# Maintaining Qualitative Spatial Knowledge*†

Daniel Hernández
Fakultät für Informatik
Technische Universität München
80290 Munich, Germany

e-mail: danher@informatik.tu-muenchen.de

**Abstract**

We present mechanisms used to maintain the consistency of a knowledge base of spatial information based on a qualitative representation of 2-D positions. These include the propagation heuristics used when inserting new relations as well as the reason maintenance mechanisms necessary to undo the effects of propagation when deleting a relation. Both take advantage of the rich structure of the spatial domain.

## 1 Introduction

For a representation to be of any use, we have to consider not only its constituents and how they correspond to what is being represented, but also the mechanisms operating on them. In this paper we look into the mechanisms that allow us to reason with qualitative representations of 2-D positions. These mechanisms are determined in part by the tasks for which qualitative reasoning is used, such as: Inferring knowledge implicit in the knowledge base; answering queries given partial knowledge and a specific context; maintaining various types of consistency; acquiring new knowledge; and, particularly in the case of spatial knowledge, building cognitive maps and visualizing qualitatively represented spatial situations.

Even though the qualitative approach has been extensively used for modeling physical phenomena (Bobrow 1984; Weld and de Kleer 1990), it is only recently that research on qualitative models of space has been undertaken. Allen (1983) introduced an interval-based temporal logic, in which knowledge about time is maintained qualitatively by storing comparative relations between intervals. Freksa (1992a) presents a generalization of Allen's temporal reasoning approach based on semi-intervals and

*The project on which the work reported here is based has been funded by the German Ministry for Research and Technology (BMFT) under FKZ ITN9102B. The author is solely responsible for the contents of this publication.

†To appear in: COSIT 93, Proc. of the European Conference on Spatial Information Theory, Elba, Italy, 19–22 Sept. 1993.

introduces the notion of "conceptual neighborhood" of qualitative relations. There have been some prior efforts to extend Allen's temporal approach to spatial dimensions (Guesgen 1989; Mukerjee and Joe 1990). However, these extensions just use Cartesian tuples of the one-dimensional relations, loosing the "cognitive plausibility" that Allen's approach has in the temporal domain. Our representation of positions in 2-D space establishes different qualitative relations for the two relevant dimensions topology and orientation. Related research on topological relations in the context of Geographic Information Systems has been done by Egenhofer (1989, 1991), Egenhofer and Al-Taha (1992), Egenhofer and Sharma (1993), and Smith and Park (1992). Kuipers and Levitt (1988) describe a series of influential systems for navigation and mapping in large-scale space. Of particular interest here is the QUALNAV model (see also Levitt and Lawton 1990), which includes a coordinate-free, topological representation of relative spatial location, and integrates metric knowledge of relative or absolute angles and distances. The symbolic projection schema introduced by Chang, Shi, and Yan (1987) in the context of pictorial databases represents two dimensional spatial arrangements by projecting them into two "2D-strings" along the vertical and horizontal axes. Various extensions of this model have been proposed including further operators and a combination with quad-trees (Chang and Li 1988), local operators to handle overlapping objects (Jungert 1988). Frank (1991) presents a qualitative algebra for reasoning about cardinal directions, which are easier to analyze then relative orientations because the frame of reference is fixed in space. Cohn, Cui, and Randell (1992) summarize a theory of space and time based on a calculus of individuals founded on "connection" and expressed in the many sorted logic LLAMA. A basic set of dyadic topological relations is defined using the primitive C(x,y) = 'x connects with y', x and y being regions. Freksa (1992b) and Freksa and Zimmermann (1992) present an approach to qualitative spatial reasoning based on directional orientation information. They distinguish 15 possible positions and orientations of a point based on the left/straight/right distinction w.r.t. a vector $ab$ as well as the front/neutral/back distinction w.r.t. the lines orthogonal to $ab$ on the end points of $a$ and $b$. For a general overview of recent literature in the area of spatial reasoning see, for example, McDermott (1992) and Topaloglou (1991).

In previous work we have explored various aspects of the qualitative representation of space (Hernández 1991) including mechanisms used to transform between different frames of reference (these transformations are necessary to obtain canonical reference frames, which are a pre-requisite for qualitative inference); methods for the efficient computation of composition tables for positional relations based on the structure of the relational domains, and "abstract maps", which allow the solution of some tasks by diagrammatical means. In (Hernández and Zimmermann 1992) we discuss a method for constraint relaxation that uses the structure of the relational domain to weaken constraints by including other neighboring relations in their disjunctive definitions, instead of retracting them as a whole. This approach leads faster to solutions of meaningfully modified sets of otherwise unsatisfiable constraints.

In the following section we first briefly introduce the representation model and concentrate in the later sections on the algorithms required to maintain the consistency of a qualitative knowledge base of spatial information. These include the propagation heuristics used when inserting new relations as well as the reason maintenance mechanisms required to undo the effects of propagation when deleting a relation.

# 2 Qualitative Representation of Positions in 2-D

We focus on 2-D projections of 3-D scenes. Two factors determine the qualitative position of objects in 2-D space: the relative orientation of objects to each other and the extension of the involved objects. Considering these factors independently from each other results in two classes of spatial relations:

- topological relations (ignore orientation)
- orientation relations (ignore extension, i.e., objects = points)

Our goal is to combine these two classes of relations to provide a model of orientation that accounts for extended objects. For this purpose we define a small set of spatial relations from the two relevant dimensions topology and orientation.

Topological[1] relations describe how the boundaries of the two objects relate. A complete set of topological relations can be derived from the combinatorial variations of the point set intersection of boundaries and interiors of the involved objects by imposing the constraints of physical space on them (Egenhofer and Franzosa 1991). The resulting set of eight mutually exclusive relations is: `disjoint` (`d`), `tangent` (`t`), `overlaps` (`o`), `contains-at-border` (`c@b`), `included-at-border` (`i@b`), `contains` (`c`), `included` (`i`), `equal` (`=`).

