ABSTRACT SCENE SPECIFICATIONS

by

Hong-Yee Wong

Department of Computer Science

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

Faculty of Graduate Studies
The University of Western Ontario
London, Ontario
March 1997

0-612-21323-4

ABSTRACT

One of the most efficient methods for the transfer of information from a computer to a human being is the graphical-visual form. The choice of specification method is crucial — it determines the flexibility with which the information can be used for a variety of purposes or output by the various output devices.

In this thesis, I investigate the properties of a declarative scene specification method for the description of pictures composed only of rectangles. A careful analysis is made and significant geometrical information is extracted from the basic declarations. I also examine the realizability of such specifications. Even restrictive versions of the realizability problem are surprisingly difficult, and the decidability of realizability remains an open problem.

To my dearest parents

献给我亲爱的父母

ACKNOWLEDGEMENTS

I wish to express my sincere thanks to my advisor Dr. Helmut Jürgensen for his knowledgeable advice and inspiring guidance. His infectious drive and enthusiasm will be a continuing source of inspiration for me.

I am most grateful to my first teachers — my parents — for their support and encouragement throughout my academic career. I would also like to thank my children Victoria, Ayden, Anson and Vienna for constantly reminding me how to smile and be happy, even when the going gets tough.

Finally my deepest heartfelt thanks go to my wife Stephanie for all the love and support she has given me over the years. Her unselfishness and devotion to our children made it all possible.

TABLE OF CONTENTS

List of Symbols

Rectangles and squares

$Rect^{w,h}_{x,y}$	(real) rectangle with centre (x,y), width w and height h	22
$S^{\mathbb{Z}}$	unit integer square	53
$\mathbb{Q}Rect$	rational rectangle	58
$\mathbb{Z}Rect$	integer rectangle	54
Rect	the set of all (real) rectangles	22
$\mathbf{R}_{\mathbb{Q}}$	the set of all rational rectangles	58
$\mathbf{R}_{\mathbb{Z}}$	the set of all integer rectangles	54
$\mathbf{S}_{\mathbb{Z}}$	the set of all unit squares	53

Functions

$X_l(a)$	horizontal coordinate of the left edge of rectangle a	23
$X_c(a)$	horizontal coordinate of the centre of rectangle a	23
$X_r(a)$	horizontal coordinate of the right edge of rectangle a	23
$Y_b(a)$	vertical coordinate of the bottom edge of rectangle a	23
$Y_c(a)$	vertical coordinate of the centre of rectangle a	23
$Y_t(a)$	vertical coordinate of the top edge of rectangle a	23
$C(a)$	coordinates of the centre of rectangle a	23
$M(a,b)$	midpoint of touching for adjacent rectangles a and b	25

Binary relations on rectangles

$\bar{\nu}, \bar{\eta}$	geometric "North of" and "East of" compass relations	24
$\bar{\nu}_\psi, \bar{\eta}_\psi$	geometric compass relations for a picture ψ	24
ν, η	abstract "North of" and "East of" compass relations	27
$\omega_{\bar{\nu}}, \omega_{\bar{\eta}}$	geometric order relations	63
ω_ν, ω_η	abstract order relations	66
$\rho_{\bar{\nu}_\psi}, \rho_{\bar{\eta}_\psi}$	geometric extended compass relations	86
$\sigma_{\bar{\nu}_\psi}, \sigma_{\bar{\eta}_\psi}$	geometric extended compass relations	86
ρ_ν, ρ_η	abstract extended compass relations	87
σ_ν, σ_η	abstract extended compass relations	87

Others

List of Figures

CHAPTER 1

INTRODUCTION

The flow of information from a computer to a human being takes many forms. One of the most efficient so far, if not the most efficient, seems to be the graphical-visual form where the computer presents data in a graphical manner and the user assimilates the information visually. This graphical presentation is usually two-dimensional, and includes images, animations, diagrams as well as text. Text is considered as graphics, as it is displayed by graphical fonts representing the letters and words of a language.

A wide range of media is available for the output of this graphical information. They include dynamic displays such as CRTs as well as static displays such as paper output from a printer. They vary widely in characteristics as well as capabilities. As a consequence, the choice of specification method for graphical information is crucial — it determines the flexibility with which the information can be used for a variety of purposes or output by the various output media.

1.1 Problem description

There are many ways to describe graphical information. They can be classified into three major classes: *declarative*, *procedural*, and *image-based*. Declarative techniques describe the information as a set of objects. The relationships among the objects are stated as a set of conditions. They are also known as *semantic-based* techniques, as the relationships among objects convey the structure and meaning of the scene. Procedural and image-based techniques describe the information explicitly at a low level. A procedural technique represents the information by a collection of low level drawing commands for drawing lines, circles, etc. Coordinates have to be used to position these drawing commands. An image-based technique represents visual information by storing a likeness of the desired output, usually in the form of a two-dimensional array of color values. Both procedural and image-based techniques can be considered as *non-semantic-based* techniques.

Clearly these techniques are useful in different situations. Image-based techniques are employed when fast rendering is required on a device for which the image is used. Procedural techniques make it easier to specify and port through the interface or software as long as the devices are not vastly different. Declarative techniques are used when having information about the contents or meaning of the image is crucial.

Let us consider what an ideal system for the description of graphical information might look like. It should possess the following characteristics:

1. a clear distinction is made between the declarative specification and the possibly procedural rendering of graphical information;

2. semantic information is present, or at least easily available, so that it is possible to reason about the contents of the image;

3. structure existing in the image is reflected by structure in the specification;

4. the specification is unambiguous, so that complex information can be accurately described and output;

5. the specification is intuitive so that a user can manipulate the information with relative ease;

6. the specification should not make reference to any device-specific features such as coordinates or resolutions so that the information can easily be translated for output on different devices and/or media.

I focus on a very much simplified abstraction of the ideal system to be called *picture specification*. Consider a picture that consists only of rectangles. The rectangles touch each other, but do not overlap. They are allowed to be stretched or shrunk in both the horizontal or vertical directions. An example is given in Figure 1.1.

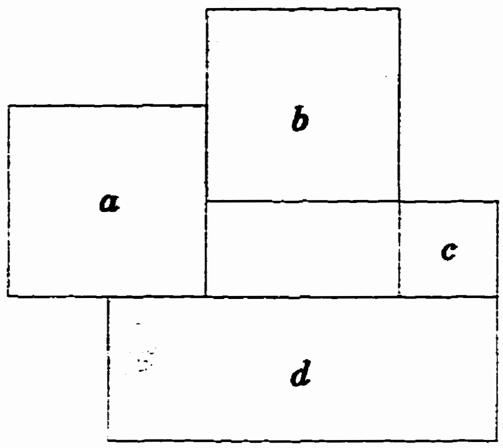


Figure 1.1: Example of a picture.

One way to specify such a picture is to locate the corners of the rectangles by coordinates. If one does not wish to use coordinates, one can use the spatial relationships between the rectangles to specify the picture.

I want to use abstract relations to describe a geometric layout of rectangles. Let us define two binary relations ν and η on the rectangles. For rectangles a and b, $(a.b) \in \nu$ if a touches b and is to the North of b. Similarly, $(a.b) \in \eta$ if a touches b and is to the East of b.

In the example in Figure 1.1, the corresponding relations to describe the picture are

$$\nu = \{(a,d).(c.d)\} \qquad \text{and} \qquad \eta = \{(b,a)\}.$$

Obviously, a given pair of ν and η is by no means sufficient to describe all or even just all of the important aspects of a picture. Descriptions of this form are investigated to gain some basic insights in the possibilities and pitfalls of coordinate-free scene description methods.

1.2 Motivation

In this section, I will discuss the motivation for adopting a declarative description of a picture by highlighting some of the problems associated with procedural or image-based methods.

Note that the focus of the thesis is to study the techniques that can be used in declarative methods. The results of the thesis are not directly applicable to the solution of the problems to be presented.

1.2.1 Semantic information

In many situations, the semantic information contained in graphical information is required explicitly. If it is not provided as part of the description of the image, a series of involved computations, probably using algorithms in image analysis, object recognition or reasoning and other techniques, is required to extract the semantic information. I give two examples of when semantic information could be useful or necessary and when the cost for image analysis may be unacceptable.

Visually impaired user

Consider a computer system that represents scenes by declarative techniques. The scene can be translated to be output in different forms. For a sighted user, a graphical output of the scene can be used. A graphical display device such as a CRT can be used to display a rendering of the scene. For a blind user, the same information can be translated to other output devices. For example, the scene can be translated to a verbal format. With the presence of semantic information in a declarative technique, it would be relatively easy to produce a detailed spoken explanation of the scene. An example of a scene described with semantics is

> "This scene is composed of a table, two chairs, a book and a pencil in a room. The book is on the table and the pencil is on the book. The chairs are to the right of the table."

Such relational description allows information to be derived from the initial declarations. For example, if the user queries

> "Is the pencil on the floor?"

The system would be able to state

> "No."

If the scene was represented by an image, a photograph, for example, or even as a collection of individual entities with coordinates to position them, none of the functionality described above would be available immediately as no semantic information would be present.

Information retrieval

A procedural description of a picture lacks information regarding the relationships between the components within the picture. Suppose one has a database of pictures or movies. Without semantics, one is unable to implement any semantic search capabilities for information retrieval. Suppose one needs to retrieve a scene as follows:

> "There is a table. A jar of chrysanthemums is on the table. The table is next to a window."

Such a query can only be performed in a database that includes semantic information [27]. Currently the only available methods for this functionality involves adding the semantic information manually — to the extent queries can be foreseen.

1.2.2 Typesetting

The typesetting of text and graphics poses many challenges. I present some of the problems encountered when procedural techniques are used.

Resolution problem

In order to visualize pictures, one has to render them on some kind of output device. The class of raster-based devices is the most common by far. The following is a partial list of such devices, with their approximate resolution in dots-per-inch (dpi).

- Computerized Braille tactile graphics display or Braille printer (about 7 dpi)
- Dot matrix impact printers (60–240 dpi)
- CRT monitors (70–100 dpi)

- Facsimile machines (100–200 dpi)
- Ink-jet printers (300–600 dpi)
- Laser printers (300–1200 dpi)
- Photo typesetters (2400 dpi or more)

When a coordinate-based picture is created, one usually has to consider the capability of the output device. If this picture is displayed on devices with different resolutions, one may get some undesirable results.

Consider the example shown in Figure 1.2. Suppose one has a simple picture consisting of two squares that are non-touching. When rendered at a higher resolution, the two indeed do not touch each other. But when the same picture is rendered on a lower resolution device, they may touch. This is undesirable in most cases.

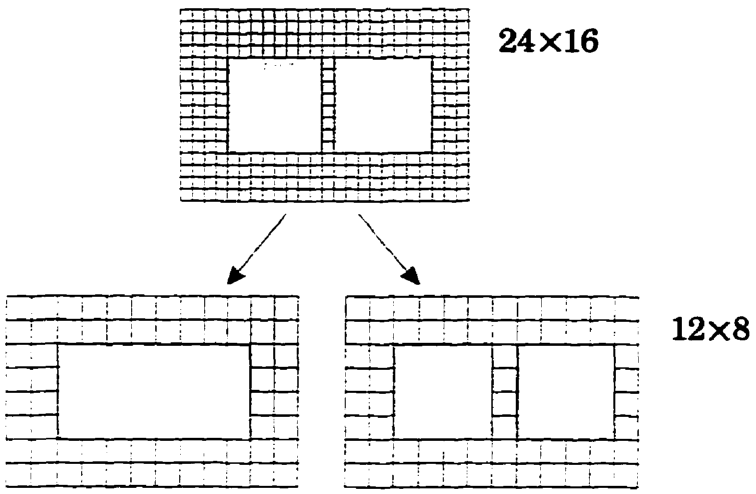


Figure 1.2: Resolution problem.

In a declarative description, important conditions that have to be met for each rendering are given in the description. Then the rendering program would ensure these conditions are satisfied in the rasterization process.

The resolution problem is especially bad when very low resolution devices such as a computerized Braille output device are considered. Obviously, graphical objects must be greatly simplified to be displayed at very low resolution. If this is done by simple scaling, shapes may be lost, and the crucial semantics information contained in a picture may be missing in the representation. This is devastating to a visually impaired user.

According to [23], the conversion process has to ensure that the renderings at low resolution satisfy the following requirements:[1]

- features must be large and have simple shapes;
- the distance between unconnected parts has to be large;
- only straight lines parallel to the axes can be used;
- features should only appear in predictable contexts.

Clearly a prerequisite to this conversion process is a clear understanding of the structure and meaning of the picture to be displayed. This can only be achieved when semantic information is provided.

Alignment problem

Suppose one needs to compose a picture of a musical note. The note is composed by putting together a flag, a vertical staff and the head. To produce a correct image, the position of the head may have to be shifted slightly, depending on the *rendered* thickness of the staff. Figure 1.3 illustrates two possible cases. The correct alignment is shown on the left, and an incorrect alignment on the right. If the composition is described by coordinates, it would be difficult to ensure the correct alignment of all objects over different output devices.

This problem can be eliminated in a declarative approach by specifying that the right edge of the stem must be aligned with the right edge of the oval. Then the device-specific rendering procedures would ensure that the composition is perfectly aligned.

Music

In the specification of musical score, the purposes are

- printing/displaying;
- playing by synthesizer;
- searching for a motif.

[1]Beyond Braille displays there are other devices for tactile output that could have slightly different requirements. The point here is that low resolution imposes particularly restrictive conditions on the rendering.

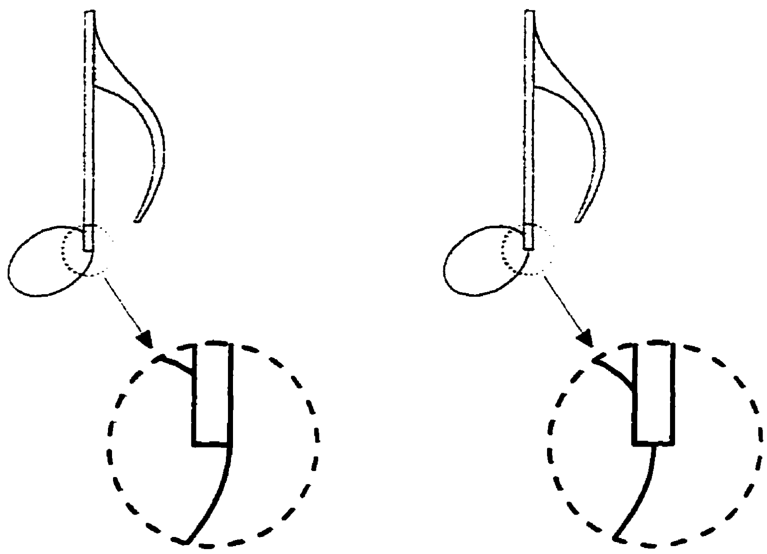


Figure 1.3: Alignment problem.

This requires that the specification reflect the musical structure, not the layout of the score or the sound-track [16], [15], [12].

PostScript

In [13], Dunne discovered a problem with the use of coordinates in the PostScript page description language. PostScript is commonly used in laser printers to control the rendering of images. Line and rectangle drawing are not precisely defined in PostScript. Due to the use of coordinates and discrete units for measures, the rendered output is incorrect. For example, it is impossible to correctly render a rectangle that is one device pixel high or wide.

Symbols for the blind

Consider a graphical rendering system for the blind. Suppose one needs to represent a scene containing a table and several other objects. Figure 1.4 shows four representations for a table. They are derived from drawings made by blind people. Representation (a) is seldom used, since a projection view of a three-dimensional object has no meaning to a congenitally blind person. Representation (d) is used when the table is surrounded by other objects. Representations (b) and (c) are used when there are other objects to be placed on the tabletop [29], [30].

Ideally, the description of the scene is hierarchical. The top level description

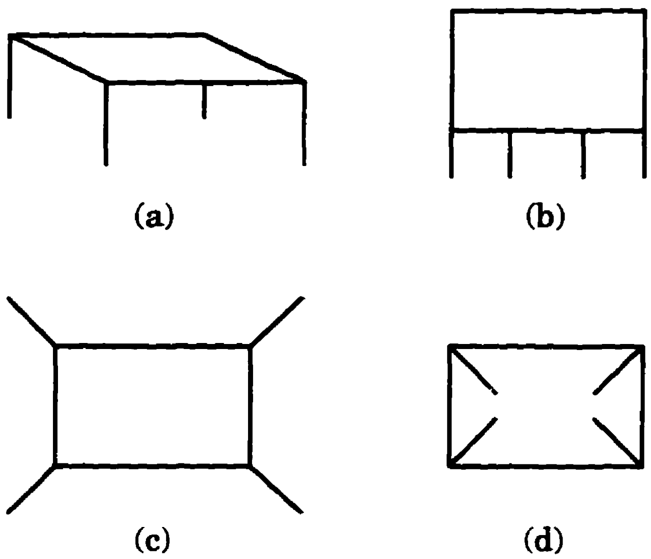


Figure 1.4: Representations of a table for the blind.

would describe the relative positions of the objects in the scene. The next level would describe the properties and shapes of the objects. In this way, an appropriate form for the table will be selected for rendering, depending on the objects around it.

In a procedural approach, the scene is specified by a collection of drawing instructions for all the objects in the scene. There is no information regarding the relationships among objects. There may not even be a distinction between the objects. This means that the graphical representation for all objects, including the table for example, must be preselected and explicitly described in the specification.

Design of Ideographic Characters

In the computerized typesetting of ideographic languages, the high-resolution output of ideographs has always been one of the most difficult and challenging problems. The difficulty arises in part from the huge number of characters required for such languages. Storing the complete high-resolution raster information is not only prohibitively expensive, but also prevents the individual users from making any significant style modifications to existing font libraries.

The analogy of typesetting a Chinese character is the typesetting of an English word. English words are basically a one-dimensional string of letters. The alphabet is small and difficulties in inter-letter spacing can be handled by special cases. On the other hand, the structure of a Chinese character is two dimensional. The analogy of the English alphabet is the set of sub-characters, sometimes known as (word) radicals. One radical can be composed of others and the set of radicals is very large (somewhere between 200 and 1000). Relative sizing of the radicals, as well as inter-radical spacing (in two dimensions) cannot be specified by fixed amounts, nor can they be handled by special cases.

One strategy for the output of Chinese characters is to use a font generator that is capable of generating fonts in different styles. A distinction is made between the style-independent and style-dependent aspects of font specification. The style-independent information consists of a specification of the generic structure of characters independent of font styles. Supplied with the appropriate font style characteristics, the images can be generated.

Figure 1.5 shows a Chinese character printed in different font styles. As one can observe from the sample characters, it is not possible for the generic structures of characters to be specified with fixed coordinate positions, hence a procedural description cannot be used. Placement positions of strokes and sub-characters vary with different styles. The relative sizes of the sub-characters are also different. These parameters must be specified by higher-level declarations such as "radical x is above radical y and they are both to the right of radical z". This information is then interpreted during font generation, where font-dependent routines compute the final sizes, shapes and positions of all strokes and radicals [47], [24].

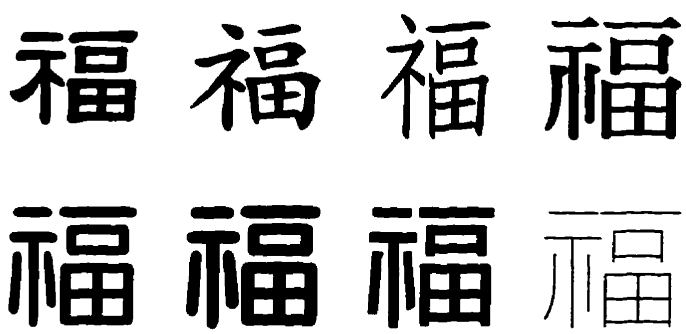


Figure 1.5: Sample of Chinese font styles.

1.3 Some high-level graphics systems

I have discussed some drawbacks of scene description without semantics. Let us now look at some high-level graphics systems and study how they represent and render graphical information.

1.3.1 Sketchpad

Sutherland's sketchpad [44] pioneered the use of constraints in graphics systems. It provides an interactive graphical interface, and its users could construct drawings by defining geometric constraints interactively.

Points, lines and circular arcs are primitive graphical objects in Sketchpad. Any drawing can be used as a primitive object by turning it into a "macro". A macro has a set of attachment points that are used to merge an instance of it into another drawing.

The geometric constraints include making two lines parallel, perpendicular, or of equal length. Lines can also be made horizontal or vertical, and a point can be constrained to lie on a line or arc. With the use of such constraints, a user can compose a drawing without explicitly stating the coordinates of every primitive object.

The macro feature allows objects to be built, but does not allow constraints to be specified with the objects. Hierarchy is not used effectively and as a consequence constraints can only to be stated with the primitive graphical objects, making it difficult to draw pictures.

1.3.2 IDEAL

Van Wyk's IDEAL [45] is a language for typesetting graphics into documents. IDEAL allows images to be built hierarchically, using boxes. For example, this is a definition for a rectangle:

```
rect {
     var ne, nw, sw, se, center, height, width;
     ne = se + (0, 1) * height;
     nw = sw + (0, 1) * height;
     ne = nw + width;
     center = (ne  + sw) / 2;
```

```
conn ne to nw to sw to se to ne;
}
```

IDEAL uses constraint satisfaction to allow the positions and sizes of objects to be stated as relationships, which makes it very easy to lay out complex figures. For example, one can place some rectangles side by side across the width of a page without specifying their individual widths, except that they are each to be of equal width. The width of each object will then be determined by the width of the page, even if the width of the page changes at a later time. This makes the description of images more flexible and natural.[2]

However, the only primitive data type in IDEAL is a point. All relations between the nonprimitive data types must be expressed in terms of primitives, which are points. So to specify that "rectangle *a* touches rectangle *b* and is to the East of rectangle *b*", one needs to state the low level constraint: $\mathtt{ne}_\mathtt{b} = \mathtt{nw}_\mathtt{a}$. As in Sketchpad, the hierarchy of construction is not used in the specification of constraints. It would be more intuitive if one were able to define constraints in terms of the nonprimitive objects. Semantical information in the sense of our "ideal system" cannot be specified.

1.3.3 Juno

Nelson's Juno [34] is a system which integrates a constraint-based language with an image editor. With Juno, a constraint-language program can be represented either in its textual form or by the image it produces. The user can edit either form, and the changes will be reflected back into the program.

Juno's intended domain is very limited. Its only data object is the point, and there are only four constraints on points. New constraints can only be added if they can be expressed as a conjunction of the four primitive constraints. If one wants to use Juno for a different application, such as three-dimensional graphics, the underlying system would have to be modified extensively.

1.3.4 COOL

In [26], a constraint-based system named COOL is built to visualize abstract objects and relations. First data are translated into the relational structure

[2] TeX offers similar capabilities for boxes.

representation of abstract objects and relations. Second, abstract objects are mapped to graphical objects, and abstract relations between them are mapped to graphical relations between the corresponding graphical objects. Finally, an actual layout of graphical objects is computed by solving graphical constraints, and then a picture is generated.

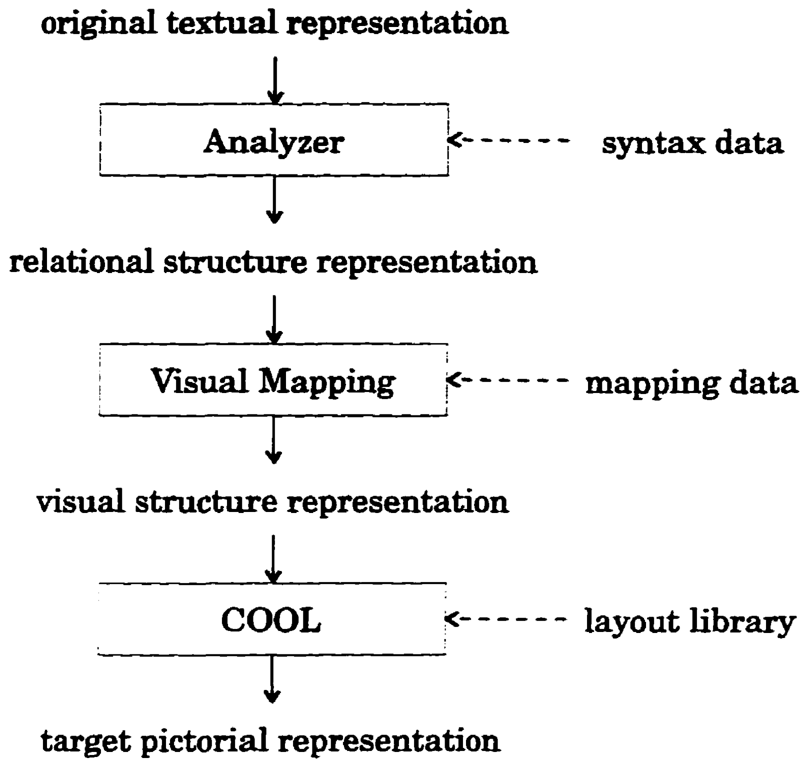


Figure 1.6: Visualization model proposed by Kamada (from [26]).

The positions of graphical objects are computed automatically from the specified graphical relations by the system. The geometric relations are expressed as algebraic constraints among the variables which characterize the graphics objects. A constraint hierarchy is built to handle over- or under-constrained situations.

Graphical objects in COOL are similar to boxes in IDEAL. They have local variables that are related to one another by equations. Graphical relations among graphical objects are expressed as extra constraints among the variables of the objects. Some examples of pictures generated by COOL is shown in Figure 1.7.

The constraint solver in COOL is based on a simple equation solver. If a set of constraints is over- or under-constrained, the system reports an error. The

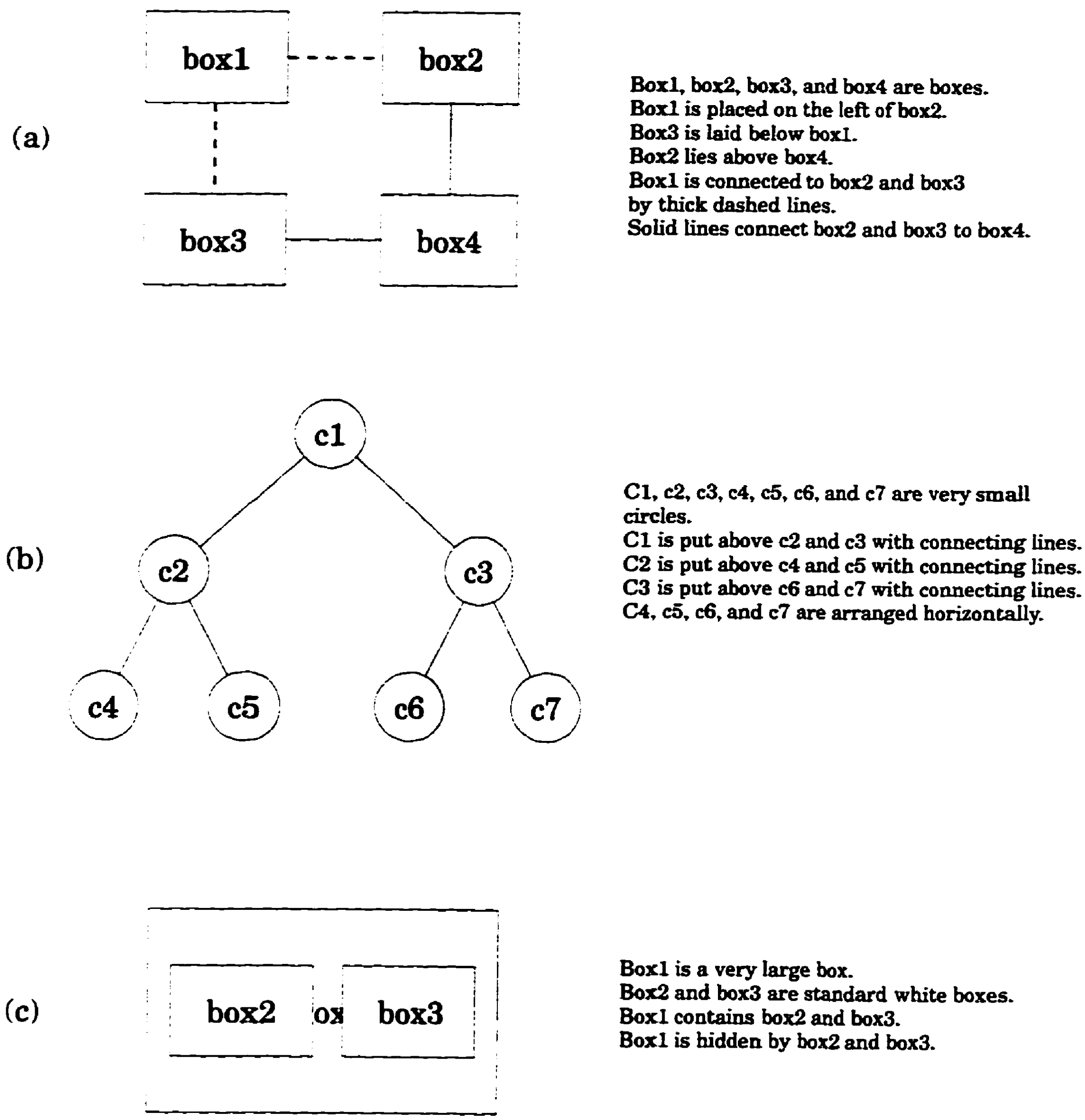


Figure 1.7: Examples of picture generation in COOL (from [26]).

user is then responsible to divide the constraints into two types, the "rigid" constraints which must be satisfied exactly and the "pliable" constraints that need to be satisfied not exactly but approximately. Rigid constraints are eliminated like Gaussian elimination. After that, pliable constraints are solved by the least square method.

COOL does have many of the characteristics of an ideal system. It maintains a clear distinction of the specification and rendering of pictures, and is able to generate pictures with the use of constraint-based techniques. However, as with most constraint-based techniques, over- and under-constrained situations cannot be resolved easily.

If a constraint solver is unable to solve a set of constraints, it reports an error. The conflict involves local conditions, and may not be easily expressed in the global layout scheme. This makes it difficult to identify the high-level constraints that have to be modified to resolve the problem.

1.3.5 Constraint-Based Reasoning

In [20], the "office world", consisting of windows, desks, chairs, computers, etc, is presented as an example domain for different basic forms of spatial reasoning. One way of describing the sketch of an office as given in figure 1.8 is by the following proposition:

> *The chair is left of both the desk and the lamp, and the desk is left of the lamp.*

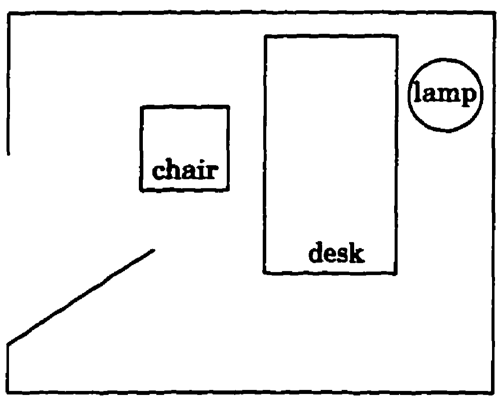


Figure 1.8: A sketch of an office layout (from [20]).

The *left of* relations between the objects can then be said to *constrain* the positions of the objects. So, these relationships can be represented by a constraint network on the variables *chair, desk,* and *lamp*, where each variable specifies the distance of the respective object to a reference point on some horizontal axis. Suppose that the domains

$$D_{chair} = \{4.5.6\} \quad D_{desk} = \{5.6.7\} \quad D_{lamp} = \{6.7.8\}$$

are associated with the variables *chair, desk,* and *lamp*, respectively. Then the assignment *chair* $= 4$, *desk* $= 7$, *lamp* $= 8$ are a solution of the constraint network.

Figure 1.9 shows the set of relations used in [20]. A set of spatial propositions can be represented as a network consisting of two types of nodes: circle representing the objects and rectangles representing the relations. Reasoning about spatial relationships in a constraint reasoning setting now can be viewed as modifying the labels of the rectangles, that is, the constraints, and inserting new constraints into the network.

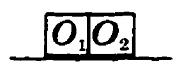
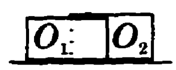
Relationship	Symbol	Symbol for Converse	Picture
O_1 *left of* O_2	$\prec$	$\succ$	
O_1 *attached to* O_2	$\preceq$	$\succeq$	
O_1 *overlapping* O_2	$\Leftarrow$	$\Rightarrow$	
O_1 *inside* O_2	$\sqsubset$	$\sqsupset$	

Figure 1.9: The possible relationships between two objects (from [20]).

Consider, for example, the network of Figure 1.10(a). Since O_4 is between O_1 and O_3, intuitively the spatial relation between O_1 and O_3 must be $O_1 \prec O_3$ (as shown in Figure 1.10(b)). From this, together with $O_2 \preceq O_3$, one can conclude that $\Rightarrow$ is not a possible relation between O_1 and O_2, because O_1 is at least as far left as O_2 (as shown in Figure 1.10(c)).

