Consistency Checking for Qualitative Spatial Reasoning with Cardinal Directions

Spiros Skiadopoulos[1] and Manolis Koubarakis[2]

[1] Dept. of Electrical and Computer Engineering
National Technical University of Athens
Zographou 157 73 Athens, Greece
spiros@dblab.ntua.gr, www.dblab.ntua.gr/~spiros

[2] Dept. of Electronic and Computer Engineering
Technical University of Crete
Chania 73100 Crete, Greece
manolis@intelligence.tuc.gr, www.intelligence.tuc.gr/~manolis

Abstract. We present a formal model for qualitative spatial reasoning with cardinal directions and study the problem of checking the consistency of a set of cardinal direction constraints. We present the first algorithm for this problem, prove its correctness and analyze its computational complexity. Utilizing the above algorithm we prove that the consistency checking of a set of basic cardinal direction constraints can be performed in $\mathcal{O}(n^5)$ time while the consistency checking of an unrestricted set of cardinal direction constraints is NP-complete. Finally, we briefly discuss some extensions to the basic model.

1 Introduction

Qualitative spatial reasoning has received a lot of attention in the areas of Geographic Information Systems [6–8], Artificial Intelligence [4, 6, 13, 17], Databases [14] and Multimedia [19]. Qualitative spatial reasoning problems have recently been posed as constraint satisfaction problems and solved using traditional algorithms e.g., path-consistency [18]. One of the most important problems in this area is the identification of useful and tractable classes of spatial constraints and the study of efficient algorithms for consistency checking, minimal network computation and so on [18]. Several kinds of useful spatial constraints have been studied so far e.g., topological constraints [6, 17], cardinal direction constraints [11, 9] and qualitative distance constraints [7].

In this paper we concentrate on *cardinal direction constraints* [9, 11, 14, 20]. Cardinal direction constraints describe how regions of space are placed relative to one another. Typically, direction constraints are of the form $a\ R\ b$, where a is the primary region, b is the reference region and R is a direction relation (e.g., *a north of b*). Earlier qualitative models for cardinal direction relations approximate a spatial region by a representative point (most commonly the centroid) or by a representative box (most commonly the minimum bounding box) [7, 8, 11, 13, 14]. Depending on the particular spatial configuration these approximations may be too crude [9]. Thus, expressing direction relations on these approximations can be misleading and contradictory. For instance, with respect to the point-based approximation Spain is northeast of Portugal, but most people would agree that "northeast" does not describe accurately the relation between Spain and Portugal on a map.

Bearing that in mind Goyal and Egenhofer [9], and Skiadopoulos and Koubarakis [20] presented a model that only approximates the reference region (using the minimum bounding box) while using the exact shape of the primary region. Relations in the above model are clearly more expressive than point and box-based models. The model that we will present in this paper is closely related to the model of [9, 20] but there is a significant difference. [9, 20] deal only with extended regions that are connected and with connected boundaries while our approach allows regions to be disconnected and have holes. The regions that we consider are very common in Geography, Multimedia and Image Databases [1, 3, 19]. For example, countries are made up of separations (islands, exclaves, external territories) and holes (enclaves) [3].

We will study the problem of checking the consistency of a given set of cardinal direction constraints in our model. This is a fundamental problem of Logic Programming, Artificial Intelligence and Spatial Reasoning and has received a lot of attention in the literature [11, 14, 18]. Algorithms for consistency checking, are of immediate use in various situations including:

– Propagating relations and detecting inconsistencies in a given set of spatial relations [11, 18].
– Preprocessing spatial queries so that inconsistent queries are detected or the search space is pruned [15].

The technical contributions of this paper can be summarized as follows:

1. We present a formal model for qualitative reasoning about cardinal directions. This model is related to the model of [9, 20] and is currently one of the most expressive models for qualitative reasoning with cardinal directions.
2. We use our formal framework to study the problem of checking the consistency of a given set of cardinal direction constraints in the proposed model. We present the first algorithm for this problem and prove its correctness.
3. We present an analysis of the computational complexity of consistency checking. We show that the consistency checking of a set of *basic* cardinal direction constraints in n variables can be performed in $\mathcal{O}(n^5)$. Moreover, we prove that checking the consistency of an unrestricted set of cardinal direction constraints is NP-complete.
4. Finally, we consider the consistency checking of a set of cardinal direction constraints expressed in some interesting extension of the basic model and outline algorithms for this task.

The rest of the paper is organized as follows. Section 2 presents our model. In Section 3 we discuss the problem of consistency checking, present the main

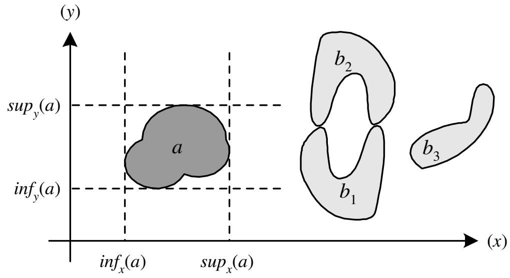

Fig. 1. Regions

algorithm and prove its correctness. Section 4 discusses the computational complexity of consistency checking. Finally, Section 5 concludes the paper by summarizing some interesting extensions of the basic model that we have already completed.

2 A Formal Model for Cardinal Direction Information

We consider the Euclidean space $\Re^2$. *Regions* are defined as non-empty and bounded sets of points in $\Re^2$. Let a be a region. The *greatest lower bound* or the *infimum* [12] of the *projection* of region a on the x-axis (respectively y-axis) is denoted by $inf_x(a)$ (respectively $inf_y(a)$). The *least upper bound* or the *supremum* of the *projection* of region a on the x-axis (respectively y-axis) is denoted by $sup_x(a)$ (respectively $sup_y(a)$). We will often refer to sup and inf as *endpoints*.

