Combining Lightweight Description Logics with the Region Connection Calculus*

Özgür L. Özçep and Ralf Möller

Institute for Software Systems (STS)
Hamburg University of Technology
Hamburg, Germany

August 5, 2011

Abstract

Providing reasoning in GIS domains is a demanding task due to the size of the data that are stored in secondary memory. This is also the case for query answering over spatio-thematic ontologies which act as an interface to the GIS data and filter unintended models. Solving this problem by compiling the ontology into an SQL query works only in those cases in which such a compilation, first-order logic (FOL) rewritability, is theoretically possible. Combining lightweight description logics like DL-Lite that are tailored towards FOL-rewritability with spatial calculi like the region connection calculus in a naïve way can prohibit FOL-rewritability. But if assumption on the completeness and the consistency of the spatial data are made and the combination is done in a controlled way combinations of DL-Lite and the region connection calculus result that are sufficiently expressive to model GIS data and at the same time allow for computationally feasible query answering.

1 Introduction

There is a need for reasoning over geographical data in almost any area in which geographical information systems (GIS systems for short) are used, e.g., damage classification for flooding scenarios, development of eco systems in forestry, or analysis of sociological and demoscopic aspects in urban areas—to mention just a few. But providing reasoning services over geographical data is a demanding task because of at least two reasons which we are going to explain on the basis of the TIGER/Line® GIS data of the US Census Bureau[1], a well known free set of geographic data.

The first main problem is to specify the concepts and relations of the geographical domain over which one wants to reason; the intended meanings are not given in a formal

*The presented work resulted from the DFG funded project "Optimierungstechniken für beschreibungslogische Schlussfolgerungssysteme im Bereich der Geo-Daten (GeoDL) "

[1] http://www.census.gov/geo/www/tiger/

logical language with a precise semantics but in most cases with some feature codes and explanations of the codes in natural language. This holds also for the TIGER/Line® GIS data which specify features like parks, rivers, hospitals with the MAF/TIGER Feature Class Code and describe the intended meanings in the manual (Johnson et al., 2009). Only basic subsumption relations, e.g. "All parks are governmental area" are directly modelled in the data. If one wanted to provide a consistency test that checks whether the intended semantics of the feature codes are indeed in accordance with the data, one would have to do the hard of work translating the natural language specifications in some formal language and then apply a theorem or tableau prover over the resulting set of axioms.

But even if one had success in translating the natural language descriptions into some formal language, it would not be guaranteed that the formal language—which would have to be expressive enough in order to capture the natural language descriptions[2]—is computationally feasible. And here enters the second main problem for reasoning over GIS-data: the huge amount of geographical data which are usually stored persistently in secondary memory and maintained with sophisticated indexing mechanisms restricts the possibilities for expressive declarative knowledge representation and reasoning. It is common sense knowledge that higher representation capabilities lead to more complex reasoning services over the representation in terms of time and space resources—and this is more than true for geographical data which already consume space resources before any reasoning has started; for example, loading the TIGER/Line® shapefiles for the state New York in the relational database management system like SQLServer 2008 results in a database of roughly 7 GB.

Nonetheless, we will argue in this paper that for some reasoning scenarios over GIS data sufficiently expressive logics can be defined that are computationally feasible. The idea of providing a conceptualization over the data is to filter unintended models of the data. The more expressive the logic is the more unintended models can be filtered. But for some GIS scenarios it is not necessary to give such a complete conceptualization, it suffices to filter out some smaller set of the unintended models. And so the idea is to make the logic for representing the knowledge only as much expressible as is necessary to exclude this smaller set. For example, think of a city planning bureau that wants to use the TIGER/Line® data to plan additional parks in New York and so adds some features like *ParkContainingLakeAtTheBorder* (*ParkWithLake* for short) and *ParkContainingPlayingAreaAtTheBorder* (*ParkPlaying* for short) and states some necessary conditions for *ParkWithLake* and *ParkPlaying*, e.g. in case of the concept *ParkWithLake*: If x is a *ParkWithLake*, then x is a *Park* and x *contains* a *Lake* such that the *Lake touches* the *Park* from within. Now, a user wants to find out whether—with respect to the knowledge base of the city planning bureau—there are parks with a playing area that does not reside as an island in some lake of the park. Then, a complete reasoning service would also have to take into account all parks that are instances of the newly introduced concepts *ParkWithLake* and *ParkPlaying*, because such a park would contain a playing area that is not an island in

[2] For example take the explanation of Lake/Pond in the manual (Johnson et al., 2009, Appendix, F-7) that reads "A standing body of water that is surrounded by land"

the park. This reasoning step can be realized without a more complete conceptualization of the concepts *ParkWithLake* or *ParkPlaying* that also provides sufficient conditions.

The reasoning services over GIS data do not only involve pure geographical concepts (area, region etc.) and geographical relations (contains, touches etc.) but also thematic concepts and relations that can be combined with the geographical concepts and relations to build mixed concepts. The combination of thematic concepts with geographical concepts calls for a representation in a combined logic with a thematic component and a spatial component. In this paper, we will investigate combinations of lightweight descriptions logics (DL-Lite) that are well suited for reasoning over large databases with the region connection calculus which can model topological relations like that of containment. As we will show, the representation capabilities of the combined logic that are needed to adequately model mixed concepts like voting district containing a hospital or a park containing a lake may lead to reasoning mechanisms that can not be captured by simple SQL queries. On the other hand it is possible to give combinations that allow for feasible reasoning mechanisms; besides being feasible, these reasoning mechanisms are sufficient to allow for complete answering services that can be used in scenarios like the one of the city planning bureau mentioned above.

2 Logical Preliminaries

In this section, we introduce the two components of the combined logics that we are going to investigate.

2.1 RCC8-calculus

The Region Connection Calculus (RCC) (Randell et al., 1992) is the most widely known member of qualitative spatial reasoning calculi that take regions and not points as the basic entities for representing and reasoning over spatial knowledge. In the axiomatic representation of Randell et al. (1992) a primitive binary relation C is intended to model the connectedness relation between regions, and is therefore axiomatically restricted to be reflexive and symmetric. The connectedness relation is used to define different relations between regions that are termed base relations. One family of base relations, denoted $\mathcal{B}_{RCC8} = \{\mathsf{dc}, \mathsf{ec}, \mathsf{eq}, \mathsf{po}, \mathsf{ntpp}, \mathsf{tpp}, \mathsf{ntppi}, \mathsf{tppi}\}$ henceforth, constitutes the calculus RCC8. Further calculi of the region connection calculus can be defined by considering other sets of base relations, but we will focus on RCC8 as its base relations are quite natural[3] and sufficiently expressive to model many spatial configurations. The base relation dc is intended to model the disjointness relation and is defined by the axiom

$$\mathsf{dc}(x, y) \text{ iff } \neg C(x, y)$$

The other base relations are defined similarly. We omit the definitions but give the intended meanings of the base relations in Figure 1. The axioms imply that the base relations

[3]The claim that $\mathcal{B}_{RCC8}$ is a natural set of base relations can be justified by the fact that another qualitative spatial model, the 9-intersection model of Egenhofer (1991), results in the same set of eight base relations if it is constrained to regularly closed regions.

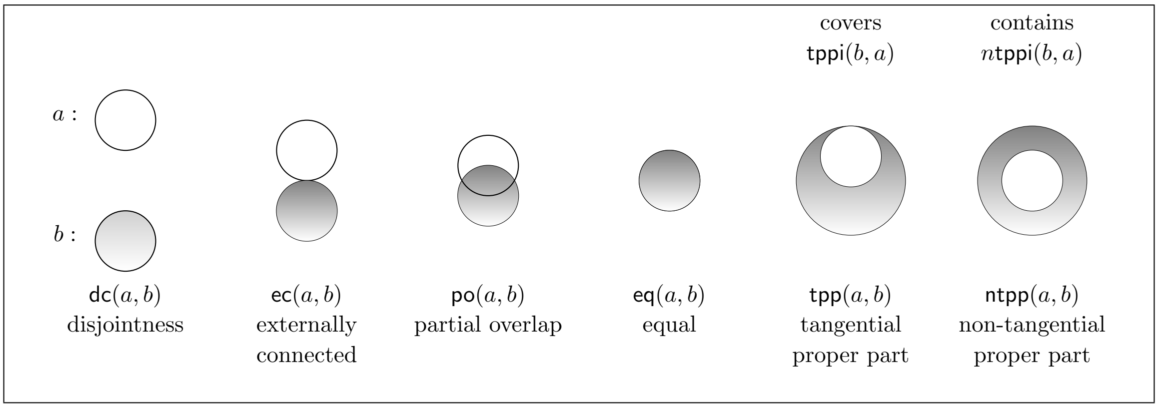

Figure 1: Base relations of RCC8 and their intended meanings

have the JEPD-property, i.e., the eight base relations are jointly exhaustive and pairwise exclusive.

With the help of the base relations real-world spatial configurations can be represented in the form of networks which can be processed by constraint satisfaction procedures. A network consists of a set of sentences that have the form $r_1(a, b) \vee \cdots \vee r_k(a, b)$ where a, b are constants and $r_1, \ldots, r_k$ are base relations from $\mathcal{B}_{RCC8}$. These sentences are presented in the more succinct algebraic notation as $\{r_1, \ldots, r_k\}(a, b)$. The set of all possible disjunctions of base relations $Pot(\mathcal{B}_{RCC8})$ is denoted Rel_{RCC8}. For example, the network $\{\mathsf{ec}(a, b), \mathsf{dc}(b, c), \mathsf{dc}(a, c)\}$ represents a spatial constellation of regions a, b, c where a and b are touching each other and are disjoint from c, respectively. With disjunctions of base relations indefinite knowledge on the spatial relations of regions can be expressed. These networks can be represented with labelled graphs in which the vertices are the constants of the network and in which an edge labelled $\{r_1, \ldots, r_k\}$ is drawn between (a, b) iff $\{r_1, \ldots, r_k\}(a, b)$ is contained in the network.

A practically relevant question is whether a network is satisfiable with respect to the RCC8-axioms. For example, $\{\mathsf{tpp}(a, b), \mathsf{tpp}(b, c), \mathsf{tpp}(a, c)\}$ is satisfiable while the network $\{\mathsf{tpp}(a, b), \mathsf{ntpp}(b, c), \mathsf{tpp}(a, c)\}$ is not satisfiable. Testing the satisfiability of networks can be carried out by path consistency algorithms (Mackworth, 1977). These algorithms are based on composition tables for the base relations which contain for every pair of base relations r_1, r_2 its composition $r_1 \circ r_2$. The composition $\circ$ of two relations r_1 and r_2 is defined as $r_1 \circ r_2 = \{(x, y) \mid \exists z. r_1(x, z) \wedge r_2(z, y)\}$.

The composition table for RCC8[4] is in fact a table of weak compositions. For two relations r_1, r_2 the weak composition $r_1; r_2$ is the minimal disjunction of base relations that cover their composition $r_1 \circ r_2$, i.e., $r_1 \circ r_2 \subseteq r_1; r_2$.[5] For example the weak composition table entry for the pair $(\mathsf{ec}, \mathsf{ntpp})$ is $\mathsf{ec}; \mathsf{ntpp} = \{\mathsf{po}, \mathsf{tpp}, \mathsf{ntpp}\}$. This composition table entry can be described by the following FOL-sentence:

$$\forall x \forall y \forall z. (\mathsf{ec}(x, y) \wedge \mathsf{ntpp}(y, z) \rightarrow (\mathsf{po}(x, z) \vee \mathsf{ntpp}(x, z) \vee \mathsf{tpp}(x, z))$$

[4] Compare (Renz, 2002, p. 45)

[5] The set $r_1; r_2$ is an approximation of $r_1 \circ r_2$ from above, in other words, $r_1; r_2$ is implied by $r_1 \circ r_2$. Therefore the operator ; is termed *weak* composition.