(a)

(b)

(c)

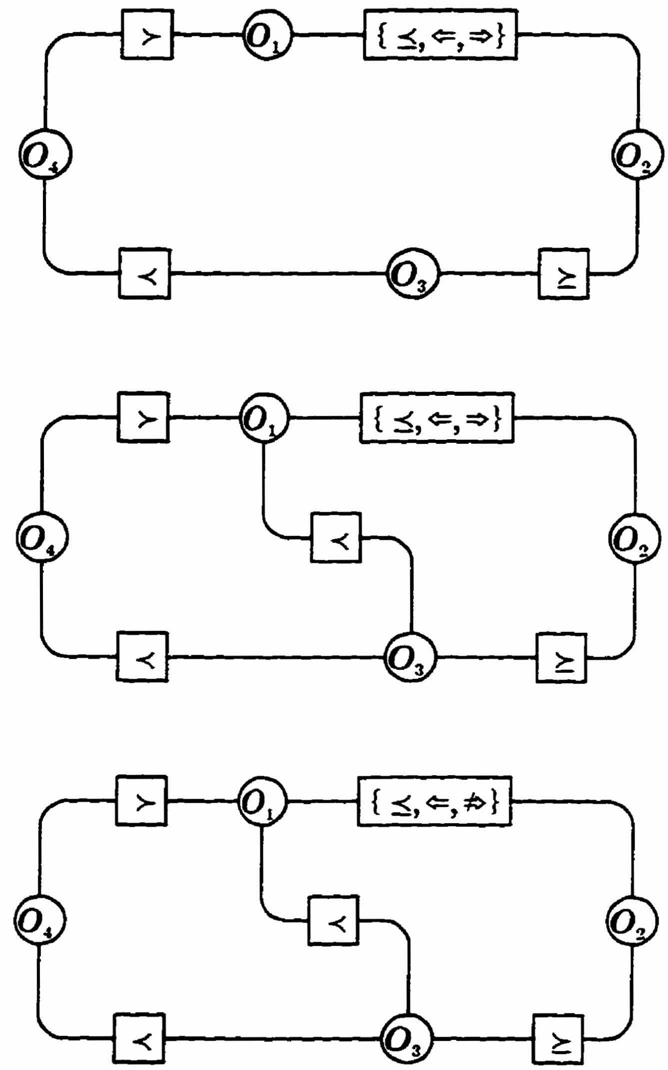


Figure 1.10: Reasoning about spatial relationships.

The example shows that there are two different reasoning steps on a network of spatial relations:

1. Computing the composition of spatial relations, i.e., inserting new constraints into the network.

2. Deleting all relations that are inconsistent.

Standard constraint satisfaction algorithms can be used to remove inconsistencies from the network. These algorithms result in different levels of consistency (ranging from local consistency to global consistency), and they are of different complexity (up to exponential).

The aim of the study in [20] is to introduce concepts and algorithms around the notion of dynamic constraints. One severe limitation that prevents it from being used as a method for scene description is the fact that it only deals with spatial relations in one dimension. As I shall show in a later chapter, it is not possible to treat two-dimensional spatial relations as two separate one-dimensional spatial relations.

1.4 Geometric dimensioning and tolerancing in computer aided design systems

In computer aided design (CAD) systems, computer models of object components are specified precisely with the use of coordinates. Problems arise in the specification of assemblies of components that are coordinate based. Assembly constraints are used to provide information on how component are connected. These geometric constraints present themselves as a set of algebraic constraints, to be solved by a constraint solver.

The satisfaction of assembly constraints can be performed by solving the algebraic equations that were derived from the geometric assembly constraints. The problem of constraint-solving is difficult as the constraints are highly coupled and non-linear [8],[39].

1.4.1 Summary

Besides the systems mentioned above, there are several other systems such as Bertrand [32], ThingLab [6], DeltaBlue [18] [40]. They are in many ways similar to the systems presented before as they all rely on constraint-based techniques. Constraint satisfaction by itself is known to be a very difficult problem. Techniques that can be applied to solve general constraint problems are inefficient and as a consequence most constraint-based systems usually employ domain-specific techniques and are not easily extensible. In IDEAL, one can define new structures that are like objects, but not new types of constraints on those objects. In Sketchpad and Juno, one can define new operations on their existing data types, but cannot define new types of objects.

Another problem with constraint-based techniques is the issue of numerical-stability. Systems such as Juno use iterative numeric techniques to solve constraints. They may fail to terminate even when the constraints have a solution. Moreover for constraints with more than one solution, an arbitrary solution is given.

Coordinates are imposed in a constraint-based graphics system. Even if coordinates are not used in the declaration of high-level constraints, they cannot be avoided in the constraint-solving process. Inconsistencies in the description of pictures present themselves as under- or over-constraints. These can only be detected by the failure in finding a set of numerical solutions for all the coordinate values. Although some form of constraint-solving may be inevitable in the construction of a realization, one should be able to determine the realizability of a picture by means other than to attempt to construct one. To determine the realizability of pictures, one needs to have a clear understanding of the underlying geometry, and derive rules to decide what is possible and what is not. A constraint-solver provides no understanding of the problem at hand, merely finding a solution by "brute-force" if it exists.

1.5 Thesis

So far I have shown the merits of declarative methods, and I have presented an overview of how existing systems use them. I have discussed the shortcomings of these existing systems due to their heavy dependence on constraint solving. What is lacking is a good understanding of the fundamental properties that are embedded in pictures specified with declarative methods.

In this thesis, I investigate the properties of pictures specified with declarative methods. The aim is to extract as much information as possible from the given specifications. Instead of considering general scene layouts, I focus on a simplified abstraction of declarative methods – a specification of rectangle pictures – as as proposed in [25]. Of particular interest is the question of realizability. I show that the proposed framework is insufficient for complete specification even under some very loose limitations to "similarity classes" of realizations. I exhibit a set of necessary conditions, for such a specification to be realizable. Moreover, I demonstrate that even restricted versions of the realizability problem are surprisingly difficult. The ultimate issue, that of deciding whether a given specification is realizable and, if so, to find a realization remains open. I believe, however, that my analysis will contribute to the understanding and eventual solution of the problem. The thesis is structured as follows:

- Chapter 2 introduces the basic geometric and abstract notions to lay the groundwork for the study.
- Chapter 3 presents the characteristics of specifications. The limitations of specifications are also discussed.
- Several examples of specifications are studied in Chapter 4 in order to help gain clues on the conditions required for realizability.
- In Chapter 5, the original specification problem is reduced by placing restrictions on the rectangles.
- In Chapter 6, "order relations" are introduced to augment the specification problem. In particular I focus on the "ω-relations" and exploit its use in order to infer a significant amount of geometric information.
- Chapter 7 summarizes the necessary conditions required for the realizability of specifications.
- The conclusion of the thesis and a discussion of further work is presented in Chapter 8.

CHAPTER 2

BASIC NOTIONS

This chapter lays the groundwork for the study of rectangle picture specifications. Geometric notions such as rectangles and their spatial relationships are introduced. An abstraction is made from these geometric notions to create a system of abstract rectangle symbols and their relations. The abstraction is free from coordinates and I will describe how it is used to specify pictures.

2.1 Binary relations

In this section I review some terminology and notations concerning binary relations. Let R be a set and let ξ be a binary relation on R, that is, $\xi \subseteq \mathsf{R} \times \mathsf{R}$.

- The *inverse* of ξ is defined as $\xi^{-1} = \{(r,r') | r,r' \in \mathsf{R}, (r',r) \in \xi\}$.
- ξ is *reflexive* if $(r,r) \in \xi$ for all $r \in \mathsf{R}$.
- ξ is *anti-reflexive* if $(r,r) \notin \xi$ for all $r \in \mathsf{R}$.
- ξ is *symmetric* if, for all $r,r' \in \mathsf{R}$, $(r,r') \in \xi$ implies $(r',r) \in \xi$.
- ξ is *anti-symmetric* if, for all $r,r' \in \mathsf{R}$ with $r \neq r'$, $(r,r') \in \xi$ implies $(r',r) \notin \xi$.
- ξ is *transitive* if, for $r,r',r'' \in \mathsf{R}$, $(r,r'),(r',r'') \in \xi$ implies $(r,r'') \in \xi$.
- ξ is *anti-transitive* if, for $r,r',r'' \in \mathsf{R}$, $(r,r'),(r',r'') \in \xi$ implies $(r,r'') \notin \xi$.
- A ξ-*cycle* is a finite sequence $\langle r_0, r_1, \ldots, r_n \rangle$ of symbols $r_0, r_1, \ldots, r_n \in \mathsf{R}$ such that $n \geq 1$, $r_0 = r_n$, $(r_i, r_{i+1}) \in \xi$ for $i = 0, 1, \ldots, n-1$, and $r_i \neq r_j$ for $0 \leq i < j < n$.
- If there are no ξ-cycles then ξ is anti-reflexive and anti-symmetric.

- When ξ is transitive then there are no ξ-cycles if and only if ξ is anti-symmetric.
- The *transitive closure* of ξ is defined as

$$\xi^+ = \{(r,r')|r,r' \in \mathsf{R}, \exists r'' \in \mathsf{R}\ (r,r''),(r'',r') \in \xi\}.$$

- ξ is a *partial order* if it is reflexive, anti-symmetric, and transitive.
- ξ is a *strict order* if it is anti-symmetric and transitive.

2.2 Geometric notions

I begin with defining the compass directions as an intuitive way of specifying directions in a two-dimensional plane.

DEFINITION 2.2.1 (COMPASS DIRECTIONS) *Consider a two-dimensional plane with the usual horizontal and vertical axes. The following directions are defined:*

1. *North is the direction of increasing vertical axis value;*
2. *South is the direction of decreasing vertical axis value;*
3. *East is the direction of increasing horizontal axis value;*
4. *West is the direction of decreasing horizontal axis value;*

The basic shapes to be considered are rectangles. A rectangle is defined as a set of points in $\mathbb{R}^2$.

DEFINITION 2.2.2 (RECTANGLE) *Let* $x_0, y_0, w, h \in \mathbb{R}$ *and* $w, h > 0$. *The* rectangle $Rect^{w,h}_{x_0,y_0}$ *is the set of all points within the region with the center* (x_0, y_0), *width* w, *and height* h, *that is,*

$$Rect^{w,h}_{x_0,y_0} \stackrel{\text{def}}{=} \{(x,y) \in \mathbb{R}^2 : x_0 - {}^w/_2 \le x \le x_0 + {}^w/_2, y_0 - {}^h/_2 \le y \le y_0 + {}^h/_2\}.$$

Let **Rect** *denote the set of all rectangles, that is,*

$$\mathbf{Rect} \stackrel{\text{def}}{=} \{Rect^{w,h}_{x_0,y_0} : x_0, y_0, w, h \in \mathbb{R}, w, h > 0\}.$$

In the sequel the notation $Rect_{x_0,y_0}^{w,h}$ is intended to imply, without special mention, that $x_0, y_0, w, h \in \mathbb{R}$ and $w, h > 0$. The following gives some useful abbreviations for representing the coordinates of the sides and centre of rectangles. For a rectangle $Rect_{x_0,y_0}^{w,h}$ the following items are defined:

1. $X_l(r) = x_0 - {}^w\!/_2$
2. $X_r(r) = x_0 + {}^w\!/_2$
3. $X_c(r) = x_0$
4. $Y_t(r) = y_0 + {}^h\!/_2$
5. $Y_b(r) = y_0 - {}^h\!/_2$
6. $Y_c(r) = y_0$
7. $C(r) = (x_0, y_0)$

These items are illustrated in Figure 2.1.

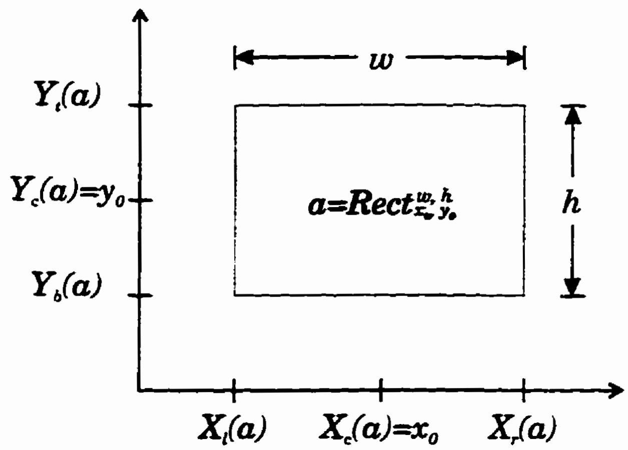


Figure 2.1: A rectangle and its coordinate abbreviations.

DEFINITION 2.2.3 (EXTENTS OF A RECTANGLE) *For a rectangle* $r = Rect_{x_0,y_0}^{w,h}$, *the open intervals of real values* $\left(X_l(r), X_r(r)\right)$ *and* $\left(Y_b(r), Y_t(r)\right)$ *are called the* horizontal *and* vertical extents *of r, respectively.*

Next we need the two binary relations to describe the spatial relationships between the rectangles.

Definition 2.2.4 (Geometric compass relations) *Binary relations $\bar{v}$ and $\bar{\eta}$ for rectangles to represent "North of" and "East of" are defined as follows:*

$$\bar{v} = \left\{(a,b),\ a,b \in \mathbf{Rect}:\ Y_b(a) = Y_t(b) \wedge \left(X_l(a),X_r(a)\right) \cap \left(X_l(b),X_r(b)\right) \neq \emptyset\right\}.$$
$$\bar{\eta} = \left\{(a,b),\ a,b \in \mathbf{Rect}:\ X_l(a) = X_r(b) \wedge \left(Y_b(a),Y_t(a)\right) \cap \left(Y_b(b),Y_t(b)\right) \neq \emptyset\right\}.$$

These relations are called the geometric compass relations.

Notice that open intervals are used. This excludes cases called "corner kissing" in [25]. An example of corner kissing can be found in Figure 1.1, where $(b,c) \notin \bar{v} \cup \bar{v}^{-1} \cup \bar{\eta} \cup \bar{\eta}^{-1}$.

Now I introduce the notion of a picture, which is a collection of non-overlapping rectangles.

Definition 2.2.5 (Picture) *A* picture $\psi \subset \mathbf{Rect}$ *is a set of rectangles with the property that no rectangles overlap, that is, for any distinct rectangles r_1 and r_2 in ψ,*

$$\left(X_l(r_1),X_r(r_1)\right) \cap \left(X_l(r_2),X_r(r_2)\right) = \emptyset \quad or \quad \left(Y_b(r_1),Y_t(r_1)\right) \cap \left(Y_b(r_2),Y_t(r_2)\right) = \emptyset$$

Next I present a lemma that is useful for later proofs. It is based on the fact that rectangles are not allowed to overlap.

Lemma 2.2.6 *Let r_1, r_2 be any two rectangles in a picture.*

1. *If $X_l(r_1) = X_l(r_2)$ or $X_r(r_1) = X_r(r_2)$ then*
$$\left(Y_b(r_1),Y_t(r_1)\right) \cap \left(Y_b(r_2),Y_t(r_2)\right) = \emptyset.$$
2. *If $Y_b(r_1) = Y_b(r_2)$ or $Y_t(r_1) = Y_t(r_2)$ then*
$$\left(X_l(r_1),X_r(r_1)\right) \cap \left(X_l(r_2),X_r(r_2)\right) = \emptyset.$$

Proof:

1. If $X_l(r_1) = X_l(r_2)$ or $X_r(r_1) = X_r(r_2)$ then
$$\left(X_l(r_1),X_r(r_1)\right) \cap \left(X_l(r_2),X_r(r_2)\right) \neq \emptyset.$$

   Since no overlaps are allowed, one has, by Definition 2.2.5,
$$\left(Y_b(r_1),Y_t(r_1)\right) \cap \left(Y_b(r_2),Y_t(r_2)\right) = \emptyset.$$

2. If $Y_b(r_1) = Y_b(r_2)$ or $Y_t(r_1) = Y_t(r_2)$ then
$$\left(Y_b(r_1), Y_t(r_1)\right) \cap \left(Y_b(r_2), Y_t(r_2)\right) \neq \emptyset.$$
Again, by Definition 2.2.5, one has
$$\left(X_l(r_1), X_r(r_1)\right) \cap \left(X_l(r_2), X_r(r_2)\right) = \emptyset.$$

□

The geometric compass relations are defined on the set **Rect** of all rectangles in $\mathbb{R}^2$. We want to be able to refer to the geometric compass relations for a picture.

DEFINITION 2.2.7 (GEOMETRIC COMPASS RELATIONS FOR A PICTURE) *The relations $\bar{\nu}_\psi$ and $\bar{\eta}_\psi$ denote the geometric compass relations for the rectangles in a picture ψ. That is to say, $\bar{\nu}_\psi = \{(a,b), a,b \in \psi : (a,b) \in \bar{\nu}\}$ and $\bar{\eta}_\psi = \{(a,b), a,b \in \psi : (a,b) \in \bar{\eta}\}$.*

DEFINITION 2.2.8 (MIDPOINT OF TOUCHING) *Let a and b be rectangles in a picture ψ such that $(a,b) \in \bar{\nu}_\psi \cup \bar{\nu}_\psi^{-1} \cup \bar{\eta}_\psi \cup \bar{\eta}_\psi^{-1}$. The intersection intervals of the rectangles are computed by*
$$(y_1, y_2) = \left(Y_b(a), Y_t(a)\right) \cap \left(Y_b(b), Y_t(b)\right)$$
and
$$(x_1, x_2) = \left(X_l(a), X_r(a)\right) \cap \left(X_l(b), X_r(b)\right).$$
The midpoint of touching *is defined as:*
$$M(a,b) = \begin{cases} \left(\frac{x_1+x_2}{2}, Y_b(a)\right) & \text{if } (a,b) \in \bar{\nu}_\psi \\ \left(\frac{x_1+x_2}{2}, Y_b(b)\right) & \text{if } (b,a) \in \bar{\nu}_\psi \\ \left(X_l(a), \frac{y_1+y_2}{2}\right) & \text{if } (a,b) \in \bar{\eta}_\psi \\ \left(X_l(b), \frac{y_1+y_2}{2}\right) & \text{if } (b,a) \in \bar{\eta}_\psi \end{cases}$$

The next notion to be introduced is the connection curve. It is illustrated in Figure 2.2.

DEFINITION 2.2.9 (CONNECTION CURVE) *Let a and b be rectangles in a picture ψ such that $(a,b) \in \bar{\nu} \cup \bar{\nu}^{-1} \cup \bar{\eta} \cup \bar{\eta}^{-1}$.* A connection curve *$C(a,b)$ from a to b is obtained by drawing straight line segments through the points*
$$C(a), M(a,b), C(b)$$

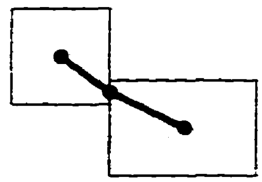


Figure 2.2: Connection curve for two adjacent rectangles.

There is a symmetry that exists for pictures. It is defined in the following.

DEFINITION 2.2.10 (SYMMETRY IMAGES OF A PICTURE) *For a picture* ψ, *the* symmetry images *of* ψ *are defined as the pictures obtained from applying a sequence of the following symmetry transformations on* ψ:

1. *mirroring about the horizontal axis* $y = 0$;
2. *mirroring about the vertical axis* $x = 0$;
3. *clockwise rotation by* $\pi/2$ *about the origin.*

The total of eight unique symmetry transformations (including the identity mapping) form the dihedral group of order 8 [43].

2.3 Abstract notions

I would like to extract an abstraction from the geometric rectangles and their relations. Instead of sets of rectangles I consider an alphabet of symbols denoting rectangles, *rectangle symbols*. I begin by defining a mapping of abstract symbols to rectangles.

DEFINITION 2.3.1 (ARRANGEMENT) *Let* R *be a set of symbols denoting rectangles. An* arrangement *of* R *is an injective mapping* $\alpha : \mathsf{R} \to \mathbf{Rect}$ *such that the resulting rectangles form a picture.*

Next I define an abstract specification of a picture. The specification requires a set of abstract symbols and two relations representing the abstraction of the geometric compass relations.

DEFINITION 2.3.2 (SPECIFICATION) *A* rectangle picture specification, *or in short a* specification, *is given by*

$$\mathcal{S} \stackrel{\text{def}}{=} (\mathsf{R}, \mathsf{v}, \eta)$$

where R *is a set of rectangle symbols, and* $\nu, \eta \subseteq R \times R$. ν *and* η *are called the* abstract compass relations. *The specification* $\mathcal{S}$ *is finite when* R *is finite.*

Most of the results derived in this thesis hold true regardless of whether $\mathcal{S}$ is finite or infinite; for the decidability issues introduced further below and for the intended application for image specifications, $\mathcal{S}$ will be required to be finite.

In the sequel, let $\mathcal{S} = (R, \nu, \eta)$ be an arbitrary, but fixed specification.

REMARK 2.3.3 *An arrangement* α *of a set* R *of rectangle symbols defines the geometric compass relation* $\bar{\nu}_\psi$ *and* $\bar{\eta}_\psi$ *on the set* $\psi = \alpha(R)$ *of rectangles. By slight abuse of terminology I use* $\bar{\nu}_\psi$ *and* $\bar{\eta}_\psi$ *also to denote the corresponding induced relations on the set* R *of rectangle symbols, that is,*

$$(r, r') \in \bar{\nu}_\psi \textit{ if and only if } (\alpha(r), \alpha(r')) \in \bar{\nu}_\psi$$

and similarly for $\bar{\eta}_\psi$. *Also, for the sake of simplicity, for* $\beta = \{X_l, X_r, X_c, Y_t, Y_b, Y_c, C\}$ *and any rectangle symbol* $r \in R$, *I write* $\beta(r)$ *instead of* $\beta(\alpha(r))$.

More definitions built on the abstract compass relations are introduced. We continue with the definition of adjacency (or touching).

DEFINITION 2.3.4 (ADJACENCY) *Let* $r_1, r_2 \in R$. r_1 *and* r_2 *are* ν-adjacent *if* $(r_1, r_2) \in \nu \cup \nu^{-1}$; *they are* η-adjacent *if* $(r_1, r_2) \in \eta \cup \eta^{-1}$; *they are* adjacent *if they are* ν*-adjacent or* η*-adjacent.*

A sequence of adjacent rectangle symbols forms a path. Open and closed paths are defined in the following.

DEFINITION 2.3.5 (PATH) *Let* $r_a, r_b \in R$. *A* path *from* r_a *to* r_b *is a sequence*

$$\mathcal{P} = \langle r_a = r_0, r_1, \ldots, r_n = r_b \rangle, \; n \geq 1$$

of rectangle symbols such that

$$(r_p, r_{p+1}) \in \nu \cup \nu^{-1} \cup \eta \cup \eta^{-1} \quad \textit{for } p = 0, \ldots, n-1$$

and

$$r_p \neq r_q \quad \textit{for } 0 \leq p < q \leq n.$$

Here n is called the length of the path, denoted by $|\mathcal{P}|$.

The path $\mathcal{P}$ *is said to be* closed *if* $r_a = r_b$; *it is* open *otherwise.*

Notice that the discussion is restricted to open and closed paths containing distinct rectangle symbols only. Paths that contain repeated rectangle symbols also exist, but they can be broken down into two or more paths that do not contain repeated rectangle symbols. Then each part can be examined on its own.

Next I define connectedness. It can be considered as the transitive extension of adjacency.

DEFINITION 2.3.6 (CONNECTEDNESS) r_1 *is said to be* connected *to* r_2 *if there exists a path from* r_1 *to* r_2.

DEFINITION 2.3.7 (CONNECTED SPECIFICATION) *A specification* $\mathcal{S}$ *is said to be* connected *if every pair of rectangle symbols in* $\mathcal{S}$ *is connected.*

In the remainder of the thesis, I assume that every specification is connected without explicitly stating so. Specifications that are not connected can be separated into two or more connected specifications. Then each connected specification can be dealt with individually.

If one is given a specification, one of the fundamental questions is to ask if there is a picture that corresponds to it. I call this realizability and define it in the following.

DEFINITION 2.3.8 (REALIZABILITY) *Consider a specification* $\mathcal{S} = (\mathsf{R}, \nu, \eta)$ *and an arrangement* α. *Let* $\psi = \alpha(\mathsf{R})$.

1. *If* $\bar{\nu}_\psi = \nu$ *and* $\bar{\eta}_\psi = \eta$, *then* α *is* acceptable *(or a* realization*) and* $\mathcal{S}$ *is* **Rect**-realizable. $\mathcal{S}$ *is* realizable *if it is* **Rect**-realizable.

2. *For a subset* $\mathbf{B} \subseteq \mathbf{Rect}$, *if* $\bar{\nu}_\psi = \nu$, $\bar{\eta}_\psi = \eta$, *and* $\psi \subseteq \mathbf{B}$, *then* α *is* **B**-acceptable *(or a* **B**-realization*) and* $\mathcal{S}$ *is* **B**-realizable.

Note: In the sequel, when it does not matter, the *realization* of a specification refers to α, as well as the picture ψ generated by α. Also, I sometimes write $\alpha(\mathcal{S})$ instead of $\alpha(\mathsf{R})$ for the sake of convenience.

The symmetry images of pictures were defined earlier. Now I define the symmetry images at the abstract level.

DEFINITION 2.3.9 (SYMMETRY IMAGES OF A SPECIFICATION) *The* symmetry images *of a specification* $\mathcal{S} = (\mathsf{R}, \nu, \eta)$ *and the symbols for denoting them are given by:*

$$\begin{aligned} {}_1\mathcal{S} &= (\mathsf{R}, \mathsf{v}, \eta) & {}_5\mathcal{S} &= (\mathsf{R}, \eta, \mathsf{v}) \\ {}_2\mathcal{S} &= (\mathsf{R}, \mathsf{v}, \eta^{-1}) & {}_6\mathcal{S} &= (\mathsf{R}, \eta, \mathsf{v}^{-1}) \\ {}_3\mathcal{S} &= (\mathsf{R}, \mathsf{v}^{-1}, \eta) & {}_7\mathcal{S} &= (\mathsf{R}, \eta^{-1}, \mathsf{v}) \\ {}_4\mathcal{S} &= (\mathsf{R}, \mathsf{v}^{-1}, \eta^{-1}) & {}_8\mathcal{S} &= (\mathsf{R}, \eta^{-1}, \mathsf{v}^{-1}) \end{aligned}$$

PROPOSITION 2.3.10 *Let* $\mathcal{S} = (\mathsf{R}, \mathsf{v}, \eta)$ *be a specification and let* α *be an arrangement of* R *such that* $\psi = \alpha(\mathsf{R})$ *is a realization of* R. *There is a one-to-one correspondence* ι *between the symmetry images* ${}_i\mathcal{S}$ *of* $\mathcal{S}$ *and the eight symmetry transformations in the dihedral group of order eight such that, up to translations of the plane,*

$$\iota[{}_i\mathcal{S}](\psi) = [\iota[{}_i\mathcal{S}] \circ \alpha](\mathcal{S}).$$

The correspondence stated above indicates that the abstract symmetry images correctly represent the geometric notions of the symmetry transformations.

Next, we want to relate a path in the abstract specification to a composition of connection curves in the picture.

PROPOSITION 2.3.11 *Let* $\mathcal{S} = (\mathsf{R}, \mathsf{v}, \eta)$ *be a specification and let* α *be a realization of* $\mathcal{S}$. *There is a one-to-one correspondence* π *between the paths in* $\mathcal{S}$ *and the connection curves in* $\alpha(\mathsf{R})$ *such that, for a path* $\mathcal{P} = \langle r_0, r_1, \ldots, r_n \rangle$, $\alpha(\mathcal{P})$ *is the curve resulting from the composition of the curves* $C(r_i, r_{i+1})$ *for* $i = 0, 1, \ldots, n-1$.

DEFINITION 2.3.12 (CONNECTION CURVE FOR A PATH) *With* $\mathcal{S}$, α *and* π *as in Proposition 2.3.11, if* $\mathcal{P}$ *is a path then* $\pi(\mathcal{P})$ *is said to be the* connection curve *of* $\mathcal{P}$.

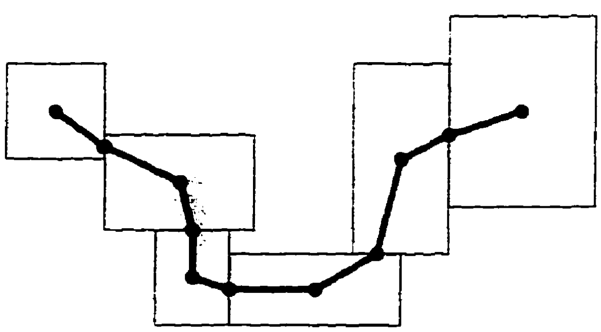


Figure 2.3: Connection curve for a path.

Figure 2.3 shows the connection curve for a path. Now we are going to examine the characteristics of connection curves. I present the following propositions. The first proposition states that a connection curve fits in the rectangles that correspond to the path. The second proposition states that a connection curve cannot intersect itself.

PROPOSITION 2.3.13 *Suppose one has a realization* ψ *of a specification* $\mathcal{S} = (\mathsf{R}, \mathsf{v}, \eta)$. *Let* $\mathcal{P} = \langle r_0, r_1, \ldots, r_n \rangle$ *be a path in* $\mathcal{S}$. *Then the connection curve* $\pi(\mathcal{P})$ *of* $\mathcal{P}$ *is contained in* $\bigcup_{i=0}^{n} \alpha(r_i)$.

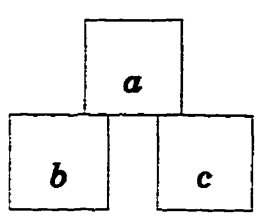
Proof: Let us consider the connection curve for any two adjacent rectangle symbols a and b in the path. Let $a' = \alpha(a)$ and $b' = \alpha(b)$. The connection curve is defined as the two straight line segments L_1 from $C(a')$ to $M(a', b')$, and L_2 from $M(a', b')$ to $C(b')$. By definition, $C(a'), M(a', b') \in a'$ and $M(a', b'), C(b') \in b'$. A straight line that connects any two points inside a rectangle is itself contained entirely within the rectangle, therefore $L_1 \in a'$ and $L_2 \in b'$. Since this is true for any two adjacent rectangle symbols, it is true for the entire path. □

PROPOSITION 2.3.14 *Suppose one has a realizable specification* $\mathcal{S}$. *The connection curve of a path is closed if and only if the path is closed. The connection curve of a path does not intersect itself.*

Proof: The connection curve of a path is contained entirely within the geometric rectangles that correspond the path. For a connection curve to intersect itself, the path must "cross itself" by having a rectangle symbol being used more than once in the path (recall that corner kissing has been eliminated). This is not allowed in the definition of a path. □

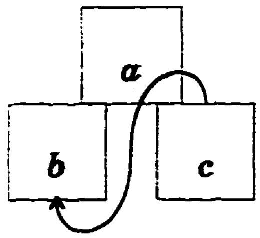
2.4 Examples

Figures 2.4 and 2.5 give simple examples of a realizable and an unrealizable specification.



$$\mathsf{v} = \{(a, b), (a, c)\}$$
$$\eta = \emptyset$$

Figure 2.4: A realizable specification.



$$\nu = \{(a,b),(a,c),\underline{(b,c)}\}$$
$$\eta = \emptyset$$

Figure 2.5: An unrealizable specification.

2.5 Approaches not taken

2.5.1 Graph drawing approach

One possible alternative is to use directed graphs to represent spatial relations. Vertices in the graph represent rectangles, and the edges represent the spatial relationships. If the spatial information can be correctly represented, then realizability testing would be equivalent to the straight-line planarity testing of the directed graph.

However, it seems that the spatial relations ν and η cannot be directly replaced by edges, as there are no restrictions to the orientations of edges in a general directed graph. One approach to constrain the orientations of edges is the concept of *upward* graph drawings. A drawing of a directed graph is straight-line upward if every edge is a straight line and is monotonically nondecreasing in the y-direction. If this approach is to be adopted, two graphs would be required — one for ν and one for η. As necessary conditions for realizability, each graph must admit a straight-line upward drawing. Unfortunately this is not sufficient for realizability as I shall demonstrate in a later chapter that the realizability problem cannot be solved by treating ν and η separately.

CHAPTER 3

EARLY FINDINGS

In this chapter, I present some early findings of the characteristics of specifications. We begin by showing the properties of the geometric and abstract compass relations. Then I show that the abstract compass relations are insufficient to specify unique realizations. I introduce equivalence classes to group similar realizations. Following that, several examples are presented to illustrate how the abstract relations are used to specify pictures. In doing so, some of the limitations of the system is discovered. In the final section, we examine an interesting construct called a "cage".

3.1 Properties of the geometric compass relations

Let us examine the properties of the geometric compass relations $\bar{\nu}$ and $\bar{\eta}$.

PROPOSITION 3.1.1 *$\bar{\nu}$ and $\bar{\eta}$ have the following properties:*

1. *There are no $\bar{\nu}$-cycles and no $\bar{\eta}$-cycles;*
2. *$\bar{\nu}$ and $\bar{\eta}$ are anti-transitive;*
3. *$\bar{\nu}$, $\bar{\nu}^{-1}$, $\bar{\eta}$, and $\bar{\eta}^{-1}$ are pairwise disjoint.*

Proof:

1. (No cycles) Let us assume, on the contrary, that there is a $\bar{\nu}$-cycle such that $(r_0, r_1), (r_1, r_2), \ldots, (r_{n-1}, r_n) \in \bar{\nu}$, $r_0 = r_n$, for some $n \geq 1$. Then one has $Y_b(r_0) = Y_t(r_1) > Y_b(r_1) = Y_t(r_2) > \cdots > Y_b(r_{n-1}) = Y_t(r_n) > Y_b(r_n) = Y_b(r_0)$, which is a contradiction. The proof for $\bar{\eta}$-cycles is analogous.
2. (Anti-transitivity) Suppose on the contrary that one has $(r_1, r_2), (r_2, r_3), (r_1, r_3) \in \bar{\nu}$. Then one has $Y_b(r_1) = Y_t(r_2)$, $Y_b(r_2) = Y_t(r_3)$, and $Y_b(r_1) = Y_t(r_3)$.

