Qualitative Spatial Reasoning on Three-Dimensional Orientation Point Objects

Julio Pacheco, Mª Teresa Escrig, Francisco Toledo

Universitat Jaume I
Dpto. Ingeniería y Ciencia de los Computadores
Campus Riu Sec 12071
Castellón (Spain)
{pacheco, escrigm, toledo}@icc.uji.es

Abstract

An approach for representing and reasoning with 3-D qualitative orientation of point objects is presented in this paper. The model in 3-D is an extension of the Zimmerman and Freksa's orientation model in 2-D. The paper presents several attempts to represent 3-D spatial orientation, why those first attempts did not work and how problems which have been found are solved in a 3-D model. An iconical notation for 3-D spatial orientation relations is also presented with the aim of representing conceptual neighbourhood. The algebra of this model is also explained.

Keywords: Spatial Reasoning, Qualitative Reasoning, Qualitative Orientation.

Introduction

One of the main aims of the Artificial Intelligence field is to simulate human behaviour in general and build robots with a human-like performance in particular. The principal goal of the Qualitative Spatial Reasoning field is to represent our everyday common sense knowledge about the physical world, and the underlying abstractions used by engineers and scientists when they create quantitative models. Kak [10] points out that the behaviour of the intelligent machine of the future might carry out temporal reasoning, spatial reasoning and also reason over interrelated entities occupying space and changing in time with respect to their attributes and spatial interrelationships. Spatial information that we obtain through perception is coarse and imprecise, thus qualitative models which reason with distinguishing characteristics rather than with exact measures seems to be more appropriate to deal with this kind of knowledge.

Supposing that we want to know the qualitative orientation of the workmate' s office in our university building (which has more than one floor) with respect the position that we have. Or we know the relative orientation between some offices (in that building) and we want to know the orientation of every office with respect the rest of the officer. In that case, we need to know the height in which every office is situated, that is, we need to represent and reason with a 3-dimension orientational model.

Among the approaches that deal with qualitative spatial orientation, it is important to distinguish between models based on projections and models not based on projections. In models based on projections, the relative orientation of objects is obtained by using (orthogonal or non-orthogonal) projections of objects into external axes, and then reasoning in one-dimension, by using Allen' s temporal logic. There exist mainly three qualitative approaches for orientation which are based on projections: Guesgen' s approach [8] is a straightforward extension of Allen' s temporal reasoning; Chang and Jungert approach [1], and Mukerjee and Joe' s approach [12]. Models based on projections might provide inconsistent representation of objects whose sides are not parallel to the axes. To overcome this problem, qualitative models not based on projections have been developed.

There exist mainly three qualitative models for orientation which are not based on projections into external reference systems (RS): Freksa and Zimmermann' s model [3, 4, 5, 6]; Hernández' s approach [9]; and Frank' s approach [7]. In these models not based on projections, space is divided into qualitative regions by means of RSs, which are centered on the reference objects (i.e. the RS are local and egocentric). Spatial objects are always simplified to points, which are the representational primitives.

From the three models not based on projections, the Zimmerman and Freksa' s model is considered more cognitive because no extrinsic reference system (such as magnetic poles) is necessary. This model has been chosen for its extension to 3-D.

In order to deal with 3-D orientation, first of all we are going to represent this spatial aspect and secondly we are going to reason with it. In the reasoning process we are going to distinguish two parts: the Basic Step of the Inference Process (BSIP) and the Full Inference Process (FIP). For those models not based on projections, the BSIP can be defined in general terms such as: given a spatial relationship between point *c* with respect to a RS, and another spatial relationship between point *d* with respect to another RS, point *c* being part of that RS, the BSIP consists of obtaining the spatial relationship of point *d* with respect to the first RS. The RS will be different depending on the model. When more relationships among several spatial landmarks are provided, then the FIP is necessary. It consists of repeating the BSIP as many times as possible, with the initial information and the information provided by some BSIP, until no more information can be inferred.

In order to accomplish the integration of orientation, distance and cardinal directions into the same spatial model we will use the following three steps:

(1) the representation of the spatial aspect to be integrated;

(2) the definition of the BSIP for each represented spatial aspect; and

(3) the definition of the FIP for this spatial aspect.

These three steps have been applied for integrating into the same model.

In this paper, we are going to focus our attention on the representation part and on the BSIP.

The structure of the rest of the paper is as follows: firstly, the original 2-D orientation model will be introduced. Secondly, the extended 3-D orientation model will be explained including the representation, the algebra of this model, the Basic Step of Inference and the Full Inference Process.

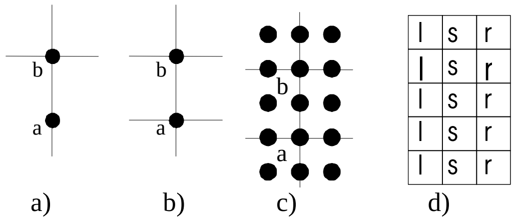


Figure 1. ***a) The coarse 2-D orientational RS; b) the fine RS; c) the 15 qualitative regions and d) their names in iconical representation.***

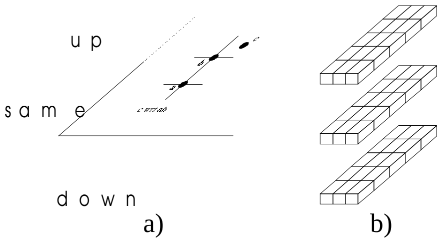


Figure 2. ***a) Three different regions are firstly considered in height: the plane where the 2-D orientation RS is studied, the region upper this plane and the region downer this plane; b) the corresponding iconical representation.***

The 2-D Zimmerman and Freksa's orientation representation

In the [3,4,5,6] approach, the orientation RS is defined by a point and a director vector *ab,* which defines the left/right dichotomy. It can be interpreted as the direction of movement. The RS also includes the perpendicular line by the point *b*, which defines the first front/back dichotomy, and it can be seen as the straight line that joins our shoulders. This RS divides space into 9 qualitative regions (figure 1 a). A finer distinction could be made in the back regions by drawing the perpendicular line by the point *a*. In this case, the space is divided into 15 qualitative regions (figure 1 b). The point *a* defines the second front/back dichotomy of the RS. An iconical representation of the fine RS and the names of the regions are shown in figure 1 c) and d).

The information represented in both coarse and fine RSs is where is the point *c* with respect to the RS *ab*, that is, *c wrt ab*. This information can also be expressed of four different ways as a result of applying the following four operations: Homing, Homing Inverse, Shortcut and Shortcut Inverse.

The 3-D Orientational Model

Representation

We can consider a plane which contains the two points which defines de 2-D RS, *a* and *b.* Another point *c* might be in the same plane, in an upper height or in a lower height (see the figure 2). Therefore we extend the 2-D grid to a 3-D grid where we consider the three planes where the point *c* could be.

However, two points (*a* and *b*) define infinite planes (see figure 3). The plane chosen will tell us the qualitative height of a third point *c*.

