A Logic-based Framework for Shape Representation[1]

José C. Damski and **John S. Gero**

Key Centre of Design Computing
Department of Architectural and Design Science
University of Sydney, NSW, 2006, Australia
fax: +61-2-3513031
email : {jose,john}@arch.su.edu.au

Abstract

Shapes represent a very important way with which we perceive and reason about the world. In this article we develop a logic-based framework to represent graphical shapes in two dimensions. Based on the concept of halfplanes this framework allows us to represent regions as predicates in logic. This representation is applied to demonstrate shape concepts associated with topology and emergence.

Key words: shape, logic representation, design.

1 Introduction

1.1 Shape and Design

Fundamental to any computer-aided design system is the need to represent the objects under consideration. When those objects are conceptual a strictly symbolic representation is sufficient. For example, the concept that a physical object has a particular behaviour which can be derived from it can be represented by algebraic expressions which represent the theory of the derivation of that behaviour from that object. However, much of the information our senses bring to us is visual and is shape oriented. Much of our knowledge about the physical world comes to us in the form of shape information. There has been considerable effort on developing representations for computer vision purposes. However, there are very few consistent symbolic representation of shape because of the difficulty in developing such a representation. There is clear evidence that visual information is somehow treated differently by the human brain to many other forms of information. Coherent visual information has long been treated as either part of geometry or part of topology. Both of these have been represented and manipulated mathematically. If we are to include shape in our formal computational systems we need better methods of representation than the mathematics of geometry and topology. The vast majority of shape representation schemes are concerned with the geometry and to a lesser extent with the internal topology of the shape. This is important since the object needs to be represented on the screen. Thus, they are particularly useful at that stage of designing when the object's shape is known. However, these representations pose severe limitations on designers when they want to use such

[1]To appear in *Computer-Aided Design*

a system at the early stages of designing when the object's shape is still being created. Further such representation systems do not readily lend themselves to some of the shape analysis tasks required by designers; tasks such as the determination of topological relationships between shapes and shape emergence. These tasks are often carried out using symbolic manipulations in addition to geometric ones. We need computationally manipulable representations which can fit into our other areas of symbolic representation.

There are two fundamental concerns when considering shape: depiction and semantics. Depiction is concerned with the representation of a shape, whilst semantics is concerned with its meaning within a defined context. In this paper we are concerned initially with depiction issues in terms of representation and leave shape semantics as applications using the representation. The remainder of the paper shows the beginnings of a consistent logic-based representation of shapes using the concept of the halfplane as its basis. We have chosen a logic-based representation because of the formal characteristics of logic and because it allows us to represent shapes and the knowledge about their manipulation in the same formalism. In addition, first-order logic has been well studied and its strengths and weaknesses are well known.

1.2 Shape Semantics

There is a wide variety of shape semantics possible. Of particular interest here are those which are difficult or even impossible to derive using standard geometric representations of shape. Two classes of shape semantics will be used as tests for the representation. They are: topological relationships between shapes and shape emergence.

Topological relationships between shapes refer to the qualitative positional relationships between shapes, these relationships often play an important role in design. Much of the formal approaches to topological relationships has been based on symbolic rather than algebraic representation of shape and the symbolic expressions used in determining the qualitative positional relationships between shapes. Examples of such relationships are shown in Figure 1.

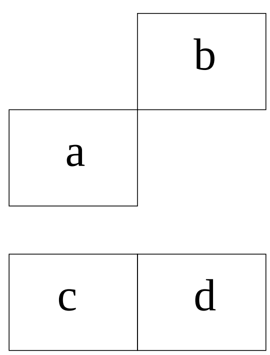
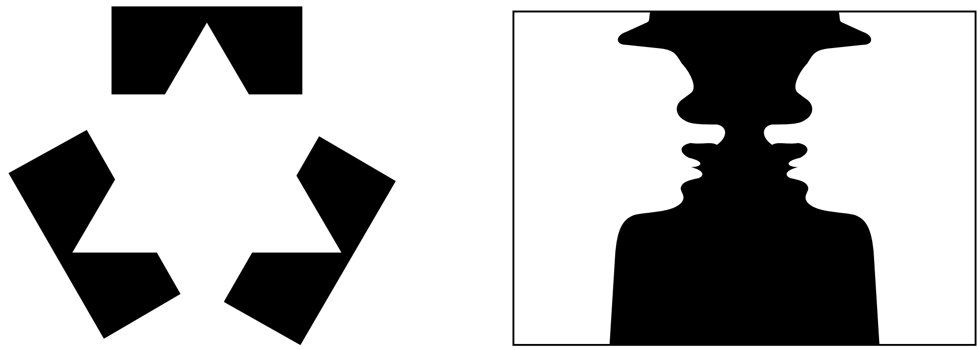


Figure 1: Examples of topological relationships between shapes: `a` *touches* `b`, `a` is *above* `c`, `d` is *adjacent* to `c`, `c` is *left of* `d`

Shape emergence is the process of discovering possible shapes that were not explicitly represented in the primary shape (Stiny 1980, Mitchell 1993). For example, in Figure 2(a) a star and many triangles can be discovered from the three initial shapes shown in black. This discovery is natural in human vision but very difficult to be reproduced in computers. Designers tend to use this additional knowledge as a new feature in the original design. In this example, the designer could keep the emerged triangle when the pieces were moved, for instance. Another example is shown in Figure 2(b) where two faces or a vase can be seen. The problem addressed in these examples can be tackled as a re-representantion of the primary shape, in computational terms. This re-representations allows us to re-interpret, and therefore have another perception of the primary shape.



(a) (b)

Figure 2: Example of shape emergence where the black shapes are the only ones drawn.

2 Representing Shapes using Logic

In this paper we define a logic formalism for representing shapes based on halfplanes. This formalism was briefly introduced in Damski & Gero (1993) using propositional logic. Here we extend that representation to first-order logic and show that it also applies to halfplanes with non-straight boundaries. We then develop some applications using this representation.

This representation is idealised to be a symbolic alternative to numerical representation used in CAD systems. The relation between these two paradigms of representation is shown in Figure 3 for the case when a new shape is deduced from the original one. This is an important aspects, because the logic representation needs to be 'grounded' in a numerical entity in order to be drawn.

2.1 Background

The concept of the halfplane (a similar concept is used in Giraud (1984) with a different formalisation) originated in our basic question about how to represent spaces symbolically without any reference to a particular coordinate system or any other numerical reference.

In this paper we define halfplane slightly differently to its definition in geometry. In geometry two halfplanes are divided by a line, and the points on that line do not belong to any of halfplanes. In our concept there is no line, only a conceptual border that divides two sets of points. Each set of points defines one halfplane. More formally:

- U is a region defined by a set of points $p(x, y)$.
  $U = \{p(x, y)\}$
- U always can be divided into exactly two subsets A and B, defined by:
  $A = \{p(x, y) : f(x, y) > 0\}$, and
  $B = \{p(x, y) : f(x, y) \leq 0\}$
  and $f(x, y)$ is a continuous function in U.

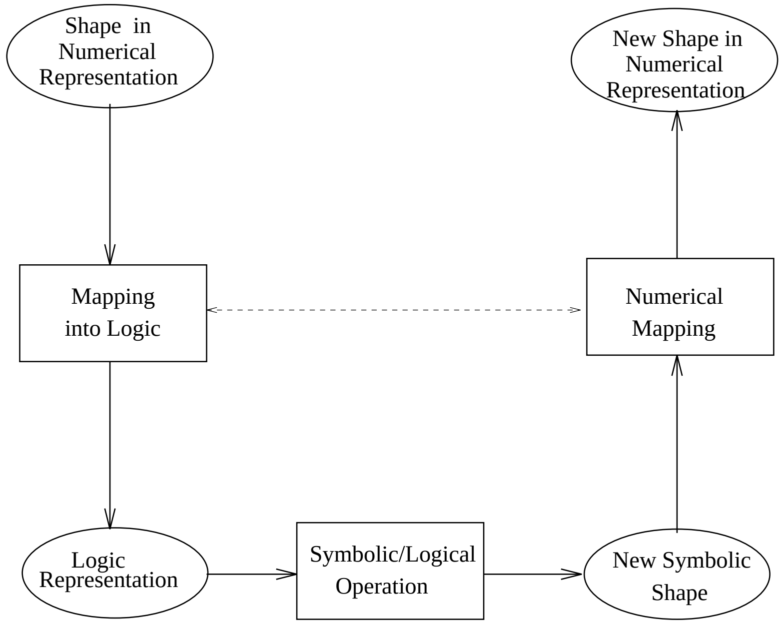


Figure 3: Connection between numerical and logical representations for the deduction of new shapes.