Let a be a region. We say that a is a *box* iff a is a rectangular region formed by the straight lines $x = c_1$, $x = c_2$, $y = c_3$ and $y = c_4$ where c_1, c_2, c_3 and c_4 are real constants such that $c_1 \leq c_2$ and $c_3 \leq c_4$. Moreover, iff $c_1 < c_2$ and $c_3 < c_4$ hold, we say that a is a *non-trivial box*. A box is *trivial* if it is a point or a vertical line segment or a horizontal line segment.

The *minimum bounding box* of a region a, denoted by $mbb(a)$, is the box formed by the straight lines $x = inf_x(a)$, $x = sup_x(a)$, $y = inf_y(a)$ and $y = sup_y(a)$ (see Fig. 1). Obviously, the projections on the x-axis (respectively y-axis) of a region and its minimum bounding box have the same endpoints.

We will consider throughout the paper the following types of regions:

– Regions that are homeomorphic to the *closed unit disk* ($\{(x, y) : x^2 + y^2 \leq 1\}$). The set of these regions will be denoted by REG. Regions in REG are *closed*, *connected* and have *connected boundaries* (for definitions see [2, 12]). Class REG excludes disconnected regions, regions with holes, points, lines and regions with emanating lines. Connected regions have been previously studied in [9, 16] and form the first approach to model areas in Geographic Information Systems [7]. Notice that our results are not affected if we consider regions that are homeomorphic to the *open unit disk* (as in [16]).

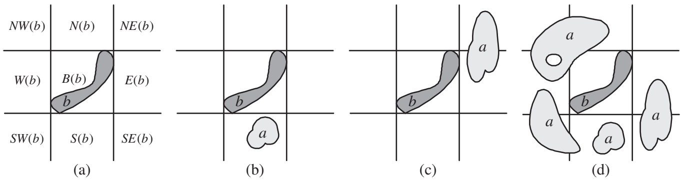

Fig. 2. Reference tiles and relations

– Regions in REG cannot model the variety and complexity of geographic entities [3]. Thus, we extend class REG in order to accommodate *disconnected regions* and *regions with holes*. The set of these regions will be denoted by REG^*. Set REG^* is a natural extension of REG which is useful to model (possibly disconnected) land parcels and countries in Geographic Information Systems [3, 7] or areas of an image containing similar chromatic arrangements [1].

In Fig. 1, regions a, b_1, b_2 and b_3 are in REG (also in REG^*) and region $b = b_1 \cup b_2 \cup b_3$ is in REG^*. Notice that region b is disconnected and has a hole.

Proposition 1. *If $a \in REG^*$ then $mbb(a)$ is a non-trivial box. Equivalently, the following inequalities hold:*

$$inf_x(a) < sup_x(a) \text{ and } inf_y(a) < sup_y(a).$$

Let us now consider two arbitrary regions a and b in REG^*. Let region a be related to region b through a cardinal direction relation (e.g., a is north of b). Region b will be called the *reference* region (i.e., the region *to* which the relation is described) while region a will be called the *primary* region (i.e., the region *from* which the relation is described). The axes forming the minimum bounding box of the reference region b divide the space into 9 tiles (Fig. 2a). The peripheral tiles correspond to the eight cardinal direction relations *south*, *southwest*, *west*, *northwest*, *north*, *northeast*, *east* and *southeast*. These tiles will be denoted by $S(b)$, $SW(b)$, $W(b)$, $NW(b)$, $N(b)$, $NE(b)$, $E(b)$ and $SE(b)$ respectively. The central area corresponds to the region's minimum bounding box and is denoted by $B(b)$. By definition each one of these tiles includes the parts of the axes forming it. The union of all 9 tiles is $\Re^2$.

If a primary region a is included (in the set-theoretic sense) in tile $S(b)$ of some reference region b (Fig. 2b) then we say that *a is south of b* and we write $a\ S\ b$. Similarly, we can define *southwest* (SW), *west* (W), *northwest* (NW), *north* (N), *northeast* (NE), *east* (E), *southeast* (SE) and *bounding box* (B) relations.

If a primary region a lies partly in the area $NE(b)$ and partly in the area $E(b)$ of some reference region b (Fig. 2c) then we say that *a is partly northeast and partly east of b* and we write $a\ NE{:}E\ b$.

The general definition of a cardinal direction relation in our framework is as follows.

Definition 1. *An* atomic cardinal direction relation *is an element of the set* $\{B, S, SW, W, NW, N, NE, E, SE\}$. *A* basic cardinal direction relation *is an atomic cardinal direction relation or an expression* $R_1{:}\cdots{:}R_k$ *where* $2 \leq k \leq 9$, $R_1, \ldots, R_k \in \{B, S, SW, W, NW, N, NE, E, SE\}$ *and* $R_i \neq R_j$ *for every* i, j *such that* $1 \leq i, j \leq k$ *and* $i \neq j$.

Example 1. The following are basic cardinal direction relations:

$$S, \quad NE{:}E \quad \text{and} \quad B{:}S{:}SW{:}W{:}NW{:}N{:}E{:}SE.$$

Regions involved in these relations are shown in Figures 2b, 2c and 2d respectively.

In order to avoid confusion we will write the atomic elements of a cardinal direction relation according to the following order: B, S, SW, W, NW, N, NE, E and SE. Thus, we always write $B{:}S{:}W$ instead of $W{:}B{:}S$ or $S{:}B{:}W$. The readers should also be aware that for a basic relation such as $B{:}S{:}W$ we will often refer to B, S and W as its *tiles*.