The (weak) composition relation ; is defined for non-base relations $r_1 = \{r_1^1, \dots r_1^k\}$ and $r_2 = \{r_2^1, \dots, r_2^l\}$ in the usual way by pairwise composing the contained base-relations: $r_1; r_2 = \bigcup_{1 \leq i \leq k; 1 \leq j \leq l} r_1^i; r_2^j$.

By a translation into the modal logic S4 it can be shown that the satisfiability test for RCC8-networks is in NP (Bennett, 1996). By showing that the decidability problem 3SAT—i.e. the problem of deciding whether a propositional formula in CNF with clauses that contain at most 3 literals—is reducible to the satisfiability of RCC8-networks, the NP-hardness follows (Renz and Nebel, 1999). Consequently, testing the satisfiability of arbitrary RCC8-networks is NP-complete and therefore a computationally intensive task. Tractability of the satisfiability of RCC8-networks can be gained by restricting the labels to a specific subclass of all RCC8-relations Rel_{RCC8}. A maximally tractable subset of RCC8-relations in the sense that the satisfiability test is polynomial in time complexity is defined by Renz and Nebel (1999). If one constrains the RCC8-networks to so-called conjunctive RCC8-networks (Grigni et al., 1995), i.e. networks that contain only a base relation from $\mathcal{B}_{RCC8}$ or the whole set $\mathcal{B}_{RCC8}$ as label, then the complexity of the satisfiability test can be more specifically described as lying in NC (Nebel, 1995). Intuitively, NC (Nick's Class) is the class of problems that are decidable in polylogarithmic time on a parallel computer with a polynomial number of processors.[6] This can be made precise by boolean circuit complexity. NC is the class of all problems that can be decided by a uniform system of Boolean circuits with a polylogarithmic depth and polynomial size (polynomial number of gates.) If one restricts in this definition the depth to a constant, the complexity class AC^0, which will be used in the propositions below, is obtained. AC^0 is the class of problems that can be decided instantly (in constant time) with the help of polynomially many processors.

2.2 DL-Lite + UNA

DL-Lite denotes a family of light weight description logics that are tailored towards reasoning over ontologies with large ABoxes. We will focus on the member of the DL-Lite family allowing functional roles, role hierarchies and role inverses. The syntax is given in Figure 2.[7] The semantics of the logic is defined in the usual Tarskian style with the additional constraint of the unique name assumption (UNA): Different constants are mapped to different elements in the domain of the interpretations.[8]

Logics of the DL-Lite family have the remarkable property that checking the satisfiability of ontologies as well as answering queries issued to ontologies can be reduced to model checking. As in the logical perspective a relational database is nothing else than a finite interpretation (model), DL-Lite thus offers in particular the possibility to keep the ABox data in a relational database and reduce consistency checks and query answering to SQL queries w.r.t. the database. These properties of DL-Lite are formally described by

[6] Though it is known that $NC \subseteq P$, it is not known whether $NC \subsetneq P$.

[7] Note that we excluded qualified existential restrictions on the right hand side of the TBox axioms. This is no essential restriction on the expressibility as an axiom of the form $B \sqsubseteq \exists R.C$ can be formulated equisatisfiably by $\{A \sqsubseteq \exists R_{new}, \exists R_{new}^{-} \sqsubseteq C, R_{new} \sqsubseteq R\}$ (Calvanese et al., 2009, S.286)

[8] The UNA is needed for FOL-rewritablity (Calvanese et al., 2009, Theorem 6.6).

Let RN be the set of role symbols and $P \in RN$, CN be a set of concept symbols and $A \in CN$, $Const$ be a set of individual constants and $a, b \in Const$.

$$\begin{array}{rcl} R & \longrightarrow & P \mid P^{-} \\ B & \longrightarrow & A \mid \exists R \\ C & \longrightarrow & B \mid \neg B \\ TBox & : & B \sqsubseteq C, (\text{funct } R), R_1 \sqsubseteq R_2 \\ ABox & : & A(a), R(a,b) \end{array}$$

In order to keep the complexity low, the interplay of functionality assertions and inclusion axioms is restricted in the following way: If R occurs in a functionality assertion, then R and its inverse do not occur on the right hand side of a role inclusion axiom.

Figure 2: DL-Lite

the term *first order logic rewritability* or *FOL-rewritability* for short. In order to give the definitions, we have to introduce some further concepts.

An *FOL-query* $Q = \psi(\vec{x})$ is an FOL-formula $\psi(\vec{x})$ whose free variables are the ones in the n-ary vector of variables $\vec{x}$; the variables in $\vec{x}$ are called *distinguished variables*. If $\vec{x}$ is empty, the FOL-query is called boolean. For an ontology $\mathcal{O}$ let $Sign(\mathcal{O})$ denote the signature of $\mathcal{O}$, i.e., the set of concept symbols, role symbols and constants in $Sig(\mathcal{O})$. If $\vec{a}$ is a vector of constants, we write $\vec{a} \in Sig(\mathcal{O})$ to denote the fact that all components of $\vec{a}$ appear in some axiom of $\mathcal{O}$. The semantics of n-ary FOL-queries with respect to an interpretation $\mathcal{I}$ is given by the set $Q^{\mathcal{I}}$ of n-ary tuples $\vec{d}$ over the domain $\Delta^{\mathcal{I}}$ such that $\mathcal{I}_{[\vec{x} \mapsto \vec{d}]} \models \psi(\vec{x})$. The semantics of FOL-queries w.r.t. an ontology $\mathcal{T} \cup \mathcal{A}$ is given by the set of certain answers $cert(Q, \mathcal{T} \cup \mathcal{A})$. This set consists of n-ary tuples of constants $\vec{a} \in Sig(\mathcal{O})$ such that $\psi[\vec{x}/\vec{a}]$ (i.e. the formula resulting from $\psi(\vec{x})$ by applying the substitution $[\vec{x}/\vec{a}]$) follows from the ontology.

$$cert(\psi(\vec{x}), \mathcal{T} \cup \mathcal{A}) \quad = \quad \{\vec{a} \mid \mathcal{T} \cup \mathcal{A} \models \psi[\vec{x}/\vec{a}]\}$$

Two well investigated subclasses of FOL-queries are *conjunctive queries (CQ)* and *unions of conjunctive queries (UCQ)*. A conjunctive query is a FOL-query in which $\psi(\vec{x})$ is an existentially quantified conjunction of atomic formulas at_j, $\psi(\vec{x}) = \exists \vec{y} \bigwedge_i at_i(\vec{x}, \vec{y})$. The more general UCQs allow disjunctions of conjunctive queries, i.e. $\psi(\vec{x})$ can have the form

$$\psi(\vec{x}) = \exists \vec{y_1} \bigwedge_i at_i^1(\vec{x}, \vec{y_1}) \vee \exists \vec{y_2} \bigwedge_i at_i^2(\vec{x}, \vec{y_2}) \vee \cdots \vee \exists \vec{y_n} \bigwedge_i at_i^n(\vec{x}, \vec{y_2})$$

The existential quantifiers in UCQs are interpreted in the same way as for FOL-formulas (natural domain semantics) and not with respect to a given set of constants or individuals of a specific domain (active domain semantics). This fact has some implausible consequences with respect to the interpretation of the outcomes of queries. Consider an ABox states the affiliation relation of professors and universities:

$$\mathcal{A} = \{\overbrace{\text{profAt}(\text{franz}, \text{tud})}^{\alpha_1}, \overbrace{\text{profAt}(\text{ralf}, \text{tuhh})}^{\alpha_2}\}$$

The TBox states that if someone is in profAt-relation to something, then it is a professor. Moreover, the TBox contains an axiom according to which all professors have a ranking.

$$\mathcal{T} = \{\overbrace{\exists\text{profAt} \sqsubseteq \text{Professor}}^{\tau_1}, \overbrace{\text{Professor} \sqsubseteq \exists\text{hasRanking}}^{\tau_2}\}$$

As there are no ranking data in the ABox, we get a conflict between the second axiom of the TBox τ_2 and the ABox that leads to unexpected answers of the following two queries:

$$\begin{aligned} Q_1 &= \exists y.\text{hasRanking}(x, y) \\ Q_2 &= \text{hasRanking}(x, y) \end{aligned}$$

The set of answers of Q_1 is just the set of all professors mentioned in the ABox, formally $cert(Q_1, \mathcal{T} \cup \mathcal{A}) = \{\text{franz}, \text{ralf}\}$, the reason lying in the existence of the second TBox axiom. But the set of answers to Q_2 is the empty set, $cert(Q_2, \mathcal{T} \cup \mathcal{A}) = \emptyset$, the reason being the fact that there are no rankings mentioned in the ABox for the professors. This implausible divergence between Q_1 and Q_2 could be overcome by providing for Q_2 some kind of pinpointing or explanation which produces a new constant indexed with the axioms needed to prove the existence of variable bindings, e.g.,

$$cert^{expl}(Q_2, \mathcal{T} \cup \mathcal{A}) = \{(\text{ralf}, rank_{\tau_1,\tau_2,\alpha_2}), (\text{franz}, rank_{\tau_1,\tau_2,\alpha_1})\}$$

Though this approach to cope with the implausibilities can be developed in an intuitive query semantics, here, we will proceed in a different direction and consider a weaker query language GCQ^+ (see Definition 4 below) which allows for existential quantifiers interpreted in natural domain semantics only if, informally speaking, they are embedded in a tree like structure.

With the technical notions introduced so far we are in a position to give the definition for FOL-rewritability. In the following, let the canonical model of an ABox $\mathcal{A}$, denoted $DB(\mathcal{A})$, be the minimal Herbrand model of $\mathcal{A}$. *Checking the satisfiability of ontologies is FOL-rewritable* iff for all TBoxes $\mathcal{T}$ there is a boolean FOL-query $Q_\mathcal{T}$ such that for all ABoxes $\mathcal{A}$ it is the case that the ontology $\mathcal{T} \cup \mathcal{A}$ is satisfiable just in case the query $Q_\mathcal{T}$ evaluates to false in the model $DB(\mathcal{A})$.

Answering queries from a subclass $\mathcal{C}$ of FOL-queries w.r.t. to ontologies is FOL-rewritable iff for all TBoxes $\mathcal{T}$ and queries $Q = \psi(\vec{x})$ in $\mathcal{C}$ there is a FOL-query $Q_\mathcal{T}$ such that for all ABoxes $\mathcal{A}$ it is the case that $cert(Q, \mathcal{T} \cup \mathcal{A}) = Q_\mathcal{T}^{DB(\mathcal{A})}$.

