Fachbereich Informatik der Universität Hamburg

Vogt-Kölln-Str. 30 ♢ D-22527 Hamburg / Germany

University of Hamburg - Computer Science Department

Mitteilung Nr. 277 • Memo No. 277

Spatioterminological Reasoning Based on Geometric Inferences: The $\mathcal{ALCRP}(\mathcal{D})$ Approach

Ralf Möller, Volker Haarslev, Carsten Lutz

Arbeitsbereich KOGS

FBI-HH-M-277/97

Dezember 1997

Spatioterminological Reasoning Based on Geometric Inferences: The $\mathcal{ALCRP}(\mathcal{D})$ Approach

Ralf Möller, Volker Haarslev, Carsten Lutz

University of Hamburg, Computer Science Department,
Vogt-Kölln-Str. 30, 22527 Hamburg, Germany
`http://kogs-www.informatik.uni-hamburg.de/~<name>/`

Abstract

This report presents a method for reasoning about spatial objects and their qualitative spatial relationships. In contrast to existing work, which mainly focuses on reasoning about qualitative spatial relations alone, we integrate quantitative and qualitative information with terminological reasoning by providing an admissible concrete domain for the description logic $\mathcal{ALCRP}(\mathcal{D})$. The theory is motivated as a basis for knowledge representation and query processing in the domain of environmental geographic information systems.

1 Introduction

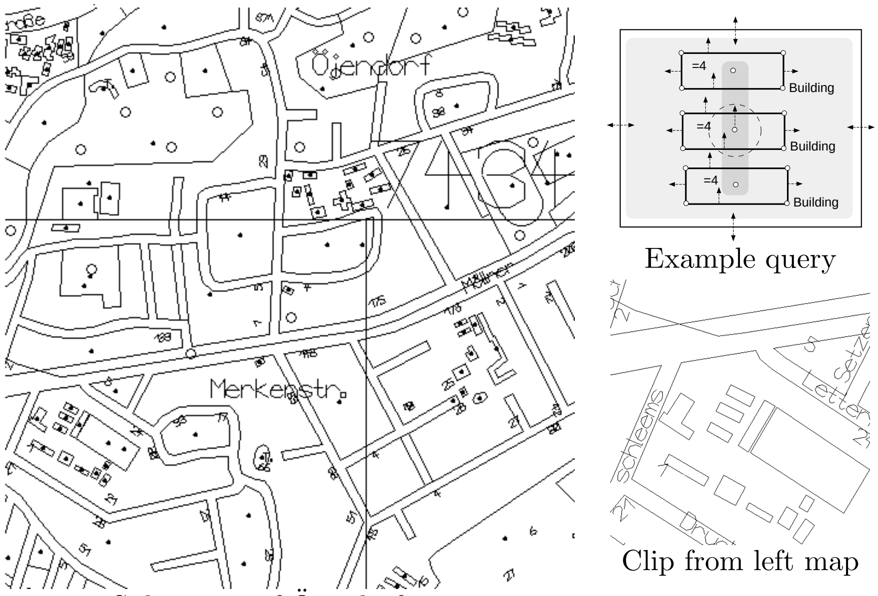
Qualitative relations play an important role in formal reasoning systems that can be part of, for instance, geographic information systems (GIS). In this context, inferences about spatial relations should not be considered in isolation but should be integrated with formal inferences about structural descriptions of domain objects (e.g. automatic consistency checking and classification) and inferences about quantitative data. In our opinion, the abstractions provided by qualitative spatial relations can be interpreted as an interface from a conceptual model about the world to quantitative spatial data representing spatial information about domain objects. The combination of formal conceptual and spatial reasoning serves as a theoretical basis for knowledge representation in GIS and can be used to solve important application problems. Continuing our work presented in [Haarslev and Möller 1997] and [Haarslev, Möller, and Schröder 1994] we demonstrate the importance of terminological inferences with spatial relations in the domain of map databases and spatial query processing.

For instance, in order to answer a GIS query, concept terms are computed on the fly and have to be checked for consistency. Furthermore, we assume that computed concept terms must be automatically inserted into the subsumption hierarchy of a knowledge base. Concerning applications in the area of map interpretation we would like to emphasize a characteristic of these problems. It is often very difficult to describe a fixed algorithm that defines an exact sequence of “interpretation steps” because several different “cues” have to be integrated. In other words: the solution has to be computed by adequately integrating partial information about domain objects. Therefore, we need sound and complete formalisms such as description logics that do not depend on any serialization steps. We developed the description logic $\mathcal{ALCRP}(\mathcal{D})$ [Lutz, Haarslev, and Möller 1997] that is well suited for modeling GIS objects. The information about objects is given by conceptual background knowledge and GIS data.

As an example, a small subsection of a vector map from the city of Hamburg is shown in the left part of Figure 1. We assume that basic map objects are predefined in a GIS. Furthermore, spatial areas are defined by polygons. Map elements (e.g. polylines, polygons) are partly annotated with labels like “living-area”, “building”, “ordinary-road” etc. The upper right part of Figure 1 contains a sketch describing a visual query. In this example we search for a constellation where three buildings are aligned in parallel. The form and position of the buildings may vary. We refer to [Haarslev and Wessel 1997] for more details about the visual query language. The right part of Figure 1 also contains a small magnified clip from the lower left corner of the map. This clip shows in its center three parallel rectangles aligned in north-east direction. These rectangles represent buildings that are an example match for the visual query of Figure 1. Because of the intended vagueness of visual queries, qualitative spatial relations like touching, overlapping, disjoint etc. are used to define the semantics of a query sketch (see [Haarslev and Wessel 1997] for details). Reasoning about spatial relations has to be combined with reasoning about conceptual information attached to the visual objects (e.g. house, street, living-area).

For formalizing reasoning about spatial structures many theories have been published (see e.g. [Stock 1997] for an overview). Ignoring decidability, Borgo et al. [Borgo, Guarino, and Masolo 1996] have developed a first order theory of space which formalizes different aspects such as mereology etc. An algebraic theory about space has been proposed by [Pratt and Lemon 1997; Pratt and Schoop 1997]. The well-known RCC theory [Cohn, Bennett, Gooday, and Gotts 1997] also formalizes *qualitative* reasoning about space. While first axiomatizations used first-order logic, recently, the spatial relations used in RCC have been defined in terms of intuitionistic logic and propositional modal logic [Bennett 1995]. Although qualitative reasoning with RCC can be used in many applications, in GIS also conceptual knowledge combined with quantitative data has to be considered. Therefore, another approach is required.

In order to adequately support decidable reasoning (i) about qualitative relations between spatial regions and (ii) about properties of quantitative data,



Subsection of Öjendorf map

Example query

Clip from left map

Figure 1: City map and query example.

we extend the description logic $\mathcal{ALC}(\mathcal{D})$ [Baader and Hanschke 1991a]. The main idea of our approach is to deal with spatial objects and their relations using predicates over concrete domain objects (see below for a formal introduction) and to deal with knowledge about abstract domain objects using the well-known description logic theory. Although description logics in general, and $\mathcal{ALC}$ in particular, are known to be strongly related to propositional modal logics [Schild 1991; De Giacomo and Lenzerini 1994], it is not clear how Bennett's modal logic can be extended to handle quantitative data. Because Bennett's modal logic formalization of RCC [Bennett 1995] uses the transitivity axiom, decidability problems can be expected if, for instance, concrete domains over the reals have to be treated by the satisfiability tester, too. This is due to the fact that $\mathcal{ALC}(\mathcal{D})$ with transitive closure of roles is known to be undecidable [Baader and Hanschke 1991b].

Extending the work on $\mathcal{ALC}(\mathcal{D})$, we have developed a new description logic called $\mathcal{ALCRP}(\mathcal{D})$ [Lutz, Haarslev, and Möller 1997] in order to provide a foundation for spatioterminological reasoning with description logics. The part of our theory dealing with spatial relations is based on a set of topological relations in analogy to Egenhofer [Egenhofer 1991] or RCC-8 [Randell, Cui, and Cohn 1992]. The goal was to develop a description logic that provides modeling constructs which can be used to represent topological relations as defined roles. In a specific domain model, spatial areas can be represented as concrete objects, which, in turn, are associated to individuals via specific features. Then, roles representing topological relations can be defined that are based on predicates over concrete objects. $\mathcal{ALCRP}(\mathcal{D})$ supports this modeling technique by providing role terms that refer to predicates over a concrete domain. With these constructs $\mathcal{ALCRP}(\mathcal{D})$ extends the expressive power of $\mathcal{ALC}(\mathcal{D})$ (for a comparison, see [Lutz, Haarslev, and Möller 1997]). However, in order to ensure termination of the satisfiability algorithm, we impose restrictions on the syntactic form of the set of terminological axioms. Although modeling is harder, syntactic restrictions on terminologies ensure decidability of the language.

In our earlier work presented in [Haarslev, Möller, and Schröder 1994] and [Haarslev 1995], [Haarslev and Wessel 1996], as well as [Haarslev 1998], we used topological relations as primitives in the sense of logic. However, these relations were based on properties of concrete objects and we had to rely on external mechanisms for synchronizing the actual properties of objects with their logical counterparts. With the help of $\mathcal{ALCRP}(\mathcal{D})$ we can extend the treatment of topological relations with respect to terminological reasoning. Thus, the theory presented in this paper allows one to detect both inconsistencies and implicit information in formal conceptual models for spatial domain objects.

In contrast to our earlier work presented in [Haarslev, Möller, and Schröder 1994], [Haarslev 1995], [Haarslev and Wessel 1996] and [Haarslev 1998], where topological relations are used as primitives in the sense of logic, we extend the treatment of topological relations with respect to terminological reasoning. Thus, the theory presented in this paper allows one to detect both inconsistencies and implicit information in formal conceptual models for spatial domain objects.

The next section outlines the formal foundations for spatioterminological reasoning. It is followed by a section introducing a concrete domain for polygonal space that is used for integrating spatial and terminological reasoning. This section also discusses an extended example. We conclude the report with a summary and point out topics for ongoing work.

2 Foundations of Spatioterminological Reasoning with Description Logic

The previous section motivated the formalization of space with the help of terminological and spatial inference services. This section introduces the formal tools necessary for spatioterminological reasoning. We define spatial regions and their qualitative relationships and present the description logic $\mathcal{ALCRP}(\mathcal{D})$ [Lutz, Haarslev, and Möller 1997] in terms of extensions to the description logic $\mathcal{ALC}(\mathcal{D})$ [Baader and Hanschke 1991a].