Hence $Y_t(r_2) = Y_b(r_2)$. This means that rectangle r_2 has zero height. Since degenerate rectangles are not allowed, one arrives at a contradiction. The case for $\bar{\eta}$ is similar.

3. (Disjointness) If there are no $\bar{\nu}$-cycles and no $\bar{\eta}$-cycles, then $\bar{\nu}$ and $\bar{\eta}$ are anti-symmetric. Therefore $\bar{\nu} \cap \bar{\nu}^{-1} = \bar{\eta} \cap \bar{\eta}^{-1} = \emptyset$. Let us consider the case for $\delta \cap \gamma$, for $(\delta, \gamma) = \{(\bar{\nu}, \bar{\eta}), (\bar{\nu}, \bar{\eta}^{-1}), (\bar{\nu}^{-1}, \bar{\eta}), (\bar{\nu}^{-1}, \bar{\eta}^{-1})\}$. Let $(r_1, r_2) \in \delta \cap \gamma$. If $(r_1, r_2) \in \delta$ then one has $Y_b(r_1) = Y_t(r_2)$ or $Y_t(r_1) = Y_b(r_2)$. This means that
$$\Big((Y_b(r_1), Y_t(r_1)\Big) \cap \Big(Y_b(r_2), Y_t(r_2)\Big) = \emptyset. \tag{3.1}$$
But if $(r_1, r_2) \in \delta$, then one has $\Big((Y_b(r_1), Y_t(r_1)\Big) \cap \Big(Y_b(r_2), Y_t(r_2)\Big) \neq \emptyset$. This contradicts equation 3.1. Hence $\delta \cap \gamma = \emptyset$.

□

PROPOSITION 3.1.2 *For rectangles r_1 and r_2 in a picture, if*
$$[X_l(r_1), X_r(r_1)] \cap [X_l(r_2), X_r(r_2)] = \emptyset$$
or
$$[Y_b(r_1), Y_t(r_1)] \cap [Y_b(r_2), Y_t(r_2)] = \emptyset$$
then
$$(r_1, r_2) \notin \bar{\nu} \cup \bar{\nu}^{-1} \cup \bar{\eta} \cup \bar{\eta}^{-1}.$$

Proof: If $[X_l(r_1), X_r(r_1)] \cap [X_l(r_2), X_r(r_2)] = \emptyset$ then $X_l(r_1) \neq X_r(r_2)$ and $X_l(r_2) \neq X_r(r_1)$. Thus $(r_1, r_2) \notin \bar{\nu} \cup \bar{\nu}^{-1} \cup \bar{\eta} \cup \bar{\eta}^{-1}$.

If $[Y_b(r_1), Y_t(r_1)] \cap [Y_b(r_2), Y_t(r_2)] = \emptyset$ then $Y_b(r_1) \neq Y_t(r_2)$ and $Y_b(r_2) \neq Y_t(r_2)$. Thus $(r_1, r_2) \notin \bar{\nu} \cup \bar{\nu}^{-1} \cup \bar{\eta} \cup \bar{\eta}^{-1}$. □

COROLLARY 3.1.3 *Suppose a specification $\mathcal{S} = (\mathrm{R}, \nu, \eta)$ is realizable, then ν and η have the following properties:*

1. *There are no ν-cycles and no η-cycles;*

2. *ν and η are anti-transitive;*

3. *ν, ν^{-1}, η, and η^{-1} are pairwise disjoint.*

Proof: Let $\psi = \alpha(\mathrm{R})$ be a realization. Then $\nu = \bar{\nu}_\psi$ and $\eta = \bar{\eta}_\psi$. It is proven in Proposition 3.1.1 that for any picture ψ, $\bar{\nu}_\psi$ and $\bar{\eta}_\psi$ have all the properties listed above. □

3.2 Equivalence of realizations

Specifications do not stipulate the sizes of rectangles in the realizations. If a specification is realizable, it can be mapped by different arrangements to an infinite number of different pictures. We want to be able to classify them and group them into classes. We will deal with the concept of equivalence informally in this section and present a formal definition in Chapter 6.

Suppose one has a simple specification given by $\left\{\{a,b\},\{(a,b)\},\emptyset\right\}$. All the realizations shown in Figure 3.1 are to be considered as equivalent.

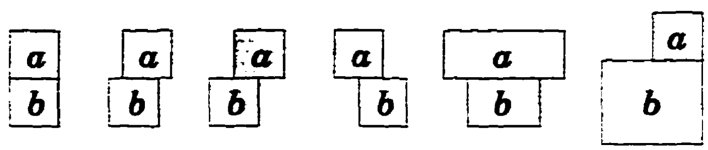


Figure 3.1: Realizations that are equivalent.

Suppose one has another specification given by $\left\{\{a,b,c\},\{(a,b),(a,c)\},\emptyset\right\}$. Realizations shown in Figure 3.2 are to be considered equivalent. However, the realizations shown in Figure 3.3 are to be distinguished.

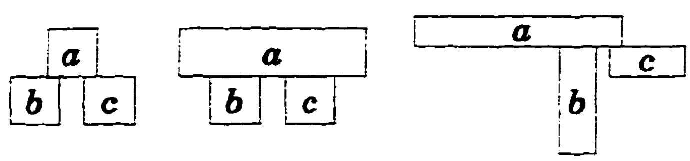


Figure 3.2: Realizations that are equivalent.

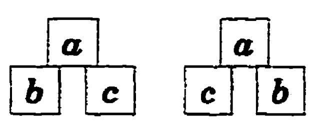


Figure 3.3: Realizations that are not equivalent.

If only specifications with $\eta = \emptyset$ are considered, Figure 3.4 lists all the equivalence classes for specifications with one to five rectangles. Each entry may represent more than one class, as classes that are horizontal/vertical mirrors are not shown. The total number of classes contained in each entry is shown in the second column.

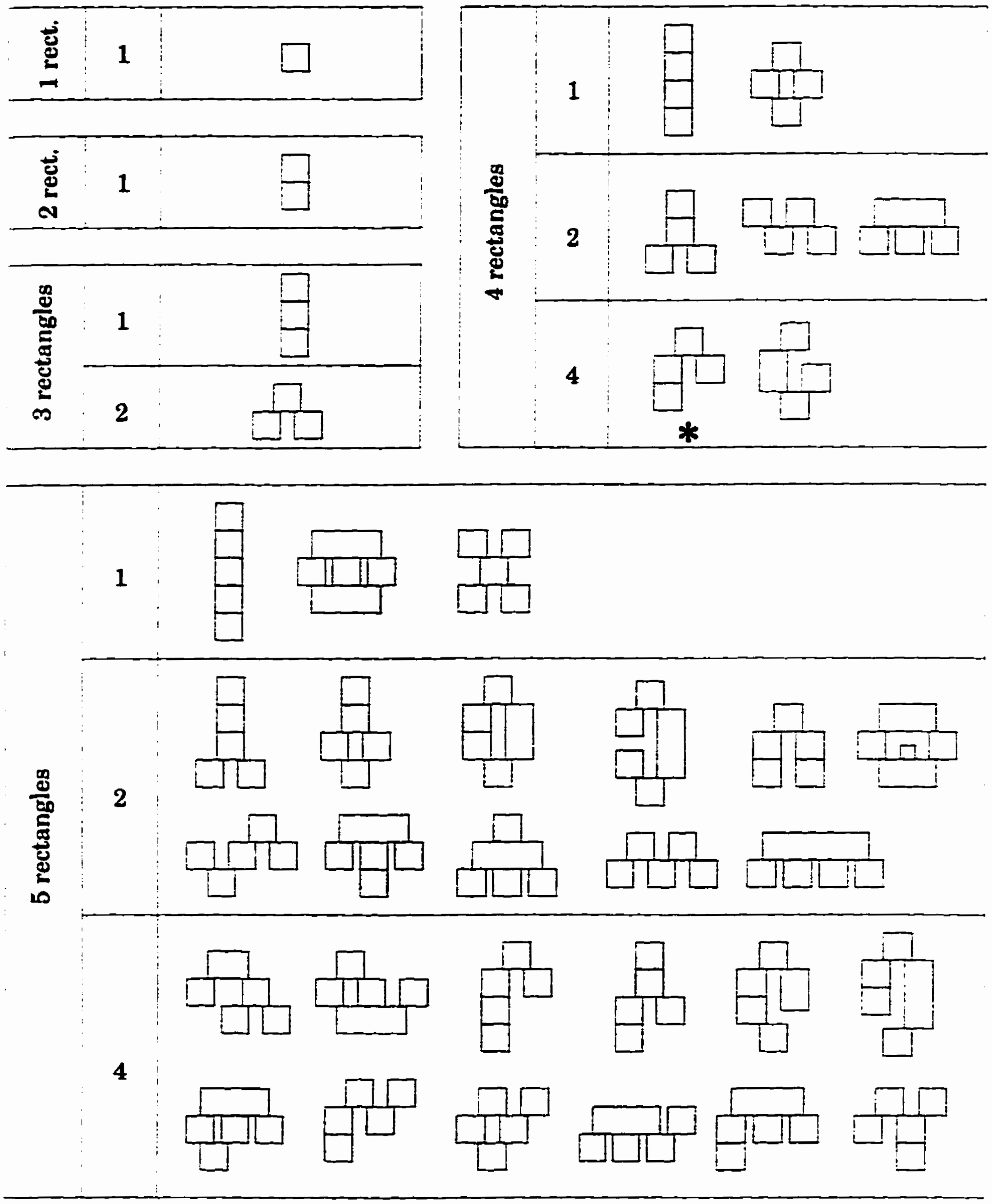


Figure 3.4: Equivalence classes.

Let us look at an example to clarify the presentation of Figure 3.4. Consider the entry labeled with the $*$. The entry represents a total of four classes, shown in Figure 3.5.

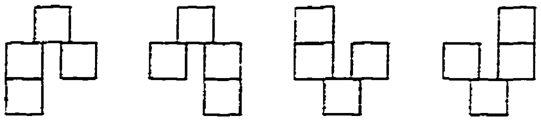


Figure 3.5: The four classes represented by the entry labeled with the $*$ in Figure 3.4.

That is to say, whenever you have three connected rectangles, their relative positions must fall into one of the two general classes. Similarly for four rectangles, their relative positions must fall into one of the seven classes.

As expected, the number of classes increases dramatically with the number of connected rectangles. There also does not seem to be a pattern that can be extracted.

As mentioned before, these classes cannot be specified formally at this point as some important concepts are still missing. The notion of equivalence will be re-visited again in Chapter 6. There a formal definition will be given.

3.3 Examples

Let us look at some examples of specifications and examine their realizability. In the first example, the specification is realizable and the possible realization classes are shown. The specification shown in the second example is not realizable, and a proof of this is presented.

EXAMPLE 3.3.1 *Let*

$$\begin{aligned} \mathsf{R} &= \{a,b,c,d,e,f,g\}, \\ \mathsf{v} &= \{(b,a),(c,a),(d,b),(d,c),(e,a),(f,a),(g,e),(g,f)\}, \\ \eta &= \emptyset. \end{aligned}$$

We can see that there are three classes of possible realizations (remember that arbitrary scaling and vertical mirroring are allowed). They are shown in Figure 3.6. The picture shown in Figure 3.7 is not acceptable because rectangles are not allowed to overlap.

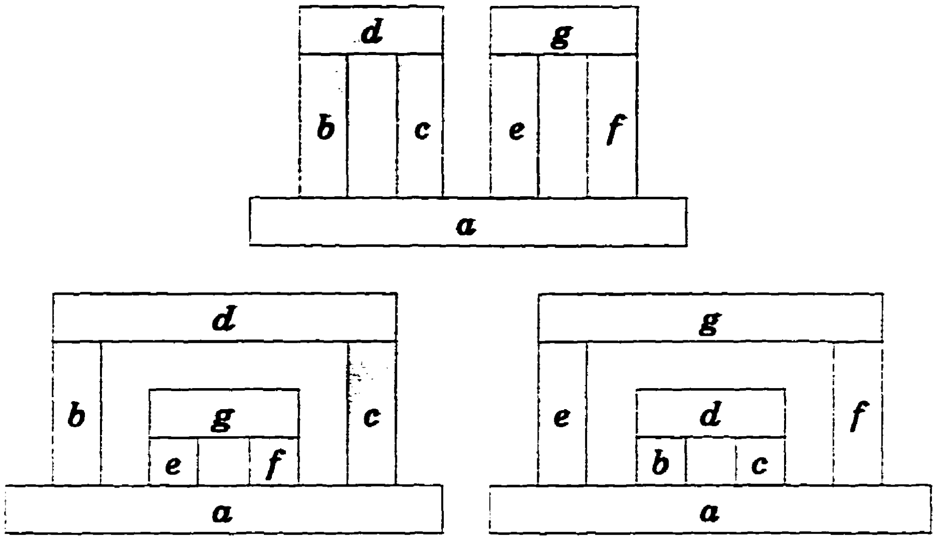


Figure 3.6: Possible realization classes.

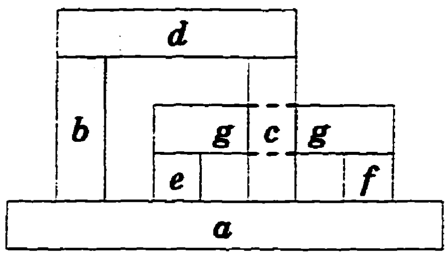


Figure 3.7: Overlaps are not allowed.

EXAMPLE 3.3.2 *Let us now look at an example that cannot be realized. Let*

$$
\begin{aligned}
\mathsf{R} &= \{a,b,c,d,e,f,g,h,i,j\}, \\
\mathsf{v} &= \{(a,h),(b,a),(c,a),(d,a),(e,b),(e,c),(f,c),(f,d),(g,e),(g,f),(i,h),(j,i), \\
&\quad (j,c)\}, \textit{ and} \\
\eta &= \emptyset.
\end{aligned}
$$

The rectangle symbol c is enclosed in a "cage" formed by the closed path $\langle a,b,e,g,f,d,a\rangle$. It is not possible for c to be adjacent to j which is outside of the "cage". This situation is shown in Figure 3.8.

Note that the cage used in this example is constructed deliberately in a careful manner so that one can assert that c is inside and j is outside. The following proves that this example is not realizable. Since the tools for dealing with the inside and outside of cages have not yet been developed, the proof will not use any properties of cages.

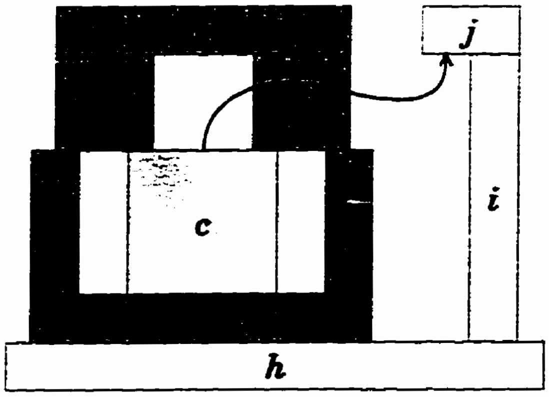


Figure 3.8: No realization is possible.

Proof: Let us assume that there is a realization. From v one knows that

$$Y_t(b) = Y_b(e) = Y_t(c) = Y_b(f) = Y_t(d) = Y_b(j).$$

For $r_1, r_2 \in \{e, f, j\}, r_1 \neq r_2$, one has, by Definition 2.2.5,

$$\Big(X_l(r_1), X_r(r_1)\Big) \cap \Big(X_l(r_2), X_r(r_2)\Big) = \emptyset. \tag{3.2}$$

Similarly for $r_3, r_4 \in \{b, c, d\}, r_3 \neq r_4$, one has, by Definition 2.2.5,

$$\Big(X_l(r_3), X_r(r_3)\Big) \cap \Big(X_l(r_4), X_r(r_4)\Big) = \emptyset. \tag{3.3}$$

I have established that the horizontal extents of rectangles e, f, and j, as well the horizontal extents of rectangles b, c, and d do not overlap.

Looking at rectangles b, c, and d, there are six permutations to order them from West to East:

1. $X_c(b) < X_c(c) < X_c(d)$ implies $X_l(b) < X_r(b) < X_l(c) < X_r(c) < X_l(d) < X_r(d)$.
2. $X_c(d) < X_c(c) < X_c(b)$ implies $X_l(d) < X_r(d) < X_l(c) < X_r(c) < X_l(b) < X_r(b)$.
3. $X_c(b) < X_c(d) < X_c(c)$ implies $X_l(b) < X_r(b) < X_l(d) < X_r(d) < X_l(c) < X_r(c)$.
4. $X_c(c) < X_c(d) < X_c(b)$ implies $X_l(c) < X_r(c) < X_l(d) < X_r(d) < X_l(b) < X_r(b)$.
5. $X_c(d) < X_c(b) < X_c(c)$ implies $X_l(d) < X_r(d) < X_l(b) < X_r(b) < X_l(c) < X_r(c)$.
6. $X_c(c) < X_c(b) < X_c(d)$ implies $X_l(c) < X_r(c) < X_l(b) < X_r(b) < X_l(d) < X_r(d)$.

One has $(e,b),(f,d),(e,c),(f,c) \in \mathrm{v}$. Cases 3 to 6 can be eliminated because they force rectangles e and f to overlap. I will show case 3 in detail and omit cases 4 to 6 as the proofs are similar.

In case 3, $X_l(e) < X_r(b)$ because of $(e,b) \in \mathrm{v}$; moreover, $X_r(b) < X_l(c)$ as above, thus $X_l(e) < X_l(c)$. By $(f,d),(f,c) \in \mathrm{v}$ and the inequality above one has $X_l(f) < X_r(d) < X_l(c) < X_r(f)$. Thus by $(e,c) \in \mathrm{v}$ $X_l(e) < X_l(c) < X_r(e)$ and $X_l(f) < X_l(c) < X_r(f)$, hence $\left(X_l(e),X_r(e)\right) \cap \left(X_l(f),X_r(f)\right) \neq \emptyset$. However, this contradicts equation 3.2.

Let us now look at the two remaining possible cases.

1. $X_c(b) < X_c(c) < X_c(d)$.

   Since $(e,b),(f,d) \in \mathrm{v}$, $X_c(e) < X_c(f)$. Looking at rectangles e, f, and i, one has three permutations to order them.

   (a) $X_c(j) < X_c(e) < X_c(f)$ implies $X_l(j) < X_r(j) < X_l(e) < X_r(e) < X_l(f) < X_r(f)$.

   (b) $X_c(e) < X_c(j) < X_c(f)$ implies $X_l(e) < X_r(e) < X_l(j) < X_r(j) < X_l(f) < X_r(f)$.

   (c) $X_c(e) < X_c(f) < X_c(j)$ implies $X_l(e) < X_r(e) < X_l(f) < X_r(f) < X_l(j) < X_r(j)$.

   Since $(e,b),(e,c),(f,c),(f,d) \in \mathrm{v}$ one has $X_l(e) < X_l(c) < X_r(e)$ and $X_l(f) < X_r(c) < X_r(f)$.

   For cases (a) and (c) one has $(X_r(j) < X_l(e) < X_l(c)$ or $X_r(c) < X_r(f) < X_l(j)$, and therefore $\left(X_l(j),X_r(j)\right) \cap \left(X_l(c),X_r(c)\right) = \emptyset$. Thus $(j,c) \notin \mathrm{v}$. But this contradicts the fact that $(j,c) \in \mathrm{v}$. Therefore cases (a) and (c) can be eliminated.

   For case (b), one knows that $X_l(c) < X_r(e) < X_l(j) < X_r(j) < X_l(f) < X_r(c)$. Since $(j,i) \in \mathrm{v}$, one has $\left(X_l(j),X_r(j)\right) \cap \left(X_l(i),X_r(i)\right) \neq \emptyset$. Hence $\left(X_l(c),X_r(c)\right) \cap \left(X_l(i),X_r(i)\right) \neq \emptyset$. Since $(j,c),(j,i) \in \mathrm{v}$, one has $\left(Y_b(i),Y_t(i)\right) \cap \left(Y_b(c),Y_t(c)\right) \neq \emptyset$. Thus rectangles i and c overlap. Therefore we have arrived at a contradiction.

2. $X_c(d) < X_c(c) < X_c(b)$.

   The proof of the impossibility of this case is similar. It can be obtained by simply exchanging e and f, as well as b and d in the proof of part 1.

Hence I have shown that no realization exists for the specification. □

The proof above by case distinction is tedious and specific. It cannot be easily generalized to other situations. This is one of the motivations for attempting to characterize the cage situation in subsequent sections of this thesis.

It seems rather simple to identify a cage in a realization and to decide which rectangle are inside and which are outside. But in fact, if we only look at the abstract relations ν and η, this turns out to be a surprisingly difficult task.

3.4 Cages

Let us take a closer look at cages. Consider a specification $\mathcal{S} = (\mathsf{R}, \nu, \eta)$. Suppose, $\mathcal{S}$ has a closed path $\mathring{\mathcal{P}}$, called a cage in the sequel. If there is a rectangle symbol r not occurring in $\mathring{\mathcal{P}}$ such that, in any realization α of $\mathring{\mathcal{P}}$, $\alpha(r)$ would have to be attached to a rectangle inside the cage and also to a rectangle outside the cage then, clearly, $\mathring{\mathcal{P}}$ is not realizable. It seems natural, therefore, to develop the abstract notions of *inside* and *outside* for cages in specifications. This turns out to be surprisingly difficult.

DEFINITION 3.4.1 *Let $\mathcal{S} = (\mathsf{R}, \nu, \eta)$ be a realizable specification and suppose there is a closed path $\mathring{\mathcal{P}} = \langle r_0, \ldots, r_n, r_0 \rangle$ in $\mathcal{S}$. A rectangle symbol $r \in \mathsf{R}$ not occurring in $\mathring{\mathcal{P}}$ is* inside *$\mathring{\mathcal{P}}$ if, for every arrangement α of $\mathcal{S}$, $\alpha(r)$ is contained in the area $\mathcal{A}_\alpha(\mathring{\mathcal{P}})$ enclosed by the connection curve of $\alpha(\mathring{\mathcal{P}})$. The rectangle symbol $r \in \mathsf{R}$ is outside $\mathring{\mathcal{P}}$ if, for every arrangement α of $\mathcal{S}$, the interior of $\alpha(r)$ is contained in the area*

$$\mathbb{R}^2 \setminus \left(\bigcup_{i=0}^{n} \alpha(r_i) \cup \mathcal{A}_\alpha(\mathring{\mathcal{P}}) \right).$$

In this section, I will present a series of attempts to capture the concept of inside and outside of cages. We restrict our discussion here to specifications with $\eta = \emptyset$ to reduce the complexity of the cases to be examined. I begin with a very simple form of a cage, called an LR cage.

3.4.1 LR cages

DEFINITION 3.4.2 (LR CAGE) *Whenever one has two paths*

$$\mathcal{P}_s = \langle r_{\mathrm{bot}} = s_0, s_1, \ldots, s_m = r_{\mathrm{top}} \rangle$$

and

$$\mathcal{P}_t = \langle r_{\mathrm{bot}} = t_0, t_1, \ldots, t_n = r_{\mathrm{top}} \rangle$$

where

$$s_i \neq t_j,\ 0 < i < m,\ 0 < j < n$$

and

$$(s_i, s_{i-1}), (t_j, t_{j-1}) \in \mathrm{v},\ \ 1 \leq i \leq m,\ 1 \leq j \leq n$$

one has an LR cage $\dot{\mathcal{P}} = \langle r_{\text{bot}} = s_0 = t_0, s_1, \ldots, s_m = r_{\text{top}} = t_n, t_{n-1}, \ldots, t_1, t_0 = r_{\text{bot}} = s_0 \rangle$. r_{bot} *and* r_{top} *are called the* bottom *and the* top *of the cage, respectively. The remaining rectangle symbols* $\{s_1, \ldots, s_{m-1}\}$ *and* $\{t_1, \ldots, t_{n-1}\}$ *compose the two* walls *of the cage.*

Note that at the abstract level, it is not possible to identify the left and right walls. One knows that $(s_1, r_{\text{bot}}), (t_1, r_{\text{bot}}) \in \mathrm{v}$ but there is no information available to order s_1 and t_1. The same can be said for s_{m-1}, t_{n-1}, and r_{top}. The notion of left and right walls does exist in the realizations though, after a specific arrangement α is employed. We now present several attempts at abstracting the notions of inside and outside from some scenarios in which these look intuitively obvious.

ATTEMPT 3.4.3 *An LR cage is empty if (a) there are no rectangle symbols* r_1 *other than* s_1 *and* t_1 *such that* $(r_i, r_{\text{bot}}) \in \mathrm{v}$*; and (b) there are no rectangle symbols* r_2 *other than* s_{m-1} *and* t_{n-1} *such that* $(r_{\text{top}}, r_j) \in \mathrm{v}$.

Disproof: The only remaining candidates that can be inside the cage must be adjacent to the side walls with respect to v. But they can always be "flipped out" in the realizations. □

Note that the converse of the above statement is also not true. Even if the rectangle symbols r_1 and r_2 do exist, they can again be "flipped out" of the cage in the realizations.

ATTEMPT 3.4.4 *A rectangle symbol is inside an LR cage if it is connected to both walls of the cage.*

Disproof: A counter-example is shown in Figure 3.9. The two realizations are constructed from

$$\mathrm{v} = \{(i, g), (g, b), (g, e), (h, f), (h, c), (b, a), (c, a), (e, d), (f, d).$$

Rectangle d can be inside or outside the cage. □

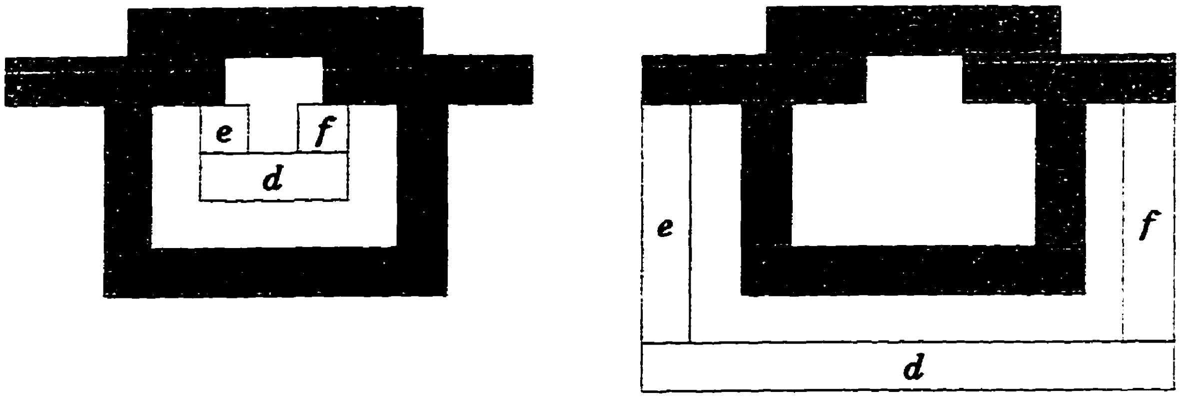


Figure 3.9: Counter-example for Observation 3.4.4. Rectangle d can be inside or outside the cage.

ATTEMPT 3.4.5 *A rectangle symbol is inside an LR cage if it is connected to both walls of the cage and also connected to either the top or the bottom of the cage.*

Disproof: A counter-example is shown in Figure 3.10. □

Let us add more conditions to attempt to "force" rectangle symbols to be inside cages.

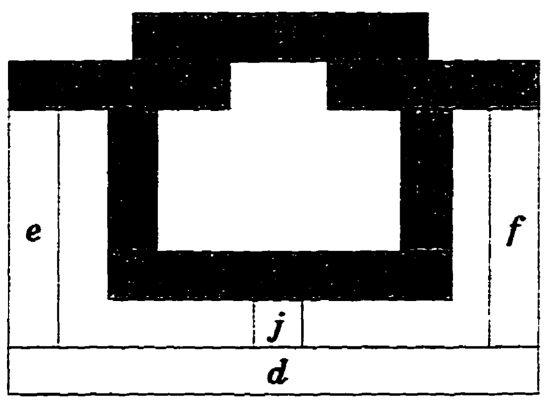


Figure 3.10: Counter-example for Observation 3.4.5. d is not in the LR cage.

ATTEMPT 3.4.6 *A rectangle symbol inside in an LR cage if it is connected to both walls of the cage and also connected to either the top from the South or connected to the bottom from the North.*

At first glance, there seem to be enough conditions to prevent rectangle symbols to be flipped out of a cage. An example is shown in Figure 3.11. But yet again, a counter-example can be found. It is shown in Figure 3.12. The problem is that the "connectedness" condition provides too much freedom. We will have to use the more restrictive adjacency condition in our next attempt.

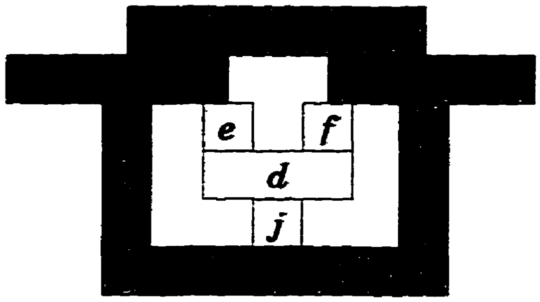


Figure 3.11: Observation 3.4.6: is d is in the LR cage?

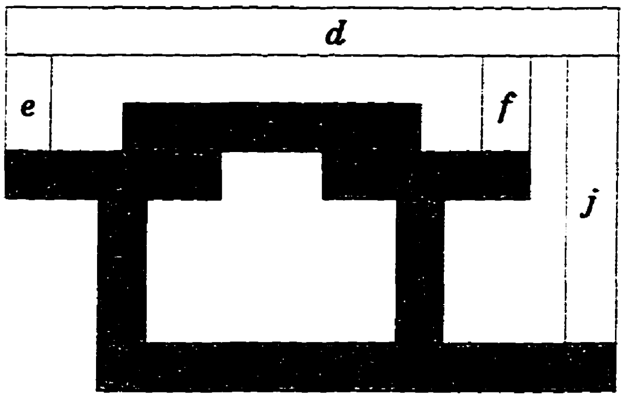


Figure 3.12: Counter-example for Observation 3.4.6.

ATTEMPT 3.4.7 *A rectangle symbol is inside an LR cage if it is adjacent to a rectangle symbol from each of the two walls.*

We have finally collected enough conditions that is sufficient, but by no means necessary, to state that a rectangle symbol is in an LR cage. This is used as a basis to construct Example 3.3.2.

Proof: Let the LR cage be $\mathring{\mathcal{P}} = \langle r_0 = r_{\text{bot}}, r_1, \ldots, r_n = r_{\text{top}}, r_{n+1}, \ldots, r_{n+m} = r_{\text{bot}} \rangle$. Let r be a rectangle symbol adjacent to r_i and r_j with $0 < i < m$ and $0 < j < n$.

Let α be an arrangement of the specification. Then

$$X_c(r_i) < X_r(r_i) = X_l(r) < X_r(r) = X_l(r_j) < X_c(r_j)$$

or

$$X_c(r_j) < X_r(r_j) = X_l(r) < X_r(r) = X_l(r_i) < X_c(r_i).$$

Thus $\pi(\mathring{\mathcal{P}}) \notin \alpha(r)$. Without loss of generality assume the former. Suppose $\alpha(r)$ is not contained in $\mathcal{A}_\alpha(\mathring{\mathcal{P}})$. Hence $\pi(\mathring{\mathcal{P}})$ intersects the connection curve $\pi(\mathring{\mathcal{P}})$ of $\mathring{\mathcal{P}}$. Hence there is a rectangle symbol r_i such that $C(r_i, r_{(i+1)\bmod(n+m)})$ intersects $\alpha(r)$. By Proposition 2.3.13 $C(r_i, r_{(i+1)\bmod(n+m)})$ is contained in $\alpha(r_i) \cup \alpha(r_{i+1})$. As $\alpha(r)$ does not overlap with $\alpha(r_i)$ and $\alpha(r_{i+1})$ it follows that $\mathcal{M}(r_i, r_{i+1}) \in \alpha(r)$. However, this again implies an overlap. □

The series of attempts presented shows that it is not easy to derive properties for rectangle pictures, even for very simple cases. The apparently simple notion of inside and outside cannot be defined easily, even in an extremely simplified cage situation.

3.4.2 General cages

We now turn our attention to other cage situations.

DEFINITION 3.4.8 (GENERAL CAGE) *A closed path* $\mathring{\mathcal{P}} = \langle p_0, p_1, \ldots, p_n, p_0\rangle$ *forms a* cage.

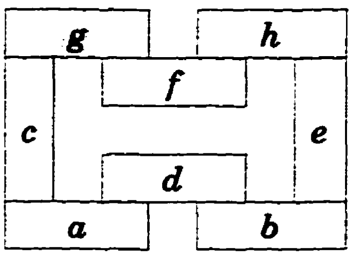


Figure 3.13: Another cage situation. Here, *a*, *b*, and *f* are local minima, *g*, *h* and *d* are local maxima.

Given the difficulties we had with determining the inside and outside of an LR cage there is little hope for a simple characterization of the inside and outside of a general cage. In the next section I present another sequence of attempts to generalize the idea of Attempt 3.4.7, along with their counter-examples.

3.4.3 Local maxima and minima

In the general cage the roles of the rectangle symbols r_{top} and r_{bot} of an LR cage seem to be taken over by "local minima" and "local maxima". Intuitively, a rectangle symbol is a local minimum if no rectangle symbols is forced to be the South of it in every realization.

