Several Models on Qualitative Motion as instances of the CSP*

M. Teresa Escrig*, Lledó Museros*Ψ, Julio Pacheco* and Francisco Toledo*

*Universitat Jaume I, Egineering and Computer Science Department,
Campus de Riu Sec, 12170 Castellón (Spain)
ΨAlicer, Ceramic Design Tecnology Institute, Avda. Mar 42, E-12003 Castellón (Spain)
email: {escrigm, museros, pacheco, toledo}@icc.uji.es

Abstract

Space and time has been integrated into three qualitative models in this article: a qualitative motion model which integrates topology and time, a qualitative velocity model in 2-D and a qualitative velocity model in 3-D. The integration has been accomplished thanks to the definition of an approach with the following three steps: (1) the definition of the algebra of the spatial aspect to be integrated. The representation of each aspect is seen as an instance of the Constraint Satisfaction Problem (CSP); (2) the definition of the Basic Step of the Inference Process (BSIP) for each spatial aspect to be integrated. In general, the BSIP consists on given two relationships which relate three objects A, B, and C (one object is shared among the two relationships, for instance B), we will find the third relationship between objects A and C; and (3) the definition of the Full Inference Process (FIP) for each spatial aspect to be integrated which consists on 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. The corresponding algorithm is solved using Constraint Logic Programming extended with Constraint Handling Rules (CLP+CHRs) as tool.

Keywords: Qualitative reasoning, Spatial Reasoning, Temporal Reasoning, Qualitative motion.

1 Introduction

Historically, the effort of scientists has consisted in measuring and naming everything in the environment, in order to understand and classify the physical world. However, human way of thinking is qualitative in nature, and in the last years a contrary effort has been perceived. In order to model and manage uncertainty information and commonsense reasoning, a set of quantitative models have been developed for dealing with spatial concepts, such as orientation in 2-D [Guesgen 89], [Jungert 92], [Mukerjee & Joe 90], [Freksa 92, Freksa & Zimmermann 92], [Hernández 94], orientation in 3-D [Pacheco et al. 01], named distances [Zimmermann 93], [Jong 94], [Clementini et al. 95], [Escrig & Toledo 00], compared distances [Escrig & Toledo 01], cardinal directions [Frank 92], and so on. See [Escrig & Toledo 98] for a review of these approaches. Some approaches represent and reason with one or two of the previous mentioned spatial aspects.

Spatio-temporal data are dealt in several approaches, such as [Belussi et al. 98], [Böhlen et al. 99], [Grumbach et al. 98], [Raffaetà & Fruehwirth 01], and [Mukerjee & Schnorrenberg 91], among others.

The concept of qualitative motion has also been dealt in [Zimmermann & Freksa 93], and [Musto et al. 99, 00]. In these approaches, motion has been modelled as a

Revista Iberoamericana de Inteligencia Artificial. No.17 (Otoño 2002), 55-71, ISNN: 1137-3601 @AEPIA (www.aepia.org)

sequence of changes of positions, taking into account conceptual neighbourhood, but without integrating the concept of space and time into the same model.

In [Escrig & Toledo 98], many spatial aspects such as orientation in 2-D and in 3-D, named and compared distances, and cardinal directions, have been integrated into the same model, thanks to represent each spatial aspect as an instance of the Constraint Satisfaction Problem (CSP) and to reason using Constraint Logic Programming extended with Constraint Handling Rules (CLP+CHRs) as tool. We are going to use this approach for integrating space and time in this article.

The CSP can be defined for binary constraints such as: given a set of variables $\{X_1,...,X_n\}$, a discrete and finite domain for each variable $\{D_1,...,D_n\}$, and a set of binary constraints $\{c_{ij}(X_i,X_j)\}$, which define the relationship between every pair of variables X_i,X_j $(1 \leq i<j\leq n)$; the problem is to find an assignment of values $<v_1,...,v_n>$, $v_1 \in D_i$, to variables such that all the constraints are satisfied.

Generate and test and backtracking are algorithms which solve the CSP, although in a very inefficient way. These algorithms have an exponential cost. Research in the field tries to improve efficiency of the backtracking algorithm. Some of the improved algorithms are based on the idea of making explicit the implicit constraints by means of the constraint propagation process. Unfortunately, the complete constraint propagation process is also hard, therefore the process is approximated by local constraint propagation, as path consistency. If the constraint graph is complete (that is, there is a pair of arcs, one in each direction, between every pair of nodes) it suffices to repeatedly compute paths of two steps in length at most. This means that for each group of three nodes (i,k,j) we repeatedly compute the following operation until a fix point is reached [Fruehwirth 94]:

$$c_{ij} := c_{ij} \oplus c_{ik} \otimes c_kj \qquad (1)$$

This operation computes the composition of constraints (⊗) between nodes *ik* and *kj,* and the intersection (⊕) of the result with constraints between nodes *ij*. The complexity of this algorithm is $O(n^3)$, where *n* is the number of nodes in the constraint graph (that is, the number of objects involved in the reasoning process) [Mackworth & Freuder 85].

CHRs are a tool which helps to write the above algorithm. They are an extension of CLP which facilitate the definition of constraint theories and algorithms which solve them. They facilitate the prototyping, extensions, specialization and combination of Constraint Solvers [Fruehwirth 94]. The composition part of the formula $(c_{ik} \otimes c_{kj})$ is implemented by propagation CHRs which are of the form:

$$H_1,..., H_i ==> G_1,...Gj \mid B_1,...,B_k \quad (i>0, j\geq 0, k\geq 0) \qquad (2)$$

It means that if a set of constraints matches the head of a propagation CHR $(H_1,..., H_i)$ and the guards $(G_1,...Gj)$ are satisfied, then the set of constraints $(B_1,...,B_k)$ is added to the set of initial constraints $B_1,...,B_k$.

The part of the formula which refers to intersection $(c_{ij} \oplus)$ is implemented by simplification CHRs which are of the form:

$$H_1,..., H_i <=> G_1,...Gj \mid B_1,...,B_k \quad (i>0, j\geq 0, k\geq 0) \qquad (3)$$

It means that if a set of constraints (H') matches the head of a simplification CHR $(H_1,..., H_i)$ and the guards $(G_1,...Gj)$ are satisfied, the set of constraints $(H_1,..., H_i)$ is substituted by the set of constraints $(B_1,...,B_k)$. The set of constraints $(B_1,...,B_k)$ is simpler than the set of constraints $(H_1,..., H_i)$ and preserves logical equivalence. CHRs are included as a library in ECLiPSe [Brisset et al. 94].

The approach described in [Escrig & Toledo 98] defines three steps for each spatial aspect to be integrated:

(1) The definition of the algebra of the spatial aspect to be represented. The algebra includes a) the explanation of the meaning of the relations or constraints to be establish among spatial objects; b) how many spatial objects are involved in the relation[1]; and c) the operations which can be defined on these relationships[2].

(2) The definition of the Basic Step of the Inference Process (BSIP) for each spatial aspect to be integrated. In general, the BSIP consists on given two relationships which relate three objects A, B, and C (one object is shared among the two relationships, for instance B), we will find the third relationship between objects A and C.

(3) the definition of the Full Inference Process (FIP) for each spatial aspect to be integrated. When several relationships among spatial landmarks are provided, the

[1] Normally the relationships will be binary or tertiary.
[2] If the relationships are binary, the converse operation will be defined. If the relationships are tertiary, five operations can be defined [Escrig & Toledo 98].

FIP will repeat 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.

Following this approach, and therefore the three steps above described, we are going to integrate several spatio-temporal models: topology and time (in section 2), velocity in 2-D (in section 3); and velocity in 3-D (in section 4).

Due to the fact that all these spatio- temporal models are dealt using this common representation, they are also integrated with the previous mentioned spatial aspects: orientation in 2-D and in 3-D, named and compared distances, cardinal directions, and with any other spatial aspect which in the future will be dealt with using this approach.

2 Integrating topology and time: motion

Motion is seen in this section as a form of spatio-temporal change. A model for integrating qualitatively time and topological information is defined. It will allow us to reason about dynamic worlds in which spatial relations between regions may change with time.