2.1 Defining Basic Cardinal Direction Relations Formally

Now we can formally define the atomic cardinal direction relations B, S, SW, W, NW, N, NE, E and SE of our model as follows:

$a\ B\ b$	iff	$inf_x(b) \leq inf_x(a)$, $sup_x(a) \leq sup_x(b)$, $inf_y(b) \leq inf_y(a)$ and $sup_y(a) \leq sup_y(b)$.
$a\ S\ b$	iff	$sup_y(a) \leq inf_y(b)$, $inf_x(b) \leq inf_x(a)$ and $sup_x(a) \leq sup_x(b)$.
$a\ SW\ b$	iff	$sup_x(a) \leq inf_x(b)$ and $sup_y(a) \leq inf_y(b)$.
$a\ W\ b$	iff	$sup_x(a) \leq inf_x(b)$, $inf_y(b) \leq inf_y(a)$ and $sup_y(a) \leq sup_y(b)$.
$a\ NW\ b$	iff	$sup_x(a) \leq inf_x(b)$ and $sup_y(b) \leq inf_y(a)$.
$a\ N\ b$	iff	$sup_y(b) \leq inf_y(a)$, $inf_x(b) \leq inf_x(a)$ and $sup_x(a) \leq sup_x(b)$.
$a\ NE\ b$	iff	$sup_x(b) \leq inf_x(a)$ and $sup_y(b) \leq inf_y(a)$.
$a\ E\ b$	iff	$sup_x(b) \leq inf_x(a)$, $inf_y(b) \leq inf_y(a)$ and $sup_y(a) \leq sup_y(b)$.
$a\ SE\ b$	iff	$sup_x(b) \leq inf_x(a)$ and $sup_y(a) \leq inf_y(b)$.

Using the above atomic relations we can define all non-atomic ones. For instance relation $NE{:}E$ (Fig. 3a) and relation $B{:}S{:}SW{:}W{:}NW{:}N{:}E{:}SE$ (Fig. 3b) are defined as follows:

$a\ NE{:}E\ b$ iff there exist regions a_1 and a_2 in REG^* such that $a = a_1 \cup a_2$, $a_1\ NE\ b$ and $a_2\ E\ b$.

$a\ B{:}S{:}SW{:}W{:}NW{:}N{:}SE{:}E\ b$ iff there exist regions $a_1, \ldots, a_8$ in REG^* such that $a = a_1 \cup \cdots \cup a_8$, $a_1\ B\ b$, $a_2\ S\ b$, $a_3\ SW\ b$, $a_4\ W\ b$, $a_5\ NW\ b$, $a_6\ N\ b$, $a_7\ SE\ b$ and $a_8\ E\ b$.

In general each non-atomic cardinal direction relation is defined as follows.

Definition 2. *If* $2 \leq k \leq 9$ *then* $a\ R_1{:}\cdots{:}R_k\ b$ *iff there exist regions* $a_1, \ldots, a_k \in REG^*$ *such that* $a_1\ R_1\ b, \ldots, a_k\ R_k\ b$ *and* $a = a_1 \cup \cdots \cup a_k$.

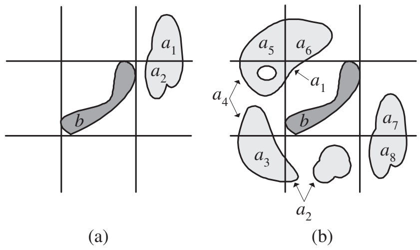

Fig. 3. Relations and component variables

The variables $a_1, \ldots, a_k$ in any equivalence such as the above (which defines a basic cardinal direction relation) will be called the *component variables* corresponding to variable a. Notice that for every i, j such that $1 \leq i, j \leq k$ and $i \neq j$, a_i and a_j have disjoint interiors but may share points in their boundaries.

The set of basic cardinal direction relations in our model contains $\sum_{i=1}^{9} \binom{9}{i} = 511$ elements. We will use $\mathcal{D}^*$ to denote this set. Relations in $\mathcal{D}^*$ are jointly exclusive and pairwise disjoint. Elements of $\mathcal{D}^*$ can be used to represent *definite information* about cardinal directions, e.g., $a\ N\ b$.

Using the 511 relations of $\mathcal{D}^*$ as our basis, we can define the *powerset* $2^{\mathcal{D}^*}$ of $\mathcal{D}^*$ which contains 2^{511} relations. Elements of $2^{\mathcal{D}^*}$ are called *cardinal direction relations* and can be used to represent not only definite but also *indefinite information* about cardinal directions e.g., $a\ \{N,\ W\}\ b$ denotes that region a is north *or* west of region b.

The model presented in this section is related to the model of [9, 20] but these proposals deal only with the connected regions of REG. Our approach accommodates a wider set of region (i.e., regions in REG^*) that also allows regions to be disconnected and have holes. As a result we have 511 relations while the model of [9, 20] has 218. This enables us to express several natural spatial arrangements (e.g., $a\ S{:}W\ b$ or $a\ S{:}N\ b$) that are not possible in [9, 20].

3 Constraints and Consistency Checking

In this section we present an algorithm for deciding the consistency of a set of cardinal direction constraints involving only basic relations. Let us first define cardinal direction constraints formally.

Definition 3. *A* cardinal direction constraint *is a formula $a\ R\ b$ where a, b are variables ranging over regions in REG^* and R is a cardinal direction relation from the set $2^{\mathcal{D}^*}$. Moreover, a cardinal direction constraint is called* atomic *(respectively* basic*) if R is an atomic (respectively basic) cardinal direction relation.*

Example 2. The following are cardinal direction constraints:

$$a_1\ S\ b_1, \quad a_2\ NE{:}E\ b_2 \ \text{ and } \ a_3\ \{B, S{:}SW\}\ b_3.$$

The constraint $a_1\ S\ b_1$ is an atomic one (also basic). The constraint $a_2\ NE{:}E\ b_2$ is a basic one.

Definition 4. *Let* C *be a set of cardinal direction constraints in variables* $a_1, \ldots, a_n$. *The* solution set *of* C, *denoted by* $Sol(C)$, *is:*

$$\{(o_1, \ldots, o_n) : o_1, \ldots, o_n \in REG^* \text{ and for every } c \in C,\ (o_1, \ldots, o_n) \text{ satisfies } c\}.$$

Each member of $Sol(C)$ *is called a* solution *of* C. *A set of cardinal direction constraints is called* consistent *iff its solution set is non-empty.*

In this paper we will also be interested in special kinds of order constraints which are defined below.