- A and B are non-empty, closed sets.

Therefore A and B have the following characteristics:

- $A \bigcap B$ is $\emptyset$, and
- $A \bigcup B$ is U

The set B is the *complement* of set A, denoted by A', where we only consider elements in U. Symbolically,

$$A\text{'} = \{p(x, y) : p(x, y) \notin A\}$$

Suppose another set C={T,F} defined by the function $g : A \rightarrow C$. The function g is:

$$g(x, y) = \begin{cases} \text{T} & \text{if } p(x, y) \in A \\ \text{F} & \text{if } p(x, y) \notin A \end{cases}$$

With this set C, we define the predicate $hp(x)$ with the following truth value:

- $hp(x)$ is True if T $\in C$ and False otherwise.

The predicate $hp(x)$ is a general representation of a halfplane, according to its truth value. For instance, in Figure 4 the halplane a is shown by the shaded area. Its complement, $\overline{a}$ is the unshaded area bounded by U and can be represented by $\neg hp(a)$, regardless of the specific truth value assigned to it.

By convention, we assign with truth value True to the halfplane where its name lies, as $hp(a)$ shows in Figure 4.

Given the sets $A = \{p(x, y)\}$ and $J = \{p(x, y)\}$, if $A \subset J$, the logical representation equivalent to this condition using the function g presented above, is:

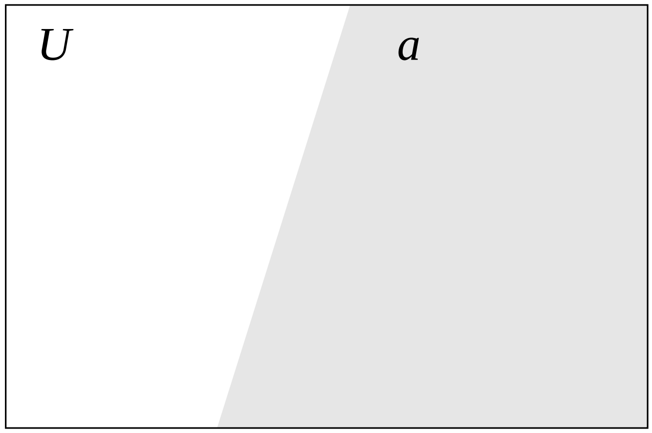


Figure 4: Halfplane a in U.

$$\boxed{hp(a) \rightarrow hp(j)}$$

where a and j are halfplanes for A and J.

According to these principles, all shapes expressed by halfplanes can be rewritten as a Wff (Well Formed Formula) in first order logic. The aim of this representation is to map shapes into halfplanes, and from them, into predicates to be manipulated according to first order logic principles.

2.2 Definitions

Constraint A *constraint* $\mathcal{C}$ is the set of logical formulas equivalent to a given topology.

Region Given n halfplanes, a *region* R is defined by a conjunctive formula of n $hp(x)$, as

$$R \text{ is } hp(a_1) \wedge hp(a_2) \wedge ... \wedge hp(a_n) \tag{1}$$

Since each halfplane can have truth value True or False, each *region* is an *interpretation* [2] of the formula 1. This means, for a given n halfplanes we have 2^n different regions.

Visibility A region R is said to be *visible* iff R has the truth value True under $\mathcal{C}$, i.e., $R \rightarrow \mathcal{C}$ is true for all interpretations of $hp(a_i)$. This means, of all possible regions, we have a special interest in those which are True under $\mathcal{C}$. The concept of visibility is important because we can distinguish what regions can be "seen" in a geometrical topology (if drawn) from all possible regions.

Minimal Description Given a region F defined by $hp(a_1) \wedge hp(a_2) \wedge ... \wedge hp(a_n)$, F has a *reduced* form F' if an $hp(a_i)$ were removed from F, and F and F' define the same region unambigously. If F can not be reduced further, F is said to be the *minimal description* and is represented by F_{min}. There always exists an F_{min} for a given F.
Graphically, F_{min} means a formula with only $hp(a_i)$ that bounds the region F.

Shape A shape is a disjunctive formula of regions.

In order to illustrate these definitions, Figure 5 shows three halfplanes a,b and c. The region R_1 is described by the formula:

[2] By *interpretation* we mean the assignment of truth value to each predicate in a formula.

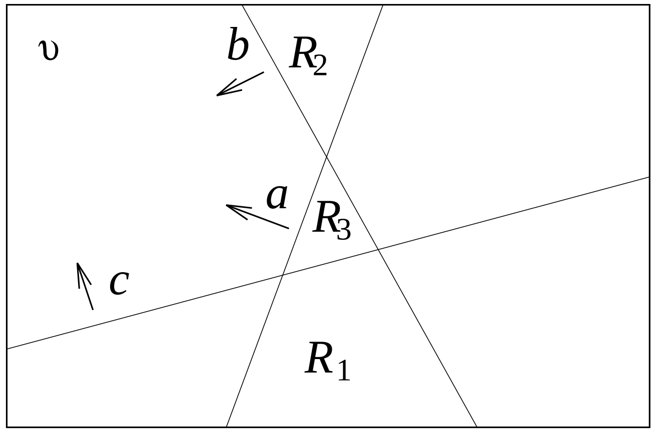


Figure 5: Halfplane a, b and c.

$$\neg hp(a) \wedge hp(b) \wedge \neg hp(c)$$

The regions R_1, R_2 and R_3 are examples of *visible* regions. Since we have three halfplanes, there are 8 (2^3) possible regions. In this particular example only 7 regions are visible. The regions not visible is:

$$hp(a) \wedge \neg hp(b) \wedge \neg hp(c)$$

The description for region R_2 is $hp(a) \wedge \neg hp(b) \wedge hp(c)$, but if the halfplane $hp(c)$ were removed from this formula, as $hp(a) \wedge \neg hp(b)$, it will represent the same region. The last formula is the *minimal description* for the region R_2 since it is not possible to remove any further halfplane. Note that the minimal description is composed of halfplanes that bound the region R_2. A shape is a combination (disjunctive normal formula) of regions. For example, a region S_1 can be $R_1 \vee R_3$, which can be expanded to:

$$(\neg hp(a) \wedge hp(b) \wedge \neg hp(c)) \vee (\neg hp(a) \wedge hp(b) \wedge hp(c))$$

which can be simplified further to:

$$\neg hp(a) \wedge hp(b)$$

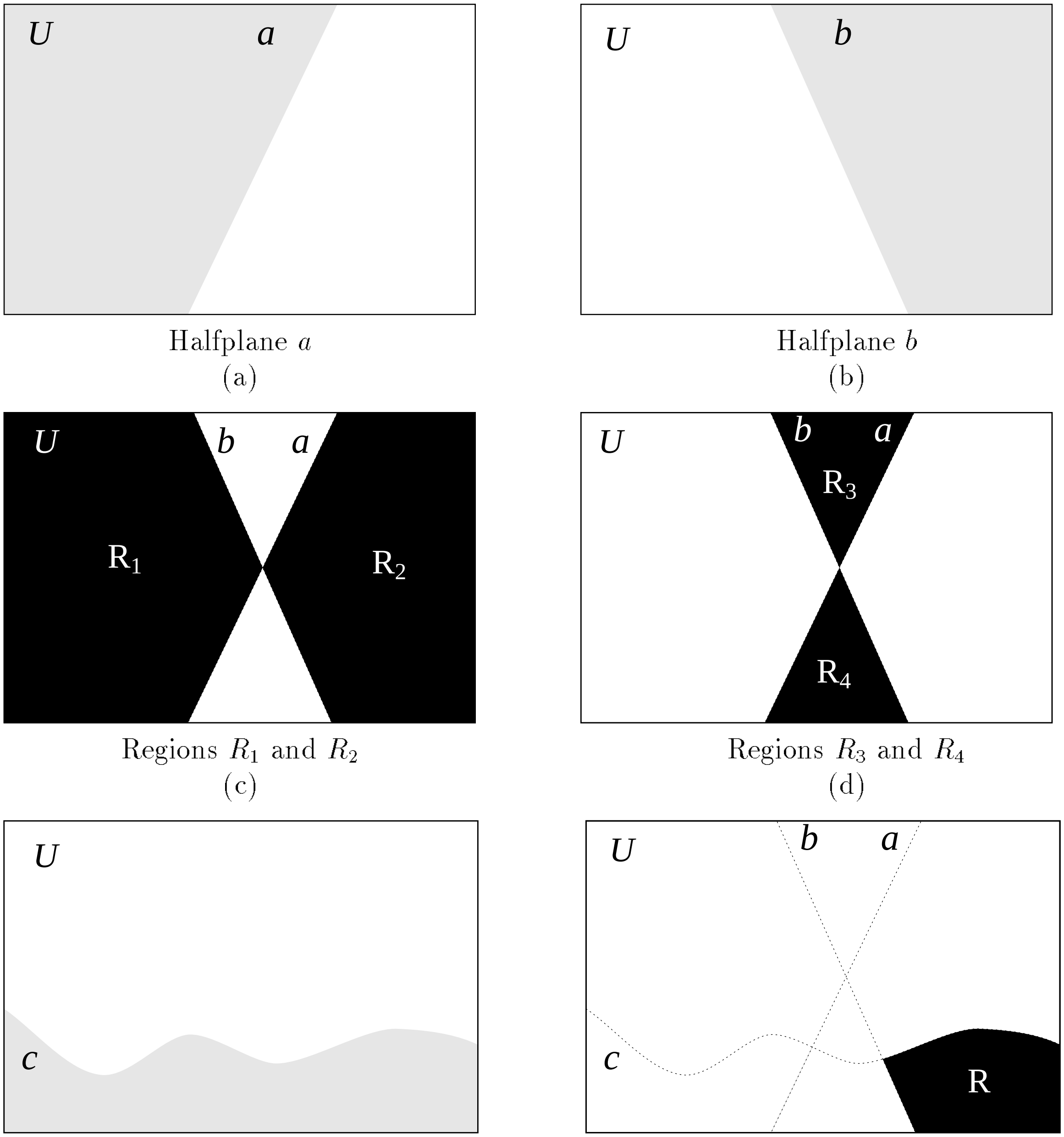
Finally, the *constraint* which describes this topology is given by the formula:

$$\neg hp(b) \wedge \neg hp(c) \rightarrow \neg hp(a)$$

The way to specify this constraint is shown in the following subsection.

2.3 Specifying a topology

Here we describe how to represent geometrical topologies using first-order logic, hereafter called *logic* for short. One advantage of using logic is its capacity to express additional knowledge about geometry in the same formalism. In addition, once we have all facts and production rules we can apply an inference process to determine desired results as theorem proving (Chang & Lee 1973). This will enable us to process any other knowledge about geometrical constraints using production rules. Initially it is necessary to represent a topology. For example, given two halfplanes a and b in U as shown in Figures 6(a) and 6(b), we can describe regions as intersections between a and b. There are four regions, which can be written using simple formulas:



Halfplane a
(a)

Halfplane b
(b)

Regions R_1 and R_2
(c)

Regions R_3 and R_4
(d)

Halfplane c
(e)

Region R
(f)

Figure 6: Possible topologies and regions

R_1 as $hp(a) \wedge \neg hp(b)$
R_2 as $\neg hp(a) \wedge hp(b)$
R_3 as $hp(a) \wedge hp(b)$
R_4 as $\neg hp(a) \wedge \neg hp(b)$

as shown in Figures 6(c) and (d).

Adding a third halfplane c as in Figure 6(e), we can generate a new possible topology as in Figure 6(f). In order to describe logically this topology it is necessary to describe the relation of a region to other halfplanes. If we choose a in Figure 6(a), the following formula is the definition $\mathcal{C}$ for this topology:

$$\boxed{hp(b) \wedge hp(c) \rightarrow \neg hp(a)} \qquad \text{Formula } F_1$$

because region R $(hp(b) \wedge hp(c))$ is completely inside the halfplane $\neg hp(a)$. As no other restrictions were applied, the truth table for F_1 is given in Table T_1.

$hp(a)$	$hp(b)$	$hp(c)$	F_1
True	True	True	False
True	True	False	True
True	False	True	True
True	False	False	True
False	True	True	True
False	True	False	True
False	False	True	True
False	False	False	True

Table T_1

From the truth Table T_1 we can infer that, given $hp(x)$ as non-empty halfplanes and F_1, all the following regions are non-empty:

$hp(a) \wedge hp(b) \wedge \neg hp(c)$;
$hp(a) \wedge \neg hp(b) \wedge hp(c)$;
$hp(a) \wedge \neg hp(b) \wedge \neg hp(c)$;
$\neg hp(a) \wedge hp(b) \wedge hp(c)$;
$\neg hp(a) \wedge hp(b) \wedge \neg hp(c)$;
$\neg hp(a) \wedge \neg hp(b) \wedge hp(c)$;
$\neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c)$.

As more restrictions are added to a description, the fewer non-empty regions we get. For example, in Figure 6(f) if we move the boundary of a to the right until it reaches where b crosses c, the region $\neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c)$ will be empty. As we already had F_1 as $hp(b) \wedge hp(c) \rightarrow \neg hp(a)$, now we have a new restriction, described by F_2, as:

$$\boxed{\neg hp(b) \wedge \neg hp(c) \rightarrow hp(a)} \qquad \text{Formula } F_2$$

Since F_1 and F_2 have to be true together, we have F_3 as $F_1 \wedge F_2$, as shown in Table T_2. In Table T_2, the formula $\neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c)$ now has the truth value False, which means it is an empty, or non-visible, region. These are important concepts which support shape emergence.

$$\boxed{(hp(b) \wedge hp(c) \rightarrow \neg hp(a)) \wedge (\neg hp(b) \wedge \neg hp(c) \rightarrow hp(a))} \qquad \text{Formula } F_3$$

$hp(a)$	$hp(b)$	$hp(c)$	F_3
True	True	True	False
True	True	False	True
True	False	True	True
True	False	False	True
False	True	True	True
False	True	False	True
False	False	True	True
False	False	False	False

Table T_2

2.4 Region Adjacency

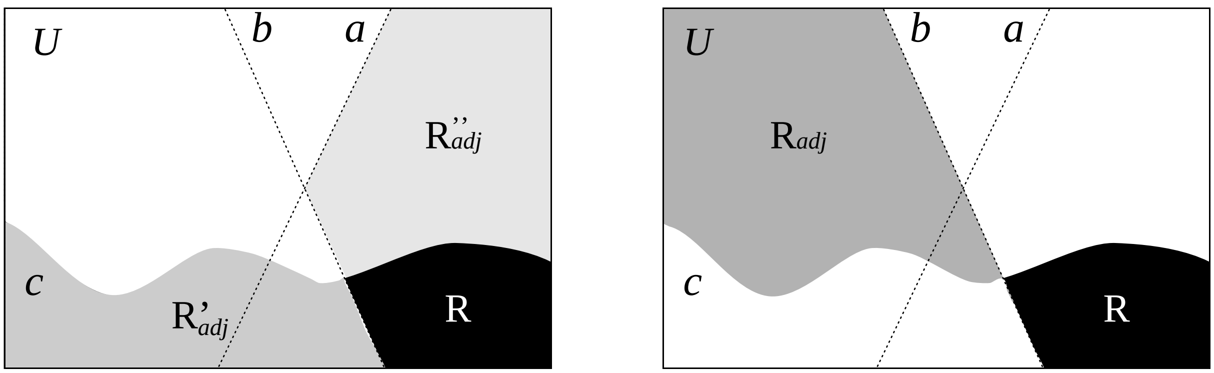
The region adjacent to any visible region R is defined as any region R' which graphically bounds R. There are two types of adjacency: border adjacency and corner adjacency. These are important concepts which support topological determinations.

2.4.1 Border Adjacency

Given a minimal description of a region R expressed by $hp(x_1) \wedge hp(x_2) \wedge ... \wedge hp(x_n)$, a region R_{adj} is **border adjacent** to R iff it differs in only one $hp(x_i)$, such as $hp(x_i)$ in R is $\neg hp(x_i)$ in R_{adj}.

