Hierarchical Reasoning about Direction Relations

Dimitris Papadias

Artificial Intelligence Research Division, Institute for Applied Information Technology, GMD,
D-53754 Sankt Augustin, Germany.
E-mail: dimitris@nathan.gmd.de, fax: +49-2241-142072, tel: +49-2241-142329

Max J. Egenhofer

National Center for Geographic Information and Analysis, Department of Spatial Information Science and Engineering, Department of Computer Science, University of Maine, Orono, ME 04469-5711, USA.
E-mail:max@spatial.maine.edu, fax: 207- 581-2206, tel: 207-581-2127

Jayant Sharma

NEDC Oracle Engineering , 110 Spit Brook Rd, ZK02-1/019, Nashua, NH 03062, USA.
E-mail: jsharma@us.oracle.com, tel: 603-881-6322

Abstract Spatial reasoning is an important area of GIS and Spatial Databases research. This paper deals with reasoning about direction relations (east, northeast) in geographic hierarchies. We assume a database that stores the direction relations between objects in the same geographic entity and propose algorithms for the inference of relations between objects in different entities. We present two types of inference: the first one uses the relations of ancestor regions, while the second is based on composition of spatial relations and path consistency. For both types we provide inference rules, illustrate examples, and study the computational complexity. Although we use a specific set of relations for demonstration purposes, the algorithms are applicable to any set of direction relations provided with appropriate inference rules.

Keywords: Spatial Reasoning, Direction Relations, Path Consistency

1. INTRODUCTION

The hierarchical representation of space has a strong psychological motivation (Hirtle and Jonides, 1985) and numerous computational advantages that have been exploited in a number of areas such as Data Structures (Guttman, 1984) and Wayfinding (Car and Frank, 1994). This paper studies hierarchical reasoning about direction relations in spatial databases that cannot resort to coordinate-based representations. Such situations are typically found in narratives, trip reports, and scientists' field notes. We assume that there only exists relative information about the objects within a region, and inclusion relations (i.e., the hierarchical structure). The goal is to infer the direction relations between objects located in different regions and detect potential inconsistencies.

As an example consider that you have the information that location X_1 is east of X_2 in the map (or more generally, spatial representation) of region A_1, and that X_2 is east of X_3 in the map of (neighbouring) region A_2. In addition you learn that X_3 belongs also to region A_3 which is northeast of A_1. The above data contains an inconsistency about the relation between X_1 and X_3: from their relation with X_2, we can infer through transitivity that east(X_1,X_3); using the relation of their ancestor regions (northeast(A_3,A_1)) we can also infer that northeast(X_3,X_1). Such inconsistencies may occur by combining spatial knowledge from different sources and alternative representations like images, topographic surveys or verbal descriptions (for an extended discussion see Frank, 1992). Spatial inference mechanisms are essential for explicating relations and enforcing consistency in the database.

The rest of the paper is organized as follows: Section 2 defines direction relations between points and regions, and describes spatial databases preserving directions. Section 3 discusses the retrieval of explicit relations and outlines a framework for the computation of the cost. An algorithm for the inference of the relation between points using the relations of their ancestor regions is presented in Section 4. Section 5 describes a complementary form of inference that uses chains of common points and achieves path consistency for the whole database. Section 6 concludes with comments.

2. DIRECTION RELATIONS AND SPATIAL DATABASES

Unlike topological relations where there is a set (Egenhofer and Franzosa, 1991) of widely used relations in both research literature and commercial products, there are not universally accepted definitions for direction relations. People have applied different types of direction relations to match different needs that range from cognitive modelling (Herskovits, 1986) to image similarity retrieval (Lee et al., 1992) and from robot navigation (Holmes and Jungert, 1992) to user interfaces (Roussopoulos et al., 1988; Papadias and Sellis, 1995). Although in this paper we use projection-based relations, the proposed methods are applicable to other types of directions (for a discussion on alternative types see Frank, 1992; Hernandez, 1994).

According to our model, the relation between two points is determined by the position of the primary object with respect to the projection lines from the reference object to the coordinate axes (Freksa, 1992; Papadias and Sellis, 1994). In this way, nine mutually exclusive relations can be defined between points:

$NW(P_1,P_2)$ $X(P_1)<X(P_2)$ $Y(P_1)>Y(P_1)$,

$RN(P_1,P_2)$ $X(P_1) = X(P_2)$ $Y(P_1)>Y(P_1)$,

$NE(P_1,P_2)$ $X(P_1) >X(P_2)$ $Y(P_1)>Y(P_1)$,

$RW(P_1,P_2)$ $X(P_1)<X(P_2)$ $Y(P_1)=Y(P_1)$,

$SP(P_1,P_2)$ $X(P_1)=X(P_2)$ $Y(P_1)=Y(P_1)$, and the converse relations:

$RE(P_1,P_2)$ $RW(P_2,P_1)$, $SW(P_1,P_2)$ $NE(P_2,P_1)$, $RS(P_1,P_2)$ $RN(P_2,P_1)$, $SE(P_1,P_2)$ $NW(P_2,P_1)$.