For DL-Lite, it can be shown that the satisfiability check is FOL-rewritable. If, e.g., $\mathcal{T} = \{A \sqsubseteq \neg B\}$ and $\mathcal{A} = \{A(a), B(a)\}$, then the satisfiability test would be carried out by querying $Q_{\mathcal{T}} = \exists x.A(x) \wedge B(x)$ w.r.t. $DB(\mathcal{A})$, resulting in the answer yes and indicating that $\mathcal{T} \cup \mathcal{A}$ is unsatisfiable. Moreover, answering UCQs in DL-Lite can be shown to be FOL-rewritable. FOL-rewritability of satisfiability is a prerequisite for answering queries because in case the ontology is not satisfiable the set of certain answers is identical to all tuples of constants in $Sig(\mathcal{O})$.

The main technical tool for proving the rewritability results is the chase construction known from database theory. The idea of the chase construction is to repair the ABox with respect to the constraints formulated in the TBox. If, e.g., the TBox contains the axiom $A_1 \sqsubseteq A_2$ and the ABox contains $A_1(a)$ but not $A_2(a)$, then it is enriched by the atom $A_2(a)$. This procedure is applied stepwise to yield a sequence of ABoxes S_i starting with the original ABox as S_0. The resulting set of ABox axioms $\bigcup S_i$ may be infinite but induces a canonical model $can(\mathcal{O})$ for the ABox and the axioms of the TBox that are used in the chasing process (see below). We will summarize the chase construction for DL-Lite as we will use it in our proofs.

Let $\mathcal{T}$ be a DL-Lite TBox, let $\mathcal{T}_p$ denote the subset of positive inclusion axioms (no negation symbol allowed) in $\mathcal{T}$ and let $\mathcal{A}$ be an ABox and $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$. Chasing will be carried out with respect to positive inclusion axioms only. Let $S_0 = \mathcal{A}$. Let S_i be the set of ABox axioms constructed so far and α be a positive inclusion axiom in $\mathcal{T}_p$. Let α be of the form $A_1 \sqsubseteq A_2$ and let $\beta \in S_i$ an ABox axiom. The positive inclusion axiom α is called applicable to β if β is of the form $A_1(a)$ and there is no ABox axiom $A_2(a)$ in S_i. The applicability of positive inclusion axioms of the other forms are defined similarly.[9]

As there may be many possible applications of positive inclusion axioms to atoms[10], one has to impose an order on the TBox axioms and the assertions in the ABox. So we assume that all strings over the signature $Sig(\mathcal{O})$ of the ontology and some countably infinite set of new constants $Const_{chase}$ are well ordered. Such a well ordering exists and has the order type of the natural numbers $\mathbb{N}$ because the set of strings w.r.t. $Sig(\mathcal{O}) \cup Const_{chase}$ is countable.[11] If there is a positive inclusion axiom α applicable to an atom β in S_i, one takes the minimal pair (α, β) with respect to the ordering and produces the next level $S_{i+1} = S_i \cup \{\beta_{new}\}$; here β_{new} is the atom that results from applying the chase rule for (α, β) as listed in Figure 3. The primed constants in the figure are the chasing constants from $Const_{chase}$.

The chase is defined as the union of all the ABox axioms, $chase(\mathcal{O}) = chase(\mathcal{T}_p \cup \mathcal{A}) = \bigcup_{i \in \mathbb{N}} S_i$. The canonical model $can(\mathcal{O})$ is the minimal Herbrand model of $chase(\mathcal{O})$.

The essential property of the canonical model $can(\mathcal{O})$ is that it is a universal model of $\mathcal{T}_p \cup \mathcal{A}$ with respect to homomorphisms, i.e., $can(\mathcal{O}) \models \mathcal{T}_p \cup \mathcal{A}$ and $can(\mathcal{O})$ can be mapped homomorphically to all models of $\mathcal{T}_p \cup \mathcal{A}$. As existentially quantified positive sentences are invariant under homomorphisms, this property has the consequence that every UCQ Q issued to $\mathcal{T}_p \cup \mathcal{A}$ can be answered by computing $Q^{can(\mathcal{O})}$.

[9] For the whole definition see (Calvanese et al., 2009, Definition 4.1, p. 287).

[10] More than one positive inclusion axiom may be applicable to the same ABox axiom, and the same positive inclusion axiom may be applicable to more than one ABox axiom.

[11] This ordering is different from the one of Calvanese et al. (2009) but has the advantage that it can be used also for infinite ABoxes.

If $\alpha = A_1 \sqsubseteq A_2$ and $\beta = A_1(a)$ then $\beta_{new} = A_2(a)$

If $\alpha = A_1 \sqsubseteq \exists R$ and $\beta = A_1(a)$ then $\beta_{new} = R(a, a')$

If $\alpha = \exists R \sqsubseteq A$ and $\beta = R(a, b)$ then $\beta_{new} = A(a)$

If $\alpha = \exists R_1 \sqsubseteq \exists R_1$ and $\beta = R_1(a, b)$ then $\beta_{new} = R_2(a, a')$

If $\alpha = R_1 \sqsubseteq R_2$ and $\beta = R_1(a, b)$ then $\beta_{new} = R(a, b)$

Figure 3: Chasing rules for DL-Lite

The chasing is carried out with respect to the positive axioms in the TBox $\mathcal{T}$, and this is indeed sufficient as Calvanese et al. (2009) have shown. More concretely, (some finite closure of) the negative inclusions axioms and the functionality axioms are only relevant for checking the satisfiability of the ontology. With induction on the stepwise construction of the chase they can show that $can(\mathcal{O})$ is a model of the whole ontology $\mathcal{O}$ iff the negative inclusion axioms and functionality axioms are in accordance with the original ABox—and this can be tested by a simple FOL-query.

The idea of introducing the concept of FOL-rewritability is motivated by the demand to enable computationally feasible reasoning services over large ABoxes. Because the size of the TBox (and the queries) is small with respect to the size of the ABoxes, the computational feasibility is measured with respect to the size of the ABox alone, thereby fixing all other parameters (TBox, query resp.). The resulting type of complexity is called *data complexity*. Aiming at FOL-rewritability is indeed a successful venture with respect to computational feasibility. This is due to the fact that the data complexity of answering FOL-queries is in the low boolean circuits complexity class AC^0. This fact can be used contrapositively to deduce that a given reasoning service (satisfiability check or query rewriting) is not FOL-rewritable if its data complexity is not in AC^0.

3 Combinations of lightweight DLs with RCC8 that are not FOL-rewritable

The NP-completeness of consistency tests for RCC8-constraint networks poses a severe problem when trying to define tractable or—even stronger—FOL-rewritable spatio-thematic description logics that use the RCC8-calculus as the spatial domain. The main challenge in constructing a computationally tractable logic is to restrict the way the concrete domain can be accessed from within the logic; one has—so to speak—to control the flow of information from the spatial domain to the abstract thematic domain of the underlying lightweight logic. As will be shown in this section, a naïve combination of ω-admissible domains with any simple FOL-rewritable logic already results in logics that inherit the high complexity of the concrete domain. This fact is formalized in the propositions below which demonstrate that arbitrary complex network structures can be generated by the TBox-axioms in the naïvely combined logic.

We intend to construct spatial-thematic description logics in which the combinations between the thematic abstract domain and the spatial domain are maintained by constructors available in the logic $\mathcal{ALC}$(RCC8) of Lutz and Miličić (2007). Our approach diverges from the one of Lutz and Miličić (2007) in the point that we do not presuppose an ω-admissible domain but some finite set of FOL-sentences that has the corresponding properties of ω-admissible domains. So we explicitly represent the axioms of the domain rather than making calls to an oracle. The main reason for this shift from a concrete domain to a theory is the fact that it is simpler to use known techniques for query answering (e.g. the chase construction) with respect to some axioms than with respect to a concrete domain. Formally, let *Rel* be a finite set of binary relation symbols, *Const* be a set of constants and T_ω be a finite set of sentences with respect to a signature containing *Rel* and *Const*. A network $\mathcal{N}$ is a set of sentences over $Rel \cup Const$ of the form $r_1(a^*, b^*) \vee \cdots \vee r_k(a^*, b^*)$ for $r_1, \ldots, r_k \in Rel$ and $a^*, b^* \in Const$. The network $\mathcal{N}$ is called complete if it contains only atomic sentences $r(a^*, b^*)$ and for all constants a^*, b^* in $\mathcal{N}$ there is a $r \in Rel$ such that $r(a^*, b^*) \in \mathcal{N}$. For two complete finite networks $\mathcal{N}, \mathcal{M}$ let $I_{\mathcal{N},\mathcal{M}}$ denote the atoms $r(a^*, b^*) \in \mathcal{N}$ such that a^*, b^* occur in both $\mathcal{N}$ and $\mathcal{M}$. The restriction $\mathcal{N}_{Const'}$ of a network to the set of constants $Const'$ is the subset of $\mathcal{N}$ restricted to those sentences containing only constants from $Const'$.

T_ω is an ω-admissible theory iff it fulfills the following conditions:

1. Satisfiability: T_ω is satisfiable.
2. JEPD-Property: T_ω implies the JEPD-property for the relations in *Rel*.
3. Decidability: Testing whether a finite complete syntactic network $\mathcal{N}$ is satisfiable with respect to T_ω, i.e., testing whether $T_\omega \cup \mathcal{N}$ is satisfiable, is decidable.
4. Patchwork Property: If $\mathcal{N}, \mathcal{M}$ are finite complete networks that are satisfiable relative to T_ω, respectively, and if $I_{\mathcal{N},\mathcal{M}} = I_{\mathcal{M},\mathcal{N}}$, then $\mathcal{N} \cup \mathcal{M}$ is satisfiable relative to T_ω, too.
5. Compactness: A complete network $\mathcal{N}$ is satisfiable relative to T_ω iff for every finite set of constants X occurring in $\mathcal{N}$ the restriction $\mathcal{N}_X$ is satisfiable relative to T_ω. (This property is trivially fulfilled by all FOL-theories because FOL has the compactness property.)

We will especially look at the ω-admissible theory of the Region Connection Calculus (Randell et al., 1992). A problematic axiom within the Region Connection Calculus is the axiom of non-atomicity $\forall x \exists y \mathsf{ntpp}(y, x)$ which says that all objects have a non-tangential proper part and which therefore implies that there are no atomic regions. This axiom enforces all models of RCC to be infinite. This axiom does not follow from the other axioms or the definition of C but is explicitly stated.[12] Rather than using the original axioms (Randell et al., 1992) which are based on a reflexive symmetric connectedness-relation C, we use the axioms that directly state that the eight base relations $\mathcal{B}_{RCC8}$ have

[12] Perhaps it is possible to integrate the non-atomicity axiom in the following considerations. We decided to skip it in order to simplify the proofs.

the JEPD-property together with the axioms corresponding to the composition table and the axiom $\forall x \mathsf{eq}(x,x)$. This theory is named Ax_{RCC8} and is shown in Figure 4; it is weaker than the theory of sentences over the signature $\mathcal{B}_{RCC8}$ that follow from the original set of axioms but it can be better integrated into the chasing process.

$$\{\forall x,y.\ \bigvee_{r \in \mathcal{B}_{RCC8}} r(x,y)\} \qquad \text{(joint exhaustivity)}$$

$$\cup \{\forall x,y.\ \bigwedge_{r_1,r_2 \in \mathcal{B}_{RCC8}, r_1 \neq r_2} r_1(x,y) \rightarrow \neg r_2(x,y)\} \qquad \text{(pairwise disjointness)}$$

$$\cup\, \forall x,y,z. r_1(x,y) \wedge r_2(y,z) \rightarrow r_3^1(x,z) \vee \cdots \vee r_r^k(x,z) \mid r;s = \{r_3^1,\ldots,r_3^k\}\} \qquad \text{(weak composition axioms)}$$

$$\cup \{\forall x. \mathsf{eq}(x,x)\} \qquad \text{(reflexivity of } \mathsf{eq}\text{)}$$

Figure 4: Axiom set Ax_{RCC8}

All ABoxes that we will consider here can contain axioms of the form $A(a)$ for some concept symbol A or $R(a,b)$ for some role symbol R or $r(a^*,b^*)$ for a relation symbol r from T_ω or $loc(a,a^*)$; a,b are individual constants intended to denote abstract thematic objects; a^*,b^* are constants intended to denote objects in the theory T_ω. Let $\mathcal{N}_\mathcal{A}$ denote the constraint network encoded in $\mathcal{A}$, i.e., $\mathcal{N}_\mathcal{A} = \{r(a^*,b^*) \mid r(a^*,b^*) \in \mathcal{A}, r \in \mathcal{B}_{RCC8}\}$.