DEFINITION 3.4.9 *Suppose one has a closed path* $\dot{\mathfrak{P}} = \langle p_0, p_1, \ldots, p_m, p_0 \rangle$.

1. *A rectangle symbol* p_i *is a* local minimum *of* $\dot{\mathfrak{P}}$ *if* $(p_i, p_j) \notin \mathsf{v}$ *for all* $0 \leq j \leq m, i \neq j$. *Let* R_N *denote the set of all local minima.*

2. *A rectangle symbol* p_i *is a* local maximum *of* $\dot{\mathfrak{P}}$ *if* $(p_j, p_i) \notin \mathsf{v}$ *for all* $0 \leq j \leq m, i \neq j$. *Let* R_X *denote the set of all local maxima.*

3. *A rectangle symbol* p_i *is a* side wall *of* $\dot{\mathfrak{P}}$ *otherwise. Let* R_W *denote the set of all side wall rectangle symbols.*

4. *Let* $p_i \in R_N \cup R_X$ *and let* $p_k, p_l \in \dot{\mathfrak{P}}$. p_k *and* p_l *are said to be* on opposite sides *of* p_i *if there is a* $p_j \in R_N \cup R_X$, $j \neq i$, *such that one of the following conditions is satisfied*

   (a) *If* $j < i$ *then* $k \in \{j+1, \ldots, i-1\}$ *and* $l \in \{0, \ldots, j-1, i+1, \ldots, n\}$ *or vice versa.*

   (b) *If* $i < j$ *then* $k \in \{i+1, \ldots, j-1\}$ *and* $l \in \{0, \ldots, i-1, j+1, \ldots, n\}$ *or vice versa.*

ATTEMPT 3.4.10 *A rectangle symbol is inside a cage if it is connected to a local maximum or a local minimum of the cage, and to two other side wall rectangle symbols on opposite sides.*

Disproof: A counter-example is given in Figure 3.12. d is connected to the local minimum a and side wall rectangle symbols g and h. □

ATTEMPT 3.4.11 *A rectangle symbol* r *is inside a cage if there exists a rectangle symbol* $r_x \in R_X$ *such that* $(r_x, r) \in \mathsf{v}$ *or if there exists a rectangle symbol* $r_n \in R_N$ *such that* $(r, r_n) \in \mathsf{v}$.

Disproof: A counter-example is shown in Figure 3.14. □

ATTEMPT 3.4.12 *A rectangle symbol* r *is inside a cage if there exists a rectangle symbol* $r_x \in R_X$ *and a rectangle symbol* $r_n \in R_N$ *such that* $(r, r_x), (r_n, r) \in \mathsf{v}$.

Disproof: An example of this is shown in Figure 3.15. However this attempt is not true either as shown in Figure 3.16. □

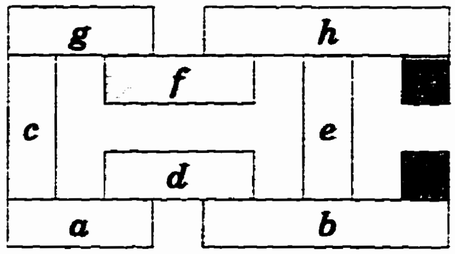


Figure 3.14: Counter example to Observation 3.4.11.

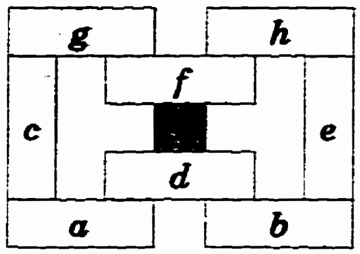


Figure 3.15: Inside a cage.

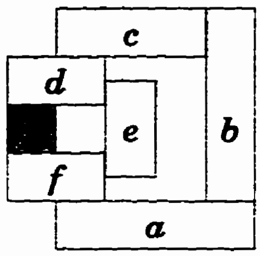


Figure 3.16: Outside a cage.

3.5 Need to reduce the problem

It is shown that even seamingly simple ideas cannot be expressed easily, even in specifications involving only one of the two abstract compass relations. Recall that the motivation to define the inside and outside of a closed path was to enable us to identify certain specifications that are not realizable.

CHAPTER 4

EXAMPLES

We would like to derive rules to determine the realizability of specifications in the following chapters. It is difficult to do this by just looking at the abstract relations. In this chapter, we will look at geometric examples to gather clues on how to formulate conditions for realizability testing.

4.1 Specifications with $\eta = \emptyset$

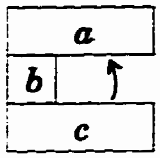
We begin with simple specifications that involve ν only. The following lists some specific situations that may cause realizability problems.

1. Degenerate dimensions (Figure 4.1(a)).
   For this case to be realizable, rectangle b has to have zero height. This is characterized by a transitive relationship in ν.

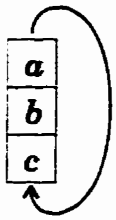
2. Impossible stretching (Figure 4.1(b)).
   It is not possible for c to be to the North of a. This is characterized by a ν-cycle.

3. Forced overlapping (Figure 4.1(c)).
   Rectangles a and b act as obstructions to prevent x from touching y. However, in the absence of information ordering the positions of a and y, their positions can be swapped without any change in ν. Then it would be possible to obtain a realization.

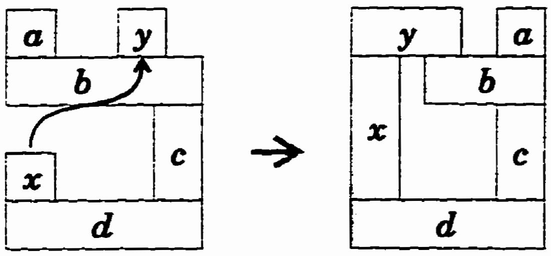
4. Inside/outside (Figure 4.1(d)).
   Consider the closed path $\mathring{\mathcal{P}} = \langle a, b, c, d, a \rangle$. If x is inside and y is outside of the cage, then x cannot be adjacent to y without overlapping some part of

(a) 

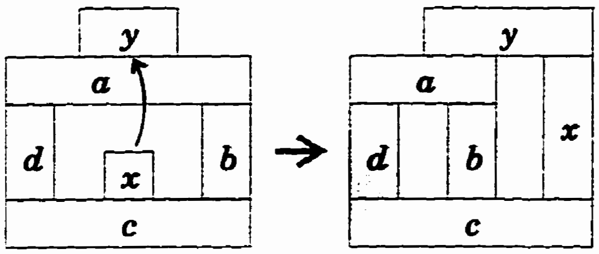
$\nu = \{(a.b).(b.c).\underline{(a.c)}\}$

(b) 

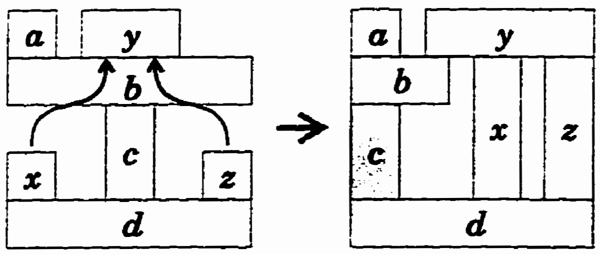
$\nu = \{(a.b).(b.c).\underline{(c.a)}\}$

(c) 

$\nu = \{(a.b).(y.b).(b.c).$
$(c.d).(x.d).\underline{(y.x)}\}$

(d) 

$\nu = \{(y.a).(a.b).(a.d).$
$(b.c).(d.c).(x.c)$
$\underline{(y.x)}\}$
$\dot{\mathcal{P}} = \langle a.b.c.d.a\rangle$

(e) 

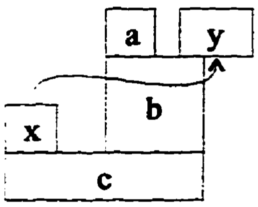
$\nu = \{(a.b).(y.b).(b.c).$
$(c.d).(x.d).(z.d).$
$\underline{(y.x)}.\underline{(y.z)}\}$

Figure 4.1: Some possible causes of unrealizability ($\eta = \emptyset$).

the cage. But again, without additional ordering information, a realization can be obtained by "flipping" x out of the cage. We have seen examples of this situation in the previous chapter.

5. Forced overlapping (Figure 4.1(e)).
   At first glance, this situation seems to be unrealizable. b acts as a barrier to prevent y from touching x and z. But again, without the ordering of x, c, and z, one can "flip" c to the leftmost position. d and y are extended to the East, and x and z are extended to the North, forming a realization.

Notice that examples 3, 4, and 5 are realizable *in the absence of additional information*. If the "order" of all rectangles is fixed (to prohibit "flipping"), then they are no longer realizable.



$$\nu = \{(a.b).(y.b).(b.c).(x.c).\underline{(y.x)}\}$$

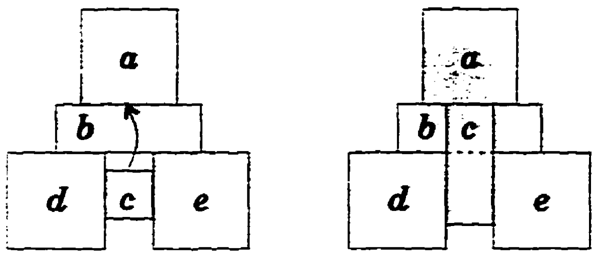
Figure 4.2: Another example of unrealizability ($\eta = \emptyset$).

4.2 Specifications with ν and η

Suppose one has the situation given in Figure 4.3. The specification is unrealizable, but this fact cannot be detected by examining ν or η alone. To illustrate this point, consider the same specification with $\eta = \emptyset$. Then it is realizable and a realization is given in Figure 4.4(a). If we instead let $\nu = \emptyset$, it is also realizable and a realization is given in Figure 4.4(b).

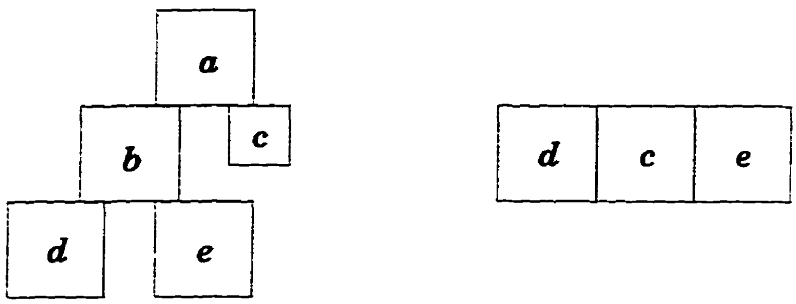
This shows the two-dimensional nature of the problem: ν and η cannot be treated separately.

Let us look at another example involving both ν and η, shown in Figure 4.5. This time, let us assume that, for each rectangle, the ordering of the rectangles adjacent to it is fixed. Therefore the only possible transformations are translation and scaling.



$$\eta = \{(e,c),(c,d)\}$$
$$\nu = \{(a,b),(b,d),(b,e),\underline{(a,c)}\}$$

Figure 4.3: Unrealizable because of overlap (ν and η).



(a) (b)

Figure 4.4: (a) $\eta = \emptyset$; (b) $\nu = \emptyset$.

Observations:

1. x cannot be adjacent to y' because of a combined-anti-transitivity formed by ν and η.

2. x cannot be adjacent to y because of Proposition 6.2.4.

3. x cannot be adjacent to z because it would have to cut across or reach around b.

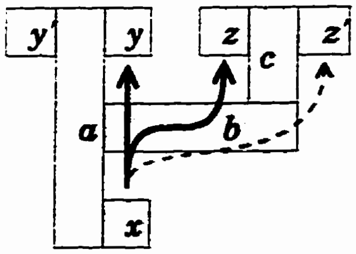
4. x cannot be adjacent to c because it would have to reach around b.

5. x can be adjacent to z'.

4.3 Big example

I now show a slightly larger example, variants of which are used in several spots later in this work. The specification is as follows:

$$\mathsf{R} = \{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t\},$$
$$\nu = \{(a,c),(a,d),(a,e),(a,f),(b,t),(c,g),(d,h),(e,i),(f,i),(g,k),(h,k),$$



$$\eta = \{(c,b),(b,a),(c,z),(y,a),(x,a)\}$$
$$\nu = \{\underline{(y,x)},\underline{(z,x)}\}$$
Ordering is fixed.

Figure 4.5: Another non-realizable example.

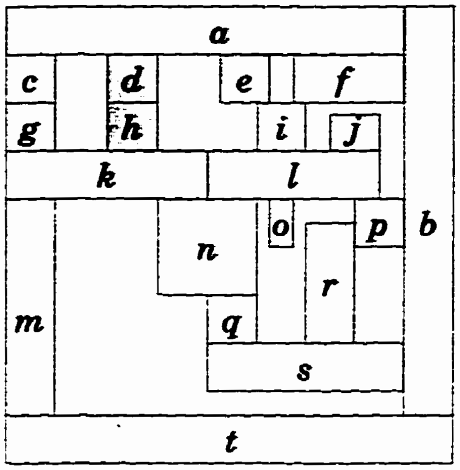


Figure 4.6: Big example.

$$(i,j),(j,l),(k,m),(k,n),(l,n),(l,o),(l,p),(m,t),(n,q),(q,s),(r,s)\}.$$
$$\eta \;=\; \{(b,a),(b,f),(b,p),(b,s),(l,k),(p,r)\}.$$

It is realizable as shown in Figure 4.6. I show this example to make the point that it is indeed very difficult to determine abstractly whether rectangles can or cannot be adjacent to each other. For instance, none of the properties derived so far indicates that $(o,m) \in \eta$ would cause the specification to be unrealizable even with q and s absent, that is, without the cage formed by b,s,q,n,l,p.

CHAPTER 5

LIMITED PROBLEMS

We have seen in the previous chapters that there are many difficulties associated with the original problem that cannot be dealt with easily. One reason is the infinite number of possibilities for the sizes and placement of positions when real values are used. In this chapter I discuss a few variants of the original problem. I am going to reduce the problem by placing certain restrictions. Instead of working with rectangles that have sizes and positions in real numbers, I consider rational and also integer sizes and positions.

5.1 Unit squares

We begin by considering a simplification of picture specifications by restricting the rectangles to have unit dimensions.

DEFINITION 5.1.1 *A* unit square *is a rectangle with unit dimensions,*

$$\begin{aligned} S_{x_0,y_0} &\stackrel{\text{def}}{=} Rect^{1,1}_{x_0,y_0} \\ &= \{(x,y) \in \mathbb{R}^2 : x_0 - 1/2 \le x \le x_0 + 1/2, y_0 - 1/2 \le y \le y_0 + 1/2\} \end{aligned}$$

Let $\mathbf{S}$ *denote the set of all unit squares. That is,*

$$\mathbf{S} \stackrel{\text{def}}{=} \{S_{x,y} : x, y \in \mathbb{R}\}$$

Further, the range of positions that the squares can be placed is also going to be limited. The following enforces a "unit grid" on which to place the squares.

DEFINITION 5.1.2 *A* unit $\mathbb{Z}$-square *is a unit square with integer-valued positions. That is,*

$$S^{\mathbb{Z}}_{i,j} \stackrel{\text{def}}{=} S_{i,j}, \quad i, j \in \mathbb{Z}$$

Let $\mathbf{S}_{\mathbb{Z}}$ denote the set of all unit $\mathbb{Z}$-squares. That is,

$$\mathbf{S}_{\mathbb{Z}} \stackrel{\text{def}}{=} \{S^{\mathbb{Z}}_{i,j} : i, j \in \mathbb{Z}\}$$

The layout of unit $\mathbb{Z}$-squares is not ambiguous because, for any pair of adjacent squares, there can only be exactly one relative position to arrange them.

REMARK 5.1.3 *Suppose $S^{\mathbb{Z}}_1$ and $S^{\mathbb{Z}}_2$ are two unit $\mathbb{Z}$-squares in a picture. If $(S^{\mathbb{Z}}_1, S^{\mathbb{Z}}_2) \in \bar{\nu} \cup \bar{\nu}^{-1}$ then $X_c(S^{\mathbb{Z}}_1) = X_c(S^{\mathbb{Z}}_2)$. If $(S^{\mathbb{Z}}_1, S^{\mathbb{Z}}_2) \in \bar{\eta} \cup \bar{\eta}^{-1}$ then $Y_c(S^{\mathbb{Z}}_1) = Y_c(S^{\mathbb{Z}}_2)$.*

Without ambiguity in the placement of unit $\mathbb{Z}$-squares, it should be possible to decide the realizability of a specification. This can be done by attempting to construct the realization.

PROPOSITION 5.1.4 *The $\mathbf{S}_{\mathbb{Z}}$-realizability of a finite specification S is decidable, and, if there is a realization, one can produce it algorithmically.*

Proof: I shall prove this by describing an algorithm for constructing a realization. The aim is to place all the squares down on a unit grid. Since these squares are constrained, they can only be placed exactly onto a grid position. Once a grid position is occupied, no other square can be placed on it.

Let us place the first square s_1. The grid is initially empty, so we can place it arbitrarily. Although there is an infinite number of positions to place the first square, the choice has no consequence in the proof.

Now we place the next square s_2. We choose one that is adjacent to s_1, in other words $(s_2, s_1) \in \nu \cup \nu^{-1} \cup \eta \cup \eta^{-1}$. Since (s_2, s_1) occurs in exactly one of ν, ν^{-1}, η, η^{-1}, Proposition 5.1.3 states that there can only be one exact position to place s_2. The grid positions surrounding the first one are empty so far, and therefore the second square can be placed.

Now we want to place another square s. We need to choose one that is adjacent to a square that is already on the grid. We can always find one since all the squares are connected. If the grid position is empty, then we can place it. If the grid position is occupied, then we stop.

We continue in this fashion until

1. we have placed all the squares; or

2. we have encountered a situation where overlapping of squares occurs.

If we have placed all the squares, then we clearly have produced a realization and therefore proved that the arrangement is realizable. Otherwise we can conclude that the arrangement is unrealizable. □

The above is easily proven because there are essentially no choices to be made in the layout of unit $\mathbb{Z}$-squares. The realizations are all similar, differing only in the absolute positions of the entire picture in the plane.

5.2 Integer rectangles

Let us relax the conditions slightly and examine a situation where there are more decisions to be made. Consider the layout of a set of rectangles instead of unit squares. These rectangles have integer dimensions, and can only be placed at integer-valued positions in the plane. We define such rectangles in the following.

DEFINITION 5.2.1 *A* $\mathbb{Z}$-rectangle *is a rectangle* $\mathbb{Z}\mathit{Rect}^{x,y}_{w,h} = \mathit{Rect}^{x,y}_{w,h}$ *such that* $w, h, x, y \in \mathbb{Z}$, $w, h > 0$. *Let* $\mathbf{R}_{\mathbb{Z}} = \{\mathbb{Z}\mathit{Rect}^{x,y}_{w,h} : w, h, x, y \in \mathbb{Z}, w, h > 0\}$ *be the set of all* $\mathbb{Z}$*-rectangles.*

We would like to consider the decidability of $\mathbf{R}_{\mathbb{Z}}$-realizability for specifications. Unlike unit $\mathbb{Z}$-squares that have only one size, the set of possible sizes of the $\mathbb{Z}$-rectangles is infinite. In order to bound the search space to be finite, an upper bound is imposed on the size for the $\mathbb{Z}$-rectangles. I will prove that the realizability of such a layout is decidable in the following.

PROPOSITION 5.2.2 *Suppose one has a finite specification* $\mathcal{S} = (\mathsf{R}, \nu, \eta)$. *We impose a size limit* $l \in \mathbb{Z}^+$ *on every rectangle symbol* $r_i \in \mathsf{R}$ *such that for any arrangement* α, *if* $r_i' = \alpha(r_i) = \mathit{Rect}^{x_i,y_i}_{w_i,h_i}$, *one has* $w_i \le l$ *and* $h_i \le l$. *Then the* $\mathbf{R}_{\mathbb{Z}}$*-realizability of* $\mathcal{S}$ *is decidable and, if there is a realization, one can produce it algorithmically.*

Proof: I shall prove the proposition by describing an algorithm for constructing a realization.

Since we are dealing with an infinite plane to place the rectangles, the number of realizations for every specification is clearly infinite. However in the proof of decidability, one needs not distinguish layouts that differ only in the absolute positions of the entire picture in the infinite plane. This corresponds to the selection of the absolute position of the first rectangle to be positioned. The choice of this initial position has no consequence in the proof.

We know that the sizes as well as the placement positions of every rectangle are integers. Together with the size limit imposed on rectangles, one knows that there can only be a finite number of layouts to be considered. For example, suppose one has $(a,b) \in \eta$ and the limit on size set at 2 units. Then a and b can have sizes $\{1 \times 1, 2 \times 1, 1 \times 2, 2 \times 2\}$. For the cases when both a and b have height 2 units, there are exactly three possible relative positions between a and b. They are shown in Figure 5.1.

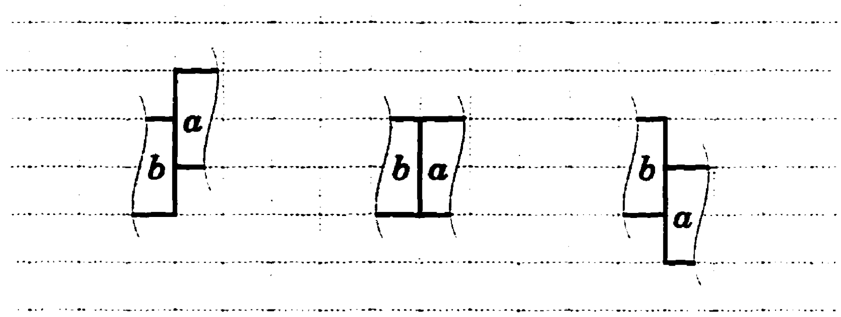


Figure 5.1: Relative placement positions for two touching rectangles of height 2 on a unit grid.

In general, if the size limit is l units, then each rectangle can have one of l^2 shapes. For each pair of adjacent rectangles, if m and n are the lengths of their touching sides, there are exactly $m+n-1$ possible placement positions. Therefore the upper bound on the number of cases to be considered is

$$|R|^{l^2} \times (|\nu| + |\eta|)^{2l-1}.$$

Although the total number of cases could be large it is, nevertheless, finite. All layout possibilities can be enumerated and we attempt to construct a realization for each case. In each construction, no further decisions are required. The construction algorithm is analogous to the one used in the proof of the previous proposition. It places the rectangles on the unit grid one after the other as specified. If all the rectangles can be placed without overlapping, then one has a realization.

If one is successful in constructing one realization, then the arrangement is realizable, otherwise it is not realizable. □

The previous proof enumerates all possible arrangements and attempts to construct them. The construction technique can also be used to determine the "spread" of the picture. We refer to the extents of a realization as the width and height of a smallest bounding rectangle that contains the realization. The following gives an upper bound to the extents of any realization.

COROLLARY 5.2.3 *Suppose one has a finite specification* $\mathcal{S} = (\mathsf{R}, \nu, \eta)$. *We impose a size bound* $l \in \mathbb{Z}^+$ *on every rectangle symbol* $r \in \mathsf{R}$ *such that for any arrangement* α, *if* $r' = \alpha(r) = \mathit{Rect}^{x,y}_{w,h}$, *one has* $w \leq l$ *and* $h \leq l$. *Then one can compute the maximum vertical and horizontal extents of* $\psi = \alpha(\mathsf{R})$ *for any* α.

Proof: We begin by arbitrarily selecting a starting rectangle $r_1 \in \mathsf{R}$. Let us take the bottom left corner of r_1, $\left(X_l(r_1), Y_b(r_1)\right)$, as the point of reference. Let $Y_{\max}$, $Y_{\min}$, $X_{\max}$, and $X_{\min}$ represent the extremes that can be reached by rectangles in the North, South, East and West directions respectively. The precise placement position for the next rectangle is not known, but one knows the range of possible positions. The horizontal range of the current rectangle is represented by $(x_{\min}, x_{\max})$ and its vertical range is represented by $(y_{\min}, y_{\max})$. This means that for the current rectangle r, $l \times x_{\min} \leq X_l(r) < X_r(r) \leq l \times x_{\max}$ and $l \times y_{\min} \leq Y_b(r) < Y_t(r) \leq l \times y_{\max}$. In the beginning, set

$$X_{\max} = Y_{\max} = x_{\max} = y_{\max} = 1 \quad \text{and} \quad X_{\min} = Y_{\min} = x_{\min} = y_{\min} = 0.$$

We traverse the specification starting from r_1.For every step taken from r_i to r_j, where $(r_i, r_j) \in \nu \cup \nu^{-1} \cup \eta \cup \eta^{-1}$, one performs the following:

1. Compute the range of positions for the next rectangle: for each of

$$\gamma = (x_{\min}, y_{\min}, x_{\max}, y_{\max}), \quad \gamma = \gamma + \Delta\gamma,$$

where

$$(\Delta x_{\min}, \Delta y_{\min}, \Delta x_{\max}, \Delta y_{\max}) = \begin{cases} (-1, 1, 1, 1) & \text{if } (r_j, r_i) \in \nu \\ (-1, -1, 1, -1) & \text{if } (r_j, r_i) \in \nu^{-1} \\ (1, -1, 1, 1) & \text{if } (r_j, r_i) \in \eta \\ (-1, -1, -1, 1) & \text{if } (r_j, r_i) \in \eta^{-1} \end{cases}$$

2. Update the extents if necessary:

$$\begin{array}{llll} \text{If } & y_{\max} > Y_{\max} & \text{then} & Y_{\max} = y_{\max} \\ \text{If } & y_{\min} < Y_{\min} & \text{then} & Y_{\min} = y_{\min} \\ \text{If } & x_{\max} > X_{\max} & \text{then} & X_{\max} = x_{\max} \\ \text{If } & x_{\min} < X_{\min} & \text{then} & X_{\min} = x_{\min} \end{array}$$

I will explain one case in step 1 to show how the Δ's are determined. Consider the case when $(r_j, r_i) \in \mathsf{v}$. It is clear the vertical range is shifted upwards by the height of one rectangle. The horizontal range of positions for r_j relative to r_i is shown in Figure 5.2. It increases the maximum horizontal range by almost the width of one rectangle and decreases the minimum horizontal range by the same amount. Therefore one has $(\Delta x_{\min}, \Delta y_{\min}, \Delta x_{\max}, \Delta y_{\max}) = (-1, 1, 1, 1)$.

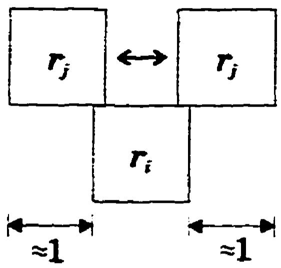


Figure 5.2: Range of possible positions for $(r_j, r_i) \in \mathsf{v}$.

After the entire specification is traversed, $Y_{\max}$, $Y_{\min}$, $X_{\max}$, and $X_{\min}$ store the extreme positions that can be reached by all the rectangles. Each rectangle is limited to size $l \times l$, therefore the maximum horizontal extent is $l \times (X_{\max} - X_{\min})$ and the maximum vertical extent is $l \times (Y_{\max} - Y_{\min})$. □

The converse of the previous corollary is also true. That is to say, given the size of an area, one can compute the maximum size limits for the $\mathbb{Z}$-rectangles such that any realization would be able to fit into it.

COROLLARY 5.2.4 *Suppose one has a finite specification $\mathcal{S} = (\mathsf{R}, \mathsf{v}, \eta)$ that is $\mathbf{R}_{\mathbb{Z}}$-realizable. Given an area of $w \times h$ square units, one can determine the maximum size limit $l \in \mathbb{Z}^+$ to be imposed on the $\mathbb{Z}$-rectangles such that it is always possible to display any $\mathbf{R}_{\mathbb{Z}}$-realization of $\mathcal{S}$ in the said area.*

Proof: Corollary 5.2.3 states that given a specification, the maximum extents of all $\mathbf{R}_{\mathbb{Z}}$-realizations can be computed. The maximum horizontal extent is given as $l \times (X_{\max} - X_{\min})$ and the maximum vertical extent is given as $l \times (Y_{\max} - Y_{\min})$. By setting the extents to be the maximum area available to display the realizations, one gets

$$w = l_1 \times (X_{\max} - X_{\min}) \;\Rightarrow\; l_1 = w/(X_{\max} - X_{\min})$$

and

$$h = l_2 \times (Y_{\max} - Y_{\min}) \;\Rightarrow\; l_2 = h/(Y_{\max} - Y_{\min})$$

Then the required maximum size limit is $l = \min(l_1, l_2)$. □

5.3 Rational rectangles

Now we are going to move away from integer coordinates to rational coordinates. We consider rectangles that have rational dimensions and positions.

DEFINITION 5.3.1 *A* $\mathbb{Q}$-**rectangle** *is a rectangle* $\mathit{QRect}^{x,y}_{w,h} = \mathit{Rect}^{x,y}_{w,h}$ *such that* $w, h, x, y \in \mathbb{Q}$. *Let* $\mathbf{R}_{\mathbb{Q}} = \{\mathit{QRect}^{x,y}_{w,h} : w, h, x, y \in \mathbb{Q}\}$ *denote the set of all* $\mathbb{Q}$*-rectangles.*

PROPOSITION 5.3.2 *A finite specification* $\mathcal{S} = (\mathsf{R}, \nu, \eta)$ *is* $\mathbf{R}_{\mathbb{Q}}$*-realizable if and only if it is* $\mathbf{R}_{\mathbb{Z}}$*-realizable.*

Proof: If $\mathcal{S}$ is $\mathbf{R}_{\mathbb{Z}}$-realizable, then there exists an arrangement α such that $\alpha(\mathsf{R}) \subseteq \mathbf{R}_{\mathbb{Z}}$. But $\mathbf{R}_{\mathbb{Z}} \subset \mathbf{R}_{\mathbb{Q}}$, therefore $\mathcal{S}$ is $\mathbf{R}_{\mathbb{Q}}$-realizable.

The next step establishes that if a specification has a $\mathbf{R}_{\mathbb{Q}}$-realization then it also has a $\mathbf{R}_{\mathbb{Z}}$-realization. To do that, one computes a common denominator for all the rational numbers used in all the $\mathbb{Q}$-rectangles in the $\mathbf{R}_{\mathbb{Q}}$-realization. Then the entire $\mathbf{R}_{\mathbb{Q}}$-realization is scaled up by this common denominator. The resulting realization would therefore contain only integer values.

For a $\mathbf{R}_{\mathbb{Q}}$-realization ψ, $d \in \mathbb{Z}^+$ is a *common denominator* of ψ if for all $r = \mathit{Rect}^{x,y}_{w,h} \in \psi$, there exist integers i_1, i_2, i_3, i_4 such that $w = i_1/d$, $h = i_2/d$, $x = i_3/d$, and $y = i_4/d$.

We need to show that a common denominator exists for all $\mathbf{R}_{\mathbb{Q}}$-realizations. The following shows how one can be computed.

Let $\psi = \{r_i = \mathit{Rect}^{x_i,y_i}_{w_i,h_i}\}$ be a $\mathbf{R}_{\mathbb{Q}}$-realization. Then $\forall i\,[w_i, h_i, x_i, y_i \in \mathbb{Q}]$. Every rational number is represented by a fraction, say

$$
\begin{aligned}
w_i &= N_{i1}/D_{i1} \\
h_i &= N_{i2}/D_{i2} \\
x_i &= N_{i3}/D_{i3} \\
y_i &= N_{i4}/D_{i4}
\end{aligned}
$$

for some $N_{ij}, D_{ij} \in \mathbb{Z}$. Then a common denominator of ψ can be computed by

$$d = \operatorname*{lcm}_{\forall i,j}(D_{i,j})$$

where lcm() is the function computing the least common multiple.

Let ψ be a realization of $\mathcal{S}$ where every rectangle is a $\mathbb{Q}$-rectangle. Let d be a common denominator for ψ. A new picture ψ' is constructed by scaling every $\mathbb{Q}$-rectangle by d. That is to say, for every $\mathbb{Q}$-rectangle $r = Rect^{x,y}_{w,h} \in \psi$, a new rectangle $r' = Rect^{x',y'}_{w',h'}$ is obtained, where:

$$
\begin{aligned}
w' &= d \times w \\
h' &= d \times h \\
x' &= d \times x \\
y' &= d \times y
\end{aligned}
$$

By the definition of the common denominator, one knows that $w'.h'.x'.y' \in \mathbb{Z}$. Since the same proportional scaling is applied to all rectangles, the geometric compass relations remain unchanged, that is to say $\bar{\nu}_{\psi'} = \bar{\nu}_{\psi}$ and $\bar{\eta}_{\psi'} = \bar{\eta}_{\psi}$. Then ψ' is a realization of $\mathcal{S}$ where every rectangle is a $\mathbb{Z}$-rectangle. Hence $\mathcal{S}$ is $\mathbf{R}_{\mathbb{Z}}$-realizable. $\square$

5.4 Real rectangles

In this section, we will try to associate the restricted cases with the original problem. We want to establish the relationship between realizability and $\mathbf{R}_{\mathbb{Q}}$-realizability.

PROPOSITION 5.4.1 *A finite specification $\mathcal{S} = (\mathsf{R}, \nu, \eta)$ is realizable if and only if it is $\mathbf{R}_{\mathbb{Q}}$-realizable.*

Proof: If $\mathcal{S}$ is $\mathbf{R}_{\mathbb{Q}}$-realizable, then there exists an arrangement α such that $\alpha(\mathsf{R}) \subset \mathbf{R}_{\mathbb{Q}}$. But $\mathbf{R}_{\mathbb{Q}} \subset \mathbf{Rect}$, therefore $\mathcal{S}$ is realizable.