Definition 5. *An* order constraint *is a formula in any of the following forms:*

$$\begin{array}{lll} inf_x(a) \sim inf_x(b), & sup_x(a) \sim sup_x(b), & inf_x(a) \sim sup_x(b), \\ inf_y(a) \sim inf_y(b), & sup_y(a) \sim sup_y(b), & inf_y(a) \sim sup_y(b) \end{array}$$

where a *and* b *are variables ranging over regions in* REG^* *and* $\sim$ *can be any operator from the set* $\{<, >, \leq, \geq, =\}$.

Definition 6. *A set of order constraints is called* canonical *iff it includes the constraints* $inf_x(a) < sup_x(a)$ *and* $inf_y(a) < sup_y(a)$ *for every region variable* a *referenced in the set.*

Definition 7. *Let* O *be a canonical set of order constraints in region variables* $a_1, \ldots, a_n$. *The* solution set *of* O, *denoted by* $Sol(O)$, *is the set of* $4n$*-tuples* $(a_1^{xl}, a_1^{xu}, a_1^{yl}, a_1^{yu}, \ldots, a_n^{xl}, a_n^{xu}, a_n^{yl}, a_n^{yu}) \in \Re^{4n}$ *such that the constraints in* C *are satisfied by assigning*

$$a_i^{xl} \text{ to } inf_x(a_i), \quad a_i^{xu} \text{ to } sup_x(a_i), \quad a_i^{yl} \text{ to } inf_y(a_i), \quad a_i^{yu} \text{ to } sup_y(a_i)$$

for every i, $1 \leq i \leq n$.

According to the above definition when we talk of a solution for a canonical set of order constraints, it is the expressions $sup(\cdot)$ and $inf(\cdot)$ that should interest us and not the region variables of the set. As with cardinal direction constraints, a set of order constraints is called *consistent* if its solution set is non-empty.

3.1 Checking the consistency of a set of basic cardinal direction constraints

The "iff" statements of Definition 2 can be used to equivalently map a set of arbitrary basic cardinal direction constraints C to a set $\mathcal{S}$ consisting of the following two types of constraints:

- *atomic* cardinal direction constraints.
- *set-union* constraints.

Example 3. Let C be a set of basic cardinal direction constraints in variables $a_1, \ldots, a_n$. Let us consider an arbitrary constraint $a_i\ R_1{:}\cdots{:}R_k\ a_j$ in C. For this constraint, the mapping of Definition 2 will introduce to $\mathcal{S}$:

– k atomic cardinal direction constraints, i.e., $d_1\ R_1\ a_j,\ \ldots,\ d_k\ R_k\ a_j$ and
– one set-union constraint, i.e, $a_1 = d_1 \cup \cdots \cup d_k$.

where $d_1, \ldots, d_k$ are component variables corresponding to region variable a_i.

If we had an algorithm for deciding the consistency of sets like $\mathcal{S}$, we could use it to solve our problem. Given the unavailability of such an algorithm, we develop Algorithm Consistency that checks whether C (equivalently $\mathcal{S}$) is consistent. This algorithm proceeds as follows:

1. Initially, Algorithm Consistency considers each constraint c in C and maps the atomic cardinal direction constraints of its definition into a set of order constraints O. To this end, it uses the definitions of Section 2.1, Page 5. Moreover, the algorithm introduces into set O additional order constraints that are implied by the cardinal direction constraint c. These constraints will be discussed in the sequel.
2. Then Algorithm Consistency finds a solution of the set of order constraints O or if no solution exists it returns 'Inconsistent'. To this end we use the well-known algorithms of [22, 5]. This solution assigns regions to the regions variables of set C. Moreover, the second step of the algorithm extends the above regions without falsifying any constraint in C.
3. Finally, the algorithm considers the set-union constraints that correspond to every constraint c in C and checks whether they are satisfied by the solution of Step 2. If all set-union constraints are satisfied the algorithm returns 'Consistent'; otherwise it returns 'Inconsistent'.

Let us now describe the 3 steps of Algorithm Consistency. More details can be found in [21].

Step 1. In this first step, for each cardinal direction constraint $a_i\ R_1{:}\cdots{:}R_k\ a_j$ in C, we consult Definition 2. More specifically, this step introduces new region variables $d_1, \ldots, d_k$ which denote the component variables corresponding to variable a_i. Then, this step introduces order constraints encoding all atomic cardinal direction relations between the reference region a_j and the component variables corresponding to the primary region a_i, i.e., $d_1\ R_1\ a_j,\ \ldots,\ d_k\ R_k\ a_j$ (see Section 2.1, Page 5). We also introduce the obvious order constraints relating the endpoints of the projections of the component variables (Proposition 1). These constraints make the final set O canonical.

Example 4. Let $C = \{a_1\ S{:}SW\ a_2,\ a_1\ NW{:}N{:}NE\ a_3\}$ be the given set of basic cardinal direction constraints (see Fig. 5). Step 1.1 considers constraint $a_1\ S{:}SW\ a_2$ first and introduces component variables d_1 and d_2 representing subregions of a_1 such that $d_1\ SW\ a_2$ and $d_2\ S\ a_2$. Then, the definitions of relations SW and S are consulted (see Section 2.1, Page 5) and the following order constraints are added to O:

$$sup_x(d_1) \leq inf_x(a_2), \quad inf_x(a_2) \leq inf_x(d_2), \quad sup_x(d_2) \leq sup_x(a_2),$$
$$sup_y(d_1) \leq inf_y(a_2), \quad sup_y(d_2) \leq inf_y(a_2).$$

The obvious order constraints relating the endpoints of each projection are also added to O during this step (Proposition 1). Similarly, the order constraints corresponding to the second cardinal direction constraint of set C are introduced.