For example, in Figure 6(f), the region R has the minimal description $hp(b) \wedge hp(c)$, therefore it has two border adjacent regions:

$$R'_{adj} \text{ is } \neg hp(b) \wedge hp(c)$$
$$R''_{adj} \text{ is } hp(b) \wedge \neg hp(c)$$



Regions R'_{adj} and R''_{adj} border adjacent to R
(a)

R_{adj} corner adjacent to R
(b)

Figure 7: Border and corner adjacency

as shown in Figure 7(a).

2.4.2 Corner Adjacency

Given a minimal description of a region R expressed by $hp(x_1) \wedge hp(x_2) \wedge ... \wedge hp(x_n)$. A region R_{adj} is **corner adjacent** to R iff it differs in exactly two literals $hp(x_i)$ and $hp(x_j)$, where $hp(x_i)$ and $hp(x_j)$ in R are $\neg hp(x_i)$ and $\neg hp(x_j)$ in R_{adj}.

For the example in Figure 6(f) the region R has the minimal description $hp(b) \wedge hp(c)$, therefore it has one corner adjacent region R_{adj}: $\neg hp(b) \wedge \neg hp(c)$, as shown in Figure 7(b).

2.5 Closed Region

For a given constraint $\mathcal{C}$ (see section 2.2), a region R expressed by $hp(a_1) \wedge hp(a_2) \wedge ... hp(a_n)$ is said to be **closed** iff:

- R is visible;

- for all $hp(a_i)$ in the region R there exists a border adjacent region R_{adj} which is *visible* under $\mathcal{C}$;
- all corner adjacent regions R_{adj} are visible under $\mathcal{C}$ for any two $hp(a_i)$ and $hp(a_j)$ in R; and
- R must have, at least, 3 different halfplanes $hp(a_1), hp(a_2)$ and $hp(a_3)$.

For example, in Figure 6(f) the region defined by $\neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c)$ is closed since it satisfies the definitions.

2.6 Embedability

Embedability is the property of a shape to be embedded into another shape, in a geometrical sense. Suppose two shapes S_1 and S_2 described as:

S_1 as $R_1 \vee R_2 \vee ... \vee R_n$
S_2 as $T_1 \vee T_2 \vee ... \vee T_m$

where R_i and T_j are regions. S_1 is said to be *embedded* in S_2 ($S_1 \subset S_2$) if the following expression is true:

$$\forall i \; \exists j \mid R_i = T_j$$

where $R_i = T_j$ means these regions are defined by the same halfplanes.

Similarly, S_1 *overlaps* S_2 if:

$$\exists i \; \exists j \mid R_i = T_j$$

Finally, S_1 is *disjoint* to S_2 if:

$$\forall i \; \nexists j \mid R_i = T_j$$

2.7 Relative position

As the mapping from a numeric to a logic representation does not carry any information on spatial relation among halfplanes, it is necessary to give a semantic denotation for each halfplane. This denotation can be giving by the following declaration

- The universe of discourse U is represented by a rectangle, whose boundaries are numbered from 1 to 4 in a clockwise direction, as shown in Figure 8.
- Each halfplane has its symbol and denotation according to the endpoints of its border. Figure 9 shows the possible combinations among sides, where the border of each halfplane is represented by a dotted line. Halfplane labels are consistently placed above or to the left of the boundary of the halfplane. Arbitrarily name **above** as the upwards direction and **left** as westwards direction.
- The denotation for each halfplane is shown in Table T_3, assuming Left(L) as opposite direction of Right(R) and Above(A) as opposite direction of Below (B).
- The name of each halfplane is given to the side its denotation is, as shown in the example Figure 9.

Halfplane	Begin-end	Denotation	Abbreviation
a	1-1	Left	L
b	1-2	Above-Left	AL
c	1-3	Above	A
d	1-4	Above-Right	AR
e	2-2	Above	A
f	2-3	Above-Right	AR
g	2-4	Left	L
h	3-3	Left	L
i	3-4	Above-Left	AL
j	4-4	Above	A

Table T_3

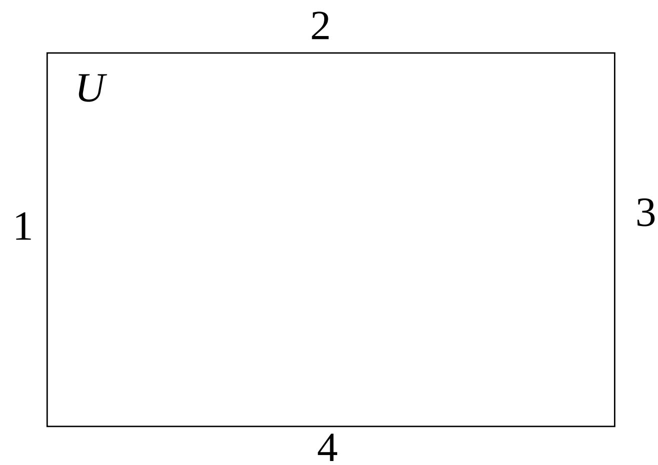


Figure 8: Rectangle of the universe of discourse

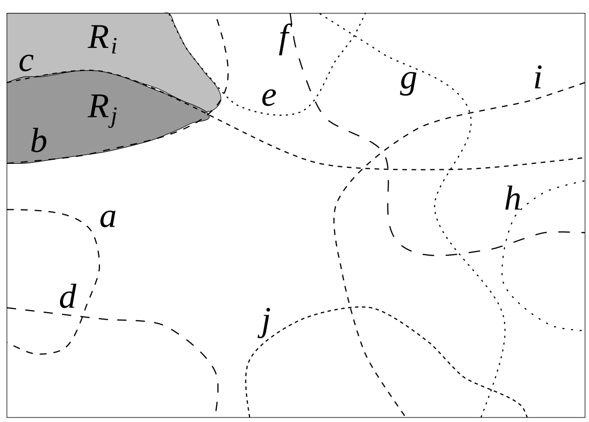


Figure 9: Halfplanes $a - j$

With this systematic way of labeling halplanes and denotations, we can infer the relative position of a region to another, in relation to a given halplane.

In order to compare the relative position between two regions it is necessary to eliminate the halfplanes in common and analyse the differences between the resulting formulas.

Where the predicate components are the same the two regions lie in the same halfplane and no topological information is available from the formula; only differences contain topological information. Take as an example the following two formulas representing two arbitrary regions R_i and R_j:

R_i $hp(a) \wedge hp(b) \wedge hp(f)$
R_j $hp(a) \wedge \neg hp(b) \wedge hp(f)$

The difference is:

R_i $hp(b)$
R_j $\neg hp(b)$

i.e. R_i lies on the other side of the boundary of halfplane b to R_j. The specific interpretation depends on the denotation of the specific case.

3 Characteristics of the Representation

Any representation should ideally have the properties (Mantyla 1988, p. 51) described in the two first columns of Table T_4, where the third column expresses the particular interpretation for the halfplanes.

