Foundations of Spatioterminological Reasoning with Description Logics

Volker Haarslev, Carsten Lutz, Ralf Möller
Computer Science Department
University of Hamburg
Vogt-Kölln-Str. 30
22527 Hamburg, Germany

Abstract

This paper presents a method for reasoning about spatial objects and their qualitative spatial relationships. In contrast to existing work, which mainly focusses on reasoning about qualitative spatial relations alone, we integrate quantitative and qualitative information with terminological reasoning. For spatioterminological reasoning we present the description logic $\mathcal{ALCRP}(\mathcal{D})$ and define an appropriate concrete domain $\mathcal{D}$ for polygons. The theory is motivated as a basis for knowledge representation and query processing in the domain of deductive 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 [11] and [13] we demonstrate the importance of terminological inferences with spatial relations in the domain of map databases and spatial query processing. In order to answer a query, concept terms are computed on the fly and must be checked for consistency. Furthermore, we assume that computed concept terms must be automatically inserted into the subsumption hierarchy of a knowledge base.

For formalizing reasoning about spatial structures many theories have been published (see e.g. [27] for an overview). Ignoring decidability, Borgo et al. [5] have developed a first order theory of space which formalizes different aspects such as mereology etc. An algebraic theory about space has been proposed in [21]. The well-known RCC theory [6] 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 [4]. 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, we extend the description logic $\mathcal{ALC}(\mathcal{D})$ [2]. 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 [26, 7], it is not clear how Bennett's modal logic approach can be extended to handle conceptual modeling and quantitative data.

Extending the work on $\mathcal{ALC}(\mathcal{D})$, we have developed a new description logic called $\mathcal{ALCRP}(\mathcal{D})$ [16] in order to provide a foundation for spatioterminological reasoning with description logics ($\mathcal{RP}$ stands for **r**ole definitions based on **p**redicates). The part of our theory dealing with spatial relations is based on a set of topological relations in analogy to Egenhofer [8] or RCC-8 [22]. 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, roles representing topological relations can be defined based on properties between concrete objects which, in turn, are associated to individuals via specific features. Thus, $\mathcal{ALCRP}(\mathcal{D})$ provides 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 [16]). 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, the restrictions on terminologies ensure decidability of the language.

In contrast to our earlier work presented in [13], [10] and [14] 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 (for a longer introduction, see [16], [18], [12], and [11]).

The paper is structured as follows. In the second section we introduce the language $\mathcal{ALCRP}(\mathcal{D})$. Unfortunately, without restrictions the satisfiability problem for $\mathcal{ALCRP}(\mathcal{D})$ is undecidable. We give a proof in the second section. Afterwards, we discuss the notion of a restricted terminology. A proof for the decidability of the satisfiability problem for restricted $\mathcal{ALCRP}(\mathcal{D})$ terminologies is sketched. Section 3 discusses an extended example for applying $\mathcal{ALCRP}(\mathcal{D})$ to spatioterminological reasoning. The conclusion in Section 4 demonstrates the general significance of $\mathcal{ALCRP}(\mathcal{D})$ by pointing out its applicability to other important reasoning problems such as terminological reasoning about temporal relations.

2 The Description Logic $\mathcal{ALCRP}(\mathcal{D})$

The description logic $\mathcal{ALC}(\mathcal{D})$ as defined in [2] supports the representation of concrete knowledge by incorporating concrete domains. The concrete domain approach can be used to integrate conceptual knowledge with concrete spatial knowledge. However, to define meaningful spatial concepts, it is also necessary to represent qualitative spatial relations and to exploit their various properties for reasoning. Since quantification over these relations is also needed in a description logic formalism, they should be represented as roles. Furthermore, the formalism has to ensure that knowledge about qualitative relations is represented in a way that is consistent with quantitative spatial knowledge. In this section, the description logic $\mathcal{ALCRP}(\mathcal{D})$ is introduced. $\mathcal{ALCRP}(\mathcal{D})$ extends $\mathcal{ALC}(\mathcal{D})$ by a role-forming operator which is based on concrete domain predicates. The new operator allows the definition of roles with very complex properties and provides a close coupling of roles with concrete domains.

2.1 The Formalism

First, the concept language is introduced. Concrete domains are defined by Baader and Hanschke [2] as follows.

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.

In the following we define role and concept terms in $\mathcal{ALCRP}(\mathcal{D})$.

Definition 2. Let R and F be disjoint sets of role and feature names[1], respectively. Any element of $\mathsf{R} \cup \mathsf{F}$ is an *atomic* role term. A composition of features (written $f_1 f_2 \cdots$) is called a feature chain. A simple feature can be viewed as a feature chain of length 1. 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. Let S be a role name and let T be a role term. Then $S \doteq T$ is a *terminological axiom*.

Definition 3. Let C be a set of concept names which is disjoint to R and F. Any element of C is a *concept term* (*atomic* concept term). If C and D are concept terms, R is a role term, $P \in \Phi_\mathcal{D}$ is a predicate name with arity n, and $u_1, \ldots, u_n$ are feature chains, then the following expressions are also concept terms: $C \sqcap D$ (*conjunction*), $C \sqcup D$ (*disjunction*), $\neg C$ (*negation*), $\exists R.C$ (*exists restriction*), $\forall R.C$ (*value restriction*), and $\exists u_1, \ldots, u_n.P$ (*predicate exists restriction*). For all kinds of exists and value restrictions, the role term or the 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 $\mathcal{T}$

[1] In the following, the notion *feature* is used as a synonym for feature name.

is 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.

We can now assign a meaning to $\mathcal{ALCRP}(\mathcal{D})$ concept terms by giving a set-theoretic semantics as usual.

