Zenon Kulpa

FROM PICTURE PROCESSING TO INTERVAL DIAGRAMS

4/2003

WARSZAWA 2003

ISSN 0208-5658

Praca wpłynęła do Redakcji dnia 15 kwietnia 2003 r.

recenzent - Doc. dr hab. Wojciech Mokrzycki

Praca habilitacyjna

Instytut Podstawowych Problemów Techniki PAN
Nakład 100 egz. Ark. wyd. 15,65 Ark. druk. 19,50
Oddano do drukarni w sierpniu 2003 r.

ATOS - Poligrafia-Reklama, W-wa, ul. Jana Kazimierza 35/37

Preface

> And it is through the eyes
> that I intend to open up a new world to you.
> [Fred Hoyle, *The Black Cloud* (1957)]

The sense of sight is undoubtedly the most important sense of human beings (and many animals too) with respect to the quantity and richness of perceived information. Most of the information we receive from the world, including that communicated between human beings, is of visual nature. Hence there should be no doubt about the importance of studying this kind of information, and developing appropriate technical means—in recent times this means mostly computer hardware and software—to aid and facilitate its use.

Taking that into account, it may be surprising that despite the widespread use of visual methods in science and other areas of human activity, a serious scientific study of this representation, communication and reasoning tool has started only recently. Compare that with centuries of studying language and associated disciplines—the difference appears quite striking. Moreover, in certain areas (notably mathematics), the very use of diagrammatic representations has been (and still often is) discouraged, or even attempts were made to expel it completely. Mathematicians wrote books on geometry—undoubtedly, the most diagrammatic field of mathematics—without a single diagram in them, and were proud of that.

Fortunately, the profound importance of visual reasoning methods for the scientific work becomes recently acknowledged by philosophers of science:

> ... historians, philosophers, and sociologists of science are finally becoming aware of how much of science has been done, and increasingly is being done, using pictorial and diagrammatic modes of representation. ... It is my view that studying visual modes of representation in science provides an entrée into fundamental debates within the philosophy of science, ...
>
> [Ronald N. Giere, *Science without Laws* (1999)]

Finally, the discipline of *diagrammatics* is born, and growing numbers of researchers and publications in this area bring hope of filling yet another gap in our knowledge about means of representing and processing complex information in humans and machines. This also means the development of new tools and methods that will boost effectiveness of knowledge and information processing, necessary to meet the challenges put forward by the emerging Information Age. This work adds another segment to that broad field of study, by discussing some basic issues of diagrammatics in a novel way, and by developing a new diagrammatic notation for interval algebra and computation.

Aims of the work. The main aim of the work is the development and presentation of the novel diagrammatic notation for interval algebra and computation developed by the author, and showing its usefulness in some areas of interval algebra. Additionally, as a background for that undertaking, a state of the art survey and partially novel systematization of basic issues of diagrammatics is attempted, including a unified framework for relating various subfields and aspects of the pictorial information handling domain.

Contents of the work. The work consists of three chapters. The first one starts from general discussion of the field of using pictures as tools for information storage and communication, based on the processes of *interpretation of pictures* and *pictorial representation of knowledge*, both by humans and in computers. A three-level integrated framework combining basic types of corresponding subprocesses of interpretation and representation, and data structures used by them is proposed and then used to delineate selected research and application areas in the field. Main issues in these areas are reviewed, with this author's contributions to them outlined on that general background.

The second chapter contains an introduction to the area of *diagrammatics* that has been recently established within that field and constitutes the main subject of recent research by this author. The main issues and problems investigated within that area are presented, with special emphasis on applications in mathematics. Several contributions by this author are also included there. They span the range from novel ways of interpreting certain examples, through new proposals for treating several basic issues, to a novel general organization of the area itself.

One of the main research themes of diagrammatics is the development and investigation of diagrammatic notation systems for diverse applications, including various branches of mathematics. The latter range from rigidly formalised systems for some small subject areas, serving mostly theoretical purposes (like various recent formalizations of Venn diagrams and Euler circles), through more complex systems of more practical significance (like formal diagrammatic systems for elementary geometry), to less formal but more practical systems to be used as everyday research tools of human researchers, possibly aided by computers. Of the latter kind is the diagrammatic system for the field of *interval algebra and computation*, developed by this author.

The field of interval computations is a comparatively recent branch of pure and applied algebra, still in the stage of vigorous development. It has important applications, mostly in numerical computation domain, like the *guaranteed accuracy* computations, global optimization, and others. Albeit various simple diagrams appeared in the literature of the field, they played no significant role in its development. This stays in marked contrast to the development of complex number theory and analysis long ago, where the diagrammatic notation based on the complex plane diagram played an important role in the acceptance of complex numbers and in the development of their theory. Even today new capabilities of this notation are being discovered.

The diagrammatic system for interval algebra described in Chapter III is based on a diagrammatic representation of a space of intervals, called an *MR-diagram*. It is then applied to several subareas of interval algebra, namely *interval relations*, *interval arithmetic*, and *interval linear equations*. To meet specific needs of these subareas, additional diagrammatic tools are developed and used. It is hoped that the diagrammatic system developed might play a role in further development of interval algebra similar to that the complex plane diagram played in complex analysis research.

Bibliography. The bibliography is divided into two main parts. The first contains works authored or co-authored by this author, referred to in the text by numerical indices in brackets. The works are classified into several bibliographical categories and ordered in each category anti-chronologically (the most recent first).

The second part contains works by other authors, divided into two categories. The first contains general article collections and proceedings, referred to by [Name Year] references and ordered alphabetically by name. The second category contains the individual books and papers, referred to by [Author Year] references and ordered alphabetically by author.

Acknowledgements. The research leading to this work took many years of labour spanning many subject matters and was conducted in collaboration with many people, to whom the author owes much and feels obliged to acknowledge their help and contribution. As usual in such cases, it is practically impossible to mention all of those who possibly deserve mentioning, hence the author should ask for forgiveness for any possible omissions.

First of all, the author is greatly indebted to Prof. Michał Kleiber for creating a friendly and stimulating workplace during the last ten years, as well as for his direct collaboration with the author and encouragement of the line of research that led to this work.

The early research started within the department of pattern recognition headed by Prof. Juliusz L. Kulikowski whom I would like to thank for creating a creative and unrestrictive working environment. Thanks are also due to all members of the team there, especially Janusz Dernałowicz and Marek Chmielewski (main hardware designers of the first Polish image processing systems) and my numerous collaborators on the theoretical and software side, most of all Maria Piotrowicz, Hanna Szydło, Andrzej Bielik, Marek Doros, Henryk T. Nowicki, and Aleksander Radomski. Special thanks are due to Michał (Michael) Sobolewski for his sharing of stimulating ideas and joint work on several artificial intelligence issues, also during our stay at Concurrent Engineering Research Center in Morgantown, WV.

The author is also most thankful to Dr. Ewa Grabska, who persuaded the author to start lecturing on diagrammatics [44] which resulted in the development of much of the contents of Chapter II of this work. She also read and commented on an early draft of this work. Dr. Wojciech Mokrzycki inspired and facilitated the edition of a special issue on diagrammatics of the *Machine GRAPHICS & VISION* journal [78]. The collaboration of Ph.D. students Karol Rosłaniec and Truong Lan Le is also appreciated.

Great thanks are also due to many researchers throughout the world who shared their ideas and provided freely their publications and advice. The author would like to express his gratitude especially to Prof. Svetoslav Markov from Sofia, Prof. Jiři Rohn from Prague, and Prof. Edgar Kaucher from Karlsruhe. Thanks are also due to Nathaniel Miller, Mark Greaves, and Yuri Engelhardt, as well as many other members of the `<diagrams>`, `<reliable_computing>`, and `<infoDesignCafe>` e-mail lists for many stimulating discussions. The author would also like to thank Prof. Tristan Needham for the inspiration provided by his award-winning book [Needham 1997].

Thanks are also due to my wife Elżbieta, whose loving attention has made the burden of life so much lighter.

— Zenon Kulpa

Contents

Chapter I

Picture processing in humans and machines

Nihil est in intellectu
quod non prius fuerit in sensu.
[*A maxim of empiricist school of thought*]

The old and venerable maxim quoted above ("Nothing appears in the mind that has not appeared earlier in the senses") can be traced as far to the past as ancient Greeks of the Democritus school (5th/4th centuries B.C.), especially Epicurus (4th/3rd centuries B.C.). In its radical form ("sensual stimuli are the only substrate of thinking") it is probably highly disputable, and in fact is hotly disputed till our times [Giere 1999]. However, when understood in its more moderate form (one may say, more "physiological"), it can serve well as a summary of the main idea of this chapter. Namely, we will interpret it here in the following sense [44]:

- data from the senses constitute an important, if not the main, substrate of thinking;
- to process data during thinking, the mind makes also important use of the brain apparatus devoted originally to process (interpret) data from the senses.

The first claim is rather obvious and indisputable. The second one constitutes the kernel of the famous "imagery debate" [Kosslyn 1980, Pylyshyn 1981, Tye 1991]. Recent neurological data seem to support it rather decisively [Kosslyn 1994]. Both sentences taken together stimulate an integrated treatment of the processes of data interpretation (input), data processing (thinking) and data representation (output). For long these areas were usually treated and studied in separation, as signified by the separation, at the human studies side, of disciplines of perceptual physiology, perceptual psychology,[1] cognitive psychology, and graphic design, as well as, in computer science field, of disciplines of image processing, automated reasoning and computer graphics. An integrated treatment of these areas constitutes a basis of the emerging discipline of *diagrammatics* to which this work attempts to contribute.

[1] It is worth to mention here that visual illusions are still often called "optical illusions" despite the fact that practically all of them have nothing to do with optics of the eye, but are the results of often very high-level processing and interpretation of visual stimuli in our brains, see e.g. several examples discussed throughout this work.

The chapter starts from an exposition of general notions concerning the structure of a two-way communication between a system (whether living or artificial) and the world, with particular emphasis on and detailed exposition of specific features of visual modality. Main similarities and differences between humans and contemporary computers in this respect are also briefly outlined. The general framework proposed stresses the integration and conceptual parallelism of the opposite processes of data input (interpretation) and output (representation). Using this framework, the chapter is continued with a brief survey of several selected research areas in the field, with this author's contributions to them briefly outlined on that background.

I.1 Picture information systems

We need and want to rebuild the bridge
between perception and thinking.
[Rudolf Arnheim, *Visual Thinking* (1969)]

To organize the discussion of the field, a unified framework for classifying and relating various subfields and aspects of that large domain would be helpful. Such a framework is presented in this section, in the form of an integrated structural schema of a generic picture information system, relating basic types of corresponding processes of interpretation and representation, and data structures used by them.

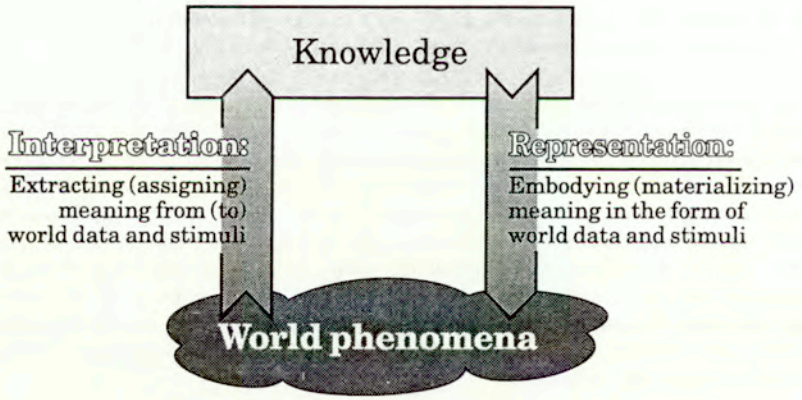


Figure I.1: Interpretation and representation:
communication between an information processing system and the world.

I.1.1 Interpretation and representation

Why, the other end of the stick always points the opposite way.
It depends whether you get hold of the stick by the right end.
[Gilbert Keith Chesterton, *The Mistake of the Machine* (1874-1936)]

Any system functioning within the real world must, in order to perform its function, be in a two-way contact with the world. That is, it must sense the surrounding world (including possibly other such systems) obtaining information from it, and it must act in the world, communicating information to it (possibly, also to other such systems). This is illustrated by the general schema shown in Fig. I.1.

The interpretation process extracts useful information from external world data and stimuli, endowing them with *meaning*, i.e., interpreting them in terms of the current system's knowledge about the world. That assignment of system-generated meaning can take a quite literal form, actually changing the sensory stimulus to the extent of adding to it new elements not present in the original physical data. It often occurs in many real-life situations, when we actually see not what is really there, but instead what we expect or want it to be. Judges examining witnesses know the effect all too well.

A neat illustration is provided by a visual illusion shown in Fig. I.2, see [Kanizsa 1974]. The arrangement of appropriately aligned black shapes stimulates our visual interpretation apparatus to recognize a white triangle in the centre. The perception of this triangle in turn influences the initial perception of the physical input to such an extent that we persistently *see* that the triangle is actually *whiter* that the surrounding area. The recognition of a triangle forces the visual apparatus to supply the missing parts of the defining element of the figure, namely its contour, producing in turn a perception of a physically nonexistent brightness difference between the apparent figure and its background. Hence the effect is also known by the name of *"illusory contours."* The effect may be also considered as one of manifestations of the emergence phenomenon discussed in Chapter II.

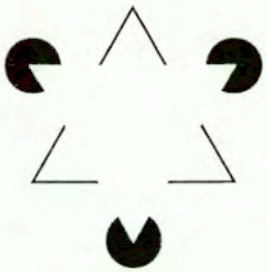


Figure I.2: Kanizsa triangle.

The representation part materializes the meaning of some piece of knowledge in the form of real world data and stimuli, either for the purpose of communicating the knowledge to some other system (possibly back to its originator as well, like during thinking aided by written notes or drawings of partial results, or when storing the data for further reference), or to produce direct physical effect in the world (like taking some meaningful action aimed at shaping some part of the world according to an internally devised design).

Despite its simplicity, the schema in Fig. I.1 leads to some more insights. First, we see that the communication system must contain at least three main subsystems: information input device (*receptor*), internal information processing and storage unit, and

information output (*effector*), see Section II.6. Second, it is obvious that the input and output subsystems cannot perform their tasks directly in a single step, thus they should consist of a whole sequence of smaller processing units using various intermediate forms of processed information. With that, the apparent parallelism of these two processing sequences, as depicted in Fig. I.1, suggests a possibility of deeper connections and similarities between them. Indeed, as indicated at the beginning of this chapter, and as recent psychophysiological findings seem to confirm (see the long raging "imagery debate" [Kosslyn 1980, Pylyshyn 1981, Tye 1991, Kosslyn 1994]), humans, when thinking visually (including visualization of information for pictorial output), apparently use to a certain extent just these parts of their brains that are otherwise used to process and interpret the input data from the eyes. The influence of the "downward" representation process on the working of the interpretation part is confirmed also by many illusory effects, like the Kanizsa contours above.

Also, the disciplines of computer image processing and computer graphics, for a long time developing independently, display since some time a marked tendency to integrate their methods and technical means, see e.g. [Pavlidis 1982]. Basic issues of that parallelism and integration of interpretation and representation tasks, and their consequences, are discussed in the next section in more detail.

I.1.2 The three-level conceptual model

This ... shows a belvedere with three floors ...:
its top and bottom are mutual contradictions.
[Maurits C. Escher, *(Lecture notes for an American tour)* (1964)]

A more detailed structure of an integrated interpretation/representation system can be rendered as in Fig. I.3, see [30, 44, 84]. It shows the main steps needed for both of these two processes, together with intermediate information structures. The schema is generic, independent on sensory modality [44], though for the purpose of this work it is specialized specifically for visual information. Note the thick "arrows" representing various subprocesses transforming information between different information structures: their thickness reflects the fact that various research and technology disciplines covering that problem area are usually defined in terms of just these processes, not in terms of information structures represented by the boxes in the diagram. With a more detailed view, the stages and processes discriminated here can be of course subdivided even further, see [48, 59, 84]. However, the level of detail considered here seems to be optimal for the purpose of understanding and classifying the basic structure of the discipline. The terms chosen to name various elements in the diagram may not always conform exactly to their usage in the field—partially due to sometimes significant terminological differences and variations in local jargons of various subfields, and partially because the short terms used have often, by their nature, larger semantic domains than the specialized usage needed in the schema. For example, the term *picture processing* in more general usage means often not only the process of transformation of the encoded picture within the system, as shown in the schema, but also the associated processes of picture coding and generation. Also, the term *recognition* is often understood to mean the whole process of interpretation, leading from a raw picture to its understanding at the knowledge level.

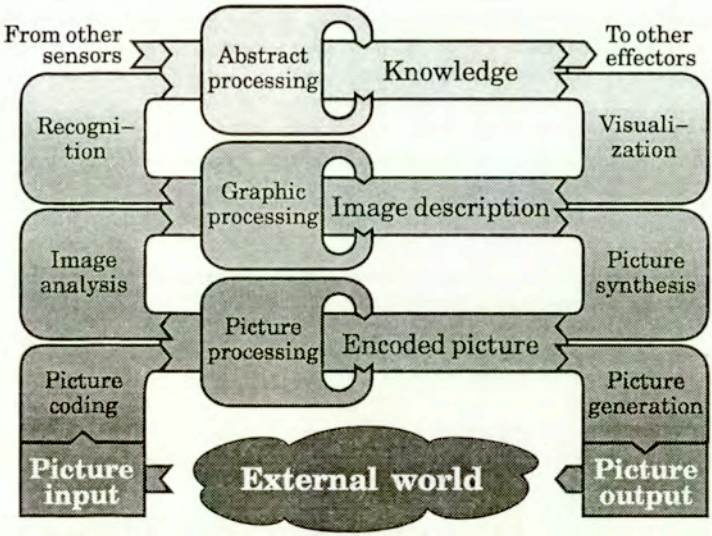


Figure I.3: The general three-level schema
of visual information interpretation and representation.

The system modules and information structures shown in the schema of Fig. I.3 can be briefly characterized as follows.

External world contains physical objects whose pictures can be "seen" by the system, as well as physical objects affected by the system, like those "painted" by the system. It also may contain other similar systems, with which the system may communicate, exchanging information through the appropriate objects in the world, like pictures.

Picture input is a device capable of taking pictures of the external world and transferring them to the inside of the system.

Picture output is a device capable of producing pictures, generated within the system, as objects in the external world.

Encoded picture is an internal form of the picture as obtained by appropriate encoding of the external world picture, or as prepared by the system for output to the world. It is generally characterized by the fact that primitive elements of the picture correspond to elementary pictorial stimuli, like brightness and colour of otherwise undifferentiated picture points.

Image description is an internal partial interpretation of the contents of the encoded picture, or the structural description of visualized knowledge, in terms of some larger elements (like edges, lines, regions, and characteristic local configurations of them), their more complex attributes (like line direction, length, thickness, figure area and shape), and relations between them (like adjacency, relative size, position), cf. Section II.2.

Knowledge is the most abstract form of information, gathered and stored by the system as a result of interpretation and processing of external stimuli. It can be used to affect the external world in various ways, among others by producing pictures representing some chunks of the knowledge. At this level all data gathered by various sense organs of the system are ultimately integrated, although various links and influences between them can occur at lower levels as well.

The processes linking the modules and transforming the appropriate information structures comprise:

Picture coding produces an internal form of the raw external picture from the data provided by the picture input device.

Picture processing transforms the encoded picture in various ways, mostly in order to enhance it by emphasizing some its aspects or filter out the data considered to be noise. This may be an end in itself, or it may be a preparatory step to facilitate further analysis of the picture, or generation of its external form. Another popular kind of processing aims at size reduction of the picture encoding for its more effective storage or transmission.

Image analysis transforms the encoded picture into a higher-level description in terms of more meaningful parts, their attributes, and relations between them. In the context of traditional pattern recognition discipline, this process is usually called *feature extraction.*

Graphic processing transforms the image description in various ways, like calculating and adding to the description new, derivative attributes and relations, preparing the image for further analysis or picture synthesis.

Recognition may be said to give meaning to the information conveyed by the image description, producing usable knowledge about objects and relations depicted in the input picture. Often the result of the recognition process consists simply of a name (of a class) of the object found in the image.

Abstract processing, that can be also termed *abstract reasoning,* transforms the system knowledge producing new items of knowledge or restructuring it for various reasons, like facilitating compact storage or knowledge retrieval.

Visualization presents selected chunks of knowledge in a visual manner, producing a structural description of an appropriate image according to some selected visual language, cf. Section II.2.

Picture synthesis takes the image description and renders it in the form suitable for generation of the output picture, possibly after some amount of final processing. Because image descriptions have often the form of parameterised, continuous geometrical primitives, while encoded pictures are usually represented as raster images, i.e. digital arrays of picture points (pixels), this process is also known as *rasterization* (sometimes the term *digitization* is also used, not very properly in this context).

Picture generation feeds the picture output device with appropriate data produced from the internal encoded picture.

I.1.3 Humans versus machines

"I mean Man," said Father Brown,
"the most unreliable machine I know of."
[Gilbert Keith Chesterton, *The Mistake of the Machine* (1874-1936)]

Before going further, it would be instructive to compare, within the above framework, certain basic features of humans and contemporary computer systems in handling of visual information.

As human beings, we are endowed with a very efficient and sophisticated apparatus for picture interpretation, especially concerning its versatility and ability to recognize tremendous numbers of different objects, usually obscured by complex contexts of other objects and high levels of various kinds of noise. The situation is quite different as concerns our pictorial representation abilities. Unfortunately, the Nature did not endow us with a *picture effector* of comparable throughput, efficiency, and sophistication as our receptor devices. That striking asymmetry[2] has very serious consequences, though rarely fully recognised. Some of the consequences will be discussed further in this work, see especially Sections II.3.1.1 and II.5.1.3.

The situation is markedly different with computer systems. The advances of computer graphics makes them able to produce tremendous amounts of complex pictures, already allowing simulation in real time of quite sophisticated and realistic three-dimensional virtual realities. On the other hand, analysis and recognition of complex images by computers is still much inferior as compared with human abilities. Computer recognition systems are far less general, require comparatively simple and clean pictures without much noise, and exhibit high error rates. This leads, among others, to problems with computer implementation of diagrammatic systems, as discussed in Section II.6.

Comparing both kinds of systems, we see that humans and computers are complementary concerning their abilities to handle pictorial information. Therefore, hybrid man-machine systems, combining effectively the strengths of humans and machines while mutually compensating for their respective weaknesses, seem to be the best solutions in graphically intensive applications.

I.1.4 From picture processing to diagrammatics

... human cognition [is] a unitary process,
which leads without break from the elementary acquisition
of sensory information to the most general theoretical ideas.
[Rudolf Arnheim, *Visual Thinking* (1969)]

Development of computer picture processing systems led to the development of many new scientific and technical disciplines, aimed at investigation of methods of automatic processing of pictorial information and developing their various practical applications. Many of these disciplines followed directly the divisions illustrated in the three-level schema of Fig. I.3, while others were structured according to other criteria (e.g., following the di-

[2]The reasons for that asymmetry, dictated by the course of biological evolution, are rather obvious and will not be discussed here.

visions of various application areas). Certain selected problem areas are surveyed in the rest of this chapter.

The survey starts, in Section I.2, from the lower part of the schema in Fig. I.3 which constituted most of the early systems. Since, until quite recently, the evolution of the whole field was practically determined by the evolution of the image input and output devices, the exposition focuses mostly on them. The section ends with a historical note on the first Polish picture processing systems, in particular these in whose development this author took an active part. Next, in Section I.3, basic notions of a theory of discrete pictures and operations on them are presented, as they constitute a necessary basis for discussion of most other areas here, including one of possible techniques for implementing diagrammatic reasoning, as described in Section II.6.2.1. Basic issues of discrete image analysis, i.e., methods of producing a higher-level description of the contents of the picture on the basis of its unstructured raster image, are introduced in Section I.4. The discipline of scene interpretation and understanding, whose main concern is the problem of three-dimensional interpretation of pictures of the real three-dimensional world, is introduced in Section I.5. The phenomenon of so-called *"impossible figures"* is discussed there in more detail, as it especially conspicuously reveals some basic properties and mechanisms of spatial interpretation of pictures, important for understanding of both human and computer vision. Finally, in Section I.6, the field of diagrammatics is briefly introduced as well. The whole next Chapter II is devoted to its more detailed treatment.

The survey of the field of picture processing is far from complete here. The selection of the few covered issues has been made on the basis of three criteria: the need to provide a possibly brief but general outline of the field, the demands of the exposition of the field of diagrammatics contained in the next chapter, and, in part, the fact of the author's active work in the chosen areas. A recent survey of major current directions and problems can be found in [81].

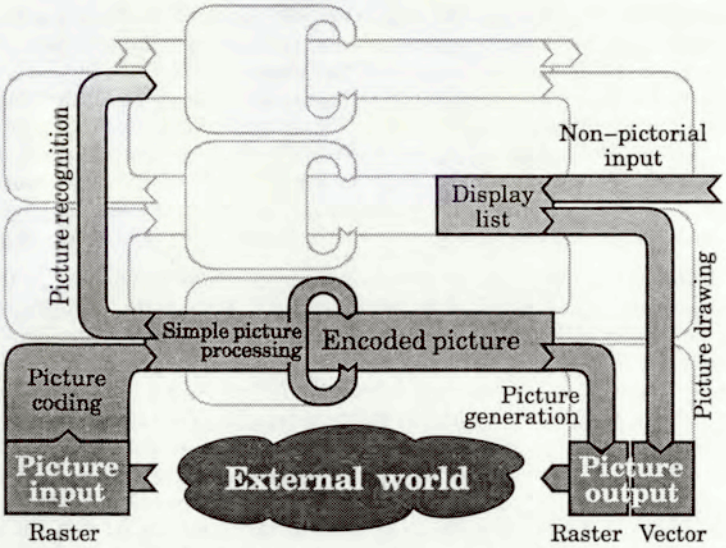


Figure I.4: Early (raster) picture processing (left and bottom part), and (vector) computer graphics (right part).

I.2 Computer picture processing systems

"So it does!" said Pooh. "It goes in!"
"So it does!" said Piglet. "And it comes out!"
[Alan A. Milne, *Winnie-the-Pooh* (1926)]

The development of computer systems for handling pictorial information started from constructing devices for pictorial input and output with rather rudimentary processing of pictorial information. The *picture processing* systems were mostly restricted to the lowest level of the schema in Fig. I.3—the external picture was encoded, subjected to some simple processing like local noise filtering or contrast enhancement, and then displayed or output into a hardcopy of some kind. Also, attempts at perceptron-based [Minsky & Papert 1969] pattern recognition at the level of the encoded picture started early. On the other hand, in the field of *computer graphics*, the main emphasis was on the generation of linear drawings according to geometrical specification. Early systems started from the second, image description level (created in most cases directly by a human user, often in a non-pictorial way) and drawn the final image in a single sweep, omitting the encoded picture level, as shown symbolically in Fig. I.4. It is therefore not surprising that the disciplines of picture processing and computer graphics used quite different graphical devices and different philosophies of using visual information, and in consequence were treated as almost completely separated and independent. That was exemplified by discerning two separate types of graphical techniques, namely:

Raster graphics. Here the external image is sampled according to a fixed, regular *raster* of points arranged in an array (usually rectangular), and the encoded picture is therefore a two-dimensional array of picture elements (called *pixels*), characterized by their positions (two-dimensional coordinates) in the image and their values, usually brightness (or gray level), or colour. The picture input device samples the points in a fixed sequence independent of the contents of the picture and digitizes their brightness or colour value into a fixed number of discrete levels. The output device directly displays the pixel array point by point, usually in the same fixed sequence as the input device, in a fixed time also independent of the contents of the picture.

Vector graphics. In this technique, the image to be produced is described in terms of geometric primitives, mostly straight line segments (hence *vector graphics*), defined by coordinates of their endpoints and arranged in the memory into a sequential display list. The output device takes the list as the input and directly draws the segments one by one, in the order defined by the list. As a result, no intermediate encoded image is internally produced. The generation of the image depends on its contents—elements further on the list may overwrite the earlier ones, and the time of producing the whole image depends on the number of primitives. The picture is defined usually with non-pictorial means, either as a result of some computer calculations, or provided by the user in textual form, sometimes with the help of appropriate picture generation language (like the simple language provided by the popular TEX typesetting system, still used sometimes for simpler figures, like that in Section II.1.2.1 of this work).

With raster graphics one may process and produce the realistic images with smooth transitions of brightness and colour, while classic vector graphics was restricted mostly to line drawings (called also *wire-frame* pictures), due to limitations of the output devices. Raster graphics started early to use the inherently raster television technique which made them simpler to construct and cheaper, in contrast to vector graphics displays with their dedicated hardware needed to control deflection voltages of a cathode-ray tube directly from the data in the display list. Raster input devices were usually of two kinds: television cameras for general purpose, though not very accurate input, while electromechanical devices—ancestors of present-day scanners—were used where precision and parameter stability were important. Hardcopy output was for a considerable time a weak spot of raster devices, and was for long dominated by cumbersome photographic devices until comparatively recent development of high resolution colour printers and computer-controlled imagesetters. Vector hardcopy was much easier with pen plotters, capable of precise large format drawing in colour from early times.

Finally, though, raster graphics won the battle at the input/output device level, so that vector graphics, when needed, is now usually simulated on raster devices. Namely, the single process of picture drawing, previously done in hardware, splits into two processes: a software algorithm synthesizing internally a raster-encoded picture from the vector description (called often a *rasterization* or *scan conversion* algorithm), and a hardware raster device producing the final picture from that encoded picture. As a result, the restriction to line segment elements has been basically removed, and with wider repertoire of graphic primitives, including curves and various shapes filled with solid colour, the vector graphics changed into a so-called *surface graphics* or *object graphics*, although the term *vector graphics* is still widely used. On the other hand, because picture recognition

starting from raw raster images did not give good results (cf. [Minsky & Papert 1969] for analysis of some reasons for that), the approach called *structural pattern recognition* has been introduced. With it, a single interpretation (picture recognition) process was split into the *image analysis* phase and the proper *recognition* phase. The image analysis (called also *feature extraction*) determines the structural description of the image in terms of detected primitive graphical objects, their various local attributes and structural relations between them. The recognition phase recognizes the meaning (contents) of the image on the basis of the structure of the image description.

With that integration of raster and vector devices at the low level, and the refinement of the processes used in both disciplines, the full three-level schema of Fig. I.3 emerged and the two approaches started finally to integrate.

From the very beginning it was realized that the homogeneous structure of raster images and the prevalent kind of their processing is especially suited for parallel processing, where the same local operation is performed in parallel for neighbourhoods of all pixels on the raster. It was already known that visual systems of humans and animals use this principle for early processing in the eye's retina and in lower levels of brain visual centres. The first embodiment of the idea appeared in the ILLIAC III computer [McCormick 1963]. It contained a square array of 32×32 logical processors corresponding to pixels of a binary image of this size, and operating in parallel, taking as inputs the immediate neighbours of every pixel in the array. The resulting class of local parallel operations was then, and is till now, extensively studied and used in picture processing, see Section I.3.2.

In standard, serial computers these local parallel operations were often implemented in a way simulating to some extent the physical parallelism of execution, gaining in this way much on the computation speed. These so-called *semi-parallel* implementations were based on the fact that a single computer instruction, say of some logical operation, works in parallel on all bits of the computer word (and the word-oriented computers then in use possessed rather long words, of the order of 32 bits). Thus, if (binary) pixels of the raster image were not placed one to a whole word, but packed into its individual bits, many pixels can be processed in parallel by a single machine instruction. The principle was often used [33, 92, 93], assuring fast and efficient picture processing on rather slow serial computers of the early times of image processing. The technique was extended to multilevel images as well. For that, they were represented as stacks of binary images, such that the i-th image of the stack contained the i-th bits of all image pixels [93]. The binary images in the stack were then packed into bits of machine words as explained above. An access to individual pixel values is troublesome in this representation, but homogeneous operations, like adding two pictures pixel-wise, could be performed very efficiently with this data structure.

I.2.1 Early computer image processing systems in Poland

It seems worth to include in this place a historical note on the early development of computer image processing field in Poland. The first two Polish computer image processing systems were developed in the early seventies, in the department of pattern recognition headed by J.L. Kulikowski, [ICS Report 1972]. Their parameters and capabilities placed them in the mainstream of similar designs constructed thorough Europe in that period [32, 87, 92]. Main design of the hardware part was due to J. Dernałowicz and

M. Chmielewski [Dernałowicz 1972, Dernałowicz et al. 1977, 38, 73, 88], with some contributions from this author too, while the development of the software support was due in main part to this author. The CESARO series of systems was designed by another group in Cracow, headed by R. Tadeusiewicz, during late seventies and eighties.

The CPO-1 system. The first of these systems, named CPO-1 (*Cyfrowy Przetwornik Obrazu*: Digital Picture Transducer) was constructed using transistor technology, see [Dernałowicz 1972]. A black and white analog input from a modified industrial TV camera was quantized to obtain a raster image of 128×128 two-bit pixels (i.e., 4 gray levels). The digitized image was stored in the frame buffer memory of the device and displayed on one of its two TV displays. The frame buffer could be accessed from a Polish minicomputer of that period, called ODRA 1204 (executing about 40 thousand machine instructions per second on 24 bit words), for processing and analysis. The pictorial results could be send back to the frame buffer for inspection on the screen.[3]

The software support developed in most part by this author consisted of heavily reprogrammed operating system of the minicomputer, an appropriate modifications made to the vendor-provided compiler (called ALGOL 1204) of the ALGOL-60 high level language, and a picture processing software package PICTURE ALGOL 1204 [42, 98]. The functionality of the package was modelled upon the software developed for the ILLIAC III parallel image processing machine [McCormick 1963, Narasimhan 1964]. The main group of operations in the package worked on binary images. The package was later extended by S. Tyszko from another group of users to handle multi-valued images using a so-called *packed coding* scheme. A list-processing extension to the package was also experimentally implemented (by this author and A. Bielik).

The ALGOL procedures of the package made extensive use of the possibility offered by the ALGOL 1204 translator to insert machine code instructions within an ALGOL program. This feature allowed for implementation of the *semi-parallel* mode of image processing, leading to very significant improvements in processing time, necessary for conducting any practical image processing with the slow computers of those times.

Several basic research projects and experimental applications were conducted using this system, among others [22, 99].

The CPO-2 system. The second system, named CPO-2, operational since 1974, was built with solid state technology and offered much better image parameters, [32, 67, 80, 87, 88]. The input image contained 512×512 four-bit pixels (16 gray levels). Quantization thresholds were controlled by hand or from the computer. The input analog or digitized images from the camera, or the digitized image from the frame buffer, could be observed on a black and white display, while the contents of the frame buffer could be also seen on a colour display[4] using a pseudo-colour look-up table controlled from the computer (with 16 intensity levels available for every R, G, and B colour component). The system contained

[3]The younger readers may not realize that at those times the only visual interface with a computer consisted of a teleprinter or at most an alphanumeric "green on black" monitor (switch on to a DOS screen on your PC to see how it approximately looked like). Hence, one could not use the computer monitor screen to display input and output raster images. The display must have been custom-build as a part of the image device instead.

[4]See the previous footnote.

also a joystick manipulator controlling a cursor on the screen. The device was connected to a small, but quite powerful for those times, Polish experimental minicomputer K-202 designed by J. Karpiński.

Because the K-202 computer did not yet at that time offer a compiler for an appropriate high-level language, the software support for the system was written in an assembly language. Besides small additions to the operating system, it comprised a large library of image processing subroutines called PICASSO (abbreviated from *PICture ASSembly-programmed Operations*) [33, 94], extensively using the semi-parallel implementation approach [93] for efficiency, and a series of simple interpreted languages collectively called PICASSO-SHOW [33, 37, 67, 88, 95]. The PICASSO library was developed by a team of programmers, including, and supervised by, this author. The PICASSO-SHOW language was developed by this author and implemented in most part by H.T. Nowicki [37, 88, 95]. In implementation of further versions of the language took part A. Radomski, who also implemented the final version in the series, renamed LIP (*Language for Image Processing*), which included, among others, the Pascal-like structural programming constructs and list processing operations (the paper [33] describes briefly a preliminary proposal for that extension, named there PICASSO-SHOW 3).

As a final element of the software support, the PAL (Picture Analysis Language) implementation was started. The PAL language, with a general structure combining some features of PL/1 and Algol 68 languages, was first proposed in [41] and further detailed in [33, 50, 88]. In the thesis [109] the systematic design of the language was conducted, based on comprehensive analysis and survey of basic methods and requirements of picture processing and analysis, leading to a final formal definition of the language. However, the language has never been fully implemented and used.

Besides image processing and analysis [34–36, 38, 62, 63, 68, 88], the system was used for computer graphics tasks as well, e.g. [17, 18, 21].

The Cesaro systems. These systems were constructed and tested in various applications by a group of researchers, headed by R. Tadeusiewicz, working in the Cracow Academy of Mining and Metallurgy. The first system of the series has been conceived around 1977, see [Tadeusiewicz 1977], but became fully operational in 1982, see [Tadeusiewicz 1985]. The TV-camera input was quantized into a digital raster image of 128×128 pixels with 16 gray levels. An experimental microcomputer PRS-4 served as the processing unit. The system was used in many image recognition applications, mostly in medicine and mineral industry [Tadeusiewicz 1977, Kordek et al. 1980].

The second system of the series, named CESARO 2, became fully operational in 1990. It used up to three TV cameras and operated on the gray scale image of 256×256 pixels with 8 bits per pixel. The image was processed and analyzed by a custom-made multiprocessor system build with industrial CAMAC technology, and its main applications were in vision systems of industrial robots, see [Tadeusiewicz 1992].

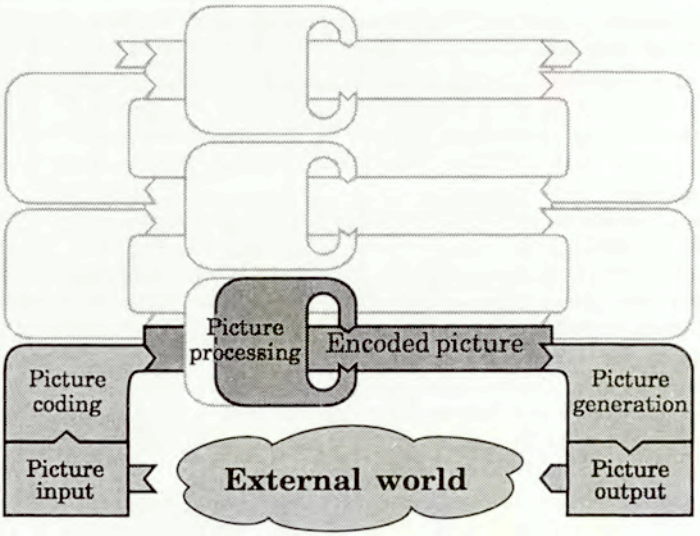


Figure I.5: The basic raster-level picture processing.

I.3 Discrete picture processing

Basic discrete picture processing at the level of encoded pictures, see Fig. I.5, has been investigated and formalized early, as a discrete two-dimensional extension of standard signal processing formalism [Rosenfeld 1969, Granlund & Knutsson 1996]. The early treatments were rather eclectic; a more uniform formulation (after [23, 109], modified and extended) is outlined here. Other models of the raster-level processing are possible, as indicated in Fig. I.5 by the dark gray regions filling only a part of the encoded picture and processing boxes. The raster level processing is closely related to the use and properties of the input and output devices, as marked by light gray fill in the figure. This level of processing often constitutes a whole self-contained application, in the cases when the task is to produce a filtered, enhanced, or in another way processed version of the input picture, like in preprocessing raw pictures from remote sensing devices (say, in space research), or preparing them for quality printing. Such applications do not involve any structural analysis or recognition of pictures, or synthesizing them from non-pictorial data.

A basic notation for binary relations follows (see Section III.3 for more). Let X, Y, Z, W be sets. Then $\diamondsuit \subseteq X \times Y$ and $\heartsuit \subseteq Y \times Z$ are (*binary*) *relations*; $x \diamondsuit y$ means $(x, y) \in \diamondsuit$. The notation $\diamondsuit \circ \heartsuit$ (shortly: $\diamondsuit\heartsuit$) is used for a *composition* of relations: $x \diamondsuit\heartsuit z \iff (\exists y) x \diamondsuit y$ **and** $y \heartsuit z$. A relation $\diamondsuit^{-1}$ is an *inverse* of $\diamondsuit$ when $y \diamondsuit^{-1} x \iff x \diamondsuit y$. The notation $W\diamondsuit$ denotes an *image* of the set W under the relation $\diamondsuit$, namely: $W\diamondsuit = \{y \in Y \mid w \in W$ **and** $w \diamondsuit y\}$. Similarly, $\diamondsuit W$ denotes a *coimage* of W under $\diamondsuit$. For every set X, the symbol $\mathbf{I}_X = \{(x, x) \mid x \in X\}$ will denote the *identity* (or *diagonal*) *relation* in X. Therefore, $\mathbf{I}_W\diamondsuit$ is a *left restriction* and $\diamondsuit\mathbf{I}_W$ is a *right restriction* of $\diamondsuit$ to the set W. As functions are also relations, this notation will be used for them also.

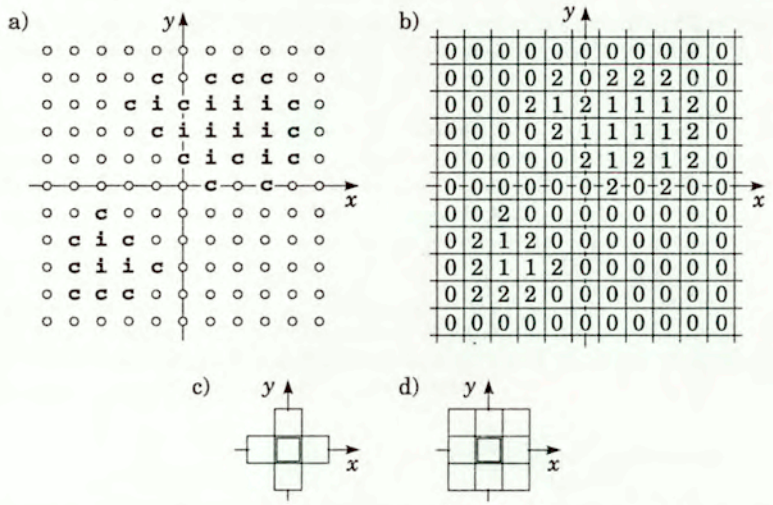


Figure I.6: The raster (shown in point-oriented (a) and cell-oriented (b) rendering) and basic neighbourhoods: 4-adjacent (c), and 8-adjacent (d), together with example pictures: over an abstract alphabet {c, i} (a), and using numerical values with $v_{\max} = 2$ (b).

I.3.1 Discrete pictures

Let $\mathbb{J}$ denotes a set of (all) integer numbers. The set $U = \mathbb{J} \times \mathbb{J}$ is called a *raster*, while every pair $(i, j) \in U$ is called a *raster point* (or simply a *point*), see Fig. I.6. Two points (a, b) and (i, j) are called 4-adjacent (8-adjacent) when $|a - i| + |b - j| \leq 1$ (respectively, $|a - i| \leq 1$ **and** $|b - j| \leq 1$). The names of these two types of adjacency come from the number of points that are adjacent to any given point under the appropriate definition. A subset $V \subseteq U$ is called 4-connected (8-connected) when every pair of its points can be connected by a sequence of consecutively 4-adjacent (8-adjacent) points, all of them belonging to V.[5]

A bounded subset $N \subset U$ is called a *neighbourhood*. The two basic neighbourhoods most often used in picture processing are illustrated in Fig. I.6. They consist of all raster points 4-adjacent (Fig. I.6c) or 8-adjacent (Fig. I.6d) to the central one with coordinates $(0, 0)$ (marked by a double contour in the figure).

A function $d_{ij} : U \to U$ such that[6] $d_{ij}(x, y) = (x + i, y + j)$ is called in the sequel a *displacement*. For any $V \subseteq U$, we have $d_{ij}(V) = d^{-1}_{-i,-j}(V) = V d_{ij} = d_{-i,-j} V$. Let A be some set (finite, if not stated otherwise), called an *alphabet*. By $\circ \notin A$ we will denote a *blank symbol*; moreover, $A_\circ = A \cup \{\circ\}$.

[5]Other raster structures are sometimes considered, like *hexagonal* (honeycomb) raster, or *nonuniform circular* raster. The latter was used by [Funt 1980] in the WHISPER system introduced in Section II.6.2.1.

[6]For functions f with the domain U the term $f((x, y))$ will be abbreviated to $f(x, y)$.

Definition I.1 (Discrete picture) *A* discrete picture *(shortly a* picture*) is any function $P : U \to A_o$ for which PA is a bounded (here finite) set of points. A set of all pictures will be denoted by $\Pi(A)$, or simply Π when A is known (or irrelevant).*

An example picture using the alphabet {c, i} is shown in Fig. I.6a. A picture P is 4-connected (8-connected) when the set PA is 4-connected (8-connected). A *subpicture* is a function $p_N = \mathbf{I}_N P$, where N is a neighbourhood, and P a picture. The set of all subpictures over A is denoted here by $\pi(A)$, or briefly π. The symbol $\pi_N(A)$ denotes the set of all subpictures over A defined over the common neighbourhood N.

For some $P \in \Pi(A)$ and $s \in A$, when $P_o\{s\} \neq \emptyset$ it is said that the symbol s *occurs* in P. A *blank picture* is a picture Q for which $UQ = \{o\}$, i.e., in which no other symbol occurs except a blank symbol. A point (x, y) together with its value in some picture P, i.e., formally a pair $((x, y), P(x, y))$, is commonly called a *pixel* (of the picture P). Pictures can be thus conveniently defined also as (finite) sets of their non-blank pixels.

I.3.2 Picture processing operations

Definition I.2 (Picture processing operation) *A* picture processing operation *is any function $\varphi : \Pi \to \Pi$. A picture processing operation $\sigma_{ij} = \{(P, d_{ij}P) \mid P \in \Pi\}$, where d_{ij} is some displacement, is called a* shifting operation.

The shifting operation σ_{ij} simply moves the value of every point (x, y) of the argument picture to the point $(x - i, y - j)$, so that for every $(a, b) \in U$ we have $P(a + i, b + j) = [\sigma_{ij}(P)](a, b)$.

Definition I.3 (Position-invariant operation) *A picture processing operation ψ is called* position-invariant *if for every shifting operation σ_{ij} we have $\psi\sigma_{ij} = \sigma_{ij}\psi$.*

Definition I.4 (Local parallel operation) *A* local parallel picture processing operation *(or (*local*)* parallel operation *for short) is such a picture processing operation λ for which there exists a neighbourhood N and the function $f_\lambda : \pi_N(A) \to A_o$ such that for every $P \in \Pi$ we have $[\lambda(P)](a, b) = f_\lambda(\mathbf{I}_N d_{ab} P)$.*

In other words, the result of the local parallel operation for any given point of the raster depends only on the values of points in some neighbourhood of that point in the argument picture, the neighbourhood being the same (except for shifting) for all points. Such an operation can be specified by specifying the function f_λ, consisting of a finite set of pairs $(p_N(A), s)$ with $p_N(A) \in \pi_N(A)$ and $s \in A_o$. Execution of the operation can then be accomplished by applying the function f_λ independently at every point of the raster, so that it can be done in parallel, i.e., for all the points simultaneously. It is easy to see that every local parallel operation is position-invariant, but *not* the other way round.

The shifting operation σ_{ij} is also a local parallel operation, with $N = \{(i, j)\}$ and $f_{\sigma_{ij}}(p_N) = p_N(i, j)$.

Definition I.5 (Point operation) *A* point operation *is such a local parallel operation for which $N = \{(0, 0)\}$, i.e., the result of the operation for any given point depends on the value of only this same point in the argument picture.*

Generalization of the above to n-argument picture operations of the form $\varphi : \Pi^n \to \Pi$ is straightforward. It can be done, e.g., by redefining them as single-argument operations using an extended alphabet $A^n_\square$, consisting of ordered n-tuples of symbols from the original alphabet. All the discussion above generalizes easily to so defined n-argument operations.

Example I.1 (Parallel contour extraction) A simple example of a useful parallel operation is the contour- and interior-extraction operation γ presented in Fig. I.7. In the example, the alphabet A contains three symbols $\{1, c, i\}$ and the neighbourhood is the basic 4-adjacent one from Fig. I.6c. The function f_γ is defined graphically in Fig. I.7a. An empty cell means here that any symbol can occur in the cell, while a symbol set name in the cell means that any symbol from that set can occur in it. Due to occurrence of symbol set names, diagrams containing them define a whole family of conditions, one for every combination of symbols taken from the specified set(s). The resulting picture is

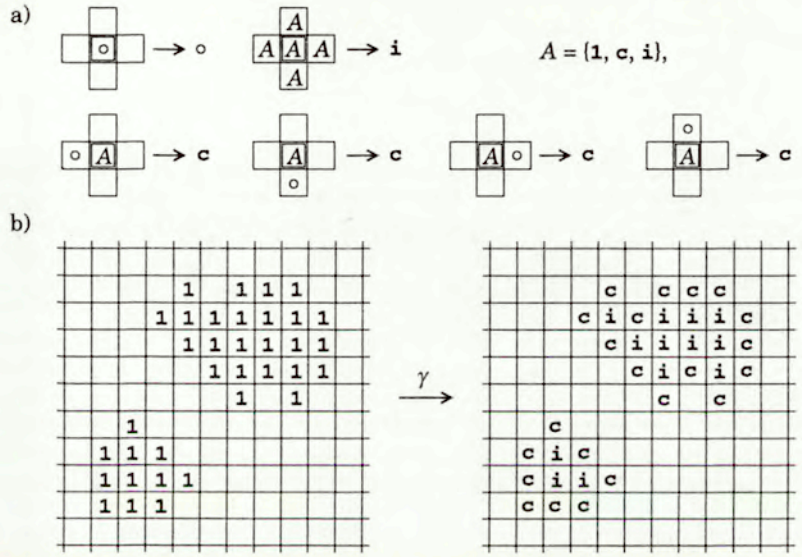


Figure I.7: The parallel contour extraction operation: graphical specification of the f_γ function (a), and the result of the operation for some example picture (b).

the same as in Fig. I.6a, with symbols c denoting contour points, and symbols i interior points of the regions marked by 1-symbols in the input picture. Points with the blank symbol value are left empty (differently than in the specification of the f_γ function). ■

Theorem I.1 (Realisation of parallel operations) *Every parallel operation λ can be realized as a superposition of n shifting operations and an n-argument point operation, where $n =$* **card** *N and N is the neighbourhood used by the operation λ.*

Proof. The theorem is quite obvious, nevertheless it may be instructive to derive it more formally. Let $N = \{(i_1, j_1), (i_2, j_2), \ldots, (i_n, j_n)\}$. For every $P \in \Pi$ and $(a, b) \in U$, from Definition I.4 it follows:

$$\begin{aligned}[\lambda(P)](a,b) &= f_\lambda(\{((i_k,j_k),P(a+i_k,b+j_k))\,|\,1\le k\le n\}) = \\ &= h_\lambda(P(a+i_1,b+j_1),P(a+i_2,b+j_2),\ldots,P(a+i_n,b+j_n)),\end{aligned}$$

where $h_\lambda : A^n \to A$ is some n-argument function over A. Let us now construct an n-argument point operation φ, defined by its neighbourhood function f_φ as follows:

$$f_\varphi(p_1,p_2,\ldots,p_n) = h_\lambda(p_1(0,0),p_2(0,0),\ldots,p_n(0,0)),$$

where $p_k = \{((0,0),s_k)\}$; $s_k \in A_0$. Then from Definition I.4:

$$[\varphi(P_1,P_2,\ldots,P_n)](a,b) = h_\lambda(P_1(a,b),P_2(a,b),\ldots,P_n(a,b)),$$

where $P_1,P_2,\ldots,P_n \in \Pi$. Now substituting $P_k = \sigma_{i_kj_k}(P)$ (where $\sigma_{i_kj_k}$ is the shifting operation) and taking into account that $P(a+i_k,b+j_k) = [\sigma_{i_kj_k}(P)](a,b)$ we have:

$$\begin{aligned}&[\varphi(\sigma_{i_1j_1}(P),\sigma_{i_2j_2}(P),\ldots,\sigma_{i_nj_n}(P))](a,b) = \\ &= h_\lambda([\sigma_{i_1j_1}(P)](a,b),[\sigma_{i_2j_2}(P)](a,b),\ldots,[\sigma_{i_nj_n}(P)](a,b)) = \\ &= h_\lambda(P(a+i_1,b+j_2),P(a+i_2,b+j_2),\ldots,P(a+i_n,b+j_n)) = \\ &= [\lambda(P)](a,b),\end{aligned}$$

as required. □

The above theorem serves as a basis for implementation of local parallel picture operations on ordinary serial computers. According to the theorem, it suffices to implement an appropriate set of point operations on pictures (possibly using the semi-parallel implementation technique described in Section I.2), and an universal operation of shifting the picture by any given distance (i,j), in order to be able to program any local picture processing operation.

The number of points in the neighbourhood N and its extent—the distance of its points from the point $(0,0)$—may serve to some extent as a measure of complexity of the parallel operation defined over this neighbourhood. Operations using large neighbourhoods provide larger "productivity" in a single step, but are usually more cumbersome in implementation. However, similarly to Theorem I.1, it has been shown by [Kruse 1973] that any local parallel operation can be realized as a sequence of local operations using the basic 8-adjacent neighbourhood (see Fig. I.6d). That fact is extensively exploited in picture processing systems and hence that neighbourhood (together with its subset, the 4-adjacent neighbourhood, Fig. I.6c) are the most common and are considered basic in implementations and applications.

I.3.3 Number-valued pictures

In practical picture processing, usually the values of points of the input picture are not symbols from some abstract alphabet A. Instead, the values represent the (discrete) brightness levels of appropriate points, see Section I.2. In such cases the set $\mathbb{N}$ of natural numbers can be used as an alphabet, with the number 0 playing the role of the blank symbol. Moreover, in practice only some finite subset $\{0,1,\ldots,v_{\max}\} \subseteq \mathbb{N}$ is used, which conforms with the usual assumption that the alphabet is finite. When discussing operations on such pictures, it is implicitly assumed that $v_{\max}$ is big enough so that the results of the discussed operations do not exceed $v_{\max}$ (or do not become negative). Sometimes, a special rules (e.g., thresholding or a modulo $v_{\max}+1$ arithmetic) are applied to

enforce that requirement. Analogically, sometimes also pictures with values taken from the set $\mathbb{J}$ of integers (i.e., containing also negative numbers) are considered. The class of number-valued pictures will be denoted by $\Pi(\mathbb{N})$ or $\Pi(\mathbb{J})$, depending on the allowed set of values. An example number-valued picture with $v_{\max} = 2$ is shown in Fig. I.6b.

An important special case is represented by *binary pictures*, for which $v_{\max} = 1$, i.e., $A = \{1\}$, with 0 playing the role of the blank symbol, $A_0 = A_0 = \{0, 1\}$. It is sometimes convenient to interpret binary pictures as simply subsets of U. For some ordinary picture B over the binary alphabet $A_0 = \{0, 1\}$, a binary picture in that sense is defined as $B_U = \{(i, j) \mid B(i, j) = 1\} \subseteq U$. The whole raster U is in this case called a *universal picture*, although formally it cannot be interpreted as a picture in the sense of Definition I.1, because it contains an unbounded set of points with non-blank values.

In the class of pictures with numerical values of their points, we distinguish a *unit picture* J_{ij} defined by the condition $J_{ij}(i, j) = 1$ **and** $(\forall (a, b) \neq (i, j)) J_{ij}(a, b) = 0$. The picture J_{00} will be denoted shortly as J. Obviously, the equality $J = d_{ij} J_{ij}$ holds here.

I.3.4 Operations on number-valued pictures

For number-valued pictures it is useful to extend arithmetical operations to picture arguments. Namely, the operation $P_1 \diamond P_2$ for $P_1, P_2 \in \Pi(\mathbb{J})$ and $\diamond \in \{+, -, \cdot, /\}$ is a two-argument point operation with its defining function $f_\diamond$ given as:

$$f_\diamond(p_1, p_2) = p_1(0, 0) \diamond p_2(0, 0),$$

where p_k are subpictures over the neighbourhood $\{(0, 0)\}$, see Definitions I.4 and I.5. In a similar way one may define operations with one argument being a number and the other a picture. For an operation $n \diamond P$; $n \in \mathbb{J}$, the defining function will be obviously:

$$f_{\mathbb{J}\diamond}(p) = n \diamond p(0, 0).$$

Other algebraic notation and operations (e.g., max, min, etc.) can be extended to arithmetic on pictures analogously.

Definition I.6 (Linear homogeneous operation) *A* linear homogeneous parallel picture operation *(*linear operation *for short) is a picture operation φ for which the equality*

$$\varphi(n_1 P_1 + n_2 P_2) = n_1 \varphi(P_1) + n_2 \varphi(P_2)$$

holds for every $n_1, n_2 \in \mathbb{J}$ and $P_1, P_2 \in \Pi(\mathbb{J})$.

Theorem I.2 (Weighted-average operations) *Every parallel operation φ on number-valued pictures with the defining function f_φ of the form:*

$$f_\varphi(p) = \sum_{(i,j) \in N} a_{ij} p(i, j), \quad \text{with } a_{ij} \in \mathbb{J} \tag{I.1}$$

is linear homogeneous. *In particular, the point operation ψ with $f_\psi(p) = n\, p(0, 0)$ is linear homogeneous. Operations of this form ale called* (local) weighted-average operations.

Proof. For every point $(a,b) \in U$, from Definition I.4 we have:

$$\begin{aligned}
[\varphi(n_1P_1 + n_2P_2)](a,b) &= f_\varphi(\mathbf{I}_N d_{ab} \circ (n_1P_1 + n_2P_2)) = \\
&= \sum a_{ij}[n_1P_1 + n_2P_2](i+a, j+b) = \\
&= \sum a_{ij}(n_1P_1(i+a, j+b) + n_2P_2(i+a, j+b)) = \\
&= n_1 \cdot \sum a_{ij}P_1(i+a, j+b) + n_2 \cdot \sum a_{ij}P_2(i+a, j+b) = \\
&= n_1 f_\varphi(\mathbf{I}_N d_{ab} P_1) + n_2 f_\varphi(\mathbf{I}_N d_{ab} P_2) = \\
&= n_1[\varphi(P_1)](a,b) + n_2[\varphi(P_2)](a,b),
\end{aligned}$$

as required (all sums in the above are of course over $(i,j) \in N$, as in equation (I.1)). □

The weighted average operations are the most often used operations in early stages of picture processing. Many useful operations belong to this class, like local averaging (low-pass filtering), contour sharpening, "salt and pepper" noise filtering, calculation of local spatial derivatives and gradient, and extraction of various local features in pictures, see e.g. [Rosenfeld 1969, Pratt 1978].

A simple non-local function of number-valued pictures is of much practical importance, see Section I.4. It is called *weight*, see e.g. [Narasimhan 1964], [33, 42], and is defined as:

$$w(P) = \sum_{(x,y)\in U} P(x,y). \tag{I.2}$$

Theorem I.3 (Impulse dispersion picture) *Every picture $P \in \Pi(\mathbb{J})$ can be represented in the form:*

$$P = \sum_{(i,j)\in U} P(i,j)J_{ij} = \sum_{(i,j)\in U} P(i,j)\sigma_{-i,-j}(J).$$

Moreover, if φ is a linear operation, then for every $P \in \Pi(\mathbb{J})$:

$$\varphi(P) = \sum_{(i,j)\in U} P(i,j)\varphi(\sigma_{-i,-j}(J)) = \sum_{(i,j)\in U} P(i,j)\sigma_{-i,-j}(\varphi(J)).$$

That is, the effect of the linear operation φ for any picture P is fully determined by its effect on the unit picture J. The picture $\varphi(J)$ (denoted also as J_φ) is called an impulse dispersion response *(in this context sometimes also* impulse dispersion picture*) of the linear operation φ.*

Proof. Immediate. □

Further on, we have:

$$\begin{aligned}
[\varphi(P)](a,b) &= \sum_U P(i,j)[\sigma_{-i,-j}(J_\varphi)](a,b) = \\
&= \sum_U P(i,j)J_\varphi(a-i, b-j) = [P * J_\varphi](a,b).
\end{aligned} \tag{I.3}$$

The operation "$*$" is called a *convolution*. Thus, the effect of a linear operation φ on a picture P is simply the convolution of this picture with the impulse dispersion picture of the operation φ. This fact is of large practical importance, as it points to a close kinship

between linear parallel operations and the *correlation function* of two pictures. Indeed, the correlation function for two functions, say f and g, of two variables, in the discrete case has the form:

$$[\varrho(f,g)](a,b) = \sum_{(i,j)\in U} f(i,j)g(i-a,j-b) = [f * \overline{g}](a,b). \tag{I.4}$$

where $\overline{g}(x,y) = g(-x,-y)$. Thus, we have immediately

$$\varphi(P) = P * J_\varphi = \varrho(P,\overline{J}_\varphi). \tag{I.5}$$

For two pictures F and G, the value of the correlation function $\varrho(F,G)$ for some point (a,b) can be used as a good measure of similarity between the picture F and the picture G shifted to this point. Thus, finding in the picture F some fragments (features) maximally similar to the prescribed shape or brightness distribution can be reduced to finding of maxima of the correlation function between the picture F and the picture G containing a template of the feature searched for. That is, it can be done by executing some linear operation (with F as an argument) whose impulse dispersion picture is equal to the symmetric reflection (with respect to the origin point $(0,0)$) of the picture G containing the template. That procedure is called *template matching* (or *matched filtering*).

I.3.5 Operations on binary pictures

Besides arithmetic operations defined in the previous section, natural operations on binary pictures include boolean point operations, treating the point values 0 and 1 as logical values **false** and **true**, respectively. The boolean operations are also naturally extendable to number-valued pictures (for negation we then have the formula $v_{\max} - p(0,0)$ for the defining function). For binary pictures treated as subsets of the raster U, boolean point operations map onto set-theoretic operations in a standard way.

The most important subclass of picture processing operations producing binary pictures (and often applied to binary pictures as well) is the class of thresholding operations.

Definition I.7 (Linear thresholding operation) *A* linear thresholding operation *is a picture operation φ for which the defining function f_φ has the form:*

$$f_\varphi(p) = \begin{cases} 1, & \text{if } \sum_{(i,j)\in N} a_{ij}p(i,j) \geq t_\varphi, \\ 0 & \text{otherwise,} \end{cases}$$

where $a_{ij}, t_\varphi \in \mathbb{J}$ and t_φ is called the threshold *of the operation φ.*

It is easy to show [Sklansky 1970] that linear thresholding operations are not linear in general. The usual method of implementation of such operations is by applying the point thresholding operation

$$f_t(p) = \begin{cases} 1, & \text{if } p(0,0) \geq t, \\ 0 & \text{otherwise,} \end{cases} \tag{I.6}$$

to the result (a number-valued picture) of some linear homogeneous operation on the argument picture. Again, operations of this sort are commonly used in picture processing. The weight operation (I.2) for binary pictures produces simply the number of non-blank pixels in the picture.

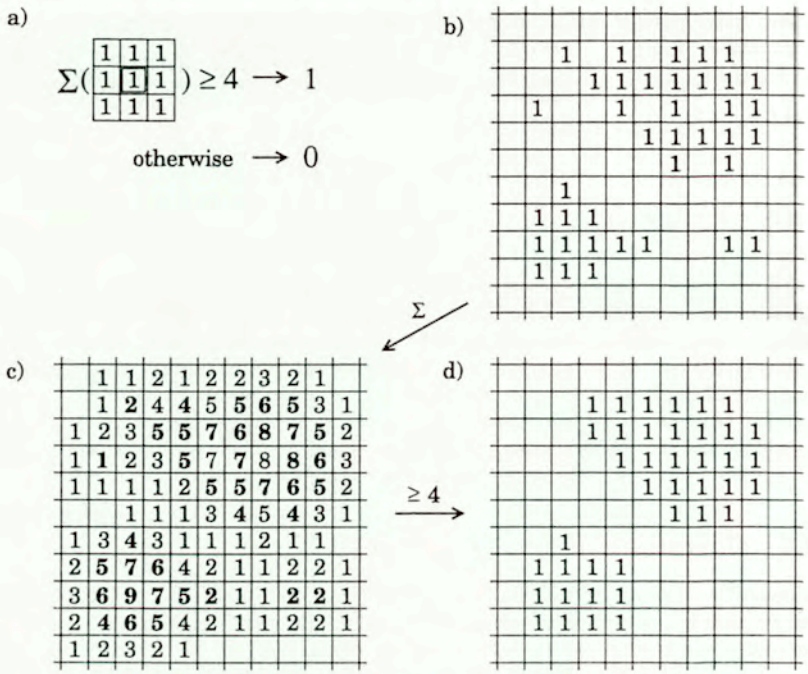


Figure I.8: Noise filtering with linear thresholding operation: graphical specification of the f_δ function (a), the input picture (b), the result after linear operation (c), and after subsequent thresholding (d).

Example I.2 (Local noise filtering) A simple example of a useful linear thresholding operation is the "salt and pepper" noise filtering operation δ shown in Fig. I.8. In the example, the input is the binary picture (like that in Fig. I.6a, but with some noise added). The function f_δ is defined graphically in Fig. I.8a. It uses the basic 8-adjacent neighbourhood with uniform weights $a_{ij} = 1$. Points with the value of 0 are left empty. In Fig. I.8c, boldfaced digits mark pixels with the value 1 in the input picture. ■

I.3.6 Mathematical morphology

A set of simple local parallel picture operations on number-valued pictures, investigated and applied under the name of *mathematical morphology*, (see e.g. [Serra 1989, Nieniewski 1998]) gained certain popularity. The morphological operations are usually described in a rather involved and intricate way. A more simple and clear formulation is attempted here, based on the formalism developed earlier in this section. Two different definitions of basic morphological operations are in use (see [Nieniewski 1998] for their comparison). Here the considerations are restricted to the variant developed by [Serra 1989].

In the sequel, for neighbourhoods and subpictures we will use an additional notation $\overline{N} = \{(-i,-j) \mid (i,j) \in N\}$ and $\overline{p}_N = \{((-i,-j), p(i,j)) \mid (i,j) \in N\}$ for some $p \in \pi_N$. That is, $\overline{N}$ (respectively, $\overline{p}_N$) is the neighbourhood (subpicture) obtained by symmetrical reflection of N (respectively, p) with respect to the origin point $(0,0)$.

Binary pictures. Consider a binary picture B_U defined as a subset of the raster U, and some neighbourhood N (in the morphological terminology, called a *structuring element*). In this simple case, the morphological operations can be defined as follows.

Definition I.8 (Binary erosion and dilation) *The operations* $\oplus$ *(*dilation*) and* $\ominus$ *(*erosion*) on binary pictures with structuring element* N *are local parallel operations defined as:*

$$\begin{aligned} B_U \oplus N &= \bigcup_{(i,j)\in\overline{N}} d_{ij}(B_U) = \bigcup_{(i,j)\in N} d_{ij} B_U, \\ B_U \ominus N &= \bigcap_{(i,j)\in\overline{N}} d_{ij}(B_U) = \bigcap_{(i,j)\in N} d_{ij} B_U. \end{aligned} \tag{I.7}$$

Alternatively, the operations can be defined in a manner used in Definition I.4, through the subpicture functions $f_\oplus$ *and* $f_\ominus$ *defined, for* $p \in \pi_N$, *as follows:*

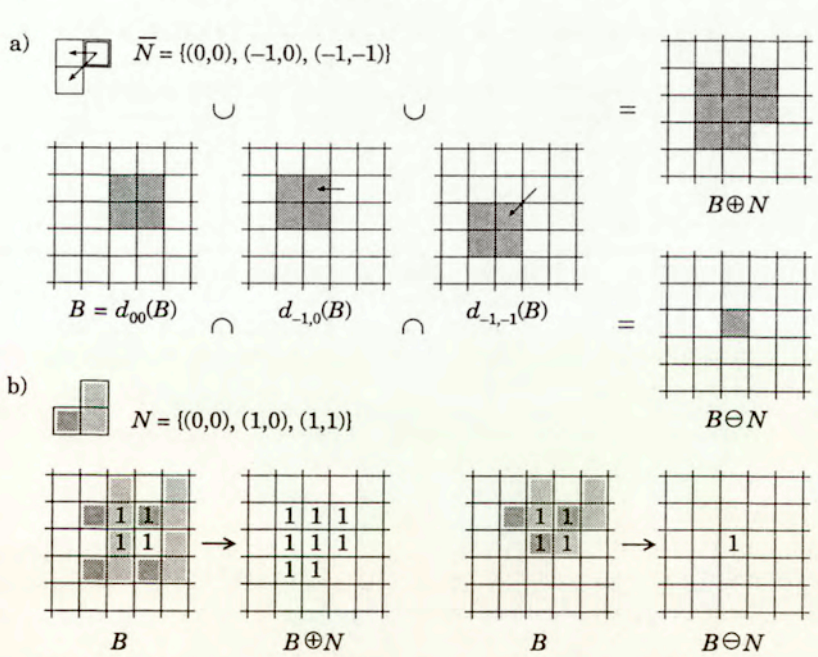


Figure I.9: Morphological operations realized according to formulae (I.7) (a) and (I.8) (b). In the latter case, a few selected positions of the neighbourhood N are shown.

$$f_{\oplus}(p) = \max\{p(i,j) \mid (i,j) \in N\} = \begin{cases} 1 & \textit{if symbol 1 occurs in } p, \\ 0 & \textit{otherwise.} \end{cases}$$
$$f_{\ominus}(p) = \min\{p(i,j) \mid (i,j) \in N\} = \begin{cases} 0 & \textit{if blank symbol occurs in } p, \\ 1 & \textit{otherwise.} \end{cases} \tag{I.8}$$

From this formulation, the pictorial effect of the operations is clear: dilation expands the non-blank components of the picture by all points contained in appropriately shifted neighbourhoods $\overline{N}$ placed on every non-blank point of the picture, while erosion shrinks the picture components analogously. See Fig. I.9 for an illustration of the definition.

Note that with the structuring element $N = \{(i,j)\}$, both morphological operations are simply shifting operations, namely $\sigma_{ij}(P) = P \oplus \{(i,j)\} = P \ominus \{(i,j)\}$. Therefore, if arbitrary n-argument point operations are also allowed, then due to Theorem I.1, all local parallel operations can be realized. However, such universality of morphological operations is rather trivial, as the whole processing burden is laid on the set of point operations instead of on the morphological ones.

Example I.3 (Contour extraction with morphological operations) Although the morphological operations as such cannot directly produce pictures of the sort shown in Fig. I.7, an appropriate combination of them with point operations can produce pictures containing separately contour and interior of picture components, as shown in Fig. I.10, cf. Fig. I.7. ■

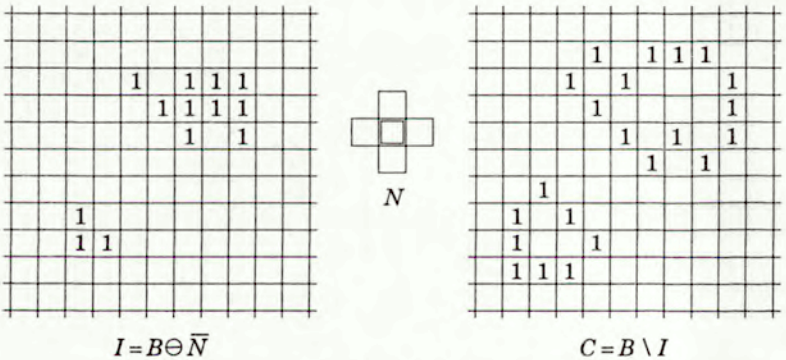


Figure I.10: Contour extraction with morphological and point operations. The input picture B is the same as in Fig. I.7b.

Number-valued pictures. The morphological operations can be easily extended to number-valued arguments, as follows. The simpler version uses, like for binary pictures, a structuring element that is a neighbourhood:

Definition I.9 (Many-valued erosion and dilation) *The operations* $\oplus$ *(*dilation*) and* $\ominus$ *(*erosion*) on number-valued pictures with a structuring element* $N \subset U$ *are local parallel operations defined by the subpicture functions:*

$$\begin{aligned} f_{\oplus}(p) &= \max\{p(i,j) \mid (i,j) \in N\}, \\ f_{\ominus}(p) &= \min\{p(i,j) \mid (i,j) \in N\}. \end{aligned} \tag{I.9}$$

Note that the above definition is in essence the same as for binary pictures treated as number-valued pictures, cf. formulae (I.8).

The generalization of morphological operations to use as a structuring element not a neighbourhood, but a subpicture, must take into account the values of points in the subpicture. This is done in a natural way as follows:

Definition I.10 (General many-valued erosion and dilation) *The operations* $\oplus$ *(*dilation*) and* $\ominus$ *(*erosion*) on number-valued pictures with a number-valued structuring element* $q \in \pi_N$ *are local parallel operations defined by the subpicture functions:*

$$\begin{aligned} f_{\oplus}(p) &= \max\{p(i,j) + q(i,j) \mid (i,j) \in N\}, \\ f_{\ominus}(p) &= \min\{p(i,j) - q(i,j) \mid (i,j) \in N\}. \end{aligned} \tag{I.10}$$

Note that in the above, p *is the subpicture cut out from the appropriately shifted argument picture* P *by the neighbourhood* N*, cf. Definition I.4.*

In this way, the simpler version from Definition I.9 is equivalent to the general version with a blank subpicture used as a structuring element. In practical applications, also some combined operations are used, namely:

Definition I.11 (Opening and closing) *The morphological operations* $\square$ *(*opening*) and* $\blacksquare$ *(*closing*) are composite operations defined for any* $P \in \Pi(\mathbb{J})$ *as:*

$$\begin{aligned} P \square S &= (P \ominus \overline{S}) \oplus S, \\ P \blacksquare S &= (P \oplus \overline{S}) \ominus S, \end{aligned} \tag{I.11}$$

where S *is a structuring element—either some neighbourhood* N*, or a subpicture* $p \in \pi_N$ *defined over* N*.*

The effects of opening and closing operations are similar to the local filtering like that shown in Example I.2. If the value 0 (blank symbol) is meant to represent white and 1 to represent black in the picture, the opening operation in general deletes small (as defined by the structuring element size and shape) black elements on white background, whereas closing deletes white elements on black background, cf. Figs. I.9 and I.11.

Example I.4 (Local filtering with morphological operations) Results of noise filtering of the picture from Fig. I.8b with morphological operations of opening and closing (using as structuring element the 4-adjacent neighbourhood of Fig. I.6c) are shown in Fig. I.11. Compare the results with those of Fig. I.8d in Example I.2. ■

I.3.7 Sequential picture operations

Another class of local picture operations is called sequential, because a single step of application of some local operator is for them done not for all points simultaneously, but only in a single place at a time. Therefore, the next application of the operator uses the picture already changed in the previous step(s). There are several schemes for performing such sequential operations, differing mostly by the manner the subsequent local operator applications are controlled. Three main types of such schemes are the *two-dimensional Turing machine*, *sequential operations of Rosenfeld and Pfaltz*, and *planar grammars*.

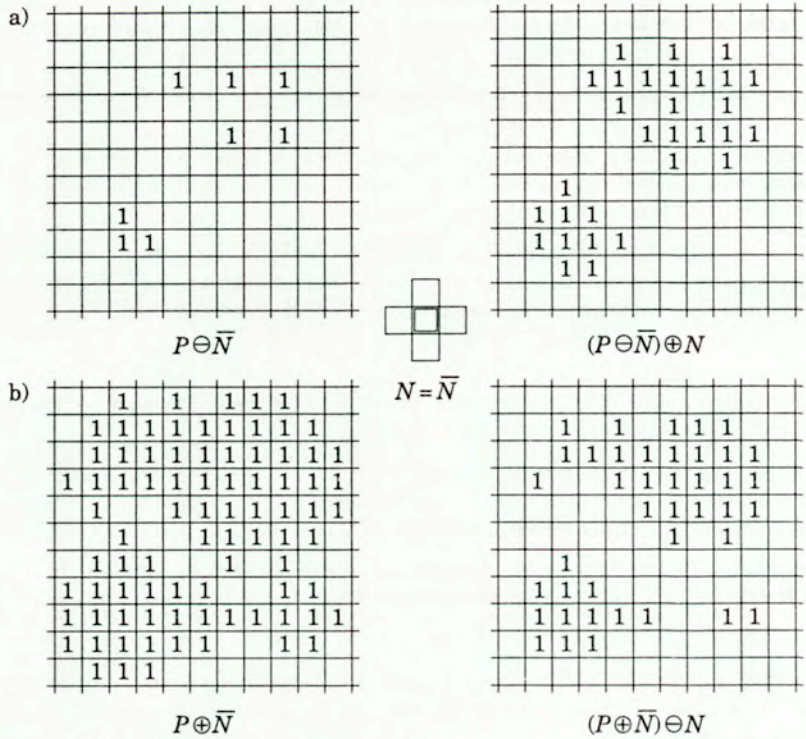


Figure I.11: Local filtering with morphological operations: opening (a) and closing (b). The input picture P is the one used in Fig. I.8.

Two-dimensional Turing machine. Extension of the Turing machine to a two-dimensional "tape" is straightforward, requiring only addition of a few new commands for moving the machine head in two dimensions. This model has mostly theoretical value, see e.g. [Minsky & Papert 1969], though a few attempts to use it in practical applications were made, usually after some significant modifications like taking into account not only a single point under the head, but some its neighbourhood. As an example one may take the BUGSYS system of [Ledley et al. 1966]. Note that in this control model the choice of the next point to consider is made as a result of observing the current point (or neighbourhood), thus it significantly depends on the contents of the picture.

Sequential operations of Rosenfeld and Pfaltz. In this model, the picture is systematically scanned line by line (usually in the same sequence as is customary in television systems) and at every point a local operator using the basic 8-adjacent neighbourhood, of the same kind as the f_λ functions of Definition I.4 is applied, see [Rosenfeld & Pfaltz 1966]. At any given point the operator changes the point value according to its defining function before proceeding to the next point in sequence. Thus, in this control scheme

the sequence of points considered in subsequent steps is rigid and does not depend on the contents of the picture. Such operations were realized in hardware by [Kruse 1973]. Every parallel operation available in his machine (called PICAP) can be applied also in the sequential manner described above.

Authors of the paper [Rosenfeld & Pfaltz 1966] proved that the class of such sequential operations is computationally equivalent to the class of local parallel operations using also the same basic 8-adjacent neighbourhood. They also listed some picture processing problems (like counting of connected components on a binary picture) for which sequential operations are more effective than parallel ones (when programmed on ordinary sequential computers).

Planar grammars. Planar grammars are local rewriting systems extended to two dimensions from one-dimensional string rewriting grammars. Their theory was investigated extensively, see e.g. [Milgram & Rosenfeld 1972] and [23, 39]. They can be used both as tools for defining classes of pictures (so-called *planar languages*), or, more important for practice, as picture processing operators. Concerning their latter role, in [23, 39] it was proven that they are computationally equivalent to general nondeterministic algorithms using local parallel operations.

The basic notion for this model is the notion of a (local) planar rewriting rule.

Definition I.12 (Planar rewriting rule) *A* (local) planar rewriting rule *is an ordered pair of subpictures* $(p_l, p_r) \in \pi_N(A) \times \pi_N(A)$ *so that* $Up_l \neq \{o\}$, *i.e., the left-hand side of the rule contains at least one non-blank symbol. An* application *of a planar rewriting rule* (p_l, p_r) *to the picture* P *at the point* (i, j) *produces a picture* Q *such that if* $\mathbf{I}_N d_{ij} P = p_l$ *then* $\mathbf{I}_N d_{ij} Q = p_r$ *and* $\mathbf{I}_{U \setminus N} d_{ij} P = \mathbf{I}_{U \setminus N} d_{ij} Q$, *or else* $Q = P$ *(where* $N = p_l A_o = p_r A_o$ *is the neighbourhood used by the rule). That is, when the left hand side of the rule matches the fragment of the picture* P *at the point* (i, j), *this fragment is replaced by the subpicture specified as the right hand side of the rule (leaving the rest of the picture* P *unchanged), otherwise the application has no effect.*

Because in general the rewriting rule can change several points within its (shifted) neighbourhood, it cannot be applied in parallel (simultaneously) for all points of the picture, as there might be places where several applications of the rule with overlapping neighbourhoods may be ambiguous, prescribing different changes to the same point of the picture. Therefore, the control schemes used in the planar grammar processing model usually prescribe only a single application of any given rule at some place in the picture. However, when the rule changes only one point (and is then practically only a different form of the defining function f_λ of some parallel operation, see Definition I.4) or is guaranteed (at least for the class of pictures to which it will be applied) not to exhibit the ambiguity effect, it can be applied also in parallel. The processing result can be, however, different for such parallel application of the rules, unless the applicability of every rule does not depend on application of any other rule, see Example I.5 below. In practice, rewriting rules are sometimes used in parallel, like in the diagrammatic reasoning on the raster model of [Furnas 1990], see Example I.25 in Section II.6.2.1.

The execution control in this processing model consists usually of the following steps, repeated until the stopping condition is achieved:

1. **Select a rule** from the set of rules of the operator.
2. **Choose a place** at the processed picture where the left hand side of the rule matches the contents of the picture.
3. If there is no such place and not all rules were yet checked, go to (1).
4. **Apply the rule** at the chosen place.
5. **Check the stopping condition**: if satisfied, **stop**, otherwise go to (1).

The rules used in the operator may be defined on different neighbourhoods (see Example I.5 below or Example I.25 mentioned above). There may be different methods of choosing the rule and the place of its application. The stopping condition in the picture processing applications usually consists of checking if the picture changed since the previous check. In picture grammar applications, the process stops when the picture contains no more symbols from the specified set of so-called *nonterminal symbols*.

Example I.5 (Sequential contour extraction) The parallel contour and interior extraction operation presented in Example I.1 can be easily programmed also as a sequential operation using the set of rewriting rules defined in Fig. I.12. Like in Example I.1, because

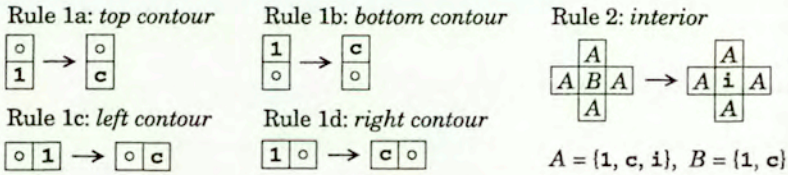


Figure I.12: The rules for sequential contour extraction operation.

Rule 2 contains names of sets of symbols, it actually represents a rule schema, i.e., a set of (in this case 162) rules with different combinations of symbols from the indicated sets (here A and B). It is assumed that in every rule derived from such a rule schema, at any given position in the neighbourhood containing a symbol set name on both sides, the symbols on both sides of the resulting rule are the same. As it is easy to see, the set of rules shown works well for any pictures over the alphabet $A = \{1, c, i\}$, differently than in the parallel version shown in Example I.1, where the input pictures must use only blanks and 1 symbols. With this restriction, the results of applying the rules of Fig. I.12 will be exactly the same as for the parallel version of Fig. I.7. Moreover, as all the rules change only a single symbol each and their applicability does not depend on the result of application of any another rule from the set, they can be applied in parallel with the same final results. ■

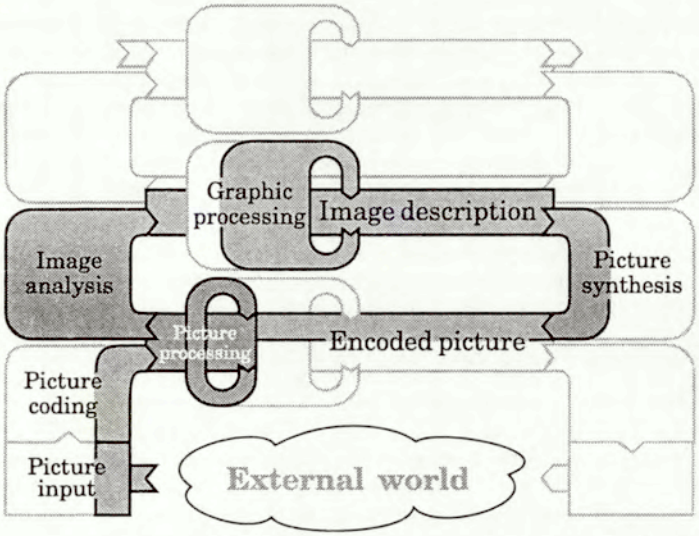


Figure I.13: Processes and information structures involved with the area of discrete image analysis.

I.4 Discrete image analysis

The exploration of the island was finished, its shape determined, its features made out, its extent calculated, ...
[Jules Verne, *The Mysterious Island* (1874)]

The task of image analysis, see Fig. I.13, divides into two main processes:

Picture segmentation, that is finding in the encoded raster picture elementary regions corresponding to meaningful input objects or their fragments.

Object measurement, that is measuring various features (qualitative and quantitative) of the objects found in the picture and determining relations between them.

A general introduction to various approaches to picture segmentation is presented in Section I.4.1. The object measurement tasks are discussed in Sections I.4.2 and I.4.3, using the example of measuring areas and perimeters of figures (regions) represented on discrete pictures.

I.4.1 Picture segmentation

The detection of meaningful objects in discrete pictures is based on analysis of distribution of pixel values (brightness levels or colours) in the picture. Before the discrete picture is obtained for segmentation, the picture input and coding device must transform the real-world picture into a discrete form. That *picture digitization* process may, in general, proceed in two ways, see Fig. I.14:

Point sampling: pixel values are obtained by sensing the input at point-sensors located at raster points. Although there are no physical sensors of zero dimensions like mathematical points, a good approximation to this schema is obtained if the size of the sensor is small enough in comparison to distance between raster points. It occurs in practice in adjustable-resolution devices that possess small sensors for high resolution imaging but can be used for coarser scanning by applying them at larger intervals (like in many electromechanical scanners).

Local averaging: here the size of the sensor is comparable to the raster point distance, and the brightness (or colour) of the input picture is averaged over the area of the sensor (aperture), often with different weights for different positions within the aperture, due to nonuniform sensitivity of the sensor. As a result, the resulting digitized picture is usually multilevel, even when the input picture is sharply black and white only.

If the input picture is of good quality, evenly lit, etc., its point-sampled discrete representation requires little further segmentation effort, as different object differ unambiguously by their (uniform) brightness or colour. The region of the input picture is then represented by the set of raster points that fall inside the region, see Figs. I.14a and d. Otherwise, similarly as with local averaging digitization, region edges became blurred and region interiors may cease to have uniform pixel values, see Fig. I.14b.

Various approaches to picture segmentation of such multilevel pictures can be divided into two groups:

Edge-based: using **differences** between pixel values.

Region-based: using **uniformities** in distribution of pixel values.

The first approach is often called *edge detection* [Levialdi 1981, Pavlidis 1982, Pratt 1978] and is usually conducted by various types of local operations based in general on local high-pass filtering (or differentiation). Currently, also methods based on *tensor analysis* are used [Cyganek 2001, Granlund & Knutsson 1996]. Edge detection produces a set of pixels constituting the *contour* of the region (marked with square cells in Fig. I.14c), and the whole region is obtained by filling this contour by its interior pixels. Note that the contour can contain both the points falling inside and the points falling outside the original region, see Fig. I.14c.

The region-based approach takes into account the opposite features of the pixel values distribution, namely detecting possibly large areas with similar pixel values. Often so-called *propagation* or *region growing* methods are used here: the operation starts from some pixel and iteratively adds to the growing region neighbour pixels with small value differences compared to the already selected pixels [Brice & Fennema 1970, Pavlidis 1982, Pratt 1978]. In simpler cases various methods based on the thresholding operation (cf. formula (I.6) in Section I.3.5) can be used. The global thresholding uses a single threshold value for all pixels, selected in a way that best differentiates the regions with different pixel values. Usually the threshold is selected on the basis of analysing the histogram of pixel values. With pictures containing regions of similar pixel values on the background of different value (e.g., all black objects on white background) the histogram is bimodal and the optimal threshold lies at the minimum between the two largest maxima. When the scene is unevenly lit, or when different objects have different surface characteristics, this simple method does not work. One of popular approaches in such

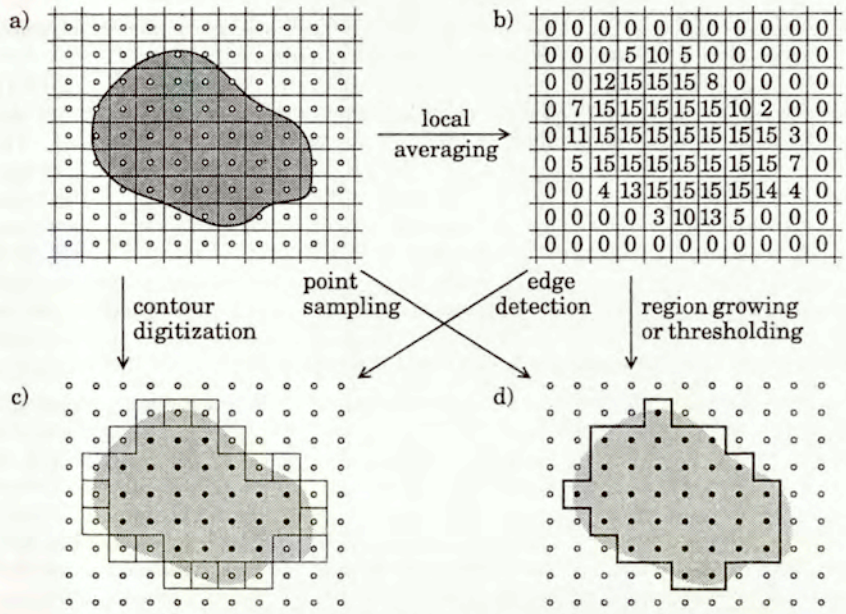


Figure I.14: From the input picture (a), through picture coding (b) to edge-based (c), and region-based (d) digital object representations.

cases is called *dynamic thresholding*, where the threshold is selected individually for each pixel on the basis of analysing the pixel values histogram in its neighbourhood (usually much wider than the basic neighbourhoods of Fig. I.6c and d, see e.g. [36, 68]).

The input picture digitization is closely related to problems of picture synthesis, that is, producing a discrete raster picture from (usually geometrical) symbolic description, see the right-hand arrow in Fig. I.13. Given a description of some region (or a curve—possibly a contour of a region), the task consists of finding a discrete object, i.e, a set of pixels, that approximates the original object in the best possible way (according to some given criterion prescribing what representations should be considered better than others). The methods used here are generally similar to that discussed above for the case of encoding of real-object pictures, dividing again into region-based approaches (like local averaging or point sampling) and edge-based approaches (like various methods of curve rasterization, called also, less properly, curve digitization). The former were discussed already, while curve rasterization methods differ significantly from the edge-detection methods, thus they will be discussed here in more detail. They divide into two types:

Cell-based, like the *point-in-a-box* method that collects all those raster points through whose *cells* the given curve passes (the cell being a unit square around the point, see Fig. I.6b).

Point-based, where the main raster point choice criterion is the minimization of distance between the chosen raster points and the curve. Different methods use different

kinds of distance calculation and different characterizations of the set of points allowed to be considered as candidates for the rasterization.

The point-in-a-box method produces 4-connected discrete "lines" (cf. Section I.3.1). Point-based methods usually produce 8-connected lines, though other possibilities also can occur, including pixel sets disconnected in the sense defined in Section I.3.1. The simplest method of the point-based kind is the *Freeman digitization* scheme [Freeman 1970, Freeman & Glass 1969] and [20–22], where distance is measured along the raster lines (horizontally and vertically only), see Fig. I.18a. Very fast and simple algorithms for various kinds of curves exist for this method, see e.g. [Bresenham 1965, Doros 1979] and [22]. They produce in general 8-connected lines (unless the curve contains small details of the size comparable to raster points distance). In some cases, problems can be caused by the so-called *ambiguity points*, see the crossed points in Fig. I.18a, which lead to non-uniqueness of rasterization of some lines, see [22].

Example I.6 (Discrete object representations) A simple real-world flat object and its various digital representations are shown in Fig. I.14. After superimposing the object on a raster (Fig. I.14a) it can be either point-sampled producing the representation in Fig. I.14d, or digitized into a multi-level picture (say, with 16 gray levels in this case) of Fig. I.14b by local averaging (with the aperture equal to the raster cell in this case), or contour-digitized into Fig. I.14c (here, using Freeman curve digitization). The gray level picture of Fig. I.14b can be subsequently segmented by some edge detection method (here, a simple choice of pixels with values intermediate between 0 and 15) producing again Fig. I.14c, or by thresholding (here with the global threshold of 8 (cf. formula (I.6)) to obtain Fig. I.14d. The region-growing method with the maximal gray level difference between region pixels set at 7 produces the same result. With the zero tolerance condition for region pixels, the interior of the region is obtained, as in Fig. I.14c (black dots). In Figs. I.14c and d the original object is superimposed in gray over the discrete representation for comparison. In Fig. I.14c, the representation of the object consists of both the internal pixels (filled black circles) and the 8-connected contour pixels (marked by outlining them with raster cell boxes). In Fig. I.14d, as the contour of the region serves the between-cells boundary (drawn with a thick line). ■

The resulting discrete representations of the original object shown in Figs. I.14c and d are essentially different and cannot be simply transformed into each other without reference to the original object. Therefore, also any measurements of properties of the object must directly take into account the differences that arise from different methods of object digitization and extraction.

I.4.2 Area measurement

An obvious measure of the area of some object in a discrete picture is the number of pixels comprising the object, of course multiplied by the physical area of the pixel cell as projected back to the real world. However, it is not correct in all cases, because the method of producing (extracting) the discrete representation of the object may influence substantially the correctness of such a simple approach. Thus, for different digitization and picture segmentation methods, possibly different area measurement methods will be needed. It is indeed the case with the two main types of discrete object representation discussed before.

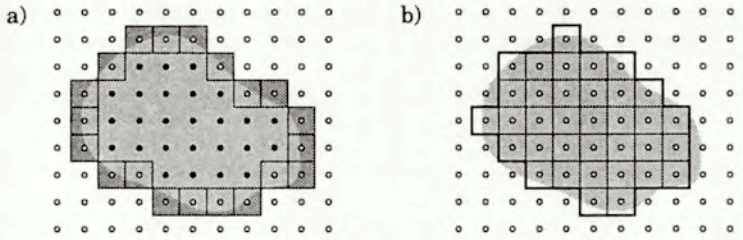


Figure I.15: Region area measurement for edge-based (a), and region-based (b) representations.

Areas of edge-detected regions. Superimposing the original region on the result of its edge-based representation as in Fig. I.15a (cf. Fig. I.14c) it is easily seen that simple pixel counting will produce a large systematic error as a region area measure. Counting all pixels of the region (both interior and contour) will produce an excess area (marked by dark gray in Fig. I.15a), while counting only interior pixels will produce a deficiency in the area (consisting of the light gray parts of contour cells in Fig. I.15a). It seems that a mean value of these two measurements will be much better. Indeed, the correct formula is given by the following theorem by Pick (after [Steinhaus 1950][7]):

Theorem I.4 (G. Pick 1899) *The area of an arbitrary polygon with all vertices at the points of a square raster is given by the formula:*

$$S_P = S_i + S_c/2 - 1,$$

where S_i is the number of raster points lying inside the polygon and S_c is the number of raster points lying on the sides of the polygon.

The polygon with which the theorem is concerned is here obtained by connecting the 8-adjacent contour points of the region, see Fig. I.16a. As can be seen in that figure, the polygon approximates the contour of the original region fairly well. The Pick's Theorem was first proposed for area measurement in the discrete picture analysis context in [75]. In [22] it was proved that the resulting area is on average the same for all polygons with vertices at raster points which have the same Freeman-digitized 8-connected contour. The result is valid, with good accuracy, also for other contour digitization schemes using the distance-to-curve approach. The paper [18] contains further clarifications of applicability conditions of the Pick's area measurement formula (in answer to certain objections raised by [Rosen 1980]). The approach was also tested in [21] on non-polygonal figures, namely Freeman's digitization of circles, with excellent results.

Areas of point-sampled regions. As it is clear from Fig. I.15b, the pixel counting applied to region-based representation, as obtained, e.g., by point-sampling of the original region, in average approximates the original region area fairly well. It can be proved formally that for many measurements with different positions of the region with respect

[7] See also `http://www.cut-the-knot.com/ctk/Pick.shtml` for comprehensive web pages on the Pick's Theorem (including the proof) and its uses.

to the raster the mean error of measuring its area by counting raster points falling inside the region tends to zero, see [Steinhaus 1950]. Thus, for point-sampling representation of regions the simple pixel counting method is certainly valid, whereas the analysis for Freeman-digitized circles conducted in [21] indicates a considerable systematic error of this method.

Example I.7 (Area measurement) It is easy to calculate the area of the example region of Fig. I.14a. The Pick's Theorem applied to Fig. I.15a produces the area $S_P = S_i + S_c/2 - 1 = 26 + 20/2 - 1 = 35$ units, while pixel counting applied to Fig. I.15b gives also 35 units. It is interesting to compare that result with measurements done with pixel counting for Fig. I.15a and Pick's Theorem for Fig. I.15b. All the results are summarized in Table I.1. Bold-faced entries indicate results for the cases calculated above—where the method properly matches the representation. The results are the same, indicating the equivalence of both methods when applied in their proper circumstances. When the method does not match the representation, however, large errors may result, as indicated by remaining entries in the table. ■

Table I.1: Comparison of area measurement methods.

Region representation	Pick's theorem	pixel count inner	pixel count outer
edge-based	**35.0**	26.0	46.0
region-based	22.5	**35.0**	

I.4.3 Perimeter measurement

Measurement of the perimeter of a region, and more generally, measurement of length of any real-world line, suffers from the so-called "British Isles" paradox. Namely, when the line is inspected for measurement in more and more detail, its measured length grows indefinitely. Fortunately, in the discrete region case there are natural limits to the effect. The basic limit comes from the limited precision of the raster—details smaller than the distance between raster points cannot be represented in the discrete picture. Moreover, in practice a certain model of the original region contour is assumed, such that the shape of the assumed contour is represented in the discrete version with sufficient detail for the given application. The assumption may, for example, prescribe minimal radius of curvature for the contour detail which is of interest for the application. The most common assumption is the same as that adopted for area measurement in the preceding Section I.4.2, because it leads to simpler considerations. Namely it is assumed that the original region is a polygon, with sides larger than the distance between raster points.[8] Further simplification of the analysis, without much harm for its accuracy, is obtained by assuming polygon vertices to coincide with raster points. Now the discrete contour can be considered as a sequence of digitizations of straight line segments.

[8]Note that one cannot set for polygons a lower limit for radius of curvature of the contour, as it equals zero at polygon vertices.

Thus, an obvious perimeter measurement method is to find first the polygon, whose digitization according to the method used in the picture input process coincides with the given discrete contour. Indeed, such a polygonal approximation method is often used, cf. [22]. It has certain drawbacks, however: the polygonal approximation algorithms tend to be complex and may produce sometimes rather awkward results, partly due to the non-uniqueness of the operation, as there can be many different polygons with the same digitization [Montanari 1970].

Fortunately, there are more direct methods based on that polygonal model which do not require actual finding of the polygon. Of course, again the perimeter measurement rule must take into account the picture digitization and segmentation method by which the final discrete picture representation of the region (and its contour) has been produced.

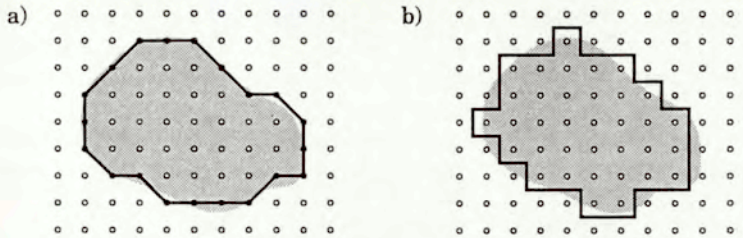


Figure I.16: Region perimeter measurement for edge-based (a), and region-based (b) representations.

Perimeters of edge-detected regions. The 8-connected discrete contour of a region extracted by edge-detection approximates the original blob contour fairly well, see Fig. I.16a. One can measure its exact length by the obvious formula

$$l_8 = n + s\sqrt{2}, \tag{I.12}$$

where n is the number of horizontal and vertical segments, and s is the number of diagonal segments between adjacent pixels. Unfortunately, that formula produces a value larger than the length of the smooth original contour. However, assuming the polygonal model of the original region contour introduced above, it is possible to calculate exactly the excess length and introduce appropriate correcting factor to the above formula, see [22].