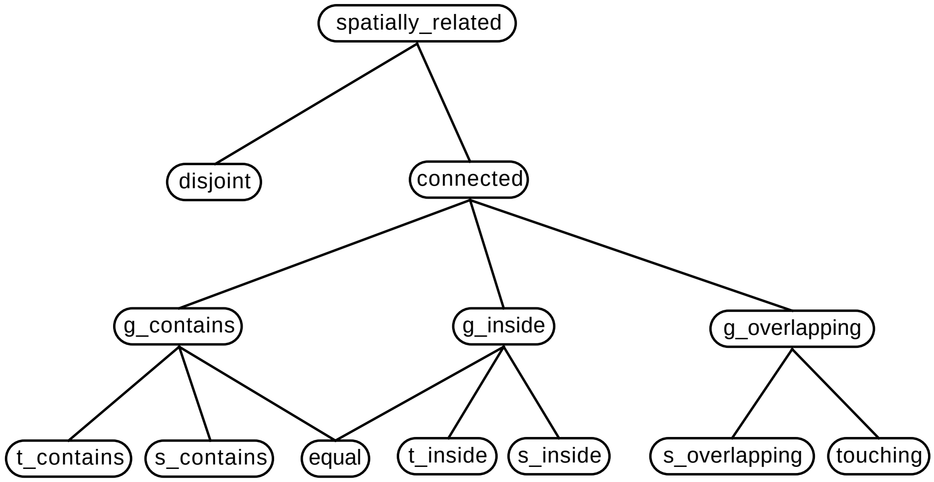


Figure 2: Subsumption hierarchy of spatial relations.

2.1 Qualitative Spatial Relationships

In the following we assume the usual concepts of point-set topology with open and closed sets [Spanier 1966]. For a set λ_i we denote its *interior* as λ_i^o and its *boundary* as $\partial\lambda_i$. In a similar way as [Grigni, Papadias, and Papadimitriou 1995] we define 13 binary topological relations that are organized in a subsumption hierarchy (see Figure 2). The leaves of this hierarchy represent eight mutually exclusive relations. The eight relations are equivalent to the set defined by Egenhofer [Egenhofer 1991] or RCC-8 [Randell, Cui, and Cohn 1992]. We assume the usual restrictions for sets in analogy to [Egenhofer 1991]. The eight relations are also referred to as *elementary* relations. Figure 3 illustrates five elementary relations.

- **spatially_related**: Two objects have a *spatial relationship* between each other iff they are either disjoint or connected.

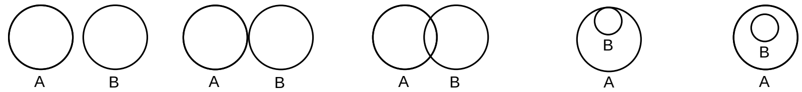
  $$\mathsf{spatially\_related}(\lambda_1, \lambda_2) \equiv \mathsf{disjoint}(\lambda_1, \lambda_2) \vee \mathsf{connected}(\lambda_1, \lambda_2)$$

- **disjoint**: Two objects are *disjoint* iff their intersection is empty; disjoint is symmetric.

  $$\mathsf{disjoint}(\lambda_1, \lambda_2) \equiv \lambda_1 \cap \lambda_2 = \emptyset$$

- **connected**: Two objects are *connected* iff their intersection is non-empty; connected is symmetric.

  $$\mathsf{connected}(\lambda_1, \lambda_2) \equiv \lambda_1 \cap \lambda_2 \neq \emptyset$$



disjoint touching s_overlapping t_contains s_contains

Figure 3: Elementary spatial relations between two regions A and B. The inverses of t_contains and s_contains as well as the relation equal have been omitted.

- **g_overlapping**: Two objects are *generally overlapping* iff they are either touching or strictly overlapping; g_overlapping is symmetric.

  $$\mathsf{g\_overlapping}(\lambda_1, \lambda_2) \equiv \mathsf{touching}(\lambda_1, \lambda_2) \vee \mathsf{s\_overlapping}(\lambda_1, \lambda_2)$$

- **touching**: Two objects are *touching* iff only their boundaries are intersecting; touching is symmetric.

  $$\mathsf{touching}(\lambda_1, \lambda_2) \equiv (\partial\lambda_1 \cap \partial\lambda_2 \neq \emptyset) \wedge (\lambda_1^{\circ} \cap \lambda_2^{\circ} = \emptyset)$$

- **s_overlapping**: Two objects are *strictly overlapping* iff their interiors are intersecting and the intersection is not equal to either of them; s_overlapping is symmetric.

  $$\mathsf{s\_overlapping}(\lambda_1, \lambda_2) \equiv (\lambda_1 \cap \lambda_2 \neq \lambda_1) \wedge (\lambda_1 \cap \lambda_2 \neq \lambda_2) \wedge (\lambda_1^{\circ} \cap \lambda_2^{\circ} \neq \emptyset)$$

- **g_contains/g_inside**: An object λ_1 *generally contains* an object λ_2 iff it either tangentially or strictly contains λ_2 or it is equal to λ_2; g_inside is the inverse of g_contains; g_contains and g_inside are reflexive, antisymmetric and transitive.

  $$\begin{aligned}\mathsf{g\_contains}(\lambda_1, \lambda_2) \equiv\ & \mathsf{t\_contains}(\lambda_1, \lambda_2) \vee \mathsf{s\_contains}(\lambda_1, \lambda_2) \vee \\ & \mathsf{equal}(\lambda_1, \lambda_2)\end{aligned}$$

  $$\mathsf{g\_inside}(\lambda_1, \lambda_2) \equiv \mathsf{g\_contains}(\lambda_2, \lambda_1)$$

- **equal**: Two objects are *equal* iff they describe the same set of points.

  $$\mathsf{equal}(\lambda_1, \lambda_2) \equiv \lambda_1 = \lambda_2$$

- **t_contains/t_inside**: An object λ_1 *tangentially contains* an object λ_2 iff their intersection is equal to λ_2 and the intersection of their boundaries is non-empty; the inverse of $\mathsf{t\_contains}$ is $\mathsf{t\_inside}$; $\mathsf{t\_contains}$ and $\mathsf{t\_inside}$ are asymmetric.

$$\mathsf{t\_contains}(\lambda_1, \lambda_2) \equiv (\lambda_1 \cap \lambda_2 = \lambda_2) \land (\lambda_1 \cap \lambda_2^{-1} \neq \emptyset) \land (\partial\lambda_1 \cap \partial\lambda_2 \neq \emptyset)$$
$$\mathsf{t\_inside}(\lambda_1, \lambda_2) \equiv \mathsf{t\_contains}(\lambda_2, \lambda_1)$$

- **s_contains/s_inside**: An object λ_1 *strictly contains* an object λ_2 iff their intersection is equal to λ_2 and only the interiors of their regions intersect; the inverse of $\mathsf{s\_contains}$ is $\mathsf{s\_inside}$; $\mathsf{s\_contains}$ and $\mathsf{s\_inside}$ are asymmetric and transitive.

$$\mathsf{s\_contains}(\lambda_1, \lambda_2) \equiv (\lambda_1 \cap \lambda_2 = \lambda_2) \land (\lambda_1 \cap \lambda_2^{-1} \neq \emptyset) \land (\partial\lambda_1 \cap \partial\lambda_2 = \emptyset)$$
$$\mathsf{s\_inside}(\lambda_1, \lambda_2) \equiv \mathsf{s\_contains}(\lambda_2, \lambda_1)$$

2.2 Description Logic

We represent terminological knowledge about spatial (e.g. GIS) domains using description logic (DL) theory that has been proven to be a useful formalism for modeling in technical domains. The concept language $\mathcal{ALCRP}(\mathcal{D})$ is shortly introduced in this report and is described in detail in our companion report [Lutz, Haarslev, and Möller 1997]. $\mathcal{ALCRP}(\mathcal{D})$ is based on $\mathcal{ALC}(\mathcal{D})$ as defined in [Baader and Hanschke 1991a]. It extends $\mathcal{ALC}(\mathcal{D})$ by a role-forming operator that is based on concrete domain predicates. In the following we define concrete domains and their admissibility in analogy to [Baader and Hanschke 1991a].

Definition 1
A *concrete* domain $\mathcal{D}$ is a pair $(\Delta_\mathcal{D}, \Phi_\mathcal{D})$, where $\Delta_\mathcal{D}$ is a set called the domain, and $\Phi_\mathcal{D}$ is a set of predicate names. Each predicate name P from $\Phi_\mathcal{D}$ is associated with an arity n, and an n-ary predicate $P^\mathcal{D} \subseteq \Delta_\mathcal{D}^n$.

A concrete domain $\mathcal{D}$ is called *admissible* iff

1. the set of its predicate names is closed under negation and contains a name for $\Delta_\mathcal{D}$.

2. the satisfiability problem for finite conjunctions of predicates is decidable.

The next definitions present the syntax and model-theoretic semantics of $\mathcal{ALCRP}(\mathcal{D})$ (see also our companion report [Lutz, Haarslev, and Möller 1997]).

Definition 2
Let R and F be disjoint sets of role and feature names, respectively. Any element of R and any element of F is an *atomic* role term. The elements of F are also called *features*. A composition of features (written $f_1 f_2 \cdots$) is called a feature chain. A feature chain of length one is also a feature chain. If $P \in \Phi_{\mathcal{D}}$ is a predicate name with arity $n+m$ and $u_1, \ldots, u_n$ as well as $v_1, \ldots, v_m$ are feature chains, then the expression $\exists(u_1, \ldots, u_n)(v_1, \ldots, v_m).P$ (*role-forming predicate restriction*) is a *complex* role term.[1] Let S be a role name and let T be a role term. Then $S \doteq T$ is a terminological axiom.

The next definition introduces concept terms of $\mathcal{ALCRP}(\mathcal{D})$. As we will see later, if decidable reasoning algorithms are needed for a certain terminology, not all of these concept terms can be freely combined.

Definition 3
Let C be a set of concept names which is disjoint from R and F. Any element of C is an *atomic* concept term. If C and D are concept terms, R is an arbitrary role term (it may also be a feature), $P \in \Phi_{\mathcal{D}}$ is a predicate name with arity n, and $u_1, \ldots, u_n$ are feature chains, then the following expressions are also concept terms:

1. $C \sqcap D$ (*conjunction*),
2. $C \sqcup D$ (*disjunction*),
3. $\neg C$ (*negation*),
4. $\exists R.C$ (*concept exists restriction*) and
5. $\forall R.C$ (*concept value restriction*), and
6. $\exists u_1, \ldots, u_n.P$ (*predicate exists restriction*).

For all kinds of exists and value restrictions, the role term or list of feature chains may be written in parentheses.
Let A be a concept name and let D be a concept term. Then $A \doteq D$ is a terminological axiom as well. A finite set of terminological axioms $\mathcal{T}$ is called a *terminology* or *TBox* if no concept or role name in $\mathcal{T}$ appears more than once on the left hand side of a definition and, furthermore, if no cyclic definitions are present.

The semantics for the language is defined as usual for $\mathcal{ALC}$ languages with concrete domains (cf. [Baader and Hanschke 1991a]). In the following, we define only those interpretation function extensions concerning concept terms and role terms that refer to the concrete domain.

[1] Note that there have to be at least one u and one v.