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 $f_1^{\mathcal{I}}, \ldots, f_n^{\mathcal{I}}$. Let the symbols C, D, R, P, $u_1, \ldots, u_m$, and $v_1, \ldots, v_m$ be defined as in Definition 2 and 3, respectively. Then the interpretation function can be extended to arbitrary concept and role terms as follows:

$$
\begin{aligned}
(C \sqcap D)^{\mathcal{I}} &:= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &:= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} &:= \Delta_{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(\exists R.C)^{\mathcal{I}} &:= \{a \in \Delta_{\mathcal{I}} \mid \exists b \in \Delta_{\mathcal{I}} : \\
&\qquad (a, b) \in R^{\mathcal{I}}, b \in C^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} &:= \{a \in \Delta_{\mathcal{I}} \mid \forall b \in \Delta_{\mathcal{I}} : \\
&\qquad (a, b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\} \\
(\exists u_1, \ldots, u_n.P)^{\mathcal{I}} &:= \{a \in \Delta_{\mathcal{I}} \mid \exists x_1, \ldots, x_n \in \Delta_{\mathcal{D}} : \\
&\qquad (a, x_1) \in u_1^{\mathcal{I}}, \ldots, (a, x_n) \in u_n^{\mathcal{I}}, \\
&\qquad (x_1, \ldots, x_n) \in P^{\mathcal{D}}\} \\
(\exists (u_1, \ldots, u_n)(v_1, \ldots, v_m).P)^{\mathcal{I}} &:= \\
&\{(a, b) \in \Delta_{\mathcal{I}} \times \Delta_{\mathcal{I}} \mid \\
&\qquad \exists x_1, \ldots, x_n, y_1, \ldots, y_m \in \Delta_{\mathcal{D}} : \\
&\qquad (a, x_1) \in u_1^{\mathcal{I}}, \ldots, (a, x_n) \in u_n^{\mathcal{I}}, \\
&\qquad (b, y_1) \in v_1^{\mathcal{I}}, \ldots, (b, y_m) \in v_m^{\mathcal{I}}, \\
&\qquad (x_1, \ldots, x_n, y_1, \ldots, y_m) \in P^{\mathcal{D}}\}
\end{aligned}
$$

An interpretation $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$ iff it satisfies $A^{\mathcal{I}} = D^{\mathcal{I}}$ for all terminological axioms $A \doteq D$ in $\mathcal{T}$. A concept term C *subsumes* a concept term D w.r.t. a TBox $\mathcal{T}$ (written $D \preceq_{\mathcal{T}} C$), iff $D^{\mathcal{I}} \subseteq C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$. A concept term C is *satisfiable* w.r.t. a TBox $\mathcal{T}$ iff there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $D^{\mathcal{I}} \neq \emptyset$.

The basic reasoning service for a description logic formalism is computing the subsumption relationship. This inference is needed in the TBox to build a hierarchy of concepts w.r.t. specificity. Subsumption and satisfiability can be mutually reduced to each other since $C \preceq_{\mathcal{T}} D$ iff $C \sqcap \neg D$ is not satisfiable. The following definition introduces the assertional language of $\mathcal{ALCRP}(\mathcal{D})$, which can be used to represent knowledge about individual worlds.

Definition 5. Let O_D and O_A be two disjoint sets of object names. If C is a concept term, R a role term, f a feature name, P a predicate name with arity n, a and b are elements of O_A and x, and $x_1, \ldots, x_n$ are elements of O_D, then the following expressions are *assertional axioms*.

$$a : C,\ (a, b) : R,\ (a, x) : f,\ (x_1, \ldots, x_n) : P$$

A finite set of assertional axioms is called *ABox*. An *interpretation* for the concept language can be extended to the assertional language by additionally mapping every object name from O_A to a single element of $\Delta_{\mathcal{I}}$ and every object name from O_D to a single element from $\Delta_{\mathcal{D}}$. We assume that the unique name assumption does not hold, that is $a^{\mathcal{I}} = b^{\mathcal{I}}$ may hold even if $a \neq b$. An interpretation satisfies an assertional axiom

$$
\begin{gathered}
a : C \text{ iff } a^{\mathcal{I}} \in C^{\mathcal{I}}, \quad (a, b) : R \text{ iff } (a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}, \\
(a, x) : f \text{ iff } f^{\mathcal{I}}(a^{\mathcal{I}}) = x^{\mathcal{I}}, \\
(x_1, \ldots, x_n) : P \text{ iff } (x_1^{\mathcal{I}}, \ldots, x_n^{\mathcal{I}}) \in P^{\mathcal{D}}
\end{gathered}
$$

An interpretation is a *model* of an ABox $\mathcal{A}$ w.r.t. a TBox $\mathcal{T}$, iff it is a model of $\mathcal{T}$ and furthermore satisfies all assertional axioms in $\mathcal{A}$. An ABox is *consistent* w.r.t. a TBox $\mathcal{T}$ iff it has a model.

The ABox consistency problem is to decide whether a given ABox $\mathcal{A}$ is consistent w.r.t. a TBox $\mathcal{T}$. Satisfiability of concept terms can be reduced to ABox consistency as follows: A concept term C is satisfiable iff the ABox $\{a : C\}$ is consistent. In the next section we show that the reasoning problems just introduced are undecidable if the full logic $\mathcal{ALCRP}(\mathcal{D})$ is considered.

2.2 Undecidability of the Full Logic

In the following we prove the undecidability of reasoning in $\mathcal{ALCRP}(\mathcal{D})$ if no restrictions are posed on terminologies.

Theorem 6. *The problem whether an $\mathcal{ALCRP}(\mathcal{D})$ concept term C is satisfiable w.r.t. a TBox $\mathcal{T}$ is undecidable.*

Proof. The proof works by reducing the Post Correspondence Problem (PCP) to the satisfiability problem for $\mathcal{ALCRP}(\mathcal{D})$ concept terms and is similar to the reduction given in [3]. The PCP is defined as follows. Given a nonempty finite set $S = \{(l_i, r_i); i = 1, \ldots, m\}$, where the l_i and r_i are words over an alphabet Σ, a *solution of* S is a sequence of indices $i_1, ..., i_k$ with $k \geq 1$ such that the left concatenation $w_l = l_{i_1} \cdots l_{i_k}$ and the right concatenation $w_r = r_{i_1} \cdots r_{i_k}$ denote the same word. The PCP is known to be undecidable if Σ contains at least two symbols [20].