Consider the straight line segment AB (constituting one of the original polygon sides) shifted to the first octant of coordinate axes as shown in Fig. I.17a. Its true length is:

$$l_E = b/\sin\varphi,$$

while the length of its 8-connected digitization according to the formula (I.12) is:

$$l_8 = (a-b) + b\sqrt{2} = l_E(\cos\varphi - \sin\varphi) + l_E\sqrt{2}\sin\varphi,$$

for $0 \leq \varphi \leq \pi/4$, because here $a = l_E\cos\varphi$ and $b = l_E\sin\varphi$. Therefore,

$$l_E = l_8/(\cos\varphi + (\sqrt{2}-1)\sin\varphi).$$

For $0 \leq \varphi \leq \pi/4$, the graph of the function $\epsilon = \cos\varphi + (\sqrt{2}-1)\sin\varphi$ is shown in Fig. I.17b; for other orientations φ of the polygon side, the error function ϵ repeats with

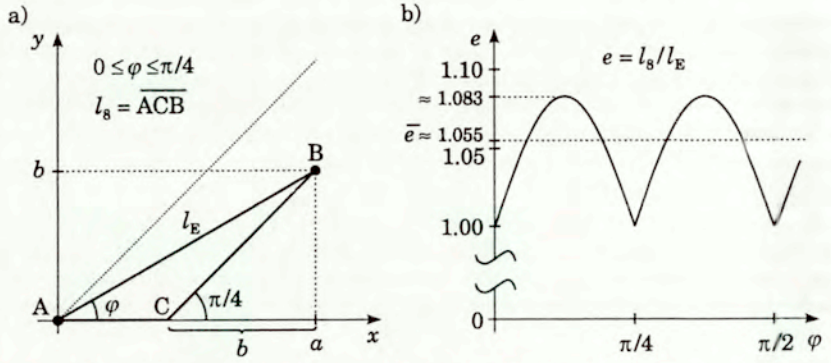


Figure I.17: Calculating correction factor for 8-connected discrete lines: lengths l_E and l_8 of a line segment in the first octant (a), and the error function ϵ with its average (b).

the cycle of $\pi/4$ (though the formula in terms of the angle φ will be different). It is easy to calculate the average value $\bar{\epsilon}$ of this error factor over all possible slopes of segments:

$$\bar{\epsilon} = \frac{4}{\pi}\int_0^{\pi/4}(\cos\varphi + (\sqrt{2}-1)\sin\varphi)d\varphi =$$
$$= \left[\frac{4}{\pi}(\sin\varphi - (\sqrt{2}-1)\cos\varphi)\right]_0^{\pi/4} = \frac{8(\sqrt{2}-1)}{\pi} \approx 1.055. \quad \text{(I.13)}$$

Now because $l_E = l_8/e$, then assuming that the original region is a polygon with orientations of sides, taken modulo $\pi/4$, spanning uniformly the interval $[0, \pi/4]$, the length of the polygon perimeter is well approximated by the formula:

$$l_K = l_8/\bar{\epsilon} = \frac{\pi(\sqrt{2}+1)}{8} l_8 \approx 0.948(n + s\sqrt{2}). \quad \text{(I.14)}$$

Determination of the numbers n and s can be made directly from the picture containing the region by simple parallel local picture operations akin to that of the Example I.1 in Section I.3.2, combined with the "weight" operation, see formula (I.2) in Section I.3.4. The maximal errors of the formula can be easily calculated to be +2.5 percent and −5.3 percent. These maximal errors are attained when all segments have orientations corresponding to maxima or minima, respectively, of the function ϵ, see Fig. I.17b.

Perimeters of point-sampled regions. The contour of a point-sampled region is given as the between-pixels cell boundary, see Fig. I.16b. Shifting the boundary diagonally by half the distance of raster points we cause all corners to fall at the raster points, thus obtaining a 4-connected discrete line. The length of this line is given by:

$$l_4 = m, \quad \text{(I.15)}$$

where m is the number of horizontal and vertical segments of the line. This length, as is clear from Fig. I.16, overestimates the true object perimeter even more than the length of the 8-connected line in the edge-detection case. In fact, when the region does

not contain too deep concavities, all regions having the same circumscribing rectangle have their l_4 perimeter length equal to the perimeter of the rectangle, irrespectively of their shapes. That overestimation was proposed to be compensated by means of an appropriate correction factor, see [Proffitt & Rosen 1979, Ellis et al. 1979, Rosen 1980]. The correction factor of $\pi/4$ nullifies the average error, but leaves the standard deviation as large as 12 percent [Proffitt & Rosen 1979]. Therefore, in the above paper, a so-called "corner counting rule" was proposed. Namely, the corrected length was calculated as:

$$l_R = \alpha m - \beta k, \tag{I.16}$$

where m is the number of horizontal and vertical segments, and k is the number of corners in the 4-connected discrete line. Now it remains to find the appropriate coefficients α and β that nullify the average error and minimize the deviation over all orientations of the digitized straight segment. The coefficients with these properties were indeed found in [Proffitt & Rosen 1979], leading to the final formula:

$$l_R = \frac{\pi(\sqrt{2}+1)}{8} m - \frac{\pi\sqrt{2}}{16} k \approx 0.948m - 0.278k. \tag{I.17}$$

The standard deviation of this formula is about 2.3 percent. Note that it is not the *maximal* error, so that it cannot be directly compared with the maximal error given for the l_K formula above.

The values of m and k can be also easily found directly from the picture by simple parallel local picture operations and the "weight" operation.

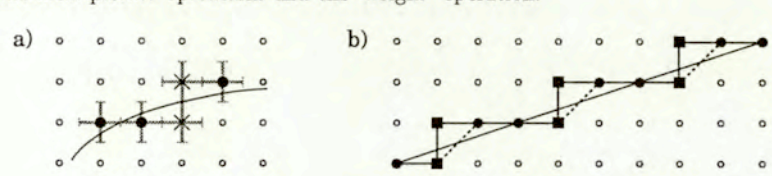


Figure I.18: Freeman's digitization of a line (a), and the relation between 4-connected and 8-connected digitization of a straight segment (b).

The 4-connected digitization of straight segments is closely related to their 8-connected Freeman's digitization, see Fig. I.18b. It is clear from the figure that every diagonal link of the 8-connected encoding corresponds to exactly two adjacent corners of the 4-connected encoding and replaces here two links of the latter. Therefore we have:

$$k = 2s, \quad m = n + k = n + 2s.$$

Substituting the above into (I.17) we get:

$$\begin{aligned} l_R &= \frac{\pi(\sqrt{2}+1)}{8}(n+2s) - \frac{\pi\sqrt{2}}{16}2s = \\ &= \frac{\pi(\sqrt{2}+1)}{8}n + \frac{\pi(\sqrt{2}+1)}{8}s\sqrt{2} = l_K . \end{aligned}$$

Therefore, the formulae (I.14) and (I.17) are in fact equivalent for respectively 8-connected and 4-connected digitizations of polygons.

Example I.8 (Perimeter measurement) For the 8-connected contour of Fig. I.16a we have $n = 12$ and $s = 8$, so that the application of the formula (I.14) produces

$l_K \approx 0.948(12 + 8\sqrt{2}) \approx 22.1$ units. The cell boundary of Fig. I.16b gives $m = 30$ and $k = 22$, hence from (I.17) we have $l_R \approx 0.948 \cdot 30 - 0.278 \cdot 22 \approx 22.3$ units which is very close to l_K. In a similar way one may try to use the formula (I.14) for the 8-connected contour of the region in Fig. I.16b, obtaining $l_K \approx 0.948(8 + 9\sqrt{2}) \approx 19.7$ units, and the formula (I.17) to the cell boundary of Fig. I.16a, obtaining now $l_R \approx 0.948 \cdot 32 - 0.278 \cdot 20 \approx 24.8$ units. The results for representations of Fig. I.16 are summarized in Table I.2, in a similar way as for the area measurement in Example I.7. Again they show the significance of applying the appropriate measurement method in its proper circumstances. ■

Table I.2: Comparison of perimeter measurement methods.

Region representation	l_K length	l_R length
edge-based	**22.1**	24.8
region-based	19.7	**22.3**

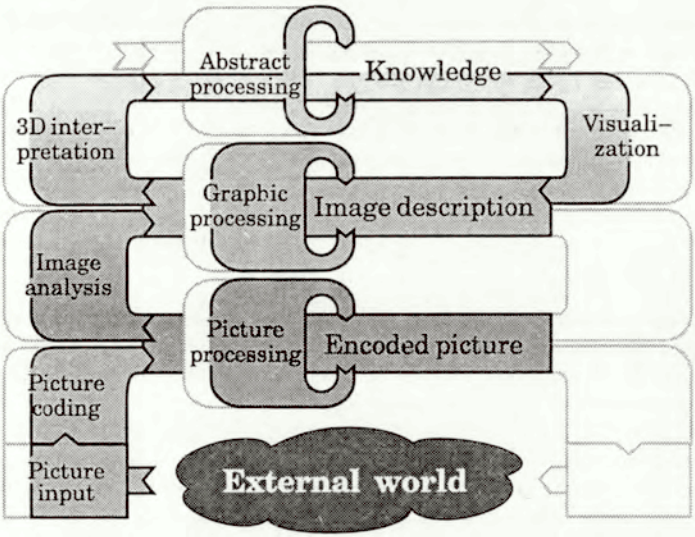


Figure I.19: Processes and information structures involved with three-dimensional scene interpretation and understanding.

I.5 Scene interpretation and understanding

... few if any of ... men seem to react
in the same way to all they see around them.
[Maurits C. Escher, *The Graphic Work of M.C. Escher* (1967)]

In order to be able to perceive and understand properly the real world scenes around us, the visual system, living or artificial, must be able to both recognize the objects in its field of vision and localize them in space. The spatial localization problem is the one that distinguishes the task of scene interpretation and understanding from other varieties of picture analysis. The problem, called *stereovision*, has been approached in various ways. They can be divided into two main types:

Passive methods involve analysis of the passively obtained flat image or images of the scene, and then trying to infer the spatial arrangement of the visible objects from the analysis of the images, possibly aided by some additional knowledge (like the information about possible objects that can occur in the scene, and their properties).

Active methods involve special devices (called often *range finders*) for measuring distance of selected points in the scene from a given position or positions and then trying to construct a three-dimensional model of the surroundings from that range data. To this class belong also the methods based on taking images of the scene under special lighting conditions actively generated by the system (like striped lighting) which makes the determination of the three-dimensional structure of the scene from such images much easier.

The first type above is the one used by human vision, hence investigation of methods of this type is of relevance to both psychology of vision and computer vision. Also, they can be investigated without the need of building any special equipment or sensors other than the ordinary picture processing systems of the sort described in Section I.2. Not surprisingly, the passive methods constitute the mainstream methods of scene understanding research and applications till today. The passive methods, in turn, divide into two classes:

Single image methods use only a single (flat) image to infer the spatial relations between objects on the basis of so-called *depth cues*. It is further discussed in Section I.5.1 below.

Multi-image methods use many (usually two) images taken from different viewpoints, using the differences between the images (called *disparities*, or *binocular disparities* in case of two images) to find spatial structure of the objects and spatial relations between them.

The multi-image methods use the fact that images of objects in space taken from different viewpoints differ depending on the position in depth of the objects. To conduct the analysis, usually two images suffice (the situation called *binocular vision*, like in human vision), though sometimes more images can be used to obtain more precise results. The research field investigating and using such multi-image methods is usually called *stereogrammetry* [Mokrzycki 1992a, STEREO 2001], and uses often sophisticated local processing (e.g., tensor analysis) to accurately find corresponding fragments of objects on different images and calculate reliably the exact value of disparity and the corresponding position in depth.

I.5.1 Monocular depth perception

The single-image depth analysis is a unique ability of humans, allowing in most normal circumstances for quite reliable inference of proper spatial structures. The analysis, called *monocular depth perception*, is based on so-called *depth cues*, i.e., certain common properties of image parts correlating with high probability with appropriate spatial arrangements. For example, usually the smaller object of the two of the same kind or shape is located farther on; also, the object partly obscuring another must be nearer of the two (assuming we have some reliable method to decide which of the two blobs of colour on the image obscures another). Researchers distinguish several different cues of various complexity and reliability (the most common catalogue lists nine of them, see [44]). Most of them are too complex or elusive to allow easy and efficient computer implementation, especially for usually complex real-life images. As usual in such cases, the work on computer monocular depth perception started from scenes of severely restricted complexity (usually the so-called *blocks world*) for which simple depth cues can be used. For the simple case of *trihedral solids* (i.e., blocks with exactly three surfaces meeting at every vertex), the depth cues can be formulated as a catalogue of allowed interpretations of edges of the solids that meet at corners of various types, see Fig. I.20 [Clowes 1971, Huffman 1971]. Such catalogues, called *labelling schemata*, were also compiled for wider classes of scenes, e.g. [Huffman 1971], including multi-block scenes containing blocks (not necessarily trihedral) with shadows [Waltz 1975].

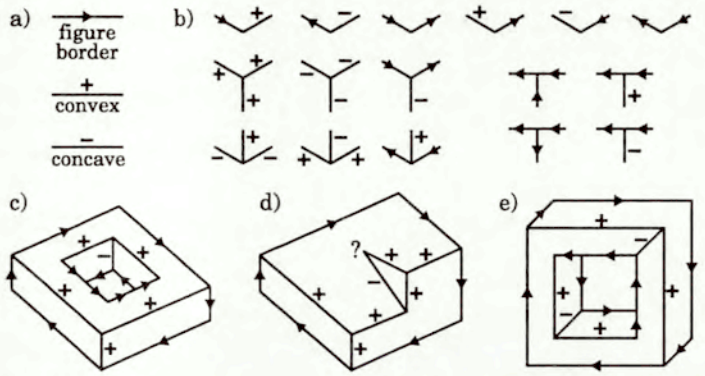


Figure I.20: Basic edge labels (a), the basic library of allowed labellings for corners of trihedral solids (b), and labelling examples: consistent labelling for a proper solid (c), and inconsistent (d) and consistent (e) labellings for impossible figures.

To obtain the final spatial configuration of blocks from a proper labelling of the block scene, usually the model-fitting stage is added, where possible flat views of the allowed kinds of blocks (see Fig. I.21) are fitted to the appropriate parts of the image (see [Mackworth 1976] for a survey of main approaches). The model-fitting approach is still widely used, both in monocular and binocular stereovision, with emerging general theory of view changes of complex solids when seen from changing viewpoints [Dąbkowska & Mokrzycki 1998].

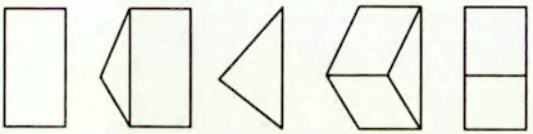


Figure I.21: All possible topologically distinct views of a certain solid.

The work on labelling schemata soon revealed a curious phenomenon. Some line drawings did not admit any consistent labelling—they seemingly represented impossible objects. An early, mostly forgotten work on *impossible figures* [Penrose & Penrose 1958] was recalled and the investigation of this curious phenomenon was started within the artificial intelligence field.

One of the results of this research was the recognition of limits of the edge-based approach. One of the symptoms was that while some impossible figures could be detected by the lack of consistent labelling for them (Fig. I.20d), many others still admitted a consistent labelling (Fig. I.20e). This led to the development of a surface-based reasoning about spatial structures of solids, as exemplified in the next subsections, especially Section I.5.3. Possibly the most systematic and rigorous approach to spatial interpretation of line drawings was conducted in numerous papers by Sugihara, culminating in a comprehensive book [Sugihara 1986].

I.5.2 "Impossible figures": errors of spatial interpretation

Everyone knows that dragons don't exist.
But while this simplistic formulation may satisfy the layman,
it does not suffice for the scientific mind.
[Stanisław Lem, *The Cyberiad* (1965)]

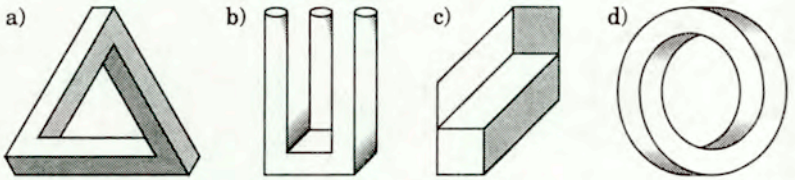


Figure I.22: Four common *impossible figures*: Penrose's (or Reutersvärd's) triangle (a), devil's fork (b), Thiéry's figure (c), and impossible ring (d).

In early papers on the so-called "impossible figures" it was assumed that the notion is so obvious that it was never explicitly defined, merely demonstrated by some pictorial examples, like these in Fig. I.22. Despite this lack of definition of its subject, one of the main goals of the early research was to find a *formal* (which usually meant *mathematical*) criterion of impossibility which would rigorously and unambiguously separate impossible drawings from possible ones. The first proposal was to use inconsistency of labelling as such a criterion [Huffman 1971]. However, in practice it become clear that such impossibility criteria are little more than tests of membership in some narrow classes of polyhedrons (like trihedral solids) having structures assumed by the particular labelling scheme used. Attempts at producing more and more complicated labelling schemata soon showed that there is no universal schema that can cover all practically needed, not to say conceivable, three-dimensional objects. Attempts were also made to find impossibility tests for some specific families of figures, like multibars, see [Cowan 1977] (called there "cornered toruses"). However, the classifications offered by such criteria too often differed considerably from the human judgement to consider them as capturing the nature of the phenomenon. The final blow was given to these attempts by a growing number of discoveries of possible spatial realizations for various impossible figures, starting from the impossible triangle[9] in Fig. I.22a, [Gregory 1970]. Should the formal and strict impossibility criteria automatically "change their minds," so to speak, after each such discovery? Thus, the need arose to formulate at last the definition of impossible figures that would be not necessarily formal, but rather able to follow the human judgement closely enough. Such a definition was proposed in [16, 19, 57]:

Definition I.13 (Impossible figures) *An* impossible figure *is a flat drawing that produces an* ***impression of a three-dimensional object*** *such that that object, suggested by the* ***human spatial interpretation of the drawing cannot exist****, i.e., its spatial interpretation contains geometrical contradictions* ***clearly visible to the human observer****.*

[9] Somehow, the earlier, first photograph of the possible model of the impossible staircase published in the first paper on impossible figures [Penrose & Penrose 1958] went completely unnoticed.

The definition contains three elements (boldfaced in the text) which are of crucial significance to understanding the phenomenon and will be thus analyzed in some detail.

First, the figure, to be at all amenable to a judgement of its impossibility, should be first subjected to a spatial interpretation. Every drawing, by virtue of its existence, is possible. What may be impossible in the drawing is only its interpretation: in the case considered here, its interpretation as a view of some three-dimensional object. This brings us also to the second part of Definition I.13, and can be summarized as the following basic observation:

What is impossible? *The property "to be an impossible figure" is not the property of the drawing, but the property of its spatial interpretation chosen by a human observer.*

Impossible figures are thus *visual illusions of spatial interpretation.*

The stress given to the role of *human observer*, both in Definition I.13 and in the observation above signifies the fact that it is the human that has the final saying on what constitutes the "proper" spatial interpretation of the drawing, as the only truly competent being in this matter. To this day, computers are far less competent than humans in the task of spatial interpretation of images, though they are already much more competent in the inverse task—generating flat views of complicated spatial scenes, also in motion, cf. also Section I.1.3, as every contemporary cinema or TV spectator is well aware of.

That points to the fact that the phenomenon is of the psychological rather than geometric nature. As such, it may vary with viewing conditions or with changing observers, so that the impossibility judgement is a matter of averaging over larger audiences and circumstances rather that a simple "yes or no" decision by a single individual endowed with some rigid impossibility test. Also, the ability to see impossible figures is a learned one and appears comparatively late in children's development. It critically depends on the ability to build internal spatial models (spatial interpretations) of the observed drawing [Young & Deręgowski 1981].

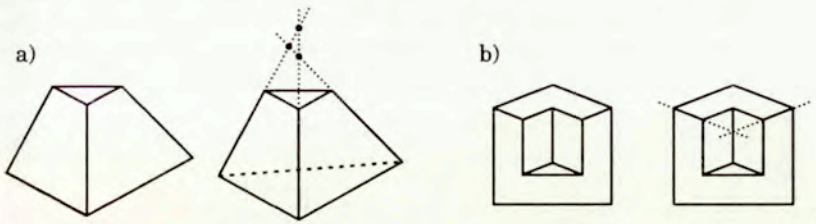


Figure I.23: Two *likely figures* with demonstrations of their impossibility: the truncated triangular pyramid (a), and the Huffman's corner (b).

The third part of the definition invites a more thorough study of possible types of contradictions that may occur in impossible figures (see the next subsection) and raises the issue of the recognizability of these contradictions by the observer. As it was first observed by [Huffman 1971], there are figures whose usual spatial interpretations are impossible, but the fact goes unnoticed by most observers, see Fig. I.23. The pyramid shown in this figure was published as an interesting example by many authors, while two families of its possible interpretations were published in [Térouanne 1983] and [19], respectively. The Huffman's corner first appeared in [Huffman 1971].

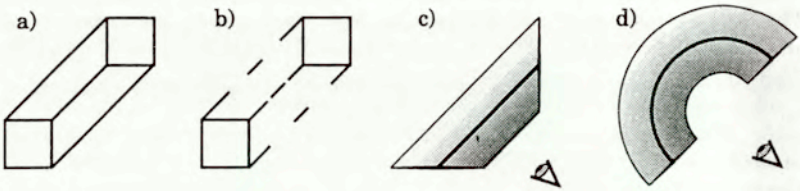


Figure I.24: An impossible beam (a), its impossible interpretation (b), and two easy to find possible interpretations (c,d).

Such figures were called *likely figures*. Huffman discerned also *unlikely figures*, defined by him as those that look impossible, but have easily found possible interpretations, see Fig. I.24. The possible interpretations are shown in the figure from the side, while the eye icon indicates the viewing direction which results in the "impossible" image of Fig. I.24a.

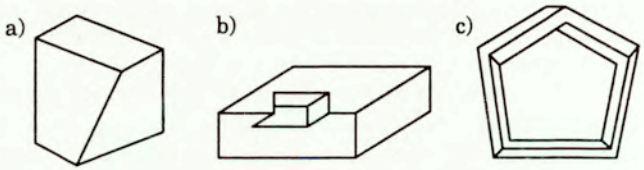
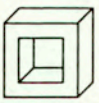
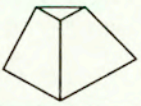
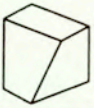


Figure I.25: Three *unlikely figures*: a skewed die (a), Huffman's plank (b), and Césari's pentabar (c).

However, such figures are not truly opposite to the likely ones, being rather ordinary impossible figures with low "degree of impossibility." Truly opposite would be the figures whose usual spatial interpretations are possible, but are judged as being impossible. Three examples of such figures are given in Fig. I.25. The Huffman's plank appeared in [Huffman 1971], and the pentabar was discussed in [Térouanne 1983]. Thus, in [16, 19, 57] the term

Table I.3: Four impossibility classes of figures.

Interpretation is judged as:	Interpretation is: possible	Interpretation is: impossible
possible	Possible figures 	Likely figures 
impossible	Unlikely figures 	Impossible figures 

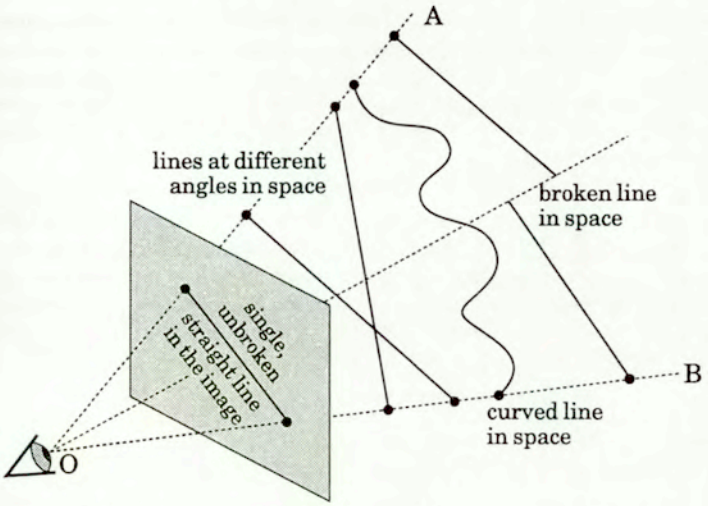


Figure I.26: Inherent ambiguity of spatial interpretation of drawings: an infinity of possible lines in space lying in the plane OAB produces the same straight line in the drawing.

unlikely figures was applied to just this kind of figures instead.

All these types of figures can be arranged into a schema shown in Table I.3. It classifies into four classes various line drawings representing three-dimensional objects according to the actual (im)possibility of their interpretations usually chosen by observers and the (im)possibility judgement of them made by these observers.

As a result of the analysis sketched above it became clear that spatial interpretation of flat drawings is an *inherently ambiguous* process—there is no single, absolutely correct spatial interpretation for any flat drawing, as there is an infinity of such interpretations possible. As shown in Fig. I.26, many different spatial objects can generate the same flat projections, hence any feature of the drawing can have many different spatial interpreta-

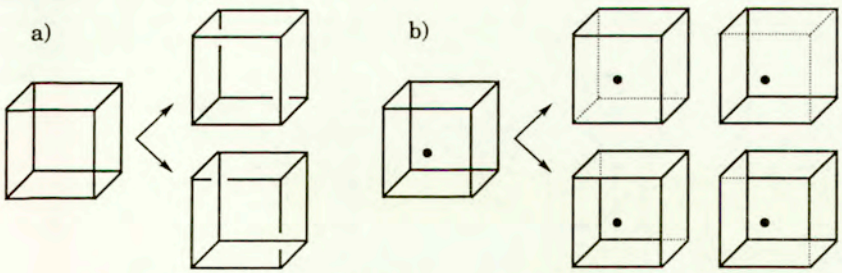


Figure I.27: Two interpretations of an ambiguous Necker's cube (a), and four for a Necker's box (b).

tions. Some of them may be considered as more probable in the given context than others, but it does not guarantee their correctness, nor even the consistency of the interpretations of various parts of the figure. The impossible triangle from Fig. I.22a illustrates that neatly: every corner of the triangle has its strong and convincing local spatial interpretation, whereas these three interpretations are obviously inconsistent globally when joined as indicated in the drawing.

Before that, several ambiguous figures, like the well-known Necker's cube (and its even more ambiguous version, the Necker's box), see Fig. I.27, and Thiery's figure [Thiéry 1895], see Fig. I.22c (a canonical form of the so-called "moon craters" illusion or convex-concave illusion) were studied and copied in psychology of vision papers over and over. However, they were treated mostly as merely rare curiosities, whereas now it was realized that ambiguities must occur in *all* flat images when only one attempts to interpret them spatially.

I.5.3 Impossibility sources

They were all, ..., nonexistent,
but each nonexisted in an entirely different way.
[Stanisław Lem, *The Cyberiad* (1965)]

According to Definition I.13, the drawing looks impossible when the local interpretations of its parts are visibly inconsistent. The contradiction occurs when certain elementary part of the interpreted figure obtains several (usually two) inconsistent spatial properties (like being at the same time vertical and horizontal), or when some two (or more) such parts become related by an inconsistent relation (like one being simultaneously below and above the other). Trying to discern such possibly contradictory local spatial features one can find three general classes of such contradictions [16, 19].

Figure-background contradiction. This contradiction is the most noticeable and hard to reinterpret into a possible interpretation, as it violates the most basic distinction between the part of the object and the background against which it is pictured. See Fig. I.28 for two simple examples: a standard one in Fig. I.28a, and the one using shadows—a self-shadow (figure) and a cast shadow (bakground) in Fig. I.28b (first published in [19]; see also Fig. II.39 in Section II.5.2.1).

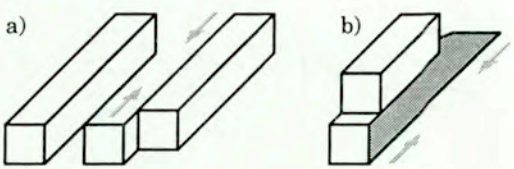


Figure I.28: Two figures with figure-background contradiction: a usual case (a), and self-shadow/cast shadow variety (b).

Position estimation contradiction This contradiction is very common in impossible figures. Usually it involves two parts of the object whose mutual position in space is estimated differently depending on the local context used for the estimating. Examples of estimating contradictory vertical position are shown in Fig. I.29ab, and position in depth in Fig. I.29c.

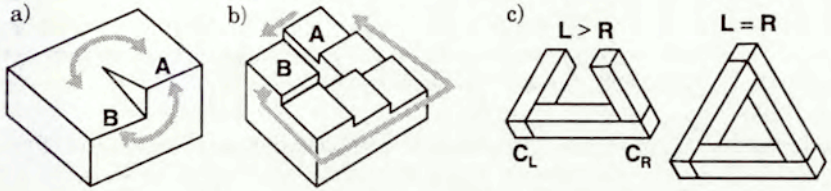


Figure I.29: Two figures with vertical position contradiction: Huffman's block (a), impossible staircase (b), and the position in depth contradiction in the impossible triangle (c).

In the Huffman block in Fig. I.29a [Huffman 1971] local analysis of the edges A and B with the vertical segment joining them indicates that A is above B, while noticing that both A and B are edges of the upper horizontal face of the block suggests that the edges must be at the same level. The impossible staircase in Fig. I.29b constitutes a variety of Penrose's staircase [Penrose & Penrose 1958] with the minimal number of steps (according to the general formula developed in [19]). From local analysis of the steps A and B it is concluded that A is above B, while considering the sequence of steps from B to A around the right-hand corner of the staircase one obtains a contradictory relation—that A is below B. Such position contradiction in space explains also the structure of the impossible triangle in Fig. I.29c. In the orientation in which the triangle is usually drawn the position in depth of every corner is contradictory. Considering the top corner, from the depth cues provided by bottom corners C_L and C_R one concludes that the upper ends of the side beams must point in opposite directions in space, so that the L end is farther, while the R end is nearer than the bottom beam. However, in the completed triangle both L and R are at the same distance, hence the contradiction.

Surface form contradiction. This effect is very common and often arises as a byproduct of other sources. It occurs when an area in the drawing (representing a fragment of the interpreted object surface) obtains different spatial conformations depending on which partial context of the area is taken into account.

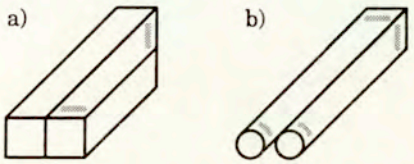


Figure I.30: Two figures with surface form contradiction: a twisted plane contradiction in Ernst's dibar (a), and planar-curved contradiction in the double rod figure (b).

There are two common special cases of this type of contradiction. In the *surface orientation contradiction* (called sometimes a *twisted-plane contradiction*), an apparently planar face of the object admits two contradictory orientations in space, like horizontal versus vertical, as in the Ernst's dibar [Ernst 1986] in Fig. I.30a. In the *planar-curved contradiction* different shapes of the surface, namely planar and curved, are clashing with each other, as in the double-rod figure in Fig. I.30b.

This contradiction can be often reinterpreted as possible by admitting a smooth transition from one of the estimated forms into the other, as usually they are assigned to the given area at different places along its boundary, see Fig. I.30.

Mixed cases. In many impossible figures several contradictions of the same or different types occur, sometimes one being a consequence of the other. For example, in Fig. I.28a the offending strip with figure-background contradiction contains also vertical position contradiction, being both at the ground level (as a background) or above it (as a topmost surface of the beam). In Fig. I.28b, the area of the shadow contains also the contradiction between vertical and horizontal orientation of the same plane. The vertical position contradiction in Fig. I.29b admits also an explanation in terms of planar-curved contradiction for the upper surface of the block.

Final remark. The phenomenon of impossible figures obtains a new and broader significance in the context of research on diagrammatics, as an example of problems that may arise from partial analogicity of diagrammatic representations, see Section II.3.1.3 and [2].

I.6 Diagrammatics

Diagrammatics is a very young discipline—a more or less official beginning of it can be ascribed to the year of the first AAAI Symposium on the subject [DIAGRAMS 1992]. Its aims concern theoretical investigation and practical applications of diagrams as a general tool for information and knowledge representation and reasoning, see Section II.1.4. Diagrams were used in these roles since times immemorial, but, surprisingly, no systematic and scientifically rigorous study of them and their use has been undertaken until recently. Diagrammatics as a scientific discipline aims to play similar role with respect to diagrams as linguistics plays with respect to ordinary languages, natural or artificial (like programming languages). One of the main goals of diagrammatics is such a systematization and formalization of the knowledge about diagrams and their use that they can be implemented on computers in a useful way. Computer implementation of diagrams may be attempted in order either to help humans to use diagrammatic tools more effectively, or to serve as a module for artificial intelligence systems to make them capable to reason diagrammatically, see Section II.6.

As will follow from Chapter II, which contains a much more detailed exposition of basic issues of diagrammatics, the use of diagrams, either by humans or by machines, effectively spans the whole area of the interpretation/representation schema in Fig. I.3, as discussed in this chapter. Diagrams may appear at the lower level, externally with respect to the system, being here used as an external reasoning aid, external information storage, or a communication medium (with other such systems, human or machine). They can be also used as an internal information representation, on both encoded picture and image description levels. Therefore, diagrammatics must draw extensively from approaches and results of the whole field introduced in this chapter. Moreover, it must also use findings of several other disciplines, especially cognitive psychology and psychology of vision (concerning internal mechanisms of using diagrams by humans), and information design (to tap a vast store of knowledge on practical recipes for producing good diagrammatic representations), and mathematics (both as a source of formal research tools and as a challenging application area, see Section II.5). Hence, diagrammatics is a truly interdisciplinary field, trying to integrate various research tools and sources of knowledge, and promising to develop new tools and applications, and refine old ones, in practically all domains of human activity.

As mentioned above, Chapter II contains a review of basic issues of diagrammatics, while Chapter III contains an exposition of a diagrammatic notation and representation system for interval algebra, developed by the author, together with several examples of its use in interval algebra research.

Chapter II

Diagrammatics: an introduction

These forty years now, I've been speaking in prose without knowing it!
How grateful am I to you for teaching me that!
[Moliere, *The Bourgeois Gentleman* (1622-1673)]

Diagrammatics, or, as it is still sometimes descriptively called, *diagrammatic (knowledge) representation and reasoning*, is a quite young field of study, although, like with the Monsieur Jourdain's prose, diagrams were in common use since times immemorial, see Section II.3. The beginning of diagrammatics as a scientific discipline can be formally ascribed to the year 1992, when the first workshop on the topic was organized, as an AAAI Spring Symposium on "*Reasoning with Diagrammatic Representations*" [DIAGRAMS 1992]. Most of the issues raised and discussed there are still hot topics of debate among diagrammatic researchers, despite several books and article collections that appeared since then, like [DIAGRAMS 1995, Hammer 1996, DIAGRAMS 1996, DIAGRAMS 2000a, DIAGRAMS 2000b, DIAGRAMS 2001, DIAGRAMS 2002, 77, 78]. Diagrammatics is thus still a discipline in the process of development. A formulation of its subject, scope and goals is discussed in more detail in Section II.1.4 below.

In this chapter only a general survey of the most important topics, especially concerning the issue of more or less formal diagrammatic reasoning, will be included. The survey in part relates findings reported in the relevant literature, and partially introduces new ideas and results obtained by this author. The latter range from proposals of new terminology (e.g., "divergence" in Section II.4.3), or clarification of some common but often imprecisely used notions (e.g. Section II.1.3), through several new illustrative examples and reformulation or new analysis of old ones, to several new ideas and proposals. The latter include a new look at several alleged problems with and limitations of diagrams (Section II.3.2), a new general classification of diagrammatic reasoning types (Section II.4.1), some topics of visual language design and choice (Section II.2), especially a survey and classification of visual language styles used in mathematical diagrams (Section II.5.4), with thorough refutation of main arguments being raised against the use of diagrams in mathematics (Section II.5), and development of the *diagrammatic spreadsheet* concept for computer implementation of diagrammatic tools (Section II.6.4).

II.1 Knowledge representation

> ... solving a problem simply means representing it
> so as to make the solution transparent.
> [Herbert A. Simon, *The Sciences of the Artificial* (1981)]

One of the important early findings of artificial intelligence research has been the recognition of importance of problem representation for finding the solution of a problem. Therefore, the main task of a problem solver, whether human or artificial, is first to find a proper representation of the problem (i.e., the formulation of the problem itself and the knowledge pertinent to its solution). The wording of that recipe given by Simon in the motto above may seem too far-reaching at the first sight, but it ceases to feel as such after a little consideration. Indeed, we are acting exactly as prescribed in the motto most of the time when solving various problems. For example, solving a simple algebraic equation consists of a series of transformations of an algebraic representation of some unknown value(s) until one achieves the representation transparent enough, like "$x = 2.5$."

The paper [Amarel 1968] is considered to be the first important work explicitly addressing the general issue of designing good representations of problems. It is often pointed out as a beginning of the knowledge representation subfield of AI research. Amarel devised there a series of representations for the generalized version of the classic *"Missionaries & Cannibals"* problem, starting from a predicate calculus representation (augmented with a production system notation), and compared their effectiveness. Interestingly, the subsequent, more efficient representations designed by Amarel were increasingly diagrammatic in their nature. Thus it may seem surprising that further research on knowledge representations concentrated almost exclusively on logical (propositional) representations. Amarel's diagrams became practically forgotten for a long time [Brachman 1990]. Possible reasons for that will be considered in various places of this chapter.

One of the results of knowledge representation research has been the discrimination of two main types of representations: the so-called *analogical* (called also *direct*, or *homomorphic*) representations and *propositional* (or *Fregean*, or *sentential*) representations. The term "Fregean" was introduced in [Sloman 1971] in honour of Gottlob Frege, whose work laid foundations for the predicate logic notation. The term "sentential" is used more often, e.g. in [Larkin & Simon 1987]), but it seems less fitting than "propositional," as it may lead to confusion of propositional representations with descriptions using sentences in some (natural) language.

II.1.1 Analogical versus propositional representations

> One resource alone remains, ...
> I must try the method of Analogy.
> [Edwin A. Abbott, *Flatland: A Romance of Many Dimensions* (1884)]

The following definition of the analogical knowledge representation was formulated in [Sloman 1975], see also [Sloman 1971] and [AI Handbook 1981]:

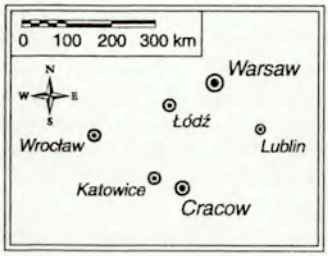
Definition II.1 (Analogical representations) *If A is an analogical representation of* [factual domain] *F, then there must be parts of A representing parts of F,* [...] *and it must be possible to specify some sort of correspondence, possibly context-dependent, between properties or relations of parts of A and properties or relations of parts of F.*

In spite of this direct relation between parts of the representation A and the represented domain F, relationships within F do not need to be explicitly named in A, i.e., there need not be a part of A corresponding to relation names like "south of" or "far off" in F (see the example below). Also, the relationship between A and F need not be an exact isomorphism.

That somewhat imprecise definition can be possibly clarified a little by means of a simple example.

Example II.1 (Analogical versus propositional) Compare the two types of representations shown in Fig. II.1.

a)



b) *Cracow: The city 250 km south of Warsaw. Katowice lies 65 km west of Cracow. ...*

c) *Cracow* = (ι *x*) (CITY(*x*)
and DIR (*Warsaw*, *x*, S)
and DIST (*x*, *Warsaw*, 250 km)),
DIST (*Cracow*, *Katowice*, 65 km),
DIR (*Cracow*, *Katowice*, W), ...

Figure II.1: Analogical (a) versus propositional (b, c) representations of some geographical facts (see Section II.1.2 below for explanation of the notation used in (c)).

In the analogical representation, size of, direction to and distance between marks on a map directly represent size of, direction to, and distance between cities, see Fig. II.1a. In contrast, in a sentential (propositional) representation, like in the phrase *"The city 250 km south of Warsaw"* in Fig. II.1b, its parts (e.g., the word *"Warsaw"*) or relationships between them (e.g., that *"Warsaw"* appears after *"south"* in the phrase) need not correspond to any parts and relations within the thing denoted. The city denoted by the phrase, namely *Cracow*, contains neither *Warsaw* nor *south* as a part, and *Warsaw* is not after *south* on the surface of the Earth. On the other hand, the relation between *Cracow* and *Warsaw*, explicitly named *"south of"* in the sentence does not appear explicitly, as an object one can point to, in the analogical representation of the fact (on a map). The same is true for a more formal propositional representation, like predicate calculus formulae provided in Fig. II.1c. ■

The distinction may be also stated in another way:

Definition II.2 (Analogical versus propositional) An analogical representation *has a structure whose syntax parallels (models), to a significant extent, the semantics of the problem domain.* A propositional representation *has a structure that has no direct bearing on the semantics of the problem domain.*

Not surprisingly, analogical representations are usually specialized for some specific application domain, whereas propositional representations can be made easily, by their very nature, quite universal. However, it makes them much less effective as a result. They suffer from the lack of semantic (or heuristic) guidance necessary to effectively navigate through usually vast search spaces during information retrieval and reasoning.

Whether it is possible to construct a significantly general, analogical knowledge representation seems to be still an open question. There is a strong evidence that it may be impossible, see e.g. [Barwise & Etchemendy 1996a]. Even if it were possible, it is probably not practical. The more practical approach is to use *hybrid* (*heterogeneous*, *mixed*) representation schemes [Barwise & Etchemendy 1996b], e.g., with separate domain-dependent analogical representations linked by a universal propositional superstructure [Iwasaki et al. 1995, 10, 14]. Most working systems that use diagrammatic representations are hybrid, as were the early systems [Gelernter 1959, Gelernter et al. 1960, Funt 1980].

The distinction between analogical and propositional representations is often wrongly interpreted. This sometimes leads to misunderstandings in discussions between proponents of both approaches. The interested reader may find the comprehensive listing and discussion of the most common misrepresentations of the distinction in [Sloman 1975]; see also [Stenning 2000]. The matter is still not fully settled—yet another definitions and thorough discussions of the distinction are given in [Barwise & Etchemendy 1996b] and [Barwise & Hammer 1996]. A more recent discussions of the issue are due to [Stenning 2000, Shimojima 2001], whereas [Gurr 1999] adds to that a more deep analysis of properties of analogical representations.

It is important to bear in mind that the propositional versus analogical distinction is neither absolute nor sharp. There are various degrees of being analogical (see e.g. the discussion of verbal versus visual thinking in [Arnheim 1969]), or the representation may be analogical to the represented domain along certain dimensions (or aspects), but propositional along others. The limiting case is reasoning directly with (or simulating) the target domain itself: if you cannot infer which lid fits the jar from the available information, try them on in turn. At the other extreme one can place, e.g., a Morse code, where the syntactic structure of the representation bears little if any discernible relationship to the structure of whatever the message is about.[1]

Hybrid representations. The representation may contain elements of both kinds discussed here, becoming thus a hybrid representation (called also *heterogeneous* [Barwise & Etchemendy 1996b] or *multimodal* [Pineda & Garza 1998]). This situation is ubiquitous in practice—in fact it is very hard, if not impossible, to find an example of indisputably pure case. Also when the representation itself seems to be purely analogical, its interpretation and reasoning with it usually involve essentially propositional components as well, see e.g. [Pineda & Garza 1998]. Whether a representation is analogical or not depends on the representation of what information it is, and how it represents this information. E.g., a set of logical formulae as written on paper (or computer screen) is a representation of the knowledge encoded with the formulae, but it is not an analogical representation of this knowledge. However, it is an analogical representation of the set of logical formulae! Or, as it was pointed out by [Sloman 1975], it may be considered as an analogical representation of the algorithmic *procedure* by which the underlying knowledge can be identified.

[1] The two examples mentioned here are taken from [Barwise & Etchemendy 1996b].

or computed.[2] The graphical (geometrical) relations between written graphical symbols directly correspond to the (syntactic) structure of the formulae, although rather not to the structure of the knowledge represented by them.

Note, however, that visual and essentially two-dimensional representations that are nevertheless propositional (like various kinds of mathematical notation) gain certain advantages (hence their origination and widespread usage) in comparison to textual, purely one-dimensional representations (like linear notations for mathematical expressions used in traditional high-level programming languages). It stems from the fact that they are analogical with respect to the structure of the expressions, though not (or to a significantly lesser degree) with respect to the knowledge represented by the expressions. Such analogical elements embedded in otherwise strictly propositional representations occur quite often—from natural language (where sometimes ordering of words or sentences directly represents some underlying facts or relations), to graph-structure organization of large knowledge bases of expert systems (see Section II.1.2). Maps, as yet probably the most successful and well developed system of analogical and diagrammatic representation (see Example II.1 and Section II.1.3) clearly exhibit that hybrid character, as summarized in the quote:

> It is difficult to imagine a map without language. However separate the evolution of iconic and linguistic representation, the map has, for millennia, embraced both. ... The map is simultaneously ... language *and* image.
>
> [Denis Wood, *The Power of Maps* (1992)]

Let us illustrate these issues by yet another example.

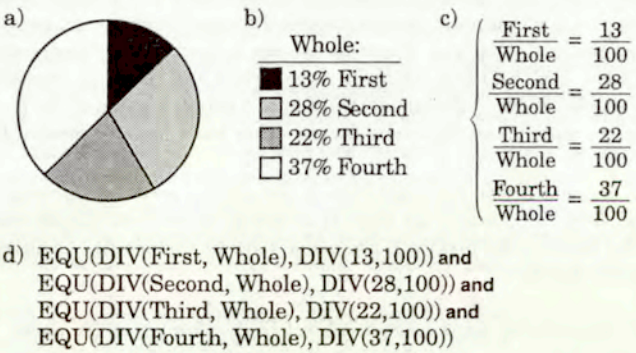


Figure II.2: Mixing propositional and analogical aspects in a representation (see text).

Example II.2 (Hybrid representations) In Fig. II.2, a knowledge about proportions of certain quantities is presented both diagrammatically (with a pie chart, Fig. II.2a), and propositionally (using "unadorned" predicate calculus, Fig. II.2d). The *legend* in Fig. II.2b links corresponding symbols in both domains (being thus a part of the definition

[2] See the embodiment of that idea in a number of programming languages based on predicate calculus formalism, with Prolog as the best known (and used) example.

of the visual language used, see Sections II.2 and II.3.3.1). The more usual mathematical notation of Fig. II.2c is thus propositional with respect to the domain knowledge (about proportions of parts in the whole), but (partially) diagrammatic with respect to the structure of the formula given in Fig. II.2d. ■

The role of the legend in such hybrid representations, only briefly mentioned in the example above, is well summarized by another quote:

> In the legend, semantic connections are made between classes of graphic images or image attributes and linguistic representations of the phenomena to which they refer. In this capacity, the legend acts as interpreter between the unique semiological system of the individual map and the culturally universal system of language ...
>
> [Denis Wood, *The Power of Maps* (1992)]

II.1.2 Logical representation

> ... the Old Entish ..., a lovely language,
> but it takes a very long time to say anything in it ...
> [John R.R. Tolkien, *The Two Towers* (1954)]

The most widespread type of knowledge representation in contemporary AI systems (especially *expert systems*) is the *logical representation* based on predicate calculus of the first order. It is considered to be a standard type of formalized propositional knowledge representation, thus serving as a reference point with which other proposed methods, including the diagrammatic representations discussed here, are compared. Referring the interested reader to the vast literature on the subject, e.g., [Genesereth & Nilsson 1987, Davis 1990, EXPERTSYS 2001] for more formal and thorough exposition, we will present only an elementary, informal description of a simple version of the method, to the extent needed to understand the examples and other basic issues discussed further on in this work.

The basic element of the representation is the *atomic predicate formula* of the form PNAME($arg_1, arg_2, \ldots, arg_n$), see Figs. II.1c and II.2d for some simple examples. The PNAME is the *name of the predicate*, while arg_i's are its *arguments*. As an argument can stand any *term*, that is:

A constant representing some concrete object in the domain of discourse. Here it will be denoted by a name (possibly in italics) starting with an upper-case letter, possibly with indices, or by some other notation pertinent to the domain (like number constants).

A variable representing some unknown object in the domain of discourse. Here it will be denoted by lower-case letters (in italics), possibly with indices.

An expression in some appropriate domain algebra (e.g., arithmetic) with function and operator symbols from the algebra, and constants and variables as their terminal arguments.

Another kind of term used sometimes is the "defining quantifier" described below. As a notational convention, some atomic formulae are often written as relational expressions with relation symbols taken from the domain. E.g., instead of LESS(x, 5), one usually writes $x < 5$. One-argument predicates can be interpreted to represent some properties of objects, while many-argument predicates represent relations between them.

From these atomic formulae one may then build more complex *logical formulae* with logical operators **and** (often implicitly assumed as binding all expressions in a list of expressions), **or**, **not**, **implies**, etc. Various symbols are used alternatively for the logical operators, like "$\wedge$" or "&" for **and**, "$\vee$" for **or**, "$\sim$" or "$\neg$" for **not**, and "$\Rightarrow$" for **implies**. The implication formula *ante* **implies** *conse* is often written in the form **if** *ante* **then** *conse* and called a *rule*. The subformula *ante* of a rule is called its *antecedent*, while *conse* is called its *consequent*. *Sentential logic* calculus can be interpreted as a subsystem of predicate calculus by identifying sentential variables with zero-argument predicates.

The formulae containing variables can be quantified with *existential* and *universal* quantifiers ($\exists x$) and ($\forall x$), so that ($\exists x$)F(x) means "there exists an x so that F(x) holds" and ($\forall x$)F(x) means "for all x holds F(x)" for some predicate formula F(x). For practical convenience sometimes another "defining quantifier" (usually called a *definite descriptor* or *iota operator*) is used, with (ιx)F(x) meaning "that (unique) x for which F(x) holds." Note that it is not a quantifier, also syntactically, because the result is not a formula but a term. See its use in Fig. II.1 in Example II.1. Variables in a formula that occur in some quantifier enclosing that formula are called *bounded variables* (within that formula), while the other variables are called *free variables*.

The logical representation of some knowledge has a form of some (possibly huge) logical formula. Note that a single formula suffices—a set of formulae, interpreted naturally as containing all formulae true for the represented situation, is equivalent to a single formula obtained from the set by connecting all the formulae from the set by **and** logical operator.

If the representation states some problem to solve, it must additionally include the statement of the problem itself—in this representation, another formula which should be shown to follow from the problem representation (including general logical axioms).

A concrete representation cannot contain free variables. All formulae containing variables in the representation are thus assumed to be appropriately quantified (see below).

For practical purposes, such an unstructured single formula is inconvenient. Therefore, the usual format of the representation, used in standard expert systems, divides that single formula into some subformulae of different character and mode of use.

Example II.3a (Logical formulation) An example will explain the general format of such a representation for a typical problem. It is taken from [15], with several notational modifications. The example originally appeared in [Larkin & Simon 1987], where it was differently formulated, using a computer-oriented algorithmic notation.[3] Table II.1 shows the representation of a simple mechanical problem, whose physical identity will remain undisclosed for a while. ■

[3]The version used in the paper [15], and its simplified variant used in [14], employ a somewhat informal notation directly modelled on the computer-oriented coding used in [Larkin & Simon 1987]. The standard, formal solution procedure described here would not work properly in that formulation. Hence it is here reformulated to follow the standard predicate calculus practice. There are several possibilities to reformulate this particular example—the one adopted here is not necessarily the simplest possible.

Table II.1: A logical representation example.

FACTS: structural description and object properties

$\mathrm{PA}(C_1)$, $\mathrm{PA}(C_2)$,
$\mathrm{PB}(C_3)$, $\mathrm{PB}(C_4)$, $\mathrm{PB}(C_5)$, $\mathrm{PB}(C_6)$, $\mathrm{PB}(C_7)$, $\mathrm{PB}(C_8)$, $\mathrm{PB}(C_9)$,
$\mathrm{PC}(C_{10})$, $\mathrm{PC}(C_{11})$, $\mathrm{PC}(C_{12})$,
$\mathrm{PD}(C_{13})$,
$\mathrm{PE}(C_3, C_{10}, C_4)$, $\mathrm{PE}(C_7, C_{11}, C_8)$, $\mathrm{PE}(C_8, C_{12}, C_9)$,
$\mathrm{PF}(C_1, C_3)$, $\mathrm{PF}(C_{10}, C_7)$, $\mathrm{PF}(C_{11}, C_6)$, $\mathrm{PF}(C_{12}, C_5)$,
$\mathrm{PG}(C_2, C_4, C_5)$, $\mathrm{PG}(C_{13}, C_6, C_9)$
$\mathrm{PH}(C_1, 1)$

RULES: how to derive new facts from the old ones

Rule 1:
if $\mathrm{PA}(x_1)$, $\mathrm{PB}(x_2)$, $\mathrm{PF}(x_1, x_2)$, $\mathrm{PH}(x_1, x_3)$
then $\mathrm{PH}(x_2, x_3)$

Rule 2:
if $\mathrm{PC}(x_4)$, $\mathrm{PB}(x_5)$, $\mathrm{PB}(x_6)$, $\mathrm{PE}(x_5, x_4, x_6)$ **or** $\mathrm{PE}(x_6, x_4, x_5)$, $\mathrm{PH}(x_6, x_7)$
then $\mathrm{PH}(x_5, x_7)$

Rule 3:
if $\mathrm{PC}(x_8)$, $\mathrm{PB}(x_9)$, $\mathrm{PB}(x_{10})$, $\mathrm{PB}(x_{11})$, $\mathrm{PE}(x_9, x_8, x_{10})$, $\mathrm{PF}(x_8, x_{11})$, $\mathrm{PH}(x_9, x_{12})$, $\mathrm{PH}(x_{10}, x_{13})$
then $\mathrm{PH}(x_{11}, x_{12} + x_{13})$

Rule 4:
if $\mathrm{PA}(x_{14})$, $\mathrm{PB}(x_{15})$, $\mathrm{PB}(x_{16})$, $\mathrm{PG}(x_{14}, x_{15}, x_{16})$, $\mathrm{PH}(x_{15}, x_{17})$, $\mathrm{PH}(x_{16}, x_{18})$
then $\mathrm{PH}(x_{14}, x_{17} + x_{18})$

GOAL: find a value for y such that...

$\mathrm{PH}(C_2, ?y)$

As it can be seen in the table, the representation of a problem is usually divided into three parts:

Facts, listing the objects occurring in the problem and their properties (or types), and relations holding between them. The formulae in this part are considered to be universally quantified, and often (as in this example), they do not contain variables.

Rules, being usually stated as formulae in the **if**...**then**... form and used to derive new facts (represented by appropriate predicate formulae) from the already established facts (initial ones or those previously derived). The formulae in this part are considered to be universally quantified.

A goal, stating the thesis that should be proven on the basis of the given facts. Often the sought of answer to the problem is not the mere statement that the goal formula can

be proven, but rather the valuation of some variable(s) for which it becomes true (in the example, the "answer variable" is marked by a question mark). Unless otherwise specified, the formulae in this part are considered to be existentially quantified.

Usually, the rules capture the general knowledge about the domain laws common to all problems solvable by the system. They are compiled and put into the system once and for all at the time of its creation. It is also implicitly assumed that they include all needed laws of logic (logical axioms). On the other hand, facts and goal constitute a description of a particular problem to be solved by the system, that is, an input data to be discarded after solving this problem. Facts and rules together constitute a *knowledge base* of the system for a given problem.

The example knowledge base utilizes also the two related principles stated below, often used in knowledge bases of that type.

Domain closure assumption. *It is assumed that the only objects that exist in the represented situation and are relevant to the problem at hand are those named by ground terms (i.e., constants and applications of existing functions to them).*

Closed world assumption. *It is assumed that the base of facts in the representation contains* all *atomic predicate formulae (stating valid statements about object properties and relations between them) relevant to the problem at hand.*

In consequence, one can conclude that if some ground atomic formula (a formula without variables and logical connectives, including negation) cannot be inferred from the knowledge base, then its negation should be assumed true. Unfortunately, the above conclusion is undecidable in general case, because the statement of the form "$\mathrm{P}(C)$ cannot be inferred from a given set of formulae" is generally undecidable. Therefore, in practice, a stronger, decidable version of this conclusion is used, formulated as the following rule:

Negation by omission. *When the fact (ground atomic formula stating some property of objects or relations between them) does not occur in the base of facts, it is assumed that its negation is true.*

Thanks to that assumption, representations of that type can be made much simpler: great numbers of statements excluding properties and relations that are known not to hold in the represented situation need not be explicitly stated as facts. E.g., in the example above it was possible to omit facts like **not** $\mathrm{PA}(C_6)$, **not** $\mathrm{PA}(C_4)$, **not** $\mathrm{PB}(C_1)$, **not** $\mathrm{PF}(C_2, C_4)$, **not** $\mathrm{PE}(C_1, C_{11}, C_5)$, **not** $\mathrm{PH}(C_1, 2)$, and so on. The rule makes the representation much simpler and more efficient for the price of decreasing expressive power of the representation: one cannot distinguish between facts that are false and those that are only unknown. This requires therefore a special provision for the new facts generated by application of the rules. Despite that they are not present in the base of facts, so that, by the negation by omission rule, they should be assumed false from the start, they are nevertheless added as true, on the assumption that actually they were only unknown, not false. That may sometimes introduce hidden contradictions when the knowledge base is improperly constructed. The situation is even worse when the knowledge base is not constructed once and for all, but must be updated many times when new knowledge or additional data arrive. Questions concerning these problems are studied under the heading of *truth maintenance*. See Section II.3.2.3 for an example of a similar problem in the context of diagrammatic representations.

Logic needed to handle properly such situations is called *nonmonotonic logic*; in fact most practical systems using logical representations are nonmonotonic logical systems [Genesereth & Nilsson 1987], though the fact is not always explicitly stated.

II.1.2.1 Reasoning with logical representation

They argued about it backwards and forwards for a long while ...
[John R.R. Tolkien, *The Hobbit* (1937)]

Solving a problem stated in the form described above requires, in general, finding a sequence of *rule applications* linking the facts with the goal. A standard rule application consists of the following steps:

1. Finding a *valuation* of all variables occurring in the rule, i.e., a substitution of domain algebra expressions for the variables in the rule, such that the antecedent of the rule becomes true (taking into account the facts of the problem statement, both initial ones and those already derived).

2. Adding to the set of facts the new, derived facts obtained from the consequent of the rule by applying to it the substitution found in the previous step.

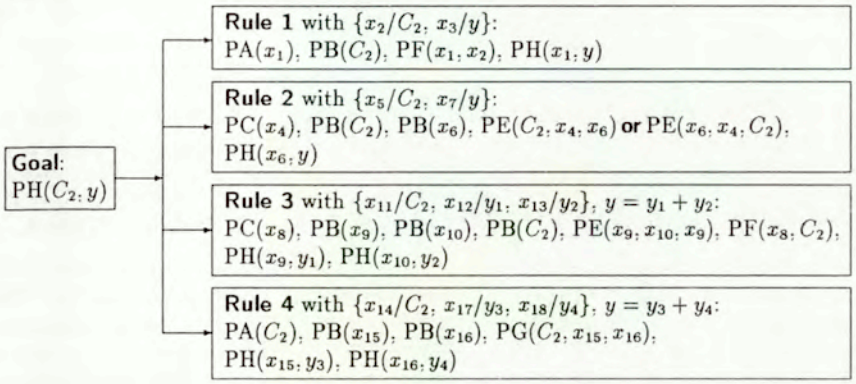
Finding an appropriate substitution in the first step of the procedure above is usually called the *unification problem* or *matching problem* and constitutes one of the two main problems in using this representation, the other one being the selection of the best rule to consider at every step in the process.

The appropriate sequence of rule applications can be sought of in several different ways. The two main types of methods used here are called *forward chaining* and *backward chaining*. With forward chaining, we proceed from facts toward the goal, adding new, derived facts at every step with the hope that one of them will finally match the goal formula. With backward chaining, we proceed from the goal formula, applying rules in reverse direction and thus producing new (sub)goals (in practice, a whole so-called *and/or tree* of subgoals), in the hope that finally the appropriate subgoals will math the facts. The tree can be partially pruned during the process when at some step a formula contradictory to the facts is produced. Additional problem in this method is the selection of a subgoal to be expanded next into a subtree (by a backward rule application). Many systems combine in some way both the above approaches.

Example II.3b (Forward and backward chaining) Let us illustrate and clarify the procedures described above using our example again.

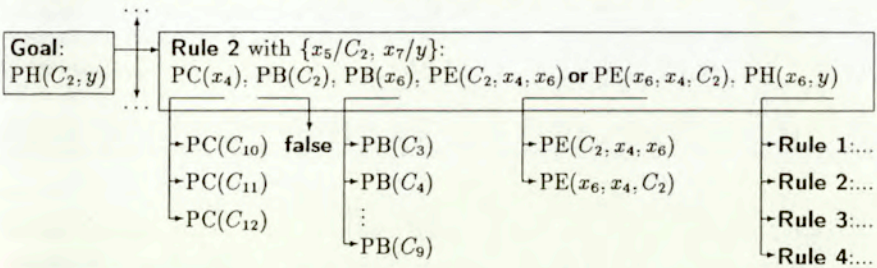
With forward chaining, the inspection of the search space for possible valuations reveals, after considerable search, that only the first rule can be applied to the facts. The substitution $\{x_1/C_1,\ x_2/C_3,\ x_3/1\}$ (meaning "substitute C_1 for x_1, C_3 for x_2, and 1 for x_3"), applied to **Rule 1** produces **if** $\mathrm{PA}(C_1)$, $\mathrm{PB}(C_3)$, $\mathrm{PF}(C_1, C_3)$, $\mathrm{PH}(C_1, 1)$ **then** $\mathrm{PH}(C_3, 1)$. Because all the terms in the antecedent match exactly the initial facts, the application of the rule yields a derived fact $\mathrm{PH}(C_3, 1)$ to be added to the knowledge base. Similarly, at the next step also only one rule can be applied (**Rule 2**), producing the fact $\mathrm{PH}(C_4, 1)$, etc. Only at the fifth step two rules are applicable, but only one of them lies on the way to the goal formula $\mathrm{PH}(C_2, 5)$ with the final valuation $\{y/5\}$.

With backward chaining, we can see that at the first step all the rules can be applied to the goal formula, producing the first part of the subgoals tree—an *or*-type root node with the original goal formula and four *and*-type nodes produced by backward applications of the four rules:



Note the introduction of new variables y_i that are needed to handle properly the calculation of the final value of the existentially quantified "answer variable" y.

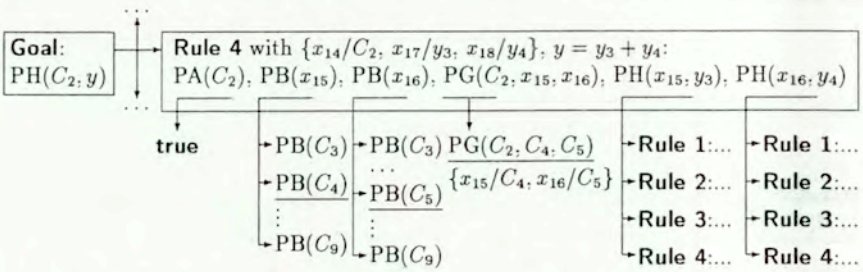
The *or*-node can be pruned only if all its descendant nodes were pruned, while the *and*-node can be pruned if at least one of its descendant nodes was pruned. To show the process, let us expand the node produced by the backward application of **Rule 2**:



The descendants of the *and*-node (here, the components of the conjunction associated with the node), can be expanded further either by a backward application of some rule or by matching them with the base of facts. As seen above, the expansion of a component produces usually an *or*-node, whose descendants are the matching facts (as for the nodes obtained from the predicates PC(x_4) and PB(x_6)), or antecedents of the applicable rules (as with the PH(x_6, y) predicate), or components of the disjunctive subformula (here PE(C_2, x_4, x_6) **or** PE(x_6, x_4, C_2)).

However, the component PB(C_2) cannot be further expanded in either way; moreover, the fact it represents does not occur in the base of facts, so that, according to the negation by omission rule, it is considered incompatible with the data. Therefore, it can be pruned from the tree, and because it is a descendant of the *and*-node, the whole *and*-node

produced by the **Rule 2** can be pruned too. Because the same component $\mathrm{PB}(C_2)$ occurs also in the *and*-nodes produced by **Rule 1** and **Rule 3**, they also can be pruned. This leaves only the node produced by **Rule 4** for further expansion, which will look as follows:



Now $\mathrm{PA}(C_2)$ is contained in the base of facts, while $\mathrm{PG}(C_2, x_{15}, x_{16})$ can match only the single fact $\mathrm{PG}(C_2, C_4, C_5)$ which results in the substitution $\{x_{15}/C_4, x_{16}/C_5\}$. That substitution causes in turn the predicates $\mathrm{PB}(x_{15})$ and $\mathrm{PB}(x_{16})$ to match appropriate facts as shown, and allows for expanding further the node by a backward application of the rules to remaining components of the conjunction, namely the predicates $\mathrm{PH}(C_4, y_3)$ and $\mathrm{PH}(C_5, y_4)$. Proceeding in such a manner, we finally find the full sequence linking the goal with the facts and will be able to calculate the value of y following backward the obtained sequence of substitutions for the y_i variables. ■

II.1.2.2 Problems with logical representation

... if it was so, it might be; and if it were so, it would be;
but as it isn't, it ain't. That's logic.
[Lewis Carroll, *Through the Looking Glass* (1871)]

As the patient reader might observe, solving the example problem in the representation described above is extremely tedious for a human. The main reasons may be formulated as follows:

Meaninglessness. There is no indication concerning the *sense* of the problem—an ingredient very important for a human. Even worse, any explanation of the nature (say, physical, or specifically, mechanical in this case) of the problem lies completely outside this kind of representation, as it even cannot be sensibly stated within it.

Mechanicalness. As a result, transformations of the representation leading to the solution are purely formal and hence mechanical, giving little, if any, guidance to the human solver concerning the direction of the process, distance to the goal, etc., or warning of possible errors made in the process (see below).

Confusing names. One of the elements contributing to that meaninglessness are the standard, formal names of predicates, constants (objects), and variables. They do not convey any feeling of the nature of the objects denoted and are easy to confuse with each other.

Many combinations. Despite the relative simplicity of the problem, the number of alternative variants to consider at every step (e.g., the number of possible substitutions) is quite large, rather unwieldy for a human reasoner. For more complex practical problems it can easily become unwieldy for a computer as well.

Error-proneness. The inevitable result of the above is the great possibility of making various types of hard to detect errors, not only at the stage of solving the problem, but also during encoding it into this kind of representation.

All the above indicates that this kind of representation is not very suitable for use by humans. There are some techniques, as mentioned above, located mostly outside the representation itself, and partially alleviating these problems. Also, there are techniques of easing the search space problems in a computer implementation of the method. All these tricks in effect replace the original logical representation by some other, hybrid representation of which the logical formulae and formal transformations constitute only a part. We will discuss some of these problems and techniques in more detail below.

Formality and meaninglessness. The predicate calculus technique described above is purposefully made strictly formal, which, allegedly, requires that it should be devoid of and independent of any meaning of the underlying concepts or domain objects, see Pasch's quote in Section II.5.3. Therefore, any attempt to add somehow that meaning would in fact violate the very idea of this representation. Any such attempt would thus necessarily lie outside the representation itself, as a sort of external decoration making it easier to comprehend by a human, but in no way influencing the nature of the representation or the solution procedure. Nevertheless, the need caused several such techniques of "humanization" of this representation to be developed and commonly used.

Meaningful names. Possibly the simplest such technique is the use of more meaningful names for predicates, constants and variables.

Example II.3c (With meaningful names) The version of a statement of our mysterious mechanical example using more meaningful names is shown in Table II.2. Note that still the identity (hence, the meaning) of the example remains rather obscure to most readers unfamiliar with the full description of the example. ■

Work with such a form of the problem seems easier and more "user-friendly", so to speak, for a human, though its solution by hand would actually require more writing. It would be useful also in the case of computer implementation, easing the process of formulation of the problem, its input to the computer, and understanding of the results and explanations produced by the computer during the solution process.

This line of approach led to the use of a restricted "natural language" formulation and input of facts and rules to systems based on this representation [27]. For example, the input form of some rules concerning truss structures in such a system might look like that in Table II.3 (see [10, 14]).

Note, however, that using long, semi-natural names is impractical, especially for larger problems, while shorter names, like that in Table II.2, are still rather obscure, both to the originator of the problem (especially after some time away from it) and especially to

Table II.2: A logical representation example with "meaningful" names (cf. Table II.1).

FACTS: structural description and object properties

WT(P), WT(Q),
RP(Rp), RP(Rq), RP(Rs), RP(Rt), RP(Rx), RP(Ry), RP(Rz),
PL(A), PL(B), PL(C),
CL(T),
PLS(Rp, A, Rq), PLS(Rx, B, Ry), PLS(Ry, C, Rz),
HNG1(P, Rp), HNG1(A, Rx), HNG1(B, Rt), HNG1(C, Rs),
HNG2(Q, Rq, Rs), HNG2(T, Rt, Rz),
VAL($P, 1$)

RULES: how to derive new facts from the old ones

Rule 1:
if WT(w_1), RP(r_1), HNG1(w_1, r_1), VAL(w_1, n_1)
then VAL(r_1, n_1)

Rule 2:
if PL(p_1), RP(r_2), RP(r_3), PLS(r_2, p_1, r_3) **or** PLS(r_3, p_1, r_2), VAL(r_3, n_2)
then VAL(r_2, n_2)

Rule 3:
if PL(p_2), RP(r_4), RP(r_5), RP(r_6), PLS(r_4, p_2, r_5), HNG1(p_2, r_6),
VAL(r_4, n_3), VAL(r_5, n_4)
then VAL($r_6, n_3 + n_4$)

Rule 4:
if WT(w_2), RP(r_7), RP(r_8), HNG2(w_2, r_7, r_8), VAL(r_7, n_5), VAL(r_8, n_6)
then VAL($w_2, n_5 + n_6$)

GOAL: find a value for n such that...

VAL($Q, ?n$)

Table II.3: Example rules in stylised "natural language" form, cf. [10, 14].

Rule 1: Elimination of two bars		**Rule C:** Equilibrium of two bars	
if	number of bars at node is 2	**if**	number of bars at node is 2
and	support is none	**and**	support is none
and	external load is none	**and**	external load is none
and	bars (b_1, b_2) are not collinear	**and**	bars (b_1, b_2) are collinear
then	eliminate bars (b_1, b_2)	**and**	bar b_1 is stretched
and	eliminate node	**then**	bar b_2 is stretched

other people. Short or long, the names by themselves usually are not meaningful enough without some other, more directly meaningful representation, e.g. a drawing. It is worth to recall here again the work of [Amarel 1968], where starting from purely predicate calculus representation, a sequence of more and more efficient and transparent representations has been constructed. These further representations were also more meaningful and contained a more and more pronounced diagrammatic component (see also the next section).

Large search spaces. In order to check if the given rule is applicable in the current state of the knowledge base, one must find in the base of facts all atomic formulae that match atomic formulae in the antecedent of the rule for some possible substitutions of terms (constants in the case of our running example). With an unstructured set of facts (which is the case with a pure predicate logic representation) it involves search within usually very large space of possibilities.

Example II.3d (Search space: unstructured) In the example described by Table II.1. there are 23 atomic formulae in the base of facts. Checking the applicability of the **Rule 1** with its four atomic formulae in the antecedent therefore requires searching through $23^4 = 279\ 841$ possible combinations of facts. As there are two facts involving the PA predicate, seven for PB, etc., the search produces a reduced subspace of $2 \cdot 7 \cdot 4 \cdot 1 = 56$ combinations of facts that must be then checked for consistency of assignments of constants to the three variables that occur in the antecedent. For the **Rule 3**, with its eight atomic formulae in the antecedent, the numbers would be $23^8 \approx 78.3 \cdot 10^9$ and $3 \cdot 7 \cdot 7 \cdot 7 \cdot 3 \cdot 4 \cdot 1 \cdot 1 = 12\ 348$ respectively. ■

These numbers, especially the sizes of the initial search spaces, are quite large even for computers, considering the simplicity of the example as compared to real life problems that may easily involve thousands of facts and rules. Therefore, the knowledge bases of large problems are structured in a way facilitating the search. The structuring takes the form of various indexes listing facts according to their predicates, constants or terms occurring as their arguments, etc. Although that usually considerably reduces the sizes of search spaces involved, they still remain rather large, limiting the size of problems that can be solved in a reasonable time.

Example II.3e (Search space: structured) Let us introduce some such structuring to the example. After listing the atomic formulae in the base of facts according to their predicates, as it was already done in Table II.1, the search for a fact will now involve, first, searching for the name of the predicate (within the space of eight possibilities), and next the enumeration of only the facts within the already selected set of facts containing this predicate name. Thus, for the **Rule 1** we must now search through only $(8+2) \cdot (8+7) \cdot (8+4) \cdot (8+1) = 16\ 200$ possible combinations to select the 56 combinations for final checking of variable assignments. For the **Rule 3**, the number would be $(8+3) \cdot (8+7) \cdot (8+7) \cdot (8+7) \cdot (8+3) \cdot (8+4) \cdot (8+1) \cdot (8+1) = 396\ 940\ 500$, an improvement by a factor of almost 200 with respect to the unstructured case, but still rather large. ■

Such structuring of the knowledge base converts it effectively into a kind of graph, which can be interpreted as an internal form of some diagram, see Section II.6.2.2 and Example II.3i there. Recall also a sequence of representations by [Amarel 1968] mentioned at the end of the previous section. One may thus claim that practical knowledge bases using the logical representation are actually in most cases diagram-aided...

II.1.2.3 Perceptual rules

You certainly usually find something, if you look,
but it is not always quite the something you were after.
[John R.R. Tolkien, *The Hobbit* (1937)]

An interesting feature of logical representations, both contrasting and linking them with analogical (especially diagrammatic) representations is the occurrence of so-called *perceptual rules*. They are used to recognize certain important patterns or substructures in the knowledge base and name them as new properties or relations to be used as single predicate expressions in other rules. It is possible to avoid their use at the cost of detecting and encoding "by hand" all such necessary configurations as single predicate expressions during preparation of the knowledge base of the system. This, however, usually leads to a large base of facts, with the possibility that many of them may be not needed, or that some important configurations may get overlooked. Inclusion of only basic facts in the knowledge base with a bunch of perceptual rules to detect certain substructures when needed is often more advisable, though it complicates the work of the system during problem solving by enlarging ever more the search space needed to find a route to the goal formula.

Example II.3f (Perceptual rules) In our mechanical example, the two kinds of the HNG*i* predicate, see Table II.2, are an obvious target for using perceptual rules. The appropriate fragment of the knowledge base using them is shown in Table II.4. The base of facts contains there a single generic HNG predicate augmented with two perceptual rules recognizing the patterns with single and double associations of a RP-type object with some other object in the arguments of the HNG predicate. As can be easily checked, all possible applications of the perceptual rules will produce exactly the set of HNG1 and HNG2 predicate expressions occurring in Table II.2, as required. ■

Table II.4: Perceptual rules for the HNG1 and HNG2 predicates (cf. Table II.2).

FACTS: structural description and object properties

...
HNG(P, Rp), HNG(A, Rx), HNG(B, Rt), HNG(C, Rs),
HNG(Q, Rq), HNG(Q, Rs), HNG(T, Rt), HNG(T, Rz),
...

PERCEPTUAL RULES: recognizing meaningful patterns

Rule HNG1:
if HNG(x_1, r_1), $(\forall r_2)(r_1 = r_2$ **or not** HNG(x_1, r_2))
then HNG1(x_1, r_1)
Rule HNG2:
if HNG(x_1, r_1), HNG(x_1, r_2), $(\forall r_3)(r_1 = r_3$ **or** $r_2 = r_3$ **or not** HNG(x_1, r_3))
then HNG2(x_1, r_1, r_2)

As can be seen in Table II.4, the application of the perceptual rules shown there is rather costly, as it involves search of a substantial part of the base of facts to assure that there are no other associations than necessary of the x_1 object with other objects in arguments of the HNG predicate. This is often the case with perceptual rules: they tend to be rather complicated and costly in practical applications. This stays in contrast with diagrammatic representations, where such perceptual inferences are much easier, especially if the representation is to be used by humans, see Section II.3, where this example is represented in diagrammatic form.

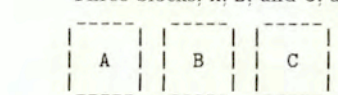
II.1.2.4 Only logical framework?

... the picture will always reward you
by bringing you nearer to the Truth.
[Tristan Needham, *Visual Complex Analysis* (1997)]

When the logical representations became fashionable in AI, many researchers even claimed that it is the only possible type of representation for computer implementations of knowledge bases and machine reasoning using them. An instructive analysis of an example of such a claim (made by [Moore, R.C. 1982][4]) follows. In that paper, its author fervently defended the thesis that some kinds of reasoning "can be implemented only within a logical framework,"[5] but illustrated it by an example using a diagram as an essential part of its statement:

Example II.4a (Three blocks) An exact quote from [Moore, R.C. 1982] goes as follows:

"Three blocks, A, B, and C, are arranged as shown:



A is green, C is blue, and the colour of B is unstated. In this arrangement of blocks, is there a green block next to a block that is not green?" ■

Moore then asserts that the answer to the posed problem "should be clear with no more than a moment's reflection" and gives it in natural language, without producing any logical procedure of actually solving the problem. Then he claims that a person solving this problem will use such "logical factors" as "the ability to see that an existentially quantified proposition is true." However, it seems highly doubtful whether solving this problem by a human with only "a moment's reflection" can be notably aided by using any existential quantifiers or other predicate calculus notation. The example can be of course stated using predicate calculus representation and solved within it, but the process will require significantly more than only "a moment's reflection," as will be shown below.

[4]Not to be confused with Ramon E. Moore, the author of the basic book on interval arithmetic [Moore, R.E. 1966], see Chapter III.

[5]All phrases in quotes in this subsection are quotes from the paper [Moore, R.C. 1982].

Example II.4b (Logical formulation) In predicate calculus there are no blocks and their arrangements, nor their colours. There are only abstract constants, variables, and predicates. Hence, the example should be first appropriately translated into the logical framework. The first attempt at such a translation may look like that:

> There are three objects, A, B, and C. A relation P holds between A and B as well as between B and C. The object A has property Q, and the object C has property R. Is there an object with property Q which is in the relation P with another object without that property?

On closer scrutiny, it becomes clear that such a formulation does not allow to solve the problem at all, because there is no possibility to ascertain that the object C does *not* have the property Q.[6] What has been omitted from the original formulation? A single important thing—the unstated, implicit properties of colours, namely that when something is blue, it cannot be green.[7] Note that this condition cannot be purely logically extracted from the abstract formulation of the problem, as it is the *physical* property of colours in this particular situation.[8] Therefore, the rigorous statement of the problem should look rather like that:

> There are three objects, A, B, and C. A relation P holds between A and B as well as between B and C. The object A has property Q, and the object C has property R. It is understood that when an object has property R, it cannot have property Q. Is there an object with property Q which is in the relation P with another object without that property?

Now we are at last ready to state formally the problem in the language of logic, see Table II.5 (with **and** operators replaced by commas for simplicity). ■

Table II.5: The three blocks example: formal predicate calculus statement.

Facts:
$\mathrm{P}(A,B),\ \mathrm{P}(B,C),\ \mathrm{Q}(A),\ \mathrm{R}(C)$
Rule:
$(\forall x)\,(\mathrm{R}(x)\,\textbf{implies not}\,\mathrm{Q}(x))$
Goal:
$(\exists y,z)\,(\mathrm{P}(y,z),\ \mathrm{Q}(y),\ \textbf{not}\,\mathrm{Q}(z))$

Arriving thus to the realm of formal and pure logic (almost), we see that despite that, solving the problem does not seem to require merely "no more than a moment's reflection."

[6] One may propose to use here the *negation by omission* rule, see Section II.1.2. However, it would dictate both **not** $\mathrm{Q}(B)$ and **not** $\mathrm{R}(B)$, contrary to the intended sense of the original problem, so it cannot be applied here.

[7] At least in the particular, also implicit, context of this example: in real life, objects partially blue and partially green, or of some intermediate shade of blue-green, can occur as well.

[8] Again, because of the lack of indications to the contrary, the simplest colour ontology is implicitly assumed, cf. the previous footnote.

Solving it in this form requires quite a lot of formula rewriting, following closely a set of not-too-obvious rules. Formal solution of this problem can be used to illustrate a popular technique of automatic theorem proving called the *resolution method*, some time ago considered the best that may ever happen to computer implementors of logical reasoning techniques.

The resolution method is based on two principles:

Proof by contradiction. Here it means that as a first step a negation of the goal is produced and added to the knowledge base. The proof is now conducted by trying to show that this leads to contradiction, i.e., that the resulting formula (consisting of the whole new knowledge base) is false.

The resolution principle. This means the use of the following logical tautology:

$$\begin{aligned}&(A_1 \textbf{ or } A_2 \textbf{ or } \ldots \textbf{ or } B) \textbf{ and not} B = \\ &= (A_1 \textbf{ or } A_2 \textbf{ or } \ldots) \textbf{ and not} B = \\ &= (A_1 \textbf{ or } A_2 \textbf{ or } \ldots \textbf{ or } B) \textbf{ and not} B \textbf{ and } (A_1 \textbf{ or } A_2 \textbf{ or } \ldots).\end{aligned}$$

Thus, after the knowledge base (with the negated goal included) is transformed to the form of a conjunction of disjunctions of atomic formulae (possibly negated)—which is always possible—it can be transformed further equivalently by adding new simpler disjunctions according to the resolution principle. As a result, when in this manner an empty disjunction is produced (which has a logical value false: B **and not** B = **false**) we obtain a contradiction, thus proving that the goal follows from the knowledge base. It follows that generally resolution is a kind of forward chaining strategy, and like with other logical methods, the main problem here is the selection of the appropriate sequence of resolution steps with appropriate choice of substitutions at every step.

Example II.4c (Solution by resolution) The solution of the three blocks problem using the method of resolution is presented in Table II.6. The knowledge base, by default universally quantified over all variables, is presented as a list (meaning conjunction) of disjunctions (here commas replace **or** connectives), numbered from 1 to 5. Note at the position 5 the rule converted to a disjunction. Next, the base is augmented by the negated goal, also converted to a disjunction (position 6). The existentially quantified goal changes here into a universally quantified formula due to negation.

In subsequent steps, the resolution principle is applied several times to the indicated disjunctions, after an appropriate substitutions are made for the unification of terms. In the final step an empty disjunction is produced, hence the possibility of validating the goal statement is proven. The substitutions made on the way provide the conditions under which the goal becomes valid—the two objects having different values of the property Q but being in relation P with each other can be either the objects A and B, or otherwise B and C. ■

In another place of the paper [Moore, R.C. 1982], its author states that solving of the example needs logic because we have incomplete description of the situation: we do not know the colour of the middle block and do not know which blocks are to be tested to satisfy the given condition. This is one of the standard arguments against diagrams, called the *incomplete information* problem, or more specifically, its special case—the problem of representation of *disjunctive knowledge*. These arguments are discussed in

Table II.6: The three blocks example: solution by resolution method.

Solution:

1.	$\mathrm{P}(A, B)$	– *fact*
2.	$\mathrm{P}(B, C)$	– *fact*
3.	$\mathrm{Q}(A)$	– *fact*
4.	$\mathrm{R}(C)$	– *fact*
5.	**not** $\mathrm{Q}(x)$, **not** $\mathrm{R}(x)$	– *rule*
6.	**not** $\mathrm{P}(y, z)$, **not** $\mathrm{Q}(y)$, $\mathrm{Q}(z)$	– *negated goal*
.....		
7.	**not** $\mathrm{Q}(C)$	– 4, 5 $\{x/C\}$
8.	**not** $\mathrm{Q}(A)$, $\mathrm{Q}(B)$	– 1, 6 $\{y/A, z/B\}$
9.	$\mathrm{Q}(B)$	– 3, 8
10.	**not** $\mathrm{Q}(B)$, $\mathrm{Q}(C)$	– 2, 6 $\{y/B, z/C\}$
11.	**not** $\mathrm{Q}(B)$	– 4, 10
12.	—	– 9, 11 (*contradiction*)

detail in Section II.3.2.2, where also a simple diagrammatic solution to this three-blocks problem is given (Example II.4d), without any trouble caused by the incompleteness of the information provided in the example. Note also that in the logical solution above there is no meaningful way of finding which blocks should be tested: the only way offered by the method to find appropriate substitutions is by exhaustive search.

II.1.3 Diagrammatic representation

[Diagrams are] naturally analogous to the thing represented.
[Charles S. Peirce, *The Collected Papers of C. S. Peirce* (1839–1914)]

What is exactly meant by *diagrammatic representation* is not yet fully agreed even between researchers in the field of diagrammatics themselves. Discussions and controversies persist—see especially the series of books and article collections grouped at the beginning of the bibliography. Here we may start with the following tentative definition.

Definition II.3 (Diagrammatic = analogical + visual) *Diagrammatic representation is an* analogical (direct, homomorphic) representation *(i.e., such that its syntax models the semantics of the problem domain) presented by* visual means, *of some (not necessarily visual) data or knowledge.*

One may consider the definition to be too broad, e.g. because it seemingly includes such things as photographs, while traditionally diagrams were identified rather with some linear abstract drawings. Others may complain that it may be too narrow, as one may imagine diagrams in which the analogical component seems to be quite small, like with some abstract graphs of functional dependencies. However, it seems that this definition captures the core idea of diagrammaticity quite well, so that it gives an easy to comprehend general feeling of the subject matter. For the few border cases one may always provide some more specific subdefinitions in case of real need.

There are other, more specific definitions of the term. A fairly comprehensive survey of the most important approaches to distinguish diagrammatic representations from propositional ones can be found in [Shimojima 2001]. In this paper, Shimojima also advocated his criterion which basically boils down to calling diagrammatic those representations that exhibit emergence effect, see Section II.4.2. For our purposes here, the following definition quoted from [Stenning & Lemon 2001] will be also of interest. It is more narrow than Definition II.3 above, and is based on the notion of *directness* of the representation (discussed further in Section II.3.1.3).

Definition II.4 (Diagrammatic = planar + direct) *Diagrammatic representation is a* plane structure *in which representing tokens are objects whose mutual spatial and graphical relations are* directly *interpreted as relations in the target structure.*

For better understanding of those definitions and related notions, some terminological clarification is due. First, several common terms, often interchangeably used in the place of the term "diagrammatic," should be clearly differentiated. In this work, when used in more formal context, they should be understood according to the following specifications. The first of them is:

Visual means *perceivable by sight*, thus, generally at least two-dimensional (even when the concepts represented are one-dimensional, like time intervals).

"Visual" does not necessarily mean analogical. The visual/non-visual distinction is to a large extent independent of the analogical/propositional distinction. It refers to *sensual modality* of the representation (in opposition to *aural*, *tactile*, etc.), not to the structure of the encoding involved in the analogical/propositional distinction. Thus, both a diagram and a set of formulae (if written on paper or displayed on a screen) are visual representations. The two main subspecies of "visual" are:

Pictorial means *involving pictures*, i.e. realistic (representational) images of world objects and scenes (like in figurative painting, photography, etc.).

Graphical means *involving symbolic or simplified* visual depictions of objects, notions, etc. (like in drawings, schemata, information signs, etc.).

Historically, the distinction between "pictorial" and "graphical" also carried a connotation of restricted means of rendering (for graphical type—only black and white, with no intermediate tones or colour, only line drawings, etc.), now mostly irrelevant due to proliferation of advanced graphics techniques in art, design, printing industry, and computer graphics. The distinction is currently a matter of degree rather than principle. It is also different, though related, to the distinction observed in visual arts between *painting* and *graphics*. These art terms correspond not so much to the specific visual features of the artifact or the intent of its creation, but rather denote the *technique* of its creation. Namely, painting means the execution by hand producing only one original, while graphics denotes there the use of more or less mechanical means of reproduction capable of producing multiple originals.

Often used to characterize diagrams, the term:

Spatial is rather ambiguously meant once as *more than one-dimensional*, but in other cases, as *more than two-dimensional*.

The former meaning of the term is usually understood in the context of diagrammatic representations, so that the term is often used to contrast diagrams, characterized by their *spatiality*, with only *linear* propositional representations, see Definition II.4 and especially Section II.3.1.2.

The next term of interest here is:

Geometrical means *involving geometry*, i.e., a branch of mathematics studying some abstract mathematical objects abiding in a metric space, especially their properties invariant with respect to distance-preserving transformations.

It is a quite common practice to identify geometrical objects with their renderings (drawings). That may, however, lead to confusion, with often detrimental consequences (cf. Sections II.3.1.3 and II.4.2.2). It should be stressed here that the study of geometry (or any things geometrical) does not have to use diagrams (drawings) at all. Thus, *geometrical* does not bear any specific relation to *diagrammatic*. Also, the term *geometrical interpretation* often used in such contexts needs some extra explanation. The term is in itself rather confusing—for the sake of precision, it should have been rather changed to *geometrical representation*. Anyway, it is meant to denote an isomorphic rendering of some set of objects or notions (usually, of mathematical or some other abstract nature) into geometrical terms. Again, this may, but as well may not, involve diagrams. It will lead to some diagrammatic representation if the geometrical model obtained will in turn be represented using some (geometrical) diagrams.

It is also important to note that diagrams, even geometrical diagrams, can (and quite often do) contain visual language elements that are not geometrical (i.e., do not represent any geometrical objects or notions). Consider for example such elements as arrowheads, various thickness or dot patterns of lines, colour, shading or cross-hatching of areas, etc., see Section II.2 for more detailed discussion of the matter.

To put these considerations in a practical setting, a simple example (taken from [44]) may be handy.

Example II.5 (Three representations of a set) In Fig. II.3, three different representations of some set called S are shown. The first representation (Fig. II.3a) is a description in (mostly) natural language (hence *propositional*), but, as we see, in geometrical terms—it mentions points, circles, coordinate axes, etc. Hence, the representation is also *geometrical*, though it does not involve nor contain any diagram (geometrical or not). At the same time the representation is *visual*, as being written on paper for perception through the eyes. It would instead become aural when read aloud. The second representation (Fig. II.3b) is given as an algebraic formula, and is hence *propositional*, *algebraic*, and again *visual*. The third one (Fig. II.3c) involves a (geometrical) drawing, representing the structure (especially the metric one) of the set S *analogically*. Hence, as being at the same time *visual*, it may be called *diagrammatic*. Moreover, it is obviously *geometrical* as well, and concerning its drawing style—*graphical*.

a) *S* is the set of points lying inside of a unit circle centered at the point on the horizontal axis with the coordinate equal to 1.

b) $S = \{ (x, y) \mid x^2 - 2x + y^2 < 0 \}$

c)

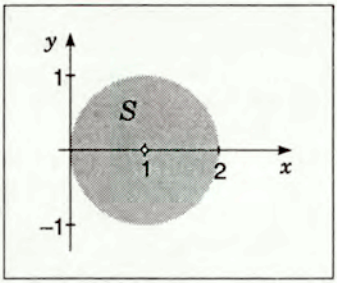


Figure II.3: Three representations of a set: propositional geometric (a); propositional algebraic (b); and diagrammatic (c).

Note also that the diagram contains elements of visual language that do not have geometrical meaning, like arrows, shading of the interior of the circle (note also that the lack of explicit contour of the circle serves as a visual statement that the circumference of the circle does not belong to the set S), small circle denoting a point (circle centre), tickmarks on the axes, and letter and number labels (that constitute a propositional component of the representation, making it in fact a hybrid one). Also, the lines representing axes are not fully direct representations of the geometric entity called "straight line," as the latter is infinite, which is not directly representable on a real diagram (see Section II.3.1.3 for further discussion of this issue). ■

II.1.4 The field of diagrammatics

Probably the shortest possible definition of diagrammatics can be formulated as:

Definition II.5 (Field of diagrammatics) *Diagrammatics is the field of study of diagrammatic representations.*

The above definition fixes also the *subject* of diagrammatics. Much more trouble comes from an attempt to delineate the *scope* of this new, emerging discipline. The problem comes mostly from the question of specifying what should be considered a diagrammatic representation, as discussed in the previous sections. An all-inclusive formulation claims for diagrammatics all graphical and pictorial representations together with techniques of their production and use. This is probably an impractically extreme view, making unexpectedly diagrammatics researchers from too many scholars and practitioners of long ago established disciplines, from psychology of vision and graphic design to computer graphics. Another extreme classifies as diagrams only some kind of graphical representations, like those used in mathematics and some other related disciplines of science, like physics, excluding even such representations as maps and statistical graphs. Some intermediate stance would be obviously more practical, but this author is not in a position to specify the exact boundary of the field, leaving that to future development of the discipline. Avoiding in this way the decision which of the old disciplines concerned with graphical representations should or should not be considered as parts of diagrammatics, one may

safely say that diagrammatics nevertheless does, and should, use the findings and results of all those old disciplines as a source and foundation of diagrammatics research.

From the methodological point of view, diagrammatics can be divided into three basic types of research:

Cognitive and psychological research concerns psychological mechanisms of understanding, producing, and generally using diagrams by human beings, including the search for principles and rules that may make diagrams more effective and easy to use by humans.

Theory of diagrammatic representations builds and investigates theoretical (especially formalized) models of diagrammatic representations, visual languages, their various properties, and the methodology of their use to represent, communicate, and process information and knowledge.

Applications of diagrams cover analysis, improvement and design of various systems of diagrammatic notations for various applications, including problems and methods of their computer implementation.

In this work, the first of these areas is not explicitly pursued, although some basic findings are implicitly touched on in several places in Chapter II. Basic issues of diagrams theory constitute the main material of Chapter II, albeit treated there mostly informally, while Chapter III, where a diagrammatic notation for interval algebra is developed, belongs fully to the third area of research, as does Section II.6 on computer implementation.

One may formulate many goals of research in diagrammatics, from very general ones, like gathering all possible knowledge on the subject, to particular ones, like explaining how this or that particular effect or diagram works. In the context of this work, one particular goal, involving theoretical and application-oriented research, should be stressed:

The diagrammatics challenge: *To formalize knowledge about diagrammatic representations and reasoning to such an extent as to make possible its wide computer implementation, allowing for:*

- *Computer-aided or fully automatic design of diagrammatic representations of extensive varieties of data and knowledge.*
- *Efficient diagrammatic reasoning (data and knowledge processing), by human-computer systems, using diagrammatic representations.*

II.2 Visual languages

Do you know languages?
What's the French for fiddle-de-dee?
[Lewis Carroll, *Through the Looking Glass* (1871)]

In order to represent anything in a diagram, or to read off the diagram the represented information, one must have an encoding system with which to construct or interpret a diagram in terms of appropriate building blocks combined with appropriate composition and interpretation rules. In other words, one must have a *visual language*.

Definition II.6 (Visual language) *A* visual language *is an encoding system prescribing in what manner graphical (pictorial, diagrammatic) primitives should be composed in order to produce diagrammatic representations for some kind of data (information, knowledge), and in what manner the diagram should be interpreted in order to read off the data (information, knowledge) encoded in it.*

It should be stressed here that the full specification of a visual language must contain both the rules for constructing diagrams and the rules for interpreting them. The latter are seldom simple inversions of the construction rules—consider for example the problems of spatial interpretations of flat projections discussed in Section I.5. Neglecting to specify the interpretation rules may lead to misunderstanding, often caused by using rules from some other visual language, more familiar to the reader.

Various, mostly uncodified, visual languages were in use since human beings started to make pictures. Informal attempts to codify visual languages originated only quite recently in the graphic design community, starting from the birth of statistical graphs in XIX century [Tufte 1983, Hankins 1999], and culminating in works of modern graphic designers and statisticians, like [Bowman 1968, Bertin 1967/83, Dondis 1975] and others. A recent work of [Engelhardt 2002] excellently systematizes and unifies most of these partial approaches, putting the research on visual languages on a more ordered and firm conceptual foundations.

It is important to note here that despite the frequent use in these works of the phrases like "*the* language of graphics," there is no single universal visual language. Different languages are constructed for different purposes and different represented domains. In different languages the same visual tokens or relations may have quite different, sometimes even opposite meaning. The language may be also hybrid, as shown in Section II.1.1 (Example II.2), containing both diagrammatic and propositional elements in various degrees. This is important also because quite often certain findings concerning properties (usually limitations) of some particular visual language are presented as applying universally to all diagrammatic representations. The most important such misunderstandings concerning the alleged general limitations of diagrammatic representations are discussed in some detail in Section II.3.2.

Some work on visual languages for description of complex pictures has been done as well in the context of image analysis and pattern recognition, see e.g. [Narasimhan 1964, Fu 1982] and [23, 39, 76]. However, sufficiently rigorous and formal definitions of visual languages appeared only recently, within the borders of diagrammatics [Hammer 1996, Luengo 1995, Miller 2001], or in the context of visual computer interfaces and pro-

gramming languages [VISLANG 1998]. In fact, the study of visual languages constitutes one of the main branches (if not *the* main branch) of diagrammatics. In spite of its importance, it is still in its infancy, due to the complexity of the issue (visual languages have much richer structure than ordinary ones), and a rather short length of time since its more serious study has been attempted. A very general introduction to this fundamental issue is given in this section. The remaining sections of this chapter (except possibly Section II.6 on computer implementation of diagrams) in fact concern (mostly) various semantic and pragmatic issues of visual languages studied in diagrammatics (see also [Gurr 1999]). Section II.5.4 contains also a review of the specific visual language styles used in mathematics.

An important subset of visual languages, vigorously investigated and developed since the eighties, comprises languages for visual programming, software visualization, visual database access, and the like—generally, visual languages used in computation [Myers 1990]. Their proper name of "visual *programming* languages" is often abbreviated to "visual languages" only, which may lead to confusion. The field has already a well-established conference and a scientific journal (publishing papers on other subjects of diagrammatics as well, see e.g. [Gurr 1999, Wang & Lee 1993]). Results and knowledge gained within this research, see e.g. [Shu 1988, VISPROG 1990, VISLANG 1990a, VISLANG 1990b, VISLANG 1998], is of great importance to visual language research in particular, and diagrammatics in general. Somehow, however, these results are seldom used or even noticed by the core diagrammatics community. E.g., the recent systematization of issues of visual languages done in the work [Engelhardt 2002] mentioned before does not mention visual programming languages at all.

The structure of visual languages used in diagrammatic representations is essentially different than the structure of one-dimensional propositional representations (with possible exception of some borderline cases like certain "one-dimensional diagrams"). Therefore, translation from a propositional statement into a diagrammatic one cannot be made "word for word," and much more so than in the case of natural languages. The representation must be usually restated in a completely different manner, conforming to the specific nature of diagrammatic representation.

Despite these essential differences between two-dimensional visual languages and one-dimensional ordinary languages, many notions developed in linguistics are still used, with due modifications, in the study of visual languages, in want of possibly more appropriate notions. Most of all, the basic way of structuring specifications and uses of languages is generally preserved, namely, a visual language is considered to consist of:

Visual vocabulary which contains *pictorial* (like icons) and *graphical* (like drawings) *elements* (called also *primitives*) and their visual *properties* that can be used to encode pieces of data.

Composition rules that govern how the elements of the vocabulary may and should be combined to represent the required data (information, knowledge). The rules are further divided, following again the example from linguistics, into three main classes, called by the familiar terms:

Syntax: prescribing allowable composition of the visual *elements* using allowable *relations* between them.

Semantics: prescribing how the *elements*, their *properties* and *relations* should be chosen in order to convey an *intended meaning* of the visual statement.

Pragmatics: prescribing how to assure *effective* construction and comprehension of the visual statement, as well as greater *usefulness* of the resulting representation for its intended use.

Some basic knowledge concerning the general issues outlined above is briefly discussed in the following.

II.2.1 Visual vocabulary and syntax

> ... among everything that's visually observable
> we can refer only to relationships and to contrasts.
> [Maurits C. Escher, *White-Gray-Black* (1951)]

Every particular visual language has its own vocabulary of graphical elements than can be used to construct visual messages within the language. A common repository of visual building blocks—graphical primitives and their visual properties—from which these particular vocabularies may be build up, can be nevertheless formulated in general terms. The overall classification, with some illustrative examples, of this set of elements is shown in Fig. II.4. It was sketched in a similar way already in [Bowman 1968, Bertin 1981] and little of greater significance has been added to this schema since then. The version shown here is based on the diagram from [Bertin 1981], though extended and significantly more detailed. Also, a different interpretation for some properties of regions is adopted here.

The general vocabulary is classified here along two main dimensions: the general element type and the kind of visual properties of elements that can be used to convey useful information. There are three element types discerned here:

Points, whose basic, defining function is just to *point*, in the sense of specifying some (comparatively) precise *position* in the diagram, usually within some system of coordinates (e.g., a one-dimensional scale or a two-dimensional Cartesian system).

Lines are characterized mainly by being significantly more long than wide, and their main uses are to specify *orientation* (possibly changing, as in the case of a curve) and to serve as *borders* of regions.

Regions serve to specify a two-dimensional areas in the diagram, with *shape* as their basic characteristics.

As it is evident from the above and from the examples in Fig. II.4, the types should be understood in the functional, not geometrical sense. An element used to specify position is thus a point, despite the fact that it can look as a small region, or consist of two crossing lines. Also, the division into these three classes is rather fuzzy and relative. E.g., it may be hard to decide how thick the line should be to be classified as a region, and genuine regions can be used also to specify orientation (like with an elongated rectangle), or exact position (say, with the centre of a circle). Also, contour of a region can play a role of a line and participate in different meaningful relations than the region itself, while some

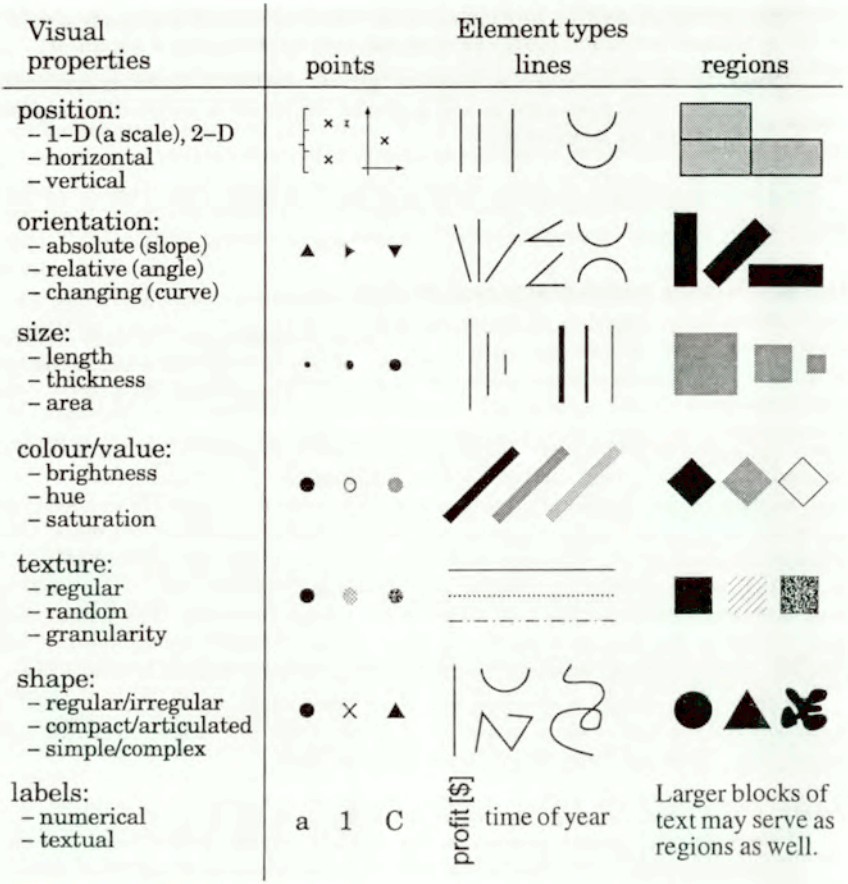


Figure II.4: A repertoire of basic elements of graphical vocabularies for visual languages.

elements of lines and regions (like endpoints, cross-points, corners, centres) can play a role of explicit or implicit points.

