Computationally Feasible Query Answering over Spatio-thematic Ontologies

Özgür Lütfü Özçep and Ralf Möller
Institute for Software Systems (STS)
Hamburg University of Technology
Hamburg, Germany
Email: {oezguer.oezcep,moeller}@tu-harburg.de

***Abstract*—Providing query answering facilities at the conceptual level of a geographic data model requires deduction, and deduction in geographical information systems (GIS) is a demanding task due to the size of the data that are stored in secondary memory. In particular, this is the case for deductive query answering w.r.t. spatio-thematic ontologies, which provide a logical conceptualization of an application domain involving geographic data. For specific logics (so-called lightweight description logics) and query languages (conjunctive queries) the query answering problem can be solved by compiling the ontology-based query into an SQL query that is posed to the database. Thus, ontology-based query answering becomes as feasible as standard database query answering. In the literature, this kind of query answering by compilation is formalized using the notion of first-order logic (FOL) rewritability. In this paper we show that lightweight description logics such as DL-Lite can be combined with spatial calculi such as the region connection calculus such that FOL rewritability is retained and the expressive power is sufficient for modeling important aspects of GIS data.**

***Keywords*-description logics; qualitative spatial reasoning; deductive query answering; FOL rewritability**

I. Introduction

In almost any area in which geographical information systems (GIS for short) are used, e.g., damage classification for flooding scenarios, development of eco systems in forestry, analysis of sociological and demoscopic aspects in urban areas, or semantic web applications over GIS data [1] there is a need to formalize relevant concepts and relations in a conceptual data model in order to answer queries. The preferred technical tool for providing a conceptualization is an ontology. Ontologies are represented in some logical language (e.g., a description logic) that has a formal semantics and allows for automated reasoning, in particular deductive query answering. The idea of exploiting a conceptualization over the domain is to provide a convenient query language for building applications, given some mappings of basic concepts and relations to expressive database queries are defined in terms of SQL. Mappings to SQL are required because the encodings of spatial information in databases might vary and might be rather cumbersome in practice. This holds in particular for GIS databases. As we will show in this paper, the conceptualization can be used to find GIS database queries (views) relevant for complete query answering.

For example, think of a planning problem for additional parks in New York, say. An engineering office responsible for this task uses geographical data, such as, e.g., the TIGER/Line® data—a well known free set of geographic data from the US Census Bureau. Among many others, the engineering consultants have declared concepts such as (i) a park that covers a lake (*Park+Lake* for short), and (ii) a park *Park4Playing* intended to denote all parks covering playing areas. Imagine the office has defined mappings from these concepts to SQL queries possibly involving GIS extensions in order to find instances of these concepts. Quite a large library of similar concepts and relations with mappings will emerge pretty soon. Moreover, assume that for quality assurance purposes the engineering office would like to formulate queries for identifying objects with certain design flaws, e.g., a park with a lake such that inside the lake there is a playing area. Intuitively, the query searches for parks except those for which there are objects on the righthand side of tuples in the relation *hasLake* and *hasPlayingArea* such that the locations of the objects are related by the proper containment relation. Note that *hasLake* and *hasPlayingArea* are relations declared in the conceptual domain model (see below for a more formal account).

We argue that despite the fact that it is not syntactically apparent, in order to find all parks that contain this type of design flaw (for being complete, that is) one also has to take into consideration the mappings that produce instances of *Park+Lake* and *Park4Playing*. The query rewriting process ensures that query answering w.r.t. the semantics of the ontology is implemented by answering a properly rewritten query using a standard SQL query (see [2] for details).

The goal of this paper is to demonstrate that the SQL queries relevant for complete query answering in our spatial domain can be automatically determined as well, and that qualitative spatial reasoning is required to achieve this. The only additional effort for finding relevant mappings is a very weak axiomatization for spatio-thematic concepts such as *Park+Lake* and *Park4Playing* in the form of necessary conditions that approximate the intended meanings. The axiomatization is part of the conceptual domain model.