For the reduction, the words over Σ are encoded as natural numbers which are then represented as concrete domain objects. The elements of Σ are interpreted as digits of numbers at base B, where $B := |\Sigma| + 1$. $\overline{w}$ denotes the nonnegative integer at base 10 which the (nonempty) word w represents at base B. $w \mapsto \overline{w}$ is a $1-1$-mapping from Σ^* into the set of nonnegative integers. Concatenation is encoded as an operation over natural numbers that can be captured by a concrete domain predicate: If vw is the concatenation of two words $v, w \in \Sigma^*$, then $\overline{vw} = \overline{v} * B^{|w|} + \overline{w}$, where $|w|$ is the length of the word w.

Now for every instance S of the PCP, an $\mathcal{ALCRP}(\mathcal{D})$ concept *C(S)* can be defined whose models have an infinite tree-like structure that encodes the set of possible solutions of the PCP instance S. Additionally, the concept is defined in a way such that it is satisfiable if and only if none of the possible solutions in fact is a solution. Hence, if the satisfiability of the concept *C(S)* could be decided then one would be able to decide the satisfiability of the PCP S. Since this is impossible, the satisfiability problem for $\mathcal{ALCRP}(\mathcal{D})$ concept terms must be undecidable. Let l, r, w_l, w_r and f_1,...,f_m be attributes and R be a role name, then for a given instance S of the PCP we define a concept *C(S)*:

$$\begin{aligned}
\textit{C(S)} \doteq\; & \exists w_l.\textit{zero-p} \sqcap \exists w_r.\textit{zero-p} \sqcap && (1)\\
& \sqcap_{i=1}^{m} \exists w_l, f_i \circ w_l.\textit{cnstr-p}_l^i \sqcap && (2)\\
& \sqcap_{i=1}^{m} \exists w_r, f_i \circ w_r.\textit{cnstr-p}_r^i \sqcap && (3)\\
& \forall R. \sqcap_{i=1}^{m} \exists w_l, f_i \circ w_l.\textit{cnstr-p}_l^i \sqcap && (4)\\
& \forall R. \sqcap_{i=1}^{m} \exists w_r, f_i \circ w_r.\textit{cnstr-p}_r^i \sqcap && (5)\\
& \forall R. \exists w_l, w_r.\textit{notequal-p} && (6)
\end{aligned}$$

$$R \doteq \exists (w_l, w_r)(w_l, w_r).\textit{trans-p}$$

The predicates used in the definition of *C(S)* are defined as follows (l_i and r_i are the words used in the definition of the PCP problem, see above):

$$\begin{aligned}
\textit{trans-p}(a,b,c,d) &:= a < c \wedge b < d\\
\textit{zero-p}(a) &:= a = 0\\
\textit{cnstr-p}_l^i(a,b) &:= b = \overline{l_i} + a * B^{|l_i|}\\
\textit{cnstr-p}_r^i(a,b) &:= b = \overline{r_i} + a * B^{|r_i|}\\
\textit{notequal-p}(a,b) &:= a \neq b
\end{aligned}$$

To complete the proof of theorem 6, we need to prove the following proposition.

Proposition 7. *The concept C(S) is satisfiable if and only if the PCP S has no solution.*

Proof. It has to be shown that (i) if *C(S)* is satisfiable then S has no solution and (ii) if S has no solution then *C(S)* is satisfiable.

The first point can be seen by examining the definition of *C(S)*. If *C(S)* is satisfiable then there exists an interpretation $\mathcal{I}$ with $C(S)^{\mathcal{I}} \neq \emptyset$. As intended, this interpretation encodes all possible sequences that could be a solution of the PCP S. It has the form of an infinite tree. Each node (abstract object) in the tree has m successors. The edges (attributes) to these successors are labeled with f_1 to f_m, respectively. The edges represent single indices and the nodes represent sequences of indices defined by the labels of the path from the root node. The role R is used as a "universal role" to propagate concepts to every node in the tree: every node in the tree is an R-role filler of the root-node. In lines 4 and 5, predicate-operators which use the predicate *constr-p* are propagated to every node in the tree. These concepts enforce the successors of each node. The concepts in lines 2 and 3 do the same for the root node. Each node in the tree has concrete fillers of the *wl* and *wr* attributes. These fillers are enforced by the concept terms in lines 1-5. The fillers of these attributes encode the two concatenations that are defined by the index sequence which the node represents. Line 6 enforces that for all objects in the tree the fillers of *wl* and *wr* (and thus the concatenations for all possible sequences) differ. From this it is clear that the concept *C(S)* can only be satisfiable if the PCP S has no solution.

To prove the second point we give an interpretation with $C(S)^{\mathcal{I}} \neq \emptyset$ for a given PCP S for which it is known that no solution exists. This interpretation has the same structure as described above.

$$\Delta_{\mathcal{I}} = \{a_{ij}; i \geq 0, 0 \leq j < m^i\};$$
$$\forall i \geq 0, 0 \leq j < m^i :$$
$$f_1^{\mathcal{I}}(a_{ij}) = a_{i+1\ j*m}, \ \ldots, \ f_m^{\mathcal{I}}(a_{ij}) = a_{i+1\ j*m+m-1},$$
$$w_l^{\mathcal{I}}(a_{ij}) = \overline{\phi_l(i,j)}, \quad w_r^{\mathcal{I}}(a_{ij}) = \overline{\phi_r(i,j)}$$

where ϕ_l and ϕ_r are two recursively defined concatenation functions (*concat* concatenates words and $\lfloor\rfloor$ denotes the *floor* function):

$$\phi_l(0,0) = \epsilon$$
$$\phi_r(0,0) = \epsilon$$
$$\phi_l(i,j) = concat(\phi_l(i-1, \lfloor j/m \rfloor), l_{j+1-(m*\lfloor j/m \rfloor)})$$
$$\phi_r(i,j) = concat(\phi_r(i-1, \lfloor j/m \rfloor), r_{j+1-(m*\lfloor j/m \rfloor)}).$$

□

As already noted, in [3] a similar proof is used to prove the undecidability of subsumption for $\mathcal{ALC}(\mathcal{D})$ extended by a transitivity operator for roles. The proof given there differs in that it uses a concept which is satisfiable if and only if the corresponding PCP is has a solution. A transitive attribute is used to distribute concepts to any node. This cannot be directly done in $\mathcal{ALCRP}(\mathcal{D})$. However, with a role-forming predicate we can define a transitive role to achieve similar effects (see [17] for further details).