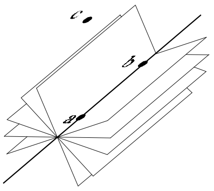
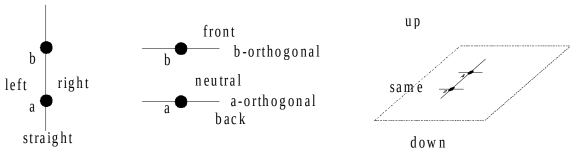


Figure 3. ***Depending on the plane we might say that the point c is up, in the same plane or down with respect to the line which defines ab.***

We are interested not only in representing orientation in 3-D, but also in reasoning with this concept in 3-D. Three points define a plane perfectly. When a new point appears, it might be that the new point do not belong to the *abc* plane. It is impossible to reason changing completely the plane in every three different points.

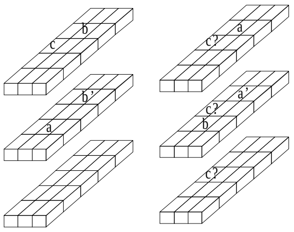
We do not need three points to construct our reference plane. We must decide on the plane before choosing the points [14, 17]. In that case, our 3-D orientation RS will be based on a point and a reference plane. The reference plane chosen will be a plane parallel to the floor (or to the base of the robot in a robotic application). In the case we do not have any specific plane to make reference, we must decide it first. When we said a reference plane (as the point *a* could be in any height) we refer to all the family of planes parallel to the reference plane.

Once the reference plane has been chosen, we consider two heights more that define the upper and downer height, respectively. Therefore, the 2-D orientation model has been extended to the third dimension, as it is shown in figure 4.



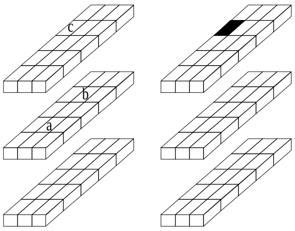
a) b) c)

Figure 4. ***a) left, straight and right; b) front, b-orthogonal, neutral, a-orthogonal and back; c) up, same and down.***



a) b)

Figure 5. ***a) When* a *and* b *do not belong to the same reference plane, the point* b *is projected onto the plane which contains* a *in* b'; *b) Considering that the point* b *is downer the point* a, *the height of another point* c *is not determined.***



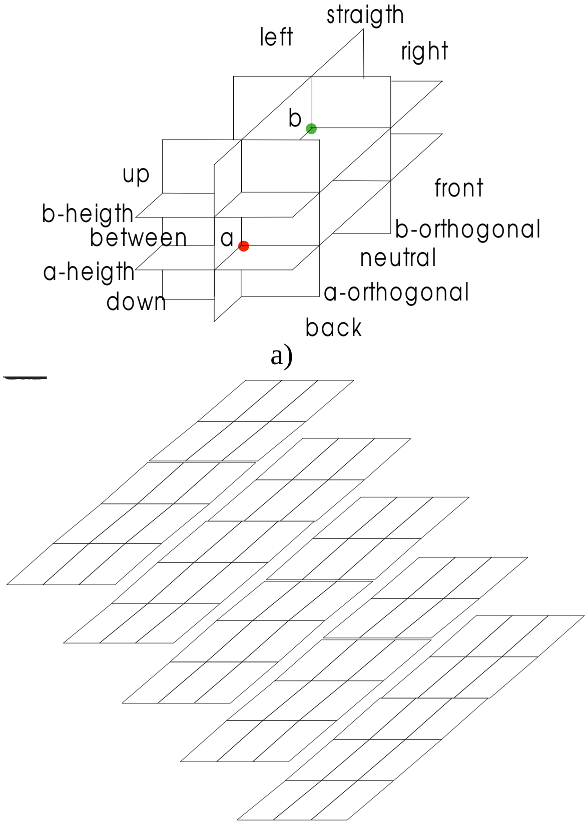
a) b)

Figure 6. ***If a point* c *is in the position given in figure a), it is iconicaly represented as in figure b).***

Considering these three heights we had extended the 15 qualitative regions in to 45 qualitative regions (figure 5).

Having these three heights (the **up** height, the **same** height and the **down** height with respect to the plane passes by the point a) implies that *a* and *b* have always to be in the same plane. In most of the cases this fact did not happen. When a and *b* are not in the same plane, it might happen that the point *b* is upper than the point a or that the point a is upper than the point *b*. If the point *b* is upper than the point *a*, the reference plane passes by the point a and we work with the orthogonal projection of the point *b* onto the reference plane (in figure 6 the point *c* is upper the point *a*). When we considered the RS with the reference plane passing by the point *b*, we had only information about the point *c* with respect to the point *a*, but we did not have information about th eposistion of the popint *c* with respect to the point *b*; the third point could be in any height (see figure 6 b).

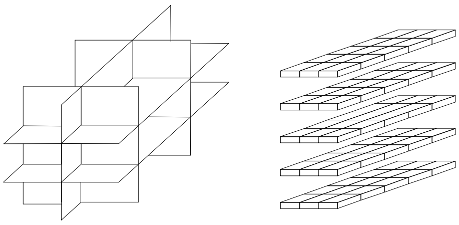
As the point *a* and the point *b* are not always in the same height, we can define a fine 3-D orientation RS by including five different heights: up, **b-height**, between, **a-height** and down. (figure 7 a).



a)

b)

Figure 7. ***a) The 3-D orientation RS b)the names inside of the iconic representation.***



a) b)

Figure 8. ***a) The division of the 3-D space into 75 regions; b) and the corresponding 3-D iconic representation.***

In this case, the 3-D orientational RS divides space into 75 qualitative regions (figure 8 a), which arise from the three width parts, the five length parts and the five high parts (3 x 5 x 5 = 75). An 3-D iconical representation of the RS is shown in figure 8 b).

The names of every region are defined according to the position they are. We will use acronyms as `ulf` if it position is in *up-left-front*; `usf` for *up-straight-front*; `urf` for *up-right-front*, and so on (figure 7 b).

As a matter of clarity, the 3-D representation has been translated into 2-D iconical representation, as it is shown in figure 9. In this 2-D iconical representation it is easier to perceive conceptual neighbourhood.

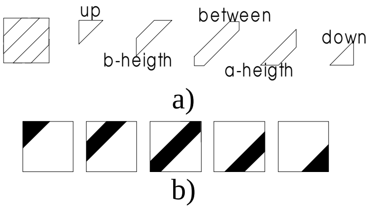


Figure 9. ***a) A single cell divided into five heights and the names of every part; b) the representation of the different heights.***

By agreement, in what follows we are going to reason with the point *b* above the point *a*.

For the cases in which the second point of the front/back dichotomy is not in the same plane or above the first point of the front/back dichotomy, we rotate the RS 180 degrees by using the spin operation (figure 10). The algebra of this operation will be defined in the next section.

