TECHNISCHE UNIVERSITÄT DRESDEN

Technische Universität Dresden
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS-Report

Practical Query Rewriting for DL-Lite with Numerical Predicates (Extended Version)

Christian Alrabbaa, Patrick Koopmann, Anni-Yasmin Turhan

LTCS-Report 19-06

Postal Address:
Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
01062 Dresden

http://lat.inf.tu-dresden.de

Visiting Address:
Nöthnitzer Str. 46
Dresden

Contents

Abstract

We present a method for answering ontology-mediated queries for DL-Lite extended with a concrete domain, where we allow concrete domain predicates to be used in the query as well. Our method is based on query rewriting, a well-known technique for ontology-based query answering (OBQA), where the knowledge provided by the ontology is compiled into the query so that the rewritten query can be evaluated directly over a database. This technique reduces the problem of query answering w.r.t. an ontology to query evaluation over a database instance. Specifically, we consider members of the DL-Lite family extended with unary and binary concrete domain predicates over the real numbers. While approaches for query rewriting DL-Lite with these concrete domain have been investigated theoretically, these approaches use a combined approach in which also the data is processed, and require the concrete domain values occurring in the data to be known in advance, which makes the procedure data-dependent. In contrast, we show how rewritings can be computed in a data-independent fashion.

Practical Query Rewriting for DL-Lite with Numerical Predicates (Extended Version)

Christian Alrabbaa, Patrick Koopmann, Anni-Yasmin Turhan

July 8, 2019

1 Introduction

Formal ontologies are useful to augment application data in order to be able to extract more consequences from the data by use of the background knowledge than from a query over the plain data alone. In recent years, *ontology-based query answering* (OBQA) by means of Description Logics (DLs) has become a prominent example of this setting. In many practical applications such as medical or stream-reasoning applications, where data is produced by sensors, the data need not always be symbolic, but can be numerical. Concepts from such applications can be characterized by relating their instances to numerical values. For example, patients with high blood pressure can be modelled as patients with a value for blood pressure over 180. Such statements can be expressed in an ontology by the use of concrete domains [3].

In OBQA applications, the expressiveness of the underlying DL can lead to high complexity of query answering, which limits a fast execution of ontology-based queries [**?**, **?**, **?**]. This has lead to the development of the DL-Lite family of DLs that are designed such that their expressiveness admits to perform query answering by the well-known rewriting approach for answering conjunctive queries [4, **?**]. In the classical rewriting approach, the query is rewritten such that the resulting query incorporates the relevant knowledge from the ontology. Then, the rewritten query is answered over the plain (or possibly enriched) data by a database engine directly. DLs that admit this approach are called *first-order rewritable*. The rewriting approach has strong benefits. Since answering conjunctive queries has a complexity of AC^0 measured in the size of the data, rewritability means that query answering is of the same complexity. Furthermore, the rewriting is *data-independent*, and thus only has to be performed once, after which the rewritten query can be executed on different databases without further adaptations or further reasoning steps. This is especially useful for querying big data or frequently changing data. Another approach to solve OBQA by means of standard database query answering is the *combined rewriting approach*, in which the data is enriched based on the ontology before the rewritten query is executed [**?**, **?**, **?**].

Combinations of DL-Lite and concrete domains have been investigated in regard of query answering early on [7, 8, 1]. In these combinations either the concrete domain predicates are only unary [7, 8, 1] or the query language does not admit the use of concrete domain predicates [7]. Both restrictions are severe limitations on the expressiveness.

Recently, Baader et al. identified in [2] a criterion for concrete domains with n-ary predicates that admits *combined rewritability* when used in combination with DL-Lite. This so-called *cr-admissibility* consists of several properties that the concrete domain must fulfill. Among others,

the concrete domain must be convex and admit polynomial reasoning, it must contain equality in its set of predicates, and it must be functional, i.e. for any predicate (of non-zero arity) applied to a tuple of variables, where one of these variables has a fixed value, there is at most one solution. See [2] for a discussion of all the properties required by cr-admissibility.

Two concrete domains that are identified as cr-admissible in [2], are those over the rational numbers with predicates including equality and one comparison $\sim_d \in \{<_d, >_d\}$ to arbitrary values d and a predicate to state a distance $+_d(v, w)$ from one value to another. This concrete domain may seem inexpressive, but it has infinitely many different unary and also binary predicates. The latter gives more expressiveness as the concrete domains considered in earlier approaches. This concrete domain also admits to express the example from above:

$$\mathsf{HighBloodPressurePatient} \sqsubseteq \mathsf{Patient} \sqcap \exists \mathsf{hasBloodPressure}.\, >_{180} .$$

Furthermore it can express that high risk patients are patients whose systolic blood pressure is above their diastolic blood pressure by 90 (mmHg):

$$\mathsf{HighRiskPatient} \sqsubseteq \mathsf{Patient} \sqcap \exists \mathsf{hasDiastolicBloodPressure}, \mathsf{hasSystolicBloodPressure}.+_{90} .$$

Despite being a polynomial method for evaluating queries, the technique proposed in [2] is a combined rewriting approach, which means that the data has to be processed before the rewritten query can be executed. Furthermore, the query rewriting procedure requires full knowledge of the concrete domain values that occur in the data. This limits the practical applicability of the technique for large or frequently changing data sets—one of the main advantages of DLs admitting full first-order rewritability. To solve this issue, we present a *data-independent* rewriting procedure for DL-Lite extended with the aforementioned cr-admissible concrete domains.

Our rewriting procedure proceeds in two phases. The first phase, is to saturate the terminological part of the ontology, and the second is to rewrite the input query based on the saturated ontology. When answering conjunctive queries by a data-independent rewriting approach, the rewritten query must cater for all possibilities how concept membership or satisfaction of a concrete domain predicate can be derived based on the information in the ontology. For instance, if a concept implies a positive distance between a pair of values, and the ontology also states a bound for the first value, then a bound on the second value can be inferred. Such information does only depend on the ontology and not on the data. In the first phase, our rewriting approach makes such information explicit and adds it to the TBox in the form of new axioms. This TBox saturation can be done even independently of the query. In the second phase, our algorithm computes a rewriting of a given query in regard to the saturated TBox. In contrast to the classical rewriting for DL-Lite without concrete domains, here the challenge is that our rewriting procedure needs to cope with a potentially infinite set of predicates.

This paper is structured as follows. In the next section we introduce the two concrete domains $\mathbb{R}_>$ and $\mathbb{R}_<$, the resulting logic $\mathsf{DL\text{-}Lite}_\sqcap(\mathbb{R}_\sim)$ and answering of conjunctive queries. In Section 3, we give an overview of the rewriting method and in Section 4, we describe the TBox saturation and its properties. In Section 5, we introduce the algorithm to compute the query rewriting and we show that it is complete and terminating. Finally, we provide our conclusion in Section 6.

2 Query Answering in $\mathsf{DL\text{-}Lite}_\sqcap(\mathbb{R}_>)$ and $\mathsf{DL\text{-}Lite}_\sqcap(\mathbb{R}_<)$

We recall syntax and semantics of $\mathsf{DL\text{-}Lite}_\sqcap(\mathcal{D})$ and the main task conjunctive query answering.

2.1 The Concrete Domains $\mathbb{R}_>$ and $\mathbb{R}_<$

We first define two concrete domains over the real numbers, $\mathbb{R}_>$ and $\mathbb{R}_<$, that are used in our DL. In general, a concrete domain [3] is a tuple $\mathcal{D} = \langle \Delta^{\mathcal{D}}, \mathcal{P}^{\mathcal{D}}, \mathsf{ar}^{\mathcal{D}}, \cdot^{\mathcal{D}} \rangle$ of a set $\Delta^{\mathcal{D}}$ of *concrete domain elements*, a set of $\mathcal{P}^{\mathcal{D}}$ of *concrete domain predicates*, where to each $\Pi \in \mathcal{P}$ an arity $\mathsf{ar}(\Pi) \in \mathbb{N}$ is associated, and an *interpretation function* $\cdot^{\mathcal{D}}$ which assigns to each $\Pi \in \mathcal{P}$ with $\mathsf{ar}(\Pi) = n$ a set $\Pi^{\mathcal{D}} \subseteq (\Delta^{\mathcal{D}})^n$. We focus on two concrete domains, $\mathbb{R}_>$ and $\mathbb{R}_<$, defined by $\mathbb{R}_\sim = \langle \mathbb{R}, \mathcal{P}^{\mathbb{R}_\sim}, \mathsf{ar}^{\mathbb{R}_\sim}, \cdot^{\mathbb{R}_\sim} \rangle$, for one comparison predicate $\sim \in \{<, >\}$ per concrete domain, with the set $\mathcal{P}^{\mathbb{R}_\sim}$ of predicates defined as $\mathcal{P}^{\mathbb{R}_\sim} = \{\top^1_{\mathbb{R}_\sim}, \top^2_{\mathbb{R}_\sim}, \bot^1_{\mathbb{R}_\sim}, \bot^2_{\mathbb{R}_\sim}\} \cup \{=_d, \sim_d, +_d \mid d \in \mathbb{R}\}$, arities $\mathsf{ar}^{\mathbb{R}_\sim}(=_d) = \mathsf{ar}^{\mathbb{R}_\sim}(\sim_d) = \mathsf{ar}^{\mathbb{R}_\sim}(\top^1_{\mathbb{R}_\sim}) = \mathsf{ar}^{\mathbb{R}_\sim}(\bot^1_{\mathbb{R}_\sim}) = 1$ and $\mathsf{ar}^{\mathbb{R}_\sim}(+_d) = \mathsf{ar}^{\mathbb{R}_\sim}(\top^2_{\mathbb{R}_\sim}) = \mathsf{ar}^{\mathbb{R}_\sim}(\bot^2_{\mathbb{R}_\sim}) = 2$, and an interpretation function defined as

$$(\top^1_{\mathbb{R}_\sim})^{\mathbb{R}_\sim} = \mathbb{R} \qquad (\bot^1_{\mathbb{R}_\sim})^{\mathbb{R}_\sim} = (\bot^2_{\mathbb{R}_\sim})^{\mathbb{R}_\sim} = \emptyset \qquad (\sim_d)^{\mathbb{R}_\sim} = \{d' \mid d' \in \mathbb{R}, d' \sim d\}$$
$$(\top^2_{\mathbb{R}_\sim})^{\mathbb{R}_\sim} = \mathbb{R} \times \mathbb{R} \qquad (=_d)^{\mathbb{R}_\sim} = \{d\} \qquad (+_d)^{\mathbb{R}_\sim} = \{\langle d_1, d_2 \rangle \mid d_1, d_2 \in \mathbb{R}, d_1 + d = d_2\}.$$

Given two predicates Π_a and Π_b of the same arity, we write $\Pi_a \models \Pi_b$ iff $(\Pi_a)^{\mathbb{R}_\sim} \subseteq (\Pi_b)^{\mathbb{R}_\sim}$.