As testing the satisfiability of arbitrary[13] RCC8 constraint networks is not FOL-rewritable, the envisioned combination of some lightweight DL with the RCC8-domain cannot be expected to be FOL-rewritable in the standard sense of FOL-rewritability. If we can presume that the ABox is complete and consistent with respect to the RCC8-knowledge, we have a chance to test the satisfiability of ontologies in the envisioned combined logic with some FOL-query posed to the ABox. Consider, e.g., the simple boolean query $Q = \mathsf{ntpp}(a,b)$ which asks whether regions a,b in the database are related such that a is a non-tangential proper part of b. The composition axiom for the pair $(\mathsf{ntpp},\mathsf{ntpp})$ states that ntpp is a transitive relation; but the transitiveness condition can not be compiled into a finite FOL-query. Intuitively, at least one would have to take into account all ntpp-paths from a,b, i.e., one would have to query the database with queries $Q'_n = \exists x_1 \ldots \exists x_n . \mathsf{ntpp}(a,x_1) \wedge \mathsf{ntpp}(x_1,x_2) \wedge \cdots \wedge \mathsf{ntpp}(x_{n-1},x_n), \mathsf{ntpp}(x_n,b)$ for all $n \in \mathbb{N}$, because the database may be of the form $\{\mathsf{ntpp}(a,c_1),\mathsf{ntpp}(c_1,b)\}$ or of the form $\{\mathsf{ntpp}(a,c_1),\mathsf{ntpp}(c_1,c_2),\mathsf{ntpp}(c_2,b)\}$ etc. Therefore we define the following completeness and consistency condition for ABoxes and weaken the notion of FOL-rewritability of satisfiability to FOL-rewritability of satisfiability with respect to these ABoxes. An ABox $\mathcal{A}$ is called *spatially complete* iff $\mathcal{N}_\mathcal{A}$ is a complete and satisfiable constraint network. We hope that presuming ABoxes to be complete in this sense may allow for FOL-rewritability of satisfiability checks and query answering.

Though the spatial completeness of ABoxes reduces the set of inconsistencies with respect to a TBox, a naïve combination with simple description logics may lead to ontologies that still do not allow for FOL-rewritable satisfiability checks.

[13] Clearly, if the RCC8 network contains only base relations and is complete, then the satisfiability check comprises nothing more than looking up the entries in the composition table.

Proposition 1. *Consider the following simple description logic, called* $\mathcal{L}_{\mathcal{F}}^{0}(RCC8)$*: Let* U_i *denote paths of maximal length 2 which may contain abstract features, i.e.* $U_i = loc$ *or* $U_i = R \circ loc$*.* $\mathcal{L}_{\mathcal{F}}^{0}(RCC8)$ *allows for atomic concepts on the left-hand side of TBox axioms, and it allows for* $\exists U_1, U_2.r$ *constructs*[14] *on the right-hand side.*

$$\begin{array}{rcl} U & \longrightarrow & R \mid R \circ loc \\ C_l & \longrightarrow & A \\ C_r & \longrightarrow & \exists U_1, U_2.r \text{ for } r \in Rel_{RCC8} \\ TBox & : & C_l \sqsubseteq C_r, (\text{funct } R), (\text{funct } loc) \\ T_\omega & = & Ax_{RCC8} \end{array}$$

The data complexity for checking satisfiability of $\mathcal{L}_{\mathcal{F}}^{0}(RCC8)$ *ontologies with respect to spatially complete ABoxes is in* NP*, and therefore satisfiability checking is not FOL-rewritable.*

Proof. We construct a generic TBox that allows one to encode any RCC8 constraint network so that checking the consistency of RCC8-constraint networks is reducible to a satisfiability check of this TBox and a spatially complete ABox. Because the consistency check for general RCC8-networks is NP-complete, the data complexity of checking satisfiability of ontologies in the DL $\mathcal{L}_{\mathcal{F}}^{0}(RCC8)$ must also be in NP.

Let Rel_{RCC8} be the set of all 2^8 RCC8-relations and let, for every $r \in Rel_{RCC8}$, be given role symbols R_r^1, R_r^2. The generic TBox $\mathcal{T}_g$ has for every $r \in Rel_{RCC8}$ a concept symbol A_r and a corresponding axiom with the content that all instances of A_r have paths over the abstract features R_1 resp. R_2 to regions that are r-related.

$$\mathcal{T}_g = \{A_r \sqsubseteq \exists (R_r^1 \circ loc), (R_r^2 \circ loc).r, (\text{funct } loc, R_r^1, R_r^2) \mid r \in Rel_{RCC8}\}$$

Now, let $\mathcal{N}$ be an arbitrary RCC8 constraint network which has to be tested for relational consistency. We define an ABox $\mathcal{A}_{\mathcal{N}}$ such that for every $r(a,b)$ in $\mathcal{N}$ three new constants are introduced: x_{ab}, x_a, x_b. The constants x_a, x_b are intended to denote abstract pendants of a and b and x_{ab} is an abstract object that instantiates A_r—thereby forcing the relation r between a and b.

$$\mathcal{A}_{\mathcal{N}} = \{A_r(x_{ab}), R_r^1(x_{ab}, x_a), R_r^2(x_{ab}, x_b) \mid r(a,b) \in \mathcal{N}\}$$

The size of the ABox $\mathcal{A}_{\mathcal{N}}$ is linear in the size of $\mathcal{N}$, and $\mathcal{A}_{\mathcal{N}}$ can be constructed from $\mathcal{N}$ using only logarithmic space. The construction immediately implies the following fact:

$$\mathcal{T}_g \cup \mathcal{A}_{\mathcal{N}} \cup Ax_{RCC8} \text{ is satisfiable iff } \mathcal{N} \cup Ax_{RCC8} \text{ is satisfiable}$$

If the data complexity of the satisfiability check for $\mathcal{L}_{\mathcal{F}}^{0}(RCC8)$-ontologies were in AC^0, then the consistency of constraint networks could be tested in AC^0, too. Note that $\mathcal{T}_g$ is

[14] The semantics of $\exists U_1, U_2.r$ under an interpretation $\mathcal{I}$ is defined according to the semantics described by Lutz and Miličić (2007). According to this semantics, for example, $(\exists (R \circ loc), (loc).r)^{\mathcal{I}}$ is the set $\{d \in \Delta^{\mathcal{I}} \mid$ there exist $e, d', e' \in \Delta^{\mathcal{I}}$ s.t. $(d,e) \in R^{\mathcal{I}}, (e,e') \in loc^{\mathcal{I}}, (d,d') \in loc^{\mathcal{I}}$ and $(e',d') \in r^{\mathcal{I}}\}$.

a fixed TBox. But checking the consistency of RCC8 constraint networks is NP-hard. As NLogSpace $\subseteq$ NP, every NP-hard problem is NLogSpace-hard, too. But because AC^0 $\subsetneq$ NLogSpace, we get a contradiction. □

The reason for this negative result is the fact that abstract features make it possible to identify the regions of the abstract objects, and therefore arbitrary constraint networks can be constructed. One can circumvent this problem by defining a combined logic that forbids abstract features; but if, on the other hand, one allows for the use of $\forall U_1, U_2.r$-constructs on the right hand side of TBox axioms as well as inverse roles, one can use almost the same construction to get a similar negative result on FOL-rewriting of satisfiability. Hereby, $\forall U_1, U_2.r$ denotes in an interpretation $\mathcal{I}$ all elements which have only paths along U_1 resp. U_2 to objects that are related by the RCC8-relation r.

Proposition 2. *Consider the following simple description logic, called $\mathcal{L}^1_{\mathcal{F}}(RCC8)$ which allows only atomic concepts on the left-hand side but allows for $\forall U_1, U_2.r$ constructs with paths U_i of maximal length 2 on the right-hand side:*

$$\begin{array}{rcl} U & \longrightarrow & R \mid R \circ loc \\ C_l & \longrightarrow & A \mid \exists R \mid \exists R^{-1} \mid \exists loc \\ C_r & \longrightarrow & \forall U_1, U_2.r \textit{ for } r \in Rel_{RCC8} \\ TBox & : & C_l \sqsubseteq C_r, (\text{funct } loc) \\ T_\omega & = & Ax_{RCC8} \end{array}$$

The data complexity for checking satisfiability of $\mathcal{L}^1_{\mathcal{F}}(RCC8)$ ontologies with respect to spatially complete ABoxes is in NP*, and therefore satisfiability checking is not FOL-rewritable.*

Proof. The generic TBox $\mathcal{T}_g$ contains for every $r \in Rel_{RCC8}$ an atomic concept A_r and a corresponding axiom saying that all instances of A_r can only have paths over R^r_1 resp. R^r_2 to regions that are r-related. Furthermore, the TBox has axioms which allow only for fillers of R^1_r resp. R^2_r that have locations.

$$\begin{array}{rcl} \mathcal{T}_g & = & \{A_r \sqsubseteq \forall(R^1_r \circ loc), (R^2_r \circ loc).r, (\text{funct } loc) \mid r \in Rel_{RCC8}\} \\ & & \cup \{\exists(R^1_r)^{-1} \sqsubseteq \exists loc, \exists(R^2_r)^{-1} \sqsubseteq \exists loc \mid r \in Rel_{RCC8}\} \end{array}$$

The proof continues in the same way as the proof for Proposition 1. □

As the formula $B \sqsubseteq \forall U_1, U_2.r$ for r can be (modulo Ax_{RCC8}) equivalently written as $\exists U_1, U_2.(\mathcal{B} \setminus r) \sqsubseteq \neg B$, the proposition implies, that the $\exists$-constructs on the left-hand side of axioms can lead to non-FOL-rewritability. This fact has also consequences for the scenario of the city planning bureau from the introduction. If the planning bureau wants to have necessary and sufficient definitions for parks with lakes that touch the park from within, then a resulting TBox could have the form as in Fig. 5.[15] Even such simple definitions of concepts like parks with lakes over the TIGER/Line® database may prohibit FOL-rewritability.