The new role-forming operator plays an important role in the definition of the concept *C(S)*. As already noted, the complex role R is used as a universal role, i.e. it connects the root node of the tree to all other nodes. Using value restriction over R, a concept term which contains exists restrictions creating new succeeding nodes is propagated to every node of the tree. It is this mechanism that enforces the infinite tree structure of the model of *C(S)*. In the following section a set of restrictions for terminologies is developed. The idea is that these restrictions prohibit the definition of "dangerous" concepts such as *C(S)*.

2.3 Restricted Terminologies

The analysis of the concept *C(S)* at the end of the last section gives a first idea on how to avoid undecidability. Complex roles seem to cause problems if they are used in certain combinations with exists and value restrictions. A more thorough analysis reveals that there exist at least two options for restricting the language such that reasoning becomes decidable. First, restrictions could be posed on the structure of the concrete domain predicates. This is not very promising since all interesting applications, e.g. modeling spatial or temporal relations, require fairly complex predicate definitions that inevitably cause undecidability of reasoning in $\mathcal{ALCRP}(\mathcal{D})$. Second, some critical combinations of concept-forming operators could be restricted. We will pursue the second approach. Before we can introduce our structurally restricted terminologies, we have to make some technical definitions.

A concept term C is said to be *unfolded* w.r.t. a TBox $\mathcal{T}$ iff none of the concept and role names used in the concept term occur on the left side of any terminological axiom in $\mathcal{T}$. Any concept term can be transformed into an unfolded form by iteratively replacing concept and role names by their defining terms. This algorithm terminates since the terminology is required to be acyclic. Any unfolded concept term can then be transformed to an equivalent one in *negation normal form* (NNF). An unfolded concept term is said to be in NNF iff negation occurs only in front of concept names. The transformation to NNF can be done by iteratively applying transformation rules that propagate the negations "down" to the atomic concepts. For example, $\neg(C \sqcap D)$ has to be transformed to $\neg C \sqcup \neg D$. We omit details since the transformation rules are the same as for $\mathcal{ALC}(\mathcal{D})$ [2]. We are now ready to define restricted terminologies.

Definition 8. A concept term X is called *restricted* w.r.t. a TBox $\mathcal{T}$ iff its equivalent *X'* which is unfolded w.r.t. $\mathcal{T}$ and in NNF fulfills the following conditions:

(1) For any (sub)concept term C of *X'* 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 $\exists R_2.E$ where R_2 is also a complex role term.

(2) For any (sub)concept term C of *X'* 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 $\forall R_2.E$ where R_2 is also a complex role term.

(3) For any (sub)concept term C of *X'* 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 (i) quantify over attribute chains of length 1 and (ii) are not contained inside any value and exists restrictions that are also contained in D.

A terminology is called restricted iff all concept terms appearing on the right-hand side of terminological axioms in $\mathcal{T}$ are restricted w.r.t. $\mathcal{T}$. An ABox $\mathcal{A}$ is called restricted w.r.t. a TBox $\mathcal{T}$ iff $\mathcal{T}$ is restricted and all concept terms used in $\mathcal{A}$ are restricted w.r.t. the terminology $\mathcal{T}$.

Consider for example the following three very simple terminologies that are already in unfolded NNF. None of them is restricted because they all violate one of the above conditions. Let C and D be concept names, R_a be an atomic role term, R_c be a complex role term, f be a feature, and u be a feature chain with a length greater than 1.

$$\mathcal{T}_1 : \{C \doteq \forall R_c.\exists R_c.D\},$$
$$\mathcal{T}_2 : \{C \doteq \exists R_c.\exists u.P\},$$
$$\mathcal{T}_3 : \{C \doteq \forall R_c.\forall R_a.\exists f.P\}$$

We can now examine the decidability of the standard reasoning problems with respect to restricted terminologies of $\mathcal{ALCRP}(\mathcal{D})$.

Theorem 9. *The ABox consistency problem for restricted $\mathcal{ALCRP}(\mathcal{D})$ ABoxes is decidable.*

We prove this by giving an algorithm which is sound and complete.

2.4 The Tableau Calculus

The algorithm is a standard tableau-based one as it is used for first order or modal logics as well as other description logics. To decide the satisfiability of a concept C, the algorithm starts with an initial ABox $\mathcal{A}_0 := \{a : C\}$ and then iteratively applies completion rules creating one or more descendant ABoxes. Thus, rule application constructs a tree of ABoxes Υ. Finally, either all ABoxes that are leaves of Υ happen to be contradictory which means that C is not satisfiable or a non-contradictory ABox is obtained to which no more completion rules are applicable. In the latter case, the ABox mentioned is called *complete* and defines a model for C. The purpose of the completion rules can be understood as making implicit facts explicit. The algorithm can be considered as a step-by-step model construction.

The rule set is an extension of the one used for deciding ABox consistency in $\mathcal{ALC}(\mathcal{D})$ (see [2]). Before the rules can be given, some technical terms need to be defined. 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 $f_1 \circ \cdots \circ f_k$, and let $u_1, \ldots, u_n$ and $v_1, \ldots, v_m$ be arbitrary feature chains. For convenience we define three functions as follows:

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

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

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

An ABox $\mathcal{A}$ is said to contain a *fork* (for a feature f) if it contains the two axioms $(a, b) : f$ and $(a, c) : f$, where a and b are either both from O_A or O_D. A fork can be eliminated by replacing all occurrences of c in $\mathcal{A}$ with b. We assume that during rule application forks are eliminated as soon as they appear. Before any rules are applied to initial ABox $\mathcal{A}_0$, fork elimination also takes place. The completion rules can now be defined.

Definition 10. The following *completion rules* will replace an ABox $\mathcal{A}$ by a single 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 $f_1, \ldots, f_n$ denote feature names, and $u_1, \ldots, u_m$ as well as $v_1, \ldots, v_m$ 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 exists restriction rule.
Premise: $a : \exists R.C \in \mathcal{A}, \quad \neg\exists b \in \mathsf{O}_A :$
$(\mathit{filler?}_{\mathcal{A}}(a, b, R) \wedge b : C \in \mathcal{A})$
Consequence: $\mathcal{A}' = \mathcal{A} \cup \{(a, b) : R, \ b : C\}$
where $b \in \mathsf{O}_A$ is not used in $\mathcal{A}$.