2.2 The Description Logics DL-Lite$_\sqcap(\mathbb{R}_\sim)$

We recall DL-Lite$_\sqcap(\mathbb{R}_\sim)$ with the two concrete domains just introduced. Let $\mathsf{N_C}$, $\mathsf{N_R}$, $\mathsf{N_A}$ and $\mathsf{N_I}$ be pairwise disjoint, countably infinite sets of respectively *concept names*, *role names*, *attribute names* and *individual names*. A *role* R is an expression of the form r or r^-, where $r \in \mathsf{N_R}$. DL-Lite$_\sqcap(\mathcal{D})$ *concepts* C, D and *axioms* $\mathfrak{a}$ are defined according to the following syntax rule, where $A \in \mathsf{N_C}$, R, S are roles, $U_1, \ldots, U_n \in \mathsf{N_A}$, and $\Pi_n \in \mathcal{P}^{\mathbb{R}_\sim}$ s.t. $\mathsf{ar}^{\mathbb{R}_\sim}(\Pi_n) = n$:

$$C ::= \top \mid A \mid C \sqcap C \mid \exists R \mid \exists U_1, \ldots, U_n.\Pi_n \qquad D ::= \bot \mid C \mid \forall U_1, \ldots, U_n.\Pi_n$$
$$\mathfrak{a} ::= C \sqsubseteq D \mid R \sqsubseteq S \ .$$

We assume (nested) conjunctions to be represented as sets, that is, they never contain duplicates, and the order of conjuncts is not important.

A *TBox* is a finite set of axioms. Let $A \in \mathsf{N_C}$, $a, b \in \mathsf{N_I}$, $r \in \mathsf{N_R}$, $U \in \mathsf{N_A}$, and $d \in \mathbb{R}$. An *ABox* is a set of *assertions*, which are of the forms $A(a)$, $r(a, b)$, and $U(a, d)$. A *knowledge base* (KB) is a tuple $\langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ is a TBox and $\mathcal{A}$ an ABox.

Example 1. *Assume that $A, B_1, B_2 \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, $U, U_1, U_2 \in \mathsf{N_A}$, and $b \in \mathsf{N_I}$. Then, $\mathcal{T}$ is a TBox s.t. $\mathcal{T} = \{B_1 \sqsubseteq \exists r^-,\ A \sqsubseteq \exists U.>_{3.5},\ B_2 \sqsubseteq \forall U_1, U_2.+_{10}\}$; $\mathcal{A}$ is an ABox s.t. $\mathcal{A} = \{A(b), B_2(b), U_1(b, 11), U_2(b, 21)\}$; and $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is a knowledge base.*

The semantics of KBs is defined in terms of *interpretations*. An interpretation is a tuple $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}, \mathbb{R}_\sim \rangle$, where $\Delta^{\mathcal{I}}$ is a set called the *domain*, $\cdot^{\mathcal{I}}$ is a function, and $\mathbb{R}_\sim$ is a concrete domain. The function $\cdot^{\mathcal{I}}$ maps every $a \in \mathsf{N_I}$ to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, every concept name A to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, every role name r to a relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and every attribute $U \in \mathsf{N_A}$ to a relation $U^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \mathbb{R}$. We require the domain $\Delta^{\mathcal{I}}$ to be disjoint with the concrete domain: $\Delta^{\mathcal{I}} \cap \mathbb{R} = \emptyset$. The interpretation function is extended to roles by setting $(r^-)^{\mathcal{I}} = (r^{\mathcal{I}})^-$, and to concepts by $(\top)^{\mathcal{I}} = \Delta^{\mathcal{I}}$, $(\bot)^{\mathcal{I}} = \emptyset$,

$$(C_1 \sqcap C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}, \qquad (\exists R)^{\mathcal{I}} = \{e \in \Delta^{\mathcal{I}} \mid \exists e' \in \Delta^{\mathcal{I}} : \langle e, e' \rangle \in R^{\mathcal{I}}\},$$
$$(\exists U_1, \ldots, U_n.\Pi)^{\mathcal{I}} = \{e \in \Delta^{\mathcal{I}} \mid \exists \langle e, d_1 \rangle \in U_1^{\mathcal{I}}, \ldots, \exists \langle e, d_n \rangle \in U_n^{\mathcal{I}} : \langle d_1, \ldots, d_n \rangle \in \Pi^{\mathbb{R}_\sim}\},$$
$$(\forall U_1, \ldots, U_n.\Pi)^{\mathcal{I}} = \{e \in \Delta^{\mathcal{I}} \mid \forall \langle e, d_1 \rangle \in U_1^{\mathcal{I}}, \ldots, \forall \langle e, d_n \rangle \in U_n^{\mathcal{I}} : \langle d_1, \ldots, d_n \rangle \in \Pi^{\mathbb{R}_\sim}\}.$$

Let X, Y be concepts or roles. An interpretation $\mathcal{I}$ *satisfies an axiom* $X \sqsubseteq Y$ (in symbols $\mathcal{I} \models X \sqsubseteq Y$) iff $X^{\mathcal{I}} \subseteq Y^{\mathcal{I}}$. $\mathcal{I}$ *satisfies an assertion* $A(a)$ iff $a^{\mathcal{I}} \in A^{\mathcal{I}}$, $r(a, b)$ iff $\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in r^{\mathcal{I}}$,

and $U(a,d)$ iff $\langle a^{\mathcal{I}}, d\rangle \in U^{\mathcal{I}}$. $\mathcal{I}$ is a *model* of a KB (TBox) if it satisfies all axioms and assertions in it. Two TBoxes $\mathcal{T}, \mathcal{T}'$ are equivalent (in symbols $\mathcal{T} \equiv \mathcal{T}'$) if they have the same set of models. An axiom/assertion $\mathfrak{b}$ is *entailed* by a KB $\mathcal{K}$ (in symbols $\mathcal{K} \models \mathfrak{b}$) if $\mathcal{I} \models \mathfrak{b}$ for all models $\mathcal{I}$ of $\mathcal{K}$.

2.3 Conjunctive Queries for DL-Lite$_{\sqcap}(\mathbb{R}_{\sim})$ KBs

Let $\mathsf{N_V}$ be a countably infinite set of *variables* pairwise disjoint with $\mathsf{N_C}$, $\mathsf{N_R}$, $\mathsf{N_A}$, and $\mathsf{N_I}$. Elements from $\mathsf{N_I} \cup \mathsf{N_V}$ are *abstract terms* and elements from $\mathbb{R} \cup \mathsf{N_V}$ are *concrete terms*. The union of abstract and concrete terms is called *terms*. Let $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, $U \in \mathsf{N_A}$, $\Pi \in \mathcal{P}^{\mathbb{R}_{\sim}}$ with arity n, t_a, t'_a be abstract terms and $t_{c_1}, \ldots, t_{c_n}$ be concrete terms. An *atom* is an expression of the forms $A(t_a)$, $r(t_a, t'_a)$, $=(t_a, t'_a)$, $U(t_a, t_c)$ or $\Pi(t_{c_1}, \ldots, t_{c_n})$. A *conjunctive query* (CQ) is an expression of the form $\phi = \exists x_1, \ldots, x_n.\alpha_1 \wedge \ldots \wedge \alpha_m$, where $x_1, \ldots, x_n \in \mathsf{N_V}$, and $\alpha_1, \ldots, \alpha_n$ are atoms. We denote terms in α ($/\phi$) by $\mathbf{terms}(\alpha)$ ($/\mathbf{terms}(\phi)$), and variables in α ($/\phi$) by $\mathbf{var}(\alpha)$ ($/\mathbf{var}(\phi)$). Variables in ϕ that are not bound by an existential quantifier are called *answer variables*. A *union of CQs* (UCQ) is a non-empty set of CQs, where each CQ has the same set of answer variables. We denote the set of answer variables of a UCQ Ψ by $\mathsf{var}_{\mathrm{ans}}(\Psi)$. A UCQ Ψ is called *Boolean* if $\mathsf{var}_{\mathrm{ans}}(\Psi) = \emptyset$. Given a UCQ Ψ, an *answer to* Ψ is a mapping $\mathbf{a} : \mathsf{var}_{\mathrm{ans}}(\Psi) \to \mathsf{N_I} \cup \mathbb{R}$, and we denote by $\mathbf{a}(\Psi)$ the Boolean UCQ obtained by replacing every answer variable x by $\mathbf{a}(x)$. Answer variables and answers are defined accordingly for CQs.