For example, in case of *Park+Lake* we have a *Park* that *contains* a *Lake* such that the *Lake touches* the *Park* from within. The formalization of these notions is done with an

ontology language, and the details of the formalization and its application to the sample scenario are given in this paper. The query is automatically compiled into a GIS database query, and the mappings are considered appropriately. Note that query compilation (or rewriting) is a method that cannot be reduced to simple macro expansion but requires reasoning over the axiomatization as demonstrated by the perfect rewriting algorithm of [2].

The contribution of our paper is the result that query rewriting can also be used in GIS scenarios, but rewriting has to be significantly enhanced because the logical language in which the ontology is represented has to be expressive enough in order to represent spatio-thematic concepts. We identify DL-Lite(RCC8) (Section IV) as an appropriate combined logic and show that it allows for the compilation of the ontology into the query, i.e., in technical terms, allows for FOL rewritability of query answering. The query language defined in Section IV and GeoSPARQL (see OGC draft specification, http://www.w3.org/2011/02/GeoSPARQL.pdf) have common features. But in contrast, DL-Lite(RCC8) allows for the controlled use of RCC8 relations in the ontology. Preparing the results of Section IV, Section II introduces the logical components of DL-Lite(RCC8) and Section III describes the obstacles in finding a combination.

II. Logical Preliminaries

The logic DL-Lite(RCC8) to be introduced in this paper is a combination of two logics described in the following subsections: the region connection calculus RCC8, which can model topological relations like that of containment; and a member of the family of lightweight description logics DL-Lite, which is well suited for reasoning over large databases and which can model many elements of the Unified Modeling Language (UML).

A. RCC8-calculus

The Region Connection Calculus (RCC) [3] is one of the most widely known qualitative spatial reasoning calculi that take regions and not points as the basic entities for representing spatial knowledge and reasoning about it. In the axiomatic representation of RCC [3], a primitive binary relation C is intended to model the connectedness relation between regions; C is therefore axiomatically restricted to be reflexive and symmetric. C is used to define different relations between regions that are termed *base relations*. One family of base relations, denoted $\mathcal{B}_{RCC8} = \{$dc (disconnected), ec (externally connected), eq (equal), po (partially overlapping), ntpp (non-tangential proper part), tpp (tangential proper part), ntppi (inverse of ntpp), tppi (inverse of tpp)$\}$ henceforth, is the building block of RCC8. Further calculi of RCC can be defined by considering other sets of base relations. The base relation dc is intended to model disconnectedness and is defined by $\mathsf{dc}(x, y)$ iff $\neg C(x, y)$. The other base relations are defined similarly [3]. The axioms imply that the eight base relations are jointly exhaustive and pairwise exclusive (JEPD property).

With the help of the base relations, real-world spatial configurations can be represented in the form of constraint networks, which can be efficiently processed by constraint satisfaction procedures. A network is defined by a set of formulas 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}. With disjunctions of base relations, indefinite knowledge on spatial relations of regions can be expressed. The networks are labelled graphs derived from the formulas such that the vertices of the network are the constants used in the formulas, and edges (a, b) labelled $\{r_1, \ldots, r_k\}$ are derived iff $\{r_1, \ldots, r_k\}(a, b)$ is contained in the set of formulas.