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

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

Rr∃P The complex role rule.
Premise:
$(a, b) : \exists(u_1, \ldots, u_n)(v_1, \ldots, v_m).P \in \mathcal{A},$
$\neg\exists x_1, \ldots, x_n, y_1, \ldots, y_m \in \mathsf{O}_D :$
$(\mathit{filler}_{\mathcal{A}}(a, u_1) = x_1 \wedge \ldots \wedge \mathit{filler}_{\mathcal{A}}(a, u_n) = x_n \wedge$
$\mathit{filler}_{\mathcal{A}}(b, v_1) = y_1 \wedge \ldots \wedge \mathit{filler}_{\mathcal{A}}(b, v_m) = y_m \wedge$
$(x_1, \ldots, x_n, y_1, \ldots, y_m) : P \in \mathcal{A})$
Consequence:
$\mathcal{A}' = \mathcal{A} \cup \{(x_1, \ldots, x_n, y_1, \ldots, y_m) : P\} \cup$

$chain_{\mathcal{A}}(a, x_1, u_1) \cup \ldots \cup chain_{\mathcal{A}}(a, x_n, u_n) \cup$
$chain_{\mathcal{A}}(b, y_1, v_1) \cup \ldots \cup chain_{\mathcal{A}}(b, y_m, v_m)$
where the objects $x_i \in \mathsf{O}_D$ and $y_i \in \mathsf{O}_D$ are not used in $\mathcal{A}$.

RChoose The choose rule.
Premise:
$a : \forall(\exists(u_1, \ldots, u_n)(v_1, \ldots, v_m).P).C \in \mathcal{A}$,
$\exists b \in \mathsf{O}_A,\ x_1, \ldots, x_n, y_1, \ldots, y_m \in \mathsf{O}_D$:
$(\mathit{filler}_{\mathcal{A}}(a, u_1) = x_1 \wedge \ldots \wedge \mathit{filler}_{\mathcal{A}}(a, u_n) = x_n \wedge$
$\mathit{filler}_{\mathcal{A}}(b, v_1) = y_1 \wedge \ldots \wedge \mathit{filler}_{\mathcal{A}}(b, v_m) = y_m \wedge$
$(x_1, \ldots, x_n, y_1, \ldots, y_m) : P \notin \mathcal{A} \wedge$
$(x_1, \ldots, x_n, y_1, \ldots, y_m) : \overline{P} \notin \mathcal{A})$
Consequence:
$\mathcal{A}' = \mathcal{A} \cup \{(x_1, \ldots, x_n, y_1, \ldots, y_m) : P\}$,
$\mathcal{A}'' = \mathcal{A} \cup \{(x_1, \ldots, x_n, y_1, \ldots, y_m) : \overline{P}\}$

The notion of a contradictory ABox still needs to be formally defined.

Definition 11. Let the same naming conventions be given as in Definition 10. Additionally, let f be a feature. An ABox $\mathcal{A}$ is called *contradictory* 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.

Similar completion rules can also be found in algorithms for related languages such as $\mathcal{ALC}(\mathcal{D})$. The clash triggers are identical to those of $\mathcal{ALC}(\mathcal{D})$. The new completion rules are Rr∃P and RChoose. The use of the *related* function is also new and is necessary because the new role-forming operator is introduced. In the following we will discuss only the novelties as compared to the rules needed for $\mathcal{ALC}(\mathcal{D})$. First, consider the rule R∀C. The use of the *related* function is necessary because not all role fillers might appear explicitly as constraints of the form $(a, b) : R$. If R is a complex role then b could be an R role filler of a, although there is no constraint of the above form. Two objects can be related simply because a predicate holds over the concrete fillers of feature chains starting from a and b, respectively. This can be seen by considering the semantics of the role-defining operator. The two possible types of role fillers (explicit and implicit) are both captured by the *related* function.

The meaning of the rule Rr∃P is straightforward. If it is known that two objects are related via a complex role, we can immediately create concrete objects as fillers of those feature chains being used in the definition of the complex role. It is then also known that the predicate used in the definition of the role holds over the newly created concrete objects and thus appropriate constraints can be added. This is what Rr∃P does.

The meaning of RChoose can be understood as follows. If a set of feature chains is used in the definition of a complex role R and (loosely spoken) there are the appropriate concrete fillers for two objects a and b for all the feature chains in this set, then b might be an R role filler of a. But unless there is an explicit constraint which states that the predicate P used in the definition of the role R holds (or does not hold), we do not know if this is really the case. So, if there is no such constraint, we have to try both alternatives and test whether P holds or does not hold. If any of these two alternatives is the wrong one, we will end up with a concrete domain clash in the corresponding branch of the ABox tree Υ. Like R⊔, an application of RChoose creates a branch in Υ.

A full proof of the soundness and completeness of the algorithm given above can be found in [16]. Soundness of the algorithm follows from the local soundness of the completion rules. Completeness can be proven by showing that any complete ABox computed by the algorithm defines a model for the initial ABox $\mathcal{A}_0$. However, proving termination is less straightforward. The restrictions posed on terminologies are required to ensure that all satisfiable concepts from the language are also satisfied by at least one finite model. If this finite model property does not hold, termination of the tableau algorithm can not be guaranteed.

We give a short sketch of the termination proof and then motivate the definition of the restrictions on concept terms. Each ABox that is computed by the application of completion rules can be mapped to a forest, i.e. a collection of trees, in the following way: Each element of O_A that is already present in $\mathcal{A}_0$ is the root of a tree. The edges and further nodes of the trees correspond to role filler relationships and abstract objects, respectively. But only the explicit role filler relationships introduced by the rules R∃C, R∃P and Rr∃P are taken into account. The number of the trees obviously doesn't grow during rule application. It can be shown that (i) infinite rule application implies infinite growth of at least one of the trees in at least one sequence of forests[2] and (ii) that there exist upper bounds for the degree and the depth of all the trees. From this it follows that the algorithm terminates. The restrictions on concept terms ensure the existance of the upper

[2] A sequence of forrests corresponds to one path in Υ.

bound for the depth of the trees. For the detailed termination proof see [16].