The main kinds of visual properties of elements are listed in the left column in Fig. II.4. Within the listed classes, there are many subclasses of properties as well as different methods of specification of descriptors for these properties. The most important of these are listed in Fig. II.4 as well. For example, the space of colour values is generally three-dimensional, and any, or all, of these dimensions can be used independently to convey useful information, while shape is a rather elusive property, without established and generally useful quantitative descriptors (see [da Fontoura Costa & Cesar 2001, Leyton 2001]).

The meaning of some properties may vary for different element types, e.g., size of a line may mean its length or its thickness, but for a region it usually means its area. For most

properties, various value scales are in use, both quantitative and qualitative. Some values of properties play especially important role and are used more often than others, e.g., vertical and horizontal orientations, right angles, some basic shapes (circle, square, triangle). There are also many interdependencies between various properties that may limit the expressiveness of visual languages using them. For example, texture usually induces particular colour or value; in fact colours or values are often produced with appropriate textures, using rasterization technique, as was done in Fig. II.4 as well.

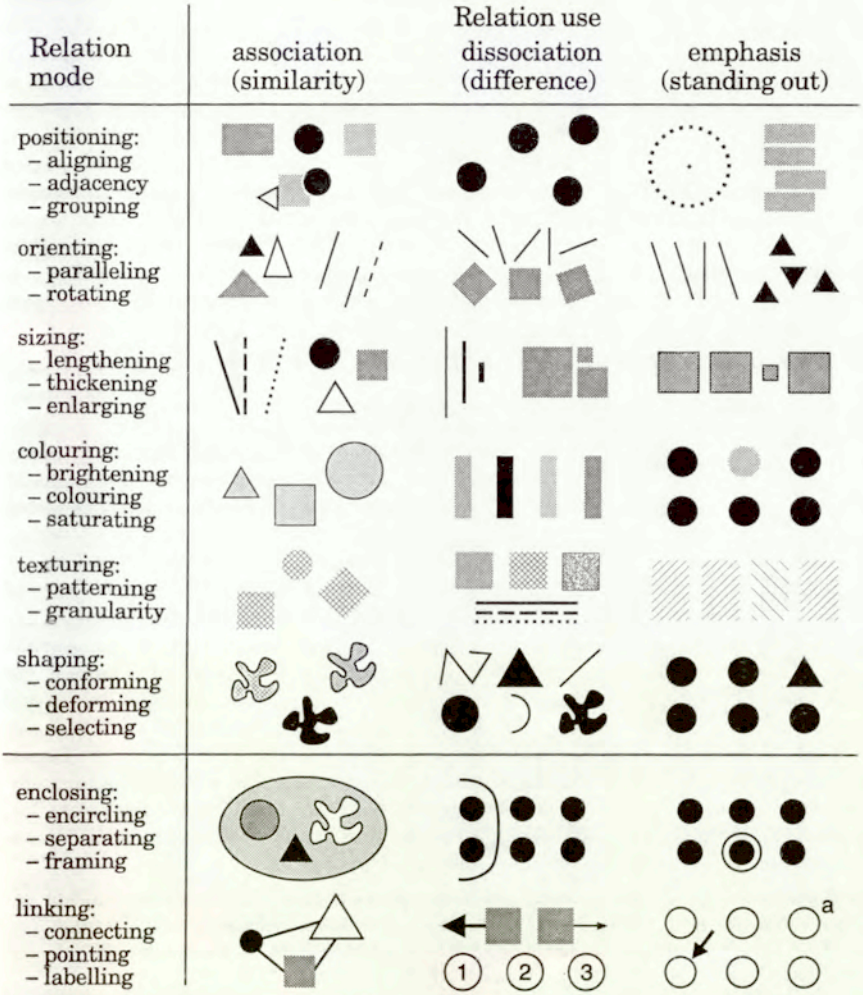


Figure II.5: Basic kinds of possible visual relations between elements of a graphical vocabulary.

Combining graphical elements to build a diagram puts them into various spatial and graphical relations with each other. These relations can be prescribed by the syntax of the given language, or can appear as consequences of the choice of elements and other relations. The latter are often called *emergent properties* and will be discussed later on, e.g. in Section II.4.2. Both the graphical elements (together with their properties), and relations between them, can be used to express useful meaning (see the next section). The repertoire of useful relations in visual languages used in diagrams is much richer than in other representations, due to both spatiality of the representation (see Section II.3.1.2), and the large number of visual properties of graphical elements. That multitude of possible relations can be roughly classified along two dimensions as shown in Fig. II.5. The *relation mode* dimension lists first the various visual properties which can participate in relations, generally in accordance with the list in Fig. II.4. The two additional modes of specifying relations (enclosing and linking) are not based on relating individual properties of elements. The *relation use* dimension lists three general uses of relating the elements of a diagram (or rather two, with an important special case of the second one singled out as a separate case). Of course, these effects do not exhaust all possible meaningful uses of visual relations. Similarly as in the case of the visual vocabulary, there are interdependencies between various relations, for example differences in texture may induce differences in colour or value, and differences in orientation can be perceived as differences in shape as well. Also, a part of an element (e.g., a contour of a region) may participate in different relationships than its other parts, or the element as a whole.

II.2.2 Expressiveness of visual languages

> ... in these signs there is a one-to-one correspondence between expression and content.
>
> [Umberto Eco, *A Theory of Semiotics* (1976)]

The main semantic requirement that the language must fulfill is that of *expressiveness*, i.e., the ability of the language to express *all* the facts (or classes of facts) it is intended to represent, and as few as possible of the *unintended* facts (especially false ones). While the first part of the requirement is rather obvious, the second part is much more complicated. Usually languages can express much more facts that is intended—the more so the more universal the language aspires to be. Especially propositional languages like natural languages and predicate calculus can easily express not only unintended facts, but also all kinds of falsehoods and contradictions (they are not *self-consistent*, see e.g. [Lemon & Pratt 1997]). As will be further discussed in Section II.3.1.3, analogical representations behave much better in this respect—usually, the more analogical the representation, the smaller the possibility of expressing with it false (impossible, contradictory) facts, although the full self-consistency can rarely be achieved.

Therefore, a good rule of thumb for constructing visual languages that will not express falsehoods is to use visual elements and relations as much as possible similar (i.e., analogical) in their properties to domain objects and relations to which they are to refer.

Example II.6a (Wrong visual language) Consider two simple facts about mutual positions of certain countries, stated in Fig. II.6a (inspired by [Mackinlay & Genesereth 1985]). When encoded with the visual language of Fig. II.6b,they give rise to a dia-

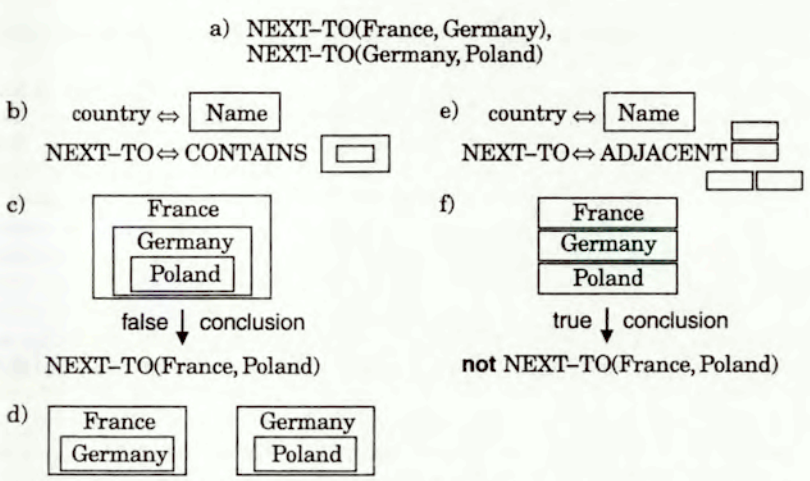


Figure II.6: Two simple facts (a) represented with an inappropriate visual language (b) may result in a diagram encoding a false fact (c) or a more complex diagram (d); with correct language (e), a simple diagram (f) may encode additional true fact.

gram in Fig. II.6c which, however, allows for concluding directly from it an unintended third fact, as shown, which is moreover obviously false.[9] We can prevent that interpretation of the diagram by drawing it differently, e.g., as a composition of two separate subdiagrams (connected by an implicit **and** connective), see Fig. II.6d. That has its disadvantages, however, as the resulting diagram is more complex and violates even more the analogicity of the representation, see Section II.3.1.3. The reason for the problem lies in the visual language used—the graphical relation CONTAINS used to encode the domain relation NEXT-TO has different properties (being transitive) than the latter (which is intransitive). In this way, analogicity of the representation is not preserved, causing an expressiveness mismatch between the language and the represented domain. Thus, a better way to correct the error is here to change the visual language. This is done in Fig. II.6e—the graphical relation used has now the same properties as the domain relation, and the resulting diagram in Fig. II.6f does not produce the false conclusion from Fig. II.6c. However, another (possibly) unintended fact can be read off the new diagram, namely **not** NEXT-TO(France, Poland). Fortunately, the fact is true and can be thus considered as an added bonus of this kind of representation. ■

Indeed, appearance of such cheap conclusions as shown in Fig. II.6c and f (called usually *emergent facts*) is quite common in diagrammatic representations, moreover, it can be considered as a consequence of their analogicity (see Section II.3.1.3). In most cases the effect is considered beneficial, as it effectively increases expressiveness of the language, and is especially useful in diagrammatic reasoning, see Section II.4.2. However, if for some reasons the appearance of such unintended facts is undesirable, special measures must

[9] We assume here the exact correspondence of arguments, i.e., NEXT-TO$(a,b)\Leftrightarrow$CONTAINS(a,b), but *not* NEXT-TO$(a,b)\Leftrightarrow$CONTAINS(b,a). Otherwise, it would be possible to construct another representation using the language of Fig. II.6b, but without generating the false conclusion.

be taken, usually decreasing the analogicity of the representation (like the subdiagram solution in Fig. II.6d, see also Fig. II.30d).

One may claim that there are apparently many other facts represented in Figs. II.6c, d, and f that were not stated in Fig. II.6a, and were nevertheless not mentioned above. They are facts like BIGGER(France, Germany) and WITHIN(Poland, Germany) (in Fig. II.6c) or NORTH-OF(Poland, Germany) and AREA(Poland) = AREA(France) (in Fig. II.6f). They are properly *not* listed, because they fall outside the universe of discourse provided by the definitions of visual languages used, as given in Figs. II.6b and e. Because of that, they in fact cannot be actually read off the diagrams at all, as it is only the visual language that prescribes what is the meaning of any given visual element or property of the diagram. A diagrammatic representation consists of a diagram *and* the visual language: a diagram alone does not suffice. Therefore, a diagram can be properly understood and used only if it is accompanied, explicitly or implicitly, by the correct visual language for its interpretation. Errors may arise when the diagram is interpreted with different visual language than that used or intended by its creator. That may easily occur when the language is only implicitly suggested by the diagram, as is often the case with informal use of diagrams outside some well-codified diagrammatic notation system.

II.2.3 Pragmatic criteria

As for a picture, if it isn't worth a thousand words,
the hell with it
[Ad Reinhardt (1913-1967)]

Two important criteria that should govern the construction of both visual languages and diagrammatic representations using them are worth to mention here as well. They belong to pragmatics of visual languages, as it was explained above.

Effectiveness criterion prescribes that the language and representation should minimize:

- costs of producing the representation of the given facts;
- costs (difficulty) of reading off the facts from the representation;
- possibility of making errors during production or use of the representation.

Intended function (goal) criterion prescribes that the language and the representation should be closely matched to:

- the information-seeking goals of the prospective user, and
- the kind of task(s) the representation is expected to aid.

The importance of these issues and profound influence that consideration of them may import on the visual language and resulting diagrammatic representation is shown with two simple examples that follow.

Example II.7 (Effectiveness) This example is a much simplified and reworked version of an example discussed in [Mackinlay & Genesereth 1985]. A few simple facts about three lectures and seasons during which they are delivered (Fig. II.7a) is represented first with the visual language specified in Fig. II.7b. When executed as in the diagram of

a) DURING(PL, Fall) & DURING(OS, Winter) & DURING(CO, Spring) & REQUIRES(OS, PL) & REQUIRES(CO, PL)
[PL: Programming Languages; OS: Operating Systems; CO: COmpilation]

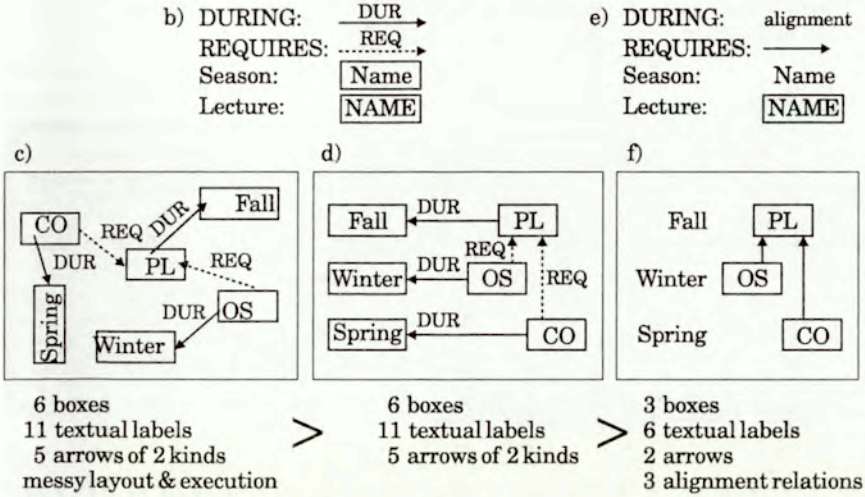


Figure II.7: Several facts (a) represented with the visual language of (b) may result in a messy diagram (c) or a much clearer one (d); with simpler language (e), the diagram is even more neat (f).

Fig. II.7c, it results in a complex drawing, containing many graphical elements which are superfluous (e.g., redundant arrow labels, boxes around all names), and is very messy, with sloppily executed arrows, imprecise placement of labels, unnecessary rotation of the "Spring" label, etc. Hence, the cost of such a representation is high, in all three areas listed above, especially concerning the difficulty of reading the data off the diagram. With the same language the diagram can be drawn far better, simpler and more clear, see Fig. II.7d. Note that in this version the original visual language is in fact augmented by the use of certain visual relations to reinforce the meaningful similarities and differences between elements encoding different kind of data (e.g., horizontal and vertical alignment of boxes). This hints at a possible simplification of the visual language, as shown in Fig. II.7e, with the resulting neat diagram of Fig. II.7f which allows for easy grasping of all encoded data at a glance. ■

A rule of thumb for decreasing costs of diagramming is to make the visual language (and rendering of the diagram itself) as simple as possible. However, it is not good to push that advice too far. For example, in [Tufte 1983] the author advocates an unlimited maximization of "data-ink" (meaning deleting all elements not directly coding some information), illustrating that with examples so simplified that they become hard to read precisely and became prone to error due to a total lack of information redundancy.[10] Thus, as it often happens, here also an extremal solution is not necessarily the best one.

[10] See especially Section 6 there, with figures on pages 124 and 132.

a) $\{(x,y): (88, 33.29), (85, 18.76), (87, 58.31), (86, 45.14)\}$

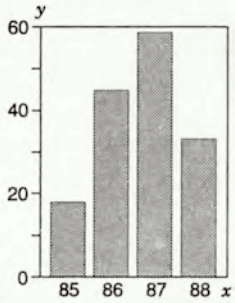
b) for accurate value lookup (find y for given x):

use table (sorted by x)

x	y
85	18.76
86	45.14
87	58.31
88	33.29

c) for fast value comparison:

use ordinary bar chart



d) to embellish your presentation:

use fancy bar chart

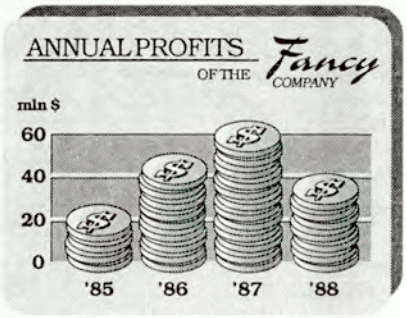


Figure II.8: A list of number pairs (a) should be represented with different diagrammatic means, depending on the goal (b, c, d).

The goal of the representation—both the needs of the user and the nature of the task for which the diagram is to be used—influence the choice of the visual langauge and appearance of a diagram even more, see [Roth & Mattis 1990, Roth & Mattis 1991].

Example II.8 (Presentation goal) Let us consider a few pairs of numbers given in Fig. II.8a. When the representation is to be used for finding accurate numerical value of y for the given x, a simple table, sorted by x, is the best solution (Fig. II.8b). However, if we want to be able to compare fast the (qualitative) relation between values of y's for different x's, a simple bar chart as in Fig. II.8c is much better. But if the goal is simply to embellish one's presentation with a nice picture, a decorative bar chart is a must, see Fig. II.8d (obviously, executed in full colour, which this author is unfortunately not allowed to use here)! ■

Some goals can be combined in a single diagram. For example, if the diagram should fulfill both goals specified in Fig. II.8b and c, a bar chart augmented with exact numerical values put at the bars will do that. Significant results on more or less formal specification of such goals have been reported, see e.g. [Roth & Mattis 1990], and computer systems capable of automatic design of appropriate presentations have been implemented (see e.g. the SAGE system described in [Roth & Mattis 1991]).

II.3 Diagrammatic representations

... there is simply no physical or mathematical conception
that is not open to graphic representation ...
[David W. Brisson, *Visual Comprehension in n-dimensions* (1978)]

There are many issues concerned with the use of diagrammatic representations. First, one should discern and analyze important specific properties of diagrams making them different from other types of representations. Second, the general classification of different types of practical uses of diagrams is needed to see their application domains in proper perspective (Section II.3.3). As for the former issue, the properties are usually divided into advantageous ones (Section II.3.1) and those that may cause problems (Section II.3.2). The distinction is a little artificial, as practically all of them have both positive and negative aspects. The latter are often advanced as arguments against wider application of diagrammatic representations. Fortunately, these arguments are often based on misconceptions or on a too restrictive understanding of the issue, e.g., by generalizing features of some particular visual language to diagrammatic representations as a whole.

Let us begin from a diagrammatic formulation of that mysterious mechanical problem used in Section II.1.2 to introduce basic ideas of logical representations.

Example II.3g (Diagrammatic formulation) A diagram of the example mechanical system shown in Fig. II.9 reveals the nature and structure of the object described with predicates in Tables II.1 and II.2. The elements of the system of pulleys are appropriately labelled with the names used in these tables. In this form it is much more understandable, and solving it becomes now quite trivial for a human. The solution is shown in Fig. II.10. The inference rules are rephrased as diagrammatic rewriting rules in Fig. II.10a, while the step-by-step process of solving the problem is shown in Fig. II.10b. ■

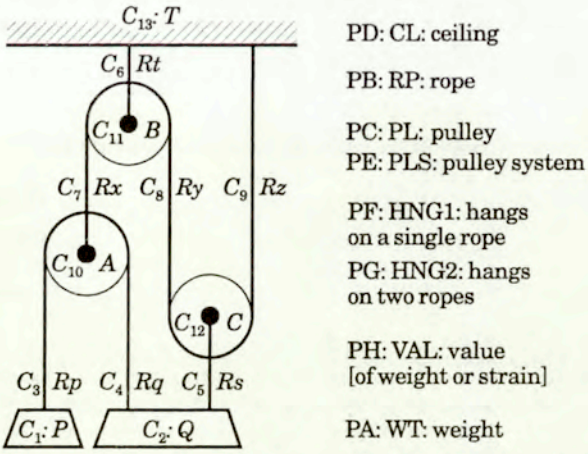


Figure II.9: The mysterious mechanical example from Tables II.1 and II.2 in a diagrammatic form. The problem is to find the ratio Q/P of weights in equilibrium.

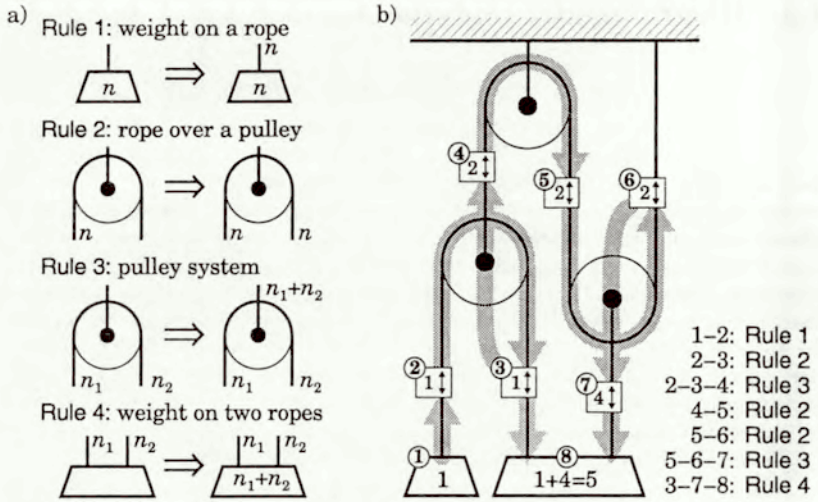


Figure II.10: The pulley example solved diagrammatically: diagrammatic inference rules (a), and the record of solution process (b).

The application of diagrammatic inference rules shown in Fig. II.10a involves first recognition in the diagram of the subpatterns provided in the left hand side of each rule. This is easy for human solvers, due to abilities of the perceptual apparatus of the eyes, see Section II.3.1.1. In contrast, logical formulations make such perceptual inferences much more complicated, see discussion in Section II.1.2.3 concerning the *perceptual rules*.

Other advantages of diagrammatic representations, at least for a human user, become also quite apparent when comparing the above formulation with the logical one. These advantages are systematically discussed in the next section.

II.3.1 Advantages of diagrammatic representations

耳聞不如目見

(Ěrwén bùrú mùjiàn:
Ear hears not so good as eye sees.)
[*An old Chinese saying*]

The main advantages of diagrammatic representations in comparison with propositional ones come from their analogicity (called also *directness*), and their (usually) two-dimensionality (called also *spatiality*). Some other advantages are also mentioned, like the possibility to get rid (at least partially) of reference labels, or easy detection and exploitation of various symmetries in the represented data. For human users, a great advantage comes also from the possibility of a full use of the sense of vision, undoubtedly the most effective human perceptual and information-processing system.

II.3.1.1 Effective visual apparatus

Of your sense organs, the best suited
to the receiving of complex information is your eyes.
[Fred Hoyle, *The Black Cloud* (1957)]

Possibly the main advantage of diagrammatic representations for humans is that they are endowed with a very efficient and sophisticated apparatus for picture interpretation, as already pointed out in Chapter I, especially in Section I.1.3. This is not the case with our ability to produce diagrams, where our abilities are much less developed than on the interpretation side. Fortunately, computers are already much more proficient in the picture generation task, hence they can be used as a sort of prosthetic devices for our underdeveloped pictorial effector, see Section II.6. Thus, the full potential of diagrammatic representation and reasoning will be probably achieved in hybrid man-machine systems combining the appropriate strengths of both parties, at the same time compensating for their respective weaknesses.

Moreover, the human visual apparatus most probably has not said, so to speak, its last word. As it will be also discussed in Section II.5.1 in the context of using diagrams in mathematics, our picture processing abilities are, most probably, not yet utilized in their full possible potential. Our still mostly verbal educational system, not aimed at developing and training human pictorial thinking and processing skills, but in many cases even actively discouraging them (see Section II.5.1.2), does not allow for the full development of these skills. With coming changes in this area, it is hoped that human abilities of using pictures in thinking and other activities may rise to still higher levels of competence and effectiveness.

II.3.1.2 Spatiality of diagrams

You know how on a flat surface, which has only two dimensions,
we can represent a figure of a three-dimensional solid, ...
[Herbert G. Wells, *The Time Machine* (1898)]

Except for some rather special cases of one-dimensional diagrams, most diagrammatic representations are at least two-dimensional. The advantage of that is rather obvious—one has much richer repertoire of possible relations between elements of the representation (that can be used to represent relations between domain objects), not only as to their kind, but also as to the number of objects that can be related to the given one. In essentially one-dimensional propositional representations, modelled after (spoken) language structure, one has only one relation directly available (*concatenation*, or succession) and only two objects that can be directly related to the given one (the *previous* and the *next* one).

To represent richer structures, one-dimensional representations must resort to long-range reference labels of various kinds, like names, pronouns, matching parentheses, etc., see Section II.3.1.4. That ability of two-dimensional representations to relate many objects directly and simultaneously to a given one has two main advantages from the point of view of retrieval of the needed information. Namely, it reduces:

Size of search space, as it is usually sufficient to consider only the objects directly related to a given one.

Search costs, due both to the smaller search space, and to the direct access to related objects, without a need for additional search for matching labels or the like.

Let us illustrate the above with the pulley system example:

Example II.3h (Limited diagrammatic search) Contrary to the situation with the logical representation of the pulley system in Section II.1.2.2, solving the problem in the diagram in Fig. II.10 involves much less search. For example, at the first step, when focusing on the only known weight P, one needs only to search which of the four rules has such beginning, but after that finding the rope to which the strain should be transferred involves no search. This is only a little worse at the step, say, third—only two adjacent objects should be considered (the pulley and the weight), with three rules possibly applicable (rules 2, 3, or 4). Two of them are immediately filtered out (one due to the lack of data, the other as making no contribution to the process). While filtering out these rules and applying the third, finding the three needed objects (ropes Rp, Rs and Rx) again involves no search. Compare that with the amount of search estimated for some steps in Section II.1.2.2. ■

In view of the above, the often advanced opinion [Stenning & Oberlander 1995] that diagrammatic representations take their efficiency from limited expressiveness of the diagrammatic media does not sound convincing. See e.g. [Lemon & Pratt 1997] for a discussion of that issue—instead, the efficiency is attributed there to *spatial constraints*, coming from the spatiality of diagrams, and leading to limiting search costs as in the Example II.3h above. Thus, the effectiveness comes rather from richer instead of from limited expressive possibilities of diagrams. The limitation of expressiveness does occur, but on another level. Namely, when the structure of the represented domain is richer than a spatial structure of a two-dimensional Euclidean plane holding the diagram, that may limit the expressiveness of the diagram (like in the case of making certain set structures not representable with Euler circles [Lemon & Pratt 1997]), or lessen analogicity of possible representations. The latter forces non-analogical components into the representation which may lead to errors of the "impossible cases" kind, see Section II.3.1.3, and limits the expressiveness of the representation, but at a much higher level, because a significant part of the information is still encoded analogically.

Even richer expressive possibilities are provided by diagrams with more dimensions than two. For a computer implementation, adding more dimensions does not cause much trouble (see Section II.6.2.2). The situation is much worse with human users, despite that there are various techniques for representing three-dimensional diagrams, from two-dimensional projections and sections to using stereo vision and virtual reality. However, the human efficiency deteriorates with the increase of dimensionality above two dimensions. Many people have considerable difficulty with even three-dimensional imagination. The difficulties grow steeply for higher dimensions, for which the potential expressive gain is all but lost to the inefficiency of producing and using the representation by humans. However, in certain cases it is possible to produce useful representations of many-dimensional data as well [HYPERGRAPHICS 1978]. Intensive research on new techniques of that sort is under way, especially within the discipline of so-called *scientific visualization.*

II.3.1.3 Analogicity of representation

Surely it is a bit absurd to draw a few lines
and then claim: "This is a house."
[Maurits C. Escher, *The Graphic Work of M.C. Escher* (1967)]

A number of important consequences follows from the analogicity of representation, see Section II.1.1. Some of them are advantageous, while other may lead to troubles. The three main features of the first kind are discussed in the sequel; some troublesome features are discussed in Section II.3.2.

Filtering out impossible cases. One of the problems with representations of more complex information is assuring internal consistency of the representation, i.e., eliminating the possibility of representing "impossible cases" that cannot occur in the represented domain. Analogical representations are able in principle to solve that problem. Assuming the represented domain is consistent, the full analogicity of the representation assures consistency of the representation as well (the feature called *self-consistency* in [Lemon & Pratt 1997]). Thus, analogical representations make production of inconsistent representations harder, allowing for more easy spotting of such a kind of gross error as in the "Ghost Town" case in Example II.12 (Fig. II.17d) below.

Unfortunately, fully analogical representations rarely, if at all, occur in practice. First, practical representations are usually hybrid, see Section II.1.1, with non-analogical elements often added to overcome certain limitations of analogical representations, see Section II.3.2. Second, the representation method or medium used may have limited ability to represent analogically all necessary data, as discussed in the previous section.

Example II.9 (Impossible triangle) Representation of three-dimensional objects on a flat plane is one of those tasks that are not possible to solve with full analogicity, see Section I.5.2. Thus, it is easy to represent in a diagram an impossible triangle of Fig. II.11, despite that the sum of its angles equals 270 degrees. ■

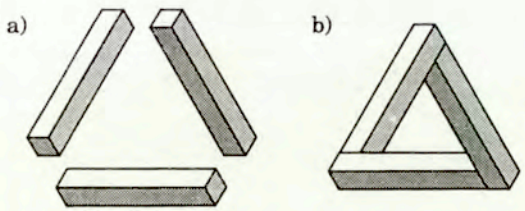


Figure II.11: Three identical rectangular beams (a) can be apparently joined together in three dimensions to form an impossible triangle (b).

Generally, impossible objects are easy to produce when the diagram represents only a part of important components of the situation, e.g., only external appearance of the object, without the important details of its internal fine structure. Then convincing drawings of mermaids or centaurs are not too hard to procure.

Yet another cause of generation of impossible cases comes from the inevitable imprecision of a drawing in representation of continuous quantities, see Section II.3.2.1, as well as

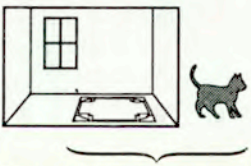
examples there (Example II.11a). It is also a common cause of the *false divergence* effect discussed in Section II.4.3.2 (Example II.23). Also the kind of error illustrated by the "Ghost Town" example would be much harder to spot if the value of the distance specified in Fig. II.17d would have been close enough (compared with the accuracy of the map) to half of the distance between Warsaw and Cracow. Therefore, the full self-consistency of diagrammatic representations can rarely be achieved in practice. See [2] for a deeper discussion of this problem.

Emergence. One of the most interesting consequences of analogicity of the representation is the effect of *emergence*. It was already mentioned in Section II.2.2, and is discussed in more detail in Section II.4.2 on diagrammatic reasoning, because it is of much significance to this area of diagrams use, being actually an important cause of its attractiveness. The following example may indicate that emergence follows to a large degree from the analogicity of the representation.

Example II.10a (Cat not on the mat) Consider two simple facts stated in natural language in Fig. II.12a. Concluding formally from them that *the cat is not on the mat* requires considerable amount of juggling with meanings of the words *mat*, *cat*, *room*, *in*, *outside*, etc., and syntactic and semantic relations between them. However, representing these two facts analogically, as in Fig. II.12b, makes the conclusion in Fig. II.12c immediate, with no need for any additional processing of the representation. See also Section II.3.2.5 for further discussion of this example, and Example II.19a in Section II.4.2 for a more abstract example with essentially the same syntactic structure. ■

a) The mat is in the room.
The cat is outside the room.

b)



c) The cat is not on the mat.

Figure II.12: Two facts (a), represented analogically (b), produce a third emergent fact (c).

Directness. Analogical representations are often called *direct*, to account for a much more direct correspondence between the structure of the representation and the thing represented that occurs in these representations, as compared with propositional ones, see Sections II.1.1 and II.1.3. This directness has also important consequences. It makes the representation better comprehensible to human users, as it is in essence a *model*, rather than an encoded description, of the thing represented. It makes analysis and processing of information so represented simpler, either for a human or a machine, contributing also, like spatiality discussed in the previous section, to the decrease of sizes of search spaces to be considered.

However, the directness of analogical representations is never full, in part due to the inevitable imprecision of the representation, see Section II.3.2.1. A line in a diagram may be said to represent a geometric line more directly than the words "straight line," or the predicate symbol $SL(L_1)$, say, but still it is not *the* straight line, only an approximate representation of it. The line drawn in the diagram has certain thickness and some unevenness of its contour—the elements which certainly do not represent any components of the geometrical concept of a straight line, and thus must be filtered out in the process of understanding and using the diagram. See Section II.3.2.1 for a more thorough discussion of the drawing imprecision issue. Also, a straight line of geometry is by definition infinitely extended in both directions—the feature that *cannot* be directly represented by any real diagram, so that it must be encoded in the diagram indirectly in some way. It can be done, for example, by using a default assumption of infiniteness unless some explicit symbol (like a dot or dash), denoting the endpoint of the segment of the line, appears in the diagram. More explicitly, some graphical symbol denoting an infinite extent can be used, like an ellipsis "...".

Not surprisingly, cartography, where the correspondence between spatial structures of the earth surface and the paper surface is one of the most natural and *direct*, constitutes one of the most successful applications of diagrammatic representations. An added complexity of nonconformity, for larger areas, of the curved surface of the earth with the planar surface of a map did not hamper its development. It led only to the development of a rich variety of projection schemes, preserving either distances, or angles, or areas, etc., depending on the specific purpose of the map.

II.3.1.4 Getting rid of reference labels

... the legend makes symbolic conventions meaningful ...
[Eric M. Hammer, *Logic and Visual Information* (1995)]

As mentioned in Section II.3.1.2, one-dimensional representations must use various long-distance labels, like names of objects taking part in multiple relations, or structuring elements like matching parentheses. Due to much richer possibilities for relating many objects in various ways, that necessity rarely occurs in diagrammatic representations. This goes in accordance with the *specificity* property to be discussed in Section II.3.2.5, namely, the common assumption that different graphical elements in a diagram necessarily represent different objects. Although the assumption can be overruled, (e.g., just by using labels, see Section II.3.2.5), that is rarely needed. In problem solving using diagrams it means that identification of objects needed at a given step does not involve searching of (usually huge) lists of reference labels. This is easily seen in Example II.3g above. For solving the pulley problem diagrammatically, no names of the ropes, pulleys, and other objects were necessary, while the solution using the logical representation needed repeated searches for matching names of objects and relations, see [Larkin & Simon 1987].

Labels in diagrams are, however, often used, see e.g. the discussion of styles of mathematical diagrams in Section II.5.4. There is even a graphical relation of labelling set aside for that purpose, see Fig. II.5 in Section II.2.1. In most such cases, they are useful only for connecting the diagram with the accompanying text or formulae. Labels are used for linking objects named in the text or formula with their representations in the diagram.

Sometimes (like on maps or quantitative graphs), such labels are not attached directly to the graphical elements of the diagram proper, in order not to clutter them unduly. Instead, a *legend* is used, which links labels to certain properties of visual elements (like colours, shapes, etc.) which are then used in the diagram to mark appropriate objects, see Figs. II.2 and II.23 for simple examples. There was even an attempt to use such graphical labels directly in the text (see [Byrne 1847, Tufte 1990], discussed in Section II.5.4), with rather mixed results, however, and the proposal did not catch. A rare cases when it is used do occur, see two examples in Sections III.5.2.2 and III.5.2.4, the latter in the proof of Proposition III.10.

II.3.1.5 Exploitation of symmetries

This place, however, had an over-all symmetry and pattern,
though one so complex that it eluded the mind.
[Arthur C. Clarke, *Rendezvous with Rama* (1973)]

Rarely mentioned in the literature (see, however, [Barwise & Etchemendy 1996a]) is the fact that various symmetries are usually easy to spot in a diagram, at least for human reasoners. Such symmetries are often very useful to reduce the number of cases to be considered, as it may suffice to analyze in detail only a single case, because for other cases the argument repeats symmetrically. A proper exploitation of symmetries may thus drastically reduce the amount of processing needed due to the divergence effect, see Section II.4.3. That is often so naturally taken for granted that the fact that the argument should properly have been repeated for those several cases is rarely even noticed, see the footnote to Example II.22 in Section II.4.3.1. That may sometimes lead to an error due to *overlooked divergence*, by omitting the case which actually needs separate consideration, as in the above-mentioned example.

Various symmetries (mostly due to sign change of the involved quantities) are commonly exploited in Chapter III of this work, see e.g. Figs. III.32 and III.47, considerably reducing the number of diagrams needed to represent different possible cases.

II.3.2 Problems with diagrammatic representations

... I have to let you see my little difficulties,
if you are to understand the situation.
[Arthur Conan Doyle, *A Scandal in Bohemia* (1892)]

There are many features of diagrammatic representations that may cause problems with using this kind of representation. There are also some prejudices concerning capabilities of diagrammatic representations—it is often asserted that diagrams cannot do this or that. In many cases, these allegations are unfounded, or there are more or less easy remedies for the problems. In this section, the most common such problems are briefly discussed. Some other problems of this sort, connected specifically with diagrammatic reasoning, are discussed in Section II.4. Specific problems connected with the application of diagrams in mathematics are further discussed in Section II.5.

II.3.2.1 Imprecision of diagrams

Possibly the most serious problem with diagrammatic representations comes from a limited precision of diagrams. By definition, analogous representations should represent directly the properties of domain objects and relations between them, including continuous quantities. E.g., the length of the domain object should be represented by the length of appropriate graphical element, the parallelism of object's edges by parallelism of corresponding lines, etc. However, as the precision of the drawing (sometimes also the precision of necessary measurements in the object domain) is limited, diagrammatic representation of such quantities or relations (called in the sequel *metric features*) usually cannot be fully analogous and precise. That may lead to even gross errors in the use of diagrammatic representations if proper caution is neglected. Let us start a more detailed analysis of the problem from an entertaining example.

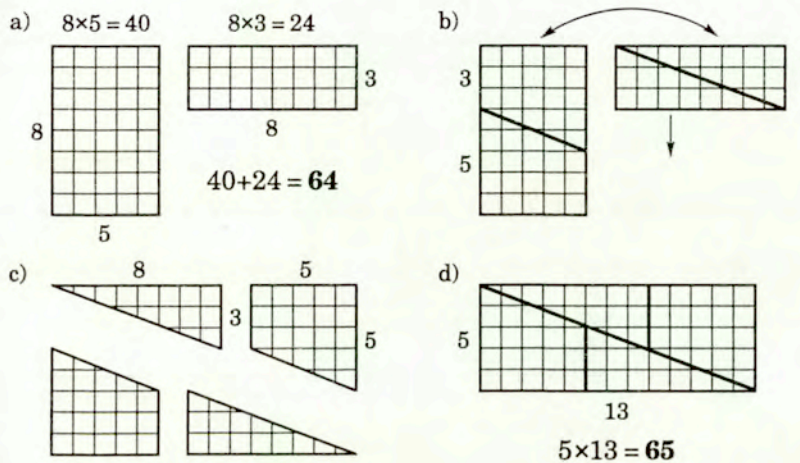


Figure II.13: The *"64=65"* diagrammatic puzzle.

Example II.11a ("64=65" puzzle) This is a well-known puzzling argument apparently "proving" diagrammatically that 64 = 65.[11] The reasoning goes as follows, see Fig. II.13. We start from two rectangles, containing together 64 unit cells.[12] Then we dissect each rectangle into two equal parts with diagonal straight segments having endpoints exactly at the corners of the square cells. The obtained parts are shifted on the plane to form another configuration and then glued together into a single rectangle. The resulting rectangle contains obviously $5 \times 13 = 65$ square cells. Since we neither removed nor added any parts of the pieces used, obviously 64 = 65. ■

[11]The origin of the example is uncertain; some attribute it to Lewis Carroll. Here it was adapted, with modifications, from a book on entertaining mathematics [Jeleński 1968]. There are many other puzzles based on the same principle, some of them producing as much as six different results for different arrangements of parts.

[12]A version with an 8 × 8 square as the starting figure is more usual [Jeleński 1968].

The cause of error lies here in the particular kind of diagram imprecision which one may describe, somewhat jokingly, by:

The Grand Drafting Theorem. *Through any three (or more) points one may draw a straight line, provided it is thick enough.*[13]

Overlooking the fact that diagrams can be neither constructed nor read with an infinite precision may lead to wrong results. Any diagram interpretation relying on such precision can be invalid, and therefore needs some additional checking and confirmation besides only the visual obviousness.

Example II.11b ("64=65" puzzle: solution) To see where the extra square is hiding in the example,[14] one may draw it more precisely, i.e., with lines sufficiently thin (relatively to the dimensions of the diagram). After doing this, one finds the mystery cleared, see Fig. II.14. Indeed, in the previous drawing (Fig. II.13d), the diagonal of the rectangle has been drawn thick enough to hide the fact that it actually passes through four *non-collinear points*, comprising vertices of an elongated quadrilateral, producing that extra area of one square unit. Note, however, that the construction of Fig. II.14 and the explanation above

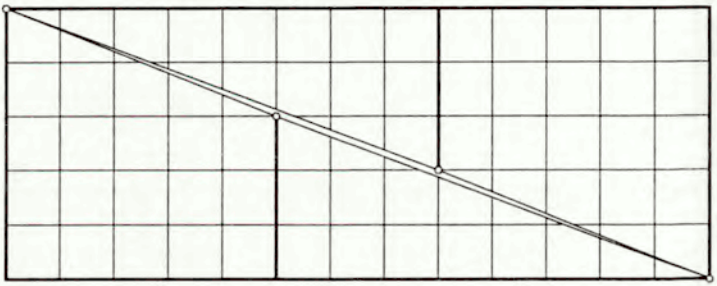


Figure II.14: Explanation of the *"64=65"* diagrammatic puzzle.

also do not by themselves constitute a proof that the four points marked in Fig. II.14 are *not* collinear. The apparent gap in this figure can be caused by an imprecise drawing in the same way as the lack of the gap in Fig. II.13d was! The existence of that gap should be ascertained by other means, see e.g. Example II.11d below.

Let us observe, moreover, that the original reasoning leading to the equality $64 = 65$ may be easily considered as a proof that *there must be a gap* in the final rectangle. That is, that seemingly erroneous reasoning is actually fully correct—it only proves another thesis instead of the equality $64 = 65$ with which the puzzle attempts to impress the reader. Namely, the diagrams prove that there must be a gap in the middle of Fig. II.14 and that its area equals exactly one—a fact not so straighforward to demonstrate by other means, see Example II.11e in Section II.4.1.3. ■

Generally, the problem of imprecision comes from the impossibility to directly represent *infinity* in a diagram. There are two distinct kinds of infinity to consider in this context, cf. [Rucker 1982], with different effects on diagrammatic representations:

[13] Based on a mathematical joke of unknown origin.

[14] The error analysis provided here as well as further discussion of the example is due to this author.

Infinity in the large: infinitely long lines, infinite extent of the plane, infinite length of a series, etc. This situation is rarely harmful; a number of devices to overcome that difficulty are in use, cf. [Jamnik et al. 1999], the discussion on limitation of directness in Section II.3.1.3, and an example in Fig. II.26.

Infinity in the small: impossibility of infinitely precise representation of continuous domains—infinite division of the plane and line, continuity, etc. This is a much more harmful difficulty, as the example above and many other to follow signify.

General ways of coping with the diagram imprecision are discussed in the sequel; still more about that can be found in Section II.4.1 on diagrammatic reasoning. Let us start here from an analysis of several possible methods of establishing some metric feature. For example, Fig. II.15 shows three ways to establish the equality of sides of a triangle.

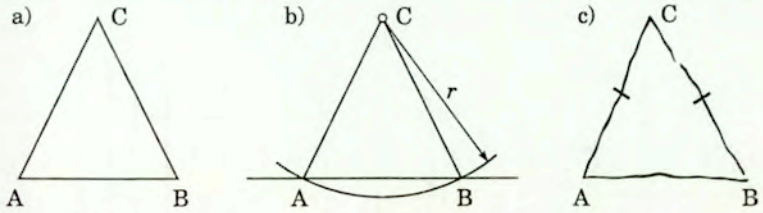


Figure II.15: Different ways of establishing the equality of sides AC and BC: using metric isomorphism (a), structural reasoning (b), and explicit statement (c).

Metric isomorphism. The first method, with a drawing of the sort shown in Fig. II.15a, conveys the required property by using the isomorphism between an abstract equality of triangle sides and (apparent) physical equality of lengths of certain marks on the paper. As such, it is in principle analogical, but vulnerable to the imprecision effect, hence unreliable, because no method of physical measurement can definitely prove the exact equality of segments AC and BC. Despite that, such representations are in practice quite common and considered reliable. Their relative reliability is based on a common interpretation convention that may be called:

Apparent look rule. *Elements of a diagram that look significantly close to having some (usually metric) property (e.g., a line being straight), or being in some relation (usually metric) to each other (e.g., segments being of equal length) should be assumed as having (reliably) that property or relation.*

The common practice of taking that rule for granted is summarized by the quote:

> Perception, on the other hand, is unreliable, ... and can refer only to actual, physically given objects, which are always imperfect. However, physical objects must not be confused with the percepts derived from them. ... When a person reports that he sees a square, he is referring not to a physically deficient specimen but to the pure shape of the perfect square, with which geometry is concerned. He sees a figure with truly right angles and truly equal sides.
>
> [Rudolf Arnheim, *Visual Thinking* (1969)]

With that rule in effect, it is thus the responsibility of the constructor of the diagram to assure that when some property or relation is intended *not* to be conveyed by the diagram, that fact must be explicitly and reliably indicated in it. For example, if the constructor wants to state that the segments are not equal, he must draw them to look convincingly unequal, e.g. using qualitative means (like large difference in length) or symbolic means (like explicit marks provided by the visual language to mean inequality). Otherwise they may be considered equal by the user on the basis of the apparent look rule.

As the apparent look rule is commonly assumed, but rarely explicitly and rigorously stated within the specification of the used visual language, the rule is a common source of error. One of the basic problems here is the problem of ambiguity, or scope of the rule. Namely, which of possibly many specific properties of objects apparently looking to hold in the drawing should be chosen to be fixed by the rule? The problem is aggravated by various visual effects like illusions, dependence of perception on various side circumstances (e.g., an orientation of the figure), and the like, see some examples in Fig. II.16.

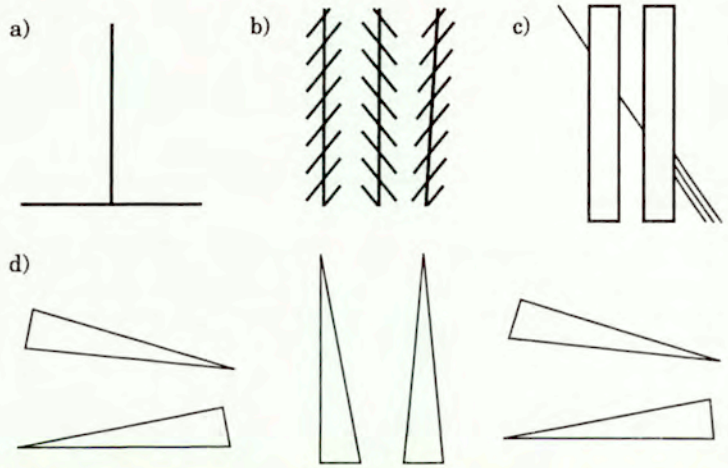


Figure II.16: Influence of illusions and orientation on metric judgements: are the two segments equal? (a); which two vertical lines are parallel? (b); which pairs of slanted segments are collinear? (c); which triangles are right, and which isosceles? (d).

A related convention is the *general position* rule, see Section II.3.2.4. For example, if the intention is to draw a triangle representing *any* triangle, it should not be drawn in a manner suggesting some particular kind of triangle only, like an isosceles triangle (as in Fig. II.15a) or a right-angled triangle, etc. That again raises the issue of the scope of the *apparent look* rule, also in connection with the so-called *particularity* feature, and the representation of sets of objects, see Section II.3.2.3. Namely, does the drawing of a triangle, with two sides apparently equal, represent this particular triangle only, or all triangles geometrically similar to it, or any isosceles triangles, or else any triangles whatsoever? If these conventions are not clearly clarified in the visual notation used in every particular case, that may lead to many interpretation, communication, and reasoning errors.

Structural reasoning. Another way, shown in Fig. II.15b, uses structural reasoning (see Section II.4.1.2) allowing the observer to derive (reliably) the equality, independently of any length measurements. As the example shows, a genuinely metric feature like the equality of segment lengths can be nevertheless represented in a way not requiring an infinite drawing precision, and thus essentially reliable.

However, in the final account the derivation of a feature by such structural reasoning must inevitably depend on some more elementary features of the diagram, on which the reliability of that derivation is indirectly dependent. And those more elementary features may be stated with metric means, possibly augmented by the apparent look rule. E.g., the construction in Fig. II.15b is based, among others, on the apparent look of the curve, suggesting its interpretation as a circular arc, reinforced by the visual language convention of the radial arrow with accompanying letter "r" for *radius* (of a circle, supposedly).

Explicit statement. In the last case (Fig. II.15c), the equality is explicitly *stated* with the corresponding hatch marks, despite that even a crude visual measurement gives strong evidence to the contrary. Moreover, the very sides of the triangle are here drawn so sloppily that the property of being straight sides of a triangle must be also ascertained with the resort to the apparent look rule, relaxed far enough to function also for such a sketchy-drawing visual language.

Such explicit stating of some features despite visual appearances constitutes an obvious departure from the analogicity of representation, with equally obvious dangers, like a possibility of generating impossible cases, see Section II.3.1.3. Not surprisingly, features of this type can be, and often are, stated propositionally rather than diagrammatically, in an accompanying text or within the diagram itself. Here the formula $\overline{AC} = \overline{BC}$ put besides the drawing of the triangle might serve the purpose.

II.3.2.2 Incomplete information and disjunctive knowledge

It is often claimed that diagrams, or more generally, analogical representations, cannot represent *incomplete information* about the objects. This belief is neatly summarized by the following quote from a respectable handbook [AI Handbook 1981]:

> ... analogical representations become unwieldy for certain kinds of incomplete information. That is, if a new city is added to a map, its distance from other cities is obtained easily. But suppose that its location is known only indirectly, for example, that it is equidistant from cities Y and Z. Then the distance to other cities must be represented as equations, and the power of the analogue has been lost.
>
> [*The Handbook of Artificial Intelligence* (1981)]

Contrary to the claims expressed above, there are various remedies to that apparent deficiency of diagrams, as remarked in, e.g., [Lemon & Pratt 1997]. Let us take the example from the quote above. It is true that the conventional visual language of geographical maps, developed just for representing *exact* positions, does not contain standard means for representing such kind of incomplete data about the positions. This does not mean that representing such incomplete information is impossible with a diagram—one needs

only to add appropriate visual language tools for doing that. Even a standard geographical map notation contains means for representing some types of incomplete information. E.g., the sizes or shapes of symbols denoting cities are often used to represent the size (or population) of the city in question. However, rarely the correspondence is continuous and complete; usually, there are only a few discrete sizes (or shapes) of symbols available, representing quite wide intervals of real city sizes (e.g., a single symbol can be used for a whole population range from, say, 100,000 to 1,000,000). Thus, when looking at the symbol of that size (or shape) we get a quite incomplete information about the possible population of the city.

Representation of incomplete information of the kind exemplified in the quote can be done with appropriate visual language extension, as the following example shows.

Example II.12 (Incomplete information) Let us start from the simple map shown in Fig. II.17a, containing two towns (Warsaw and Cracow—the current and the old capital of Poland) at their appropriate coordinates as specified (completely) in the figure. When we receive another piece of data, namely concerning the distance of a third city (Lublin) from Warsaw, see Fig. II.17b, adding that information to the map is not difficult—e.g., by using the visual device of a gray circle showing possible positions of the third city. With yet another piece of data—about the distance of Lublin from Cracow—the information becomes less incomplete, allowing for only two possible positions of Lublin, shown with gray dots in Fig. II.17c (the two large dotted circles marked there are not part of the map proper; they were added for illustrative purposes only).

In Fig. II.17d it is shown how the visual language devices used above can be also used for easy spotting of certain gross errors or inconsistencies in the data, like the contradictory data about the position of a hypothetical Ghost Town. Finding that inconsistency with the use of only propositional representation provided in Figs. II.17a and d will be much more cumbersome, involving calculations, especially complex if we had to make precise account of the changing distance between meridians. ■

Representing sets of objects. Essentially, this problem boils down to the question of representing *sets of objects*. With incomplete information about an object we cannot identify a single object, as there can be many objects sharing this partial characteristics. In some particular cases, difficulties with representing such sets of possible objects may arise, especially when the visual language used does not have standard provisions for that, but it is not impossible in principle. In general, two main ways of overcoming the possible trouble can be used here:

Extended visual language: devised just for the purpose of representing the required kind of knowledge.

Divergence: separate the representation into several cases represented in different diagrams (see Section II.4.3 for more discussion of divergence in a diagrammatic reasoning context).

The latter solution is practical when the set contains a few discrete elements, like in Fig. II.17c. The situation in this case is sometimes called a problem of *disjunctive knowledge* representation, and is discussed in detail below. The former approach can be used also for continuous infinite sets, as in Fig. II.17b and d. Often the extension of

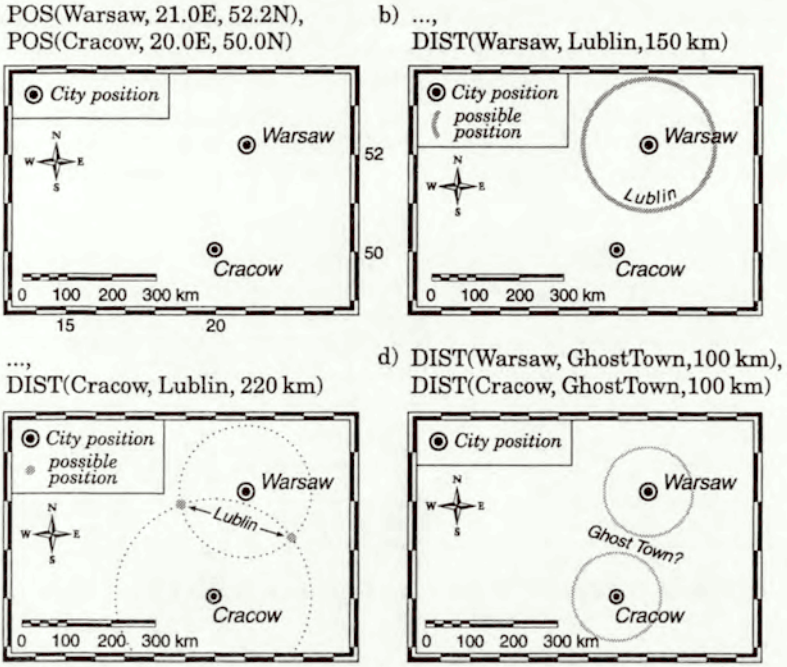


Figure II.17: Representing incomplete information about positions on a map: two cities at given positions (a), possible positions of the third city (b, c), and filtering out the Ghost Town (d).

the language leads not in the direction of representing a set explicitly (as in Fig. II.17b), but rather towards representation of a process of generating the required set from the particular instance represented in the diagram, see Section II.3.2.3 below.

Disjunctive knowledge. The claim here is that diagrams, or analogical representations in general, cannot represent alternative pieces of information about which we do not know which of the alternatives truly holds. Here the incompleteness is usually limited to a number of discrete alternatives, not necessarily closely related to each other. This objection is often formulated in the literature, usually that advocating a logical approach to knowledge representation, see e.g. [Moore, R.C. 1982, Levesque 1986] and many others, but also in the diagrammatics literature, like in [Stenning & Oberlander 1995, Lemon & Pratt 1997].

In the *three blocks* problem discussed in Section II.1.2.4, Example II.4a, a statement was made by the author of the example [Moore, R.C. 1982] that the problem can be solved only by logical means because it requires representation of incomplete information of the disjunctive type which cannot be represented by other means, especially diagrammatic. Contrary to that claim, it is easy to solve it diagrammatically, with disjunctive knowledge easily represented in this form too.

Example II.4d (Diagrammatic formulation) For a diagrammatic solution, it is advisable to translate the original Moore's formulation, see Example II.4a, into a more consistent form (see the discussion in Example II.4b), for example as follows:

> There are three squares arranged in a row. The first square is dark, the last one is not dark, while the middle one may be dark or not. In this configuration, is there always a dark square that is next to a square that is not dark?

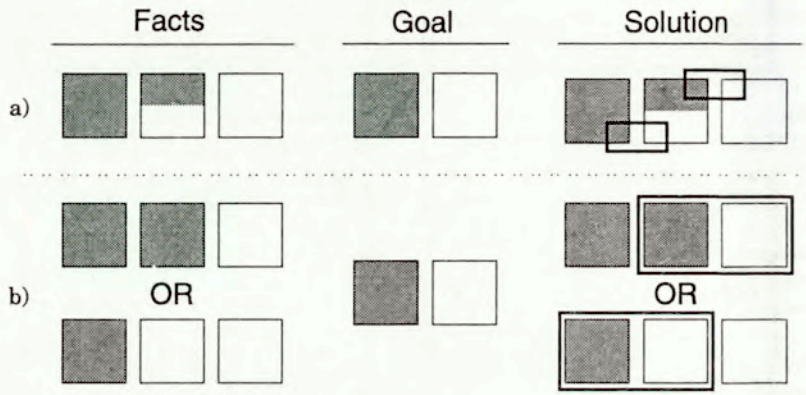


Figure II.18: Diagrammatic representation and solution of the three blocks problem: using extended visual language (a), or divergence (b).

The facts are represented diagrammatically in Fig. II.18a with the disjunctive knowledge about the middle square represented with darkening of half of it, and the goal as a subconfiguration that must occur in the diagram. The solution shows two possible positions of the goal configuration, corresponding to two alternative colourings of the middle square. No matter how the middle square is coloured, the goal configuration can always be found. In Fig. II.18b, two (sub)diagrams are used for two possible cases. ■

The example shows two main methods of coping with the representation of disjunctive knowledge, already mentioned above, namely extending the visual language appropriately, or using divergence, i.e., several (sub)diagrams for different possible cases constituting the disjunctive knowledge in the problem. See Fig. II.33 in Section II.4.3 for another example.

II.3.2.3 Particularity

> An instance will do more than a volume of generalities
> to make my meaning clear.
> [Edwin A. Abbott, *Flatland: A Romance of Many Dimensions* (1884)]

The incomplete information problem for diagrammatic representations in a yet another disguise is called the problem of *particularity* (sometimes also *determinacy* or *indeterminacy*). It was noticed already by Aristotle:

> [In geometrical proofs,] though we do not for the purpose of the proof make any use of the fact that the quantity in the triangle (for example, which we have drawn) is determinate, we nevertheless draw it determinate in quantity.
>
> [Aristotle, *On Memory and Reminiscence* (350 B.C.E.)]

That is, when representing some knowledge about or conducting some reasoning with a whole class of objects (for example all right triangles), we cannot draw the argument (say, the representation of the Pythagoras theorem or its proof, see Example II.15 in Section II.4) for all such objects. We can only draw a determinate *particular instance* of such an object. In consequence, a question arises whether the represented knowledge or reasoning applies only to this particular object, or to a wider class of somehow similar objects, and exactly to which class? It is thus another problem involving representation of sets, see Section II.3.2.2. In such situations, additional mechanisms should be used to delineate precisely the set involved, either in the diagram itself, or outside it (e.g., in an additional part of the proof of the Pythagoras theorem showing that the diagrammatic argument conducted for only some particular triangle can be safely generalized to all right triangles), see also [Jamnik et al. 1999].

This style of proof is a diagrammatic counterpart of a kind of sentential proofs involving the universal quantifier, see e.g. [Winterstein et al. 2002]. In a sentential proof we often quantify over a set, possibly infinite, of objects: "for every $x \in X$...," while a diagram focuses on a specific example in that set, i.e., we are in effect stating: "let x_0 be an arbitrary element of X..." [Barker-Plummer & Bailin 1997]. To complete such a diagrammatic proof step, it is necessary to define in some way the quantified set X and show that x_0 is a truly representative example within that range (i.e., such that the reasoning concerning x_0 that follows applies equally to all other elements of X). This requirement can be met by presenting all other elements of X as simple structural variations of x_0 (or simply defining the set X as a set of all such variations generated by x_0). This is usually implicitly assumed, especially when the definition of a set involved is obvious and the needed transformations of x_0 equally obvious and easy, like in the case of the diagrammatic proof of Pythagoras theorem shown in Section II.4, Example II.15. In less obvious cases, it can often be done by a *structure variation diagram*, see Section II.5.4.5, which rather than representing the set X directly represents instead a process, or recipe, of generating the set from the given instance x_0 of a member of the set.

Another formulation of the problem in terms used here is due to [Ioerger 1992]:

> Knowledge expressed in propositional format can determine part of the state of the world while conveniently leaving other parts undetermined; whereas, an image determines everything about a particular state of the world.
>
> [Thomas R. Ioerger, *Diagrammatic semantics for spatial prepositions* (1992)]

Trying to add such a partial-knowledge fact to the diagram, we are often forced, by the logic of the diagram, to fix also additional details which we possibly do not want to state—the situation, essentially, of unwanted emergence, see e.g. Section II.2.2.

Example II.13a (Particularity) When asked to represent in a diagram the simple fact HIGHER(Fly, Table), where we should place the fly? If it is placed as in Fig. II.19a, the diagram also represents some unintended emergent information about the horizontal

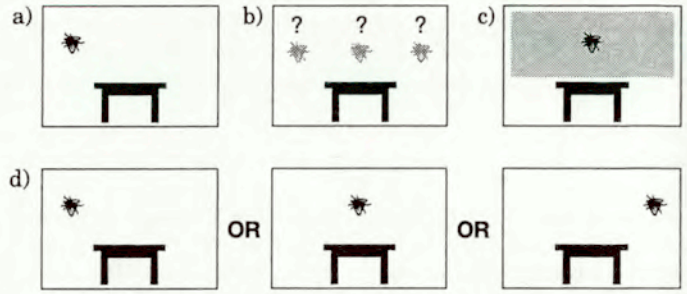


Figure II.19: Representing the fact HIGHER(Fly, Table) seemingly requires also fixing the horizontal position of the fly (a, b); the problem may be solved by extending visual language (c), or by divergence (d).

position of the fly (here—that it is to the left of the table). Placing it in other places does not solve the problem, only the unintended information will be different, Fig. II.19b. In consequence, when using later the diagram we do not know which parts of the encoded data are necessary, and which are arbitrarily assumed because of that diagram-enforced particularity. Moreover, the assumed data may be false in reality, so that we may become prey to the effect of *false emergence* (see Sections II.2.2 and II.4.2.1). ■

Note also that assuming the *negation by omission* rule, we would generate even more possibly unintended facts (negations of those options that we did not choose when deciding where to place the fly), see Section II.3.2.5 below.

Like in the case of incomplete information and disjunctive knowledge, see Section II.3.2.2, the problem can be solved in two main ways, namely using:

Extended visual language, allowing explicit representation of sets of positions, see Fig. II.19c. In this case, all positions of a fly higher than the table are represented by an appropriate gray region, with the image of the fly used as a graphical label for the whole region (see Section II.2).

Divergence, using several (sub)diagrams for different possible cases, see Fig. II.19d. It is practical when there are a few such cases, like here, where possibly only three shown values {LEFT, CENTER, RIGHT} of the horizontal position of the fly will suffice.

Of course, both methods have their disadvantages, especially when the information to be encoded is more complex, but it is obvious that the problem can be solved, at least in principle.

Yet another approach was pursued in [Ioerger 1992], in the context of *truth maintenance*, i.e., the task of adding new knowledge (facts) to a knowledge base without creating inconsistencies in it, see Section II.1.2. The knowledge base in Ioerger's model consists of a list of propositional assertions already processed and a single diagram constituting an instance of the class of all diagrammatic representations that satisfy all these assertions. With a new assertion arriving, the diagram is transformed to incorporate the new knowledge in a manner that does not violate (too much) all the previous assertions.

Example II.13b (Dynamic adjustment) For example, asserting HIGHER(Fly, Table) at the beginning would result in one of the diagrams listed in Fig. II.19d, let us say the first one. Subsequent arrival of the assertion RIGHT(Fly, Table) would result in adjusting the diagram by shifting the fly to the right of the table, but maintaining it still at the upper part of the diagram. ■

In a more philosophical vein, one may spot here an analogy between handling of the particularity problem and the situation where a deeper understanding of a solution to a particular problem may lead to finding some general principle, applicable to many other problems, often quite dissimilar at the first sight. Understanding how apple falls may lead to understanding how planets move... By the way, the latter problem was solved by Newton in his *"Principia..."* in most part diagrammatically. Richard Feynman, trying to repeat Newton's derivation in one of his famous lectures on physics, encountered much trouble at some spots and finally did some parts differently than Newton. As Goodsteins point out in [Goodstein & Goodstein 1996], it was due to the fact that Newton apparently made use of some special properties of conic sections, well known and widely discussed in his times, but apparently forgotten since then. One may wonder how many other diagrammatic techniques were lost in this way during the period of time when diagrams were practically expelled from mathematics...

II.3.2.4 Accidental alignments and general position

Closely related to particularity is the effect of *accidental alignments*, to the extent that, as shown below, some of the cases listed under this name should be rather considered as a kind of particularity. Accidental alignments usually occur due to more or less sloppy execution of the diagram, or due to the very logic of its construction in the particular case considered (e.g., its closeness to some special case). In such situation, some elements of the diagram accidentally become so close to having certain property or to be specifically related to other elements that the apparent look rule is triggered, producing false conviction that the property in question really holds, or should be considered to hold.

Example II.11c (Accidental alignment) This effect can be considered to be responsible for the "64=65" puzzle as well, see Example II.11a. The logic of the construction makes the corner points that are near the diagonal to become accidentally aligned so as to look almost like lying on a single straight line. Taking that property for granted, as a property enforced by the apparent look rule, becomes then the source of error in the example. ■

Another kind, as already remarked, is rather a case of particularity mixed with overlooked divergence, as observed in [Shimojima 1996]. Namely, trying to represent some general case we can draw only a particular instance of the object(s) involved, in this case such an instance which accidentally fixes some important feature at the value valid only for some subclass of these objects. This leads to error if further use of the representation generalizes some result to all objects of the kind, while the reasoning leading to that result critically depends on this particular value of the feature. It is then said that the reasoning referred to an *accidental feature* of the diagram, and the effect is classified as a case of accidental alignment.

Example II.14 (Accidental feature) When investigating properties of triangle heights, one may draw a triangle with only acute angles (Fig. II.20a), and then find and prove for it that all its heights intersect at a single point *inside* the triangle. Generalizing that thesis to all triangles, including obtuse ones (Fig. II.20b)[15] would lead to an error based on an accidental feature—here the *acuteness of all angles* in the triangle used in the proof. Note that the situation can be also easily reinterpreted as a case of *overlooked divergence*,

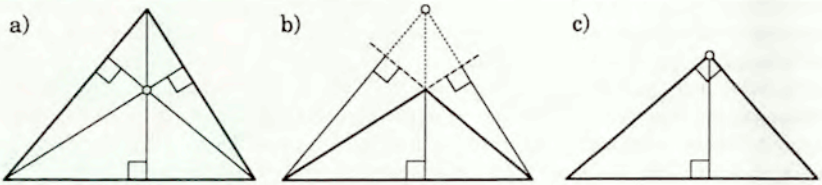


Figure II.20: Triangles with heights intersecting at a point inside the triangle (a), outside it (b), or at the vertex (c).

see Section II.4.3.1, namely, overlooking the fact that there are other cases (here, triangles with obtuse angles) that should either be investigated separately, or else the generalization of the conclusion to them should be explicitly and separately justified. One may also insist on considering also the third case (of right triangles), with the intersection point falling at the vertex (Fig. II.20c). ■

As a remedy for accidental alignments, usually a general advice concerning the proper choice of particular examples of objects to be used for conducting diagrammatic arguments is formulated as the *general position* rule:

General position *rule.* *The object that is intended to represent some general class of objects, about which we want to reason, should be selected and drawn in such a way that it does not exhibit explicit features or relations between them that may be significant for the conducted argument, but which do not hold for all objects of intended class.*

Thus, the rule simply requires, in a more elaborate wording, that no accidental alignments should appear in the diagram. For example, if the intention is to draw a triangle representing *any* triangle, it should not be drawn in a manner suggesting some particular kind of triangle only, like an isosceles triangle (as in Fig. II.15a), or right-angled triangle.

Unfortunately, that is not always possible to attain, either because the structure of the needed class of objects does not allow to represent all relevant cases with a single diagram, or when some important features of the objects involved cannot be freely selected, being, say, fixed from outside, in the formulation of the problem. The former case is actually the case of *forced divergence*, discussed in more detail in Section II.4.3. It occurs in Example II.14, where it is impossible to draw a triangle having simultaneously both all acute angles and one of them obtuse. Therefore, two (or three) separate cases should be considered there for any arguments depending on types of angles in the triangle.

[15]Note the interesting symmetry (see Section II.3.1.5) between constructions in Fig. II.20a and b. It would occur for any side of the triangle taken as a base.

II.3.2.5 Specificity and negation by omission

The feature called *specificity*, or more elaborately, *enforcement of non-identity between tokens*, means that two distinct elements of the representation cannot denote the same object in the represented domain. In logical representations this means that constants with different names always denote distinct objects. Therefore, specificity is in fact the *closed world assumption* (see Section II.1.2) applied to the equality predicate [Davis 1990]. In a diagram it means that two distinct graphical elements or properties denote different objects as well.

In logical representations, the assumption can be obviously either taken or not. Many authors claim, however, that in diagrammatic representations we do not have that choice—allegedly, in a diagram we are always forced to be specific in this way. Fortunately (or not), this is not true. Examples of the same objects represented by several graphical tokens in a diagram abound. See for example Figs. II.6d, II.28, II.29, II.30d, and II.32-II.34. In these figures, the identity of objects denoted by different graphical elements is fixed by the use of textual labels, hence, one might argue, by propositional means. However, it can be easily done by other means as well, like in Fig. II.13b, c, and d (pieces of rectangles), and in Figs. II.21 and II.22 (corresponding triangles within two squares). In these cases the identity is ascertained by the sameness of shape (and orientation) of the pieces, reinforced in the latter case by a consistent gray fill. In Fig. II.18b the identity is indicated also by alignment of squares, despite different gray fills of the middle ones. A more complex example is provided by technical drawings of machine parts, where the same object is very often depicted by different graphics showing its view or projection from different sides. The specificity also does not hold in multithematic maps of the same area, where different tokens (often placed in different submaps or inserts) denote the same object (e.g., a city) without problems, see e.g. Fig. II.17. The argument that in those cases we have several different diagrams, not a single one, sounds rather hollow, as the distinction between a single diagram and several distinct diagrams is never absolute and seldom clear, and the examples using undoubtedly a single diagram abound, see Fig. II.33.

A similar issue concerns the use of the *negation by omission* (NBO, for short) rule. In diagrams, it means that if some object, or relation between objects, is not explicitly drawn, it should be considered that it does not exist in the represented domain. It is often claimed that diagrammatic representations force the rule—i.e., that it is actually impossible to draw diagrams without the rule becoming automatically assumed [Levesque 1986, Stenning & Oberlander 1995]. Fortunately, it is not true—in practical uses of diagrams often the opposite is assumed. Consider first the "cat on the mat" example in some more detail.

Example II.10b (Cat not on the mat and the NBO rule) In Fig. II.12b there is no cat drawn on the mat. It seems that the reasoning there is based exactly on the application of the negation by omission rule, automatically assumed for the diagram. If this was the case, the conclusion in Fig. II.12c would be "no cat is on the mat." However, our conclusion in Fig. II.12c concerns not any cat, but only *the* particular cat which is outside the room, as required by the input data and as accordingly drawn. *This* cat certainly is not on the mat, NBO rule or not.

It is important, however, to admit here that the absence of the drawing of a cat on the mat plays a significant role in the reasoning. Presence of an image of a cat there would

mean that there indeed is a cat on the mat, although not necessarily *the same cat* as that outside the house. Its presence would confuse issues, the more because of the specificity issue, i.e., possible doubts if there are two distinct cats, or the same cat (in the latter case such a construction can be used to express the fact that we do not know whether the cat is on the mat, or outside it). ■

Generally, the use of the NBO rule is practical in some circumstances only, no matter whether in propositional or diagrammatic representations. Not always it is possible to get rid harmlessly of the distinction between facts that are not true and facts that are merely unknown (and just this distinction is eliminated by the NBO rule). For example, in Example II.13a the absence, in Fig. II.19c, of the fly within the top gray rectangle in regions to the left and to the right of the table does not mean that there is no fly there, but merely that we do not know its exact position relative to the table (except that it is above the table).

The possible use (or not) of the NBO rule in diagrams may produce also problems with interpreting emergence. Namely, if we assume the NBO rule, then all implicit facts generated by it should be considered as already stated, thus they will not qualify as emergent facts, see e.g. Fig. II.6f in Section II.2.2 and Example II.2 in Section II.4.2.1.

All the above considerations tell us that many confusion and errors in using diagrammatic representations may come from implicit assumptions concerning some interpretation conventions, like specificity or the NBO rule discussed in this section. Despite claims, these conventions are in no way automatic and compulsory in interpreting any particular diagram. To avoid misunderstanding, they (or their negations) should be explicitly stated as part of the specification of the visual language used in any given case.

II.3.3 Diagram application modes

Diagrammatic representations are used in a wide variety of applications, for different purposes and in different ways. The manner of their use can be classified according to the following main criteria:

What kind of activity is to be aided:
an information *encoding*, or its *processing*?

Who created the diagram:
a *human* or a *computer*?

Who is to use the diagram:
its *creator*, or *someone else*, and again,
a *human* or a *computer*?

Where is the diagram held:
within the *agent's mind*, or *externally*?

The first of these classification criteria divides diagram applications into two major kinds described below. This division is apparent in the descriptive name of diagrammatics as "diagrammatic representation and reasoning."

II.3.3.1 Information representation (recording)

Here diagrams are used mostly as means of recording (encoding) the information or knowledge. The main purpose of such a representation is usually as a means of more or less direct visual *communication* of that data to the *other party*. This kind of application is sometimes called *infographics* or *presentation graphics*, although these terms are usually understood in a more restricted senses (like assuming only human recipient of the diagram, or communicating only quantitative information). Depending on the nature of the agents involved and the direction of information flow, one may distinguish the following subspecies of this domain:

Human → human: here we have the traditional disciplines of *graphic design*—of information signs and signboards, posters, advertisements, (statistical) graphs, publication layouts, etc., for communication, advertising, instruction, etc.

Computer → human: here we have computer *data (knowledge) visualization*, including *scientific visualization*, and computer aided or automatic *presentation design*.

Human → computer: here we have *graphical (diagrammatic) data input*.

Human ↔ computer: this is the domain of man-machine *graphical interfaces*.

Diagrams for communication purposes tend to contain comparatively small amounts of data, the time from creation to use of the diagram is usually short, and the diagram is often discarded soon after its use. Because of that there is little possibility to accompany the diagram with a specification of the visual language used. Hence, the language is usually provided only implicitly, as either a general convention customary in the given context (usually then it cannot be too complex and intricate), or an already codified notation well-known to the specialists involved in this particular diagrammatic communication task.

Another branch of diagrammatic representation applications uses diagrams mostly for storage or archiving purposes, as (part of) data or knowledge bases, to be used later by the same (or the other) party. Diagrams for this purpose usually contain rather large amount of data (often one of the basic issues here is how to achieve high information density for compact storage), the time from creation to use of the diagram may be arbitrarily long, and the diagram is often used many times. As examples may serve pictorial or graphical catalogues and manuals, detailed maps, libraries of construction and engineering drawings, circuit schemata, etc. Here the visual language used can be explicitly specified together with the diagram (or a set of diagrams). That allows for use of much richer and intricate visual languages, specifically tailored to the particular application at hand.

II.3.3.2 Information processing (reasoning)

In this mode of application, diagrams are used by humans, machines, or man-machine systems, mainly for *information processing*, especially for inferring new facts from the ones encoded in the representation. In the latter case, the domain is called also *diagrammatic reasoning*, see Section II.4 for more detailed discussion of issues involved here. Of course, a prerequisite for making diagrammatic inferences is to first *represent* all the relevant

facts in a diagram. Hence, the issues of diagrammatic representation of knowledge are of basic importance for diagrammatic reasoning as well.

Further important distinction within this mode of diagram use is between:

Internal use of diagrams, when they exist only in the mind of their creator and are used privately by the creator; if it is a human, we have the case of *visual imagery*, [Kosslyn 1980, Kosslyn 1994].

External use of diagrams, when they are produced outside the processing agent, to be read off and possibly modified repeatedly in the course of reasoning, with the possibility to be shared with other agents.

As the diagrammatic reasoning constitutes the most important type of diagrams application in mathematics (and related disciplines), which in turn is of particular interest for this work, the next section is devoted entirely to a more deep discussion of diagrammatic reasoning.

II.4 Diagrammatic reasoning

Diagrammatic reasoning is the only really fertile reasoning.
[Charles S. Peirce (1839–1914)]

The use of diagrammatic representations for reasoning, i.e., to *process* encoded knowledge or information (premises, assumptions, already known or proven facts, etc.) in a way that allows for production, with the help of that representation, of some new knowledge (conclusions, inferences, theorems), is the most important application of diagrams in areas like mathematics, which are of most interest for us here (see Chapter III). Therefore, *diagrammatic reasoning* is discussed here in more detail.

A single act of diagrammatic reasoning can be considered to consist of three main steps:

Encoding: first, the starting information (premises, assumptions, already known or proven facts, etc.) comprising the formulation of the reasoning problem should be represented in the diagram (or diagrams) in accordance with the *visual language* appropriate for the given application.

Processing: next, the diagram should undergo a sequence of *transformations* selected from the repertoire allowed for the given reasoning task, just like transformations of formulae used in solving problems in propositional representations.

Decoding: finally, the required result of processing (conclusion, thesis of a theorem, etc.) should be *read off* the resulting diagram, and possibly represented in some other required form (e.g., a formula).

In some cases the second step can be reduced to nothing: the very process of producing the diagram to encode the premises produces the structures in the diagram that can be immediately read off as conclusions, without any need for further operations. This so-called *emergence effect*, originating from the analogicity of diagrammatic representations (see Section II.3.1.3), is very important for efficiency of diagrammatic reasoning and is discussed in more detail in Section II.4.2 below.

One of the common mechanisms of diagram transformations during (especially formal) reasoning uses diagram rewriting rules, modelled on the *rewriting systems* concept used in propositional representation systems, especially in the theory of computations (e.g., the Post system) and mathematical linguistics (rewriting grammars). The expert system-like reasoning with logical representations, discussed in Section II.1.2.1, generally also belongs to this type. The diagrammatic version of the concept is used in the pulley system example, see Example II.3g in Section II.3 (Fig. II.10). It can be used to advantage also in computer implementations of diagrammatic reasoning, either the ones based on graph grammars and transformations [Grabska 1993a], see Section II.6.2.2, or the kind based on a local rewriting on a raster [Furnas 1990, Furnas et al. 2000, Gardin & Meltzer 1989], see Section II.6.2.1.

Two other simple examples will introduce other aspects of the matter at hand.

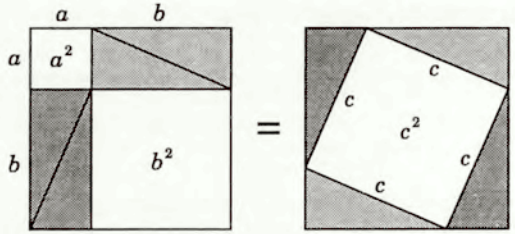


Figure II.21: A simple diagrammatic proof of the Pythagoras Theorem.

Example II.15 (Pythagoras Theorem) The classic diagrammatic proof of the Pythagoras theorem shown in Fig. II.21[16] involves two squares of equal size obtained from each other by rearrangement of the four right triangles inside. Thus, we have there already two diagrams—the original and transformed one. Reading off the conclusion amounts to performing the following simple calculation:

$$\begin{aligned} a^2 + b^2 + 4\triangle &= c^2 + 4\triangle, \\ a^2 + b^2 &= c^2, \end{aligned}$$

which can be also done visually in the diagram by reducing out the identical gray triangles occurring on both sides of the equal sign.

The alphanumerical labels used in the diagram are not necessary for the proof itself; they only serve to connect it with the standard formula expressing the theorem. Another diagram (Fig. II.22) for essentially the same proof performs even better without labels. Note also that several significant features (like rightness of certain angles) are stated here with the apparent look rule in mind (see Section II.3.2.1), while other features (like equality of several line segments) are stated explicitly using textual labels (compare with Figs. II.15 and II.31, where graphical labels are used). It may be also interesting to note that the first of the pair of diagrams in Fig. II.21 contains also a diagrammatic proof of the binomial formula, see the vignette to Section II.4.1.2 below.

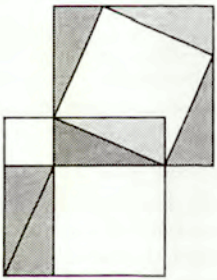


Figure II.22: Behold!

■

A collection of such diagrammatic proofs has been published in a book [Nelsen 1993], see Section II.5.3.1. To make such a proof fully rigorous, additional considerations are necessary, in order to assure generality (here, that the argument safely applies to all right triangles, independently of their sizes, proportions, etc., see e.g. [Jamnik et al. 1999, Winterstein et al. 2000]), and to assure absence of some common errors, see Section II.5.2.

The use of diagrams for reasoning is not restricted to mathematical proofs. Another common example is an extraction of useful information from numerical and statistical data visualized on various types of graphs and charts.

[16] Attributed by [Nelsen 1993] to an old Chinese treatise, and by [Jeleński 1968] to Pythagoras himself.

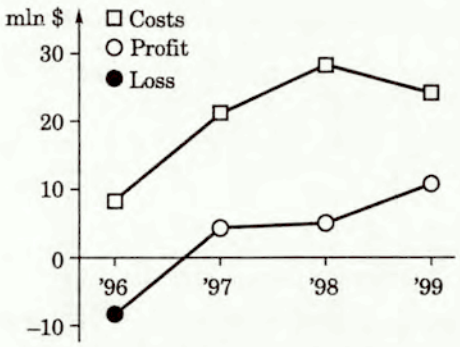


Figure II.23: A simple statistical graph about financial history of a company.

Example II.16 (Company statistics) In Fig. II.23, a set of twelve numbers representing the history of costs and profits for some company over the period of four years is presented with a simple graph. What can be read off that graph?

Concerning the original numbers, they can be easier and more accurately read off from a list of numbers, without the expense of producing that graph. What such graphs are really for is something different—namely, a possibility to see at a glance some general conclusion, i.e., a result of some *reasoning* that follows from the interaction of these numbers. Here, the diagrammatic reasoning may produce something like "Seemingly, we finally ended the investment period and are starting to reap the benefits." Reasoning using graphs may involve much more as well, including sophisticated graphical calculations, as is the case with various kinds of nomograms, see Section II.4.1.1 below. ■

The use of various graphical presentations of data, also quantitative or statistical, as tools for finding new information in the data, is as common as the use of them for simply presenting some data for easy access. See e.g. [Tufte 1997] for a case study of how the wrong graphical presentation of statistical data hampered proper conducting of a simple reasoning that might have prevented the tragic Challenger space shuttle disaster in 1986. There also exist sophisticated techniques of *graphical processing* of data to reveal hidden regularities and dependencies, see e.g. [Bertin 1981]. The entire subfield of *scientific visualization* aims at helping scientists to conduct their reasonings by proper visual presentation of vast multitudes of experimental numerical data.

Despite the lack of rigorous and comprehensive methodology of diagrammatic reasoning, some general issues and mechanisms can be distinguished. These issues include the three general diagrammatic reasoning modes (Section II.4.1), as well as the effects of *emergence* (Section II.4.2), and *divergence* (Section II.4.3. Some other effects pertaining to the use of diagrams for reasoning were also mentioned on the occasion of discussing various advantages of and problems with diagrammatic representations in Section II.3.

One of the important issues here is the problem of *reliability* of diagrammatic reasoning—advanced as one of the main arguments against the possibility of rigorous use of diagrams in mathematics, see Section II.5. Therefore, the problem of unravelling possible causes for making errors in diagrammatic reasoning will be given a particular attention in this section (see also Section II.5.2).

II.4.1 Quantitative and qualitative reasoning

Basically, features of diagram elements, relevant to the process of diagrammatic reasoning, can be divided into two main classes:

Quantitative: concerning, or based on, some quantitative measurement or observation with a numerical value, like line length, number of points, etc. Basically, such features as alignment of points along a straight line or perpendicularity and parallelism of lines also belong here. Although they have a certain qualitative flavour (see below), they are nevertheless limit cases of some quantitatively defined property, like a distance from a line or an angle between lines.

Qualitative: concerning features not based on fine numerical distinctions, but due to qualitatively distinct configurations or gross value differences.

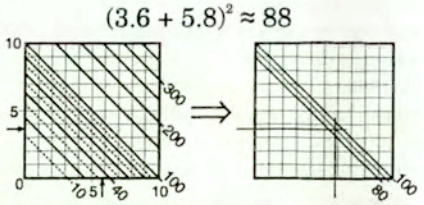
The difference between these two classes is not always clear-cut and doubtful cases are not uncommon. E.g., it is not clearly defined what exactly qualifies as a "gross value difference" due to which some quantitative difference becomes qualitative. Despite that, the distinction is quite useful in most practical cases, as will be shown below.

A certain class of intermediate situations, having some properties of both the above types, is worth to consider separately:

Discrete counting: here quantitative distinctions are based not on a potentially continuous-valued measurement, requiring in principle an infinite precision, but instead on a discrete counting of structurally (qualitatively) distinct elements or tokens.

To these three classes of features correspond naturally three different modes of dealing with diagrams in reasoning, with different properties concerning their reliability. These modes are based on, respectively, *metric reasoning*, *structural (topological) reasoning*, and *discrete token counting*.[17] Let us discuss them in more detail.

II.4.1.1 Metric reasoning



In this kind of reasoning, the argument concerns, or relies on, a precise measurement of some continuous quantity, like length, distance, angle, area, etc., or on testing of some property requiring such precise measurement or alignment, like equality (or inequality) of such quantities, parallelism of lines, collinearity of more than two points, etc. For a fully reliable reasoning with such *metric features*, both the construction of a diagram and reading off data from it should have to be infinitely precise, which is obviously impossible in practice, see Section II.3.2.1.

It does not mean that *all* inferences involving such features are necessarily unreliable. There are several possibilities available here. One consists of checking the necessary conditions by other (non-metric) arguments, for example discrete, structural, or qualitative (see below and Section II.4.2).

[17] This classification is in principle unrelated to that used in [Jamnik et al. 1999], though the token-counting type may superficially resemble one of the classes discerned there.

Example II.11d (Checking metric features) The above prescription can be easily applied to the *"64=65"* puzzle. The problem would not arise there if we observed the rule that such an essentially metric feature as collinearity should not be taken for granted or judged "by the eye" alone, without closer inspection and verification using a more reliable evidence. In this case, as one should (easily) verify using a reliable discrete token counting in the diagram of Fig. II.13b (see Example II.11e), the slopes of the cuts of the two rectangles are not equal, hence the corresponding diagonal cuts cannot be aligned into a single line suggested by Fig. II.13d. ■

As follows from the above, the error does not necessarily follow from some inescapable, intrinsic unreliability of diagrams. The blame should be rather laid on the human reasoner for not observing the proper precautions specific to the diagrammatic mode of reasoning. This source of error is the most common one in diagrammatic reasoning. Other kinds of errors are often based on this foundation as well, see e.g. emergence and divergence errors discussed below.

Graphical computation. When quantitatively approximate results are acceptable, the reasoning based on metric representations can be made completely valid and quite useful. Numerous generations of engineers using nomograms, slide rules, and other graphical calculation methods may certify to that. A brief but informative history of such numerical graphs can be found in [Hankins 1999]. In Fig. II.24 some basic examples are shown. Another one is used as a vignette to this section. The true nomograms (of the kind shown in Fig. II.24b) were invented by d'Ocagne in 1884, see [d'Ocagne 1899, Hankins 1999], and since then they were, together with a slide rule (in itself a kind of easily adjustable nomogram) in extensive use in engineering calculations till the spreading of computers. Nomograms can be made quite complex, connecting many variables related by complicated functional dependencies, like 105 physicochemical parameters of blood in the famous nomogram of 1928 by L.J. Henderson, see [Hankins 1999]. A more recent source on various graphical computation techniques is, among many others, [Hoelscher et al. 1952].

These techniques of approximate calculation are based exactly on a more or less diagrammatic reasoning involving continuous quantities, usually marked on and read off various numerical scales. This makes them partially qualitative, as described in the next paragraph.

Qualitative metric reasoning. With the use of scales, continuous measurement theoretically needed to conduct continuous graphical calculations is effectively replaced by token counting (of marks on scales) and qualitative reasoning (finding the scale marks between which some other mark lies). With that, the computation is made more reliable, within the limited accuracy specified by the numerical distances associated with marks on the scales. Some other, essentially metric inferences can be made reliable if it is possible to formulate or conduct them in a qualitative way, as shown with the following example.

Example II.17 (Inclusion of points in a set) In Fig. II.25, the set S is defined with a formula and then represented diagrammatically as a shaded triangle within a coordinate system. If the question is to decide which particular points (represented in the diagram by small circles) belong to the set S, we can see that for some of them, e.g. the points

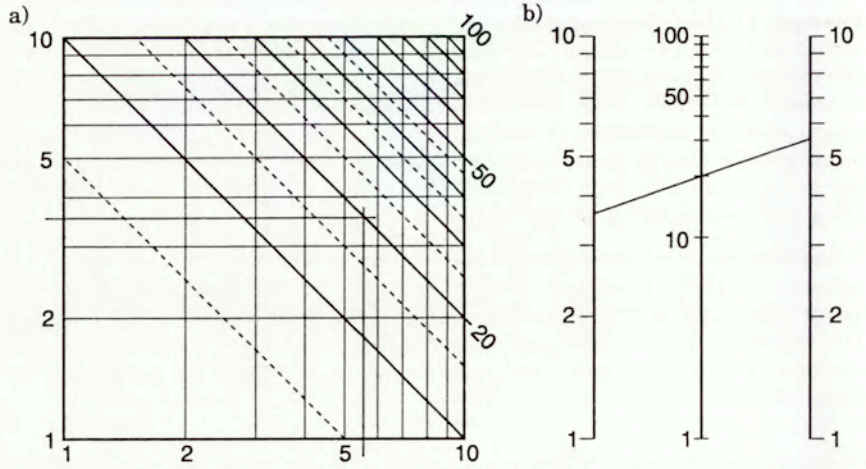


Figure II.24: Multiplication with metric diagrams: an isoline Cartesian graph (a), and a classic nomogram (b); both with the calculation $3.6 \cdot 5.5 \approx 20$ shown.

with coordinates $(-1, 0)$, $(0.5, 0.5)$ and $(1.5, 1)$ it can be done reliably "by the eye." It is so because their position with respect to the set S is sufficiently qualitative, not requiring much precision of drawing or sight to ascertain the proper inclusion relation. This is different for the points $(0.49, 0.76)$ and $(1.01, 0.49)$: to decide their inclusion relation with the set S there is practically no other way (within a reasonable drawing accuracy) than calculation involving the equation for the hypothenuse of the triangle. The point $(0.5, 0.01)$ represents a sort of an intermediate case—while purely diagrammatic inference may be not reliable enough for it, a mere glance at its second coordinate and noticing that it is greater than zero suffices to decide on its inclusion in S (provided the condition $y > 0$ is reliably readable off the diagram with the visual language used). ■

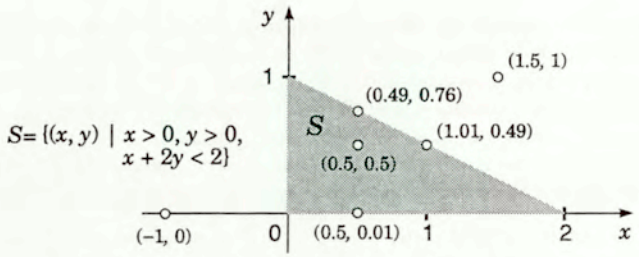
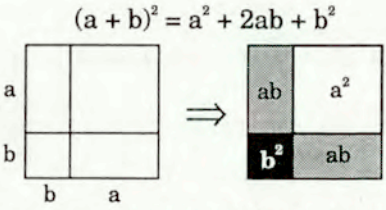


Figure II.25: Testing visually inclusion of points in a set: quantitative versus qualitative metric reasoning.

Replacing metric reasoning with structural one. An example of this kind of a solution is provided in Section II.3.2.1 (Fig. II.15b). It was adopted in the formal systems for geometrical diagrams developed in [Luengo 1995] and [Miller 2001]. From these systems (almost) all metric features were cast out, so that rules of diagram construction and inference allow only topological features and relations between diagram elements. In this way, a (geo)metric representation became *structuralised*, see the next section. The word "almost" used above refers to the fact that in Luengo's system a single metric feature—namely, that straight lines should be always actually straight in a diagram—was retained. Unfortunately, that led to a serious flaw in her system, as explained in [Miller 2001]. The same structuralisation method is also used in the GROVER diagrammatic preprocessor for an automatic theorem prover called "&" of [Barker-Plummer & Bailin 1997].

Such structuralisation avoids problems with unreliable metric features for the price of proliferation of structurally different cases, often impossible to construct in the given situation (see the *false divergence* problem in Section II.4.3.2) which therefore must be separately checked and eliminated—a problem that is NP-hard in general, see [Miller 2000, Miller 2001].

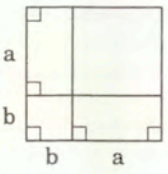
II.4.1.2 Structural reasoning



Reasoning is here based on properties of a structural and qualitative character, not affected much by the correctness of the drawing or the metric accuracy of reading of its features. Such features (and the corresponding reasoning mode) are sometimes called "topological." Features of this type include such things as an existence of certain regions and their inclusion relations, existence of intersection points, and immediate inferences like recognizing a square from equality of sides meeting at right angles. Inferences based on such features are generally reliable. That does not mean that one cannot make an error here. The most common sources of error include mistaking of such features for essentially metric ones (see Section II.4.1.1), and falsely assuming a possibility of occurrence of certain invalid configurations. The latter type of error (called here *false divergence* effect) can be often attributed to the drawing (im)precision anyway, see the examples in Section II.3.2.1 and Section II.4.3.2.

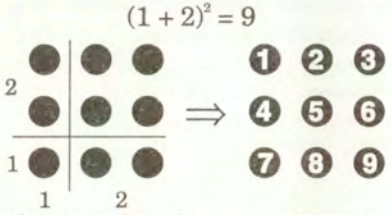
This reasoning mode is most common in diagrammatic and diagram-aided mathematical proofs, due to its reliability and generality. Not surprisingly, the formal diagrammatic reasoning systems for geometry developed in [Luengo 1995, Miller 2001] are based (almost) exclusively on this reasoning mode, despite the fact that geometry concerns metric features as well, see Section II.4.1.1 above for more details. Most of the diagrammatic reasoning examples included in Section II.4 are of this kind, hence only the simple example of a binomial formula, used as a vignette to this section, is provided here.

Mistaking metric and structural features comes often from the careless use of a too simple visual language, cf. Section II.2. For example, the fact that the lines in the figure above are perpendicular, thus forming perfect squares and rectangles, is not explicitly indicated by appropriate visual language means (see also Section II.4.2.2 below). One must thus



implicitly assume that fact, prompted by the looks of the drawing which can easily be deceptive, as essentially metric in this case. Rather harmless in this particular situation, such simplifications of the visual expression, quite common in many "proofs without words" of this type (see Section II.5.3.1) may easily lead to erroneous assumptions, as Example II.23 in Section II.4.3.2 demonstrates. To avoid such consequences, a more reliable diagrammatic expression for the first part of the figure should be provided, like that shown here, or else the visual language used must explicitly rely on the *apparent look* rule, see Section II.3.2.1.

II.4.1.3 Discrete token counting



The diagram here consists of discrete objects (tokens), and the reasoning is based on a simple counting of (various groups of) these tokens. This is the type of reasoning conducted by a computer-implemented *Diamond* system of [Jamnik et al. 1999]. As the distinctions to be made in the diagram are here basically structural, independent of precise measurement of continuous quantities, this mode is usually reliable, provided one does not make some gross misinterpretation or counting error. Various form of abacuses used for centuries were also based on this principle.

Example II.18 (Sum of odd naturals) A classic diagrammatic proof (attributed to Nicomachus of Gerasa, see [Nelsen 1993]) of a formula for the sum of odd number series is shown in Fig. II.26. Despite the quantitative nature of the result, the proof uses a reliable structural reasoning, see the previous section. ■

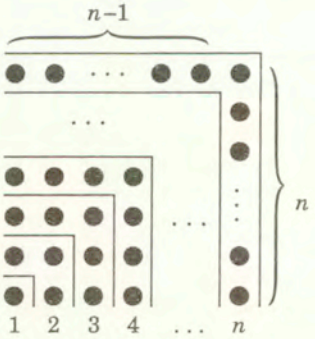


Figure II.26: A diagrammatic proof of the formula $1 + 3 + 5 + 7 + \ldots + (2n - 1) = n^2$ using token-counting reasoning.

Example II.11e (Reliable token counting?) Note that the *"64=65"* puzzle, see Section II.3.2.1, is disguised as just this kind of problem, with its counting of discrete square cells. However, the crucial element of the reasoning (collinearity of the four points along

the diagonal) is of a quite different nature, as it was shown in Section II.4.1.1. Nevertheless, the reasoning leading to precise assessment of the directions of sides of the diagonal quadrangle in Fig. II.14 can be conducted reliably with a token counting argument. After counting cells spanned horizontally and vertically by the diagonal lines in Fig. II.14b or c, one sees that the slope of the cut of the larger rectangle equals 2/5, while in the smaller one it equals 3/8.

Note, however, that the mere observation of that inequality of slopes does not resolve in itself the question whether there is a gap, or else an overlap, along the diagonal of Fig. II.13d. To settle that, one must find which of the slopes is smaller than the other. It can be done either propositionally, by comparing the corresponding fractions above, or diagrammatically, using a discrete counting argument again, on the square lattice of Fig. II.14. The appropriate diagrammatic argument is shown in Fig. II.27. Otherwise, one may conclude that there must be a gap (of area equal to 1) in the final rectangle from the comparison of its area with the total area of initial rectangles, see Example II.11b. ■

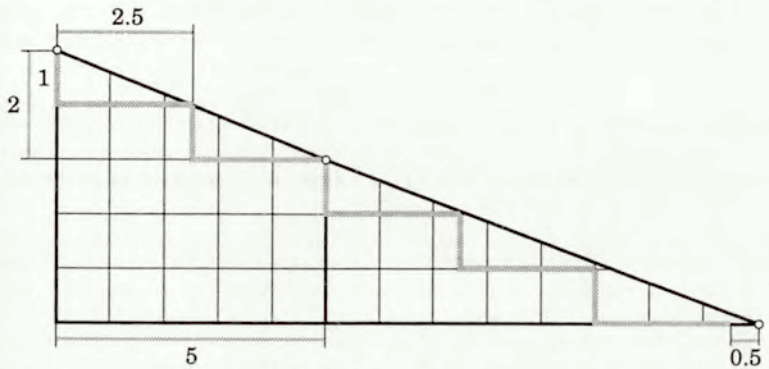


Figure II.27: Comparing slopes of segments of the diagonal using reliable token counting (using the discrete token with half of the original cell width).

Remark. The example diagrams used for vignettes of the three preceding subsections allow an interpretation that structural reasoning is rather abstract and general, while the other two modes are concrete and particular. It is not a valid impression, as the true difference is between *qualitative* and *quantitative* character of these modes which, although related to the *general* versus *particular* distinction, is nevertheless markedly different from the latter. As shown in Example II.18, the token counting mode can be used to prove general theorems without any basic restrictions. Also, the graphical computation examples (including that shown in the vignette diagram of the metric reasoning section) do not represent some particular calculations only, but rather quite general functional dependencies that may be used for countless particular calculations, not only those shown in the provided examples.

II.4.2 Emergence

The behaviour of the body as a whole will then emerge
as a consequence of interactions of its parts.
[Richard Dawkins, *The Blind Watchmaker* (1986)]

The Internet *Dictionary of Philosophy of Mind*[18] defines emergence traditionally as:

Definition II.7 (Emergence: summation of properties) *Properties of a complex physical system are* emergent *just in case they are neither*

(i) *properties had by any parts of the system taken in isolation nor*
(ii) *resultant of a mere summation of properties of parts of the system.*

Emergence effects occur in many diverse contexts, from biology and psychology, through design of physical and information artefacts [Grabska 2001], to art. There is a multitude of definitions of the concept, more or less different than Definition II.7. This definition is also not very useful for our purposes, especially as it is rather unclear what is meant there by "mere summation of properties." Here a better formulation seems to be:

Definition II.8 (Emergence: implicit made explicit) Emergence *occurs when a component or property implicit in some structure (due to its particular composition from parts, but not belonging to any of these parts) appears (i.e., emerges) as an explicit component or property of the structure as a whole.*

In the context of knowledge representation it occurs when the act of encoding some items of information produces a representation in which encodings of some other, not originally considered items of information pop up explicitly. As indicated in Section II.3.1.3 (Example II.10), it is a consequence of analogicity of representations. The effect may occur unexpectedly, being not intended by the creator of the representation, or it can be consciously introduced, for example to increase efficiency of the representation. In this sense, it is closely related to the notion of "conversational implicatures," see e.g. [Marks & Reiter 1990, Mackinlay & Genesereth 1985], the difference being that these implicatures are not "physically" present in the representation (as in the case of emergence) but are rather assumed on the basis of context or certain pragmatic conventions customary in acts of communication of a given type [Mackinlay & Genesereth 1985].

In diagrammatic systems devised specifically as reasoning tools, emergence can be an important factor simplifying the process, as the following example shows. The example involves the simple (and extensively studied, see e.g. [Hammer 1996, Shimojima 1996]) diagrammatic language of *Euler circles*, used to solve syllogisms or for other simple logical or set-theoretic reasoning. Here (and in other places of this work) we will assume the semantic convention for these circles stating that the regions produced by *overlapping* of several circles always represent *nonempty* sets.

Example II.19a (Euler circles: emergence) Let us conduct a simple abstract reasoning, shown in Fig. II.28a, while in Fig. II.28b one of the possible real life concretisations of the schema is shown.