2.1 The algebra

For developing the algebra on motion, first of all we are going to introduce the algebra on topology, then the algebra on time, and afterwards the algebra on topology integrated with time.

2.1.1 The algebra on topology

For the topological calculus we are going to used the approach by [Isli et al. 00]. This topological calculus is presented as an algebra alike to Allen's temporal interval algebra [Allen 83]. It allows us to represent and reason about point-like, linear and areal entities or objects, which are the variables. This way of reasoning also permits the use of different granularities of the same map. The information to be represented is the topological relationship of an object *h1* wrt another object *h2*. The calculus defines 9 topological relations: touch (T), cross (C), overlap (O), disjoint (D), equal (E), completely-inside (CI), touching-from-inside (TFI), completely-inside-inverse (CIi), and touching-from-inside-inverse (TFIi), which are formally defined in table 1. These relationships are mutually exclusive, that is, given 2 entities, the relation between them must be one and only one of the nine relations defined. In table 1, a topological relation r between two entities *h1* and *h2*, denoted by (h_1,r,h_2), is defined on the right hand side of the equivalence sign in the form of a point-set expression. In order to understand these definitions, it is necessary the following basic topological concepts.

Definition 1. The boundary of an entity h, called δh, is defined as:
- We consider the boundary of a point-like entity to be always empty.
- The boundary of a linear entity is the empty set in the case of a circular line or the two distinct endpoints otherwise.
- The boundary of an area is the circular line consisting of all the accumulation points of the area.

Definition 2. The interior of an entity h, called h^o is defined as $h^o = h - \delta h$.

Definition 3. The function *dim*, which returns the dimension of an entity of either the types we consider, or the dimension of the intersection of two or more of such entities, is defined as follows:

If $S \neq \emptyset$ then
  dim(S) = 0: if S contains at least a point and no lines and no areas.
  dim(S) = 1: if S contains at least a line and no areas.
  dim(S) = 2: if S contains at least an area.
Else dim(S) is undefined.

The operation to be defined in such a binary relation is the converse operation. The converse table for topological relations can be found in [Isli et al. 00].

2.1.2 The algebra on time for change

We define a time algebra in which variables represent time points. There are five primitive constraints: prev, next, <<, >>, ==. These primitive constraints are defined as follows:

Definition 4. Given two time points, t and t', t **==** t' iff it has not occurred a change between t and t' (or between t' and t) on any relation.

Definition 5. Given two time points, t an t', t' **next** t iff t' > t and some relation or relations have changed to a neighbor relation between t and t'.

Definition 6. Given two time points, t and t', t' **prev** t iff t' < t and some relation or relations have changed to a neighbor relation between t and t'.

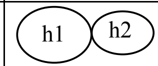
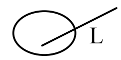
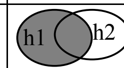
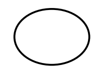
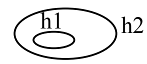
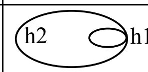
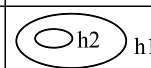
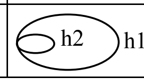
Relation	Definition	Iconically
T	$(h_1,touch,h_2) \leftrightarrow h°_1 \cap h°_2 = \emptyset \wedge h_1 \cap h_2 \neq \emptyset$	
C	$(h_1,cross,h_2) \leftrightarrow dim(h°_1 \cap h°_2) = max(dim(h°_1), dim(h°_2)) - 1 \wedge h_1 \cap h_2 \neq h_1 \wedge h_1 \cap h_2 \neq h_2$	
O	$(h_1,overlap,h_2) \leftrightarrow dim(h°_1) = dim(h°_2) = dim(h°_1 \cap h°_2) \wedge h_1 \cap h_2 \neq h_1 \wedge h_1 \cap h_2 \neq h_2$	
D	$(h_1, disjoint, h_2) \leftrightarrow h_1 \cap h_2 = \emptyset$	
E	Given that $(h_1,in,h_2) \leftrightarrow h_1 \cap h_2 = h_1 \wedge h°_1 \cap h°_2 \neq \emptyset$: if (h_2,in,h_1) then $(h_1,equal,h_2)$	
CI	Given that (h_1,in,h_2) and not $(h_1,equal,h_2)$: if $h_1 \cap \delta h_2 \neq \emptyset$ then $(h_1,touching\text{-}from\text{-}inside,h_2)$	
TFI	If (h_1,in,h_2), not $(h_1,equal,h_2)$ and not then $(h_1,touching\text{-}from\text{-}inside,h_2)$ then: $(h_1,completely\text{-}inside,h_2)$	
CIi	$(h_1, completely\text{-}inside_i, h_2) \leftrightarrow$ $(h_2, completely\text{-}inside, h_1)$	
TFIi	$(h_1, touching\text{-}from\text{-}inside_i, h_2) \leftrightarrow$ $(h_2, touching\text{-}from\text{-}inside, h_1)$	

Table 1. Topological relations of the calculus.

Definition 7. Given two time points, t and t', t' >> t iff t' > t and a relation has changed strictly more than once to a neighbor relation.

Definition 8. Given two time points, t and t', t' << t iff t' < t and a relation has changed strictly more than once to a neighbor relation.

According to these definitions, time is represented by disjunctive binary constraints of the form $X\{r_1, ..., r_n\}Y$, where each r_i is a relation that is applicable to X and Y. $X\{r_1, ..., r_n\}Y$ is a disjunction of the way $(Xr_1Y) \vee .... \vee (Xr_nY)$ and r_i is also called primitive constraints.

We have chosen this type of qualitative time constraints because we are only interested in the point of the time in which one region is transformed into its topological neighbourhood.

The topological neighborhood of a region is that region to which the original region can be transformed to by a process of gradual, continuous change which does not involve passage through any third region.

To reason about these temporal constraints we need to define the converse operation on a general relation.

Definition 9. A **general relation** R of the calculus is any subset of the set of all atomic relations.

Definition 10. The **converse** of a general relation R, called $R^{\cup}$ is defined as:

$$\forall(X,Y)\ ((X,R,Y) \Leftrightarrow (Y,R^{\cup},X)) \qquad (4)$$

Table 2 shows the converse operation for the time algebra.

2.1.3 The algebra on motion

The representational model of topology and qualitative time points follows the formalism used by Allen for temporal interval algebra [Allen 83]. The Allen style formalism will provide to our approach the possibility of reasoning with topology in dynamic worlds by applying the Allen's constraint propagation algorithm.

In the representational model for motion, binary relations between two objects, which can be points, lines or areas,

r	$r^{\cup}$
==	==
<<	>>
>>	>>
next	prev
prev	next

Table 2. The converse table for the time algebra.

called h_1 and h_2, in a point of time t are defined as tertiary constraints or propositions where the topological relation r between h_1 and h_2 in the point of time t is denoted by $(h_1,r,h_2)_t$. Therefore, we define a general relation on motion and the corresponding converse operation such as:

Definition 11. **A general relation R** of the algebra during time t as:

$$\forall(h_1,h_2)\ ((h_1,R,h_2)_t \Leftrightarrow \cup_{r\in R}\ (h_1,r,h_2)_t) \qquad (5)$$

Definition 12. The **converse** of a general relation R in time t, denoted as $R^{\cup}$, is defined as follows:

$$\forall(h_1,h_2)\ ((h_1,R,h_2)_t \Leftrightarrow (h_2,R^{\cup}, h_1)_t) \qquad (6)$$

From this definition we observe that the converse of the topology and time algebra is the same as the converse defined only for topological relations because the converse is calculated in the same point of time, therefore time does not affect to the converse operation.

2.2 The BSIP

In the BSIP section we are also going to distinguish among topology, time and motion. The operation which define the BSIP is the composition.

2.2.1 The BSIP on topology

The composition table for topological relations can be found in [Isli et al. 00].

r_1 r_2	<<	prev	==	next	>>
<<	{<<}	{<<}	{<<}	{prev,<<}	{<<,prev,==, next,>>}
prev	{<<}	{<<,prev}	{prev}	{==,prev, next}	{next,>>}
==	{<<}	{prev}	{==}	{next}	{>>}
next	{<<,prev}	{prev,==,next}	{next}	{>>,next}	{>>}
>>	{<<,prev,==, next,>>}	{>>,next}	{>>}	{>>}	{>>}

Table 3. The composition table for the time algebra.