To show the converse, one must provide a way of constructing a $\mathbf{R}_{\mathbb{Q}}$-realization given a realization. In any realization, there is only a finite set of real numbers used to locate the corners of all the rectangles. These corners can be shifted slightly to a position that involves only rational numbers. Then the new picture is a $\mathbf{R}_{\mathbb{Q}}$-realization.

Let ψ be a realization of $\mathcal{S}$. For all rectangles $r_i \in \psi$ we collect the real values representing the coordinates of the corners of r_i, namely $X_l(r_i)$, $X_r(r_i)$, $Y_b(r_i)$, and $Y_t(r_i)$. These numbers are sorted in ascending order. Let the sorted sequence be $p_1, p_2, \ldots, p_n$. For every pair of real numbers (p_{i-1}, p_i) in the sequence, one can find a rational number q_{i-1} such that $p_{i-1} < q_{i-1} \leq p_i$. One can also find a rational number $q_n > p_n$. We construct the corresponding sequence of rational numbers $q_1, q_2, \ldots, q_n$.

A new picture ψ' is constructed by "shifting" all real values p_i to the rational values q_i. For the coordinates of the corners of every rectangle, real value p_i is replaced with the corresponding rational value q_i. The geometric compass relations remain unchanged, that is, $\bar{\nu}_{\psi'} = \bar{\nu}_{\psi}$ and $\bar{\eta}_{\psi'} = \bar{\eta}_{\psi}$; moreover rectangles in ψ' do not overlap. Thus ψ' is a realization of $\mathcal{S}$ where every rectangle is a $\mathbb{Q}$-rectangle. Hence $\mathcal{S}$ is $\mathbf{R}_{\mathbb{Q}}$-realizable. □

Propositions 5.3.2 and 5.4.1 can be summarized by the following corollary.

COROLLARY 5.4.2 *A finite specification $\mathcal{S}$ is realizable if and only if it is $\mathbf{R}_{\mathbb{Q}}$-realizable.*

CHAPTER 6

AUGMENTED PROBLEMS

In Chapter 3, we have seen that specifications involving only ν and η may have many possible realizations. The large number of cases causes difficulties when we are trying to establish properties of the specifications.

In this chapter, additional ordering relations for the rectangles are explored. These ordering relations are first introduced on the realizations. Then an abstract equivalent of such relations is extracted for the specifications.

6.1 Clockwise ordering

Figure 3.3 shows two pictures with the same $\bar{\nu} = \{(a,b),(a,c)\}$ and $\bar{\eta} = \emptyset$. By looking only at the geometric compass relations, one is unable to distinguish them. An ordering of rectangles b and c with respect to a is required.

An obvious way to order them is to use the spatial relationships between the centres of the rectangles. In our example, we construct directed edges from a to b and from a to c. The orientation of the edges is used to order b and c.

DEFINITION 6.1.1 (θ) *For two adjacent rectangles r_1 and r_2, $\theta_{r_1r_2}$ is defined as the anti-clockwise angle from a horizontal edge directed to the East and the edge $\overrightarrow{c_1c_2}$, where $c_1 = C(r_1)$ and $c_2 = C(r_2)$.*

The intention is to use θ to order rectangles, but there is a problem. Figure 6.1 shows two possible pictures with $\bar{\nu} = \{(a,b),(a,c)\}$ and $\bar{\eta} = \emptyset$. In case (a), one has $\theta_{ac} > \theta_{ab}$. But in case (b), one has $\theta_{ac} < \theta_{ab}$. The ordering is reversed by "stretching" a. This is undesirable as we want to consider the ordering of b and c to be the same for both cases.

The ordering technique is improved by using a point on the the edge of intersection instead of the centres. The midpoint on the edge of intersection is

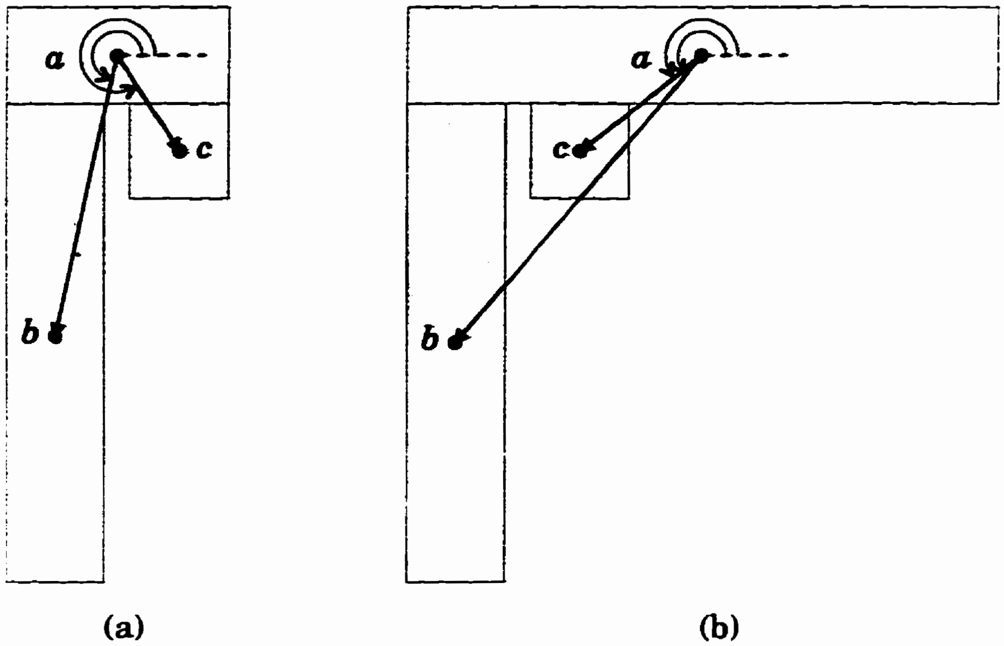


Figure 6.1: Problem with using centres of rectangles: (a) $\theta_{ac} > \theta_{ab}$ (b) $\theta_{ac} < \theta_{ab}$.

used.

DEFINITION 6.1.2 (θ') *For two adjacent rectangles r_1 and r_2, $\theta'_{r_1 r_2}$ is defined as the anti-clockwise angle from a horizontal edge directed to the East and the edge $\overrightarrow{c_1 m}$, where $c_1 = C(r_1)$ and $m = M(r_1.r_2)$.*

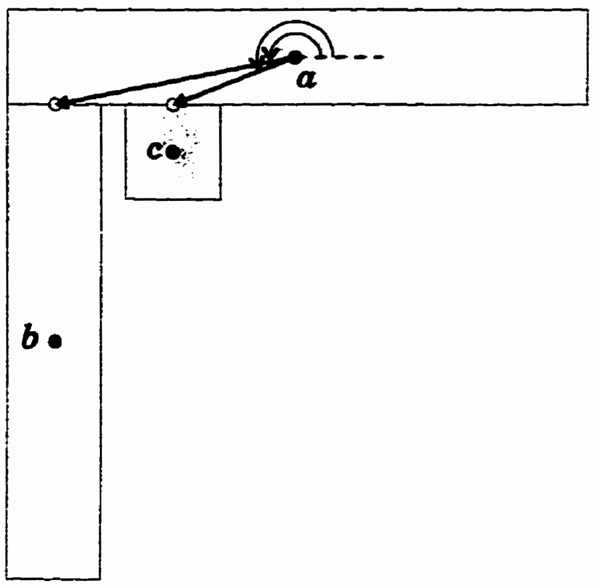


Figure 6.2: Adding the centres of common edge segments.

Now θ' is used to define a relation $\bar{\tau}$ in an attempt to order adjacent rectangles.

DEFINITION 6.1.3 *$\bar{\tau}$ is a binary relation on rectangles such that $(r_2, r_3) \in \bar{\tau}$ if*

1. *There exists $r_1 \in R$ such that $(r_2, r_1), (r_3, r_1) \in \bar{\nu}$ or $(r_1, r_2), (r_1, r_3) \in \bar{\nu}$; and*
2. $\theta_{r_1 r_2} > \theta_{r_1 r_3}$.

Let us derive $\bar{\tau}$ for an example. Consider the example given in Figure 6.3. If we consider the ordering of the rectangles s_1, s_2, and s_3 with respect to r_1, we would obtain

$$(s_3, s_2), (s_2, s_1) \in \bar{\tau}.$$

However, if we consider the ordering with respect to r_2, we would obtain

$$(s_1, s_2), (s_2, s_3) \in \bar{\tau}.$$

This is again undesirable, as the ordering of the s_i's are different depending on the reference rectangles.

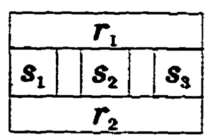


Figure 6.3: The ordering of the s_i's are different depending on which of the reference rectangles r_1 and r_2 is used.

The motivation for using an ordering based on directions is to attempt to obtain one additional ordering relation $\bar{\tau}$ to order the rectangles for both $\bar{\nu}$ and $\bar{\eta}$. However it is shown that this method is unsuccessful because $\bar{\tau}$ does not give an ordering that is globally consistent, that is, antisymmetric.

6.2 Geometric order relations $\omega_{\bar{\nu}}$ and $\omega_{\bar{\eta}}$

In our next attempt to order rectangles, I introduce a pair of relations, one for each of the geometric compass relations $\bar{\nu}$ and $\bar{\eta}$.

DEFINITION 6.2.1 (GEOMETRIC ORDER RELATIONS) *Let ψ be a picture. I define the binary relations $\omega_{\bar{\nu}_\psi}$ and $\omega_{\bar{\eta}_\psi}$ on the set of rectangles in ψ as follows. Let r_1 and r_2 be rectangles in ψ.*

1. *One has* $(r_1, r_2) \in \omega_{\bar{\nu}_\psi}$ *if* $X_c(r_1) > X_c(r_2)$ *and there exists a rectangle* r_3 *in* ψ *such that* $(r_1, r_3), (r_2, r_3) \in \bar{\nu}_\psi$ *or* $(r_3, r_1), (r_3, r_2) \in \bar{\nu}_\psi$.

2. *One has* $(r_1, r_2) \in \omega_{\bar{\eta}_\psi}$ *if* $Y_c(r_1) > Y_c(r_2)$ *and there exists a rectangle* r_3 *in* ψ *such that* $(r_1, r_3), (r_2, r_3) \in \bar{\eta}_\psi$ *or* $(r_3, r_1), (r_3, r_2) \in \bar{\eta}_\psi$.

I call $\omega_{\bar{\nu}_\psi}$ *and* $\omega_{\bar{\eta}_\psi}$ *the* **geometric order relations.**

Note on symmetry

There is a symmetry regarding ν and η. Properties of ν and η are identical, with the appropriate mirroring and/or rotation. The same symmetry extends to ω_ν and ω_η. In the sequel, we will exploit this symmetry to simplify the discussion by avoiding the unnecessary duplication of arguments for ν and η, and for ω_ν and ω_η. Therefore all propositions that are stated may be specific to one orientation, but are also true for other orientations, with the appropriate substitutions.

PROPOSITION 6.2.2 $\omega_{\bar{\nu}}$ *and* $\omega_{\bar{\eta}}$ *have the following properties:*

1. *There are no* $\omega_{\bar{\nu}}$*-cycles and no* $\omega_{\bar{\eta}}$*-cycles;*

2. $\omega_{\bar{\nu}}$ *and* $\omega_{\bar{\eta}}$ *are not transitive in general;*

3. $\omega_{\bar{\nu}}$, $\omega_{\bar{\nu}}^{-1}$, $\omega_{\bar{\eta}}$, *and* $\omega_{\bar{\eta}}^{-1}$ *are pairwise disjoint.*

Proof:

1. (No cycles) Let us assume, on the contrary, that there is a $\omega_{\bar{\nu}}$-cycle such that $(r_0, r_1), (r_1, r_2), \ldots, (r_{n-1}, r_n) \in \omega_{\bar{\nu}}$, $r_0 = r_n$, for some $n \geq 1$. Then by definition one has $X_c(r_0) > X_c(r_1) > \cdots > X_c(r_n) = X_c(r_0)$, which is a contradiction.

2. (Non-transitivity) Consider the counter-example shown in Figure 6.4. In the layout, one has $(r_2, r_1), (r_3, r_2) \in \omega_{\bar{\nu}}$. But $(r_3, r_1) \notin \omega_{\bar{\nu}}$ because it does not satisfy part 1 of the definition of $\omega_{\bar{\nu}}$.

3. (Disjointness) $\omega_{\bar{\nu}} \cap \omega_{\bar{\nu}}^{-1} = \omega_{\bar{\eta}} \cap \omega_{\bar{\eta}}^{-1} = \emptyset$ has been shown above by the fact that there are no $\omega_{\bar{\nu}}$-cycles and no $\omega_{\bar{\eta}}$-cycles. Let us consider the case for $\delta \cap \gamma$, for

$$(\delta, \gamma) = \{(\omega_{\bar{\nu}}, \omega_{\bar{\eta}}), (\omega_{\bar{\nu}}, \omega_{\bar{\eta}}^{-1}), (\omega_{\bar{\nu}}^{-1}, \omega_{\bar{\eta}}), (\omega_{\bar{\nu}}^{-1}, \omega_{\bar{\eta}}^{-1})\}.$$

Let $(r_1, r_2) \in \delta \cap \gamma$. If $(r_1, r_2) \in \delta$ then $Y_t(r_1) = Y_t(r_2)$ or $Y_b(r_1) = Y_b(r_2)$. Hence $((Y_b(r_1), Y_t(r_1)) \cap (Y_b(r_2), Y_t(r_2)) \neq \emptyset$. If $(r_1, r_2) \in \gamma$ then $X_l(r_1) = X_l(r_2)$ or $X_r(r_1) = X_r(r_2)$. Hence $((X_l(r_1), X_r(r_1)) \cap (X_l(r_2), X_r(r_2)) \neq \emptyset$. Thus, r_1 overlaps r_2. Overlaps are not allowed, therefore $\delta \cap \gamma = \emptyset$.

□

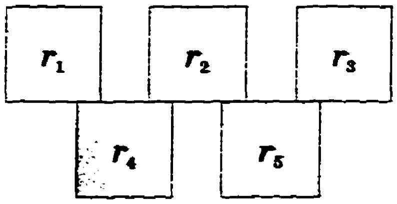


Figure 6.4: Counter-example for non-transitivity of $\omega_{\bar{v}}$.

PROPOSITION 6.2.3 *If* $(r_1, r_2) \in \omega_{\bar{v}}$ *then* $X_l(r_1) \geq X_r(r_2)$.

Proof: Assume that one has $(r_1, r_2) \in \omega_{\bar{v}}$. By definition the rectangles r_1 and r_2 have a common top or bottom boundary, that is, $Y_t(r_1) = Y_t(r_2)$ or $Y_b(r_1) = Y_b(r_2)$. They are not allowed to be degenerate, therefore their vertical extents overlap,

$$(Y_b(r_1), Y_t(r_1)) \cap (Y_b(r_2), Y_t(r_2)) \neq \emptyset.$$

Since rectangles in a picture do not overlap, one has

$$(X_l(r_1), X_r(r_1)) \cap (X_l(r_2), X_r(r_2)) = \emptyset.$$

Since $X_c(r_1) > X_c(r_2)$ and $X_l(r) < X_c(r) < X_r(r)$ for any rectangle r, we must have $X_l(r_1) \geq X_r(r_2)$. □

PROPOSITION 6.2.4 *If* $(a, b), (b, c) \in \omega_{\bar{v}}$ *then* $(a, c) \notin \bar{v} \cup \bar{v}^{-1} \cup \bar{\eta} \cup \bar{\eta}^{-1}$.

Proof: b is "in between" a and c. For a and c to be adjacent, b would have to be degenerate, which is not allowed.

If $(a, b), (b, c) \in \omega_{\bar{v}}$, then $X_l(a) \geq X_r(b) > X_l(b) \geq X_r(c)$. This means that

$$\Big(X_l(a), X_r(a)\Big) \cap \Big(X_l(c), X_r(c)\Big) = \emptyset$$

and therefore $(a, c) \notin \bar{v} \cup \bar{v}^{-1} \cup \bar{\eta} \cup \bar{\eta}^{-1}$. □

LEMMA 6.2.5 *If* $(r_1, r_2) \in \omega_{\bar{v}}^{+}$ *then* $X_c(r_1) > X_c(r_2)$ *and* $X_l(r_1) \geq X_r(r_2)$.

Proof: If $(r_1, r_2) \in \omega_{\bar{\nu}}^{+}$, then there are $s_1, \ldots, s_n$ such at $(r_1, s_1), (s_1, s_2), \ldots, (s_n, r_2) \in \omega_{\bar{\nu}}$. Thus $X_c(r_1) > X_c(s_1) > X_c(s_2) > \ldots > X_c(s_n) > X_c(r_2)$ and $X_l(r_1) \geq X_r(s_1) > X_l(s_1) \geq X_r(s_2) > \cdots \geq X_r(s_n) > X_l(s_n) \geq X_r(r_2)$. □

PROPOSITION 6.2.6 *$\omega_{\bar{\nu}}^{+}$ and $\omega_{\bar{\eta}}^{+}$ have the following properties:*

1. *There are not $\omega_{\bar{\nu}}^{+}$-cycles and no $\omega_{\bar{\eta}}^{+}$-cycles;*

2. *$\omega_{\bar{\nu}}^{+}$ and $\omega_{\bar{\eta}}^{+}$ are transitive;*

Proof:

1. (No cycles) Let us assume, on the contrary, that there is a $\omega_{\bar{\nu}}^{+}$-cycle such that $(r_0, r_1), (r_1, r_2), \ldots, (r_{n-1}, r_n) \in \omega_{\bar{\nu}}^{+}$, $r_0 = r_n$, for some $n \geq 1$. Then, from Lemma 6.2.5, one has $X_c(r_0) > X_c(r_1) > \cdots > X_c(r_n) = X_c(r_0)$, which yields a contradiction.

2. (Transitivity) $\omega_{\bar{\nu}}^{+}$ is transitive by definition.

□

I believe that in addition to the properties listed in Proposition 6.2.6, the relations $\omega_{\bar{\nu}}^{+}$ and $\omega_{\bar{\eta}}^{+}$ are disjoint, but I do not have a complete formal proof at this point.

6.3 Abstract order relations ω_ν and ω_η

Now we need to extend the specification to include the ω orderings. I define a new form of specification called ω-specification.

DEFINITION 6.3.1 (ω-SPECIFICATION) *An* ω-specification *is given by*

$$\mathcal{S}_\omega \stackrel{\text{def}}{=} (\mathsf{R}, \nu, \eta, \omega_\nu, \omega_\eta)$$

where $\nu, \eta, \omega_\nu, \omega_\eta \subseteq \mathsf{R} \times \mathsf{R}$. *I call ω_ν and ω_η the abstract order relations.*

DEFINITION 6.3.2 (REALIZABILITY WITH ω) *Consider an ω-specification $\mathcal{S}_\omega = (\mathsf{R}, \nu, \eta, \omega_\nu, \omega_\eta)$ and an arrangement α. Let $\psi = \alpha(\mathsf{R})$. α is* acceptable *(or a* realization*) if $\bar{\nu}_\psi = \nu$, $\bar{\eta}_\psi = \eta$, $\omega_{\bar{\nu}_\psi} = \omega_\nu$, and $\omega_{\bar{\eta}_\psi} = \omega_\eta$.*

PROPOSITION 6.3.3 *Suppose an ω-specification $\mathcal{S}_\omega = (\mathsf{R}, \nu, \eta, \omega_\nu, \omega_\eta)$ is realizable, then*

1. ω_ν *and* ω_η *have the following properties:*

   (a) *There are no* ω_ν*-cycles and no* ω_η*-cycles;*

   (b) ω_ν *and* ω_η *are not transitive;*

   (c) ω_ν, ω_ν^{-1}, ω_η, *and* ω_η^{-1} *are pairwise disjoint.*

   (d) *For* $\beta = \{\nu, \eta\}$, *if* $(r_1, r_2) \in \omega_\beta$ *then there is a* $r_3 \in \mathrm{R}$ *such that* $(r_1, r_3), (r_2, r_3) \in \beta$ *or* $(r_3, r_1), (r_3, r_2) \in \beta$.

2. ω_ν^+ *and* ω_η^+ *have the following properties:*

   (a) *There are no* ω_ν^+*-cycles and no* ω_η^+*-cycles;*

   (b) ω_ν^+ *and* ω_η^+ *are not transitive;*

Proof: Let $\psi = \alpha(\mathrm{R})$ be a realization. Then $(\bar{\nu}_\psi, \bar{\eta}_\psi, \omega_{\bar{\nu}_\psi}, \omega_{\bar{\eta}_\psi}) = (\nu, \eta, \omega_\nu, \omega_\eta)$.

1. Proposition 6.2.1 states that for any picture ψ, $\omega_{\bar{\nu}_\psi}$ and $\omega_{\bar{\eta}_\psi}$ have the properties 1(a) to 1(c). 1(d) is an immediate consequence of Definition 6.2.1.

2. Proposition 6.2.6 states that for any picture ψ, $\omega_{\bar{\nu}_\psi}^+$ and $\omega_{\bar{\eta}_\psi}^+$ have the properties 2(a) and 2(b).

□

EXAMPLE 6.3.4 *Consider the specification*

$$\mathcal{S} = \{\{r_1, \ldots, r_5\}, \{(r_1, r_2), (r_1, r_3), (r_1, r_4), (r_2, r_5), (r_4, r_5)\}, \emptyset\}.$$

Without additional ordering, we have several possible realizations classes, shown in Figure 6.5(a-f). If we add to the specification $\omega_\nu = \{(r_3, r_2), (r_4, r_3), (r_4, r_2)\}$, *there is only a unique realization class, the one shown in Figure 6.5(a).*

Now we need to extend the definitions of symmetry images so that they include ω-specifications.

DEFINITION 6.3.5 (SYMMETRY IMAGES OF AN ω-SPECIFICATION) *The* symmetry images *of an* ω*-specification* $\mathcal{S}_\omega = (\mathrm{R}, \nu, \eta, \omega_\nu, \omega_\eta)$ *and their names are given by:*

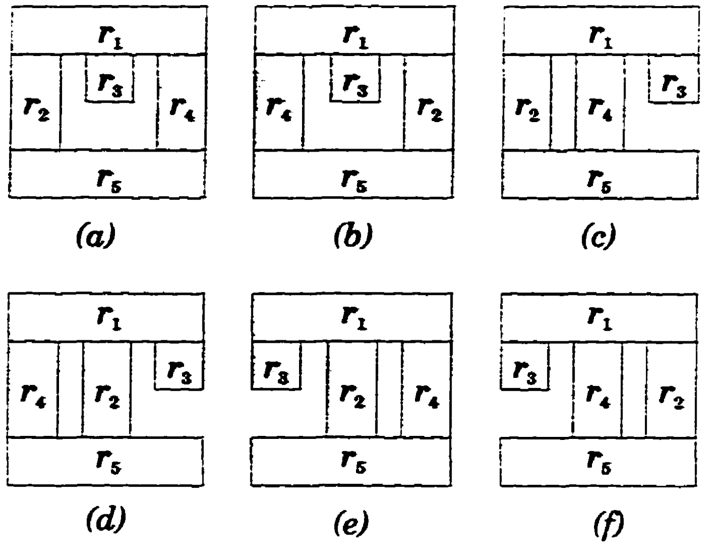


Figure 6.5: Realizations with or without ω_ν of Example 6.3.4.

$$
\begin{aligned}
{}_1\mathcal{S}_\omega &= (\mathsf{R}, \nu, \eta, \omega_\nu, \omega_\eta) \\
{}_2\mathcal{S}_\omega &= (\mathsf{R}, \nu, \eta^{-1}, \omega_\nu^{-1}, \omega_\eta) \\
{}_3\mathcal{S}_\omega &= (\mathsf{R}, \nu^{-1}, \eta, \omega_\nu, \omega_\eta^{-1}) \\
{}_4\mathcal{S}_\omega &= (\mathsf{R}, \nu^{-1}, \eta^{-1}, \omega_\nu^{-1}, \omega_\eta^{-1}) \\
{}_5\mathcal{S}_\omega &= (\mathsf{R}, \eta, \nu, \omega_\eta, \omega_\nu) \\
{}_6\mathcal{S}_\omega &= (\mathsf{R}, \eta, \nu^{-1}, \omega_\eta^{-1}, \omega_\nu) \\
{}_7\mathcal{S}_\omega &= (\mathsf{R}, \eta^{-1}, \nu, \omega_\eta, \omega_\nu^{-1}) \\
{}_8\mathcal{S}_\omega &= (\mathsf{R}, \eta^{-1}, \nu^{-1}, \omega_\eta^{-1}, \omega_\nu^{-1})
\end{aligned}
$$

One verifies that Proposition 2.3.10 can be extended to ω-specifications using ${}_i\mathcal{S}_\omega$ instead of ${}_i\mathcal{S}$.

6.3.1 ω does not restrict the problem

In the previous example, ω_ν was considered as an addition to the specification. However, since R is finite, there can only be a finite number of different ω_ν and ω_η for a particular specification. So for any specification, all possible ω_ν and ω_η can be enumerated to obtain ω-specifications. Another possibility is to begin with subsets of the ordering relations, possibly provided as additional information together with the abstract compass relations. Then the remaining orderings can be automatically enumerated and tested.

Not all ω-specifications will be realizable. But the important point to note is that if one such ω-specification is realizable, then we can say that the same specification defined without ω is realizable.

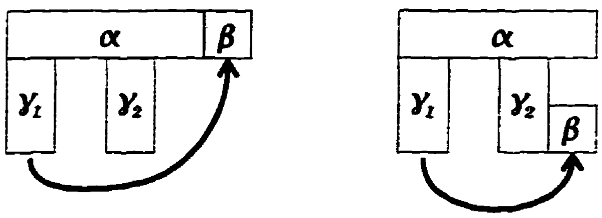
OBSERVATION 6.3.6 *A specification $\mathcal{S} = (\mathsf{R}, \nu, \eta)$ is realizable if and only if there are ω_ν, ω_η such that the ω-specification $\mathcal{S}_\omega = (\mathsf{R}, \nu, \eta, \omega_\nu, \omega_\eta)$ is realizable.*

COROLLARY 6.3.7 *Realizability is decidable for specification $\mathcal{S}$ if and only if it is decidable for an ω-specification $\mathcal{S}_\omega$.*

6.3.2 Using ω

In this section we explore some basic properties of ω_ν and ω_η. This is intended to set the stage for the more sophisticated cases studied in Section 6.7 and to show some typical proofs.

PROPOSITION 6.3.8 *Consider 4 rectangle symbols α, β, γ_1, and γ_2 in a realizable ω-specification $\mathcal{S}_\omega = (\mathsf{R}, \nu, \eta, \omega_\nu, \omega_\eta)$. Suppose one has $(\alpha, \gamma_1), (\alpha, \gamma_2) \in \nu$ and $(\gamma_2, \gamma_1) \in \omega_\nu$. If $(\beta, \alpha) \in \eta \cup \omega_\nu^+$ or $(\beta, \gamma_2) \in \eta \cup \omega_\nu^+$ then $(\beta, \gamma_1) \notin \nu \cup \nu^{-1} \cup \eta \cup \eta^{-1}$. This is also true for the symmetry images ${}_2\mathcal{S}_\omega$ to ${}_8\mathcal{S}_\omega$ of $\mathcal{S}_\omega$.*



(a) (b)

Figure 6.6: Roadblock.

Proof: The situation for this proposition is shown in Figure 6.6. As the specification is realizable, there exists an arrangement α such that $\psi = \alpha(\mathcal{S}_\omega)$ and $(\nu, \eta, \omega_\nu, \omega_\eta) = (\bar{\nu}_\psi, \bar{\eta}_\psi, \omega_{\bar{\nu}_\psi}, \omega_{\bar{\eta}_\psi})$. Lemma 6.2.5 is used to provide the geometric properties of the rectangles in this proof.

Consider $(\beta,\alpha) \in \eta \cup \omega_\nu^+$. It is known that $\eta \cup \omega_\nu^+ = \bar{\eta}_\psi \cup \omega_{\bar{\nu}_\psi}^+$, so one has $X_r(\alpha) \leq X_l(\beta)$. Since $(\alpha,\gamma_2) \in \nu \cup \nu^{-1} = \bar{\nu}_\psi \cup \bar{\nu}_\psi^{-1}$, one has $X_l(\gamma_2) < X_r(\alpha)$. $(\gamma_2,\gamma_1) \in \omega_\nu = \omega_{\bar{\nu}_\psi}$, thus $X_r(\gamma_1) \leq X_l(\gamma_2) < X_r(\alpha) \leq X_l(\beta)$.

Now consider $(\beta,\gamma_2) \in \eta \cup \omega_\nu^+$. It is known that $(\beta,\gamma_2) \in \eta \cup \omega_\nu^+ = \bar{\eta}_\psi \cup \omega_{\bar{\nu}_\psi}^+$, so one has $X_r(\gamma_2) \leq X_l(\beta)$. $(\gamma_2,\gamma_1) \in \omega_\nu = \omega_{\bar{\nu}_\psi}$, thus $X_r(\gamma_1) \leq X_l(\gamma_2) < X_r(\gamma_2) \leq X_l(\beta)$.

I have shown in both cases that $X_r(\gamma_1) < X_l(\beta)$, and by Proposition 3.1.2, one has $(\beta,\gamma_1) \notin \bar{\nu} \cup \bar{\nu}^{-1} \cup \bar{\eta} \cup \bar{\eta}^{-1} = \nu \cup \nu^{-1} \cup \eta \cup \eta^{-1}$. The proofs for ${}_2S_\omega$ to ${}_8S_\omega$ are obtained by symmetry. □

Now we look at another "road block" situation, shown in Figure 6.7. γ_2 acts as a road block to prevent γ_1 to be adjacent to β or δ.

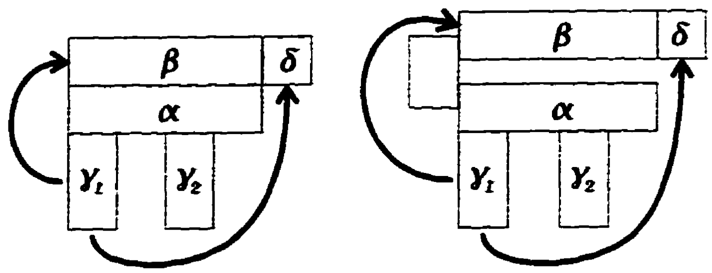


Figure 6.7: Roadblock.

PROPOSITION 6.3.9 *Consider 5 rectangles* $\alpha,\beta,\delta,\gamma_1$, *and* γ_2 *in a realizable* ω-*specification. Suppose one has* $(\alpha,\gamma_1),(\alpha,\gamma_2) \in \nu$, $(\beta,\alpha) \in \nu \cup \omega_\eta^+$, $(\delta,\beta) \in \eta \cup \omega_\nu^+$, *and* $(\gamma_2,\gamma_1) \in \omega_\nu$, *then* $(\gamma_1,\beta),(\gamma_1,\delta) \notin \nu \cup \nu^{-1} \cup \eta \cup \eta^{-1}$. *This is also true for the symmetry images* ${}_2S_\omega$ *to* ${}_8S_\omega$ *of* S_ω.

Proof: As the specification is realizable, there exists an arrangement α such that $\psi = \alpha(S_\omega)$ and $(\nu,\eta,\omega_\nu,\omega_\eta) = (\bar{\nu}_\psi, \bar{\eta}_\psi, \omega_{\bar{\nu}_\psi}, \omega_{\bar{\eta}_\psi})$. Again, Lemma 6.2.5 is used to provide the geometric properties of the rectangles in this proof.

$(\beta,\alpha) \in \nu \cup \omega_\eta^+ = \bar{\nu}_\psi \cup \omega_{\bar{\eta}_\psi}^+$, thus

$$Y_t(\alpha) \leq Y_b(\beta).$$

$(\delta,\beta) \in \eta \cup \omega_\nu^+ = \bar{\eta}_\psi \cup \omega_{\bar{\nu}_\psi}^+$, thus

$$X_r(\beta) \leq X_l(\delta).$$

$(\alpha,\gamma_2) \in \nu = \bar{\nu}_\psi$, thus

$$X_l(\gamma_2) < X_r(\alpha).$$

$(\alpha, \gamma_1) \in \nu = \bar{\nu}_\psi$, thus

$$Y_t(\gamma_1) = Y_b(\alpha) < Y_t(\alpha) \leq Y_b(\beta).$$

Since $Y_t(\gamma_1) < Y_b(\beta)$, by Proposition 3.1.2,

$$(\gamma_1, \beta) \notin \bar{\nu} \cup \bar{\nu}^{-1} \cup \bar{\eta} \cup \bar{\eta}^{-1} = \nu \cup \nu^{-1} \cup \eta \cup \eta^{-1}.$$

Now I show the same for δ. There are two cases, either $X_r(\alpha) \leq X_l(\delta)$ or $X_r(\alpha) > X_l(\delta)$.

1. Assume $X_r(\alpha) \leq X_l(\delta)$.
   It is known that $(\gamma_2, \gamma_1) \in \omega_\nu = \omega_{\bar{\nu}_\psi}$, thus

   $$X_r(\gamma_1) \leq X_l(\gamma_2) < X_r(\alpha) \leq X_l(\delta).$$

2. Now assume $X_r(\alpha) > X_l(\delta)$. Again I separate into two cases, $(\beta, \alpha) \in \nu$ or $(\beta, \alpha) \in \omega_\eta^+$.