Given an interpretation $\mathcal{I}$, a Boolean CQ ϕ is *satisfied by* $\mathcal{I}$ (in symbols $\mathcal{I} \models \phi$) if there exists a *homomorphism* $h : \mathbf{terms}(\phi) \to \Delta^{\mathcal{I}} \cup \mathbb{R}$ s.t. for every $=(t_a, t'_a) \in \phi$ we have $h(t_a) = h(t'_a)$, for every $d \in \mathbf{terms}(\phi) \cap \mathbb{R}$ we have $h(d) = d$, for every $a \in \mathbf{terms}(\phi) \cap \mathsf{N_I}$ we have $h(a) = a^{\mathcal{I}}$, for every $A(t) \in \phi$ we have $h(t) \in A^{\mathcal{I}}$, for every $r(t_1, t_2) \in \phi$ we have $\langle h(t_1), h(t_2)\rangle \in r^{\mathcal{I}}$, for every $U(t_1, t_2) \in \phi$ we have $\langle h(t_1), h(t_2)\rangle \in U^{\mathcal{I}}$, and for every $\Pi(t_1, \ldots, t_n)$, we have $\langle h(t_1), \ldots, h(t_n)\rangle \in \Pi^{\mathbb{R}_{\sim}}$. We might then also write $\mathcal{I} \models h(\phi)$. A Boolean UCQ Φ is satisfied by $\mathcal{I}$ iff $\mathcal{I} \models \phi$ for some $\phi \in \Phi$. A Boolean UCQ is entailed by a KB $\mathcal{K}$ if it is satisfied in every model of $\mathcal{K}$. A Boolean CQ/UCQ Φ is unsatisfiable in a TBox $\mathcal{T}$ if for every model $\mathcal{I}$ of $\mathcal{T}$, we have $\mathcal{I} \not\models \Phi$. An answer $\mathbf{a}$ to a UCQ Ψ is a *certain answer to* Ψ if $\mathcal{K} \models \mathbf{a}(\Psi)$.

Example 2. *Let $\mathcal{K}$ be a KB as shown in Example 1. Let ϕ_1 and ϕ_2 be CQs s.t. $\phi_1 = \exists x, v.U(x, v)$, and $\phi_2 = \exists v_1, v_2.(U_1(x, v_1) \wedge U_2(x, v_2) \wedge +_{10}(v_1, v_2))$. We have $\mathsf{var}_{ans}(\phi_1) = \emptyset$, and $\mathsf{var}_{ans}(\phi_2) = \{x\}$. Furthermore, ϕ_1 is Boolean and $\mathcal{K} \models \phi_1$. The answer $\mathbf{a}$ for ϕ_2 with $\mathbf{a}(x) = b$ is a certain answer to ϕ_2 in $\mathcal{K}$, since $\mathcal{K} \models \mathbf{a}(\phi_2)$.*

3 Overview of the Query Rewriting Method

Our aim is to develop a practical rewriting method for answering UCQs with concrete domain predicates over $\mathbb{R}_{\sim}$ w.r.t. DL-Lite$_{\sqcap}(\mathbb{R}_{\sim})$ ontologies. While in principle, a single rewriting step can be sufficient to enable query answering, it does not yield a practical algorithm that lends itself to implementation. This is mainly due to consequences that follow from the concrete domain alone, or from their combination with the TBox. For example, axioms of the form $C \sqsubseteq \forall U_1, U_2.+_d$ make both attributes U_1 and U_2 functional for instances of concept C. Consequences of this sort hold independently of the query and the data, and thus would need to be re-discovered for each answered UCQ. A more efficient way is to compute these consequences once and reuse them. To this end, we present an approach that consists of two steps. The first one is a preprocessing step of the TBox $\mathcal{T}$, which is called *TBox saturation*, and the second is the rewriting of the query w.r.t. the saturated version of $\mathcal{T}$.

In the TBox saturation step, a set of *saturation rules* augments the TBox with additional axioms. The rewriting step is then similar to the classical one for DL-Lite [4], and employs a set of *rewriting rules*. Starting from some CQ ϕ in the input UCQ Φ, and for every rewriting rule

$\mathbf{R}$, if ϕ satisfies the premise of $\mathbf{R}$, and the side condition of $\mathbf{R}$ is also met, then the conclusion of $\mathbf{R}$, as a new CQ ϕ, is added to Φ. This process is repeated for every CQ in Φ until a fixed-point is reached. As we present later on, the side conditions of the rewriting rules check for the existence of certain axioms that follow from $\mathcal{T}$, but are not necessarily present in $\mathcal{T}$. But since the query rewriting step is based on the saturated version of $\mathcal{T}$, a syntactic check suffices to inspect the satisfaction of these side conditions. This method yields a sound and complete procedure for query rewriting for $\mathsf{DL\text{-}Lite}_{\sqcap}(\mathbb{R}_{\sim})$.

4 TBox Saturation

We introduce the calculus for generating the saturated version of a given TBox $\mathcal{T}$, which is then used by the query rewriting procedure. Afterwards, we discuss properties of this calculus.

4.1 TBox Saturation Calculus

Our calculus consists of the rules shown in Figure 1. In these rules, $\Pi_1, \Pi_2 \in \mathcal{P}_{\mathbb{R}_\sim}$ are some predicates from $\mathbb{R}_\sim$, $\varpi \in \{<, =, >\}$ is a comparison operator from $\mathcal{P}^{\mathbb{R}_\sim}$, that together with a value d gives rise to the unary predicate ϖ_d, and $\mathsf{Q} \in \{\forall, \exists\}$ is a quantor. The preconditions are to be checked syntactically and the derived statements are added as axioms to the TBox.

The rules in the calculus are grouped according to the kind of inference they yield. The rules in $\mathbf{R_{init}}$ infer the straightforward properties of attributes. For example, rule $\mathbf{R_{init}}$-6 states that if an element has two attribute values with distance d, and both of these are (locally) functional, then all pairs of values of these attributes must have a distance d. The rules in $\mathbf{R}_+$ infer implicit distances between attribute values, since the binary predicate $+_d$ behaves additive for the real numbers, and distances can simply be propagated down (/up) the number line. Rules in $\mathbf{R}_\varpi$ lead to the inference of an attribute value based on the following: if the distance between two attribute successors and the value of one of them are known, then the value of the other attribute successor can be inferred. Lastly, the rules in $\mathbf{R}_\bot$ lead to axioms stating which concepts cannot have certain attribute successors or which concepts are unsatisfiable. Observe that the saturation rules can refer to (the presence of) data while staying data-independent. This is achieved by the use of $\exists U.\top_{\mathbb{R}_\sim}$ in the left-hand side of the inferred statements.

4.2 Properties of the Saturation Calculus

In Algorithm 1, the saturation rules are used to infer all axioms from the TBox which are relevant to our rewriting procedure. The relevance of these axioms is determined mainly by two criteria, namely *rewritability* and *termination*. To illustrate relevance of axioms for rewritablity, let us take the following case as an example. Assume we are given the TBox $\mathcal{T} = \{C_2 \sqsubseteq \exists U_1.=_3,\ \top \sqsubseteq \exists U_2.\top_{\mathbb{R}_\sim},\ C_1 \sqsubseteq \forall U_1, U_2.+_2\}$, the ABox $\mathcal{A} = \{C_1(b), C_2(b)\}$, and the CQ $\phi = \exists v.(U_2(x, v) \wedge =_v(5))$. It is easy to see that $\langle \mathcal{T}, \mathcal{A}\rangle \models \phi$, but $\langle \emptyset, \mathcal{A}\rangle \not\models \phi$. In order to be able to rewrite ϕ into Φ s.t. $\langle \emptyset, \mathcal{A}\rangle \models \Phi$, we need some axiom $\mathfrak{a}$ in $\mathcal{T}$ of the form $C_1 \sqcap C_2 \sqsubseteq \exists U_2.=_5$. Thus, $\mathfrak{a}$ is a *relevant* axiom from the *rewritability* aspect, and therefore, a rule to generate such an axiom, from such a TBox, is needed. In this example, this rule is $\mathbf{R}_\varpi$-1. Let us consider another case where $\phi' = \exists v.(U_2(x, v))$. Then, $\mathfrak{a}_1 = (C_1 \sqcap C_2 \sqsubseteq \exists U_2.\top_{\mathbb{R}_\sim})$ is needed to get the rewriting of ϕ, but actually $\mathfrak{a}_1$ is not a relevant axiom because its effect is already covered, since $(C_1 \sqcap C_2 \sqsubseteq \exists U_2.=_5) \models (C_1 \sqcap C_2 \sqsubseteq \exists U_2.\top_{\mathbb{R}_\sim})$. This type of entailments are handled by the rewriting rules from Section 5.

The other criterion for relevance is *termination*. The interaction between the rules, or even

$\mathbf{R_{init}}$-rules:

$$\frac{C \sqsubseteq \mathsf{Q}U_1, U_2.\Pi}{C \sqsubseteq \mathsf{Q}U_1.\top_{\mathbb{R}_\sim},\ C \sqsubseteq \mathsf{Q}U_2.\top_{\mathbb{R}_\sim}} \quad (1) \qquad \frac{C_1 \sqsubseteq \exists U.\Pi \quad C_2 \sqsubseteq \forall U, U.+_0}{C_1 \sqcap C_2 \sqsubseteq \forall U.\Pi} \quad (2)$$

$$\frac{C_1 \sqsubseteq \forall U.\Pi_1 \quad C_2 \sqsubseteq \exists U.\Pi_2}{C_1 \sqcap C_2 \sqsubseteq \exists U.\Pi_1} \quad (3) \qquad \frac{C_1 \sqsubseteq \forall U_1, U_2.+_d}{\exists U_1.\top_{\mathbb{R}_\sim} \sqcap \exists U_2.\top_{\mathbb{R}_\sim} \sqcap C_1 \sqsubseteq \exists U_1, U_2.+_d} \quad (4)$$

$$\frac{C \sqsubseteq \mathsf{Q}U_1, U_2.+_d}{C \sqsubseteq \mathsf{Q}U_2, U_1.+_{-d}} \quad (5) \qquad \frac{C_1 \sqsubseteq \exists U_1, U_2.+_d \quad C_2 \sqsubseteq \forall U_1, U_1.+_0 \quad C_3 \sqsubseteq \forall U_2, U_2.+_0}{C_1 \sqcap C_2 \sqcap C_3 \sqsubseteq \forall U_1, U_2.+_d} \quad (6)$$