As already noted at the end of section 2.2, in the case of unrestrictedness there are concept terms which are only satisfied by infinite models. To shed more light on details about the restrictions given in Definition 8, it is necessary to analyze how these concepts would be treated by the tableau calculus given in this section. Using a value restriction over a complex role $\forall R_c.C$, a concept term $C \doteq \exists R.D$ can be "propagated" to an object o_1. Then, the application of the R∃C rule creates an object o_2 with $(o_1, o_2) : R$. The concept term D can be crafted in a way such that concrete objects are generated as fillers of some attributes of o_2. This leads to the inference of a new role filler relationship between another object and the newly created object o_2. Again a value restriction is used to propagate an exists restriction along the role filler relationship just inferred to the object o_2. Thus, a cycle is obtained. To prevent this, one has to prevent the generation of concrete fillers for the attributes of o_2. Concrete objects are only created by the rules R∃P and Rr∃P. The rule R∃P can only be applied if there is a concept-forming predicate operator inside the concept term D. The rule Rr∃P can only be applied if the concept term C contains an exists restriction quantifying over a complex role. Summarizing, concept-forming predicate operators inside of value restrictions quantifying over complex roles and also nestings of value and exists restrictions which both quantify over complex roles have to be prohibited. Further elaboration yields the restrictions given in definition 8.

Corollary 12. *The subsumption problem and the satisfiability problem for $\mathcal{ALCRP}(\mathcal{D})$ concept terms are decidable w.r.t. terminologies for which the considered concept terms are restricted.*

Proof. This follows from Theorem 9 together with the reduction of subsumption to satisfiability and of satisfiability to ABox consistency (see Section 2.1). Please note that if the concept terms C and D are restricted w.r.t. a terminology $\mathcal{T}$, then the concept term $C \sqcap \neg D$ is also restricted w.r.t. $\mathcal{T}$ since the set of restricted concept terms is closed under negation.

In order use $\mathcal{ALCRP}(\mathcal{D})$ for knowledge representation in general and for spatial reasoning in particular, an admissible concrete domain must be defined. This will be discussed in the next section.

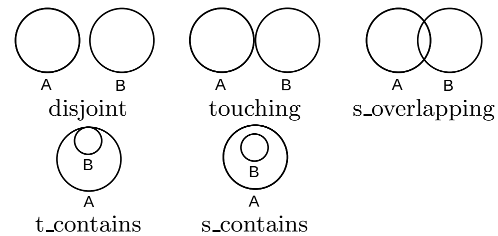

Figure 1: Elementary relations between two regions A and B. The inverses of t_contains and s_contains as well as the relation equal have been omitted.

3 A Concrete Domain for Polygonal Space

Rather than dealing with arbitrary point sets in $\Re^2$, we restrict the predicates for the spatial domain to the description of polygons because efficient algorithms (e.g. the simplex procedure) are known for the polygon inclusion and polygon intersection problems (see below). 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. Polygons are defined as usual as a list of polylines, i.e. polygons describe point sets that are not necessarily internally connected.

3.1 Predicates for Qualitative Spatial Relationships

The set $\Phi_{\mathcal{D}_\mathcal{P}}$ contains the names of eight elementary binary predicates (equal, disjoint, touching, strictly_overlapping, tangentially_contains/ tangentially_inside, strictly_contains/strictly_inside) representing spatial relationships as illustrated in Figure 1. In analogy to Egenhofer [8] or RCC-8 [22] we have given a formal definition of the elementary relations in [11]. The definition is based on the interior, the complement and the boundary of spatial objects (point sets). The interior of a set λ is defined to be the union of all open sets in λ. The boundary of a polygon, i.e. the intersection of the closure of the interior and the closure of the complement of an object, is defined by the "border polyline."

For convenience, we extend the set $\Phi_{\mathcal{D}_\mathcal{P}}$ by names for so-called composite predicates consisting of disjunctions of elementary predicates. For instance, in this paper we use the universal relation spatially_related and the relation generally_inside (g_inside $\equiv$ t_inside $\vee$ s_inside $\vee$ equal). We define one-place predicates which are denoted as sr_p where sr is an elementary or composite predicate and p is a concrete object representing a polygon constant.

3.2 Satisfiability of Conjunctions of Predicates

The admissibility criterion for $\mathcal{D}_{\mathcal{P}}$ concerns the satisfiability of finite conjunctions of (possibly negated) predicate terms. Negated unary and binary terms can be resolved into disjunctions of elementary spatial predicates by simple syntactic transformations because the elementary predicates are mutually exclusive and exhaustive. Therefore, we can restrict our analysis to conjunctions of unnegated binary terms $\bigvee_{j=1}^{k} \mathsf{ep}^j(x, y)$ where $1 \leq k \leq 8$ 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 relations 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 a 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. $\mathsf{s\_inside}(A, B) \wedge \mathsf{s\_inside}(B, C) \Rightarrow \mathsf{s\_inside}(A, C)$.

However, in general, path consistency is not a sufficient criterion for consistency. Thus, an additional step is required to ensure *global* consistency. Algorithms for solving these constraint problems are discussed in [15] and [19, 24]. According to Nebel and Renz [19, 24], the worst case complexity depends on the relations (disjunctions of base relations) actually encountered in a constraint network. To achieve global consistency, in the worst case, exponential algorithms are required. However, Nebel and Renz [19, 24] 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 consistency (Ladkin and Reinefeld [15] also discuss speedup techniques).

Grigni et al. [9] consider objects that are internally connected regions in the plane and propose two notions of satisfiability, relational consistency and realizability. First, a conjunction may violate the so-called *relational consistency* criterion which is identical to the global consistency of a (spatial) constraint network (see above). 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 (for examples see [9]). Thus, according to their semantics (regions are internally connected), there are conjunctions of predicates that are relationally consistent but not realizable in the plane.