Definition 4
An *interpretation* $\mathcal{I} = (\Delta_{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a set $\Delta_{\mathcal{I}}$ (the abstract domain) and an interpretation function $\cdot^{\mathcal{I}}$. The sets $\Delta_{\mathcal{D}}$ and $\Delta_{\mathcal{I}}$ must be disjoint. The interpretation function maps each concept name C to a subset $C^{\mathcal{I}}$ of $\Delta_{\mathcal{I}}$, each role name R to a subset $R^{\mathcal{I}}$ of $\Delta_{\mathcal{I}} \times \Delta_{\mathcal{I}}$, and each feature name f to a partial function $f^{\mathcal{I}}$ from $\Delta_{\mathcal{I}}$ to $\Delta_{\mathcal{D}} \cup \Delta_{\mathcal{I}}$, where $f^{\mathcal{I}}(a) = x$ will be written as $(a, x) \in f^{\mathcal{I}}$. If $u = f_1 \cdots f_n$ is a feature chain, then $u^{\mathcal{I}}$ denotes the composition $f_1^{\mathcal{I}} \circ \cdots \circ f_n^{\mathcal{I}}$ of the partial functions $f_1^{\mathcal{I}}, \ldots, f_n^{\mathcal{I}}$. Be $P \in \Phi_{\mathcal{D}}$ a predicate name with arity $n + m$, $u_1, \ldots, u_n$ feature chains, and $f_1, \ldots, f_m$ feature names. Let R be an arbitrary role term (it may also be a feature). Then the interpretation function is extended for the following constructs:

$$
\begin{aligned}
(\exists u_1, \ldots, u_n.P)^{\mathcal{I}} := \{a \in \Delta_{\mathcal{I}} \mid \exists x_1, \ldots, x_n \in \Delta_{\mathcal{D}} &: \\
(a, x_1) \in u_1^{\mathcal{I}}, \ldots, (a, x_n) \in u_n^{\mathcal{I}}, (x_1, \ldots, x_n) &\in P^{\mathcal{D}}\} \\
(\exists (u_1, \ldots, u_n)(v_1, \ldots, v_m).P)^{\mathcal{I}} &:= \\
\{(a, b) \in \Delta_{\mathcal{I}} \times \Delta_{\mathcal{I}} \mid \exists x_1, \ldots, x_n, y_1, \ldots, y_m &\in \Delta_{\mathcal{D}}: \\
(a, x_1) \in u_1^{\mathcal{I}}, \ldots, (a, x_n) &\in u_n^{\mathcal{I}}, \\
(b, y_1) \in v_1^{\mathcal{I}}, \ldots, (b, y_m) &\in v_m^{\mathcal{I}}, \\
(x_1, \ldots, x_n, y_1, \ldots, y_m) &\in P^{\mathcal{D}}\}
\end{aligned}
$$

It has been shown in [Lutz and Möller 1997] that, in general, the satisfiability problem for $\mathcal{ALCRP}(\mathcal{D})$ is undecidable. However, in our companion report [Lutz, Haarslev, and Möller 1997], we show the decidability of so-called *restricted terminologies* of $\mathcal{ALCRP}(\mathcal{D})$ provided an *admissible* concrete domain is specified. The *restrictedness* criterion is defined as follows.

Definition 5
A terminology $\mathcal{T}$ is called *restricted* iff its unfolded negation normal form (see [Lutz, Haarslev, and Möller 1997]) fulfills all of the following conditions:

1. For any (sub)concept term C in $\mathcal{T}$ that is of the form $\forall R_1.D$ where R_1 is a complex role term, D does not contain any terms of the form $\exists R_2.D$ where R_2 is also a complex role term.

2. For any (sub)concept term C in $\mathcal{T}$ that is of the form $\exists R_1.D$ where R_1 is a complex role term, D does not contain any terms of the form $\forall R_2.D$ where R_2 is also a complex role term.

3. For any (sub)concept term C in $\mathcal{T}$ that is of the form $\forall R.D$ or $\exists R.D$ where R is a complex role term, D contains only predicate exists restrictions that quantify over attribute chains of length of 1 and, furthermore, do not occur inside any value and exists restrictions that are also contained in D.

The restrictedness criterion appears to be a rather severe modeling constraint but the next section will demonstrate that $\mathcal{ALCRP}(\mathcal{D})$ with restricted terminologies is still a very useful description logic for integrating conceptual and spatial reasoning.

3 A Concrete Domain for Polygonal Space

This section introduces a concrete domain for modeling spatial objects and, based on the spatial relations defined in the previous section, a language for specifying predicates over this domain is given. We demonstrate that the admissibility criterion can be fulfilled. Our concrete domain implements the satisfiability test for a conjunction of spatial predicate terms.

Rather than dealing with arbitrary point sets in $\Re^2$, we restrict the spatial predicates to the description of polygons because efficient algorithms (e.g. the simplex procedure) are known for the polygon inclusion and polygon intersection problems. In accordance to Definition 1 we define the concrete domain $\mathcal{D}_\mathcal{P}$ as consisting of a set $\Delta_{\mathcal{D}_\mathcal{P}}$ of polygons and a set $\Phi_{\mathcal{D}_\mathcal{P}}$ of predicate names.

3.1 Polygons as Concrete Objects

We describe polygons as n-tuples of coordinates, i.e. a polygon represents a single (connected) set of points from $\Re^2$ without holes. The interior of the polyline is assumed to be on the left hand side of the chain of edges. The edges of a polygon must not intersect one another. If we syntactically denote polygons as lists of coordinates, a normalization function can transform adjacent collinear edges into a single edge. We consider the description logic $\mathcal{ALCRP}(\mathcal{D}_\mathcal{P})$ in the following subsections.

3.2 Predicates for Qualitative Spatial Relationships

We define $\mathcal{SP}$ as the set of names[2] for the 13 spatial predicates implementing the spatial relations defined in Section 2.1. Each predicate in $\mathcal{SP}$ has a unique name. For each predicate, a negated counterpart can be easily defined, i.e. the set of predicate names $\mathcal{SP}$ can be extended in a way that $\mathcal{SP}$ is closed under negation (cf. the admissibility criterion for concrete domains in Definition 1). The set $\mathcal{SP}$ contains the two disjoint sets $\mathcal{EP}$ and $\mathcal{CP}$. $\mathcal{EP}$ contains the eight *elementary* predicates, $\mathcal{CP}$ the five *composite* predicates. The set of composite predicates is defined to facilitate modeling in our

[2] In the following we do not distinguish between predicates and their names. We also use the terms relation and predicate interchangeably.

spatial domain. A composite predicate can always be reduced to a disjunction of elementary predicates. The (transitive) composition of predicates is defined by a *composition table* representing one-step inferences (for details see [Egenhofer 1991]). For instance, $\mathsf{touching}(a, b) \wedge \mathsf{touching}(b, c)$ yields $\mathsf{disjoint}(a, c) \vee \mathsf{touching}(a, c) \vee \mathsf{equal}(a, c) \vee \mathsf{s\_inside}(a, c) \vee \mathsf{s\_contains}(a, c) \vee \mathsf{s\_overlapping}(a, c)$.

We define one-place predicates (denoted as $\mathsf{sr_p}$) where sr is an elementary or composite predicate and where p is a concrete object representing a constant polygon. We also assume that the set $\mathcal{SP}$ is properly extended by a set of one-place predicate names and their negated counterparts. The semantics of one-place predicates is as follows.

$$\mathsf{sr_p}^{\mathcal{I}} := \{x \in \Delta_{\mathcal{D_P}} \,|\, (x, \mathsf{p}) \in \mathsf{sr}^{\mathcal{I}}\} \text{ with } \mathsf{sr}^{\mathcal{I}} \subseteq \Delta_{\mathcal{D_P}} \times \Delta_{\mathcal{D_P}}$$

It is interesting to note that the spatial predicate $\mathsf{g\_contains}$ has the same properties as the subsumption relation (i.e. it is reflexive, antisymmetric, transitive). Therefore, a polygon $\mathsf{p_1}$ *spatially subsumes* another polygon $\mathsf{p_2}$ iff $\mathsf{p_1}$ $\mathsf{g\_contains}$ $\mathsf{p_2}$. This property will be utilized for modeling in our GIS example domain (see Section 3.4).

3.3 Satisfiability of Conjunctions of Spatial Predicates

The admissibility criterion for $\mathcal{D_P}$ concerns the satisfiability of finite conjunctions of (possibly negated) predicate terms. Negated unary and binary terms (possibly containing composite predicates) can be resolved into disjunctions of elementary spatial predicates since the elementary predicates are mutually exclusive and exhaustive. Therefore, we can restrict our analysis to conjunctions of unnegated terms $\bigvee_{j=1}^{k_i} \mathsf{ep}^j$ where $1 \leq k_i \leq 8$ and $\mathsf{ep}^j \in \mathcal{EP}$.

Consistency of a conjunction of binary predicate terms is usually considered as a binary constraint satisfaction problem. In this view, a conjunction is represented as a constraint network whose nodes are defined by variable names and whose edges are labeled by relation sets representing disjunctions of relation names valid between a pair of nodes. A standard technique for deciding the satisfiability of such a network is the *3-consistency* or *path consistency* method that is based on the composition table. This table defines the composition of spatial relations, for instance it has to hold that $\mathsf{s\_inside} \circ \mathsf{s\_inside} = \mathsf{s\_inside}$ (see above). In other words, a composition table directly encodes so-called *3-consistent* or *path consistent* spatial relations between three regions, e.g. $\mathsf{s\_inside}(A, B) \wedge \mathsf{s\_inside}(B, C) \Rightarrow \mathsf{s\_inside}(A, C)$. In some cases, an additional step is required to ensure global consistency.

The network is unsatisfiable iff the relation set becomes empty for any edge, i.e. the predicate term is inconsistent with the composition table. In case of satisfiability, this step results in a network refining all relation sets with respect to consistency.

Algorithms for solving these constraint problems are discussed in [Ladkin and Reinefeld 1997] and [Nebel 1995; Renz and Nebel 1997]. According to Nebel and Renz [Nebel 1995; Renz and Nebel 1997], the worst case complexity depends on the relations (disjunctions of base relations) actually used in a constraint network. To achieve global consistency, in the worst case, exponential algorithms are required. However, Nebel and Renz [Nebel 1995; Renz and Nebel 1997] showed that for certain subsets of $\mathcal{EP}$ global consistency is equivalent to path consistency. These findings can be used to speed up the verification of relational consistency (Ladkin and Reinefeld [Ladkin and Reinefeld 1997] also discuss speedup techniques).

Grigni et al. [Grigni, Papadias, and Papadimitriou 1995] proposed two notions of satisfiability, relational consistency and realizability, where relational consistency is a necessary condition for realizability. They showed that both notions may be the cause for the unsatisfiability of a conjunction of predicate terms. First, a conjunction may violate the *relational consistency* criterion which is identical to the global consistency of a (spatial) constraint network. The full form of satisfiability is called *realizability* and is related to *planarity*. A relationally consistent conjunction may violate realizability if planar regions are declared to be disjoint from one another (e.g. see [Grigni, Papadias, and Papadimitriou 1995]).

Our algorithm for the concrete domain satisfiability test is divided into four steps. The first two steps implement the normalization phase, while the last two steps verify the consistency of the resulting predicate terms. The last two steps can fail at any time and thus prove unsatisfiability.