[18] http://www.artsci.wustl.edu/~philos/MindDict/

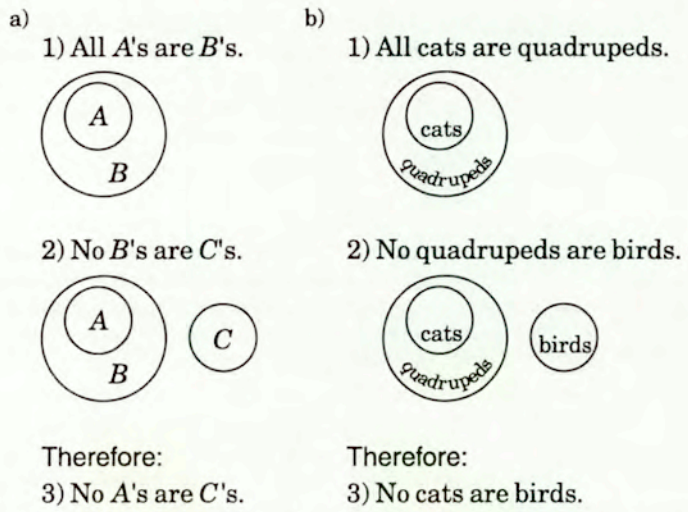


Figure II.28: Simple syllogism solved with Euler circles with the help of emergence: abstract version (a); and real life example (b).

The first premise states that all objects having the property A (being of type A) have also the property B (are of type B), i.e., propositionally, $(\forall x)\mathrm{A}(x) \Rightarrow \mathrm{B}(x)$. In a set-theoretic formulation it means simply $A \subseteq B$, which is naturally represented by the topological inclusion of appropriately labelled circles. To that we are adding the second premise, stating that no objects having the property B (being of type B) may have the property C (be of type C), i.e., $(\forall x)\mathrm{B}(x) \Rightarrow \mathbf{not}\,\mathrm{C}(x)$, or $B \cap C = \emptyset$, so that the circle labelled C is drawn outside the already present circle B. From such obtained diagram we can immediately, without further actions, read off the conclusion, as the circle C is positioned obviously outside the circle A. Combining both premises on a single diagram causes the emergence to do all the inference work for us. ■

As shown by the above example, the very fact of representing some data may cause the inclusion in the representation of some other data, ready to read from the representation without performing any additional operations. In the context of reasoning, the emergent data can often happen to be the desired reasoning results. Thus, emergence becomes one of the mechanisms of inference, especially advantageous due to being cheap and, in a sense, "automatic."

Interestingly, a similar emergence effect can be observed also in the predicate formula representation of the above reasoning. After stating the first premise using the formula $(\forall x)\mathrm{A}(x) \Rightarrow \mathrm{B}(x)$, the second premise, instead of being put down as separate formula, can be integrated into the formula for the first premise, producing as a result $(\forall x)\mathrm{A}(x) \Rightarrow \mathrm{B}(x) \Rightarrow \mathbf{not}\,\mathrm{C}(x)$. From that formula we can read immediately the conclusion $(\forall x)\mathrm{A}(x) \Rightarrow \mathbf{not}\,\mathrm{C}(x)$. Note, however, that the process is rather informal and needs additional elements to make it acceptable from the viewpoint of mathematical rigour (e.g., we should formally define some additional notational conventions utilizing the transitivity

properties of implication). In the set-theoretic formulation, though it corresponds more directly to the diagrammatic representation, the above trick does not work well, as the combined formula $A \subseteq B \cap C = \emptyset$ is not well-formed and has no obvious interpretation leading to the conclusion $A \cap C = \emptyset$. Notational conventions needed to correct the situation would be rather cumbersome and inconvenient to use.

The effect above offers also a counterexample to Shimojima's diagrammaticity criterion [Shimojima 2001] which classifies as diagrammatic all representations that exhibit the emergence effect. Nevertheless, it is true that emergence occurs rather in analogical representations which model more or less directly the structure of the represented domain, see also Section II.3.1.3. Its occurrence in some propositional representations is rather spurious and may indicate the existence of some analogical elements in them. Indeed, mathematical notation contains some amount of such analogical diagrammatic elements, cf. also Section II.5.

The formalization of diagrammatic reasoning systems goes usually in the direction similar to that taken by the formalization of propositional inference, namely the reasoning is divided into small, elementary steps involving only primitive acts like recognizing a simple diagram elements and changing them in a simple way (e.g., erasing the element). This produces a mechanical diagram rewriting system using a number of simple (individually mostly meaningless) rules. As a result, emergence is usually expelled from such a system.

Example II.19b (Euler circles: formal) The reasoning conducted in Fig. II.28 might look in a formalized system of diagrammatic reasoning in a way shown in Fig. II.29. The premises are there stated separately (note the similarity with Fig. II.6d in Section II.2.2), then combined with a unification rule (usually the most complex kind of rule in such systems) into a single diagram. Direct reading of the conclusion off that diagram would involve recognition of certain relations between graphical primitives within the context of other primitives. It is possibly considered too complex an action for a formalized system, because it is not allowed in it—instead, a rule explicitly erasing the B circle must be first invoked, after which the inference result may be read off the now elementarily simple final diagram. In this way, to assure simplicity of rules and mechanicalness of their application (a doubtful advantage for a human user) we sacrifice meaningful links with semantics

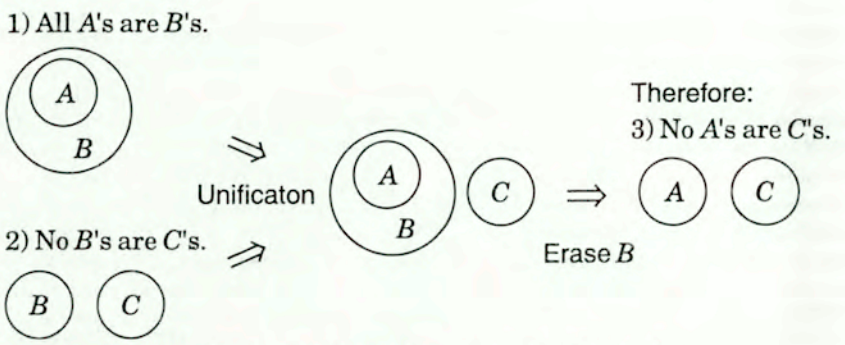


Figure II.29: Simple syllogism solved with Euler circles: formalized diagrammatic system version with no emergence used.

of the domain and effectiveness provided by the emergence effect and a possibly smaller number of (larger) inference steps (see also Section II.1.2). ■

As the above example shows, the occurrence of emergence depends on the way the diagram is used. The example in Section II.2.2 (Fig. II.6) shows also its dependence on the visual language used to encode the information. Within the same visual language, different construction of the diagram for the same input data may also affect the emergence effect significantly, cf. Fig. II.6c and d in Section II.2.2 and Fig. II.30e in the next section.

Emergence is generally considered a beneficial phenomenon, being one of the sources of increased expressiveness (see Section II.2.2) and efficiency of analogical (hence also diagrammatic) representations, see also Section II.3.1.3. This effect provides some ready to use conclusions (or additional facts stored in the representation) at little or no cost. Instead of having to be *deduced* from the represented facts, they can be merely *observed* directly in the representation. Emergence effect in certain definite situations can be defined formally, e.g. as it was done in the context of diagrammatic reasoning by [Shimojima 1996] (who called it "free ride" there).

However, emergence can lead to errors as well, and should be therefore treated with proper caution. Generally, we can discern two basic kinds of errors connected with emergence: *false emergence*, where the emergent facts are wrong, and *unreliable emergence*, where some facts may look like they emerged from the diagram while they have not.

II.4.2.1 False emergence

The false emergence phenomenon consists of an appearance of emergent facts which are, however, either not intended to emerge or outright false. They are sometimes called "false implicatures" [Mackinlay & Genesereth 1985]. The most common cause of false emergence is the use of a wrong visual language, violating to a certain extent the analogicity of the representation. It was already shown in Example II.6a (see Fig. II.6 in Section II.2.2). However, even with a proper visual language one may construct a diagram producing false emergent facts.

Example II.6b (False emergence) The same two simple facts and the (proper) visual language as used in Example II.6a (repeated here in Fig. II.30a and b) may nevertheless produce a false fact as shown in Fig. II.30c. Note that this emergent fact is false in the world, but not necessarily false with respect to the statement of facts in Fig. II.30a, unless one assumes the NBO rule for the statement, see Section II.3.2.5. Drawing the diagram properly, as it was shown in Fig. II.6f, corrects the problem, producing in that case another emergent fact, fortunately true. Alternatively, if we do not want that, the diagram can be drawn as a composite of two subdiagrams, see Fig. II.30d, producing no emergent facts. However, that has its disadvantages, as the resulting diagram is more complex (requiring repetition of the same object twice), and we are deprived of the potential cost savings provided by emergence effects, due to reduced analogicity of the representation. ■

One may insist here that the fact **not** NEXT-TO(France,Poland) is also (emergently) stated in Fig. II.30d, as certainly the rectangles labelled "France" and "Poland" are not adjacent. However, it is wrong to interpret the diagram in this way, as the two rectangles are elements of two distinct subdiagrams and it is in general not meaningful to consider relations between objects not belonging to the same diagram.

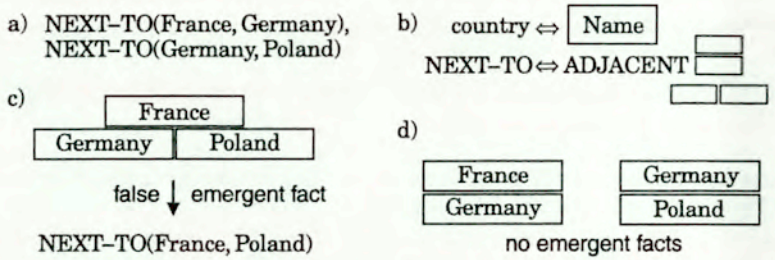


Figure II.30: Two simple facts (a) represented with a proper visual language (b) may nevertheless result in a diagram encoding a false emergent fact (c) or a diagram without emergence (d).

II.4.2.2 Unreliable emergence

This effect is akin to the previous one: again, emergent facts appear that are incorrect. However, here the blame for the error should be laid not so much on the diagram or its creator, but rather on the observer, who carelessly pulls out as emergent facts the visual relations that look like being true, but are not reliable due to the unavoidable imprecision of the drawing and/or the eye, see Section II.3.2.1.

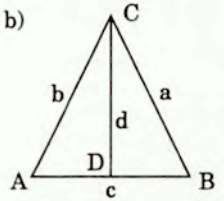
The apparent collinearity of four vertices along the diagonal in Fig. II.13 (see also Fig. II.14 in Example II.11b) is also an example of such unreliable emergence. Another examples are provided below.

Example II.20 (Reliable and unreliable emergence) In Fig. II.31, adapted from [15], first a set of geometrical facts is stated with predicate formulae. Then a diagram compatible with these facts is drawn, in two versions: one with an elementary visual language, and the other using a richer language. From the diagram one can read a number of emergent facts, not stated explicitly in the original specification in Fig. II.31a. These emergent facts can be generally of two kinds: topological (structural), Section II.4.1.2; and metric (geometric), Section II.4.1.1. The former ones, Fig. II.31d, are usually reliable, provided the diagram is drawn properly (cf. Example II.23), while the latter, Fig. II.31e, are often unreliable, as based on imprecise metric judgment, depending on the quality of drawing and precision of reading them off the diagram. If the latter facts are taken to be true without additional reliable checking or confirmation by reasoning, an error may easily occur, like with the "64=65" puzzle, see Example II.11a and Example II.11b. ■

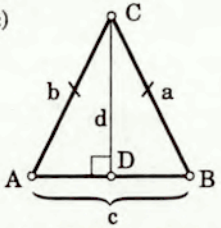
The classification of emergent facts in the above example requires some clarification. First, note the two equalities of angles in Fig. II.31d. Despite the fact that equality of angles is a metric feature, they are listed as structural, hence reliable, facts. This is because the equality here comes not from metric comparison of values of the angles, but from essentially structural property—the equaled angles are *the same* angles, differing only in their symbolic names due to a different choice of the point on one of the angle arms. This is indicated in Fig. II.31d by the use of the symbol "$\angle$" instead of the symbol "$\measuredangle$" used in Fig. II.31e to indicate the numerical *value* of the angle. Distinction between topological and metric features does not depend only on the type of the feature, but also on the manner of reading the feature off the diagram, see also Section II.3.2.1.

a) POINT(A), POINT(B), POINT(C), TRIANGLE(A, B, C),
SEG(a, B, C), SEG(b, C, A), SEG(c, A, B), EQU(length(a), length(b)),
POINT(D), ONSEG(D, c), SEG(d, C, D), PERPEND(c, d)

b)



c)



d) Structural (topological, qualitative):
reliable (usually)
$\angle ACD$, $\angle BCD$, $\angle CDA$, $\angle CDB$,
$\angle CAB$, $\angle CAD$, $\angle CBA$, $\angle CBD$,
$\angle ACB$, AD, BD, $\triangle ACD$, $\triangle BCD$,
$\angle CAB = \angle CAD$, $\angle CBA = \angle CBD$.

e) Metric (quantitative):
usually unreliable or imprecise
$\measuredangle ACD = \measuredangle BCD$, $\measuredangle CAD = \measuredangle CBD$,
$\overline{AD} = \overline{DB}$, $\overline{AB} \neq \overline{AC}$, $\overline{AB} \neq \overline{BC}$,
$\measuredangle CDA = 90°$, $\measuredangle CDB = 90°$,
$\overline{AC} = \overline{BC}$.

Figure II.31: A set of geometrical facts (a); the corresponding diagrams using elementary (b) and richer (c) visual languages, and two kinds of emergent facts (d, e) readable off the diagrams.

Second, note in Fig. II.31e the facts stating that two angles are right (90 degree) angles. They are listed under unreliable facts, because in the diagram of Fig. II.31b nothing of structural nature does indicate that they are right angles. This fact can be only inferred from the original specification which asserts that segments c and d are perpendicular, but neither this perpendicularity nor the rightness of these angles is structurally obvious from the diagram alone (but see the *apparent look* rule in Section II.3.2.1). The situation is different in the diagram in Fig. II.31c which uses a richer visual language. The language contains a construct allowing for explicit stating the perpendicularity of segments c and d, in a manner that simultaneously indicates that the appropriate angles are right angles in a structurally obvious way, making the feature a reliable one, at least concerning reliability of its reading off the diagram.

This indicates that the appropriate choice of visual language may cause the interpretation of the diagram, including emergent facts, more reliable and less prone to reading errors, by avoiding metric imprecision of reading of these facts off the diagram, see Section II.3.2.1. Note some other features of this sort provided by the visual language used in Fig. II.31c, like explicit indicators of equality of sides a and b, and disambiguation of the meaning of the label "c," affirming that it is the side AB, not a point on this side. Note also that interpretation of negated statements like inequality of sides in Fig. II.31e can be further affected by the *negation by omission* rule, see Section II.3.2.5.

II.4.3 Divergence

But what we ought to aim at is ... the recognition
of likenesses hidden under apparent divergences.
[R.M. Pirsig, *Zen and the Art of Motorcycle Maintenance* (1974)]

The divergence effect, in the context of mathematical reasoning, means simply that in some circumstances the reasoning must be conducted by cases, as the structure of the problem (or the visual language used) does not allow to represent it on a single diagram covering all possibilities. The effect has been also formalized in [Shimojima 1996], under the name of *"overdetermined alternatives."* Here a shorter term *divergence* is proposed, as it better describes the nature of the phenomenon and constitutes a neat pair with the term *emergence*. For our purposes it can be defined as follows:

Definition II.9 (Divergence) Divergence *in knowledge representation and reasoning occurs when representing the given data (especially, addition of some new piece of knowledge or premise to the already represented data) cannot be done under the given representation system without splitting the representation into separate, structurally distinct cases or reasoning routes.*

It is interesting that Shimojima considered the effect harmful for diagrammatic reasoning. He repeats this opinion twice in his paper [Shimojima 1996], despite the fact that he explicitly mentions there the Hyperproof hybrid reasoning system with its "Exhaustive Cases" inference operator, see e.g. [Barwise & Etchemendy 1996a, b] which implements exactly the divergent reasoning by cases. Also Miller's geometry system CSEG implements the algorithm generating all potentially possible cases, see [Miller 2000]. It is also worth to mention here that reasoning by cases is a well-known standard proof technique independent of the use of diagrams. The negative attitude toward divergence is probably due to the fact that quite often diagrammatic reasoning is considered to be restricted to operating with a single diagram only, while divergence forces us to create several diagrams and to conduct separate branches of reasoning with them. Another manifestation of this attitude is provided by the frequently raised objection claiming that diagrams cannot represent disjunctive knowledge (see Section II.3.2).

Example II.21a (Divergence in Euler circles) Like in Example II.19a, in Fig. II.32 we see again an application of Euler circles to solve an apparent, simple-looking syllogism. After representing graphically the first premise, and then adding the second, we can read off the emergent conclusion as in Fig. II.32a. It looks innocent enough, especially after we substitute some real-life objects for the abstract A's and B's, e.g. "cats" for A, "quadrupeds" for B and "birds" for C, resulting in both premises obviously true and so the conclusion: "No quadrupeds are birds," see Fig. II.32b. However, a single example does not prove the rule—what with substituting "reptiles" instead of "birds" for C, Fig. II.32c? Both premises are still true, but the conclusion "No quadrupeds are reptiles" becomes obviously false. What went wrong?

During addition of the second premise to the diagram we should notice that the premise establishes some relation between A and C but does not tell anything about the relation between C and B. Hence, while maintaining the required relation between A and C, we are free to place C in relation to B in three significantly distinct ways. These ways are shown

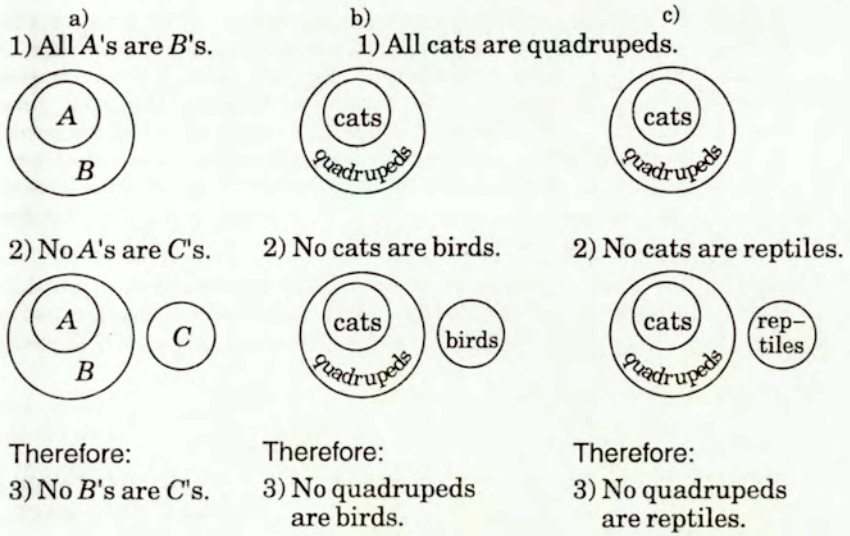


Figure II.32: Divergence in Euler circles: yet another syllogism? (a); it looks quite true (b), but what in this case? (c).

in Fig. II.33: thus, to add properly the second premise we should *diverge* the diagram into three cases. Reading the resulting relation between B and C off these diagrams we obtain the conclusion shown—admittedly not very useful, but true, in contrast to the one obtained in Fig. II.32. ■

Note that the divergent cases can be represented in the diagram either by separate (sub)diagrams for the different cases (the bottom row in Fig. II.33), or using a single

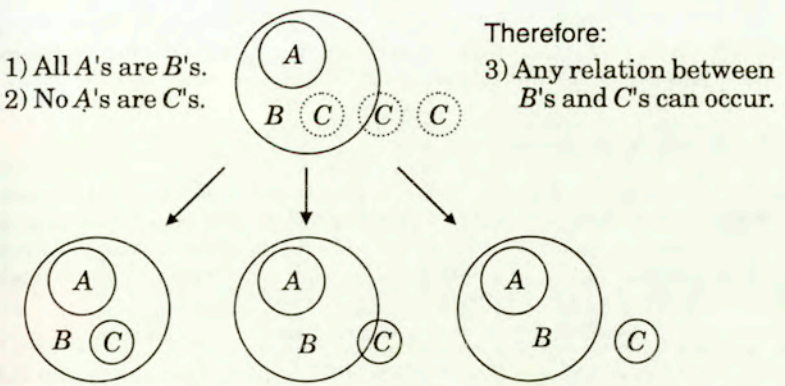


Figure II.33: Divergence in Euler circles: three divergent cases showing that no useful conclusion can be drawn in this case.

diagram with enriched visual language (dotted circles at the top of Fig. II.33). See Section II.3.2.2 for more discussion of representations for such disjunctive knowledge. An occurrence of divergence, similarly as for emergence, depends thus on the visual language chosen. First, if we used another semantic convention for interpreting the circles, namely that assumed in [Hammer 1996], instead of the nonemptiness requirement for overlapping circles assumed here, only a single diagram—the middle one of the bottom row in Fig. II.33—would suffice to represent properly the premises. Also, the same reasoning as above, but represented with quite another visual language may produce no divergence effect, as the following example shows.

Example II.21b (No divergence in Venn diagrams) The same reasoning introduced in Example II.21a can be represented with the visual language of Venn diagrams (see e.g. [Hammer 1996]), as in Fig. II.34. Here the circles always intersect each other in all possible

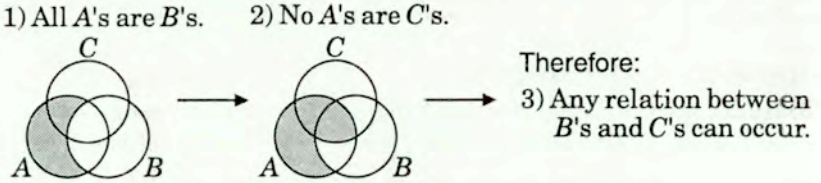


Figure II.34: The reasoning from Figs. II.32 and II.33 represented with Venn diagrams with no divergence effect.

ways, and shading is used to denote emptiness of the appropriate (sub)sets. Thus, the first premise is represented by shading that part of the circle A which lies outside B, indicating that there are no members of the set A that do not belong also to the set B, as required. Similarly, the second premise is added to the diagram by shading the intersection of circles A and C, indicating that there are no elements belonging simultaneously to both sets A and C. No divergence occurs, and from the final diagram it follows easily that all relations between B's and C's are possible, as indicated. ■

As Example II.21a used above to illustrate the effect of divergence shows, divergence may sometimes lead to errors in diagrammatic reasoning. One can distinguish two general kinds of such errors: *overlooked divergence* and *false divergence*.

II.4.3.1 Overlooked divergence

Example II.21a discussed above illustrates the error caused by overlooking that some new fact that we would like to add to the diagram cannot be represented directly in it. Its encoding requires taking into account several different cases (or a change of the visual language). Another example, of a different nature, involves a diagrammatic derivation of a simple algebraic inequality.

Example II.22 (Overlooked divergence) The diagram in Fig. II.35a has been taken from the collection of "Proofs Without Words" [Nelsen 1993],[19] see also Section II.5.3.1.

[19]Attributed there to Nelsen himself. Interestingly, the problem with correctness of the derivation based on that diagram has been overlooked there.

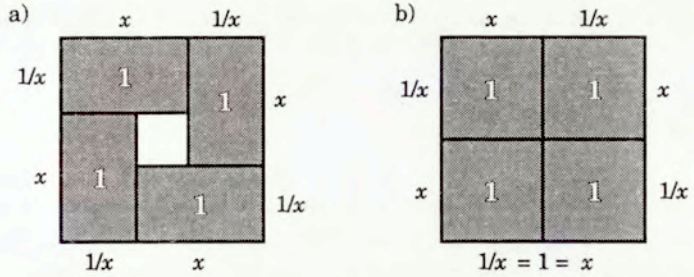


Figure II.35: The diagrammatic proof of a simple inequality: the original Nelsen diagram (a), and the overlooked divergent case (b).

First, obviously $x > 0$, and the area of every gray rectangle equals 1, as indicated. The side of the square equals $x + 1/x$, and because its area is obviously larger than the area of the sum of the four rectangles, we have $(x + 1/x)^2 > 4$, hence $|x + 1/x| > |2|$ and finally $x + 1/x > 2$. Obvious, but consider $x = 1$ (fulfilling the condition $x > 0$ stated at the beginning). Substituting it to our inequality we get $2 > 2$, which is obviously wrong. What happened? We simply overlooked a divergence situation. The diagram of Fig. II.35a is valid for all $x > 0$, except for the case $x = 1$.[20] In the omitted case the diagram looks differently, see Fig. II.35b. From this last diagram we obtain easily the equality $x + 1/x = 2$ (valid for $x = 1$) which, combined with the previous inequality, produces the final formula $x + 1/x \geq 2$, now truly valid for all $x > 0$.

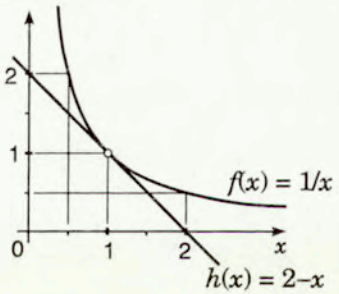


Figure II.36: Another diagrammatic proof of the inequality without divergence effect.

Also here the problem can be made to vanish by a different choice of the visual language. Another diagrammatic proof of the same inequality (also taken, with modifications, from [Nelsen 1993]) is shown in Fig. II.36. Obviously, $f(x) \geq h(x)$ in the whole range $x > 0$ (as the diagram shows, equality occurs only for $x = 1$, as expected from the previous proof), hence $1/x \geq 2 - x$ and finally $x + 1/x \geq 2$. This time, no divergence occurs. ■

[20] Strictly speaking, the diagram only covers the case $x > 1/x$, but this is harmless, as for $x < 1/x$ the diagram remains the same and reasoning proceeds in the same way, only the labels x and $1/x$ should be interchanged. See Section II.3.1.5 for discussion of significance of such *symmetry arguments* in diagrammatic reasoning.

Some cases of overlooked divergence can be also interpreted as a kind of *accidental alignments* discussed in Section II.3.2.4, Example II.14.

II.4.3.2 False divergence

... when you have eliminated the impossible,
whatever remains, however improbable, must be the truth.
[Arthur Conan Doyle, *The Sign of Four* (1890)]

Here the error comes from taking into account apparent divergent cases that are actually impossible. This is usually due to the limited analogicity of the representation, allowing for representation of inconsistent configurations, see Section II.3.1.3, or due to the limited accuracy of drawing (see Section II.3.2.1), allowing for drawing an impossible configuration that is similar enough to some otherwise possible one, so that one is deceived, through triggering of the apparent look rule, to consider it a valid case after all, see [2]. Also in formalized systems which generate all potentially possible cases algorithmically the task of filtering out the impossible cases may not be too easy, see [Miller 2000]. A classic example of an erroneous proof that all triangles are isosceles may serve as a demonstration of that effect. This is a significantly expanded and more precise version of the example mentioned in [Jeleński 1968].

Example II.23 (All triangles are isosceles) Let us take an arbitrary triangle $\triangle ABC$ and draw the following two lines: the line perpendicular to the side AB going through its centre X, and another line bisecting the angle $\angle ACB$, see Fig. II.37.

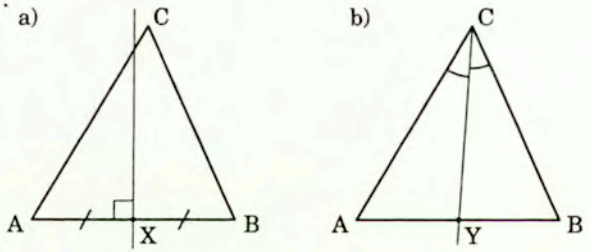
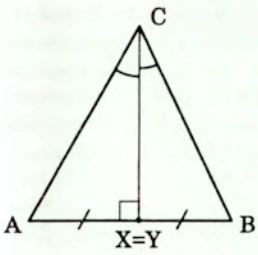


Figure II.37: An arbitrary triangle $\triangle ABC$ with the line perpendicular to AB at its centre X (a), and the line bisecting the angle $\angle ACB$ (b).

Apparently, trying to combine the diagrams into a single one, we see that the two lines drawn can either:

Case 1: Coincide.
Case 2: Have no points in common.
Case 3: Intersect at a single point O lying inside the triangle $\triangle ABC$.
Case 4: Intersect at a single point O lying outside the triangle $\triangle ABC$.

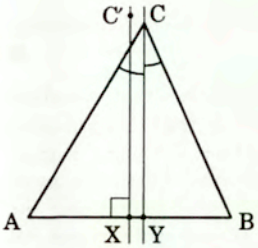
Let us consider these cases in turn.



Case 1. When the line bisecting the angle $\angle ACB$ coincides with the centreline of AB, then

$\triangle ACX = \triangle BCX$ by the SAS rule:
$AX = XB$; $\angle AXC = \angle BXC = 90°$; $CX = CX$.

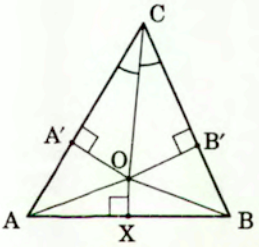
Therefore, $AC = BC$ and $\triangle ABC$ is isosceles.



Case 2. If the line bisecting the angle $\angle ACB$ and the centreline of AB have no common points, they must be parallel, i.e. $XC' \parallel CY$. As $XC' \perp AB$, then also $CY \perp AB$. Thus,

$\triangle AYC = \triangle BYC$ by the SAA rule:
$CY = CY$; $\angle AYC = \angle BYC = 90°$; $\angle ACY = \angle BCY$.

Therefore, $AC = BC$ and $\triangle ABC$ is isosceles.



Case 3. Here the line bisecting the angle $\angle ACB$ and the centreline of AB intersect at a single point O *inside* the triangle $\triangle ABC$. Let us draw additional line segments AO and BO, as well as segments OA' and OB' perpendicular to the sides AC and BC, respectively. Now:

$\triangle A'OC = \triangle B'OC$ by the SAA rule:
$OC = OC$; $\angle OA'C = \angle OB'C = 90°$; $\angle A'CO = \angle B'CO$.

Hence (a) $A'C = B'C$, and (b) $A'O = B'O$. Next:

$\triangle AOX = \triangle BOX$ by the SAS rule:
$XO = XO$; $\angle AXO = \angle BXO = 90°$; $AX = XB$.

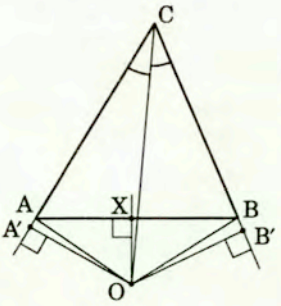
Thus (c) $AO = BO$, and

$\triangle AOA' = \triangle BOB'$ by the SSA rule (for right triangles):
$AO = BO$ (c); $A'O = B'O$ (b); $\angle AA'O = \angle BB'O = 90°$.

Therefore, (d) $AA' = BB'$, and adding (a) and (d) by sides, we get:

$$AA' + A'C = BB' + B'C$$

so that $AC = BC$ and $\triangle ABC$ is isosceles again.



Case 4. Here the line bisecting the angle $\angle ACB$ and the centreline of AB intersect at a single point O *outside* the triangle $\triangle ABC$. Let us now draw additional line segments AO and BO, as well as segments OA′ and OB′ perpendicular to the extensions of sides AC and BC, respectively. Now, in exactly the same way as in **Case 3**, we obtain:

$$A'C = B'C \text{ and } AA' = BB',$$

and subtracting these equalities by sides, we get:

$$A'C - AA' = B'C - BB'$$

so that AC = BC and $\triangle ABC$ is isosceles in this last case as well.

Because for all possible cases we equally obtained that our *arbitrary* triangle is isosceles, it follows that all triangles are isosceles. ■

A more careful geometrical analysis reveals that only **Case 1** above is actually possible, displaying the only configuration of the additional lines defined in Fig. II.37 that can occur in an isosceles triangle. For **Case 2** it is easy to show that the lines C′X and CY must coincide, effectively turning this case into **Case 1**. But **Case 3** cannot occur—the point O always falls outside the triangle $\triangle ABC$. **Case 4** is correct in this respect, but the placement of points A′ and B′ is wrong: they cannot fall on the same side of the base AB of the triangle. Always one of these points falls below and the other above the base, as shown in Fig. II.38a. In that proper configuration it is impossible to prove that AC = BC, while it is easy to show the opposite, namely that $AC \neq BC$.

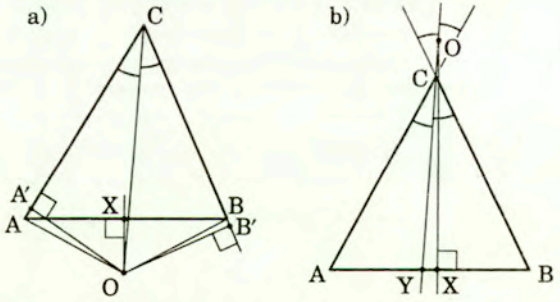


Figure II.38: The proper configuration for **Case 4** (a), and another omitted (impossible) configuration (b).

Thus, we have here a clear case of false divergence, caused by the lack of careful checking whether the configurations that seem to be correct are in fact possible. Therefore, it is at the bottom due to the inevitable imprecision of the drawing followed by uncritical reading off it those (geo)metric features whose visual correctness would require an unattainable infinite precision, see Section II.3.2.1.[21]

[21] Of course, presentation of such examples in books like [Jeleński 1968] deliberately introduces the errors to confound and entertain the readers. However, the history of errors made occasionally by professional mathematicians, not to mention students and schoolboys, shows that the problem is real.

It may be interesting to note that this example shows also an effect which can be called "overlooked false divergence," namely, omitting of additional impossible cases. If one is committed (consciously or mistakenly) to introduce impossible cases, it is reasonable to expect that *all* such cases will be listed. In the example, **Case 2** above is usually omitted—possibly because it is too easy to spot here that it is impossible. Another impossible case shown in Fig. II.38b (a subcase of **Case 4** this time) is omitted too—now probably because here it is impossible to prove that AC = BC (at least with the method used in **Case 3** and **Case 4**). For this reason the latter case has been omitted here, while both cases were omitted from the presentation of the example in [Jeleński 1968].

II.5 Diagrams in mathematics

[The diagram] is dispensable as a proof-theoretic device;
indeed, ... it has no proper place in the proof as such.
[Neil Tennant, *The withering away of formal semantics?* (1986)]

... many properties of mathematical systems
can be unified *and simplified* by a presentation with diagrams ...
[Saunders Mac Lane, *Categories for a Working Mathematician* (1971)]

For the purpose of this work, of most interest is the use of diagrammatic methods as working tools in the mathematical field. Because mathematics now serves as a foundation of many other branches of science and technology, that question is very important to those other branches as well. In many disciplines of science diagrams are used without much reservation, albeit usually also without much awareness of their potential and proper methodology of usage. Also in mathematics the use of diagrams has a long and respectable history. The most naturally diagrammatic field is obviously geometry, so it is not surprising that *"The Elements"* by Euclid, the first mathematics text on geometry foreshadowing the modern axiomatic approach to mathematics, relied heavily on geometric diagrams. Euclid even started a study of the diagrammatic tool as such, attempting, in his treatise entitled *"Pseudaria"* ("Fallacies"; unfortunately, not surviving to our times), an analysis of fallacies in geometric reasoning, see Section II.5.2. Diagrams were later applied not only to geometry. A simple diagram of a number axis played an instrumental role in eventual acceptance of negative numbers as a respectable mathematical entity, for a long time considered to be of doubtful status, as their then common name *numeri ficti* (fictitious numbers) signified. A similar story repeated itself with the invention of complex plane diagram, which not only stimulated the acceptance of, nomen omen, "imaginary numbers," but played a significant role in the development of complex analysis, see [Needham 1997].[22]

However, with the invention of predicate calculus by Frege and the birth of Hilbert's program of formalization of mathematics at the end of the XIX century, diagrams went out of fashion as a respectable research tool in mathematics and their use in this role is actively discouraged till now, as confirmed by the first motto above. The trend went so far that some mathematicians (like Dieudonnè, a member of the Bourbaki team) wrote books on geometry without a single diagram in them and were proud of that. An anecdote about Dieudonnè says, however, that during a lecture when somebody asked him to prove a theorem, he went to the side of the blackboard, drew a diagram, which he hunched over to prevent anyone else from seeing, clicked the chalk over it, and then erased it. Finally, he went to the middle of the board and wrote out a proof without any reference to the diagram.[23]

A notable (although neglected for years) exception were the works of Peirce on diagrammatic notation for predicate calculus, revived quite recently by Sowa as a knowledge representation tool [Sowa 1984]. The fashion persisted despite the fact that many promi-

[22] In this role, the complex plane diagram apparently has not yet said its last word, as shown by the invention of diagrammatic representations for some operations on complex numbers, until recently thought to be not representable in this way, see [Needham 1997] for details.

[23] As related on the Internet discussion list on diagrams <diagrams@csli.stanford.edu>.

nent mathematicians admitted the use of visual images and diagrammatic aids in their mathematical research [Hadamard 1945]. Another interesting example is the case of Edgar Kaucher, a prominent founder of interval algebra whose work has given a name to one of its important branches, called "Kaucher arithmetic," see Section III.4.4. The (unpublished) Kaucher dissertation [Kaucher 1973] was illustrated with a number of diagrams, some of them using essentially the same concept of interval space as that used in this work (see Chapter III). However, further published works by Kaucher contained no diagrams at all. Also, his idea for a diagram for the space of so-called "Kahan intervals" [Kaucher 1999], see Section III.4.5, was never published by him, though he claims that it was instrumental in devising the calculation method described in [Kaucher 1977].

That may be partially due to the fact that formalization of propositional reasoning, using one-dimensional strings of symbols, is much simpler. Formalization of diagrammatic representation and reasoning remains still in its beginnings, partly due to the complexity of the task, partly because work in that direction started only quite recently, after long years of relegating diagrams to low grade classrooms, as a mere teaching aid for uninitiated.

The situation changes recently, mostly due to accumulating results in the area of formalizing various mathematical applications of diagrammatic reasoning, see e.g. [DIAGRAMS 1996, Hammer 1996, Jamnik et al. 1999, Winterstein et al. 2000]. A publication of an award-winning book [Needham 1997], reviving the diagrammatic exposition of complex analysis is also symptomatic. As these and other examples indicate, diagrams *are* used by mathematicians, although mostly only in private, because an open admittance of the use of diagrams is still not in vogue in many mathematical circles (see again [Hadamard 1945]). Publications without diagrams are considered so much more "professional"... Despite that, in some branches of modern mathematics the use of diagrams begins to be openly admitted, e.g. in category theory, see the second motto above. See also their use in discrete group theory [Coxeter & Moser 1957/80].

It would be instructive here to consider in detail the main arguments against the use of diagrammatic methods in mathematics. In the order of growing importance, they are:

Difficulty argument: Diagrammatic reasoning is much more difficult than sentential reasoning to specify, interpret, check for correctness, and in other uses.

Unreliability argument: Diagrammatic reasoning is inexact and notorious as a source of errors and hidden assumptions.

Informality argument: Diagrammatic reasoning is incompatible with the basic philosophy and nature of formal reasoning.

Let us examine these arguments in more detail.

II.5.1 Are diagrams difficult?

You can be perfectly easy in your mind.
We shall certainly find some way out of your difficulties.
[Arthur Conan Doyle, *The adventure of the three students* (1905)]

It is interesting that even some prominent users and advocates of diagrammatic methods expressed opinions about the difficulty of using them, as the two quotes below signify:

> It is not easy to use the geometrical method to discover things, but the elegance of the demonstrations after the discoveries are made is really very great.
> [Richard Feynman, *The Motion of Planets Around the Sun* (1964)]

> ... while it often takes more imagination and effort to find a picture than to do a calculation, the picture will always reward you by bringing you nearer to the Truth.
> [Tristan Needham, *Visual Complex Analysis* (1997)]

There are several answers to that concern which will be discussed in turn.

II.5.1.1 Individual abilities answer

Different people are differently endowed with various abilities. While for some people it may be quite difficult to think pictorially, for others this can be actually an easier mode of thinking than the propositional one. As psychological findings indicate, in this respect the human population is divided roughly in half—one half being more proficient in verbal, and the other half in visual thinking. However, with the educational bias to be discussed below, visual thinkers have rough times in school and after it, so that their visual abilities are seldom properly trained and used to their full potential. This applies especially to people seeking a career in science. Continually, the phrase *Man of Letters* is used as a synonym for a highly educated intellectual. *Men of Pictures* are thus relegated to "lesser" professions like crafts and arts. Not surprisingly, the proportion of people for whom diagrammatic reasoning is easy seems to be much smaller among scientists than in the overall population.

II.5.1.2 Skill training answer

Like any other skill, also the ability to design and manipulate diagrams is to a large extent a learned skill—if it is exercised, it becomes more proficient and easy. It should not be surprising that people of our times have problems with diagrammatic thinking when we realize that our educational system, since long ago using mostly verbal learning and devoted to training mainly language skills, does not encourage the use of diagrams, and even actively discourages it, especially at earlier stages, just when children express greatest interest and proficiency in visual thinking, see [Piaget 1951].

Symptomatic in this respect might be the case of Jakob Steiner (1796–1863), the famous Swiss mathematician, considered the greatest geometer after Appollonius of Perga. He did not learn reading and writing before the age of 14 and went to school only at the age of 18. Not being forced to unlearn his visual thinking skills in his early age, he exhibited an outstanding geometrical imagination[24] that finally made him so prominent in this area. He hated algebra and analysis, saying that calculation simply replaces thinking, while geometry stimulates it.

In summary, one may claim that once diagrammatic skills are seriously taught in school and used without reservations in all intellectual professions where they can be useful, the finding of an appropriate picture will not take more imagination and effort than writing a coherent paragraph of several tens of words.

[24] Of course, it is true that a single example, however spectacular, does not necessarily prove a rule—but it is hard to stop wondering...

II.5.1.3 Pictorial effector answer

Besides perceiving and imagining diagrams, an effective use of them requires an actual drawing of (often many, and intricate) diagrams (see Chapter III for examples). With the equipment provided by human biology it is indeed difficult to produce diagrams of any complexity and accuracy—humans do not have a proper and effective *visual effector*, comparable in efficiency to that one we use for spoken language. Our eyes flood us with tremendous amount of information every second, orders of magnitude greater than that we are usually able to produce in the form of pictures with our hands, even if these hands belong to a skilled and highly trained draftsman. Now realizing the tremendous effects on human development caused by our mastering of sufficiently complex spoken language, one starts to wonder how the situation might look like if we were endowed by similarly effective organ for also *producing* complex pictures.

Fortunately, in our times we do have technical means very effective in producing complex pictures. We can use them as a sort of prosthetic devices compensating for our deficiencies in this area. The use of computers makes it possible to make diagrammatic methods accessible to all *visual thinkers* there can be. For how it can be actually done, see the section on computer implementation of diagrams (Section II.6).

Difficulty argument: conclusion. To sum up this section, the difficulty objection cannot in fact be meaningfully held against diagrams. It is rather a challenge asking for creation of conditions in which the great unused potential of visual thinking lying mostly dormant within the human population can be finally freed and put to more effective use.

II.5.2 Are diagrams unreliable?

If we threw out every form of reasoning
that could be misapplied by a careless reasoner,
we would have little if anything left.

[Jon Barwise and John Etchemendy, *Heterogeneous logic* (1991)]

As [Arnheim 1969] candidly observes:

> Perception ... is unreliable, as shown by the many optical illusions, and can refer only to actual, physically given objects, which are always imperfect.
>
> [Rudolf Arnheim, *Visual Thinking* (1969)]

It is thus not surprising that representation of knowledge in diagrams, and reasoning conducted with them, is prone to various errors. This, however, applies as well to all other representations and reasoning tools. The differences here boil down to different causes and types of error situations, and different ways to avoid them. Once properly recognized, analyzed in detail, and taught to the users of diagrams, they are no more harmful that in any other human activity. To achieve that goal, appropriate knowledge about possible sources and causes of such errors should be developed, and sufficient level of education and skill training of prospective diagrammatic reasoners should be reached. The work in this direction was started long ago, already by Euclid, who devoted an entire treatise (under the title *"Pseudaria"*) to the subject of erroneous diagrammatic reasoning.

The work unfortunately did not survive to our times; the only (short) relation about its contents was given by Proclus, the author of an extensive commentary on Euclid's work. Unfortunately, due in part to long neglect of these problems, especially when the diagrams were seemingly safely expelled from mathematics at the end of the XIXth century, little has been done in this matter since Euclid. The necessary knowledge is still very inadequate and even that limited information that we do have is not well known to a wider audience, hence the (unwarranted) feeling that diagrams are somehow intrinsically unreliable as tools of rigorous and precise reasoning.

Various possible error situations that can occur in diagrams are discussed in Sections II.3 and II.4, see also Section II.2. A general, although probably not yet full and precise summary of the most important error causes is provided below.

Readability errors. Here the error comes from unreliable reading of diagrams due to their sloppy design or execution, see Section II.2.3, unaccounted for visual illusions (Fig. II.16), or too intricate or graphically inadequate visual language. (Fig. II.7c).

Imprecision errors. These are due to the argument relying on imprecise metric features, like in the "64=65" puzzle, see Example II.11a in Section II.3.2.1. This kind of error becomes often an important cause of several other kinds of errors like *accidental alignments* (Section II.3.2.4), *unreliable emergence* (Section II.4.2.2), or *false divergence* (Section II.4.3.2).

Emergence errors. The emergence effect (Section II.4.2), although generally beneficial, may nevertheless lead to errors as well. Here we have *false emergence* which can be due to improper visual language (Section II.2.2) or diagram design (Section II.4.2.1), and *unreliable emergence* (Section II.4.2.2), usually caused by diagram imprecision.

Divergence errors. The common technique of reasoning by cases (called here the *divergence effect*, Section II.4.3) can be also conducted erroneously. The errors can be, first, of the *overlooked divergence* type (Section II.4.3.1), due to an inadequate case analysis or *particularity* (Section II.3.2.3), or some kind of *accidental features* effect (Example II.14 in Section II.3.2.4). Second, there is also *false divergence* (Section II.4.3.2), due to imprecision or limited analogicity of the representation (Section II.3.1.3).

Particularity errors. These errors come from the necessity of using a particular diagram to stay for a whole class of situations (Section II.3.2.3), and more generally, problems with precise representation of sets of objects or configurations (Section II.3.2.2). The errors of this kind are due to imprecise (possibly due to diagram imprecision, Section II.3.2.1) or erroneous delineation of the required set, and manifest themselves often as *overlooked divergence* errors (Section II.4.3.1).

Some kinds of errors listed above occur also in other representations, not only the diagrammatic ones (like errors due to sloppy execution of the representation or language misunderstanding errors), while some are rather specific to diagrammatic representations (like emergence errors). All these errors can be avoided by careful design and use of the appropriate visual language well adapted to the given task (see Section II.2), augmented by knowledge of possible error situations and methods of avoiding them. Therefore, Hilbert's warning:

> The proof can indeed be given by calling on a suitable figure ... [It merely] makes the interpretation easier, and it is a fruitful means of discovering new propositions. *Nevertheless, care, since it can easily be misleading.*
>
> [David Hilbert, *Lecture on Geometry* (1894)]

should clearly be observed, though *not* to the extreme of banning diagrams altogether from formal mathematical reasoning, as can be ascertained from the following quote from [Hadamard 1945] about the work of Hilbert himself:

> Logically, of course—and this is all that is essential—the result announced is fully attained and every intervention of geometrical sense eliminated: that is, theoretically unnecessary to follow the reasoning from the beginning to the end. Is it the same from the psychological point of view? Certainly not. There is no doubt that Hilbert, in working out his *Principles of Geometry*, has been constantly guided by his geometrical sense. If anybody could doubt that (which no mathematician will), he ought simply to cast one glance at Hilbert's book. Diagrams appear at practically every page. They do not hamper mathematical readers in ascertaining that, logically speaking, no concrete picture is needed.
>
> [Jacques Hadamard, *The Psychology of Invention in the Mathematical Field* (1945)]

The last sentence of the above quote needs additional comment. Possibly, "logically speaking" no concrete picture is needed—but then equally logically speaking no concrete formulae are needed in mathematical texts, as all their content can be presented with sentences in plain natural language, say English. Actually, so it was done through centuries of history of mathematics. Intricate formulae and mathematical notations are a quite recent invention, and as yet nobody proposed to get rid of them as "logically speaking" not needed. So why trying to ban diagrams? See also the next Section II.5.3, where the remaining part of the above Hilbert's quote is discussed.

II.5.2.1 Are formulae reliable?

The reasoning based on sentential representations—e.g. mathematical formulae—is also prone to error and may be equally unreliable, as it is signified by the long history of wrong proofs published by professional mathematicians, and often amusing blunders constantly produced by schoolchildren and college students.

If the mere existence of fallacious inferences would be sufficient to dismiss the reasoning method, the propositional reasoning, including mathematical formulae, would qualify for such a dismissal as well. A wide variety of mistaken proofs and fallacious inferences has been produced without any help of diagrams. These range from the traditional informal fallacies, to the misapplication of formal rules, to mistakes far harder to classify or categorize. Moreover, the sources, and indeed the very existence of such fallacies, was in no way self-evident, but come from careful attention and analysis of this form of reasoning.

Table II.7: A sentential "proof" of the equality $1 = 2$.

$$\begin{aligned} -2 &= -2 \\ 1-3 &= 4-6 \\ 1-3+9/4 &= 4-6+9/4 \\ 1^2-2\cdot 1\cdot(3/2)+(3/2)^2 &= 2^2-2\cdot 2\cdot(3/2)+(3/2)^2 \\ (1-3/2)^2 &= (2-3/2)^2 \\ 1-3/2 &= 2-3/2 \\ 1 &= 2 \end{aligned}$$

As an amusing example, see the "proof" that $1 = 2$, shown in Table II.7.[25] To avoid such blunders one must also observe certain rules and precautions, just like those sketched above for diagrammatic reasoning. Which of these rules (and where) were violated in Table II.7 is left as an exercise to the reader. Finally, consider that the diagrammatic proof of the same equality looks more elegant and entertaining, see Fig. II.39 on the right (and also Section I.5.2).

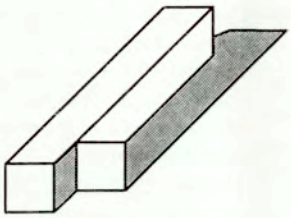


Figure II.39: A diagrammatic "proof" of the equality $1 = 2$.

Unreliability argument: conclusion. To sum up this section, the unreliability objection also cannot be credibly held against diagrams. Diagrams are no more unreliable than formulae, provided the proper caution is observed, and reasoning rules suitable for the specific features of diagrammatic reasoning are used. The only remaining difficulty so far is due to the fact that the proper error-avoidance rules for diagrams are not yet fully investigated, codified and taught. This problem constitutes then another challenge to be confronted by the researchers in the field of diagrammatics.

II.5.3 Are diagrams intrinsically informal?

The more formal we made the visit
the less information we might obtain.
[Arthur Conan Doyle, *The Hound of the Baskervilles* (1902)]

The most serious argument against the use of diagrams as a standard tool in rigorous mathematical work says that by their nature diagrams cannot be made formal enough to be acceptable as valid components of rigorous mathematical proofs. To that argument one can answer shortly that it is already disproved by many fully formalized systems of diagrammatic reasoning that have been developed, see e.g. [Shin 1994, Hammer 1996,

[25] One of many similar examples circulating around. This particular version is adapted, with modifications, from a book on entertaining mathematics [Jeleński 1968].

Luengo 1995, Jamnik et al. 1999, Miller 2001]. However, because analysis of arguments of this type can be quite instructive, this argument will be reviewed here in some detail.

Diagrams were treated with suspicion by many mathematicians since long ago. This concerns especially logicians, who, by the nature of their subject[26] were conditioned to use almost exclusively the linguistic mode of thinking. The most serious blow against mathematical diagrams was struck at the end of the XIXth century by the Hilbert programme of total formalization of mathematics.[27] The following quotes by Pasch and Hilbert himself express the position of that formalistic attitude nicely:

> In fact, if geometry is to be genuinely deductive, then the process of inferring must always be *independent of the sense of geometrical concepts*, just as it must be *independent of diagrams*. ... In the course of a deduction, it is certainly legitimate and useful, though in no way necessary, to think of the reference of the concepts concerned. Indeed, if it is necessary to do so, then the inadequacy of the deduction is revealed, and even the insufficiency of the proof method.
>
> [Moritz Pasch, *Vorlesungen über Neuere Geometrie* (1882)]

> The proof can indeed be given by calling on a suitable figure, but this appeal is not at all necessary. [It merely] makes the interpretation easier, and it is a fruitful means of discovering new propositions. Nevertheless, care, since it can easily be misleading. A theorem is only proved when the proof is *completely independent of the diagram*. The proof must call step by step on the preceding axioms. The *making of figures is the experimentation of the physicist*, and experimental geometry is already over with the axioms.
>
> [David Hilbert, *Lecture on Geometry* (1894)]

The argument in these quotes says, in effect, that the formal mathematical reasoning must be restricted to mechanical manipulation of symbols only, without reference to any "sense" of the underlying mathematical concepts (while diagrams, allegedly, constitute the direct embodiment of just these concepts). Laying aside the question whether mechanical manipulation of symbols is indeed all that there is (and should be) significant in mathematics, let us first ask if diagrams necessarily must contain directly "the sense of geometrical concepts" in order to be useful?

So, does the sensible use of a diagram in a proof hang on it being like the real-world objects on which physicists conduct their experiments? This is certainly doubtful. A line in a diagram is *not* a line understood as a geometric object—it is at most only a (crude and approximate) *representation* of the appropriate geometrical concept. It then has the similar status as any other symbol or notation used in a "normal" proof, i.e., one containing no diagrams. Thus, a diagram may be considered to be merely a different symbolic *notation*, useful to represent geometrical entities in the same way as certain other squiggles on paper are useful as representations of numbers, logical or arithmetic operations, and the like. Moreover, diagrams are essentially also finite arrangements of

[26] Note that "logos" means "word"...

[27] Interestingly, while the formalist attack was directed by logicians against the essential use of diagrams in geometry, resulting in practical expulsion of them from that discipline, at the same time diagrams started to be used in logic itself, due to theh work of Venn and Peirce, see [Gardner 1958].

symbols from a finite alphabet. One cannot draw an infinite line with infinite precision no more than one is able to write in a formula an infinitely precise letter "x" exactly identical in shape to the previous one. To claim that diagrammatic reasoning relies on such allegedly infinite or infinitely precise features is to fall in the kind of error thoroughly discussed in Sections II.3.2.1 and II.4.1.1. Indeed, no formalized system of diagrammatic reasoning can base its inference rules on such features (and none does). Thus, diagrams can also be constructed, transformed and inspected in a finitistic way, just like the formulae.

It is true, however, that *directness* (or *analogicity*) of diagrams, so that they seem for a human reasoner to more vividly represent the underlying mathematical concepts, their important features, and relations between them, does seem to introduce into the reasoning process "the reference of the concepts concerned." But it actually helps in creatively conducting rigorous reasoning (as Pasch and Hilbert themselves admit in the quotes above), provided of course that one observes necessary precautions against misuse of the notation, as one must do also when using formulae (see Section II.5.2.1). The question whether that implies that one again introduces back the seemingly safely expelled ghosts of Platonic forms to which mathematical notations seem to refer, can be, for the time being, left to ponder by philosophers, as it has little if any direct relevance to the everyday mathematical practice.

Diagrams might be here, due to the functional structure of our brains, especially useful and well suited to represent certain *geometric* entities. But they can be also, with equal benefit, used to represent many other abstract entities in various branches of mathematics, quite far from geometry (e.g., complex numbers [Needham 1997]), or in any other discipline, from biology to quantum electrodynamics to geography to relativity theory.

Consider also that no mathematician actually works by a purely mechanical juggling of meaningless symbols. Practically nobody even works with the pure logical notation of the sort introduced in Section II.1.2. That would greatly impede the efficiency of mathematical work, cf. Section II.1.2.2. Instead, mathematicians use, and constantly expand, quite intricate symbolic notation systems, containing also lots of essentially diagrammatic elements like two dimensional fractions, subscript and superscript systems, matrices, morphism diagrams, often very pictorial operator or relation symbols, and the like. The purist may try to accept diagrams by considering them just as another level of (or step to) even more intricate notation, devised for the sole purpose of increasing the efficiency of work of mathematicians (or any other formal thinkers, for that matter).

The importance of having, and actually using, different notational systems to generate different insights and suggesting different directions of possible development of represented concepts has been neatly summed up by Richard Feynman and Stanisław Ulam:

> Theories of the known, which are described by different physical ideas, may be equivalent in all their predictions and hence scientifically indistinguishable. However, they are not psychologically identical when trying to move from that base into the unknown. For different views suggest different kinds of modifications which might be made and hence are not equivalent in the hypotheses one generates from them in one's attempt to understand what is not yet understood.
>
> [Richard Feynman, *Nobel Lecture* (1966)]

> I have to admit that there are cases when formalism by itself has great value—for example, the technique, or rather the notation of Feynman graphs in physics. It is purely a typographical idea, it does not bring in itself any tangible input into a physical picture, nevertheless, by being a good notation it can push thoughts in directions that may prove useful or even novel and decisive. [Stanisław Ulam, *Adventures of a Mathematician* (1976)]

Thus, rather than waste efforts on trying to expel diagrams from the mathematical reasoning practice, we should rather redirect them into investigation of conditions for effective and safe (from the point of view of rigour and correctness) use of diagrams.

II.5.3.1 "Proofs without words"

> Better say nothing at all.
> Language is worth a thousand pounds a word.
> [Lewis Carroll, *Through the Looking Glass* (1871)]

A mathematical journal *Mathematics Magazine* includes a column under the heading "Proofs without words," featuring drawings demonstrating various mathematical relations or theorems. A collection of these drawings was published in a book form too [Nelsen 1993]. One might be tempted to treat it as an indication of an "official" use of diagrammatic tools in mainstream mathematical research, but such an opinion would be rather premature. First, the column contains illustrations of rather elementary and simple mathematical facts, well known since long ago, without any pretence for a scientific novelty—as far as this author knows, the magazine never published a regular paper containing solely such a diagrammatic exposition. The column plays merely the role of an entertaining exercise for the readers, like puzzle columns in newspapers (the more so because, without additional words, it is often hard to understand what the included diagram attempts to demonstrate and how it does that).

The title of the column is rather misleading too. First, those diagrams hardly can be called proofs, as they lack necessary ingredients of a proof, like generality. They usually demonstrate the thesis for some special case only, with little, if any, indication of how it can be generalized to other cases (see e.g. [Jamnik et al. 1999] for discussion on what a complete diagrammatic proof should contain). Second, these "proofs" are seldom completely without words, considering, obviously, mathematical formulae and textual labels as words too. One should also realize that diagrammatic proofs are not distinguished by mere lack of words (or formulae), but by the essential role a diagram must play in the reasoning. Not all proofs containing diagrams can be thus called diagrammatic: some of them can be called at most "diagram-aided," while others are merely only *illustrated* with diagrams, as the reasoning in them is quite independent of the attached diagrams.

All that does not in any way mean that the column and its contents are worthless. They provide many interesting examples, from diverse branches of mathematics, of how various mathematical facts can be represented diagrammatically, providing inspiration and simple test cases for the proper research on diagrammatic reasoning in mathematics, see e.g. [Jamnik et al. 1999, Winterstein et al. 2000]. However, one should not mistake the column as an indication for already widely accepted, "official" status of diagrammatic methods in mathematics.

Informality argument: conclusion. This author's position on the issue can be best described by first starting from the following two quotes:

> It is a basic principle in the study of mathematics, and one too seldom emphasized, that a proof is not really understood until the stage is reached at which one can grasp it as a whole and see it as a single idea. In achieving this end, much more is necessary than merely following the individual steps in the reasoning.
>
> [G.F. Simmons, *Introduction to Topology and Modern Analysis* (1963)]

> Diagrams ... play an essential role in both the comprehension and communication of mathematical proofs. This role is to make the content of the proof "real" rather than formal. ... the validity of a proof can be "seen" in the diagram rather than justified as a step-by-step application of formal rules. ... Visualization, then, is a means by which mathematics sheds its purely formal character and takes on meaning. As such, it is a key aspect not just of mathematical learning but also of mathematical discovery. Diagrams, in turn, are a vehicle for communicating the visualized images. Far from being expendable aid, diagrams play an essential role in the communication of mathematical meaning.
>
> [Dave Barker-Plummer and Sidney C. Bailin, *The role of diagrams in mathematical proofs* (1997)]

To that it should be only added that in order to be able to devise appropriate diagrams for a particular proof or problem, it is much needed and useful to develop first a sufficiently comprehensive *system* of diagrammatic representation of objects, notions and relations of a given mathematical (or other) domain. Such a system, with its sufficiently rich visual language (see Section II.5.4 below), provides ready-made tools and building blocks for producing appropriate, unambiguous, and uniformly interpretable diagrams for any particular problem or proof within the domain. One of historical examples of such a diagrammatic system, of great significance for the development of its respective domain, is the diagram notation for complex numbers plane, see [Needham 1997]. A current example of the development of a similar system, this time for interval algebra and computations, constitutes the main subject of this author's research and is presented in Chapter III of this work.

II.5.4 Visual languages of mathematics

> We want something more practical
> than the vision of equations curling their way through air.
> [Isaac Asimov, *Forward the Foundation* (1993)]

When talking about mathematical diagrams, the question of designing an appropriate system of diagrammatic notation, i.e., a *visual language* (see Section II.2) for the given fragment of mathematics cannot be avoided. As an introduction to the problem, let us survey the main styles of mathematical diagrams, both old ones and new possibilities offered by modern computer technology (see Section II.6, especially II.6.4).

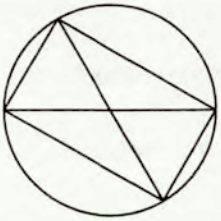
Let us repeat once more that the use of diagrams in mathematics has a long and respectable history. Their mode of use varied with time, and with culture differences. Geometricians of ancient China and India used extensively the method of visual rather than propositional argument [Arnheim 1969, Tufte 1990]. Ancient Greeks, and western mathematics after them, followed another path. Euclid's *"The Elements"* contain many drawings, but they usually play a role of mere illustrations, helping to understand the statement of the problem, whereas the whole argument is presented textually, in a formal, step-by-step sequence of logically connected propositions. This practice laid foundations for development of formal deductive scientific method, but the possible advantages of direct visual grasp were put aside and all but lost to generations of scholars in geometry and other sciences.

Euclid's proof of the Pythagoras theorem is intricate and nonintuitive. The accompanying diagram is complicated and its major role is to help the reader in not becoming completely lost in the tangle of argument. An interesting attempt to "pictorialize" Euclid was made by [Byrne 1847] in the middle of the XIXth century (see [Tufte 1990] for excerpts of some interesting fragments). Instead of textual labels marking points, lines, and figures in the drawings, Byrne used colour to code the elements and areas in the drawing, and inserted corresponding coloured graphical symbols in appropriate places in the text. However, the structure and style of Euclid's exposition remained unchanged. Thus, what he made was simply to replace, as stated in the title of his book, the usual letter labels used to refer to geometrical objects discussed in the text by appropriate elements of the diagram. This led in just the opposite direction than that prescribed by one of the postulated advantages of diagrammatic representations, see e.g. [Larkin & Simon 1987] and Section II.3.1.4, namely the possibility of getting rid of superfluous labels. Instead of removing textual labels, Byrne replaced them with graphical ones. Not surprisingly, the attempt seems hardly successful. The intricate proofs remained intricate, the train of underlying reasoning obscure, and the references between the text and the diagram even more hard to follow.

That Byrne's failed proposal certainly shows the crucial importance of taking properly into account visual language issues when attempting to construct a diagrammatic notation for any application, especially in mathematics. Let us thus consider the main styles of mathematical diagrams and what they offer the user. The survey will use an example from geometry, undoubtedly the most diagrammatic of all branches of mathematics, consisting of a diagram-aided proof of a simple geometry theorem suggested by an example from [Arnheim 1969].

II.5.4.1 A simple style

The simplest style illustrated in Fig. II.40 uses single drawings consisting of lines with undifferentiated thickness and no other graphical elements. More popular in old times, due to costs of engraving of more complicated drawings, it is, surprisingly, quite often encountered in mathematical texts nowadays, usually with detrimental effects on the diagram usefulness. This diagram style requires usually detailed textual explanations, like that provided in the figure, and can be used only for simple drawings. This makes such diagrams suitable only as a mere supplementary illustration to the text, with no important contribution to the discussion.



In order to prove that the triangle based on the diameter of the circle is always right-angled, draw a line from the vertex of the triangle through the centre of the circle and arrive thereby at a rectangle, located symmetrically within the circle. By its position in this rectangle, the angle at the vertex of the triangle is an angle of 90°.

Figure II.40: A simple mathematical diagram with necessary textual explanation.

To illustrate more complicated mathematical arguments, comprising several reasoning steps, sometimes an elaboration of such diagrams containing several subdiagrams illustrating the individual phases of the argument are used, like in Fig. II.41. It does not make much of a change as concerns the other features discussed above.

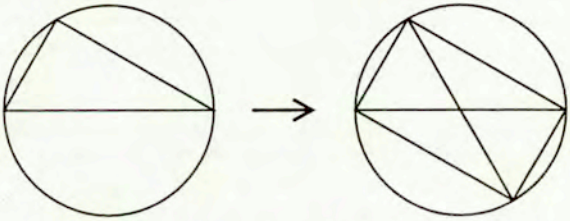
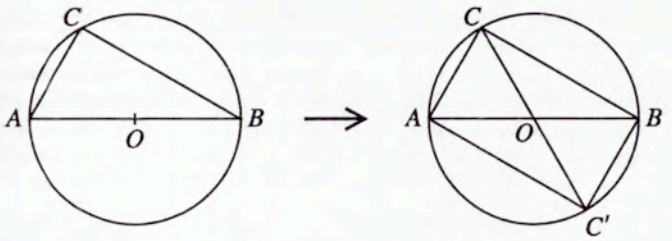


Figure II.41: A simple mathematical diagram with *premises* and *conclusions* subdiagrams.

II.5.4.2 A standard textbook style

The most widely used style of contemporary mathematical diagrams is presented in Fig. II.42 (here, in the version using subdiagrams for subsequent reasoning steps). Its simplified version was used in [Arnheim 1969]. In such diagrams, the still undifferentiated linear drawing is augmented with textual labels marking usually important points on the drawing, and sometimes also other elements (line segments, regions). Such diagrams still require textual explanations, but the use of labels in the text allows for simpler and much more convenient linking between the text and elements of the diagram. This facilitates more complex and sophisticated use of such diagrams, up to and including a possibility to produce true diagrammatic proofs in which the diagram plays a crucial role in the argument. For example, note that most of the "proofs without words" in [Nelsen 1993] (see Section II.5.3.1) also use this style. As these proofs then must contain (textual) labels, they hardly can be claimed to be entirely "without words."

Let us recall here the attempt to replace textual labels by graphical ones, proposed long ago by [Byrne 1847] and discussed above at the beginning of Section II.5.4. The result was hardly an improvement, indicating that diagrammatic proofs should rather take advantage of the possibility, offered by diagrammatic representations (see Section II.3.1.4), of entirely



In order to prove that the triangle $\triangle ABC$ based on the diameter AB of the circle is always right-angled (i.e., $\angle ACB = 90°$), draw a line from the vertex C of the triangle through the centre O of the circle to the point C′ on the circle, and arrive thereby at a rectangle ACBC′, located symmetrically within the circle. By its position in this rectangle, the angle $\angle ACB$ at the vertex C of the triangle is an angle of 90°.

Figure II.42: A "textbook style" diagram with subdiagrams and textual explanation.

getting rid of (most) labels instead of making them graphical. The next Section shows how it can be done in practice.

II.5.4.3 A pure diagrammatic style

Here most of the burden carried by textual explanations necessary in the more traditional styles can be taken over by an enriched visual language, see Fig. II.43. In this way, with the appropriately elaborated visual language, one may ultimately get rid of textual labels and explanations, making the representation (and proofs) wholly "without words." The problem is that any sufficiently rich visual language ceases to be self-explanatory, requiring prior learning by the user, or often elaborate explanations accompanying its usage. This is worsened by the fact that, despite the claims that our culture already became mostly visual, the "visual illiteracy" [Dondis 1975] of the prospective users of mathematical (and other) diagrams still remains significant. Also, there are no sufficiently rich, well established and codified graphical notational conventions ready for use in such cases. Hence, even perfectly willing and graphically competent makers of such diagrams

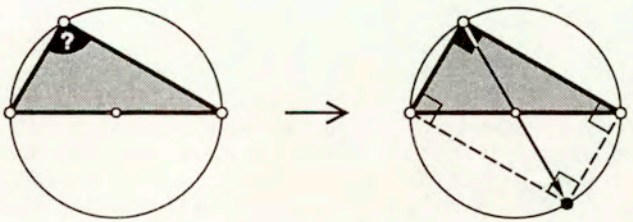


Figure II.43: A "purely diagrammatic" diagram (with subdiagrams).

must often improvise and design new conventions and symbols on the spot, with the significant risk of being not understood or misunderstood by the users. Thus, one of the most important tasks facing researchers on diagrammatics is to propose, design, codify and disseminate appropriate graphical conventions and visual languages for representing diagrammatically complex knowledge in various disciplines, especially those ones in which diagrams were not routinely used for some time (or even shunned off, like in mathematics).

II.5.4.4 A hybrid diagrammatic style

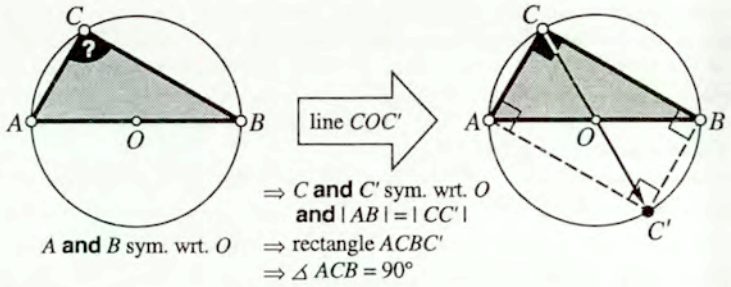


Figure II.44: A "hybrid diagrammatic" diagram (with subdiagrams).

In this style, see Fig. II.44, one adds textual labels and explanations to the diagram drawn with the richer visual language as described in the previous Section. These textual elements serve partially as explanations of the visual language constructs, partially repeat the information represented diagrammatically in terms more understandable to textual-oriented users, and partially provide the information hard to represent diagrammatically (at least within the limits imposed by the visual language used). Due to the information provided by the richer visual language, textual explanations can be here made simpler and more symbolic which facilitates their incorporation into the diagram itself, as in Fig. II.44. Until fully diagrammatic representations in the given field are developed and fully assimilated by users, such a hybrid representation seems the best way to advance a more diagrammatic practice in the field, at least within the realm of still the most abundant static diagrammatic representations. This kind of diagrammatic style is also used in most of the Chapter III for interval algebra diagrams.

II.5.4.5 Dynamic styles

In many proposals for more extensive use of diagrams, a notion of *dynamic diagrams* is often advanced [Arnheim 1969, Barwise & Etchemendy 1996a]. Different authors usually mean different things by that term. The two most important meanings are:

Sequence of steps. A dynamic diagram of this sort is an ordered sequence of diagrams showing consecutive steps of some complex diagram construction, or of many-step reasoning process. It can be a discrete sequence of "snapshots" (like film frames) or a continuous (at least virtually) animation of the diagram.

Structure variation. In this case, what is being depicted is not a linearly ordered set of steps of a construction or a reasoning process, but rather different variations of an essentially the same diagram structure, not necessarily linearly ordered, though clearly constituting a more or less continuous transformations of each other.

The *step sequence* type is the most common. Two-step sequences of this sort were used in the examples considered above. There are several technical means of displaying such sequences, namely the use of *step indicators*, *static diagram sequences*, or *true animation*, to be discussed in the sequel.

Step indicators. A diagram in this style usually consists of a single diagram, but the sequence of steps in constructing the diagram and/or conducting the reasoning is indicated by appropriate ordering labels (most often consecutive numbers, as in Fig. II.45). This

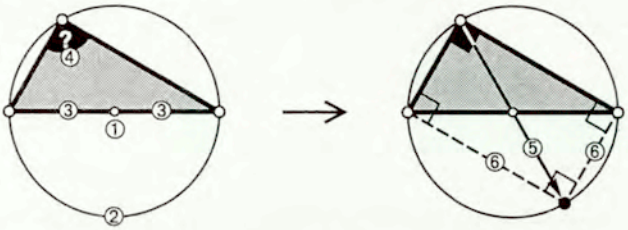


Figure II.45: A diagram with step indicators.

method can provide the possibility to grasp the whole idea of the reasoning at a single glance, at the same time allowing for easy analysis of individual steps when necessary. Note the use of this style in some interval arithmetic diagrams in Section III.4. Note also that there are more than one of same-numbered indicators for some steps. They are used in cases when two (or more) elements of the diagram can be drawn simultaneously (i.e., the order of their construction is irrelevant).

This trick works well for simple diagrams, when the step indicators do not unduly clutter the diagram and are easy to find and order in their proper sequence. Otherwise, they better should be avoided. For example, in Fig. II.45 one should add, for completeness, a final ⑦ indicator, repeated four times at the straight-angle indicators in the vertices of the rectangle. It was omitted, however, as these indicators would clutter the diagram too much making it partially illegible.

Static diagram sequences. An obvious construction for many-framed "static" animation is the use of a many-step sequence, as shown in Fig. II.46. Such a design can show unambiguously a sequence of construction or reasoning steps even in the case of complex diagrams. The main drawback is the amount of space it may require. Moreover, in many cases, even with many small steps, the perception of a dynamical change can be quite poor.

True animation. The standard, real-time film-like animation is often proposed as a best method of making truly dynamic diagrams. Practice does not seem to confirm that

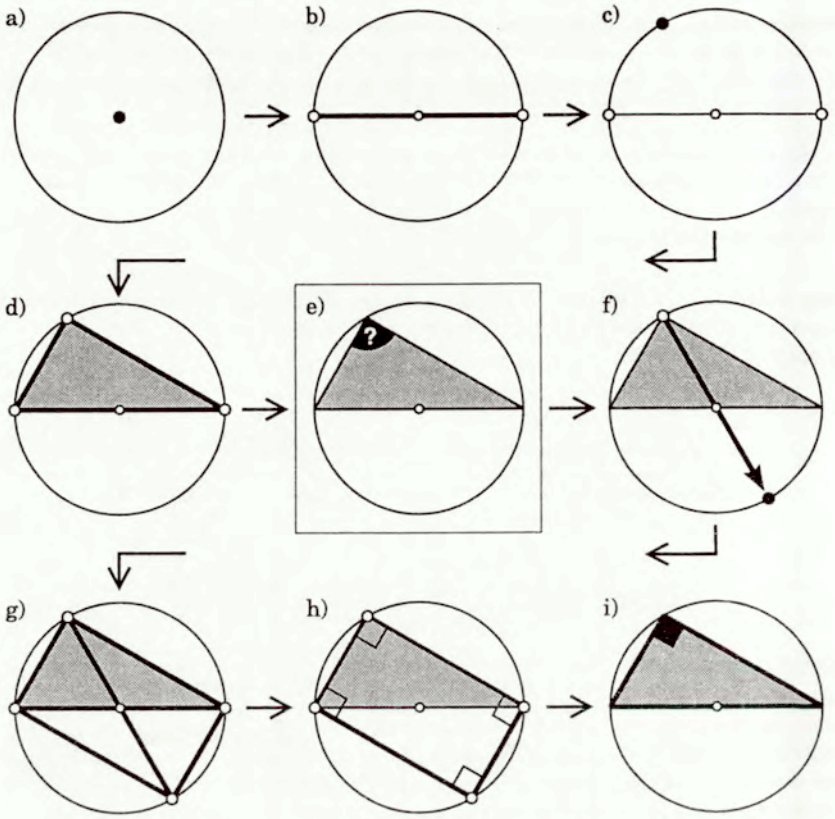


Figure II.46: A diagram sequence (animation frames): construction (a, b, c, d) of the problem statement (e), and reasoning steps with auxiliary constructions (f, g, h) leading to the conclusion (i).

view. The usefulness of such animated diagrams is limited, mostly due to passive role they assign to the user, imposing on him/her a fixed sequence and rate of information flow, and actually hampering a unified view of the whole argument at a glance. An only partial remedy for that may be provided by giving to the user a minimal video-like control over the animation: stopping, playing/rewinding back and forth, and changing the playing speed, all that without loss of quality of the presentation.

Yet another drawback of this approach is that preparation of such animated diagrams is still rather costly and difficult for an average user. There are no widely accessible software tools to make animated diagrams, with the ease of use comparable to popular drawing editors. And it is not likely to change soon, as the usefulness of such an animation rarely seems to be worth the effort, as indicated above, especially for the day-to-day use by a larger number of people.

However, as a technique to produce a general educational film material, diagram animation has been often used, also in mathematics, like in the classical film titled "*The Hypercube: Projection and Slicing*" [Banchoff & Strauss 1978]. More recent examples, including also more complex techniques, albeit of rather experimental significance, can be found e.g. in [VISMATH 1997].

Structure variation diagrams. These diagrams are in most cases produced as static single diagrams with textual explanations (or special visual indicators in the diagram, see below) describing required structural transformations, see e.g. the proof of the "fast multiplication" Proposition III.3 in Section III.4.2. The ultimate form of such diagrams, combining them, moreover, with the step-sequence diagrams, is provided by the *interactive animation* diagrams concept introduced below (see also Section II.6.4).

The structure variation diagrams help to solve the problem of handling universal quantifiers in diagrammatic reasoning, see e.g. [Winterstein et al. 2002], and the closely related particularity problem, see Section II.3.2.3. Instead of representing directly the set X over which the quantification takes place, the diagram represents only some particular instance x_0 together with a rule to produce other configurations from X in a way assuring that the reasoning concerning x_0 applies equally to all other elements of X, see Section II.3.2.3 for more details.

Example II.24a (Structure variation diagram) A simple example has been provided by [Arnheim 1969], see Fig. II.47. A particular triangle is drawn, with easy diagrammatic

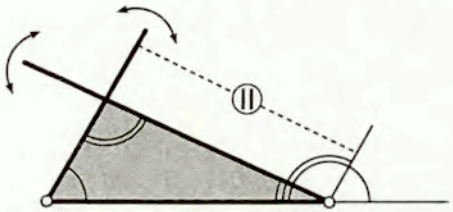


Figure II.47: A diagrammatic proof for the sum of angles in a triangle, using a structure variation diagram (after [Arnheim 1969], extended).

proof that all its internal angles sum up to 180°. At the same time, graphical indicators at the ends of extended sides of the triangle (together with the parallelism indicator) supply the necessary structure variation argument showing that whatever the directions of the sides may be (that is, effectively for all triangles) the relationship proven for this particular triangle will remain unchanged. ■

Interactive animation diagrams. Animated diagrams may acquire full functionality when the animation can be made interactive in the sense of full control by the user over the dynamic changes of the diagram or its parts. That is, when the very construction of the (partial) animation of the diagram is interactively available to the user, in a click-and-drag mode, and the description what elements can move, and in what way, can be also easily and interactively defined by the user. Consider the possibility that in the diagram of

Fig. II.47 the user can grasp one of the ends of the extended triangle sides and rotate it as indicated, with the rest of the diagram reshaping itself automatically to maintain all relevant constraints (e.g., the rotation points position, indicated line parallelism, extents of angle marks), showing that for all positions the sum of the angles remains the same.

Many examples of such diagrams were implemented for a computer in some special cases.[28] However, these implementations usually are capable of drawing only a single specific diagram, allowing the user to manipulate its elements in a preprogrammed way, but not defining easily his/her own diagrams and constraints. See Section II.6.4 for more details on an implementation of such an interactive diagram definition and animation system.

[28]See for example some diagrams on the page `http://www.cut-the-knot.com/geometry.html`

II.6 Computer implementation of diagrams

> The patriarch gave a nod, ... wrote up the program, threw the switch, lifted the lid of the Black Box and said: "Behold!"
>
> [Stanisław Lem, *The Cyberiad* (1974)]

While, on the first hand, the development of computers and their widespread use helps to spread the use of diagrammatic methods and making them more practical, on the other hand practical computer implementation of diagrammatic representation and reasoning methods encounters a lot of difficulties.

Computer implementation of diagrams can, in general, serve two main purposes:

Diagrammatic aid for humans: here the implementation serves as a sort of prosthetic device, compensating for the lack in humans of the proper and effective *visual effector*, comparable in efficiency to that we use for the spoken language.

Diagrammatic component of AI systems: here the aim is to endow the computer with yet another human capability used by humans in their intelligent behaviour.

The problem of computer implementation of diagrams divides generally into three main issues, see also Section I.1.1, which will be discussed in turn:

Input of diagrams to the computer.

Representation and processing (reasoning with) diagrams inside the computer.

Output of diagrams from the computer.

Depending on the purpose, different aspects of the implementation become important. When the direct human aid is aimed at, the most important issues concern a rich and functional input and output of diagrams, in other words, a graphical man-computer interface, while the internal processing of diagrams can be rather rudimentary. In diagrammatic AI systems, on the other hand, internal processing of diagrams, or in other words their use as internal inference tools, becomes the most important component, while input and output can be much simpler, or even nonexistent in the case when input or output data are of a non-diagrammatic nature.

At the end of this section, an idea for a computerized diagrammatic reasoning tool—called a *diagrammatic spreadsheet* and serving as an effective visual effector for human users—is briefly introduced.

II.6.1 Diagram input

Input of diagrams to a computer can be generally made in two distinct ways, paralleling to some extent the raster versus vector graphics division discussed in Section I.2.

Direct scanning, by a device like a TV-camera or electromechanical scanner, of the diagram already available on some external medium, like a drawing on paper, photograph, or a real world scene.

Human-aided drawing, involving on-line construction of the diagram from scratch with the help of some sort of graphical editor running on the computer.

The first approach requires further processing by the computer of the raster input to extract and recognize meaningful structures of the diagram, a task still hard for computers and performed by them in a way much inferior to human abilities in this area, see Section I.1.3. This method is thus still rarely used in practical man-machine diagrammatic systems. One of the rare exceptions is the process of transforming huge paper archives of engineering drawings (gathered in the course of many years by automobile, aerospace, or other machinery construction companies) into electronic form needed for modern computerized design practice. The problem led to construction of often complex hardware and software systems dedicated especially to that task. Even a dedicated *raster-to-vector conversion* subdiscipline of computer image processing with its own scientific conferences [GREC 1999] emerged as a result. It includes also a similar problem involving digitization of data stored in geographical maps, especially land register maps, see e.g. [Stąpor 2000] and [GREC 1999] as well. These systems require a significant involvement of human operators for spotting and correcting numerous errors made by the computer processing and recognition routines.

Thus, in more common systems of diagram input, the diagram is drawn by a human user, interactively and directly in electronic form, with the help of an appropriate *graphic editor* run on the computer. Common graphic editors allow to compose the diagram from geometric primitives like line segments, polylines, spline curves, simple figures (circles, ellipses, rectangles), and additional drawing operators like area fill with colour or transformations like scaling, rotation, etc. Properties of every graphical element so constructed are mostly independent of properties of other elements, so that constructing structures with specifically related elements, like drawing a line tangent to a circle at a required point, is often not an easy task and cannot be done very accurately with available tools. Also, many intended structural relations in the diagram, like the above-mentioned tangency of the line and the circle, cannot be explicitly asserted by the user and made a part of the diagram specification, which significantly impedes the usefulness of such diagrams for further internal processing, like diagrammatic reasoning. The problem is addressed by a new generation of graphic editors called *feature-based* or *constraint-based* editors [Nelson 1985, Gleicher & Witkin 1994], where new elements can be added by directly specifying their relations to the existing ones, see Section II.6.4 below for an example. With such editors, the constructed diagram contains already the meaningful structure required for internal processing or diagrammatic reasoning by the computer.

II.6.2 Internal diagram representation

... see if you can think of two devices,
so that if one fails, the other will carry on.
[Isaac Asimov, *Prelude to Foundation* (1988)]

The internal representation of a diagram should facilitate effective execution of the operations needed to perform the required form of processing within the system. In parallel to the common division of computer graphics formats into raster and vector graphics, see Section I.2, internal representations of diagrams can be:

Diagrams on a raster, where a diagram is represented generally as a two-dimensional array of homogeneous simple elements, and the processing is done with systems of *local picture operations* of the sort discussed in Section I.3. Thus, the processing here takes place at the *encoded picture* level of the model in Fig. I.3 (Section I.1.2).

Diagrams as graphs, where the meaningful structure of the diagram is represented as some kind of graph structure with vertices representing diagram elements and their individual properties, and edges representing relations between them. Processing of so encoded diagrams is based on various kinds of *graph grammars* or more general *graph transformations*. It thus in general takes place at the middle, *image description* level of the model in Fig. I.3 (Section I.1.2).

II.6.2.1 Diagrams on a raster

Diagrams are here represented as a two-dimensional array of homogeneous simple elements (called *pixels*), i.e., as some abstract digital image as defined in Section I.3. The processing in this case is based on systems of local image operations (sequential or parallel) of the sort discussed in Section I.3 as well. If considered as a raster image of the represented diagram, this kind of representation requires costly processing to extract the needed diagram structure and update the diagram with new facts and reasoning results, due to complicated raster image processing needed, see also Section II.6.1. Thus, it is rarely used in this manner (see [Funt 1980, Olivier et al. 1996]). A raster diagram can be more effectively used to represent objects or phenomena which at the outset have an internal raster structure with local homogeneous interaction pattern, see [Furnas 1990, Furnas et al. 2000, Gardin & Meltzer 1989] and Example II.25 below.

A great advantage of diagrams represented as raster images is their direct representation of spatial relations, which is a primary feature a diagrammatic representation should possess, see Section II.3.1.2.

Despite the fact that diagrams in that format are practical in a rather limited kind of applications, quite a number of computer diagrammatic systems based on this principle were implemented. Not surprisingly, most of them were aimed at testing some cognitive aspects or theories of mental imagery and diagrammatic reasoning rather than for a more practical use.

Most probably, the first system of this kind was devised by [Funt 1980] as an element of an artificial intelligence system called WHISPER for simulation of a simple physical system consisting of unstably stacked blocks. One of the main aims of the system was the simulation of retinal processing of pictures, hence the raster used was circular and non-homogeneous. Concerning the diagram representation, it was a hybrid system, using diagrams in both raster and graph-like formats.

A more recent KAP (*Kinematic Analysis Program*) system is similar in several respects to the WHISPER system. It uses diagrams on a raster (this time the standard rectangular one) for simulation of interaction of kinematic pairs of mechanical parts [Olivier et al. 1996, Olivier 1997]. The main novelty here is the use of a pyramid of raster representations at different resolutions, and of an attention window on the image, controlled by the high-level reasoner to follow the regions of interaction of the parts. This approach can be contrasted with that of the CLOCK system [Forbus et al. 1991], where the similar

kinematic reasoning is conducted in a configuration space represented as a graph (see the next section).

Another group of raster-diagram models used them as simplified models of some simple physical phenomena, see e.g. [Furnas 1990, Gardin & Meltzer 1989]. The model was a raster image representing the (simplified) state of the physical system modelled, while the "laws of physics" were stated as local pixel neighbourhood rewriting rules of the sort described by Definition I.12 in Section I.3.7. The rules were applied in parallel to all places on the image where they were applicable. In addition to simulating simple physical systems (especially in [Gardin & Meltzer 1989], though there the "substrate" was not the rigid two-dimensional raster of [Furnas 1990]), the model was able to perform also some necessary calculations with digits represented as subimages on the raster (the possibility not so surprising in the view of the computational power of the operations discussed in Section I.3.2).

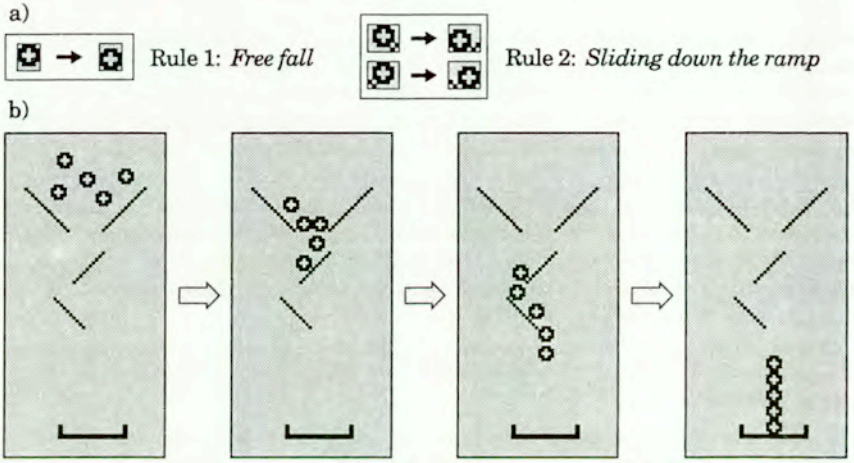


Figure II.48: Raster diagram simulating fall of balls (after [Furnas 1990]): local rewriting rules (a), and several snapshots from the simulation (b).

Example II.25 (Falling balls simulation) A model for simple simulation of falling and sliding balls using the paradigm of parallel local pixel rewriting rules is shown in Fig. II.48. The rewriting rules model free fall and sliding along slanted ramps. The simplification of physics in the model is apparent in the resulting neat stacking of the balls at the final simulation frame. With a finer raster and a more complex set of rules, more physical phenomena could be taken into account with more realistic modelling results. ■

Note that essentially this very paradigm was used in early computer games to simulate artificial environments, changing, as a result of player's actions in the course of the game, according to some homogeneous system of local "physical laws," like the laws of free fall and ramp sliding stated with the rewrite rules in Fig. II.48.

Another way of using raster images for diagrammatic reasoning has been studied within the so-called *inter-diagrammatic reasoning* of [Anderson & McCartney 1997]. Here the

diagram has the form of a tiled array (essentially a raster image, but with more general meaning of its elements—they do not need to have a pictorial nature), and the reasoning takes the form of sequences of point operations (Section I.3.2) on n-tuples of such images taken as a whole. Examples given by [Anderson & McCartney 1997] include solving the n-Queens problem on a chessboard and learning guitar chord fingering rules.

II.6.2.2 Diagrams as graphs

In this representation, the meaningful structure of the diagram is represented as some kind of an interlinked list structure, with list elements (records) representing diagram elements, their individual properties, and relations between them, while list links are used for connecting related elements. The abstract model of such a structure is best described by a *graph*, with vertices representing diagram elements and edges representing relations, though many systems representing diagrams in such a way may not mention explicitly graphs as their representation tool. The processing of so encoded diagrams is based on various kinds of *graph grammars* or more general *graph transformations*. To capture the structural richness of many diagrammatic representations, various extensions of the graph formalism are used, e.g., by adding *hyperedges* to represent multidimensional relations between nodes (*hypergraphs*, [Berge 1973]), or *hypernodes* to represent elements with multiple *attachment points* (*plex structures* [Feder 1971], *CP-graphs* [Grabska 1993a]), or adding hierarchical constructions (*higraphs* by [Harel 1988]).

Such a graph (or structural) representation of diagrams has the advantage of easier inspection and manipulation of the diagram, as all meaningful components of the diagram are explicitly available. It can be also naturally generalized into many-dimensional diagrams or a general *model based* reasoning (which is actually a logic-theoretic name for general analogical representations). The main disadvantage of this internal representation is less direct representation of spatial relations. That means less analogicity and thus more room for "impossible cases," i.e., errors due to an internal inconsistency, see Sections II.3.1.2 and II.3.1.3.

Practically all implementations of formal diagrammatic reasoning systems use this structural representation, starting from the first system of [Gelernter 1959], through GROVER of [Barker-Plummer & Bailin 1997], DIAMOND of [Jamnik et al. 1999] and others. It is interesting to compare here the CLOCK system of [Forbus et al. 1991] for qualitative analysis of interaction of kinematic pairs of mechanical parts with the KAP system of [Olivier et al. 1996, Olivier 1997] described before. While the KAP system analyzes the interaction of the parts on a raster image, the CLOCK system constructs from the drawing of the parts a graph describing qualitatively the configuration space of the parts and uses this graph to reason about possible interactions of them.

A number of formal models for such structural representation of pictures or diagrams was developed. Many of them were first proposed within the framework of picture processing and recognition [Fu 1982]. A survey of most important early proposals of that kind can be found in [109]. One of them, called a *plex language* and proposed by [Feder 1971], had the form of a graph built from a sort of hypernodes, called *NAPEs*, with explicitly named attachment points through which the node may be connected to other nodes. A contemporary extension of that idea, called *CP-graphs* (Composition graphs), was investigated in [Grabska 1993a, Grabska 1993b]. Another formalism proposed by

[Harel 1988], using so-called *higraphs*, is based on hypergraps (which may contain hyperedges that link more than two nodes, see e.g. [Berge 1973]), with nodes developed from Euler circles (see Section II.4.2) augmented by the possibility to represent Cartesian products of sets and hierarchical structures. An elaborate graph formalism, called *E-R diagrams* (Entity-Relationship diagrams) is used for conceptual specification of structures of databases since late seventies [Chen 1976], while in essence very similar *semantic nets* were commonly used for general knowledge representation [Brachman 1979]. These formalisms can be used for both external diagrammatic representation of certain domains (like state-transition diagrams for complex concurrent systems described in [Harel 1988], or of design specification investigated by [Grabska 1993b]), and as internal implementation tools for various diagrammatic representations (as the CP-graph formalism was used, see [Borkowski et al. 1999]).

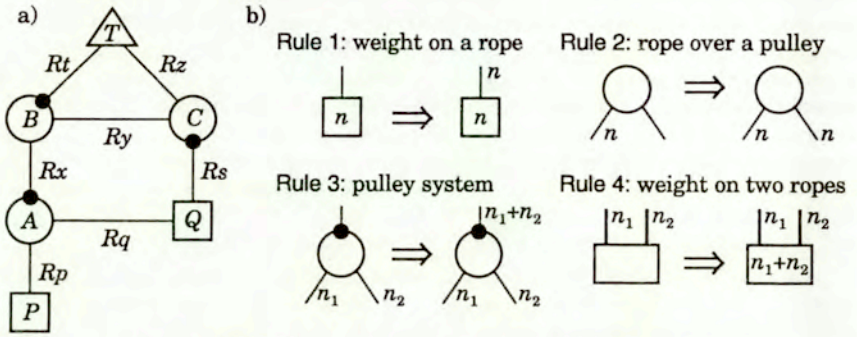


Figure II.49: The pulley example represented by a graph: the graph structure (a), and graph transformation rules (b).

Example II.3i (Graph formulation) Our example pulley system can be also easily represented as a graph structure, as shown in Fig. II.49. Note that the representation uses hypernodes to represent pulleys, with two kinds of attachment points differentiated graphically in Fig. II.49a. The reasoning rules represented by **if... then** formulae in Tables II.1 and II.2 are represented here by simple graph rewriting rules in Fig. II.49b (they rewrite only edge and node labels, leaving the structure of the graph intact). The diagram in Fig. II.49a is an external diagrammatic representation of the internal graph describing the pulley system represented in customary diagrammatic form in Fig. II.9 (see Example II.3g in Section II.3). ■

II.6.3 Diagram output

... the whole performance on the computer screen
is for the benefit of human eyes ...
[Richard Dawkins, *The Blind Watchmaker* (1986)]

Output of two-dimensional diagrams from a computer is, with the present technology, easily done in generally two ways:

On screen, for immediate display of the diagrammatic result for a human user during man-machine interaction.

On hardcopy, as printouts on paper or film, for further reference or external archiving.

In actual intensive development, and sometimes already in use in specialized applications, are further enhancements to this output options, like *animation, three-dimensional* display and printing, and an interactive combination of the above, called *virtual reality.* The main directions of development concern enhanced realism (for a human), and richer, more direct interactivity between humans and machines during cooperative problem solving. The latter direction leads also to experimenting with new man-machine interaction paradigms specific to diagrammatic applications, like the one discussed in the next section.

II.6.4 Diagrammatic spreadsheet concept

In many applications of computer systems, especially the more complex and creative ones like aiding various fields of science or of engineering design, the most successful systems are hybrid, man-machine systems combining different and complementary abilities of both parties, see Section I.1.3. Of the same kind is the interactive diagram animation concept called here a *diagrammatic spreadsheet,* see [3]. It is generally based on the old *ThingLab* idea by [Borning 1981]. The diagram constructed with the system is in essence a structure variation diagram, as described in Section II.5.4.5. Now it can be interactively animated by the user, in a click-and-drag manner, according to underlying constraint system prescribing how a change of some diagram element or feature causes appropriate changes of other diagram elements. Graphical elements of the diagram behave thus like cells in a spreadsheet, with formulae associated with the cells defined as sets of geometric (and other) constraints binding the features of this element with those of the others. The constraint solving system automatically recomputes the features dependent on those that have changed due to the action of the user, and displays the new, transformed diagram. What differentiates the proposed system from existing examples of such animated diagrams mentioned in Section II.5.4.5 is the possibility of easy interactive specification by the user of the structure of diagrams used and all the underlying constraints.

The implementation of the system will consist of three main modules:

User interface, consisting of the interactive diagram and constraint editor.

Diagram representation, using hypergraphs and a graph transformation paradigm, see Section II.6.2.2, describing both the diagram and constraint system structure and displaying the diagram in various specified ways.

Constraint solver, controlling diagram transformations according to the specified constraint system and user-induced changes.

The interactive diagram editor allows for constructing the diagram from graphical elements described in Section II.2.1 and setting their properties in various interactive ways. Construction of the diagram is aided by a possibility of setting properties of elements *in relation to properties of other elements,* like a point lying on a given line, a line tangent to the specified circle or parallel to another line, etc. In many ways it resembles (and

partially extends) the familiar compass-and-ruler paradigm of hand drafting of geometric and technical diagrams. Relations defined during construction become automatically included as elements of the set of constraints defining the internal structure of the diagram, to be maintained intact despite diagram transformations during interactive animation. The constraints can be also edited and changed independently of diagram construction, although essentially with the help of the same interactive mechanisms as described above. As it was mentioned in Section II.6.1, several constraint-based graphical editors already implement this paradigm to some extent [Nelson 1985, Gleicher & Witkin 1994].

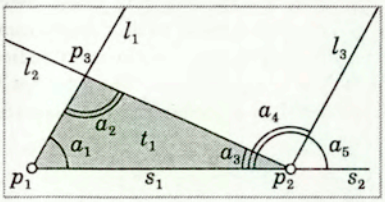
Internal diagram representation is based on the *CP-graph* concept, see [Grabska 1993a]. In this representation, the deep structure of the diagram, together with structural constraints binding properties of its elements, is represented by a graph structure, see Section II.6.2.2, independently of the definition of an external rendering of the displayed diagram, which is defined by a so-called *realization scheme*. This allows for separate transformations of a diagram structure without bothering about external display details, and also for independent change of displayed appearance without affecting internal diagram structure.

Constraint solver controls diagram transformations according to the specified constraint system, recomputing dependent properties of objects after user-induced changes due to interactive animation of the diagram. The constraint system works also during editing of the diagram, helping to specify new objects in terms of the already constructed ones (e.g., a half-line starting at a given point, or an endpoint of an existing line segment), as it was mentioned above. It may also detect emergent structural changes due to diagram transformation and report them as possible inferences concerning new properties of the defined diagram structure.

Example II.24b (Angles in a triangle) To illustrate the above, a possible external rendering of a significant part of such internal diagram description is shown in Table II.8. It represents the structure variation diagram drawn in Fig. II.47 of Example II.24a. It consists of a list of records representing geometrical objects (here listed in order of one of possible sequences of diagram construction), with fields representing their relevant structural properties and constraints. The textual labels of the records are provided here for easy reference; internally they are implemented simply as links in a list (or rather graph) structure. The accompanying figure shows correspondence between elements of the diagram and the labelled objects in the description.

For brevity and convenience of the reader, constraints are placed at one of the fields they bind, with other fields listed as arguments. In practice, they are undirected and are represented as separate nodes of the graph structure. Constraints are listed as predicates after the "|" separator; for simplicity, the equality and inequality predicates assume that their first argument is the field (or the whole object) they bind. The repetition of the intersection calculation in the fields of the point p_3 is not necessary and was added for clarity. The FIXED predicate means that the marked field is not changeable during interactive animation. Note that because the coordinates of the point p_1 are not FIXED in this way, a possible modification of the diagram involves also dragging this point to another position, albeit only horizontally, due to the FIXED orientation of the segment s_1 (of which this point serves as an endpoint), and to the left of the point p_2, due to the "$< p_2.x$" constraint.

Table II.8: An internal description of an example structure variation diagram.



p_1: POINT | $= s_1.start$
x: x_1 | $< p_2.x$
y: y_1
display: circle(...)...

s_1: LSEG
start: p_1
end: p_2
orient: 0° | FIXED

p_2: POINT | $= s_1.end$
x: x_2 | FIXED
y: y_2 | FIXED
display: circle(...)...

l_1: HALFLINE
start: p_1
orient: ... | $\geq 0°; \leq 180°$

l_2: HALFLINE
start: p_2
orient: ... | $\geq 0°; \leq 180°;$
| $> l_1.orient$

p_3: POINT | $= \text{intersect}(l_1, l_2)$
x: ... | $= \text{intersect}(l_1, l_2).x$
y: ... | $= \text{intersect}(l_1, l_2).y$
display: none

s_2: LSEG | $\text{COLLIN}(s_1, s_2)$
start: p_2
end: $p_2 + (20, 0)$ mm
orient: 0° | $= s_1.orient$

l_3: HALFLINE | $\text{PARALLEL}(l_1, l_3)$
start: p_2
orient: ... | $= l_1.orient$

a_1: ANGLE
x: ... | $= p_1.x$
y: ... | $= p_1.y$
start: 0° | $= s_1.orient$
end: ... | $= l_1.orient$
display: circarc(...)...

a_2: ANGLE
x: ... | $= p_3.x$
y: ... | $= p_3.y$
start: ... | $= l_1.orient + 180°$
end: ... | $= l_2.orient + 180°$
display: circarc(...)...

a_3: ANGLE
x: ... | $= p_2.x$
y: ... | $= p_2.y$
start: ... | $= l_2.orient$
end: 180° | $= s_1.orient + 180°$
display: circarc(...)...

a_4: ANGLE
x: ... | $= p_2.x$
y: ... | $= p_2.y$
start: ... | $= l_3.orient$
end: ... | $= l_2.orient$
display: circarc(...)...

a_5: ANGLE
x: ... | $= p_2.x$
y: ... | $= p_2.y$
start: 0° | $= s_2.orient$
end: ... | $= l_2.orient$
display: circarc(...)...

t_1: TRIANGLE
vertices: $\{p_1, p_2, p_3\}$
display: areafill(...)...