Step 1 also deals with the component variables corresponding to variable a_i. The fact that a variable r is a component variable of a_i implies that r is a subregion of a_i and thus, the constraints $inf_x(a_i) \leq inf_x(r)$, $sup_x(r) \leq sup_x(a_i)$, $inf_y(a_i) \leq inf_y(r)$ and $sup_y(r) \leq sup_y(a_i)$ hold. The above constraints are introduced by Step 1.2 for all component variables corresponding to variable a_i.

Example 5. Let us continue with the set C of Example 4. Since regions d_1 and d_2 are component variable corresponding to a_1, Step 1 adds to O the following constraints for every $r \in \{d_1, d_2\}$:

$$inf_x(a_1) \leq inf_x(r),\ sup_x(r) \leq sup_x(a_1), inf_y(a_1) \leq inf_y(r),\ sup_y(r) \leq sup_y(a_1)$$

The new order constraints corresponding to the second constraint of set C are omitted for brevity.

Finally, in Step 1 we re-examine each constraint $a_i\ R_1{:}\cdots{:}R_k\ a_j$ in C and introduce additional order constraints that establish the *strictest possible* relation between the endpoints of the projections of a_i and a_j on the x and y-axis. In this way, the strictest possible relation between the minimum bounding boxes of a_i and a_j is also established. Notice that these constraints are not implied by the other constraints of Step 1. There are 8 possible cases that need to be considered here. These cases correspond to checking whether the set of relations $\{R_1, \ldots, R_k\}$ is a subset of one or more of the following sets:

$$\{SW, S, SE\},\ \{SW, S, SE, W, B, E\},\ \{NW, W, SW\},\ \{NW, W, SW, N, B, S\},$$
$$\{NW, N, NE\},\ \{NW, N, NE, W, B, E\},\ \{NE, E, SE\},\ \{NE, E, SE, N, B, S\}.$$

For example, if the set of relations $\{R_1, \ldots, R_k\}$ is a subset of $\{SW, S, SE\}$, the constraint $sup_y(a_i) \leq inf_y(a_j)$ is introduced by this step (the full version of this paper gives the 7 constraints required by the remaining 7 cases).

Example 6. Let us continue with the set C of Example 4. If we consider the first constraint of C, we notice that relations S and SW are members of $\{SW, S, SE\}$ and $\{NW, W, SW, N, B, S\}$. As a result the constraints $sup_y(a_1) \leq inf_y(a_2)$ and $sup_x(a_1) \leq sup_x(a_2)$ are added to O (see Fig. 5). The new order constraints corresponding to the second constraint of set C are omitted for brevity.

Step 2. In the first step of Algorithm CONSISTENCY we have introduced to set O order constraints which are implied by the spatial configuration that is expressed by the cardinal direction constraints of a given set C. The second step of Algorithm CONSISTENCY uses the well-known algorithms for order constraints to find a solution of set O [22, 5]. If set O is inconsistent obviously set C is also inconsistent. If, on the other hand, a solution of set O exists then this solution assigns values to the sup's and inf's of all region and component variables

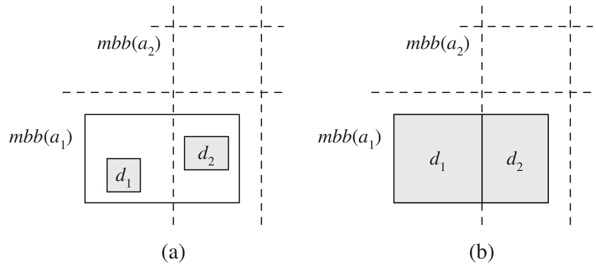

Fig. 4. Extending the solution of O

referenced in set C. Using these values we can draw non-trivial boxes that correspond to the above variables. Then, Step 2 of Algorithm CONSISTENCY extends all these non-trivial boxes as the following example suggests.

Example 7. Let us continue with the set C of Example 4. Using Step 1 we can map C into a set of order constraints O. Let us now assume that set O is consistent and $\bar{x}_0$ is a solution of O. Since O is consistent then $\bar{x}_0$ assigns values to $sup_x(r)$, $inf_x(r)$, $sup_y(r)$ and $inf_y(r)$, for every region variable $r \in \{a_1,\ a_2,\ a_3,\ d_1,\ d_2\}$. Using the above values, we can form non-trivial boxes that correspond to the minimum bounding boxes of regions a_1, a_2, a_3, d_1 and d_2 respectively.

We can now extend d_1 and d_2 in all directions until they touch whatever line is closer to them from the ones forming the minimum bounding boxes of a_1 and a_2 (see Fig. 4a). For instance, we can extend d_1 to the east to touch the vertical line $y = inf_x(mbb(a_2))$ and to the west to touch the vertical line $y = inf_x(mbb(a_1))$. Fig. 4b illustrates this idea for all regions d_1 and d_2. Now notice that the new regions d_1 and d_2 still satisfy all constraints in O.

The following lemma captures the above observation in its full generality.

Lemma 1. *Let O be the output of Step 1 of Algorithm* CONSISTENCY *when it is called with argument a set of cardinal direction constraints C in variables $a_1, \ldots, a_n$. For each variable a_i $(1 \leq i \leq n)$ let $\{d_1, \ldots, d_l\}$ be the set of component variables corresponding to a_i that have been generated by Step 1 of Algorithm* CONSISTENCY *while considering various constraints $a_i\ R_1{:}\cdots{:}R_k\ a_j$, $1 \leq j \leq n$, in C.*

Let $\bar{x}_0$ be a solution of O. Then a new solution $\bar{x}'_0$ of O can be constructed as follows. For every constraint $a_i\ R_1{:}\cdots{:}R_k\ a_j$ in C we consider each component variable d_m, $(1 \leq m \leq l)$ in turn.