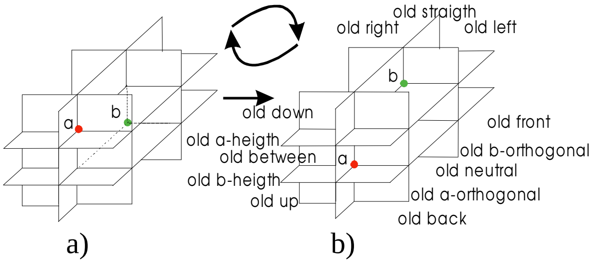


Figure 10. ***a) When the point* b *is not above the point* a *b) the spin operation is made (rotate 180º the RS).***

You can see some examples of the 3-D qualitative orientation represented in the 2-D iconical diagram drawing in figure 11.

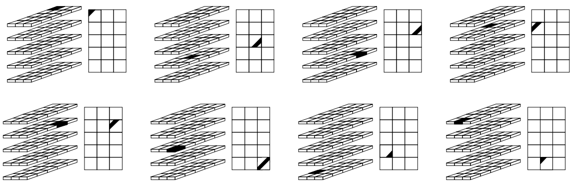


Figure 11: ***Some examples of object orientation in 3-D and their corresponding iconical representation***

The information to be represented with this 3-D orientation RS (c *wrt ab*) can also be expressed of four different ways (as well as the original 2-D orientation RS) as a result of applying the following operation whose algebra is defined in the next section: Homing, Homing Inverse, Shortcut and Shortcut Inverse.

Algebra

In our approach, the operations have been implemented as facts in a PROLOG database. In order to deal with the disjunction of relationships, the result of applying some operation to any orientation relationship is a list of relationships. Often this list contains only a relation (as in Identity, Inverse or Spin operations), but it allows us to deal with more than one relation if necessary (as in Homing, Homing Inverse, Shortcut and Shortcut Inverse) [13, 15, 16].

Identity. We will represent the identity operation as ID.

The algebraic notation is: ID(*c wrt ab*) = *c wrt ab.*

The 2-D iconical representation of this operation is presented in figure 12 with the 75 different positions.

The PROLOG facts of the Identity operation would be for instance (see figure 12): `id(ulf,[ulf])`; `id(usf,[usf])`; etc.

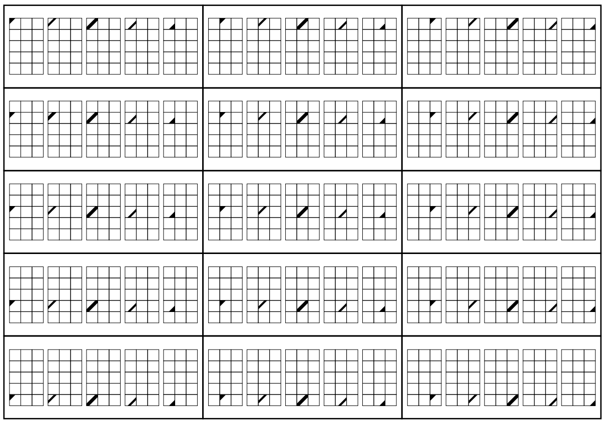


Figure 12: ***The 2-D iconical representation of the 3-D qualitative spatial orientation ``c* wrt *ab''.***

Inversion. The inversion operation (INV) corresponds to consider the point *c* with respect to the RS *ba* (see figure 13 a).

The algebraic notation is: INV(*c wrt ab*) = *c wrt* ba.

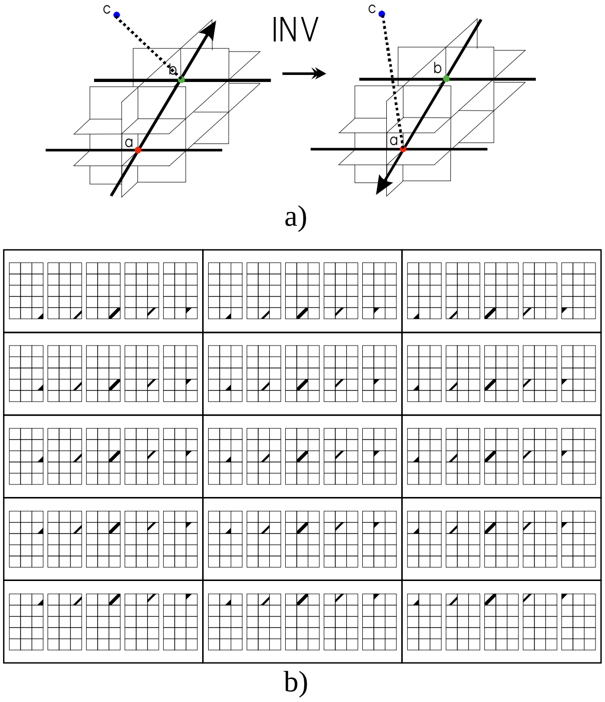
The iconical representation of this operation is shown in figure 13 b).

The PROLOG facts of the Inversion operation would be for instance (figure 13 b): `inv(ulf,[drb])`; `inv(usf,[dsb])`; etc.

Spin. The spin operation (SP) is the result of rotating 180 degrees the RS by the axis which passes by the two main points *a* and *b* of the RS (figure 10). This operation implies that anything which were up and right will be down and left, respectively, after applying the operation.

The iconical representation of the spin operation is presented in figure 14.

The PROLOG facts of the Spin operation would be for instance: `sp(ulf,[drf])`; `sp(usf,[dsf])`; etc.



a)

b)

Figure 13: ***a) The inverse operation, and b) its 2-D iconical representation.***

Homing. In the homing (HM) operation we ask about the point a with respect to the RS formed by *bc*. (See the transformation in Figure 15 a) and the iconical representation in Figure 15 b).

The algebraic notation is: HM(*c wrt ab*) = a *wrt bc*.

The PROLOG facts of the Homing operation would be for instance (see figure 15 b): `hm(ulf,[dlb])`; `hm(usf,[dsb])`; etc. Here disjunction appear, for example: `hm(us,[dlf, dsf, drf, dl, ds, dr, dln, dsn, drn, dla, dsa, dra, dlb, dsb, drb]`).