Orientation relations describe where the objects are placed relative to one another. The orientation dimension results from the transfer of distinguished reference axes from an observer to the reference object. There are various levels of hierarchically organized orientation relations of different granularities. The level with the eight distinctions most commonly used contains the following relations (abbreviations in parentheses) $\texttt{front}_3(\texttt{f}_3)$, $\texttt{back}_3(\texttt{b}_3)$, $\texttt{left}_3(\texttt{l}_3)$, $\texttt{right}_3(\texttt{r}_3)$, $\texttt{left-back}_3(\texttt{lb}_3)$, $\texttt{right-back}_3(\texttt{rb}_3)$, $\texttt{left-front}_3(\texttt{lf}_3)$, and $\texttt{right-front}_3(\texttt{rf}_3)$.

Relative orientations must be given w.r.t. a *reference frame*, which can be *intrinsic* (orientation given by some inherent property of the reference object), *extrinsic* (orientation imposed by external factors), or *deictic* (orientation imposed by point of view). When reasoning about orientations, the reference frame is *implicitly* assumed to be the intrinsic orientation of the parent object (i.e., the one containing the objects involved), unless *explicitly* stated otherwise. The relative position is given by a topological/orientation relation pair:
`<primary_object, [topological,orientation], ref_object, ref_frame>`

## 2.1 The Structure of the Topological and Orientation Domains

Topological relations have a fork-like neighboring structure, whereas orientations form a uniform circular neighborhood on each level. As we will show below taking advantage of this structure leads to more efficient propagation algorithms. Since we use pairs of topological and orientation relations to represent relative positions, it is interesting to look at their combined structure. Figure 1 gives a simplified overview of level 3 orientations and linearly adjacent topological relations. Figure 2 shows

[1] Previous work used the term "projection" instead of topological. This has been changed because of the possible confusion with the use of the word in "2-D projections of 3-D scenes" and in "projective spatial prepositions" (which are related to what we call orientation relations).

a partial detailed view of two neighboring level 2 orientations and all 8 topological relations.[2] Arcs between nodes denote neighboring topological/orientation pairs. In

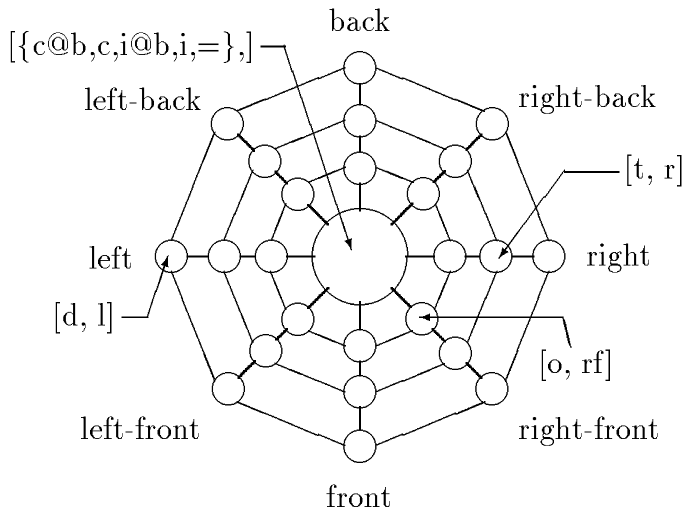



Figure 1: Combined structure of topological and orientation relations (overview)

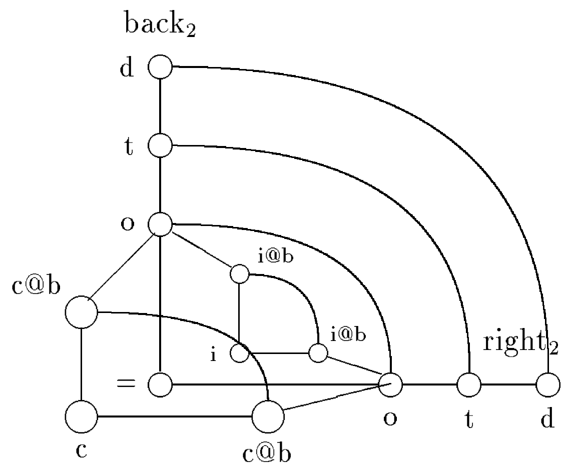



Figure 2: Combined structure of topological and orientation relations (detail)

the simplified visualization the containment and equality topological relations have been merged in a single node. Figure 2 is intended to be seen as a perspective view of a side cut of the 3-D visualization of the combined structure. In order not to clutter the figure unnecessarily, only the two neighboring orientations $\texttt{back}_2$ and $\texttt{right}_2$ are shown, together with all the links connecting neighboring relation pairs. The `c`, `=`, and `i` nodes are in the middle of the structure, because, according to our conventions, they are not oriented. The oriented nodes (e.g., [`i@b`,$\texttt{b}_2$]) are linked to neighboring

[2]The figures actually omit the links for neighboring relation pairs that result from the simultaneous change of topological and orientation relations as in [`d`,`f`] $\rightarrow$ [`t`,`rf`].

topological nodes of the same orientation (e.g., [o,$b_2$], [i,$b_2$]), and to equivalent nodes of same topology in neighboring orientations (e.g., [i@b,$r_2$], [i@b,$l_2$]).

# 3 Reasoning with Qualitative Representations

While all spatial reasoning tasks in a qualitative representation can be formulated as constraint satisfaction problems, the general techniques known in the literature incur an unnecessary computational overhead.[3] One reason for this overhead is, that they try to achieve global consistency among all constraints, whereas many spatial reasoning tasks require only local consistency. Another reason is, that they ignore the rich structure of space, which further constraints the set of possible solutions.

A qualitative description of a spatial configuration in form of a set of positional relations can be represented as a constraint network, where nodes corresponding to objects are linked by arcs corresponding to the relative positional relations between two objects. Sets of relations (relsets, for short), corresponding conceptually to disjunctions of possible relations between two objects, are ubiquitous in the algorithms to be described. Disjunctions are a way of expressing uncertainty about the "real" relation between the objects. Whenever the number of different relations is relatively small, as is always the case with qualitative representations, a bit-string representation of sets is the usual choice.