If $d_m\ S\ a_j$ holds, perform the following substitutions:

$$\begin{array}{lll} inf_x(d_m) & \text{by} & max\{\ inf_x(a_i),\ inf_x(a_j)\ \}, \\ sup_x(d_m) & \text{by} & min\{\ sup_x(a_i),\ sup_x(a_j)\ \}, \\ inf_y(d_m) & \text{by} & inf_y(a_i), \\ sup_y(d_m) & \text{by} & min\{\ sup_y(a_i),\ inf_y(a_j)\ \}. \end{array}$$

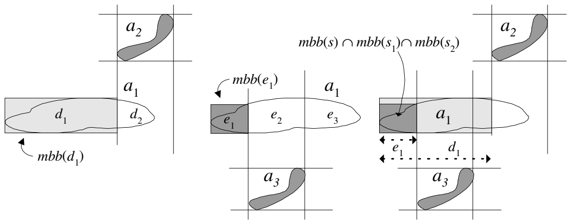

Fig. 5. Example of Constraint (T)

Similar substitutions are performed when d_m SW a_j, d_m W a_j, d_m NW a_j, d_m N a_j, d_m NE a_j, d_m E a_j, d_m SE a_j and d_m B a_j hold (the full version of this paper gives these substitutions [21]).

Example 8. Let us continue with the set C of Example 4 and consider region variable d_2 of Fig. 4a. It is easy to verify that if we perform the substitutions of Lemma 1 then d_2 can be extended as in Fig. 4b. For instance $inf_x(d_2)$ in Fig 4b is $inf_x(a_2) = max\{\ inf_x(a_1),\ inf_x(a_2)\ \}$.

Let $\bar{x}_0$ be a solution of O. Solution $\bar{x}_0$ is called *maximal* if it is not affected by the substitutions of Lemma 1. The substitutions of Lemma 1 are very useful since they allow us to check rather than enforce the set-union constraint of $\mathcal{S}$.

Step 3. The final step of Algorithm CONSISTENCY checks if the maximal solution of Step 2 satisfies the set-union constraints of $\mathcal{S}$. So far we have dealt only with order constraints, so it would be convenient if we could transform the set-union constraints into a comparable to order constraints from. To do so, we will need the following lemma.

Lemma 2. *Let a be a region in REG^*. Let us now suppose that $S_1 = \{a_1^1, \ldots, a_{k_1}^1\}$, $\ldots$, $S_m = \{a_1^m, \ldots, a_{k_m}^m\}$ are sets of subregions of a such that $a_1^1, \ldots, a_{k_1}^1$, $\ldots$, $a_1^m, \ldots, a_{k_m}^m \in REG^*$ and $a = a_1^1 \cup \cdots \cup a_{k_1}^1 = \cdots = a_1^m \cup \cdots \cup a_{k_m}^m$ holds. Then, the following constraint also holds.*

Constraint (T): *For all $s \in S_1 \cup \cdots \cup S_m$ there exists a tuple $(s_1, \ldots, s_m) \in S_1 \times \cdots \times S_m$ such that $mbb(s) \cap mbb(s_1) \cap \cdots \cap mbb(s_m)$ is a non-trivial box.*

Example 9. Let us continue with the set C of Example 4. Let $S_1 = \{d_1,\ d_2\}$ and $S_2 = \{e_1,\ e_2,\ e_3\}$ be the set of component variables corresponding to a_1 (see Fig. 5). Since d_1, d_2, e_1, e_2, e_3 are subregions of a_1 such that $d_1, d_2, e_1, e_2, e_3 \in REG^*$ and $a_1 = d_1 \cup d_2 = e_1 \cup e_2 \cup e_3$, the preconditions of Lemma 2 hold. Using Fig. 5, it is not hard to see that Constraint (T) also holds. For instance, for subregion $d_1 \in S_1$ there exists a region $s_1 \in S_1$, namely d_1, and a region in $s_2 \in S_2$, namely e_1, such that $mbb(d_1) \cap mbb(s_1) \cap mbb(s_2)$ is a non-trivial box (the corresponding bounding boxes are depicted in Fig. 5).

Lemma 2 is very important since it provides us with an alternative method to check whether the set-union constraints of $\mathcal{S}$ are satisfied. Using Lemma 2

```
For every region variable a_i in {a_1, ..., a_n} Do
  Let Σ_1, ..., Σ_m be the sets of boxes that correspond to the sets of component variables S_1, ..., S_m
  For every s in Σ_1 ∪ ··· ∪ Σ_m
    Q = {s}
    For every Σ' in {Σ_1, ..., Σ_m} Do
      Q' = ∅
      For every s' in Σ' and every q in Q Do
        If s' ∩ q is a non-trivial box Then Q' = Q' ∪ {s' ∩ q}
      EndFor
      If Q' = ∅ Then Return 'Inconsistent'
      Q = Q'
    EndFor
  EndFor
EndFor
```

Fig. 6. Step 3 of Algorithm CONSISTENCY

Algorithm CONSISTENCY equivalently checks whether the maximal solution of O satisfies Constraint (T). To this end it proceeds as follows (see also Fig. 6).

First, the algorithm considers every region variable a_i in $a_1, \ldots, a_n$ in turn. Let us assume that the input set C contains m constraints with region a_i as a primary region (i.e., of the form $a_i \ R \ a_j$). Every time each of these m constraints is processed by Step 1 of Algorithm CONSISTENCY a new set of component variables corresponding to a_i is introduced. Let $S_1, \ldots, S_m$ be these sets of component variables corresponding to a_i. The solution of Step 2 assigns non-trivial boxes to all the members of $S_1, \ldots, S_m$. Let $\Sigma_1, \ldots, \Sigma_m$ be the sets of boxes that correspond to $S_1, \ldots, S_m$ after the assignments of Step 2.