   (a) Consider $(\beta, \alpha) \in \nu$.
   It is known that $(\beta, \alpha) \in \nu = \bar{\nu}_\psi$, so one has

   $$X_l(\alpha) < X_r(\beta) \leq X_l(\delta).$$

   Thus,

   $$(X_l(\alpha), X_r(\alpha) \cap (X_l(\delta), X_r(\delta)) \neq \emptyset.$$

   Rectangles in a picture do not overlap, therefore by Definition 2.2.5 one has

   $$(Y_b(\alpha), Y_t(\alpha) \cap (Y_b(\delta), Y_t(\delta)) = \emptyset.$$

   This means that

   $$Y_b(\delta) \geq Y_t(\alpha) > Y_b(\alpha) \geq Y_t(\gamma_1).$$

   (b) Consider $(\beta, \alpha) \in \omega_\eta^+$.
   It is known that $(\beta, \alpha) \in \omega_\eta^+ = \omega_{\bar{\eta}_\psi}^+$, therefore

   $$Y_b(\beta) \geq Y_t(\alpha) > Y_b(\alpha) = Y_t(\gamma_1).$$

In all cases, I have shown that either the vertical extents or the horizontal extents of α and γ_1 do not intersect. Therefore by Proposition 3.1.2, $(\alpha, \gamma_1) \notin \bar{\nu} \cup \bar{\nu}^{-1} \cup \bar{\eta} \cup \bar{\eta}^{-1} = \nu \cup \nu^{-1} \cup \eta \cup \eta^{-1}$. The proofs for ${}_2\mathcal{S}_\omega$ to ${}_8\mathcal{S}_\omega$ are obtained by symmetry. □

I have exhibited two simple "road block" conditions. Many more such cases exist. This is explored in later in this chapter.

6.4 Equivalence of arrangements

With the introduction of the relations ω_ν and ω_η, it is now possible to state the conditions for the equivalence of arrangements.

DEFINITION 6.4.1 (EQUIVALENCE OF ARRANGEMENTS) *Let $\mathcal{S} = (\mathsf{R}, \nu, \eta)$ be a realizable specification and α and α' be two arrangements such that $\psi = \alpha(\mathsf{R})$ and $\psi' = \alpha'(\mathsf{R})$. α and α' are* equivalent *if*

$$(\bar{\nu}_\psi, \bar{\eta}_\psi, \omega_{\bar{\nu}_\psi}, \omega_{\bar{\eta}_\psi}) = (\bar{\nu}_{\psi'}, \bar{\eta}_{\psi'}, \omega_{\bar{\nu}_{\psi'}}, \omega_{\bar{\eta}_{\psi'}})$$

I would like to examine the geometric consequences of the equivalence of the arrangements. The intuition is that the pictures that result from equivalent arrangements are "similar" to each other.

The examples suggest that, if pictures ψ and ψ' are equivalent, then ψ' can be obtained from ψ by a finite sequence of basic steps, each involving some scaling of rectangles and some translations. At this points, I don't see how to cast this intuition into a formal framework that would permit a formal proof.

6.5 Left and right sides of a closed path

Recall that in Section 3.4 we had difficulties with the concepts of inside and outside of a closed path. The ω-relations will enable us to define the opposite sides of a closed path. While this will not allow us to define inside and outside we can, nevertheless, use the notions of "opposite sides" as, in no realizable specification, can a rectangle be on both sides of a closed path.

When we traverse a closed path $\dot{\mathcal{P}}$, we can partition the set of all rectangles that are adjacent to the rectangles in $\dot{\mathcal{P}}$ into two classes: those to the left and those to the right of the path. Let us denote the classes as $\mathcal{L}_{\dot{\mathcal{P}}}$ and $\mathcal{R}_{\dot{\mathcal{P}}}$ respectively.

To begin, we restrict our attention to specifications with $\eta = \emptyset$ to reduce the number of possible cases. Consider an example where one has a closed path $\mathring{\mathcal{P}} = \langle r_0, \ldots, a, b, c, \ldots, r_0 \rangle$, and $(a, b), (c, b) \in \nu$. Then we have two possibilities, either $(a, c) \in \omega_\nu$ or $(c, a) \in \omega_\nu$. If $(a, c) \in \omega_\nu$, then for any rectangle r that is adjacent to b:

$$\begin{aligned} r &\in \mathcal{R}_{\mathring{\mathcal{P}}} && \text{if } (r, b) \in \nu \text{ and } (r, c), (a, r) \in \omega_\nu; \\ r &\in \mathcal{L}_{\mathring{\mathcal{P}}} && \text{otherwise.} \end{aligned}$$

In the former case we think of r being to the right of $\mathring{\mathcal{P}}$. In the latter, r would be to the left. On the other hand, if $(c, a) \in \omega_\nu$, then situation is reversed:

$$\begin{aligned} r &\in \mathcal{L}_{\mathring{\mathcal{P}}} && \text{if } (r, b) \in \nu \text{ and } (r, c), (a, r) \in \omega_\nu; \\ r &\in \mathcal{R}_{\mathring{\mathcal{P}}} && \text{otherwise.} \end{aligned}$$

This example is illustrated in Figure 6.8 in the entry labeled with the $*$. Rectangles that are placed at the black squares are to the right, whereas those placed at the white squares are to the left of the path. The complete figure illustrates *all* the right and left partitions for all possible orientations among a, b, and c.

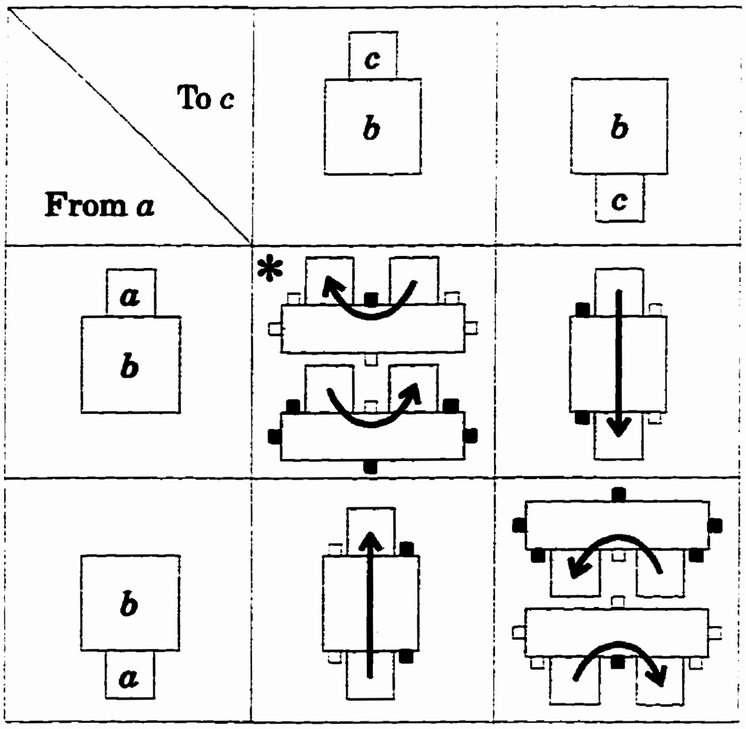


Figure 6.8: Left-adjacent (□) and right-adjacent (■) sides of a path with ν only.

Now we are going to lift the restriction of $\eta = \emptyset$ and look at specifications involving both ν and η. Consider a closed path $\mathring{\mathcal{P}} = \langle a.b.c.\ldots.a\rangle$. Again we start with an example, this time looking at the case when $(b.a) \in \nu$ and $(c.b) \in \eta$. Then for any rectangle r such that $(r,b) \in \nu \cup \nu^{-1} \cup \eta \cup \eta^{-1}$:

$r \in \mathcal{R}_{\mathring{\mathcal{P}}}$ if both $(b.r) \in \nu$ and $(r.a) \in \omega_\nu$ or both $(r.b) \in \eta$ and $(c.r) \in \omega_\eta$;

$r \in \mathcal{L}_{\mathring{\mathcal{P}}}$ otherwise.

This example is illustrated in Figure 6.9 in the entry labeled with the $*$. As before, rectangles that are placed at the black squares are to the right, whereas those placed at the white squares are to the left of the path. I define the left and right sides of a closed path formally in the next definition. Please refer to Figure 6.9. The number on the lower right corner of each entry in the figure corresponds to the part number used in the definition.

DEFINITION 6.5.1 [LEFT-ADJACENT AND RIGHT-ADJACENT SIDES OF A CLOSED PATH] *Suppose one has a closed path* $\mathring{\mathcal{P}} = \langle r_0.\ldots.a.b.c.\ldots.r_0\rangle$ *in an* ω*-specification. I denote the* left-adjacent *and* right-adjacent sides *of* $\mathring{\mathcal{P}}$ *as* $\mathcal{L}_{\mathring{\mathcal{P}}}$ *and* $\mathcal{R}_{\mathring{\mathcal{P}}}$. *Suppose one has a rectangle* d *such that* $(d.b) \in \nu \cup \nu^{-1} \cup \eta \cup \eta^{-1}$.

1. *The following applies for* $(\alpha,\beta,\delta,\gamma) = \{(a.b,c.d),(b,a,d,c)\}$.
   (a) *Suppose one has* $(\alpha,\beta) \in \nu$ *and* $(\delta,\beta) \in \eta$. *If both* $(\gamma.\beta) \in \nu$ *and* $(\gamma.\alpha) \in \omega_\nu$ *or both* $(\gamma,\beta) \in \eta$ *and* $(\gamma,\delta) \in \omega_\eta$ *then* $d \in \mathcal{L}_{\mathring{\mathcal{P}}}$, *otherwise* $d \in \mathcal{R}_{\mathring{\mathcal{P}}}$.
   (b) *Suppose one has* $(\beta,\delta) \in \nu$ *and* $(\alpha,\beta) \in \eta$. *If both* $(\beta,\gamma) \in \nu$ *and* $(\gamma.\delta) \in \omega_\nu$ *or both* $(\gamma,\beta) \in \eta$ *and* $(\alpha,\gamma) \in \omega_\eta$ *then* $d \in \mathcal{L}_{\mathring{\mathcal{P}}}$, *otherwise* $d \in \mathcal{R}_{\mathring{\mathcal{P}}}$.
2. *The following applies for* $(\alpha,\beta.\delta,\gamma) = \{(a,b.c.d).(b.a.d.c)\}$.
   (a) *Suppose one has* $(\alpha,\beta) \in \nu$ *and* $(\beta,\delta) \in \eta$. *If both* $(\gamma,\beta) \in \nu$ *and* $(\alpha.\gamma) \in \omega_\nu$ *or both* $(\beta.\gamma) \in \eta$ *and* $(\gamma,\delta) \in \omega_\eta$ *then* $d \in \mathcal{R}_{\mathring{\mathcal{P}}}$, *otherwise* $d \in \mathcal{L}_{\mathring{\mathcal{P}}}$.
   (b) *Suppose one has* $(\beta,\delta) \in \nu$ *and* $(\beta.\alpha) \in \eta$. *If both* $(\beta.\gamma) \in \nu$ *and* $(\delta.\gamma) \in \omega_\nu$ *or both* $(\beta,\gamma) \in \eta$ *and* $(\alpha.\gamma) \in \omega_\eta$ *then* $d \in \mathcal{R}_{\mathring{\mathcal{P}}}$, *otherwise* $d \in \mathcal{L}_{\mathring{\mathcal{P}}}$.
3. *The following applies for* $(\alpha,\beta.\delta) = \{(\nu.\mathcal{L}_{\mathring{\mathcal{P}}}.\mathcal{R}_{\mathring{\mathcal{P}}}).(\eta.\mathcal{R}_{\mathring{\mathcal{P}}}.\mathcal{L}_{\mathring{\mathcal{P}}})\}$.
   (a) *Suppose one has* $(a.b).(c.b) \in \alpha$.

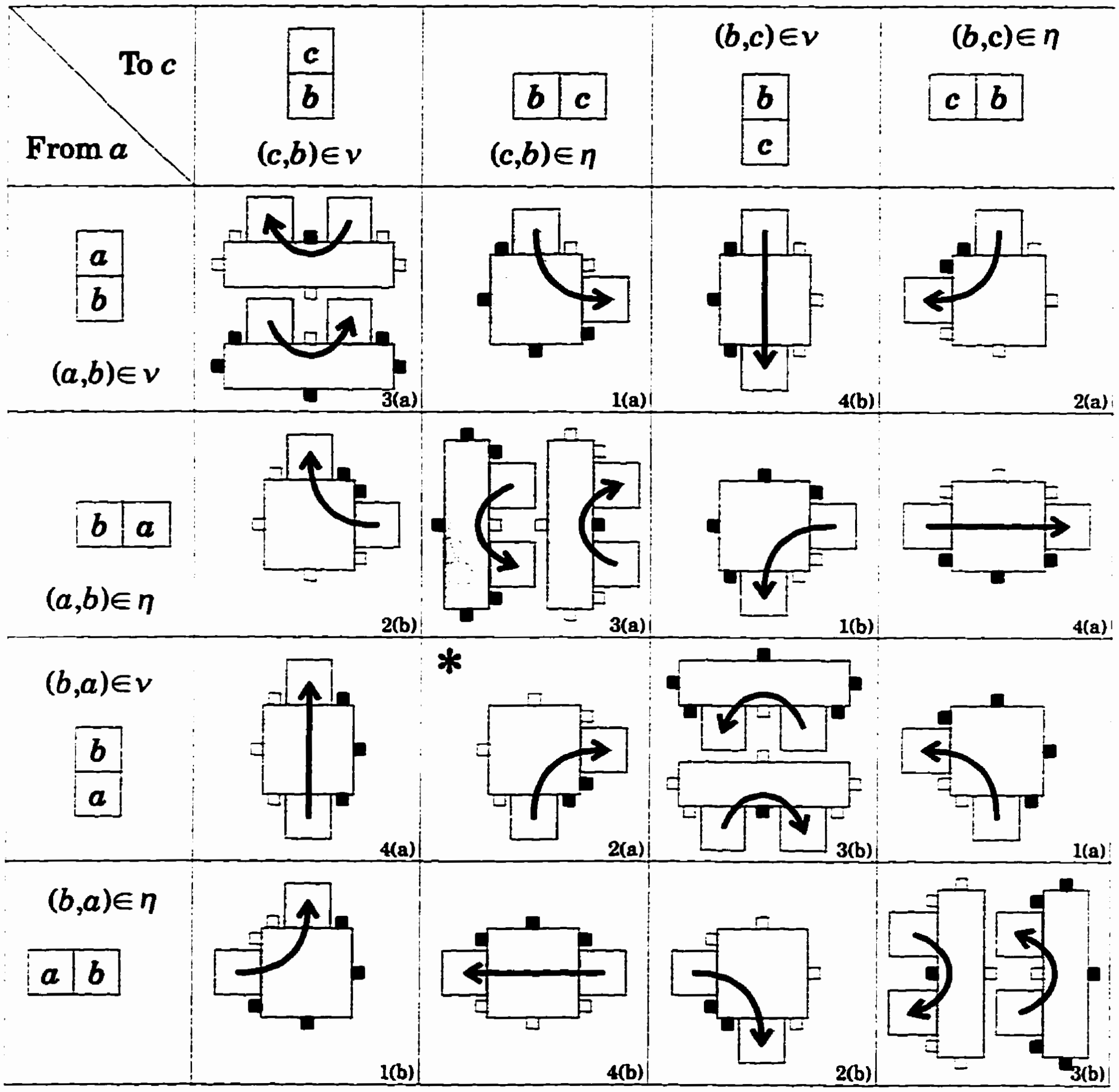


Figure 6.9: Defining the left-adjacent (□) and right-adjacent (■) of a closed path $\mathring{\mathcal{P}} = \langle a.b.c.\ldots.a \rangle$ with both ν and η.

i. *Suppose* $(a,c) \in \omega_\alpha$. *If* $(d,b) \in \alpha$ *and* $(a,d),(d,c) \in \omega_\alpha$ *then* $d \in \delta$, *otherwise* $d \in \beta$.

ii. *Suppose* $(c,a) \in \omega_\alpha$. *If* $(d,b) \in \alpha$ *and* $(c,d),(d,a) \in \omega_\alpha$ *then* $d \in \beta$, *otherwise* $d \in \delta$.

(b) *Suppose one has* $(b,a),(b,c) \in \alpha$.

i. *Suppose* $(a,c) \in \omega_\alpha$. *If* $(d,b) \in \alpha$ *and* $(a,d),(d,c) \in \omega_\alpha$ *then* $d \in \beta$, *otherwise* $d \in \delta$.

ii. *Suppose* $(c,a) \in \omega_\alpha$. *If* $(d,b) \in \alpha$ *and* $(c,d),(d,a) \in \omega_\alpha$ *then* $d \in \delta$, *otherwise* $d \in \beta$.

4. *The following applies for* $(\alpha,\beta,\delta,\gamma) = \{(\nu,\eta,\mathcal{L}_{\dot{\mathcal{P}}},\mathcal{R}_{\dot{\mathcal{P}}}),(\eta,\nu,\mathcal{R}_{\dot{\mathcal{P}}},\mathcal{L}_{\dot{\mathcal{P}}})\}$.

(a) *Suppose one has* $(c,b),(b,a) \in \alpha$. *If* $(b,d) \in \beta$ *or both* $(d,b) \in \alpha)$ *and* $(c,d) \in \omega_\alpha$ *or both* $(b,d) \in \alpha)$ *and* $(a,d) \in \omega_\alpha$ *then* $d \in \delta$, *otherwise* $d \in \gamma$.

(b) *Suppose one has* $(a,b),(b,c) \in \alpha$. *If* $(d,b) \in \beta$ *or both* $(d,b) \in \alpha)$ *and* $(d,a) \in \omega_\alpha$ *or both* $(d,b) \in \alpha)$ *and* $(d,c) \in \omega_\alpha$ *then* $d \in \delta$, *otherwise* $d \in \gamma$.

$\mathcal{L}_{\dot{\mathcal{P}}}$ and $\mathcal{R}_{\dot{\mathcal{P}}}$ contain rectangle symbols that are adjacent to a rectangle symbol in $\dot{\mathcal{P}}$. By intuition, we know that we should also be able to classify rectangles that are *connected* to a rectangle symbol in $\dot{\mathcal{P}}$. I extend the left-adjacent and right-adjacent definitions by the following.

DEFINITION 6.5.2 (LEFT AND RIGHT SIDES OF A CLOSED PATH) *Suppose one has a closed path* $\dot{\mathcal{P}}$ *in an* ω*-specification. For* $\zeta \in \{\mathcal{L}_{\dot{\mathcal{P}}},\mathcal{R}_{\dot{\mathcal{P}}}\}$, *if* $x \in \zeta$, *then for any* y *such that there is a path* $Q = \langle x = q_1,q_2,\ldots,q_n = y\rangle$, $n \geq 1$ *with* $q_i \notin \dot{\mathcal{P}}$ *for all* i, *one has* $y \in \zeta^\#$. *I call* $\mathcal{L}^\#_{\dot{\mathcal{P}}}$ *and* $\mathcal{R}^\#_{\dot{\mathcal{P}}}$ *the* left *and* right sides *of the closed path* $\dot{\mathcal{P}}$.

I am trying to capture the geometric notion of the left and right sides of a closed path by the above definitions using the abstract relations. It is known that one side is the *inside* and the other side is the *outside* of the path. However, one is unable to determine which is which. An analogy of this problem is when one has a closed continuous curve on a sphere. One is able to distinguish the points on either side of the curve, but the concept of inside and outside is not clear. The analogy is not complete as the path formed by rectangles has discrete steps. It does not have as much freedom of movement. But the consequence is similar that one is unable to discern the inside and the outside.

One of the reasons to pursue a definition of inside and outside is to enable us to express the condition that rectangles that are inside a closed path cannot be adjacent to rectangles that are outside. That condition can be replaced by using the left and right sides of a closed path without any loss of generality.

In the following, we will show the soundness of $\mathcal{L}$, $\mathcal{R}$, $\mathcal{L}^{\#}$, and $\mathcal{R}^{\#}$. First we show that a rectangle symbol cannot be simultaneously left-adjacent and right-adjacent to a closed path in a realizable ω-specification.

PROPOSITION 6.5.3 *For all closed paths $\dot{\mathcal{P}}$ in a realizable* ω*-specification* $\mathcal{S}_\omega = (\mathrm{R}, \nu, \eta, \omega_\nu, \omega_\eta)$, $\mathcal{L}_{\dot{\mathcal{P}}} \cap \mathcal{R}_{\dot{\mathcal{P}}} = \emptyset$.

Proof: Let us assume, on the contrary, that there exists a rectangle x such that $x \in \mathcal{L}_{\dot{\mathcal{P}}}$ and $x \in \mathcal{R}_{\dot{\mathcal{P}}}$. Let $\dot{\mathcal{P}} = \langle a, b, c, \ldots, a', b', c', \ldots, a \rangle$.

Assume that $x \in \mathcal{R}_{\dot{\mathcal{P}}}$ is decided by the adjacency between x and b, and $x \in \mathcal{L}_{\dot{\mathcal{P}}}$ is decided by the adjacency between x and b'.

It is clear that $b \neq b'$ because in all cases in the definition, it is not possible to have $x \in \mathcal{L}_{\dot{\mathcal{P}}}$ and $x \in \mathcal{R}_{\dot{\mathcal{P}}}$ when $b = b'$.

Since there is a realization, one has a picture $\psi = \alpha(\mathcal{S})$ for some arrangement α. Let us look at one possible realization. Figure 6.10 shows a case when rule 4(a) and rule 2(a) from Definition 6.5.1 are used to determine that $x \in \mathcal{R}_{\dot{\mathcal{P}}}$ and $x \in \mathcal{L}_{\dot{\mathcal{P}}}$ respectively. The rules determine the direction of $\mathcal{C}_{\dot{\mathcal{P}}}$ at b and b'.

Consider the connection curves $\pi(\dot{\mathcal{P}})$ for $\dot{\mathcal{P}}$ and $\pi(\mathcal{P})$ for the path $\mathcal{P} = \langle b, x, b' \rangle$. Since $x \notin \dot{\mathcal{P}}$, $\pi(\dot{\mathcal{P}})$ is not allowed to cross $\pi(\mathcal{P})$ and it is impossible to find a realization of $\dot{\mathcal{P}}$ such that its connection curve does not intersect itself. This is true for all cases of the rules used to determine $x \in \mathcal{R}_{\dot{\mathcal{P}}}$ and $x \in \mathcal{L}_{\dot{\mathcal{P}}}$. Since the arrangement was supposed to be realizable, one has arrived at a contradiction. Therefore the proposition is true. □

Now we show that a rectangle cannot be simultaneously to the left and to the right side of a closed path.

THEOREM 6.5.4 *Suppose an* ω*-specification* $\mathcal{S}_\omega$ *has a realizable arrangement* α. *For all closed paths* $\dot{\mathcal{P}}$ *in* $\mathcal{S}_\omega$, $\mathcal{L}^{\#}_{\dot{\mathcal{P}}} \cap \mathcal{R}^{\#}_{\dot{\mathcal{P}}} = \emptyset$. *I call this the* left-right property.

Proof: Let us assume, on the contrary, that there exists a rectangle x such that $x \in \mathcal{L}^{\#}_{\dot{\mathcal{P}}}$ and $x \in \mathcal{R}^{\#}_{\dot{\mathcal{P}}}$. This means that there exists a path $\mathcal{Q} = \langle q_0, q_1, \ldots, q_m = x \rangle$ such that $q_0 \in \mathcal{L}_{\dot{\mathcal{P}}}$, and $q_i \notin \dot{\mathcal{P}}$ for all i. Let $p_q \in \dot{\mathcal{P}}$ such that $(q_0, p_q) \in \nu \cup \nu^{-1} \cup \eta \cup$

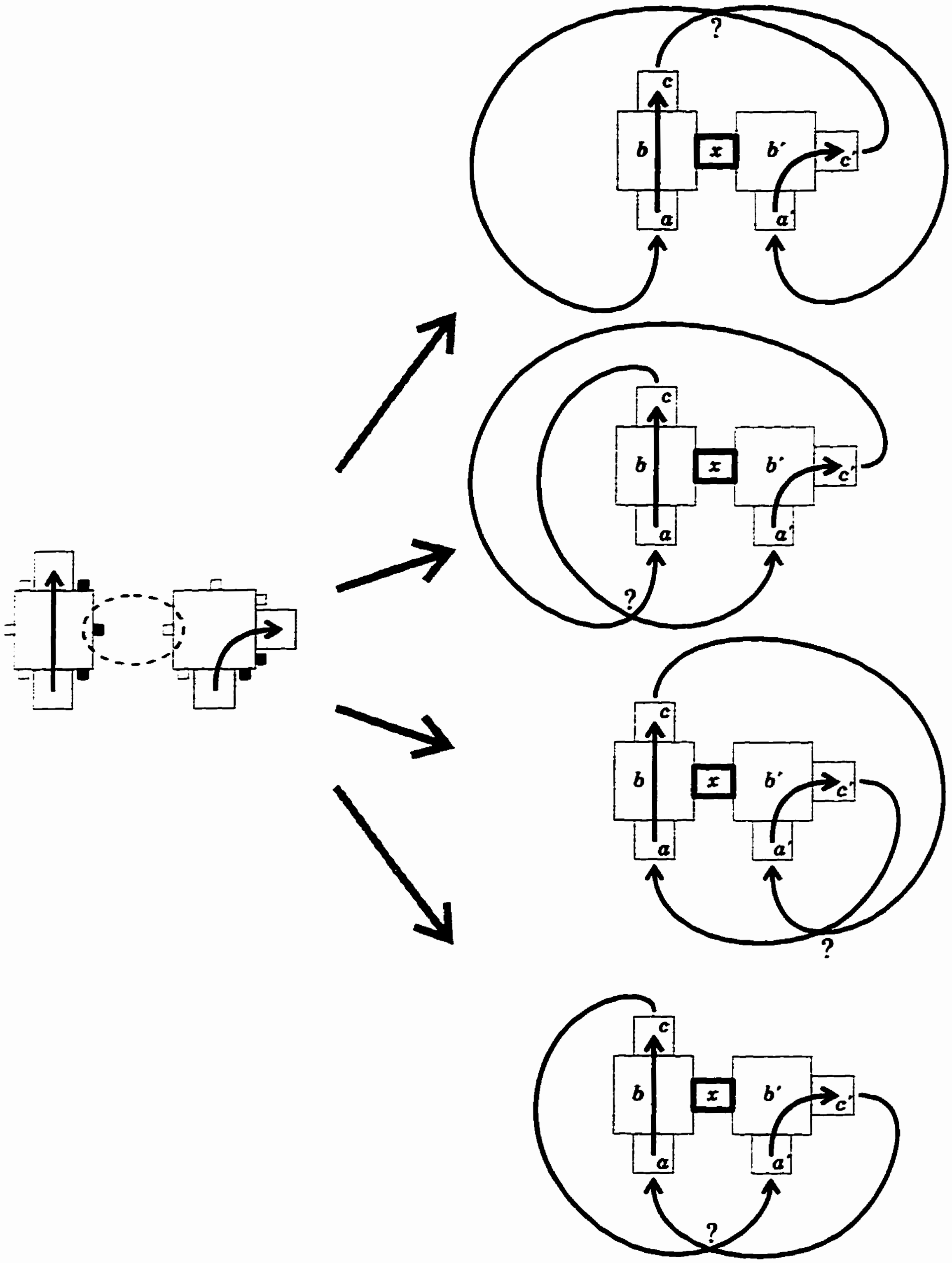


Figure 6.10: Example of x being both to the left and to the right of a closed path. It is not possible to realize the closed path without intersecting itself.

η^{-1}. There also exists another path $\mathcal{T} = \langle t_0, t_1, \ldots, t_n = x \rangle$ such that $t_0 \in \mathcal{R}_{\dot{\mathcal{P}}}$, and $t_j \notin \dot{\mathcal{P}}$ for all j. Let $p_t \in \dot{\mathcal{P}}$ such that $(t_0, p_t) \in \nu \cup \nu^{-1} \cup \eta \cup \eta^{-1}$.

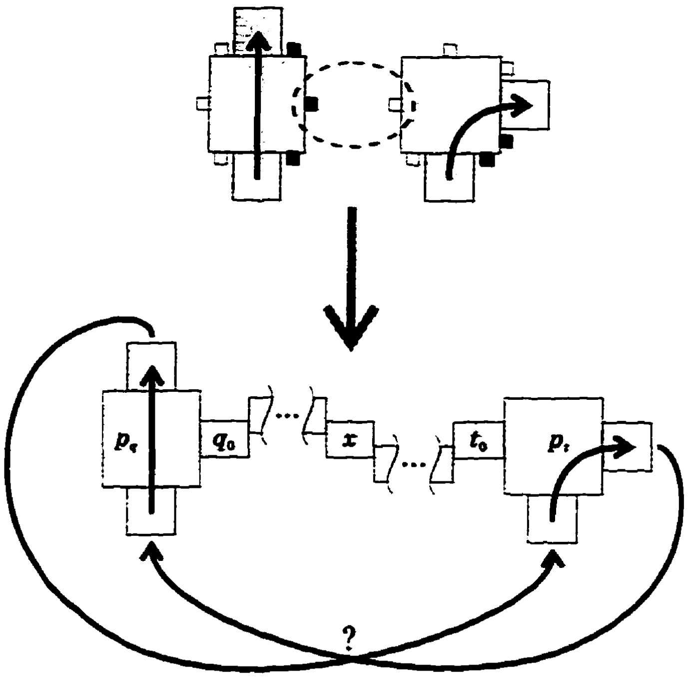


Figure 6.11: Example of $x \in \mathcal{L}^{\#}_{\dot{\mathcal{P}}} \cap \mathcal{R}^{\#}_{\dot{\mathcal{P}}}$. It is not possible to realize the closed path without intersecting itself (only one out of four possible closed paths is shown).

Consider the connection curves $\pi(\dot{\mathcal{P}})$ for $\dot{\mathcal{P}}$ and $\pi(\mathcal{P})$ for the path

$$\mathcal{P} = \langle p_q, q_0, q_1, \ldots, q_{m-1}, x, t_{n-1}, t_{n-2}, \ldots, t_0 \rangle.$$

The rules used to decide $q_0 \in \mathcal{L}_{\dot{\mathcal{P}}}$ and $t_0 \in \mathcal{R}_{\dot{\mathcal{P}}}$ determine the direction of $\pi(\dot{\mathcal{P}})$ at p_q and p_t. By definition $x, q_i, t_j \notin \dot{\mathcal{P}}$, therefore $\pi(\dot{\mathcal{P}})$ is not allowed to cross $\pi(\mathcal{P})$ and it is impossible to find a realization of $\dot{\mathcal{P}}$ such that its connection curve does not intersect itself. An example is shown in Figure 6.11.

This is true for all cases of the rules used to determine $q_0 \in \mathcal{L}_{\dot{\mathcal{P}}}$ and $t_0 \in \mathcal{R}_{\dot{\mathcal{P}}}$. Since the arrangement was supposed to be realizable, we have arrived at a contradiction. Therefore the theorem is true. □

PROPOSITION 6.5.5 *For any closed path $\dot{\mathcal{P}}$ in a realizable ω-specification $S_\omega = (\mathsf{R}, \nu, \eta, \omega_\nu, \omega_\eta)$, $\dot{\mathcal{P}}$, $\mathcal{L}^{\#}_{\dot{\mathcal{P}}}$ and $\mathcal{R}^{\#}_{\dot{\mathcal{P}}}$ are disjoint, and $\mathsf{R} = \dot{\mathcal{P}} \cup \mathcal{L}^{\#}_{\dot{\mathcal{P}}} \cup \mathcal{R}^{\#}_{\dot{\mathcal{P}}}$.*

Proof: $\dot{\mathcal{P}} \cap \mathcal{L}^{\#}_{\dot{\mathcal{P}}} = \mathcal{R}^{\#}_{\dot{\mathcal{P}}} \cap \dot{\mathcal{P}} = \emptyset$ is obtained directly from the definition of $\mathcal{L}^{\#}_{\dot{\mathcal{P}}}$ and $\mathcal{R}^{\#}_{\dot{\mathcal{P}}}$. $\mathcal{R}^{\#}_{\dot{\mathcal{P}}} \cap \mathcal{L}^{\#}_{\dot{\mathcal{P}}} = \emptyset$ is proven in Proposition 6.5.4. Thus $\dot{\mathcal{P}}$, $\mathcal{L}^{\#}_{\dot{\mathcal{P}}}$ and $\mathcal{R}^{\#}_{\dot{\mathcal{P}}}$ are disjoint.

Now we want to show that $\mathsf{R} = \mathring{\mathcal{P}} \cup \mathcal{L}^{\#}_{\mathring{\mathcal{P}}} \cup \mathcal{R}^{\#}_{\mathring{\mathcal{P}}}$. Consider a rectangle symbol $a \in \mathsf{R}$. Clearly if $a \in \mathring{\mathcal{P}}$ then the proposition is true. Now consider $a \notin \mathring{\mathcal{P}}$. It is assumed that all specifications are connected, therefore a is connected to some rectangle symbols in $\mathring{\mathcal{P}}$. By Definition 6.5.2 and Theorem 6.5.4, we know that either $a \in \mathcal{L}^{\#}_{\mathring{\mathcal{P}}}$ or $a \in \mathcal{R}^{\#}_{\mathring{\mathcal{P}}}$. Thus the proposition is true. □

Thus I have shown that the abstract definition of $\mathcal{L}^{\#}_{\mathring{\mathcal{P}}}$ and $\mathcal{R}^{\#}_{\mathring{\mathcal{P}}}$ captures the intuition of the left and right sides of a path in the geometric domain.