1. Negated predicate terms are replaced by the corresponding disjunction of elementary predicate terms. Afterwards, every conjunct consists of either a single term $\mathsf{sr}(x,y)$ or a disjunction $\bigvee_{i=1}^{k} \mathsf{sr}_i(x,y)$ with $\mathsf{sr}, \mathsf{sr}_i \in \mathcal{ER}$, $1 \leq k \leq 8$, and all $\mathsf{sr}_i(x,y)$ are involved with the same pair of objects. For instance, $\neg\mathsf{g\_inside}(x,y)$ is replaced by $\mathsf{t\_contains}(x,y) \vee \mathsf{s\_contains}(x,y) \vee \mathsf{s\_overlapping}(x,y) \vee \mathsf{touching}(x,y) \vee \mathsf{disjoint}(x,y)$.

2. For every conjunct this step adds a new conjunct representing the inverted relation term. For example, for the term $\mathsf{t\_contains}(x,y)$ we add $\mathsf{t\_inside}(y,x)$.

3. This step verifies relational consistency with the help of path consistency as outlined above. It is important to note that reasoning about quantitative geometric data is also involved in the third step because

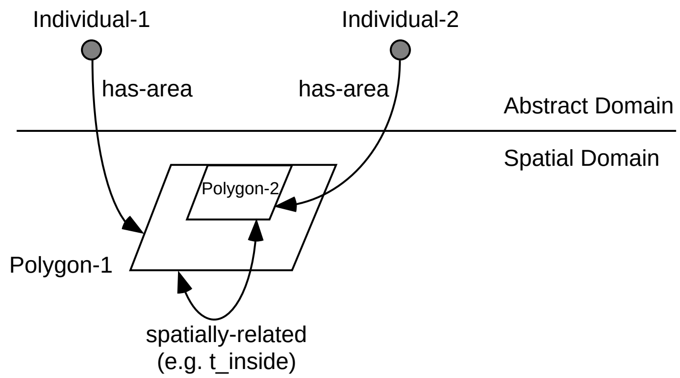


Figure 4: Relationship between abstract and concrete domain

some predicates may have polygons instead of variable names as parameters. This is caused by the one-place predicates $\mathsf{sr}^{i}_{\mathsf{p}}$ introduced above. With the help of standard algorithms from computational geometry (e.g. see [de Berg 1997]) we infer/verify every relation set labeling an edge that connects two polygons. In accordance to our definition of the spatial relations, this problem can be reduced to the intersection test for two polygons. Corresponding algorithms have a time complexity of at most $\mathcal{O}(n^2)$ where n is the average number of edges of polygons.

4. If necessary, the last step verifies planarity by constructing a set of simply connected planar regions, one for each node, any pair of regions is related by a relation name that is element of the relation set labeling the connecting edge between these nodes. Note that in the presence of quantitative data there exist more predicate conjunctions that are relationally consistent but fail the planarity test (see [Lutz and Möller 1997] for an example).

In summary, we showed that the concrete domain $\mathcal{P}$ is admissible with reference to Definition 1.

3.4 Examples for Spatioterminological Reasoning

How can predicates over the concrete domain of polygons be used to support spatioterminological inferences with the description logic $\mathcal{ALCRP}(\mathcal{D})$? First of all, as an ontological commitment, we assume that each abstract domain object is associated with its spatial representation via the feature (or attribute) $\mathsf{has\_area}$ (see Figure 4). Now, we can use the concept-forming predicate operator in combination with one-place predicates for restricting role fillers of $\mathsf{has\_area}$ to be specific spatial regions. For instance, subsumption between concept terms such as $\exists\, \mathsf{has\_area} \,.\, \mathsf{g\_inside}_{\mathsf{p_i}}$ (with $\mathsf{g\_inside}_{\mathsf{p_i}} \in \mathcal{SP}$) resembles inclusion of regions because every concept term $\exists\, \mathsf{has\_area} \,.\, \mathsf{g\_inside}_{\mathsf{p_i}}$

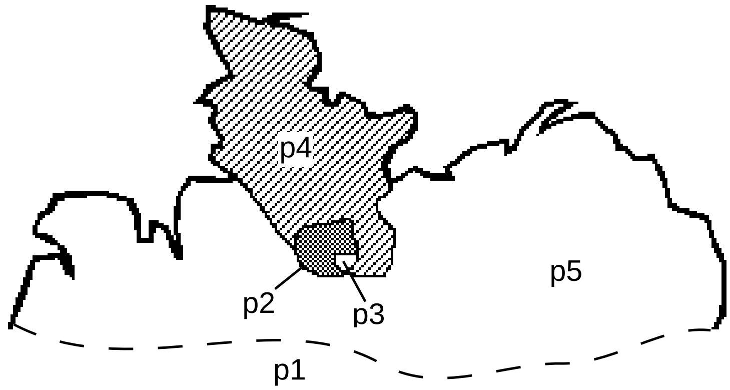


Figure 5: A sketch of the northern part of Germany with polygons for Germany (p_1), Northern Germany (p_5), the federal states Schleswig-Holstein (p_4) and Hamburg (p_2) as well as a small district of Hamburg (p_3). Polygon p_3 is assumed to be inside p_2 but p_2 is not inside p_4.

subsumes the term $\exists\,\mathsf{has\_area}\,.\,\mathsf{g\_inside}_{\mathsf{p_j}}$ iff $\mathsf{p_i}$ g_contains (i.e. spatially subsumes) $\mathsf{p_j}$.

In our GIS example we apply this technique to modeling the regions of the German federal states, of Northern Germany, of a district of the city of Hamburg, etc. The restrictedness criterion for the following set of TBox axioms (cf. Definition 5) is trivially fulfilled because they contain no nested exists or universal quantifiers.

$$\mathbf{northern\_german\_region} \doteq \exists\,\mathsf{has\_area}\,.\,\mathsf{g\_inside}_{\mathsf{p_5}}$$
$$\mathbf{district\_of\_hh} \doteq \exists\,\mathsf{has\_area}\,.\,\mathsf{g\_inside}_{\mathsf{p_2}} \sqcap \exists\,\mathsf{has\_area}\,.\,\neg\mathsf{equal}_{\mathsf{p_2}}$$

The concept northern_german_region is defined by an existential restriction for the attribute has_area whose filler is constrained to be any polygon that is g_inside of p_5 which defines the area of Northern Germany (see Figure 5). In other words: The concept denoted by $\exists\,\mathsf{has\_area}\,.\,\mathsf{g\_inside}_{\mathsf{p_5}}$ subsumes every region of Northern Germany whose associated polygon is g_inside of p_5. Therefore, district_of_hh is automatically classified as a subconcept of northern_german_region.

$$\begin{aligned}
\mathbf{german_federal_state} &\doteq \mathsf{federal_state} \sqcap \\
&\quad (\exists\,\mathsf{has_area}\,.\,\mathsf{equal}_{\mathsf{p_2}} \sqcup \exists\,\mathsf{has_area}\,.\,\mathsf{equal}_{\mathsf{p_4}} \sqcup \ldots) \\
\mathbf{federal_state_hh} &\doteq \mathsf{german_federal_state} \sqcap \exists\,\mathsf{has_area}\,.\,\mathsf{equal}_{\mathsf{p_2}} \\
\mathbf{federal_state_sh} &\doteq \mathsf{german_federal_state} \sqcap \exists\,\mathsf{has_area}\,.\,\mathsf{equal}_{\mathsf{p_4}}
\end{aligned}$$

The concept definition of $\mathsf{german\_federal\_state}$ contains a disjunction of concept terms that characterize the locations of all possible German federal states. Due to the definition of equal, the predicate $\mathsf{equal_p}$ does not subsume arbitrary regions in Germany. As a consequence, the area of, for instance, $\mathsf{district\_of\_hh}$ is not subsumed by the area of $\mathsf{german\_federal\_state}$.

We also define the concepts for the federal states Hamburg and Schleswig-Holstein. We would like to emphasize that both concepts are subsumed by the concept $\mathsf{northern\_german\_region}$. This is due to the definition of the spatial relations in the previous section. For instance, the predicate $\mathsf{equal_{p_2}}$ is subsumed by $\mathsf{g\_inside_{p_2}}$ and, in turn, this predicate is subsumed by $\mathsf{g\_inside_{p_5}}$ because the region $\mathsf{p_5}$ $\mathsf{g\_contains}$ $\mathsf{p_2}$.

In many cases, restrictions about spatial relations have to be combined with additional restrictions. For example, how can we define a concept that describes a district of Hamburg that touches the federal state Hamburg from the inside? Note that it is not sufficient that the corresponding district polygon (e.g. $\mathsf{p_3}$ in Figure 5) is inside any polygon that is equal to the state polygon (e.g. $\mathsf{p_2}$). The domain object that refers to this polygon with the role $\mathsf{has\_area}$ must also be subsumed by the concept $\mathsf{federal\_state\_hh}$ (see the example presented in Figure 4). For modeling spatial relations we declare corresponding roles as part of the TBox. The following TBox axioms fulfill the restrictedness criterion because the nested concept terms employ only the $\exists\, \mathsf{f} \,.\, \mathsf{P}$ constructor.

$$\textsf{\textbf{is\_t\_inside}} \doteq \exists\,(\mathsf{has\_area})(\mathsf{has\_area}) \,.\, \mathsf{t\_inside}$$

$$\textsf{\textbf{hh\_border\_district}} \doteq \mathsf{district\_of\_hh} \sqcap \exists\, \mathsf{is\_t\_inside} \,.\, \mathsf{federal\_state\_hh}$$

The concept $\mathsf{hh\_border\_district}$ is discussed as an example for the use of the role-forming predicate restriction introduced by $\mathsf{is\_t\_inside}$. The associated polygon of any individual that is subsumed by this concept has to be in the $\mathsf{t\_inside}$ relationship with another polygon that, in turn, is referred to by an instance that is subsumed by the concept $\mathsf{federal\_state\_hh}$.

While the subsumption relationships discussed above are quite obvious, the advantages of TBox reasoning with spatial relations become apparent if we consider more complex cases, e.g. the following axiom is computed by other (non-DL) components and added to our TBox (e.g. imagine a scenario employing machine learning techniques). The restrictedness criterion is fulfilled.

$$\begin{aligned}\textsf{\textbf{unknown}} \doteq\ & \mathsf{district\_of\_hh} \sqcap \\ & \exists\,(\exists\,(\mathsf{has\_area})(\mathsf{has\_area}) \,.\, \mathsf{spatially\_related}) \,.\, \mathsf{federal\_state\_hh} \sqcap \\ & \exists\,(\exists\,(\mathsf{has\_area})(\mathsf{has\_area}) \,.\, \mathsf{touching}) \,.\, \mathsf{federal\_state\_sh}\end{aligned}$$

If the polygon of $\mathsf{district\_of\_hh}$ touches the polygon of $\mathsf{federal\_state\_sh}$, then the polygon of $\mathsf{district\_of\_hh}$ is also $\mathsf{t\_inside}$ the polygon of $\mathsf{federal\_state\_hh}$. Therefore, it can be proven that $\mathsf{unknown}$ is subsumed by $\mathsf{hh\_border\_district}$ (see the next section). The spatial constellation defined by the concept $\mathsf{unknown}$ could also be characterized as a "Hamburg border district to Schleswig-Holstein." Note however, if $\mathsf{district\_of\_hh}$ had only been defined by the term $\exists\, \mathsf{has\_area} \,.\, \mathsf{g\_inside}_{\mathsf{p_2}}$ (see above), $\mathsf{unknown}$ would *not* have been subsumed by $\mathsf{hh\_border\_district}$ because an abstract individual whose associated polygon had been *equal* to $\mathsf{p_2}$ would have been a member of $\mathsf{unknown}$ but not a member of $\mathsf{hh\_border\_district}$.