U denotes the *universal* relation, the disjunction of all primitive relations. The relation denotes the *empty* relation (the relation that arises during inconsistencies). The above relations form a relation algebra and can be used for relation-based reasoning. They constitute the set of *high resolution* relations; we also define a set of *low resolution* relations using disjunctions: N=NW∫RN∫NE, E=NE∫RE∫SE, S=SW∫RS∫SE, W=NW∫RW∫SW, SL=RW∫SP∫RE (SameLevel), and SH=RN∫SP∫RS (SamewidtH). The projection-based definitions are applied for regions in an analogous way. There are 13 mutually exclusive relations between intervals in 1D space (Allen, 1983). If we extend Allen’s relations to 2D space we get the 169 primitive relations between regions of Figure 1.

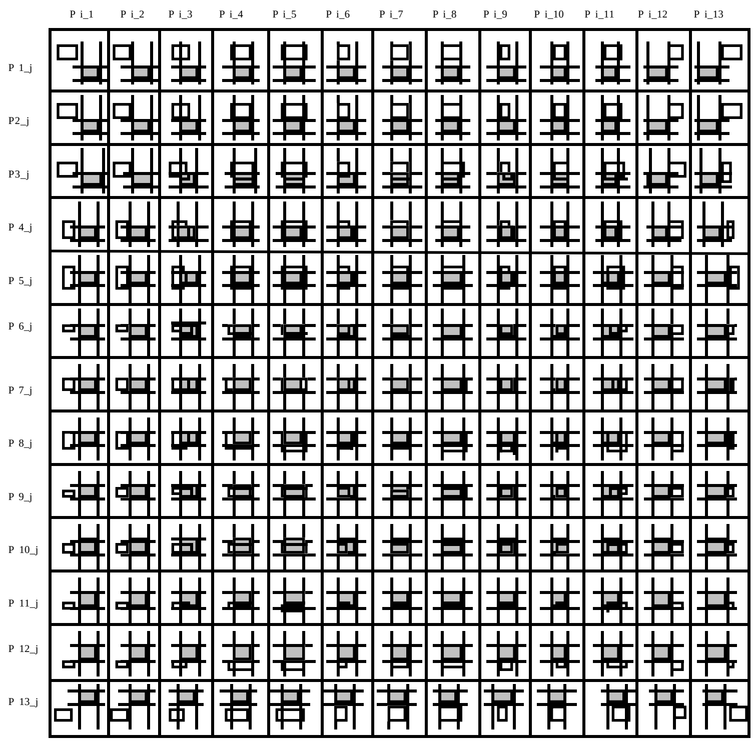


Figure 1 Projection relations in 2D space

A number of *relation-based* systems that store only the above type of relations and discard other forms of spatial information (such as shape, distance and topological relations) have been proposed. Chang et al., (1987) designed *2D strings* for iconic indexing in image databases. A 2D string is a pair of one-dimensional strings that represent the symbolic projections of the objects on the x and y axis. Glasgow and Papadias (1992) developed *symbolic arrays*, which are nested array structures that preserve directions relations among the distinct parts of complex spatial entities at different levels. Most previous work, however, has focused on the representation and processing of explicit relations and the proposed systems do not include mechanisms for inference and inconsistency checking.

Let *DB* be a spatial database of maps each corresponding to a distinct region. For every map there is a relation-based representation (2D string, symbolic array, a relational table or a set of binary predicates) that stores the relations between all pairs of objects in the region. The objects in the map can be either points or regions but not both (the regions that contain points are called *leaf regions*). Each pair of objects in a map is related by a primitive direction relation explicitly represented. The relation of each point with itself is SP and the projection relation of each region with itself is $P_{7\text{-}7}$. Converse pairs of objects are related by converse relations.

We use the notation A R(X_1,X_2) to express that objects X_1 and X_2 are related by relation R in the map of region A. The hierarchy is represented by pointers to next-level areas (IN relation). *DB* IN(X_i,A_j) denotes that object (point or region) X_i is a part of (therefore, totally contained in) the next level region A_j. IN* is the transitive closure of IN: *DB* IN*(X_i,A_j) *DB* IN(X_i,A_j) ∫ ∃ A_k [*DB* IN*(X_i,A_k) | *DB* IN*(A_k,A_j)]. IN* needs not be explicitly represented, but can be computed by a recursive function that traverses the hierarchy bottom-up, and marks all the ancestors of an object in the hierarchy. For demonstration, we use the example of Figure 2a; Figure 2b illustrates the hierarchy and the relations explicitly represented.