$\mathbf{R}_+$-rules:

$$\frac{C_1 \sqsubseteq \exists U_1, U_2.+_{d_1} \qquad C_2 \sqsubseteq \exists U_2, U_3.+_{d_2} \qquad C_3 \sqsubseteq \forall U_2, U_2.+_0}{C_1 \sqcap C_2 \sqcap C_3 \sqsubseteq \exists U_1, U_3.+_{(d_1+d_2)}} \quad (1)$$

$$\frac{C_1 \sqsubseteq \forall U_1, U_2.+_{d_1} \quad C_2 \sqsubseteq \forall U_2, U_3.+_{d_2}}{\exists U_2.\top_{\mathbb{R}_\sim} \sqcap C_1 \sqcap C_2 \sqsubseteq \forall U_1, U_3.+_{(d_1+d_2)}} \quad (2) \qquad \frac{C_1 \sqsubseteq \forall U_1, U_2.+_{d_1} \quad C_2 \sqsubseteq \exists U_2, U_3.+_{d_2}}{\exists U_1.\top_{\mathbb{R}_\sim} \sqcap C_1 \sqcap C_2 \sqsubseteq \exists U_1, U_3.+_{(d_1+d_2)}} \quad (3)$$

$\mathbf{R}_\varpi$-rules:

$$\frac{C_1 \sqsubseteq \forall U_1, U_2.+_{d_1} \qquad C_2 \sqsubseteq \exists U_1.\varpi_{d_2} \qquad C_3 \sqsubseteq \exists U_2.\Pi}{C_1 \sqcap C_2 \sqcap C_3 \sqsubseteq \exists U_2.\varpi_{(d_1+d_2)}} \quad (1)$$

$$\frac{C_1 \sqsubseteq \forall U_1, U_2.+_{d_1} \qquad C_2 \sqsubseteq \exists U_1.\varpi_{d_2}}{C_1 \sqcap C_2 \sqsubseteq \forall U_2.\varpi_{(d_1+d_2)}} \quad (2) \qquad \frac{C_1 \sqsubseteq \exists U_1, U_2.+_{d_1} \qquad C_2 \sqsubseteq \forall U_1.\varpi_{d_2}}{C_1 \sqcap C_2 \sqsubseteq \exists U_2.\varpi_{(d_1+d_2)}} \quad (3)$$

$\mathbf{R}_\perp$-rules:

$$\frac{C_1 \sqsubseteq D_1 \qquad C_2 \sqsubseteq D_2}{C_1 \sqcap C_2 \sqsubseteq \bot} \quad \text{provided } \models D_1 \sqcap D_2 \sqsubseteq \bot \quad (1)$$

$$\frac{C_1 \sqsubseteq D_1 \qquad C_2 \sqsubseteq D_2}{C_1 \sqcap C_2 \sqsubseteq \forall U.\bot_{\mathbb{R}_\sim} \ /\ \forall U_1, U_2.\bot_{\mathbb{R}_\sim}} \quad \text{provided } \models D_1 \sqcap D_2 \sqsubseteq \forall U.\bot_{\mathbb{R}_\sim} \ /\ \forall U_1, U_2.\bot_{\mathbb{R}_\sim} \quad (2)$$

$$\frac{C_1 \sqcap \exists U.\top_{\mathbb{R}_\sim} \sqsubseteq D}{C_1 \sqcap C_2 \sqsubseteq D} \quad \text{provided } C_2 \sqsubseteq \exists U.\Pi \quad (3)$$

Figure 1: TBox saturation rules.

between the conclusion and the premise of the same rule, enables infinitely many rule applications. Axioms inferred using the $\mathbf{R}_\bot$ can be utilised to prevent such behaviour: note that an axiom of the form $C \sqsubseteq \forall U.\bot_{\mathbb{R}_\sim}$ makes all other axioms of the form $C \sqsubseteq \forall U.\Pi$ superfluous, which can be used to limit the number of inferred axioms to be kept.

Let $\mathcal{T}$ be a TBox, $\mathbf{R}$ a saturation rule from Figure 1, and $\mathfrak{a}$ an axiom. The axiom $\mathfrak{a}$ is *derivable by* $\mathbf{R}$ *from* $\mathcal{T}$ (in symbols $\mathcal{T} \vdash_{\mathbf{R}} \mathfrak{a}$) iff the premise(s) of $\mathbf{R}$ occur in $\mathcal{T}$ and $\mathfrak{a}$ is of the form of the conclusion of $\mathbf{R}$. $\mathcal{T} \vdash \mathfrak{a}$ denotes that $\mathfrak{a}$ *is derivable from* $\mathcal{T}$ using any saturation rule in the calculus.

Lemma 4.1 (Soundness)**.** *Let $\mathcal{T}$ be a TBox and $\mathfrak{a}$ be an axiom. Then, $\mathcal{T} \vdash \mathfrak{a}$ only if $\mathcal{T} \models \mathfrak{a}$.*

Proof. The soundness of Rules $\mathbf{R}_{\mathbf{init}}$-1, $\mathbf{R}_{\mathbf{init}}$-3, $\mathbf{R}_{\mathbf{init}}$-4, $\mathbf{R}_{\mathbf{init}}$-5 follows directly from the semantics of $\mathsf{DL\text{-}Lite}_\sqcap(\mathbb{R}_\sim)$. Regarding $\mathbf{R}_{\mathbf{init}}$-2 and $\mathbf{R}_{\mathbf{init}}$-6, we note that the concept $\forall U, U.+_0$ expresses local partial functionality of U: therefore, if some U-successor satisfies a predicate Π, then every U-successor does (there is only one). Similary, if two attributes U_1 and U_2 are partially functional, then $\exists U_1, U_2.+_d$ implies $\forall U_1, U_2.+_d$. The same fact is used in $\mathbf{R}_+$-1: if a domain element satisfies $\exists U_1, U_2.+_{d_1}$ and $\exists U_2, U_3.+_{d_2}$, and it has maximally one U_2-successor, the difference between the U_1 and the U_3 successor must be $d_1 + d_2$. Note that the rule would not be sound without requiring $C_3 \sqsubseteq \exists \forall U_2, U_2.+_0$, since then there could be several U_2-successors. The other $\mathbf{R}_+$-rules propagate information from universal quantifications on the predicate $+_d$. Note that the concepts $\forall U_1, U_2.+_{d_1}$ and $\forall U_2, U_3.+_{d_2}$ do not directly entail $\forall U_1, U_3.+_{d_1+d_2}$, since it is possible that there is no U_2-successor, in which case $\forall U_1, U_2.+_{d_1}$ and $\forall U_2, U_3.+_{d_2}$ are trivially satisfied, regardless of any U_1 or U_3-successor. That is why the conclusion of Rule $\mathbf{R}_+$-2 needs to have the concept $\exists U_2.\top_{\mathbb{R}_\sim}$ on the left hand side. The argument for $\mathbf{R}_+$-3 is similar. The soundness of the $\mathbf{R}_\bot$-rules follows directly from the semantics of $\mathsf{DL\text{-}Lite}_\sqcap(\mathbb{R}_\sim)$ (recall that ϖ_d stands for one of $=_d$, $>_d$ and $<_d$). □

In order to be able to ensure termination of the TBox saturation process, we need to refer to certain kinds of "superfluous axioms".

Definition 4.1 (Redundant axiom)**.** Let $\mathcal{T}$ be a TBox and $\mathfrak{a}_1, \mathfrak{a}_2 \in \mathcal{T}$. Then, $\mathfrak{a}_2$ *is redundant to* $\mathfrak{a}_1$ w.r.t. $\mathcal{T}$ if at least one of the following conditions is satisfied:

1. $\mathfrak{a}_1$ is of the form $C \sqsubseteq D$, and $\mathfrak{a}_2$ is of the form $C \sqcap C' \sqsubseteq D$;
2. $\mathfrak{a}_1$ is of the form $C \sqsubseteq \forall U.\bot_{\mathbb{R}_\sim}$, and $\mathfrak{a}_2$ is of the form $C \sqcap C' \sqsubseteq \forall U.\Pi$; or
3. $\mathfrak{a}_1$ is of the form $C \sqsubseteq \forall U_1, U_2.\bot_{\mathbb{R}_\sim}$, and $\mathfrak{a}_2$ is of the form $C \sqcap C' \sqsubseteq \forall U_1, U_2.+_d$.

The axiom $\mathfrak{a}_2 \in \mathcal{T}$ is a *redundant axiom in* $\mathcal{T}$ iff there exists some $\mathfrak{a}_1 \in \mathcal{T}$ such that $\mathfrak{a}_2$ *is redundant to* $\mathfrak{a}_1$ w.r.t. $\mathcal{T}$.

From this definition, it follows that for any TBox $\mathcal{T}$, we have $\mathcal{T} \equiv \mathcal{T} \setminus \{\mathfrak{a}_2\}$ where $\mathfrak{a}_1, \mathfrak{a}_2 \in \mathcal{T}$, and $\mathfrak{a}_1$ makes $\mathfrak{a}_2$ redundant in $\mathcal{T}$. Therefore, a refinement function of $\mathcal{T}$ can be defined as $\mathsf{refine}(\mathcal{T}) = \{\mathfrak{a} \in \mathcal{T} \mid \mathfrak{a} \text{ is not redundant in } \mathcal{T} \setminus \{\mathfrak{a}\}\}$. Algorithm 1 specifies the computation of the saturated TBox (in symbols: $saturate(\mathcal{T})$). In Lemma 4.1, we prove that for any TBox $\mathcal{T}$, and due to redundancy elimination, Algorithm 1 terminates on any input.