2.2.2 The BSIP on time

Definition 13. The **composition** R1 ⊗ R2 of two general relations R1 and R2 is the most specific relation R such that:

$$\forall\ (h1,h2,h3)\ ((h1,R1,h2) \wedge (h2,R2,h3) \Rightarrow (h1,R,h3)) \qquad (7)$$

The composition table for time relations is shown in table 3.

2.2.3 The BSIP on motion

The BSIP for motion consists of: "given three objects A, B, C, if the topological relationships in time t between A and B and B and C are known, it is possible to obtain the topological relationship in time between objects A and C". In order to infer such topological relationship in time t we are going to define the composition operation for two general relations R1 and R2.

The composition for the model including topology and time has to be defined in three different ways in order to include all the possibilities.

Definition 14. The resulting general relation R obtained from the **composition** (⊗) operation could be calculated such as:

a) $(A,R1,B)t_0 \otimes (B,R2,C)t_0 \Rightarrow (A,R,C)t_0$
b) $(A,R1,B)t_0 \otimes (t_0, \text{Reltime}, t_1) \Rightarrow (A,R,B)t_1$
c) $(A,R1,B)t_0 \otimes (B,R2,C)t_1 / (t_0, \text{Reltime}, t_1) \Rightarrow ((A,R1,B)t_0 \otimes (t_0, \text{Reltime}, t_1)) \otimes (B,R2,C)t_1 \Rightarrow (A,R',B)t_1 \otimes (B,R2,C)t_1 \Rightarrow (A,R,C)t_1$

The first type of composition (14.a) is the composition of the topological relations between three regions A, B and C, in the same point of time, where A, B, C belong to {point, line, area}. Then it is the usual topological composition, where time does not affect. To calculate this composition we will use the 18 composition tables and the converse table. All these tables can be found in [Isli et al. 00].

The second type of composition (14.b) is the composition which implements Freksa's conceptual neighbourhood notion [Freksa 91]. It looks for the possible topological relations which will appear between two regions as time changes. To reason about this type we need to construct 6 composition tables that will be referred to as XYt–table where the regions X and Y belong to {point (P), line (L), area (A)} and t represents the time dimension of the algebra. We would need 9 composition tables if we consider all possibilities with X and Y being a point-like, a linear or an areal entity. However, we construct only 6 tables from which the other 3 tables can be obtained, by using the converse operation. We construct the AAt-table, LAt-table, PAt-table, LLt-table, PLt-table and the PPt-table, which are depicted in tables 4 to 9, respectively. Due to space limitation we have depicted in the same column both "next" and "prev" cases, and "<<" and ">>" cases because the corresponding entries are the same.

Notice that the "=" time relation is the identity.

As a relation t prev t' corresponds to a change of some topological relation to a neighbour relation, the tables always keep the possibility that a relation has not changed between time t and t', this situation model the fact that the time changes from t to t' because other topological relationship has changed and the relationship between X and Y (RelTop) has not changed.

The three tables which are not constructed can be obtained by applying the converse operation to the ones which are constructed. For example, the AL_t-table can be obtained using the LA_t-table and the converse operation.

This means that we have to find the most specific relation R such that, if X and Y are an areal and a linear entity respectively:

$$(X, \text{Reltop}, Y)t_0 \otimes (t_0, \text{Reltime}, t_1) \Rightarrow (X, R, Y)t_1 \qquad (8)$$

From the LA_t-table and using the converse operation we will get the relation R as follows:

Rtime / **RTop**	**next** or **prev**	<< or >>	==
T	{D,O,T}	{T,E,TFI,C,CIi,TFIi}	{T}
O	{T,TFI,O}	{O,D,E,CI,TFIi,CIi}	{O}
D	{T,D}	{D,O,E,TFI,CI,TFIi,CIi,TFIi}	{D}
E	{O,E,}	{E,T,D,TFI,CI, TFIi,CIi}	{E}
TFI	{O,CI,TFI}	{TFI,T,D,E,CI, TFIi, CIi}	{TFI}
CI	{TFI,CI}	{CI,T,O,D,E,TFIi,CIi}	{CI}
TFIi	{O,CIi,TFi}	{TFIi,T,D,E,CI,TFI}	{TFIi}
CIi	{TFIi,CIi}	{CIi,T,O,D,E,TFI,CI}	{CIi}

Table 4 AA_t-table.

Reltime / **Reltop**	**next** or **prev**	<< or >>	==
T	{C,D,T}	{T,TFI,CI}	{T}
C	{D,TFI,C}	{C,T,CI}	{C}
D	{T,D}	{D,C,TFI,CI}	{D}
TFI	{C,CI,TFI}	{TFI,T,D}	{TFI}
CI	{TFI,CI}	{CI,T,C,D}	{CI}

Table 5. LA_t-table.

Reltime / **Reltop**	**next** or **prev**	<< or >>	==
T	{D,CI,T}	{T}	{T}
D	{T,D}	{D,CI}	{D}
CI	{T,CI}	{CI,D}	{CI}

Table 6. PA_t-table.

Reltime / **Reltop**	**next** or **prev**	<< or >>	==
T	{D,CI,T}	{T}	{T}
D	{T,CI,D}	{D}	{D}
CI	{T,D,CI}	{CI}	{CI}

Table 7. PL_t-table.

Reltime Reltop	next or prev	<< or >>	==
E	{D,E}	{E}	{E}
D	{E,D}	{D}	{D}

Table 8. PP_t-table.

Reltime Reltop	Next or prev	<< or >>	==
T	{D,O,C,T}	{T,E,TFI,CI, TFi,CIi}	{T}
D	{T,C,D}	{D,O,E,TFI, CI,TFIi,CIi}	{D}
O	{T,C,O}	{O,D,E,TFI, CI,TFIi, CIi}	{O}
C	{T,D,C}	{C,O,E,TFI, CI,TFIi, CIi}	{C}
E	{T,O,E}	{E,D,C,TFI, CI,TFIi,CIi}	{E}
TFI	{C,CI,T,TFI}	{TFI,D,O,E, TFIi,CIi}	{TFI}
CI	{TFI,C,CI}	{CI,T,D,O,E, TFIi,CIi}	{CI}
TFIi	{T,C,CIi,TFi}	{TFIi,D,O,E, TFI,CI}	{TFIi}
CIi	{C,TFIi,CIi}	{CIi,T,D,O, E,TFI,CI}	{CIi}

Table 9. LL_t-table.

$$(Y, Reltop^{\cup}, X)t_0 \otimes (t_1, Reltime^{\cup}, t_0) \Rightarrow (Y, R', X)t_0 \quad (9)$$

Then the relation R that we are looking for is $R=(R')^{\cup}$.

For the third case of composition (*14.c*) we want to infer the composition R in time t_1 between three regions, X, Y and Z having the topological relation in time t_0 between X and Y, the topological relation in time t_1 between Y and Z and the qualitative time relation between times t_0 and t_1. In order to get the composition relation R, first of all we have to obtain the topological relations that can appear between X and Y in time t_1 using the composition tables defined for the case of *14.b* above described. Therefore we have the general relation R' which appears between X and Y in t_1. This relation together with the general relation R2 between Y and Z in t_1 correspond to the case *14.a*.

2.3 The FIP

In order to define a straightforward algorithm to solve the FIP, the concept of topology integrated with time is seen in our approach as an instance of the CSP.

Formula (1) to approximate the solution of the CSP is rewritten for topological relations between spatial objects in a point of time in three formulas for each definition of composition given in the BSIP, as follows:

Case 1: $c_{a,c,t} := c_{a,c,t} \oplus c_{a,b,t} \otimes c_{b,c,t}$ (10)
Case 2: $c_{a,b,t1} := c_{a,b,t1} \oplus c_{a,b,t0} \otimes c_{t0,t1}$ (11)
Case 3: $c_{a,b,t1} := c_{a,c,t0} \oplus c_{a,b,t0} \otimes c_{b,c,t1}$ (12)

Topology plus time relationships are represented as tertiary constraints by the predicate ctr_comp_top_time(TB,TA,A,B,Rel,t), where A and B are the spatial objects which holds the set of atomic topological relationships included in the set *Rel* in the point of time t, TB and TA represents the types of the objects A and B, which can be point (*p*), line (l) or area (a). Time relationship between points of time t_0 and t_1, (t_0, Rtime,t_1), is represented by the predicated ctr_comp_time(t0,t1,Rtime).