3.5 Verifying Satisfiability: An Extended Example

We illustrate the satisfiability problem for an $\mathcal{ALCRP}(\mathcal{D}_{\mathcal{P}})$ concept with the example from the previous section. In order to prove that the concept $\mathsf{unknown}$ is subsumed by $\mathsf{hh\_border\_district}$, the tableau prover constructs an initial ABox and derives that every ABox in the set of ABoxes obtained by applying a set of rules will be "obviously contradictory," i.e. it will contain a clash. The rules are described in detail in [Lutz, Haarslev, and Möller 1997]. For the reader's convenience they are repeated here in the appendix (see Section A.1 and Section A.2). Please note that the rules can be applied in arbitrary order but in the following we rely on an manually defined ordering.

We start with the ABox $\mathcal{A}_1$ and expand in ABox $\mathcal{A}_2$ the concept names from ABox $\mathcal{A}_1$.

$$\mathcal{A}_1 := \left\{ \mathsf{x} : \mathsf{unknown} \sqcap \neg \mathsf{hh\_border\_district} \right\}$$

$$\mathcal{A}_2 := \left\{ \begin{array}{l} \mathsf{x} : \mathsf{district\_of\_hh} \sqcap \exists\, \mathsf{is\_spatially\_related} \,.\, \mathsf{federal\_state\_hh} \sqcap \\ \quad\;\; \exists\, \mathsf{is\_touching} \,.\, \mathsf{federal\_state\_sh} \sqcap \\ \quad\;\; \neg(\mathsf{district\_of\_hh} \sqcap \exists\, \mathsf{is\_t\_inside} \,.\, \mathsf{federal\_state\_hh}) \end{array} \right\}$$

If we fully expand the concept terms, we get the following ABox.

$$\mathcal{A}_3 := \left\{ \begin{array}{l} \mathsf{x} : \exists\, \mathsf{has\_area} \,.\, \mathsf{g\_inside}_{\mathsf{p_2}} \sqcap \exists\, \mathsf{has\_area} \,.\, \neg \mathsf{equal}_{\mathsf{p_2}} \sqcap \\ \quad\;\; \exists\, \mathsf{is\_spatially\_related} \,.\, \exists\, \mathsf{has\_area} \,.\, \mathsf{equal}_{\mathsf{p_2}} \sqcap \\ \quad\;\; \exists\, \mathsf{is\_touching} \,.\, \exists\, \mathsf{has\_area} \,.\, \mathsf{equal}_{\mathsf{p_4}} \sqcap \\ \quad\;\; \neg(\exists\, \mathsf{has\_area} \,.\, \mathsf{g\_inside}_{\mathsf{p_2}} \sqcap \exists\, \mathsf{has\_area} \,.\, \neg \mathsf{equal}_{\mathsf{p_2}} \sqcap \\ \quad\;\;\;\; \exists\, \mathsf{is\_t\_inside} \,.\, \exists\, \mathsf{has\_area} \,.\, \mathsf{equal}_{\mathsf{p_2}}) \end{array} \right\}$$

Then, we transform this ABox in negation normal form.

$$\mathcal{A}_4 := \left\{\begin{array}{l} \mathsf{x} : \exists\, \mathsf{has\_area} \,.\, \mathsf{g\_inside}_{\mathsf{p}_2} \sqcap \exists\, \mathsf{has\_area} \,.\, \neg \mathsf{equal}_{\mathsf{p}_2} \sqcap \\ \quad \exists\, \mathsf{is\_spatially\_related} \,.\, \exists\, \mathsf{has\_area} \,.\, \mathsf{equal}_{\mathsf{p}_2} \sqcap \\ \quad \exists\, \mathsf{is\_touching} \,.\, \exists\, \mathsf{has\_area} \,.\, \mathsf{equal}_{\mathsf{p}_4} \sqcap \\ \quad (\exists\, \mathsf{has\_area} \,.\, \neg \mathsf{g\_inside}_{\mathsf{p}_2} \sqcup \forall\, \mathsf{has\_area} \,.\, \top \sqcup \exists\, \mathsf{has\_area} \,.\, \mathsf{equal}_{\mathsf{p}_2} \sqcup \\ \quad\; \forall\, \mathsf{is\_t\_inside} \,.\, (\exists\, \mathsf{has\_area} \,.\, \neg \mathsf{equal}_{\mathsf{p}_2} \sqcup \forall\, \mathsf{has\_area} \,.\, \top)) \end{array}\right\}$$

We already proved that the TBox containing the axioms introduced in Section 3.2 complies to the restrictedness criterion. Therefore, the ABox $\mathcal{A}_4$ is based on a restricted terminology and we are allowed to apply the ABox rules of the calculus for $\mathcal{ALCRP}(\mathcal{D})$ (see [Lutz, Haarslev, and Möller 1997]). First, we apply the *and rule* ($\mathbf{R}\sqcap$) and get the ABox $\mathcal{A}_6$. For convenience we use an auxiliary ABox $\mathcal{A}_5$.

$$\mathcal{A}_5 := \left\{\begin{array}{l} \mathsf{x} : \exists\, \mathsf{has\_area} \,.\, \mathsf{g\_inside}_{\mathsf{p}_2} \\ \mathsf{x} : \exists\, \mathsf{has\_area} \,.\, \neg \mathsf{equal}_{\mathsf{p}_2} \\ \mathsf{x} : \exists\, \mathsf{is\_spatially\_related} \,.\, \exists\, \mathsf{has\_area} \,.\, \mathsf{equal}_{\mathsf{p}_2} \\ \mathsf{x} : \exists\, \mathsf{is\_touching} \,.\, \exists\, \mathsf{has\_area} \,.\, \mathsf{equal}_{\mathsf{p}_4} \end{array}\right\}$$

$$\mathcal{A}_6 := \mathcal{A}_5 \cup \left\{\begin{array}{l} \mathsf{x} : (\exists\, \mathsf{has\_area} \,.\, \neg \mathsf{g\_inside}_{\mathsf{p}_2} \sqcup \forall\, \mathsf{has\_area} \,.\, \top \sqcup \\ \qquad \exists\, \mathsf{has\_area} \,.\, \mathsf{equal}_{\mathsf{p}_2} \sqcup \\ \qquad \forall\, \mathsf{is\_t\_inside} \,.\, (\exists\, \mathsf{has\_area} \,.\, \neg \mathsf{equal}_{\mathsf{p}_2} \sqcup \forall\, \mathsf{has\_area} \,.\, \top)) \end{array}\right\}$$

Afterwards we obtain four alternative ABoxes ($\mathcal{A}_7$ - $\mathcal{A}_{10}$) by resolving the disjunctions in ABox $\mathcal{A}_6$ and by applying the *exists-in over predicates rule* ($\mathbf{R}\exists\mathbf{P}$) and/or the *all rule* ($\mathbf{R}\forall\mathbf{C}$).

$$\mathcal{A}_7 := \mathcal{A}_5 \cup \left\{\begin{array}{c} \mathsf{x} : \exists\, \mathsf{has\_area} \,.\, \neg \mathsf{g\_inside}_{\mathsf{p}_2} \\ (\mathsf{x}, \mathsf{q}_2) : \mathsf{has\_area},\ \mathsf{q}_2 : \mathsf{g\_inside}_{\mathsf{p}_2},\ \mathsf{q}_2 : \neg \mathsf{g\_inside}_{\mathsf{p}_2} \end{array}\right\}$$

The ABox $\mathcal{A}_7$ contains a *concrete domain clash* because the concrete individual q_2 can not satisfy the conjunction $(\mathsf{g\_inside}_{\mathsf{p}_2} \wedge \neg \mathsf{g\_inside}_{\mathsf{p}_2})$.

$$\mathcal{A}_8 := \mathcal{A}_5 \cup \left\{\begin{array}{c} \mathsf{x} : \forall\, \mathsf{has\_area} \,.\, \top \\ (\mathsf{x}, \mathsf{q}_2) : \mathsf{has\_area},\ \mathsf{q}_2 : \mathsf{g\_inside}_{\mathsf{p}_2},\ \mathsf{q}_2 : \top \end{array}\right\}$$

The ABox $\mathcal{A}_8$ contains an *all domain clash* because the concrete individual q_2 can not be a member of both the abstract ($\top$) and the concrete (e.g. $\mathsf{g\_inside}_{\mathsf{p}_2}$) domain.

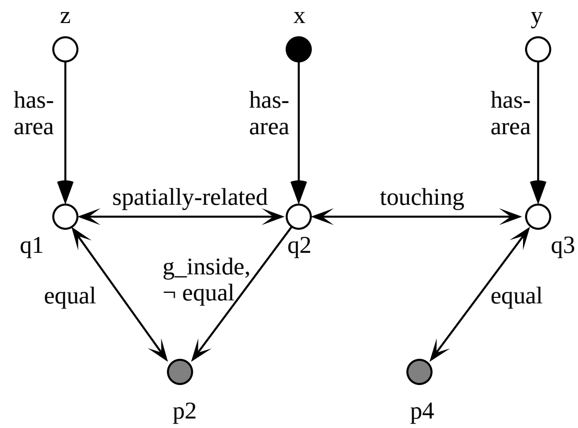


Figure 6: Initial constraint network corresponding to ABox $\mathcal{A}_{12}$. For symmetric relations the arrows point in both directions. Inverse relations have been omitted.

$$\mathcal{A}_9 := \mathcal{A}_5 \cup \left\{ \begin{array}{c} \mathsf{x} : \exists\, \mathsf{has\_area} \,.\, \mathsf{equal}_{\mathsf{p_2}} \\ (\mathsf{x}, \mathsf{q_2}) : \mathsf{has\_area},\ \mathsf{q_2} : \neg \mathsf{equal}_{\mathsf{p_2}},\ \mathsf{q_2} : \mathsf{equal}_{\mathsf{p_2}} \end{array} \right\}$$

The ABox $\mathcal{A}_9$ contains a *concrete domain clash* because the concrete individual $\mathsf{q_2}$ can not satisfy the conjunction $(\mathsf{equal}_{\mathsf{p_2}} \wedge \neg \mathsf{equal}_{\mathsf{p_2}})$.

$$\mathcal{A}_{10} := \mathcal{A}_5 \cup \left\{ \mathsf{x} : \forall\, \mathsf{is\_t\_inside} \,.\, (\exists\, \mathsf{has\_area} \,.\, \neg \mathsf{equal}_{\mathsf{p_2}} \sqcup \forall\, \mathsf{has\_area} \,.\, \top) \right\}$$