A practically relevant question is whether a constraint network is satisfiable with respect to the RCC8 axioms. Testing satisfiability of networks can be carried out on the basis of path consistency algorithms [4]. These algorithms are based on composition tables. For every pair of base relations r_1, r_2 they contain an entry for the composition $r_1 \circ r_2$. In general, 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 [5, p. 45] 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., $r_1 \circ r_2 \subseteq r_1; r_2$. For example the weak composition table entry for the pair (tpp, tppi) is $\mathsf{tpp}; \mathsf{tppi} = \{\mathsf{dc}, \mathsf{ec}, \mathsf{po}, \mathsf{tpp}, \mathsf{tppi}, \mathsf{eq}\}$. This composition table entry can be described by the following (implicitly universally quantified) FOL sentence:

$$\mathsf{tpp}(x, y) \wedge \mathsf{tppi}(y, z) \rightarrow \{\mathsf{dc}, \mathsf{ec}, \mathsf{po}, \mathsf{tpp}, \mathsf{tppi}, \mathsf{eq}\}(x, z)$$

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

Testing the satisfiability of arbitrary RCC8 networks is NP-complete and thus computationally intensive [6] [7]. Rather than using the axioms of [3], which are based on the relation C, we use axioms that directly state that the eight base relations $\mathcal{B}_{RCC8}$ have the JEPD property, together with 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 1. Adapting the term of an ω-admissible domain [8], we call Ax_{RCC8} an *ω-admissible theory*.

B. DL-Lite + UNA

DL-Lite denotes a family of lightweight description logics that are tailored towards reasoning over ontologies with large sets of data descriptions. We will focus on the member of

- $\{\bigvee_{r \in \mathcal{B}_{RCC8}} r(x,y)\} \cup$ (joint exhaustivity)
- $\{\bigwedge_{r_1,r_2 \in \mathcal{B}_{RCC8}, r_1 \neq r_2} r_1(x,y) \rightarrow \neg r_2(x,y)\} \cup$ (pairwise disjointness)
- $r_1(x,y) \wedge r_2(y,z) \rightarrow r_3^1(x,z) \vee \cdots \vee r_r^k(x,z) \mid r; s = \{r_3^1, \ldots, r_3^k\}\} \cup$ (weak composition axioms)
- $\{\mathsf{eq}(x,x)\}$ (reflexivity of eq)

Figure 1. Ax_{RCC8}. Formulas are implicitly universally quantified

the DL-Lite family allowing functional roles, role hierarchies and role inverses. The syntax of concept descriptions, axioms (a set of axioms is called TBox), and assertions for describing data (a set of assertions is called ABox) is given in Figure 2. Here P is a role symbol, A a concept symbol and a, b are constants. Moreover, in order to keep query answering complexity low, the interplay of functionality 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. The semantics of the logic is defined in the usual first-order logic style in terms of relational structures, or interpretations $\mathcal{I}$, that satisfy axioms and assertions, with the additional constraint of the unique name assumption (UNA): Different constants are mapped to different elements in the domain of the interpretations. The UNA is needed for FOL rewritability [2, Theorem 6.6].

An ontology $\mathcal{O}$ is a tuple $(Sig, \mathcal{T}, \mathcal{A})$, with a signature Sig (i.e., set of concept symbols, role symbols and constants), with a TBox $\mathcal{T}$, and with an ABox $\mathcal{A}$. An ontology is satisfiable iff there exists an interpretation satisfying $\mathcal{T}$ and $\mathcal{A}$. Given an interpretation $\mathcal{I}$, checking whether $\mathcal{I}$ satisfies $\mathcal{T}$ and $\mathcal{A}$ is called model checking (and $\mathcal{I}$ is called a model if satisfiability is given).

Given an ontology, query answering is a decision problem directly relevant for practical applications. An *FOL query* $Q = \psi(\vec{x})$ is a first-order logic 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 query is called boolean.

Logics of the DL-Lite family have the remarkable property that checking the satisfiability of ontologies as well as answering queries w.r.t. ontologies can be reduced to model checking. Since in the logical perspective a relational database is nothing else than an interpretation (or a finite part of the canonical model, aka Herbrand model, to be more precise), DL-Lite thus offers the possibility to keep data descriptions as a virtual ABox in a relational database and reduce consistency checks and query answering to SQL queries (first-order logic formulas) w.r.t. the database. These properties of DL-Lite are formally described by the term *first-order logic rewritability* or *FOL rewritability* for short.