The intersection part ($c_{a,b}\oplus$....) of formula 10, 11, and 12 are implemented by a *simplification CHR* and the composition part ($c_{a,b} \otimes$) of formula 10, 11, and 12 are implemented by *propagation CHRs.*Part of the FIP is presented in algorithm 1.

For supplying the lack of completeness of the constraint graph (because there is not a topological relation between every object in the graph), three simplification CHRs are defined: the initial one (2a) and two more CHRs (2b and 2c) defined by applying the converse operation to the first and second constraints of the initial CHR (2a), respectively.

The same fact happens for propagation CHRs,. We need to define more CHRs by applying the converse operation to the first and second constraints, respectively. In the case of the composition (propagation CHRs) we need a different number of CHRs for each case to implement (A, B and C). Therefore for the CHRs implementing the case A of the composition we will need three CHRs, the initial one (3a) and two more CHRs defined using the converse operation (3b and 3c). For the case B, we need only two CHRs (3d and 3e), because the converse is applied only

```
% Constraint definitions
(1a)  constraints (ctr_comp_top_time)/6, (ctr_comp_top_time)/8.
(1b)  label_with ctr_comp_top_time(N,TB,TA,B,A,Rel,T,I) if N>1.
(1c)  ctr_comp_top_time(N,TB,TA,B,A,Rel,T,I):-
      member(R,Rel),ctr_comp_top_time(1,TB,TA,B,A,[R],T,I).

% Initializations
(1d)  ctr_comp_top_time(TB,TA,B,A,Rel,T) <=>
      length(Rel,N),ctr_comp_top_time(N,TB,TA,B,A,Rel,T,1).
(1e)  ctr_comp_time(T1,T2,Rel) <=> length(Rel,N),
      ctr_comp_time(N,T1,T2,Rel,1).

% Special cases
(1f)  ctr_comp_top_time(N,TB,TA,B,A,Rel,T,I) <=> empty(Rel) |
      false.
(1g)  ctr_comp_top_time(N,TA,TA,A,A,Rel,T,I) <=> true.
(1h)  ctr_comp_top_time(N,p,p,A,B,Rel,T,I) <=> N=2 | true.

% Intersection
(2a)  ctr_comp_top_time(N1,TB,TA,B,A,R1,T,I),
      ctr_comp_top_time(N2,TB,TA,B,A,R2,T,J)  <=>
      intersection(R1,R2,R3), length(R3,N3), K is min(I,J)+1 |
      ctr_comp_top_time(N3,TB,TA,B,A,R3,T,K).
(2b)  ctr_comp_top_time(N1,TB,TA,B,A,R1,T,I),
      ctr_comp_top_time(N2,TA,TB,A,B,R2,T,J) <=>
      conv_op(R2,R22), intersection(R1,R22,R3), length(R3,N3),
      K is min(I,J)+1 | ctr_comp_top(N3,TB,TA,B,A,R3,T,K).
(2c)  ctr_comp_top_time(N1,TA,TB,A,B,R1,T,I),
      ctr_comp_top_time(N2,TB,TA,B,A,R2,T,J) <=>
      conv_op(R1,R11), intersection(R1,R11,R3), length(R3,N3),
      K is min(I,J)+1 | ctr_comp_top(N3,TB,TA,B,A,R3,T,K).

% Composition
%CASE A:
(3a)  ctr_comp_top_time(N1,TB,TA,B,A,R1,T,I),
      ctr_comp_top_time(N2,TC,TB,C,B,R2,T,J) ==>
      ((I=1,J<6);(J=1;I<6)),
      composition_op(TA,TB,TC,R1,R2,R3), length(R3,N3), K is
      I+J | ctr_comp_top_time(N3,TC,TA,C,A,R3,T,K).
(3b)  ctr_comp_top_time(N1,TA,TB,A,B,R1,T,I),
      ctr_comp_top_time(N2,TC,TB,C,B,R2,T,J) ==>
      ((I=1,J<6);(J=1,I<6)), singleton(R1), conv_op(R1,R11),
      composition_op(TA,TB,TC,R11,R2,R3), length(R3,N3),
      K is I+J | ctr_comp_top_time(N3,TC,TA,C,A,R3,T,K).

%CASE B:
(3d)  ctr_comp_top_time(N1,TB,TA,B,A,R1,T1,I),
      ctr_comp_time(N2,T1,T2,R2,J) ==> ((I=1,J<6);(J=1;I<6)),
      composition_time_op(TA,TB,T1,T2,R1,R2,R3),
      length(R3,N3), K is I+J |
      ctr_comp_top_time(N3,TB,TA,B,A,R3,T2,K).

%CASE C:
(3f)  ctr_comp_top_time(N1,TB,TA,B,A,R1,T1,I),
      ctr_comp_top_time(N2,TC,TB,C,B,R2,T2,J),
      ctr_comp_time(N3,T1,T2,R3,M) ==>  ((I=1,J<6);(J=1;I<6)),
      composition_time_op(TA,TB,T1,T2,R1,R3,R4),length(R4,N
      4), K is I+M | ctr_comp_top_time(N4,TB,TA,B,A,R4,T2,K),
      ctr_comp_top_time(N2,TC,TB,C,B,R2,T2,J).
```

Algorithm 1. Path consistency algorithm to propagate compositions of disjunctive topology + time relationships.

to the second constraint. And finally for the case C we need 6 CHRs, defining the five not original by applying the converse operation to the three constraints in the head respectively (3f-k). Due to limitation restrictions only some of the propagation CHRs are depicted.

Therefore, the algorithm defines a total of 3 simplification rules and 11 propagation rules.

3 Qualitative velocity in 2-D

Velocity is the physical concept which relates space travelled by an object and time needed for this movement. The quantitative formula in physics is:

$$\text{Velocity} = \text{Space} / \text{Time}$$

Velocity is always relative (among other things because the earth is moving): we compare the position of an object "a" with respect to the position of another object "b" in two different times. Velocity is the distance travelled in this period of time. In the next sections we are going to introduce the algebra of a qualitative model for representing and reasoning with velocity.

3.1 The algebra

We define a velocity algebra, in which variables represent spatial objects. Between two objects it is possible to define a binary velocity relationship or constraint, which has the following reference system.

Definition 15. The **velocity reference system** VRS={UD, UT, LAB, INT}, where UD refers to the unit of distance or space travelled by the object; UT is the unit of time, both of which are context dependent; LAB refers to the set of qualitative velocity labels which represent the primitive relationships, the amount of which depend on the granularity level; and INT refers to the intervals associated to each velocity label of LAB, which will describe the velocity label in terms of the UD and the UT.

Assuming that in a determined context we fix UD to ud and UT to ut, we are going to define as examples two different VRS at coarse and fine levels of granularity, respectively. The UD and UT values might be changed depending on the context.

For the coarse VRS we are going to use four primitive constraints: zero, slow, normal, quick. They are defined as follows:

Definition 16. Given two spatial objects, the relative velocity between them is **zero** iff there is no space travelled per unit of time.

Definition 17. Given two spatial objects, the relative velocity between them is **slow** iff the space travelled by the object in the unit of time is between zero (not included) and half of the unit of distance (included).

Definition 18. Given two spatial objects, the relative velocity between them is **normal** iff the space travelled by the object in the unit of time is between half of the unit of distance (not included) and the unit of distance (included).

Definition 19. Given two spatial objects, the relative velocity between them is **quick** iff the space travelled by the object in the unit of time is between the unit of distance (not included) and infinite.

Definition 20. Therefore, the **coarse VRS** ={UD, UT, LAB_1, INT_1} where UD=ud, UT=ut, LAB_1={zero, slow, normal, quick}, and INT_1={[0,0],]0,ud/2ut],]ud/2ut,ud/ut],]ud/ut, ∝[}.