In our approach, the geometric relationship between two concrete polygons (or its border polylines) directly corresponds to the qualitative spatial relationship of the objects. The elementary predicates for qualitative spatial relations have the property of being mutually exclusive and exhaustive. Hence, topological reasoning can be realized by inference processes based on composition tables when all spatial relations implicitly given by concrete polygons are computed and non-concrete spatial objects are not assumed to be internally connected. The work of Renz [23] shows that, in the case of regions that are not necessarily internally connected (open set semantics), satisfiability of RCC-8 constraint systems implies realizability. Thus, an algorithm for the concrete domain satisfiability test can be divided into three main steps:

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 $\mathsf{t\_contains}(x, y) \vee \mathsf{s\_contains}(x, y) \vee \mathsf{s\_overlapping}(x, y) \vee \mathsf{touching}(x, y) \vee \mathsf{disjoint}(x, y)$. Furthermore, for each conjunct a new conjunct representing the inverted relation term is added. For example, for the term $\mathsf{t\_contains}(x, y)$ we add $\mathsf{t\_inside}(y, x)$.
2. The one-place predicates $\mathsf{sr}_{\mathsf{p}}^{i}$ introduce concrete polygons. With the help of standard algorithms from computational geometry we compute the topological base relation between each pair of concrete polygons. In accordance to our definition of the spatial relations, this problem basically can be reduced to the intersection test for two polygons. For each pair of polygons, the topological information is added to the constraint system with a corresponding predicate. From now on, concrete polygons are treated as variables.
3. In the third step, relational consistency is verified. Thus, a globally consistent solution must be computed (path-consistency may be realized as a pruning step). If no globally consistent solution can be found, return "not satisfiable", otherwise return "satisfiable."

In summary, we showed that the concrete domain $\mathcal{P}$ is admissible. The next section discusses the application of the algorithm using several application examples.

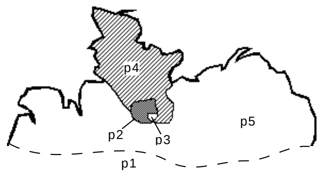

Figure 2: A sketch of the northern part of Germany with polygons for Germany (p_1), Northern Germany (p_5), the federal states Schleswig-Holstein (p_4) and Hamburg (p_2). Polygon p_2 is touching p_4 and p_3 is t_inside p_2.

3.3 Examples for Spatioterminological Reasoning

How can conrete domain predicates be used to support spatioterminological inferences? First of all, as an ontological commitment we assume that each domain object is associated with its spatial representation via the attribute has_area. We would like to define predicates that restrict the role fillers for has_area to be specific spatial regions. For instance, without being told explicitly, the inference system should automatically classify a region in Hamburg also as a Northern German region (see Figure 2).

For instance, the one-place predicate $\mathsf{g\_inside}_{\mathsf{p}}$ can be used as follows. Using the concept-forming predicate operator $\exists\, \mathsf{f}\,.\,\mathsf{P}$ (see above), we define restrictions for fillers of the feature (or attribute) has_area of a region in Northern Germany, for a district of the city of Hamburg, etc. The restrictedness criterion (cf. Definition 8) for the following two TBox axioms is trivially fulfilled because they contain no nested exists or all quantifiers.

$$\mathbf{northern\_german\_region} \doteq \exists\, \mathsf{has\_area}\,.\,\mathsf{g\_inside}_{\mathsf{p}_5}$$

$$\mathbf{district\_of\_hh} \doteq \exists\, \mathsf{has\_area}\,.\,\mathsf{g\_inside}_{\mathsf{p}_2} \sqcap \exists\, \mathsf{has\_area}\,.\,\neg\mathsf{equal}_{\mathsf{p}_2}$$

In the concept northern_german_region existential quantification for has_area is used to constrain the filler to be any polygon that is g_inside of p_5 which defines the area of Northern Germany (see Figure 2). 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. By analogy, we define the concepts for the federal states Hamburg and Schleswig-Holstein. We would like to emphasize that both concepts are subsumed by the concept northern_german_region.

$$\mathbf{federal\_state\_hh} \doteq \mathsf{german\_federal\_state} \sqcap \exists\, \mathsf{has\_area}\,.\,\mathsf{equal}_{\mathsf{p}_2}$$

$$\mathbf{federal\_state\_sh} \doteq \mathsf{german\_federal\_state} \sqcap \exists\, \mathsf{has\_area}\,.\,\mathsf{equal}_{\mathsf{p}_4}$$

These simple examples can already be represented with $\mathcal{ALC}(\mathcal{D})$. However, in many cases, restrictions about spatial relations have to be combined with additional conceptual restrictions. Thus, there is a need to quantify over roles that are defined based on predicates over a concrete domain. 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. p_3 in Figure 2) is inside any polygon that is equal to the state polygon (e.g. p_2). The domain object that refers to the polygon p_2 with the feature has_area must also be subsumed by the concept federal_state_hh. We can adequately express this restriction with the help of the *role-forming predicate restriction* ($\exists\, (\mathsf{f})(\mathsf{f})\,.\,\mathsf{P}$). For modeling spatial relations we use role axioms for declaring corresponding roles in the TBox. The following TBox axioms also fulfill the restrictedness criterion because the nested concept terms employ only the $\exists\, \mathsf{f}\,.\,\mathsf{P}$ constructor.

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

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

The concept hh_border_district is discussed as an example for the use of the role-forming predicate restriction introduced by is_t_inside. The associated polygon of any individual that is a member this concept has to be in the t_inside relationship with another polygon that, in turn, is referred to by an instance of the concept federal_state_hh.

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

$$\mathbf{spatially\_related} \doteq \exists\, (\mathsf{has\_area})(\mathsf{has\_area})\,.\,\mathsf{spatially\_related}$$

$$\mathbf{is\_touching} \doteq \exists\, (\mathsf{has\_area})(\mathsf{has\_area})\,.\,\mathsf{touching}$$

$$\mathbf{unknown} \doteq \mathsf{district\_of\_hh} \sqcap \exists\, \mathsf{spatially\_related}\,.\,\mathsf{federal\_state\_hh} \sqcap \exists\, \mathsf{touching}\,.\,\mathsf{federal\_state\_sh}$$

If the polygon of district_of_hh touches the polygon of federal_state_sh, then the polygon of district_of_hh is also t_inside the polygon of federal_state_hh. Therefore, it can be proven that unknown is subsumed by hh_border_district (see the next section). The spatial constellation defined by the concept unknown could also be characterized as a "Hamburg border district to Schleswig-Holstein."