Property	Description	Specific for halfplanes
expressive power	what can be represented amongst all possible shapes in the domain to be described ?	geometrical entities that divide U into two half-U
uniqueness	do all valid representations model one shape only ?	halfplane representation using logic is unique since each predicate has been grounded to a particular geometrical halfplane
unambiguity	do all shapes have only one representation ?	each *valid* combination of halfplanes determines a shape, therefore halfplane model are unambiguous
validity	do all representations describe some shape in the domain ?	not all combinations of half-planes are valid but can be made valid.
conciseness	how large do the representations become for interesting shapes ?	relatively concise
description languages	what kind of description languages can be based on the representation ?	logical formulas
computational ease	what kinds of algorithms can be written for the representation and what is their computational complexity ?	logic representation and inference
closure of operations	do all operations among elements of the model result in elements of that model ?	any combination of two halfplane models with a logical operation defines a new valid model; hence these operations are closed

Table T_4

4 Application of the Formalism

4.1 Applications in Topology

Consider the shapes in Figure 10. The regions R_1, R_2, R_3, R_4, R_5 and R_6 are defined by the

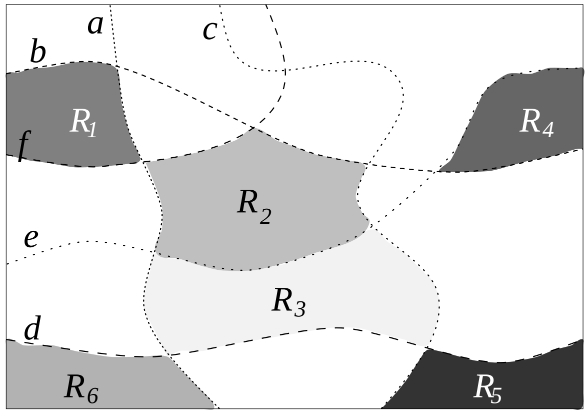


Figure 10: Regions $1 - 6$

following formulas:

R_1 $hp(a) \wedge \neg hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f)$
R_2 $\neg hp(a) \wedge \neg hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge \neg hp(f)$
R_3 $\neg hp(a) \wedge \neg hp(b) \wedge hp(c) \wedge hp(d) \wedge \neg hp(e) \wedge \neg hp(f)$
R_4 $\neg hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge \neg hp(e) \wedge \neg hp(f)$
R_5 $\neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge \neg hp(f)$
R_6 $hp(a) \wedge \neg hp(b) \wedge hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge \neg hp(f)$

The denotation for each halfplane is given in Table T_5.

halfplane	Denotation	$\overline{\text{halfplane}}$
$hp(a)$	L	R
$hp(b)$	A	B
$hp(c)$	L	R
$hp(d)$	A	B
$hp(e)$	A	B
$hp(f)$	AL	BR

Table T_5

Comparing R_1 and R_2, the result is:

R_1 $hp(a) \wedge hp(f)$
R_2 $\neg hp(a) \wedge \neg hp(f)$

which means:

- based on halfplane $hp(a)$ the region R_1 is Left (L) of region R_2, and
- based on halfplane $hp(f)$ the region R_1 is Above-Left (AL) of region R_2

The same reasoning is used to compare other regions, as shown in the following examples. Comparing R_2 and R_5, the result is:

R_2 $hp(c) \wedge hp(d) \wedge hp(e)$
R_5 $\neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e)$

which means:

- based on halfplane $hp(c)$ the region R_2 is Left (L) of region R_5, and
- based on halfplane $hp(d)$ the region R_2 is Above (A) region R_5, and
- based on halfplane $hp(e)$ the region R_2 is Above (A) region R_5

Comparing R_3 and R_4, the result is:

R_3 $\neg hp(c) \wedge hp(c)$
R_4 $hp(b) \wedge \neg hp(c)$

which means:

- based on halfplane $hp(b)$ the region R_3 is Below (B) region R_4, and
- based on halfplane $hp(c)$ the region R_3 is Left (L) of region R_4

Comparing R_1 and R_5, the result is:

R_1 $hp(a) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f)$
R_5 $\neg hp(a) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge \neg hp(f)$

which means:

- based on halfplane $hp(a)$ the region R_1 is Left (L) of region R_5, and
- based on halfplane $hp(c)$ the region R_1 is Left (L) of region R_5, and
- based on halfplane $hp(d)$ the region R_1 is Above (A) region R_5, and
- based on halfplane $hp(e)$ the region R_1 is Above (A) region R_5, and
- based on halfplane $hp(f)$ the region R_1 is Above-Left (AL) of region R_5

Comparing R_6 and R_4, the result is:

R_6 $hp(a) \wedge \neg hp(b) \wedge hp(c) \wedge \neg hp(d)$
R_4 $\neg hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge hp(d)$

which means:

- based on halfplane $hp(a)$ the region R_6 is Left (L) of region R_4, and
- based on halfplane $hp(b)$ the region R_6 is Below (B) region R_4, and
- based on halfplane $hp(c)$ the region R_6 is Left (L) of region R_4, and
- based on halfplane $hp(d)$ the region R_6 is Below (B) region R_4

4.2 Applications in Shape Emergence

One major concern in developing this formalism is to manipulate shapes symbolically, with applications such as shape emergence (Gero & Yan 1993), although other applications can be envisaged.

A complete model for shape emergence is not within the scope of this paper. Briefly we can give four basic heuristics to find emerged shapes. These heuristics are:

1. Complement of the primary shape.
   A primary shape is a shape which was input to the system. From a given primary shape PS, an emerged shape S is obtained by the logical negation of the formula which defines the primary shape PS, as:
   $S = \neg\ PS$

2. Wholeness of the primary shape.
   If the primary shape is composed by two or more regions, they can be composed into a more general shape, which is emerged from the primary shape. This emerged shape is obtained by the elimination of the two complementary halfplanes in common in two formulas of the primary shape. This procedure is similar to the resolution principle. Suppose the primary shape PS is composed by two regions R_i and R_j, defined as:

$$\begin{array}{lll} R_i & \text{as} & hp(a_1) \wedge hp(a_2) \wedge ... \wedge hp(a_n) \\ R_j & \text{as} & hp(a_1) \wedge hp(a_2) \wedge ... \wedge hp(a_n) \end{array}$$

   the emerged shape is obtained eliminating from R_i the $hp(a_i)$ and from R_j the $\neg hp(a_i)$. For example, a shape S_1 is defined by $R_1 \wedge R_2$, where:

$$\begin{array}{lll} R_1 & \text{as} & hp(a) \wedge \neg hp(b) \wedge hp(c) \wedge hp(d) \\ R_2 & \text{as} & hp(a) \wedge hp(b) \wedge hp(c) \wedge hp(d) \end{array}$$

   The emerged shape, or the more general shape, is:

$$S_e \quad \text{as} \quad hp(a) \wedge hp(c) \wedge hp(d)$$

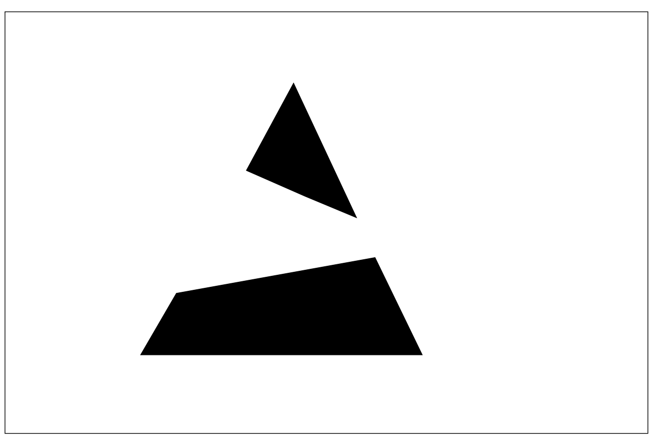


Figure 11: Primary shape

   In order to illustrated this, consider the primary shape shown in Figure 11. It is composed of the two regions shown in black. These regions are bounded by line segments. Extending

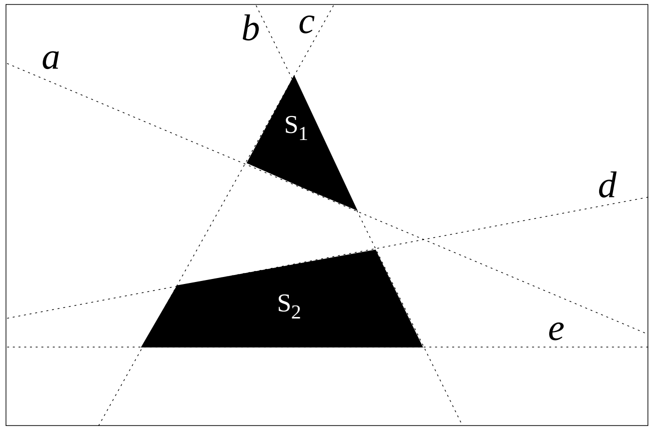


Figure 12: Primary shape decomposed into halfplanes

each line segment to define halfplanes a, b, c, d, and e, we have Figure 12. The constraint $\mathcal{C}$ of the topology shown in Figure 12 is:

- $\neg hp(a) \leftarrow hp(b) \wedge hp(c) \wedge \neg hp(d) \wedge hp(e)$
- $\neg hp(a) \leftarrow hp(b) \wedge hp(c) \wedge \neg hp(d) \wedge \neg hp(e)$
- $\neg hp(a) \leftarrow hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge hp(e)$
- $\neg hp(a) \leftarrow hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e)$
- $\neg hp(a) \leftarrow \neg hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e)$
- $hp(a) \leftarrow \neg hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e)$

The primary shape PS can be represented as the formula: $S_1 \vee S_2$, where S_1 and S_2 are the formulas:

$$\begin{array}{lll} S_1 & \text{is} & hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge hp(e) \\ S_2 & \text{is} & \neg hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge hp(e) \end{array} \quad (2)$$

In the primary shape PS we verify that the regions S_1 and S_2 have the halfplanes $hp(a)$ and $hp(d)$ in common and complementary. Therefore if we apply the *wholeness criteria* on the primary shape PS we obtain the formula:

$$S_3 \quad \text{is} \quad hp(b) \wedge \neg hp(c) \wedge hp(e)$$

The final shape S_3 defined by this formula is represented in Figure 13.