For the fine VRS we are going to use six primitive constraints: zero, very slow, slow, normal, quick, very quick. They are defined as follows:

Definition 21. Given two spatial objects, the relative velocity between them is **zero** iff the distance between them in two different times is the same, that is, there is no space travelled per unit of time.

Definition 22. Given two spatial objects, the relative velocity between them is **very slow** iff the space travelled by the object in the unit of time is between zero (not included) and a quarter of the unit of distance (included).

Definition 23. Given two spatial objects, the relative velocity between them is **slow** iff the space travelled by the object in the unit of time is between a quarter of the unit of distance (not included) and half of the unit of distance (included).

Definition 24. Given two spatial objects, the relative velocity between them is **normal** iff the space travelled by the object in the unit of time is between half of the unit of distance (not included) and the unit of distance (included).

Definition 25. Given two spatial objects, the relative velocity between them is **quick** iff the space travelled by the object in the unit of time is between the unit of distance (not included) and double the unit of distance.

Definition 26. Given two spatial objects, the relative velocity between them is **very quick** iff the space travelled by the object in the unit of time is between double the unit of distance (not included) and infinite.

Definition 27. Therefore, the **fine VRS** ={UD, UT, LAB_1, INT_1} where UD=ud, UT=ut, LAB_2={zero, very slow, slow, normal, quick, very quick}, and INT_2={[0,0],]0,ud/4ut],]ud/4ut,ud/2ut],]ud/2ut,ud/ut],]ud/ut,2ud/ut],]2ud/ut, ∝[}.

It is possible to define other VRSs, and it will be possible to reason using velocity information expressed in different VRS, by following the steps introduced in [Escrig & Toledo 01].

The information that we are going to represent is the relative velocity of an object b with respect to (wrt) the position of an object a, namely, b wrt a. Velocity information is a binary relationship. Therefore, it can also be expressed by: a wrt b, as result of applying the converse operation.

Definition 28. The converse of a general velocity relation R, called $R^{\cup}$ is defined as:

$$\forall(X,Y)\ ((X,R,Y) \Leftrightarrow (Y,R^{\cup},X)) \quad (13)$$

The converse is an identity operation, in the sense that if an object "b" is travelled at a particular speed from another object "a", which is considered as reference point, the speed at which object "a" is separating from object "b", which is now consider as reference point, is the same.

3.2 The BSIP

For the concept of velocity, the BSIP can be defined such as: given two velocity relationships: (1) object "b" wrt object "a", and (2) object "c" wrt object "b"; the BSIP consists of obtaining the velocity relationship of object "c" wrt object "a".

In the BSIP we can distinguish two different situations: 1) when the relative movement of the implied objects is in the same direction, and 2) when the relative movement of the implied objects is in any direction. They are both developed in the next sections.

3.2.1 Composition of velocity in the same direction

The BSIP for the concept of velocity, when the relative movement of the implied objects is in the same direction, is given in the composition table 10, for the coarse VRS_1, and in the composition table 11 for the fine VRS_2.

The first column of both tables refers to the velocity relationship "b wrt a", and the first raw of both columns refers to the velocity relationship "c wrt b". The rest of the cells of both tables refers to the composition, the velocity relationship "c wrt a", which are included into brackets because sometimes they contain disjunction of relations. The first label in both VRS (zero) are the neutral element, therefore the second column and the second raw of both tables correspond to the original relationship. The last labels of both VRS (quick and very quick, respectively) are the absorbent elements, that is, everything composed with these last relationships provide the last relationships of both VRS, respectively.

Both tables are symmetrical with respect to the main diagonal, therefore it is necessary to represent only the upper or the downer part of both tables.

It is also important to notice that the results in both inference tables maintain conceptual neighbourhood.

3.2.2 Composition of velocity in any direction

The BSIP for the velocity concept when the relative movement of the implied objects is in any direction, is solved by integrating the concept of velocity with the qualitative orientation model of Freksa & Zimmermann [Freksa 92, Freksa & Zimmermann 92]. The Freksa & Zimmeramnn's coarse orientation reference system contains nine orientation regions, as it is shown in figure 1. The reference system per se is represented by the perpendicular lines, whereas the black dots are only included if the corresponding orientation region is referred. For representing the velocity of b wrt a, integrated with this orientation reference system, we consider that the reference object is always on the cross point of both perpendicular lines which defines the coarse reference system. Therefore, a finer reference system is not necessary, because we are not going to use so many distinctions.

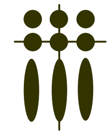


Figure 1. The Freksa & Zimmerman's coarse orientation reference system.

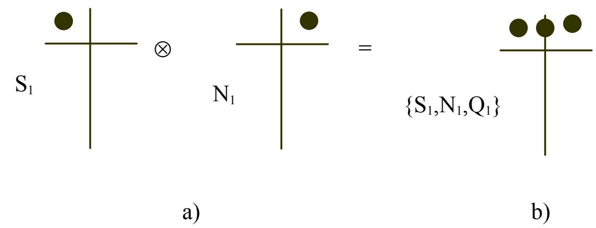


Figure 2. An example of composition: a) "slow" velocity towards the "left front" orientation with "normal" velocity towards the "right front" orientation, gives as result b) "slow", "normal" or "quick" velocities towards "left front", "straight front" or "right front" orientations.

	Z_1	S_1	N_1	Q_1
Z_1	$\{Z_1\}$	$\{S_1\}$	$\{N_1\}$	$\{Q_1\}$
S_1	$\{S_1\}$	$\{S_1,N_1\}$	$\{N_1,Q_1\}$	$\{Q_1\}$
N_1	$\{N_1\}$	$\{N_1,Q_1\}$	$\{N_1,Q_1\}$	$\{Q_1\}$
Q_1	$\{Q_1\}$	$\{Q_1\}$	$\{Q_1\}$	$\{Q_1\}$

Table 10. The composition table which solves the BSIP for the coarse VRS_1 with LAB_1={zero (Z_1), slow (S_1), normal (N_1), quick (Q_1)}.

The composition table which solves the BSIP for the integration of the coarse VRS_1 (4 velocity distinctions) with the coarse orientation reference system (9 qualitative regions), has 36×36 entries. One of these entries is shown in figure 2.

The composition table which solves the BSIP for the integration of the fine VRS_1 (6 velocity distinctions) with the coarse orientation reference system (9 qualitative regions), has 54×54 entries.

For the management of velocity, we can consider the orientation regions as intervals of orientations as well.

	Z_2	VS_2	S_2	N_2	Q_2	VQ_2
Z_2	$\{Z_2\}$	$\{VS_2\}$	$\{S_2\}$	$\{N_2\}$	$\{Q_2\}$	$\{VQ_2\}$
VS_2	$\{VS_2\}$	$\{VS_2,S_2\}$	$\{S_2,N_2\}$	$\{N_2,Q_2\}$	$\{Q_2,VQ_2\}$	$\{VQ_2\}$
S_2	$\{S_2\}$	$\{S_2,N_2\}$	$\{N_2\}$	$\{N_2,Q_2\}$	$\{Q_2,VQ_2\}$	$\{VQ_2\}$
N_2	$\{N_2\}$	$\{N_2,Q_2\}$	$\{N_2,Q_2\}$	$\{Q_2\}$	$\{Q_2,VQ_2\}$	$\{VQ_2\}$
Q_2	$\{Q_2\}$	$\{Q_2,VQ_2\}$	$\{Q_2,VQ_2\}$	$\{Q_2,VQ_2\}$	$\{VQ_2\}$	$\{VQ_2\}$
VQ_2	$\{VQ_2\}$	$\{VQ_2\}$	$\{VQ_2\}$	$\{VQ_2\}$	$\{VQ_2\}$	$\{VQ_2\}$

Table 11. The composition table which solves the BSIP for the fine VRS_2 with LAB_2={zero (Z_2),very slow, (V_2S), slow (S_2), normal (N_2), quick (Q_2), very quick (VQ_2)}.