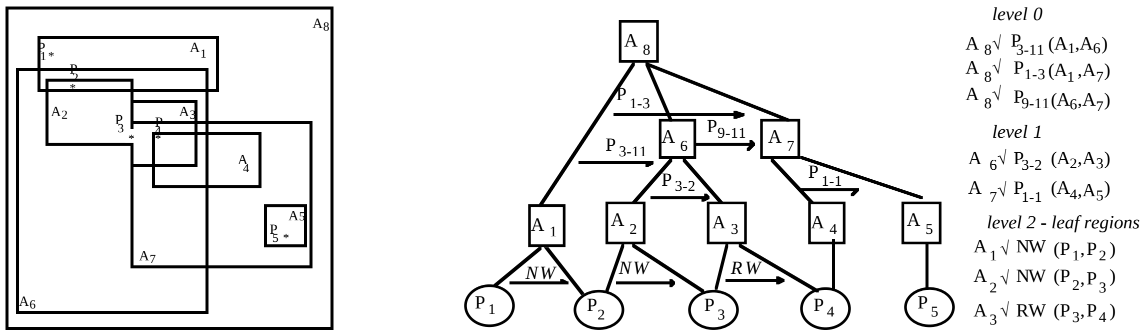


Figure 2 Example of inference through regions

R, R_1, R_2 ... denote relation variables between points, and r, r_1, r_2 ... between regions. The general problem is to retrieve the (explicit, or implicit) relation between any pair of points P_i and P_j in the database: *DB* $R(P_i,P_j)$. There are three cases regarding the direction relations between points: explicit retrieval, inference through regions and inference through points. In the next sections we discuss algorithms that extract the relation between all pairs of points and detect inconsistencies. For each case we provide rules of inference, describe extensive examples, and obtain formulas for the cost.

3. EXPLICIT RETRIEVAL

According to explicit retrieval the relation between points P_i and P_j is R, if there is a leaf region A in which the two points are related by R: $\exists$ A (A $R(P_i,P_j)$) *DB* $R(P_i,P_j)$. Inconsistencies in this case arise when P_i and P_j exist together in multiple maps and their relations in these maps are different (an inconsistency of this form would be: A_1 $NW(P_1,P_2)$ and A_2 $NE(P_1,P_2)$). The following algorithm performs explicit retrieval by retrieving all leaf regions and examining the relations between all pairs of points in them[1]. We assume an *initialization* process that assigns *U* to the relation between each pair of distinct points (*SP* for identical points).

```
Explicit_retrieval
for each (leaf) region A_k
        retrieve A_k;
        for each point P_i such that DB IN(P_i,A_k)
                        for each point P_j such that DB IN(P_j,A_k) and i<j
                                get the relation R' : A_k  R'(P_i,P_j);
                                R(P_i,P_j)= R(P_i,P_j)R';
                                if R= then return INCONSISTENCY DUE TO EXPLICIT RETRIEVAL;
                                        else R(P_j,P_i)=converse(R(P_i,P_j));
                        end-for
        end-for
end-for
```

[1] All the algorithms assume that information in each region is arc consistent: $AR(P_i, P_j)$ $Aconverse(R(P_j, P_i))$ and work

In order to obtain formulas for the cost of the algorithms we make the following simplifications (although such simplifications may not apply for real applications, they provide a good measure for the expected cost in most cases). Each region contains *k* objects (points or other regions). Each object belongs to *m* regions in the upper hierarchy level, except for the region at the top (0 level) that does not belong to any region, and the objects at level 1 that belong only to the top-level region. It is always the case that $k/m>1$ and in regular applications $k/m>>1$. *N* is the total number of points in the database and *h* is the height of the hierarchical structure. We assume that there is a buffer that stores the $N(N-1)/2$ relations between all pairs of points.

The cost is a function of the *number of map retrievals* because such operations require access to secondary storage (i.e., retrieval of the disk pages that contain the map). This is common practice in database literature where indexing methods are compared on the number of accessed pages from the disk (Guttman 1984; Papadias and Theodoridis, in press). In the case of explicit retrieval we have to retrieve all leaf regions. Due to the fact that leaf regions store all points and their copies[2], their number is *mN/k*, resulting in the same number of map retrievals. Unlike explicit retrieval which is straightforward, the other two cases require inference mechanisms that potentially search large parts of the database.

4. SPATIAL INFERENCE THROUGH REGIONS

According to *inference through regions*, P_i and P_j do not exist in the same region, but their relation can be inferred from the relations between their ancestor regions. The notation rR means that when the relation r holds true between two regions, then the relation R holds between all pairs of points in the regions. For instance, P_{1-1} NW; if two regions are related by projection relation P_{1-1}, the relation between any two points, each belonging to one region, is NW. Inference through regions can be described as: $[\exists A_k \exists A_l$ (*DB* IN*(P_i,A_k) *DB* IN*(P_j,A_l) *DB* $r(A_k,A_l))$ | (r R)] *DB* $R(P_i,P_j)$. Inconsistencies during inference through regions arise when the relation between P_i and P_j in some map is not consistent with the relation between some of their ancestors regions. As an example consider: A $RN(P_1,P_2)$, DB IN*(P_1,A_1), DB IN*(P_2,A_2), and DB $P_{1-1}(A_1,A_2)$.