The ABox $\mathcal{A}_{10}$ is subject to further rule application. We apply the *exists-in over predicates rule* (**R∃P**) and the *role-forming exists-in over predicates rule* (**Rr∃P**) and create two abstract domain individuals y and z such that z is a filler of the role $\mathsf{is\_spatially\_related}$ and y is a filler of $\mathsf{is\_touching}$. We also create three concrete domain individuals $\mathsf{q_2}$, $\mathsf{q_3}$, and $\mathsf{q_1}$ that are associated with their corresponding abstract individuals via the attribute $\mathsf{has\_area}$. The rules also establish spatial relations that have to hold between concrete individuals. After firing all applicable rules except the *choose rule* (**RChoose**), we get ABox $\mathcal{A}_{12}$ whose spatial constraints are illustrated in Figure 6. For convenience we use an auxiliary ABox $\mathcal{A}_{11}$.

$$\mathcal{A}_{11} := \mathcal{A}_5 \cup \left\{ \begin{array}{l} (\mathsf{x}, \mathsf{q_2}) : \mathsf{has\_area},\ \mathsf{q_2} : \mathsf{g\_inside}_{\mathsf{p_2}},\ \mathsf{q_2} : \neg \mathsf{equal}_{\mathsf{p_2}} \\ (\mathsf{x}, \mathsf{y}) : \exists\, (\mathsf{has\_area})(\mathsf{has\_area}) \,.\, \mathsf{touching} \\ (\mathsf{q_2}, \mathsf{q_3}) : \mathsf{touching} \\ (\mathsf{y}, \mathsf{q_3}) : \mathsf{has\_area},\ \mathsf{q_3} : \mathsf{equal}_{\mathsf{p_4}} \\ (\mathsf{x}, \mathsf{z}) : \exists\, (\mathsf{has\_area})(\mathsf{has\_area}) \,.\, \mathsf{spatially\_related} \\ (\mathsf{q_2}, \mathsf{q_1}) : \mathsf{spatially\_related} \end{array} \right\}$$

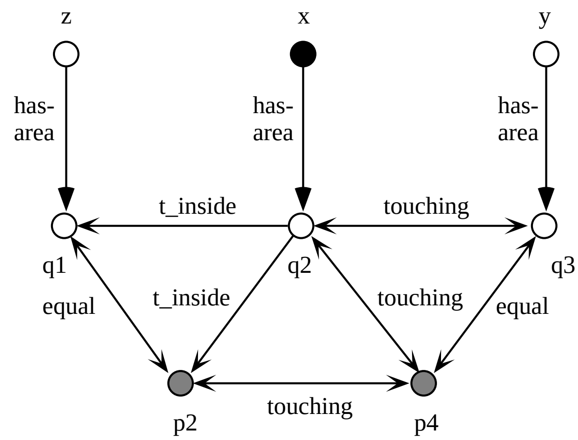


Figure 7: Final constraint network (most implicit constraints added) that corresponds to ABox $\mathcal{A}_{13}$. For symmetric relations the arrows point in both directions. Inverse relations have been omitted.

$$\mathcal{A}_{12} := \mathcal{A}_{11} \cup \left\{ \begin{array}{r} \mathsf{x} : \forall\, \mathsf{is\_t\_inside} \,.\, (\exists\, \mathsf{has\_area} \,.\, \neg \mathsf{equal}_{\mathsf{p_2}} \sqcup \forall\, \mathsf{has\_area} \,.\, \top) \\ (\mathsf{z}, \mathsf{q_1}) : \mathsf{has\_area},\ \mathsf{q_1} : \mathsf{equal}_{\mathsf{p_2}} \qquad\qquad\qquad\qquad \end{array} \right\}$$

In the next step, the *choose rule* (**RChoose**) has to decide whether the relation $\mathsf{t\_inside}$ or its negation holds between any two concrete individuals in ABox $\mathcal{A}_{12}$. Without loss of generality we can assume that only the following two alternative ABoxes ($\mathcal{A}_{13}$, $\mathcal{A}_{15}$) are created by selecting the concrete individuals $\mathsf{q_1}, \mathsf{q_2}$.

$$\mathcal{A}_{13} := \mathcal{A}_{11} \cup \left\{ \begin{array}{l} \qquad \mathsf{x} : \forall\, \mathsf{is\_t\_inside} \,.\, (\exists\, \mathsf{has\_area} \,.\, \neg \mathsf{equal}_{\mathsf{p_2}} \sqcup \forall\, \mathsf{has\_area} \,.\, \top) \\ \ (\mathsf{z}, \mathsf{q_1}) : \mathsf{has\_area},\ \mathsf{q_1} : \mathsf{equal}_{\mathsf{p_2}} \\ (\mathsf{q_2}, \mathsf{q_1}) : \mathsf{t\_inside} \end{array} \right\}$$

The spatial constraints that have to hold in ABox $\mathcal{A}_{13}$ are illustrated in Figure 7. This ABox assumes that $\mathsf{t\_inside}(\mathsf{q_2}, \mathsf{q_1})$ holds and makes the implicit spatial constraints from ABox $\mathcal{A}_{12}$ explicit (see also Figure 6). Due to the last assertion in ABox $\mathcal{A}_{13}$, now the *all rule*(**R**$\forall$**C**) is applicable to $\mathsf{is\_t\_inside}$ and creates the ABox $\mathcal{A}_{14}$.

$$\mathcal{A}_{14} := \mathcal{A}_{11} \cup \left\{ \begin{array}{l} \qquad \mathsf{x} : \forall\, \mathsf{is\_t\_inside} \,.\, (\exists\, \mathsf{has\_area} \,.\, \neg \mathsf{equal}_{\mathsf{p_2}} \sqcup \forall\, \mathsf{has\_area} \,.\, \top) \\ \ (\mathsf{z}, \mathsf{q_1}) : \mathsf{has\_area},\ \mathsf{q_1} : \mathsf{equal}_{\mathsf{p_2}} \\ (\mathsf{q_2}, \mathsf{q_1}) : \mathsf{t\_inside} \\ \qquad \mathsf{z} : \exists\, \mathsf{has\_area} \,.\, \neg \mathsf{equal}_{\mathsf{p_2}} \sqcup \forall\, \mathsf{has\_area} \,.\, \top \end{array} \right\}$$

Caused by the disjunction in the last assertion, we get two descendants of ABox $\mathcal{A}_{14}$. However, both descendants contain clashes for the concrete individual q_1 (i.e. either an *all domain clash* or a *concrete domain clash*) and eliminate this branch.

It remains the ABox $\mathcal{A}_{15}$ as the second alternative descendant of ABox $\mathcal{A}_{12}$. The ABox $\mathcal{A}_{15}$ assumes that $\neg\mathsf{t\_inside}(\mathsf{q}_2, \mathsf{q}_1)$ holds. It has all spatial constraints expanded.

$$\mathcal{A}_{15} := \mathcal{A}_5 \cup \left\{ \begin{array}{rl} (\mathsf{x}, \mathsf{q}_2) : & \mathsf{has\_area},\ \ \mathsf{q}_2 : \mathsf{g\_inside}_{\mathsf{p}_2},\ \ \mathsf{q}_2 : \neg\mathsf{equal}_{\mathsf{p}_2} \\ (\mathsf{x}, \mathsf{y}) : & \exists\,(\mathsf{has\_area})(\mathsf{has\_area})\,.\,\mathsf{touching} \\ (\mathsf{q}_2, \mathsf{q}_3) : & \mathsf{touching} \\ (\mathsf{y}, \mathsf{q}_3) : & \mathsf{has\_area},\ \ \mathsf{q}_3 : \mathsf{equal}_{\mathsf{p}_4} \\ (\mathsf{x}, \mathsf{z}) : & \exists\,(\mathsf{has\_area})(\mathsf{has\_area})\,.\,\mathsf{spatially\_related} \\ (\mathsf{q}_2, \mathsf{q}_1) : & \mathsf{spatially\_related} \\ \mathsf{x} : & \forall\,\mathsf{is\_t\_inside}\,.\,(\exists\,\mathsf{has\_area}\,.\,\neg\mathsf{equal}_{\mathsf{p}_2} \sqcup \forall\,\mathsf{has\_area}\,.\,\top) \\ (\mathsf{z}, \mathsf{q}_1) : & \mathsf{has\_area},\ \ \mathsf{q}_1 : \mathsf{equal}_{\mathsf{p}_2} \\ (\mathsf{q}_2, \mathsf{q}_1) : & \neg\mathsf{t\_inside} \end{array} \right\}$$

ABox $\mathcal{A}_{15}$ contains a *concrete domain clash*. In order to check for concrete domain clashes, the calculus for $\mathcal{ALCRP}(\mathcal{D}_{\mathcal{P}})$ invokes the satisfiability test for $\mathcal{D}_{\mathcal{P}}$ with the conjunction $\mathcal{C}_0$ of spatial predicate terms. This conjunction represents all spatial relations that have to hold between the concrete individuals in ABox $\mathcal{A}_{15}$.

$$\mathcal{C}_0 := \left\{ \begin{array}{c} \mathsf{g\_inside}(\mathsf{q}_2, \mathsf{p}_2) \wedge \neg\mathsf{equal}(\mathsf{q}_2, \mathsf{p}_2) \wedge \mathsf{touching}(\mathsf{q}_2, \mathsf{q}_3) \wedge \mathsf{equal}(\mathsf{q}_3, \mathsf{p}_4) \wedge \\ \mathsf{spatially\_related}(\mathsf{q}_2, \mathsf{q}_1) \wedge \mathsf{equal}(\mathsf{q}_1, \mathsf{p}_2) \wedge \neg\mathsf{t\_inside}(\mathsf{q}_2, \mathsf{q}_1) \end{array} \right\}$$

Some of the terms in the conjunction $\mathcal{C}_0$ contain references to the two concrete polygons p_2 and p_4 that have known positions (see Figure 5). During the satisfiability test the elementary spatial relation that holds between these polygons is computed. This results in the conjunction $\mathcal{C}_0 \wedge \mathsf{touching}(\mathsf{p}_2, \mathsf{p}_4)$ that is unsatisfiable in our spatial domain $\mathcal{D}_{\mathcal{P}}$.

The examples in this section demonstrate that reasoning about consistency and subsumption of TBox concepts is a nontrivial task. Sound and complete inference algorithms are a necessity in these application domain. In the next section we will extend the discussion to ABox reasoning and demonstrate the usefulness of $\mathcal{ALCRP}(\mathcal{D})$ for environmental planning problems.