Theorem 4.1 (Termination of saturation)**.** *Algorithm* 1 *always terminates.*

Proof. We prove the lemma by a sequence of claims. In the following, let $\mathcal{T}$ be fixed.

Claim 1. The number of concepts occurring on the left hand side of any axiom added by the algorithm is bounded.

Proof of claim. Let $\mathbf{C}$ be the set of concepts that appear on the left hand side of an axiom in $\mathcal{T}$, as well as of all concepts of the form $\exists U.\top_{\mathbb{R}_\sim}$, where U is an attribute name occurring in $\mathcal{T}$. Since $\mathcal{T}$ is finite, so is $\mathbf{C}$. By inspection of the rules, we see that every left hand side of a derived axiom is a conjunction of concepts from $\mathbf{C}$. Recall that we represent conjunctions as sets, that is, no conjunct can appear twice in a conjunction. As a result, we obtain that there are at most $2^{|\mathbf{C}|}$ different concepts that can occur on the left hand side of an axiom added by the algorithm. ■

Claim 2. For a fixed concept C and fixed attribute names U, U_1, $U_2 \in \mathsf{N_A}$, the number of axioms of the forms $C \sqsubseteq \forall U.{=_d}$ and $C \sqsubseteq \forall U_1, U_2.\Pi$ that are added by the algorithm is bounded.

Proof of claim. First note that all binary predicates in our concrete domain are of the form $+_d$ for some $d \in \mathbb{R}$, and therefore only predicates of the form $=_d$ and $+_d$ are relevant for this claim. Suppose that at some point during the computation of $\mathit{saturate}(\mathcal{T})$, the current TBox $\mathcal{T}'$ contains two axioms $\mathfrak{a}_1, \mathfrak{a}_2$ such that $\mathfrak{a}_1 = C \sqsubseteq \forall U.{=_{d_1}}$ and $\mathfrak{a}_2 = C \sqsubseteq \forall U.{=_{d_2}}$, where $d_1 \neq d_2$. Because we always apply the $\mathbf{R}_\bot$-rules before any other rule, and $\mathbf{R}_\bot$-2 applies on $\mathfrak{a}_1$ and $\mathfrak{a}_2$, then $\mathfrak{a}_3 = C \sqsubseteq \forall U.\bot$ is added in the next iteration. By Definition 4.1, $\mathfrak{a}_3$ makes any further axioms of the form $C \sqsubseteq \forall U.{=_{d'}}$ redundant, so that they are not added by the algorithm anymore. We can apply the same argument for axioms of the form $C \sqsubseteq \forall U_1, U_2.{+_d}$. ■

Claim 3. The calculus generates only finitely many axioms of the form $C \sqsubseteq \exists U_1, U_2.{+_d}$.

Proof of claim. Axioms of this shape are only generated by $\mathbf{R}_+$-1 and $\mathbf{R}_+$-3, as well as $\mathbf{R}_{\mathbf{init}}$-4 and $\mathbf{R}_{\mathbf{init}}$-5. $\mathbf{R}_{\mathbf{init}}$-4 only generates one per axiom of form $C \sqsubseteq \forall U_1, U_2.{+_d}$, of which we already established there can be only finitely many. It therefore suffices to focus on $\mathbf{R}_+$-1, $\mathbf{R}_+$-3 and $\mathbf{R}_{\mathbf{init}}$-5. Note that these rules take an axiom of our shape also as premise.

Assume for a proof by contradiction that the calculus infers infinitely many axioms of this form. It follows that there must be a sequence $\mathfrak{a}_1, \ldots, \mathfrak{a}_n$ of inferred axioms, where for $i \in \{2, \ldots, n\}$, $\mathfrak{a}_i$ was inferred using $\mathfrak{a}_{i-1}$, $\mathfrak{a}_1 = C \sqsubseteq \exists U_1, U_2.{+_{d_1}}$, $\mathfrak{a}_n = C \sqsubseteq \exists U_1, U_2.{+_{d_n}}$, $d_1 \neq d_n$, and each axiom has C on the left-hand side. (Since the number of left-hand sides is bounded, it would eventually converge to C, which is why we can assume the left-hand side to be the same.) These inferences are only possible using one of the rules $\mathbf{R}_+$-1, $\mathbf{R}_+$-3 and $\mathbf{R}_{\mathbf{init}}$-5, . While $\mathbf{R}_{\mathbf{init}}$-5 only switches the order of the attribute names, $\mathbf{R}_+$-1 and $\mathbf{R}_+$-3 exchange one of the attribute names. We can thus reorder the inference so that it first step-wise changes the first attribute name U_1 into U_2 or back into U_1, and then proceeds to change the second attribute name, until

Algorithm 1: Computation of $\mathit{saturate}(\mathcal{T})$

Input: TBox $\mathcal{T}$.
while $\mathcal{T} \vdash \mathfrak{a}$ *for some* $\mathfrak{a}$ *and* $\mathfrak{a}$ *not redundant in* $\mathcal{T}$ **do**
  while $\mathcal{T} \vdash_{\mathbf{R}_\bot} \mathfrak{a}'$ *for some* $\mathfrak{a}'$ *and* $\mathfrak{a}'$ *not redundant in* $\mathcal{T}$ **do**
    set $\mathcal{T} := \mathsf{refine}(\mathcal{T} \cup \{\mathfrak{a}'\})$.
  set $\mathcal{T} := \mathsf{refine}(\mathcal{T} \cup \{\mathfrak{a}\})$.
return $\mathcal{T}$;

we get the required axiom $C \sqsubseteq \exists U_1, U_2.+_{d_n}$. Thus, $\mathit{saturate}(\mathcal{T})$ contains the following axioms:

$$\begin{aligned}\mathfrak{a}_1 &= C \sqsubseteq \exists U_1', U_2.+_{d_1}\\ \mathfrak{a}_2 &= C \sqsubseteq \exists U_2', U_2.+_{d_2}\\ &\vdots\\ \mathfrak{a}_m &= C \sqsubseteq \exists U_m', U_2.+_{d_m},\end{aligned}$$

where $U_1' = U_1$, $U_m' \in \{U_1, U_2\}$ and for $i \in \{2, m\}$, $\mathfrak{a}_i$ is obtained from $\mathfrak{a}_{i-1}$ using one of $\mathbf{R}_+$-1 and $\mathbf{R}_+$-3, and possibly inferences using $\mathbf{R}_{\mathbf{init}}$-5 to get the order of the attribute names right. Note that if $U_m' = U_1$, then $m = n$. Otherwise, $U_m' = U_2$ and the sequence of inferences continues and we have axioms

$$\begin{aligned}\mathfrak{a}_{m+1} &= C \sqsubseteq \exists U_2, U_{m+1}'.+_{d_{m+1}}\\ &\vdots\\ \mathfrak{a}_m &= C \sqsubseteq \exists U_2, U_n'.+_{d_n'},\end{aligned}$$

where $U_n' = U_1$ and $d_n' = -d_n$.

We first show that for every $i \in \{1, \ldots, m-1\}$, there exists $\mathfrak{b}_i = C_i' \sqsubseteq \forall U_i', U_i'.+_0$, where $C_i' \subseteq C$. If $\mathfrak{a}_{i+1}$ was inferred using $\mathbf{R}_+$-1, this is $\mathfrak{b}_i$ was one of the premises. Otherwise, $\mathfrak{a}_{i+1}$ was inferred using $\mathbf{R}_+$-3, then we have as other premise $\mathfrak{b}_i' = C_i'' \forall U_{i+1}', U_i'.+_{d_{i+1}-d_i}$, and $\exists U_{i+1}.\top_{\mathbb{R}_\sim} \in C$. We can thus use $\mathbf{R}_{\mathbf{init}}$-5 and $\mathbf{R}_+$-2 on $\mathfrak{b}_i'$ to derive the required axiom of the form $\mathfrak{b}_i = C_i' \sqsubseteq \forall U_i', U_i'.+_d$.

Using this, we can now show that for every $i \in \{1, m-1\}$, we have an axiom of the form $\mathfrak{c}_i = D_i \sqsubseteq \forall U_i', U_{i+1}'.+_{d_{i+1}-d_i}$. If $\mathfrak{a}_{i+1}$ was inferred using $\mathbf{R}_+ - 3$, this is directly the other premise. Otherwise, the other premise is $D_i' \sqsubseteq \exists U_i', U_{i+1}'.+_{d_{i+1}-d_i}$. First, we already showed, we have the axiom $\mathfrak{b}_i = C_i' \sqsubseteq \forall U_i', U_i'.+_d$, and if $i < m-1$, the axiom $\mathfrak{b}_{i+1} = C_{i+1}' \sqsubseteq \forall U_{i+1}', U_{i+1}'.+_0$. If $i = m-1$, we distinguish the cases based on whether $U_m' = U_1$ or $U_m' = U_1$, and show that in each case, there also exists such an axiom. If $U_m' = U_1$, this axiom is $\mathfrak{b}_1 = C_1' \sqsubseteq \forall U_1, U_1.+_d$. If $U_m' = U_2$, recall that then the sequence continues with $\mathfrak{a}_{m+1} = C \sqsubseteq \exists U_2, U_{m+1}'.+_{d_{m+1}}$, and we can argue as above that there exists the axiom $\beta_{m+1} = C \sqsubseteq \forall U_2, U_2.+_0$. We obtain that, if $\mathfrak{a}_{i+1}$ was inferred using $\mathbf{R}_+$-1, we have the following three axioms:

$$\begin{aligned}D_i' &\sqsubseteq \exists U_i', U_{i+1}'.d_{i+1} - d_i\\ C_i' &\sqsubseteq \forall U_i', U_i'.+_0\\ C_{i+1}' &\sqsubseteq \forall U_{i+1}', U_{i+1}'.+_0,\end{aligned}$$

on which we can apply $\mathbf{R}_{\mathbf{init}}$-6 to infer $\mathfrak{c}_i = D_i \sqsubseteq \forall U_i', U_{i+1}'.d_{i+1} - d$, the same as if $\mathfrak{a}_i$ was inferred using $\mathbf{R}_+$-3.