Definition 29. Freksa & Zimmermann's orientation reference system (ORS) is redefined to ORS={LABo, INTo}, where LABo refers to the set of qualitative orientation labels and INTo refers to the intervals associated to each orientation label of LABo, which will describe the orientation label in terms of ranges of angles. For instance, for the coarse orientation reference system, which we consider:

LABo={front-left (fl), straight front (sf), front-right (fr), left (l), none (n), right (r), back-left (bl), straight-back (sb), back-right (br)}; and

INTo={]90,180[,[90,90],]0,90[,[180,180],_,[0,0],]180,270[, [270,270],]270,360[}.

In order to obtain the composition of velocity at any orientation, we have implemented an algorithm which is based on the idea of reasoning with the extreme points/angles which define the INT part of the velocity/orientation reference systems, respectively. In general, we want to compose a velocity with label V1, at an orientation with label O1, with a velocity with label V2 at an orientation with label O2. Each velocity label V is defined by the interval: open or closed initial velocity to open or closed final velocity (*o/c, vi, o/c, vf*). Each orientation label O is defined by the interval: open or closed initial orientation to open or closed final orientation, (*o/c, oi, o/c, of*).

There exists sixteen possible combinations of extreme velocity/orientation for computing each one of the entries of the composition table (see table 12 for the example corresponding to figure 2). They are calculated in algorithm 2.

In general, for the composition of (OorCv1, IorFv1, V1, OorCo1, IorFo1, O1) ⊗ (OorCv2, IorFv2, V2, OorCo2, IorFo2, O2), where OorC corresponds to the fact that the xy interval is open or closed, and IorF corresponds to the fact that the xy interval is initial or final, algorithm 1 will obtain both result components (velocity and orientation):

For each on of the sixteen orientation/velocity extreme cases we repeat the following:

1) If one of both velocities is equal to zero, then the module of the velocity and the orientation corresponds to the one which is different from zero.

2) If the module of the subtraction of both orientations is zero, then the resulting module of velocity will be the sum of velocities. The orientation will correspond to the same orientation.

3) If the module of the subtraction of both orientations is 90 degrees, then the Pythagoras theorem is applied in order to obtain the velocity module, that is:
   $V= \mathrm{sqrt}(V1^2 + V2^2 - 2 \times V1 \times V2 \times \cos \alpha)$
   where $\alpha=|O1-O2|/2$.
   The orientation will be given by the following sequence:
   maxmin(O1,O2,Omax,Omin) {Omax and Omin contain the maximun and minimun of O1 and O2, respectively}
   O=Omin+(Omax-Omin)/2

4) If the module of the subtraction of both orientations is 180 degrees, then the resulting module of velocity will be the subtraction of velocities. If the velocity is zero then the orientation is not important. Otherwise, the orientation will correspond to the higher orientation.

V1	O1	V2	O2			qual. orient	qual. vel.
i	i	i	i	0 ud/2ut	ud/2ut	front-right	normal
i	f	i	i	0 ud/2ut	ud/2ut	front-right	normal
f	i	i	i	ud/2ut ud/2ut	√2 ud/2ut	front-right	normal
f	f	i	i	ud/2ut ud/2ut	V=0 O=?	?	zero
i	i	i	f	+ud/2ut 0	ud/2ut	front-right	normal
i	f	i	f	ud/2ut 0	ud/2ut	front-right	normal
f	i	i	f	+ud/2ut +ud/2ut	ud/ut	straight-front	quick
f	f	i	f	ud/2ut ud/2ut	√2 ud/2ut	front-left	normal
i	i	f	i	0 ud/ut	ud/ut	front-right	normal
i	f	f	i	0 ud/ut	ud/ut	front-right	normal
f	i	f	i	ud/2ut ud/ut	√5/4 ud/2ut	front-right	quick
f	f	f	i	ud/2ut ud/ut	ud/2ut	front-right	{slow, normal}
i	i	f	f	ud/ut 0	ud/ut	front-right	normal
i	f	f	f	ud/ut 0	ud/ut	front-right	normal

Table 12.a. Composition example of figure 2, where O1=front_left, i.e.]90,180[, V1=slow1, i.e.]0,ud/2ut], O2=front_right, i.e.]0,90[, and V2=normal1, i.e.]ud/2ut,ud/ut] (first part)

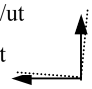
f	**i**	**f**	**f**	 ud/ut ud/2ut	 ud/2ut	front-right	slow
f	**f**	**f**	**f**	ud/ut ud/2ut 	 √5/4 ud/2ut	front-left	quick
UNION						**{front-right**, **straight-front, front-left}**	**{zero, slow, normal, quick}**

Table 12.b. Composition example of figure 2, where O1=front_left, i.e.]90,180[, V1=slow1, i.e.]0,ud/2ut], O2=front_right, i.e.]0,90[, and V2=normal1, i.e.]ud/2ut,ud/ut] (second part)

```
Entries:   (OorCv1, IorFv1, V1, OorCo1, IorFo1, O1) ⊗
           (OorCv2, IorFv2, V2, OorCo2, IorFo2, O2)

REPEAT for each one of the sixteen orientation/velocity extreme
cases.
   CASE
      V1=0:          O=O2; V=V2;
      V2=0:          O=O1; V=V1;
      |O1-O2|=0:     O=O1; V=V1+V2;
      |O1-O2|=90:    V= sqrt(V1^2 + V2^2 - 2 ×V1 ×V2 ×cos α)
                     α=|O1-O2|/2
                     maxmin(O1,O2,Omax,Omin)
                     O=Omin+(Omax-Omin)/2
      |O1-O2|=180:   V=|V1-V2|;
                   if V=0 then O=?
                   else O=max(O1,O2)
   ENDCASE;

   if O=O1 then
      if IorFo1=I then O=O+1
      else O=O-1
   else if O=O2 then
         if if IorFo2=I then O=O+1
         else O=O-1;

   Vlabel= LAB[pos(interval(V,INT))]
   Olabel= LAB[pos(interval(O,INT))]
   Vresult= Vresult ∪ Vlabel
   Oresult= Oresult ∪ Olabel

ENDREPEAT

Output: Vresult, Oresult
```

Algorithm 2. BSIP for each entry of the composition table for velocity in any direction.

In any case, if the resulting orientation corresponds to the beginning of the interval, one degree is added to the result (because the interval is open and the exact orientation is not Included in the result); if the resulting orientation corresponds to the end of the interval, one degree is subtracted to the result.

The result of each of the previous compositions have two components: 1) the module of the velocity, which will be a quantitative amount in terms of the unit of distance and the unit of time; and 2) an angle which corresponds to a quantitative orientation. The two quantitative components (velocity and orientation) which deliver the previous algorithm are translated into qualitative terms, by using the corresponding reference systems and the following formula:

$$\text{label}=\text{LAB}[\text{pos}(\text{interval}(\text{VorO},\text{INT}))] \quad (14)$$

That is, the label associated with a quantity (label) is obtained by looking at the label set (LAB) for the position of the interval (INT) where the quantity of velocity or orientation (VorO) belongs.

The union of the resulting velocity labels will define the resultant velocity, and the union of the resulting orientation labels will define the resultant orientation. Both results for velocity and orientation might be a disjunction of relations.

A graphical example of the algorithm, corresponding to figure 2, is shown in table 12.