For every component variable s in $\Sigma_1 \cup \cdots \cup \Sigma_m$ Algorithm CONSISTENCY checks whether there exists a tuple $(s_1, \ldots, s_m) \in \Sigma_1 \times \cdots \times \Sigma_m$ such that $mbb(s) \cap mbb(s_1) \cap \cdots \cap mbb(s_m)$ is a non-trivial box. To this end, it utilizes set Q. Initially Q contains only the component variable s. Then, the algorithm considers every set of component variables Σ' in $\Sigma_1, \ldots, \Sigma_m$ in turn. Let us now suppose that algorithm has processed sets Σ_1 to $\Sigma_{\mu-1}$ where $1 \leq \mu - 1 < m$. In this case, set Q contains all non-trivial boxes of the form $s \cap \sigma_1 \cap \cdots \cap \sigma_{\mu-1}$ where $\sigma_i \in \Sigma_i$. Then, the algorithm considers the component variables of Σ_μ and the non-trivial boxes of Q. Algorithm CONSISTENCY finds all regions $s' \in \Sigma_\mu$ and $q \in Q$ such that $s' \cap q$ is a non-trivial box and puts them into a new set Q'. In other words set Q' contains all non-trivial boxes of the form $s \cap \sigma_1 \cap \cdots \cap \sigma_\mu$ where $\sigma_i \in \Sigma_i$. Hence, if Q' is empty, Constraint (T) is violated and Algorithm CONSISTENCY returns 'Inconsistent'. Otherwise Q' is assigned to Q and the algorithm continues with the remaining sets of boxes $\Sigma_{\mu+1}, \ldots, \Sigma_m$ that correspond to variable a.

Our discussion has now been completed. The following result summarizes what we have achieved.

Theorem 1. *Let C be a set of basic cardinal direction constraints. Algorithm* CONSISTENCY *correctly decides whether C is consistent.*

Proof. (sketch) Notice that the constraints in O and Constraint (T) are indeed implied by the given set of basic cardinal direction constraints C. This is trivial to see for Step 1; for Step 3, it follows from Lemma 2. Thus, if Algorithm CONSISTENCY returns 'Inconsistent' then C is inconsistent.

If Algorithm CONSISTENCY returns 'Consistent', we start with a solution of O, and form non-trivial boxes (rectangles) that are solutions of the atomic cardinal direction constraints involving only component variables (these atomic constraints are in the set $\mathcal{S}$ and have been produced by consulting the righthand sides of "iff" definitions of Section 2.1). Then, for each variable a_i of C we form a family of unions of non-trivial boxes such that each union is a solution of a constraint in C with a_i as the primary region. The constraints introduced by Step 1 can be used here to guarantee that each one of these unions has the same bounding box with the bounding box of a_i. Finally, the intersection of all unions in the family for each a_i results in a region $a_i^0 \in REG^*$ (the constraints introduced by Step 3 are used here). All a_i^0's constructed in this way form a tuple which is a solution of the original constraint set C.

For the details of the proof, the interested reader can consult the full version of this paper [21]. ■

4 Complexity of Consistency Checking

We will now turn our attention to the complexity of consistency checking of a set of basic cardinal direction constraints. As we have seen in the previous section checking the consistency of a set of basic cardinal direction constraints can be done using Algorithm CONSISTENCY. The following theorem discusses its complexity.

Theorem 2. *Checking the consistency of a set of* basic *cardinal direction constraints in n variables can be done using Algorithm* CONSISTENCY *in $\mathcal{O}(n^5)$ time.*

Proof. To decide the consistency of a set of basic cardinal direction constraints we can use Algorithm CONSISTENCY introduced above. Let us now calculate its complexity. Let the input to CONSISTENCY be a set C of basic cardinal direction constraints in n region variables. The number of constraints in C is $\mathcal{O}(n^2)$.

The output of Step 1 of Algorithm CONSISTENCY is a set of order constraints O. Algorithm CONSISTENCY introduces at most 9 new variables each time Step 1 is executed. So the total number of region variables is $\mathcal{O}(n^2)$. Step 1 of Algorithm CONSISTENCY adds to O, $\mathcal{O}(n^2)$ atomic order constraints. Therefore, to find a solution of O we have to find a solution of a set of $\mathcal{O}(n^2)$ atomic order constraints in $\mathcal{O}(n^2)$ variables. This can be done in $\mathcal{O}(n^4)$ time using Algorithm CSPAN of [22]. Then Algorithm CONSISTENCY applies Lemma 1. This, can be performed in $\mathcal{O}(n)$ time. Summarizing the first two steps of Algorithm CONSISTENCY can be done in $\mathcal{O}(n^4)$ time.

In the third step, the algorithm checks if Constraint (T) is satisfied. Before we proceed let us measure the size of set Q. For a given variable a_i, $1 \leq i \leq n$, set C contains $\mathcal{O}(n)$ constraints of the form $a_i\ R\ a_j$, $1 \leq i \leq n$. Now since a_i participates in $\mathcal{O}(n)$ constraints of C, it follows that the region represented by variable a_i is divided by $\mathcal{O}(n)$ horizontal and $\mathcal{O}(n)$ vertical lines. Thus a_i is divided into $\mathcal{O}(n^2)$ pieces. Now notice that set Q cannot contain more members than the possible pieces of a_i, thus the size of Q is $\mathcal{O}(n^2)$. In order to check whether Constraint (T) is satisfied Algorithm Consistency performs four nested loops. Each of the outer two loops is executed $\mathcal{O}(n)$ times. Both the inner loops are performed at most $\mathcal{O}(n^3)$ times. Thus checking whether Constraint (T) is satisfied can be done in $\mathcal{O}(n^5)$ time.

Summarizing, the complexity of Algorithm Consistency is $\mathcal{O}(n^5)$. ■

Now turning our attention to the general problem we have the following result.