3.6 ABox Reasoning for GIS Applications Concerning Environmental Planning

Environmental information systems can benefit from spatioterminological reasoning in many ways. First, queries can be posed as concepts composed

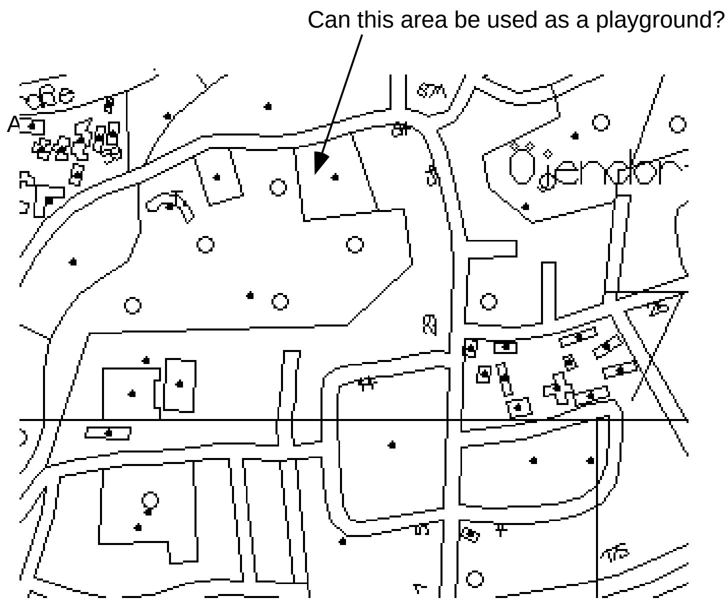


Figure 8: A clip from the Oejendorf map (see text).

with respect to an ontology underlying a certain TBox. The description logic reasoner will answer the query by finding all instances that are subsumed by the query concept. Second, the ability to test ABox instances for consistency can be used to implement a planning system which is based on hypothesize and test strategies. For instance, let us assume the following TBox fragment is used to model domain objects shown in the map of Figure 1. The restrictedness criterion for TBoxes is fulfilled.

$$\textbf{is\_touching} \doteq \exists\,(\mathsf{has\_area})(\mathsf{has\_area})\,.\,\mathsf{touching}$$
$$\textbf{is\_connected} \doteq \exists\,(\mathsf{has\_area})(\mathsf{has\_area})\,.\,\mathsf{connected}$$
$$\textbf{dangerous\_object} \doteq \mathsf{freeway} \sqcup \mathsf{chemical\_plant} \sqcup \ldots$$
$$\textbf{insecure\_object} \doteq \mathsf{dangerous\_object} \sqcup (\mathsf{unfenced\_object} \sqcap \exists\,\mathsf{is\_connected}\,.\,\mathsf{dangerous\_object})$$
$$\textbf{secure\_playground} \doteq \mathsf{playground} \sqcap \forall\,\mathsf{is\_touching}\,.\,\neg\mathsf{insecure\_object}$$

We suppose that the objects depicted in Figure 8 are represented in an ABox as instances of general concepts such as building, region, road etc. These concepts directly model the information given in the underlying database. The geometry is assumed to be represented by corresponding polygons as fillers for the attribute has_area as required by the ontology underlying our TBox domain model. In order to check whether, for instance, the region area_1 which is indicated by an arrow in Figure 8 can be used as a secure playground, we simply add the ABox axiom area_1 : secure_playground.[3] If the description logic reasoner computes that the ABox is consistent, all constraints imposed by secure_playground are satisfied. Hence, according to our (simple) domain model, area_1 might be suitable for a playground. Note that these inferences are correct only if the semantics of spatial relations is adequately considered by the TBox reasoner as described before.

4 Conclusion

Based on the description logic language $\mathcal{ALCRP}(\mathcal{D})$, we have shown how spatial and terminological reasoning can be combined in the TBox. Thus, the fruitful research on description logics has been extended to cope with qualitative spatial relations and quantitative spatial data. One of the main ideas is to introduce constructors for roles whose definitions are based on properties of concrete objects. The abstract domain is used to represent terminological knowledge about spatial domains on an abstract logical level. The concrete domain (space domain) extends the abstract domain and provides access to spatial reasoning algorithms. We have shown that the concrete domain given by polygons and predicates about spatial relations is admissible. Our approach for testing satisfiability of finite conjunctions relies on current work in qualitative spatial reasoning theory [Renz and Nebel 1997]. If required, even quantitative data (concrete polygons) are considered by applying algorithms known from computational geometry (e.g. see [de Berg 1997]). Techniques for spatial indexing can easily be integrated. Note that although realizability makes reasoning much harder [Grigni, Papadias, and Papadimitriou 1995], in some cases where quantitative information is available for polygons, inconsistencies caused by relational structures can be easily detected by applying polynomial algorithms.

We admit that the $\mathcal{ALCRP}(\mathcal{D})$ restrictedness criterion for terminologies does impose tight constraints on modeling spatioterminological structures. However, in a specific application, many interesting concepts can be represented in a TBox with the additional advantage of having a decidable satisfiability algorithm. Using an example from a GIS domain, we have demonstrated that, on the one hand, topological relations directly influence the kind of conceptual or terminological knowledge that can (and must) be derived by a formal inference engine. On the other hand, assertions about concepts restrict the set of possible spatial relations between different individuals.

Considering the general mechanism for integrating concrete domains, it becomes clear that another instance of $\mathcal{ALCRP}(\mathcal{D})$ can deal with temporal

[3] ABox statements can be automatically generated by a graphical interface (see [Haarslev and Wessel 1996]).

relations. Corresponding constraint satisfaction algorithms known from the literature (e.g. [Ladkin and Reinefeld 1997]) can be employed. Future work will reveal the relationship between $\mathcal{ALCRP}(\mathcal{D})$ and, for instance, the temporal description logic developed in [Artale and Franconi 1997]. Although this approach does not impose restrictions on terminologies, it does not provide facilities for expressing value restrictions over complex relations. This is not a problem in $\mathcal{ALCRP}(\mathcal{D})$ as long as the terminology fulfills the restrictedness criterion. Defined qualitative relations that are "grounded" in quantitative data provide a bridge to conceptual knowledge and support more extensive reasoning services to be exploited for solving practical problems.

A The Calculus for $\mathcal{ALCRP}(\mathcal{D})$

The following two sections are excerpts taken from [Lutz, Haarslev, and Möller 1997]. They are repeated here for the convenience of the reader.

A.1 Completion Rules

Before the completion rules can be defined, we introduce some technical terms. Let $\mathcal{A}$ be an ABox, R be a role term, a and b be object names from O_A, γ be a symbol that is not element of O_D, u be a feature chain $f_1 \ldots f_k$, and let $u_1, \ldots, u_n$ and $v_1, \ldots, v_m$ (possibly with index) be arbitrary feature chains. For convenience we define three functions as follows:

$$
\mathit{filler}_{\mathcal{A}}(a,u) := \begin{cases} x \text{ where } x \in \mathsf{O}_D \text{ such that} \\ \quad \exists b_1, \ldots, b_{k-1} \in \mathsf{O}_A : \\ \quad\quad ((a,b_1) : f_1 \in \mathcal{A}, \ldots, (b_{k-1}, x) : f_k \in \mathcal{A}) \\ \gamma \text{ if no such } x \text{ exists.} \end{cases}
$$

$$
\mathit{createchain}_{\mathcal{A}}(a,x,u) := \{(a,c_1) : f_1, \ldots, (c_{k-1},x) : f_k\} \\ \text{where } c_1, \ldots, c_{k-1} \in \mathsf{O}_A \text{ are not used in } \mathcal{A}.
$$

$$
\mathit{related}_{\mathcal{A}}(a,b,R) := \begin{cases} \mathit{true} & \text{if } (a,b) : R \in \mathcal{A} \\ \mathit{true} & \text{if } R \text{ is of the form } \exists(u_1, \ldots, u_n)(v_1, \ldots, v_m).P, \\ & \text{and } \exists x_1, \ldots, x_n, y_1, \ldots, y_m \in \mathsf{O}_D \text{ such that} \\ & \mathit{filler}_{\mathcal{A}}(a,u_1) = x_1, \ \ldots \ , \mathit{filler}_{\mathcal{A}}(a,u_n) = x_n, \\ & \mathit{filler}_{\mathcal{A}}(b,v_1) = y_1, \ \ldots \ , \mathit{filler}_{\mathcal{A}}(b,v_m) = y_m, \\ & (x_1, \ldots, x_n, y_1, \ldots, y_m) : P \in \mathcal{A} \\ \mathit{false} & \text{otherwise} \end{cases}
$$

Let $\mathcal{A}$ be an ABox, f be a feature name, a, b, c be object names from O_A, and x, y be object names from O_D. If $\mathcal{A}$ contains the constraints $(a,b) : f$ and $(a,c) : f$ (resp. $(a,x) : f$ and $(a,y) : f$) then this pair of constraints is called a *fork* in $\mathcal{A}$. Since f is interpreted as a partial function, b and c (resp. x and y) have to be interpreted as the same objects. Each ABox is checked for forks immediately after a completion rule was applied. If a fork is detected, all occurrences of c in $\mathcal{A}$ are replaced by b (resp. y by x). Before any rule is applied to the initial ABox $\mathcal{A}_0$, any forks in $\mathcal{A}_0$ have to be eliminated. It is easy to prove that fork elimination preserves (in)consistency by showing that a model $\mathcal{I}$ for an ABox $\mathcal{A}$ is also a model for an ABox $\mathcal{A}'$ which is obtained from $\mathcal{A}$ by fork elimination.

Definition 1
The following *completion rules* will replace an ABox $\mathcal{A}$ by an ABox $\mathcal{A}'$ or by two ABoxes $\mathcal{A}'$ and $\mathcal{A}''$ (*descendants* of $\mathcal{A}$). In the following C and D

denote concept terms, R denotes a role term, and P denotes a predicate name from $\Phi_{\mathcal{D}}$. Let $f_1, \ldots, f_n$ as well as $g_1, \ldots, g_n$ denote feature names, and $u_1, \ldots, u_m$ denote feature chains. a and b denote object names from O_A.

R⊓ The conjunction rule.
Premise: $a : C \sqcap D \in \mathcal{A}, \quad a : C \notin \mathcal{A} \vee a : D \notin \mathcal{A}$
Consequence: $\mathcal{A}' = \mathcal{A} \cup \{a : C,\ a : D\}$

R⊔ The disjunction rule.
Premise: $a : C \sqcup D \in \mathcal{A}, \quad a : C \notin \mathcal{A} \wedge a : D \notin \mathcal{A}$
Consequence: $\mathcal{A}' = \mathcal{A} \cup \{a : C\},\ \mathcal{A}'' = \mathcal{A} \cup \{a : D\}$

R∃C The concept exists restriction rule.
Premise: $a : \exists R.C \in \mathcal{A}, \quad \neg\exists b \in \mathsf{O}_A \colon (\mathit{related}_{\mathcal{A}}(a, b, R) \wedge b : C \in \mathcal{A})$
Consequence: $\mathcal{A}' = \mathcal{A} \cup \{(a, b) : R\ ,\ b : C\}$ where $b \in \mathsf{O}_A$ is not used in $\mathcal{A}$.
This rule may create a fork if R is a feature.

R∀C The concept value restriction rule.
Premise: $a : \forall R.C \in \mathcal{A}, \quad \exists b \in \mathsf{O}_A \colon (\mathit{related}_{\mathcal{A}}(a, b, R), \wedge b : C \notin \mathcal{A})$
Consequence: $\mathcal{A}' = \mathcal{A} \cup \{b : C\}$

R∃P The predicate restriction rule.
Premise: $a : \exists u_1, \ldots, u_n.P \in \mathcal{A}, \neg\exists x_1, \ldots, x_n \in \mathsf{O}_D \colon$
$(\mathit{filler}_{\mathcal{A}}(a, u_1) = x_1 \wedge \ldots \wedge \mathit{filler}_{\mathcal{A}}(a, u_n) = x_n \wedge$
$(x_1, \ldots, x_n) : P \in \mathcal{A})$
Consequence: $\mathcal{A}' = \mathcal{A} \cup \{(x_1, \ldots, x_n) : P\} \cup$
$\mathit{createchain}_{\mathcal{A}}(a, x_1, u_1) \cup \ldots \cup \mathit{createchain}_{\mathcal{A}}(a, x_n, u_n)$
where the $x_i \in \mathsf{O}_D$ are not used in $\mathcal{A}$.
This rule may create forks.