COROLLARY 6.5.6 *Let $\mathcal{S}_\omega = (\mathsf{R}, \nu, \eta, \omega_\nu, \omega_\eta)$ be an ω-specification. If there is a closed path $\mathring{\mathcal{P}}$ in $\mathcal{S}_\omega$ such that, for some $r \in \mathsf{R}$, $r \in \mathcal{L}^{\#}_{\mathring{\mathcal{P}}} \cap \mathcal{R}^{\#}_{\mathring{\mathcal{P}}}$ then $\mathcal{S}_\omega$ is not realizable.*

In Corollary 6.5.6 we capture the "cage" problems that escaped us before in Chapter 3. For example, in the cage situation of Figure 3.8 one has the closed path $\mathring{\mathcal{P}} = \langle a, b, e, g, f, d, a \rangle$. The rectangle symbol c is to the right of $\mathring{\mathcal{P}}$ and the symbols h, i, j, c are to the left. Thus, the symbol c is both to the right and the left of this path, hence the specification is unrealizable.

6.6 Local normalization

In this section, we present a technique called *local normalization* to replace a rectangle in a picture by a group of new rectangles. This can be useful as a proof technique in an induction proof, but it is not used further in this thesis.

We begin by defining the process of normalizing a rectangle symbol. We refer to Figure 6.12 and Figure 6.13 for the following definition.

DEFINITION 6.6.1 (LOCAL NORMALIZATION) *Consider an ω-specification $\mathcal{S}_\omega = (\mathsf{R}, \nu, \eta, \omega_\nu, \omega_\eta)$. A new ω-specification $\mathcal{S}'_\omega = (\mathsf{R}', \nu', \eta', \omega'_\nu, \omega'_\eta)$ is produced by replacing a rectangle symbol $x \in \mathsf{R}$ with a set of five new rectangle symbols x_T, x_B, x_R, x_L, x_C. The subscripts of the replacement rectangle symbols represent top, bottom, right, left and centre, respectively. Then*

$$\mathsf{R}' = \mathsf{R} \cup \{x_T, x_B, x_R, x_L, x_C\} \setminus \{x\}$$

ν' is defined by

$$\begin{aligned} \nu' &= \nu \cup \{(x_T, x_C), (x_C, x_B)\} \cup \{(a, x_T) \mid (a, x) \in \nu\} \\ &\quad \cup \{(x_T, b) \mid (b, x) \in \nu\} \setminus \{(a, x), (x, b) \mid a, b \in \mathsf{R}\} \end{aligned}$$

η' is defined analogously using η instead of ν.
Let

$$
\begin{aligned}
\omega'_\nu \;=\; & \omega_\nu \cup \left\{ (e,x_T),(x_T,f) \;\middle|\; \begin{array}{l} e,f \in \mathrm{R}, e \neq f, \textit{ with } (e,x),(x,f) \in \omega_\nu \\ \textit{and for some } y \in \mathrm{R} \\ (y,x) \textit{ and } (y,e) \in \nu \textit{ or } (y,f) \in \nu \end{array} \right\} \\
& \cup \left\{ (e,x_B),(x_B,f) \;\middle|\; \begin{array}{l} e,f \in \mathrm{R}, e \neq f, \textit{ with } (e,x),(x,f) \in \omega_\nu \\ \textit{and for some } y \in \mathrm{R} \\ (x,y) \textit{ and } (e,y) \in \nu \textit{ or } (f,y) \in \nu \end{array} \right\} \\
& \setminus \{(e,x),(x,f) \mid (e,x),(x,f) \in \omega_\nu\}.
\end{aligned}
$$

The definition of ω'_η is analogous using η instead of ν.

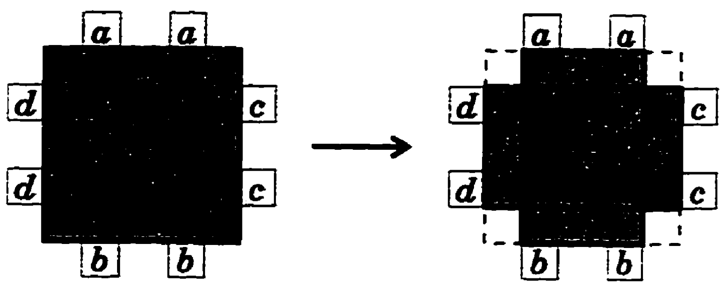


Figure 6.12: Rectangle replacement.

We need to specify how paths are affected by the normalization. For a path $\mathcal{P}$ in $\mathcal{S}_\omega$, we define $\mathcal{P}'$ to be the sequence of rectangle symbols obtained as follows:

1. If $\mathcal{P} = \langle \ldots, p_i, x, p_j, \ldots \rangle$, then

$$
\mathcal{P}' = \begin{cases} \langle \ldots, p_i, \beta, p_j, \ldots \rangle & \text{if } (p_i,x),(p_j,x) \in \gamma \quad \text{for } (\beta,\gamma) \in \\ & \quad \{(x_T,\nu),(x_B,\nu^{-1}),(x_R,\eta),(x_L,\eta^{-1})\} \\ \langle \ldots, p_i, \beta, x_C, \delta, p_j, \ldots \rangle & \text{otherwise, where} \\ & \quad \beta = \beta_0 \text{ if } (p_i,\beta_0) \in \gamma_1 \text{ for } (\beta_0,\gamma_1) \in \\ & \qquad \{(x_T,\nu),(x_B,\nu^{-1}),(x_R,\eta),(x_L,\eta^{-1})\} \\ & \quad \delta = \delta_0 \text{ if } (p_j,\delta_0) \in \gamma_2 \text{ for } (\delta_0,\gamma_2) \in \\ & \qquad \{(x_T,\nu),(x_B,\nu^{-1}),(x_R,\eta),(x_L,\eta^{-1})\} \end{cases}
$$

2. If $\mathcal{P} = \langle x, p_j, \ldots \rangle$, then

$$\mathcal{P}' = \langle x_C, \delta, p_j, \ldots \rangle.$$

where $\delta = \delta_0$ if $(p_j,\delta_0) \in \gamma_3$ for $(\delta_0,\gamma_3) \in \{(x_T,\nu),(x_B,\nu^{-1}),(x_R,\eta),(x_L,\eta^{-1})\}$.

3. If $\mathcal{P} = \langle \ldots, p_i, x \rangle$, then

$$\mathcal{P}' = \langle \ldots, p_i, \beta, x_C \rangle$$

where $\beta = \beta_0$ if $(p_i, \beta_0) \in \gamma_4$ for $(\beta_0, \gamma_4) \in \{(x_T, \nu), (x_B, \nu^{-1}), (x_R, \eta), (x_L, \eta^{-1})\}$.

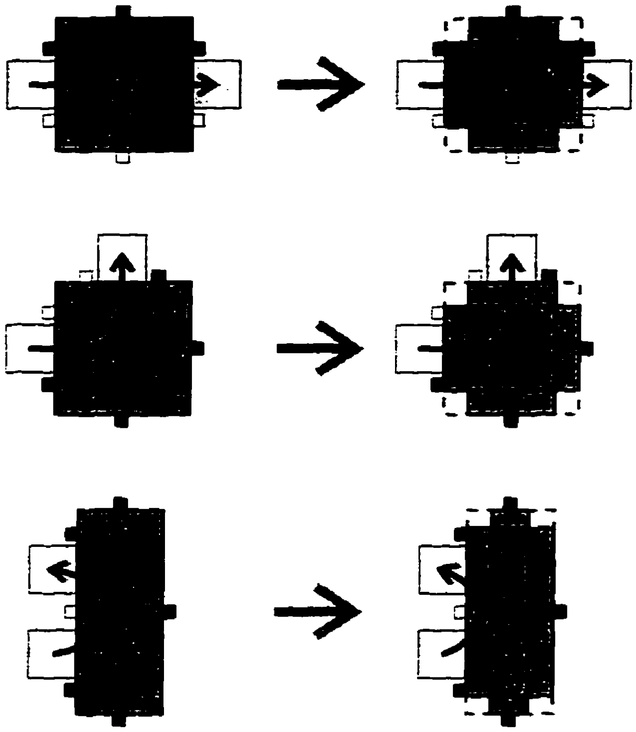


Figure 6.13: Examples of path redirection in rectangle replacement.

PROPOSITION 6.6.2 *$\mathcal{P}'$ is a path; it is closed if and only if $\mathcal{P}$ is closed.*

Proof: Those parts of $\mathcal{P}$ not involving x are the same in $\mathcal{P}'$. The steps reaching and leaving x are replaced by 4 steps via the new rectangles. Thus $\mathcal{P}'$ is a path. This assumes that x is not the first or last element of the path. If it is then the step leaving or reaching x, respectively, is replaced by two steps via the new rectangle symbols. The rest is obvious. □

PROPOSITION 6.6.3 *If $\mathcal{S}_\omega$ is realizable, then $\mathcal{S}'_\omega$ obtained by replacing a rectangle x in $\mathcal{S}_\omega$ for local normalization, is also realizable.*

Proof: We begin with a realizable ω-specification $\mathcal{S}_\omega$. If we look at the geometric realization, the replacement pattern for x can always be fitted into the space occupied by x by scaling the replacement pattern appropriately. All adjacent rectangles can also be attached correctly. To achieve this, imagine scaling x_C to be almost as large as x, leaving x_R, x_L, x_T, x_B to be very thin/narrow rectangles. No overlaps are introduced, therefore the conditions of a picture are still satisfied. This resulting picture is a realization of $\mathcal{S}'_\omega$. □

Next we want to show that local normalization does not affect the left and right sides of a closed path.

PROPOSITION 6.6.4 *Suppose one has an arrangement* α *of an* ω*-specification* $\mathcal{S}_\omega = (\mathsf{R}, \nu, \eta, \omega_\nu, \omega_\eta)$*. Now we perform a local normalization on a rectangle symbol* $x \in \mathsf{R}$ *to obtain* $\mathcal{S}'_\omega = (\mathsf{R}', \nu', \eta', \omega'_\nu, \omega'_\eta)$*. For a closed path* $\mathring{\mathcal{P}}$ *in* $\mathcal{S}_\omega$*, we denote the corresponding closed path in* $\mathcal{S}'_\omega$ *by* $\mathring{\mathcal{P}}'$*. Then for all rectangle symbols* $y \in \mathsf{R}$*,* $y \neq x$*, and all closed paths* $\mathring{\mathcal{P}}$ *in* $\mathcal{S}$*,*

$$y \in \mathcal{L}^{\#}_{\mathring{\mathcal{P}}} \Rightarrow y \in \mathcal{L}^{\#}_{\mathring{\mathcal{P}}'} \quad and \quad y \in \mathcal{R}^{\#}_{\mathring{\mathcal{P}}} \Rightarrow y \in \mathcal{R}^{\#}_{\mathring{\mathcal{P}}'}$$

Proof: The following proof applies for $(\zeta, \zeta') = \{(\mathcal{L}_{\mathring{\mathcal{P}}}, \mathcal{L}_{\mathring{\mathcal{P}}'}), (\mathcal{R}_{\mathring{\mathcal{P}}}, \mathcal{R}_{\mathring{\mathcal{P}}'})\}$.
Please refer to Figure 6.14 for this proof. If $y \in \zeta^{\#}$ then there exists a path $Q = \langle q_0, q_1, \ldots, q_n = y \rangle$ in $\mathcal{S}$ such that $q_0 \in \zeta$ and $q_i \notin \mathring{\mathcal{P}}$ for all i.

1. $x \notin \mathring{\mathcal{P}}$
   If $x \notin \mathring{\mathcal{P}}$, then $\mathring{\mathcal{P}}' = \mathring{\mathcal{P}}$ and $x_L, x_R, x_T, x_B, x_C \notin \mathring{\mathcal{P}}'$.

   (a) If $x \notin Q$, then $q_0 \in \zeta'$ and $q_i \notin \mathring{\mathcal{P}}$ for all i. (since q_0 and $\mathring{\mathcal{P}}$ are unchanged) Therefore $y \in \zeta'^{\#}$.

   (b) If $x \in Q$, then letting $x = q_j$, $j \neq 0$, one has

   $$Q' = \langle q_0, q_1, \ldots, q_{j-1}, x_1, q_{j+1}, \ldots, q_n \rangle$$

   or

   $$Q' = \langle q_0, q_1, \ldots, q_{j-1}, x_1, x_C, x_2, q_{j+1}, \ldots, q_n \rangle$$

   for $x_1, x_2 \in \{x_L, x_R, x_T, x_B\}$ as defined in the local normalization definition. Since $x_L, x_R, x_T, x_B, x_C \notin \mathring{\mathcal{P}}'$, one has $q'_i \notin \mathring{\mathcal{P}}'$ for all $q'_i \in Q'$. Also $q'_0 = q_0 \in \zeta \Rightarrow q'_0 \in \zeta'$. Therefore $y \in \zeta'^{\#}$.
   If $x = q_0$ then let x_q be the rectangle that replaced q_0 in the same position, where $x_q \in \{x_L, x_R, x_T, x_B, x_C\}$. Then $q'_0 = x_q \in \zeta \Rightarrow q'_0 \in \zeta'$. Therefore $y \in \zeta'^{\#}$.

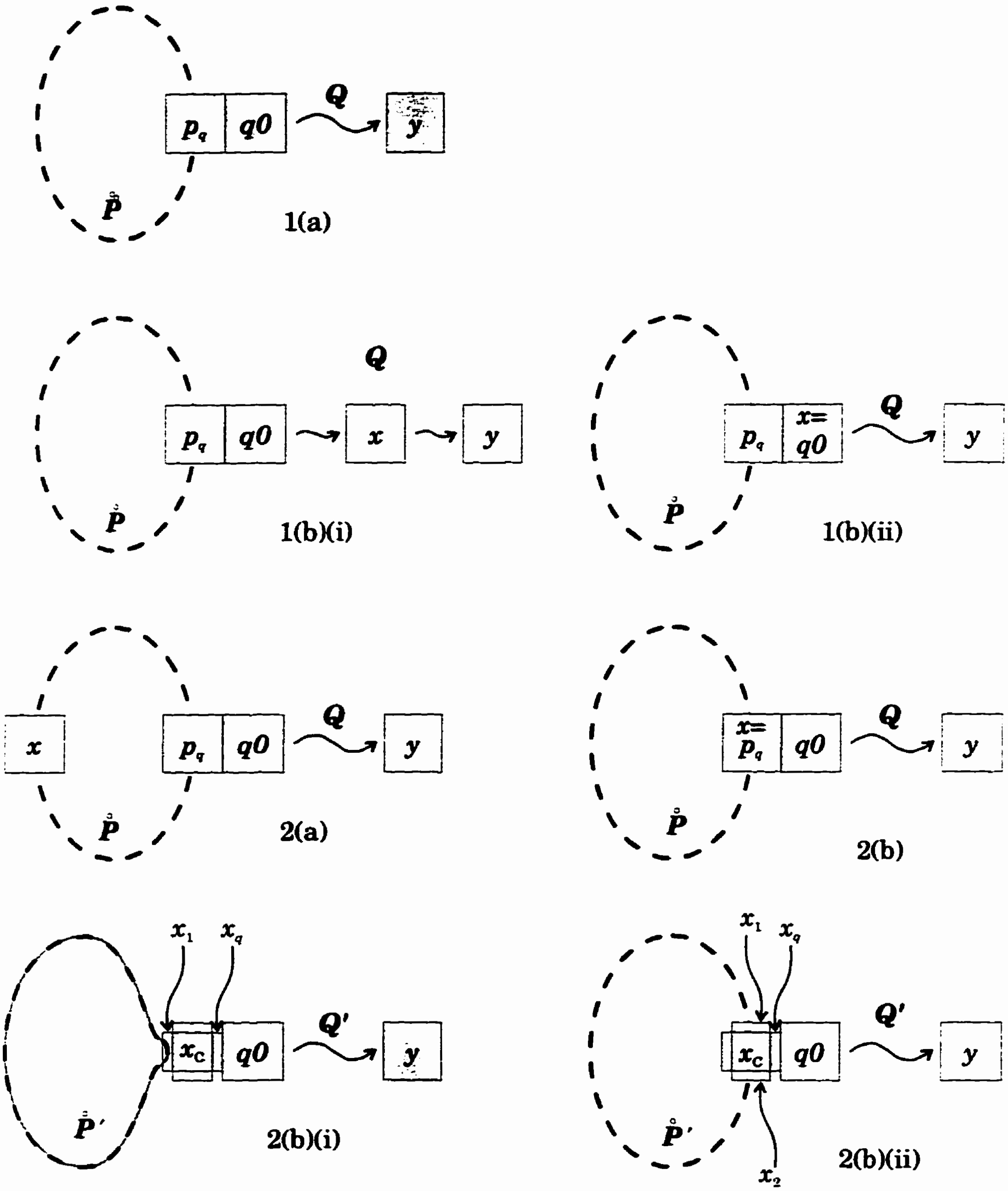


Figure 6.14: Cases for the proof of Proposition 6.6.4.

2. $x \in \dot{\mathcal{P}}$
Let $\dot{\mathcal{P}} = \langle p_0, p_1, \ldots, p_m = p_0 \rangle$. If $x = p_l$, one has

$$\dot{\mathcal{P}}' = \langle p_0, p_1, \ldots, p_{l-1}, x_1, p_{l+1}, \ldots, p_0 \rangle \tag{6.1}$$

or

$$\dot{\mathcal{P}}' = \langle p_0, p_1, \ldots, p_{l-1}, x_1, x_C, x_2, p_{l+1}, \ldots, p_0 \rangle \tag{6.2}$$

for $x_1, x_2 \in \{x_L, x_R, x_T, x_B\}$ as defined in the local normalization definition.
Since $q_0 \in \zeta$, $(q_0, p_q) \in \nu \cup \nu^{-1} \cup \eta \cup \eta^{-1}$ for some $0 \leq q \leq m-1$.

(a) If $x \neq p_q$ then $q_0 \in \zeta'$, since the same p_q is used. Since $x \in \dot{\mathcal{P}} \Rightarrow x \notin Q$, $Q' = Q$. Since $q_i \notin \dot{\mathcal{P}}$ and $q_i \neq x$ for all i, $q'_j \notin \dot{\mathcal{P}}'$. Therefore $y \in \zeta'^{\#}$.

(b) If $x = p_q$ then let x_q be the replacement rectangle, where $x_q = \{x_L, x_R, x_T, x_B, x_C\}$, such that $(x_q, q_0) \in \nu \cup \nu^{-1} \cup \eta \cup \eta^{-1}$.

   i. $\dot{\mathcal{P}}'$ is defined as in equation 6.1.
   This would mean that $(p_{l-1}, x), (p_{l+1}, x), (p_{l-1}, x_1), (p_{l+1}, x_1) \in \delta$ for $\delta = \{\nu, \nu^{-1}, \eta, \eta^{-1}\}$.
   If $(q_0, x) \in \delta$ then $x_q = x_1$ replaced x in $\dot{\mathcal{P}}'$. $q_0 \in \zeta \Rightarrow q_0 \in \zeta'$. Therefore $y \in \zeta'^{\#}$.
   If $(q_0, x) \notin \delta$, consider merging x_C with x_1 so that x_C is part of $\dot{\mathcal{P}}'$. Then the orientation between q_0 and x is the same as between x_q and x_c. Since $q_0 \in \zeta$, $x_q \in \zeta'$, and since q_0 is adjacent to x_q, $q_i \in \zeta'^{\#}$ for all i. Therefore $y \in \zeta'^{\#}$.

   ii. $\dot{\mathcal{P}}'$ is defined as in equation 6.2.
   If $x_q \in \dot{\mathcal{P}}'$ the orientation between q_0 and x is the same as between q_0 and x_q. Since $q_0 \in \zeta$, $q_0 \in \zeta'$, and therefore $y \in \zeta'^{\#}$.
   If $x_q \notin \dot{\mathcal{P}}'$ then the orientation between q_0 and x is the same as between x_q and x_C. Since $q_0 \in \zeta$, $x_q \in \zeta'$, and since q_0 is adjacent to x_q, $q_i \in \zeta'^{\#}$ for all i. Therefore $y \in \zeta'^{\#}$.

□

6.7 Extended compass relations

We wish to extract as much information as we can from the geometric compass relations. The geometric compass relations describe the relative orientation of touching rectangles. We are now going to explore the relative orientations of rectangles that do not touch by introducing four relations $\rho_{\bar{v}}$, $\rho_{\bar{\eta}}$, $\sigma_{\bar{v}}$, and $\sigma_{\bar{\eta}}$ on the geometric rectangles. Then we extract the abstract extended compass relations ρ_v, ρ_η, σ_v, and σ_η.

I introduce the relations $\rho_{\bar{v}_\psi}$, $\rho_{\bar{\eta}_\psi}$, $\sigma_{\bar{v}_\psi}$, and $\sigma_{\bar{\eta}_\psi}$ on the rectangles in a picture ψ by

$$\begin{array}{lll} (a,b) \in \rho_{\bar{v}_\psi} & \text{if and only if} & Y_b(a) > Y_t(b). \\ (a,b) \in \rho_{\bar{\eta}_\psi} & \text{if and only if} & X_l(a) > X_r(b). \\ (a,b) \in \sigma_{\bar{v}_\psi} & \text{if and only if} & Y_b(a) \geq Y_t(b). \\ (a,b) \in \sigma_{\bar{\eta}_\psi} & \text{if and only if} & X_l(a) \geq X_r(b). \end{array}$$

PROPOSITION 6.7.1 *The properties of* $\rho_{\bar{v}_\psi}$, $\rho_{\bar{\eta}_\psi}$, $\sigma_{\bar{v}_\psi}$, *and* $\sigma_{\bar{\eta}_\psi}$ *are:*

1. *There are no* $\rho_{\bar{v}_\psi}$*-cycles, no* $\rho_{\bar{\eta}_\psi}$*-cycles, no* $\sigma_{\bar{v}_\psi}$*-cycles, and* $\sigma_{\bar{\eta}_\psi}$*-cycles;*

2. $\rho_{\bar{v}_\psi}$, $\rho_{\bar{\eta}_\psi}$, $\sigma_{\bar{v}_\psi}$, *and* $\sigma_{\bar{\eta}_\psi}$ *are transitive.*

Proof:

I begin with the proof for $\sigma_{\bar{\eta}_\psi}$.

1. (No cycles) Let us assume, on the contrary, that there is a $\sigma_{\bar{\eta}_\psi}$-cycle such that
$$\langle (r_1,r_2),(r_2,r_3),\ldots,(r_{n-1},r_n),(r_n,r_1)\rangle \in \sigma_{\bar{\eta}_\psi}.$$
Then one has
$$X_l(r_1) \geq X_r(r_2) > X_l(r_2) \geq X_r(r_3) > \cdots > X_l(r_n) \geq X_r(r_1) > X_l(r_1).$$
a contradiction.

2. (Transitivity) Suppose $(a,b),(b,c) \in \sigma_{\bar{\eta}_\psi}$. Then one has $X_l(a) \geq X_r(b) > X_l(b) \geq X_r(c)$. Hence $(a,c) \in \rho_{\bar{\eta}_\psi}$.

Now I show the proof for $\rho_{\bar{\eta}_\psi}$. Since $\rho_{\bar{\eta}_\psi} \subseteq \sigma_{\bar{\eta}_\psi}$ there are no $\rho_{\bar{\eta}_\psi}$-cycles. Suppose $(a,b),(b,c) \in \rho_{\bar{\eta}_\psi}$. Then one has $X_l(a) > X_r(b) > X_l(b) > X_r(c)$. Hence $\rho_{\bar{\eta}_\psi}$ is transitive.

The proof for the other relations is obtained by symmetry. $\square$

6.7.1 The abstract relations ρ and σ

We now want to define the abstract counterparts to the extended geometric compass relations. In the following, I will define ρ_η and σ_η. As before, the definitions of ρ_ν and σ_ν and all the associated theorems and proofs can be obtained by symmetry. To improve the presentation of the definition, I will first introduce a set of conditions to be used in the definition.

DEFINITION 6.7.2 (BLOCKING CONDITIONS) *Consider an ω-specification $\mathcal{S}_\omega = (\mathsf{R}, \nu, \eta, \omega_\nu, \omega_\eta)$ and two rectangle symbols $\lambda, \delta \in \mathsf{R}$.*

1. (λ, δ) *satisfies the* blocking 1 condition *if there exist rectangle symbols $\gamma, \beta, \kappa \in \mathsf{R}$ such that*
   (a) $(\delta, \kappa), (\beta, \kappa), (\gamma, \beta), (\lambda, \beta) \in \nu$ *and*
   (b) $(\beta, \delta), (\lambda, \gamma) \in \omega_\nu$.
2. (λ, δ) *satisfies the* blocking 2 condition *if there exist integers m and n, and rectangle symbols $\gamma, \beta, \kappa, \phi, s_1, \ldots, s_m, t_1, \ldots, t_n \in \mathsf{R}$ such that*
   (a) $(\phi, s_1), (s_1, s_2), \ldots, (s_{m-1}, s_m), (s_m, \delta), (\delta, t_1), (t_1, t_2), \ldots, (t_{n-1}, t_n), (t_n, \kappa),$ $(\phi, \gamma), (\gamma, \beta), (\lambda, \beta), (\beta, \delta) \in \nu$ *and*
   (b) $(\beta, t_n), (\lambda, \gamma), (\gamma, s_1) \in \omega_\nu$.
3. (λ, δ) *satisfies the* blocking 3 condition *if there exists rectangle symbols $\gamma, \beta \in \mathsf{R}$ such that*
   (a) $(\lambda, \delta) \notin \eta$,
   (b) $(\gamma, \delta), (\beta, \delta) \in \eta$,
   (c) $(\lambda, \gamma) \in \nu$,
   (d) $(\beta, \gamma) \in \omega_\nu$.

(λ, δ) *satisfies the* blocking conditions *if it satisfies one of the blocking conditions above.*

DEFINITION 6.7.3 *Suppose one has an ω-specification $\mathcal{S}_\omega = (\mathsf{R}, \nu, \eta, \omega_\nu, \omega_\eta)$ and two rectangle symbols $\lambda, \delta \in \mathsf{R}$. Let σ be a binary relation on R. (λ, δ) satisfies the*

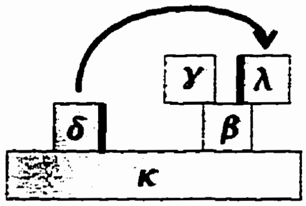


Figure 6.15: Blocking 1.

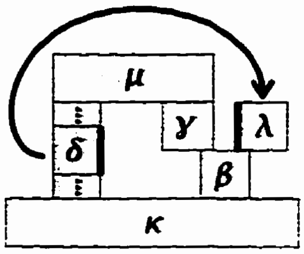


Figure 6.16: Blocking 2.

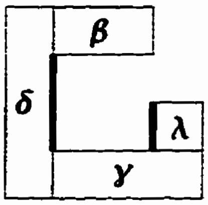


Figure 6.17: Blocking 3.

extended transitivity condition with respect to σ *if there exist rectangle symbols* $\beta, \gamma \in R$ *such that*

1. $(\alpha, \beta), (\gamma, \delta) \in \sigma$*; and*

2. $(\gamma, \beta) \in \nu \cup \nu^{-1} \cup \omega_\eta \cup \omega_\eta^{-1}$.

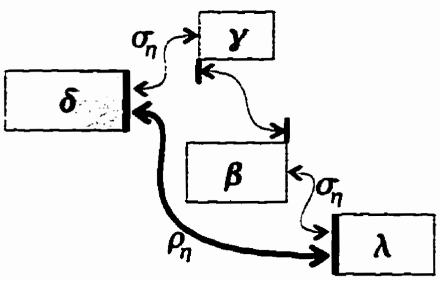


Figure 6.18: Extended transitive.

Now I will define ρ_η and σ_η using the above conditions. ρ_η and σ_η are defined simultaneously.

DEFINITION 6.7.4 (ρ AND σ) *For an ω-specification $\mathcal{S}_\omega = (\mathsf{R}, \nu, \eta, \omega_\nu, \omega_\eta)$, ρ_η and σ_η are defined as the smallest binary relations on R satisfying the following conditions:*

1. $(\eta^+ \cup \omega_\nu^+) \setminus \eta \subseteq \rho_\eta$ *and* $\eta^+ \cup \omega_\nu^+ \subseteq \sigma_\eta$.
2. *If* (λ, δ) *satisfies the blocking conditions for* ${}_1\mathcal{S}_\omega$ *or* ${}_3\mathcal{S}_\omega$*, then* $(\lambda, \delta) \in \rho_\eta$ *and* $(\lambda, \delta) \in \sigma_\eta$.

   If (λ, δ) *satisfies the blocking conditions for* ${}_2\mathcal{S}_\omega$ *or* ${}_4\mathcal{S}_\omega$*, then* $(\lambda, \delta) \in \rho_\eta^{-1}$ *and* $(\lambda, \delta) \in \sigma_\eta^{-1}$.
3. *If* (λ, δ) *satisfies the extended transitivity condition with respect to* σ_η *for* ${}_1\mathcal{S}_\omega$ *or* ${}_3\mathcal{S}_\omega$*, then* $(\lambda, \delta) \in \rho_\eta$ *and* $(\lambda, \delta) \in \sigma_\eta$.

   If (λ, δ) *satisfies the extended transitivity condition with respect to* σ_η *for* ${}_2\mathcal{S}_\omega$ *or* ${}_4\mathcal{S}_\omega$*, then* $(\lambda, \delta) \in \rho_\eta^{-1}$ *and* $(\lambda, \delta) \in \sigma_\eta^{-1}$.
4. $\rho_\eta^+ \subseteq \rho_\eta$ *and* $\sigma_\eta^+ \subseteq \sigma_\eta$.

PROPOSITION 6.7.5 *ρ_η and σ_η are well-defined and can be computed.*

Proof: The definition is recursive with (1) and (2) as base cases and (3) and (4) defining the recursion steps. □

PROPOSITION 6.7.6 *If an ω-specification $\mathcal{S}_\omega$ is realizable then, for any realization ψ, $\rho_\eta \subseteq \rho_{\bar{\eta}_\psi}$ and $\sigma_\eta \subseteq \sigma_{\bar{\eta}_\psi}$.*

Proof: Since $\mathcal{S}$ is realizable, there is an arrangement α such that $\psi = \alpha(\mathcal{S})$. Therefore $(\nu, \eta, \omega_\nu, \omega_\eta) = (\bar{\nu}_\psi, \bar{\eta}_\psi, \omega_{\bar{\nu}_\psi}, \omega_{\bar{\eta}_\psi})$ for some ω_ν and ω_η.

$\rho_\eta \subseteq \rho_{\bar{\eta}_\psi}$ and $\sigma_\eta \subseteq \rho_{\bar{\eta}_\psi}$ is proven simultaneously. The definitions of ρ_η and σ_η are recursive. I will prove the proposition by induction. Let $(\lambda, \delta) \in \rho_\eta$ and $(\lambda', \delta') \in \sigma_\eta$. It suffices to prove that $X_l(\lambda) > X_r(\delta)$ and $X_l(\lambda') \geq X_r(\delta')$. In the computation of ρ_η consider the step when (λ, δ) is first obtained and let $\hat{\rho}_\eta$ be the part of ρ_η constructed up to but not including this step. $\hat{\sigma}_\eta$ is also defined in the same way. Four cases are distinguished according to the condition by which (λ, δ) and (λ', δ') are added.

1. Suppose (λ, δ) is added to ρ_η because $(\lambda, \delta) \in (\eta^+ \cup \omega_\nu^+) \setminus \eta$. If $(\lambda, \delta) \in \eta^+ \setminus \eta$ then there are $\beta_1, \ldots, \beta_r$, $r \geq 1$, such that $(\lambda, \beta_1), (\beta_1, \beta_2), \ldots, (\beta_{r-1}, \beta_r), (\beta_r, \delta) \in$

η. Then $X_l(\lambda) = X_r(\beta_1) > X_l(\beta_1) = X_r(\beta_2) > \cdots > X_l(\beta_r) = X_r(\delta)$, hence $X_l(\lambda) > X_r(\delta)$.

If $(\lambda, \delta) \in \omega_\nu^+$ then by Lemma 6.2.5 one has $X_l(\lambda) \geq X_r(\delta)$. But since $(\lambda, \delta) \notin \eta$, $X_l(\lambda) \neq X_r(\delta)$. Hence one has $X_l(\lambda) > X_r(\delta)$.

Suppose (λ', δ') is added to σ_η because $(\lambda', \delta') \in \eta^+ \cup \omega_\nu^+$. If $(\lambda', \delta') \in \eta^+$ then there are $\beta_1, \ldots, \beta_r$, $r \geq 0$, such that $(\lambda', \beta_1), (\beta_1, \beta_2), \ldots, (\beta_{r-1}, \beta_r), (\beta_r, \delta') \in \eta$. Then $X_l(\lambda') = X_r(\beta_1) > X_l(\beta_1) = X_r(\beta_2) > \cdots > X_l(\beta_r) = X_r(\delta')$, hence $X_l(\lambda') \geq X_r(\delta')$.

If $(\lambda', \delta') \in \omega_\nu^+$ then by Lemma 6.2.5 one has $X_l(\lambda') \geq X_r(\delta')$.