Theorem 3. *Checking the consistency of an unrestricted set of cardinal direction constraints is NP-complete.*

Proof. Let C be a set of cardinal direction constraints. Deciding the consistency of C is easily seen to be in NP. To prove NP-hardness use a reduction from 3SAT. To this end, we map each literal of 3SAT to a region variable and each clause of 3SAT to a set of cardinal direction constraints. This proof borrows some ideas form a proof that appears in [23] (like the use of a center region). It differentiates in the way we use relations and auxiliary regions. ■

Following the line of work of [18] for topological constraints, we are currently working on investigating tractable subclasses of cardinal direction relations.

5 Extensions

There are two interesting extensions of the basic model. The first extension is to accommodate any region in $\Re^2$ (i.e., to include points and lines). Points and lines have been excluded carefully from REG^* (they are not homeomorphic to the unit disk). They can be easily included by dividing the space around the reference region into 25 areas (9 two-dimensional areas, 8 semi-lines, 4 line segments and 4 points). This should be contrasted to the model presented in Section 2 that divides the space into 9 areas. The new set contains $\sum_{i=1}^{25} \binom{25}{i} = 33,554,431$ jointly exclusive and pairwise disjoint cardinal direction relations. Algorithm Consistency can be easily modified to handle this case (details will be given in the long version of the paper). This extension is similar to [10].

The second extension is to modify the framework so that it also covers the case of connected regions as envisioned originally in [9, 20]. This can be done by considering only 218 relations that are possible between connected regions. Algorithm Consistency also works in this case (details in the long version of the paper [21]).

References

1. A. Berretti, A. Del Bimbo, and E. Vicario. Modeling Spatial Relationships between Color Sets. In *Proceeding of IEEE Workshop on Content-based Access of Image and Video Libraries (CVPR-2000)*, June 2000.

2. W.G. Chinn and N.E. Steenrod. *First Concepts of Topology*. The Mathematical Association of America, 1966.
3. E. Clementini, P. Di Fellice, and G. Califano. Composite Regions in Topological Queries. *Information Systems*, 7:759–594, 1995.
4. E. Clementini, P. Di Fellice, and D. Hernandez. Qualitative Representation of Positional Information. *Artificial Intelligence*, 95:317–356, 1997.
5. J.P. Delgrande, A. Gupta, and T. Van Allen. Point Based Approaches to Qualitative Temporal Reasoning. In *Proceedings of the AAAI-99*, pages 739–744, 1999.
6. B. Faltings. Qualitative Spatial Reasoning Using Algebraic Topology. In *Proceedings of COSIT-95*, volume 988 of *LNCS*, 1995.
7. A.U. Frank. Qualitative Spatial Reasoning about Distances and Directions in Geographic Space. *Journal of Visual Languages and Computing*, 3:343–371, 1992.
8. C. Freksa. Using Orientation Information for Qualitative Spatial Reasoning. In *Proceedings of COSIT-92*, volume 639 of *LNCS*, pages 162–178, 1992.
9. R. Goyal and M.J. Egenhofer. Cardinal Directions Between Extended Spatial Objects. *IEEE Transactions on Data and Knowledge Engineering*, (in press), 2000. Available at `http://www.spatial.maine.edu/~max/RJ36.html`.
10. R. Goyal and M.J. Egenhofer. Consistent Queries over Cardinal Directions across Different Levels of Detail. In *Proceedings of the 11th International Workshop on Database and Expert Systems Applications*, 2000.
11. G. Ligozat. Reasoning about Cardinal Directions. *Journal of Visual Languages and Computing*, 9:23–44, 1998.
12. S. Lipschutz. *Set Theory and Related Topics*. McGraw Hill, 1998.
13. A. Mukerjee and G. Joe. A Qualitative Model for Space. In *Proceedings of AAAI-90*, pages 721–727, 1990.
14. D. Papadias. *Relation-Based Representation of Spatial Knowledge*. PhD thesis, Dept. of Electrical and Computer Engineering, National Technical University of Athens, 1994.
15. D. Papadias, N. Arkoumanis, and N. Karacapilidis. On The Retrieval of Similar Configurations. In *Proceedings of 8th International Symposium on Spatial Data Handling (SDH)*, 1998.
16. C.H. Papadimitriou, D. Suciu, and V. Vianu. Topological Queries in Spatial Databases. *Journal of Computer and System Sciences*, 58(1):29–53, 1999.
17. D.A. Randell, Z. Cui, and A. Cohn. A Spatial Logic Based on Regions and Connection. In *Proceedings of KR'92*. Morgan Kaufmann, October 1992.
18. J. Renz and B. Nebel. On the Complexity of Qualitative Spatial Reasoning: A Maximal Tractable Fragment of the Region Connection Calculus. *Artificial Intelligence*, 1-2:95–149, 1999.
19. A.P. Sistla, C. Yu, and R. Haddad. Reasoning About Spatial Relationships in Picture Retrieval Systems. In *Proceedings of VLDB-94*, pages 570–581, 1994.
20. S. Skiadopoulos and M. Koubarakis. Composing Cardinal Directions Relations. In *Proceedings of the 7th International Symposium on Spatial and Temporal Databases (SSTD-01)*, volume 2121 of *LNCS*, pages 299–317, July 2001.
21. S. Skiadopoulos and M. Koubarakis. Qualitative Spatial Reasoning with Cardinal Directions: Semantics, Algorithms and Computational Complexity. Manuscript in preparation, 2002.
22. P. van Beek. Reasoning About Qualitative Temporal Information. *Artificial Intelligence*, 58:297–326, 1992.
23. M. Vilain, H. Kautz, and P. van Beek. Constraint Propagation Algorithms for Temporal Reasoning: A Revised Report. In *Readings in Qualitative Reasoning about Physical Systems*, pages 373–381. Morgan Kaufmann, 1989.