3. Any combination of regions

   There are many possible shapes that could be emerged from the primary shapes by combining different regions. The criteria as to which one is better or more likely to be emerged are not described here.

   Using the example shown in Figure 12, we can obtain shapes that are border adjacency to S_2. The expression for S_2 is:

$$\neg hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge hp(e) \quad (3)$$

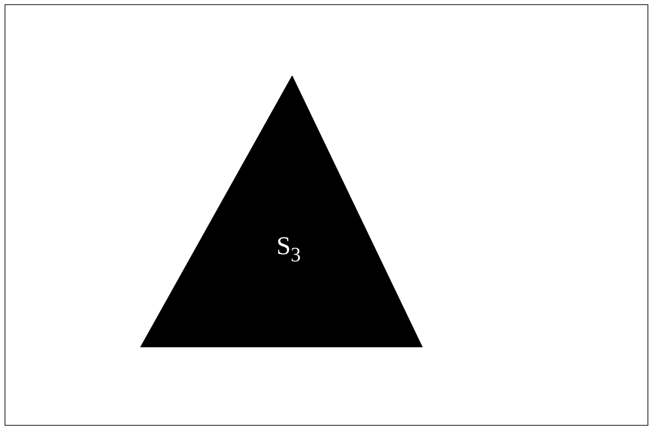


Figure 13: Emergent shape S_3

with *minimal description* S_{2min}:

$$hp(b) \land \neg hp(c) \land \neg hp(d) \land hp(e) \quad (4)$$

Once we have four halfplanes, there are four regions border adjacency to S_{2min}, as:

$$\begin{array}{ll} S^1_{2min} & \neg hp(b) \land \neg hp(c) \land \neg hp(d) \land hp(e) \\ S^2_{2min} & hp(b) \land hp(c) \land \neg hp(d) \land hp(e) \\ S^3_{2min} & hp(b) \land \neg hp(c) \land hp(d) \land hp(e) \\ S^4_{2min} & hp(b) \land \neg hp(c) \land \neg hp(d) \land \neg hp(e) \end{array}$$

These four regions are shown shaded in Figure 14.

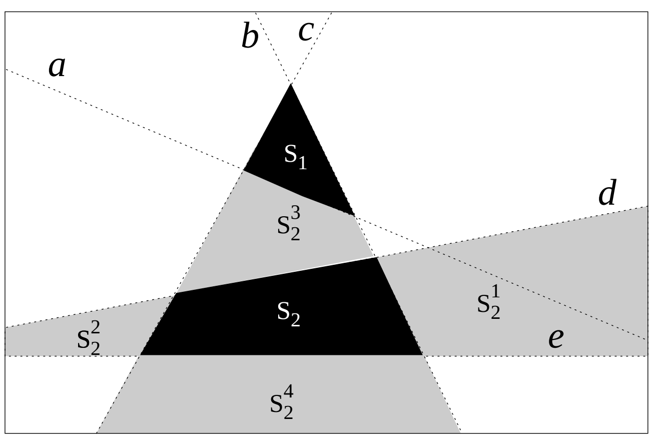


Figure 14: Regions adjacents on border to S_2

From these regions we can obtain emergent shapes, as shown in Figure 13 and Figure 15.

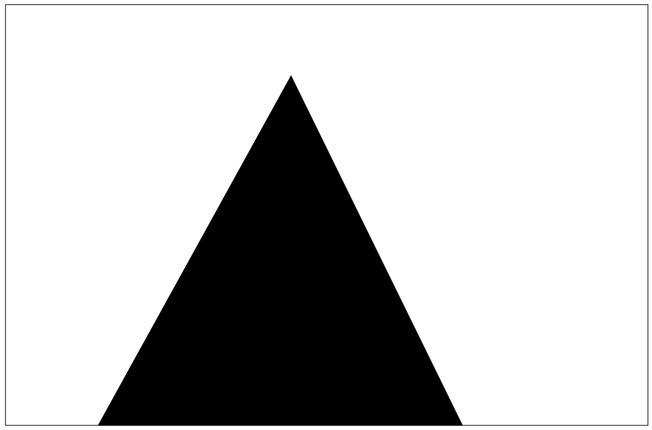


Figure 15: Emergent shape

4.3 Application in Architectural Plans

In this section we present two applications of this formalism with an elaboration of one of them in the domain of architecture. Figure 16(a) shows the Casalecchio town in south-western suburb of Bologna Italy, designed by James Stirling and Michael Wilford (Anonymous 1994, pp 118). The 2D view of it is shown in Figure 16(b). The central part of the drawing is extracted, re-represented using halfplanes and is shown in Figure 16(c). Through operations using the regions defined by these halfplanes, a number of possibles shapes can be emerged from it, as shown in Figures 16(d)(e)(f)(g) and (h). Such emerged shapes can be used to further development of the design. For example, the Figure 16(e) can represent the flux of people/cars between buildings. Figure 16(g) is a more surprising shape, and is hard to imagine the designer with this shape in mind when s/he designed the building. Once this shape had been made explicit to the designer, it can influence further development of the design taking the star as part of it.

The logic representation of the shapes in Figure 17 is shown in the Appendix.

Shapes S_1, S_2 and S_3 are defined as:

S_1

$(hp(a) \wedge hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(i) \wedge hp(j) \wedge hp(k) \wedge \neg hp(l)) \vee$
$(hp(a) \wedge hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(i) \wedge \neg hp(j) \wedge \neg hp(k) \wedge hp(l)) \vee$
$(hp(a) \wedge hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge hp(h) \wedge hp(i) \wedge \neg hp(j) \wedge hp(k) \wedge \neg hp(l)) \vee$
$(hp(a) \wedge hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge hp(h) \wedge \neg hp(i) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)) \vee$
$(hp(a) \wedge hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge hp(h) \wedge hp(i) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l))$

S_2

$(\neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge hp(h) \wedge \neg hp(i) \wedge hp(j) \wedge \neg hp(k) \wedge \neg hp(l)) \vee$
$(\neg hp(a) \wedge \neg hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge hp(h) \wedge \neg hp(i) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)) \vee$
$(\neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge \neg hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge \neg hp(i) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)) \vee$
$(\neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge \neg hp(i) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)) \vee$
$(\neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge \neg hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge hp(h) \wedge \neg hp(i) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l))$

S_3

$(hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(i) \wedge hp(j) \wedge hp(k) \wedge \neg hp(l)) \vee$