In the case that the projections of two regions are disjoint on both axes (projections P_{1-1}, P_{1-13}, P_{13-13}, and P_{13-1} in Figure 1), then high resolution information can be inferred for both south-north and west-east directions. However, not all projections allow such inferences regarding the relations between points. When the projections are disjoint on only one axis, low resolution relations about this axis can be derived, but information on the other axis is lost. Figure 3 summarises the relations that can be derived about points given the projection relation between regions. The entries with *U*, correspond to overlapping projections on both axes (in this case no conclusion can be drawn about the relations between points).

---

[2] Since an object appears *m* times in the next hierarchical level, we say that it has one original representation and m-1

P	1	2	3	4	5	6	7	8	9	10	11	12	13
1	NW	NWRN	N	N	N	N	N	N	N	N	N	NERN	NE
2	NWRW	NWRNRWSP	NSL	NSL	NSL	NSL	NSL	NSL	NSL	NSL	NSL	NERNRESP	NERE
3	W	WSH	U	U	U	U	U	U	U	U	U	ESH	E
4	W	WSH	U	U	U	U	U	U	U	U	U	ESH	E
5	W	WSH	U	U	U	U	U	U	U	U	U	ESH	E
6	W	WSH	U	U	U	U	U	U	U	U	U	ESH	E
7	W	WSH	U	U	U	U	U	U	U	U	U	ESH	E
8	W	WSH	U	U	U	U	U	U	U	U	U	ESH	E
9	W	WSH	U	U	U	U	U	U	U	U	U	ESH	E
10	W	WSH	U	U	U	U	U	U	U	U	U	ESH	E
11	W	WSH	U	U	U	U	U	U	U	U	U	ESH	E
12	SWRW	SWRSRWSP	SSL	SSL	SSL	SSL	SSL	SSL	SSL	SSL	SSL	SERSRESP	SERE
13	SW	SWRS	S	S	S	S	S	S	S	S	S	SERS	SE

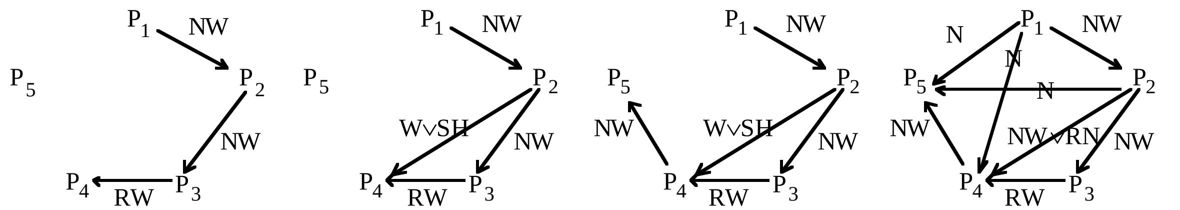
Figure 3 Direction relation between points implied by the relation of ancestor regions (rR)

Initially the relation between any pair of points is given by explicit retrieval. Inference through regions retrieves one by one all non-leaf regions A and gets the relation r: A r(A_k,A_l) for all pairs of regions IN A. Let R' be the relation implied by r: rR' (according to the rules of Figure 3). If R' U, the relation between all pairs of points P_i such that DBIN*(P_i,A_k), and P_j such that DBIN*(P_j,A_l) is updated to R(P_i,P_j)=R(P_i,P_j)R'.

```
Inference_through_regions
for each non-leaf region A
  retrieve A;
  for each region A_k such that DBIN(A_k,A)
     for each region A_l such that DBIN(A_l,A) and k<l
       get the relation r : A  r(A_k,A_l);
       lookup R': r  R';
       if R' U then
           for each point P_i such that DBIN*(P_i,A_k)
             for each point P_j such that DBIN*(P_j,A_l)
              R(P_i,P_j)=R(P_i,P_j)R' ;
              if R(P_i,P_j)= then return INCONSISTENCY DUE TO INFERENCE THROUGH REGIONS;
                            else R(P_j,P_i)=converse(R(P_i,P_j));
             end-for
           end-for
     end-for
  end-for
end-for
```

For demonstration of the algorithm we use the example of Figure 2. Figure 4 illustrates the explicit relations between all pairs of points in the form of a constraint network. First A_6 is retrieved and the relation between A_2 and A_3 is found to be $P_{3\text{-}2}$. Since $P_{3\text{-}2}$WSH, the relation between P_2 (that belongs to A_2) and P_4 (that belongs to A_3) is refined to U(WSH)= WSH (Figure 4b). The relation between P_3 and the other points of A_2 and A_3 remains unchanged because NW(WSH)=NW and RW(WSH)=RW (for (P_2,P_3) and (P_3,P_4) respectively). Then A_7 is retrieved and the relation NW between P_4 and P_5 is inferred because the ancestor regions of the two points (A_4 and A_5) are related by $P_{1\text{-}1}$ and, $P_{1\text{-}1}$ NW (Figure 4c). After the retrieval of A_8 (the last non-leaf region) the network takes its final form of Figure 4d. From A_8 $P_{1\text{-}3}$(A_1,A_7), and $P_{1\text{-}3}$N, the relation