For brevity, the representation in Table II.8 does not contain the rotation handles and the explicit parallelism indicator shown in Fig. II.47. They can be easily defined and added to the representation. Alternatively, they can be made a part of the realisation scheme (prescribing the external rendering of structure elements, see [Grabska 1993a] and [3]) instead to be a part of the structural description of the diagram. Other external rendering details (defined in the realisation scheme) are also in most part omitted from the description in Table II.8, except for some objects, where they are only generally indicated using the "display" field). Also, the rendering of half-lines in the figure assumes that they are drawn "to infinity," i.e., practically till the window frame containing the diagram, so that they will not end in midair as in Fig. II.47. ■

Chapter III

Diagrammatic interval algebra

lucid interval *n*: a temporary period of rationality
between periods of insanity or delirium.
[*Merriam-Webster's Medical Dictionary* (1997)]

The birth of the new discpline of *calculation with intervals*, including, and sometimes identified with, the fields of *interval arithmetic*, *interval algebra*, and *interval analysis*, etc., can be attributed to early papers by Warmus [Warmus 1956] and Sunaga [Sunaga 1958] (see also [Warmus 1961] and [Markov & Okumura 1999]). Numerical intervals were mentioned earlier in various contexts, mostly as convenient notation for argument and function ranges, and in error and tolerance analysis. But in these papers for the first time the formal rules for arithmetic on intervals, as separate mathematical objects on their own, were formulated, and their use for approximate solution of equations and inequalities was outlined.

However, the authors of these papers did not pursue the interval research further (with only a small exception provided by a short paper by Warmus several years later [Warmus 1961]), hence the main credit for laying firm mathematical foundations for the new discipline went to Ramon E. Moore with his seminal classic published ten years later [Moore, R.E. 1966]. With that book, the research on interval analysis started in earnest. After some time, another comprehensive textbook [Alefeld & Herzberger 1983] appeared, followed by more specialized books on various aspects and application areas of intervals: calculation of ranges of functions [Ratschek & Rokne 1984], systems of interval equations [Neumaier 1990], interval global optimization [Hansen 1992], and algorithmic complexity results [Kreinovich et al. 1997]. A recent book [Jaulin et al. 2001] addresses application issues, systematizing the interval approach to computation and providing all basic algorithms and practical application examples.

At the same time, many software packages for interval computations were developed, in Fortran, C, and various other programming languages, most of them free. Recently, a commercial "Interval Fortran" compiler has been released as well. Information on the available software, current research and application results, active research groups and the like can be found easily on the Internet via the main interval computation site at `http://cs.utep.edu/interval-comp/main.html`.

In this chapter, after a short introduction to basic concepts of interval calculations in Section III.1, the diagrammatic notation for interval algebra developed by this author is

described in Section III.2, and then its use for various interval problems is elaborated in some detail. The areas covered here include the theory of interval relations (Section III.3), interval arithmetic (Section III.4), and interval linear equations (Section III.5).

III.1 Interval algebra and computation

It was not until after careful calculation and deep thought
that the timbers were laid on the keel.
[Jules Verne, *The Mysterious Island* (1874)]

Let $(E, \leq)$ be a partially ordered set (called in the sequel a *base set*). Then, an interval can be generally defined as follows:

Definition III.1 (Intervals) *An* interval *(over the base set E) is an ordered pair $u = [e_1, e_2]$, where $e_1, e_2 \in E$, called* endpoints *of the interval, fulfill the condition $e_1 \leq e_2$.*

The interval is called *thick* if $e_1 < e_2$; *thin* (or *point*) interval if $e_1 = e_2$. Thin intervals can be for most purposes identified with the corresponding element of the base set, i.e., $[e, e] = e$. The *beginning* and *end* of the interval u are denoted by $\underline{u}$ and $\overline{u}$, respectively. Thus, $u = [\underline{u}, \overline{u}]$. When u stands for a more complex expression, an operator notation is more convenient, namely $\underline{u} = \mathbf{lb}\, u$ (for *lower bound*), and $\overline{u} = \mathbf{ub}\, u$ (for *upper bound*).

Usually, an interval can be identified with the set of elements lying between its endpoints (including the endpoints), namely $u = \{e \,|\, \underline{u} \leq e \leq \overline{u}\}$. An *interior* of an interval is then defined as $\mathbf{int}\, u = \{i \,|\, \underline{u} < i < \overline{u}\}$. When the base set E is taken to be the set of real numbers $\mathbb{R}$, the corresponding intervals are usually defined alternatively as:

Definition III.2 (Real intervals) *A* (proper) real interval *is a closed, compact and bounded subset of $\mathbb{R}$. The set of all real intervals is denoted by $\mathbb{IR}$ and called a* (real) interval space.

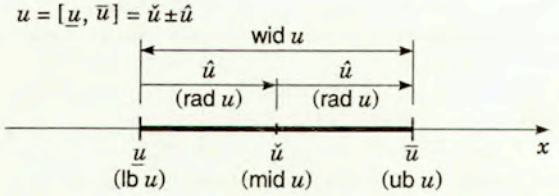


Figure III.1: A real interval and its basic parameters.

The term "proper" used above refers to the fact that in the algebra of *directed* (or *Kaucher*) intervals (see Section III.4.4) another kind of intervals, called "improper," appears as well. Moreover, in that extension intervals cannot be identified with sets of numbers contained in them, as two distinct intervals can correspond to the same number set. Thus, the above definition is not applicable to the extended intervals without modifications.

For real intervals, the *midpoint*, *radius* and *width* of the interval are also defined, respectively, as follows, see Fig. III.1:

$$\check{u} = \mathbf{mid}\, u = (\underline{u} + \overline{u})/2,$$
$$\hat{u} = \mathbf{rad}\, u = (\overline{u} - \underline{u})/2,$$
$$\mathbf{wid}\, u = \overline{u} - \underline{u} = 2\,\hat{u}\,.$$

With these parameters, another notation for real intervals, called sometimes a *centred* notation (introduced already in [Warmus 1956]) becomes possible:

$$\check{u} \pm \hat{u} = [\check{u} - \hat{u}, \check{u} + \hat{u}]. \tag{III.1}$$

The "$\pm$" notation was introduced in [7] as a convenient shorthand for the Warmus' "Λ-notation":

$$u = \check{u} \pm \hat{u} = \check{u} + \Lambda\,\hat{u}, \quad \text{where } \Lambda = [-1, 1].$$

An interval s with the midpoint equal to zero, i.e. such that $s = 0 \pm r = [-r, r] = r\,\Lambda$; $r \in \mathbb{R}^+$ is called a *zero-symmetric*, or simply a *symmetric interval*.

Two more parameters of intervals are in use, namely *mignitude* and *magnitude* (the latter called also an *absolute value*), giving the smallest (respectively the largest) distance from zero of the approximate number represented by the interval:

$$\langle u \rangle = \mathbf{mig}\, u = \min\{|\tilde{u}| \mid \tilde{u} \in u\} = \begin{cases} 0, & \text{if } 0 \in u, \\ \min\{|\underline{u}|, |\overline{u}|\} & \text{otherwise;} \end{cases}$$
$$|u| = \mathbf{mag}\, u = \max\{|\tilde{u}| \mid \tilde{u} \in u\} = |\,\check{u}\,| + \hat{u} = |\,\check{u} + \hat{u}\ \mathbf{sgn}\ \check{u}\,|. \tag{III.2}$$

With real intervals interpreted as sets of reals, a set-theoretic operations and relations on intervals can be used, in particular:

$$u \cap v = \{\tilde{u} \in \mathbb{R} \mid \tilde{u} \in u \ \mathbf{and}\ \tilde{u} \in v\} = \begin{cases} \emptyset, & \text{when } \overline{u} < \underline{v} \ \mathbf{or}\ \overline{v} < \underline{u}, \\ [\max\{\underline{u}, \underline{v}\}, \min\{\overline{u}, \overline{v}\}] & \text{otherwise.} \end{cases} \tag{III.3}$$

$$u \subseteq v \iff (\forall \tilde{u} \in u)\tilde{u} \in v \iff \underline{v} \leq \underline{u} \ \mathbf{and}\ \overline{u} \leq \overline{v} \iff |\,\check{v} - \check{u}\,| \leq \hat{v} - \hat{u}\,. \tag{III.4}$$

A diagrammatic illustration of the inclusion formula (III.4) is given in Fig. III.9 in Section III.2.3.3.

In this way, a real interval can be interpreted as an approximate (uncertain) number, such that its unknown exact value lies somewhere within the interval (between its endpoints). The notation $\tilde{u}$ is used to represent such an arbitrary number included in the interval u.

In addition to this endpoint interpretation, in many practical and theoretical situations it is natural to use the *midpoint-radius* (or *centred*) representation. In the centred representation, we consider the "nominal" value of an uncertain entity (midpoint or centre of the corresponding interval) together with an upper bound of its measurement error (radius of the interval). In consequence, when calculating interval bounds of some function with interval arguments, we try to calculate first the centre of the result and then estimate the maximal error of the result from the errors of the arguments.

The notion of an interval is straightforwardly extended to multidimensional intervals as well, see Section III.1.1.1.

III.1.1 Calculating with intervals

... racing across the bridge
at intervals carefully calculated ...
[Arthur C. Clarke, *The Fountains of Paradise* (1978)]

Taking into account the interpretation of a real interval as an approximate number, one may easily extend the arithmetic of real numbers to intervals. To convey the interpretation of approximate numbers to the results of operations on them, the result (i.e., the set of possible values for the resulting number) should be obviously a set of all results of the given operation on possible values of argument numbers. The same rule should apply for any functions on approximate numbers. Thus, for $\diamond \in \{+, -, \cdot, /\}$ and $f : \mathbb{R}^n \to \mathbb{R}$ we get:

$$\begin{array}{l} x \diamond y = \{r_1 \diamond r_2 \,|\, r_1 \in x \textbf{ and } r_2 \in y\}, \\ f(x_1, x_2, \ldots, x_n) = \{f(r_1, r_2, \ldots, r_n) \,|\, r_i \in x_i, i = 1, 2, \ldots, n\}. \end{array} \tag{III.5}$$

For arithmetical operations, the resulting sets of values are themselves intervals, and their parameters can be calculated using only parameters of argument intervals, according to the formulae (given already by [Warmus 1956] and [Sunaga 1958]):

$$x + y = [\underline{x} + \underline{y}, \overline{x} + \overline{y}] = (\check{x} + \check{y}) \pm (\hat{x} + \hat{y}), \tag{III.6}$$

$$-x = [-\overline{x}, -\underline{x}] = -\check{x} \pm \hat{x}, \tag{III.7}$$

$$x - y = x + (-y) = [\underline{x} - \overline{y}, \overline{x} - \underline{y}] = (\check{x} - \check{y}) \pm (\hat{x} + \hat{y}), \tag{III.8}$$

$$r\,x = r\,\check{x} \pm |r|\,\hat{x} = \begin{cases} [r\,\underline{x}, r\,\overline{x}] & \text{for } r \geq 0, \\ [r\,\overline{x}, r\,\underline{x}] & \text{for } r < 0, \end{cases} \tag{III.9}$$

$$x \cdot y = [\min\{\underline{x}\underline{y}, \overline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\overline{y}\}, \max\{\underline{x}\underline{y}, \overline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\overline{y}\}] = \tag{III.10}$$

$$\begin{array}{l} = (\check{x}\check{y} + \min\{\hat{x}\,|\check{y}|, |\check{x}|\hat{y}, \hat{x}\hat{y}\} \cdot \textbf{sgn}\ \check{x}\check{y}) \\ \quad \pm(\hat{x}\,|\check{y}| + |\check{x}|\,\hat{y} + \hat{x}\hat{y} - \min\{\hat{x}\,|\check{y}|, |\check{x}|\,\hat{y}, \hat{x}\hat{y}\}), \end{array} \tag{III.11}$$

$$\begin{array}{ll} x/y = & x \cdot (1/y) = x \cdot [1/\overline{y}, 1/\underline{y}] = x \cdot y/\underline{y}\overline{y} = x \cdot y/(\check{y}^2 - \hat{y}^2) \\ & \text{if } 0 \notin y, \text{ undefined otherwise.} \end{array} \tag{III.12}$$

However, sometimes (e.g., when the function is not continuous, see the *wrapping effect* in Section III.1.1.4 below) the resulting set may not be an interval. If it is bounded, it can be approximated by an interval obtained by the operation of *interval hull*, defined as:

$$\textbf{hull}\ S = [\inf S, \sup S], \tag{III.13}$$

with S a bounded subset[1] of $\mathbb{R}$. In such a case, the formulae (III.5) should be used in an extended version:

$$\begin{array}{l} x \diamond y = \textbf{hull}\,\{r_1 \diamond r_2 \,|\, r_1 \in x \textbf{ and } r_2 \in y\}, \\ f(x_1, x_2, \ldots, x_n) = \textbf{hull}\,\{f(r_1, r_2, \ldots, r_n) \,|\, r_i \in x_i, i = 1, 2, \ldots, n\}. \end{array} \tag{III.14}$$

[1] For unbounded subsets, the hull may be either left undefined, or else considered to be an *exterval*, see Section III.4.5.

III.1.1.1 Interval vectors and matrices

Vectors and matrices whose elements are intervals (possibly some of them thin) are called *interval vectors* and *interval matrices*. Interval vectors are often treated as *multidimensional* intervals (*boxes*). As vectors are, in a sense, a special case of matrices, most of the discussion concerning matrices in the sequel applies equally to vectors.

Most operations on intervals can be extended to interval matrices by applying them componentwise to all matrix elements. In particular, *infimum*, *supremum*, *midpoint*, *radius*, *magnitude*, *intersection* and *inclusion*, with corresponding notation, are defined in this way, as are *addition* and *subtraction*. Hence, an interval matrix $A \in \mathbb{IR}^{n\times m}$ can be also considered as a *set of real matrices*: $A = \{\tilde{A} \mid \underline{A} \le \tilde{A} \le \overline{A}\}$, or as a *matrix interval*: $A = [\underline{A}, \overline{A}] = [\check{A} - \hat{A}, \check{A} + \hat{A}]$. Matrix multiplication is defined like for real matrices, but usually the formula (III.14) should be used, because the set $\{\tilde{A}\tilde{B} \mid \tilde{A} \in A \text{ and } \tilde{A} \in B\}$ in general may not be an interval matrix (see Example III.2. It is also important to remember that multiplication of interval matrices (contrary to the non-interval matrices and scalar intervals) is not associative, thus in general $A(BC) \neq (AB)C$, unless A and C are thin (i.e., real) matrices.

Definition III.3 (Vertex matrix) *The* vertex matrix *of an interval matrix A is any real matrix from the* vertex set **vert** A *of A, defined as:*

$$\textbf{vert}\, A = \{\tilde{A} \in A \mid \tilde{a}_{ij} \in \{\underline{a}_{ij}, \overline{a}_{ij}\}\}.$$

The number of vertex matrices is equal to 2^t, where t is the number of thick interval coefficients of A. They were introduced by [Rohn 1989]. For a one-dimensional interval u, obviously $\textbf{vert}\, u = \{\underline{u}, \overline{u}\}$. Also, both $\underline{A}$ and $\overline{A}$ are vertex matrices.

Definition III.4 (Regular and singular interval matrices) *A square interval matrix $A \in \mathbb{IR}^{n\times n}$ is called* regular *(or* non-singular*) if all real matrices $\tilde{A} \in A$ are non-singular; otherwise it is called* singular.

Definition III.5 (Inverse of an interval matrix) *An* inverse A^{-1} *of a regular interval matrix A is defined as:*

$$A^{-1} = \textbf{hull}\,\{\tilde{A}^{-1} \in \mathbb{R}^{n\times n} \mid \tilde{A} \in A\}.$$

It is important to note that usually the set of inverses of matrices belonging to A is not an interval matrix, hence the **hull** operator in the above formula is in general necessary. If both $\underline{A}$ and $\overline{A}$ are non-singular and $\underline{A}^{-1}, \overline{A}^{-1} \ge 0$, we have simply $A^{-1} = [\underline{A}^{-1}, \overline{A}^{-1}] \ge 0$.

To avoid confusion with intervals, the double brackets $[\![\cdots]\!]$ will be used in the sequel for explicit matrices and vectors.

III.1.1.2 Nonstandard properties of interval arithmetic

> ... what is out of the common
> is usually a guide rather than a hindrance.
> [Arthur Conan Doyle, *A Study in Scarlet* (1887)]

It is often useful to think of intervals as simply *approximate numbers* and thus consider interval arithmetic as a natural extension of number arithmetic. However, one must

constantly bear in mind that algebraic properties of interval arithmetic are significantly different than that of the number arithmetic. First, quite naturally, negation of an interval (see (III.7)) negates only its midpoint, leaving radius unchanged. As a consequence, subtraction of intervals (see (III.8)) subtracts midpoints, but adds the radii, actually increasing them. Hence, for thick u one has $u - u = 2\hat{u} \neq 0$, which means that subtraction ceases to be an opposite operation to addition. Therefore, the algebraic transformation rule of moving a term (with opposite sign) to the other side of an equation does not work, causing solving even simple interval equations to become a nontrivial task. For example, the simplest equation $a + x = b$ for thick a does not always have a solution, see Section III.4.1.3, and even when it has, generally $x \neq b - a$. Similar situation occurs for multiplication and division: for thick u one has $u \cdot (1/u) \neq 1$, the equation $a \cdot x = b$ does not always have a solution, and when it has, generally $x \neq b/a$, see Section III.4.2.3. As a result, even the notion of solution for such an equation must be extended: several different types of solutions to this equation (and its multidimensional generalization $A \cdot x = b$ with $A \in \mathbb{IR}^{n \times n}$ and $b \in \mathbb{IR}^n$) are in use, see Section III.5.1.

Moreover, interval addition and multiplication cease to be distributive; instead, a weaker *subdistributivity* property holds:

$$x \cdot (y + z) \subseteq x \cdot y + x \cdot z. \tag{III.15}$$

Therefore, expanding factors in an interval expression may lead to a wider result than the original one. In general, the arithmetic expressions equivalent within the real number arithmetic are not necessarily equivalent in the interval arithmetic, which can lead to significant errors or overestimations in computation if not properly taken into account.

Before analysing the problems of overestimation we must first rigorously define what it means to approximate a real function by its *interval enclosures*.

III.1.1.3 Interval enclosures

Interval computations are generally concerned with approximation of real-valued quantities and functions with appropriate intervals containing all their possible values in a given situation. Especially, one usually seeks the smallest interval enclosing the range of values of some function for the given (interval) range of its arguments. The basic notions involved and their basic properties are as follows.

Let $f : \mathbb{R}^n \to \mathbb{R}$ denotes some real function, while $f_I, h_I : \mathbb{IR}^n \to \mathbb{IR}$ denote interval functions.

Definition III.6 (Range function) *The* range function *of a real function f is a function $f_R : \mathbb{IR}^n \to 2^{\mathbb{R}}$ defined as (cf. formulae (III.5)):*

$$f_R(u) = \{f(x) \,|\, x \in u\}.$$

The range function thus provides the set of all values the function takes for arguments from any given interval.

Definition III.7 (Interval enclosure) *An interval function $f_I(u)$ is called the* interval enclosure *(or* inclusion function*, see* [Jaulin et al. 2001]*) for a real function f if:*

$$(\forall u \in \mathbb{IR}^n) f_R(u) \subseteq f_I(u).$$

The interval enclosure is called minimal *if $f_I(u) = \textbf{hull}\, f_R(u)$.*

Interval enclosures are thus the basic objects of interval computations. The more precise estimation (the smaller overestimation) of the range function the better. The best estimation possible is obviously given by an interval hull of the range function.

Definition III.8 (Interval extension) *An interval function $f_I(u)$ is called the* interval extension *of a real function f if:*

$$(\forall x \in \mathbb{R}^n) f(x) = f_I([x, x]).$$

As real numbers can be practically considered a subset of real intervals (they are isomorphic to the set of thin real intervals), the condition simply asks for the interval extension to agree with the real function for all real arguments.

Definition III.9 (Inclusion isotonicity) *An interval function $f_I(u)$ is* inclusion isotonic *if:*

$$u \subseteq v \Rightarrow f_I(u) \subseteq f_I(u).$$

This property requires that for the subset (subinterval) of the set of argument values the set of values of the function should be also a subset of the set of function values for the larger set of argument values. Although inclusion isotonicity seems so natural a property of interval functions, it is not hard to give examples of functions that are not inclusion isotonic.

Example III.1 (Non-inclusion isotonic function) Consider $h_I(u) = u + \overline{u}$. We have $h_I([0,1]) = [1,2]$ and $h_I([0,2]) = [2,4]$, so that while $[0,1] \subseteq [0,2]$, obviously $h_I([0,1]) \not\subseteq h_I([0,2])$. ■

Theorem III.1 (Fundamental theorem of interval calculation) *If an* interval extension f_I *of a real function f is* inclusion isotonic, *then it is also an* interval enclosure *of f.*

The fundamental role of this theorem comes from the fact that it assures that with a straightforward interval extensions of real functions, if only they are inclusion isotonic (as almost all functions of practical interest are), we can safely compute the guaranteed ranges of these functions for every possible interval arguments.

The most important interval extension for arithmetic operations and elementary real functions is defined as:

Definition III.10 (Natural interval extension) *Let a real function $f(x_1, \ldots, x_n)$ be expressed as a superposition of the arithmetic operators $+, -, \cdot, /$ and elementary functions like* $\sin, \cos, \exp, \mathrm{sqrt}$, *etc. Then the expression obtained by replacing all real variables in the expression for f by corresponding interval variables and each operator and function by its interval extension defines the* natural interval extension f_N.

Theorem III.2 (Natural interval enclosure) *The natural interval extension f_N given by* Definition III.10 *is also an* inclusion isotonic interval enclosure *of f. If, moreover, the expression for f involves only continuous operators and functions, and each of its variables occurs at most once in the expression, then that interval enclosure is also* minimal.

III.1.1.4 Overestimation

The subdistributivity law (III.15) is an example of the *overestimation* problem in interval arithmetic. Due to that effect, one obtains interval results that are wider (contain more values) than the set of all results of the computation on members of argument intervals as given by (III.5). The effect may be harmful when one tries to conduct interval computations in a naive way, simply replacing numerical values by intervals in the original expressions and performing computation on them with the rules of interval arithmetic given by formulae (III.6–III.12), i.e., by relying only on the natural interval extensions (Definition III.10) of the functions one wants to compute. Fortunately, there are various methods of avoiding or substantially decreasing these effects that can be used to advantage in computations, see e.g. [Jaulin et al. 2001, Neumaier 2003].

There are generally two sources of overestimation in interval computations, called *wrapping effect* and *variable dependence effect*.

Wrapping effect. When the range function of some real function obtained according to (III.5) is not an interval, it must be "wrapped" in an interval according to (III.14). This may introduce into the resulting interval many additional values not contained in the original set, especially for certain shapes of that set in multidimensional cases. In a one-dimensional case that occurs only when the function is not continuous, while in higher dimensions the effect is much more common.

Example III.2 (Wrapping effect) A simple one-dimensional example is provided by the sign function $f(x) = \mathbf{sgn}\, x$. For it we have $\mathbf{sgn}_R[-1,1] = \{-1,0,1\}$, i.e., a set of only three numbers, while $\mathbf{hull}\,\mathbf{sgn}_R[-1,1] = [-1,1]$, i.e., the whole interval $\Lambda = 0 \pm 1$, see Fig. III.2a. A two-dimensional example is shown in Fig. III.2b, where:

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix},\ u = \begin{bmatrix} [-1,0] \\ [1,2] \end{bmatrix},\ \text{and}\ A \cdot u = \mathbf{hull}\, B = \begin{bmatrix} [1,4] \\ [1,2] \end{bmatrix} \neq B = \{A \cdot x \mid x \in u\}.$$

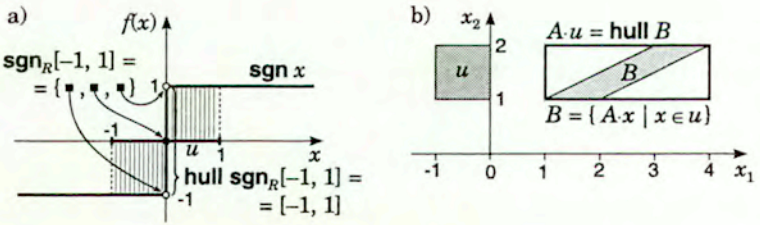


Figure III.2: Wrapping effect examples in one (a) and two (b) dimensions. ■

The wrapping effect may become especially harmful in certain kinds of iterative computations where the resulting multidimensional interval is transformed (e.g., rotated) in every iteration in a way making its subsequent wrapping significantly increasing its size.

Variable dependence effect. Definitions of interval arithmetic operations and functions quite sensibly assume that possible changes of approximated arguments within their

intervals are independent of each other. Thus, when the same interval argument is repeated in the expression to be computed, standard interval arithmetic cannot properly account for the fact that for that argument, in all its occurrences in the expression, the approximated value should be considered as being the same. In consequence, usually the result computed in a standard way can be significantly overestimated.

Example III.3 (Variable dependence) Taking a simple real function $h(x) = x^2 + x$, one can see that its natural interval extensions that are based on different forms of expressions for the function, see Fig. III.3a, are not equivalent, as shown in Fig. III.3b. E.g.,

a)

$$\begin{aligned} h(x) &= x^2 + x = x \cdot x + x = \\ &= x(x+1) = (x+1/2)^2 - 1/4. \end{aligned}$$

$$\begin{aligned} h_{I1}(u) &= u^2 + u \\ h_{I2}(u) &= u \cdot u + u, \\ h_{I3}(u) &= u \cdot (u+1), \\ h_{I4}(u) &= (u+1/2)^2 - 1/4. \end{aligned}$$

b)

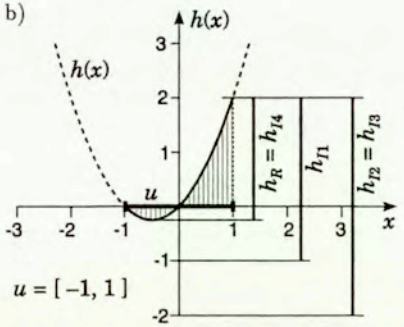


Figure III.3: Overestimation due to variable dependence: the real function $h(x)$ and its four interval enclosures (a), with corresponding range estimations for $u = [-1, 1]$ (b).

for the interval argument $u = [-1, 1]$, only the extension $h_{I4}(u)$ which contains the argument variable u only once gives no overestimation, while other forms produce substantial overestimations of the range $h_R(u) = [-1/4, 2]$ of the function. ■

The remedy for that is to transform the expression to a form with only a single occurrence of every variable (like $h_{I4}(u)$ in the example, see the condition for minimality of the enclosure in Theorem III.2) or to some other forms that do not produce overestimation of this sort, see e.g. [Alefeld & Herzberger 1983]. Other remedies include the use of other interval extensions than the natural one, see e.g. [Jaulin et al. 2001, Neumaier 2003].

III.1.2 Applications of interval computation

There's no sense in being precise
when you don't even know what you're talking about.
[John von Neumann (1903-1957)]

Computing with intervals instead of exact numbers is useful, or even recommended, in all cases where the quantities involved are not known accurately, or cannot be accurately computed. In practice it means effectively in all computations involving real-world data and performed on our limited-accuracy computers. Some other, often unexpected applications, like global optimization or proving certain types of mathematical theorems (see e.g. [Frommer 2001]) emerged as well.

Rounding errors. Ordinary floating-point computations, due to possible cumulation of rounding errors, cannot guarantee that the final result will be ever approximately accurate, without an additional, complicated and tedious (hence rarely performed with necessary thoroughness) error analysis. Many practical examples show that convincingly—many of them are reported in the literature under the quite appropriate name of *numerical monsters*, see e.g. [Essex et al. 2000]. To add insult to injury, the standard numerical procedures do not produce any indication or warning of possible occurrence of the error. Worse still, many intuitively obvious remedies unfortunately do not solve the problem.

Example III.4 (Increasing precision may increase error) Increasing the precision of computer arithmetic does not necessarily decrease the computation error; actually, in some cases it may even increase it, as it is shown in Table III.1. Increasing the precision of calculating the product $0.90005 \cdot 0.90005$ from two to four decimal places actually increases the error of the result, while the interval computation produces estimating intervals a hundred times narrower, as expected. ■

Table III.1: Increasing precision sometimes increases rounding errors (ϵ_2, ϵ_4: absolute errors for numerical computations; w_2, w_4: widths of resulting intervals for interval calculations).

Exact:	$0.90005 \cdot 0.90005$	$=$	0.8100900025			
Two decimal places:						
	$0.90 \cdot 0.90$	$=$	$0.81,$	ϵ_2	$=$	0.0000900025
	$[0.90, 0.91] \cdot [0.90, 0.91]$	$=$	$[0.81, 0.8281]$			
		$\approx$	$[0.81, 0.83];$	w_2	$=$	0.02
Four decimal places:						
	$0.9001 \cdot 0.9001$	$=$	0.81018001			
		$\approx$	$0.8102,$	ϵ_4	$=$	$0.0001099975 > \epsilon_2$
	$[0.9000, 0.9001] \cdot [0.9000, 0.9001]$	$=$	$[0.8100, 0.81018001]$			
		$\approx$	$[0.8100, 0.8102];$	w_4	$=$	$0.0002 < w_2$

Computing with uncertain data. The situation becomes even worse when the input data are not accurately known. With standard techniques, there is practically no other way to take that into account than repetition of the whole calculation for different possible values of the data, in some systematic or randomized way (cf. *Monte Carlo* methods). Still there is no guarantee that the range of obtained results will approximate in any accuracy the true range of possible outcomes. Sensitivity analysis gives acceptable results only for small perturbations of the input data. Stochastic approach is complicated, requires knowledge about the probability distribution of input data (rarely available with sufficient accuracy in practice), and gives only a probability of the result falling within a given range, without any guarantee that it will not fall out of the (safe) range. Thus, the worst case analysis needed in many practical situations is impossible to perform with that approach. Moreover, it cannot sensibly account for rounding errors.

Interval computations, if properly applied, can solve most of these problems. Uncertain data are easily modelled as intervals giving possible ranges of uncertain quantities. An outward rounding of interval endpoints during computation takes account of rounding errors. Further on, the properties of interval algebra discussed in Section III.1.1.3 assure that the interval obtained at the end is *guaranteed* to contain the true outcome, and its width summarizes all uncertainty sources in the problem. In this way, a *guaranteed accuracy* of computation is achieved. If the resulting interval seems much too wide than expected, it is a warning that something may be wrong with the formulation of the problem or the choice of computational algorithm for the problem at hand (e.g., that the matrix is too close to singularity). Another cause of too wide a result may be an improper formulation of the interval computation method, e.g., due to not taking proper precautions against overestimation effects discussed in Section III.1.1.4. The reader interested in these application issues is referred to a comprehensive literature on the theme, especially the recent book by [Jaulin et al. 2001].

Global optimization. The great problem with traditional optimization methods was that they cannot guarantee finding the global optimum in any given region of the parameter space. There is always a possibility that the search for the optimum becomes stuck in some local extremum, and, moreover, there is no general method to even detect that fact: we can never be sure whether we found the global, or only a local extremum. Interval methods can solve this problem, and, moreover, they do not in principle require any restrictive conditions on the behaviour of the objective function, like continuity or differentiability. What is only required is the existence of a method for calculation of a reasonable outer interval estimate of the range of the function for any given interval of its arguments, i.e., an interval enclosure of the objective function must be available, see Section III.1.1.3. The interested reader is referred to the comprehensive book of [Hansen 1992], see also [Jaulin et al. 2001].

III.1.3 Diagrams for interval algebra

> Where was the ... knowledge that could perhaps
> give some meaning to the empty equations?
> [Isaac Asimov, *Prelude to Foundation* (1988)]

Although in the context of computation one may often think of an interval as simply an approximate number, a too literal transfer of understanding of number arithmetic to operations on intervals may lead to severe blunders. Interval arithmetic possesses significantly different properties than ordinary arithmetic, see Section III.1.1.2, which may be hard to intuitively grasp on the basis of analogy with number arithmetic only. Similarly as it happened with complex numbers, one of the devices that can make the understanding of these nonstandard properties of "interval numbers" easier is an appropriate diagrammatic tool, like that developed by this author [7, 12, 13, 25, 105, 107] and presented in this chapter. Besides facilitating such understanding, the interval diagrams can be also used as research tools to study properties of various mathematical objects based on intervals, like interval relations ([12, 13], Section III.3), interval arithmetic operations ([7], Section III.4), and interval linear equations ([4, 5, 26], Section III.5).

The diagrammatic notation developed in this chapter is generally aimed for the use by humans, both as an education tool and a research tool. However, its usefulness will much benefit from a proper computer implementation. Of course, the implementation would *not* be practical for conducting interval *calculations*. Its main role is to help the user in the process of gaining understanding of various properties of interval operations, algorithms, etc., in search for and exploration of useful properties, and in reasoning about them and proving them. For these purposes, an implementation like the *diagrammatic spreadsheet* described in Section II.6.4 might possibly turn out to be practical.

Already all the diagrams used in the research reported in this chapter were produced by the author himself on a computer (sometimes, but not always, on the basis of some hand-made sketches), although the tools used were only a standard graphic editor.

III.2 Interval space diagrams

"The diagram is now complete," he said. ...
"It is not only complete, there are no extra lines. ..."
[Robert L. Forward, *Dragon's Egg* (1980)]

Simple diagrams appeared in the interval literature from the very beginning [Sunaga 1958, Warmus 1961, Moore, R.E. 1966, Kaucher 1973, Gardeñes et al. 1981], also in the time-interval research [Rit 1986, Nökel 1991], albeit rather sporadically and to a limited extent, only as informal illustrations for some concepts and properties. They were neither systematically investigated, nor more widely applied in interval research and applications. The case of the Kaucher dissertation [Kaucher 1973] is especially interesting here. The unpublished dissertation contained many diagrams, some of them using essentially the same concept of interval space as that presented here, and were used to illustrate various concepts and properties of *directed* interval arithmetic (see Section III.4.4). However, further published works by Kaucher do not contain any diagrams of this sort.

It seems that systematic investigation of interval space diagrams and their prospective applications started only with the works by this author, see especially [7, 12, 13, 26, 107].

III.2.1 The E-diagram and other proposals

The first step on the way to construct a diagrammatic representation and reasoning system for interval algebra is to construct an appropriate diagram capable of representing arbitrary intervals and their sets in a uniform graphical way. In other words, we need a diagram for interval space. Because two parameters are needed to uniquely characterize an interval (see the previous section), the interval space is basically two dimensional. There are many possible choices of the two parameters to be used as coordinates of the space. As the endpoint representation of an interval was the most common, it seemed all too natural to choose the endpoints as coordinates. This led to an interval space representation called here an *E-diagram* (from *endpoints diagram*), see Fig. III.4a. A drawing using these coordinates appeared already in [Warmus 1961], as well as in other early works like [Sunaga 1958, Moore, R.E. 1966, Ratschek 1973]. It was also used in [Rit 1986], which inspired this author [107]. However, this representation has several drawbacks making it inconvenient to use for more complicated interval diagrams. Among others, the set of proper intervals occupies an inconveniently skewed upper-left half-plane above the diagonal, with coordinate axes spanning it diagonally. This makes the reading of other interval parameters awkward, notably midpoint and radius, see Fig. III.4b.

Therefore, the author tried other choices. After first experiments with the midpoint-width coordinates [107], soon the midpoint-radius combination proved to be much better [12, 13], resulting in the diagram called here an *MR-diagram* (from *midpoint-radius diagram*). These coordinates also appeared in [Warmus 1961], and later in some drawings in the dissertation of [Kaucher 1973]. This choice goes in accord with several opinions to the effect that the midpoint-radius representation has many advantages in various applications and theoretical considerations—the opinion already implicitly expressed in the early papers [Warmus 1956, Sunaga 1958, Warmus 1961], see also the recent paper by [Markov 2001b].

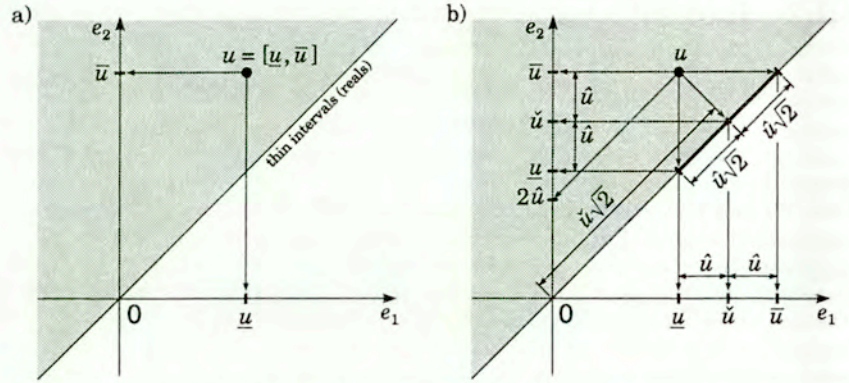


Figure III.4: The E-diagram: the proper interval space marked in gray (a); and constructions for reading off of midpoint and radius parameters (b).

Other, rather exotic coordinate choices were considered in [Ratschek 1980], with the aim of finding a space in which the multiplication can be done componentwise, like addition in standard vector spaces. A three-dimensional space with this property has been indeed found by him and presented there; he also proved that there does not exist a space in which both addition and multiplication of intervals can be simultaneously done componentwise.

III.2.2 The MR-diagram

A graphical definition of the MR-diagram, first introduced in [7, 105] is given in Fig. III.5. The (m, r) coordinate system, its axes, characteristic points and regions with names and symbolic denotations are given in Fig. III.5a. Note that the region used to represent intervals consists only of the upper half-plane above the **Om** axis.

The representation of an interval $u = \check{u} \pm \hat{u} = [\underline{u}, \overline{u}]$ is shown in Fig. III.5b. This figure defines also additional graphical elements, notably the *constant midpoint line* and *constant radius line*, as well as diagonal *constant lb-line* and *ub-line*. All points lying on these lines have the same value as the interval u of one of their parameters (*midpoint*, *radius*, *beginning* and *end*, respectively). Note the line style (light dotted) used for diagonal lines in the diagram. In all subsequent diagrams, this line style will be used for these lines only. That is, all lines drawn with this style are assumed to make the 45° angle with the **Om** axis. In the midpoint-radius representation given here, the endpoints of an interval are conveniently represented too, with the help of the diagonal lines. Actually, any pair of the four basic parameters $\check{u}$, $\hat{u}$, $\underline{u}$, and $\overline{u}$ of an interval u uniquely determines the point representing that interval, by an intersection of two of the lines shown. As a result, the standard interpretation of a real interval—as a segment of the real number line (here, the **Om** axis)—can be also represented, and the endpoint representation of intervals can be easily handled, contrary to the situation with the E-diagram which is awkward to use with the centred representation, see Fig. III.4b.

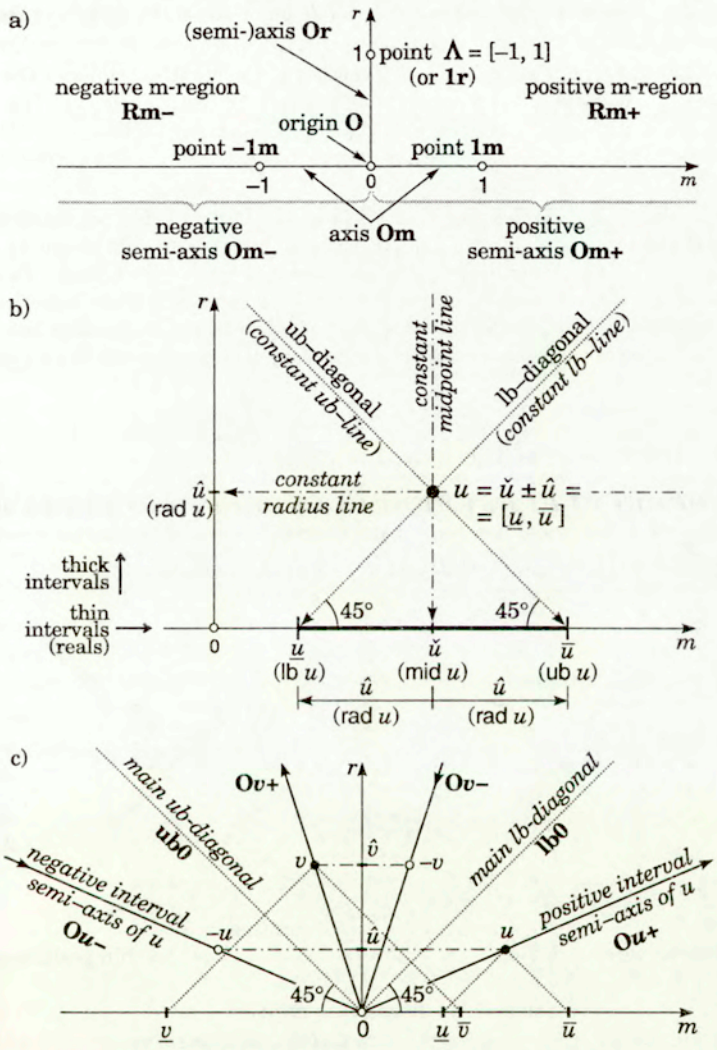


Figure III.5: The MR-diagram: axes, points and regions (a); representation of an interval (b); interval axes and main diagonals (c).

In Fig. III.5c, *main diagonals* and *interval axes* are defined. The main diagonals are diagonal lines for the interval 0. The *lb-diagonal* **lb0** groups all intervals beginning at zero, while the *ub-diagonal* **ub0** groups all those that end at zero. They also serve as a dividing line between intervals *containing zero* (they all lie on or above the main diagonals) and those *without zero* (below the main diagonals), see below for more on interval types.

The *interval axis*[2] **Ou** of the interval u consists of the negative semi-axis **Ou**$-$ going from infinity through the interval $-u$ to the origin, and the positive semi-axis **Ou**$+$ going from the origin through the interval u to infinity. The axis of all thin intervals (reals) coincides with the **Om** axis, while the axis of zero-symmetric intervals coincides with the **Or** axis.

Observe also that a number (say a) marked on the **Om** axis denotes simultaneously some numerical value of an interval midpoint (for all intervals straight above it), as well as a thin interval (a real number) $[a, a] = a$ represented by that very point. This is not so with a number, say b, marked on the **Or** axis: it denotes only some numerical value of an interval radius (for all intervals lying on the horizontal constant radius line passing through the mark), while the interval represented by this point is the thick symmetric interval $[-b, b]$, not simply b.

III.2.3 Basic uses of the MR-diagram

In this section, the MR-diagram will be used to represent some basic notions concerning various interval types, some functions characterizing them, and the lattice structure of the interval space.

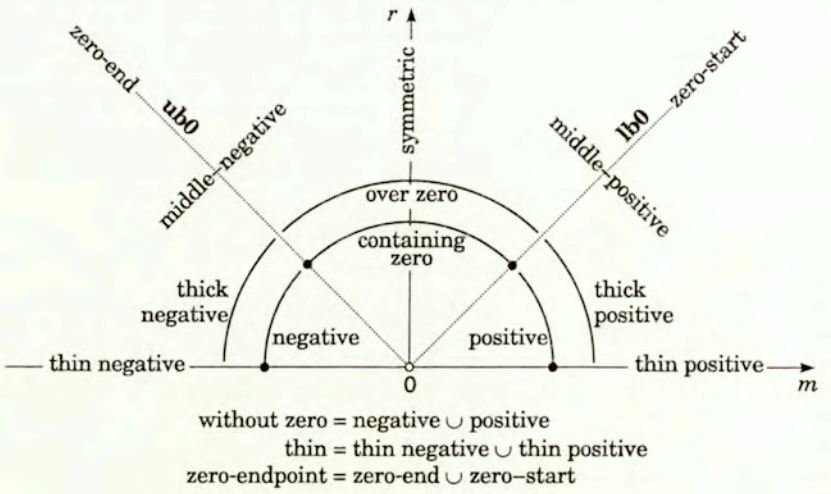


Figure III.6: The main types of intervals.

[2] In some of the previous papers and reports [7, 26, 105] it was called a "radial line." The term "interval axis" seems more appropriate, especially in the light of its properties discussed in the following.

III.2.3.1 Interval types

Several types of intervals are discerned, featuring different properties with respect to arithmetic (especially multiplication and division) and other operations on them. They can be characterized by their relation to the number 0 (or interval [0, 0]), see Section III.3.5. These types can be straightforwardly represented as regions (one- or two-dimensional) in the MR-diagram, see Fig. III.6, and their definitions in terms of basic interval parameters can be extracted from the diagram, see Table III.2 (cf. also Fig. III.5).

Note that the complexity of definitions of the types based on the centred representation compares favourably with those based on the endpoint representation—the centred conditions are of smaller or at most the same complexity.

Table III.2: Interval types distinguished by endpoint and centred parameters.

Type of u	MR-diagram region	Endpoints	Midpoint-radius
Without zero	Below **lb0-ub0**	$0 < \underline{u}$ **or** $\overline{u} < 0$	$0 < \hat{u} < \lvert \check{u} \rvert$
Positive	Below **lb0**	$\underline{u} > 0$	$\check{u} > \hat{u}$
Negative	Below **ub0**	$\overline{u} < 0$	$\check{u} < -\hat{u}$
Thick positive	Between **lb0** and **Om+**	$0 < \underline{u} < \overline{u}$	$0 < \hat{u} < \check{u}$
Thick negative	Between **ub0** and **Om−**	$\underline{u} < \overline{u} < 0$	$\hat{u} < -\hat{u} < 0$
Thin	On **Om**	$\underline{u} = \overline{u}$	$\hat{u} = 0$
Thin positive	On **Om+**	$\underline{u} = \overline{u} > 0$	$\check{u} > 0 = \hat{u}$
Thin negative	On **Om−**	$\underline{u} = \overline{u} < 0$	$\check{u} < 0 = \hat{u}$
Containing zero	On or above **lb0-ub0**	$\underline{u} \leq 0 \leq \overline{u}$	$-\hat{u} \leq \check{u} \leq \hat{u}$
Over zero	Above **lb0-ub0**	$\underline{u} < 0 < \overline{u}$	$-\hat{u} < \check{u} < \hat{u}$
Zero-start	On **lb0**	$\underline{u} = 0$	$\check{u} = \hat{u}$
Zero-end	On **ub0**	$\overline{u} = 0$	$\check{u} = -\hat{u}$
Zero-endpoint	On **ub0** or **lb0**	$\underline{u} = 0$ **or** $\overline{u} = 0$	$\lvert \check{u} \rvert = \hat{u}$
Symmetric	On **Or+**	$\underline{u} = -\overline{u}$	$\check{u} = 0$
Middle-positive	Within **Rm+** or on **Om+**	$\underline{u} > -\overline{u}$	$\check{u} > 0$
Middle-negative	Within **Rm−** or on **Om−**	$\underline{u} < -\overline{u}$	$\check{u} < 0$

III.2.3.2 Extent functions

Certain functions of basic interval parameters are useful to quantitatively describe the type of an interval. They characterise the *relative extent* of an interval with respect to its position (midpoint, or endpoints), that is, the relative *degree of uncertainty* of the number approximated by the interval. The basic functions here are the *endpoint-ratio* function **epr** and its reciprocal:

$$\begin{aligned} \mathbf{epr}\, u &= \overline{u}/\underline{u}, \\ 1/\mathbf{epr}\, u &= \underline{u}/\overline{u}. \end{aligned} \tag{III.16}$$

When $\underline{u} = 0$ (for $\mathbf{epr}\, u$) or $\overline{u} = 0$ (for $1/\mathbf{epr}\, u$), the value of the corresponding function is undefined; for some purposes it may be considered to equal $\pm\infty$. The most used in the literature is the so-called χ functional introduced by [Ratschek 1972] which is defined in terms of the above functions:

$$\chi u = \mathbf{chi}\, u = \begin{cases} -1, & \text{if } u = 0; \\ 1/\mathbf{epr}\, u, & \text{if } |\underline{u}| \le |\overline{u}|; \\ \mathbf{epr}\, u, & \text{otherwise.} \end{cases}$$

In this way, the χ function is always defined, and $-1 \le \chi u \le 1$. Finally, the *relative-extent* function **rex** (introduced in [7] and called also κ (kappa) by [Markov 2001b]), is used thorough this work because it is natural for the midpoint-radius representation:

$$\mathbf{rex}\, u = \hat{u} \,/\, \check{u} = (\mathbf{epr}\, u - 1)/(\mathbf{epr}\, u + 1) = \tan\alpha, \tag{III.17}$$

where $\alpha = \mathbf{arg}\, u$ is the angle between the $\mathbf{Om}+$ axis and the interval axis $\mathbf{Ou}+$.

For $\check{u} = 0$ the value of $\mathbf{rex}\, u$ is undefined; for some purposes it can be assumed to equal $\pm\infty$. For $u = [0, 0]$ the value of both χu and $\mathbf{rex}\, u$ is the same as for zero-symmetric intervals, i.e., the zero interval is assumed to belong to the $\mathbf{Or}+$ axis.

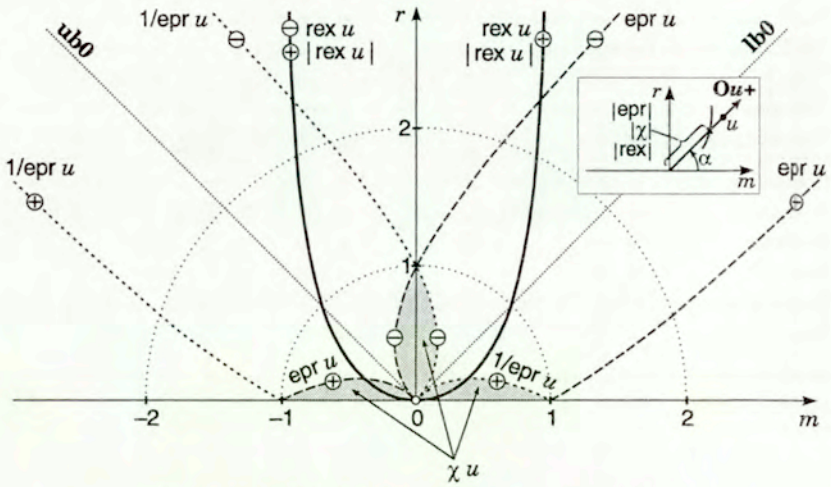


Figure III.7: Relative extent functions: polar graphs;
the graph for $|\mathbf{rex}\, u|$ differs from that for $\mathbf{rex}\, u$ by the sign of the left branch only.

As it was indicated for the **rex** function, the value of any of these functions is the same for intervals with the same $\alpha = \mathbf{arg}\, u$ parameter, i.e., for intervals lying on the same interval axis in the MR-diagram (excluding the zero interval). It is therefore possible to draw polar graphs of them directly in the MR-diagram, see Fig. III.7. Note the labels $\oplus$ and $\ominus$ on the graphs that indicate the sign of an appropriate branch of the given function. The functions **epr**, $1/\mathbf{epr}$, and **rex**, considered as functions of $\alpha = \mathbf{arg}\, u$, have well-defined inverses and are of essentially the same shape (namely, that of $\tan\alpha$), only phase-shifted

Table III.3: Interval types distinguished by extent functions.
Note the change of the last two types with respect to Table III.2, (see text).

Type of u	MR-diagram region	Using **rex** u	Using χu
Without zero	Below **lb0-ub0**	$0 \leq \lvert\mathbf{rex}\, u\rvert < 1$	$0 < \chi u \leq 1$
Positive	Below **lb0**	$0 \leq \mathbf{rex}\, u < 1$	—
Negative	Below **ub0**	$-1 < \mathbf{rex}\, u \leq 0$	—
Thick positive	Between **lb0** and **Om+**	$0 < \mathbf{rex}\, u < 1$	—
Thick negative	Between **ub0** and **Om−**	$-1 < \mathbf{rex}\, u < 0$	—
Thin	On **Om**	$\mathbf{rex}\, u = 0$	$\chi u = 1$
Thin positive	On **Om+**	—	—
Thin negative	On **Om−**	—	—
Containing zero	On or above **lb0-ub0**	$\lvert\mathbf{rex}\, u\rvert \geq 1$	$\chi u \leq 0$
Over zero	Above **lb0-ub0**	$\lvert\mathbf{rex}\, u\rvert > 1$	$\chi u < 0$
Zero-start	On **lb0**	$\mathbf{rex}\, u = 1$	—
Zero-end	On **ub0**	$\mathbf{rex}\, u = -1$	—
Zero-endpoint	On **ub0** or **lb0**	$\lvert\mathbf{rex}\, u\rvert = 1$	$\chi u = 0$
Symmetric	On **Or+**	$\mathbf{rex}\, u = \pm\infty$	$\chi u = -1$
Thick middle-positive	Within **Rm+**	$\mathbf{rex}\, u > 0$	—
Thick middle-negative	Within **Rm−**	$\mathbf{rex}\, u < 0$	—

by $\pm 45°$ (rotated in the polar graphs of Fig. III.7), while the χ function is a piece-wise combination of **epr** and $1/\mathbf{epr}$, and has no (single-valued) inverse.

The interval u is called *more extended* (or *more uncertain*) than the interval v when $|\mathbf{rex}\, u| > |\mathbf{rex}\, v|$, i.e., when u lies above the interval axis $\mathbf{O}v$ of the interval v in the MR-diagram (and hence, v lies below the interval axis $\mathbf{O}u$ of u). Obviously, thick intervals are always more extended (more uncertain) than thin intervals (i.e., reals), the symmetric intervals distinct from 0 are also considered more extended that all nonsymmetric ones. When $|\mathbf{rex}\, u| = |\mathbf{rex}\, v|$ (thus, $\mathbf{O}u = \mathbf{O}v$, ignoring the orientation), u and v have the same extent (are equally uncertain). As it easily follows from the definitions, u and v have the same extent iff $u = t \cdot v$ for some $t \in \mathbb{R}, t \neq 0$. The absolute value of the relative extent that is larger than or equal to unity, i.e. $|\mathbf{rex}\, u| \geq 1$, characterises intervals containing zero, which exhibit some peculiar properties, especially with respect to multiplication and division, while less extended intervals behave much more like ordinary real numbers. The over-unity extent may be considered to mean that the interval becomes relatively "too wide" to still remain similar to a "sharp," ordinary real number.

The basic interval types can be also distinguished by simple conditions on the (range of) values of these functions, as shown in Table III.3 (cf. also Figs. III.6 and III.7). Note that several types of intervals cannot be distinguished using conditions on the values of the χ function alone (unless we use conditions on the differential of the function), while the **rex** function distinguishes them easily. This is due to χ being not a one-to-one function of α. However, two types, "thin positive" and "thin negative," cannot be distinguished by specific values of either **rex** or χ function. In consequence, these functions also cannot define properly "middle-positive" and "middle-negative" types. Nevertheless, the **rex** function

is able to define the closely related "thick middle-positive" and "thick middle-negative" types, as shown in Table III.3. Anyway, the **rex** function seems to work better as a tool to characterise important types of intervals.

III.2.3.3 Interval lattices and lozenges

The interval space $\mathbb{IR}$ is partially ordered by interval inclusion defined by (III.4). Therefore it has a lattice structure, with the lattice *join* operator "$\vee$" defined in terms of the inclusion as:

$$u \vee v = \inf_{\subseteq}\{w \mid u \subseteq w \textbf{ and } v \subseteq w\}. \tag{III.18}$$

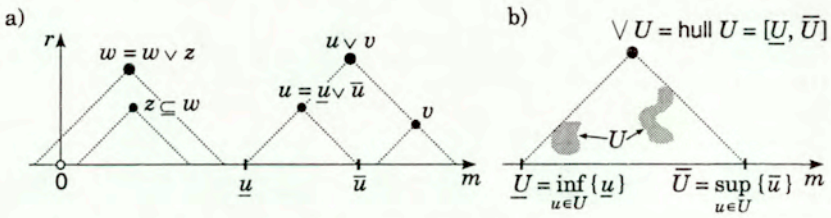


Figure III.8: Interval lattice operation *join* (a) and generalized interval hull (b).

Obviously, $u = u \vee v \iff v \subseteq u$ as well as $u = \underline{u} \vee \overline{u}$ and $u \vee v = v \vee u$. The join operation and its properties find easy representation in the MR-diagram, see Fig. III.8a. E.g., it is easy to see from the diagram that:

$$\textbf{rad}\,(u \vee v) \geq \max\{\hat{u}, \hat{v}\},$$
$$\min\{\check{u}, \check{v}\} \leq \textbf{mid}\,(u \vee v) \leq \max\{\check{u}, \check{v}\}.$$

The join operation coincides with the **hull** operation of (III.13), appropriately extended to intervals (treated as sets of reals):

$$u \vee v = \textbf{hull}\,\{u, v\} = \textbf{hull}\,(u \cup v) = [\min\{\underline{u}, \underline{v}\}, \max\{\overline{u}, \overline{v}\}]. \tag{III.19}$$

The hull operation can be naturally extended to arbitrary bounded sets of intervals, as shown in Fig. III.8b.

The dual *meet* operation, coinciding with interval intersection defined by (III.3), is rarely used (and needed) here, as it is not always defined (or alternatively one must add an empty interval into the algebra). It obtains full significance in directed (Kaucher) interval algebra, see Section III.4.4.

In addition to the lattice structure imposed by the "$\subseteq$" relation, another, orthogonal (in a way) lattice structure (the *precedence lattice*) is produced by the relations "$\preccurlyeq$" (*is-followed-by* or *precedes*) and its inverse "$\succcurlyeq$" (*follows* or *succeeds*). They are defined as follows:

$$u \preccurlyeq v \iff \underline{u} \leq \underline{v} \textbf{ and } \overline{u} \leq \overline{v},$$
$$u \succcurlyeq v \iff v \preccurlyeq u.$$

Diagrammatic comparison of these two structures is shown in Fig. III.9 which contains *coimages* of the interval u under the four ordering relations (anticipating the more general definitions of (III.20) in Section III.3: for any relation $\Diamond$, the *coimage* of an element u under this relation is a set $\Diamond u = \{v \mid v \Diamond u\}$). The figure contains also illustration of the numerical formulae for interval inclusion given by (III.4). The (dotted) diagonal lines belong to both adjoining regions. From the diagram it thus follows that when two intervals u and v do not lie on the same diagonal line, exactly one of the four ordering relations $\{\subseteq, \supseteq, \preccurlyeq, \succcurlyeq\}$ is sure to hold between them. For different intervals arranged on the same diagonal line, exactly two relations hold—one of the inclusion type, one of the precedence type. Obviously, all four relations hold when $u = v$.

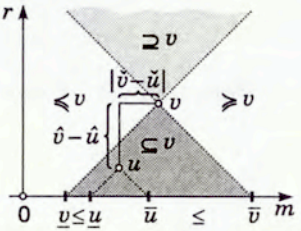


Figure III.9: Coimages of an interval v under interval partial order relations.

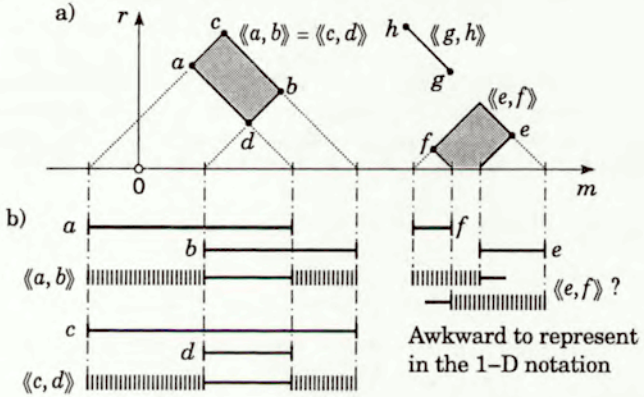


Figure III.10: Examples of lozenges (a) with one-dimensional representation included for comparison (b).

Definition III.11 (In-between interval relation) *When $u \lesseqgtr v$ for some ordering relation $\lesseqgtr \in \{\subseteq, \supseteq, \preccurlyeq, \succcurlyeq\}$, an interval x for which $u \lesseqgtr x \lesseqgtr v$ is called to lie* in-between *u and v.*

As one of these ordering relations must hold between any pair of intervals u and v, for every two distinct intervals there are some intervals lying in-between them.

Proposition III.1 (In-between interval relation) *Whenever some interval x lies in-between u and v, then also* $\min\{\underline{u}, \underline{v}\} \le \underline{x} \le \max\{\underline{u}, \underline{v}\}$ **and** $\min\{\overline{u}, \overline{v}\} \le \overline{x} \le \max\{\overline{u}, \overline{v}\}$, *i.e., endpoints of the interval x lie between the corresponding endpoints of u and v. The reverse is also true.*

Proof. See Fig. III.10 for a diagrammatic demonstration (cf. also Definition III.12). □

This allows for the definition of certain important interval sets called here *lozenges* [12, 13] (or *metaregions*):

Definition III.12 (Lozenge) *A* lozenge $\langle\!\langle u, v \rangle\!\rangle$, *defined by a pair od intervals* (u, v), *is a set of intervals whose endpoints lie in-between the intervals* u *and* v, *i.e.* $\langle\!\langle u, v \rangle\!\rangle = \{x \mid \min\{\underline{u}, \underline{v}\} \leq \underline{x} \leq \max\{\underline{u}, \underline{v}\}$ **and** $\min\{\overline{u}, \overline{v}\} \leq \overline{x} \leq \max\{\overline{u}, \overline{v}\}\} = \{x \mid u \lesseqgtr x \lesseqgtr v\}$ *for some* $\lesseqgtr \in \{\subseteq, \supseteq, \preccurlyeq, \succcurlyeq\}$.

As it is easily seen, the lozenge does not depend on the order of its arguments, so that $\langle\!\langle u, v \rangle\!\rangle = \langle\!\langle v, u \rangle\!\rangle$.

In Fig. III.10a some examples of lozenges are shown. Lozenges are closely related to the so-called *twins* [Gardeñes et al. 1980, Gardeñes et al. 1981], or *metaintervals* (intervals of intervals), that is, intervals with base sets $(\mathbb{IR}, \subseteq)$ or $(\mathbb{IR}, \preccurlyeq)$, see Definition III.1. Namely, a lozenge can be also defined as the set of intervals belonging to some twin. As it is easily seen in Fig. III.10a, some lozenges can be defined using either the inclusion relation, or else the precedence relation (usually using two different pairs of intervals, like $\langle\!\langle a, b \rangle\!\rangle = \langle\!\langle c, d \rangle\!\rangle$, except for thin lozenges like $\langle\!\langle g, h \rangle\!\rangle$), while other lozenges can be defined only by the precedence relation (using a single pair of intervals, like $\langle\!\langle e, f \rangle\!\rangle$). The latter are those "cut off" by the **Om** axis. The situation is different for the directed interval space of Section III.4.4, where all lozenges are of the first type. The first kind of lozenges afford a one-dimensional representation, as an interval with uncertain endpoints. That does not work well, however, for the second kind of lozenges, see Fig. III.10b.

III.3 Interval relations

... neither Man nor Woman shall be approached so closely
as to destroy the interval between the approximator and the approximated.
[Edwin A. Abbott, *Flatland: A Romance of Many Dimensions* (1884)]

The study of interval relations originated in AI research [Allen 1983], where they were used as tools for reasoning about time [Rit 1986, van Beek & Cohen 1990, Vilain et al. 1990, Nökel 1991], and for qualitative spatial reasoning [Mukerjee & Joe 1990]. As will be shown in Section III.5.2.1, they can be also useful in interval computations. The use of diagrams in that research has been limited to simple one-dimensional representations of intervals as segments of the number axis (the most advanced form of them described by [Schlieder 1996]), and the use of the lattice diagram of basic interval relations by [Freksa 1992]. In this section, several diagrammatic tools, useful in the study of interval relations (especially the two-dimensional *W-diagram*, Section III.3.2, and its derivatives) are presented, their basic properties discussed, and their usefulness shown with the help of several examples, including diagram-aided proofs of two theorems on different characterizations of certain important classes of interval relations [9, 12, 13].

First, let us recall the basic notation of algebra of relations. Let X, Y, Z, W be sets. Then $\diamondsuit \subseteq X \times Y$, $\heartsuit \subseteq Y \times Z$ are (binary) relations; $x \diamondsuit y$ means $(x, y) \in \diamondsuit$. As relations are sets, the union $\diamondsuit \cup \heartsuit$ and intersection $\diamondsuit \cap \heartsuit$ of relations are defined straightforwardly:

$$x(\diamondsuit \cup \heartsuit)y \iff x \diamondsuit y \text{ or } x \heartsuit y,$$
$$x(\diamondsuit \cap \heartsuit)y \iff x \diamondsuit y \text{ and } x \heartsuit y.$$

Notation $\diamondsuit \circ \heartsuit$ (shortly: $\diamondsuit\heartsuit$) is used for a composition of relations:

$$x \diamondsuit\heartsuit z \iff (\exists y) x \diamondsuit y \text{ and } y \heartsuit z.$$

A relation $\diamondsuit^{-1}$ such that $y \diamondsuit^{-1} x \iff x \diamondsuit y$ is called an inverse of the relation $\diamondsuit$. Obviously, $(\diamondsuit^{-1})^{-1} = \diamondsuit$. Note that the notion of inverse relation is in most part a notational convention—only positions of the arguments are exchanged, without real change of the underlying relationship between them.

The notation:

$$\begin{aligned} W\diamondsuit &= \{y \in Y \mid w \in W \text{ and } w \diamondsuit y\}, \text{ and} \\ \diamondsuit W &= \{x \in X \mid w \in W \text{ and } x \diamondsuit w\}. \end{aligned} \tag{III.20}$$

denotes an *image* and *coimage*, respectively, of the set W under the relation $\diamondsuit$. An image of a set under $\diamondsuit$ coincides with its coimage under the inverse of $\diamondsuit$, i.e., $W\diamondsuit = \diamondsuit^{-1} W$. In the following, $\{w\}\diamondsuit$ and $\diamondsuit\{w\}$ will be simplified to $w\diamondsuit$ and $\diamondsuit w$, respectively. The following distributive laws are also useful:

$$W(\diamondsuit \cup \heartsuit) = W\diamondsuit \cup W\heartsuit, \qquad (\diamondsuit \cup \heartsuit)W = \diamondsuit W \cup \heartsuit W, \tag{III.21}$$
$$W(\diamondsuit \cap \heartsuit) = W\diamondsuit \cap W\heartsuit, \qquad (\diamondsuit \cap \heartsuit)W = \diamondsuit W \cap \heartsuit W, \tag{III.22}$$
$$W(\diamondsuit\heartsuit) = (W\diamondsuit)\heartsuit, \qquad (\diamondsuit\heartsuit)W = \diamondsuit(\heartsuit W). \tag{III.23}$$

III.3.1 Arrangement interval relations

I never was so ordered about before, in all my life, never!
[Lewis Carroll, *Alice's Adventures in Wonderland* (1865)]

Definition III.13 (*AIR*: Arrangement Interval Relation) *An* arrangement interval relation[3] *is a relation between intervals (i.e., a subset of* $\mathbb{IR} \times \mathbb{IR}$ *for real intervals) that can be defined by an expression using only the equality and the order relation defined in their base set applied to endpoints of the interval arguments, and logical connectives.*

There are 8191 *AIR*s, not taking into account the empty relation. Note also that some simple relations between intervals, like the *equal width* relation, see Section III.3.5, are not arrangement relations.

Remark. Due to certain complications with defining arrangement interval relations involving thin intervals, most of the contents of this section, unless otherwise specified, assumes that only thick intervals are taken into account. This is sufficient for most practical purposes in this area, see [Allen 1983, Freksa 1992], while simplifying considerably the exposition.

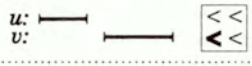
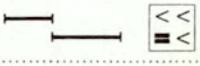
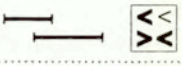
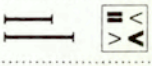
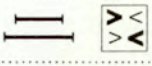
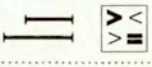
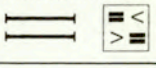
Definition III.14 (*BIR*: Basic Interval Relation) *The* basic interval relation *is an AIR that is minimal (within the set of AIRs) under the union of relations, i.e., is not a union of any other AIRs.*

There are 13 *BIR*s (again, not counting the empty relation). See Table III.4 for a listing and definitions of all *BIR*s. They are grouped in six pairs of mutually inverse relations and one self-inverse (symmetric) relation $= = =^{-1}$. New graphical symbols for the basic relations (proposed by this author in [12, 13]) are also shown there. The form of the symbols was chosen to conform with the graphical arrangement of the intervals that belong to the given relation—compare them with the diagrams in the second column of Table III.4. However, some of the symbols, as a result, violate the natural convention that symbols for mutually inverse relations are usually made to be mirror reflections in the vertical axis (as it holds for the first three pairs of relations, namely "$<$, $>$", "$\urcorner$, $\ulcorner$" and "$\sqcup$, $\sqcap$" in Table III.4). The remedy for this would be to rotate the symbols by 90 degrees, preferably anticlockwise. Unfortunately, this would in turn degrade the intuitive similarity of the relation symbols to corresponding arrangements of intervals, i.e., the very reason for this particular choice of symbols. However, the reflection with respect to horizontal axis does produce inverse relations for all new symbols (now only the pair "$<$" and "$>$" becomes an exception). Also, the proposed symbols better conform with the structure of the W-diagram below, hence their final adoption.

The relations are defined with classic one-dimensional interval diagrams and with the help of square *conjunction diagrams* denoting the conjunction of indicated relations between interval endpoints. Relations shown in boldface are required; the other follow from

[3] In the AI literature, the general term *interval relation* is commonly used to denote just this class of relations. However, because only a small subset of all possible interval relations is of this type, in a more general discussion a special term is needed to avoid possible confusion. For this reason, the term *arrangement relation* has been introduced by this author in [12, 13]. The acronym *AIR* may be also read as *Allen's Interval Relation*, after the author that originated their investigation in [Allen 1983].

Table III.4: Basic interval relations and the conjunction diagram.
Symbols within the conjunction diagrams denote relations between interval endpoints.

Basic interval relation		Symbols and names: here used	classic [Allen 1983]	
	$\begin{array}{\|cc\|}\hline < & < \\ > & < \\ \hline\end{array}$	$u < v$ $v > u$	before after	**<** **>**
	$\begin{array}{\|cc\|}\hline < & < \\ = & < \\ \hline\end{array}$	$u \llcorner v$ $v \lrcorner u$	meets met-by	**m** **mi**
	$\begin{array}{\|cc\|}\hline < & < \\ > & < \\ \hline\end{array}$	$u \sqcap v$ $v \sqcap u$	overlaps overlapped-by	**o** **oi**
	$\begin{array}{\|cc\|}\hline = & < \\ > & < \\ \hline\end{array}$	$u \sqsubset v$ $v \sqsupset u$	starts started-by	**s** **si**
	$\begin{array}{\|cc\|}\hline > & < \\ > & < \\ \hline\end{array}$	$u \sqsubset v$ $v \sqsupset u$	during contains	**d** **di**
	$\begin{array}{\|cc\|}\hline > & < \\ > & = \\ \hline\end{array}$	$u \sqsubset v$ $v \sqsupset u$	finishes finished-by	**f** **fi**
	$\begin{array}{\|cc\|}\hline = & < \\ > & = \\ \hline\end{array}$	$u = v$ $v = u$	equal	**=**

The conjunction diagram, $\begin{array}{\|cc\|}\hline \mathbf{a} & \mathbf{b} \\ \mathbf{c} & \mathbf{d} \\ \hline\end{array}$ a shortcut for: $\begin{array}{c\|cc\|}\hline & \underline{v} & \overline{v} \\ \hline \underline{u} & \mathbf{a} & \mathbf{b} \\ \overline{u} & \mathbf{c} & \mathbf{d} \\ \hline\end{array}$ depicts the formula: $\underline{u}\,\mathbf{a}\,\underline{v}$ **and** $\underline{u}\,\mathbf{b}\,\overline{v}$ **and** $\overline{u}\,\mathbf{c}\,\underline{v}$ **and** $\overline{u}\,\mathbf{d}\,\overline{v}$

bold – required; normal – follows from others; *empty* – no restriction

the interval definition (for thick intervals assumed here—from the inequality $\underline{u} < \overline{u}$) and transitivity of the ordering relation. Contrary to the initial claim by [Freksa 1992, p. 202] that all basic relations can be defined by at most two relations between the interval endpoints, two of them (the overlap relations) require the specification of at least three such relations.

All *AIR*s are unions of *BIR*s, and the set *AIR* is closed under union, intersection and composition of relations. There are some other useful subclasses of *AIR*s, wider than the *BIR* class. Two most important of them are:

CIR—**Convex Interval Relations:** (82 relations), important in applications [van Beek & Cohen 1990, Vilain et al. 1990, Nökel 1991, Freksa 1992, Bettini 1994], and discussed in more details in Section III.3.3.

PIR—**Pointisable Interval Relations:** (187 relations), a wider class than *CIR*, sharing with it some useful properties [van Beek & Cohen 1990, Bettini 1994], and discussed in Section III.3.4.

The sharp inclusions $BIR \subset CIR \subset PIR \subset AIR$ hold for these classes.

III.3.2 The W-diagram and L-diagram

Representing the images (or coimages) of basic interval relations in the MR-diagram produces the *W-diagram*, see Fig. III.11. The diagram is very useful in investigations of the properties of *AIRs*. The lines and regions in the *image W-diagram* (Fig. III.11a) constitute the images of an arbitrary thick interval u under all 13 basic interval relations. The images are labelled by graphical symbols of corresponding relations. As $u\Diamond = \Diamond^{-1}u$, the coimage diagram (Fig. III.11b) is essentially the same as the image diagram, only with all relations changed to their inverses (inverse relations are placed in the diagram symmetrically with respect to the "=" relation). An important feature of the diagram, responsible for its usefulness, is that the structure and shape of its elements (images and coimages of some thick interval) do not depend on the choice of the interval u, hence most of the properties of these images and coimages can be also safely considered as appropriate properties of the corresponding relations.

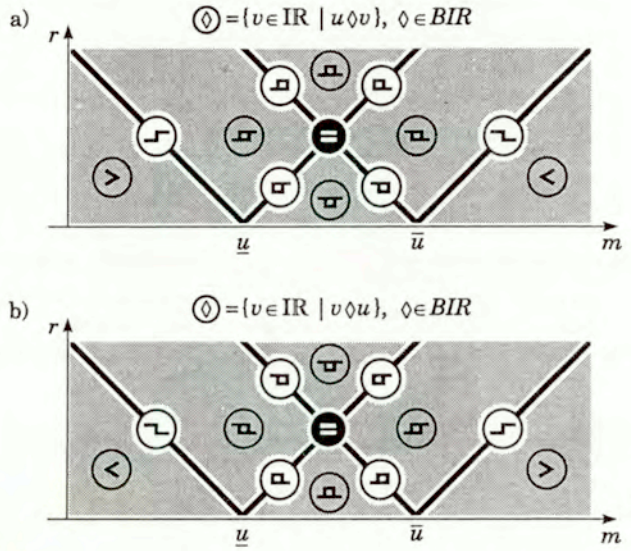


Figure III.11: The image (a) and coimage (b) W-diagrams of basic interval relations.

As can be easily seen from the diagram, all *BIRs* are disjoint and cover the whole space of intervals, i.e., every pair of intervals belongs to exactly one of these relations.[4] The diagrams show also that the basic relations fall into three distinct classes according to the dimensionality of their images: 0-dimensional (a point; the "=" relation only), 1-dimensional (lines), and 2-dimensional (regions). The circular labels of the images in Fig. III.11

[4]Strictly speaking, this holds exactly only when one excludes the thin intervals coinciding with endpoints $\underline{u}$ and $\overline{u}$ of the interval u. They belong to images (or coimages) of two relations, as can be seen in the diagrams. The exception, minor as it is, can be eliminated either by introducing additional basic relations, or by restricting the analysis to thick intervals only. The latter approach is taken here, see the remark at the beginning of Section III.3.1.

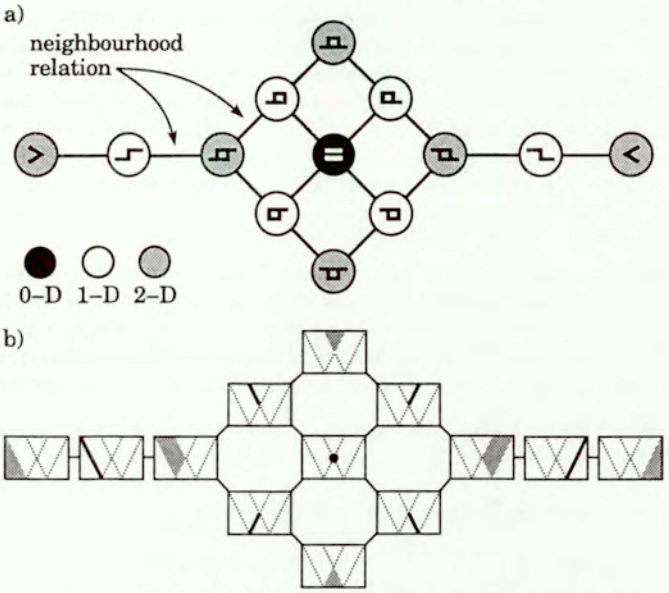


Figure III.12: The lattice diagrams of basic interval relations L_{BIR}, using relation symbols (a), and W-diagram based icons (b).

are colour-coded according to the dimensionality of the appropriate relation, see also Fig. III.12. Not surprisingly, the dimensionality corresponds (inversely) to the number of the equality conditions in the conjunction of terms relating the interval endpoints in the definition of the relation, see Table III.4. Note also that all images of *BIR*s are open, i.e. regions do not contain their borders and lines their endpoints.[5]

Definition III.15 (Neighbour interval relations) *Two basic interval relations are considered neighbours if their images on the W-diagram are* diagonally adjacent, *i.e., if there exists a pair of intervals placed along a diagonal line,*[6] *one belonging to the image of the first relation, the other to the image of the second relation, such that the diagonal line joining these intervals is fully contained within the union of images of these two relations.*

Linking symbols of two basic relations with an edge when they are neighbours produces a graph[7] (see Fig. III.12a) which, when turned into a vertical position (with the right-hand node of the diagram at the top), can be considered as a *Hasse diagram* of the lattice of basic relations L_{BIR} [Nökel 1991]. In [Freksa 1992], simplified diagrams of this kind

[5]After restricting the analysis to thick intervals, see the previous footnote.

[6]To be more orthodox, we should use here lozenges (cf. Definition III.12) instead of lines, but lines suffice here and allow for simpler formulation.

[7]Called the *A-neighbourhood graph* in [Freksa 1992], where it was defined with non-diagrammatic means.

were used as iconic symbols for *AIRs* (see also examples in Fig. III.13 below). As in Fig. III.11, the nodes of the lattice graph in Fig. III.12a are colour-coded according to the dimensionality of the corresponding relations. Note that neighbour nodes differ in dimensionality by exactly one. The structure of the lattice mirrors closely the structure of the W-diagram (compare Fig. III.11a and Fig. III.12a). An isomorphic dual lattice (with the lattice diagram put upside-down) corresponds to the coimage W-diagram. The nodes of the lattice can be also labelled with the W-diagrams of corresponding relations, as shown in Fig. III.12b. To conserve space, these small W-diagrams were narrowed horizontally a little.

As *AIRs* are unions of *BIRs*, they can be represented on the W-diagram as unions of the images (or coimages) of their constituent relations, on the basis of the distributive law (III.21). Small simplified versions of such W-diagrams can be used as convenient icons to denote various *AIRs*, like the icons used to represent basic interval relations in Fig. III.12b. By convention, *image W-diagrams* are used here as a basis for these icons.

Example III.5 (Some *AIRs*) In Fig. III.13, four example *AIRs* are specified using several different representations for comparison: descriptive names (based on the idea proposed by [Freksa 1992]), W-diagram icons, conjunction diagrams, and icons based on the lattice diagram (similar to those used in [Freksa 1992]). ■

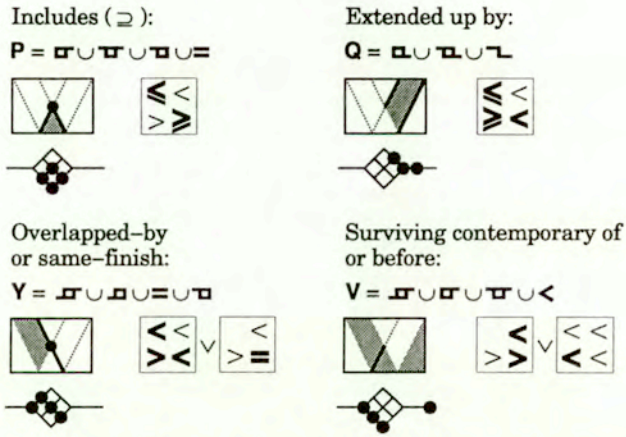


Figure III.13: Four example *AIRs* **P**, **Q**, **Y**, and **V**, represented in various ways (see text).

It is easy to see that the ordering relations discussed in Section III.2.3.3 are also *AIRs*. Comparing Fig. III.9 with Fig. III.11b (note that the relations in Fig. III.9 are represented by their coimages) it is easy to list all the ordering relations as unions of *BIRs*:

$$< \;=\; >^{-1} \;=\; b \cup m \cup o,$$
$$\leq \;=\; \geq^{-1} \;=\; b \cup m \cup o \cup = \quad \text{(see also Fig. III.46)},$$
$$> \;=\; <^{-1} \;=\; e \cup f \cup d,$$
$$\geq \;=\; \leq^{-1} \;=\; e \cup f \cup d \cup = \;=\; \mathbf{P} \text{ (see Figs. III.13 and III.46)};$$

$$\prec = \succ^{-1} = <\cup\frown\cup\sqsubset\cup\sqsupset\cup\sqsubset,$$
$$\preccurlyeq = \succcurlyeq^{-1} = <\cup\frown\cup\sqsubset\cup\sqsupset\cup\sqsubset\cup=,$$
$$\succ = \prec^{-1} = \sqsubset\cup\sqsupset\cup\sqsupset\cup\smile\cup>,$$
$$\succcurlyeq = \preccurlyeq^{-1} = \sqsubset\cup\sqsupset\cup\sqsupset\cup\smile\cup>\cup=.$$

III.3.3 Convex interval relations

The concept of *convexity* is important in many respects and has a lot of uses, both in theoretical considerations and practical applications. It is also true for interval algebra. Once the in-between relation is defined, the concept of a convex interval set follows naturally.

III.3.3.1 Convexity of interval sets and relations

Definition III.16 (Convex interval sets: algebraic) *The set S of intervals is called* convex *when for every pair of intervals $u, v \in S$, also $w \in S$ for every interval w lying* between *u and v.*

It is easy to formulate the definition in diagrammatic terms also, cf. Definitions III.11 and III.12.

Definition III.17 (Convex interval sets: diagrammatic) *The set S of intervals is called* convex *when any lozenge with opposite corners in S is fully contained in S.*

It is easy to show that intersection of convex sets is also convex, while the property does not hold for the set union.

Example III.6 (Non-convex interval sets) As lozenges used to define convex interval sets are different objects than line segments used for that purpose in ordinary (linear) spaces, shapes of convex (and non-convex) interval sets differ from those in the more ordinary spaces. In Fig. III.14, two examples of some *non-convex* interval sets are shown. Note that under the ordinary definition of convexity using line segments, both these sets would be convex. ■

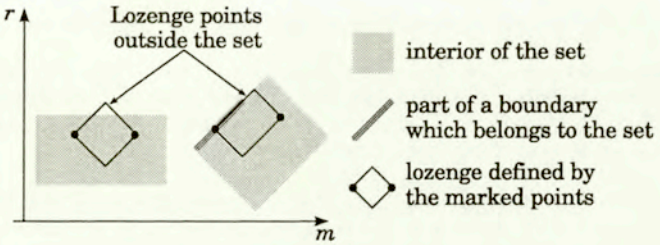


Figure III.14: Examples of two non-convex sets of intervals.

When looking at the examples in Fig. III.13, one may notice that the relations **Y** and **V** have a different "look" than **P** and **Q**: the former seem more complicated, require disjunctive logical formulae to define, and their W-diagrams and lattice diagrams look

differently too—one may say, they look less compact. The observation is indeed significant: the **P** and **Q** relations belong to the important subclass of *convex relations*, whereas the relations **Y** and **V** are not convex.

The convexity of arrangement interval relations can be then defined in a simple way:

Definition III.18 (Convex interval relations) *The arrangement interval relation is called* convex *when an image (and coimage) of any single interval under this relation is a convex interval set.*

The importance of convex interval relations (*CIR*s) comes from at least two facts:

- They are common and natural in everyday reasoning involving intervals, especially reasoning about time and space [Mukerjee & Joe 1990, van Beek & Cohen 1990, Nökel 1991, Freksa 1992, Hernández 1994].
- Algorithms for solving several important problems involving networks of constraints between intervals are tractable (of polynomial complexity) within the *CIR* class[8] whereas these problems often become NP-complete when other relations are also allowed [Allen 1983, van Beek & Cohen 1990, Vilain et al. 1990, Nökel 1991].

III.3.3.2 The convex relations characterization theorem

There are several ways to characterise compactly the class of convex interval relations. Three of them are the most common; the fourth one, involving the W-diagram, should be also added to this list. All of these characterisations are equivalent, as stated by the following theorem. The usefulness of the diagrammatic tools developed in this chapter can be shown with the help of the diagram-aided proof of this theorem.

Theorem III.3 *For arrangement interval relations, all the definitions (listed below) of convex interval relations are equivalent. An interval relation $\Diamond$ is in the class CIR if and only if:*

[P_C] Primary characterization. *For any interval i and every pair of intervals u, v such that $i \Diamond u$ and $i \Diamond v$, also $i \Diamond w$ for every interval w lying between u and v.*

[T_C] Term characterization. *It can be defined as a conjunction of simple terms involving only relations in the set $\{\leq, \geq, <, >, =\}$ defined in the base set between interval endpoints, (i.e., the order relation, its inverse and their negations, plus equality)* [van Beek & Cohen 1990].

[W_C] W-diagram characterization. *Its image (or coimage) in the W-diagram includes together with any two intervals also the lozenge defined by these intervals.*

[L_C] Lattice characterization. *It is a union of all relations belonging to some interval (including thin intervals) over a lattice L_{BIR} of basic interval relations* [Nökel 1991].

[8] This holds also for some other subclasses of *AIR*s, especially the class *PIR* of pointisable relations, see Section III.3.4 and [Drakengren & Jonsson 1998].

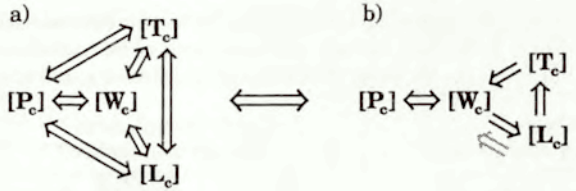
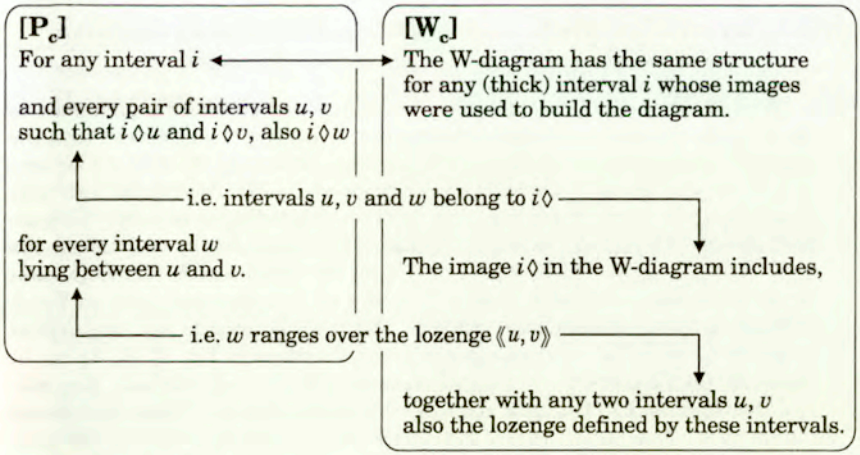


Figure III.15: The proof plan for Theorem III.3: equivalences to be proven (a), and an equivalent set of implications actually proven (b).

Proof. To prove the theorem, in principle six equivalences should be proven, as shown in Fig. III.15a. Actually, as the equivalence amounts to two implications going in opposite directions and the implication is transitive, it suffices to prove only a subset of the 12 implications shown in the figure. The single equivalence and three implications shown in black in Fig. III.15b will be proven here. Additionally, also the implication shown in gray in the figure will be explicitly demonstrated (though it is not necessary to complete the proof), as it easily follows as a byproduct of the proof for the $[\mathbf{L_c}] \Rightarrow [\mathbf{T_c}]$ implication.

In the sequel, calling a relation [C]-*convex* means that it fulfills the characterization [C]. As the interval relations addressed by this theorem are arrangement interval relations (*AIRs*), images of all relations considered here are obtainable as unions of regions depicted on the W-diagram.

$[\mathbf{P_c}] \Leftrightarrow [\mathbf{W_c}]$ equivalence. This part is the easiest, because the W-diagram characterization is actually a diagrammatic paraphrase of the primary characterization (and vice versa), as the schema below shows.



Additionally, as the coimage W-diagram has the same structure as the image W-diagram (only with inverse relations interchanged), we see that the $[\mathbf{P_c}]$ characterization can be formulated equivalently as:

[$\mathbf{P_C^{-1}}$] Primary characterization, inverse form. An interval relation $\Diamond$ is in *CIR* if and only if for any interval i and every pair of intervals u, v such that $u \Diamond i$ and $v \Diamond i$, also $w \Diamond i$ for every interval w lying between u and v.

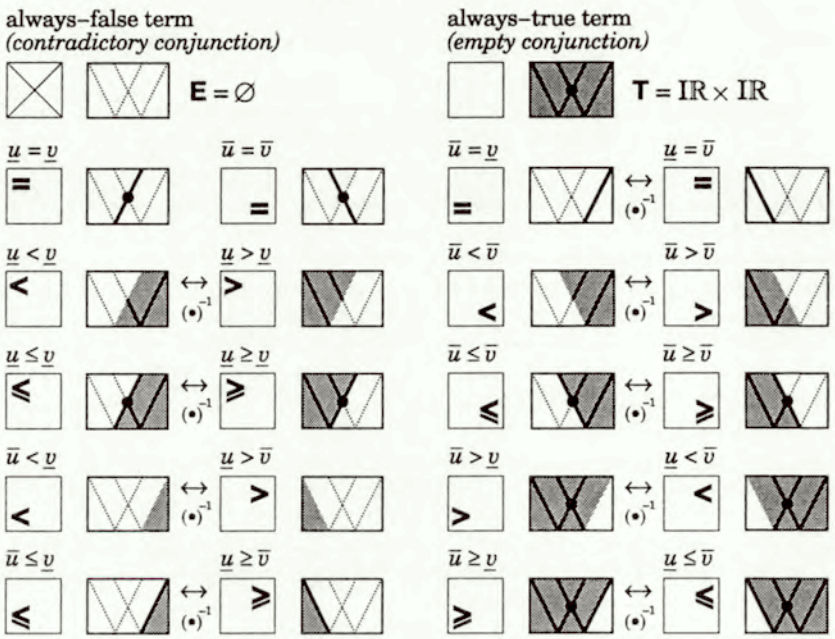


Figure III.16: The interval relations E and T defined by degenerate conjunctions, and relations defined by all 20 one-element conjunctions.

[$\mathbf{T_C}$] $\Rightarrow$ [$\mathbf{W_C}$] implication. Consider a simplest conjunction of the form prescribed by the [$\mathbf{T_C}$] characterization, namely the one containing only a single basic term (involving a single comparison using a relation from the set $\{\leq, \geq, <, >, =\}$ between one of the four possible combinations of endpoints of the two related intervals). Two *degenerate terms*: the *always true* term (corresponding to an empty conjunction) and the *always false* term (corresponding to a contradictory conjunction, like $\underline{u} = \overline{v}$ **and** $\underline{u} > \overline{v}$) should be added to this set for completeness. The degenerate terms correspond to the total relation $\mathsf{T} = \mathbb{IR} \times \mathbb{IR}$, and the empty relation $\mathsf{E} = \emptyset$, respectively (both are obviously convex). All these degenerate and *one-element* conjunctions with their corresponding relations are shown in Fig. III.16. As can be seen from the diagrams there, they all represent [$\mathbf{W_C}$]-convex relations. Any relation characterized by [$\mathbf{T_C}$] is defined by a conjunction of some of these one-element conjunctions. As a conjunction of logical definitions of relations corresponds to an intersection of their images, and intersection of convex sets is also convex, then every conjunction described by [$\mathbf{T_C}$] represents a [$\mathbf{W_C}$]-convex relation as well.

Note in Fig. III.16 that the degenerate conjunctions and two of the one-element conjunctions are self-inverses, while the remaining relations divide into nine pairs of

mutually inverse relations. If a relation is convex (or not), so is its inverse, therefore it would suffice in the proof to consider only one relation from every pair of inverses (as it was actually done in [13]).

[Lc] ⇒ [Tc] and [Lc] ⇒ [Wc] implications. The proof of the **[Lc]⇒[Wc]** implication is not really necessary. However, it easily follows from the reasoning leading to the **[Lc] ⇒ [Tc]** implication, hence it is mentioned here.

Note first that as the L_{BIR} lattice defines partial order in a set, it can be used to define intervals (called also *quotient sublattices*) over the set *BIR*. As shown in Fig. III.17a, the convex relations **P** and **Q** are unions of relations belonging to two such intervals. Moreover, any interval $l = [\underline{l}, \overline{l}]$ in L_{BIR} can be obtained as an intersection of two other intervals: one beginning at the lowest element in the lattice (the "<" relation in Fig. III.17a) and ending at $\overline{l}$, and the other beginning at $\underline{l}$ and ending at the uppermost element of the lattice (the ">" relation in Fig. III.17a). Such "lower" and "upper" intervals in the lattice are called *ideals* and *filters* in the lattice theory terminology. Every element of a lattice defines its ideal and filter, and the sets of ideals and filters constitute their own lattices, isomorphic with the original one, see Fig. III.17b and c, respectively. The W-diagram icons for the ideal and filter relations are obtained as unions of the diagrams for basic relations (Fig. III.17a) included in the given ideal or filter. The filter and ideal relations were also endowed with their conjunction diagrams in the figure. As can be seen, all of them require only single conjunction diagrams, that is, can be defined by a single conjunction of basic terms (see the proof of the previous implication). Note also that every relation corresponding to an ideal (respectively filter) is an inverse of a relation corresponding to a filter (respectively ideal) that is placed symmetrically to it in the other lattice, compare Figs. III.17b and c.

As thus follows from these constructions, an interval relation defined as a union of relations included in some interval l in L_{BIR} is equal to the intersection of two interval relations corresponding to the ideal and filter whose intersection produces the interval l. The construction is shown in Fig. III.17d for the example relations **P** and **Q** (see also Fig. III.13). Because an intersection of relations corresponds to conjunction of their logical definitions, the definition of the resulting relation is also a conjunction of basic terms (recall that ideal and filter relations are defined by single conjunctions). Thus, every interval in L_{BIR} defines an interval relation definable by some conjunction of basic terms, proving the **[Lc] ⇒ [Tc]** implication.

As every relation defined by those ideals and filters is also **[Wc]**-convex, the implication **[Lc] ⇒ [Wc]** also follows immediately.

[Wc] ⇒ [Lc] implication. This implication will be proven by contradiction: it will be shown that **not [Lc] ⇒ not [Wc]**.

Let us first see which subsets of the lattice L_{BIR} are *not* intervals in the lattice. A subset of a lattice is an interval in it (a *quotient sublattice* in the lattice theory parlance) if it contains all elements between its endpoints (hence, it must be connected), and with any two elements m, n it must also contain the elements $m \vee n$ (join) and $m \wedge n$ (meet). Thus, we have two types of non-interval subsets:

Disconnected subsets. As the L_{BIR} lattice is based on the neighbourhood relation between elements of the W-diagram, Fig. III.12, a disconnected subset of

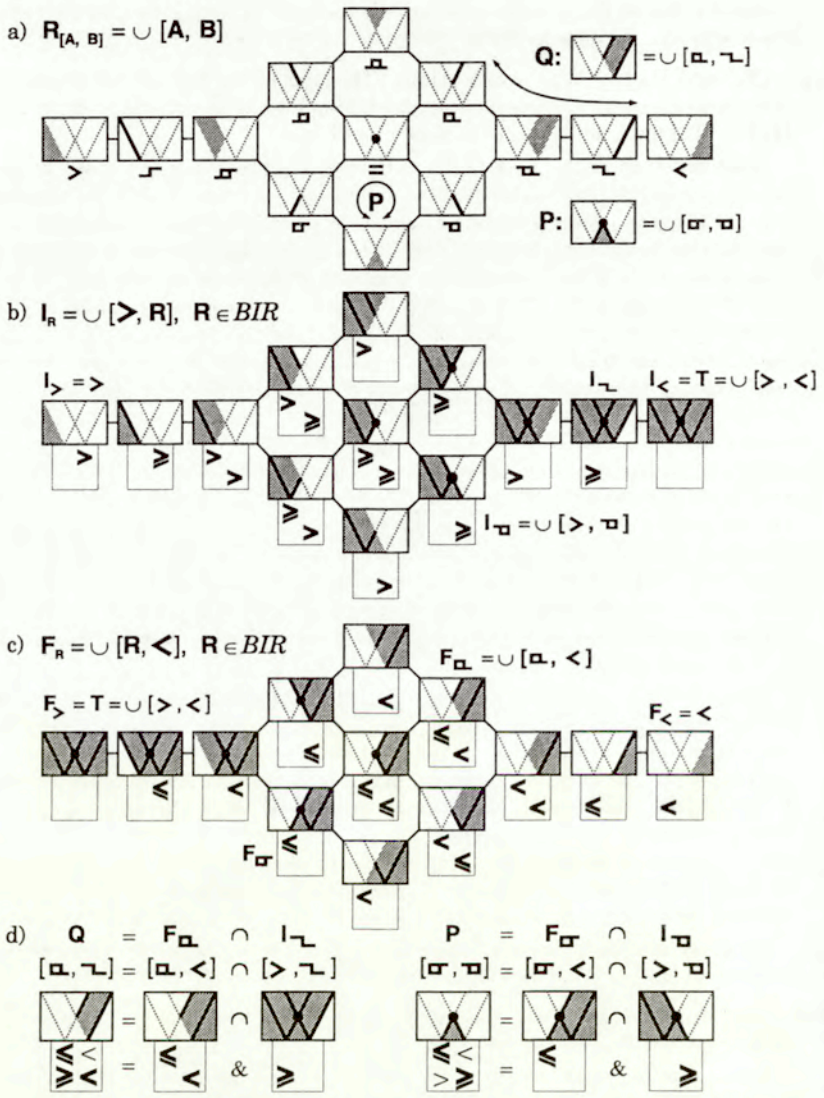


Figure III.17: The L_{BIR} lattice with two example relations P and Q represented as intervals in the lattice (a), lattices of filters and ideals of L_{BIR} (b, c), and construction of relations P and Q as intersections of appropriate ideals and filters (d).

L_{BIR} corresponds to a disconnected set of elements of the W-diagram. See Fig. III.18 for two examples of such configurations. Yet, a relation represented by disconnected set on the W-diagram is necessarily not $[\mathbf{W_c}]$-convex. To show this, consider choosing two intervals belonging to different disconnected components of such a set. The lozenge defined by these intervals, being connected, must contain some intervals outside the set, making it not $[\mathbf{W_c}]$-convex.

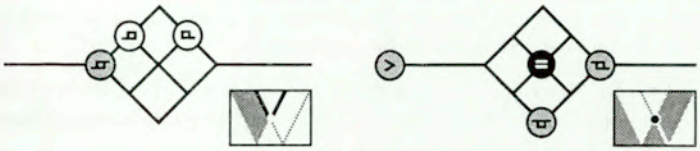


Figure III.18: Two examples of disconnected configurations on the L_{BIR} lattice and their W-diagrams.

Connected subsets with some necessary element missing. As it is obvious from the diagram of the L_{BIR} lattice, every connected subset of the L_{BIR} lattice that is not an interval must include at least one of the three basic types of three-node configurations, see Fig. III.19 (where the W-diagrams for selected configurations of each type are also shown):

(a) Missing central 0-D node (in the W-diagram: missing corner point between two border lines); four cases.

(b) Missing 1-D node in the neighbourhood of the centre (in the W-diagram: missing border line); eight cases.

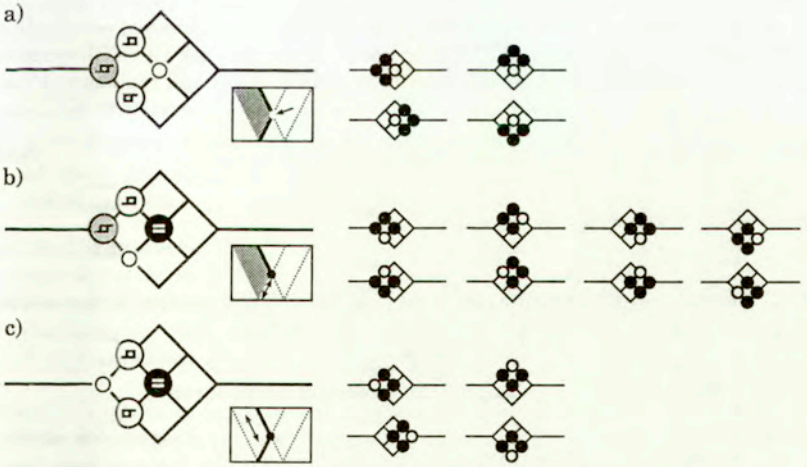


Figure III.19: Basic non-convex connected configurations on the L_{BIR} lattice and their W-diagrams.