If we now step-wise combine all $\mathfrak{c}_i$ for $i \in \{0, \ldots, m-1\}$ using $\mathbf{R}_+$-2, we obtain the axiom $\mathfrak{c} = D \sqsubseteq \forall U_m', U_2.+_{d_m}$, where $D \subseteq C$. If $U_m' = U_1$, $m = n$, and by initial assumption, $d_m \neq d_n$. We can thus apply $\mathbf{R}_\bot$-1 on $\mathfrak{c}$ and $\mathfrak{a}_1 = C \sqsubseteq \exists U_1, U_2.+_{d_1}$ and obtain $C \sqsubseteq \bot$, which makes all remaining axioms redundant. Otherwise, $U_m' = U_2$, and we can apply the same argument on the remaining sequence to obtain $D \sqsubseteq \forall U_1, U_2.+_{d_n}$, and again we infer $C \sqsubseteq \bot$. In all cases, we contradict the initial assumption that $\mathit{saturate}(\mathcal{T})$ contains two axiom $C \sqsubseteq \exists U_1, U_2.+_{d_1}$ and $C \sqsubseteq \exists U_1, U_2.+_{d_n}$ where $d_1 \neq d_n$.

As a consequence, there cannot be an unbounded number of axioms of the form $C \sqsubseteq \exists U_1, U_2.+_d$ in $\mathit{saturate}(\mathcal{T})$. ■

Claim 4. The calculus generates only finitely many axioms of the form $C \sqsubseteq \mathsf{Q}U.\varpi_d$.

Proof of claim. The only rules that generate axioms of the form $C \sqsubseteq \mathsf{Q}U.\Pi$ are $\mathbf{R_{init}}$-1, $\mathbf{R_{init}}$-2, $\mathbf{R_{init}}$-3, $\mathbf{R}_{\varpi}$-1, $\mathbf{R}_{\varpi}$-2 and $\mathbf{R}_{\varpi}$-3. The $\mathbf{R_{init}}$-rules do not introduce any new predicates, so the only interested inferences are by the $\mathbf{R}_{\varpi}$-rules. Assume these rules generate an unbounded number of axioms of the form $C \sqsubseteq \mathsf{Q}U_1.\varpi_d$. Since the number of attribute names, as well as the number of concepts on the left hand side of any derived axiom, is finite, it follows that for a fixed C and a fixed U_1, we infer more than two axioms of the form $C \sqsubseteq \mathsf{Q}U_1.\varpi_d \in \mathit{saturate}(\mathcal{T})$ using the $\mathbf{R}_{\varpi}$-rules.

Each of the $\mathbf{R}_{\varpi}$ has a premise of the form $C \sqsubseteq \mathsf{Q}U_1, U_2.+_d$, and we already showed that we only infer a bounded number of those axioms. These inferences would be witnessed by the following sequence of pairs of axioms in $\mathit{saturate}(\mathcal{T})$, each serving as the premises for the next inference,

$$\begin{aligned}
\mathfrak{a}_1 = C \sqsubseteq \mathsf{Q}_1 U_1.\varpi_{d_1} \qquad & \mathfrak{a}'_1 = C_1 \sqsubseteq \mathsf{Q}'_1 U_1, U_2.+_{d'_1} \\
\mathfrak{a}_2 = C \sqsubseteq \mathsf{Q}_2 U_2.\varpi_{d_2} \qquad & \mathfrak{a}'_2 = C_1 \sqsubseteq \mathsf{Q}'_1 U_2, U_3.+_{d'_2} \\
& \vdots \\
\mathfrak{a}_n = C \sqsubseteq \mathsf{Q}_n U_n.\varpi_{d_n} \qquad & \mathfrak{a}'_n = C_n \sqsubseteq \mathsf{Q}'_n U_n, U_1.+_{d'_n},
\end{aligned}$$

together with the final axiom

$$C \sqsubseteq \mathsf{Q}_1 U_1.\varpi_d,$$

where $\sum_{1 \leq i \leq n} d'_i = d - d_1$ and $d \neq d_1$. As the sequence of inferences can then be continued indefinitely, we assume in the following that all indices are for $i \in \{1, \ldots, n\}$ are used modulo n, that is, an index of 0 corresponds to n, and an index of $n+1$ corresponds to 1.

We first show that we can assume without loss of generality that for all $i \in \{1, \ldots, n\}$, $\mathsf{Q}_i = \forall$.

Fix i so that $\mathsf{Q}_i = \exists$. Note that the only $\mathbf{R}_{\varpi}$-rule that takes an axiom of the form $\mathfrak{a}'_i = C_i \sqsubseteq \exists U_i, U_{i+1}.+_{d'_i}$ as a premise is $\mathbf{R}_{\varpi}$-3, in which case we must have $\mathsf{Q}_i = \forall$ and $\mathsf{Q}_{i+1} = \exists$ (where we set $i+1 = 1$ in case $i = n$). The only rule that produces an axiom of the form $C \sqsubseteq \forall U_i.\varpi_{d'_i}$ is $\mathbf{R}_{\varpi}$-2, which means that there exists an axiom $\mathfrak{a}'_{i-1} = C_{i-1} \sqsubseteq \forall U_{i-1}, U_i.+_{d_{i-1}} \in \mathit{saturate}(\mathcal{T})$. Using $\mathbf{R_{init}}$-5 and $\mathbf{R}_{+}$-2, we derive $C_{i-1} \sqcap \exists U_{i-1}.\top_{\mathbb{R}_\sim} \sqsubseteq \forall U_i, U_i.+_0$. Note that we have $\mathfrak{a}_{i-1} = C \sqsubseteq \exists U_{i-1}.\varpi_{i-1}$, so that we can use $\mathbf{R}_{\bot}$-3 to infer $C \sqsubseteq \forall U_i, U_i.+_0$. The only rules that take an axiom of the form $\mathfrak{a}_{i+1} = C \sqsubseteq \exists U_{i+1}.\varpi_{d+1}$ as a premise are $\mathbf{R}_{\varpi}$-1 and $\mathbf{R}_{\varpi}$-2, which means that $\mathfrak{a}'_{i+1} = C_{i+1} \sqsubseteq \forall U_{i+1}, U_{i+2}.+_{d_{i+1}}$. Again, we can use $\mathbf{R_{init}}$-5 and $\mathbf{R}_{+}$-2 to infer $C_{i+1} \sqcap \exists U_{i+2}.\top_{\mathbb{R}_\sim} \sqsubseteq \forall U_{i+1}, U_{i+1}.+_0$, and $\mathbf{R}_{\bot}$-3 to infer $C \sqsubseteq \forall U_{i+1}, U_{i+1}.+_0$.Now we can use $\mathbf{R_{init}}$-6 with the three axioms

$$\begin{aligned}
\mathfrak{a}_i = C_i \sqsubseteq \exists U_i, U_{i+1}.+_{d_i} \\
C \sqsubseteq \forall U_i, U_i.+_0 \\
C \sqsubseteq \forall U_{i+1}, U_{i+1}.+_0
\end{aligned}$$

to obtain $\mathfrak{a}''_i = C \sqsubseteq \forall U_i, U_{i+1}.+_{d_i}$. Based on this argument, we assume in the following, that for all $i \in \{1, n\}$, we have $\mathfrak{a}'_i = C_i \sqsubseteq \forall U_i, U_{i+1}.+_{d_i}$.

We can now use $\mathbf{R}_{+}$-2 stepwise on all axioms $\mathfrak{a}'_1, \ldots \mathfrak{a}'_n$ to obtain an axiom of the form

$$D \sqsubseteq \forall U_1, U_1.+_{d-d_1},$$

where every conjunct in D either occurs in C, or is of the form $\exists U_i.\top_{\mathbb{R}_\sim}$, where $C \sqsubseteq \exists U_i.\varpi_{d_{i+1}}$. Thus, in case we do not have $D' \subseteq C$, we can use $\mathbf{R}_{\bot}$-3 to infer $C \sqsubseteq \forall U_1, U_1.+_{d-d_1}$. Since by assumption, $d - d_1 \neq 0$, we can now use $\mathbf{R}_{\bot}$-1 on $C \sqsubseteq \exists U_1.\varpi_d$ to infer $C \sqsubseteq \bot$, which makes all preceding axioms redundant. We have derived a contradiction, since all the previous inferences are not included in $\mathit{saturate}(\mathcal{T})$ ■

It follows from these bounds that both the number of left-hand sides and the number of right-hand sides occuring in any axiom generated by Algorithm 1 is bounded. It follows that only finitely many axioms are added to the set, and that the algorithm terminates. □

In the following, we show an example of the saturation process of a given TBox $\mathcal{T}$.