[15] Please note that the concept *ParkWithLake* can be defined by the even simpler axioms $ParkWithLake \sqsubseteq \exists parkHasLake$ and $ParkWithLake \sqsupseteq \exists parkHasLake$ if the TBox contains the additional axioms $Lake \sqsubseteq \exists loc$ and $Park \sqsubseteq \exists loc$.

$$\begin{array}{rcl} \mathit{ParkWithLake} & \sqsubseteq & \exists(\mathit{parkHasLake}\ \mathit{loc})(\mathit{loc}).\mathsf{tpp} \\ \mathit{ParkWithLake} & \sqsupseteq & \exists(\mathit{parkHasLake}\ \mathit{loc})(\mathit{loc}).\mathsf{tpp} \\ \exists \mathit{parkHasLake} & \sqsubseteq & \mathit{Park} \\ \exists \mathit{parkHasLake}^{-1} & \sqsubseteq & \mathit{Lake} \\ \mathit{Park} & \sqsubseteq & \forall(\mathit{parkHasLake}\ \mathit{loc})(\mathit{loc}).\mathsf{tpp} \end{array}$$

Figure 5: A definition of parks with lakes in $\mathcal{ALC}$(RCC8)

How can we react to such non-rewritability results (Propositions 1, 2) that seem to show that combinations of lightweight description logics with the RCC8 calculus are probably to yield logics that do not allow for FOL-rewritable satisfiability checks and more generally for FOL-rewritable query answering? There are at least four perspectives in this context whose modification could make the tractable combination a more successful venture.

- First of all, one can try to use different descriptions logics that further restrict the interaction of the thematic and the spatial domain. The resulting logics will of course be very weak with respect to the spatial-thematic concepts and the models that can be expressed.[16] We will show that extending DL-Lite-logics with $\exists U_1, U_2.r$-constructs on the right side where U_1 and U_2 may be *loc* or $R \circ loc$ for some role R (not abstract feature) allows for FOL-rewritable satisfiability checks and FOL-rewritable query answering for spatially complete ABoxes and restricted GCQ^+-queries.

- The second perspective concerns the additional knowledge of the special properties of the ABox that one has in advance, i.e., before the reformulation. We already did some modifications concerning this perspective by restricting the ABoxes to be spatially complete, and will look in future work at further stronger completeness conditions which can be fulfilled by some real-world database (or more correctly the virtual ABox induced by the database and the underlying mappings). E.g., the constructions in the propositions above do not work for ABoxes $\mathcal{A}$ which are complete in the sense that all individual constants appearing in $\mathcal{A}$ have locations and the induced network $\mathcal{A}_{\mathcal{N}}$ is a complete and consistent constraint network. This perspective is intended to be more general than exemplified with our completeness notions as it also should incorporate, among others, the notion of combined rewriting (Lutz et al., 2009). In the case of combined rewriting one may use the ABox to merge it with the TBox into some interpretation that can be used as a universal model; this interpretation is then used as the database to which the rewritten queries are issued. That means that in case of the combined rewriting method, one has total knowledge on the ABox before the rewriting step.

  The use of integrity constraints for mappings developed in the thesis of Rodríguez-Muro (2010) is another way to convey information from the ABox.

[16] In this context, one may also try to answer the question whether there is some spatial calculus which by itself has a complexity as low as AC^0, e.g., the very weak logic $RCC3$. The combination of such a spatial calculus with DL-Lite is a potential logic which has data complexity AC^0 itself.

- The third perspective is only relevant for query answering and concerns the expressiveness of the query language in which the query is issued. We argued above that the reformulation idea combined with general (ungrounded) conjunctive queries and undistinguished variables within these type of queries has some unintuitive consequences. We will therefore consider reformulation aspects for querying answers in a fragment of conjunctive queries.
- The fourth perspective concerns the expressiveness of the query language in which the rewritten queries have to be formulated. Using, e.g., Datalog instead of FOL enhances the compilation capabilities.

4 Combinations of lightweight DLs with RCC8 allowing for FOL-rewritability

We consider the following very weak extension DL-Lite(RCC8) of DL-Lite in which concepts of the form $\exists U_1, U_2.r$ may appear on the right-hand side of TBox axioms and in which only the concrete attribute *loc* is allowed to be functional.

$$
\begin{array}{rcl}
R & \longrightarrow & P \mid P^- \\
U & \longrightarrow & R \mid R \circ loc \\
B & \longrightarrow & A \mid \exists R \mid \exists loc \\
C & \longrightarrow & B \mid \neg B \mid \exists U_1, U_2.r \text{ for } r \in Rel_{RCC8} \\
 & & \text{and not simultaneously } U_1 = U_2 = loc \text{ and } \mathtt{eq} \notin r \\
TBox & : & B \sqsubseteq C, (\text{funct } loc), R_1 \sqsubseteq R_2 \\
T_\omega & = & Ax_{RCC8}
\end{array}
$$

Figure 6: The combined logic DL-Lite(RCC8)

The restriction for concepts of the form $\exists U_1, U_2.r$ in Fig. 6 assures that we do not get empty concepts from the beginning (without any interesting deduction over the TBox); clearly, $\exists loc, loc.r$ denotes an empty concept with respect to Ax_{RCC8} if r does not contain the relation eq. We could also handle empty concepts in the rewriting algorithms, but deciding to exclude empty concepts facilitates the rewriting process.

Excluding the special case that $U_1 = U_2 = loc$, one can see that concepts of the form $\exists U_1, U_2.r$ on the right side of TBoxes are not relevant for satisfiability checks; the reason is that at least one of U_1 or U_2 will contain a role symbol that leads to totally new regions which cannot be identified by regions already taken into consideration. In short, DL-Lite(RCC8) does not essentially generate new potential inconsistencies with ABoxes in comparison with the potential inconsistencies of the pure DL-Lite-part because DL-Lite(RCC8) offers only a weak means for restricting the models of the ABox. Therefore it is possible to use the satisfiability check of pure DL-Lite ontologies by calculating the DL-Lite concepts subsuming $\exists U_1, U_2.r$; e.g., if $B \sqsubseteq \exists R \circ loc, loc.r \in \mathcal{T}$, then one has to take into account that $\mathcal{T} \models B \sqsubseteq \exists R$. The resulting proposition which states that

Chasing Rule (R)
If $B(x) \in S_i$ and there are no y, y^*, x^* such that $\{R_1(x, y), loc(y, y*), loc(x, x*), r_1(y^*, x^*)\}$ is contained in S_i, then let S_{i+1} be the superset of S_i that results by adding the elements in $\{R_1(x, y), loc(y, y*), loc(x, x^*), r_1(y^*, x^*)\}$ with individual constants y, y^*, x^*. The individual constants y, y^* are completely new constants not appearing in S_i. The constant x^* is the old x^* if already in S_i, otherwise it is also a completely new constant symbol.

Figure 7: Additional chasing rule that accounts for $\exists U_1, U_2.r$-concepts

checking the satisfiability of DL-Lite(RCC8)-ontologies with spatially complete ABoxes is FOL-rewritable is a corollary of (Calvanese et al., 2009, Theorem 4.14).

Proposition 3. *Checking the satisfiability of DL-Lite(RCC8)-ontologies whose ABox is spatially complete is FOL-rewritable.*

Proof. Let $\mathcal{T} \cup \mathcal{A} \cup Ax_{RCC8}$ be an ontology with a spatially complete ABox $\mathcal{A}$ and the ω-admissible background theory Ax_{RCC8}. We built a simple closure $\mathcal{T}'$ of the pure DL-Lite part of $\mathcal{T}$ in the following way. Every DL-Lite axiom of $\mathcal{T}$ is in $\mathcal{T}'$. For every $B \sqsubseteq \exists R_1 \circ loc, R_2 \circ loc.r \in \mathcal{T}$ let $\{B \sqsubseteq \exists R_1, B \sqsubseteq \exists R_1, B \sqsubseteq \exists loc\} \subseteq \mathcal{T}'$ and similarly for the other axioms of the form $B \sqsubseteq \exists U_1, U_2.r$. We claim that

$$\mathcal{T} \cup \mathcal{A} \cup Ax_{RCC8} \text{ is satisfiable iff the DL-Lite ontology } \mathcal{T}' \cup (\mathcal{A} \setminus \mathcal{N}_\mathcal{A}) \text{ is satisfiable.}$$

The difficult direction is the one from right to left which we will prove in the following. Let $\mathcal{T}' \cup (\mathcal{A} \setminus \mathcal{N}_\mathcal{A})$ be satisfiable by some model $\mathcal{I}$ which can w.l.o.g. be assumed to be the canonical model for the chase $chase(\mathcal{T}'_p \cup (\mathcal{A} \setminus \mathcal{N}_\mathcal{A}))$. Here, $\mathcal{T}'_p$ denotes the positive inclusion axioms in $\mathcal{T}$. As $\mathcal{A}$ is spatially complete, there exists a model $\mathcal{I}' \models \mathcal{N}_\mathcal{A} \cup Ax_{RCC8}$. We let $X = chase(\mathcal{T}'_p \cup \mathcal{A})$ which is satisfiable by an interpretation $\mathcal{J}$ built as a merge of $\mathcal{I}$ and $\mathcal{I}'$. Now, we will extend X by further chasing steps in the following way. We will use the restricted chase construction of Calvanese et al. (2009) as explained in Section 2 on logical preliminaries. Different from the chase construction of Calvanese et al. (2009), we start with the set X which may already be infinite. This fact poses no problems as we chose an ordering over set of all possible strings over the signature of the ontology and the chasing constants.

In addition to the chasing rules listed in Figure 3, we will use the additional chasing rule for axioms of the form $B \sqsubseteq \exists R_1 \circ loc, loc.(r_1 \vee \cdots \vee r_k) \in \mathcal{T}$ and the other axioms of the form $B \sqsubseteq \exists U_1, U_2.R \in \mathcal{T}$ (Figure 4). Let S_i denote the sets created during the chasing process.

Note that we do not take into account the disjunction of the relations $r_1, \ldots, r_k$ but just take the first basic RCC8-relation r_1 of the disjunction. Note further that the functionality of *loc* is directly encoded into the chasing rule. Directly after this chasing rule a completion step is applied in order to make the generated constraint network have a unique model modulo isomorphism. For every node n^* appearing in S_i an atom $r_{n*}(n^*, y^*)$ with $r_{n^*} \in \mathcal{B}_{RCC8}$ being some basic relation is added. We may have infinite many localities already in S_0 but these are not constrained in anyway. We can assume that these nodes are pairwise related by the disjointness relation dc. So, in every chasing step there can be defined two disjoint sets of localities V_i^{dc} and V_i^{fin} with the following properties: For all pairwise distinct nodes in V_i^{dc} it is the case that $\mathsf{dc}(a^*, b^*) \in S_i$ and for all nodes $a^* \in V_i^{\mathsf{dc}}$ and nodes $b^* \in V_i^{fin}$ it is the case that $\mathsf{dc}(a^*, b^*) \in S_i$ and both networks are complete and relationally consistent. Now, the complete constraint network induced by V_i^{fin} is finite and is consistent with $r_1(y^*, x^*)$. This fact follows, e.g., from the patch-work property of ω-admissible theories. We use a path-consistency algorithm or some other appropriate algorithm to find a complete and consistent set induced by $V_i^{fin} \cup \{y^*\}$ that extends the networks induced by V_i^{fin} and y^*, resp. The new node y^* is related to the nodes in V_i^{dc} by dc-edges. This step does not disturb the consistency of the whole network because every composition of some basic relation with dc results in a disjunction which again contains dc. Proceeding in this way, we can finally define the union $\bigcup_{i=0}^{\infty} S_i$ which induces a canonical model of $\mathcal{T} \cup \mathcal{A} \cup Ax_{RCC8}$.