(c) Missing 2-D node in the corner of the centre diamond configuration (in the W-diagram: non-aligned lines connected by a point); four cases.

Arrows in Fig. III.19 mark the missing elements on the W-diagram. It is easy to show that all three configurations are not $[\mathbf{W_C}]$-convex. By choosing appropriate pairs of intervals belonging to the configuration—namely, in the cases (a) and (c) one on the upper and another on the lower edge, while in the case (b) one at the (central) corner and another inside the 2-D region—we see that the lozenges defined by them necessarily contain some intervals belonging to the missing component of the configuration.

As in both possible cases discussed above from the non-$[\mathbf{L_C}]$-convexity of the configuration follows its non-$[\mathbf{W_C}]$-convexity, the $[\mathbf{W_C}] \Rightarrow [\mathbf{L_C}]$ implication is thus proven.

That also concludes the whole proof of the theorem. □

III.3.4 Pointisable interval relations

The *pointisable interval relations* (*PIRs*) constitute a wider subset of arrangement interval relations than convex relations (and there is more than twice as many *PIRs* as *CIRs*, see Section III.3.1), but they have the same nice properties as the latter concerning tractability of algorithms for solving networks of constraints on intervals. Therefore, although from the practical point of view *PIRs* are not so common as *CIRs*, they constitute an important class of *AIRs*, hence the problem of finding simple characterizations of relations from this class becomes also of interest.

III.3.4.1 Full-line relations

For further considerations, certain four interval relations $\{F_i\}_{i=1}^4$ are of special importance. They are here called the *full-line relations*, as their images (or coimages) constitute the longest possible straight lines in the W-diagram. As such, they are all one-dimensional; they are convex as well. Representations of these four relations and their complements (which are not convex) are shown in Fig. III.20. The set F of these relations consists of:

$$F = \{\text{⌜}, \text{⌟}, \text{⌅}, \text{⌆}\}, \tag{III.24}$$

where:

$$\text{⌅} = (\text{⌜} \cup \text{=} \cup \text{⌟}),$$
$$\text{⌆} = (\text{⌜} \cup \text{=} \cup \text{⌟}).$$

The full-line relations will appear again in Section III.5.2.1 as important *border relations* defining boundaries of solution sets of linear interval equations.

III.3.4.2 The pointisable relations characterization theorem

Like with convex relations, there can be several characterizations of pointisable relations. The *term characterization* included below is commonly used in the literature [van Beek & Cohen 1990]; the other three below were first defined in [9] by this author. The equivalence of these four characterizations is stated by the following theorem.

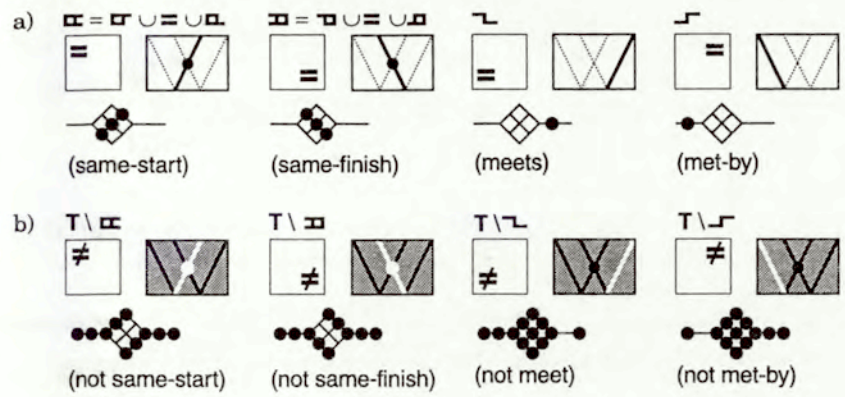


Figure III.20: The set F of four full-line relations (a) and their complements (b).

Theorem III.4 *For arrangement interval relations, all the definitions (listed below) of pointisable interval relations are equivalent. An interval relation $\Diamond$ is in the class PIR if and only if:*

[T_p] Term characterization. *It can be defined as a conjunction of simple terms involving only relations in the set $\{\leq, \geq, <, >, =, \neq\}$ defined in the base set between interval endpoints, (i.e., the order relation, its inverse, equality, and their negations)* [van Beek & Cohen 1990].

[W_{p1}] W-diagram characterization 1. *For every two intervals u, v belonging to the image (or coimage) of the relation $\Diamond$, the set difference between the lozenge $\langle\!\langle u, v \rangle\!\rangle$ defined by these intervals and the image (respectively coimage) of $\Diamond$ belongs to the image (respectively coimage) of the union $\cup f$ of some subset $f \subseteq F$ (possibly empty) of the set of four full-line relations of Fig. III.20, such that the image (respectively coimage) of $\cup f$ and the image (respectively coimage) of $\Diamond$ are disjoint.*

[W_{p2}] W-diagram characterization 2. *An image (or coimage) of $\Diamond$ is obtained from an image (respectively coimage) of some convex relation C by deleting from it a subset (possibly empty) of the set of four full-line relations, see Fig. III.20.*

[L_p] Lattice characterization. *It is a union of all relations belonging to some interval over one of the* 16 *lattices L_i, see Fig. III.23, obtained from the lattice L_{BIR} of basic interval relations by deleting from it all the nodes corresponding to any subset f (including an empty subset, when we obtain simply the lattice L_{BIR} itself) of the set F of four full-line relations, see Fig. III.20a.*

Proof. Similarly as for Theorem III.3, the proof plan for this theorem is shown in Fig. III.21. Five implications will be explicitly proven here. Several fragments of the proof proceed exactly as in the proof of Theorem III.3. In these cases, instead of repeating the whole argument, a reference to the appropriate fragment of the proof of Theorem III.3 will be given. Also, the reasoning will be conducted for images of the relations only, being

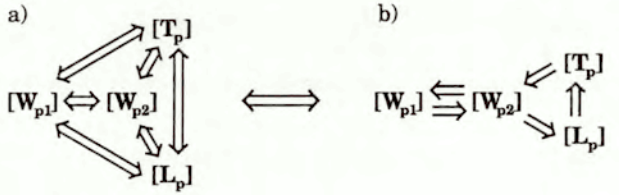


Figure III.21: The proof plan for Theorem III.4: equivalences to be proven (a) and the equivalent set of implications actually proven (b).

equally valid for coimages (after simply replacing the word "image" by "coimage" thorough the text).

$[W_{p1}] \Rightarrow [W_{p2}]$ implication. For all possible pairs of intervals belonging to the image of $\Diamond$ we can determine exactly which full line relations $F_i \in F$ (of the four shown in Fig. III.20) satisfy the characterization $[W_{p1}]$. The image of $\Diamond$ in the W-diagram is a union of some basic regions, and the full line images of F_i relations that do not belong to the image of $\Diamond$ are then the only obstacle preventing the image of $\Diamond$ from fulfilling the condition $[W_C]$ for convexity of the relation $\Diamond$. Hence, the image of $\Diamond$ with these full line relations (or their constituent relations) added creates an image of some convex relation C needed in the characterization $[W_{p2}]$.

$[W_{p2}] \Rightarrow [W_{p1}]$ implication. Obviously, $[W_{p1}]$ holds for the trivial case, when the set difference between the lozenge $\langle\!\langle u, v \rangle\!\rangle$ (defined by some two intervals u, v belonging to the image of $\Diamond$) and the image of $\Diamond$ is empty. Otherwise, for all the rest of such pairs, if $[W_{p2}]$ holds, then the image of $\Diamond$ is included in the image of C. So, if the image of $\Diamond$ contains some pair of intervals then so does the image of C. Hence the image of C includes the lozenge defined by these intervals, because C is convex (according to the $[W_C]$ condition of Theorem III.3). Thus, the set difference between that lozenge and the image of $\Diamond$ must be a subset of an image of $\cup f$ (i.e., of a union of the subset of full-line relations mentioned in the characterization). Furthermore, these images of full-line relations cannot belong to the image of $\Diamond$, as follows directly from $[W_{p2}]$, so $[W_{p1}]$ must hold as well.

$[T_p] \Rightarrow [W_{p2}]$ implication. This implication is proven in the same way as the $[T_C] \Rightarrow [W_C]$ implication in Theorem III.3, only the set of allowable relations that can occur in conjunctions is extended by adding the inequality $\neq$. As a result, an additional row consisting of four one-element conjunctions (and associated relations) should be added to Fig. III.16, namely the complements of full-line relations presented in Fig. III.22.

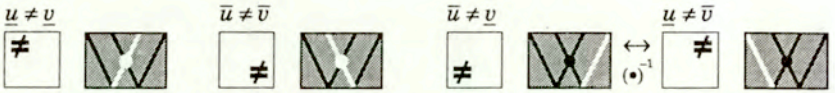


Figure III.22: The four relations defined by one-element conjunctions involving inequality.

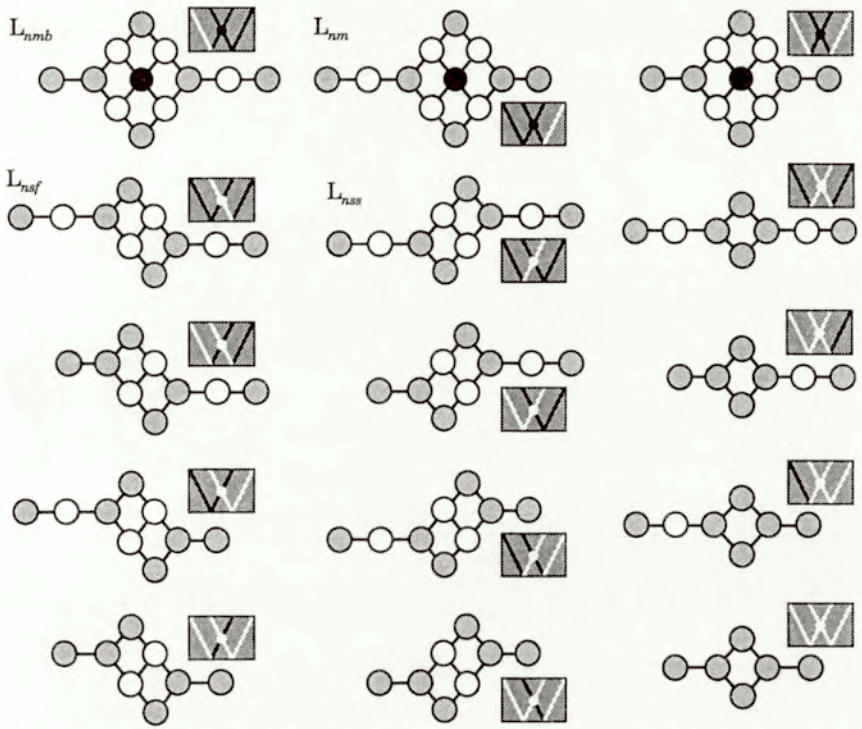


Figure III.23: The 15 sublattices L_i of the lattice L_{BIR} obtained by deleting from L_{BIR} the nodes corresponding to all nonempty subsets of the set of four full-line relations from Fig. III.20. The W-diagrams show complements of the deleted full-line relations.

Now it suffices to observe additionally that, as can be seen from the W-diagrams in Figs. III.16 and III.22, all one-element conjunction relations fulfill the condition $[\mathbf{W_{p2}}]$. As a conjunction of logical definitions of relations corresponds to the intersection of their images, intersections of convex relations are convex, and intersections of subsets of the set of full-line relations are also subsets of the same set, the conjunctions of relations in Figs. III.16 and III.22 fulfill the condition $[\mathbf{W_{p2}}]$ as well.

$[\mathbf{W_{p2}}] \Rightarrow [\mathbf{L_p}]$ implication. The convex relation C occurring in the formulation of the characterization $[\mathbf{W_{p2}}]$ is, according to the characterization $[\mathbf{L_c}]$ of Theorem III.3, a union of relations belonging to some interval over the lattice L_{BIR}. However, a pointisable relation can have, according to $[\mathbf{W_{p2}}]$, some of the constituent relations missing, namely one or more of the full-line relations F_i, see Fig. III.20. This can be accomplished easily by taking the defining interval from a sublattice of L_{BIR} with just those full-line relations missing, i.e., from one of the L_i sublattices, see Fig. III.23. Combined with the argument concerning the construction of the convex

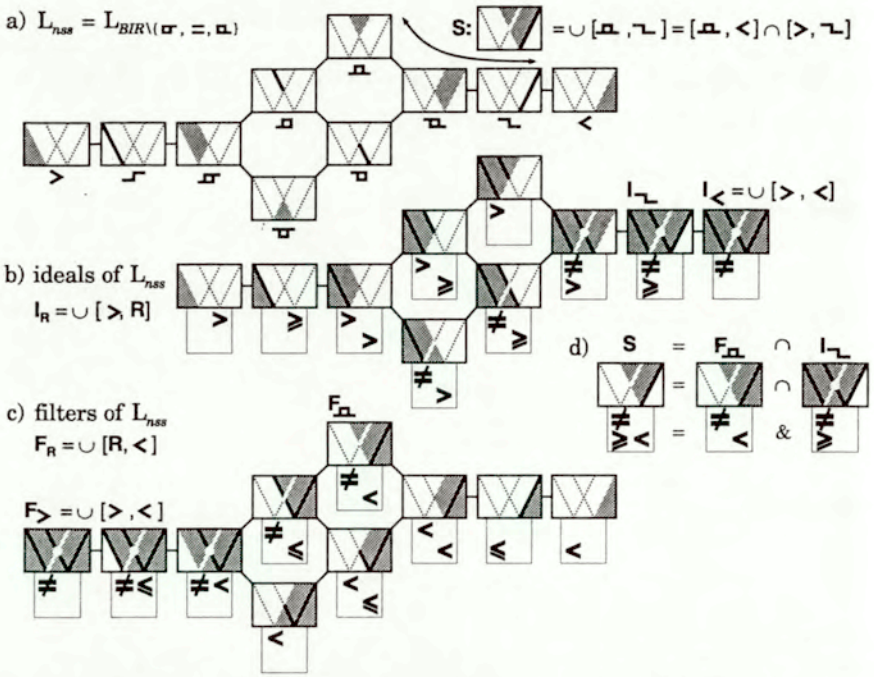


Figure III.24: The L_{nss} sublattice with an example relation S represented as an interval in the lattice (a), lattices of filters and ideals of L_{nss} (b, c), and construction of the relation S as an intersection of the appropriate ideal and filter (d).

relation C in the proof of the $[\mathbf{L_C}] \Rightarrow [\mathbf{W_C}]$ implication in Theorem III.3 it proves this implication as well.

$[\mathbf{L_P}] \Rightarrow [\mathbf{T_P}]$ implication. This part goes in the same way as the $[\mathbf{L_C}] \Rightarrow [\mathbf{T_C}]$ part of the proof of Theorem III.3, only this time the reasoning must be conducted separately for every of the 16 sublattices of L_{BIR} (those 15 from Fig. III.23 and L_{BIR} itself). Here we will show how the argument will work for the L_{nss} lattice obtained by deleting nodes constituting the $\sqsubset$ relation from the L_{BIR} lattice, see Fig. III.24. Note that filter and ideal relations from which that full-line relation was deleted contain in their conjunction diagrams the inequality at the place where it occurs in the conjunction diagram of the complement of the deleted relation $\sqsubset$, cf. Fig. III.20. The rest of the argument goes exactly as in Theorem III.3.

That also concludes the whole proof of the theorem. □

III.3.5 Non-arrangement interval relations

Some other interval relations which are not *AIR*s are also important. Figure III.25 shows images and coimages of a number of such relations, generated by dividing the MR-diagram with a *constant position* (*midpoint*) line, a *constant radius* line, and an *interval axis* for a given interval u. For the sake of easy reference, tentative graphical symbols for these relations are introduced here. The relations are defined as:

$$\begin{aligned}
\bullet| &= |\bullet^{-1} = \{(u,v) \in \mathbb{IR} \times \mathbb{IR} \mid \check{u} < \check{v}\},\\
\updownarrow &= \updownarrow^{-1} = \{(u,v) \in \mathbb{IR} \times \mathbb{IR} \mid \check{u} = \check{v}\} \text{ (the same position)},\\
|\bullet &= \bullet|^{-1} = \{(u,v) \in \mathbb{IR} \times \mathbb{IR} \mid \check{u} > \check{v}\},\\
\overline{\bullet} &= \underline{\bullet}^{-1} = \{(u,v) \in \mathbb{IR} \times \mathbb{IR} \mid \hat{u} < \hat{v}\},\\
\bullet\!\bullet &= \bullet\!\bullet^{-1} = \{(u,v) \in \mathbb{IR} \times \mathbb{IR} \mid \hat{u} = \hat{v}\} \text{ (the same radius)},\\
\underline{\bullet} &= \overline{\bullet}^{-1} = \{(u,v) \in \mathbb{IR} \times \mathbb{IR} \mid \hat{u} > \hat{v}\},\\
\diagup &= \diagdown^{-1} = \{(u,v) \in \mathbb{IR} \times \mathbb{IR} \mid |\mathbf{rex}\, u| < |\mathbf{rex}\, v|\},\\
\times &= \times^{-1} = \{(u,v) \in \mathbb{IR} \times \mathbb{IR} \mid |\mathbf{rex}\, u| = |\mathbf{rex}\, v|\} \text{ (the same relative extent)},\\
\diagdown &= \diagup^{-1} = \{(u,v) \in \mathbb{IR} \times \mathbb{IR} \mid |\mathbf{rex}\, u| > |\mathbf{rex}\, v|\}.
\end{aligned}$$

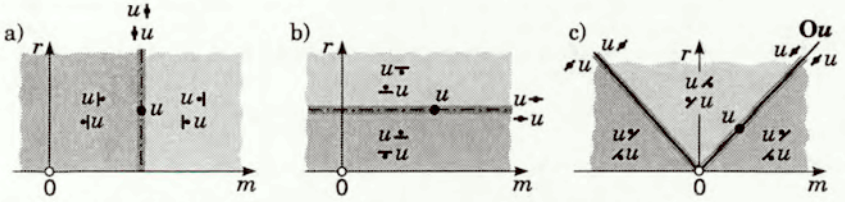


Figure III.25: Some important non-arrangement interval relations: relating interval midpoints (a), radii (b), and extents (c).

From the comparison of Fig. III.6 with Figs. III.11 and III.25 one can see that interval classes defined in Section III.2.3 can be easily defined as images (or coimages) of the interval $0 = [0,0]$ under appropriate interval basic relations and non-arrangement relations defined in this section. This assigns a rigorous meaning to the sentence at the beginning of Section III.2.3. The appropriate images and coimages are listed in Table III.5.

Note that for a thin interval, the central (=) point lies on the **Om** axis, hence the W-diagram for this case should be rather named a V-diagram, with some images of *BIR*s depicted in a W-diagram missing and other coinciding (imagine moving down the central (=) point in Fig. III.11 until it falls at the **Om** axis). For example, images of the interval 0 under relations ⊐ and ⌐, (as well as ⊏ and ¬) coincide. As a result, some of the interval types can be defined as images (coimages) of 0 in several ways using different relations, as shown in the table.

Table III.5: Interval types defined as images and coimages of zero under certain interval relations.

Interval type	MR-diagram region	Image of 0	Coimage of 0
Without zero	Below **lb0-ub0**	$0 \not\subseteq$	$\not\supseteq 0$
Positive	Below **lb0**	$0 <$	> 0
Negative	Below **ub0**	$0 >$	< 0
Thick positive	Between **lb0** and **Om+**	$0\,(\mathrel{\overline{\cdot}} \cap <)$	$(\doteq \cap >)\,0$
Thick negative	Between **ub0** and **Om−**	$0\,(\mathrel{\overline{\cdot}} \cap >)$	$(\doteq \cap <)\,0$
Thin	On **Om**	$0 \mathrel{-\!\!\cdot}$	$\mathrel{-\!\!\cdot} 0$
Thin positive	On **Om+**	$0\,(\mathrel{-\!\!\cdot} \cap <)$	$(\mathrel{-\!\!\cdot} \cap >)\,0$
Thin negative	On **Om−**	$0\,(\mathrel{-\!\!\cdot} \cap >)$	$(\mathrel{-\!\!\cdot} \cap <)\,0$
Containing zero	On or above **lb0-ub0**	$0 \subseteq$	$\supseteq 0$
Over zero	Above **lb0-ub0**	$0 \sqcup$	$\sqcap 0$
Zero-start	On **lb0**	$0 \sqsubset = 0 \urcorner$	$\sqcap 0 = \lrcorner 0$
Zero-end	On **ub0**	$0 \sqcap = 0 \lrcorner$	$\sqcap 0 = \urcorner 0$
Zero-endpoint	On **ub0** or **lb0**	$0\,(\sqsubset \cup \sqcap) =$ $= 0\,(\lrcorner \cup \urcorner)$	$(\sqcap \cup \sqcap)\,0 =$ $= (\urcorner \cup \lrcorner)\,0$
Symmetric	On **Or+**	$0 \mathrel{+\!\!\cdot}$	$\mathrel{+\!\!\cdot} 0$
Middle-positive	Within **Rm+** or on **Om+**	$0 \bullet\!\mid$	$\mid\!\bullet 0$
Middle-negative	Within **Rm−** or on **Om−**	$0 \mid\!\bullet$	$\bullet\!\mid 0$

III.4 Interval arithmetic

... and then the different branches of Arithmetic—
Ambition, Distraction, Uglification and Derision.
[Lewis Carroll, *Alice's Adventures in Wonderland* (1865)]

Diagrammatic representations of arithmetic operations on intervals provide better understanding of their nonstandard behaviour and help in finding and proving their interesting and possibly useful properties. As already mentioned in Section III.1.3, these representations are not practical for conducting actual interval calculations, and that is certainly not their purpose.

We will start in Section III.4.1.1 from the simplest operation of addition, which produces no nonstandard effects. However, subtraction and negation presented in Section III.4.1.2 behave already in a nonstandard way as compared to real number arithmetic (cf. Section III.1.1.2). As a result, interval subtraction ceases to be the opposite operation to addition, resulting in nonstandard properties of the simplest interval equation $a + x = b$, see Section III.4.1.3. The diagrams explain clearly the underlying causes of that behaviour, helping to understand better the properties of interval calculations.

Next, the much more complex operation of interval multiplication is analyzed diagrammatically in Section III.4.2, with the construction developed for the first time by this author, see [7, 105]. As it was found later, two other authors [Ratschek 1973, Gardeñes et al. 1981] independently approached the idea, but failed to produce the finished solution. The basic properties of the important interval equation $a \cdot x = b$ are also presented in Section III.4.2.3. Detailed analysis of the equation and its multidimensional generalization $\Sigma_{i=1}^{n} a_i \cdot x_i = b$ is further continued in Section III.5.

The construction for interval division (Section III.4.3), also first developed by this author, works properly also for the division by an interval containing zero, producing in such cases the so-called extervals used in Kahan arithmetic, see Section III.4.5. It is also useful in the analysis of interval linear equations in Section III.5 (especially Section III.5.2.4), compare in this respect Figs. III.43 and III.47.

The contents of this section is based on [7, 105].

III.4.1 Interval addition, negation and subtraction

He modified the time intervals, ...
adding and subtracting tasks to make sure ...
[Robert L. Forward, *Rocheworld* (1990)]

While *addition of intervals* works exactly like ordinary two-dimensional vector addition (or addition of complex numbers), *interval negation* and, in consequence, *interval subtraction* behave already in a nonstandard way (cf. Section III.1.1.2). Diagrammatic constructions developed in this section help to understand the underlying reasons for that.

To show better the underlying mechanisms of interval arithmetic, the diagram for addition will be first constructed in a way different than for real vectors. Instead, it will be based on the generic definition of interval arithmetic operations, as defined by (III.14). This way of constructing interval operations will be also useful later, especially

for multiplication in Section III.4.2, where it will also reveal a surprising diagrammatic analogy between interval addition and multiplication.

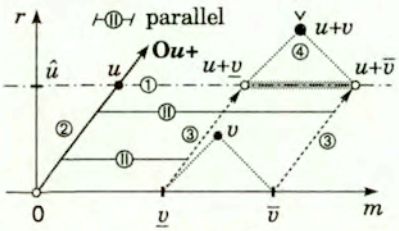
The nonstandard behaviour of subtraction (and negation necessary for its definition), see Section III.1.1.2, will be explained with the diagrams developed in Section III.4.1.2, as will be done for similarly nonstandard properties of the simplest interval equation $a+x=b$, see Section III.4.1.3. The constructions will also suggest an obvious extension of interval algebra towards the so-called *Kaucher* (or *directed*) arithmetic, see Section III.4.4.

III.4.1.1 Addition of intervals

According to (III.14), to add two intervals one should take an interval hull of results of adding all numbers from the first interval to those from the second. Adding two numbers in the MR-diagram (recall that real numbers are represented there as points on the **Om** axis) amounts to moving one of them along the axis by the distance given by the other number (using the other's sign to indicate the direction of movement). Thus, adding a number to an interval amounts to a horizontal movement of the interval by the distance given by the number. Adding in such a way all numbers from the second interval to the first interval we obtain a set of intervals whose hull will give the final sum. In formulae:

$$\begin{aligned} u+v &= \mathbf{hull}\{\tilde{u}+\tilde{v} \,|\, \tilde{u}\in u \,\mathbf{and}\, \tilde{v}\in v\} \\ &= \mathbf{hull}\{u+\tilde{v} \,|\, \tilde{v}\in v\}. \end{aligned} \tag{III.25}$$

The diagrammatic construction for the above is described, step by step, in Fig. III.26.



$u+v=(u+\underline{u})\vee(u+\overline{u})$:
① draw a constant radius line $r=\hat{u}$,
② draw a line from the origin **O** to u,
③ draw lines parallel to ② from $\underline{u}$ and $\overline{u}$ to intersect $r=\hat{u}$, i.e., add $\underline{u}$ and $\overline{u}$ to u,
④ the hull of the intersection points gives the result $u+v$.

Figure III.26: Step by step construction for addition of two thick intervals u and v according to the generic rule of (III.14).

Note how the movement of the interval u is constructed by drawing lines parallel to the axis **Ou** and intersecting them with the constant radius line of u. In this way, the metric operation of measuring and marking out the value of the distance is avoided, see Section II.4.1.1 for reasons why it is desirable. The thick gray line represents the final set in the formula (III.25) above. As it is obvious from the construction, this set is always a single horizontal straight line segment. Therefore, its hull is obviously equal to the hull of its endpoints, as indicated by the formula in Fig. III.26.

The idea of the construction based on the generic definition (III.14) can be repeated for other operations on intervals, especially for multiplication, see Section III.4.2. It does not work directly for thin intervals (reals). However, it is easy to transform addition of numbers into addition of a number to an interval: just construct any interval with the

midpoint given by the first number, add to it the second number using the construction in Fig. III.26 and take the midpoint of the result. For completeness, the construction is shown in Fig. III.27. In this way we can avoid again the metric operations of measuring and marking out values on axes, obtaining a more qualitative (structural) insight (see Section II.4.1.2) into the properties of arithmetic operations.

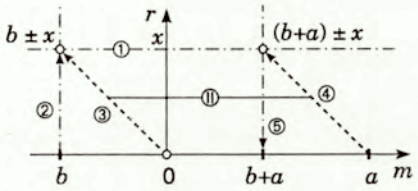


Figure III.27: Addition of reals in the MR-diagram avoiding metric measurements.

Moreover, in the construction in Fig. III.27 one may take any slope of the (parallel) lines used in steps ② and ⑤, not only the vertical constant midpoint lines shown.

The familiar parallelogram construction for adding vectors also works here, as it is obvious from the formula (III.6), or as can be geometrically inferred from the construction of Fig. III.26. As the parallelogram construction will be useful in the sequel, it is recalled here, in two versions, in Fig. III.28. Note how the two versions of the order of the addition in Fig. III.28a prove also commutativity of interval addition, much more tedious to show directly from the generic construction of Fig. III.26.

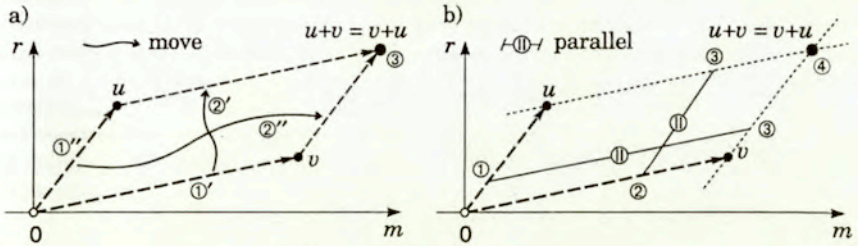


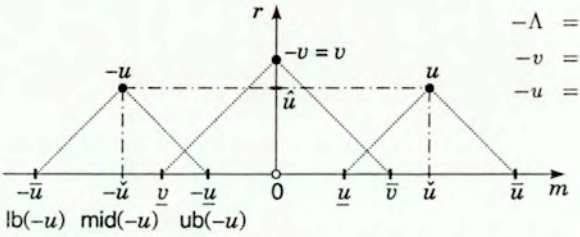
Figure III.28: Two versions of the standard parallelogram construction for interval addition.

Note various diagrammatic conventions used in the diagrams above (see also Fig. III.5) which will be used consistently in other arithmetic diagrams in the following.

III.4.1.2 Negation and subtraction of intervals

Negation (change of sign) in arithmetic of reals is represented on a real number axis as a symmetrical reflection in the point 0. Negation of an interval is also represented in the MR-diagram as a symmetrical reflection—now in the (semi)-axis **Or+**, see Fig. III.29. It follows directly from (III.5): $-u = \{-\tilde{u} \mid \tilde{u} \in u\}$, with all real numbers $-\tilde{u}$ obtained according to a rule for reals from members of u, that is as symmetrical reflections in the

point 0 of the **Om** axis. Note that in accordance with the formula (III.7), from among the four basic parameters of an interval, the radius is not negated by the change of sign of the interval. That has important consequences for interval arithmetic, as will become clear later. Also, only one real number is equal to its negation (the number 0), while there are infinitely many intervals of this property, namely, all zero-symmetric intervals (lying on the **Or**+ axis), see Section III.2.3 (Fig. III.6).



$$-\Lambda = -[-1,+1] = [-1,+1] = \Lambda,$$
$$-v = -\Lambda\hat{v} = \Lambda\hat{v} = v,$$
$$-u = -\breve{u} + \Lambda\hat{u} = [-\overline{u}, -\underline{u}].$$

Figure III.29: Negation (change of sign) of an interval.

Subtraction of intervals, defined as addition of negation, see (III.8), works just like addition after first mirroring the subtracted interval in the **Or**+ axis. Thus, the construction of Fig. III.26 can be used here as well, only augmented by first negating the second argument. In Fig. III.30a, a vector construction for subtraction of intervals is shown instead, in both $u - v$ and $v - u$ order. As it is easily seen, subtraction of intervals does not work like subtraction of ordinary vectors—while midpoints subtract as expected, radii actually add (just like in interval addition). Thus, the result is always wider than the widest of arguments (or at least as wide, when the other argument is thin). In consequence, while for reals (and real vectors) we have always $u - u = 0$, for thick intervals it is not so. In fact, $u - u$ is always a zero-symmetric interval with the radius twice the radius of u, see Fig. III.30b. Therefore, subtraction ceases to be an opposite operation to addition, and the algebraic transformation rule of moving a term (with an opposite sign) to the other side of an equation does not work in interval algebra.

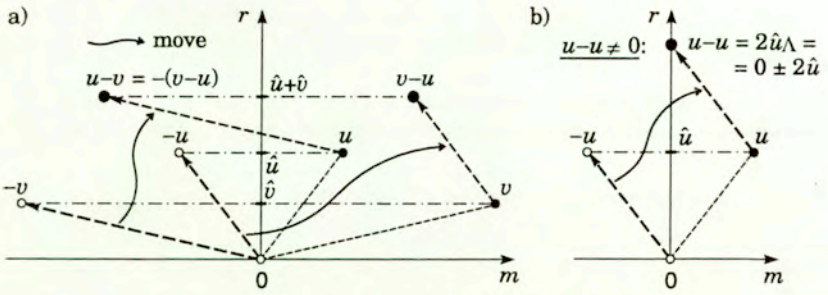


Figure III.30: Subtraction of intervals: differences
$u - v = u + (-v)$ and $v - u = v + (-u)$ (a), and $u - u$ (b).

III.4.1.3 The $a + x = b$ equation

The property of interval subtraction shown above has significant consequences for the solvability of the simplest interval equation $a + x = b$, where a and b are some given constant intervals and x is an unknown interval. For reals, we have simply $x = b - a$; however, it is not so for (thick) intervals. The interval solution to this equation is described by the following:

Proposition III.2 (The $a + x = b$ equation) *The interval equation $a + x = b$ has a solution (in the set of ordinary intervals $\mathbb{IR}$) only when $\hat{a} \leq \hat{b}$, and this solution in general (unless a is thin) does not equal $b - a$. It has the same midpoint, but different (smaller) radius, because $x = (\check{b} - \check{a}) \pm (\hat{b} - \hat{a})$, whereas $b - a = (\check{b} - \check{a}) \pm (\hat{b} + \hat{a})$.*

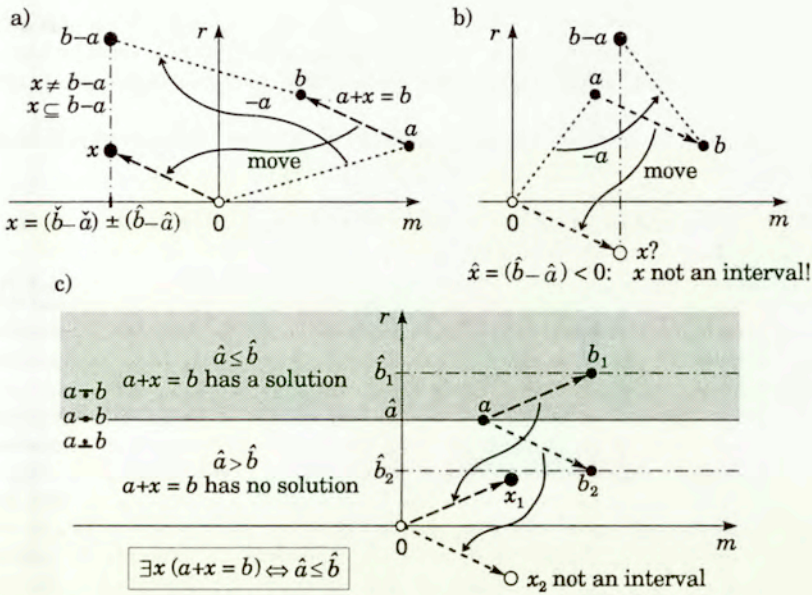


Figure III.31: The $a + x = b$ equation: solution exists (a), solution does not exist (b), and condition for existence of a solution (c).

Proof. As it is obvious from the diagram in Fig. III.31a, the solution x of the equation must be such that the vector from 0 to x, when moved to begin at a, will end up exactly at b. An ordinary interval with such a property exists only when b lies above (or at the same level) as a in the MR-diagram, i.e., when $\hat{a} \leq \hat{b}$, because only then the vertical component of x can be nonnegative. Otherwise (see Fig. III.31b), the required x would have to lie below the **Om** axis, and hence it would not be an interval. Moreover, $\hat{x} = \hat{b} - \hat{a}$ is smaller than (or at most equal to) $\mathbf{rad}\,(b - a) = \hat{b} + \hat{a}$. The rule can be neatly summarised diagrammatically, see Fig. III.31c: for a given interval a, all intervals b for

which the equation $a + x = b$ has a solution must lie on or above the constant radius line $r = \hat{a}$ of the interval a, while for intervals b below that line the equation has no solution (in the set of ordinary intervals). □

The construction readily suggests an extension of the interval arithmetic to include the points below the **Om** axis (so-called *improper intervals*). Such an extension has been proposed already by Warmus in his precursory paper [Warmus 1961], see Section III.4.4 and is now called *Kaucher* (or *directed*) *interval arithmetic*. In this arithmetic the equation $a + x = b$ has always a unique solution, and certain other problems with the ordinary interval arithmetic can be avoided.

III.4.2 Interval multiplication

You can make C-A-G-E out of B-A-C-H
by multiplying all the intervals by $3\,1/3$, ...
[Douglas R. Hofstadter, *Gödel, Escher, Bach: an Eternal Golden Braid* (1979)]

Similarly as for addition, to construct a product of intervals it is first necessary to find a construction for multiplying an interval by a real number.

III.4.2.1 Multiplication of an interval by a number

As the interval axis of u (excluding the origin $[0, 0]$) groups all intervals with the same value of the **rex** function, and $\mathbf{rex}\, u = \mathbf{rex}\,(a \cdot u)$ for any positive real a, while $\mathbf{rex}\, u = -\mathbf{rex}\,(b \cdot u)$ for any negative real b (see Section III.2.3.2), the interval axis of u is the locus of products of this interval by all real numbers, symbolically: $\mathbf{O}u = \mathbb{R} \cdot u$.

To find the product of an interval u and a real number m it thus suffices to map appropriately the point on the **Om** axis with the coordinate m onto the interval axis $\mathbf{O}u$. It is useful to define the mapping as a function called here *λ-mapping*: $\lambda_u : \mathbf{Om} \to \mathbf{O}u$; $\lambda_u(m) = m \cdot u$. Its inverse allows to find the real number (a point on the **Om** axis) by which the interval u has been multiplied to obtain the given point on the axis $\mathbf{O}u$. Diagrammatic constructions for the mapping (and therefore, for diagrammatic multiplication of an interval u by a real number) are shown in Fig. III.32. Depending on the need, one may use either the mapping lines parallel to the lines from the points **1m** and **−1m** to u and $-u$, respectively, or from **1m** to $-u$ and from **−1m** to u, taking into account the equalities:

$$\lambda_{-u}(m) = \lambda_u(-m) = -\lambda_u(m). \tag{III.26}$$

The mapping orients the interval axis according to the sign of the midpoint of interval u, hence dividing the axis $\mathbf{O}u$ into $\mathbf{O}u+$ and $\mathbf{O}u-$ half-lines, cf. also Fig. III.5c. The distance of the interval from the origin defines the unit of scale on the interval axis. Note the fundamental role of the points **1m** and **−1m** for the definition of the mapping, and hence for all subsequent constructions for multiplication (and division) of intervals.

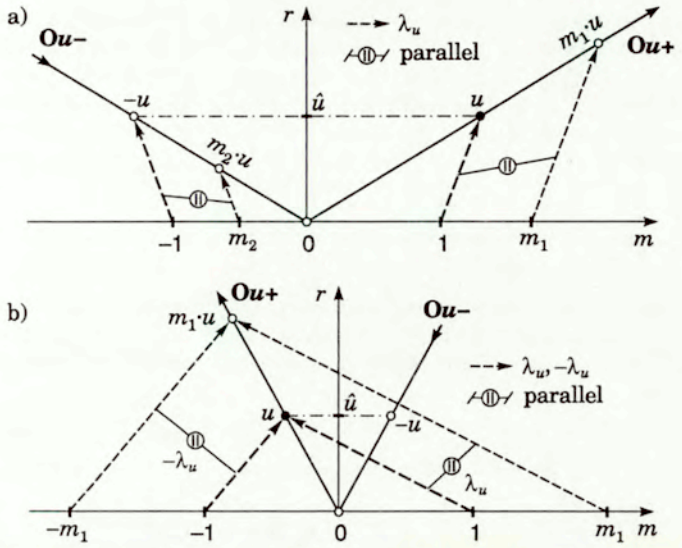
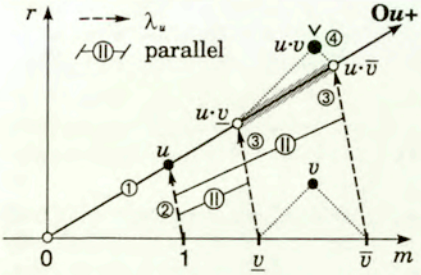


Figure III.32: Multiplication of an interval by a number (*λ-mapping*) for middle-positive (a) and middle-negative (b) intervals.

III.4.2.2 Multiplication of intervals

Similarly as for addition, it is easy to show (cf. Section III.4.1.1), that multiplication of intervals u and v reduces to finding a hull of products of u by $\underline{v}$ and $\overline{v}$. The generic construction for positive multiplicands requires two applications of the λ-mapping (i.e., it reduces to two multiplications of an interval by a number, according to Fig. III.32), and is shown in Fig. III.33.



$u \cdot v = (u \cdot \underline{v}) \vee (u \cdot \overline{v})$:

① draw the interval axis **Ou** of u,
② draw a line from the point **1m** to u,
③ draw lines parallel to ② from $\underline{v}$ and $\overline{v}$ to intersect with **Ou**, i.e., multiply u by $\underline{v}$ and $\overline{v}$,
④ the hull of the intersection points gives the result $u \cdot v$.

Figure III.33: A generic construction for multiplication: here for two positive intervals u and v.

Addition-multiplication analogy. There is an important structural similarity of this construction and the construction for addition of intervals (compare Figs. III.26 and III.33). The procedure is basically the same, only the role of the constant radius line

$r = \hat{u}$ is here played by the interval axis $\mathbf{O}u$, while the role of the axis (i.e., the line from the origin $\mathbf{O}$ to the interval u) is played by the line from the point $\mathbf{1m}$ (or $-\mathbf{1m}$) to the interval u (or $-u$). Consider next that the constant radius line $r = \hat{u}$ is the locus of sums of u and all real numbers, i.e., $\mathbb{R} + u$, while the interval axis is the locus of products $\mathbb{R} \cdot u$, and that the number zero (marked on the $\mathbf{O}m$ axis at the origin $\mathbf{O}$) is the additive identity, while the number one (marked on the $\mathbf{O}m$ axis at the point $\mathbf{1m}$) is the multiplicative identity.

Multiplication of other types of intervals uses the same general principle, but the actual constructions differ (superficially), as shown in the following example.

Example III.7 (Multiplication of intervals) The examples in Fig. III.34 help to understand the structure of interval multiplication better by showing constructions for different types of arguments. For clarity, the parallelism and step indicators have been omitted.

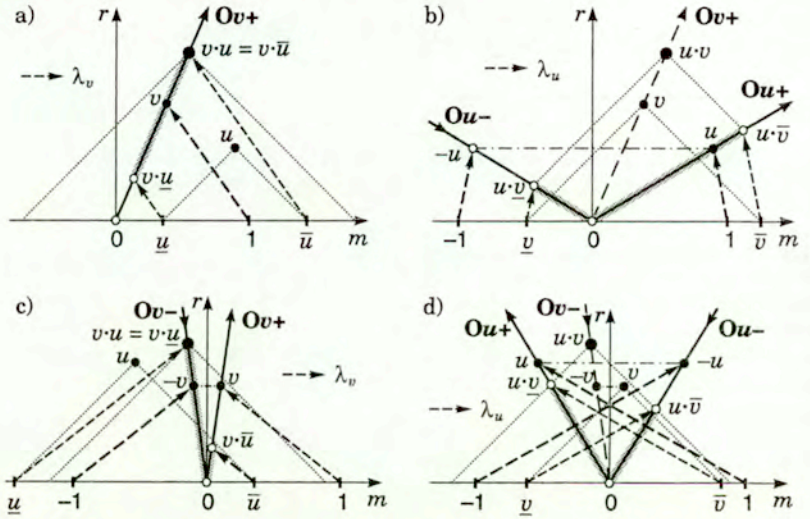


Figure III.34: Examples of multiplication of various types of intervals: u positive and v middle-positive, containing zero (a), the same in opposite order (b); both over zero, u middle-negative and v middle-positive (c), and in opposite order (d).

In the examples always at least one argument contains zero. The cases with both arguments without zero are covered essentially by the construction in Fig. III.33, only with some parts mirrored in the $\mathbf{O}r$ axis when one or both intervals are negative. Note that the product in the cases in Fig. III.34 has always the same absolute extent $|\mathbf{rex}|$ as the more extended of the arguments (i.e., the product lies on the higher of the interval axes of the arguments). Moreover, in the case when that higher interval axis is chosen for the first step of the construction (see Fig. III.33), the final hull operation is trivial, as the product lies on one of the endpoints of the thick gray line (i.e., equals either $v \cdot \underline{u}$ or $v \cdot \overline{u}$), and it suffices to apply the λ-mapping once and omit the hull operation altogether. ■

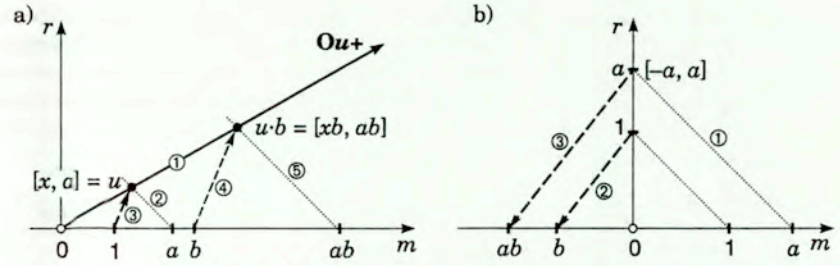


Figure III.35: Multiplication of reals in the MR-diagram: of the same sign (a) and opposite signs (b).

Similarly as for addition, the constructions of Figs. III.33 and III.34 do not work for multiplication of reals (thin intervals). In this case, however, essentially the same approach as that demonstrated for addition in Fig. III.27 can be used, as shown in Fig. III.35a—this time using not the midpoint, but the endpoint of the auxiliary interval u. As with addition, in fact any slope of the auxiliary lines is steps ② and ⑤ can be taken. In Fig. III.35b the construction originally derived from interval multiplication (using the **Or** axis as an auxiliary interval axis) was then substantially transformed to simplify the diagram. Many other constructions for multiplication of reals are also possible, see [105] for more examples.

The case of the *fast multiplication* described in Example III.7 is covered by the following proposition.

Proposition III.3 (Fast interval multiplication) *Whenever an interval v contains zero (i.e., $|\mathbf{rex}\, v| \geq 1$) and the interval axis of v lies above that for u (i.e., $|\mathbf{rex}\, v| \geq |\mathbf{rex}\, u|$), the product $v \cdot u$ depends only on one of the endpoints of u and lies on the interval axis of v, namely, it is equal to a product of v and one of the endpoints of u (see Fig. III.34a and c), according to the formula:*[9]

$$v \cdot u = \begin{cases} v \cdot \underline{u} & \text{for } \check{u} \leq 0, \\ v \cdot \overline{u} & \text{for } \check{u} \geq 0. \end{cases} \tag{III.27}$$

Proof. For u without zero, an immediate demonstration is given by the diagram in Fig. III.34a (or a symmetrically analogous diagram for $\check{v} \leq 0$). This is so because in this case the interval axis **Ov** makes with the **Om** axis an angle larger than 45° and the λ-mapping of both $\underline{u}$ and $\overline{u}$ will lie on the same side of the **Or** axis. Thus, always one of the intervals $v \cdot \underline{u}$ or $v \cdot \overline{u}$, namely that lying higher than the other, will contain the other one, so that the hull of them will be equal to that higher interval (see also Section III.2.3.3, Fig. III.8a).

For u containing zero, the demonstration is more complicated, see Fig. III.34c (taking $\check{u} \leq 0$ for variety). Demonstrating the validity of (III.27) amounts in this case to showing that the point $v \cdot \overline{u}$ never falls above the ub-diagonal of $v \cdot u$, provided the interval u is less extended than v, i.e., u lies below the interval axis **Ov**. It is clearly seen in the diagram that the point $v \cdot \overline{u}$ goes continuously up along the axis **Ov** when $\overline{u}$ grows moving to the

[9] It is easy to see that for symmetric u (when $\check{u} = 0$ and $\underline{u} = -\overline{u}$) v must also be symmetric and both cases in the formula (III.27) give the same result (also symmetric).

right along the **Om** axis (while $\underline{u}$ stays fixed). However, if only $v \cdot \overline{u}$ stays below the ub-diagonal of $v \cdot \underline{u}$, the interval represented by the point $v \cdot \underline{u}$ equals $v \cdot u$ and the formula (III.27) remains valid. The point $\overline{u}$ may move to the right only so far as it is allowed by the requirement of u staying below the axis **Ov**, hence its position is always lower (more to the left) than that attained for u lying on the **Ov** axis. Thus, it remains to show that when u lies on **Ov**, the point $v \cdot \overline{u}$ lies at most at the ub-diagonal of $v \cdot \underline{u}$.

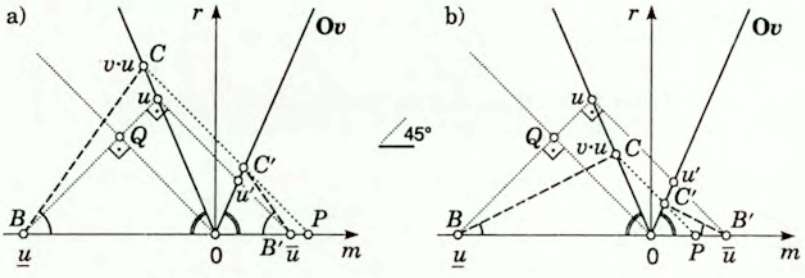


Figure III.36: Two cases of u lying on the axis **Ov** (to the proof of Proposition III.3, see text).

Indeed, the following argument (see Fig. III.36) shows that when u lies on **Ov**, the $v \cdot \overline{u}$ point (C′ in Fig. III.36) lies exactly on the ub-diagonal of $v \cdot \underline{u}$ (CP in Fig. III.36). The figure shows two possible cases, differing by the slope of the line BC (from $\underline{u}$ to $v \cdot u$): whether it is larger or smaller than 45°. The main course of the demonstration goes the same way for both cases, namely:

$\measuredangle \mathrm{OBC} = \measuredangle \mathrm{OB'C'}$ – by definition of the λ-mapping;
$\measuredangle \mathrm{BO}u = \measuredangle \mathrm{BOC} = \measuredangle \mathrm{B'O}u' = \measuredangle \mathrm{BOC'}$ – by definition of the interval axis **Ov**;
$\triangle \mathrm{OBC} \sim \triangle \mathrm{OB'C'}$ – because corresponding angles at B, B′ and O are equal, and
$\triangle \mathrm{OB}u \sim \triangle \mathrm{OB'}u'$ – as above (angles $\measuredangle \mathrm{OB}u$ and $\measuredangle \mathrm{OB'}u'$ both equal 45°).

Therefore:

$\overline{\mathrm{OB}}/\overline{\mathrm{OC}} = \overline{\mathrm{OB'}}/\overline{\mathrm{OC'}}$ – as corresponding pairs of sides in similar triangles;
$\overline{\mathrm{OB}}/\overline{\mathrm{O}u} = \overline{\mathrm{OB'}}/\overline{\mathrm{O}u'}$ – as above.

Thus, dividing the above by sides:

$\overline{\mathrm{OC}}/\overline{\mathrm{O}u} = \overline{\mathrm{OC'}}/\overline{\mathrm{O}u'}$,

which means that $\mathrm{CC'} \| u\mathrm{B'}$, thus $\mathbf{ub}\,\mathrm{C} = \mathbf{ub}\,\mathrm{C'}$ and, in consequence, $\mathbf{ub}\,(v \cdot u) = \mathbf{ub}\,(v \cdot \overline{u})$. Whenever $\overline{u} < \mathrm{B'}$ (i.e. when u moves towards Q), C′ will move towards u', which means $\mathbf{ub}\,\mathrm{C} < \mathbf{ub}\,\mathrm{C'}$, hence $\mathbf{ub}\,(v \cdot u) > \mathbf{ub}\,(v \cdot \overline{u})$, as expected (see Fig. III.34c). □

Diagram-aided proofs. Note certain characteristic features of the diagram-aided demonstration given above, shared by many proofs of that kind:

- There are several structurally distinct types of diagrammatic configurations representing the situation described by the theorem to be proved, leading naturally to the proof by cases (the effect called "divergence" in diagrammatic reasoning theory, see Section II.4.3).

- Proofs of some of these cases are immediate (i.e., follow directly from the diagram, see Fig. III.34a for the case of u without zero). That "immediacy" needs, of course, some additional support to be fully acceptable on more formal grounds, otherwise it may lead to errors, see Sections II.5.2 and II.3.2.

- Diagrammatic arguments are often based on analysing how the diagram may change when some parameter is varied. Thus, the diagrammatic proofs can be made easier and more effective with the use of interactively animated diagrams, like those discussed in Sections II.5.4.5 and II.6.4.

Historical remark. As mentioned at the beginning of this section, two other authors independently came near to a diagrammatic construction for interval multiplication, though they failed to produce the finished solution. First, [Ratschek 1973] presented a construction based on his three-dimensional interval space in which multiplication can be done component-wise (formal derivation of this space was published much later in [Ratschek 1980]). The construction, drawn in an E-diagram (see Section III.2.1), was very intricate and non-intuitive, and covered only the case of *fast multiplication* of intervals containing zero, where only one endpoint of the multiplier is used, see Proposition III.3. Later on Gardeñes et al., in the paper [Gardeñes et al. 1981], concerning essentially the arithmetic of twins (see a note in Section III.2.3.3), presented some separate fragments of the construction. Namely, in a very sketchy E-diagram, they illustrated the locus of products of an interval by real numbers (a line called here *interval axis*), and, within the context of twin arithmetic, they illustrated the analogue of the formula $u \cdot v = (u \cdot \underline{v}) \vee (u \cdot \overline{v})$, i.e., the construct needed in the last step of the construction for interval multiplication. However, the construction for actually obtaining the product of a given interval by a given number (the λ-mapping), and all the other necessary steps of the construction did not appear there.

III.4.2.3 The $a \cdot x = b$ equation

The interval equation $a \cdot x = b$ constitutes the basic (one-dimensional) case of a general linear system of interval equations, a construct of great theoretical and practical importance (see Section III.5), hence the additional significance of this simple equation. As with the analogous equation containing addition, a and b are some given constant intervals and x is the sought of unknown interval. Using the construction for interval multiplication (Figs. III.33 and III.34), we can easily construct a diagrammatic solution to the equation. For a without zero, the construction is shown in Fig. III.37. One can spot the analogy with the case of the $a + x = b$ equation (Fig. III.31), but this time solution does not exist when b lies below the axis $\mathbf{O}a$ of a, not below its constant radius line. It is so also in the case with a containing zero (the construction is essentially the same as in Fig. III.37b, see [105] for the appropriate diagram).

Indeed, the general condition for the existence of a solution to this equation can be represented diagrammatically as in Fig. III.38, according to the following:

Proposition III.4 ($a \cdot x = b$: Existence of solutions) *The interval equation $a \cdot x = b$ has a solution (in the set of ordinary intervals $\mathbb{IR}$) only when $|\mathbf{rex}\, a| \leq |\mathbf{rex}\, b|$, i.e., when the interval b lies above or on the axis $\mathbf{O}a$.*

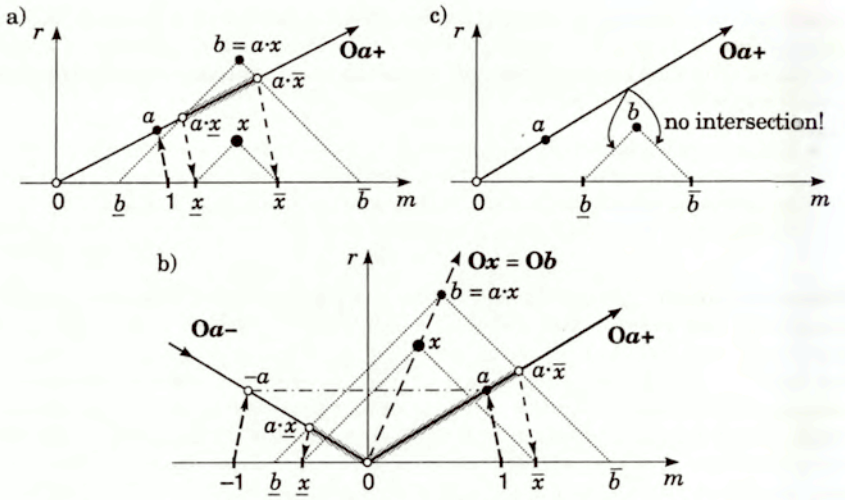


Figure III.37: Solving the $a \cdot x = b$ equation diagrammatically for a without zero: for b also without zero (a), b containing zero (b), and when the solution does not exist (c).

Proof. The condition for the existence of solutions follows immediately from Fig. III.37 (see also [105] for more cases), as summarised in Fig. III.38. □

Moreover, there is also a curious case of non-unique solutions to the equation:

Proposition III.5 ($a \cdot x = b$: Non-unique solutions) *When both a and b contain zero and lie on the same interval axis (i.e., when $|\mathbf{rex}\, a| = |\mathbf{rex}\, b| \geq 1$), the solution to the interval equation $a \cdot x = b$ is not unique: there is actually a set of solutions $X_=$ which constitutes a segment of the diagonal line with endpoints x_m on the* **Om** *axis and x_a on the* **Oa** *axis (coinciding with* **Ob**, *possibly with opposite orientation when $\check{a}$ and $\check{b}$ are of different signs), as defined by:*

$$x_m = \check{b} \,/\, \check{a} = \hat{b} \,/\, \hat{a} \cdot \mathbf{sgn}\, \check{a}\check{b}, \tag{III.28}$$

$$x_a = b \cdot \mathbf{sgn}\, \check{a} \,/|a| = ((\check{b}\, \mathbf{sgn}\, \check{a}) \pm \hat{b})/|a|. \tag{III.29}$$

When a is symmetric, of course b is also symmetric, and $X_=$ consists of two *diagonal segments: one from $x_a = \Lambda\, \hat{b} \,/\, \hat{a}$ to $x_{m1} = \underline{x_a}$ and the other to $x_{m2} = \overline{x_a}$ (so that $x_{m1} = -x_{m2}$).*

Proof. The formulae (III.28) and (III.29) can be derived using Fig. III.39. Namely, we have immediately $a \cdot x_m = b$, hence:

$(\check{a} \pm \hat{a}) \cdot x_m = \check{b} \pm \hat{b}$, i.e., see (III.10):
$\check{a} \cdot x_m \pm \hat{a} \cdot |x_m| = \check{b} \pm \hat{b}$, thus:
$\check{a} \cdot x_m = \check{b}$ and $\hat{a} \cdot |x_m| = \hat{b}$,

from which the formula (III.28) follows directly.

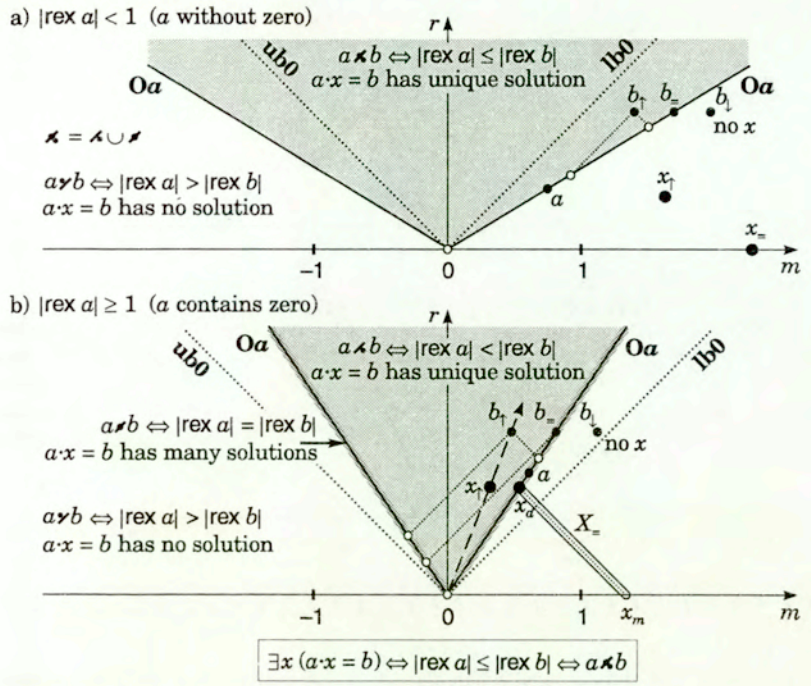


Figure III.38: The $a \cdot x = b$ equation, condition for existence of a solution: a without zero (a); a contains zero (b).

Of course, we have also $a \cdot x_a = b$, and because x_a lies on the axis $\mathbf{O}\boldsymbol{a}$, then x_a contains zero, hence, from (III.27):

$$a \cdot x_a = b = \begin{cases} \underline{a} \cdot x_a & \text{for } \check{a} \le 0, \\ \overline{a} \cdot x_a & \text{for } \check{a} \ge 0. \end{cases}$$

Because $\underline{a}$ and $\overline{a}$ are real numbers, and $\underline{a} = \check{a} - \hat{a}$ and $\overline{a} = \check{a} + \hat{a}$, see (III.1), we have:

$$\begin{aligned} x_a &= \begin{cases} b/\underline{a} & \text{for } \check{a} \le 0, \\ b/\overline{a} & \text{for } \check{a} \ge 0, \end{cases} \\ &= \begin{cases} b/(\check{a} - \hat{a}) & \text{for } \check{a} \le 0, \\ b/(\check{a} + \hat{a}) & \text{for } \check{a} \ge 0. \end{cases} \end{aligned}$$

From the definition of magnitude in (III.2) we have $\check{a} - \hat{a} = -|a|$ when $\check{a} \le 0$, and $\check{a} + \hat{a} = |a|$ when $\check{a} \ge 0$, so that, finally:

$$x_a = b/|a| \,\mathbf{sgn}\, \check{a},$$

as required for validity of (III.29).

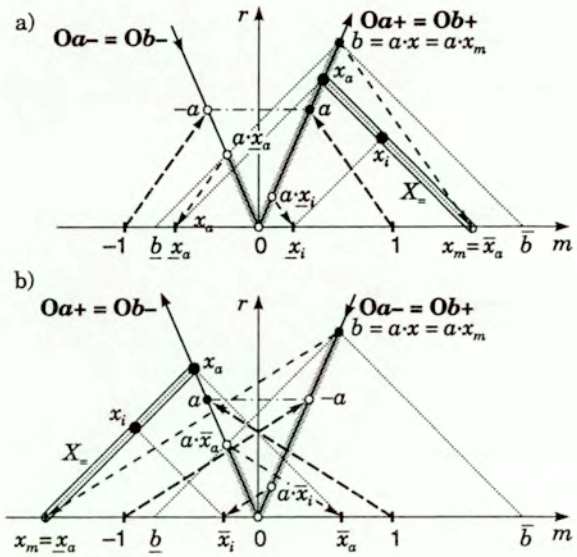


Figure III.39: The $a \cdot x = b$ equation, non-uniqueness of a solution: with midpoints of a and b of the same (a) and different (b) signs.

The special case of symmetric a can also be easily covered using the above reasoning (omitted here for brevity). □

The solution to the equation considered here is sometimes called an *algebraic* (or *formal*) *solution* and denoted as x_A, to differentiate it from other kinds of solutions defined for this equation. A detailed discussion of these other solutions is conducted in Section III.5, with appropriate diagrammatic constructions. Various relations between x_A and the other solutions in the one-dimensional case are established in Section III.5.2.4.

III.4.3 Interval inverse and division

Division of intervals is defined in terms of an *inverse* (or *reciprocal*) of an interval which in turn is defined in terms of an inverse of a number, see (III.12). There are several possible constructions for an inverse of a number (see [105] for several examples). The basic one is presented in Fig. III.40. It is a direct adaptation of the construction in Fig. III.35a for multiplication of reals, as $b = 1/a$ when $ab = 1$, so we simply seek such a b that its product with the given a equals 1. As with multiplication of reals, the auxiliary lines can be of any slope, not necessarily the 135° diagonals used in the figure.

III.4.3.1 Inverse of an interval

With the construction for an inverse of a number in place, inverting an interval according to (III.12) amounts to inverting its endpoints, see Fig. III.41a. The figure shows the

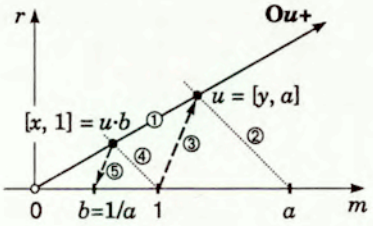


Figure III.40: Inverse (reciprocal) of a number.

construction using the arrangement of Fig. III.40 twice (with different slopes of auxiliary lines). The first is used to invert $\overline{u}$, and the second to invert $\underline{u}$. It is particularly simple because all the λ-mapping lines it requires can be of the same slope. Due to the choice of a negative interval u in this example, the construction uses the negation of the λ-mapping, see Fig. III.32b, taking advantage of the fact that $\mathbf{sgn}\, u = \mathbf{sgn}\,(1/u)$.

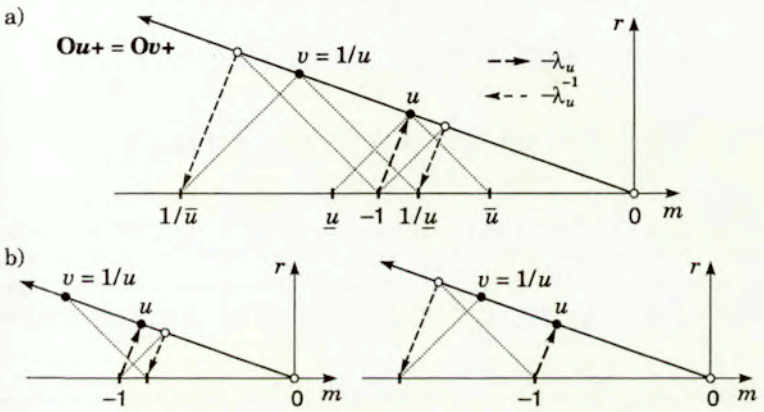


Figure III.41: Inverse of an interval: the combined construction for negative interval u containing -1 (a), and its simplified variants (b).

A simple though remarkable property of the interval inverse that can be observed in Fig. III.41a says that for any interval without zero (for which an inverse exists) $\mathbf{rex}\, u = \mathbf{rex}\,(1/u)$, i.e., both u and $1/u$ lie on the same interval axis $\mathbf{O}u+$. It means that $1/u = a \cdot u$, where $a = 1/\underline{u}\overline{u} = 1/(\check{u}^2 - \hat{u}^2)$, see (III.12). That makes it possible to construct an inverse of an interval u with only a single application of the construction for inverting a number. Namely, it suffices to find an inverse of only one of its endpoints ($\underline{u}$ or $\overline{u}$), and then draw a ub-diagonal (or an lb-diagonal, respectively) from it to intersect the $\mathbf{O}u+$ axis at the point $1/u$, as the simple schemes of Fig. III.41b demonstrate.

In the above, it was assumed that $0 \notin u$. When u is over zero, one can still invert its endpoints, and construct the interval $v = [1/\underline{u}, 1/\overline{u}]$. However, the interval v is not an inverse of u in the sense of (III.5) or (III.14), as it contains all those real numbers that are *not* inverses of any $\tilde{u} \in \mathbf{int}\, u$. The true inverse of an over-zero interval u is not an

(ordinary) interval, but a set $\mathbb{R} \setminus \mathbf{int}\, v$ comprising two half-lines: from $-\infty$ to $1/\underline{u}$ and from $1/\overline{u}$ to $+\infty$ (or one of these for a zero-endpoint interval). Such a set, called an *exterval*, or *Kahan interval*, belongs to the extension of interval arithmetic called *Kahan arithmetic*, see Section III.4.5.

III.4.3.2 Division of intervals

Division of intervals can be constructed diagrammatically as a multiplication by an inverse, as shown in Fig. III.42. This construction, however, is a little involved and hence

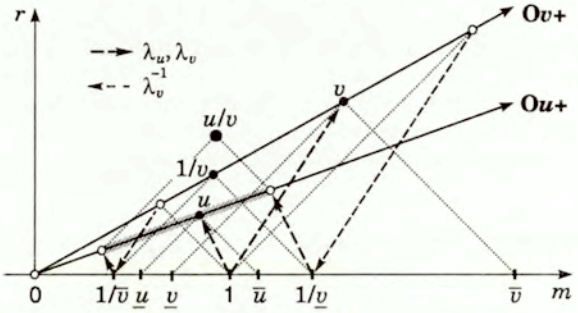


Figure III.42: Division of intervals: the standard construction using multiplication by an inverse.

impractical. A much simpler construction exists, as shown in Fig. III.43 for two types of arguments. That *direct* construction, however, is not so obvious and requires some additional justification. A mostly algebraic justification for the basic case with both arguments being positive intervals is given by the following proposition. A more diagrammatic argument is provided in the proof of Theorem III.6 in Section III.5.2.4.

Proposition III.6 (Direct interval division) *For positive intervals u and v, the interval u/v can be obtained by the construction shown in Fig. III.43a, i.e., as an interval with endpoints obtained through the inverse λ-mapping of the points* A *and* B *obtained, in turn, as intersections of the interval axis* $\mathbf{O}v+$ *with the lb-diagonal of $\overline{u}$ and the ub-diagonal of $\underline{u}$.*

Proof. As the points between the diagonals used in the construction represent all intervals that share some common point with u, the segment AB of the axis $\mathbf{O}v+$ is the set of all multiplies of v (by some real number m) which have a non-empty intersection with the interval u, i.e.:

$$\mathrm{AB} = \{x \mid x = m \cdot v \ \textbf{and}\ m \in \mathbb{R} \ \textbf{and}\ x \cap u \neq \emptyset\}.$$

The construction produces (with the inverse λ-mapping λ_v^{-1}) the interval $[a, b]$ containing all numbers m with this property, which are thus the solutions of the inequality:

$$m \cdot v \cap u \neq \emptyset.$$

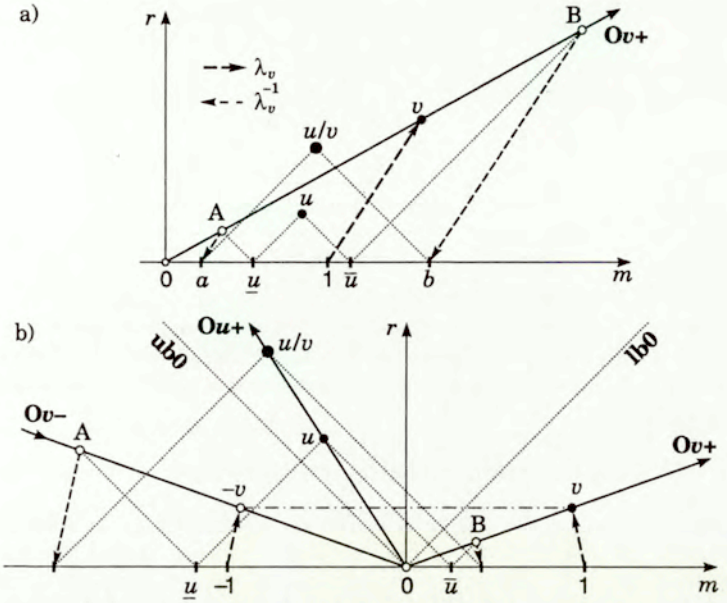


Figure III.43: Division of intervals: the direct construction for positive u and v (a), and for u middle-negative and over zero (b).

Because in the considered case (for positive u and v) we have $m > 0$, then:

$[\underline{u}, \overline{u}] \cap [m \cdot \underline{v}, m \cdot \overline{v}] \neq \emptyset$, that is:
$\underline{u} \leq m \cdot \overline{v}$ **and** $\overline{u} \geq m \cdot \underline{v}$, or:
$\underline{u}/\overline{v} \leq m \leq \overline{u}/\underline{v}$.

As it can be easily verified, for positive u and v:

$u/v = [\underline{u}/\overline{v}, \overline{u}/\underline{v}]$, hence:
$u/v = \{m \,|\, \overline{u}/\underline{v} \leq m \leq \overline{u}/\underline{v}\} = [a, b]$,

which proves the validity of the construction for this case. □

Other cases can be handled in exactly the same way (possibly with some additional juggling with the signs). An example is shown in Fig. III.43b, for u (the dividend) being an over-zero interval with the sign of the midpoint opposite to that of the divisor v. In this case u/v lies on the same interval axis as u, that is $|\mathbf{rex}\,(u/v)| = |\mathbf{rex}\,u|$, exactly as for multiplication. It is not surprising, given that $u/v = u \cdot (1/v)$, and $1/v$ never contains zero. This allows us to construct u/v for dividends containing zero even more simply, using only half of the construction in Fig. III.43b, like for an interval inversion, see Fig. III.41b. This is analogous to the fast multiplication rule, see Fig. III.34.

Division of reals can be also easily constructed on the basis of multiplication of reals (Fig. III.35), because $a/b = c$ means $a = b \cdot c$ for $a, b, c \in \mathbb{R}$. One possible construction, for both a/b and b/a, is shown in Fig. III.44, cf. Fig. III.35. For variety, we used here

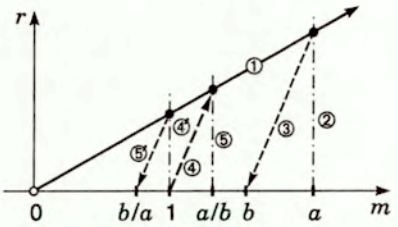


Figure III.44: Division of real numbers.

vertical auxiliary lines instead of ub-diagonals used in other similar constructions before. Justification of this construction in terms of interval arithmetic would be a little tedious, but it is easy to derive using Thales' theorem (one of several called by that name).

III.4.4 Kaucher arithmetic (directed intervals)

An extension of the MR-diagram to represent the so-called *Kaucher arithmetic*, see [Kaucher 1973, Kaucher 1980, Shary 1996], or *directed interval* algebra [Markov 1995], is quite straightforward. The way to do that was first suggested already by Warmus [Warmus 1961], and an extensive investigation of that extension (with operations defined differently than in the early Warmus proposal) was conducted by Kaucher [Kaucher 1973], hence the name *Kaucher arithmetic*. Because in that arithmetic the requirement that $\underline{u} \leq \overline{u}$ is dropped, so that the so-called *improper* intervals (with negative radius) are allowed, the lower, negative-radius half-plane of the MR-diagram becomes a natural place to represent these new objects. Not all properties of the ordinary interval space transfer naturally to the extended space—e.g., intervals can no longer be identified with sets of numbers included in them, and the inclusion relation for improper (and mixed) intervals does not coincide with set inclusion.[10] However, constructions for arithmetic operations extend naturally to that full interval space, showing even more closed ties with the lattice structure of the space. E.g., for multiplication we have, see Fig. III.45:

$$u \cdot v = \begin{cases} \bigvee\{u \cdot \tilde{v} \mid \tilde{v} \in v\} & \text{for } v \text{ proper,} \\ \bigwedge\{u \cdot \tilde{v} \mid \tilde{v} \in v\} & \text{for } v \text{ improper.} \end{cases}$$

Also, diagrammatic representation helps to clarify reasons for some, at the first sight peculiar, properties of some operations on directed intervals, like the existence of divisors of zero (that is, that the product of any two intervals containing zero of which one is proper and the other improper must always equal zero).

Further development and refinement of diagrammatic analysis of the directed interval space will be the subject of further research by this author.

[10]The modal interpretation of directed intervals due to [Gardeñes et al. 2001] restores the possibility for sensible treatment of the extended interval space in terms of sets of numbers.

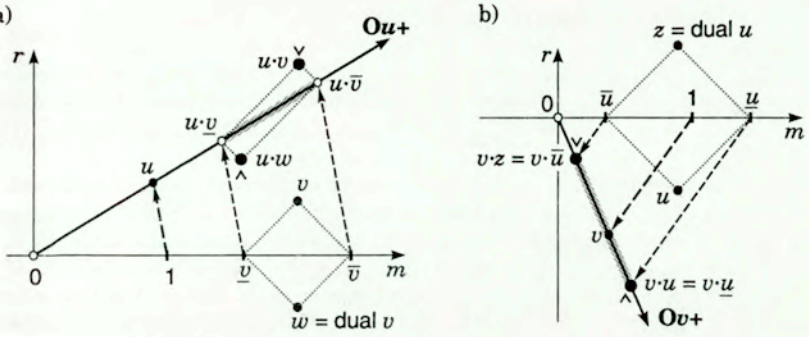


Figure III.45: Multiplication in the directed interval space, two examples: intervals without zero, w improper (a), and v under zero, improper, with u, z without zero, u improper (b).

III.4.5 Kahan arithmetic (extervals)

The so-called *Kahan arithmetic* [Kahan 1968, Laveuve 1975] is another extension adding to the directed interval space yet other objects—the so-called *extervals*, which can be defined informally as $]a, b[= \langle -\infty, a] \cup [b, +\infty\rangle$. Extervals can be thought of as results of division by an interval containing zero[11] and they pop up and are useful in many types of computations using intervals, e.g. in solving interval linear equations, see Section III.5, or calculating ranges of functions with discontinuities of the $1/(1 + 1/x)$ type, [Kaucher 1977]. Already some interval software packages implement them, see [Hickey et al. 2001].

Hovewer, differently than for directed intervals, incorporation of extervals into the interval space diagram is not so straightforward. In the space of directed intervals, occupying already the whole plane, there seems to be no room for any new objects. Fortunately, an idea due to Kaucher [Kaucher 1999] looks hopeful in this matter. It will constitute a subject of further study by this author.

[11] Though not all extervals can be so obtained.

III.5 Interval linear equations

He slowly learned [how] ... to ferret out and manipulate the space-like solutions of the equations and to project them as visual displays.
[James P. Hogan, *The Genesis Machine* (1978)]

With basic constructions for interval space, interval relations and interval arithmetic developed, we are equipped with tools allowing for diagrammatic analysis of more advanced questions of interval analysis. In this section, a unified, diagrammatic approach to determine simultaneously all the basic solution sets of the interval linear equation of the form $\Sigma_{i=1}^{n} a_i \cdot x_i = b$ is presented. First, the correctness of the phrase "linear equation" is discussed. In consequence, the above formula is properly renamed as an *interval relational expression*, and its four basic solution sets Σ, $\Sigma_{\supseteq}$, $\Sigma_{\subseteq}$, and $\Sigma_{=}$ are defined.

Then, the simplest, one-dimensional interval relational expressions of the form $a \cdot x \Diamond b$, where $\Diamond \in \{\not\subseteq, \supseteq, \subseteq, =\}$ are analyzed in detail. The analysis also uses the *quotient sequence* concept, and leads to the full classification of possible structural cases and solution types.

Next, the analysis of the two-dimensional interval relational expressions $a_1 \cdot x_1 + a_2 \cdot x_2 \Diamond b$ and types of their solution sets are presented in detail. Finally, the generalization of the approach to n-dimensional interval systems and avenues for further research are outlined. Much of this material has been published in [4, 5, 26].

Some additional notation will be used in this section. Following [Markov 1995], Greek letters α, β, σ, etc. will denote *sign variables*, taking values from the set $\{-,+\}$. In sequences of signs and sign variables, commas will be omitted where it will not lead to confusion, hence $(+-+)$ and $(\alpha-\beta)$ are to be read as $(+,-,+)$ and $(\alpha,-,\beta)$, respectively. Angle brackets are used to force actual "multiplication" of signs, with standard rules: $\langle -+\rangle = \langle +-\rangle = -, \langle --\rangle = \langle ++\rangle = +$, while e.g. $(-+)$ denotes the sequence $(-,+)$. When used before an ordinary numerical variable or expression, the sign variable will be understood as simply a sign, hence $\sigma 1$ means 1 or -1, depending on whether $\sigma = +$ or $\sigma = -$, respectively. The symbol $\sigma \mathrm{O}x_i$ will denote the $\mathrm{O}x_i$ coordinate axis when $\sigma = +$, or the axis coinciding with $\mathrm{O}x_i$, but with opposite orientation, when $\sigma = -$. The convention $u^- = \underline{u}$, $u^+ = \overline{u}$ will be used as well.

III.5.1 Linear equations or relational expressions?

Although [we] have been testing and modifying them for decades, we can never be sure what the equations mean.
[Isaac Asimov, *Forward the Foundation* (1993)]

When coefficients of the matrices A and b in the system $A \cdot x = b$ are allowed to be intervals, it is usually still called an interval system of *linear equations* [Rohn 1989, Neumaier 1990]. Precisely speaking, however, it is no longer *linear*, and usually is not treated as a system of *equations* either. First, the space of real intervals is not a linear space, only a so-called *quasilinear* space, see e.g. [Markov 2001a]. For the purpose of this work, this is fortunately of secondary importance—at least the system has a *form* of a linear expression. More importantly, the use of the word *equation* is quite misleading here, as it is

only justified in the situation when one considers the *algebraic solution* (called also *formal solution* [Shary 2002]) to the system, which is rarely the case. This solution is defined as an interval x_{A} which fulfills the equation $A \cdot x_{\mathrm{A}} = b$ in the sense of interval arithmetic, see Section III.4.2.3. In most cases, other definitions of a solution are considered, usually as sets of *real* vectors (not necessarily intervals), defined as follows (see e.g. [Shary 1996]):

United Solution Set:

$$\Sigma(A,b) = \{x \in \mathbb{R}^n \,|\, A \cdot x \cap b \neq \emptyset\} = \\ = \{x \in \mathbb{R}^n \,|\, (\exists \tilde{A} \in A)(\exists \tilde{b} \in b)\tilde{A} \cdot x = \tilde{b}\} = \Sigma_{\exists\exists}(A,b),$$

Control Solution Set:

$$\Sigma_{\supseteq}(A,b) = \{x \in \mathbb{R}^n \,|\, A \cdot x \supseteq b\} = \\ = \{x \in \mathbb{R}^n \,|\, (\forall \tilde{b} \in b)(\exists \tilde{A} \in A)\tilde{A} \cdot x = \tilde{b}\} = \Sigma_{\exists\forall}(A,b),$$

Tolerance Solution Set:

$$\Sigma_{\subseteq}(A,b) = \{x \in \mathbb{R}^n \,|\, A \cdot x \subseteq b\} = \\ = \{x \in \mathbb{R}^n \,|\, (\forall \tilde{A} \in A)(\exists \tilde{b} \in b)\tilde{A} \cdot x = \tilde{b}\} = \Sigma_{\forall\exists}(A,b).$$

None of the above is actually a solution to the original equation. They are sets of real solutions to a system of interval *relational expressions*, with different relations put in the place of the equal sign, namely:

$A \cdot x \between b$ for the Σ set,

$A \cdot x \supseteq b$ for the $\Sigma_{\supseteq}$ set,

$A \cdot x \subseteq b$ for the $\Sigma_{\subseteq}$ set, respectively,

where $u \between v$ stands for $u \cap v \neq \emptyset$. Using this convention, the equation $A \cdot x = b$ would have a solution set $\Sigma_{=}$ equal to $\Sigma_{\supseteq} \cap \Sigma_{\subseteq}$, which is different than the algebraic solution. From the definitions it follows also that $\Sigma_{\subseteq} \subseteq \Sigma$ and $\Sigma_{\supseteq} \subseteq \Sigma$.

There is no established short name for this sort of a formula—in which not only an equality, but any other relation can stand as a main connective. The term *inequality* is not appropriate, as we do not want to exclude equality—it is as good a relation as any other to be used here. Therefore, in the sequel the phrase *relational expression* will be used, or sometimes simply *relation* for short.

III.5.2 The one-dimensional relational expression

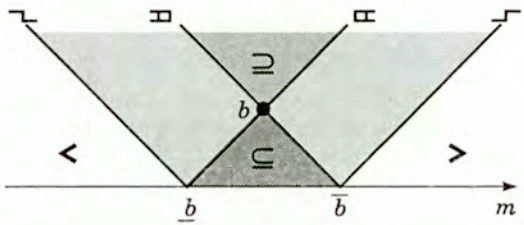
> You know of course that a mathematical line,
> a line of thickness nil, has no real existence ...
> [Herbert G. Wells, *The Time Machine* (1898)]

In the one-dimensional case, the matrix A shrinks to a single interval a, as does the vector b, hence the relational expression becomes simply $a \cdot x \lozenge b$, where $\lozenge \in \{\between, \supseteq, \subseteq, =\}$. Diagrammatic analysis of solution sets for this case is comparatively simple, but leads to very useful insights and constructions, as well as proves to be indispensable for the analysis of general multidimensional cases, see Section III.5.3.

III.5.2.1 Solving the relation diagrammatically

In the sequel we assume that both $a \neq 0$ and $b \neq 0$. The degenerate cases of $a = 0$ or $b = 0$ will be treated separately in Section III.5.2.5.

For the given interval a and all possible real x's, the products $a \cdot x$ represent simply the axis $\mathbf{O}\boldsymbol{a}$, see Section III.4.2.1. Thus, to find the solution set for the expression $a \cdot x \,\Diamond\, b$ for the given $\Diamond \in \{\mathfrak{X}, \supseteq, \subseteq, =\}$, one must first find the subset of $\mathbf{O}\boldsymbol{a}$ that stays in relation $\Diamond$ to b. This is obviously the intersection $\mathbf{O}\boldsymbol{a} \cap (\Diamond\, b)$ of $\mathbf{O}\boldsymbol{a}$ with the coimage $\Diamond\, b$. In Fig. III.46, the coimages of interval b under those interval relations that occur in the interval relational expressions of interest here are shown using the *W-diagram*, see Section III.3. The interval relations $\supseteq$, $\subseteq$, $\mathfrak{X}$, and $=$, as well as the border relations (none other than the full-line relations of Section III.3.4.1) $\urcorner$, $\sqcap$, $\sqcap$, and $\lrcorner$ are unions of the basic relations (relations $\urcorner$, $\lrcorner$, and $=$ are already basic, so they can be considered as "single component unions").



$\supseteq = \sqcap \cup = \cup \lrcorner \cup \sqcap$
$\subseteq = \sqcap \cup = \cup \lrcorner \cup \sqcap$
$\mathfrak{X} = \cup\,(BIR \setminus \{<, >\})$
$\sqcap = \urcorner \cup = \cup \sqcap$
$\sqcap = \sqcap \cup = \cup \lrcorner$

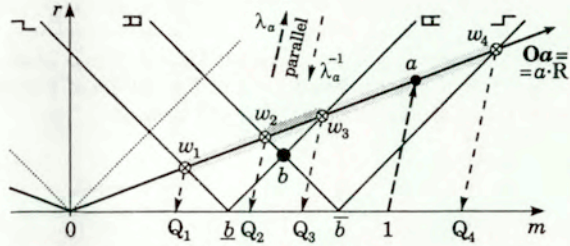
$\supseteq b = \{\, u \mid u \supseteq b \,\}$
$\subseteq b = \{\, u \mid u \subseteq b \,\}$
$\mathfrak{X} b = \{\, u \mid u \cap b \neq \varnothing \,\}$
$< b = \{\, u \mid \overline{u} < \underline{b} \,\}$
$> b = \{\, u \mid \underline{u} > \overline{b} \,\}$

Border relations:
$\urcorner\, b = \{\, u \mid \overline{u} = \underline{b} \,\}$
$\sqcap\, b = \{\, u \mid \overline{u} = \overline{b} \,\}$
$\sqcap\, b = \{\, u \mid \underline{u} = \underline{b} \,\}$
$\lrcorner\, b = \{\, u \mid \underline{u} = \overline{b} \,\}$

Figure III.46: Coimages of an interval b under certain set-theoretic interval relations.

Now, because $a \cdot x = \lambda_a(x)$, then $x = \lambda_a^{-1}(a \cdot x)$, and the solution set $\Sigma_\Diamond$ is a back projection of the intersection $\mathbf{O}\boldsymbol{a} \cap (\Diamond\, b)$ onto the $\mathbf{O}\mathbf{m}$ axis, see Fig. III.47 for several examples—three regular ones (with a without zero, types **Z**, **N**, and **U**), and two singular (with a containing zero, types **C** and **X**, see also Fig. III.50). Note that some solution sets in the singular case become so-called *extervals*, or *Kahan intervals*, see Section III.4.5. The configuration types are denoted by one-letter names; the reason for the choice of these particular letters will become clear in Section III.5.2.3 below.

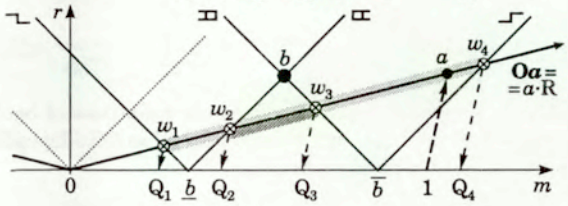
Because the coimage $(=b) = (\subseteq b) \cap (\supseteq b)$ contains only one point (namely b, see Fig. III.46), the set $\Sigma_=$ is nonempty only when the axis of a passes exactly through b, i.e., when $a \neq 0$ lies on the axis $\mathbf{O}\boldsymbol{b}$ of b, so that $|\mathbf{rex}\, a| = |\mathbf{rex}\, b|$ (a situation not shown in Fig. III.47). Hence, because $(=b)$ is a single point, the set $\Sigma_=$, when nonempty, contains also only a single number, namely $\lambda_a^{-1}(b)$.



Type Z

a, b without zero,
$0 < |\operatorname{rex} b| < |\operatorname{rex} a| < 1$

$\Sigma(a,b) = [Q_1, Q_4]$
$\Sigma_{\supseteq}(a,b) = [Q_2, Q_3]$
$\Sigma_{\subseteq}(a,b) = \emptyset$

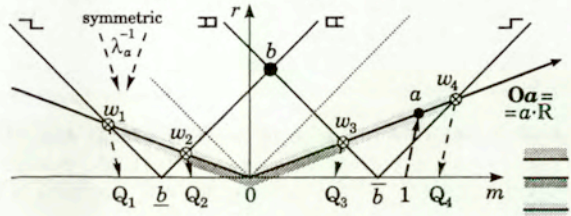


Type N

a, b without zero,
$0 < |\operatorname{rex} a| < |\operatorname{rex} b| < 1$

$\Sigma(a,b) = [Q_1, Q_4]$
$\Sigma_{\supseteq}(a,b) = \emptyset$
$\Sigma_{\subseteq}(a,b) = [Q_2, Q_3]$

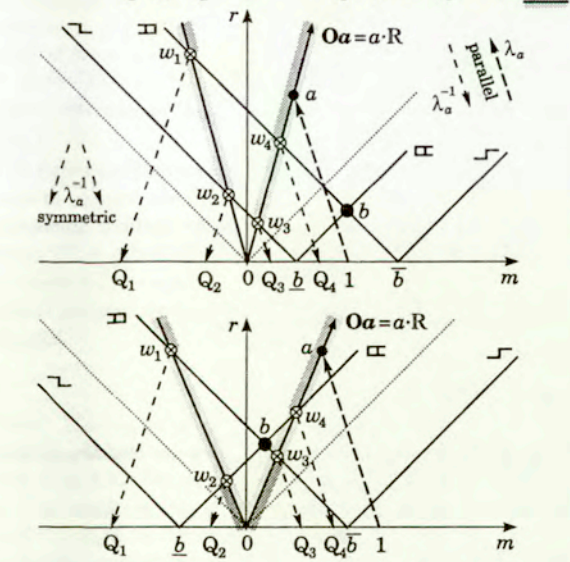
(The same for N and U)



Type U

a without zero,
b contains zero:
$0 < |\operatorname{rex} a| < 1 < |\operatorname{rex} b|$

$= (\supseteq b) \cap \mathbf{O}a$
$= (\subseteq b) \cap \mathbf{O}a$
$= (\mathfrak{X} b) \cap \mathbf{O}a$



Type C

a contains zero,
b without zero:
$0 < |\operatorname{rex} b| < 1 < |\operatorname{rex} a|$

$\Sigma(a,b) =]Q_2, Q_3[$
$\Sigma_{\supseteq}(a,b) =]Q_1, Q_4[$
$\Sigma_{\subseteq}(a,b) = \emptyset$

External notation:
$]u,v[= \langle -\infty, u] \cup [v, +\infty \rangle$

Type X

a, b contain zero:
$1 < |\operatorname{rex} a|, |\operatorname{rex} b|$

$\Sigma(a,b) = \mathbb{R}$
$\Sigma_{\supseteq}(a,b) =]Q_1, Q_4[$
$\Sigma_{\subseteq}(a,b) = [Q_2, Q_3]$

Figure III.47: Diagrammatic solutions of the $a \cdot x \Diamond b$ relation: regular cases (types Z, N, and U), and singular cases (types C and X).

III.5.2.2 Quotient sequences

The endpoints of the solution sets are thus given by the points $Q_i = \lambda_a^{-1}(w_i)$, where w_i denotes the corresponding intersection point (marked with $\otimes$ in Fig. III.47) of $\mathbf{O}\boldsymbol{a}$ with the coimage of one of the border relations. Thus, $w_i = Q_i \cdot a$, and then:

$$w_i = [\underline{w}_i, \overline{w}_i] = \begin{cases} [Q_i\underline{a}, Q_i\overline{a}] & \text{for } Q_i \geq 0, \\ [Q_i\overline{a}, Q_i\underline{a}] & \text{for } Q_i \leq 0. \end{cases}$$

Therefore:

$$Q_i = \begin{cases} \underline{w}_i/\underline{a} = \overline{w}_i/\overline{a} & \text{for } Q_i \geq 0, \\ \underline{w}_i/\overline{a} = \overline{w}_i/\underline{a} & \text{for } Q_i \leq 0. \end{cases} \tag{III.30}$$

It is now clear why the numbers Q_i are called *quotients*. From the definitions of border relations $\barwedge$, $\sqsupset$, $\sqsubset$, and $\barvee$, see Figs. III.46 and III.47, we get values for $\underline{w}_i$ and $\overline{w}_i$, depending on the border relation to whose coimage the given w_i belongs:

$$\underline{w}_i = \begin{cases} \underline{b} & \text{for } w_i \sqsubset b, \\ \overline{b} & \text{for } w_i \barvee b; \end{cases} \quad \text{and} \quad \overline{w}_i = \begin{cases} \underline{b} & \text{for } w_i \barwedge b, \\ \overline{b} & \text{for } w_i \sqsupset b. \end{cases} \tag{III.31}$$

This leads to the formulation of:

Lemma III.1 (Calculating quotients) *Depending on the relation coimage intersected by* $\mathbf{O}\boldsymbol{a}$ *at* w_i *and the sign of the respective quotient* $Q_i(a,b) = \lambda_a^{-1}(w_i)$, *the quotient is obtained as* $Q^{\beta\alpha}(a,b) = b^\beta/a^\alpha$, *where* $\beta, \alpha \in \{-,+\}$, *and* $u^- = \underline{u}$, $u^+ = \overline{u}$, *according to the rule:*

	border relation			
$\operatorname{sgn} Q_i$	$\barwedge$	$\sqsupset$	$\sqsubset$	$\barvee$
$+$	S	T	L	Z
$-$	L	Z	S	T

where the shorthands L, S, Z, *and* T *for the different kinds of quotients (chosen to mimic the graphical structure formed by dashes and division operator in the quotient expressions) are defined as:*

$$\begin{aligned} \mathrm{L} &= Q^{--} = \underline{b}/\underline{a}, \\ \mathrm{S} &= Q^{-+} = \underline{b}/\overline{a}, \\ \mathrm{Z} &= Q^{+-} = \overline{b}/\underline{a}, \\ \mathrm{T} &= Q^{++} = \overline{b}/\overline{a}. \end{aligned}$$

As in the above, the arguments (a,b) *will be usually omitted if it does not lead to ambiguity.*

Proof. Immediate from (III.30) and (III.31). □

Definition III.19 (Characteristic quotient sequence) *The sequence of all four possible quotients* $\mathbb{Q}(a,b) = Q_1Q_2Q_3Q_4$, $Q_i \in \{\mathrm{L}, \mathrm{S}, \mathrm{Z}, \mathrm{T}\}$ *ordered so that* $Q_1 \leq Q_2 \leq Q_3 \leq Q_4$, *is the* characteristic quotient sequence *for the given* a *and* b.

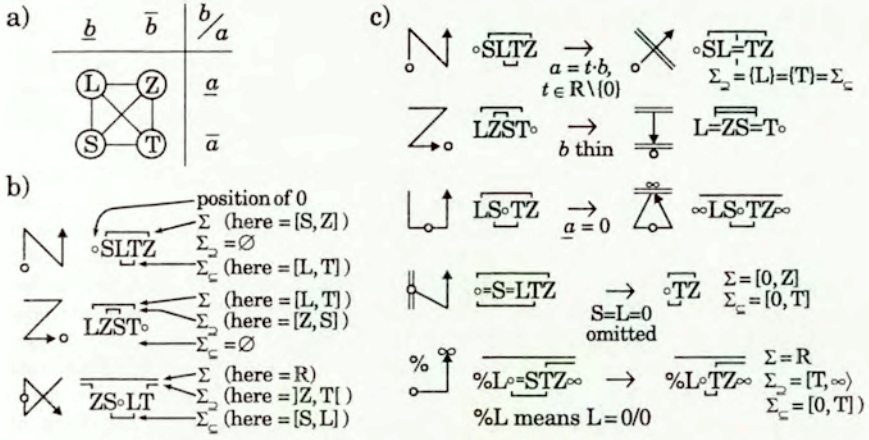


Figure III.48: The quotient diagram (a) and quotient sequence notation with example quotient sequence diagrams (b, c).

The significance of quotient sequences (and analogical objects defined for other interval arithmetic operations besides division) was pointed out by [Markov 1995]. The four quotients can be neatly arranged into a *quotient diagram*, see Fig. III.48a. Then, quotient sequences can be represented diagrammatically too, as examples in Fig. III.48 show. The basic sequence can be, when needed, augmented by an indication of the structure of the solution sets, Fig. III.48b, as well as of the position of 0 (with the "o" symbol), of the equality of some quotients (the "=" symbol), of running of some quotient to infinity (the "∞" symbol), or of the occurrence of an indefinite operation 0/0 (the "%" symbol), see for more examples of the notation in Fig. III.48c, as well as in Fig. III.49 and especially in Fig. III.51.

The following Lemma states an important property of quotient sequences:

Lemma III.2 (Radial move invariance) *The quotient sequence $\mathbb{Q}(a, b)$ remains the same (including its position relative to zero) when a, b, or both, move along their respective (positive) interval semi-axes (excluding the origin $[0, 0]$), i.e., when* **rex** *a and* **rex** *b remain unchanged.*

Proof. Moving an interval along its positive semi-axis (i.e., that on which the interval actually is placed) is equivalent to multiplication of the interval by some $t \in \mathbb{R}^+$, cf. Section III.4.2.1. For such t we have $t \cdot u = [t \cdot \underline{u}, t \cdot \overline{u}]$. Therefore, for every quotient $Q_i = b^\beta / a^\alpha$ we have $Q_i(t \cdot a, b) = Q_i(a, b)/t$, $Q_i(a, t \cdot b) = Q_i(a, b) \cdot t$, and $Q_i(t_1 \cdot a, t_2 \cdot b) = Q_i(a, b) \cdot (t_2/t_1)$. As the order of the quotients and their relation to zero does not change after dividing or multiplying all quotients by the same positive number, the thesis follows immediately. □

III.5.2.3 Basic solution types

The solutions are all simple—
after you have arrived at them.
[R.M. Pirsig, *Zen and the Art of Motorcycle Maintenance* (1974)]

Drawing diagrammatic constructions to solve various configurations of the W-diagram centred at b and the axis $\mathbf{O}a$ of a, one may find that there are five basic configuration types, as shown by the examples in Fig. III.47. Inspecting various combinations of signs (and extents) of the intervals involved, and actually determining the characteristic quotient sequences for them, one can find that the set of basic structural types further subdivides into a number of subtypes, now uniquely characterized by their characteristic quotient sequences. Figure III.49 lists a complete catalogue of 5 *basic types* and 16 *subtypes* possible. The quotient sequences are augmented with indications of the intervals (or "extervals") defining various types of solution sets (see the explanation of this notation in Fig. III.48). The quotient sequence diagrams are shown as well. The origin of the one-letter type names becomes clear now—they were chosen to mimic the shape of corresponding quotient sequence diagrams.

In all the basic cases $\Sigma_{=} = \emptyset$, because in all the corresponding diagrams in Fig. III.47 the axis $\mathbf{O}a$ never passes through the coimage $(=b) = \{b\}$ (as reflected in Fig. III.49 by the fact that all conditions exclude equality of $|\mathbf{rex}\, a|$ and $|\mathbf{rex}\, b|$). When $|\mathbf{rex}\, a| = |\mathbf{rex}\, b|$, i.e., when the axis $\mathbf{O}a$ passes through b, we do not obtain one of the basic types listed, but some *intermediate* (or *degenerate*) type, constituting a border case between a pair of basic types, see Sections III.5.2.5 and III.5.2.7 below. Intermediate types occur also for a and b lying on the **Or** or **Om** axes, or on the main diagonals **lb0** or **ub0**. All these cases are also explicitly excluded by the conditions given in Fig. III.49.

The formal justification of these results is provided by the following Theorem.

Theorem III.5 (Basic solution types) *For all possible basic combinations of values of the coefficients a and b (fulfilling the conditions for basic cases listed in Fig. III.49), the quotient sequences $\mathbb{Q}(a, b)$ corresponding to them are those listed in Fig. III.49.*

Proof. As follows from Lemma III.2, only the change of $\mathbf{rex}\, a$ or $\mathbf{rex}\, b$ may change a type of solution. As it is seen from the conditions given in Fig. III.49, all possible basic combinations (excluding the intermediate ones) of values of $\mathbf{rex}\, a$ and $\mathbf{rex}\, b$ are exhausted there (see also Figs. III.53 and III.54 in Section III.5.2.6 for another, diagrammatic way of enumerating all cases).

What thus remains to show is that, for all subtypes listed in Fig. III.49, the quotient sequence listed indeed corresponds to the conditions on a and b given for the subtype. Let us show the appropriate reasoning for two of the subtypes.