North is inferred between all points of A_1 and the ones IN* A_7, resulting in $N(P_1,P_4)$, $N(P_1,P_5)$, $N(P_2,P_5)$ and $NW(P_2,P_4)$ $RN(P_2,P_4)$ (the last relation is obtained by N(WSH)). The relations $P_{3-11}(A_1,A_6)$ and $P_{9-11}(A_6,A_7)$ do not allow any inferences because P_{3-11} U and P_{9-11} U.



initial relations — after retrieval of A_6 — after retrieval of A_7 — after retrieval of A_8

Figure 4 Illustration of the algorithm

In order to measure the cost of inference through regions we need to calculate the number of non-leaf regions, because all these regions are retrieved. There is only one node at level 0, *k* nodes at level 1, and k^2 at level 2. Out of these k^2 nodes, k^2/m correspond to objects and the rest to copies. Level 3 contains *k* nodes for each original node of the previous level resulting in a total of k^3/m nodes out of which only k^3/m^2 are original and represented at level 4. Similarly, at level h-1 there are k^{h-1}/m^{h-2} nodes that correspond to actual leaf regions. Since the number of leaf regions is mN/k, we have the following equation that provides a formula for *h*:

$$\frac{k^{h-1}}{m^{h-2}} = \frac{m \cdot N}{k} \Rightarrow h = \left\lceil \log_{(k/m)}\left(\frac{N}{m}\right) \right\rceil \tag{1}$$

The number of non-leaf regions (and therefore the number of map retrievals during inference through regions) is the sum of original regions from level 0 to level h-2. Substituting the height of equation 1 we get the following approximation for the cost of inference though regions:

$$1 + \sum_{i=1}^{h-2} \frac{k^i}{m^{i-1}} \cong m^2 \, \frac{N-k}{k(k-m)} \tag{2}$$

Since the algorithm generates the permitted relations for all pairs of points, it needs to be performed only once and its results can be stored for future use. The above algorithm produces fast and high resolution relations in many situations. However, in cases where we have overlapping projections with multiple common points (as in Figure 2) further refinements are possible by using the common points.

5. SPATIAL INFERENCE THROUGH POINTS

According to this form of inference the relation between P_i and P_j that belong to different maps is derived through a chain of common points by composition[3] of spatial relations: [∃P (*DB* $R_k(P_i,P)$ *DB* $R_l(P, P_j)$) | ($R_k * R_l = R$)] *DB* $R(P_i,P_j)$. Inconsistencies in this case arise when different relations are inferred by different chains of points, or when the inferred relation contradicts the

[3]The problem of composition can be defined as "if the spatial relation between P_i and P, and between P and P_j is known, what are the possible relations between P_i and P_j?". The symbol * denotes *path composition* (Frank, 1992): $R_1 * R_2 =$

results of explicit retrieval or inference through regions (e.g., A_1 NW(P_1,P), A_2 NW(P,P_2) and A_3 RS(P_1,P_2)).

Inference using common points, can be formulated as a path consistency problem in a network of binary direction constraints. Each constraint in the network is a disjunction of primitive relations and represents the permitted relations between a pair of points after explicit retrieval and inference through regions have taken place (e.g., Figure 4d). Path consistency uses the relative positions of common points to derive the relation between any two points as they are implied by the given constraints. Inference is achieved by excluding relations that cause inconsistencies and maintaining only the ones that could participate in a solution of the network.

In order to apply some path consistency algorithm we need a set of composition rules for direction relations. Figure 5 describes the rules that are applied in order to produce the possible direction relations between P_i and P_j when their relation with respect to a third point P is known. Frank (in press) describes composition of direction relations based on the concepts of *projections* and *cone-shaped directions*. Unlike Frank who uses the notion of *Euclidean approximate* to deal with uncertainty, our system generates a disjunction of the potential primitive relations (which are expressed by the low resolution relations).