The propagation of constraints in the network is necessary to check the consistency of the relations among adjacent objects, for example, after insertion of a new relation or deletion of a previously assumed relation. In the following subsections, we will describe algorithms for these two tasks that take advantage of the structure of space described in the previous section.

## 3.1 Inserting New Relations

Inserting a new relation between two objects[4] affects not only those two objects but might yield additional constraints on the relations between other objects in the scene through constraint propagation. Allen (1983) introduced an algorithm for updating an interval-based temporal network that is based on constraint propagation. This algorithm effectively computes the closure of the set of temporal assertions after each new insertion. In what follows, we first describe Allen's original algorithm and introduce then several important enhancements based on structural properties of space.

### 3.1.1 Allen's Propagation Algorithm

The first step in the process of inserting a new relation $R_{ij}$ between $i, j$ is intersecting the new relset with whatever relset was known before (this is here the universal relset, in case no relation had been previously inserted, which behaves as "identity" for the set intersection operation). In case this intersection results in a more constrained

[3]Comprehensive reviews of the constraint satisfaction literature can be found in, e.g., Mackworth (1987), Meseguer (1989), Kumar (1992).

[4]We assume the relation has been transformed to the canonical implicit frame of reference.

relset, the nodes $i, j$ are placed in the queue for propagation.[5]

The computation of the closure is achieved by repeatedly calling a procedure PROPAGATE as long as there are entries in the queue. PROPAGATE does the main work by propagating the effects of the new constraint to "comparable" nodes (for now, assume all nodes in the network to be comparable). This is done by determining if the new relation between $i$ and $j$ can be used to constrain the relation between $i$ and other nodes, or between those other nodes and $j$ (Fig. 3). If one of these relations can indeed be constrained, then it is placed in the queue for further propagation. Furthermore, contradictions, characterized by an empty resulting relset, are signaled if found in this process. Contradictions will normally trigger a constraint relaxation process.

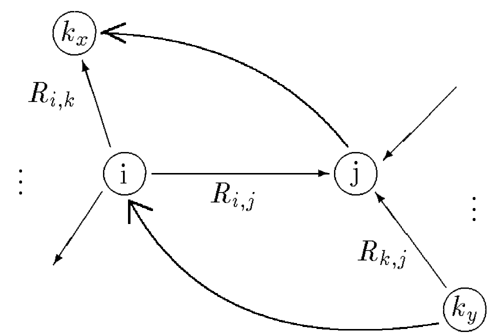



Figure 3: Propagation algorithm (visualization)

### 3.1.2 Example

To illustrate the propagation algorithm, we use the following set of initial relations, which have already been transformed to a canonical frame of reference (with possible corresponding verbal descriptions to the right):

1. `<T, [t,f], F>` *The table (T) is at the window (F).*
2. `<S, {[d,f],[t,f]}, T>` *In front of it there is a chair (S).*
3. `<W, {[d,l],[d,r]}, T>` *A bookcase (W) is next to it.*

Adding $R_{TF}$ creates the first link in the network (Fig. 4a). Adding $R_{ST}$ triggers the computation of $R_{SF}$ through composition of $R_{ST}/R_{TF}$ (Fig. 4b). Adding $R_{WT}$ leads to the computation of $R_{WF}$, and, if we allow inverting links, to $R_{WS} = R_{WT}/R_{TS}$ (Fig. 4c). Note, that the resulting relations in this last case are rather unspecific, because we do not know if the bookcase (W) is left or right of the table (T). Now suppose such information becomes indirectly available through a statement such as *"The bookcase (W) is to the right of the chair (S)."*, i.e., $R_{WS}$ = [d,r]. The intersection with the previously computed relset [{d,t},{l,lb,b,rb,r}] results again in [d,r]. The propagation algorithm computes new values for $R_{WT} = R_{WS}/R_{ST}$ and $R_{WF} = R_{WS}/R_{SF}$ both equal to [{d,t},{r,rf,f}]. The intersection with previously computed relsets leads finally to $R_{WT}$ = [d,r] and $R_{WF}$ = [{d,t},{r,rf,f}] (Fig. 4d).

[5]Note that, because the new link is obtained by intersection with the old one, it suffices to test if the new one is different from the old one, which might be cheaper.