Type $\mathbf{Z}_{++}$: Inspecting the diagram of Fig. III.47 **Z** which corresponds to that subtype, one may see that wherever one moves the points representing intervals a and b, provided they do not move outside their allowed regions (i.e., both a and b are positive and b always stays below the $\mathbf{O}a$ axis), then neither the signs of the Q_i points, nor the correspondence between them and the intersections w_i with border relations

Regular cases: $|\operatorname{rex} a| < 1$ (i.e., $0 \notin a$)

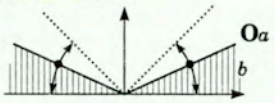
Type Z: $0 < |\operatorname{rex} b| < |\operatorname{rex} a| < 1$

$\check{a} < 0, \check{b} > 0$ $\quad$ $\check{a} > 0, \check{b} > 0$

$\mathbf{Z}_{-+}$ TSZL $\quad$ $\mathbf{Z}_{++}$ STLZ

$\check{a} < 0, \check{b} < 0$ $\quad$ $\check{a} > 0, \check{b} < 0$

$\mathbf{Z}_{--}$ ZLTS $\quad$ $\mathbf{Z}_{+-}$ LZST



$\Sigma(a,b) = [Q_1, Q_4]$
$\Sigma_{\supset}(a,b) = [Q_2, Q_3]$
$\Sigma_{\subset}(a,b) = \varnothing$

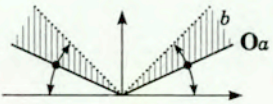
Type N: $0 < |\operatorname{rex} a| < |\operatorname{rex} b| < 1$

$\check{a} < 0, \check{b} > 0$ $\quad$ $\check{a} > 0, \check{b} > 0$

$\mathbf{N}_{-+}$ TZSL $\quad$ $\mathbf{N}_{++}$ SLTZ

$\check{a} < 0, \check{b} < 0$ $\quad$ $\check{a} > 0, \check{b} < 0$

$\mathbf{N}_{--}$ ZTLS $\quad$ $\mathbf{N}_{+-}$ LSZT



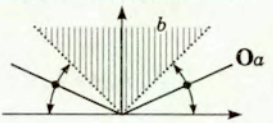
$\Sigma(a,b) = [Q_1, Q_4]$
$\Sigma_{\supset}(a,b) = \varnothing$
$\Sigma_{\subset}(a,b) = [Q_2, Q_3]$

(The same for types **N** and **U**)

Type U: $0 < |\operatorname{rex} a| < 1 < |\operatorname{rex} b|$

$\check{a} < 0$ $\quad$ $\check{a} > 0$

$\mathbf{U}_{-}$ TZ∘LS $\quad$ $\mathbf{U}_{+}$ LS∘TZ

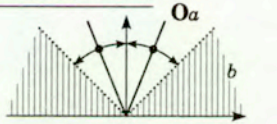


Singular cases: $|\operatorname{rex} a| \geq 1$ (i.e., $0 \in a$)

Type C: $0 < |\operatorname{rex} b| < 1 < |\operatorname{rex} a|$

$\check{b} < 0$ $\quad$ $\check{b} > 0$

$\mathbf{C}_{-}$ ST∘ZL $\quad$ $\mathbf{C}_{+}$ ZL∘ST



$\Sigma(a,b) =]Q_2, Q_3[$
$\Sigma_{\supset}(a,b) =]Q_1, Q_4[$
$\Sigma_{\subset}(a,b) = \varnothing$

Exterval notation:
$]u,v[= \langle -\infty, u] \cup [v, +\infty \rangle$

Type X: $1 < |\operatorname{rex} a|, |\operatorname{rex} b|$

$|\operatorname{rex} a| < |\operatorname{rex} b|$

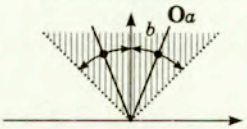
$\check{a} < 0$ $\quad$ $\check{a} > 0$

$\mathbf{X}_{a-}$ SZ∘LT $\quad$ $\mathbf{X}_{a+}$ ZS∘TL

$|\operatorname{rex} b| < |\operatorname{rex} a|$

$\check{b} < 0$ $\quad$ $\check{b} > 0$

$\mathbf{X}_{b-}$ SZ∘TL $\quad$ $\mathbf{X}_{b+}$ ZS∘LT



$\Sigma(a,b) = \mathbb{R}$
$\Sigma_{\supset}(a,b) =]Q_1, Q_4[$
$\Sigma_{\subset}(a,b) = [Q_2, Q_3]$

Figure III.49: A catalogue of *basic* types and subtypes of solutions to the $a \cdot x \Diamond b$ relation; the defining conditions for the (sub)types explicitly exclude the intermediate cases.

change. Hence, the quotient sequence obtained using the table in Lemma III.1 does not change and is as prescribed in Fig. III.49 for this case.

Alternatively, the same can be shown with formulae. First, in this case obviously $0 < \underline{a} < \overline{a}$ and $0 < \underline{b} < \overline{b}$. Therefore, all quotients are positive, and the inequalities $L < Z$, $S < T$, $S < L$, and $T < Z$ may be easily derived (e.g., dividing the $\underline{b} < \overline{b}$ inequality by $\underline{a}$ we get $\underline{b}/\underline{a} < \overline{b}/\underline{a}$, that is, $L < Z$, etc.). The obtained conditions are, however, not sufficient to determine the exact sequence of quotients (as the ordering of T and L remains undetermined). However, from the condition $|\mathbf{rex}\, a| < |\mathbf{rex}\, b|$ it follows, after some transformations, that $\overline{b}\,\underline{a} < \underline{b}\,\overline{a}$, hence $T < L$, which finishes the determination of the sequence as $0 < S < T < L < Z$, as required.

Type $\mathbf{U_+}$: Diagrammatic argument goes here in the same way; one should only use the diagram in Fig. III.47**U** corresponding to this subtype. Using formulae, we have obviously $a > 0$, hence $0 < \underline{a} < \overline{a}$. Next, $1 < |\mathbf{rex}\, b|$, which, after considering two possible cases for the sign of $\breve{b}$, boils down to $\underline{b} < 0 < \overline{b}$. From the above, inequalities $L < S < 0$, and $0 < T < Z$ are easily derived, like in the previous case, which gives just the quotient sequence listed for this type.

Similar simple reasoning can be easily conducted for all other cases listed in Fig. III.49 in order to conclude the proof of the theorem. □

III.5.2.4 Other characterizations of solution sets

For the one-dimensional case, some solution sets coincide with certain simple expressions or other solutions, as shown by the following series of theorems and propositions.

Theorem III.6 ($\Sigma(a,b)$ and division) *The united solution set coincides with the interval division b/a (the Kahan, or exterval division when a contains zero, see Section III.4.5), i.e., $\Sigma(a,b) = b/a$.*

Proof. From the definition of interval arithmetic operations, see (III.5), we have:

$$b/a = \{\tilde{b}/\tilde{a} \mid \tilde{a} \in a \text{ and } \tilde{b} \in b\}.$$

Substituting $\tilde{b}/\tilde{a} = q \in \mathbb{R}$, we get, see also Figs. III.46 and III.47:

$$\begin{aligned} b/a &= \{q \mid \tilde{a} \in a \text{ and } q \cdot \tilde{a} \in b,\} = \{q \mid q \cdot a \cap b \neq \emptyset\} = \\ &= \{q \mid \lambda_a(q) \between b\} = \lambda_a^{-1}(\mathbf{O}a \cap (\between b)) = \\ &= \Sigma(a,b). \end{aligned}$$

The reasoning remains valid for a containing zero—informally, we can assume $q = \pm\infty$ when $\tilde{a} = 0$ and proceed with such extended real arithmetic appropriately; a diagrammatic counterpart of the last steps of the derivation will use "extervals," in the manner shown in Fig. III.47 for the singular cases. It can be made formal by a (tedious) passing to the limit; an interval formalism to handle such cases, based on the Kahan arithmetic, see Section III.4.5, was proposed in [Kaucher 1977]. □

Another diagram-aided approach to prove the theorem, though basically valid only for a without zero, was used in [7, 105], see Section III.4.3.

Theorem III.7 ($\Sigma_{\supseteq}(a,b)$, $\Sigma_{\subseteq}(a,b)$ **and inner division**) *If a does not contain zero, $\Sigma_{\supseteq}(a,b) \cup \Sigma_{\subseteq}(a,b) = b/^{-}a$, where $b/^{-}a$ denotes the* inner division *of b and a* [Markov 1995]. *In alternative formulation, if $0 \notin a$, this set from among $\Sigma_{\supseteq}(a,b)$ and $\Sigma_{\subseteq}(a,b)$ that is nonempty is equal to $b/^{-}a$.*

Proof. The inner division $b/^{-}a$ for $0 \notin a$ is defined by [Markov 1995] in terms of quotients, for the case when all quotients are different, as $b/^{-}a = [Q_2, Q_3]$. Because for basic types the quotients are always distinct, and for $0 \notin a$ either $\Sigma_{\supseteq}(a,b) = [Q_2, Q_3]$ and $\Sigma_{\subseteq}(a,b) = \emptyset$, or else $\Sigma_{\subseteq}(a,b) = [Q_2, Q_3]$ and $\Sigma_{\supseteq}(a,b) = \emptyset$, the thesis follows immediately for this case.

The definition of inner division can be naturally extended to those non-basic (intermediate) cases for which $0 \notin a$, see Fig. III.51. For them some quotients become equal, but irrespectively of the ordering of the equal quotients in the sequence, the interval $[Q_2, Q_3]$ remains the same. Additionally, even when in some of those cases both $\Sigma_{\supseteq}(a,b)$ and $\Sigma_{\subseteq}(a,b)$ are nonempty, they are then equal, so that both $\Sigma_{\supseteq}(a,b) = \Sigma_{\supseteq}(a,b) \cup \Sigma_{\subseteq}(a,b)$ and $\Sigma_{\subseteq}(a,b) = \Sigma_{\supseteq}(a,b) \cup \Sigma_{\subseteq}(a,b)$. Thus, the thesis is valid for all cases with $0 \notin a$. □

Let us now consider the relations between the tolerance solution set $\Sigma_{\subseteq}(a,b)$ and the algebraic solution x_{A} (see Section III.5.1) of the equation $a \cdot x = b$, see [7, 105] and Section III.4.2.3.

Proposition III.7 ($\Sigma_{\subseteq}(a,b)$ **and** x_{A}**: emptiness**) *If the tolerance solution set $\Sigma_{\subseteq}(a,b)$ is empty, the algebraic solution x_{A} does not exist.*

Proof. The set $\Sigma_{\subseteq}(a,b)$ is empty in the cases **Z** and **C**, where the axis $\mathbf{O}a$ does not intersect the set ($\subseteq b$) of intervals contained in b, see Figs. III.47 and III.49. In both cases, $|\mathbf{rex}\, a| > |\mathbf{rex}\, b|$, which is the condition for nonexistence of the algebraic solution x_{A}, cf. Section III.4.2.3. □

The opposite implication does not hold, see Proposition III.10 below.

Proposition III.8 ($\Sigma_{\subseteq}(a,b)$ **and** x_{A}**: equality**) *If the algebraic solution x_{A} exists and is unique, it equals the tolerance solution set $\Sigma_{\subseteq}(a,b)$.*

Proof. The condition for existence and uniqueness of x_{A} is $|\mathbf{rex}\, a| < |\mathbf{rex}\, b|$, cf. Section III.4.2.3. This corresponds to the cases of the **N** and **U** types, and the $\mathbf{X}_{a+}$ and $\mathbf{X}_{a-}$ subtypes, see Figs. III.47 and III.49. The diagrams in Fig. III.47 for these cases demonstrate the thesis easily. In all of them, $\Sigma_{\subseteq}(a,b) = [Q_2, Q_3]$, and because $w_i = \lambda_a(Q_i)$, we have immediately $a \cdot \Sigma_{\subseteq} = w_2 \vee w_3 = b$ (see Section III.4.2). The diagrams in Fig. III.47 cover three of the 8 (sub)cases involved; the other cases can be demonstrated analogically after constructing similar diagrams for them. Note also that $w_2 \vee w_3 \neq b$ for other diagrams in Fig. III.47.

To conduct the proof using formulae, one should again consider several cases, corresponding to the 8 subtypes concerned. We will restrict the proof to three of them, corresponding to the same cases as above. To simplify the argument, we will use without proof the specific formulae for multiplication of intervals of the required types (see e.g. [Neumaier 1990]).

Type N_{++}: In this case $a > 0$ and $\Sigma_{\subseteq} = [\mathrm{L}, \mathrm{T}] > 0$ as well (see Fig. III.49 **N**). Therefore:

$$a \cdot \Sigma_{\subseteq} = [\underline{a}, \overline{a}] \cdot [\mathrm{L}, \mathrm{T}] = [\underline{a}\mathrm{L},\ \overline{a}\,\mathrm{T}] = [\underline{a}\,\underline{b}/\underline{a},\ \overline{a}\,\overline{b}/\overline{a}] = [\underline{b}, \overline{b}] = b, \text{ as required.}$$

Type U_{+}: Now $a > 0$, but $\Sigma_{\subseteq} = [\mathrm{S}, \mathrm{T}]$ contains zero (see Fig. III.49 **U**), therefore:

$$a \cdot \Sigma_{\subseteq} = [\underline{a}, \overline{a}] \cdot [\mathrm{S}, \mathrm{T}] = [\overline{a}\,\mathrm{S},\ \overline{a}\,\mathrm{T}] = [\overline{a}\,\underline{b}/\overline{a},\ \overline{a}\,\overline{b}/\overline{a}] = [\underline{b}, \overline{b}] = b, \text{ as required.}$$

Type X_{a+}: Here both a and $\Sigma_{\subseteq} = [\mathrm{S}, \mathrm{T}]$ contain zero, hence:

$$\begin{aligned} a \cdot \Sigma_{\subseteq} &= [\underline{a}, \overline{a}] \cdot [\mathrm{S}, \mathrm{T}] = [\min\{\underline{a}\,\mathrm{T},\ \overline{a}\,\mathrm{S}\}, \max\{\underline{a}\,\mathrm{S},\ \overline{a}\,\mathrm{T}\}] = \\ &= [\min\{\underline{a}\,\overline{b}/\overline{a},\ \overline{a}\,\underline{b}/\overline{a}\}, \max\{\underline{a}\,\underline{b}/\overline{a},\ \overline{a}\,\overline{b}/\overline{a}\}] = \\ &= [\min\{\underline{a}\,\overline{b}/\overline{a},\ \underline{b}\}, \max\{\underline{a}\,\underline{b}/\overline{a},\ \overline{b}\}]. \end{aligned}$$

Next, as for this subtype $\mathbf{rex}\ a$, $\mathbf{rex}\ b > 0$ and $|\mathbf{rex}\, a| < |\mathbf{rex}\, b|$, therefore $\mathbf{rex}\ a < \mathbf{rex}\ b$ as well. Also, $\overline{a} + \underline{a} > 0$, $\overline{b} + \underline{b} > 0$, and $\overline{a} > 0$. Thus:

$$\begin{aligned} \mathbf{rex}\ a = (\overline{a} - \underline{a})/(\overline{a} + \underline{a}) &< (\overline{b} - \underline{b})/(\overline{b} + \underline{b}) = \mathbf{rex}\ b, \\ (\overline{a} - \underline{a})(\overline{b} + \underline{b}) &< (\overline{a} + \underline{a})(\overline{b} - \underline{b}), \\ \overline{a}\,\underline{b} - \underline{a}\,\overline{b} &< \underline{a}\,\overline{b} - \overline{a}\,\underline{b}, \\ \overline{a}\,\underline{b} &< \underline{a}\,\overline{b}, \\ \underline{b} &< \underline{a}\,\overline{b}/\overline{a}, \end{aligned}$$

that is $\min\{\underline{a}\,\overline{b}/\overline{a},\ \underline{b}\} = \underline{b}$.

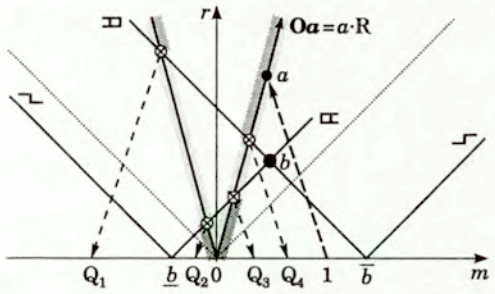
Now, $-\underline{a} < \overline{a}$, $-\underline{b} < \overline{b}$, and both sides of these inequalities are positive, thus we can multiply them side by side obtaining $\underline{a}\,\underline{b} < \overline{a}\,\overline{b}$, therefore $\underline{a}\,\underline{b}/\overline{a} < \overline{b}$ and finally $\max\{\underline{a}\,\underline{b}/\overline{a},\ \overline{b}\} = \overline{b}$. Combining the above:

$$a \cdot \Sigma_{\subseteq} = [\min\{\underline{a}\,\overline{b}/\overline{a},\ \underline{b}\}, \max\{\underline{a}\,\underline{b}/\overline{a},\ \overline{b}\}] = [\underline{b}, \overline{b}] = b, \text{ as required.}$$

The argument goes in a similar way for other cases as well. □

Proposition III.9 ($\Sigma_{\subseteq}(a,b)$ **and** x_{A}**: non-uniqueness)** *If the algebraic solution x_{A} is not unique, i.e. there is a whole set $X_{=}$ of intervals fulfilling the equation, the tolerance solution set $\Sigma_{\subseteq}(a,b)$ equals the maximal, with respect to inclusion, element of the set $X_{=}$.*

Proof. The condition for non-uniqueness of the x_{A} solution is $|\mathbf{rex}\, a| = |\mathbf{rex}\, b| \geq 1$, see Proposition III.5 in Section III.4.2.3. Thus, both a and b contain zero and lie on the same interval axis, which signifies an intermediate solution type (see Section III.5.2.5 for further discussion of such cases). For such a and b, there are four such types, intermediate between appropriate **X** subtypes: $\mathsf{X}_{a+}|\mathsf{X}_{b+}$, $\mathsf{X}_{a+}|\mathsf{X}_{b-}$, $\mathsf{X}_{a-}|\mathsf{X}_{b+}$, and $\mathsf{X}_{a-}|\mathsf{X}_{b-}$. Let us consider the first of them (compare Fig. III.47 **X** with Fig. III.50); the remaining ones can be analyzed analogically. In this case, we have obviously L = T: to show that, move the interval b in Fig. III.47 **X** (or in Fig. III.50) to put it on the axis $\mathbf{O}a$ and observe that $\mathrm{Q}_3 = \mathrm{Q}_4$, or alternatively, use the fact that $\mathbf{rex}\ a = \mathbf{rex}\ b$, so that $b = ta$, $t \in \mathbb{R}^+$, to obtain $\mathrm{L} = \mathrm{T} = t$. Therefore, the quotient sequence for this type is ZS∘L=T, thus $\Sigma_{\subseteq}(a,b) = [\mathrm{S}, \mathrm{L}] = [\underline{b}/\overline{a}, \underline{b}/\underline{a}] = [\mathrm{S}, \mathrm{T}] = [\underline{b}/\overline{a}, \overline{b}/\overline{a}]$. Now, for $\check{a} > 0$, the maximal element of the set $X_{=}$ equals $x_a = b/\overline{a}$, see the formula (III.29) in Section III.4.2.3. Thus, $x_a = b/\overline{a} = [\underline{b}/\overline{a}, \overline{b}/\overline{a}] = [\mathrm{S}, \mathrm{T}] = \Sigma_{\subseteq}(a,b)$, as required. □



Subtype $\mathbf{X}_{b+}$

a, b contain zero:
$|\mathrm{rex}\, a| > |\mathrm{rex}\, b| > 1$;
$\check{a} > 0$ & $\check{b} > 0$

$\Sigma(a,b) = \mathbb{R}$
$\Sigma_{\supset}(a,b) =]Q_1, Q_4[$
$\Sigma_{\subseteq}(a,b) = [Q_2, Q_3] \neq \emptyset$,
but x_{A} does not exist

Figure III.50: The $\mathbf{X}_{b+}$ subtype diagram: inequality of $\Sigma_{\subseteq}(a,b)$ and x_{A}.

Proposition III.10 ($\Sigma_{\subseteq}(a,b)$ **and** x_{A}**: inequality)** *For subtypes* $\mathbf{X}_{b+}$ *and* $\mathbf{X}_{b-}$*, i.e., if both* a *and* b *contain zero and* $|\mathbf{rex}\, a| > |\mathbf{rex}\, b|$*, the set* $\Sigma_{\subseteq}(a,b)$ *is nonempty, but* x_{A} *does not exist, hence* $\Sigma_{\subseteq}(a,b) \neq x_{\mathrm{A}}$.

Proof. Immediate from Fig. III.49**X**: the set $\Sigma_{\subseteq}(a,b)$ is nonempty in the cases given in the theorem, while x_{A} does not exist, because $|\mathbf{rex}\, a| > |\mathbf{rex}\, b|$. Fig. III.50 shows one of the relevant configurations: $a \cdot \Sigma_{\subseteq}$ is equal to the point $w_3 = \lambda_a(Q_3)$, marked by ⊠, and certainly different from b. Using formulae, and applying the appropriate multiplication rule for this case, we have:

$$a \cdot \Sigma_{\subseteq} = [\underline{a}, \overline{a}] \cdot [\mathrm{S}, \mathrm{L}] = [\underline{a}, \overline{a}] \cdot \mathrm{L} = [\underline{a}, \overline{a}] \cdot \underline{b}/\underline{a} = [\underline{b}, \overline{a}\,\underline{b}/\underline{a}] \neq b,$$

so indeed $\Sigma_{\subseteq}$ is not a solution to the equation in this case. □

The above propositions are no longer true for the multidimensional case—by definition, x_{A}, if it exists, is an interval, while $\Sigma_{\subseteq}(A,b)$ usually is not. Propositions III.8 and III.9 hold in the weaker sense of inclusion ($x_{\mathrm{A}} \subseteq \Sigma_{\subseteq}(A,b)$), see e.g. [Shary 2002].

III.5.2.5 The MR-diagram representation and intermediate types

The enumeration of solution types can be also done explicitly in appropriate MR-diagrams (larger versions of the small configuration diagrams in Fig. III.49). This representation is especially convenient for the study of multidimensional cases, as will be shown in Section III.5.3. It also allows for representation of intermediate types mentioned before.

Besides the basic types and subtypes listed in Fig. III.49, there is a number of intermediate (or degenerate) cases, when one (or both) intervals a and b happen to lie on a main diagonal, or on the **Om** axis (when they become thin intervals), or on the same interval axis. In quotient sequences for such cases some quotients coincide or run off to infinity, but the pattern of solution sets remains essentially the same.

In Fig. III.51, all basic and intermediate types are shown as labels of the regions (for basic (sub)types), or labels of the lines (for intermediate types) in, or on which, the coefficient a lies for the given type. Types are indicated by their quotient sequences; for basic subtypes the symbolic names are also used. Diagrams in Fig. III.51a-d show the types for $\check{b} > 0$, while those in Fig. III.51e-i depict the case $\check{b} \leq 0$. In all diagrams, except Fig. III.51g, the quotient sequence placed over the **Or** axis refers to the whole region

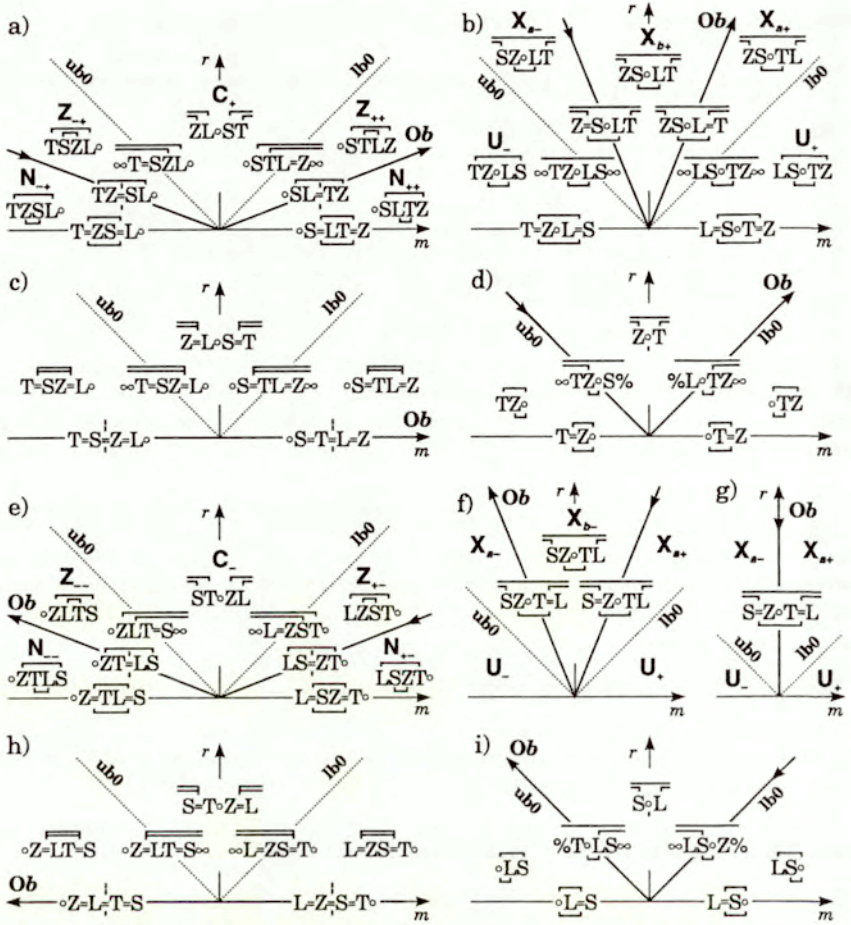


Figure III.51: The MR-diagram catalogue of types and subtypes (including intermediate ones).

either above the Ob axis, or above the main diagonals **ub0** and **lb0**, whichever comes higher (including the **Or** axis as well). In Fig. III.51g, representing the case of symmetric b (i.e. $\breve{b} = 0$), the indicated quotient sequence concerns only the a coefficients lying on the **Or** axis. In Fig. III.51f, g, the regions below the Ob axis (not labelled by quotient sequences) have the types identical as in Fig. III.51b. Quotient sequence diagrams are omitted for clarity. They may be easily reconstructed, if need arises, using the examples shown in Fig. III.48 as guides. Indications of the values of the solution sets should be clear after the examples shown in Fig. III.48. Additionally, a short vertical line denotes a one-element set containing the indicated value.

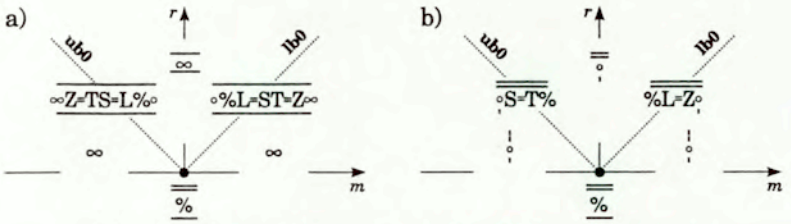


Figure III.52: The MR-diagram catalogue of degenerate types for $a = 0$, i.e. $0 \cdot x = b$ (a), and $b = 0$, i.e. $a \cdot x = 0$ (b).

For completeness, the remaining degenerate cases (for $a = 0$, or $b = 0$) are catalogued in Fig. III.52. For $a = 0$, the regions of the diagram represent different possible values of b, while for $b = 0$ the regions represent possible values of a. Positioning a sequence above the gap in the **Om** axis means that it is valid for both thin and thick intervals (on and above the axis) lying below the main diagonals. A quotient sequence containing only a single symbol ("$\circ$", "∞", or "%") means that all quotients are equal to it. For example, the sequence $\overline{\infty}$ means that all quotients are infinite, $\Sigma(0,b) = \Sigma_{\subseteq}(0,b) = \mathbb{R}$ and $\Sigma_{\supseteq}(0,b) = \emptyset$, whereas the sequence $\overline{\overline{\circ}\ \mathrm{S{=}T\%}}$ means $\mathrm{Z} = \mathrm{L} = 0$, $\mathrm{S} = \mathrm{T} = 0/0$, $\Sigma(a,0) = \Sigma_{\supseteq}(a,0) = \mathbb{R}$, and $\Sigma_{\subseteq}(a,0) = \{0\}$. The derivation of the solution sets for the case $a = 0$ assumes the convention $0 \cdot \infty = 0$.

The validity of the above catalogue of intermediate and degenerate cases can be formally justified in a similar way as shown in the proof of Theorem III.5.

III.5.2.6 RR-diagrams and graphs of types

Another useful means of representing the structure of the set of types is provided by an *RR-diagram*. It comprises a two-dimensional coordinate system with values of **rex** a and

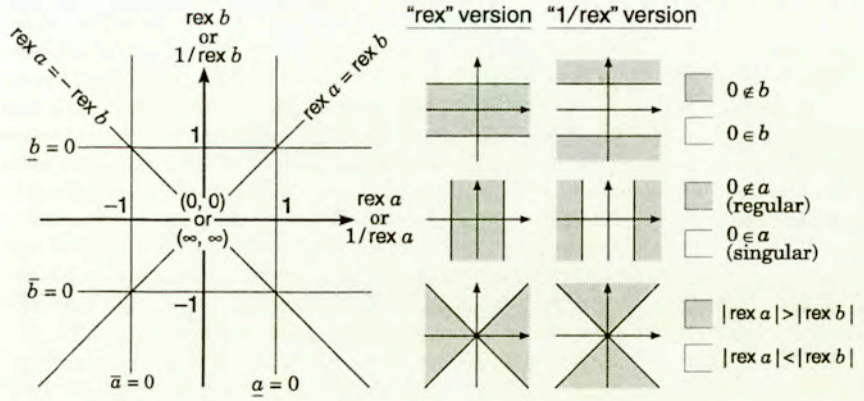


Figure III.53: The structure of an RR-diagram and a 1/RR-diagram.

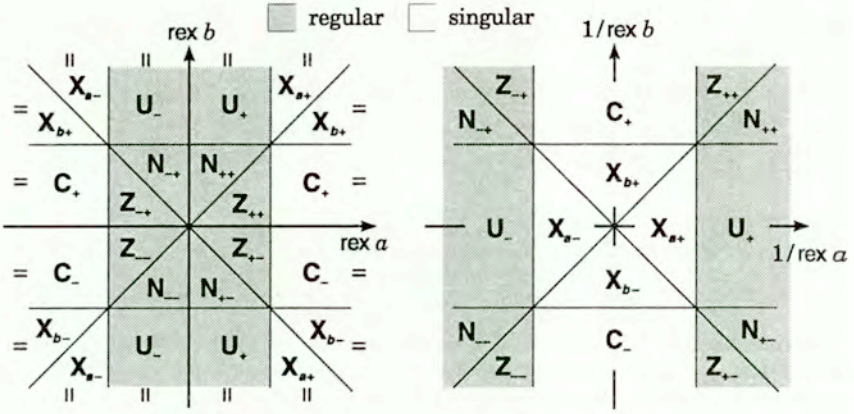


Figure III.54: Subtypes represented by regions in RR-diagrams (intermediate types correspond to lines separating the regions).

rex b (or, alternatively, 1/**rex** a and 1/**rex** b, when it may be more adequately called a *1/RR-diagram*) put on the axes, as shown in Fig. III.53. This diagram is well suited to represent relations between intervals that depend only on the values of the **rex** functions for these intervals.

The RR-diagram representation of the set of types is shown in Fig. III.54. Regions of this diagram correspond to appropriate subtypes, and their neighbourhood relations represent possible smooth transitions between types due to continuous change of the intervals a or b. Such changes pass through the lines dividing the regions, so that these lines correspond to appropriate intermediate types, similarly as in the MR-diagram representation of types shown in Fig. III.51.

The structure of the plane of this diagram is toroidal: it can be best represented on a torus, avoiding connections going through infinity. The fact that all the regions of the RR-plane are occupied by some subtype confirms the completeness of the set of basic subtypes, see the proof of Theorem III.5 in Section III.5.2.3.

The structure represented in the RR-diagram can be also captured with a *graph of types*, which can be drawn in various ways, as shown in Fig. III.55a-d (the subtype indices were omitted here for simplicity). In these drawings, nodes represent subtypes, while edges represent neighbourhood relations between them. Continuous lines correspond to possible smooth transitions between types due to continuous change of the intervals a or b, while dotted lines correspond to more abrupt changes due to the change of sign of the appropriate interval (due to crossing of the **rex** a or **rex** b axis). The graphs are nonplanar, but can be drawn without intersections of edges on a torus. An especially pleasing 3-D structure is depicted by the version of the graph shown in Fig. III.55d.

The set of subtypes, together with their defining conditions, can be also compactly represented with the annotated graph shown in Fig. III.55e. It will be especially useful in the derivation of the set of possible types for the two-dimensional case, see its other version used for enumeration of two-dimensional types in Section III.5.3.5.

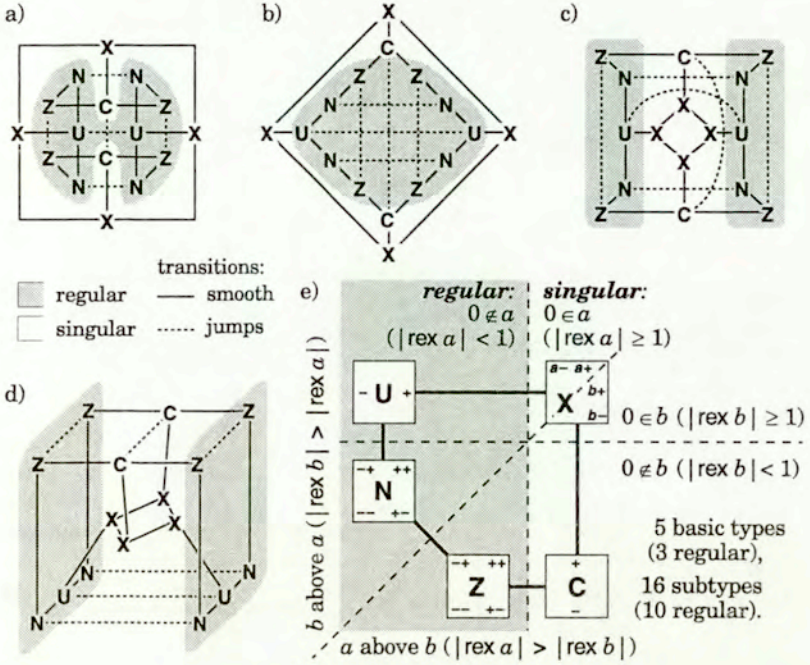


Figure III.55: Several versions of the graph of basic types and subtypes of the $a \cdot x \Diamond b$ relation.

III.5.2.7 Type changes from coefficient change

When one of the coefficients a or b (or both) changes smoothly (which corresponds to its continuous movement in the MR-diagram), the quotients change too, leading eventually to the change of the characteristic quotient sequence and, as a result, the change of the type of the relational expression (unless the coefficient moves along the positive half of its interval axis, when no type changes will occur). The structure of the space of solutions allows only for some trajectories of such type changes. As these changes are important for the analysis of multidimensional cases (see Section III.5.3.4), an example is explained in Fig. III.56, using several representations introduced earlier.

Example III.8 (Moving the a coefficient) In Fig. III.56a, the coefficient a moves along a straight line from a_1 to a_2, crossing the **O*b*** axis and main diagonals **lb0** and **ub0**. As only the coefficient a changes, in the 1/RR-diagram the trajectory is parallel to the 1/**rex** a axis, see Fig. III.56b. It crosses the 1/**rex** b axis, which means that in the RR-diagram the trajectory would consist of two half-lines joined at infinity. As the coefficient moves from the area assigned to the N_{++} subtype through areas of the Z_{++}, C_{+}, and Z_{-+} subtypes, the solution type of the corresponding relational expression changes accordingly, passing through appropriate intermediate types at the points a_b, a_l and a_u.

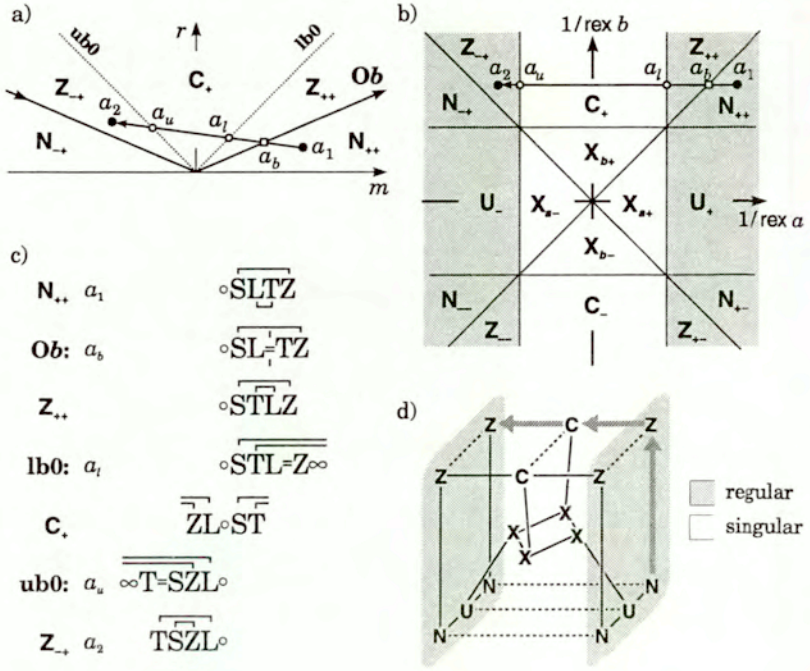


Figure III.56: Moving the a coefficient in various representations.

Figure III.56c shows the full list of characteristic quotient sequences corresponding to these subtypes, as well as the structure of the corresponding solution sets. One can observe that at the beginning (at a_1, and then until a_b) only solution sets Σ and $\Sigma_{\subseteq}$ are nonempty, both being positive intervals. Then at a_b, for the intermediate type $\mathbf{N_{++}|Z_{++}}$, the set $\Sigma_{\subseteq}$ shrinks to a point at L = T, becoming equal to the set $\Sigma_{\supseteq}$, and in consequence, to $\Sigma_{=}$ as well. In the next $\mathbf{Z_{++}}$ region, the set $\Sigma_{\subseteq}$ disappears in favour of $\Sigma_{\supseteq}$. When a approaches the **lb0** diagonal, quotients L and Z grow indefinitely, until at a_l they disappear at infinity, so that the sets Σ and $\Sigma_{\supseteq}$ become half-line positive extervals going from S (respectively, from T) to $+\infty$, and the relational expression becomes singular. In the next $\mathbf{C_+}$ region, quotients L and Z reappear from the other side of the **Om** axis, and the solution sets Σ and $\Sigma_{\supseteq}$ become standard, two-segment extervals. This time the quotients S and T grow to infinity, until the second diagonal **ub0** is reached, where at a_u we again get Σ and $\Sigma_{\supseteq}$ as half-line extervals, now negative. After the diagonal is crossed, the relation returns back to regular, and Σ and $\Sigma_{\supseteq}$ become normal intervals, also negative this time. ■

III.5.3 The two-dimensional relational expression

The two-dimensional is every bit as fictitious as the four-dimensional ...
[Maurits C. Escher, *The Graphic Work of M.C. Escher* (1967)]

Using the results obtained for the one-dimensional case, as described in Section III.5.2, the diagrammatic analysis of two-dimensional interval relational expression of the form:

$$a_1 \cdot x_1 + a_2 \cdot x_2 \Diamond b, \tag{III.32}$$

where $\Diamond \in \{\not\subseteq, \supseteq, \subseteq, =\}$, is conducted here. The solution sets of the above expression are two-dimensional regions on the Ox_1x_2 plane. The diagrammatic analysis of them will be illustrated by a concrete numerical example.

Example III.9a (Two-dimensional expression) An example two-dimensional interval relational expression is given in Fig. III.57a, in both endpoint and midpoint-radius notation. The positions of its interval coefficients in a MR-diagram are shown in Fig. III.57b, together with interval axes of the a_1 and a_2 coefficients and the W-shaped structure of border relations (see Section III.5.2.1) defined by the b coefficient. The structure of its two-dimensional solution sets is depicted in Fig. III.57c. ■

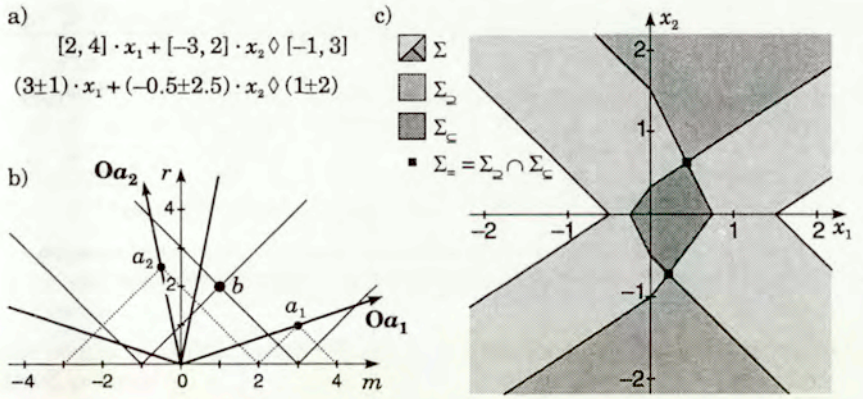


Figure III.57: An example two-dimensional relational expression (a, b) and its solution sets (c).

The main issue of connecting the MR-diagram representation of the coefficients a_1, a_2 and b of the relation with the planar representation of solution sets Σ, $\Sigma_{\supseteq}$, $\Sigma_{\subseteq}$, and $\Sigma_{=}$ (see Section III.5.1 for definitions) on the Ox_1x_2 solution plane is solved by considering one-dimensional cuts through the solution sets plane (Section III.5.3.2) and applying to them the one-dimensional analysis conducted in Section III.5.2. This allows for finding the simple boundary-line selection rule for all solution sets considered (Section III.5.3.3), finding formulae for various characteristic points on the solution plane (Section III.5.3.4), and devising a comprehensive classification of possible types of solution set configurations (Section III.5.3.5). A new diagrammatic tool called a *butterfly diagram* (see Fig. III.65 in

Section III.5.3.4), showing in the MR-diagram the trajectory of the a_i coefficients for all possible cutting axes, proves very convenient to conduct this analysis.

Unless explicitly stated otherwise, we will consider in the following the basic (non-degenerate) cases of interval expressions and solution sets. The intermediate, or degenerate cases will be treated separately in Section III.5.3.7.

III.5.3.1 Boundary lines

Solution sets of the expression (III.32) on the Ox_1x_2 plane are bounded by segments of straight lines, see Fig. III.58a. As illustrated in Fig. III.58b, these lines are defined by linear equations with real coefficients equal to the endpoints of the interval coefficients of (III.32), according to the following formal definition:

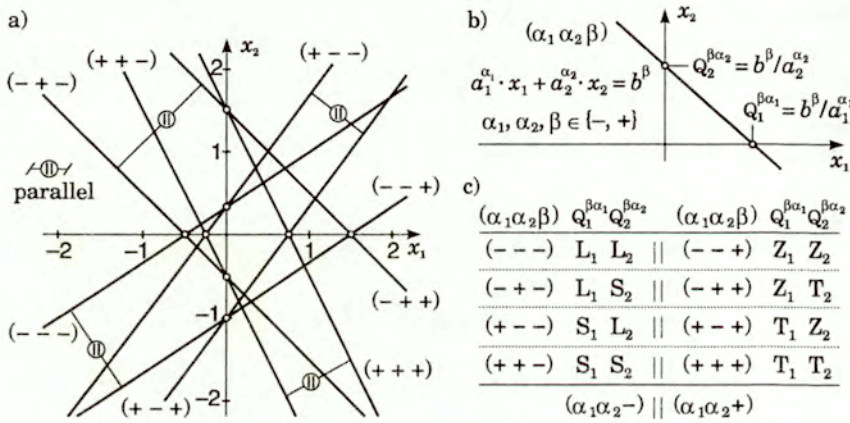


Figure III.58: Boundary lines for the example expression (a), general equation of a boundary line (b), and a code table for all eight possible lines (c).

Definition III.20 (Boundary line) *The straight line in the Ox_1x_2 plane with the equation of the form $a_1^{\alpha_1} \cdot x_1 + a_2^{\alpha_2} \cdot x_2 = b^{\beta}$, where $\alpha_1, \alpha_2, \beta \in \{-,+\}$, is the* boundary line *for (III.32), and is denoted as $(\alpha_1\alpha_2\beta)$.*

The boundary lines intersect the Ox_1 and Ox_2 axes at the points $b^{\beta}/a_1^{\alpha_1} = Q_1^{\beta\alpha_1}$ and $b^{\beta}/a_2^{\alpha_2} = Q_2^{\beta\alpha_2}$, respectively, i.e., at the quotients (see Lemma III.1 for definitions) of the one-dimensional relations $a_1 \cdot x_1 \Diamond b$ and $a_2 \cdot x_2 \Diamond b$, obtained by setting $x_2 = 0$ or $x_1 = 0$ in (III.32). Therefore, they will be also denoted by quotient pairs $Q_1^{\beta\alpha_1}Q_2^{\beta\alpha_2}$, e.g. $(- + -) = L_1S_2$, see the table in Fig. III.58c. At least *two* boundary lines pass through every quotient point on the axes. Namely, the lines $(\alpha_1 - \beta)$ and $(\alpha_1 + \beta)$ pass through $Q_1^{\beta\alpha_1}$, while the lines $(-\alpha_2\beta)$ and $(+\alpha_2\beta)$ pass through $Q_2^{\beta\alpha_2}$. There are *eight* boundary lines, in four pairs of parallel lines, because $(\alpha_1\alpha_2-) \parallel (\alpha_1\alpha_2+)$, see Fig. III.58c again. Note that the special case of a boundary line going through the origin is not included here. As mentioned already, that case belongs to the separate analysis of intermediate (degenerate) cases, see Section III.5.3.7.

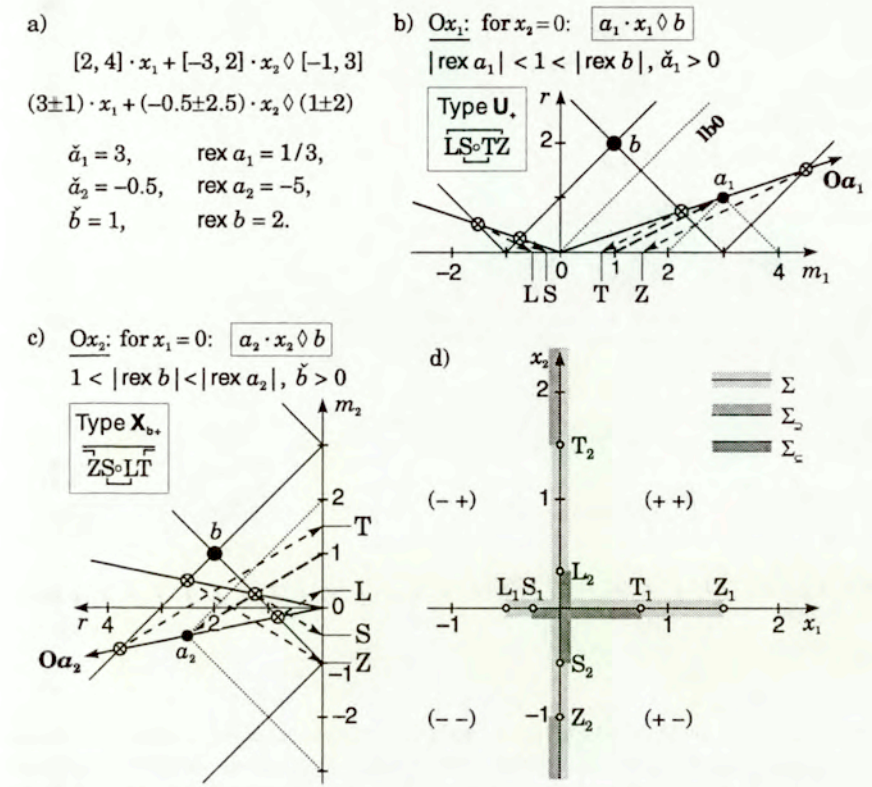


Figure III.59: Calculating cuts for the example (a) by coordinate axes Ox_1 (b) and Ox_2 (c), and the resulting traces of the solution sets on the axes (d); note the different scales on the axes $\mathbf{Om}_1$ and $\mathbf{Om}_2$, and the axes Ox_1 and Ox_2.

III.5.3.2 One-dimensional cuts

Setting $x_2 = 0$ in (III.32), we obtain $a_1 \cdot x_1 \Diamond b$, i.e., a one-dimensional expression, whose quotients are equal to the intersections of boundary lines with the Ox_1 axis. Because the Ox_1 axis cuts through regions representing the solution sets of (III.32), the arrangement of one-dimensional solution sets on the axis is the same as the arrangement of various types of solutions of (III.32) along that cut, and the quotients of the $a_1 \cdot x_1 \Diamond b$ relation indicate borders of these solutions along the cut, see Section III.5.2.2. The same applies to the cut by the Ox_2 axis (when $x_1 = 0$), see Fig. III.59. The exact diagrammatic constructions for the values of the quotients shown in Figures III.59b and c are not really necessary for the essentially qualitative analysis done here. The determination of the types (and corresponding quotient sequences) suffices for that purpose. It can be done by comparing only extents and signs of appropriate coefficients, as shown in Fig. III.59 (see Section III.5.2 for detailed procedures and explanations).

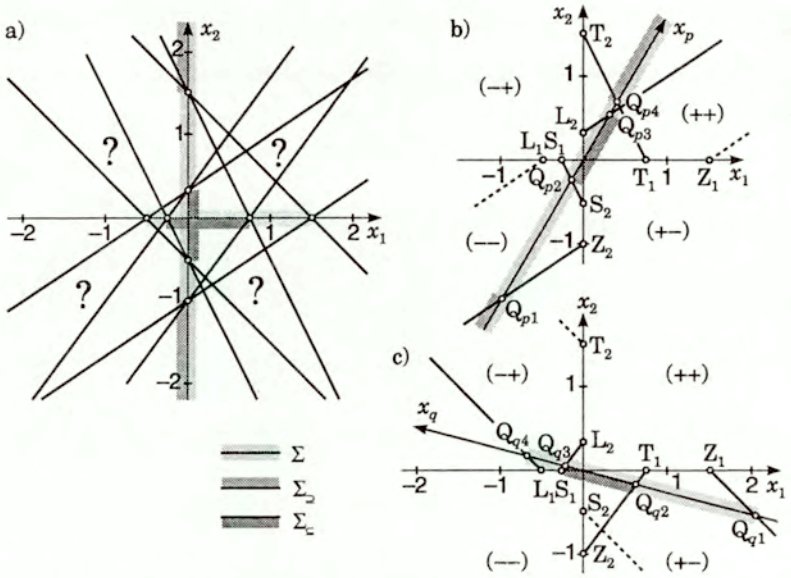


Figure III.60: Cuts by coordinate axes cannot directly determine the boundary line selection (a), but intersections of rotated axes with boundary lines can (b, c).

Rotated axes and boundary lines. Because always two boundary lines pass through every quotient point, the two perpendicular cuts by the coordinate axes Ox_1 and Ox_2 do not suffice to directly characterize the two-dimensional solution sets, see Fig. III.59 and Fig. III.60a. Let us, however, consider cuts of the solution plane by arbitrarily rotated axes, like Ox_p and Ox_q shown in Fig. III.60bc (compare these figures with Fig. III.57). Intersections of such axes with the boundary lines divide the axes into intervals contained in the solution sets, producing a pattern analogical to that on the Ox_1 and Ox_2 axes. Is it possible that these patterns correspond to solution sets of some one-dimensional relational expressions $a_p \cdot x_p \lozenge b$ and $a_q \cdot x_q \lozenge b$, respectively, where a_p and a_q are some functions of the coefficients of the original expression (III.32) and the direction of the axis? If so, then by cutting the solution plane with such an axis it should be possible to find which of the boundary lines intersecting the axis constitute borders of appropriate solution sets—namely, those intersecting the axis at appropriate quotients Q_{pi} and Q_{qi} indicated in Fig. III.60bc. It is indeed true, as the following two lemmata certify.

Some additional notation will be used in the sequel. Namely, the quadrants into which the axes divide the Ox_1x_2 plane will be denoted by pairs of signs $(\sigma_1\,\sigma_2)$, $\sigma_i = \mathsf{sgn}\, x_i$, of coordinates of points lying in the given quadrant, see Figs. III.60 and III.61. The points on the Ox_i axes with values of 1 and -1 will be denoted by $1x_i$ and $-1x_i$, respectively. The symbol $\overline{\mathrm{OX}}$ will denote the distance along some axis between the origin O and a point X on that axis, *signed* according to the relative positions of the origin and the point X with respect to the direction of the axis.

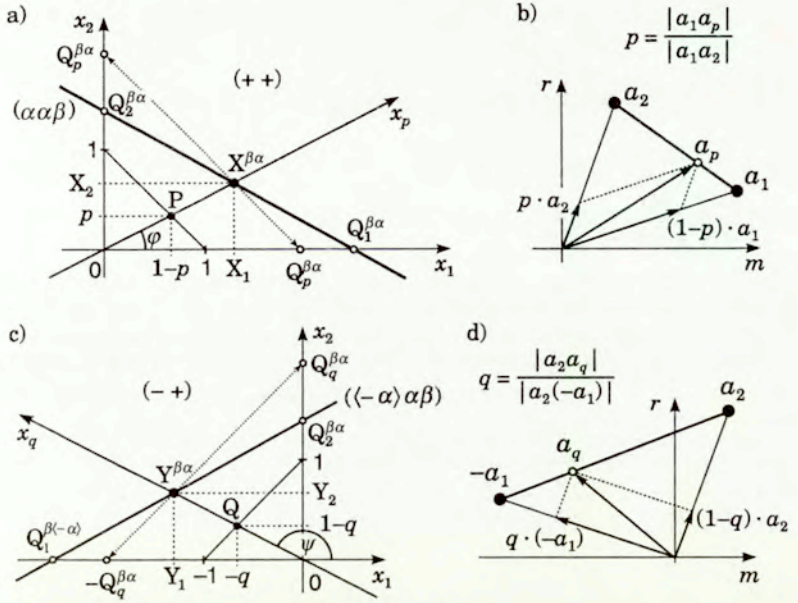


Figure III.61: The rotated cutting axes Ox_p, or $(--)\nearrow(++)$ and Ox_q, or $(+-)\nearrow(-+)$.

For the axis Ox_p going from the $(--)$ quadrant to the $(++)$ quadrant (denoted in the sequel as $(--)\nearrow(++)$), its intersections with appropriate boundary lines are given by:

Lemma III.3 (The $(--)\nearrow(++)$ axis) *Let $0 \le p \le 1$; $\alpha, \beta \in \{-,+\}$; $0 \le \varphi \le \pi/2$; and $\tan\varphi = p/(1-p)$. Then the intersection of the axis Ox_p, going through the origin and rotated by the angle φ, with the boundary line $(\alpha\alpha\beta)$ is defined by $\mathrm{X}^{\beta\alpha} = ((1-p)\mathrm{Q}_p^{\beta\alpha}, p\mathrm{Q}_p^{\beta\alpha})$, where $\mathrm{Q}_p^{\beta\alpha} = b^\beta/a_p^\alpha$ is the quotient of the interval relational expression $a_p \cdot x_p \Diamond b$ with the coefficient $a_p = (1-p)\cdot a_1 + p \cdot a_2$ lying on the straight segment between a_1 and a_2 in the MR-diagram.*

Proof. Let us examine the construction shown in Fig. III.61a. Assume the cutting axis Ox_p makes the angle φ, $0 \le \varphi \le \pi/2$, with the Ox_1 axis. Draw the straight line segment connecting the points $1x_1$ and $1x_2$ and intersecting the axis Ox_p at the point P. Denoting by p the x_2 coordinate of P, we get $0 \le p \le 1$ and $\tan\varphi = p/(1-p)$. The boundary line $(\alpha\alpha\beta)$ passes through the same-type quotients on the Ox_1 and Ox_2 axes, and there are four such lines, corresponding to four possible combinations of signs $(\alpha\beta)$, see Fig. III.58c. For non-degenerate cases, six distinct configurations of such lines are possible, with respect to the Ox_p axis and coordinate axes, see Fig. III.60b for some examples. Only these parts of boundary lines of appropriate type that pass through the quadrants containing the Ox_p axis are shown in Fig. III.60b. As a result, some of the lines consist of two disjoint half-lines—the parts drawn dotted do not intersect the Ox_p axis in its current position, but will intersect it for another angle φ. The proof goes in essentially the same way for all of the possible cases.

Namely, the intersection point $X^{\beta\alpha} = (X_1, X_2)$ of the boundary line $(\alpha\alpha\beta)$ with the line Ox_p must fulfill the following set of equations:

$$\begin{cases} x_2 = (p/(1-p))x_1 & \text{(the } Ox_p \text{ axis)}, \\ a_1^\alpha x_1 + a_2^\alpha x_2 = b^\beta & \text{(the } (\alpha\alpha\beta) \text{ boundary line)}. \end{cases}$$

Solving the equations, we get $X_1 = (1-p)Q_p^{\beta\alpha}$ and $X_2 = pQ_p^{\beta\alpha}$, where:

$$Q_p^{\beta\alpha} = \frac{b^\beta}{(1-p)\cdot a_1^\alpha + p\cdot a_2^\alpha} = b^\beta / a_p^\alpha,$$

and $a_p^\alpha = (1-p)\cdot a_1^\alpha + p\cdot a_2^\alpha$. As the above is valid for any $\alpha, \beta \in \{-,+\}$, and both p and $(1-p)$ are nonnegative, from the two equalities $a_p^\alpha = (1-p)\cdot a_1^\alpha + p\cdot a_2^\alpha$ for both possible values of α it follows also $a_p = (1-p)\cdot a_1 + p\cdot a_2$, as required. Furthermore, from basic rules of (diagrammatic) addition of intervals (see Section III.4.1 and Fig. III.61b) it is obvious that a_p lies on the straight line segment joining a_1 and a_2, dividing it in the proportion $p : (1-p)$. □

Note that $p = 0$ in the above produces the Ox_1 axis, while $p = 1$ gives the Ox_2 axis.

An analogous result holds also for the axis Ox_q passing from the $(+-)$ quadrant to the $(-+)$ quadrant:

Lemma III.4 (The $(+-)\nearrow(-+)$ axis) *Let $0 \le q \le 1$; $\alpha, \beta \in \{-,+\}$; $\pi/2 \le \psi \le \pi$; and $\tan\psi = (q-1)/q$. Then the intersection of the axis Ox_p, going through the origin and rotated by the angle ψ, with the boundary line $(\langle -\alpha\rangle\alpha\beta)$ is defined by $Y^{\beta\alpha} = (-qQ_q^{\beta\alpha}, (1-q)Q_q^{\beta\alpha})$, where $Q_q^{\beta\alpha} = b^\beta / a_q^\alpha$ is the quotient of the interval relational expression $a_q \cdot x_q \Diamond b$ with the coefficient $a_q = q\cdot(-a_1) + (1-q)\cdot a_2$ lying on the straight segment between a_2 and $-a_1$ in the MR-diagram.*

Proof. Let us examine the construction shown in Fig. III.61c. Drawing the straight line segment connecting the points $1x_2$ and $-1x_1$ and intersecting the axis Ox_q at the point Q, as well as denoting by $-q$ the x_1 coordinate of Q, we get $0 \le q \le 1$ and $\tan\psi = (q-1)/q$. The boundary line $(\langle -\alpha\rangle\alpha\beta)$ passes through the quotients $Q_1^{\beta\langle-\alpha\rangle}$ and $Q_2^{\beta\alpha}$ on the Ox_1 and Ox_2 axes, respectively. Again, there are four such lines, namely the remaining four besides the four $(\alpha\alpha\beta)$ lines covered by Lemma III.3, see Fig. III.58c. The proof then goes in essentially the same way as before.

Namely, the intersection point $Y^{\beta\alpha} = (Y_1, Y_2)$ fulfills the following set of equations:

$$\begin{cases} x_2 = ((q-1)/q)x_1 & \text{(the } Ox_q \text{ axis)}, \\ a_1^{\langle-\alpha\rangle} x_1 + a_2^\alpha x_2 = b^\beta & \text{(the } (\langle-\alpha\rangle\alpha\beta) \text{ boundary line)}. \end{cases}$$

Solving the equations, and taking into account that because $-u = [-\overline{u}, -\underline{u}]$ we have $(-u)^\alpha = -u^{\langle-\alpha\rangle}$, we finally get $Y_1 = -qQ_q^{\beta\alpha}$ and $Y_2 = (1-q)Q_q^{\beta\alpha}$, where:

$$Q_q^{\beta\alpha} = \frac{b^\beta}{-q\cdot a_1^{\langle-\alpha\rangle} + (1-q)\cdot a_2^\alpha} = \frac{b^\beta}{q\cdot(-a_1)^\alpha + (1-q)\cdot a_2^\alpha} = b^\beta / a_q^\alpha,$$

and $a_q^\alpha = q\cdot(-a_1)^\alpha + (1-q)\cdot a_2^\alpha$. Again, from the above follows also $a_q = q\cdot(-a_1) + (1-q)\cdot a_2$, and thus a_q lies on the straight line segment joining a_2 and $-a_1$, dividing the segment in the proportion $q : (1-q)$, as required, see Fig. III.61d. □

Here, $q = 0$ produces the Ox_2 axis, while $q = 1$ gives the $-Ox_1$ axis, i.e., the Ox_1 axis with its direction reversed.

Positions of the intersection points $X^{\beta\alpha}$ and $Y^{\beta\alpha}$ along the corresponding axis (Ox_p or Ox_q) are given, respectively, by $\overline{OX}^{\beta\alpha} = \overline{OP} \cdot Q_p^{\beta\alpha} = Q_p^{\beta\alpha}\sqrt{1-2p(1-p)}$ and $\overline{OY}^{\beta\alpha} = \overline{OQ} \cdot Q_q^{\beta\alpha} = Q_q^{\beta\alpha}\sqrt{1-2q(1-q)}$. Hence, positions of the quotients $Q_p^{\beta\alpha}$ and $Q_q^{\beta\alpha}$ along the rotated axes are scaled differently than along the axes Ox_1 and Ox_2, and, moreover, the scale changes nonlinearly with parameters p and q. Fortunately, this causes no trouble for the kind of analysis conducted here, especially as the exact correspondence between the positions of the intersection points and values of the quotients is easily obtained by a diagonal projection of the points onto the Ox_1 and Ox_2 axes, as shown in Fig. III.61a, c.

Note also that the points P and Q divide the segments between the points $1x_1$ and $1x_2$, as well as between $1x_2$ and $-1x_1$, in the proportions $p : (1-p)$ and $q : (1-q)$, respectively. Hence, these intersection points define the (linear) scales of the parameters p and q along the respective segments, constituting a measure of the direction (angle of rotation) of the axes Ox_p and Ox_q. These scales correspond directly to the scales defined by corresponding coefficients a_p and a_q along the segments in the MR-diagram, see Fig. III.61b, d.

The lemmata above can be formulated in the form valid for any pair of opposite quadrants $(\langle-\delta\rangle\langle-\gamma\rangle)\nearrow(\delta\gamma)$, and generalized to an n-dimensional case, see Theorem III.9 in Section III.5.4.

Remark. The linear parametrization chosen here produces straight-line trajectories for coefficients a_p and a_q in the MR-diagram, but it requires changes in the scale on the rotated axes, as explained above. A trigonometric parametrization for which $X^{\beta\alpha} = (Q_p^{\beta\alpha}\cos t, Q_p^{\beta\alpha}\sin t)$ would result in a uniform scale on all axes, but the coefficients a_p and a_q would then be defined by complex formulae producing inconveniently curved trajectories in the MR-diagram. The linear parametrization is much more convenient here.

III.5.3.3 Boundary lines selection rule

Then slowly on the surface ... faint lines appeared, ...
steadily they grew broader and clearer, until their design could be guessed.
[John R.R. Tolkien, *The Fellowship of the Ring* (1954)]

The lemmata proved in the previous section lead directly to the following theorem which gives a rule how to actually find which particular fragments of the boundary lines constitute borders of various solutions sets.

Theorem III.8 (Boundary lines selection) *The borders of the solution sets* Σ, $\Sigma_\supseteq$, *and* $\Sigma_\subseteq$ *on the* Ox_1x_2 *plane consist of the fragments of the boundary lines included in a respective quadrant according to the rule (with* $\alpha, \beta \in \{-,+\}$*):*

In quadrants:	*take lines of the type:*
(−−), (++)	$(\alpha\alpha\beta)$, *i.e.,* $L_1L_2 \parallel Z_1Z_2$; $S_1S_2 \parallel T_1T_2$;
(+−), (−+)	$(\langle-\alpha\rangle\alpha\beta)$, *i.e.,* $L_1S_2 \parallel Z_1T_2$; $S_1L_2 \parallel T_1Z_2$.

Proof. It follows almost immediately from Lemma III.3 and Lemma III.4. Namely, because the fragments of boundary lines indicated in those lemmata intersect the axes Ox_p and Ox_q at the points directly corresponding (barring the appropriate change of scale

along the axes) to the quotients $Q_p^{\beta\alpha}$ and $Q_q^{\beta\alpha}$ for the relations $a_p \cdot x_p \Diamond b$ and $a_q \cdot x_q \Diamond b$, and these quotients give the borders of the appropriate solution sets along the axes, the intersections of the indicated boundary lines with the axes must also constitute border points of the solution sets of the two-dimensional expression (III.32). As the above holds for every direction of the axes within appropriate quadrants, that is, for every point of the indicated segments of boundary lines (see Fig. III.60b, c), these segments constitute borders between appropriate solution sets of (III.32), as required. □

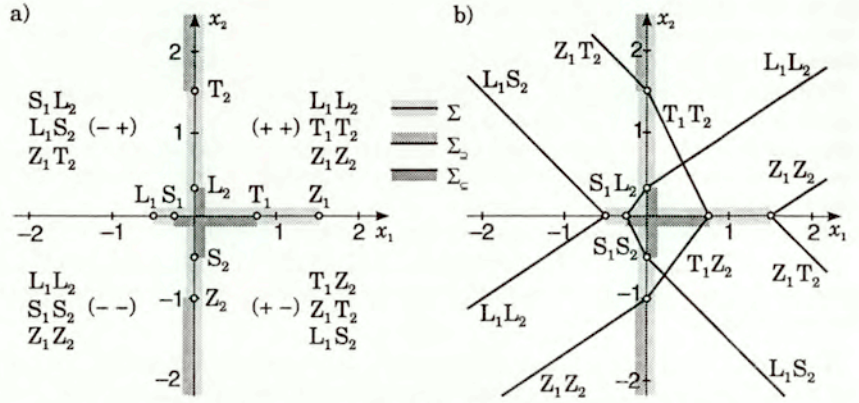


Figure III.62: Boundary lines selection rule in action: selecting boundary line segments to draw in the given quadrant (a), and the solution sets boundaries obtained (b).

Example III.9b (Boundary line selection) In Fig. III.62 one may observe the above rule in action for our running example. In Fig. III.62a, we start from the positions of quotients on the axes Ox_1 and Ox_2 (they follow directly from the types of relations $a_1 \cdot x_1 \Diamond b$ and $a_2 \cdot x_2 \Diamond b$, according to the rules determined in Section III.5.2.3). The types also determine the labelling of segments of the axes by appropriate solution set indicators, as shown by thick gray lines. Then Theorem III.8 indicates, for every quadrant, the segments of boundary lines (denoted by appropriate quotient pairs here) that constitute the borders between two-dimensional regions of the solution sets (those segments of the lines that do not cross the appropriate quadrant are omitted in the figure). Drawing the indicated fragments of the lines we obtain Fig. III.62b. What remains to be done is to propagate the solution sets indicators from the axes into appropriate regions to obtain the diagram of the solution sets for the example, as shown in Fig. III.57c. ■

III.5.3.4 Structure of solution sets

Unless things are altogether changed,
eyes that know what to look for may discover the signs.
[John R.R. Tolkien, *The Fellowship of the Ring* (1954)]

Combining the results above, one can easily proceed from the configuration of intervals a_1, a_2, and b, and the trajectories of the coefficients a_p and a_q in the MR-diagram, to

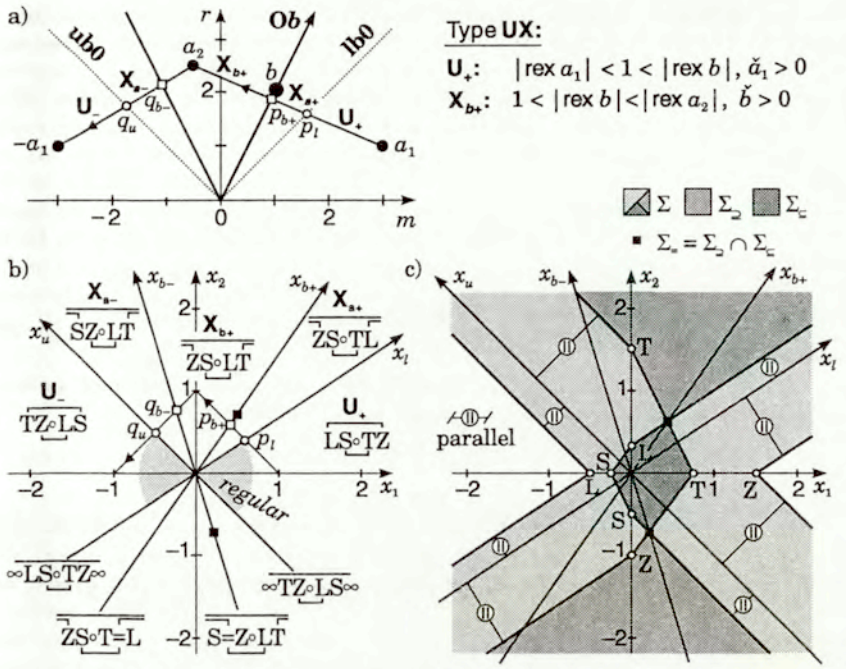


Figure III.63: A full solution analysis for the example relational expression: from the MR-diagram (a), through characteristic directions and cut types (b), to the structure of solution sets in the Ox_1x_2 plane (c).

the structure and shape of corresponding solution sets on the Ox_1x_2 plane, as shown in Fig. III.63 for the example. The trajectory of the coefficients a_p and a_q in the MR-diagram (from a_1 through a_2 to $-a_1$) corresponds directly (through parameters p and q marked along the trajectory and on the scales along the segments from $1x_1$ through $1x_2$ to $-1x_1$), to the change of direction of the cutting axes Ox_p and Ox_q. The changing solution types defined along that trajectory (see Section III.5.2.7 for more details on type changes) determine the arrangement of borders of solution sets along the directions of the cutting axes.

Characteristic points and axes. Certain characteristic values of the parameters p and q, defined by intersections of the trajectory with the main diagonals **lb0** and **ub0**, and with the interval axis **O***b*, correspond to characteristic directions in the Ox_1x_2 plane of solutions, for which the solution type of the cutting axis changes. They thus correspond to certain intermediate types, indicated in Fig. III.63b at the appropriate axes. Directions of these axes can be calculated according to the following two propositions, see also Fig. III.64:

Proposition III.11 (Singular boundary axes) *Intersections of the coefficient trajectory in the MR-diagram with the main diagonals* **lb0** *and* **ub0** *correspond to cutting axes with directions parallel to the borderlines of the solution set* $\Sigma([\![a_1\ a_2]\!], b)$*, i.e., to those directions at which the solution type of the cut changes from regular to singular. They exist only if at least one of the coefficients* a_1 *or* a_2 *lies below the main diagonals in the MR-diagram, i.e., when there exists some* regular *one-dimensional cut through the solution sets (see Fig. III.67a, Fig. III.67c, and Fig. III.67d in Section III.5.3.6 for the enumeration of appropriate types of solution set configurations). Tangents of the angles* φ_l *and* φ_u *between the* Ox_1 *axis and the cutting axes with this property are given by the formulae:*

$$\begin{aligned} \tan\varphi_l &= -a_1^{\langle -\sigma_1\rangle}/a_2^{\langle -\sigma_2\rangle} = -Q_2^{\langle -\sigma_2\rangle\beta}/Q_1^{\langle -\sigma_1\rangle\beta}; \\ \tan\varphi_u &= -a_1^{\sigma_1}/a_2^{\sigma_2} = -Q_2^{\sigma_2\beta}/Q_1^{\sigma_1\beta}; \end{aligned} \tag{III.33}$$

where $(\sigma_1\,\sigma_2)$ *denotes the quadrant into which the given axis points (determined, in turn, by the segment of the trajectory intersecting the respective diagonal).*

Proof. Let us first verify the formulae for directions of the axes. Consider the axis pointing into the $(++)$ quadrant. For this quadrant, we have $a_p = (1-p)\cdot a_1 + p\cdot a_2$, see Lemma III.3. As the interval lying on the **lb0** diagonal starts with zero, i.e., $\underline{a}_l = 0$, we have $\underline{a}_l = (1-p)\underline{a}_1 + p\underline{a}_2 = 0$. Thus, $(p/(1-p))\underline{a}_2 = -\underline{a}_1$ and $p/(1-p) = -\underline{a}_1/\underline{a}_2$. Because in this quadrant $\tan\varphi_l = p/(1-p)$, see Lemma III.3, and $\sigma_1 = \sigma_2 = +$, then $\underline{a}_1 = a_1^{\langle -\sigma_1\rangle}$, $\underline{a}_2 = a_2^{\langle -\sigma_2\rangle}$, and hence the formula for $\tan\varphi_l$ in (III.33) is valid for this case. The formula for $\tan\varphi_u$ can be verified in the same way, this time starting from $\overline{a}_u = 0$.

A similar argument proves the formula for the $(-+)$ quadrant. Here, using the formulae given in Lemma III.4, we get easily $\tan\psi_l = -\overline{a}_1/\underline{a}_2$ and $\tan\psi_u = -\underline{a}_1/\overline{a}_2$. Observing that now $\sigma_1 = -$, we see that the formulae (III.33) are valid in this case as well. As can be also verified, the formulae are also valid for the remaining two quadrants (see next sections for more discussion of the axes rotated by more than 180 degrees).

Concerning the first part of the theorem, as explained in Section III.5.2.5, solutions to one-dimensional relational systems with a zero-endpoint coefficient (lying on one of the main diagonals) are intermediate between regular and singular solutions, and are thus characterized by external quotients (marking the bounds of the united solution set Σ) going to infinity. This signifies that the axis with this property ceases to intersect the boundary lines corresponding to these external quotients, that is, to the borders of the Σ solution set. It is possible only when it becomes parallel to these boundary lines. This condition is confirmed by the fact that the formulae for the tangents of directional angles for the axes are identical to that for angle coefficients of four of the boundary lines, namely the lines with codes $(\langle -\sigma_1\rangle\ \langle -\sigma_2\rangle\ \beta)$ or $(\sigma_1\ \sigma_2\ \beta)$, $\beta \in \{-, +\}$, for the trajectory segments intersecting the diagonal **lb0** or **ub0**, respectively, see Definition III.20. Hence, every such axis is parallel to two of these four boundary lines, and only to them, as required. The requirement that at least one of the coefficients a_1 or a_2 to lie below main diagonals is obvious, cf. Fig. III.64. Also, expressing the formulae in terms of quotients is straightforward. □

Example III.9c (Singular boundary axes) For our example, the two axes of this type, marked Ox_l and Ox_u in Fig. III.63, make with the Ox_1 axis angles with tangents equal to $-a_1^{\langle -\sigma_1\rangle}/a_2^{\langle -\sigma_2\rangle} = -a_1^-/a_2^- = -\underline{a}_1/\underline{a}_2 = 2/3$ and $-a_1^{\sigma_1}/a_2^{\sigma_2} = -a_1^-/a_2^+ = -\underline{a}_1/\overline{a}_2 = -2/2 = -1$, and are parallel to the boundary lines $(---)$, $(--+)$ and $(-+-)$, $(-++)$, respectively, cf. Fig. III.58. ■

Proposition III.12 ($\Sigma_=$ axes) *In basic cases, intersections of the coefficient trajectory in the MR-diagram with the axis* $\mathbf{Ob}$ *correspond to cutting axes going through the points of the* $\Sigma_=([\![a_1\ a_2]\!], b)$ *solution set (where the boundaries of the solution sets* $\Sigma_\supseteq([\![a_1\ a_2]\!], b)$ *and* $\Sigma_\subseteq([\![a_1\ a_2]\!], b)$ *intersect). Tangents of the angles* φ_{b+} *and* φ_{b-} *between the* Ox_1 *axis and the cutting axes with this property are given by the formulae:*

$$\begin{aligned} \tan\varphi_{b+} &= \frac{a_1^{\langle -\sigma_1\rangle}\,\overline{b} - a_1^{\sigma_1}\,\underline{b}}{a_2^{\sigma_2}\,\underline{b} - a_2^{\langle -\sigma_2\rangle}\,\overline{b}}\,; \\ \tan\varphi_{b-} &= \frac{a_1^{\sigma_1}\,\overline{b} - a_1^{\langle -\sigma_1\rangle}\,\underline{b}}{a_2^{\langle -\sigma_2\rangle}\,\underline{b} - a_2^{\sigma_2}\,\overline{b}}\,; \end{aligned} \tag{III.34}$$

where $(\sigma_1\,\sigma_2)$ *denotes the quadrant into which the given axis points (determined, in turn, by the segment of the trajectory intersecting the respective branch of the* $\mathbf{Ob}$ *axis).*

Proof. As explained in Section III.5.2.1, the solution set $\Sigma_=$ is nonempty only when the coefficient a lies on the axis $\mathbf{Ob}$. Therefore, the one-dimensional cuts containing a point from the $\Sigma_=([\![a_1\ a_2]\!], b)$ solution set must correspond to the intersections of the coefficient trajectory with the $\mathbf{Ob}$ axis, i.e., to the points a_{b-} and a_{b+} shown in Fig. III.64. For the point a_{b+} lying on the positive branch of $\mathbf{Ob}$ we have $\mathbf{rex}\ b = \mathbf{rex}\ a_{b+}$, that is $\hat{b}/\check{b} = \hat{a}_{b+}/\check{a}_{b+}$, hence $(\overline{b}-\underline{b})/(\overline{b}+\underline{b}) = (\overline{a}_{b+}-\underline{a}_{b+})/(\overline{a}_{b+}+\underline{a}_{b+})$ and after a series of simple transformations: $\overline{a}_{b+}\,\underline{b} = \underline{a}_{b+}\,\overline{b}$. Because, for the $(+\ +)$ quadrant, $a_{b+} = (1-p)a_1 + p\,a_2$, we get $(\overline{a}_1 + (p/(1-p))\,\overline{a}_2)\underline{b} = (\underline{a}_1 + (p/(1-p))\,\underline{a}_2)\overline{b}$. As $\tan\varphi = p/(1-p)$ in the $(+\ +)$ quadrant, the formula for $\tan\varphi_{b+}$ is easily checked to be valid for this case. A similar argument proves it for the $(-\ +)$ quadrant as well.

An argument for the point a_{b-} lying on the negative branch of $\mathbf{Ob}$ starts from the equality $\mathbf{rex}\ b = -\mathbf{rex}\ a_{b-}$, and then proceeds as above. □

Example III.9d ($\Sigma_=$ axes) For our example, the two axes of this type, marked Ox_{b+} and Ox_{b-} in Fig. III.63, make with the Ox_1 axis angles with tangents equal to $(a_1^-\,\overline{b} - a_1^+\,\underline{b})/(a_2^+\,\underline{b} - a_2^-\,\overline{b}) = (\underline{a}_1\,\overline{b} - \overline{a}_1\,\underline{b})/(\overline{a}_2\,\underline{b} - \underline{a}_2\,\overline{b}) = (2\cdot 3 - 4(-1)/(2(-1) - (-3)\cdot 3) = 10/7$ and $(a_1^-\,\overline{b} - a_1^+\,\underline{b})/(a_2^-\,\underline{b} - a_2^+\,\overline{b}) = (\underline{a}_1\,\overline{b} - \overline{a}_1\,\underline{b})/(\underline{a}_2\,\underline{b} - \overline{a}_2\,\overline{b}) = (2\cdot 3 - 4(-1))/((-3)(-1) - 2\cdot 3) = -10/3$, respectively. ■

The $\Sigma_=([\![a_1\ a_2]\!], b)$ solution set can be determined with the help of the following proposition.

Proposition III.13 ($\Sigma_=$ solution set) *In basic cases, the* $\Sigma_=([\![a_1\ a_2]\!], b)$ *solution set is nonempty when at least one of the coefficients* a_1, a_2 *lies on or below the axis* $\mathbf{Ob}$*, and then the coordinates of the points* (ϵ_1, ϵ_2) *constituting this set are given by the formulae:*

$$\begin{aligned} \epsilon_1 &= \frac{a_2^{\sigma_2}\,\underline{b} - a_2^{\langle -\sigma_2\rangle}\,\overline{b}}{a_1^{\langle -\sigma_1\rangle}\,a_2^{\sigma_2} - a_1^{\sigma_1}\,a_2^{\langle -\sigma_2\rangle}}\,; \\ \epsilon_2 &= \frac{a_1^{\langle -\sigma_1\rangle}\,\overline{b} - a_1^{\sigma_1}\,\underline{b}}{a_1^{\langle -\sigma_1\rangle}\,a_2^{\sigma_2} - a_1^{\sigma_1}\,a_2^{\langle -\sigma_2\rangle}}\,; \end{aligned} \tag{III.35}$$

where $(\sigma_1\,\sigma_2)$ denotes the quadrant in which the given point lies (determined, in turn, by the segment of trajectory intersecting the respective branch of the $\mathbf{O}b$ axis).

Proof. The first part is essentially a corollary to Proposition III.12. Note that only when one of the coefficients a_1 or a_2 lies on or below the $\mathbf{O}b$ axis, the trajectory of cutting axes coefficients (from a_1 to a_2 to $-a_1$) can intersect the $\mathbf{O}b$ axis, producing a point belonging to the $\Sigma_=$ solution set. If both a_1 and a_2 lie above the $\mathbf{O}b$ axis, the trajectory has no points in common with the $\mathbf{O}b$ axis, and hence $\Sigma_=$ is empty.

The set $\Sigma_=([\![a_1\ a_2]\!], b)$ is a set of (real) points $\{(\epsilon_1, \epsilon_2)\}$, each of which fulfills the equation $a_1 \cdot \epsilon_1 + a_2 \cdot \epsilon_2 = b$. Depending on the signs of ϵ_1 and ϵ_2 (i.e., on the quadrant in which the point (ϵ_1, ϵ_2) lies), different pairs of real equations in terms of endpoints of coefficients a_1, a_2, and b are obtained, because for any $t \in \mathbb{R}$ we have $t \cdot u = [t\,u^{\langle -\tau\rangle}, t\,u^{\tau}]$, where $\tau = \operatorname{sgn} t$. All of these cases can be uniformly captured in the pair of simultaneous equations of the form:

$$\begin{cases} a_1^{\langle -\sigma_1\rangle}\epsilon_1 + a_2^{\langle -\sigma_2\rangle}\epsilon_2 &= \underline{b}, \\ a_1^{\sigma_1}\epsilon_1 + a_2^{\sigma_2}\epsilon_2 &= \overline{b}. \end{cases}$$

Solving them with respect to ϵ_1 and ϵ_2 we obtain the formulae given in the proposition. □

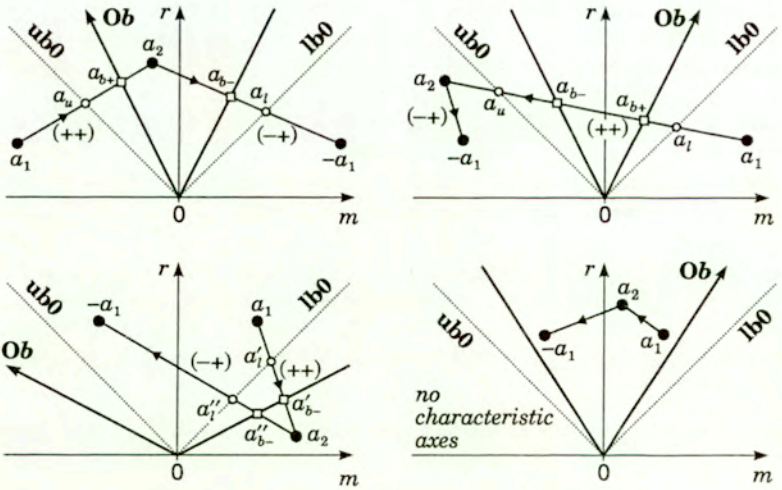


Figure III.64: Several examples of possible coefficient trajectories intersecting main diagonals and the $\mathbf{O}b$ axis; codes of quadrants in which corresponding axes lie are shown too.

If after putting appropriate values of σ_1 and σ_2 into the formulae, the values of ϵ_1 and ϵ_2 are of different sign than σ_1 or σ_2, respectively, it means that the quadrant $(\sigma_1\,\sigma_2)$ does not contain a point belonging to the set $\Sigma_=([\![a_1\ a_2]\!], b)$.

The formulae (III.34) and (III.35) can be also easily expressed in terms of quotients, like (III.33). The translation is straightforward and is left to the reader.

Example III.9e ($\Sigma_=$ solution set) As can be calculated from the formulae given above, for our example the set $\Sigma_=([\![a_1\ a_2]\!], b)$ equals $\{(7/16,\ 5/8),\ (3/14,\ -5/7)\}$, with points contained in quadrants $(++)$ and $(+-)$ respectively, cf. Fig. III.63. ■

The case of the coefficient a_p (or a_q) lying on the **Or** axis is discussed in the next section.

The butterfly diagram. As for now, we considered rotation of the cutting axes by 180 degrees only—from the direction of the Ox_1 axis through Ox_2 to $-Ox_1$. We can rotate the axis further, to complete the whole range of 360 degrees. However, further positions of the axis do not provide any essentially new information about the arrangement of two-dimensional solution sets, because the arrangements of quotients along the axes will simply repeat those for the first 180 degrees, only with reversed order of the quotients along the axis.

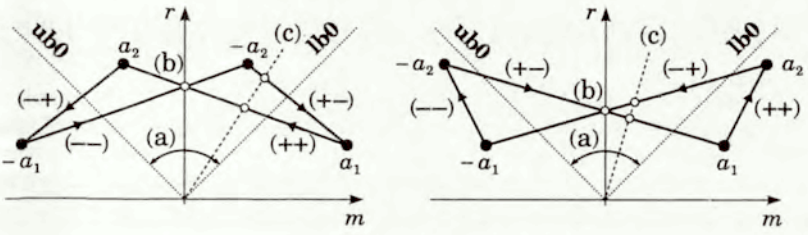


Figure III.65: Two examples of butterfly diagrams; labels (a), (b) and (c) refer to elements connected with properties stated by Propositions III.14, III.15, and III.16, respectively.

Nevertheless, an analysis of the whole trajectory of the coefficients in the MR-diagram, corresponding to the full 360 degrees rotation of the cutting axis, can provide us with some new insights concerning properties of the two-dimensional solution sets. The full trajectory produces the *butterfly diagram*, see Fig. III.65 for two examples. Except for degenerate cases, all possible butterfly diagrams are generally similar to those shown, differing only by the arrangement of coefficients a_1 and a_2 at the vertices. Arrows denote the direction of movement of coefficients corresponding to the counterclockwise rotation of the cutting axis. Symbols for the quadrant into which the corresponding cutting axis points are also shown along the sides of the diagrams.

An analysis of the diagrams directly leads to some general observations concerning the structure of two-dimensional solution sets. First, as can be verified, the formulae given by (III.33)–(III.35) remain valid for the other quadrants (namely, $(--)$ and $(+-)$) as well. Now, the pairs of formulae given by (III.33) and (III.34) can be shown to be pairwise

equivalent, i.e., $\tan\varphi_l$ transforms into $\tan\varphi_u$ and $\tan\varphi_{b+}$ into $\tan\varphi_{b-}$ (and vice versa) with the change of the quadrant involved into its opposite. To see it demonstrated in the diagram, note that whenever a branch of the coefficient trajectory in Fig. III.65 intersects one of the main diagonals (or one of the halves of the **O*b*** axis), the branch corresponding to the opposite quadrant intersects the other diagonal (or the other half of the **O*b*** axis) at the corresponding point (producing the characteristic axis with the same direction). This signifies the fact that the characteristic axes involved are given by the same straight lines, only oppositely oriented. Other significant observations are formulated below.

Proposition III.14 (Singular cuts) *Unless a_1 and a_2 are both thin, there are always singular cuts.*

Proof. The property is obvious from the diagrams (see (a) in Fig. III.65): no matter where the coefficients a_1 and a_2 are located on the MR-plane (provided at least one of them is thick), the coefficient trajectory must pass through the region *above* the diagonals **lb0** and **ub0**, i.e., the region where a_p or a_q contains zero. □

In other words, there are always cuts of the type **C** (for which both Σ and $\Sigma_{\supseteq}$ are extervals), or of the type **X** (for which Σ is the whole set of reals $\mathbb{R}$, and $\Sigma_{\supseteq}$ is an exterval), see Section III.5.2.3.

Corollary III.14.1 (Unbounded Σ and $\Sigma_{\supseteq}$) *The solution set $\Sigma([\![a_1\ a_2]\!], b)$ is always unbounded, while the set $\Sigma_{\supseteq}([\![a_1\ a_2]\!], b)$ is unbounded if at least one of the coefficients a_i is thick (or all are thin, including b).*

The first and the last part of the corollary above require of course some additional (easy) argument for the case of thin coefficients.

Despite appearances, from the fact that for all one-dimensional types the set $\Sigma_{\subseteq}$ is either empty or is an ordinary interval, it does not follow that the solution set $\Sigma_{\subseteq}([\![a_1\ a_2]\!], b)$ is always bounded.

Example III.10 (Unbounded $\Sigma_{\subseteq}$) Consider for example the expression $x_1 + x_2 = [1, 2]$. For it the sets $\Sigma_{\subseteq}([\![1\ 1]\!], [1, 2])$ and $\Sigma([\![1\ 1]\!], [1, 2])$ are equal and unbounded, comprising a strip bordered by two parallel lines $x_1 + x_2 = 1$ and $x_1 + x_2 = 2$ going through the quotient points $\mathrm{L} = \mathrm{S} = 1$ and $\mathrm{Z} = \mathrm{T} = 2$ on the x_1 and x_2 axes. The set $\Sigma_{\supseteq}([\![1\ 1]\!], [1, 2])$ is empty. ■

Proposition III.15 (Symmetric cuts) *There is always a direction in the Ox_1x_2 plane for which the cut through the solution sets has quotients (and, in effect, intersections of the cutting axis with borders of the solution sets) lying symmetrically with respect to the origin* O.

Proof. Let us consider the intersection of the trajectory with the **Or** axis, see (b) in Fig. III.65. The corresponding coefficient a_r, as lying on the **Or** axis, is a symmetric interval, i.e., such that $\underline{a}_r = -\overline{a}_r$. For this position of the coefficient in the MR-diagram, the only possible one-dimensional types of the cut are C_-, C_+, X_{b-}, and X_{b+}, see Section III.5.2.3. Quotients for these types lie symmetrically with respect to zero when the coefficient a is symmetrical. □

In the basic cases, there is only one such direction, namely that shown in the proof above. However, when the b coefficient is symmetrical, the quotients lie symmetrically around zero independently of the value of a, hence in that case *all* cuts are symmetrical.

Proposition III.16 (Proportional cuts) *Except for some special cases (like the symmetric cases described above, and some other), for a given cut there is another cut (along different direction), for which the arrangement of quotients (hence, arrangement of intersections of the cutting axis with borders of the solution sets) differs at most by some scale factor.*

Proof. As can be seen at (c) in Fig. III.65, interval axes intersecting the coefficient trajectory usually do it at two points, corresponding to different directions of the cutting axes (passing either through the same or different pair of quadrants). The coefficients lying on the same interval axis are related by a real factor: $a_p = t \cdot a_q$, $t \in \mathbb{R}^+$, and produce the same solution types. As a result, their quotients also differ only by the same factor t. □

III.5.3.5 Solution types in two dimensions

As the solutions are uniquely determined by the three intervals a_1, a_2, and b, the combined type of their structure can be uniquely characterized by the types of the constituent one-dimensional relations $a_1 \cdot x_1 \Diamond b$ and $a_2 \cdot x_2 \Diamond b$. Not all combinations of the types are possible, however, because the right-hand side interval b must be the same in both relations. Hence, excluding the cases of contradictory properties of b (the graph in Fig. III.66a, adapted from Fig. III.55e is useful here), and disregarding permutation of axes, one finds that only 9 basic types: UU, UX, XX, NN, ZZ, CC, NZ, NC, and ZC are possible, see Fig. III.66b. The basic types divide into 50 subtypes, corresponding to different possible combinations of the one-dimensional subtypes.

The counting of subtypes in Fig. III.66b was accomplished with the help of *combination graphs*, through enumeration of all combinations (denoted by edges of the graphs) of one-dimensional subtypes, except those forbidden due to sharing of the b coefficient (the forbidden combinations are those whose edges would cross the dotted horizontal lines in the graphs).

III.5.3.6 Enumeration of two-dimensional types

Figures III.67a–III.67d list typical examples of all 9 basic two-dimensional types of arrangements of solution sets for the interval expression (III.32). Every example is described by the trajectory of the a_p and a_q coefficients in the MR-diagram (with enumeration of subtypes along it, describing the corresponding rotated axes), and the diagram of the solution sets on the Ox_1x_2 plane. In order not to clutter unduly the diagram, the sequences of quotients along the Ox_1 and Ox_2 axes are shown separately from the axes themselves.

Every type is represented by some of its subtype(s). For most types, all the subtypes of the given type exhibit the same general topological pattern of solution sets on the Ox_1x_2 plane. Subtypes in these cases differ either by the quantitative values of quotients on the

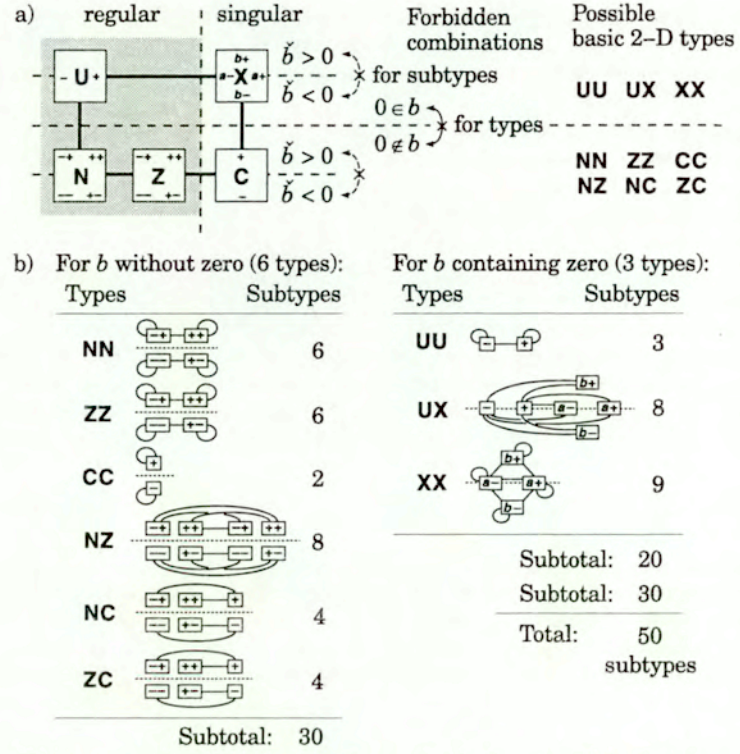


Figure III.66: A graph of one-dimensional types in the form useful for two-dimensional type analysis (a), and graphical enumeration of two-dimensional types and subtypes (b).

axes, or by reflections of the pattern in coordinate axes Ox_1 or Ox_2. For these types, only one example subtype is shown. Only for two types (**UX** and **XX**) the set of subtypes divides further into several topological subclasses (two subclasses for the **UX** type and three for the **XX** type). The subclasses differ by the distribution of the two intersection points constituting the $\Sigma_=$ solution set among different quadrants, but otherwise are of the same general structure. For these two types, an example from every topological subclass is shown also.

The representation in Figs. III.67a–III.67d is *qualitative*, i.e., no attempt is made to establish metric correctness of the correspondence between positions of coefficients a_1, a_2 and b in the MR-diagram and positions of the quotient points on the Ox_1 and Ox_2 axes. This character of the representation is indicated by the lack of numerical coordinates on the axes and by indicating the position of the b coefficient by showing only its interval axis. However, the important relation of parallelism between appropriate pairs of boundary lines is meticulously maintained in all diagrams.

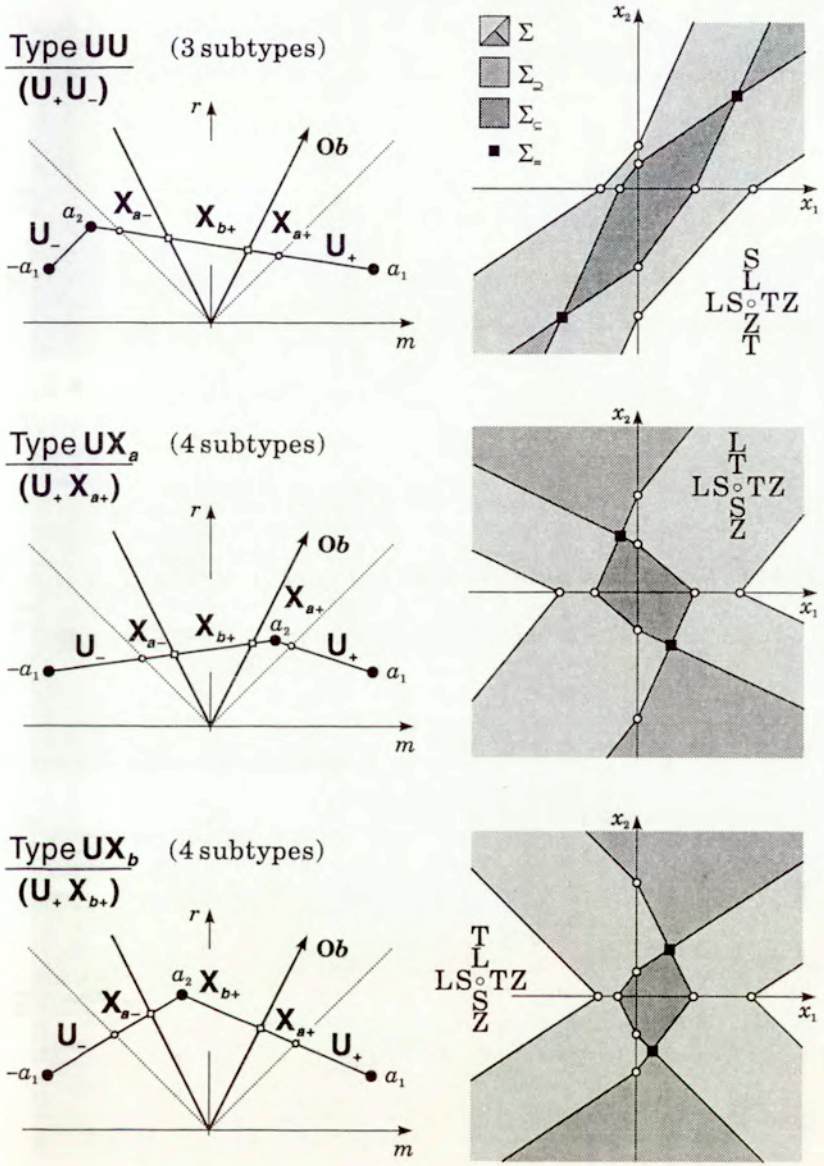


Figure III.67a: Enumeration of basic two-dimensional types (first part: the type UU, and two variants of the type UX).

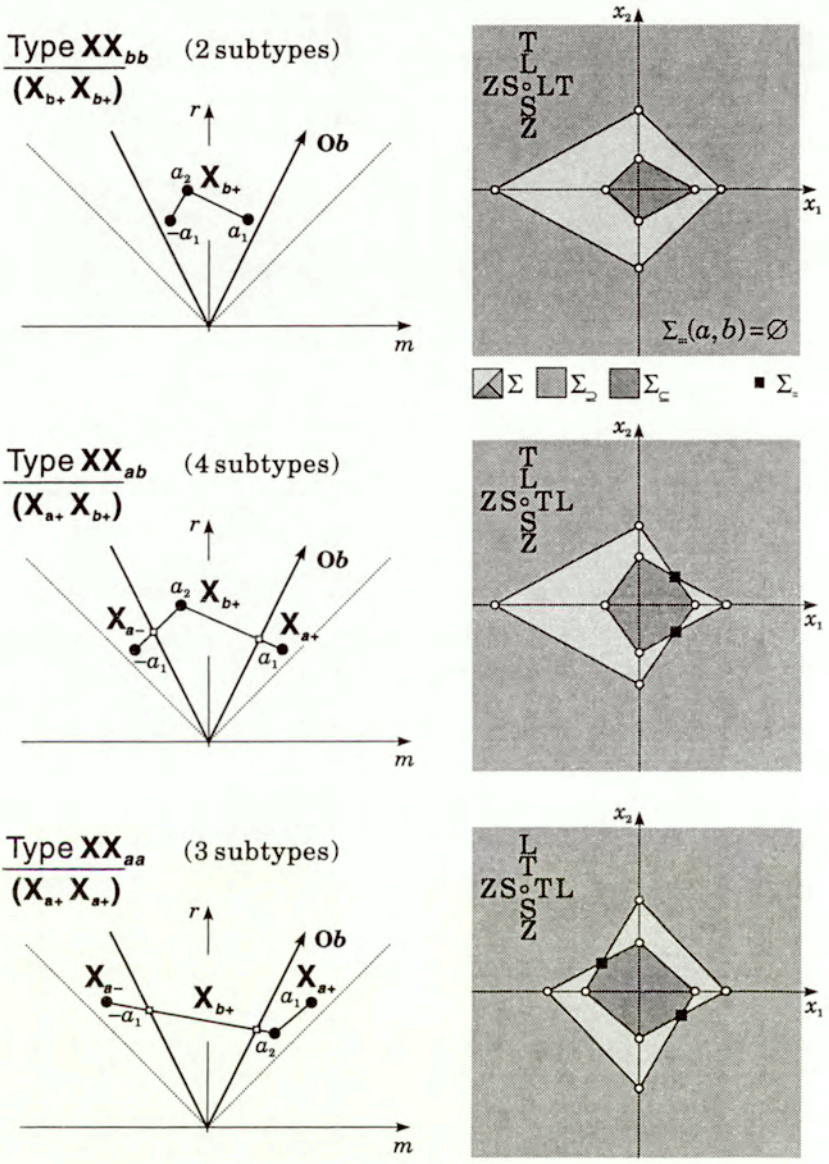


Figure III.67b: (cont.) Enumeration of basic two-dimensional types (second part: the type XX, three variants).