Some definitions are required to explain this in detail. Let $\vec{a}$ be a vector of constants from the signature of the ontology. The semantics of n-ary FOL queries with respect to an

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

Figure 2. DL-Lite

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 $\vec{a}$ from Sig 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 CQ is a FOL query in which $\psi(\vec{x})$ is an existentially quantified conjunction of atomic formulas $at(\cdot)$, $\psi(\vec{x}) = \exists \vec{y} \bigwedge_i at_i(\vec{x}, \vec{y})$. The UCQs allow disjunctions of CQs, i.e., $\psi(\vec{x})$ can have the form $\exists \vec{y_1} \bigwedge_{i_1} at_{i_1}(\vec{x}, \vec{y_1}) \vee \cdots \vee \exists \vec{y_n} \bigwedge_{i_n} at_{i_n}(\vec{x}, \vec{y_n})$. We conceive a UCQ as a set of CQs. 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 mentioned in the signature (active domain semantics).

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 [2] that the satisfiability check is FOL rewritable. Let $\mathcal{T} = \{A \sqsubseteq \neg B\}$ and $\mathcal{A} = \{A(a), B(a)\}$, then the satisfiability test is carried out by answering the query $Q_{\mathcal{T}} = \exists x.A(x) \wedge B(x)$ 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 [2]. 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 the signature.

The main technical tool for proving the rewritability results is the chase construction known from database theory. The idea is to "repair" the ABox with respect to the constraints formulated by the positive inclusion axioms $\mathcal{T}_p$. The essential property of the canonical model $can(\mathcal{O})$ resulting from the chasing process 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 posed to $\mathcal{T}_p \cup \mathcal{A}$ can be answered by computing $Q^{can(\mathcal{O})}$.

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, computational feasibility is measured with respect to the size of the ABox alone, thereby fixing all other parameters (TBox, query respectively). 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 w.r.t. DL-Lite ontologies is in the low boolean circuits complexity class AC^0, which, roughly, is the class of problems that can be decided instantly (in constant time) with the help of polynomially many processors.

III. Obstacles for combining DL-Lite and RCC8

The NP-completeness of satisfiability 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 spatial domain can be accessed from within the logic; one has to control the "flow of information" from the spatial domain to the thematic domain of the underlying lightweight logic. For example, reducing the thematical component of the logic $\mathcal{ALC}(RCC8)$ of [8] to DL-Lite is not enough to define a combined logic that allows for FOL rewritability.

As testing the satisfiability of arbitrary 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 as recapitulated in Sect. II-B. Consider, e.g., the simple boolean query $Q = \mathtt{ntpp}(a^*, b^*)$, 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 $(\mathtt{ntpp}, \mathtt{ntpp})$ 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^* to b^*, i.e., one would have to query the database for all $n \in \mathbb{N}$ with queries Q_n of the form $Q_n = \exists x_1^* \ldots \exists x_n^*.\mathtt{ntpp}(a^*, x_1^*) \wedge \cdots \wedge \mathtt{ntpp}(x_n^*, b^*)$, because the database may be of the form $\{\mathtt{ntpp}(a^*, c_1^*), \mathtt{ntpp}(c_1^*, b^*)\}$ or of the form $\{\mathtt{ntpp}(a^*, c_1^*), \mathtt{ntpp}(c_1^*, c_2^*), \mathtt{ntpp}(c_2^*, b^*)\}$ 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 the constraint network contained in $\mathcal{A}$ is a complete and satisfiable constraint network. A special case is a network in which there are no disjunctions but only base relations used for labeling edges. In practice, these networks can be computed from (consistent) quantitative geometric data.

Another obstacle for FOL rewritability with respect to query answering is the expressiveness of the query language. Though conjunctive queries are weaker than FOL queries, they allow for querying unnamed objects and building joins that are not treelike. We will therefore consider a weaker query language (GCQ^+ queries below) that is similar to the language of grounded conjunctive queries.