Now we can use the satisfiability test for DL-Lite defined by Calvanese et al. (2009). □

Proposition 3 provides a prerequisite for rewriting queries with respect to ontologies in DL-Lite(RCC8). The query language for which the rewriting is going to be implemented is derived from grounded conjunctive queries and will be denoted by GCQ^+. We chose this query mainly due to two reasons. The first reason relies in the implausible consequences of the semantics of conjunctive queries that we have discussed at the end of the section on DL-Lite. The second reason is the computational unfeasibility of answering CQs that contain base relations of RCC8—even if the ABox is assumed to be complete.

Definition 4. A *GCQ^+-atom w.r.t. DL-Lite(RCC8)* is a formula of one of the following forms:

- $C(x)$, where C is a DL-Lite(RCC8) concept without the negation symbol and x is a variable or a constant.
- $\exists R_1 \dots R_n.C$ for role symbols or inverses of role symbols R_i, a DL-Lite(RCC8) concept without the negation symbol C and a variable or a constant x[17]
- $R(x, y)$ for a role symbol or an inverse of a role symbol R
- $loc(x, y^*)$, where x is a variable or constant and y^* is a variable or constant intended to denote elements of the ω-admissible domain T_ω
- $r(x^*, y^*)$, where $r \in Rel_{RCC8}$ and x^*, y^* are variables or constants intended to denote elements of T_ω

[17]Note that we allow qualified existential restrictions in the GCQ^+-atom. This the reason why we do not just use a conjunction of predicate logical atoms as is done for grounded conjunctive queries. A grounded conjunctive query can model GCQ^+-queries for other logics in which there is no distinction concerning the left-hand side and right-hand side in the following way: If $C(x)$ is GCQ^+-atom, then one can define the grounded atomic query $A(x)$ and extend the TBox with $C \sqsubseteq A$. This is not possible for DL-Lite, as DL-Lite prohibits qualified existential restrictions on the left-hand side of TBox axioms.

A GCQ^+*-query w.r.t. DL-Lite(RCC8)* is a query of the form $\tilde{\exists}\vec{y}\vec{z^*} \bigwedge C_i(\vec{x}, \vec{w}^*, \vec{y}, \vec{z^*})$ where all $C_i(\vec{x}, \vec{w}^*, \vec{y}, \vec{z^*})$ are GCQ^+-atoms and $\tilde{\exists}\vec{y}\vec{z^*} = \tilde{\exists}y_1 \ldots \tilde{\exists}y_n\tilde{\exists}z_1^* \ldots \tilde{\exists}z_m^*$ is a sequence of existential quantifiers that have to be interpreted w.r.t. the active domain semantics.

We want to adapt the algorithm PerfectRef (Calvanese et al., 2009, Fig. 13) for reformulating UCQs w.r.t. DL-Lite ontologies to our setting in which GCQ^+-queries are issued to DL-Lite(RCC8) ontologies. So, the first step for the adaption is to translate GCQ^+-queries into CQs which can be done in the following way. Let $Q = \tilde{\exists}\vec{y}\vec{z^*} \bigwedge C_i(\vec{x}, \vec{w}^*, \vec{y}, \vec{z^*})$ be a GCQ^+. As the $\tilde{\exists}$,-quantifiers in GCQ^+-queries have to be bounded to constants in the ontology we can equivalently write Q by introducing a new concept symbol D intended to denote all individuals in the active domain. All individual constants a in the ABox are supposed to be postulated in the ABox as instances of D, and the query Q is rewritten as

$$Q' = \exists\vec{y}\vec{z^*} \bigwedge C_i(\vec{x}, \vec{w}^*, \vec{y}, \vec{z^*}) \wedge D(y_1) \wedge \cdots \wedge D(y_n) \wedge D(z_1^*) \wedge \cdots \wedge D(z_k^*)$$

Note that the existential quantifiers are now classical existential quantifiers with natural domain semantics. The GCQ^+-atoms $C_i(\vec{x}, \vec{w}^*, \vec{y}, \vec{z^*})$ can be transformed into a union of existentially quantified conjunctions of predicate logical atoms. Here we assume, that $r(x^*, y^*)$ for $r \in Rel_{RCC8}$ is understood as a predicate logical atom. By pushing the existential quantifiers to the front of the query body (and if necessary renaming the variables beforehand) the queries can be transformed into a classical UCQ so that the rewriting algorithm for DL-Lite ontologies can in principle be used for the logic DL-Lite(RCC8), too. Let $\tau_1(Q)$ be the result of the transformation to a UCQ. Using this transformation, we can explain formally the semantics of answering a GCQ^+-query with respect to an ontology $\mathcal{O}$ in the following manner.

$$cert(\mathcal{O}, Q) = cert(\mathcal{O}, \tau_1(Q))$$

For the second step of the adaptation of the PerfectRef algorithm we consider a partial type of transformation $\tau_2(\cdot, \cdot)$ with two arguments which transforms only the atom occurrences given as second argument to classical predicate logical atoms. For example, let $Q = \exists R.A(x) \wedge \exists R_1 \circ loc, loc.\mathsf{tpp}(y)$. Then

$$\begin{aligned}\tau_2(Q, \{\exists R_1 \circ loc, loc.\mathsf{tpp}(y)\}) \quad &= \quad \exists z, z^*, x^*.\exists R.A(x) \wedge R(y, z) \wedge loc(z, z^*) \wedge \\ &\quad\quad loc(x, x^*) \wedge \mathsf{tpp}(z^*, x^*)\end{aligned}$$

The resulting query $\tau_2(Q)$ is a hybrid union of conjunctive queries whose conjuncts are either classical predicate logical atoms or GCQ^+-atoms. The transformation $\tau_2(\cdot)$ with one argument is the same as the transformation τ_1 except that it does not transform the non-DL-Lite-atoms of the form $\exists U_1, U_2.r(x)$, i.e., the transformation $\tau_2(\cdot)$ with one argument can be defined by the binary transformation function $\tau_2(\cdot, \cdot)$ in the following way:

$$\tau_2(Q) = \tau_2(Q, \{at \mid at \text{ is a } GCQ^+\text{-atom in } Q \text{ not of the form } \exists U_1, U_2.r(x)\})$$

The resulting query $\tau_2(Q)$ is a union of of hybrid conjunctive queries whose conjuncts are either classical predicate logical atoms or atoms of the form $\exists U_1, U_2.r(x)$.

input : a hybrid query $\tau_1(Q) \cup \tau_2(Q)$, DL-Lite(RCC8) TBox $\mathcal{T}$
output: a UCQ pr

$pr := \tau_1(Q) \cup \tau_2(Q)$;
repeat
  $pr' := pr$;
  foreach *query* $q' \in pr'$ **do**
    foreach *atom* g *in* q' **do**
      if g *is a FOL-atom* **then** `// Case like in original PerfectRef`
        foreach PI α *in* $\mathcal{T}$ **do**
          if α *is applicable to* g **then**
            $pr := pr \cup \{q'[g/gr(g, \alpha)]\}$;
          end
        end
      else `// Case to capture the` GCQ^+`-queries`
        if g *is of the form* $\exists R_1 \circ loc, R_2 \circ loc.r_3(x)$ *and* $r_1; r_2 \subseteq r_3$ *for* $r_1, r_2, r_3 \in Rel_{RCC8}$ **then**
          $X := q'[g/(\exists R_1 \circ loc, loc.r_1(x) \wedge \exists loc, R_2 \circ loc.r_2(x))]$;
          $pr := pr \cup \{X\} \cup \{\tau_2\big(X, \{\exists R_1 \circ loc, loc.r_1(x), \exists loc, R_2 \circ loc.r_2(x)\}\big)\}$
        end
        if g *is of the form* $\exists U_1, U_2.r_1(x)$ *and* $B \sqsubseteq \exists U_1, U_2.r_2(x) \in \mathcal{T}$ *for* $r_2 \subseteq r_1$ **then**
          $pr := pr \cup \{q'[g/B(x)]\}$;
        end
        if g *is of the form* $\exists U_1, U_2.r_1(x)$ *and* $B \sqsubseteq \exists U_1, U_2.r_2(x) \in \mathcal{T}$ *for* $r_2^{-1} \subseteq r_1$ **then**
          $pr := pr \cup \{q'[g/B(x)]\}$;
        end
        if g *is of the form* $\exists R_1 \circ loc, U_1.r(x)$ *(resp.* $\exists U_1, R_1 \circ loc.r(x)$*) and* *(*$R_2 \sqsubseteq R_1 \in \mathcal{T}$ *or* $R_2^{-1} \sqsubseteq R_1^{-1} \in \mathcal{T}$*)* **then**
          $X := q'[g/(g[R_1/R_2])]$;
          $pr := pr \cup \{X\} \cup \{\tau_2\big(X, \{g[R_1/R_2]\}\big)\}$;
        end
      end
    end
    foreach *pair of FOL-atoms* g_1, g_2 *in* q' **do**
      if g_1 *and* g_2 *unify* **then**
        $pr := pr \cup \{anon(reduce(q', g_1, g_2))\}$;
      end
    end
  end
until $pr' = pr$;
return $drop(pr)$ `// see Algorithm 2`

Algorithm 1: Adapted PerfectRef

The original algorithm PerfectRef operates on the positive inclusion axioms of a DL-Lite ontology by using them as rewriting aids for the atomic formulas in the UCQ. Lines 5–12 and 28–34 of our adapted algorithm (Algorithm 1) make up the original PerfectRef. Roughly, the PerfectRef algorithm acts in the inverse direction with respect to the chasing process. For example, if the TBox contains the positive inclusion axiom $A_1 \sqsubseteq A_2$, and the UCQ contains the atom $A_2(x)$ in a CQ, then the new rewritten UCQ query contains a CQ in which $A_2(x)$ is substituted by $A_1(x)$. The applicability of a positive inclusion axiom to an atom is restricted in those cases where the variables of an atom are either distinguished variables or also appear in another atom of the CQ at hand. To handle these cases, PerfectRef— as well as also our adapted version—uses anonymous variables $\_$ to denote all non-distinguished variables in an atom that do not occur in other atoms of the same CQ. The function anon (line 31 in Algorithm 1) implements the anonymization. The application conditions for positive inclusion axioms α and atoms are as follows:

- α is applicable to $A(x)$ if A occurs on the right-hand side.
- α is applicable to $R(x_1, x_2)$, if $x_2 = \_$ and the right-hand side of α is $\exists R$; or $x_1 = \_$ and the right-hand side of α is $\exists R^-$; or α is a role inclusion assertion and its right-hand side is either R or R^-.

The outcome $gr(g, \alpha)$ of applying an applicable positive inclusion axiom α to an atom g corresponds to the outcome of resolving α with g. For example, if α is $A \sqsubseteq \exists R$ and the atom g is $P(x, \_)$, then the result of the application is $gr(g, \alpha) = A(x)$. We leave out the details here.[18] In the algorithm PerfectRef, atoms in a CQ are rewritten with the positive inclusion axioms (lines 6–11) and if possible merged by the function reduce (line 31) which unifies the atoms with the most general unifier (lines 28–34).