	NW(P,P_j)	**RN(P,P_j)**	**NE(P,P_j)**	**RW(P,P_j)**	**SP(P,P_j)**	**RE(P,P_j)**	**SW(P,P_j)**	**RS(P,P_j)**	**SE(P,P_j)**	**U(P,P_j)**
NW(P_i,P)	NW	NW	N	NW	NW	N	W	W	U	U
RN(P_i,P)	NW	RN	NE	NW	RN	NE	W	SH	E	U
NE(P_i,P)	N	NE	NE	N	NE	NE	U	E	E	U
RW(P_i,P)	NW	NW	N	RW	RW	SL	SW	SW	S	U
SP(P_i,P)	NW	RN	NE	RW	SP	RE	SW	RS	SE	U
RE(P_i,P)	N	NE	NE	SL	RE	RE	S	SE	SE	U
SW(P_i,P)	W	W	U	SW	SW	S	SW	SW	S	U
RS(P_i,P)	W	SH	E	SW	RS	SE	SW	RS	SE	U
SE(P_i,P)	U	E	E	S	SE	SE	S	SE	SE	U
U(P_i,P)	U	U	U	U	U	U	U	U	U	U

Figure 5 Composition table for low and high resolution relations

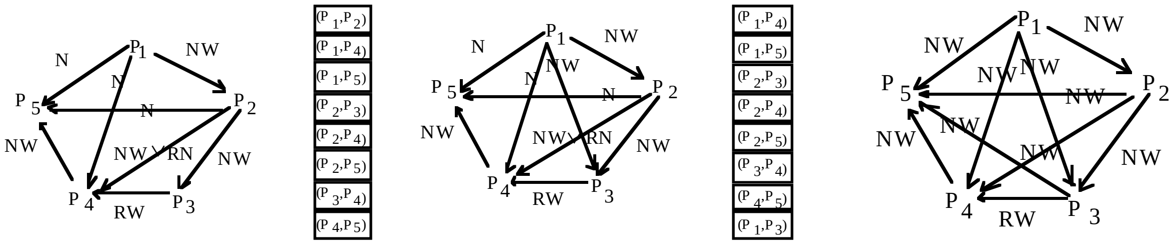
The composition constraint $R_k * R_l$ is computed by forming the cross products of the primitive relations that comprise R_k and R_l, composing each resulting ordered pair by looking up the results in the composition table, and taking the union of the resulting sets. A very important point is the type of constraints that appear in the network. If any disjunction of primitive relations is allowable, then the detection of all inconsistencies is NP-Complete even for point networks (Van Beek and Cohen,1990), and path consistency (which is polynomial) does not suffice. However, in the problem that we study here, we start with a set of 33 relations (U, , 9 primitive, 6 low resolution and 16 relations of the form NWRN that may appear during inference through regions - Figure 3) which is closed under composition and intersection (see Sharma 1996). Path consistency suffices for this case and exponential algorithms are not needed to enforce satisfiability (for more details see Papadias and Egenhofer, 1996).

A number of path consistency algorithms have been proposed (Allen, 1983; Macworth and Freuder, 1985). The following one is a variation modified for the current problem. All pairs of points whose relation is not U are inserted into a queue. Then every pair is popped from the queue and the corresponding relation is used to refine the relation between the popped points and all the other points that co-exist with them in some region. The pairs of points whose relation is refined are pushed in the queue for propagation of the update through the network.

```
Inference_through_points

for each point P_i
       for each point P_j such that i<j
                     if R(P_i,P_j)U then push-queue(P_i,P_j);
while not-empty-queue
    pop-queue(P_i,P_j);
       for each (leaf) region A_l such that DB IN(P_i,A_l)
              retrieve A_l;
              for each point P_k such that DB IN(P_k,A_l) and k  i and k  j
                     R_t(P_k,P_j)=R(P_k,P_j)  (R(P_k,P_i)*R(P_i,P_j));
                     if R_t= then return INCONSISTENCY DUE TO PATH CONSISTENCY;
                     else if R_t(P_k,P_j) R(P_k,P_j) then
                                   R(P_k,P_j)= R_t(P_k,P_j);
                                   R(P_j,P_k)=converse(R(P_k,P_j));
                                   if not in-queue(P_k,P_j) then push-queue(P_k,P_j);
              end-for
       end-for
       for each (leaf) region A_m such that DB IN(P_j,A_m)
              retrieve A_m;
              for each point P_k such that DB IN(P_k,A_m) and k  i and k  j
                     R_t(P_i,P_k)=R(P_i,P_k)  (R(P_i,P_j)* R(P_j,P_k));
                     if R_t= then return INCONSISTENCY DUE TO PATH CONSISTENCY;
                     else if R_t(P_i,P_k) R(P_i,P_k) then
                                   R(P_i,P_k)= R_t(P_i,P_k);
                                   R(P_k,P_i)=converse(R(P_i,P_k));
                                   if not in-queue(P_i,P_k) then push-queue(P_i,P_k);
              end-for
       end-for
end-while
```