IV. Combinations of lightweight DLs with RCC8 allowing for FOL rewritability

We consider the following extension of DL-Lite, denoted DL-Lite(RCC8), 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 attribute loc is allowed to be functional. The semantics $U^{\mathcal{I}}$ of role chains $U = R \circ loc$ with respect to an interpretation $\mathcal{I}$ is given by role composition of $R^{\mathcal{I}}$ and $loc^{\mathcal{I}}$. The interpretation $C^{\mathcal{I}}$ of concepts of the form $C = \exists U_1, U_2.\{r_1, \ldots r_k\}$ for $r_i \in \mathcal{B}_{RCC8}$ $(1 \leq i \leq k)$ is given as follows:

$$\begin{aligned} C^{\mathcal{I}} \quad = \quad & \{d \in \Delta^{\mathcal{I}} \mid \text{There are } e_1, e_2 \text{ with} \\ & (d, e_1) \in U_1^{\mathcal{I}} \text{ and } (d, e_2) \in U_2^{\mathcal{I}} \\ & \text{such that } (e_1, e_2) \in r_1^{\mathcal{I}} \text{ or} \ldots \text{ or } (e_1, e_2) \in r_k^{\mathcal{I}}\} \end{aligned}$$

The restriction for concepts of the form $\exists U_1, U_2.r$ in Figure

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

Figure 3. The combined logic DL-Lite(RCC8)

3 assures that we do not get empty concepts from the beginning (without any interesting deduction); 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. The resulting proposition, which states that checking the satisfiability of DL-Lite(RCC8)-ontologies with spatially complete ABoxes is FOL rewritable is a corollary of [2, Theorem 4.14].

Proposition 1: Checking the satisfiability of DL-Lite(RCC8)-ontologies whose ABox is spatially complete is FOL rewritable. (Proofs of the propositions can be found in the accompanying technical report [9].)

Proposition 1 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^+. This query language is explicitly constructed for use with DL-Lite(RCC8) and so provides only means for qualitative spatial queries. But it could be extended to allow also for quantitative spatial queries.

Definition 1: 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.
- $(\exists R_1 \dots R_n.C)(x)$ for role symbols or inverses of role symbols R_i, a DL-Lite(RCC8) concept C without the negation symbol, and a variable or a constant x
- $R(x, y)$ for a role symbol R or an inverse thereof
- $loc(x, y^*)$, where x is a variable or constant and y^* is a variable or constant intended to denote elements of the ω-admissible theory Ax_{RCC8}
- $r(x^*, y^*)$, where $r \in \mathit{Rel}_{RCC8}$ and x^*, y^* are variables or constants intended to denote elements of Ax_{RCC8}

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 $\tilde{\exists}\vec{y}\vec{z}^* = \tilde{\exists}y_1 \dots \tilde{\exists}y_n \tilde{\exists}z_1^* \dots \tilde{\exists}z_m^*$ is a sequence of existential quantifiers that have to be interpreted w.r.t. the active domain semantics.

With respect to this query language it is possible to show that a TBox in the combined logic DL-Lite(RCC8) can indeed be compiled into a UCQ and thus into an SQL query—if one presumes that the ABox is spatially complete.

Proposition 2: Answering GCQ^+ queries with respect to DL-Lite(RCC8)-ontologies whose ABox is spatially complete is FOL rewritable.

This proposition can be proved by extending the proof of Theorem 5.15 in [2]. The main component of our proof is a reformulation algorithm that is an adaption of the algorithm PerfectRef [2, 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.