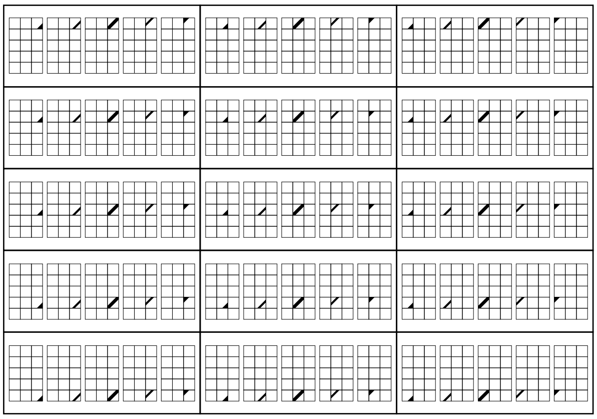
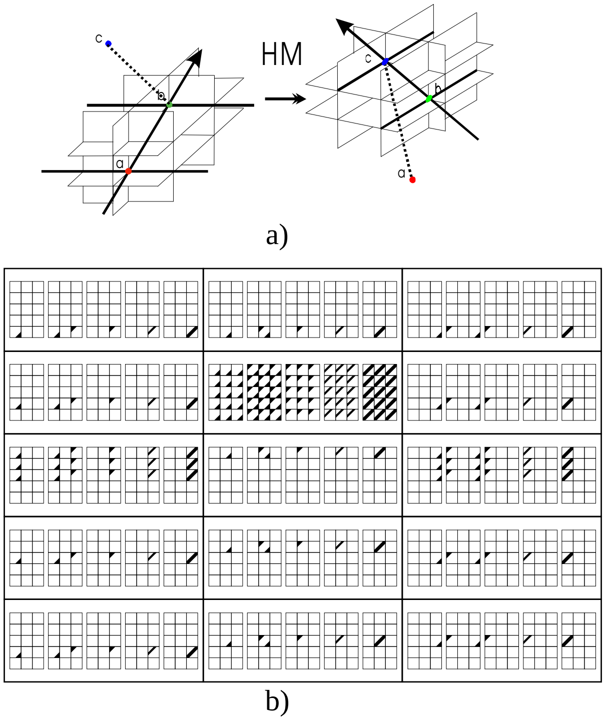


Figure 14: ***The 2-D iconical representation of spin operation.***

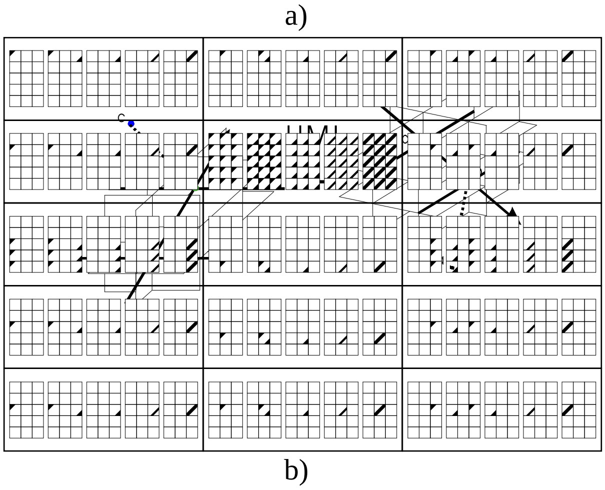
When we complete the HM operation of the 3-D orientation relationship left-front-b-height (first row, second column), it happens that *c* and *b* are in the same plane. Therefore, we reduce the five heights to three (a-height, between and b-height are the same plane). In this case, the result of the HM operation is a disjunction because we have considered the two cases in which the spin operation is not applied and when the spin operation is applied.



a)

b)

Figure 15: ***a) The homing operation and, b) its 2-D iconical representation.***



a)

b)

Figure 16: ***a) The homing inverse operation and, b) its 2-D iconical representation.***

Homing Inverse. The homing inverse (HMI) operation is the result of applying the INV operation after the HM operation.

The algebraic notation is: HMI(*c wrt ab*) = INV(HM(*c wrt ab*)) = *a wrt cb*.

The homing inverse operation is presented in figure 16.

The PROLOG facts of the Homing Inverse operation would be for instance: `hmi(ulf,[ulf])`; `hmi(usf,[usf])`; etc. Also disjunction appear here, for example: `hmi(us,[ulf, usf, urf, ul, us, ur, uln, usn, urn, ula, usa, ura, ulb, usb, urb])`.

Shortcut. In the shortcut (SC) operation we ask about the point *b* with respect to the *ac* RS.

The algebraic notation is: SC(*c wrt ab*) = *b wrt ac*.

The shortcut operation and its 2-D iconical representation is presented in figure 17.

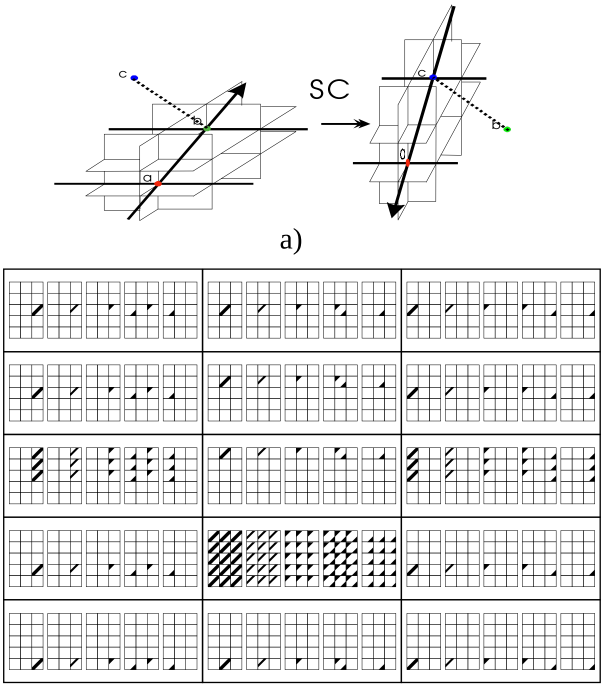
The PROLOG facts of the Shortcut operation would be for instance: `sc(ulf,[brn])`; `sc(usf,[bsn])`; etc.

Shortcut Inverse. The shortcut inverse (SCI) operation is the result of applying the INV operation after the SC operation to the original orientation representation.

The algebraic notation is: SCI(*c wrt ab*) = INV(SC(*c wrt ab*)) = *b wrt ca*.

The shortcut inverse operation is presented in figure 18.

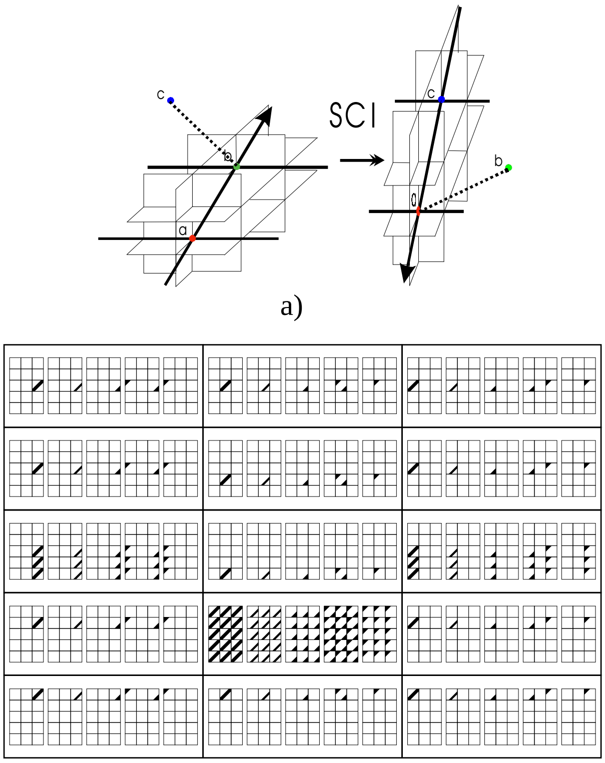
The PROLOG facts of the Shortcut Inverse operation would be for instance: `sci(ulf,[brn])`; `sci(usf,[bsn])`; etc.



a)

b)

Figure 17: ***a) The shortcut operation; and b) its iconical representation.***



a)

b)

Figure 18: a) ***The shortcut inverse operation and b) its 2-D iconical representation.***

Algebraic Combinations of Operations

There is a strong inner resemblance between the homing and the shortcut operations, for which only one table is necessary [14, 17], because all the results found in the homing 3-D iconical representation are found in the shortcut 3-D iconical representation.