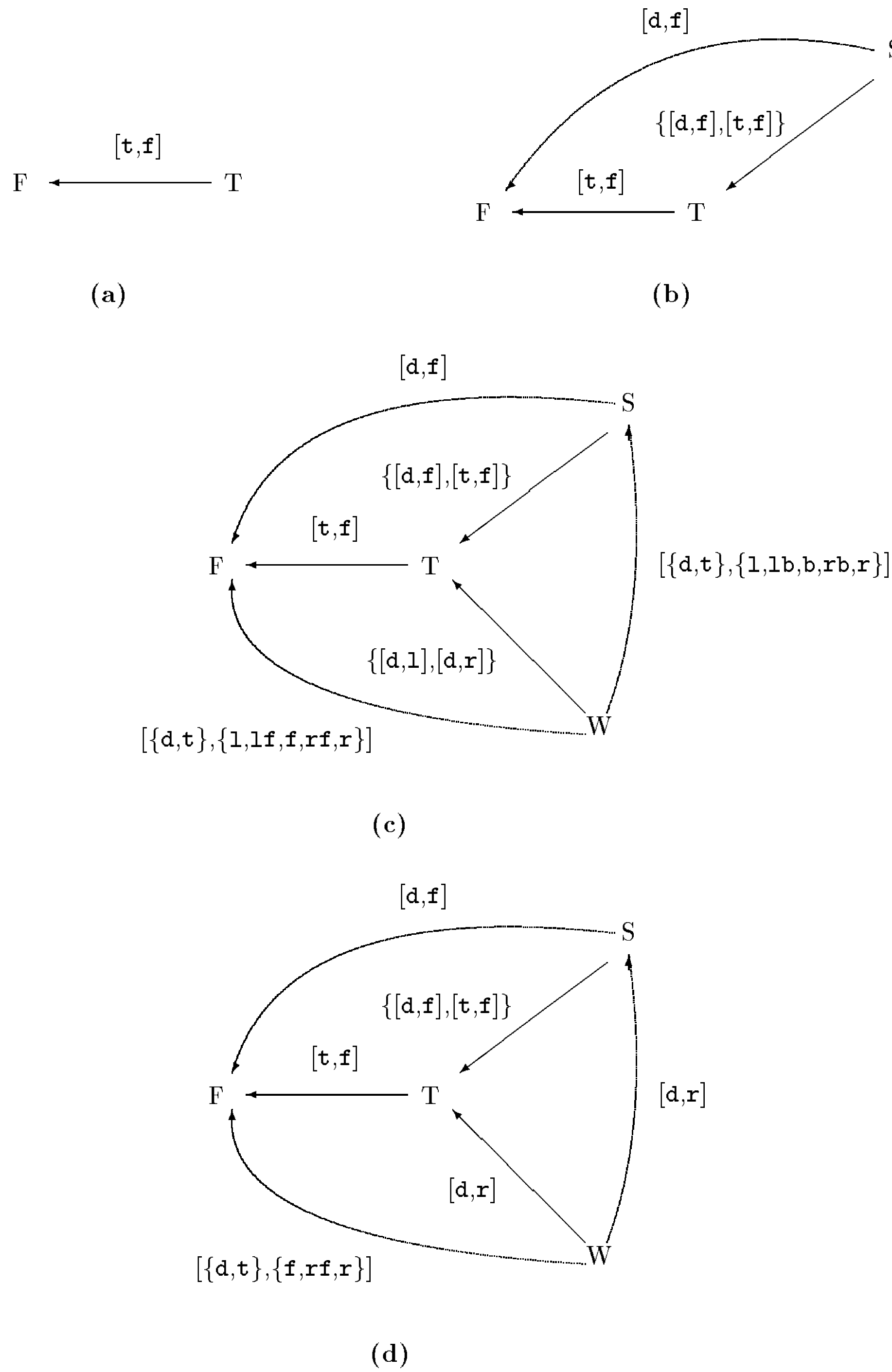



Figure 4: Propagation algorithm (example)

### 3.1.3 Exploiting the Structure of Space

**Reference Objects.** The predicate "comparable" is one place in Allen's algorithm that allows us to introduce modifications. Allen himself uses it to control propagation by introducing "reference intervals" to cluster sets of fully connected intervals, and defining comparable to be true only if the intervals share a reference interval, or one is the reference interval of the other. Similarly, we can limit the propagation by assigning parent objects (i.e., those resulting from hierarchical decomposition through containment) or functional clusters (for example, "dining-table-group" consisting of table, chairs, etc.) as "reference objects". However, we do not require full connectivity of objects with a common reference object. In most cases, it suffices if all objects are related to at least the central object in the group (e.g., the table in the dining-table-group).

**Degrees of Coarseness.** The next modification of the algorithm is not so straightforward. We want to take into account the fact that the information content of positional relations is not homogeneous. By information content we mean how much a relation constrains the relative position of objects. For example, a level 3 orientation is more constraining than a level 1 orientation, and `t` is more constraining than `d`. The specificity of a positional relation is, strictly speaking, a function of the relative size of the corresponding acceptance area.[6] However, establishing those areas is more involved than what is actually needed to differentiate among the relevant specificity classes. Thus, we use the "degree of coarseness" (doc), a number derived from the number of options left open by a relation (and confirmed by the "coarsening" factor of the resulting compositions), as a converse approximate measure. Small doc-values correspond to more constraining relations, while large doc-values correspond to less constraining relations.

| characteristics | | | | rel | doc |
|---|---|---|---|---|---|
| topological | | | size&shape restriction | `=` | 1 |
| topological | | | size&shape restriction | `c`, `i` | 4 |
| topological | oriented | boundary contact | size&shape restriction | `i@b`, `c@b` | 2 |
| topological | oriented | boundary contact | | `t` | 3 |
| topological | oriented | boundary contact | | `o` | 4 |
| topological | oriented | | | `d` | 8 |

Table 1: Degree of coarseness of topological relations (in the context of positional information)

Table 1 shows the degree of coarseness of topological relations in the context of positional information together with the factors contributing to the corresponding docs. `=` has the lowest doc, because it constrains the relative position of two objects

[6]The area in which a particular orientation is accepted as a valid description of the relative position of two objects is called "acceptance area".

to an unique value; c and i are not oriented but constrain the position of one of the objects to be within the boundaries of the other; t and o demand boundary contact between the objects, thus restricting their positions, whereas d has the highest doc and is useful only when used together with an orientation;[7] finally i@b and c@b are quite specific because they are oriented, demand boundary contact and have the size and shape restriction of the containment relations.

| characteristics | | | rel | doc |
|---|---|---|---|---|
| orientation | level 3 | corners | $\mathtt{lf}_3$, $\mathtt{rf}_3$, $\mathtt{lb}_3$, $\mathtt{rb}_3$ | 1 |
| | | sides | $\mathtt{l}_3$, $\mathtt{r}_3$, $\mathtt{f}_3$, $\mathtt{b}_3$ | 3 |
| | level 2 | | $\mathtt{l}_2$, $\mathtt{r}_2$, $\mathtt{f}_2$, $\mathtt{b}_2$ | 4 |
| | level 1 | | $\mathtt{l}_1$, $\mathtt{r}_1$ | 8 |
| | | | $\mathtt{f}_1$, $\mathtt{b}_1$ | 8 |

Table 2: Degree of coarseness of the most common orientation relations (in the context of positional information)