Example 3. *Consider the TBox* $\mathcal{T} = \{A \sqsubseteq \exists U_1.{=_{2.6}},\ B \sqsubseteq \forall U_2, U_1.{+_{0.4}}\}$. *We show the computation of some axioms in saturate*$(\mathcal{T})$*, by applying the rules from Figure 1 as follows:*

$$\begin{aligned}\mathcal{T} \cup \{\ & \xrightarrow{\mathbf{R}_{init}-5} B \sqsubseteq \forall U_1, U_2.{+_{-0.4}}\ ,\ \xrightarrow{\mathbf{R}_{init}-4} \exists U_1.\top_{\mathbb{R}_\sim} \sqcap \exists U_2.\top_{\mathbb{R}_\sim} \sqcap B \sqsubseteq \exists U_2, U_1.{+_{0.4}}\ ,\\ & \xrightarrow{\mathbf{R}_{init}-5} \exists U_1.\top_{\mathbb{R}_\sim} \sqcap \exists U_2.\top_{\mathbb{R}_\sim} \sqcap B \sqsubseteq \exists U_1, U_2.{+_{-0.4}}\ ,\\ & \xrightarrow{\mathbf{R}_{+}-2} \exists U_2.\top_{\mathbb{R}_\sim} \sqcap B \sqsubseteq \forall U_1, U_1.{+_0}\ ,\ \exists U_1.\top_{\mathbb{R}_\sim} \sqcap B \sqsubseteq \forall U_2, U_2.{+_0}\ ,\\ & \xrightarrow{\mathbf{R}_{init}-2} \exists U_2.\top_{\mathbb{R}_\sim} \sqcap A \sqcap B \sqsubseteq \forall U_1.{=_{2.6}}\ ,\ \xrightarrow{\mathbf{R}_{\varpi}-1} \exists U_2.\top_{\mathbb{R}_\sim} \sqcap A \sqcap B \sqsubseteq \forall U_2.{=_{2.2}}\ ,\ \ldots\}.\end{aligned}$$

5 Query Rewriting

This section presents the computation procedure for rewriting a given UCQ Φ w.r.t. a saturated TBox. The idea is to apply a set of rules on Φ, so that the union over the resulting set of CQs Φ' has exactly those certain answers over $\langle\emptyset, \mathcal{A}\rangle$ as Φ has over $\langle\mathcal{T}, \mathcal{A}\rangle$. We make the following *simplifying assumptions* on CQs ϕ in Φ: *i)* for every atom of the form $=(t_a, t'_a) \in \phi$, $t_a \in \mathsf{var}_{\mathrm{ans}}(\phi)$, and *ii)* $\mathbf{terms}(\phi) \cap \mathbb{R} = \emptyset$. These assumptions are w.l.o.g. since for *i)*, if $=(t_a, t'_a) \in \phi$ and $t_a \notin \mathsf{var}_{\mathrm{ans}}(\phi)$, we can replace t_a by t'_a exhaustively in the query, and equality between different constants that would make the query unsatisfiable can be easily detected; and for *ii)*, since we can replace every real number $d \in \mathbf{terms}(\phi)$ by a variable v_d, for which we add the atom $=_d(v_d)$. To keep queries in that form, we use a slightly non-standard notion of substitutions.

Definition 5.1 (Substitution)**.** Let ϕ be a CQ, α an atom s.t. $\alpha \in \phi$. A *substitution on* ϕ is a function $\sigma : \mathbf{terms}(\phi) \to \mathsf{N_V} \cup \mathsf{N_I}$, with the additional requirement that $\sigma(t) = t$ if $t \in \mathsf{N_I}$. The result of applying σ on atom α (in symbols: $\alpha\sigma$) is an atom obtained by replacing every term $t \in \mathbf{terms}(\alpha)$ by $\sigma(t)$. The result of applying σ on CQ ϕ (in symbols: $\phi\sigma$) is obtained from ϕ by replacing every atom $\alpha \in \phi$ of the form $=(t_1, t_2)$ by $=(t_1, \sigma(t_2))$ and every other atom α by $\alpha\sigma$, and adding an atom $=(x, t)$ for every answer variable x s.t. $\sigma(x) = t \neq x$.

Note the special treatment of atoms of the form $=(t_1, t_2)$: by assumption *i)*, these are only used to express equality of answer variables with other terms. Definition 5.1 ensures that we can substitute answer variables in the remaining query without affecting the answers of the query.

Before we discuss the rewriting rules, we need to address the syntactic mismatch between queries and axioms. Specifically, the rewriting rules may operate on roles or complex concepts as they can refer to axioms. Since the syntax of CQs does not admit these, we employ equivalences that "bridge" this gap between query and rules. The idea is that the syntactic matching of rules to a CQ is done modulo these equivalences. Let ϕ be a CQ, $\{x, y\} \subseteq \mathsf{N_V}$, and X be a role or a concept allowed on the left-hand side of an axiom. If X is a role and $X = r^-$, then $(\phi \wedge r^-(x, y)) \equiv (\phi \wedge r(y, x))$. Let $v, w \in \mathsf{N_V} \setminus \mathbf{var}(\phi)$. If X is a concept, then, depending on the structure of X, the CQ $\widehat{\phi} = (\phi \wedge X(x))$ is equivalent to:

- $\phi \wedge C_1(x) \wedge C_2(x)$, if $X = C_1 \sqcap C_2$;
- $\phi \wedge U(x, v) \wedge \Pi(v)$, if $X = \exists U.\Pi$; but it is $\phi \wedge U(x, w)$, if $C = \exists U.\top_{\mathbb{R}_\sim}$;
- $\phi \wedge U_1(x, v) \wedge U_2(x, w) \wedge \Pi(v, w)$, if $X = \exists U_1, U_2.\Pi$;
- $\phi \wedge A(x)$, if $X = A \in \mathsf{N_C}$;
- ϕ, if $X = \top$;
- $\phi \wedge R(x, w)$, if $X = \exists R$;

RR1	$\dfrac{\phi \wedge X(\vec{t})}{\phi \wedge Y(\vec{t})}$	$Y \sqsubseteq X'$, $\models X' \sqsubseteq X$	**RR2**	$\dfrac{\phi \wedge \Pi_1(v)}{\phi \wedge C(x) \wedge U(x,v)}$	$C \sqsubseteq \forall U.\Pi_2$, $\Pi_2 \models \Pi_1$
RR3	$\dfrac{\phi \wedge +_d(v_1,v_2)}{\phi \wedge C(x) \wedge U_1(x,v_1) \wedge U_2(x,v_2)}$				$C \sqsubseteq \forall U_1, U_2.+_d$
RR4	$\dfrac{\phi \wedge U(x,v)}{\phi \wedge C(x) \wedge =_d(v)}$				$C \sqsubseteq \exists U.=_d$
RR5	$\dfrac{\phi \wedge U_1(x,v_1) \wedge U_2(y,v_2) \wedge +_d(v_1,v_2)}{\phi[y \mapsto x] \wedge C(x) \wedge U_1(x,v_1)}$				$C \sqsubseteq \exists U_1, U_2.+_d$, $v_2 \notin \mathsf{var}(\phi)$
RR6	$\dfrac{\phi \wedge U_1(x,v_1) \wedge \varpi_{d_1}(v_1)}{\phi \wedge C(x) \wedge U_1(x,v_1) \wedge U_2(x,v_2) \wedge \varpi_{d_1+d_2}(v_2)}$				$C \sqsubseteq \forall U_1, U_2.+_{d_2}$
RR7	$\dfrac{\phi \wedge U_1(x,v_1) \wedge \varpi_{d_1}(v_1)}{\phi \wedge (C_1 \sqcap C_2)(x) \wedge U_2(x,v_2) \wedge +_{d_2}(v_1,v_2) \wedge \varpi_{d_1+d_2}(v_2)}$				$C_1 \sqsubseteq \forall U_2, U_2.+_0$, $C_2 \sqsubseteq \exists U_1, U_2.+_{d_2}$
RR8	$\dfrac{\phi \wedge U_1(x,v_1) \wedge +_{d_1}(v_2,v_1)}{\phi \wedge C(x) \wedge U_1(x,v_1) \wedge U_2(x,v_3) \wedge +_{d_1+d_2}(v_2,v_3)}$				$C \sqsubseteq \forall U_1, U_2.+_{d_2}$
RR9	$\dfrac{\phi \wedge U_1(x,v_2) \wedge +_{d_1}(v_1,v_2)}{\phi \wedge (C_1 \sqcap C_2)(x) \wedge U_2(x,v_3) \wedge +_{d_1+d_2}(v_1,v_3) \wedge +_{d_2}(v_2,v_3)}$				$C_1 \sqsubseteq \forall U_2, U_2.+_0$, $C_2 \sqsubseteq \exists U_1, U_2.+_{d_2}$

Figure 2: Query rewriting rules dependant on the TBox. (If a variable x occurs only in the conclusion of a rule, we assume ϕ contains an atom of the form $U'(x,v')$.)

RC1	$\dfrac{\phi \wedge +_{d_1}(v_1,v_2) \wedge +_{d_2}(v_2,v_3)}{\phi \wedge +_{d_1}(v_1,v_2) \wedge +_{(d_1+d_2)}(v_1,v_3)}$	**RC2**	$\dfrac{\phi \wedge +_d(v_1,v_2)}{\phi \wedge +_{(-d)}(v_2,v_1)}$
RC3	$\dfrac{\phi \wedge =_{d_1}(v_1) \wedge =_{d_2}(v_2)}{\phi \wedge =_{d_1}(v_1) \wedge +_{(d_2-d_1)}(v_1,v_2)}$	**RC4**	$\dfrac{\phi \wedge =_{d_1}(v_1) \wedge +_{d_2}(v_1,v_2)}{\phi \wedge =_{d_1}(v_1) \wedge =_{(d_1+d_2)}(v_2)}$
RC5	$\dfrac{\phi \wedge +_{d_2}(v_1,v_2) \wedge \varpi_{d_1}(v_1)}{\phi \wedge +_{d_2}(v_1,v_2) \wedge \varpi_{(d_1+d_2)}(v_2)}$	**RC6**	$\dfrac{\phi \wedge \Pi_1(v) \wedge \Pi_2(v)}{\phi \wedge \Pi_1(v)}$, $\Pi_1 \models \Pi_2$
RC7	$\dfrac{\phi \wedge U_1(x_1,v_1) \wedge U_2(x_2,v_1)}{\phi \wedge U_1(x_1,v_1) \wedge U_2(x_2,v_2) \wedge +_0(v_1,v_2)}$	**RC8**	$\dfrac{\phi \wedge \varpi_d(w)}{\phi}$
RC9	$\dfrac{\phi \wedge U(x,v) \wedge U(x,w) \wedge +_0(v_1,w)}{\phi \wedge U(x,v)}$	**RC10**	$\dfrac{\phi \wedge +_d(v,w)}{\phi}$

Figure 3: Query rewriting rules dependant on the TBox. (Here, w denotes a unique non-distinguished variable.)