Rr∃P The role-forming predicates restriction rule.
Premise: $(a, b) : \exists(u_1, \ldots, u_n)(v_1, \ldots, v_m).P \in \mathcal{A}$,
$\neg\exists x_1, \ldots, x_n, y_1, \ldots, y_m \in \mathsf{O}_D \colon$
$(\mathit{filler}_{\mathcal{A}}(a, u_1) = x_1 \wedge \ldots \wedge \mathit{filler}_{\mathcal{A}}(a, u_n) = x_n \wedge$
$\mathit{filler}_{\mathcal{A}}(b, v_1) = y_1 \wedge \ldots \wedge \mathit{filler}_{\mathcal{A}}(b, v_m) = y_m \wedge$
$(x_1, \ldots, x_n, y_1, \ldots, y_m) : P \in \mathcal{A})$
Consequence: $\mathcal{A}' = \mathcal{A} \cup \{(x_1, \ldots, x_n, y_1, \ldots, y_m) : P\} \cup$
$\mathit{createchain}_{\mathcal{A}}(a, x_1, u_1) \cup \ldots \cup \mathit{createchain}_{\mathcal{A}}(a, x_n, u_n) \cup$
$\mathit{createchain}_{\mathcal{A}}(b, y_1, v_1) \cup \ldots \cup \mathit{createchain}_{\mathcal{A}}(b, y_m, v_m)$
where the $x_i \in \mathsf{O}_D$ and $y_i \in \mathsf{O}_D$ are not used in $\mathcal{A}$.
This rule may create forks.

RChoose The choose rule.
Premise: $a : \forall(\exists(u_1, \ldots, u_n)(v_1, \ldots, v_m).P).C \in \mathcal{A}$,
$\exists b \in \mathsf{O}_A,\ x_1, \ldots, x_n, y_1, \ldots, y_m \in \mathsf{O}_D \colon$

$$\begin{aligned}&(\mathit{filler}_{\mathcal{A}}(a, u_1) = x_1 \wedge \ \ldots \ \wedge \mathit{filler}_{\mathcal{A}}(a, u_n) = x_n \wedge \\ &\ \mathit{filler}_{\mathcal{A}}(b, v_1) = y_1 \wedge \ \ldots \ \wedge \mathit{filler}_{\mathcal{A}}(b, v_m) = y_m \wedge \\ &\ (x_1, \ldots, x_n, y_1, \ldots, y_m) : P \notin \mathcal{A} \wedge \\ &\ (x_1, \ldots, x_n, y_1, \ldots, y_m) : \overline{P} \notin \mathcal{A})\end{aligned}$$

Consequence: $\mathcal{A}' = \mathcal{A} \cup \{(x_1, \ldots, x_n, y_1, \ldots, y_m) : P\}$,
$\mathcal{A}'' = \mathcal{A} \cup \{(x_1, \ldots, x_n, y_1, \ldots, y_m) : \overline{P}\}$

A.2 Clash Rules

Termination of the algorithm applying the completion rules is proved in [Lutz, Haarslev, and Möller 1997]. The proof shows that after a finite number of rule applications a tree Υ of ABoxes is obtained for which one of the following conditions holds:

1. it contains an ABox $\mathcal{A}$ which is complete or
2. all leaf ABoxes in the tree contain a clash.

In both cases no more completion rules are applicable. In the following we formalize the notion "to contain a clash."

Definition 2
Let the same naming conventions be given as in Definition 1. Additionally, let f be a feature. An ABox $\mathcal{A}$ contains a *clash* if any of the following *clash triggers* are applicable:

Primitive Clash
$a : C \in \mathcal{A}, \quad a : \neg C \in \mathcal{A}$

Feature Domain Clash
$(a, x) : f \in \mathcal{A}, \quad (a, b) : f \in \mathcal{A}$

All Domain Clash
$(a, x) : f \in \mathcal{A}, \quad a : \forall f.C \in \mathcal{A}$

Concrete Domain Clash
$(x_1^{(1)}, \ldots, x_{n_1}^{(1)}) : P_1 \in \mathcal{A}, \ldots, (x_1^{(k)}, \ldots, x_{n_k}^{(k)}) : P_k \in \mathcal{A}$ and the corresponding conjunction $\bigwedge_{i=1}^{k} P_i(x^{(i)})$ is not satisfiable in $\mathcal{D}$. This can be decided because $\mathcal{D}$ is required to be admissible.

References

Artale, A. and E. Franconi (1997). A temporal description logic for reasoning about actions and plans. Preprint.

Baader, F. and P. Hanschke (1991a, August). A scheme for integrating concrete domains into concept languages. In *Twelfth International Conference on Artificial Intelligence, Darling Harbour, Sydney, Australia, Aug. 24-30, 1991*, pp. 452–457.

Baader, F. and P. Hanschke (1991b). A scheme for integrating concrete domains into concept languages. Technical Report DFKI-RR-91-10, German Center for AI (DFKI).

Bennett, B. (1995). Modal logics for qualitative spatial reasoning. *Bull. of the IGPL 3*, 1–22.

Borgo, S., N. Guarino, and C. Masolo (1996, November). A pointless theory of space based on strong connection and congruence. In L. Aiello, J. Doyle, and S. Shapiro (Eds.), *Fifth International Conference on Principles of Knowledge Representation, Cambridge, Mass., Nov. 5-8, 1996*, pp. 220–229.

Cohn, A., B. Bennett, J. Gooday, and N. Gotts (1997). Representing and reasoning with qualitative spatial relations. See Stock (1997), pp. 97–134.

de Berg, M. (1997). *Computational Geometry: Algorithms and Applications*. Springer Verlag, Berlin.

De Giacomo, G. and M. Lenzerini (1994). Boosting the correspondence between description logics and propositional dynamic logics. In *Proceedings of the AAAI'94*, pp. 205–212. AAAI Press / MIT Press.

Egenhofer, M. (1991, August). Reasoning about binary topological relations. In O. Günther and H.-J. Schek (Eds.), *Advances in Spatial Databases, Second Symposium, SSD'91, Zurich, Aug. 28-30, 1991*, Volume 525 of *Lecture Notes in Computer Science*, pp. 143–160. Springer Verlag, Berlin.

Grigni, M., D. Papadias, and C. Papadimitriou (1995, August). Topological inference. In C. Mellish (Ed.), *Fourteenth International Joint Conference on Artificial Intelligence, Montreal, Quebec, Canada, Aug. 20-25, 1995*, pp. 901–906.

Haarslev, V. (1995, September). Formal semantics of visual languages using spatial reasoning. In *1995 IEEE Symposium on Visual Languages, Darmstadt, Germany, Sep. 5-9*, pp. 156–163. IEEE Computer Society Press, Los Alamitos.

Haarslev, V. (1998). A fully formalized theory for describing visual notations. In K. Marriott and B. Meyer (Eds.), *Visual Language Theory.* Springer Verlag, Berlin. In press.

Haarslev, V. and R. Möller (1997, June). SBox: A qualitative spatial reasoner—progress report. In L. Ironi (Ed.), *11th International Workshop on Qualitative Reasoning, Cortona, Tuscany, Italy, June 3-6, 1997, Pubblicazioni N. 1036, Istituto di Analisi Numerica C.N.R. Pavia (Italy)*, pp. 105–113.

Haarslev, V., R. Möller, and C. Schröder (1994, September). Combining spatial and terminological reasoning. In B. Nebel and L. Dreschler-Fischer (Eds.), *KI-94: Advances in Artificial Intelligence – Proc. 18th German Annual Conference on Artificial Intelligence, Saarbrücken, Sept. 18–23, 1994*, Volume 861 of *Lecture Notes in Artificial Intelligence*, pp. 142–153. Springer Verlag, Berlin.

Haarslev, V. and M. Wessel (1996, September). GenEd—an editor with generic semantics for formal reasoning about visual notations. In *1996 IEEE Symposium on Visual Languages, Boulder, Colorado, USA, Sep. 3-6*, pp. 204–211. IEEE Computer Society Press, Los Alamitos.

Haarslev, V. and M. Wessel (1997, September). Querying GIS with animated spatial sketches. In *1997 IEEE Symposium on Visual Languages, Capri, Italy, Sep. 23-26*, pp. 197–204. IEEE Computer Society Press, Los Alamitos.

Ladkin, P. and A. Reinefeld (1997). Fast algebraic methods for interval constraint problems. *Annals of Mathematics and Artificial Intelligence 19*(3-4), 383–411.

Lutz, C., V. Haarslev, and R. Möller (1997). A concept language with role-forming predicate restrictions. Technical Report FBI-HH-M-276/97, University of Hamburg, Computer Science Department.

Lutz, C. and R. Möller (1997, September). Defined topological relations in description logics. See Rousset et al. (1997), pp. 15–19.

Nebel, B. (1995, September). Computational properties of qualitative spatial reasoning: First results. In I. Wachsmuth, C.-R. Rollinger, and W. Brauer (Eds.), *Proceedings, KI-95: Advances in Artificial Intelligence, 19th Annual German Conference on Artificial Intelligence, Bielefeld, Germany, Sept. 11-13*, Volume 981 of *Lecture Notes in Artificial Intelligence*, pp. 233–244. Springer Verlag, Berlin.

Pratt, I. and O. Lemon (1997). Ontologies for plane, polygonal mereotopology. Technical Report UMCS-97-1-1, University of Manchester, Department of Computer Science.

Pratt, I. and D. Schoop (1997). A complete axiom system for polygonal mereotopology of the real plane. Technical Report UMCS-97-2-2, University of Manchester, Department of Computer Science.

Randell, D., Z. Cui, and A. Cohn (1992, October). A spatial logic based on regions and connections. In B. Nebel, C. Rich, and W. Swartout (Eds.), *Principles of Knowledge Representation and Reasoning, Cambridge, Mass., Oct. 25-29, 1992*, pp. 165–176.

Renz, J. and B. Nebel (1997, August). On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connection calculus. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI'97)*, pp. 522–527.

Rousset et al., M.-C. (Ed.) (1997, September). *Proceedings of the International Workshop on Description Logics, DL'97, Sep. 27-29, 1997, Gif sur Yvette, France*. Universite Paris-Sud, Paris.

Schild, K. (1991, August). A correspondence theory for terminological logics: Preliminary report. In *Twelfth International Conference on Artificial Intelligence, Darling Harbour, Sydney, Australia, Aug. 24-30, 1991*, pp. 466–471.

Spanier, E. (1966). *Algebraic Topology*. McGraw-Hill Book Company, New York.

Stock, O. (Ed.) (1997). *Spatial and Temporal Reasoning*. Kluwer Academic Publishers, Dordrecht.