Table 2 lists the degree of coarseness of the most common orientation relations. It corresponds roughly to the number of basic sectors covered by each of the relations. For the case of extended objects, a further distinction between more specific corner orientations and less specific side orientations can be made, as shown here for the third level. The doc of a topological/orientation pair is the sum of the docs of its components. The doc of a relset is the sum of the docs of the set members.

```
To ADD R_ij
  begin
    if doc(R_ij) > maxdoc
       then exit;
    Old ← N(i,j);
    N(i,j) ← Combine(N(i,j),R_ij);
    If N(i,j) = ∅
       then Signal contradiction;
    if N(i,j) ≠ Old
       then add <i,j,doc(N(i,j))> to Agenda;
    Nodes ← Nodes ∪ {i,j};
  end;
```

Figure 5: Adding a new relation

The doc-values are used to control propagation in the following way: While adding a new relation (Fig. 5), its doc is compared against a pre-defined constant maxdoc, above which a relation is considered too unspecific to be worth adding. The value of

[7]The unspecificity of d should not be overrated: In most applications, the distance between objects is bounded by the extension of the parent object.

`maxdoc` is application dependent. Setting it to half the sum of the docs of all possible relations at the granularity levels allowed has proven to be a useful heuristic. If the doc of the relset is below or equal to `maxdoc`, then it is inserted by combining it with the previously known relset. COMBINE can be assumed for now to be equivalent to set intersection (it will be modified below). If the new combined relation is different from the previously known, it is placed on the agenda for further propagation. An agenda is a data structure to keep track of what to do next based on some sort order. It is usually implemented as an ordered list of queues. In this case, we use the doc of the relation to be propagated as entry key, allowing us to propagate first more specific relations, i.e., those with lower docs. Relations with equal docs are processed in a first-in-first-out manner.

```
To ComputeEffects
  While Agenda is not empty do
    begin
      Get next <i,j,d> from Agenda;
      If d ≤ maxdoc
         then Propagate(i,j);
    end;
```

Figure 6: Computing the effects of a new added relation

The effects of new relations added to the network can be computed by calling COMPUTEEFFECTS (Fig. 6), which fetches the first entry from the agenda, double checks its doc to be below the limit (this is necessary because PROPAGATE also adds to the agenda, and could also be used for dynamic control through changing limits), and calls PROPAGATE.

PROPAGATE (Fig. 7) is essentially the same as in the original algorithm, except for COMBINE and the use of an agenda instead of a queue. Note, that the propagation algorithm assumes all relations, including semantically similar relations at different levels of granularity (e.g., $b_2$ and $b_3$), to be mutually exclusive. Knowing about the hierarchical structure of the relational domains used, allows us to extend the algorithm to succeed in cases where the original algorithm would fail, and signal a contradiction. For example, the intersection of a set containing only [t,$l_2$] and a set containing only [t,$l_3$] shouldn't be "empty", signaling a contradiction, but rather lead to preferring [t,$l_3$]. This can be done by looking at the range representation of $l_2$ and $l_3$, instead of viewing them as unrelated relations. Furthermore, the intersection of sets containing neighboring relations, such as for example {[t,$l_3$]} and {[t,$lb_3$]}, should not be "empty", but rather lead to a coarser relation such as [t,$l_2$]. These extensions are implemented by the modified COMBINE in Fig. 8, which checks to see if subsumed or neighboring relations are available, if the regular intersection of two relsets is empty. In the first case, the subsumed (more specific) relation is placed in the new combined set. In the second case, the "next common coarse relation" (nccr) is added to the combined set. This is a form of constraint relaxation embedded in the propagation process (Hernández and Zimmermann 1992).

Continuing our example, assume the initial relation between W and T to be

```
To  PROPAGATE i,j
  begin
    For each node k such that Comparable(i,k) do
      begin
        New ← Combine(N(i,k),Constraints(N(i,j),N(j,k)));
        If New =  ∅
           then Signal contradiction;
        If New ≠ N(i,k)
           then add <i,k,doc(New)> to Agenda;
        N(i,k) ← New;
      end;
    For each node k such that Comparable(k,j) do
      begin
        New ← Combine(N(k,j),Constraints(N(k,i),N(i,j)));
        If New =  ∅
           then Signal contradiction;
        If New ≠ N(k,j)
           then add <k,j,doc(New)> to Agenda;
        N(k,j) ← New;
      end;
  end;
```

Figure 7: Weighted Propagation

```
To  COMBINE R1, R2
  begin
    temp ← R1 ∩ R2;
    if temp ≠  ∅
       then Return temp;
    C ← ∅;
    For each r1 ∈ R1
      For each r2 ∈ R2
        begin
          if subsumes(r1,r2) then C ← C ∪ r2;
          if neighbor(r1,r2) then C ← C ∪ nccr(r1,r2);
        end;
    Return C;
  end;
```

Figure 8: Combining two relsets

{[d,$l_2$],[d,$r_2$]}, and suppose that we learn later that $R_{WT}$ = [d,$rf_3$]. Instead of returning an empty set as an intersection of the two sets would, COMBINE recognizes that $r_2$ subsumes $rf_3$ and returns [d,$rf_3$] as result. After propagation, the relations between W and F, and between W and S are constrained to be {[d,f],[d,rf]}, and [{d,t}, {b,rb,r}], respectively.

### 3.1.4 Complexity

Allen (1983) and later Vilain, Kautz, and van Beek (1990) showed the algorithm to run in polynomial time w.r.t. the number of intervals in the database. $O(n^3)$ set operations are required in the worst case for the algorithm to run to completion for $n$ intervals, because there are at most $n^2$ relations between $n$ intervals, each of which can only be non-trivially updated (and correspondingly entered on the queue for propagation) a constant number of times (each update removes at least one of 13 possible relations). In turn each of the $O(n^2)$ propagations requires $O(n)$ set operations resulting in the said $O(n^3)$ set operations (which in a bit-string implementation of sets can be assumed to take constant time each).