For obtaining the 36×36 BSIP table, we should apply the above algorithm for each one of the entries to the table.

```
% Constraint definitions
(1a)  constraints ctr_vel/4, ctr_vel/7.
(1b)  label_with ctr_vel(Nv,No,X,Y,V,O,I) if Nv>=1 and No>=1.
(1c)  ctr_vel(Nv,No,X,Y,V,O,I):- member(V1,V), member(O1,O),
      ctr_vel(1,1,X,Y,[V1],[O1],I).

% Initializations
(1d)  ctr_vel(X,Y,V,O) <=> length(L,Nv), length(O,No) |
      ctr_vel(Nv,No,X,Y,V,O,1).

% Special cases
(1e)  ctr_vel(Nv,No,X,Y,V,O,I) <=> Nv=9 | true.
(1f)  ctr_vel(Nv,No,X,Y,V,O,I) <=> No=9 | true.
(1g)  ctr_f_o(Nv,No,X,X,V,O,I) <=> true.

% Intersection
(2a)  ctr_vel(Nv1,No1,X,Y,V1,O1,I),
      ctr_vel(Nv2,No2,X,Y,V2,O2,J) <=>
      inter(V1,V2,V3),length(V3,Nv3),inter(O1,O2,O3),
      length(O3,No3), min(I,J,K) | ctr_vel(Nv3,No2,X,Y,V3,O3,K).
(2b)  ctr_vel(Nv1,No1,Y,X,V1,O1,I),
      ctr_vel(Nv2,No2,X,Y,V2,O2,J) <=>
      inv(V1,Vi1),inv(O1,Oi1),inter(Vi1,V2,V3),length(V3,Nv3),
      inter(Oi1,O2,O3), length(O3,No3), min(I,J,K) |
      ctr_vel(Nv3,No2,X,Y,V3,O3,K).
(2c)  ctr_vel(Nv1,No1,X,Y,V1,O1,II),
      ctr_vel(Nv2,No2,Y,X,V2,O2,J)
      <=> inv(V2,Vi2), inv(O2,Oi2) ,inter(V1,Vi2,V3),
      length(V3,Nv3), inter(O1,Oi2,O3), length(O3,No3),
      min(I,J,K) | ctr_vel(Nv3,No2,X,Y,V3,O3,K).

% Composition
(3a)  ctr_vel(Nv1,No1,X,Y,V1,O1,I),
      ctr_vel(Nv2,No2,Y,Z,V2,O2,J) ==> J=1,
      composition_vel(V1,O1,V2,O2,V3,O3),length(V3,Nv3),
      length(O3,No3), K is I+J | ctr_vel(Nv3,No3,X,Z,V3,O3,K).
(3b)  ctr_vel(Nv1,No1,Y,X,V1,O1,I),
      ctr_vel(Nv2,No2,Y,Z,V2,O2,J) ==> J=1, inv(V1,Vi1),
      inv(O1,Oi1), composition_vel(Vi1,Oi1,V2,O2,V3,O3),
      length(V3,Nv3), length(O3,No3), K is I+J |
      ctr_vel(Nv3,No3,X,Z,V3,O3,K).
(3c)  ctr_vel(Nv1,No1,X,Y,V1,O1,I),
      ctr_vel(Nv2,No2,Z,Y,V2,O2,J) ==> J=1, inv(V2,Vi2),
      inv(O2,Oi2), composition_vel(V1,O1,V2,O2,V3,O3),
      length(V3,Nv3), length(O3,No3), K is I+J |
      ctr_vel(Nv3,No3,X,Z,V3,O3,K).
```

Algorithm 3. Path consistency algorithm to propagate compositions of disjunctive qualitative velocity relationships.

3.3 The FIP

In order to define a straightforward algorithm to solve the FIP, the concept of qualitative velocity is seen in our approach as an instance of the CSP. The velocity relationships are represented as binary constraints such as ctr_vel(X,Y,V,O), which refers to the velocity relationship V at the orientation O which there exists between objects X and Y.

Formula (1) to approximate the solution of the CSP is also valid for binary velocity constraints.

Algorithm 3 implements the full inference process for qualitative velocity. Predicates ctr_vel/4 and ctr_vel/7 are declared in (1a). Predicate ctr_vel/4 corresponds to the user-defined constraints for qualitative velocity. For example ctr_vel(X,Y,V,O) means that the spatial object Y moves at any velocity, included in the list V, in any orientation, included in the list O, wrt the spatial object X. They are translated into ctr_vel/7 by the simplification CHR (1d), where the length of velocity and orientation lists are added, as well as the length of the shortest path from which the constraint has been derived (length 1 means that the constraint is direct, i.e. it is user-defined). These three arguments are included to increase efficiency. The two first avoid composition between constraints which does not provide more information (simplification CHR (1e) removes the constraints which contain all the velocity labels and CHR (1f) removes the constraints which contain all the orientation labels). The last argument is used to restrict the propagation CHRs to involve at least one direct constraint. In (1c), *member(V1,V)* nondeterministically chooses one primitive qualitative velocity, V1, from the disjunctive relationship, V, and one primitive orientation relationship, O1, from the disjunctive relationship, O, which implements the backtrack search part of the algorithm.

Simplification CHRs (2a), (2b) and (2c) perform the intersection part of the algorithm. When there exist two constraints which relate the same two spatial objects, the intersection of the corresponding relations (V1 and V2) is computed (plain set intersection) and the previous constraints are substituted by the new one. For the case in which the constraint graphs is not complete (i.e. there is not two arcs between each pair or nodes), two simplifications CHRs more have been defined: (2b) and (2c). In CHR (2b), the inverse operation is applied to the first constraint, in order to relate the same spatial objects in the same order. In CHR (2c), the inverse operation is applied to the second constraints, in order to relate the same spatial objects in the same order.

Propagation CHRs (3a), (3b) and (3c) compute composition. (3a) computes composition as it was originally defined by formula (1). For the cases in which the constraint graph is not complete, two propagation CHRs more (3b) and (3c) are included, with the same meaning as for computing the intersection part of formula (1).

4 Qualitative velocity in 3-D

The qualitative model explained in this section will allow us not only to represent and reason with the relative velocity between two or more objects in the 2-D plane, but in the 3D space.

4.1 The algebra

The algebra for velocity in 3D is the same as the algebra for velocity in 2D.

4.2 The BsiP

The BSIP for velocity in 3D is defined such as the BSIP on velocity in 2D. As well as for velocity in 2D, we can distinguish two different situations: 1) when the relative movement of the implied objects is in the same direction, and 2) when the relative movement of the implied objects is in any direction. The composition of velocity in the same direction is solved in 3D in the same way that it is solved in 2D. For solving composition of 3D velocity in any direction, we have integrated the concept of velocity with the 3D qualitative orientation model of [Pacheco et al. 02]. The 3D qualitative orientation model of Pacheco et al. has four different models depending the precision the space is studied. In this paper we will use the general coarse model which divides the space in 27 qualitative regions as is shown in figure 3. The reference system is represented by the three perpendicular planes passing by the point "a". For representing the velocity of b wrt a, integrated with this orientation reference system, we consider that the reference object is always on the intersection point of the three perpendicular planes which defines the coarse reference system. Therefore, a finer reference system is not necessary.

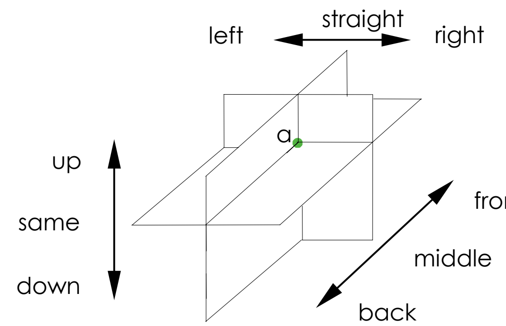


Figure 3. The general coarse qualitative orientation model of Pacheco et al.

Definition 30. The Pacheco et al.'s general coarse 3D orientation model reference system is redefined to 3DRS={3DLABo, 3DINTo}, where 3DLABo refers to the set of qualitative orientation labels and 3DINTo refers to the intervals associated to each orientation label of 3DLABo.

3DLABo={front-left-up (flu), front-straight-up (fsu), front-right-up (fru), middle-left-up (mlu), none-up (nu), middle-right-up (mru), back-left-up (blu), back-straight-up (bsu), back-right-up (bru), front-left-same (fls), front-straight-same (fss), front-right-same (frs), middle-left-same (mls), none-same (ns), middle-right-same (mrs), back-left-same (bls), back-straight-same (bss), back-right-same (brs), front-left-down (fld), front-straight-down (fsd), front-right-down (frd), middle-left-down (mld), none-down (nd), middle-right-down (mrd), back-left-down (bld), back-straight-down (bsd), back-right-down (brd)};

and

INTo={(]90,180[,u), ([90,90],u), (]0,90[,u), ([180,180],u), (_,u), ([0,0],u), (]180,270[,u), ([270,270],u), (]270,360[,u), (]90,180[,s), ([90,90],s), (]0,90[,s), ([180,180],s), (_,s), ([0,0],s), (]180,270[,s), ([270,270],s), (]270,360[,s), (]90,180[,d), ([90,90],d), (]0,90[,d), ([180,180],d), (_,d), ([0,0],d), (]180,270[,d), ([270,270],d), (]270,360[,d)}.