If district_of_hh had been defined without the term $\exists$ has_area . $\neg$equal$_{p_2}$, unknown would *not* have been subsumed by hh_border_district because an abstract individual whose associated polygon had been *equal* to p_2 would have been a member of unknown but not a member of hh_border_district (for hh_border_district the equal relation is eliminated by the constraint satisfaction process). A more detailed discussion of this example and the interaction between $\mathcal{ALCRP}(\mathcal{D})$ and the domain $\mathcal{D}_{\mathcal{P}}$ can be found in [18].

4 Conclusion

Based on the description logic $\mathcal{ALCRP}(\mathcal{D})$, 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 the notion of a role which is defined based on properties of concrete objects. The abstract domain is used to represent terminological knowledge about spatial objects on an abstract logical level. The concrete domain (space domain) extends the abstract domain and provides access to spatial reasoning algorithms. If required, even quantitative data (conrete polgyons) are considered by applying algorithms known from computational geometry. Techniques for spatial indexing can easily be integrated.

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. Our approach for testing satisfiability of finite conjunctions relies on current work in qualitative spatial reasoning theory [24]. Considering the topological spatial relations, it becomes clear that another instance of $\mathcal{ALCRP}(\mathcal{D})$ can deal with temporal relations. Similar constraint satisfaction algorithms known from the literature (e.g. [15]) can be employed. Future work will reveal the relationship between $\mathcal{ALCRP}(\mathcal{D})$ and, for instance, the temporal description logic developed in [1]. Although this approach does not impose restrictions on terminologies, it does not provide facilities for expressing value restrictions over spatial 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.

Acknowledgments

We would like to thank Franz Baader who provided some of the ideas behind the PCP-reduction. Furthermore, we acknowledge the work of the anonymous reviewers who gave fruitful comments on this paper.

References

[1] A. Artale and E. Franconi. A temporal description logic for reasoning about actions and plans. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Fourth International Conference on Principles of Knowledge Representation*, May 1994.

[2] F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. In *Twelfth International Joint Conference on Artificial Intelligence, 1991*, pages 452–457, August 1991.

[3] F. Baader and P. Hanschke. Extensions of concept languages for a mechanical engineering application. In H.J. Ohlbach, editor, *16th German Conference on Artificial Intelligence*, pages 132–143, September 1992.

[4] B. Bennett. Modal logics for qualitative spatial reasoning. *Bull. of the IGPL*, 3:1–22, 1995.

[5] S. Borgo, N. Guarino, and C. Masolo. A pointless theory of space based on strong connection and congruence. In L.C. Aiello, J. Doyle, and S. Shapiro, editors, *Fifth International Conference on Principles of Knowledge Representation*, pages 220–229, November 1996.

[6] A.G. Cohn, B. Bennett, J.M. Gooday, and N.M. Gotts. Representing and reasoning with qualitative spatial relations. In Stock [27], pages 97–134.

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

[8] M.J. Egenhofer. Reasoning about binary topological relations. In O. Günther and H.-J. Schek,

editors, *Advances in Spatial Databases, Second Symposium, SSD'91, Zurich, Aug. 28-30, 1991*, volume 525 of *LNCS*, pages 143–160. Springer-Verlag, August 1991.

[9] M. Grigni, D. Papadias, and C. Papadimitriou. Topological inference. In C. Mellish, editor, *Fourteenth International Joint Conference on Artificial Intelligence, 1995*, pages 901–906, August 1995.

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

[11] V. Haarslev and R. Möller. SBox: A qualitative spatial reasoner—progress report. In L. Ironi, editor, *11th International Workshop on Qualitative Reasoning, Pubblicazioni N. 1036, Istituto di Analisi Numerica C.N.R. Pavia*, pages 105–113, June 1997.

[12] V. Haarslev and R. Möller. Spatioterminological reasoning: Subsumption based on geometrical inferences. In Rousset et al. [25], pages 74–78.

[13] V. Haarslev, R. Möller, and C. Schröder. Combining spatial and terminological reasoning. In B. Nebel and L. Dreschler-Fischer, editors, *KI-94: Advances in Artificial Intelligence – Proc. 18th German Annual Conference on Artificial Intelligence*, volume 861 of *LNAI*, pages 142–153. Springer-Verlag, September 1994.

[14] V. Haarslev and M. Wessel. 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*, pages 204–211. IEEE Computer Society Press, Los Alamitos, September 1996.

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

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

[17] C. Lutz and R. Möller. Defined topological relations in description logics. In Rousset et al. [25], pages 15–19.

[18] R. Möller, V. Haarslev, and C. Lutz. Spatioterminological reasoning based on geometric inferences: The $\mathcal{ALCRP}(\mathcal{D})$ approach. Technical Report FBI-HH-M-277/97, University of Hamburg, Computer Science Department, 1997.

[19] B. Nebel. Computational properties of qualitative spatial reasoning: First results. In I. Wachsmuth, C.-R. Rollinger, and W. Brauer, editors, *Proceedings, KI-95: Advances in Artificial Intelligence, 19th Annual German Conference on Artificial Intelligence*, volume 981 of *LNAI*, pages 233–244. Springer-Verlag, September 1995.

[20] E.M. Post. A variant of a recursively unsolvable problem. *Bull. Am. Math. Soc.*, 52:264–268, 1946.

[21] I. Pratt and O. Lemon. Ontologies for plane, polygonal mereotopology. *Notre Dame Journal of Formal Logic*, 2(38):225–245, 1997.

[22] D.A. Randell, Z. Cui, and A.G. Cohn. A spatial logic based on regions and connections. In B. Nebel, C. Rich, and W. Swartout, editors, *Principles of Knowledge Representation and Reasoning*, pages 165–176, October 1992.

[23] J. Renz. A canonical model of the region connection calculus. In T. Cohn and L. Schubert, editors, *Sixth International Conference on Principles of Knowledge Representation*, June 1998.

[24] J. Renz and B. Nebel. 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)*, pages 522–527, August 1997.

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

[26] K. Schild. A correspondence theory for terminological logics: Preliminary report. In *Twelfth International Conference on Artificial Intelligence*, pages 466–471, August 1991.

[27] O. Stock, editor. *Spatial and Temporal Reasoning*. Kluwer Academic Publishers, Dordrecht, 1997.