As a consequence of the modified COMBINE, `New` $\neq$ `N(i,k)` in PROPAGATE being true does not imply `New` $\subset$ `N(i,k)`. However, the worst case complexity analysis of the original algorithm is not affected by this change, because every non-trivial update of a relset either removes at least one relation (intersection, original algorithm), or replaces one relation by a subsumed one, or one or more fine relations by a coarser one. The average case performance is greatly improved by the modifications described, particularly by the hierarchical decomposition, and by the preferential propagation of specific relations.

## 3.2 Deleting Relations

Deleting relations between two nodes is not just a matter of removing a link from the data structure representing the network. The consequences of the propagation of the constraint now being deleted must be taken back as well.

This requires a further modification of the insertion algorithm described in the previous section to maintain justifications for derived constraints. Instead of just modifying a link to contain the new constrained relset, we also record the link whose propagation led to the new constraint. In general, a justification is a list of the links that were used to derive the relation of the new link. To allow for multiple derivation paths, usually a list of justifications is maintained. At the same time a link has pointers to those links that it in turn served in deriving in a so called justificands list. Relations originally entered by the user (considered as "premises") have an empty justification list. This information is usually maintained in a separate "dependency network", where the links of the constraint network are the "nodes",[8] and the arcs connecting them represent the dependency structure. Fig. 9 shows a typical graphical representation of dependency networks, and illustrates the terminology introduced above (empty justifications marking premises are shown as solid rectangles). The dependency network can also serve as direct indexing and retrieval mechanism for sets

[8]Note, however, that each particular relset $R_{ij}$ between objects $i$ and $j$ is recorded as a separate node of the dependency network.

of consistent relsets. The process operating on a dependency network is called "reason

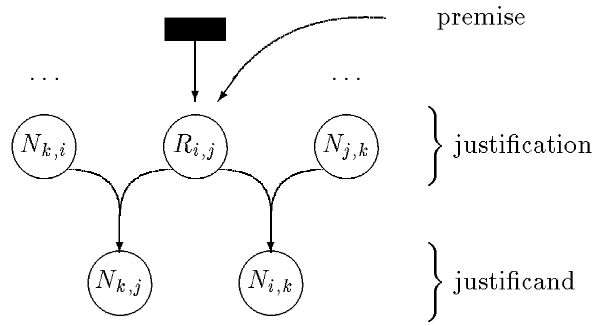


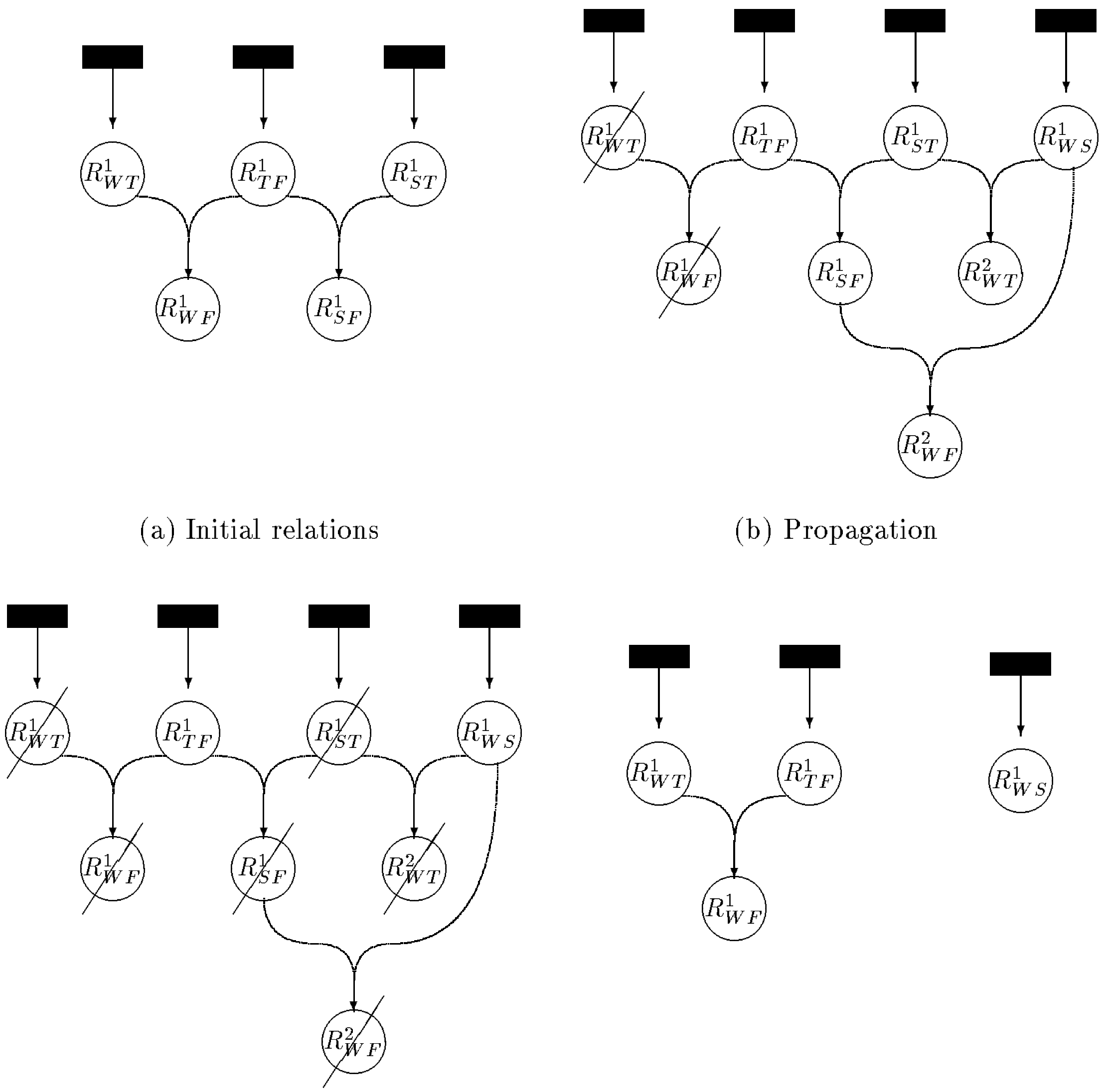

Figure 9: Graphical representation of dependency networks

maintenance". Because they can operate independently from the problem solver (the constraint propagation algorithm, in our case), reason maintenance algorithms and the corresponding dependency networks have been developed as separate systems called "Reason Maintenance Systems" (RMS). RMSs were introduced in the late 70s in the context of computer-aided circuit analysis by Stallman and Sussman (1977), and first studied as independent systems by Doyle (1979).[9] Further variants were later developed by McAllester (1980), de Kleer (1986), and others.