We notice that the resulting operation of combining two operations is other operation. For example in INV(SC(x)) SC is used first and INV after, and it results SCI(x); in SC(INV(x)) INV is used first and SC after, and HM(x) is the result.

An important feature of these operations is their idempotent property [2]. An operation is idempotent of level two if it is necessary to apply the operation twice to the original relationship in order to get the original relationship again. The inverse, homing and shortcut operations are idempotent of level two. For instance, INV(INV(*c wrt ab*)) = *c wrt ab*. An operation is idempotent of level three if it is necessary to apply the operation three times to the original relationship to get the original relationship again. The homing and shortcut inverse operations are idempotent of level three. For instance, HM(HM(HM(*c wrt ab*)))= *c wrt ab*.

The operations which are idempotent of level three have another property: the application of the homing operation twice is equivalent to the application of the shortcut inverse operation once (i.e. SCI(HM(*c wrt ab*))= *c wrt ab*), and the application of the shortcut inverse operation twice is equivalent to apply the homing operation once (i.e. HM(SCI(*c wrt ab*))= *c wrt ab*).

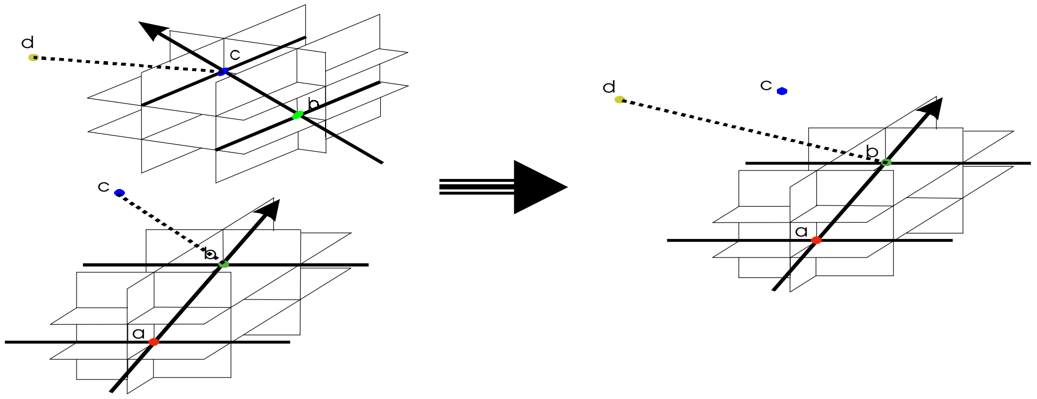


Figure 19: ***The inference process.***

The Basic Step of the Inference Process

The Basic Step of the Inference Process (BSIP) used in Freksa and Zimmermann's qualitative orientation approach is defined such as (figure 19): "given two relationships *c wrt ab* and d *wrt bc*, we want to obtain the relationship *d wrt ab*"

The inference process among the coarse qualitative orientation relationships has been represented as an inference table of 75 x 75 entries using our approach. The first column of the table shows the relationship *c wrt ab* and the first row of the table depicts the relationship *d wrt bc*. The relationship obtained in the composition table is *d wrt ab*. This inference table is complete in the sense that it will be possible to find the composition of any of the seventy-five relationships defined in the coarse division of the space. The result of the inference is always one of the seventy-five relationships or a disjunction of them. Successive compositions of disjunction of relationships with another relationship can be accomplished by compositions of each single relationship belonging to the disjunction and then adding the results.

Conceptual neighbourhood

An important concept in the common-sense reasoning is the conceptual neighbourhood, which is not a concept exclusively related to qualitative reasoning. In all spatial perception representation, and identification situations are presented [3]. Moreover, the neighbourhood of objects and conceptual neighbourhood of relations between objects provide very useful information for spatial reasoning.

Qualitative spatial reasoning (for instance the concept of orientation, distance, the treatment of extended objects, topology, and so on) divides space into qualitative regions. Conceptual neighbourhood may be explained in terms of two concepts:

1) Persistence or continuity of the properties inside the same region;

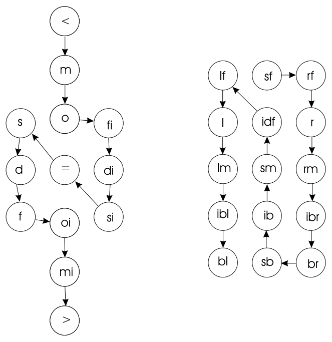
2) Discontinuity among regions that correspond to the qualitative change determined by important aspects.

Intuitively, inside each qualitative region the same "feature" persists (for instance, in the qualitative region "front/close", which implies orientation and distance information, every point in the region shares the property of being in that region). Moreover, there exist boundaries among regions where the discontinuity presented corresponds to the changes in quality.

In temporal reasoning, Freksa defines in [3] that: "Two relationships between pairs of events are conceptual neighbours if they can be directly transformed into one another by continuous deformation (i.e. shortening or lengthening) of the events".

In spatial reasoning the conceptual neighbourhood may be defined such that [2]:

"Two qualitative spatial regions, A and B, are conceptual neighbours if, and only if, in a continuous translation from a position of the qualitative region A to a position of the qualitative region B, there does not exist a position belonging to another qualitative region C".



a) b)

Figure20: ***Linearization of the topological arrangement in order to built the inference table for a) Allen's temporal logic and b) Freksa and Zimmerman RS.***

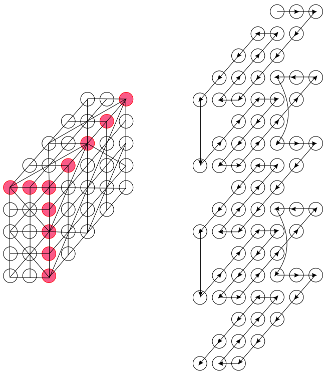


Figure 21: ***The conceptual neighbourhood in 3D.***

Lots of benefits are obtained from the use of the neighbourhood concept. Among others, the following ones expressed in [3]:

The structure of knowledge according to the conceptual neighbourhood of temporal and spatial relations allows the integration of coarse and fine knowledge. Freksa organises the world hierarchically according to the level of detail that is available [3]. The prerequisite for employing this approach is monotonicity of the reasoning process involved, i.e., the inferences carried out on the basis of coarser knowledge must remain valid when additional knowledge becomes available. Due to monotonicity properties of temporal and spatial domains, neighbouring initial conditions result at worst in neighbouring consequences; thus small uncertainties in the initial conditions do not cause drastically wrong conclusions.