The redefinition of Pacheco et al. 3D orientation model such as an interval algebra has a new argument, with respect to the 2D orientation model, the height. Therefore, the algorithm which computes each one of the 108 ×108 entries is the same as for velocity in 2D but adding the following cases (inside CASE).

1.- If one of both height labels is "same" the resulting label corresponds to the label whose height is not "same".
Example: (X,s) ⊗ (Y,H) = (Z,H) and (X,H) ⊗ (Y,s) = (Z,H) where (X ⊗ Y = Z in the 2D algorithm and H belongs to {u,s,d})

2.- If both of height labels are identical, namely LAB, the resulting label is LAB.
Example: (X,H) ⊗ (Y,H) = (Z,H) where (X ⊗ Y = Z in the 2D algorithm and H belongs to {u,s,d})

3.- In other case, we obtain complete uncertainty.
Example: (X,u) ⊗ (Y,d) = (Z,[u,s,d]) and (X,d) ⊗ (Y,u) = (Z,[u,s,d]) where (X ⊗ Y = Z in the 2D algorithm)

4.3 The FIP

The FIP for velocity in 3D is solved by the same algorithm that it is solved the FIP for velocity in 2D.

Conclusions and future work

Three models which integrates space and time is defined in this article: a qualitative motion model which integrates topology and time; a qualitative velocity model in 2D; and a qualitative velocity model in 3D. The integration is accomplished thanks to a common definition of each aspect to be integrated based on three steps: (1) the representation of each aspect to be integrated, and (2) the reasoning process which consists on (2.1) the definition of the basic step of the inference process, and (2.2) an algorithm which solves the full inference process.

This approach not only have allow us the integration of many spatial aspects [Escrig & Toledo 98], but it has also permitted the integration of time and space in a straightforward way.

We are nowadays working on the definition of a qualitative acceleration model, which is basically defined in a similar way to the 2D qualitative velocity model.

Our interest is the application of these qualitative models (together with quantitative information) to solve the autonomous and intelligent real robot navigation problem, from which we have obtained only preliminary results.

Acknowledgements

The preliminary ideas described in this paper where discussed with students of a doctoral course given by one of the authors at Santiago de Cuba. We want to specially thank their contribution. The BSIP for velocity in 2-D has been implemented by Ramón Sabater as final project of a doctoral course. Thanks to him for his contribution. This work has been partially supported by CICYT under grant number TAP99-0590-002-02.

References

[Allen 83] Allen, J. Maintaining Knowledge about Temporal Intervals. Communications of the ACM , Vol. 26, No. 11. pp. 832-843, 1983.

[Belussi et al. 98] A. Belussi, E.Bertino and B.Catania. "An extended algebra for constraint databases". *IEEE TKDE,* 10(5):686-705, 1998.

[Böhlen 99] M.H. Böhlen, C.S. Jensen and M.O. Scholl, editors. *Spatio-Temporal Database Management*, volume 1678 of *LNCS*, Springer 1999.

[Brisset et al. 94] P. Brisset et al., ECLiPSe 3.4 Extensions User Manual, European Computer-Industry Research Centre, Munich, Germany, 1994.

[Clementini et al. 95] Clementini, E., Di Felice, P., Hernández, D., "Qualitative Representation of Positional Information", TR FKI-208-95, Technische Universität München, 1995.

[Escrig & Toledo 00] Escrig, M.T., Toledo, F., "Autonomous Robot Navigation using human spatial concepts". International Journal of Intelligent Systems, Vol.15, No.3, March 2000, pp.165-196, 2000. ISBN 0884-8173.

[Escrig & Toledo 01] Escrig, M.T., Toledo, F., "Reasoning with compared distances at different levels of granularity", in Proceedings of the 9th Conference of Spanish Association of Artificial Intelligence, October 2001.

[Escrig & Toledo 98] 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.

[Frank 92] Frank, A., "Qualitative Spatial Reasoning about Distance and Directions in Geographic Space". In Journal of Visual Languages and Computing 3, pag. 343-373, 1992.

[Freksa 91] Freksa, C. Conceptual neighborhood and its role in temporal and spatial reasoning. Technical Report Tech. RPt. FK1-146-91, Institute of Information, University of Munich, 1991.

[Freksa 92] 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, pag. 162-178, 1992. Springer, Berlin.

[Freksa & Zimmermann 92] Freksa, C., Zimmermann, K., "On the Utilization of Spatial Structures for Cognitively Plausible and Efficient Reasoning", Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pag. 18-21, 1992.

[Fruehwirth 94] Fruehwirth, T., "Constraint Handling Rules", in A. Podelski (ed.), Constraint Programming: Basic and Trends, volume 910 of Lecture Notes in Computer Science, pag. 90-107, 1994. Springer.

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

[Grumbach et al. 98] S. Grumbach, P. Rigaux and L. Segoufin. "Spatio-Temporal Data Handling with Constraints". In *ACM GIS:*106-111. ACM Press, 1998.

[Hernández 94] Hernández, D. Qualitative Representation of Spatial Knowledge. In volume 804 in Lecture Notes in AI. Ed. Springer-Verlag, 1994.

[Isli, Museros et al. 00] Isli, A., Museros, Ll., Barkowsky, T. and Reinhard, M. "A Topological Calculus for Cartographic Entities". In Lecture notes in Artificial Intelligence 1849. Spatial Cognition II. Integrating Abstract Theories, Empirical Studies, Formal Methods and Practical Applications. Spatial Cognition. Springer-Verlag, pp. 225 – 239, 2000.

[Jong 94] Jong, J. H., "Qualitative Reasoning About Distances and Directions in Geographic Space". Ph.D. thesis. Department of Surveying Engineering. University of Maine, 1994.

[Jungert 92] Jungert, E., "The Observer's Point of View: An Extension of Symbolic Projections", in International Conference GIS -From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. Volume 639 of Lectures Notes in Computer Science. Ed. Springer-Verlag, 1992.

[Mackworth & Freuder 85] Mackworth, A.K., Freuder, E.C., "The complexity of some Polynomial networks consistency algorithms for constraint satisfaction problems", Journal of Artificial Intelligence 25: pag. 65-74, 1985.

[Mukerjee & Joe 90] Mukerjee, A., Joe, G., "A Qualitative Model for Space". In 8th-AAAI, pag. 721-727, 1990.

[Mukerjee & Schnorrenberg 91] A. Mukerjee and F. T. Schnorrenberg. "Hybrid Systems: Reasoning across Scales in Space and Time". *AAAI Symposium on Principles of Hybrid Reasoning*, November 15-17, 1991.

[Musto et al. 99] Musto, A. et al., "Qualitative and Quantitative representations of Locomotion and their application in robot navigation", in Proceedings of the IJCAI-99, Morgan Kaufman Publishers, 1999.

[Musto et al. 00] Musto, A. et al., "From motion observation to qualitative representation", in C. Freksa, C. Habel, C. Hendel (eds.), Spatial Cognition II. LNCS-Series, Springer Berlin, Heidelberg, New York, 2000.

[Pacheco et al. 01] Pacheco, J., Escrig, M.T., Toledo, F., "A model for representing and reasoning with 3-D Qualitative Orientation", in Proceedings of the 9th Conference of Spanish Association of Artificial Intelligence, October 2001.

[Raffaetà & Fruehwirth 01] A. Raffaetà, T. Fruehwirth. "Spatio-Temporal Annotated Constraint Logic Programming". Third international Workshop on Practical Aspects of Declarative Languages (*PADL'01)*, Las Vegas, USA, March 2001.

[Zimmermann 93] Zimmermann, K., "Enhancing Qualitative Spatial Reasoning Combining Orientation and Distance", in Frank, A. U. and Campari, I. (eds.). Spatial Information Theory. A Theoretical Basis for GIS. European Conference, COSIT'93, pag. 69-76, Marciana Marina, Italy, Volume 716 of Lecture Notes in Computer Science. Springer, Berlin.

[Zimmermann & Freksa 93] Zimmermann, K., Freksa, C., "Enhancing Spatial Reasoning by the concept of motion", in the Proceedings of the AISB'93 Conference, 1993.