For demonstration of the algorithm, we use the configuration of Figure 2 and the network of Figure 4d. After explicit retrieval and inference through regions have been applied, the pairs of points whose relation is not U are pushed into a queue. Here we assume the order of Figure 6a, but the order is not important. First the pair (P_1,P_2) is popped and all the regions that contain these points are retrieved. P_3 co-exists with P_2 in region A_2 and its relation with P_1 is updated according to: $R(P_1,P_3) = R(P_1,P_3)\ \ (R(P_1,P_2)*R(P_2,P_3)) = U(NW*NW) = NW$. Because the new relation is a refinement of the previous one (NWU) the pair (P_1,P_3) is pushed into the queue for propagation. The new network and the state of the queue at this phase are illustrated in Figure 6b. Then the pair (P_1,P_4) is popped from the queue, the regions A_1, A_3, and A_4 are retrieved, and the relations between the points (P_2,P_4), and (P_1,P_3) are updated. However the network does not change at this stage because: $R(P_2,P_4) = R(P_2,P_4)\ \ (R(P_2,P_1)*R(P_1,P_4)) = (NWRN)\ \ (SE*N) = NWRN$, and $R(P_1,P_3) = R(P_1,P_3)\ \ (R(P_1,P_4)*R(P_4,P_3)) = NW(N*RE)=NW$. Similarly the pair (P_1,P_5) will not alter the network, while the pair (P_2,P_3) will produce: $R(P_2,P_4) =NW$. The remaining pairs update the network in the same fashion; the final state after the termination of the algorithm is illustrated in Figure 6c.



initial network and queue after (P_1,P_2) has been popped final network

Figure 6 Illustration of the algorithm

In order to find the cost of inference through points we start with the observation that only 32 different constraints may appear in the network (and in which case the algorithm terminates with inconsistency). A constraint imposed by inference through regions or explicit retrieval may be refined a number of times until it reaches its final state at the end of path consistency. Figure 7 illustrates the possible refinements for the 32 constraints. The links connect each constraint with the constraints of the immediately lower level that it can be refined. The maximum number of refinements for any constraint is four. For example, a constraint between two points may initially be U and become NSL, then NWRNRWSP, then NWRN and finally NW.

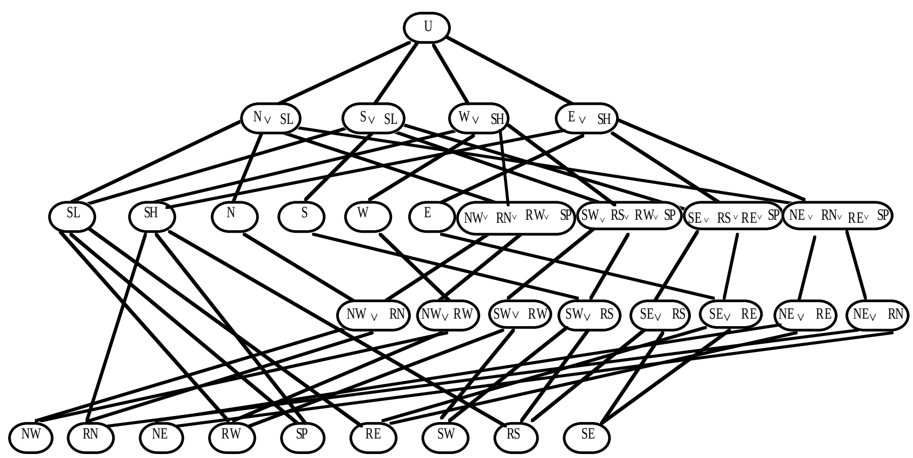


Figure 7 Refinements of direction constraints

There exist *N(N-1)/2* distinct pairs of points in the database and each may be pushed into the queue a maximum of four times. Each time a pair is popped from the queue, *2m* map retrievals are performed to retrieve the points that are related with the popped points in some region. Therefore, inference through points requires *4mN(N-1)* map retrievals in the worst case, which makes it significantly more expensive than inference through regions.

6. DISCUSSION

In the previous sections we argued that first explicit retrieval obtains the relations between pairs of points that exist in the same region, then inference through regions generates additional constraints imposed by the relations between the ancestor regions, and finally inference through points takes advantage of common points to produce further refinements. The order in which explicit retrieval and inference though regions are performed is not important. As long as the content of the database remains unaltered they will generate the same result independently on which is performed first. On the other hand, inference through points has always to be performed at the end, otherwise it may not produce all relations.

Although we have dealt with a set of projection-based direction relations often found in the literature, the methods of the paper are not relation-specific but can be applied to higher dimensions and other types of directions (for an example see Papadias and Egenhofer, 1996). In general, what is needed for the application of the algorithms is a] a set of direction relations for points and one for regions, b] rules for the inference of the relation between points given the relation between ancestor regions and finally c] composition rules. Notice, however, that depending on the choice of the relation set, inference through points may become exponential.

Hierarchical inference mechanisms are necessary to complement other qualitative spatial reasoning methods (Sharma et al., 1994; Grigni et al., 1995). Even in a single system, data about the same or overlapping areas but from different sources are stored separately. This information may be incomplete or inconsistent, and inference mechanisms are required to explicate relations and remove inconsistencies. As interoperability issues are solved, heterogeneous spatial databases and open GIS will soon become a reality. Such systems will store huge amounts of spatial data in various formats and of variable quality. Users will query the systems requiring fast and accurate results (and not answers of the form "A is north and south of B"). Spatial inference mechanisms will play an important role for the detection of inconsistencies in the data and the integration of the different systems.