It permits an extension of the representation scheme (or inference tables) in such a way that it is robust against variations or small errors in the input knowledge.

The inference table achieved by using the linear neighbourhood ensures that:

a) The relations within a disjunction always form a conceptual neighbourhood.

b) In many cases, a transition to neighbouring initial conditions results in the identical conclusion or in a subset or superset of an inference neighbourhood.

c) In no case, a transition among neighbouring initial conditions results in a jump between non-neighbouring conclusions. d) The inference table shows much symmetry that may be utilised in the inference process.

d) The inference table shows much symmetry that may be utilised in the inference process.

The inference table

It is important to remark that the inference tables are arranged in such a way that neighbouring rows and columns always correspond to conceptually neighbouring relations. As it was pointed out in [4], the topological view of the icons provides an explicit reasoning about neighbourhood. Examples of this fact are the 13 Allen's temporal relationships and Freksa and Zimmermann's division of the space. However, in order to visualise the reasoning procedure by means of tables, this topological arrangement is linearized in such a way that only a subset of the actual neighbourhood relationships is reflected. Figure 20 a) and b) corresponds to linearization of those topological representations. The arrangement of relationships in rows and columns of the fine 75 x 75 table of figure 8 a) is obtained following this idea, as it is shown in figure 21.

As it will be impossible to show the complete table in this paper, we show the first fifteen entries in the first column and the first fifteen entries in the first row (figure 22) with the aim to show how the table is built.

The PROLOG facts of the Inference Table would be for instance (see figure 28): `inf_table(ulf, ulf, [ulf, ul, uln, ula, ulb]); inf_table(usf, usf,[usf]);` etc.

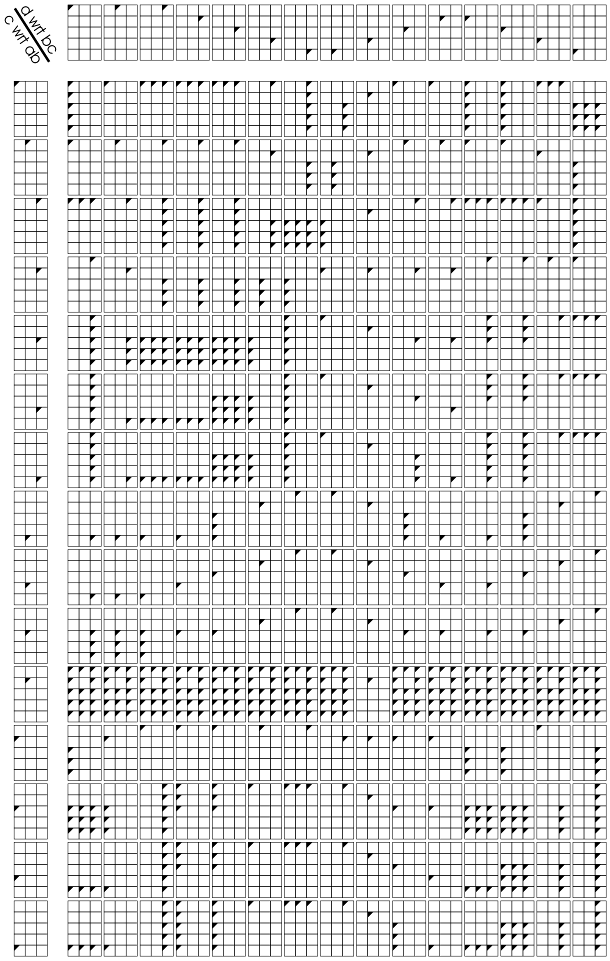


Figure 22: ***A part of the inference table (first fifteen rows and first fifteen columns).***

The Full Inference Process

The Full Inference Process (FIP) is the other part in the reasoning process. It consists of repeating the BSIP as many times as possible, with the initial information and the information provided by some BSIP, until no more information can be inferred.

When more relationships among several spatial landmarks are provided, then the FIP is necessary.

In this section a Constraint Solver (CS) for qualitative orientation will be explained. This CS, which implements a path consistency algorithm, is based on Constraint Logic Programming (CLP) extended with Constraint Handling Rules (CHRs).

Qualitative Orientation as a Constraint Satisfaction Problem

Our qualitative orientation model implies three spatial objects (*a*, *b* and *c*), therefore the constraints which deal with this information are tertiary. The Constrain Satisfaction Problem (CSP) is reformulated for these tertiary constraints (*c wrt ab*) such that: given a set of variables $\{X_1,...,X_n\}$, a discrete and finite domain for each variable $\{D_1,...,Dn\}$, and a set of constraints $\{c_{c,ab}$ $(X_c,X_a,X_b)\}$, which define the relationship between every group of three variables (X_c,X_a,X_b), $(1\leq a<b<c\leq n)$; the problem is to find an assignment of values (v_c,v_a,v_b), $v_i \in D_i$ to variables such that all constraints are satisfied, i.e. $c_{c,ab}$ (X_c,X_a,X_b) is true for every a,b,c $(1\leq a<b<c\leq n)$.

We redefine a network of tertiary constraints as path consistent for triples of nodes (c,a,b) and all paths a-b-i_1-...-i_{n-1}-i_n between them, if the direct constraint $c_{c,ab}$ is tighter (has less disjunction) than the indirect constraint along the path, i.e. the composition of constraints $c_{i1,ab} \otimes ... \otimes c_{c,i_{n-1}i_n}$ along the path. In order to determine whether a graph is complete we repeatedly compute the following operation:

$$c_{d,ab} := c_{d,ab} \oplus c_{c,ab} \otimes c_{d,bc}$$

until a fixed point is reached.

The Path Consistency Algorithm for Qualitative Orientation

The following constraint satisfaction algorithm for complete disjunctive tertiary constraints networks is defined using PROLOG extended with CHRs. PROLOG provides backtracking and CHRs are used to implement path consistency at a high level of abstraction.