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. For example, if the TBox contains the positive inclusion axiom $A_1 \sqsubseteq A_2$ (A_1 is a subconcept of A_2), and the UCQ contains the atom $A_2(x)$ in a CQ, then, among the CQ with $A_2(x)$, the rewritten UCQ query contains a CQ in which $A_2(x)$ is substituted by $A_1(x)$. In our adaption of PerfectRef, we integrate GCQ^+ atoms of the form $\exists U_1, U_2.r(x)$ into the overall reformulation process. The relevant 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 concept $\exists R_1 \circ loc, loc.r_1$ and $\exists loc, R_2 \circ loc.r_2$ is a subconcept of $\exists R_1 \circ loc, R_2 \circ loc.r_3$ where $r_3 \in \mathit{Rel}_{RCC8}$ is a superset 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. 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 a CQ, then it can be replaced by a conjunct of $\exists R_1 \circ loc, loc.r_1(x)$ and $\exists loc, R_2 \circ loc.r_2(x)$ in a new CQ for all $r_1, r_2 \in \mathit{Rel}_{RCC8}$ 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 CQ in which $\exists U_1, U_2.r_1(x)$ is substituted by $B(x)$.
- 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 CQ 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 CQ by substituting $\exists R_1 \circ loc, U_1.r(x)$ with $\exists R_2 \circ loc, U_1.r(x)$.

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

$$\begin{aligned} \mathit{Park+Lake} &\sqsubseteq \mathit{Park} \\ \mathit{Park4Playing} &\sqsubseteq \mathit{Park} \\ \mathit{Park+Lake} &\sqsubseteq \exists \mathit{hasLake} \circ loc, loc.\mathsf{tpp} \\ \mathit{Park4Playing} &\sqsubseteq \exists \mathit{hasPlAr} \circ loc, loc.\mathsf{tpp} \end{aligned}$$

The ABox $\mathcal{A}$ is derived virtually by mappings from GIS data in a database; think of mappings for *Park+Lake* and

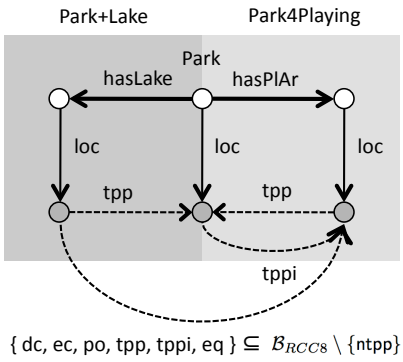

Figure 4. Interpretation satisfying the queries Q, Q', and Q''.

Park4Playing that produce an object a as instance of *Park+Lake*, *Park4Playing*, respectively. That is, assume the following: $\{\textit{Park+Lake}(a), \textit{Park4Playing}(a)\} \subseteq \mathcal{A}$.

The query asking for all parks with lakes and playing areas such that the playing area is not contained as island in the lake can be expressed as the following GCQ^+ (see Figure 4):

$$\begin{aligned} Q \quad = \quad & \textit{Park}(x) \wedge \\ & \exists \textit{hasLake} \circ \textit{loc}, \textit{hasPlAr} \circ \textit{loc}.(\mathcal{B}_{RCC8} \setminus \{\mathsf{ntpp}\})(x) \end{aligned}$$

The reformulation algorithm introduced above produces a UCQ that contains, among Q and others, the following CQ according to the first rewriting rule in the extended reformulation algorithm presented above

$$\begin{aligned} Q' \quad = \quad & (\exists \textit{hasLake} \circ \textit{loc}, \textit{loc}.\mathsf{tpp})(x) \wedge \\ & (\exists \textit{loc}, \textit{hasPlAr} \circ \textit{loc}.\mathsf{tppi})(x) \end{aligned}$$