The modification of PerfectRef (which is realized in lines 12–26 of Algorithm 1) concerns the handling of GCQ^+-atoms of the form $\exists U_1, U_2.r(x)$. These atoms may have additional implications that have to be accounted for. At the end of the adapted algorithm PerfectRef (Algorithm 1, line 35) these atoms are deleted by calling the function *drop* (Algorithm 2), so that in the end a classical UCQ results.

The only interesting implications of GCQ^+-atoms of the form $\exists U_1, U_2.r(x)$ that we have to account for are the following:

- The conjunction of $\exists R_1 \circ loc, loc.r_1$ and $\exists loc, R_2 \circ loc.r_2$ is a subconcept of the formula $\exists R_1 \circ loc, R_2 \circ loc.r_3$ where $r_3 \in Rel_{RCC8}$ results as the union of composition table entries $r_1^i; r_2^j$ for $r_1^i \in r_1$ as left and $r_2^j \in r_2$ as right argument (or r_3 is a superset of the composition). I.e., if the formula $\exists R_1 \circ loc, R_2 \circ loc.r_3(x)$ occurs as a conjunct during the rewriting of the original query, then it can be replaced by a conjunct of $\exists R_1 \circ loc, loc.r_1(x)$ and $\exists loc, R_2 \circ loc.r_2(x)$ for all $r_1, r_2 \in Rel_{RCC8}$ such that $r_1; r_2 \subseteq r_3$.
- If $\exists U_1, U_2.r_1(x)$ occurs as conjunct in the query and $B \sqsubseteq \exists U_1, U_2.r_2(x)$ with $r_2 \subseteq r_1$ is in the TBox, then create a new query in which $\exists U_1, U_2.r_1(x)$ is substituted by $B(x)$.

[18]The complete list is given in Fig. 12 in the paper of Calvanese et al. (2009).

```
input  : a hybrid UCQ pr
output: a classical UCQ pr
repeat
    pr' := pr;
    foreach query q' ∈ pr do
        if q' is not a CQ then
            pr := pr \ {q'}
        end
    end
until pr' = pr;
return pr
```

Algorithm 2: Algorithm for dropping GCQ^{+}-atoms

- If $\exists U_1, U_2.r_1(x)$ occurs as conjunct in the query and $B \sqsubseteq \exists U_2, U_1.r_2(x)$ with $r_2^{-1} \subseteq r_1$ is in the TBox, then create a new query in which $\exists U_1, U_2.r_1(x)$ is substituted by $B(x)$.
- If $\exists R_1 \circ loc, U_1.r(x)$ occurs as a conjunct in the query and $R_2 \sqsubseteq R_1$ is in the TBox, then create a new query by substituting $\exists R_1 \circ loc, U_1.r(x)$ with $\exists R_2 \circ loc, U_1.r(x)$.

These cases are handled in lines 12–26 of the algorithm.

As our algorithm is a slight modification of the PerfectRef algorithm, the following proposition concerning the FOL-rewritability can be proved with ideas similar to the ones that are used for the proof of Theorem 5.15 by Calvanese et al. (2009).

Proposition 5. *Answering GCQ^{+}-queries with respect to DL-Lite(RCC8)-ontologies whose ABox is spatially complete is FOL-rewritable.*

Proof.

Calvanese et al. (2009) make heavily use of the chase construction which we have to adapt in order to account for the disjunctions in the general RCC8-relations $r \in Rel_{RCC8}$. The main observation is that the disjunctions in the $\exists U_1, U_2.r$ constructs can be nearly handled as if they were atomic predicate symbols.

Let Q be a n-ary GCQ^{+}-query. If the ontology $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$ is not satisfiable, the set of answers $cert(Q, \mathcal{O})$ has to be the set of all n-ary tuples of constants of the ontology $\mathcal{O}$. But as we have shown in Proposition 3, satisfiability is FOL-rewritable and therefore it can be tested by a SQL-query. So we can assume, that $\mathcal{O}$ is satisfiable.

Let pr be the UCQ resulting from applying the adapted PerfectRef-algorithm to Q and $\mathcal{O}$. We have to show that $cert(Q, \mathcal{O}) = (pr)^{DB(\mathcal{A})}$. In order to prove this equation, we proceed in two main steps. First, we construct a chase-like set $chase^*(\mathcal{O})$, declare what it means to answer Q with respect to $chase^*(\mathcal{O})$, resulting in the set $ans(Q, chase^*(\mathcal{O}))$, and then show that $ans(Q, chase^*(\mathcal{O}))$ is nothing else than the certain answers of Q with respect to $\mathcal{O}$, i.e. $ans(Q, chase^*(\mathcal{O})) = cert(Q, \mathcal{O})$. In the second step, we show that the answers of the UCQ pr with respect to the minimal model $DB(\mathcal{A})$ is the same as $ans(Q, chase^*(\mathcal{O}))$, i.e. $ans(Q, chase^*(\mathcal{O})) = (pr)^{DB(\mathcal{A})}$.

First Step

We construct a chase-like set $chase^*(\mathcal{O})$ that will be the basis for proving the correctness and completeness of our algorithm. We use the chase rules of Figure 3 and the special rule (R) of Figure 4 for the new constructors to build $chase^*(\mathcal{O})$ in the following way. Every time (R) is applied to yield a new ABox S_i, the resulting constraint network in S_i is saturated by calculating the minimal labels between the new added region constants and the other region constants. Note that the application of (R) does not constrain the RCC8-relations between the old regions and even stronger: Let (R) be applied to a TBox axiom of the form $A \sqsubseteq \exists Rloc, loc.r$ and $A(a) \in S_i$ resulting in the addition of the assertions $R(a, b)$, $loc(b, b^*)$ and $r(b^*, a^*)$. Then it is enough to consider all $c^* \in S_i$ and all relations r_{c^*,a^*} such that $r_{c^*,a^*}(c^*, a^*) \in S_i$. In the composition table one looks up the outcome $r_{c^*,a^*}; r = r'_{c^*,b^*}$ and adds $r'_{c^*,b^*}(c^*, b^*)$ to S_i. In the following, this step will be called the step of triangle completion. After the triangle completion step one closes the assertions up with respect to the subset relation between RCC8-relations and with respect to symmetry. I.e., if $r_1(x^*, y^*)$ is added to S_i, then one also adds $r_2(x^*, y^*)$ for all r_2 such that $r_1 \subseteq r_2$ and $r_2^{-1}(y^*, x^*)$. For different c_1^*, c_2^*, assertions of the form $r_{c_1^*,b^*}(c_1^*, b^*)$ and $r_{c_2^*,b^*}(c_2^*, b^*)$ do not constrain each other—as can be shown by using the patch work property. As the number of regions is finite and we excluded the non-atomicity axiom, the saturation leads to a finite set S_{i+k} (for some $k \in \mathbb{N}$) that is a superset of S_i. Let $chase^*(\mathcal{O}) = \bigcup S_i$ be the union of all ABoxes constructed in this way (starting again with $S_0 = \mathcal{A}$). The set $chase^*(\mathcal{O})$ does not induce a single canonical model. But we claim that it is universal in the following sense:

(*) For every model $\mathcal{I}$ of $\mathcal{O}$ one can get a model $\mathcal{I}_c$ out of $chase^*(\mathcal{O})$ by taking a (consistent) configuration of the contained $RCC8$-network and taking the minimal model of this configuration and the thematic part of $chase^*(\mathcal{O})$. Then $\mathcal{I}_c$ maps homomorphically to $\mathcal{I}$.

The claim (*) holds because $\exists U_1, U_2.r$-constructs do not appear on the left-hand side of the positive inclusion axioms in the TBox $\mathcal{T}$, and therefore new information on the $RCC8$-network side cannot be used during the chasing process to produce new information on the thematic part.

We explain what it means to answer a GCQ^+-query with respect to $chase^*(\mathcal{O})$. We transform Q into a CQ $\tau_1(Q)$ where the the general relations $r \in Rel_{RCC8}$ are considered as atomic predicate symbols. E.g., let $Q \exists R.A(x) \wedge \exists R_1 \circ loc, loc.(\mathsf{tpp} \vee \mathsf{ntpp})(y)$. Then

$$\begin{aligned} \tau_1(Q, \{\exists R_1 \circ loc, loc.\mathsf{tpp}(y)\}) \quad &= \quad \exists z, z^*, x^*.\exists R.A(x) \wedge R(x, z) \wedge loc(z, z^*) \wedge \\ &\quad\quad loc(x, x^*) \wedge (\mathsf{tpp} \vee \mathsf{ntpp})(z^*, x^*) \end{aligned}$$

The set of answers $ans(chase^*(\mathcal{O}), Q)$ is defined by homomorphisms of the atoms of $\tau_1(Q)$ into $chase^*(\mathcal{O})$. Let $(x_1, \ldots, x_n)$ be the n-ary tuple of distinguished variables of $Q = \psi(\vec{x})$. $(a_1, \ldots, a_n) \in ans(chase^*(\mathcal{O}), Q)$ iff there is a homomorphism h from $\tau_1(Q)$ into $chase^*(\mathcal{O})$ with $h(x_i) = a_i$ (for $i \in \{1, \ldots, n\}$). The homomorphic image of $\psi(\vec{x})$ of in $chase^*(\mathcal{O})$ is called a witness of ψ w.r.t. $\vec{a}$ in $chase^*(\mathcal{O})$. Clearly, if $\mathcal{I} \models chase^*(\mathcal{O})$ and $\vec{a} \in ans(chase^*(\mathcal{O}), Q)$, then $\mathcal{I} \models \psi[\vec{x}/\vec{a}]$.

We prove that $ans(Q, chase^*(\mathcal{O})) = cert(Q, \mathcal{O})$ by proving both subset relations separately.

$\subseteq$-direction: Let $\vec{a} \in ans(Q, chase^*(\mathcal{O}))$. We have to show $\mathcal{O} \models \psi[\vec{x}/\vec{a}]$. Let $\mathcal{I} \models \mathcal{O}$ and I_c be the model according to claim (*). We have to show $\mathcal{I}_c \models \psi[\vec{x}/\vec{a}]$. Because $\mathcal{I}_c \models \psi[\vec{x}/\vec{a}]$, it follows that $\mathcal{I} \models \psi[\vec{x}/\vec{a}]$.

$\supseteq$-direction: Let $\vec{a} \in cert(Q, \mathcal{O})$. For every $\mathcal{I} \models \mathcal{O}$ consider $\mathcal{I}_c$. All these models $\mathcal{I}_c$ differ at most on the interpretations of the RCC8-Relations which are assigned to regions x^*, y^*. Consider for all x^*, y^* the assertion $r(x^*, y^*)$, $r \in Rel_{RCC8}$ where $r_i \in r$ iff there is $\mathcal{I}_c$ such that $r_i(x^*, y^*)$ is true in $\mathcal{I}_c$. Then $r(x^*, y^*)$ is in $chase^*(Q, \mathcal{O})$. Therefore we will find a homomorphism h from $\psi(\vec{x})$ onto $chase(Q, \mathcal{O})$ with $h(\vec{x}) = \vec{a}$.

Second Step

Let pr be the outcome of the adapted PerfectRef-Algorithm applied to Q. We prove $pr^{DB(\mathcal{A})} = ans(chase^*(\mathcal{O}), Q)$ by proving the subset relations separately.