The constraint $c_{c,ab}$ is represented in the algorithm by the predicate `ctr_orient(C,A,B,Rel)`, where Rel is the list of primitive spatial orientation relationships forming the disjunctive constraint. The operation of a path consistency, is implemented by means of two kinds of CHRs. The part of the operation corresponding to the intersection $c_{d,ab} \oplus ...$ is implemented by simplification CHRs:

```
ctr_orient(C,A,B,R1),
   ctr_orient(C,A,B,R2) <=>
   intersection(R1,R2,R3) |
   ctr_orient(C,A,B,R3)
```

The part corresponding to the $c_{c,ab} \otimes c_{d,bc}$ is implemented by propagation CHRs:

```
ctr_orient(C,A,B,R1),
   ctr_orient(D,B,C,R2) ==>
   composition(R1,R2,R3) |
   newc(D,A,B,R3)
```

Termination is guaranteed because the simplification rule replaces R1 and R2 by the result R3 of intersecting R1 with R2 (and R3 is the same as R1 or R2 or smaller) and because propagation CHRs are never repeated for the same constraint goals as it will be shown.

The algorithm is based on the algorithm developed in [2]. The optimisation introduced in the algorithm of [11] (named PC-2) has also been included. This optimisation is based on the idea that the constraint $c_{c,ab}$ can be computed as the converse $c_{c,ab}$ if it is needed (by applying the inverse operation to the corresponding relationship), which saves half of the computation.

Here disjunction could also appear in first and second argument; in those cases operations of union, intersection and composition of disjunctive ternary constraints must be used.

The operations of *union*, *intersection* and *composition* are formally redefined for disjunctive ternary constraints in this section.

The *union* of disjunctive ternary constraints can be formulated as follows:

$c_{c,ab} \cup c'_{c,ab} := c\{r_1,..., r_n\}ab \vee c\{s_1,..., s_m\}ab =$
$\quad c\{r_1,..., r_n\} \cup c\{s_1,..., s_m\}ab$

The *intersection* is defined as follows:

$c_{c,ab} \cap c'_{c,ab} := c\{r_1,..., r_n\}ab \wedge c\{s_1,..., s_m\}ab =$
$\quad c\{r_1,..., r_n\} \cap c\{s_1,..., s_m\}ab$

The *composition* of disjunctive ternary constraints is defined as follows:

$c_{c,ab} \otimes c'd_{,ab} := c\{r_1,...,r_n\}ab \wedge d\{s_1,...,s_m\}bc =$
$\quad d\{r{\otimes}s|r \in \{r_1,..., r_n\}, s \in \{s_1,..., s_m\}\}ab$

Where ⊗ is the basic step of inference process.

All these operations are associative.

A disjunctive ternary constraint $c_{c,ab}$ between the variables a, b and c, also written $c\ \{r_1,..., r_n\}\ ab$, is a disjunction $(c\ r_1\ ab) \vee ... \vee (c\ r_n\ ab)$ where each r_i is a relation that is applicable to *c* and *ab*.

The Algorithm. Here a part of the path consistency algorithm to propagate compositions of disjunctive qualitative orientation relationships appear.

% Constraint declarations and definitions

```
constraints (ctr_orient)/4,
   (ctr_orient)/6.
label_with ctr_orient(N,C,A,B,Rel,I)
   if N>1.
ctr_orient(N,C,A,B,Rel,I) :-
   member(R,Rel),
   ctr_orient(1,C,A,B,[R],I).
```

% Initialise

```
  ctr_orient(C,A,B,Rel) <=>
     length(Rel,N)
     |ctr_orient(N,C,A,B,Rel,1).
% Special cases
  ctr_orient(N,C,A,B,R,I) <=> empty(R)
     | false.
  ctr_orient(N,C,A,A,R,I) <=> true.
  ctr_orient(N,C,C,B,R,I) <=>
     contains_equality_a(R) | true.
  ctr_orient(N,C,A,C,R,I) <=>
     contains_equality_b(R) | true.
  ctr_orient(N,C,A,B,R,I) <=> N=75 |
     true.
% Intersection
  ctr_orient(N1,C,A,B,R1,I),
     ctr_orient(N2,C,A,B,R2,J) <=>
     intersection(R1,R2,R3),
     length(R3,N3), K is min(I,J) |
     ctr_orient(N3,C,A,B,R3,K).
  ctr_orient(N1,B,C,A,R1,I),
     ctr_orient(N2,C,A,B,R2,J) <=>
     hm_op(R1,R11),intersection(R11,R2,R
     3), length(R3,N3), K is min(I,J) |
     ctr_orient(N3,C,A,B,R3,K).
  ctr_orient(N1,C,A,B,R1,I),
     ctr_orient(N2,B,C,A,R2,J) <=>
     hm_op(R2,R22),intersection(R1,R22,R
     3), length(R3,N3), K is min(I,J) |
     ctr_orient(N3,C,A,B,R3,K).
  ...
% Composition
  ctr_orient(N1,C,A,B,R1,I),
     ctr_orient(N2,D,B,C,R2,J) ==> I=1,
     composition_op(R1,R2,R3),
     length(R3,N3), K is I+J |
     ctr_orient(N3,D,A,B,R3,K).
  ctr_orient(N1,B,C,A,R1,I),
     ctr_orient(N2,D,B,C,R2,J) ==> I=1,
     singleton(R1), hm_op(R1,R11),
     composition_op(R11,R2,R3),
     length(R3,N3), K is I+J |
     ctr_orient(N3,D,A,B,R3,K).
  ctr_orient(N1,C,A,B,R1,I),
     ctr_orient(N2,C,D,B,R2,J) ==> I=1,
     singleton(R2), hm_op(R2,R22),
     composition_op(R1,R22,R3),
     length(R3,N3), K is I+J |
     ctr_orient(N3,D,A,B,R3,K).
  ...
```

Two predicates, `ctr_orient` of arity 4 and 6, are declared in **% Constraint declarations and definitions**. Predicates `ctr_orient/4` are the kind of constraints introduced initially as qualitative orientation information. The predicates of type `ctr_orient/4` are translated into the predicates `ctr_orient/6` by rule in **%Initialize** where the length (N) of the relationship is added as well as the length of the shortest path from which the constraint is derived. A path length equal to 1 means that the constraint is direct, that is, it is user-defined, not obtained from derivation. In **% Special cases** the first one will avoid compositions between constraints which do not give more information (the last rule here) because all the qualitative orientation primitive relationships are included in the disjunction. The last argument is used to restrict the propagation CHRs to involve at least one direct constraint.

The constraints will be treated by the CLP clause if the relation, `Rel`, represents a disjunction of primitive relationships. In `member(R,Rel)` non-deterministically chooses one primitive constraint, `R`, from the disjunctive constraint `Rel` which implements the backtrack search part of the algorithm.

% Special cases are simplification CHRs. The first rule detects inconsistent constraints. When the constraint relates three spatial objects with an empty relationship, the constraint is substituted by the built-in predicate false and the full predicate fails. If it is not the expected behaviour when inconsistent information appears, substituting the inconsistent constraint can change this rule by true, that is, deleting this constraint. The second rule deletes constraints that contain only one point as base for the reference system. Third and fourth rules delete constraints that contain equality, that is, when the third point of the relationship is equal to the first or second points of the RS, respectively. And last rule deletes constraints that contain the full primitive qualitative orientation relationship set.