Due to the RCC composition table the relation between the location of the lake and the location of the playing area referred to in Q' is $\{\mathsf{dc}, \mathsf{ec}, \mathsf{po}, \mathsf{tpp}, \mathsf{tppi}, \mathsf{eq}\}$ (see Section II-A), which implies a relation $\mathcal{B}_{RCC8} \setminus \{\mathsf{ntpp}\}$ between the lake and the playing area as stated in Q. Thus, using the fact that $\exists \textit{loc}, \textit{hasPlAr} \circ \textit{loc}.\mathsf{tppi}$ can be rewritten to $\exists \textit{hasPlAr} \circ \textit{loc}, \textit{loc}.\mathsf{tpp}$ in combination with the subconcept rewriting rule for $A_1 \sqsubseteq A_2$ (see above) we get another CQ

$$Q'' = \textit{Park+Lake}(x) \wedge \textit{Park4Playing}(x)$$

Using the mappings of *Park+Lake* and *Park4Playing* to SQL, the final query to be posed to the database is obtained. This query captures the object a mentioned above such that query answering is complete and all objects with design flaws are found by taking the complement of Park w.r.t. the result set of Q.

V. Conclusion

The query language GCQ^+ allows for the SQL compilation of queries w.r.t. a DL-Lite(RCC8) conceptualization (TBox) for a geographic application domain (Propositions 1 and 2). In order to find all relevant mappings to SQL, the TBox is used to provide an axiomatization of the concepts used in the domain. In order to provide for complete query answering this formalization needs only to be quite weak as shown in the example. DL-Lite(RCC8) is not expressive enough to define sufficient conditions for concepts like that of a park containing a lake in terms of quantitative data. However, as we have argued, given the mappings of concepts (relations) to SQL, only necessary conditions on spatio-thematic concepts need to be formulated in order to automatically construct SQL queries that provide for complete query answering. The process of query rewriting requires reasoning w.r.t. the TBox and the axioms Ax_{RCC8}, and reasoning algorithms are indeed combinatorial w.r.t. query and TBox size (but we have small TBoxes and queries). However, given a compiled query, query answering is tractable in data complexity, and hence feasibility of ontology-based query answering in GIS applications is achieved.

References

[1] R. Grütter, I. Helming, S. Speich, and A. Bernstein, "Rewriting queries for web searches that use local expressions," in *Proceedings of the 5th International Symposium on Rule-Based Reasoning, Programming, and Applications (RuleML-2011 – Europe)*, ser. LNCS, N. Bassiliades, G. Governatori, and A. Paschke, Eds., vol. 6826, 2011, pp. 345–359.

[2] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodríguez-Muro, and R. Rosati, "Ontologies and databases: The DL-Lite approach," in *Semantic Technologies for Informations Systems – 5th Int. Reasoning Web Summer School (RW-2009)*, ser. LNCS, S. Tessaris and E. Franconi, Eds. Springer, 2009, vol. 5689, pp. 255–356.

[3] D. A. Randell, Z. Cui, and A. G. Cohn, "A spatial logic based on regions and connection," in *Proceedings of the 3rd International Conferecence on Knowledge Representation and Reasoning*, 1992, pp. 165–176.

[4] A. K. Mackworth, "Consistency in networks of relations," *Artif. Intell.*, pp. 99–118, 1977.

[5] J. Renz, *Qualitative Spatial Reasoning with Topological Information*, ser. Lecture Notes in Computer Science. Springer, 2002, vol. 2293.

[6] B. Bennett, "Modal logics for qualitative spatial reasoning," *Logic Journal of the IGPL*, vol. 4, no. 1, pp. 23–45, 1996.

[7] J. Renz and B. Nebel, "On the complexity of qualitative spatial reasoning: a maximal tractable fragment of the region connection calculus," *Artif. Intell.*, vol. 108, no. 1-2, pp. 69–123, 1999.

[8] C. Lutz and M. Miličić, "A tableau algorithm for description logics with concrete domains and general TBoxes," *J. Autom. Reasoning*, vol. 38, no. 1-3, pp. 227–259, 2007.

[9] Ö. L. Özçep and R. Möller, "Combining lightweight description logics with the region connection calculus," Institute for Softwaresystems (STS), Hamburg University of Technology, Tech. Rep., 2011, available online at http://www.sts.tu-harburg.de/tech-reports/papers.html.