$\subseteq$-direction: Let $q \in pr$ be a conjunctive n-ary query. We have to show $q^{DB(\mathcal{A})} \subseteq ans(chase^*(\mathcal{O}), Q)$. This can be done by induction over the number of steps that are needed in order to construct q in the PerfectRef algorithm. In the base case $q \in \tau_1(Q)$. The assertion follows directly from the fact that $DB(\mathcal{A})$ is contained in $chase^*(\mathcal{O})$. Inductive step. Let $q = q_{i+1}$ and q_{i+1} be the outcome of applying one of the steps in the algorithm to q_i. If the steps are those contained in the original PerfectRef-algorithm we can argue in the same line as in the proof of Lemma 5.13 of Calvanese et al. (2009). In the other cases the induction steps are provable because of the correctness of the implicit deductions.

$\supseteq$-direction: This is the direction showing the completeness of the algorithm. Let $\vec{a} \in ans(Q, chase^*(\mathcal{O}))$. So there is a witness of $\vec{a}$ w.r.t. $\tau_1(Q)$ in $chase^*(\mathcal{O})$. This witness lies in some S_k of the chase $chase^*(\mathcal{O})$ and shall be denoted $\mathcal{G}_k$. We have to find a $q \in pr$ such that it has a witness in the ABox $\mathcal{A}$. This can be proved by considering the pre-witness of $\vec{a}$ with respect to Q in all S_i for $i \leq k$. The pre-witness of $\vec{a}$ with respect to Q in S_i is defined by the following equation:

$$\begin{aligned}\mathcal{G}_i \quad = \quad & \bigcup_{\beta' \in \mathcal{G}_k} \{\beta \in S_i \mid \beta \text{ is an ancestor of } \beta' \text{ in } S_k \text{ and there exists no} \\ & \text{successor of } \beta \text{ in } S_i \text{ that is an ancestor of } \beta' \text{ in } S_k\}\end{aligned}$$

By induction on i (for $i \in \{0, \ldots, k\}$) one can find a $q \in pr$ such that the pre-witness of $\vec{a}$ with respect to Q in S_{k-i} is a witness for q. By induction assumption there is $q' \in pr$ such that $\mathcal{G}_{k-i+1}$ is a witness of $\vec{a}$ w.r.t. q' in S_{k-i+1}. If S_{k-i+1} results from S_{k-i} by application of one of the chase rules in Figure 3, then the argument proceeds in the same manner as in the proof of Lemma 5.13 of Calvanese et al. (2009). Otherwise, S_{k-i+1} is constructed by applying rule (R) or one of the saturation steps (triangle completion, upward closure, symmetry closure, resp.) following the application of rule (R). But all these steps have a corresponding case in the algorithm. □

As query answering in DL-Lite(RCC8) is FOL-rewritable, queries like those from the scenario of city planning can be answered correctly and completely by transforming them into SQL-queries and getting the answers from the underlying database. The TBox of the city planning bureau may contain the following axioms which formalize the necessary conditions for parks with lakes and playing areas, resp., within DL-Lite(RCC8).

$$\begin{aligned} \mathit{ParkWithLake} &\sqsubseteq \mathit{Park} \\ \mathit{ParkPlaying} &\sqsubseteq \mathit{Park} \\ \mathit{ParkWithLake} &\sqsubseteq \exists \mathit{hasLake} \circ \mathit{loc}, \mathit{loc}.\mathsf{tpp} \\ \mathit{ParkPlaying} &\sqsubseteq \exists \mathit{hasPlayingArea} \circ \mathit{loc}, \mathit{loc}.\mathsf{tpp} \end{aligned}$$

The ABox $\mathcal{A}$ is derived from GIS-data in a database and additionally contains assertions w.r.t. to an object a; the object a is still not localized with respect to the data in the database but is specified to be a park containing a lake (touching the park from within) and a playing area, i.e., $\{\mathit{ParkWithLake}(a), \mathit{ParkPlaying}(a)\} \subseteq \mathcal{A}$.

The query asking for all parks that have lakes not containing a playing area as island can be expressed as the following GCQ^{+}:

$$Q = \mathit{Park}(x) \wedge \exists \mathit{hasLake} \circ \mathit{loc}, \mathit{hasPlayingArea} \circ \mathit{loc}.\mathcal{B}_{RCC8} \setminus \{\mathsf{ntpp}\}(x)$$

Our adapted reformulation algorithm (Algorithm 1, especially lines following line 12) would produce a UCQ that contains, among others, the following CQ

$$Q' = \mathit{ParkWithLake}(x) \wedge \mathit{ParkPlaying}(x)$$

This query would correctly capture the object a.

5 Conclusion and Outlook

As a resume of this paper we may state that combining lightweight logics with spatial calculi like RCC8 is a non-trivial task that may easily lead to logics that are not appropriate to handle large GIS-data bases. This is even the case when additional assumptions on the completeness of the database resp. the (virtual) ABox are made (Propositions 1 and 2).

Weakening the interaction between the thematic and spatial domain may lead to logics like the one we termed DL-Lite(RCC8) in this paper that allow FOL-rewritability of satisfiability tests and of query answering (Propositions 3 and Proposition 5). This logic can be used to answer queries over ontologies in which mainly necessary conditions on spatio-thematic concepts are formulated.

DL-Lite(RCC8) is not expressive enough to realize more sophisticated reasoning services with a fine-grained modeling of the domain; we have seen that it is not expressive enough in order to define some simple concepts like the concept of a park containing a lake. Nonetheless, DL-Lite(RCC8) or a slight extension of it called DL-Lite$_{\sqcap}$(RCC8) may be used as a logic for approximating ontologies in more expressive description logics in the same way as done by Kaplunova et al. (2010).

More concretely, we define DL-Lite$_{\sqcap}$(RCC8) as an extension of DL-Lite(RCC8) that allows for conjunctions on the left-hand side and the $\exists U_1, U_2.r$-construct on the left-hand side restricted to the case $r = \mathcal{B}_{RCC8}$, i.e., the following rules are added to the grammar of DL-Lite(RCC8):

$$
\begin{array}{rcl}
B_1 \sqcap B_2 & \rightarrow & C \\
\exists U_1, U_2.\mathcal{B}_{RCC8} & \rightarrow & C
\end{array}
$$

Slight modifications of the proof for the rewritability of query answering with respect to DL-Lite(RCC8) ontologies show that query answering w.r.t. to ontologies in the logic DL-Lite$_{\sqcap}$(RCC8) is rewritable, too. Now, let be given an expressive combined logic $\mathcal{L}_{expr}$(RCC8) (e.g., $\mathcal{ALC}(RCC8)$) that can express an ontology over the TIGER/Line® data like the one defined in Fig. 5, and let $\mathcal{T} \cup \mathcal{A}$ be some ontology in $\mathcal{L}_{expr}$(RCC8). Answering GCQ^+-queries with respect to the ontologies in $\mathcal{L}_{expr}$(RCC8) may not be FOL-rewritable and so the idea is to approximate $\mathcal{T}$ with a TBox $\mathcal{T}'$ in DL-Lite$_{\sqcap}$(RCC8) that is stronger than $\mathcal{T}$, i.e., $\mathcal{T}' \models \mathcal{T}$. In particular, the approximation implies that all answers to the original query q with respect to $\mathcal{T} \cup \mathcal{A}$ is a subset of the answers with respect to $\mathcal{T}' \cup \mathcal{A}$. Answering queries with respect to ontologies in the expressive description logic $\mathcal{L}_{expr}$(RCC8) has thus been reduced to query answering with respect to ontologies in DL-Lite$_{\sqcap}$(RCC8) which we has been shown to be FOL-rewritable. As the approximation results in a complete but not necessarily correct query answering methodology, the set of answers $cert(Q, \mathcal{T}' \cup \mathcal{A})$ has to be verified in a post processing phase with a tableau prover that can process ontologies in $\mathcal{L}_{expr}$(RCC8). Using DL-Lite$_{\sqcap}$(RCC8) instead of DL-Lite (as realized by Kaplunova et al. (2010)) may result in better approximations in the sense that it may refine pure DL-Lite approximations $\mathcal{T}''$ of $\mathcal{T}$, because the following subset relations hold $cert(Q, \mathcal{T} \cup \mathcal{A}) \subseteq cert(Q, \mathcal{T}' \cup \mathcal{A}) \subseteq cert(Q, \mathcal{T}'' \cup \mathcal{A})$ and the last subset relation may be proper.[19]

References

Bennett, B. (1996). Modal logics for qualitative spatial reasoning. *Logic Journal of the IGPL*, 4(1):23–45.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodríguez-Muro, M., and Rosati, R. (2009). Ontologies and databases: The DL-Lite approach. In Tessaris, S. and Franconi, E., editors, *Semantic Technologies for Informations Systems – 5th Int. Reasoning Web Summer School (RW 2009)*, volume 5689 of *Lecture Notes in Computer Science*, pages 255–356. Springer.

Egenhofer, M. (1991). Reasoning about binary topological relations. In Günther, O. and Schek, H.-J., editors, *Advances in Spatial Databases*, volume 525 of *Lecture Notes in Computer Science*, pages 141–160. Springer Berlin / Heidelberg.

[19]This type of approximation from above is sufficient for query answering only if the satisfiability test of $\mathcal{L}_{expr}(RCC8)$ ontologies has been done beforehand because the strengthening of the TBox in the envisioned approximation may introduce new inconsistencies.

Grigni, M., Papadias, D., and Papadimitriou, C. H. (1995). Topological inference. In *IJCAI (1)*, pages 901–907.

Johnson, A. G., Galdi, D. E., Godwin, L., Franz, L., Boudriault, G., and Ratcliffe, M. R. (2009). Tiger/Line® shapefiles. Technical Documentation prepared by the US Census Bureau.

Kaplunova, A., Moeller, R., Wandelt, S., and Wessel, M. (2010). Towards scalable instance retrieval over ontologies. In Yaxin, B. and Mary-Anne, W., editors, *Knowledge Science, Engineering and Management, Fourth International Conference, KSEM 2010, Proceedings*, volume 6291 of *Lecture Notes in Computer Science*. Springer.

Lutz, C. and Miličić, M. (2007). A tableau algorithm for description logics with concrete domains and general TBoxes. *J. Autom. Reasoning*, 38(1-3):227–259.

Lutz, C., Toman, D., and Wolter, F. (2009). Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence IJCAI09*. AAAI Press.

Mackworth, A. K. (1977). Consistency in networks of relations. *Artif. Intell.*, pages 99–118.

Nebel, B. (1995). Computational properties of qualitative spatial reasoning: First results. In Wachsmuth, I., Rollinger, C.-R., and Brauer, W., editors, *KI*, volume 981 of *Lecture Notes in Computer Science*, pages 233–244. Springer.

Randell, D. A., Cui, Z., and Cohn, A. G. (1992). A spatial logic based on regions and connection. In *Proceedings of the 3rd International Conferecence on Knowledge Represnetation and Reasoning*, pages 165–176.

Renz, J. (2002). *Qualitative Spatial Reasoning with Topological Information*, volume 2293 of *Lecture Notes in Computer Science*. Springer.

Renz, J. and Nebel, B. (1999). On the complexity of qualitative spatial reasoning: a maximal tractable fragment of the region connection calculus. *Artif. Intell.*, 108:69–123.

Rodríguez-Muro, M. (2010). *Tools and Techniques for Ontology Based Data Access in Lightweight Description Logics*. PhD thesis, Free University of Bozen - Bolzano.