Simplification CHRs (rules in **% Intersection**) perform intersections which permit the simplification of redundant information. The first rule in **% Intersection** implements intersection in the way such as it is originally defined in the first rule in point 5.2, that is, given two constraints which relate the same three spatial objects, the more restricted relationship between both constraints is calculated by the predicate `intersection(R1,R2,R3)` and these constraints are substituted by a new one which relates the same three objects with the new relationship `R3` among them.

By applying the five operations (HM, SC, INV, HMI and SCI) to the first constraint of the two which are in the head of the original intersection rule (point 5.2), it is possible to obtain the orientation information among the same three spatial objects. (Only hm_op rule is exposed in this algorithm) Therefore, it is possible to calculate intersection if the corresponding operations are applied to the relationship or disjunction of relationships in the guard part of the rules. As it was formalised in point 5.2, the application of the above operations to a disjunction of relationships is equivalent to the application of these operations to each relationship included in the disjunction of relations.

It is important to notice that the operations that are idempotent of level three, namely HM and SCI, have a different treatment to the rest of operations. If the HM operation is applied to the usual definition of the orientation ternary constraint (*c wrt ab*), the order of the variables in the constraint becomes (*a wrt bc*). However, if the HM operation is applied again to the result (to *a wrt bc*), instead of obtaining the original result, (which is the one expected when the operations are idempotent of level two), the relationship (*b wrt ca*) is achieved. This relationship is the result of applying the SCI operation to the original relationship. Therefore, when the HM operation is applied to the first constraint to obtain the new order among the variables, the SCI operation is applied to the relationship of this constraint in the guard part of the rule, in order to achieve the correct result. With the SCI operation also happens.

Second Simplification CHR corresponds to the above explanation third simplification CHR corresponds to the case in which the operations are applied to the second constraint of the two which appear in the head of the original intersection rule (the first rule in point 5.2). A total of 11 simplification CHRs to compute intersection would be defined.

Propagation CHRs (rules in **% Composition**) perform compositions. The first rule implements the composition as originally is defined in (the second rule in point 5.2). In a similar way to what it happens to the simplification rule (the first rule in point 5.2), the application of the five operations to the first constraint of the two which define the head of the original composition rule define the next six CHRs rules. If they are applied to the second constraint, the other six more. Hence, a total of 11 propagation CHRs are needed to cover all possible combinations of constraints. The problem of those operations that are idempotent of level three is repeated here and it is solved in the same way.

In CHRs (rules in **% Composition**) another optimisation is introduced. It consists of restricting one of the two constraints involved in the propagation to be disjunction-free by adding to the guard a check that guarantees that its corresponding relationship is singleton. This not only reduces the average size of the resulting constraint but also makes composition more efficient.

Conclusions and Future Works

In this paper, we have reached a model for representing 3-D qualitative orientation and we have defined some operations to work with it.

We have left out of this paper some future work:

(1)The integration of different levels of granularity

(2)The application of the 3-D orientation model to mobile robots with an arm manipulator on it.

Acknowledgements

This work has been partially supported by CICYT under grant number TAP99-0590-002-02.

References

[1] Chang, S.K., Jungert, E., "A Spatial Knowledge structure for image information systems using symbolic projections", Proceedings of the National Computer Conference, Dallas, Texas, November 26, pag. 79-86, 1986.

[2] Escrig, M.T., Toledo, F., Qualitative Spatial Reasoning: theory and practice. Application to Robot Navigation. IOS Press, Frontiers in Artificial Intelligence and Applications, ISBN 90 5199 4125. 1998.

[3] Freksa, c., "Conceptual Neighbourhood and its role in temporal and spatial reasoning", Proceedings of the IMACS Workshop on Decision Support Systems and Qualitative Reasoning, pag. 181-187, 1991.

[4] Freksa, c., "Temporal reasoning based on semi-intervals", in Artificial Intelligence, vol. 54, pag. 199-227, 1992.

[5] Freksa, c., "Using Orientation Information for Qualitative Reasoning", in a. U. Frank, I. Campari, and U. Formentini (editors). Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. Proceedings of the International Conference on GIS-From Space to Territory, Pisa, volume 639 of Lecture Notes in Computer Science, Springer, Berlin, pag. 162-178, 1992.

[6] Freksa, c., Zimmermann, K., "On the Utilization of Spatial Structures for Cognitively Plausible and Efficient Reasoning", in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pag. 18-21, 1992.

[7] Frank, a.U., "Qualitative Spatial Reasoning with cardinal directions", in Proceedings of the Seventh Austrian Conference on Artificial Intelligence, Wien, Springer, Berlin, pag. 157-167, 1991.

[8] Guesgen, H.W., "Spatial reasoning based on Allen's temporal logic", Technical Report TR-89-049, International Computer Science Institute, Berkeley, 1989.

[9] Hernández, D., ``Qualitative Representation of Spatial Knowledge". In volume 804 of Lecture Notes in Artificial Intelligence. Ed. Springer-Verlag, 1994.

[10] Kak, a., "Spatial Reasoning", AI Magazine, vol.9, no. 2, p. 23, 1988.

[11] Mackworth, a.K., ``consistency in Networks of relations", journal of Artificial Intelligence 8: pag. 99-118, 1977.

[12] Mukerjee, a. and Joe, G., "A Qualitative Model for Space". In the 8th American Association for Artificial Intelligence, pag. 721-727, 1990.

[13] Pacheco, J., Escrig, M.T., Toledo, F. “A model for Representing and Reasoning with 3-D Qualitative Orientation” in the 4th Catalan congress of Artificial Intelligence, 2001.

[14] Pacheco, J., Escrig, M.T., Toledo, F. “An Approach to 3-D Qualitative Orientation of Point Objects” in the 9th Conference of Spanish Association of Artificial Intelligence, 2001.

[15] Pacheco, J., Escrig, M.T., Toledo, F. “Representing and Reasoning on Three-Dimensional Qualitative Orientation Point Objects” Proceedings in the 10th Potuguese Conference on Artificial Intelligence, EPIA 2001, Porto Portugal. In Pavel Brazdil, Alípio Jorge (ed.). Progress in Artifitial Intelligence. Lecture Notes in Artifitial Intelligence vol 2258, Springer, pag. 298-305, 2001. ISBN: 3-540-43030-X

[16] Pacheco, J., Escrig, M.T., Toledo, F. “Three-Dimensional Qualitative Orientation Point Objects: Model and Reasoning” Workshop on Logic programming for artifitial Intelligence and Information System (LPAI) EPIA 2001, Porto Portugal. Proceedings edited by José Alferes ans Salvador Abreu, pag. 59-74, 2001.

[17] Pacheco, J., Escrig, M.T., Toledo, F. “The First Steps towards Reasoning on 3-D Qualitative Orientation” in Inteligencia Artificial, Revista Iberoamericana de inteligencia Artificial. No 15 (2002) ISSN: 1137-3601