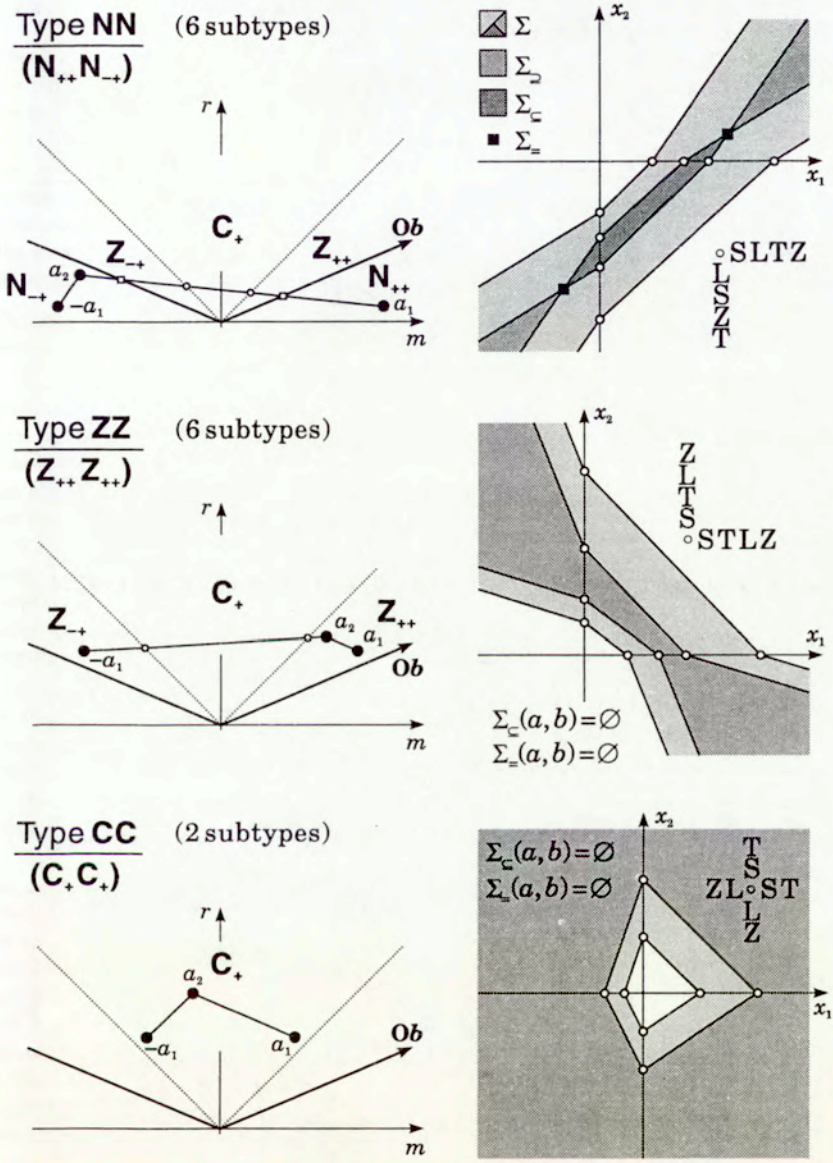


Figure III.67c: (cont.) Enumeration of basic two-dimensional types (third part: the types **NN**, **ZZ**, and **CC**).

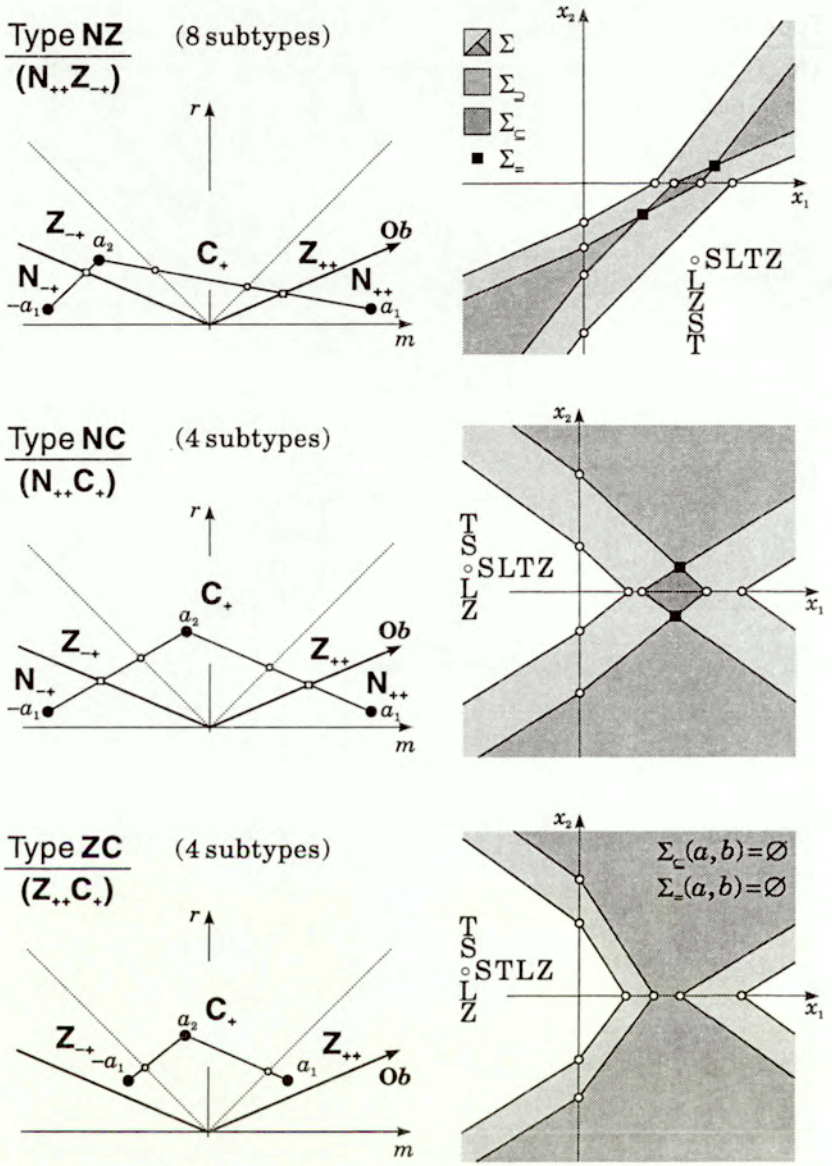


Figure III.67d: (cont.) Enumeration of basic two-dimensional types (the last part: the types NZ, NC, and ZC).

III.5.3.7 Intermediate cases

There are many intermediate (or degenerate) cases that do not fall exactly into one of the basic types enumerated in the previous section. They occur when one (or both) of the one-dimensional types of the cuts by the coordinate axes Ox_1 and Ox_2 belongs to an intermediate type, as described in Section III.5.2.5. Such situations occur when some of the coefficients a_1, a_2, or b lie on borders of the regions in the MR-diagram that define appropriate basic types. The structure and shape of the solution sets in such cases is intermediate between the basic types involved, with a number of characteristic features, like some quotients coinciding, or going to infinity (with corresponding boundary lines becoming parallel to one of the coordinate axes).

Example III.11 (Intermediate cases) The examples shown in Fig. III.68 illustrate several basic effects occurring in intermediate cases.

In the first example, both a_1 (hence $-a_1$ too), and a_2 lie on the $\mathbf{O}b$ axis. As a result, quotients L and T coincide. Because they delineate regions corresponding to solution sets $\Sigma_{\subseteq}$ and $\Sigma_{\supseteq}$ in the basic types **NN**, **ZZ**, and **NZ**, see Fig. III.67c and Fig. III.67d, that causes collapse of the set $\Sigma_{\subseteq}$ and a part of the set $\Sigma_{\supseteq}$ into a line segment, causing the segment to represent the set $\Sigma_{=}$ as well. According to Proposition III.12, because the whole segment of the coefficient trajectory corresponding to the quadrant (++) lies here on the $\mathbf{O}b$ axis, all cutting axes passing through this quadrant become the $\Sigma_{=}$ axes. The formulae (III.34) and (III.35) fail in this case, producing indeterminate results. However, the formula (III.35) which calculates the points belonging to the set $\Sigma_{=}$ gives correct results in the neighbouring quadrants (−+) and (+−), producing points on the coordinate axes that define the endpoints of the segment constituting the set $\Sigma_{=}$.

In the second example, due to the coefficient a_1, and in consequence, also $-a_1$, lying on the main diagonals in the MR-diagram, the quotients L_1 and Z_1 move to infinity, causing the corresponding boundary lines to become parallel to the Ox_1 axis. As the main diagonals in the MR-diagram constitute borders between regions of the types **Z** and **C**, the resulting structure of solution sets is intermediate between the two-dimensional types **ZC** and **CC**, cf. Figs. III.67c and III.67d.

The third example contains several effects of this kind. First, like in the second example, the Z_2 quotient runs to infinity due to the coefficient a_2 lying on the main diagonal **lb0**, causing several boundary lines to run vertically, in parallel to the Ox_2 axis. Next, like in the second example, the quotients S and L coincide due to the same coefficient a_2 lying on the $\mathbf{O}b$ axis. Moreover, as the coefficient b lies also on the main diagonal **lb0** (so that the axis $\mathbf{O}b$ coincides with the main diagonals), the quotients containing the starting point $\underline{b}$ of b, namely the quotients S and L again, become equal to zero. As a result, several boundary lines pass through the origin **O**, causing some problems with determination of their directions (the quotient-pair rule given in Theorem III.8 now cannot be directly applied). These problems are rather easy to overcome, resulting in the structure of solution sets combining features of the types **UU** and **UX**, see Fig. III.67a. ■

Defining and analysing in this manner other examples of intermediate types is an instructive exercise, giving many useful insights into the properties of two-dimensional relational expressions and their solution sets. The enumeration of all intermediate types, like it was done in Section III.5.2.5 for the one-dimensional case, is rather troublesome due to a considerable number of cases, and is omitted here.

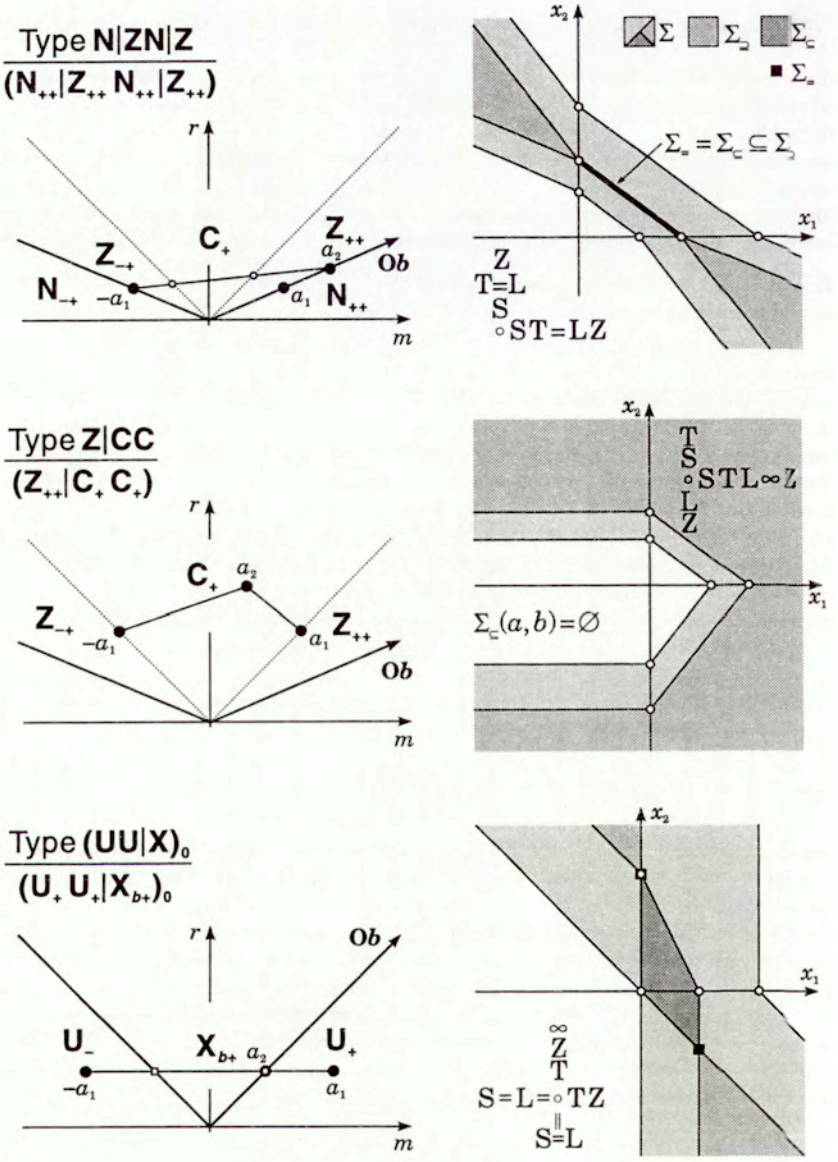


Figure III.68: Examples of intermediate two-dimensional types.

III.5.4 Generalization to n dimensions

... had ended up with elegant equations in innumerable unknowns.
[Isaac Asimov, *Prelude to Foundation* (1988)]

Generalizing the analysis to n-dimensional relational expressions, $n > 2$, of the form

$$\sum_{i=1}^{n} a_i \cdot x_i \Diamond b, \tag{III.36}$$

with $\Diamond \in \{\not\subset, \supseteq, \subseteq, =\}$, is quite straightforward, though actual drawing of some diagrams involved might be rather troublesome. In this section, the main results concerning this extension are summarized. Proofs are omitted; detailed exposition, including proofs, will be provided elsewhere. To simplify notation, in the sequel the sequence of the form $\epsilon(1)\,\epsilon(2)\ldots\epsilon(i)\ldots\epsilon(n)$ will be shortly written as $\ldots\epsilon(i)\ldots$, where $\epsilon(i)$ is any expression dependent on $i = 1, 2, \ldots, n$.

In the n-dimensional case, boundaries of solution sets become $n-1$ dimensional (hyper)planes, according to the definition:

Definition III.21 (Boundary hyperplane) *The hyperplane in the* $\mathrm{O}\ldots x_i\ldots$ *space with the equation* $\sum_{i=1}^{n} a_i^{\alpha_i} \cdot x_i = b^{\beta}$, *where* $\alpha_i, \beta \in \{-, +\}$, *is the* boundary hyperplane *for the expression (III.36), and is denoted as* $(\ldots\alpha_i\ldots\beta)$.

There are 2^{n+1} boundary hyperplanes, in 2^n pairs of parallel hyperplanes, as obviously $(\ldots\alpha_i\ldots -) \parallel (\ldots\alpha_i\ldots +)$.

Boundary hyperplanes intersect the $\mathrm{O}x_i$ axes at points $b^{\beta}/a_i^{\alpha_i} = Q_i^{\beta\alpha_i}$, i.e., at the quotients of one-dimensional relations $a_i \cdot x_i \Diamond b, i = 1, 2, \ldots, n$, obtained by setting $x_j = 0$ for all $j \neq i$ in the original expression (III.36). Therefore, they may be also denoted by quotient n-tuples $\ldots Q_i^{\beta\alpha_i}\ldots$, e.g. $(- + \cdots + -) = \mathrm{L}_1\mathrm{S}_2\ldots\mathrm{S}_n$. Through every quotient point $Q_i^{\beta\alpha_i}$ on the axes, exactly 2^{n-1} boundary hyperplanes pass, namely all the hyperplanes with codes $(\ldots\alpha_j\ldots\beta)$ exhausting all possible combinations of sign values of α_j variables other than α_i (i.e. such that $j \neq i$).

Let us denote the 2^n *orthants*, into which the axes divide the n-dimensional solution space, by n-tuples $s = (\ldots\sigma_i\ldots)$, $\sigma_i = \mathbf{sgn}\, x_i$, of signs of coordinates of the points lying in the given orthant, see Fig. III.69. Now, the axis through the origin, going from some orthant s to its opposite orthant $-s$ (where $-s = -(\ldots\sigma_i\ldots) = (\ldots\langle-\sigma_i\rangle\ldots)$, intersects an appropriate hyperplane at a single point, as described by the following theorem, generalizing Lemma III.3 and Lemma III.4:

Theorem III.9 (Generalized one-dimensional cuts) *Let* $0 \leq p_i \leq 1$; $i = 1, 2, \ldots, n$; $\sum_{i=1}^{n} p_i = 1$; $s = (\ldots\sigma_i\ldots)$, *and* $-\pi/2 \leq \varphi_i \leq \pi/2$ *such that* $\sigma_i = \mathbf{sgn}\,\varphi_i$ *and* $\cos\varphi_i = p_i/\sqrt{\sum_{i=1}^{n} p_i^2}$. *Then the axis* $\mathrm{O}x_s$, *going through the origin and through the point* $\mathrm{P} = (\ldots, \sigma_i p_i, \ldots)$, *pointing from the orthant* $-s$ *into the orthant* s, *and making angles* φ_i *with axes* $\mathrm{O}x_i$, *intersects the boundary hyperplane with the code* $(\ldots\langle\alpha\sigma_i\rangle\ldots\beta)$ *at the point* $\mathrm{X}_s^{\beta\alpha} = (\ldots, \sigma_i p_i, \ldots) \cdot Q_s^{\beta\alpha}$, *where* $Q_s^{\beta\alpha} = b^{\beta}/a_{ps}^{\alpha}$ *is the quotient of the interval relational expression* $a_{ps} \cdot x_{ps} \Diamond b$ *with the coefficient* a_{ps} *being a linear combination of coefficients* $\ldots, a_i, \ldots$ *determined by the parameter sequence* $\ldots, p_i, \ldots$ *and the signature of the orthant* s, *namely* $a_{ps} = \sum_{i=1}^{n} \sigma_i p_i a_i$.

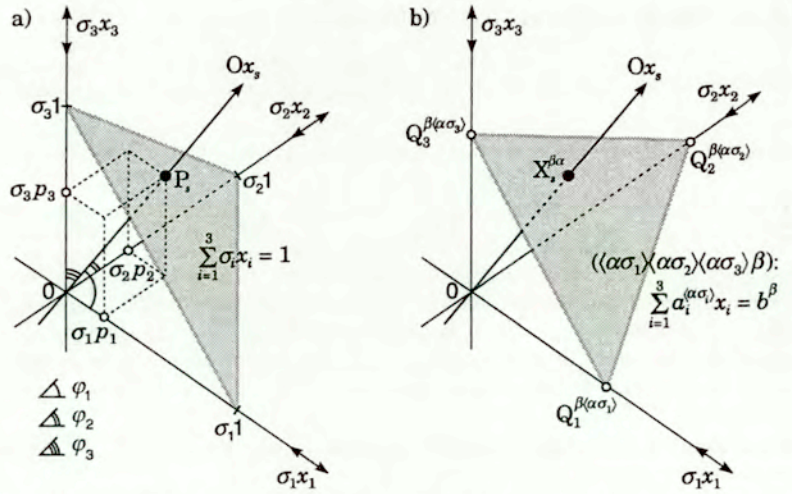


Figure III.69: Orthant specification, cutting axes, and boundary hyperplanes for $n = 3$: the Ox_s axis intersecting the hyperplane $\sum_{i=1}^{n} \sigma_i x_i = 1$ at $P = (\ldots, \sigma_i p_i, \ldots)$ (a), and the intersection of this axis with boundary hyperplane $(\ldots \langle\alpha\sigma_i\rangle \ldots \beta)$ at $X_s^{\beta\alpha}$ (b).

The proof of the theorem is quite straightforward, along the lines similar to proofs of Lemma III.3 and Lemma III.4, with the help of the construction shown in Fig. III.69. In Fig. III.69b, only one possible case of positioning the boundary hyperplane is shown, namely such that all points $Q_i^{\beta\langle\alpha\sigma_i\rangle}$ of intersection of the hyperplane with the axes lie in the orthant s (more accurately, on the σ_i-th halves of the Ox_i axes). Other cases are possible, when some of these points lie on the other halves of the coordinate axes (cf. Fig. III.60b and c).

For a given orthant $(\ldots \sigma_i \ldots)$, coefficients a_{ps} occupy in the MR-diagram an n-gon spanned over n interval coefficients $\ldots, \sigma_i a_i, \ldots$ determined by the orthant in question. As for $n > 3$ the full space of parameters $\ldots, p_i, \ldots$ is more than two-dimensional, always many different combinations of parameters $\ldots, p_i, \ldots$ project the corresponding a_{ps} onto the same point in the MR-diagram, thus generating identical one-dimensional cuts. It is an interesting phenomenon that occurs also for $n < 4$, but then only in some special cases. Namely, for $n = 2$ it occurs when $a_1 = \sigma a_2$, and for $n = 3$ also when $\sigma_1 a_1$, $\sigma_2 a_2$, and $\sigma_3 a_3$ are collinear in the MR-diagram (for some $\sigma, \sigma_i \in \{-, +\}$).

Similarly as for the two-dimensional case, from Theorem III.9 above it easily follows that borders of the solution sets can be selected from among boundary hyperplanes according to the following rule:

Theorem III.10 (Boundary hyperplanes selection) *The borders of the solution sets Σ, $\Sigma_{\supseteq}$, and $\Sigma_{\subseteq}$ in the orthants $s = (\ldots \sigma_i \ldots)$ and $-s = (\ldots \langle-\sigma_i\rangle \ldots)$ of the $O \ldots x_i \ldots$ solution space consist of exactly those parts of the boundary hyperplanes (included in this pair of opposite orthants) that have the codes of the form $(\ldots \langle\alpha\sigma_i\rangle \ldots \beta)$, where $\alpha, \beta \in \{-, +\}$.*

For a given pair of opposite orthants, there are thus exactly four (independently of n) boundary hyperplanes constituting borders of solution sets in these orthants, in two pairs of parallel hyperplanes, corresponding to all possible combinations of values of sign variables α and β in the formula $(\ldots\langle\alpha\sigma_i\rangle\ldots\beta)$ given in Theorem III.10.

The qualitative structure of solution sets is also uniquely described by the combination of types of the one-dimensional cuts by coordinate axes. The number of distinct types grows with n, but due to the limits imposed by sharing of the b coefficient, see Fig. III.66, the growth is only quadratic, instead of exponential, as might be expected otherwise. With the combinatorial analysis using the graphs of one-dimensional types, see Fig. III.66 in Section III.5.3.5, the number of types $t(n)$ and subtypes $s(n)$ can be found to conform to the formulae, for $n > 0$:

$$\begin{aligned} t(n) &= \frac{n(n+5)}{2} + 2, \\ s(n) &= 2n(9n-10) + 18. \end{aligned} \tag{III.37}$$

E.g., for $n = 3$ we have 14 basic types (**UUU**, **UUX**, **UXX**, **XXX**, **NNN**, **ZZZ**, **CCC**, **NNZ**, **NNC**, **ZZN**, **ZZC**, **CCN**, **CCZ**, **NZC**), while for $n = 4$ only 20 types. The number of subtypes is considerably larger (e.g., 120 subtypes for $n = 3$ and 386 for $n = 4$). Remember that the ordering of coordinate axes is irrelevant here, so that an n-dimensional (sub)type corresponds to a *set*, not an ordered n-tuple, of constituent one-dimensional (sub)types.

Putting $n = 2$ into the formulae in this section gives the results for the two-dimensional case described in Section III.5.3. The differences with the n-dimensional formulation of this section concern the use of directional cosines (as more customary in this context), and the use of n parameters $\ldots p_i \ldots$ with additional condition that their sum must equal 1. In the two-dimensional case we instead used tangents and a single parameter p or q (i.e., $n-1$ parameters), because it leads to simpler formulae for that case.

In addition to one-dimensional cuts, for $n > 2$ also k-dimensional cuts, $1 < k < n$, become possible. Especially the two-dimensional cuts are useful and easy to construct and analyze (using the approach developed in Section III.5.3).

III.5.5 Avenues for further research

Whether or not the goal is ever reached,
an awareness of the process brings enlightenment.
[Rudy Rucker, *Infinity and the Mind* (1982)]

There are many issues concerning interval relational expressions that were not touched in this exposition. They constitute avenues for further research in diagrammatic analysis of such expressions. Three issues of this kind are briefly introduced below.

III.5.5.1 Systems of relations

The analysis in the whole Section III.5 has been done for single relational expressions only. Solution sets for systems of such expressions are obtained as intersections of solution sets for individual expressions of the system. This leads to a much richer structure of the solution sets, producing new problems concerning qualitative classification of such systems (see [8] for some preliminary ideas). First, the composite type of the system must be represented as a *matrix of one-dimensional types*, where the order of the coefficients for a given expression (i.e., in a single row of the matrix) becomes significant. More importantly, shapes of the solution sets (and even their very existence) become dependent not only on the type matrix, but also on metric (quantitative) relations between coefficients of the constituent relational expressions. That complicates the qualitative analysis significantly, requiring new methods of type description (again, see [8] for some hints).

Extending diagrammatic analysis to general systems of interval relations will provide a necessary connection between the analysis in this section and the ideas on shape classification of solution sets started in [8].

III.5.5.2 Rohn's A_{yz} matrices

In his widely applauded talk [Rohn 2000] at the SCAN/INTERVAL 2000 Conference in Karlsruhe, Professor Rohn summarized many important results and algorithms concerning interval matrices and interval linear equations, all formulated in terms of *vertex matrices* A_{yz} and b_y, introduced by him in [Rohn 1989], see Definition III.3 in Section III.1.1.1. For these matrices, $(A_{yz})_{ij} \in \{\underline{a}_{ij}, \overline{a}_{ij}\}$ and $(b_y)_i \in \{\underline{b}_i, \overline{b}_i\}$, where a_{ij}, b_i are intervals, and y, z are vectors of the form $[\![\ldots\sigma 1\ldots]\!]^T$, $\sigma \in \{-,+\}$, used to select appropriate endpoints of the intervals a_{ij} and b_i.

As it is thus obvious, for all y and z, rows of any vertex system of equations $A_{yz} \cdot x = b_y$ correspond to boundary hyperplanes in our formulation. Hence, the analysis in terms of the vertex matrices can be also conducted diagrammatically in terms of boundary hyperplanes, which may provide new insights into these matters, and, possibly, also new interesting results in this area.

III.5.5.3 Directed (modal) intervals and generalized solution sets.

Another obvious extension of the approach concerns analysis of the interval linear equations in the algebra of directed (or modal) intervals. The extension of the MR-diagram to include also improper intervals is rather straightforward (they occupy the lower half-plane

of the diagram, for negative radius, see Section III.4.4. Also, arithmetic on directed intervals is easily expressible in such an extended MR-diagram. Thus, there should be few problems with the development of diagrammatic approach to linear equations in this algebra, especially as Shary in [Shary 1996] already obtained several basic results on dependencies between the solution sets of proper and dual equations, i.e., equations obtained by replacing (some of) their proper interval coefficients by dual improper ones. In yet another extension, the so-called *generalized solution sets* developed by Shary, see e.g. [Shary 2002], different quantifiers can be applied to different coefficients in the definition of a solution set of the form given in Section III.5.1. That produces many different new types of solution sets. However, as worked out by [Shary 2002] (with some help of the modal interpretation of directed intervals [Gardeñes et al. 2001]), again they are related to the equations in the directed interval algebra, so that they should cause little additional trouble for the diagrammatic approach developed here.

Summary

I only require a few missing links
to have an entirely connected case.
[Arthur Conan Doyle, *The Sign of Four* (1890)]

The results of the work are threefold. In a short outline, the work integrates general considerations on visual information processing in humans and machines with the exposition of the emerging research field of diagrammatics, culminating in the derivation of new diagrammatic tools for the field of interval analysis.

In Chapter I, the *general framework for the visual information processing area* is proposed and then used to structure a brief survey of several selected picture processing subfields, including the author's main contributions to them. Chapter II contains an *introduction to main issues of the newly emerging field of diagrammatics*, containing also several original contributions by the author. The main result of the work, presented in Chapter III, is the development of the *diagrammatic notation for the algebra of intervals*, including demonstration of its usefulness for several topics in this area. More detailed account of the results is provided in the sequel.

I. Picture processing in humans and machines. The first chapter starts from general discussion of the issues of using pictures as tools for information storage and communication, especially the processes of *interpretation of pictures* and *pictorial representation of knowledge*, both by humans and in computers. A unified framework for classifying and relating various subfields and aspects of that large domain is proposed by the author, in the form of a three-level integrated schema combining basic types of corresponding subprocesses of interpretation and representation, and data structures used by them. The framework is then used to delineate and relate several selected research and application areas in the field, namely, *computer picture processing systems* (Section I.2), *discrete picture processing* (Section I.3), *discrete image analysis* (Section I.4), and *scene understanding* (Section I.5). The basic issues in these areas are reviewed, with contributions to them by this author's outlined on that general background. These contributions include:

- The development of the software support for early Polish computer picture processing systems, including construction of several *picture processing programming languages* (Section I.2.1).
- The uniform formulation of the basic formalism for discrete picture operations, including *parallel* and *serial* operations on pictures and *planar grammars* (Section I.3).

- The reliable and practical methods of *area* and *perimeter* measurements of objects on discrete pictures (Sections I.4.2 and I.4.3).
- Several results concerning *impossible figures*, i.e., the illusions of spatial interpretation of pictures (Section I.5.2). They include the clarification of the definition of these figures, general classification of them, and finding all the basic types of spatial contradictions occurring in them.

The chapter concludes with a short description of the emerging field of diagrammatics, especially concerning its place in the general framework proposed in the chapter.

II. Diagrammatics: an introduction. This chapter contains a general survey of the most important topics of diagrammatics, especially concerning the issue of more or less formal diagrammatic reasoning. The survey in part relates findings reported in the relevant literature, and partially introduces new ideas and results obtained by this author. The latter range from proposals of new terminology (e.g., "divergence" in Section II.4.3), or clarification of some common but often imprecisely used notions (e.g. in Section II.1.3), to several new illustrative examples and reformulation or new analysis of the old ones, to several new ideas and proposals. The more important original ideas here include:

- The new look at several alleged problems with and limitations of diagrams (Section II.3.2), especially a comprehensive discussion of the problems of *impreciseness of diagrams* (Section II.3.2.1), representation of *incomplete information* and *disjunctive knowledge* (Section II.3.2.2), the effects of *particularity* (Section II.3.2.3), *accidental alignments* (Section II.3.2.4), and *specificity* (Section II.3.2.5), see also the summary discussion of possible errors in diagrammatic reasoning included in Section II.5.2.
- The new general classification of diagrammatic reasoning modes (Section II.4.1).
- The preliminary survey and classification of visual language styles used in mathematical diagrams (Section II.5.4).
- The comprehensive answer to the main arguments being raised against a wider use of diagrams in mathematics, showing these arguments to be generally unfounded (Section II.5).
- The *diagrammatic spreadsheet* concept for computer implementation of diagrammatic tools (Section II.6.4).

III. Diagrammatic interval algebra. Simple diagrams appeared in the interval literature from the very beginning of the field, but were used only rarely, as sketchy and informal illustrations for some basic concepts. They were neither systematically investigated, nor more widely applied in interval research and applications. It seems that systematic investigation of interval space diagrams and their prospective applications started only with the works of this author, see especially [7, 12, 13, 26, 107].

In Chapter III, after a short introduction to basic concepts of interval calculations, the diagrammatic representation for interval space developed by the author is described, and then its use in various interval analysis areas is demonstrated in some detail. Several types of diagrams and associated tools are developed and described, starting from the

main *MR-diagram* of the two-dimensional space of real intervals (in Section III.2.2) on which most of the further diagrammatic tools are based more or less directly. Further diagrammatic tools developed by the author include the *W-diagrams* and the *L-diagrams* useful in analysis of arrangement interval relations (described in Section III.3.2), a number of diagrams for interval arithmetic operations (in Section III.4), and several types of diagrams used for diagrammatic solution and analysis of basic linear interval equations (in Section III.5). The latter include the *quotient diagram* (Sections III.5.2.2 and III.5.2.3), a diagrammatic catalogue of all solution types of the basic one-dimensional equation (Sections III.5.2.3 and III.5.2.5), the *RR-diagram* and *graphs of types* (Section III.5.2.6), and the *butterfly diagram* (Section III.5.3.4)

These tools are used to treat several issues of interval algebra, producing several original results. The problems analyzed diagrammatically concern:

- The representation and characterization of *interval types*, comparison of *extent functions*, and *lattice structure* of the space of intervals (Section III.2.3.3).
- The representation and characterization of *interval relations* (especially arrangement relations), with diagram-aided proofs of two theorems on characterization of *convex* and *pointisable* interval relations (Sections III.3.3 and III.3.4), including also new properties of convex relations.
- The characterization of solution spaces of two *basic interval equations* $a + x = b$ and $a \cdot x = b$ (Sections III.4.1.3 and III.4.2.3) and a diagrammatic proof of the conditions for so-called *fast interval multiplication* (Section III.4.2.2).
- The analysis, classification, and characterization of solution sets of the *interval linear equations* of the form $\sum_{i=1}^{n} a_i \cdot x_i = b$, where a_i and b are intervals (Section III.5).

The latter application of the diagrammatic notation developed by the author led to several original results, including:

- The complete classification of *basic types* and *subtypes* of solution sets of the equation in one dimension (Section III.5.2), two dimensions (Section III.5.3) and generalization to n dimensions (Section III.5.4). The classification allows for actual construction of the solution sets and derivation of their qualitative (topological) structures (as is shown in Fig. III.67a-d in Section III.5.3.5 for the two-dimensional case).
- The comprehensive characterization of the relations between the so-called algebraic solution and other kinds of solutions in the one-dimensional case (Section III.5.2.4).
- Derivation of several general properties of the two-dimensional solution sets and finding formulas for characteristic directions in the space of solutions in Section III.5.3.4.

The above results were made possible by discovering the one-to-one correspondence between one-dimensional cuts through the solution space and points on the so-called *butterfly diagram* spanning the representations of the equation's coefficients in the MR-diagram, see Section III.5.3.2.

It is hoped that the diagrammatic system developed there may play a similar role in further development of interval algebra as the complex plane diagram in the past did in complex analysis.

Bibliography:
Author's publications

Papers in refereed international journals

[1] Z. Kulpa, S. Markov (2003) On the inclusion properties of interval multiplication: A diagrammatic study. *BIT* (accepted for publication).

[2] Z. Kulpa, (2003) Self-consistency, imprecision, and impossible cases in diagrammatic representations. In: [77], 147–160.

[3] T.L. Le, Z. Kulpa, (2003) Diagrammatic spreadsheet. In: [77], 133–146.

[4] Z. Kulpa (2003) Diagrammatic analysis of interval linear equations. Part II: The two-dimensional case and generalization to n dimensions. *Reliable Computing*, 9(3): 205–228.

[5] Z. Kulpa (2003) Diagrammatic analysis of interval linear equations. Part I: Basic notions and the one-dimensional case. *Reliable Computing*, 9(1): 1–20.

[6] K. Rosłaniec, Z. Kulpa, M. Kleiber (2002) Qualitative model-based analysis of truss structures. *Computer Assisted Mechanics and Engineering Sciences*, 9(1): 123–133.

[7] Z. Kulpa (2001) Diagrammatic representation for interval arithmetic. *Linear Algebra and Its Applications*, 324: 55–80.

[8] Z. Kulpa, K. Rosłaniec (2000) Solution sets for systems of linear interval equations. *Computer Assisted Mechanics and Engineering Sciences*, 7(4): 625–639.

[9] Z. Kulpa, T.L. Le (2000) Characterization of convex and pointisable interval relations by diagrammatic methods. *Machine GRAPHICS & VISION*, 9: 221–231.

[10] Z. Kulpa, A. Radomski, O. Gajl, M. Kleiber, I. Skalna (1999) Hybrid expert system for qualitative and quantitative analysis of truss structures. *Engineering Applications of Artificial Intelligence*, 12(1): 229–240.

[11] Z. Kulpa, A. Pownuk, I. Skalna (1998) Analysis of linear mechanical structures with uncertainties by means of interval methods. *Computer Assisted Mechanics and Engineering Sciences*, 5(4): 443–477.

[12] Z. Kulpa (1997) Diagrammatic representation for a space of intervals. In: [78], 5–24.

[13] Z. Kulpa (1997) Diagrammatic representation of interval space in proving theorems about interval relations. *Reliable Computing*, 3: 209–217.

[14] M. Kleiber, Z. Kulpa (1995) Computer-assisted hybrid reasoning in simulation and analysis of physical systems. *Computer Assisted Mechanics and Engineering Sciences*, 2(3): 165–186.

[15] Z. Kulpa (1994) Diagrammatic representation and reasoning. *Machine GRAPHICS & VISION*, 3(1-2): 77–103.

[16] Z. Kulpa (1987) Putting order in the impossible. *Perception*, 16: 201–214.

[17] Z. Kulpa, B. Kruse (1984) Algorithms for circular propagation in discrete images. *Computer Vision, Graphics and Image Processing*, 24: 305–328.

[18] Z. Kulpa (1983) More about areas and perimeters of quantized objects. *Computer Vision, Graphics and Image Processing*, 22: 268–276.

[19] Z. Kulpa (1983) Are impossible figures possible? *Signal Processing*, 5(3): 201–220.

[20] Z. Kulpa, M. Doros (1981) Freeman digitization of integer circles minimizes the radial error. *Computer Graphics and Image Processing*, 17: 181–184.

[21] Z. Kulpa (1979) On the properties of discrete circles, rings and disks. *Computer Graphics and Image Processing*, 10: 348–365.

[22] Z. Kulpa (1977) Area and perimeter measurement of blobs in discrete binary pictures. *Computer Graphics and Image Processing*, 6(5): 434–451.

[23] Z. Kulpa (1977) Planar grammars, parallel picture processing algorithms and their equivalence. *Control and Cybernetics*, 6(2): 5–16.

[24] Z. Kulpa (1977) Sistemy analiza graficheskikh izobrazhenyi i ikh programmnoye obezpechenye [Graphical image analysis systems and their software, in Russian]. *Izvestya AN SSSR - Tekhnicheskaya Kibernetika*, 1977(4): 82–88.

Papers in refereed proceedings and collections

[25] Z. Kulpa (2000) A diagrammatic notation for interval algebra. In: [DIAGRAMS 2000a], 471–474.

[26] Z. Kulpa (2001) Towards diagrammatic analysis of systems of interval "linear equations." In: [INTERVALS 2001], 115–126.

[27] Z. Kulpa, M. Sobolewski (1992) Knowledge-directed graphical and natural language interface with a knowledge-based concurrent engineering environment. In: *Proc. CARs & FOF: 8th International Conference on CAD/CAM, Robotics and factories of the Future* (Metz, France, 1992), vol. 1: 238–248.

[28] Z. Kulpa, M. Sobolewski, S.N. Dwivedi (1990) Graphical user interface with object-oriented knowledge-based engineering environment. In: S.N. Dwivedi, A.K. Verma, J.E. Sneckenberger, eds.: *CAD/CAM, Robotics and factories of the Future '90, Vol. 1: Concurrent Engineering*, Springer-Verlag, Berlin, 154–159.

[29] M.W. Sobolewski, Z. Kulpa (1984) From sentences to attribute networks. In: I. Plander, ed.: *Artificial Intelligence and Information-Control Systems of Robots* (Proc. 3rd International Conference on..., Smolenice, Czechoslovakia 1984), Elsevier (North-Holland), Amsterdam, 345–348.

[30] Z. Kulpa (1982/83) Iconics: Computer-aided visual communication. In: S. Levialdi, ed.: *Digital Image Analysis* (Proc. 2nd Conference on Image Analysis and Processing, Fasano 1982). Pitman, London 1983, 280–282.

[31] Z. Kulpa (1982/83) Impossible figures: illusion of spatial interpretation of pictures. In: S. Levialdi, ed.: *Digital Image Analysis.* (Proc. 2nd Conference on Image Analysis and Processing, Fasano 1982). Pitman, London 1983, 140–143.

[32] Z. Kulpa (1981) Universal image processing and analysis systems—an overview of the European scene. In: M. Kunt, F. de Coulon, eds.: *Signal Processing: Theories and Applications.* North-Holland, Amsterdam, 7–14.

[33] Z. Kulpa (1981) PICASSO, PICASSO-SHOW and PAL—a development of a high-level software system for image processing. In: M.J.B. Duff, S. Levialdi, eds.: *Languages and Architectures for Image Processing,* Academic Press, London, 13–24.

[34] Z. Kulpa, M. Piotrowicz (1979/85) Shape factors of figures in discrete pictures. In: *Selected papers of the Third National Conference on Biocybernetics and Biomedical Engineering* (Warsaw 1979), Polish Scientific Publishers (PWN), Warsaw 1985, 283–296.

[35] Z. Kulpa, A. Gutowska (1979/80) Measurement of limb movement coordination in cats using universal computer image processing system CPO-2. In: A. Morecki, K. Fidelius, eds.: *Biomechanics VII-A* (Proc. VIIth International Congress of Biomechanics, Warsaw 1979), Polish Scientific Publishers (PWN), Warsaw 1980, 471–477.

[36] Z. Kulpa, A. Bielik, M. Piotrowicz, M. Rychwalska (1978) Measurements of shape characteristics of moving cells using computer image processing system CPO-2. In: *Proc. International Conference on Signals and Images in Medicine and Biology (BIOSIGMA'78),* Paris, 286–292.

[37] Z. Kulpa, H.T. Nowicki (1976) Simple interactive picture processing system PICASSO-SHOW. In :*Proc. 3rd International Joint Conference on Pattern Recognition,* San Diego, CA, 218–223.

[38] Z. Kulpa, J. Dernałowicz, M. Raczkowska, M. Piotrowicz (1976/83) Digital picture processing system CPO-2 and its biomedical applications. In: M. Nałęcz, ed.: *Selected Papers of the First National Conference on Biocybernetics and Biomedical Engineering* (Warsaw 1976), Polish Scientific Publishers (PWN), Warsaw 1983, 312–326.

[39] Z. Kulpa (1974/75) On the equivalence of planar grammars and parallel picture processing algorithms. In: A. Blikle, ed.: *Mathematical Foundations of Computer Science '74,* Lecture Notes in Computer Science 28, Springer-Verlag, Berlin 1975, 307–312.

[40] Z. Kulpa (1974) Zarys konstrukcji języka analizy obrazów graficznych PAL [Outline of a construction of the language PAL for analysis of graphical images, in Polish]. In: *Proc. 6th National Automatic Control Conference*, Poznań, vol. I: 686–694.

[41] Z. Kulpa (1974) An outline description of the picture analysing language PAL. In: *Proc. 9th Yugoslav International Symposium on Information Processing (INFORMATICA 74)*, Bled, Yugoslavia, 6 pp.

[42] Z. Kulpa (1972) A picture processing system PICTURE ALGOL 1204. In: *Proc. 7th Yugoslav International Symposium on Information Processing (FCIP'72)*, Bled, Yugoslavia, 6 pp.

[43] R.S. Michalski, Z. Kulpa (1971/72) A system of programs for the synthesis of switching circuits using the method of disjoint stars. In: C.V. Freiman, ed.: *Information Processing 71* (Proc. of the IFIP Congress, Ljubljana 1971), North-Holland, Amsterdam 1972, Vol. 1: 61–65.

Invited papers and lectures

[44] Z. Kulpa (2000) *Podstawy diagramatyki* [*Foundations of Diagrammatics*]. Lectured at: Bielsko College of Business and Computer Science, Bielsko-Biała. (Unpublished lecture notes.)

[45] Z. Kulpa (1994) Diagrammatic representation and reasoning. Invited paper for: *3rd International Computer Graphics and Image Processing Conference (GKPO'94)*, Spała, Poland, 1994. (Published as [15].)

[46] Z. Kulpa (1986) Visual computing: a new quality in computer education and art. Lectured at: *Eleventh National Summer School PROGRAMMING'86*, Primorsko, Bulgaria. (Unpublished lecture notes: 7 pp.)

[47] Z. Kulpa (1983) Pictorial communication of information using digital images manipulation. Presented at: Third National Scientific-Technological Conference "Television Technology'83," Sofia, Bulgaria. Abstracts: 22; (Unpublished lecture notes: 8 pp.)

[48] Z. Kulpa (1983) Iconics: computer-aided visual communication. Lectured at: *Eight National Summer School PROGRAMMING'83*, Primorsko, Bulgaria. (Published in: M. Barneva, ed.: *PROGRAMMIRANE'83*, Sofia, Bulgaria: 145–157; and also, in Bulgarian, as [84].)

[49] Z. Kulpa (1980) Universal image processing and analysis systems—an overview of the European scene. Invited paper for: *1st European Signal Processing Conference (EUSIPCO-80)*, Lausanne. (Published as [32].)

[50] Z. Kulpa (1979) PICASSO, PICASSO-SHOW and PAL—a development of a high-level software system for image processing. Invited paper for: *Workshop on High-level Languages for Image Processing*, Windsor, England. (Published as [33].)

Other conference papers

[51] Z. Kulpa, (2002) Self-consistency, imprecision, and impossible cases in diagrammatic representations [Extended Abstract]. In: E. Grabska, Z. Kulpa, eds.: *First European Workshop on "Diagrammatics and Design": Extended Abstracts.* BCBCS, Bielsko-Biała, Poland, 45–46. (Full version published as [2].)

[52] T.L. Le, Z. Kulpa, (2002) Diagrammatic spreadsheet [Extended Abstract]. In: E. Grabska, Z. Kulpa, eds.: *First European Workshop on "Diagrammatics and Design": Extended Abstracts.* BCBCS, Bielsko-Biała, Poland, 41–42. (Full version published as [3].)

[53] K. Rosłaniec, Z. Kulpa, (2000) System ekspertowy jakościowej analizy kratownic metodą propagacji przemieszczeń [An expert system for qualitative truss analysis using the method of displacement propagation, in Polish]. In: Z. Bubnicki, A. Grzech, eds.: *Inżynieria wiedzy i systemy ekspertowe* [*Knowledge Engineering and Expert Systems*]. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, vol. 2: 297–304.

[54] Z. Kulpa, A. Radomski, O. Gajl, M. Kleiber, I. Skalna (1997) Hybrydowy system ekspertowy jakościowo-ilościowej analizy układów mechanicznych [Hybrid expert system for qualitative and quantitative analysis of mechanical structures, in Polish]. In: Z. Bubnicki, A. Grzech, eds.: *Inżynieria wiedzy i systemy ekspertowe* [*Knowledge Engineering and Expert Systems*]. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, vol. 2: 135–142.

[55] Z. Kulpa (1994) Diagrammatic representation and reasoning with applications in qualitative analysis. In: M. Akiyama, M. Kleiber, eds.: *Proc. Japan-Central Europe Joint Workshop on Advanced Computing in Engineering*, Pułtusk, Poland, 357–359.

[56] M. Kleiber, Z. Kulpa (1993) Computer-aided qualitative analysis: a key to effective simulation and analysis of physical systems? In: *Proc. Japanese-Polish Joint Seminar on Advanced Computer Simulation*, Tokyo, 123–130.

[57] Z. Kulpa (1985) Putting order in the impossible. In: *Proc. 16th Meeting of the European Mathematical Psychology Group*, Montpellier, France, 127–144.

[58] J. Dernałowicz, Z. Kulpa (1985) Komputerowe odwzorowanie schematycznych map (anatomicznych) w odniesieniu do obiektu obrazowego [Computer rendering of schematic (anatomical) maps in relation to the pictorial object, in Polish]. In: *Proc. 1st Conference "Computers in Medicine" (MIPOL-85)*, Wrocław, Poland, 246–248.

[59] Z. Kulpa (1983) Ikonika: komunikacja wizualna wspomagana komputerowo i jej zastosowania biomedyczne [Iconics: Computer-aided visual communication and its biomedical applications, in Polish]. In: *Proc. 6th National Conference "Biocybernetics and Biomedical Engineering,"* Warsaw, 363–365.

[60] J. Dernałowicz, Z. Kulpa (1983) Odwzorowanie schematycznych map anatomicznych dla celów interakcji człowiek-maszyna [Rendering of schematic anatomical maps for man-machine interaction, in Polish]. In: *Proc. 6th National Conference "Biocybernetics and Biomedical Engineering,"* Warsaw, 485–487.

[61] Z. Kulpa (1981) Image processing of biological shapes. In: *Proc. 14th European Meeting of Statisticians*, Wrocław, Poland, 50–51.

[62] M. Piotrowicz, Z. Kulpa (1980) Determination of profiles of banded chromosomes using computer image processing system CPO-2. In: *Proc. 1st European Signal Processing Conference (EUSIPCO-80)*, Lausanne, Short Communication and Poster Digest, 83-84.

[63] Z. Kulpa, A. Gutowska (1980) Limb movement coordination in cats measured by universal computer image processing system CPO-2. In: *Proc. 1st European Signal Processing Conference (EUSIPCO-80)*, Lausanne, Short Communication and Poster Digest, 85.

[64] Z. Kulpa (1980) Development of a high-level software system for image processing—a case study. In: *Proc. of The First International Workshop on Natural Communication with Computers (NCC)*, Warsaw: 71–73.

[65] Z. Kulpa, M. Piotrowicz (1979) Określanie profili chromosomów przy pomocy komputerowego systemu analizy obrazów CPO-2 [Determination of chromosome profiles using computer image analysis system CPO-2, in Polish]. In: *Proc. 4th National Conference "Biocybernetics and Biomedical Engineering,"* Poznań, 337–338.

[66] Z. Kulpa, M. Piotrowicz (1979) Współczynniki kształtu figur na obrazach dyskretnych [Shape factors of figures in discrete pictures, in Polish]. In: *Proc. 3rd National Conference on Biocybernetics and Biomedical Engineering*, Warsaw, 245–246. (Extended version published as [34].)

[67] Z. Kulpa, J. Dernałowicz (1978/81) Digital image analysis system CPO-2/K-202, general hardware and software description. In: S. Levialdi, ed.: *Pattern Recognition of Biomedical Objects* (Proc. 4th Polish-Italian Biomedical Symposium, Porto Ischia 1978), Quaderni de "la Ricerca Scientifica," vol. 108, CNR, Roma 1981, 195–201.

[68] A. Bielik, Z. Kulpa, M. Piotrowicz, M. Rychwalska (1978/81) Use of computer picture processing in quantitative morphology of biological cells. In: S. Levialdi, ed.: *Pattern Recognition of Biomedical Objects*, (Proc. 4th Polish-Italian Biomedical Symposium, Porto Ischia 1978), Quaderni de "la Ricerca Scientifica," vol. 108, CNR, Roma 1981, 77–90.

[69] Z. Kulpa, A. Bielik, M. Piotrowicz, M. Rychwalska (1978) Ilościowe pomiary zmian kształtu komórek w ruchu przy użyciu systemu cyfrowego przetwarzania obrazów CPO-2 [Quantitative measurements of shape changes of moving cells using computer image processing system CPO-2, in Polish]. In: *Proc. 2nd National Conference "Biocybernetics and Biomedical Engineering,"* Gliwice, Poland, 161–162.

[70] Z. Kulpa, M. Sobolewski (1977) Obrabotka i raspoznavanye izobrazhenyi s pomoshchyu universalnoy sistemy CPO-2/K-202 [Processing and recognition of images using the universal system CPO-2/K-202, in Russian]. In: *Proc. BIONIKA'78*, Leningrad, USSR, vol. I: 182-192.

[71] Z. Kulpa, J. Dernałowicz (1977) Digital picture processing system CPO-2 and its biomedical applications. In: *Proc. BIONIKA'77*, Dom Techniky SVTS, Bratislava, vol. III.

[72] Z. Kulpa, H.T. Nowicki (1977) Simple interactive picture processing system "PICASSO-SHOW." In: *"Experiences of Interactive System Use": Proc. International Seminar*, Szklarska Poręba; Prace Naukowe ICT PWr, Wrocław Technical University, Wrocław, 51/16: 101–115.

[73] Z. Kulpa, J. Dernałowicz (1976) System cyfrowej analizy obrazów CPO-2 i jego zastosowania biomedyczne [Picture processing system CPO-2 and its biomedical applications, in Polish]. In: *Proc. 1st National Conference "Biocybernetics and Biomedical Engineering,"* Warsaw, 182–183. (An extended version published as [38].)

[74] Z. Kulpa (1975/76) Sistemy analiza graficheskikh izobrazhenyi CPO-1 i CPO-2 i ikh programnoye obespechenye [Systems for analysis of graphical images CPO-1 and CPO-2 and their software, in Russian]. In: J. Karczewski., ed.: *Proc. Extended Meeting of the Working Group 2 KNWWT on Methods of Information Recognition, Classification and Search.* Jadwisin, Poland 1975. Instytut Organizacji i Kierowania, Warszawa 1976, 369–390.

[75] Z. Kulpa, M. Piotrowicz (1975) Practical methods for measuring area and perimeter of discrete real pictures. In: *Proc. 6th von Neumann Colloquium on Computing and Cybernetic Methods in Medicine and Biology.* Szeged, Hungary, 59–71.

[76] Z. Kulpa (1973/74) Opis struktury obrazów graficznych [Structural description of graphical images, in Polish]. In: *Proc. 1st National Symposium "System-Modelling-Control,"* Polish Cybernetical Society, Łódź, 151–161.

Article collections, special issues

[77] E. Grabska, Z. Kulpa, eds. (2003) *Diagrammatics & Design* (Selected Papers from the First European Workshop on "Diagrammatics and Design"). A Special Issue of *Machine GRAPHICS & VISION*, 12(1).

[78] Z. Kulpa, ed. (1997) *Diagrammatic representation and reasoning.* A Special Issue of *Machine GRAPHICS & VISION*, 6(1).

[79] L. Bolc, Z. Kulpa, eds. (1981) *Digital Image Processing Systems.* Lecture Notes in Computer Science 109, Springer-Verlag, Berlin.

[80] M. Nałęcz, S. Topiński, Z. Kulpa et al., eds. (1977) *System cyfrowej analizy obrazów CPO-2* [*Digital Image Analysis System CPO-2*, in Polish]. Reports of the Institute of Biocybernetics and Biomedical Engineering of the Polish Academy of Sciences, vol. 1, Warsaw.

Other papers

[81] M. Nieniewski, L. Chmielewski, Z. Kulpa (2002) Przetwarzanie obrazów, wizja komputerowa oraz graficzna reprezentacja wiedzy [Picture processing, computer vision and graphical knowledge representation, in Polish]. In: M. Kleiber, ed.: *Nauki techniczne u progu XXI wieku — Wizja rozwoju wybranych dyscyplin z perspektywy IPPT PAN* [*Technical Sciences at the Threshold of the XXI Century—The Vision of Development of Selected Disciplines from the IFTR PAS Perspective*]. Institute of Fundamental Technological Research, Warsaw, 211–228.

[82] Z. Kulpa, M. Kleiber (1998) Jakościowa analiza układów mechanicznych w zastosowaniu do analizy kratownic metodą propagacji obciążeń [Qualitative analysis of mechanical systems applied to truss analysis with the method of load propagation, in Polish]. In: A. Grzech, ed.: *Problemy Informatyki i Automatyki* [*Problems of Computer Science and Automatic Control*]. Ossolineum, Wrocław, 177–190.

[83] Z. Kulpa (1987) Impossible figures—figury niemożliwe [bilingual]. In: D. Folga-Januszewska, ed.: *Artists on Space—Artyści o przestrzeni* [bilingual], National Museum in Warsaw, Warsaw, 41–63.

[84] Z. Kulpa (1984) Ikonika: obshchuvane chrez obrazi s pomoshchta na komputar [Iconics: computer-aided pictorial communication, in Bulgarian]. *Fiziko-matematichesko spisanie*, (Sofia, Bulgaria), 1: 59–66.

[85] Z. Kulpa (1983) Ikonika: komunikacja wizualna wspomagana komputerowo [Iconics: Computer-aided visual communication, in Polish]. In: M. Nałęcz, ed.: *Wybrane problemy inżynierii biomedycznej*, IBIB PAN, Warsaw, 544–553.

[86] Z. Kulpa (1982) Some remarks on "A note on the computation of the enclosed area for contour-coded binary objects" by P. Zamperoni. *Signal Processing*, 4(1): 85–86.

[87] Z. Kulpa (1981) Universal digital image processing systems in Europe—a comparative survey. In: [79], 1–20.

[88] Z. Kulpa, J. Dernałowicz, H.T. Nowicki, A. Bielik (1981) CPO-2/K-202: a universal digital image analysis system. In: [79], 169–199.

[89] Z. Kulpa, B. Kruse (1979) Methods of effective implementation of circular propagation in discrete images. *INTERNSKRIFT LiTH-ISY-I-0274*, Institute of Electrical Engineering of Linköping University, Linköping, 1–43. (An extended version published as [17].)

[90] Z. Kulpa (1979) A note on the paper by B.K.P. Horn: "Circle generators for display devices." *Computer Graphics and Image Processing*, 9: 102-103.

[91] Z. Kulpa (1978) Errors in objects positioning with 'centre of gravity' method. *The Industrial Robot*, 5(2): 94–99.

[92] Z. Kulpa (1977) Systemy i zasady automatycznej analizy obrazów graficznych [Systems and principles of automatic analysis of graphical images, in Polish]. In: [80], 5–32.

[93] Z. Kulpa (1977) Struktura zapisu obrazów w systemie CPO-2/K-202 [Storage structure of pictures in the CPO-2/K-202 system, in Polish]. In: [80], 57–67.

[94] Z. Kulpa, H.T. Nowicki, M. Raczkowska, M. Piotrowicz, B. Wołosewicz, M. Doros, D. Gajkowicz-Dędys (1977) System podprogramów przetwarzania obrazów graficznych PICASSO [A system of PICASSO subroutines for processing of graphical pictures, in Polish]. In: [80], 69–111.

[95] H.T. Nowicki, Z. Kulpa (1977) Konwersacyjny system programowania zadań analizy obrazów PICASSO-SHOW 1 [An interactive programming system PICASSO-SHOW 1 for image analysis, in Polish]. In: [80], 113–148.

[96] Z. Kulpa (1975) Planar grammars, parallel picture processing algorithms and their equivalence. In: *Planar Grammars (Gramatyki planarne)*. Reports of the Institute of Organization and Management of the Polish Academy of Sciences and Ministry of Higher Education and Technology, series B, vol. 25, Warsaw, 25–38.

[97] Z. Kulpa (1974) Yazyki pererabotki graficheskoy informatsyi [Languages for processing of graphical information, in Russian]. In: *TANULMÁNYOK*, MTA-STAKI, Budapest, 21: 41–51.

[98] Z. Kulpa, H. Szydło (1972) PICTURE ALGOL 1204 Ab: język przetwarzania informacji graficznej [PICTURE ALGOL 1204 Ab: a language for graphical information processing, in Polish]. In: [ICS Report 1972], 55–92.

[99] Z. Kulpa (1972) Algorytmy pocieniania linii [Thinning algorithms, in Polish]. In: [ICS Report 1972], 93–147.

Popular articles

[100] Z. Kulpa (1985) Komputila bildomanipulado [Computer image processing, in Esperanto]. *Internacia Komputado*, 3(7): 8–12.

[101] Z. Kulpa (1980) Komputerowa analiza obrazów wizualnych [Computer analysis of visual images, in Polish]. *Delta*, 1980(8).

[102] Z. Kulpa (1978) Automatyczne rozpoznawanie obrazów [Automatic pattern recognition, in Polish]. *Horyzonty Techniki*, 1978(6).

[103] Z. Kulpa (1976) Obiekty nieistniejące [Nonexisting objects, in Polish]. *Problemy*, 1976(9): 20–22.

[104] Z. Kulpa (1974) Era robotów [The era of robots, in Polish]. *Problemy*, 1974(12): 5–10.

More important unpublished works

[105] Z. Kulpa (1999) Diagrammatic representation of interval space; Part I: Basics; Part II: Arithmetic. *Internal Report*, Institute of Fundamental Technological Research of the Polish Academy of Sciences, Warsaw.

[106] Z. Kulpa (1998) Qualitative model of load propagation in truss structures. *Internal Report No. B-1/1998*, Institute of Fundamental Technological Research of the Polish Academy of Sciences, Warsaw.

[107] Z. Kulpa (1995) Two-dimensional representation of interval relations: Preliminaries. *Internal Report*, Institute of Fundamental Technological Research of the Polish Academy of Sciences, Warsaw.

[108] Z. Kulpa (1992) Visual knowledge representation. *Internal Report* (unfinished), George Mason University, Fairfax, VA.

[109] Z. Kulpa (1980) *Konstrukcja języka programowania algorytmów cyfrowego przetwarzania złożonych obrazów wizualnych* [*The Design of a Programming Language for Digital Processing of Complex Visual Images*, in Polish]. Ph.D. Thesis, Institute of Computer Science of Polish Academy of Sciences, Warsaw.

[110] Z. Kulpa (1974) Oprogramowanie systemów analizy obrazów graficznych CPO-1 i CPO-2 [Software of the CPO-1 and CPO-2 systems for analysis of graphical images, in Polish]. Presented at: *Conference on "Computer Systems for Processing of Experimental Data,"* Kazimierz Dolny, Poland.

[111] Z. Kulpa (1973) Język analizy obrazów graficznych PAL [The language PAL for analysis of graphical images, in Polish]. Presented at: *Conference on "Methods of Direct Input and Output of Textual and Pictorial Information in Computer Systems,"* Jabłonna, Poland.

Bibliography:
Other publications

Proceedings, article collections

[AI Handbook 1981] A. Barr, E.A. Feigenbaum, eds.: *The Handbook of Artificial Intelligence*. HeurisTek Press, Stanford, CA, and W. Kaufmann, Los Altos, CA.

[DIAGRAMS 1992] *Reasoning with Diagrammatic Representations (1992 AAAI Spring Symposium)*. AAAI Press, Menlo Park, CA.

[DIAGRAMS 1995] J. Glasgow, N.H. Narayanan, B. Chandrasekaran, eds.: *Diagrammatic Reasoning: Computational and Cognitive Perspectives*. AAAI Press, Menlo Park, CA, and The MIT Press, Cambridge, MA.

[DIAGRAMS 1996] G. Allwein, J. Barwise, eds.: *Logical Reasoning with Diagrams*, Oxford University Press, Oxford.

[DIAGRAMS 1997] M. Anderson, ed.: *Reasoning with Diagrammatic Representations II (1997 AAAI Fall Symposium Working Notes)*. AAAI Press, Menlo Park, CA.

[DIAGRAMS 2000a] M. Anderson, P. Cheng, V. Haarslev, eds.: *Theory and Applications of Diagrams* (Proc. *First International Conference Diagrams 2000*, Edinburgh, Scotland, UK, September 1–3, 2000). Lecture Notes in Artificial Intelligence, vol. 1889, Springer-Verlag, Berlin.

[DIAGRAMS 2000b] P. Olivier, M. Anderson, B. Meyer, eds.: *Diagrammatic Representation and Reasoning*. Springer-Verlag, Berlin.

[DIAGRAMS 2001] A. Blackwell, ed.: *Thinking with Diagrams*. Kluwer Academic Publ., Dordrecht. Also as: Special Issue of *Artificial Intelligence Review*, 5(1/2).

[DIAGRAMS 2002] M. Hegarty, B. Meyer, N. Hari Narayanan, eds.: *Diagrammatic Representation and Inference* (Proc. *Second International Conference Diagrams 2002*, Callaway Gardens, GA, USA, April 18–22, 2002). Lecture Notes in Artificial Intelligence, vol. 2317, Springer-Verlag, Berlin.

[EXPERTSYS 2001] C.T. Leondes, ed.: *Expert Systems*. Academic Press, New York (6 volumes).

[GREC 1999] A.K. Chhabra, D. Dori, eds.: *Graphics Recognition. Recent Advances* (Proc. *Third International Workshop GREC'99*, Jaipur, India, September 26–27, 1999). Lecture Notes in Computer Science, vol. 1941, Springer-Verlag, Berlin.

[HYPERGRAPHICS 1978] D.W. Brisson, ed.: *Hypergraphics: Visualizing Complex Relationships in Art, Science, and Technology.* Westview Press, Boulder, CO.

[ICS Report 1972] J.L. Kulikowski et al., eds.: *Metody automatycznego przetwarzania informacji o złożonej strukturze ze szczególnym uwzględnieniem informacji obrazowej [Methods of Automatic Processing of Complex Information, in Particular Pictorial Information,* in Polish]. Reports of the Institute of Applied Cybernetics of the Polish Academy of Sciences, vol. 6, Warsaw.

[INTERVALS 1975] K.L.E. Nickel, ed.: *Interval Mathematics 1975.* Lecture Notes in Computer Science, vol. 29, Springer Verlag, Berlin.

[INTERVALS 1980] K.L.E. Nickel, ed.: *Interval Mathematics 1980.* Academic Press, New York.

[INTERVALS 1997] J. Wolff von Gudenberg, ed.: *Proc. International Conference on Interval Methods and Computer Aided Proofs in Science and Engineering INTERVAL'96,* Würzburg 1996. Special Issue of *Reliable Computing,* 3(3).

[INTERVALS 2001] W. Krämer, J. Wolff von Gudenberg, eds.: *Scientific Computing, Validated Numerics, Interval Methods* (Proc. *SCAN/INTERVAL 2000 International Conference,* Karlsruhe, Germany). Kluwer Academic/Plenum Publishers.

[KNOWLREPR 1985] R.J. Brachman, H.J. Levesque, eds.: *Readings in Knowledge Representation.* Morgan Kaufmann, San Mateo, CA.

[STEREO 2001] L. Chmielewski, ed.: Stereogrammetry and Related Topics. Special Issue of *Machine GRAPHICS & VISION,* 10(3).

[VISLANG 1990a] S.-K. Chang, ed.: *Principles of Visual Programming Systems.* Prentice Hall, Englewood Cliffs, NJ.

[VISLANG 1990b] S.-K. Chang, ed.: *Visual Languages and Visual Programming.* Plenum Press, New York.

[VISLANG 1998] K. Marriott, B. Meyer, eds.: *Visual Language Theory.* Springer-Verlag, Berlin.

[VISMATH 1997] H-C. Hege, K. Polthier, eds.: *Visualization and mathematics: Experiments, Simulations and Environments.* Springer-Verlag, Berlin.

[VISPROG 1990] E.P. Glinert, ed.: *Visual Programming Environments; Part I: Paradigms and Systems, Part II: Applications and Issues.* IEEE Computer Society Press.

Individual papers and books

[Alefeld & Herzberger 1983] G. Alefeld, J. Herzberger: *Introduction to Interval Computations.* Academic Press, New York.

[Allen 1983] J.F. Allen: Maintaining knowledge about temporal relations. *Communications of the ACM*, 26(11): 832–843.

[Amarel 1968] S. Amarel: On representations of problems of reasoning about actions. In: D. Michie, ed.: *Machine Intelligence 3.* Edinburgh University Press, Edinburgh, 131–171.

[Anderson & McCartney 1997] M. Anderson, R. McCartney: Learning from diagrams. In: [78], 57–76.

[Arnheim 1969] R. Arnheim: *Visual Thinking.* University of California Press, Berkeley, CA.

[Banchoff & Strauss 1978] T.F. Banchoff, C.M. Strauss: Real-time computer graphics analysis of figures in four-space. In: [HYPERGRAPHICS 1978], 159-168.

[Barker-Plummer & Bailin 1997] D. Barker-Plummer, S.C. Bailin: The role of diagrams in mathematical proofs. In: [78], 25–56.

[Barwise & Etchemendy 1996a] J. Barwise, J. Etchemendy: Visual information and valid reasoning. In: [DIAGRAMS 1996], 3–25. (Older, but essentially identical version appeared in: W. Zimmerman, S. Cunningham, eds.: *Visualization in Teaching and Learning Mathematics.* Mathematical Association of America, Washington, D.C. 1991, 9–24.)

[Barwise & Etchemendy 1996b] J. Barwise, J. Etchemendy: Heterogeneous logic. In: [DIAGRAMS 1996], 179–200. (Appeared also in: [DIAGRAMS 1995], 209–232.)

[Barwise & Hammer 1996] J. Barwise, E. Hammer: Diagrams and the concept of logical system. In: [DIAGRAMS 1996], 49–78.

[Berge 1973] C. Berge: *Graphs and Hypergraphs*, North Holland, Amsterdam.

[Bertin 1967/83] J. Bertin: Semiologie graphique: les diagrammes, les reseaux, les cartes. Mouton/Gauthiers-Villars, The Hague/Paris 1967. [English translation: J. Bertin: *Semiology of Graphics: Diagrams, Networks, Maps.* University of Wisconsin Press, 1983.]

[Bertin 1981] J. Bertin: *Graphics and Graphic Information Processing.* Walter de Gruyter. Berlin.

[Bettini 1994] C. Bettini: A formalization of interval-based temporal subsumption in first order logic. In: *Foundations of Knowledge Representation and Reasoning*, Lecture Notes in Artificial Intelligence, vol. 810, Springer-Verlag, Berlin, 53–73.

[Borkowski et al. 1999] A. Borkowski, E. Grabska, G. Hliniak: Function-structure computer-aided design model. *Machine GRAPHICS & VISION*, 8(3): 367–381.

[Borning 1981] A. Borning: The programming language aspects of *ThingLab*, a constraint-oriented simulation laboratory. *ACM Trans. on Programming Languages and Systems*, 3: 353–387.

[Bowman 1968] W.J. Bowman: *Graphic Communication*. J. Wiley, New York.

[Brachman 1979] R.J. Brachman: On the epistemological status of semantic networks. In: N.V. Finder, ed.: *Associative Networks: Representation and Use of Knowledge by Computer*. Academic Press, New York, 3–60.

[Brachman 1990] R.J. Brachman: The future of knowledge representation. Extended abstract. In: *Proc. 8th National Conference on Artificial Intelligence (AAAI-90)*. AAAI Press/The MIT Press, Menlo Park, CA/Cambridge, MA, 1082–1092.

[Bresenham 1965] J.E. Bresenham: Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1): 25–30.

[Brice & Fennema 1970] C.R. Brice, C.L. Fennema: Scene analysis using regions. *Artificial Intelligence*, 1: 205–226.

[Buckley & Qu 1990] J.J. Buckley, Y. Qu: On using α-cuts to evaluate fuzzy equations. *Fuzzy Sets and Systems*, 38: 309–312.

[Byrne 1847] O. Byrne: *The First Six Books of the Elements of Euclid in Which Coloured Diagrams and Symbols Are Used Instead of Letters for the Greater Ease of Learners*. William Pickering, London.

[Chen 1976] P. P.-S. Chen: The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems*, 1(1): 9–36.

[Clowes 1971] M.B. Clowes: On seeing things. *Artificial Intelligence*, 2: 79–116.

[Cowan 1977] T.M. Cowan: Organizing the properties of impossible figures. *Perception*, 6(1): 41-56.

[Coxeter & Moser 1957/80] H.S.M. Coxeter, W.O.J. Moser: *Generators and Relations for Discrete Groups*. Springer-Verlag, Berlin [4th edition: 1980].

[Cyganek 2001] B. Cyganek: Novel feature-based stereo matching method that employs tensor representation of local pixels neighbourhoods in images. In: [STEREO 2001], 289–316.

[da Fontoura Costa & Cesar 2001] L. da Fontoura Costa, R.M. Cesar, Jr.: *Shape Analysis and Classification: Theory and Practice*. CRC Press, Boca Raton.

[Danielsson 1980] P.-E. Danielsson: Euclidean distance mapping. *Computer Graphics and Image Processing*, 14: 227–248.

[Davis 1990] E. Davis: *Representations of Commonsense Knowledge*. Morgan Kaufmann, San Mateo, CA.

[Dąbkowska & Mokrzycki 1998] M. Dąbkowska, W.S. Mokrzycki: A face-dependent view model of convex polyhedra. *Machine GRAPHICS & VISION*, 7(1/2): 325–334.

[Dernałowicz 1972] J. Dernałowicz: Cyfrowy przetwornik obrazu do wprowadzania danych do maszyny cyfrowej [Digital image converter for supplying graphical data to a computer, in Polish]. In: [ICS Report 1972], 181–189.

[Dernałowicz et al. 1977] J. Dernałowicz, M. Chmielewski, W. Jarosiński, A. Dernałowicz: System cyfrowego przetwarzania obrazów CPO-2/K-202 [A system for digital image processing CPO-2/K-202, in Polish]. In: [80], 33–55.

[d'Ocagne 1899] M. d'Ocagne: Traité de nomographie: Théorie des abaques, applications pratiques. Gauthier-Villars, Paris.

[Dondis 1975] D.A. Dondis: *A Primer of Visual Literacy.* The Mit Press, Cambridge, MA.

[Doros 1979] M. Doros: Algorithms for generation of discrete circles, rings, and disks. *Computer Graphics and Image Processing*, 10: 366–371.

[Drakengren & Jonsson 1998] T. Drakengren, P. Jonsson: A complete classification of tractability in Allen's algebra relative to subsets of basic relations. *Artificial Intelligence*, 106: 205–219.

[Ellis et al. 1979] T.J. Ellis, D. Proffitt, D. Rosen, W. Rutkowski: Measurement of the lengths of digitized curved lines. *Computer Graphics and Image Processing*, 10: 333–347.

[Engelhardt 2002] Y. Engelhardt: *The Language of Graphics: A Framework for the Analysis of Syntax and Meaning in Maps, Charts and Diagrams.* Ph.D. Thesis, University of Amsterdam, Amsterdam.

[Ernst 1986] B. Ernst: *Het Begoochelde Oog: Onmogelijke en Meerzinnige Figuren* [*The Confounded Eye: Impossible and Ambiguous Figures*, in Dutch]. Meulenhof/Landshoff, Amsterdam. (A shortened English edition: B. Ernst: *Optical Illusions*, Benedikt Tashen Verlag Gmbh, Koln 1992.)

[Essex et al. 2000] C. Essex, M. Davison, C. Schulzky: Numerical monsters. *SIGSAM Bulletin*, 134: 16–32.

[Feder 1971] J. Feder: Plex languages. *Information Science*, 3: 225–241.

[Forbus et al. 1991] K.D. Forbus, P. Nielsen, B. Faltings: Qualitative spatial reasoning: the CLOCK project. *Artificial Intelligence*, 51(1-3): 417–471.

[Freeman 1970] H. Freeman: Boundary encoding and processing. In: B.S. Lipkin, A. Rosenfeld, eds.: *Picture Processing and Psychopictorics.* Academic Press, New York, 241–263.

[Freeman & Glass 1969] H. Freeman, J.M. Glass: On the quantization of line-drawing data. *IEEE Transactions on System Sciences and Cybernetics*, SSC-5: 70-79.

[Freksa 1992] C. Freksa: Temporal reasoning based on semi-intervals. *Artificial Intelligence*, 54(1-2): 199–227.

[Frommer 2001] A. Frommer: Proving conjectures by use of interval arithmetic. In: U. Kulisch, R. Lohner, A. Facius, eds.: *Perspectives on Enclosure Methods*, Springer-Verlag, Vienna, 1–13.

[Fu 1982] K.S. Fu: *Syntactic Pattern Recognition and Applications*. Prentice Hall, Englewood Cliffs.

[Funt 1980] B.V. Funt: Problem-solving with diagrammatic representations. *Artificial Intelligence*, 13(3): 201–230.

[Furnas 1990] G.W. Furnas: Formal models for imaginal deduction. In: *Proc. Twelfth Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum, Hillsdale, NJ, 662–669.

[Furnas et al. 2000] G. Furnas, Y. Qu, S. Shrivastava, G. Peters: The use of intermediate graphical constructions in problem solving with dynamic, pixel-level diagrams. In: [DIAGRAMS 2000a], 314–329.

[Gardeñes et al. 1980] E. Gardeñes, A. Trepat, J.M. Janer: SIGLA-PL/1: Development and applications. In: [INTERVALS 1980], 301–315.

[Gardeñes et al. 1981] E. Gardeñes, A. Trepat, J.M. Janer: Approaches to simulation and to the linear problem in the SIGLA system. *Freiburger Intervall-Berichte*, 81/8: 1–28.

[Gardeñes et al. 2001] E. Gardeñes, M.Á. Sainz, L. Jorba, R. Calm, R. Estell, H. Mielgo, A. Trepat: Modal intervals. *Reliable Computing*, 7: 77–111.

[Gardeñes & Trepat 1979] E. Gardeñes, A. Trepat: The interval computing system SIGLA-PL/1(0). *Freiburger Intervall-Berichte*, 79/8.

[Gardeñes & Trepat 1980] E. Gardeñes, A. Trepat: Fundamentals of SIGLA, an interval computing system over the completed set of intervals. *Computing*, 24: 161–179.

[Gardin & Meltzer 1989] F. Gardin, B. Meltzer: Analogical representations of naive physics. *Artificial Intelligence*, 38: 139–159. (Reprinted in: [DIAGRAMS 1995], 670–689.)

[Gardner 1958] M. Gardner: *Logic Machines and Diagrams*. University of Chicago Press, Chicago. [2nd edition 1982.]

[Gelernter 1959] H. Gelernter: Realization of a geometry-theorem proving machine. In: *Proc. International Conf. on Information Processing (ICIP)*. UNESCO House, Paris, 273–282. (Reprinted in: E.A. Feigenbaum, J. Feldman, eds. (1963) *Computers and Thought*. McGraw-Hill, New York, 134–152.)

[Gelernter et al. 1960] H. Gelernter, J.R. Hansen, D.W. Loveland: Empirical explorations of the geometry-theorem proving machine. In: *Proc. of the Western Joint Computer Conf. (WJCC'60)*, Vol. 17: 143–147. (Reprinted in: E.A. Feigenbaum, J. Feldman, eds.: *Computers and Thought*. McGraw-Hill, New York 1963, 153–163).

[Genesereth & Nilsson 1987] M. Genesereth, N. Nilsson: *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann, San Mateo, CA.

[Giere 1999] R.N. Giere: *Science without Laws*. The University of Chicago Press, Chicago, IL.

[Gleicher & Witkin 1994] M. Gleicher, A. Witkin: Drawing with constraints. *The Visual Computer*, 11: 39–51.

[Goodstein & Goodstein 1996] D.L. Goodstein, J.R. Goodstein: *Feynman's Lost Lecture: The Motion of Planets Around the Sun*. W.W. Norton & Co. (Polish edition: *Zaginiony wykład Feynmana: Ruch planet wokół Słońca*. Prószyński i S-ka, Warszawa 1997.)

[Grabska 1993a] E. Grabska: Theoretical concepts of graphical modeling. Part one: realization of CP-graphs. *Machine GRAPHICS & VISION*, 2(1): 3–38.

[Grabska 1993b] E. Grabska: Theoretical concepts of graphical modeling. Part two: CP-graphs grammars and languages. *Machine GRAPHICS & VISION*, 2(2): 149–178.

[Grabska 2001] E. Grabska: Emergent shapes in graphical design. In: D.M. Dubois, ed.: *Computing Anticipatory Systems* (CASYS 2000: Fourth International Conference). American Institute of Physics, 621–627.

[Granlund & Knutsson 1996] G.H. Granlund, H. Knutsson: *Signal Processing for Computer Vision*. Kluwer Academic Publ., Dordrecht.

[Gregory 1970] R.L. Gregory: *The Intelligent Eye*. Weidenfeld and Nicolson, London.

[Gurr 1999] C.A. Gurr: Effective diagrammatic communication: Syntactic, semantic and pragmatic issues. *Journal of Visual Languages and Computing*, 10: 317–342.

[Hadamard 1945] J. Hadamard: *The Psychology of Invention in the Mathematical Field*. Princeton University Press, Princeton, NJ. (Modern reprint as *The Mathematician's Mind*, Princeton University Press, Princeton, NJ 1996.)

[Hammer 1996] E. Hammer: *Logic and Visual Information*. Cambridge University Press, Cambridge.

[Hankins 1999] T.L. Hankins: Blood, dirt, and nomograms: A particular history of graphs. *Isis*, 90: 50–80.

[Hansen 1992] E. Hansen: *Global Optimization Using Interval Analysis*. Marcel Dekker, New York.

[Harel 1988] D. Harel: On visual formalisms. *Communications of the ACM*, 31(5): 514–530. (Reprinted in [VISPROG 1990], vol. 1: 171-187.)

[Hernández 1994] D. Hernández: *Qualitative representation of Spatial Knowledge*. Lecture Notes in Artificial Intelligence, vol. 804. Springer-Verlag, Berlin.

[Hickey et al. 2001] T.J. Hickey, Q. Ju, M.H. van Emden: Interval arithmetic: from principles to implementation. *Journal of the ACM*, 48: 1038–1068.

[Hoelscher et al. 1952] R.P. Hoelscher, J.N. Arnold, S.H. Pierce: *Graphic Aids in Engineering Computation*. New York.

[Huffman 1971] D.A. Huffman: Impossible objects as nonsense sentences. In: B. Meltzer, D. Michie, eds.: *Machine Intelligence 6*, Edinburgh University Press, Edinburgh, 295–323.

[Ioerger 1992] T.R. Ioerger: Diagrammatic semantics for spatial prepositions. In: [DIAGRAMS 1992], 191–194.

[Iwasaki et al. 1995] Y. Iwasaki, S. Tessler, K.H. Law: Qualitative structural analysis through mixed diagrammatic and symbolic reasoning. In: [DIAGRAMS 1995], 712–729.

[Jamnik et al. 1999] M. Jamnik, A. Bundy, I. Green: On automating diagrammatic proofs of arithmetic arguments. *Journal of Logic, Language and Information*, 8: 297–321.

[Jaulin et al. 2001] L. Jaulin, M. Kieffer, O. Didrit, É. Walter: *Applied Interval Analysis*. Springer Verlag, London.

[Jeleński 1968] S. Jeleński: *Lilavati* [In Polish]. PZWS, Warsaw.

[Kahan 1968] W.M. Kahan: *A more complete interval arithmetic*. Lecture notes, University of Toronto, Toronto.

[Kanizsa 1974] G. Kanizsa: Contours without gradients or cognitive contours. *Italian Journal of Psychology*, 1: 93–112.

[Kaucher 1973] E. Kaucher: *Über metrische und algebraische Eigenschaften einiger beim numerischen Rechnen auftretender Räume*. Ph.D. Thesis, Universität Karlsruhe, Karlsruhe.

[Kaucher 1977] E. Kaucher: Über eine Überlaufarithmetik auf Rechenanlagen und deren Anwendungsmöglichkeiten. *ZAMM*, 57: T286–T287.

[Kaucher 1980] E. Kaucher: Interval analysis in the extended interval space IR. *Computing, Suppl.* 2: 33–49.

[Kaucher 1999] E. Kaucher: *Personal communication*.

[Koedinger 1992] K.R. Koedinger: Emergent properties and structural constraints: Advantages of diagrammatic representations in reasoning and learning. In: [DIAGRAMS 1992], 151-156.

[Kordek et al. 1980] J. Kordek, R. Nipl, K. Sztaba, R. Tadeusiewicz: CESARO—the digital experimental system of analysis and recognition of images. In: *Proc. 17th International Symposium on the Application of Computers and Mathematics in the Mineral Industries*, Moscow: 393–398.

[Kosslyn 1980] S.M. Kosslyn: *Image and Mind*. Harvard University Press, Cambridge, MA.

[Kosslyn 1994] S.M. Kosslyn: *Image and Brain*. The MIT Press, Cambridge, MA.

[Kreinovich et al. 1997] V. Kreinovich, A. Lakeyev, J. Rohn, P. Kahl: *Computational Complexity and Feasibility of Data Processing and Interval Computations*. Kluwer Academic Publ., Dordrecht.

[Kruse 1973] B. Kruse: A parallel picture processing machine. *IEEE Transactions on Computers*, C-22(12).

[Larkin & Simon 1987] J.H. Larkin, H.A. Simon: Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11: 65–99.

[Laveuve 1975] S.E. Laveuve: Definition einer Kahan-Arithmetik und ihre Implementierung. In: [INTERVALS 1975], 236–245.

[Ledley et al. 1966] R.S. Ledley, J.D. Jacobsen, M. Belson: BUGSYS: a programming system for picture processing—not for debugging. *Communications of the ACM*, 9(2).

[Lemon & Pratt 1997] O. Lemon, I. Pratt: Spatial logic and the complexity of diagrammatic reasoning. In: [78], 77–88.

[Levesque 1986] H.J. Levesque: Making believers out of computers. *Artificial Intelligence*, 30: 81–108.

[Levialdi 1981] S. Levialdi: Finding the edge. In: J.C. Simon, R.M. Haralick, eds.: *Digital Image Processing*. Reidel, Dordrecht, 105-148.

[Leyton 2001] M. Leyton: *A Generative Theory of Shape*. Lecture Notes in Computer Science, vol. 2145, Springer-Verlag, Berlin.

[Luengo 1995] I. Luengo: *Diagrams in Geometry*. Ph.D. Thesis, Indiana University, Bloomington, IN.

[Luengo 1996] I. Luengo: A diagrammatic subsystem of Hilbert's geometry. In: [DIAGRAMS 1996], 149–176.

[Mackinlay & Genesereth 1985] J. Mackinlay, M.R. Genesereth: Expressiveness and language choice. *Data & Knowledge Engineering*, 1: 17–29.

[Mackworth 1976] A.K. Mackworth: Model-driven interpretation in intelligent vision systems. *Perception*, 5: 349–370.

[Markov 1995] S. Markov: On directed interval arithmetic and its applications. *Journal of Universal Computer Science*, 1: 510–522.

[Markov 2001a] S. Markov: On the algebraic properties of intervals and some applications. *Reliable Computing*, 7(2): 113–127.

[Markov 2001b] S. Markov: Computation of algebraic solutions of interval systems via systems of coordinates. In: [INTERVALS 2001], 103–114.

[Markov & Okumura 1999] S. Markov, K. Okumura: The contribution of T. Sunaga to interval analysis and reliable computing. In: T. Csendes, ed.: *Developments in Reliable Computing*, Kluwer Academic Publ., Dordrecht, 167-188.

[Marks & Reiter 1990] J. Marks, E. Reiter: Avoiding unwanted conversational implicatures in text and graphics. In: *Proc. 8th National Conference on Artificial Intelligence (AAAI'90)*. AAAI Press / The MIT Press, Menlo Park, CA, 450-456.

[McCormick 1963] B.H. McCormick: The Illinois pattern recognition computer—ILLIAC III. *IEEE Transactions on Electronic Computers*, EC-12: 791–813.

[Milgram & Rosenfeld 1972] D.L. Milgram, A. Rosenfeld: Array automata and array grammars. In: C.V. Freiman, ed.: *Information Processing 71* (Proc. IFIP Congress 1971). North Holland, Amsterdam.

[Miller 2000] N. Miller: Case analysis in Euclidean geometry: An overview. In: [DIAGRAMS 2000a], 490–493.

[Miller 2001] N. Miller: *A Diagrammatic Formal System for Euclidean Geometry*. Ph.D. Thesis, Cornell University, Ithaca, NY.

[Minsky & Papert 1969] M. Minsky, S. Papert: *Perceptrons—An Introduction to Computational Geometry*. The MIT Press, Cambridge, MA.

[Mokrzycki 1992a] W. Mokrzycki: Stereoskopowe systemy postrzegania głębi sceny: przegląd zagadnień [Stereoscopic systems of scene depth perception: A survey, in Polish]. *Machine GRAPHICS & VISION*, 2(1/2): 342–392.

[Mokrzycki 1992b] W. Mokrzycki: *Encyklopedia przetwarzania obrazów* [*Encyclopedia of Picture Processing*, in Polish]. Akademicka Oficyna Wydawnicza RM, Warsaw.

[Montanari 1970] U. Montanari: A note on minimal length polygonal approximation to a digitized contour. *Communications of the ACM*, 13: 41-47.

[Moore, R.C. 1982] R.C. Moore: The role of logic in knowledge representation and commonsense reasoning. In: *Proc. 2nd National Conference on Artificial Intelligence (AAAI-82)*. William Kaufmann, Los Altos, CA, 428–433. (Reprinted in: [KNOWLREPR 1985], 336–341.)

[Moore, R.E. 1966] R.E. Moore: *Interval Analysis*. Prentice Hall, Englewood Cliffs, NJ.

[Mukerjee & Joe 1990] A. Mukerjee, G. Joe: A qualitative model for space. In: *Proc. 8th National Conference on Artificial Intelligence (AAAI'90)*. AAAI Press/The MIT Press, Menlo Park, CA, 721–727.

[Myers 1990] B.A. Myers: Taxonomies of visual programming and program visualization. *Journal of Visual Languages and Computing*, 1(1): 97–123.

[Narasimhan 1964] R. Narasimhan: Labeling schemata and syntactic description of pictures. *Information and Control*, 7(2): 151–179.

[Needham 1997] T. Needham: *Visual Complex Analysis*. Clarendon Press, Oxford.

[Nelsen 1993] R.B. Nelsen: *Proofs Without Words: Exercises in Visual Thinking*. The Mathematical Association of America, Washington, DC.

[Nelson 1985] G. Nelson: Juno, a constraint based graphics system. *Computer Graphics*, 19: 235–243.

[Neumaier 1990] A. Neumaier: *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge.

[Neumaier 2003] A. Neumaier: Taylor forms—use and limits. *Reliable Computing*, 9(1): 43–79.

[Nieniewski 1998] M. Nieniewski: *Morfologia matematyczna w przetwarzaniu obrazów* [*Mathematical Morphology in Picture Processing*, in Polish]. Akademicka Oficyna Wydawnicza PLJ, Warsaw.

[Nökel 1991] K. Nökel: *Temporally Distributed Symptoms in Technical Diagnosis*. Lecture Notes in Artificial Intelligence, vol. 517. Springer-Verlag, Berlin.

[Olivier 1997] P. Olivier: Hierarchy and attention in computational imagery. In: [78], 77–88.

[Olivier et al. 1996] P. Olivier, K. Nakata, A.R.T. Ormsby: Occupancy array-based kinematic reasoning. *Engineering Applications of Artificial Intelligence*, 9(5): 541–549.

[Pavlidis 1982] T. Pavlidis: *Algorithms for Graphics and Image Processing*. Computer Science Press, Rockville, MD. (Polish edition: T. Pavlidis: *Grafika i przetwarzanie obrazów*, WNT, Warszawa 1987.)

[Penrose & Penrose 1958] L.S. Penrose, R. Penrose: Impossible objects: A special type of visual illusion. *British Journal of Psychology*, 49(1): 31-33.

[Piaget 1951] J. Piaget: *The Origin of Intelligence in Children*. International Universities Press, New York.

[Pineda & Garza 1998] L. Pineda, G. Garza: A model for multimodal representation and inference. In: L. Pineda, T. Rist, J. Lee, eds.: *Interpretation and Generation in Intelligent Multimodal Systems and Graphical Reasoning in Expert Systems* (Proc. of the Workshop at the 4th World Congress on Expert Systems), ITESM, Mexico City, Mexico, 6-21.

[Pratt 1978] W.K. Pratt: *Digital Image Processing*. J. Wiley & Sons, New York. [3rd ed.: 2001].

[Proffitt & Rosen 1979] D. Proffitt, D. Rosen: Metrication errors and coding efficiency of chain-encoding schemes for the representation of lines and edges. *Computer Graphics and Image Processing*, 10: 318–332.

[Pylyshyn 1981] Z.W. Pylyshyn: The imagery debate: Analogue media versus tacit knowledge. *Psychological Review*, 88: 16–45. (Reprinted in: A. Collins, E.E. Smith, eds.: *Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence*. Morgan Kaufmann, San Mateo, CA 1988, 600–614.)

[Ratschek 1972] H. Ratschek: Teilbarkeitskriterien der Intervallarithmetik. *J. Reine Angewandte Mathematik*, 252: 128–138.

[Ratschek 1973] H. Ratschek: Intervalarithmetik—mit Zirkel und Lineal. *Elemente der Mathematik*, 28(4): 93–96.

[Ratschek 1980] H. Ratschek : Representation of interval operations by coordinates. *Computing*, 24: 93–96.

[Ratschek & Rokne 1984] H. Ratschek, J. Rokne: *Computer Methods for the Range of Functions*. J. Wiley & Sons, New York.

[Rit 1986] J.-F. Rit: Propagating temporal constraints for scheduling. In: *Proc. Fifth National Conference on Artificial Intelligence (AAAI-86)*. Morgan Kaufmann, Los Altos, CA, 383–388.

[Rohn 1989] J. Rohn: Systems of linear interval equations. *Linear Algebra and Its Applications*, 126: 39–78.

[Rohn 2000] J. Rohn: Finite characterization of some linear problems with inexact data. Invited Lecture at the *SCAN/INTERVAL 2000 Conference*, Karlsruhe, Sept. 18-23, 2000. [Unpublished; for conference slides of the lecture and later additions, see `http://www.ippt.gov.pl/~zkulpa/quaphys/interval.html`]

[Rosen 1980] D. Rosen: On the areas and boundaries of quantized objects. *Computer Graphics and Image Processing*, 13: 94-98.

[Rosenfeld 1969] A. Rosenfeld: *Picture Processing by Computer*. Academic Press, New York.

[Rosenfeld & Kak 1976] A. Rosenfeld, A.C. Kak: *Digital Picture Processing*. Academic Press, New York.

[Rosenfeld & Pfaltz 1966] A. Rosenfeld, J.L. Pfaltz: Sequential operations in digital picture processing. *Journal of the ACM*, 13(4).

[Roth & Mattis 1990] S.F. Roth, J. Mattis: Data characterization for intelligent graphics presentation. In: *Human Factors in Computing Systems VII* (Proc. of the Conf. on Computer-Human Interaction (CHI'90)). ACM Press, 193-200.

[Roth & Mattis 1991] S.F. Roth, J. Mattis: Automating the presentation of information. In: *Proc. IEEE Conf. on Artificial Intelligence Applications*. IEEE Press.

[Rucker 1982] R. Rucker: *Infinity and the Mind: The Science and Philosophy of the Infinite*. Bantam Books, New York.

[Schlieder 1996] C. Schlieder: Diagrammatic reasoning about Allen's interval relations. In : *AAAI Spring Symposium on Cognitive and Computational Models of Spatial Representations* (Stanford, CA, March 25–27, 1996), Stanford University, Stanford, CA, 9 pp.

[Serra 1989] J. Serra: *Image Analysis and Mathematical Morphology*. Academic Press, New York.

[Shary 1996] S.P. Shary: Algebraic approach to the interval linear static identification, tolerance, and control problems, or one more application of Kaucher arithmetic. *Reliable Computing*, 2: 3–33.

[Shary 2002] S.P. Shary: A new technique in system analysis under interval uncertainty and ambiguity. *Reliable Computing*, 8(5): 321–418.

[Shimojima 1996] A. Shimojima: Operational constraints in diagrammatic reasoning. In: [DIAGRAMS 1996], 27–48.

[Shimojima 2001] A. Shimojima: The graphic-linguistic distinction: Exploring alternatives. In: [DIAGRAMS 2001], 5–27.

[Shin 1994] S.-J. Shin: *The Logical Status of Diagrams.* Cambridge University Press, New York.

[Shu 1988] N.C. Shu: *Visual Programming.* Van Nostrand Reinhold, New York.

[Sklansky 1970] J. Sklansky: Thresholded convolutions operations. *Journal of the ACM*, 17(1): 161–165.

[Sloman 1971] A. Sloman: Interactions between philosophy and AI: The role of intuition and non-logical reasoning in intelligence. *Artificial Intelligence*, 2: 209–225.

[Sloman 1975] A. Sloman: Afterthoughts on analogical representations. In: *Proc. 1st Workshop on Theoretical Issues in Natural Language Processing (TINLAP-1)*, Cambridge, MA, 164–171. (Reprinted in: [KNOWLREPR 1985], 432–439.)

[Sowa 1984] J.F. Sowa: *Conceptual Structures: Information Processing in Mind and Machine.* Addison Wesley, Menlo Park, CA.

[Stąpor 2000] K. Stąpor (2000) Geographic map image interpretation—survey and problems. *Machine GRAPHICS & VISION*, 9: 497–518.

[Steinhaus 1950] H. Steinhaus: *Mathematical Snapshots.* Oxford University Press, Oxford. (The 3rd edition: 1983; Polish edition: *Kalejdoskop matematyczny.* PZWS, Warszawa 1956.)

[Stenning 2000] K. Stenning: Distinctions with differences: Comparing criteria for distinguishing diagrammatic from sentential systems. In: [DIAGRAMS 2000a], 132–148.

[Stenning & Lemon 2001] K. Stenning, O. Lemon: Aligning logical and psychological perspectives on diagrammatic reasoning. In: [DIAGRAMS 2001], 29–62.

[Stenning & Oberlander 1995] K. Stenning, J. Oberlander: A cognitive theory of graphical and linguistic reasoning: logic and implementation. *Cognitive Science*, 19: 97–140.

[Sugihara 1986] K. Sugihara: *Machine Interpretation of Line Drawings.* The MIT Press, Cambridge, MA.

[Sunaga 1958] T. Sunaga: Theory of an interval algebra and its application to numerical analysis. *RAAG Memoirs*, 2: 547–564.

[Szalas 1992] A. Szalas: *Zarys dedukcyjnych metod automatycznego wnioskowania* [*An Outline of Deductive Automatic Inference Methods*, in Polish]. Akademicka Oficyna Wydawnicza RM, Warsaw.

[Tadeusiewicz 1977] R. Tadeusiewicz: Próba zastosowania rozpoznawania obrazów w diagnostyce neuroinfekcji [An attempt to apply image recognition in neuroinfection diagnostics, in Polish], In: *Systemy informatyczne w diagnostyce i terapii* [Computer Systems in Diagnostics and Therapy, in Polish], Medical Academy, Cracow, 68–76.

[Tadeusiewicz 1985] R. Tadeusiewicz: *Rozpoznawanie obrazów—zarys teorii* [*Recognition of Images—An Outline of a Theory*, in Polish], Jagiellonian University Textbooks, vol. 499, Cracow.

[Tadeusiewicz 1992] R. Tadeusiewicz: *Systemy wizyjne robotów przemysłowych* [*Vision Systems of Industrial Robots*, in Polish]. WNT, Warsaw.

[Térouanne 1983] E. Térouanne: "Impossible figures" and interpretations of polyhedral figures. *Journal of Mathematical Psychology*, 27(4): 370–405.

[Thiéry 1895] A. Thiéry: Über geometrisch-optische Täuschungen. *Philosophische Studien*, 11(3): 307–370.

[Tufte 1983] E.R. Tufte: *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT.

[Tufte 1990] E.R. Tufte: *Envisioning Information*. Graphics Press, Cheshire, CT.

[Tufte 1997] E.R. Tufte: *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press, Cheshire, CT.

[Tye 1991] M. Tye: *The Imagery Debate*. The MIT Press, Cambridge, MA.

[van Beek & Cohen 1990] P. van Beek, R. Cohen: Exact and approximate reasoning about temporal relations. *Computational Intelligence*, 6: 132–144.

[Vilain et al. 1990] M.B. Vilain, H. Kautz, P. van Beek: Constraint propagation algorithms for temporal reasoning—A revised report. In: *Readings in Qualitative Reasoning about Physical Systems*. Morgan Kaufmann, San Mateo, CA, 373–381.

[Waltz 1975] D. Waltz: Understanding line drawings with shadows. In: P.H. Winston, ed.: *The Psychology of Computer Vision*, McGraw-Hill Co., New York, 19–91.

[Wang & Lee 1993] D. Wang, J.R. Lee: Visual reasoning: Its formal semantics and applications. *Journal of Visual Languages and Computing*, 4(4): 327–356.

[Wang et al. 1995] D. Wang, J.R. Lee, H. Zeevat: Reasoning with diagrammatic representations. In: [DIAGRAMS 1995], 339–393.

[Warmus 1956] M. Warmus: Calculus of approximations. *Bull. Acad. Polon. Sci., Cl. III*, IV(5): 253–259. [For on-line scans of the paper, see `http://www.ippt.gov.pl/~zkulpa/quaphys/warmus.html#56`]

[Warmus 1961] M. Warmus: Approximations and inequalities in the calculus of approximations: Classification of approximate numbers. *Bull. Acad. Polon. Sci., Ser. Math, Astr. et Phys.*, IX(4): 241–245. [For on-line scans of the paper, see `http://www.ippt.gov.pl/~zkulpa/quaphys/warmus.html#61`]

[Winterstein et al. 2000] D. Winterstein, A. Bundy, M. Jamnik: A proposal for automating diagrammatic reasoning in continuous domains. In: [DIAGRAMS 2000a], 286–299.

[Winterstein et al. 2002] D. Winterstein, A. Bundy, C. Gurr, M. Jamnik: Using animation in diagrammatic theorem proving. In: [DIAGRAMS 2002], 46–60.

[Young & Deręgowski 1981] A.W. Young, J.B. Deręgowski (1981) Learning to see impossible. *Perception*, 10: 91–105.

Appendix:
English-Polish dictionary of basic terms

The dictionary contains Polish versions of basic terms used in the work, covering the fields of *diagrammatics* and *interval analysis* with addition of a number of terms from the *picture processing* field. For more of the picture processing terminology see [Mokrzycki 1992b], though some of the proposals included there have not been adopted in Polish practice. Certain common terms and terms from established disciplines, e.g. mathematics (like *lattice, quantifier*) or expert systems (like *expert system, forward chaining*) are usually not included, except in certain selected cases, especially when they constitute a root term of a family of specific terms in the covered disciplines (like *picture, relation, knowledge*). For such common terms, page numbers are not given. The page numbers provided do not point to all occurrences of the given term in the text, only to the places where the term is defined or more thoroughly discussed or used. The repeating main term components are abbreviated by their first letters (possibly with a -suffix added). Within lists of sub-terms, the terms so abbreviated are listed alphabetically as if the whole word was present. Polish terms in square brackets [*in italics*] constitute alternative versions that appear sometimes in Polish literature or speech, but are not recommended for use (at least by this author).

A

absolute (orientation) → visual property
accidental alignment, 103-104, 136: ustawienie przypadkowe (niezamierzone)
adjacency → visual relation mode
after → basic interval relation
aligning → visual relation mode
ambiguity point, 32: punkt niejednoznaczności
analogicity, 81-82, 89-90, 118, 140: naśladowczość [*analogiczność*]
angle → visual property
approximate number, 163-165, 171: liczba przybliżona
area → visual property
area measurement → measurement
arrangement (of intervals), 184: ułożenie (przedziałów)
articulated (shape) → visual property
association → visual relation use
assumption: założenie
  closed world a., 59, 105: z. zamkniętości świata
  domain closure a., 59: z. zamkniętości dziedziny
  negation by omission a. → rule
axis: oś
  characteristic a., 247, 252: o. charakterystyczna
  cutting a., 242, 247-251, 253, 260, 262: o. przecinająca
  interval a. → interval
    negative (interval) semi-a. → interval axis
    positive (interval) semi-a. → interval axis
  rotated a., 242-245, 253: o. obrócona
  singular boundary a., 248: o. brzegowa osobliwa

K

L

S

W

Z