ACKNOWLEDGEMENTS

This work was partially supported by the National Science Foundation under grant number No. SES 88-10917 for the NCGIA. Dimitris Papadias is currently supported by an ERCIM fellowship. Max Egenhofer's work is further supported by the National Science Foundation under grant number IRI-9309230, Rome Laboratories under grant number F30602-95-1-0042, the Scientific and Environmental Division of the North Atlantic Treaty Organization, and by a Massive Digital Data Systems contract sponsored by the Advanced Research and Development Committee of the Community Management Staff and administered by the Office of Research and Development.

REFERENCES

Allen, J.F. “Maintaining Knowledge about Temporal Intervals”. CACM 26 (11), pp. 832-843, 1983.

Car, A., Frank, A. “General Principles of Hierarchical Spatial Reasoning: The Case of Wayfinding”. In the Proceedings of the 6th International Symposium on Spatial Data Handling, 1994.

Chang, S.K., Shi, Q.Y. Yan C.W. “Iconic Indexing by 2-D String”. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, no 3, pp. 413-428, 1987.

Egenhofer, M., Franzosa R. (1991) “Point Set Topological Relations”. International Journal of Geographic Information Systems, 5, pp. 161-174, 1991.

Frank, A.U., “Qualitative Spatial Reasoning about Distances and Directions in Geographic Space”, Journal of Visual Languages and Computing, Vol. 3, pp. 343-371, 1992.

Frank, A. U. “Qualitative Spatial Reasoning: Cardinal Directions as an Example”. To appear in the International Journal of Geographic Information Systems.

Freksa, C. “Using Orientation Information for Qualitative Spatial Reasoning”. Proceedings of the International Conference GIS - From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, Pisa, Italy. Springer Verlag LNCS.

Glasgow, J.I., Papadias, D. “Computational Imagery”. Cognitive Science, 16, pp. 355-394, 1992.

Grigni, M., Papadias, D., Papadimitriou, C., “Topological Inference”, Proceedings of the International Joint Conference of Artificial Intelligence, 1995.

Guttman, A., “R-trees: a Dynamic Index Structure for Spatial Searching”, Proceedings of ACM SIGMOD International Conference on Management of Data, 1984.

Hernandez, D., “Qualitative Representation of Spatial Knowledge”, Springer Verlag LNAI, 1994.

Herskovits, A. “Language and Spatial Cognition”. Cambridge University Press, 1986.

Hirtle, S, Jonides, J. “Evidence of Hierarchies in Cognitive Maps”. Memory and Cognition 13, pp. 208-217, 1985.

Holmes, P.D., Jungert, E. “Symbolic and Geometric Connectivity Graph Methods for Route Planning In Digitized Maps”. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-14, no 5, pp. 549-565, 1993

Lee, S-Y., Yang, M-C., Chen, J-W. “Signature File as a Spatial Filter for Iconic Image Database”. Journal of Visual Languages and Computing, 3, pp. 373-397, 1992.

Mackworth, A, Freuder, E. “The Complexity of some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problems”. Artificial Intelligence, 25, pp. 65-74, 1985.

Papadias, D., Sellis, T., “Qualitative Representation of Spatial Knowledge in Two-Dimensional Space”, Very Large Data Bases Journal, Special Issue on Spatial Databases, Vol 3(4), pp. 479-516, 1994.

Papadias, D., Sellis, T. “A Pictorial Query-By-Example Language”. Journal of Visual Languages and Computing, Special Issue on Visual Query Systems, Vol. 6, No. 1, 53 -72, 1995.

Papadias, D., Theodoridis, Y. “Spatial Relations, Minimum Bounding Rectangles, and Spatial Data Structures”, To appear in the International Journal of Geographic Information Systems.

Papadias, D., Egenhofer, M., “Algorithms for Hierarchical Spatial Reasoning”. NCGIA TR-96-2 University of California, Santa Barbara, 1996. Available via ftp from: ftp://ncgia.ucsb.edu/pub/Publications/tech_reports/96/96-2/96-2.ps

Roussopoulos, N., Faloutsos,C., Sellis, T. “An Efficient Pictorial Database System for PSQL”. IEEE Transactions on Software Engineering, Special issue on Image Databases, pp. 639-650, 1988.

Sharma, J., Flewelling, D., Egenhofer, M.I., “A Qualitative Spatial Reasoner”. In the Proceedings of the 6th International Symposium on Spatial Data Handling, 1994.

Sharma, J., “Heterogeneous Spatial Reasoning in Geographic Information Systems”. Ph.D. Thesis, Department of Spatial Information Science and Engineering, University of Maine, 1996.

van Beek, P., Cohen, R. “Exact and Approximate Reasoning about Temporal Relations”. Computational Intelligence 6, 132-144, 1990.