Based on the dependency information recorded during the propagation of constraints, the RMS establishes which nodes of the dependency network are affected by the deletion of a link in the constraint network. Figure 10a shows the dependency network for the example from the previous section, after the first set of relations (the "premises" $R_{FT}$, $R_{ST}$, $R_{WT}$) and its consequences ($R_{SF}$, $R_{WF}$) have been established.[10] The one as superscript indicates that this particular relset is the first one assigned to that link. Figure 10b shows the next step in which $R_{WS}$ is added as a premise, resulting in the derivation of new values for $R^2_{WT}$ and $R^2_{WF}$ (indicated by the two as superscript). Note that the old values $R^1_{WT}$ and $R^1_{WF}$ are cancelled out. Now suppose we originally got the wrong description and are forced to delete the relation between S and T. In that case, all the consequences derived by using that relation must be removed as well, as shown in Fig. 10c. Unfortunately, we are left only with the relations $R^1_{TF}$ and $R^1_{WS}$, even though we had previously the relations $R^1_{WT}$ and $R^1_{WF}$ that did not depend on the erased relation. The RMS mechanism, however, does usually reconsider the evidence in support of all nodes based ultimately on premises, when other values are erased, and would restore in this case the old values for $R^1_{WT}$ and $R^1_{WF}$ (Fig. 10d). Thus, maintaining dependencies not only allows us to retract the consequences of deleted relations, but also helps us to avoid repeated computations. Of course, there is always a tradeoff between the overhead of caching inferences and the costs of recomputing them, particularly when restrictive propagation policies apply.

[9]They were originally called "Truth Maintenance Systems", a somewhat misleading term, still often found in the literature. We prefer the name "Reason Maintenance Systems" following McDermott (1983).

[10]For better readability, we are omitting here the propagation through inverted links (e.g., $R_{WS} = R_{WT}/R_{TS}$).




(a) Initial relations (b) Propagation

(c) Deletion (d) Old context restored

Figure 10: Example dependency network

Upon deletion of a link in the constraint network, reason maintenance is triggered, leading to the corresponding erasure of a node in the dependency network. The basic idea of reason maintenance is similar to the "garbage collection" mechanism in LISP systems: The status and support of all elements in the network are reset first, and then recomputed by a recursive mark procedure that—starting from the premises—finds all nodes with "well-founded" support. Nodes without such support are physically removed from the network. While the procedure illustrates the basic idea, it is usually not very efficient, because all nodes in the network are visited (only once, but anyway). Thus, other variants start from the erased node, and recursively mark all its justificands for erasure. However, because of the potential of circular dependencies, a somehow more involved procedure is needed (Hernández 1992). It resets the non-circular `supporter/supportee` links and assigns them a tentative status, assuming that all nodes marked inactive ($\neq$ {`IN, OUT`}) will ultimately be erased (`OUT`). If that happens not to be the case, i.e., if one such node becomes active (`IN`), because of a valid justification, all its supportees must be reconsidered, because they might have been assigned a new status based on the assumption that the supporter be `OUT`. Details of this algorithm can be found in (Hernández 1984), where several variants were implemented based on ideas in (Charniak et al. 1980) and (Doyle 1979).

# 4 Conclusion

Even though the solution techniques for the general constraint satisfaction problem available in the literature represent large efficiency improvements over the obvious backtracking algorithm, they are "limited by their generality". That is, being general, domain-independent techniques, they ignore the structure of the relational domain. Thus, in this paper we show that taking the structure of the richly constrained spatial domain into consideration leads to more efficient algorithms. One way of exploiting this structure is introducing heuristics to control the propagation of constraints by using the hierarchical and functional decomposition of space to limit constraints to physically adjacent objects. Spatial reasoning can be done at coarser or finer levels of that structure, depending on the kind of information available. In particular, if only coarse information is available, the reasoning process is less involved than if more details are known. Also, a weighting of positional relations according to their information content is used to avoid "information decay" in the network due to the propagation of weak relations. Retracting the consequences of previous propagations in order to maintain the consistency of the knowledge base requires keeping track of dependencies and a reason maintenance mechanism.

## Acknowledgements

The research reported here has benefited from numerous discussions with W. Brauer, C. Freksa, C. Habel, S. Högg, D. Kobler, I. Schwarzer, K. Zimmermann and others.

# References

Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Communications* *of the ACM*, *26*(11), 832–843.

American Association for Artificial Intelligence, 8th-AAAI (1990). *Proc. of the Eighth National Conference on Artificial Intelligence*, Menlo Park/Cambridge. AAAI Press/The MIT Press.

Bobrow, D. (Ed.). (1984). *Qualitative Reasoning about Physical Systems.* Elsevier Science Publishers, Amsterdam.

Chang, S.-K. and Li, Y. (1988). Representation of multi-resolution symbolic and binary pictures using 2DH-strings. In *Proceedings of the IEEE Workshop on Language for Automation*, pp. 190–195.

Chang, S.-K., Shi, Q.-Y., and Yan, C.-W. (1987). Iconic indexing by 2D-strings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *9*(3), 413–427.