2. Suppose (λ, δ) is added to ρ_η because of the blocking conditions.

   (a) If blocking 1 condition is satisfied, then there are rectangle symbols $\gamma, \beta, \kappa \in R$ such that $(\delta, \kappa), (\beta, \kappa), (\gamma, \beta), (\lambda, \beta) \in \nu$ and $(\beta, \delta), (\lambda, \gamma) \in \omega_\nu$. Since $(\beta, \delta), (\lambda, \gamma) \in \omega_\nu$, one has $X_l(\beta) \geq X_r(\delta)$ and $X_l(\lambda) \geq X_r(\gamma)$. Since $(\gamma, \beta) \in \nu$, one has $X_r(\gamma) > X_r(\beta)$. Therefore $X_l(\lambda) \geq X_r(\gamma) > X_r(\beta) \geq X_r(\delta)$.

   (b) If blocking 2 condition is satisfied, then there are rectangle symbols $\gamma, \beta, \kappa, \phi, s_1, \ldots, s_m, t_1, \ldots, t_n \in R$ such that

$$(\phi, s_1), (s_1, s_2), \ldots, (s_{m-1}, s_m), (s_m, \delta), (\delta, t_1), (t_1, t_2), \ldots, (t_{n-1}, t_n),$$
$$(t_n, \kappa), (\phi, \gamma), (\gamma, \beta), (\lambda, \beta), (\beta, \delta) \in \nu$$

   and

$$(\beta, t_n), (\lambda, \gamma), (\gamma, s_1) \in \omega_\nu.$$

   From $(\lambda, \gamma) \in \omega_\nu$ and $(\gamma, \beta) \in \nu$, one knows that $X_l(\lambda) \geq X_r(\gamma)$ and $X_r(\gamma) > X_l(\beta)$.

   Consider the LR cage from the bottom κ to the top ϕ. The two side walls are the paths $\langle t_j, \delta, s_i \rangle$ and $\langle \beta, \gamma \rangle$. One knows from $(\beta, t_n) \in \omega_\nu$ that $\langle t_j, \delta, s_i \rangle$ is the left wall (denoted by W_L) and $\langle \beta, \gamma \rangle$ is the right wall (denoted by W_R). Since overlaps are not allowed, one knows that for all $x \in W_L$ one cannot have both $X_l(\beta) < X_r(x)$ and $X_l(\gamma) < X_r(x)$. Hence one has $X_l(\beta) \geq X_r(\delta)$ or $X_l(\gamma) \geq X_r(\delta)$.

   Therefore one has either

$$X_l(\lambda) \geq X_r(\gamma) > X_l(\beta) \geq X_r(\delta)$$

or

$$X_l(\lambda) \geq X_r(\gamma) > X_l(\gamma) \geq X_r(\delta).$$

Hence $X_l(\lambda) > X_r(\delta)$.

(c) If blocking 3 condition is satisfied, then there are rectangle symbols $\gamma, \beta \in \mathsf{R}$ such that $(\lambda, \delta) \notin \eta$, $(\gamma, \delta), (\beta, \delta) \in \eta$, $(\lambda, \gamma) \in \nu$, and $(\beta, \gamma) \in \omega_\nu$.

From $(\gamma, \delta), (\beta, \delta) \in \eta$ one knows that $Y_t(\gamma) > Y_b(\delta)$ and $Y_b(\beta) < Y_t(\delta)$. According to $(\beta, \gamma) \in \omega_\nu$ one knows that $Y_b(\beta) \geq Y_t(\gamma)$. Therefore $Y_t(\delta) > Y_b(\beta) \geq Y_t(\gamma) > Y_b(\delta)$

From $(\lambda, \gamma) \in \nu$ one knows that $Y_b(\lambda) = Y_t(\gamma)$.

Now there are two cases, comparing $X_l(\lambda)$ and $X_r(\beta)$. If $X_l(\lambda) \geq X_r(\beta)$, then $X_l(\lambda) \geq X_r(\beta) > X_l(\beta) = X_r(\delta)$. On the other hand, if $X_l(\lambda) < X_r(\beta)$, then one has $Y_t(\lambda) < Y_b(\beta) < Y_t(\delta)$. Since overlapping is not allowed, one has $X_l(\lambda) \geq X_r(\delta)$. Since $(\lambda, \delta) \notin \eta$ one obtains $X_l(\lambda) > X_r(\delta)$.

Now suppose (λ', δ') is added to σ_η because of the blocking conditions. It is shown above that if (λ', δ') satisfies the blocking conditions, then $X_l(\lambda') > X_r(\delta')$.

3. and 4. Rules 1 and 2 form the basis for the induction. Now assume that $X_l(a) > X_r(b)$ if $(a, b) \in \hat{\rho}_\eta$. If (λ, δ) is added because of transitivity then there are $\beta_1, \ldots, \beta_r$ such that $(\lambda, \beta_1), (\beta_1, \beta_2), \ldots, (\beta_{r-1}, \beta_r), (\beta_r, \delta) \in \hat{\rho}_\eta$. By the induction assumption, $X_l(\lambda) > X_r(\beta_1) > X_l(\beta_1) > X_r(\beta_2) > \cdots > X_r(\delta)$, hence $X_l(\lambda) > X_r(\delta)$. On the other hand, if (λ, δ) is added because of extended transitivity then there are rectangle symbols β and γ such that $(\lambda, \beta), (\gamma, \delta) \in \hat{\rho}_\eta, (\gamma, \beta) \in \nu \cup \nu^{-1} \cup \omega_\eta \cup \omega_\eta^{-1}$. By the induction assumption, $X_l(\lambda) > X_r(\beta)$ and $X_l(\gamma) > X_r(\delta)$ and $X_r(\beta) > X_r(\gamma)$. Then one has $X_l(\lambda) > X_r(\delta)$, hence $(\lambda, \delta) \in \rho_{\bar{\eta}}$.

Now assume that $X_l(a) \geq X_r(b)$ if $(a, b) \in \hat{\sigma}_\eta$. If (λ', δ') is added because of transitivity then there are $\beta_1, \ldots, \beta_r$ such that

$$(\lambda', \beta_1), (\beta_1, \beta_2), \ldots, (\beta_{r-1}, \beta_r), (\beta_r, \delta') \in \hat{\sigma}_\eta.$$

By the induction assumption, $X_l(\lambda') \geq X_r(\beta_1) > X_l(\beta_1) \geq X_r(\beta_2) > \cdots \geq X_r(\delta')$, hence $X_l(\lambda') \geq X_r(\delta')$. On the other hand, if (λ', δ') is added because of extended transitivity then I have shown that $(\lambda', \delta') \in \rho_{\bar{\eta}}$, and hence $X_l(\lambda') \geq X_r(\delta')$.

□

An interesting observation is made with the case known as *Blocking 2*. It is modified slightly by inserting a new rectangle μ between β and κ. Then $(\lambda, \delta) \in \rho_{\bar{\eta}}$ is no longer true. An example exhibiting a violation is shown in Figure 6.19.

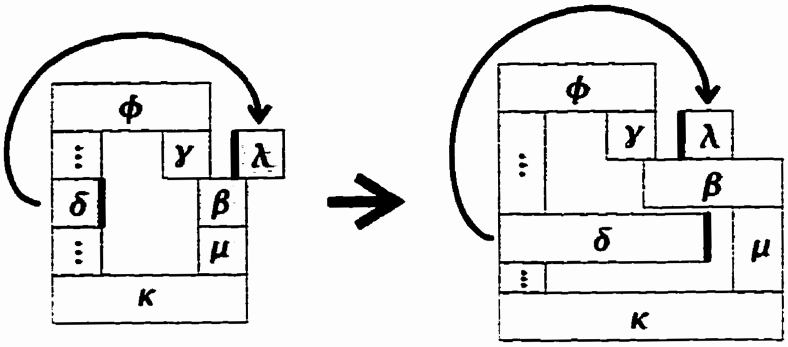


Figure 6.19: Blocking condition not included.

It is shown that for every realization, $\rho_\eta \subseteq \rho_{\bar{\eta}}$ and $\sigma_\eta \subseteq \sigma_{\bar{\eta}}$. But is $\rho_\eta \supseteq \rho_{\bar{\eta}}$ and $\sigma_\eta \supseteq \sigma_{\bar{\eta}}$? If it is true, then $\rho_\eta = \rho_{\bar{\eta}}$ and $\sigma_\eta = \sigma_{\bar{\eta}}$ for all realizations. This is clearly not possible. It is stated formally in the following.

PROPOSITION 6.7.7 *There is a realizable ω-specification $\mathcal{S}_\omega$ with a realization ψ such that $\rho_\eta \neq \rho_{\bar{\eta}_\psi}$ and $\sigma_\eta \neq \sigma_{\bar{\eta}_\psi}$.*

Proof: Consider the ω-specification $\mathcal{S}_\omega = \Big\{\{a,b,c\}, \{(a,b),(b,c)\}, \emptyset, \emptyset, \emptyset\Big\}$. Then $\psi = \alpha(\mathcal{S}_\omega)$ and $\psi' = \alpha'(\mathcal{S}_\omega)$ shown in Figure 6.20 are realizations of $\mathcal{S}_\omega$ such that $(c,a) \in \rho_{\bar{\eta}_\psi}$ and $(c,a) \notin \rho_{\bar{\eta}_{\psi'}}$

By Proposition 6.7.6, $\rho_\eta \subseteq \rho_{\bar{\eta}_\psi} \cap \rho_{\bar{\eta}_{\psi'}}$. Therefore $(c,a) \notin \rho_\eta$, hence $\rho_\eta \neq \rho_{\bar{\eta}_\psi}$. □

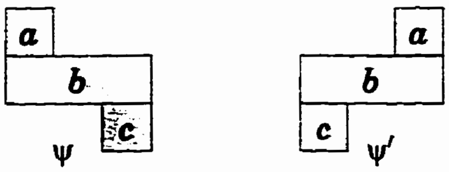


Figure 6.20: Examples for Proposition 6.7.7.

PROPOSITION 6.7.8 *The basic properties of the abstract extended compass relations ρ_ν, ρ_η, σ_ν, and σ_η are:*

1. *There are no ρ_ν-cycles, no ρ_η-cycles, no σ_ν-cycles, and no σ_η-cycles;*

2. ρ_ν, ρ_η, σ_ν, *and* σ_η *are transitive.*

Proof: It is shown in Proposition 6.7.1 that for any picture ψ, $\rho_{\bar{\nu}_\psi}$, $\rho_{\bar{\eta}_\psi}$, $\sigma_{\bar{\nu}_\psi}$, and $\sigma_{\bar{\eta}_\psi}$ have the above properties. According to Proposition 6.7.6, $\rho_\nu \subseteq \rho_{\bar{\nu}_\psi}$, $\rho_\eta \subseteq \rho_{\bar{\eta}_\psi}$, $\sigma_\nu \subseteq \sigma_{\bar{\nu}_\psi}$, and $\sigma_\eta \subseteq \sigma_{\bar{\eta}_\psi}$. Together with the fact that ρ_ν, ρ_η, σ_ν, and σ_η are transitive by definition, they have the three properties listed above. □

6.7.2 Cases not captured by σ

There are situations when one can conclude about East-West order which are not covered by σ_η. This is in contrast to the example shown in Proposition 6.7.7 where $\sigma_\eta \neq \sigma_{\bar{\eta}_\psi}$ because of some inherent ambiguity regarding the East-to-West order. I now describe a situation where $\sigma_\eta \neq \sigma_{\bar{\eta}_\psi}$ not because of such an ambiguity, but because σ_η fails to capture all cases.

DEFINITION 6.7.9 *Suppose one has a closed path* $\dot{\mathcal{P}} = \langle a_0, a_1, \ldots, a_n, a_0 \rangle$ *in an* ω-*specification* $\mathcal{S}_\omega$. *Suppose further that* $(a_0, a_i) \in \rho_\eta$ *for* $i = 2, \ldots, n-1$. *If there is no* ζ *such that* $(\zeta, a_0) \in \eta$, *then one introduces it as a new rectangle. If* $\zeta \in \mathcal{L}_{\dot{\mathcal{P}}}$ *then* $\mathcal{L}^\#_{\dot{\mathcal{P}}}$ *is the* outside *and* $\mathcal{R}^\#_{\dot{\mathcal{P}}}$ *is the* inside *of* $\dot{\mathcal{P}}$, *otherwise* $\mathcal{L}^\#_{\dot{\mathcal{P}}}$ *is the* inside *and* $\mathcal{R}^\#_{\dot{\mathcal{P}}}$ *is the* outside *of* $\dot{\mathcal{P}}$. *This definition is extended to the symmetry images* ${}_1\mathcal{S}_\omega$ *to* ${}_8\mathcal{S}_\omega$ *of* $\mathcal{S}_\omega$.

PROPOSITION 6.7.10 *Suppose one has a closed path* $\dot{\mathcal{P}} = \langle a_0, a_1, \ldots, a_n, a_0 \rangle$ *in a realizable* ω-*specification* $\mathcal{S}_\omega$. *Suppose further that* $(a_0, a_i) \in \sigma_\eta$ *for* $i = 2, \ldots, n-1$. *Then for any realization* ψ *of* $\mathcal{S}_\omega$, $(a_0, x) \in \sigma_{\bar{\eta}_\psi}$ *for all rectangles* x *inside* $\dot{\mathcal{P}}$. *This is also true for the symmetry images* ${}_2\mathcal{S}_\omega$ *to* ${}_8\mathcal{S}_\omega$ *of* $\mathcal{S}_\omega$.

Proof: Since ψ is a realization of $\mathcal{S}_\omega$, $(a_0, a_i) \in \sigma_{\bar{\nu}_\psi}$ for $i = 2, \ldots, n-1$.

Since $(a_0, a_2), (a_0, a_{n-1}) \in \sigma_{\bar{\nu}_\psi}$, then $X_r(a_1) < X_l(a_0)$ and $X_r(a_n) < X_l(a_0)$. Thus $X_l(r_i) > X_r(r_i) > X_l(a_0)$ for $1 \leq i \leq n$. Since x is inside $\dot{\mathcal{P}}$, $X_r(x) \leq X_l(a_0)$. Hence $(a_0, x) \in \sigma_{\bar{\eta}_\psi} = \sigma_\eta$.

□

EXAMPLE 6.7.11 *Consider the* ω-*specification*

$$\mathcal{S}_\omega = \Big(\{\beta, \gamma, \delta, \lambda, \kappa, \zeta\}, \{(\beta, \lambda), (\lambda, \gamma)\}, \{(\kappa, \lambda), (\delta, \beta), (\delta, \gamma), (\zeta, \delta)\}, \{(\beta, \gamma)\}\Big).$$

Figure 6.21 shows a possible realization of $\mathcal{S}_\omega$. (δ, λ) *satisfies blocking 3 condition in Definition 6.7.2 and by Definition 6.7.4* $(\delta, \lambda) \in \rho_\eta$. *Now consider the closed* *path* $\mathring{\mathcal{P}} = \langle \delta . \beta . \lambda . \gamma \rangle$. *By Definition 6.5.1* $\zeta \in \mathcal{R}_{\mathring{\mathcal{P}}}$ *and* $\kappa \in \mathcal{L}_{\mathring{\mathcal{P}}}$. *By Definition 6.7.9* $\mathcal{R}^{\#}_{\mathring{\mathcal{P}}}$ *is the outside and* $\mathcal{L}^{\#}_{\mathring{\mathcal{P}}}$ *is the inside of* $\mathring{\mathcal{P}}$. *Then Proposition 6.7.10 states that* $(\delta . \kappa) \in \rho_\eta$.

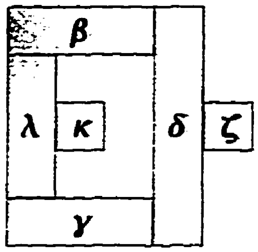


Figure 6.21: A realization of $\mathcal{S}_\omega$ in Example 6.7.11.

While I could have included the case of Proposition 6.7.10 in the definition of ρ_η, adding another case to the recursion part, I decided not to do so at this point as this case looks rather special — and its general abstract principle is not yet apparent. I still have the impression that this case is just an example of a more general situation in which East-to-West information can be deduced using cages and path directions.

6.7.3 Using ρ and η

PROPOSITION 6.7.12 *Suppose one has an* ω*-specification* $\mathcal{S}_\omega = (\mathsf{R}, \nu, \eta, \omega_\nu, \omega_\eta)$. *If* $\mathcal{S}_\omega$ *is realizable then* $(\rho_\nu \cup \rho_\nu^{-1} \cup \rho_\eta \cup \rho_\eta^{-1}) \cap (\nu \cup \nu^{-1} \cup \eta \cup \eta^{-1}) = \emptyset$.

Proof: Let ψ be a realization of $\mathcal{S}_\omega$. Suppose $(r_1, r_2) \in (\rho_\nu \cup \rho_\nu^{-1} \cup \rho_\eta \cup \rho_\eta^{-1})$ for some $r_1 . r_2 \in \mathsf{R}$. If $(r_1, r_2) \in \rho_\nu \cup \rho_\nu^{-1} \subseteq \rho_{\bar{\nu}_\psi} \cup \rho_{\bar{\nu}_\psi}^{-1}$, then $Y_b(r_1) > Y_t(r_2)$ or $Y_b(r_2) > Y_t(r_1)$. If $(r_1, r_2) \in \rho_\eta \cup \rho_\eta^{-1} \subseteq \rho_{\bar{\eta}_\psi} \cup \rho_{\bar{\eta}_\psi}^{-1}$, then $X_l(r_1) > X_r(r_2)$ or $X_l(r_2) > X_r(r_1)$. By Proposition 3.1.2, one has $(r_1, r_2) \notin \bar{\nu} \cup \bar{\nu}^{-1} \cup \bar{\eta} \cup \bar{\eta}^{-1} = \nu \cup \nu^{-1} \cup \eta \cup \eta^{-1}$ □

CHAPTER 7

REALIZABILITY

I have collected a series of conditions to determine the realizability of a specification. They are summerized in this chapter.

7.1 Necessary conditions for realizability

LEMMA 7.1.1 *Suppose one has a specification $\mathcal{S} = (\mathsf{R}, \nu, \eta)$. An ω-specification $\mathcal{S}_\omega$ is built for every possible ω_ν and ω_η. If one of $\mathcal{S}_\omega$ is realizable, then $\mathcal{S}$ is realizable.*

THEOREM 7.1.2 *A realizable ω-specification $\mathcal{S}_\omega = (\mathsf{R}, \nu, \eta, \omega_\nu, \omega_\eta)$ satisfies the following conditions:*

1. *ν and η satisfy the basic properties of the abstract compass relations. That is to say, ν and η have the following properties:*
   - *(a) There are no ν-cycles and no η-cycles;*
   - *(b) ν and η are anti-transitive;*
   - *(c) ν, ν^{-1}, η, and η^{-1} are pairwise disjoint.*
   - *(d) For $\beta = \{\nu, \eta\}$, if $(r_1, r_2) \in \omega_\beta$ then there is a $r_3 \in \mathsf{R}$ such that $(r_1, r_3), (r_2, r_3) \in \beta$ or $(r_3, r_1), (r_3, r_2) \in \beta$.*
2. *ω_ν and ω_η satisfy the basic properties of the abstract order relations. That is to say, ω_ν and ω_η have the following properties:*
   - *(a) There are no ω_ν-cycles and no ω_η cycles;*
   - *(b) ω_ν and ω_η are not transitive;*
   - *(c) ω_ν, ω_ν^{-1}, ω_η, and ω_η^{-1} are pairwise disjoint.*
3. *ω_ν^+ and ω_η^+ have the following properties:*

(a) *There are no ω_ν^+-cycles and no ω_η^+ cycles;*

(b) *ω_ν^+ and ω_η^+ are transitive;*

4. *ρ_ν, ρ_η, σ_ν, and σ_η satisfy the basic properties of the abstract extended compass relations. That is to say, ρ_ν, ρ_η, σ_ν, and σ_η have the following properties:*

   (a) *There are no ρ_ν-cycles, no ρ_η-cycles, no σ_ν-cycles, and σ_η-cycles;*

   (b) *ρ_ν, ρ_η, σ_ν, and σ_η are transitive.*

5. *S_ω satisfies the "left / right" property. That is to say, for all closed paths $\mathring{\mathcal{P}}$ in S_ω, one has $\mathcal{L}^\#_{\mathring{\mathcal{P}}} \cap \mathcal{R}^\#_{\mathring{\mathcal{P}}} = \emptyset$.*

6. *If one has a closed path $\mathring{\mathcal{P}} = \langle a_0, a_1, \ldots, a_0\rangle$ such that $(a_0, a_i) \in \rho_\eta$ for $i = 2, \ldots, n-1$, then $(a_0, x) \in \rho_\eta$ for all rectangles x inside $\mathring{\mathcal{P}}$. This is also true for ρ_ν and also for the symmetry images of S_ω.*

7. $(\rho_\nu \cup \rho_\nu^{-1} \cup \rho_\eta \cup \rho_\eta^{-1}) \cap (\nu \cup \nu^{-1} \cup \eta \cup \eta^{-1}) = \emptyset$.

Proof:

1. See Proposition 3.1.3.
2. See Proposition 6.3.3.
3. See Proposition 6.3.3.
4. See Proposition 6.7.8.
5. See Theorem 6.5.4.
6. See Proposition 6.7.10.
7. See Proposition 6.7.12.

□

CHAPTER 8

CONCLUDING REMARKS

8.1 Conclusion

- The rectangle picture specifications studied in this thesis was proposed to study semantic-based systems. It was never intended that rectangles alone would suffice in the layout of general scenes. The results obtained in this thesis, on its own, does not and was not meant to provide immediate solutions for problems that exist in non-semantic-based approaches, such as the problems discused in the section on motivation. Instead, this work explores the power and limitations of such systems in principle.

- A significant amount of geometrical information can be inferred from basic systems of rectangle picture specifications. The idea of road block conditions and the resulting relations ρ and σ capture some of the global conditions that must hold true for all realizations. This knowledge is obtained by the analysis of the the local information provided by the compass relations.

- To deal with the decidability problem, I have explored other approaches, including methods based on the Post Correspondence Problem, as well as methods based on graph grammars. They do not lead to any significant results.

- Apparently simple concepts like *inside or outside* of a cage are hard to capture; while there are some obvious simple cases for which sufficient conditions can be formulated for a rectangle symbol to be inside the cage in every realization, a general characterization of this property seems to be very difficult.

- Notions of *to-the-left-of* and *to-the-right-of* a path can be defined in such a way that, for closed paths, they meet the intuitive requirement that a rectangle cannot be both to the left and to the right of a closed path. The difficulty lies in the determination of the left and the right sides of a path in the context of rectangle picture specifications. Once the two sides are identified, the condition that they are disjoint is an application of the classical Jordan curve theorem.

- The cage situation, intuitively stated as *no rectangle can be attached to a rectangle inside the cage and also to a rectangle outside the cage*, can be re-expressed as *no rectangle can be both to the left and to the right of the closed path forming the cage.*

- The realizability problem for specifications cannot be solved by treating ν and η separately. The two relations must be considered jointly.

- A specification is realizable if and only if it is $\mathbf{R}_{\mathbb{Z}}$-realizable. Thus, the difficulty of the problem is not a result of the search space being uncountable; it is far more fundamental than this.

- If the realizations are restricted to rectangles of bounded size and bounded resolution then realizability is trivially decidable and bounds on the size of a bounding box for all realizations can be computed.

- The status of the general realizability problem or, equivalently, the $\mathbf{R}_{\mathbb{Z}}$-realizability problem is open. My feeling is, it is undecidable. Consider the picture given in Figure 8.1. Suppose it is a realization of some specification $\mathcal{S} = (\mathsf{R}, \nu, \eta)$. Let us follow the path $\langle a, b, c, \ldots, p\rangle$. One knows that a and p can be adjacent. However, if a rectangle somewhere along the path is modified slightly, then it may no longer be possible for a and p to be adjacent. The possible modifications include changes to the abstract compass relations and/or the abstract ordering relations. As an example, consider the case when one has $(m, l) \in \eta$ instead of $(l, m) \in \eta$. Then $\mathcal{S}$ is no longer realizable. As one can see from the example, there are many possibilities for modifying the path. It seems unlikely that one can find enough global rules or even a complete set of local rules to deal with all cases.

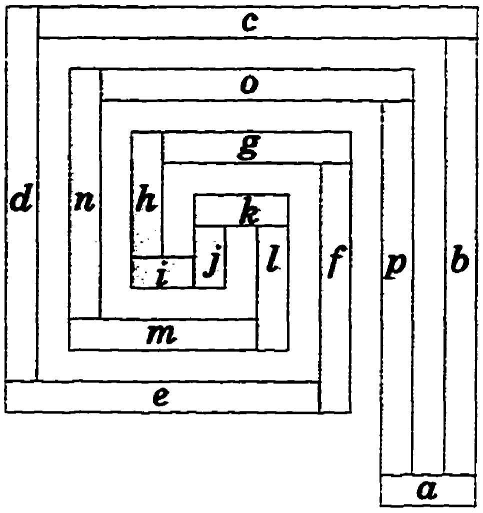


Figure 8.1: Difficult situation.

- For many cases, as was to be expected, a specification may have many non-equivalent realizations. Adding additional relational information to the specification may eliminate some of this ambiguity. Among the candidates, clockwise ordering often leads to inconsistencies. The ω-relations, while reducing ambiguity do not help regarding the decidability question.

- In Theorem 7.1.2 I list a set of necessary but not sufficient conditions for an ω-specification to be realizable. These conditions are decidable.

8.2 Further work

- Foremost, of course, the decidability of realizability has remained open. There are, however, several other unresolved issues.

- From the examples one has the impression that equivalent pictures can be obtained from each other using a finite sequence of elementary transformations involving only scaling and translation. This intuition needs to be cast into formal terms.

- A more uniform approach to the road block conditions needs to be found so as to arrive at a manageable set of sufficient conditions for realizability. So far, for any new type of conditions, I usually also find a case not covered. This could be an indicator of a lacking fundamental insight or it could point to undecidability of the realization problem.

In summary, specifications and more so ω-specifications, provide a large amount of geometrical information that can be extracted easily by algorithms. They are, however, too weak to be usable on their own for scene specifications and would have to be supplemented by additional, preferably "orthogonal", information.

REFERENCES

[1] R. G. Arrabito: Computerized Braille typesetting: Some recommendations on mark-up and Braille standards. Master's thesis, The University of Western Ontario, London, Canada, 1990.

[2] D. H. Ballard: *Computer Vision.* Prentice-Hall, 1982.

[3] G. D. Battista, P. Eades, R. Tamassia, I. G. Tollis. Algorithms for drawing graphs: An annotated bibliography, 1993.

[4] J. Beck, B. Hope, A. Rosenfeld (editors): *Human and Machine Vision.* Academic Press, 1983.

[5] S. Bhansali, G. A. Kramer: Planning from first principles for geometric constraint satisfaction. In *Proceedings of the 12th National Conference on Artificial Intelligence.* Seattle, WA, 1994.

[6] A. Borning: The programming language aspects of thinglab, a constraint-oriented simulation laboratory. *ACM Trans. Prog. Lang. and Systems* **3**(4) (October 1981), 269–295.

[7] A. Borning, R. Duisberg: Constraint-based tools for building user interfaces. *ACM Trans. Graphics* **5**(4) (October 1986), 345–374.

[8] M. Brunkhart: Interactive geometric constraint systems. Master's thesis, University of California, Berkeley, USA, 1994.

[9] K. J. Chen, K. Y. Cheng: A structured design methodology for Chinese character fonts. In *Proceedings of the 1983 International Conference on Text Processing with a Large Character Set.* 47–52, Tokyo, 1983.

[10] C. M. Creveling: *Tolerance Design: A Handbook for Developing Optimal Specifications.* Addison-Wesley, 1997.

[11] B. W. Croft, H. R. Turtle: Retrieval of complex objects. In *Advances in Database Technology: Proceedings of the Third International conference on Extending Database Technology.* 217–219, Springer-Verlag, Berlin, 1992.

[12] S. Dunne, H. Jürgensen: Formatting specialized notations. In *Proceedings of WOODMAN'89: Workshop on Object-Oriented Document Manipulation.* Rennes, France, 1989.

[13] S. Dunne: Why TeX should NOT output PostScript—yet. *TUGboat* **9**(1) (April 1988), 37–39.

[14] S. Dunne: Why TeX should NOT output PostScript—yet: Addendum. *TUGboat* **9**(2) (August 1988), 178.

[15] S. D. Dunne: A new paradigm for computer formatting of specialized notations. Master's thesis, The University of Western Ontario, London, Canada, 1988.

[16] S. D. Dunne: Syntax-directed interpretation of visual languages. Master's thesis, The University of Western Ontario, London, Canada, 1995.

[17] E. L. Fiume: *The Mathematical Structure of Raster Graphics.* Academic Press, Inc., 1989.

[18] B. Freeman-Benson, J. Maloney, A. Borning: An incremental constraint solver. *Communications of the ACM* **33**(1) (1990), 54–63.

[19] A. Garg, R. Tamassia: On the computational complexity of upward and rectilinear planarity testing. In *Proceedings Graph Drawing 1994.* 286–297, Princeton, USA, 1994.

[20] H. W. Guesgen, J. Hertzberg: *A Perspective of Constraint-Based Reasoning.* Springer-Verlag, 1992.

[21] G. Henzold: *Handbook of Geometrical Tolerancing: Design, Manufacturing and Inspection.* John Wiley and Sons Ltd., 1995.

[22] K. Hirata, T. Kato: Query by visual example – content based image retrieval. In *Advances in Database Technology: Proceedings of the Third International conference on Extending Database Technology.* 56–71, Springer-Verlag, Berlin, 1992.

[23] H. Jürgensen. Tactile computer graphics. Manuscript, 1996, 48 pp.

[24] H. Jürgensen, H.-Y. Wong: Chinese character generator. In *Proceedings 1988 International Conference on Computer Processing of Chinese and Oriental Languages.* 298–302, Toronto, 1988.

[25] H. Jürgensen, D. Wood. Structured pictures. Manuscript, 1992, 27 pp.

[26] T. Kamada: *Visualizing Abstract Objects and Relations – A Constraint-Based Approach.* World Scientific, 1989.

[27] G. A. M. Kiech: Content based retrieval of multimedia objects. Master's thesis, The University of Western Ontario, London, Canada, 1993.

[28] D. E. Knuth: *TEX and METAFONT, New Directions in Typesetting.* Published jointly by the American Mathematical Society and Digital Press, 1979.

[29] M. Kurze: Rendering drawings for interactive haptic perception. *Proc. CHI 97* (March 1997), to appear.

[30] M. Kurze, E. Holmes: 3D concepts by the sighted, the blind and from the computer. *Proc. of the 5th International Conference on Computers Helping People with Special Needs* (July 1996), 551–556.

[31] J. C. Lange: *Design Dimensioning with Computer Graphic Applications.* Marcel Dekker, Inc., 1984.

[32] W. Leler: *Constraint Programming Languages, Their Specification and Generation.* Addison-Wesley, 1988.

[33] R. D. Maddux. Some algebras and algorithms for reasoning about time and space. Internal Paper, Department of Mathematics, Iowa State University, 1990.

[34] G. Nelson: Juno, a constraint-based graphics system. *SIGGRAPH Computer Graphics* **19**(3) (1985), 235–243.

[35] A. Rosenfeld: Connectivity in digital pictures. *Journal of the ACM* **17**(1) (January 1970), 146–160.

[36] A. Rosenfeld: Adjacency in digital pictures. *Information and Control* **26**(1) (September 1974), 24–33.

[37] A. Rosenfeld: A converse to the Jordan curve theorem for digital curves. *Information and Control* **29**(3) (November 1975), 292–293.

[38] A. Rosenfeld, A. C. Kak: *Digital Picture Processing*. Academic Press, Inc., 1976.

[39] O. W. Salomons: *Computer Support in the Design of Mechanical Products: Constraint Specification and Satisfaction in Feature Based Design for Manufacturing*. PhD thesis, University of Twente, Twente, The Netherlands, 1995.

[40] M. Sannella, J. Maloney, B. Freeman-Benson, A. Borning: Multi-way versus one-way constraints in user interfaces: Experience with the deltablue algorithm. *Software–Practice and Experience* **23**(5) (1993), 529–566.

[41] J.-C. Simon: *Patterns and Operators: The Foundations of Data Representation*. McGraw-Hill, 1986.

[42] H. Y. Song, C. Y. Suen: A survey on Chinese character generation with a proposed new method. In *Proceedings of the 1986 International Conference on Chinese Computing*. 421–425, 1986.

[43] A. Speiser: *Die Theorie der Gruppen von endlicher Ordnung, mit Anwendungen auf algebraische Zahlen und Gleichungen sowie auf die Krystallographie. Lehrbücher und Monographien aus dem Gebiete der exakten Wissenschaften, Mathematische Reihe* **22**. Birkhäuser, 4th ed., 1956.

[44] I. E. Sutherland: Sketchpad: A man-machine graphical communication system. *IFIPS Proc. of the Spring Joint Computer Conf.* (1963), 329–346.

[45] C. J. Van Wyk: A high-level language for specifying pictures. *ACM Transactions on Graphics* 1(2) (April 1982), 163–182.

[46] P. H. Winston (editor): *The Psychology of Computer Vision*. McGraw-Hill, 1975.

[47] H.-Y. Wong: Computerized typesetting of Chinese: A new proposal for the design and output of Chinese characters. Master's thesis, The University of Western Ontario, London, Canada, 1992.