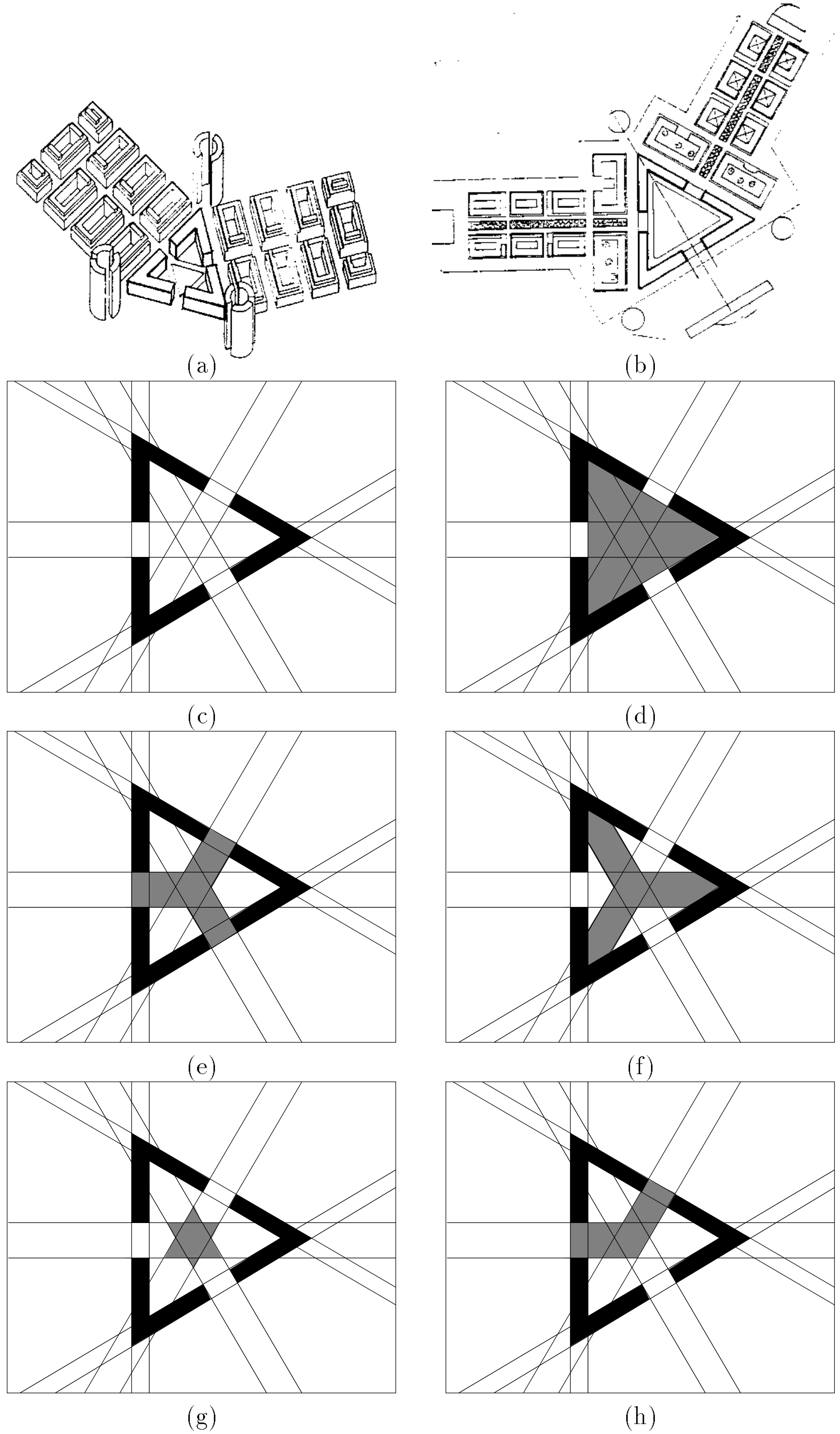


Figure 16: Casalecchio plan study designed by James Stirling and Michael Wilford

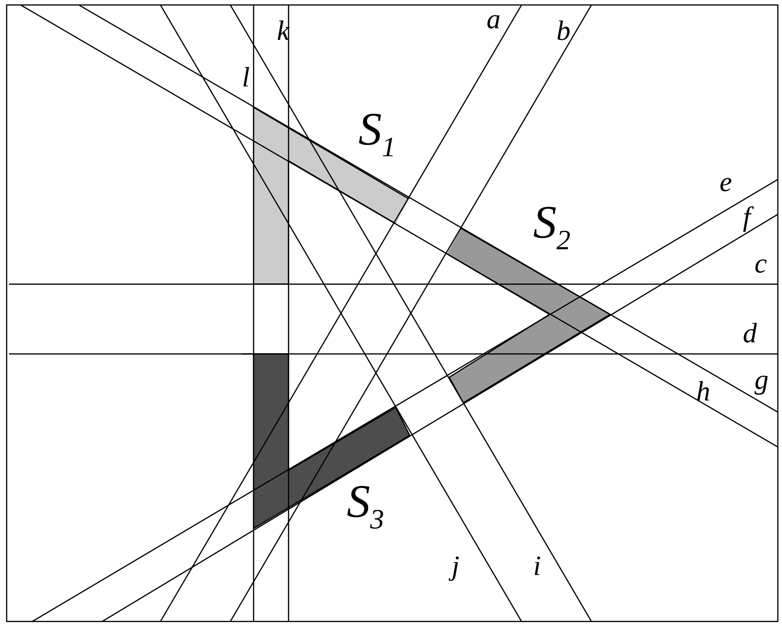


Figure 17: Halfplane representation of Figure 16(c)

$$(\neg hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(i) \wedge hp(j) \wedge hp(k) \wedge \neg hp(l)) \vee$$
$$(\neg hp(a) \wedge \neg p(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(i) \wedge hp(j) \wedge \neg hp(k) \wedge \neg hp(l)) \vee$$
$$(\neg hp(a) \wedge p(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(i) \wedge hp(j) \wedge \neg hp(k) \wedge \neg hp(l)) \vee$$
$$(\neg hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(i) \wedge hp(j) \wedge hp(k) \wedge \neg hp(l))$$

The emergent internal triangle shown in Figure 16(d) is composed of all visible regions defined by $\neg hp(h) \wedge hp(e) \wedge \neg hp(k)$, according to criteria presented in section 4.2.

5 Implementation

We have developed a prototype system in order to test the concepts presented in this paper. The system has two modules: drawing editor and inference engine. The basic modules are shown in Figure 18.

The drawing editor was developed in Tk/Tcl (Ousterhout 1994) and has basic functions to draw regions bounded by straight and non-straight lines. In addition, the system finds all the intersections among the lines and gives as output the constraints expressed in predicates form.

In the inference engine module we used Prolog to implement the theorem prover (based on examples in Bratko (1990)). In order to prove if a region R is visible under the constraint $\mathcal{C}$ given, we have to prove that $(R \rightarrow \mathcal{C})$ is valid. Using the *resolution principle* we have to find the clause *nil* (contradiction) from the clause $\neg(R \rightarrow \mathcal{C})$, or $R \wedge \neg\mathcal{C}$. Other results can be obtained as string manipulation from the formulas.

6 Conclusion

In this paper we have developed a logic-based representation for shapes using halfplanes as its basis. Of particular importance in this representation is the consistency with which shapes

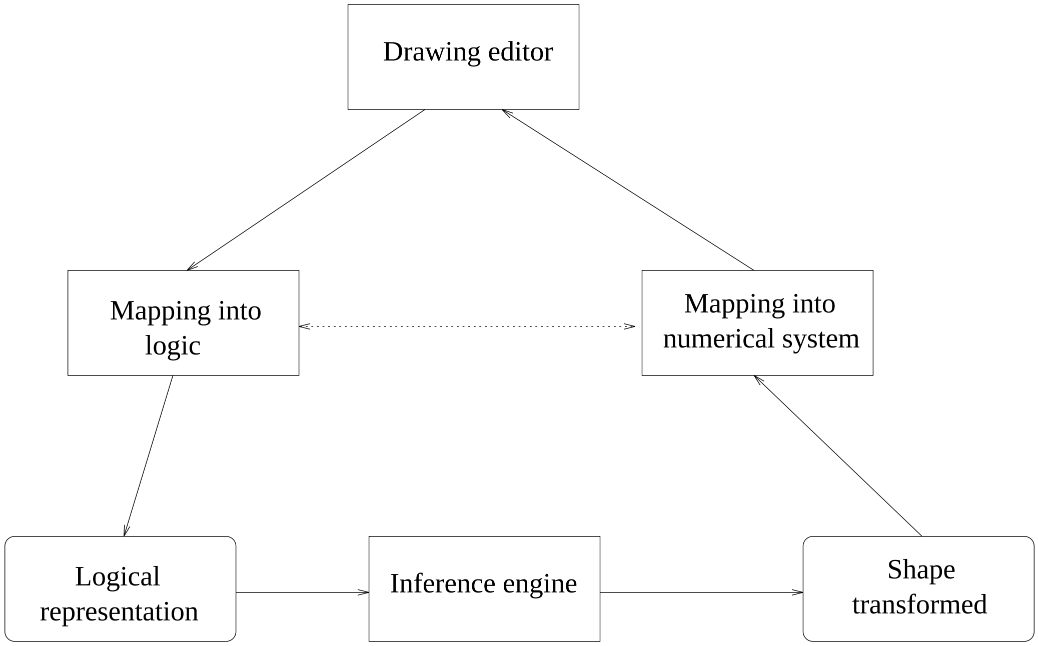


Figure 18: Modules of prototype system

can be represented whether they be bounded by straight line segments or curved line segments. This is an unusual characteristic to be able to achieve in a shape representation. Shape features such as topology, adjacency and closure are easily deducible from the representation. This logic-based representation will be used as the basis for an implementation to support early stages of design in visually-based environments such as architecture and industrial design. Any drawing editor, such as AutoCAD, can be used in order to enhance the input and display functionality.