Charniak, E., Riesbeck, C. K., and McDermott, D. V. (1980). *Artificial Intelligence Programming.* Erlbaum, Hillsdale, N.J.

Cohn, A. G., Cui, Z., and Randell, D. A. (1992). Logical and computational aspects of spatial reasoning. In Pribbenow, S. and Schlieder, C. (Eds.), *Spatial Concepts: Connecting Cognitive Theories with Formal Representations.* ECAI-92 Workshop.

Doyle, J. (1979). A truthmaintenance system. *Artificial Intelligence*, *12*, 231–272.

Egenhofer, M. J. (1989). A formal definition of binary topological relationships. In Litwin, W. and Schek, H.-J. (Eds.), *Third International Conference on Foundations of Data Organization and Algorithms*, Vol. 367 of *Lecture Notes in Computer Science*, pp. 457–472. Springer-Verlag, Berlin.

Egenhofer, M. J. (1991). Reasoning about binary topological relations. In *Second Symposium on Large Spatial Databases*, Vol. 525 of *Lecture Notes in Computer Science.* Springer-Verlag.

Egenhofer, M. J. and Al-Taha, K. K. (1992). Reasoning about gradual changes of topological relationships. In Frank et al. (1992), pp. 196–219.

Egenhofer, M. J. and Franzosa, R. (1991). Point-set topological spatial relations. *International Journal of Geographical Information Systems*, *5*(2), 161–174.

Egenhofer, M. J. and Sharma, J. (1993). Assessing the consistency of complete and incomplete topological information. *Geographical Systems*, *1*(1). To appear.

Frank, A. U., Campari, I., and Formentini, U. (Eds.). (1992). *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. Intl. Conf. GIS—From Space to Territory. Pisa, Sept. 1992*, Vol. 639 of *Lecture Notes in Computer Science*, Berlin. Springer-Verlag.

Frank, A. U. (1991). Qualitative spatial reasoning with cardinal directions. In Kaindl, H. (Ed.), *7. Österreichische Artificial Intelligence Tagung.* Springer-Verlag.

Freksa, C. (1992a). Temporal reasoning based on semi-intervals. *Artificial Intelligence*, *54*, 199–227.

Freksa, C. (1992b). Using orientation information for qualitative spatial reasoning. In Frank et al. (1992), pp. 162–178.

Freksa, C. and Zimmermann, K. (1992). On the utilization of spatial structures for cognitively plausible and efficient reasoning. In *Procs. of the 1992 IEEE International Conference on Systems, Man, and Cybernetics Proceedings, Chicago.*

Guesgen, H.-W. (1989). Spatial reasoning based on Allen's temporal logic. Technical Report TR-89-049, ICSI, Berkeley, CA.

Hernández, D. (1984). Modulare Softwarebausteine zur Wissensrepräsentation. IMMD (IV) and RRZE Universität Erlangen-Nürnberg.

Hernández, D. (1991). Relative representation of spatial knowledge: The 2-D case. In Mark and Frank (1991), pp. 373–385.

Hernández, D. (1992). *Qualitative Representation of Spatial Knowledge.* Ph.D. thesis, Institut für Informatik, Technische Universität München.

Hernández, D. and Zimmermann, K. (1992). Default reasoning and the qualitative representation of spatial knowledge. In preparation.

Jungert, E. (1988). Extended symbolic projection used in a knowledge structure for spatial reasoning. In *Proceedings of the 4th BPRA Conf. on Pattern Recognition.* Springer-Verlag, Berlin.

de Kleer, J. (1986). An assumption based truth maintenance system. *Artificial Intelligence*, *28*, 127–162.

Kuipers, B. J. and Levitt, T. (1988). Navigation and mapping in large-scale space. *AI Magazine*, *9*(2), 25–43.

Kumar, V. (1992). Algorithms for constraint satisfaction problems: A survey. *AI Magazine*, *13*(1), 32–44.

Levitt, T. and Lawton, D. (1990). Qualitative navigation for mobile robots. *Artificial Intelligence*, *44*, 305–360.

Mackworth, A. K. (1987). Constraint satisfaction. In Shapiro (1987).

Mark, D. M. and Frank, A. U. (Eds.). (1991). *Cognitive and Linguistic Aspects of Geographic Space.* NATO Advanced Studies Institute. Kluwer, Dordrecht.

McAllester, D. (1980). An outlook on truth maintenance. Memo 551, MIT Artificial Intelligence Laboratory, Cambridge, MA.

McDermott, D. V. (1983). Contexts and data dependencies: A synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *5*(3), 237–246.

McDermott, D. V. (1992). Reasoning, spatial. In Shapiro, E. (Ed.), *Encyclopedia of Artificial Intelligence* (Second edition)., pp. 1322–1334. Wiley.

Meseguer, P. (1989). Constraint satisfaction problems: An overview. *AI Communications*, *2*(1), 3–17.

Mukerjee, A. and Joe, G. (1990). A qualitative model for space. In 8th-AAAI (1990), pp. 721–727.

Shapiro, E. (Ed.). (1987). *Encyclopedia of Artificial Intelligence.* Wiley.

Smith, T. R. and Park, K. K. (1992). Algebraic approach to spatial reasoning. *International Journal of Geographical Information Systems*, *6*(3), 177–192.

Stallman, R. and Sussman, G. J. (1977). Forward reasoning and dependency-directed backtracking. *Artificial Intelligence*, *9*(2), 135–196.

Topaloglou, T. (1991). Representation and management issues for large spatial knowledge bases. Technical Report, Dept. of Computer Science, University of Toronto, Toronto, Ontario.

Vilain, M., Kautz, H., and van Beek, P. (1990). Constraint propagation algorithms for temporal reasoning: A revised report. In Weld and de Kleer (1990). Revised version of a paper that appeared in Proceedings of AAAI 1986, pp. 377–382.

Weld, D. S. and de Kleer, J. (Eds.). (1990). *Readings in Qualitative Reasoning about Physical Systems.* Morgan Kaufmann, San Mateo, CA.