The ability to reason about topology directly and to emerge shapes using a consistent formalism provides opportunities for the support of computational creative processes concerned with shapes.

Acknowledgements

José Damski thanks CNPq Brasilia/Brazil for his scholarship. Part of this work is funded by a grant from the Australian Research Council.

References

Anonymous (1994). *James Stirling, Michael Wilford and Associates: Buildings & Projects, 1975-1992*, Thames and Hudson/Verlag Gerd Hatje, London/Stuttgart.

Bratko, I. (1990). *Prolog - Programming for Artificial Intelligence*, 2nd edn, Addison-Wesley, Reading.

Chang, C. & Lee, R. (1973). *Symbolic Logic and Mechanical Theorem Proving*, Academic Press, New York.

Damski, J. C. B. & Gero, J. S. (1993). Using logic to represent graphical shapes, *in* C. Rowles, H. Liu & N. Foo (eds), *AI '93 - The Sixth Australian Joint Conference on Artificial Intelligence*, World Scientific, Singapore, pp. 90–95.

Gero, J. S. & Yan, M. (1993). Discovering emergent shapes using a data-driven symbolic model, *in* U. Flemming & S. V. Wyk (eds), *CAAD Futures '93*, Elsevier, Amsterdam, pp. 3–17.

Giraud, C. (1984). Presque half-planes: towards a general representation scheme, *Computer-Aided Design* **16**(1): 17–24.

Mantyla, M. (1988). *An Introduction to Solid Modeling*, Computer Science Press, Rockville.

Mitchell, W. J. (1993). A computational view of design creativity, *in* J. S. Gero & M. L. Maher (eds), *Modeling Creativity and Knowledge-Based Creative Design*, Lawrence Erlbaum, Hillsdale, New Jersey, pp. 25–44.

Ousterhout, J. K. (1994). *Tcl and the Tk Toolkit*, Addison-Wesley, Reading.

Stiny, G. (1980). Introduction to shape and shape grammars, *Environment and Planning B* **7**: 343–351.

Appendix

Logic Representation of Shape in Figure 17

The logic representation of the shapes in Figure 17, based on the halfplane i is defined by the following formulas:

$hp(i) \leftarrow hp(a) \wedge hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge hp(k) \wedge \neg hp(l)$

$hp(i) \leftarrow hp(a) \wedge hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge hp(k) \wedge hp(l)$

$hp(i) \leftarrow hp(a) \wedge hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge \neg hp(j) \wedge hp(k) \wedge hp(l)$

$hp(i) \leftarrow hp(a) \wedge hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge hp(l)$

$hp(i) \leftarrow hp(a) \wedge hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge hp(h) \wedge hp(j) \wedge hp(k) \wedge hp(l)$

$hp(i) \leftarrow hp(a) \wedge hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge hp(h) \wedge \neg hp(j) \wedge hp(k) \wedge hp(l)$

$hp(i) \leftarrow hp(a) \wedge hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge hp(g) \wedge hp(h) \wedge hp(j) \wedge hp(k) \wedge hp(l)$

$hp(i) \leftarrow hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge hp(k) \wedge hp(l)$

$hp(i) \leftarrow hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge hp(k) \wedge \neg hp(l)$

$hp(i) \leftarrow hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$

$hp(i) \leftarrow hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$

$hp(i) \leftarrow \neg hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$

$hp(i) \leftarrow \neg hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$

$hp(i) \leftarrow \neg hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$

$hp(i) \leftarrow \neg hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge \neg hp(k) \wedge hp(l)$

$hp(i) \leftarrow hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$

$hp(i) \leftarrow hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge hp(k) \wedge \neg hp(l)$

$hp(i) \leftarrow \neg hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge hp(k) \wedge \neg hp(l)$

$hp(i) \leftarrow \neg hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge hp(k) \wedge hp(l)$

$hp(i) \leftarrow \neg hp(a) \wedge \neg p(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$

$hp(i) \leftarrow \neg hp(a) \wedge p(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$

$hp(i) \leftarrow \neg hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge \neg hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$

$hp(i) \leftarrow \neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge \neg hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$
$hp(i) \leftarrow \neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge \neg hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge hp(k) \wedge \neg hp(l)$
$hp(i) \leftarrow \neg hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge \neg hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge hp(k) \wedge \neg hp(l)$
$hp(i) \leftarrow \neg hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge hp(k) \wedge hp(l)$
$hp(i) \leftarrow \neg hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge hp(k) \wedge \neg hp(l)$
$hp(i) \leftarrow \neg hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge \neg hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge hp(k) \wedge hp(l)$
$hp(i) \leftarrow hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge \neg hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge hp(k) \wedge hp(l)$
$hp(i) \leftarrow hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge hp(k) \wedge hp(l)$
$hp(i) \leftarrow hp(a) \wedge hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge hp(k) \wedge hp(l)$
$hp(i) \leftarrow hp(a) \wedge hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$
$\neg hp(i) \leftarrow \neg hp(a) \wedge hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge hp(g) \wedge hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$
$\neg hp(i) \leftarrow \neg hp(a) \wedge hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$
$\neg hp(i) \leftarrow \neg hp(a) \wedge \neg hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge hp(g) \wedge hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$
$\neg hp(i) \leftarrow \neg hp(a) \wedge \neg hp(b) \wedge hp(c) \wedge hp(d) \wedge \neg hp(e) \wedge hp(f) \wedge hp(g) \wedge hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$
$\neg hp(i) \leftarrow \neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge hp(h) \wedge hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$
$\neg hp(i) \leftarrow \neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge \neg hp(e) \wedge hp(f) \wedge hp(g) \wedge hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$
$\neg hp(i) \leftarrow \neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge hp(g) \wedge hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$
$\neg hp(i) \leftarrow \neg hp(a) \wedge \neg hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$
$\neg hp(i) \leftarrow \neg hp(a) \wedge \neg hp(b) \wedge hp(c) \wedge hp(d) \wedge hp(e) \wedge hp(f) \wedge hp(g) \wedge \neg hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$
$\neg hp(i) \leftarrow \neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge \neg hp(e) \wedge hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$
$\neg hp(i) \leftarrow \neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge \neg hp(e) \wedge \neg hp(f) \wedge \neg hp(g) \wedge \neg hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$
$\neg hp(i) \leftarrow \neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge \neg hp(e) \wedge \neg hp(f) \wedge hp(g) \wedge hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$
$\neg hp(i) \leftarrow \neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c) \wedge hp(d) \wedge \neg hp(e) \wedge \neg hp(f) \wedge \neg hp(g) \wedge hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$
$\neg hp(i) \leftarrow \neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge \neg hp(f) \wedge hp(g) \wedge hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$
$\neg hp(i) \leftarrow \neg hp(a) \wedge \neg hp(b) \wedge \neg hp(c) \wedge \neg hp(d) \wedge \neg hp(e) \wedge \neg hp(f) \wedge \neg hp(g) \wedge hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$
$\neg hp(i) \leftarrow \neg hp(a) \wedge \neg hp(b) \wedge hp(c) \wedge hp(d) \wedge \neg hp(e) \wedge \neg hp(f) \wedge hp(g) \wedge hp(h) \wedge \neg hp(j) \wedge \neg hp(k) \wedge \neg hp(l)$