We are now prepared to discuss the rewriting rules. The rewriting rules are grouped into TBox-dependent rules (see Figure 2) and TBox-independent rules (see Figure 3). **RR1** is the standard rewriting rule as used in most rewriting procedures for DL-Lite, and **RR2** and **RR3** are the variant dealing with universal restrictions, already used in [2]. **RR5** corresponds to a special case of **RR1** in which an additional substitution step is required. The main function of most rules in Figure 3 is to reformulate the concrete domain expressions in the query in an equivalence preserving way. While the TBox saturation already computes some inferences between concrete domain predicates, not every possible combination of concrete domain predicates that may occur in a query can be treated without knowing the query. It is thus possible that the query contains concrete domain predicates saturation has not considered yet, but which can be transformed into the required form. This is done by Rules **RC1** to **RC5** and **RC7**. The purpose of other rules is to reduce the number of occurrences of a variable, to make other rules applicable. This is the main motivation behind **RR4**, **RC6**, **RC9** and **RC10**, and is also achieved by **RR6** to **RR9**. Note that we have to be careful here not to lose the "link" of any variable occurring in the rest of the query: if we would drop a variable occurring elsewhere, we would allow for additional matches not covered by the original query. In [2], shared variables in the query are eliminated using a special *splitting-rule*, which splits shared occurrences of a variable by assigning a fixed value to them, where the set of values is determined based on the TBox and the ABox. Since we want to obtain a data-independent and goal-oriented procedure, we cannot follow this route in our rewriting procedure.

However, in order to achieve full data-independence, a bit more has to be done. Note that the fillers of concrete domain attributes can be determined by both: numbers occurring in the data and axioms in the TBox. If a predicate in the query refers to an attribute filler implicit in the data, we may need to "push" the concrete domain predicates in the query towards those attribute fillers explicit in the data. This is the purpose of rules **RR6** to **RR9**. Note that these rules, similar to some of the TBox saturation rules, may make use of local functionalities of an attribute expressed by an axiom of the form $C \sqsubseteq \forall U, U.+_0$.

To obtain a rewriting procedure that is both complete and terminating, two problems have to be addressed. First, the rules need not be applicable to a query, as some rules require certain variables to occur only once or twice in the query. If a CQ contains a variable multiple times, it is often possible to reduce the number of occurrences by applying appropriate substitutions. Second, termination of the rewriting has to be ensured. Note that rules **RR6** to **RR9** rely on predicate atoms of the form $\varpi_d(v)$ (or $+_d(v, v')$) in their precondition and generate predicate atoms with same kind of predicate, but with a new value for d and thus with a new predicate and with different variables. This generation process can continue, but it generates only redundant queries. We adress the two problems in the following.

Achieving applicability of rules can require to unify some variables which is usually achieved by applying substitutions. From a single CQ, a set of many CQs can be derived from the same CQ by such substitutions. One can easily see that not all possible substitutions are relevant here, but only those that unify different atoms. We make this intuition formal. Let $A = \{\alpha_1, \ldots, \alpha_n\}$ be a set of atoms. Set A *unifies* (to a singleton) if there exists a substitution σ s.t. $\alpha_1\sigma = \ldots = \alpha_n\sigma$. We then call σ a *unifier of* A. For each such set, we select a *most general unifier* $\mathsf{mgu}(A)$, which is defined as a unifier σ of A s.t. for every other unifier σ' of A, there exists some substitution σ'' s.t. $\sigma \circ \sigma'' = \sigma'$. We define the set of all CQs that can be obtained by unifying any set of atoms in ϕ as $\mathsf{reduce}(\phi) = \{\phi\sigma \mid \sigma = \mathsf{mgu}(A), A \subseteq \phi, A \text{ unifies}\}$.

Let Φ be a UCQ. For every $\phi \in \Phi$, due to our notion of redundancy that is to be defined next, every $\phi' \in \mathsf{reduce}(\phi)$ is redundant. This is why $\mathsf{reduce}(\phi)$ is not defined as a rewriting rule, but as a separate procedure.

Achieving termination of rule application depends on limiting the number of new predicates introduced by rules, and on limiting the number of variables introduced. The latter effect can be remedied by avoiding redundant queries in the query set.

Definition 5.2 (Redundant query)**.** Given two CQs ϕ_1 and ϕ_2, ϕ_2 is *redundant* to ϕ_1, iff there exists a substitution σ s.t. $\phi_1\sigma \subseteq \phi_2$. CQ ϕ_2 is redundant in a UCQ Φ if there exists $\phi_1 \in \Phi$ to which ϕ_2 is redundant.

By Lemma 5.1, removing redundant CQs from Φ does not affect the entailment of Φ.

Lemma 5.1 (Query redundancy elimination)**.** *Let Φ be a Boolean UCQ, and $\phi_1, \phi_2 \in \Phi$ s.t. ϕ_2 is redundant to ϕ_1. It holds for every interpretation $\mathcal{I}$, that $\mathcal{I} \models \Phi$ iff $\mathcal{I} \models \Phi \setminus \{\phi_2\}$.*

Proof. Let $\mathcal{I}$ be an arbitrary interpretation s.t. $\mathcal{I} \models \Phi \setminus \{\phi_2\}$. By the definition of entailment of UCQs, and since $\Phi \setminus \{\phi_2\} \subsetneq \Phi$, we have $\mathcal{I} \models \Phi$. For the reverse direction, let $\mathcal{I}$ be an arbitrary interpretation, and assume $\mathcal{I} \models \Phi$ and $\mathcal{I} \not\models \Phi \setminus \{\phi_2\}$. This means two things. First, $\mathcal{I} \models \phi_2$, and therefore there exists a homomorphism $h : \mathbf{terms}(\phi_2) \to \Delta^{\mathcal{I}} \cup \mathbb{R}$. Second, $\mathcal{I} \not\models \phi_1$. But since ϕ_2 is redundant to ϕ_1, there exists a substitution σ s.t. $\phi_1\sigma \subseteq \phi_2$. We know that σ is a mapping from terms in ϕ_1 to terms in ϕ_2, and h is a mapping from terms in ϕ_2 to elements in $\Delta^{\mathcal{I}} \cup \mathbb{R}$. Therefore, we can construct a mapping $h' = \sigma \circ h$ s.t. $h' : \mathbf{terms}(\phi_1) \to \Delta^{\mathcal{I}} \cup \mathbb{R}$. The function h' is indeed a homomorphism from terms of ϕ_1 into $\mathcal{I}$. Hence $\mathcal{I} \models \phi_1$, and consequently, $\mathcal{I} \models \Phi \setminus \{\phi_2\}$, which contradicts the original assumption. □

The algorithm to compute the rewriting of a UCQ Φ, written as $rew(\Phi)$, uses the rewriting rules shown in Figures 2 and 3, where X and Y are both either roles or concepts. The complete UCQ rewriting is depicted in Algorithm 2.

Example 4. *Let $\mathcal{T} = \{\mathfrak{a}_1 = A \sqsubseteq \exists U_3, U_1.{+_3},\ \mathfrak{a}_2 = A \sqsubseteq \forall U_1, U_2.{+_1},\ \mathfrak{a}_3 = B \sqsubseteq \exists U_2.\top_{\mathbb{R}_\sim}\}$ be a TBox, $\mathcal{A} = \{A(c), B(c), U_1(c, 10)\}$ an ABox, and $\phi = \exists v_3.(U_3(x, v_3) \wedge {>_5}(v_3))$ a CQ. Then, $\mathbf{a}$ with $\mathbf{a}(x) = c$ is a certain answer to ϕ in $\langle \mathcal{T}, \mathcal{A}\rangle$. In the following, we show how it is obtained. First, we compute saturate($\mathcal{T}$), which contains $\mathfrak{a}_4 = \exists U_2.\top_{\mathbb{R}_\sim} \sqcap A \sqsubseteq \forall U_1, U_1.{+_0}$, among other axioms. This axiom is derived by applying $\mathbf{R}_{\mathbf{init}}$-5 on $\mathfrak{a}_2$, and then $\mathbf{R}_+$-2 on the result and $\mathfrak{a}_2$. To obtain the CQ for which $\mathbf{a}$ is a certain answer in $\langle \emptyset, \mathcal{A}\rangle$, we compute rew($\phi$), which consists of ϕ and the following queries:*

$$
\begin{aligned}
&\overset{RR7(\mathfrak{a}_1,\mathfrak{a}_4)}{\rightsquigarrow} && A(x) \wedge U_1(x, v_1) \wedge U_2(x, v_2) \wedge {+_3}(v_3, v_1) \wedge {>_8}(v_1)\\
&\overset{RC10}{\rightsquigarrow} && A(x) \wedge U_1(x, v_1) \wedge U_2(x, v_2) \wedge {>_8}(v_1)\\
&\overset{RR1(\mathfrak{a}_3)}{\rightsquigarrow} && A(x) \wedge U_1(x, v_1) \wedge B(x) \wedge {>_8}(v_1) \quad = \phi'.
\end{aligned}
$$

In $\langle \emptyset, \mathcal{A}\rangle$, ϕ' has a match, and therefore $\mathbf{a}$ is a certain answer to rew(ϕ).

Let $\mathcal{T}$ be a saturated TBox, ϕ a CQ, and **RR** some rewriting rule. In Algorithm 2, we denote by $\phi \overset{\mathbf{RR},\mathcal{T}}{\rightsquigarrow} \phi'$ a rewriting step w.r.t. $\mathcal{T}$ s.t. some atom(s) in ϕ satisfy the premise(s) of **RR**, and the resulting query ϕ' is in the form of the conclusion of **RR**.

Soundness of the rewriting procedure. We first show that our rewriting procedure is indeed sound. Soundness of the first rewriting Rule, Rule **RR1**, is a consequence of the following lemma, which will later also become convenient in the completeness proof.

Lemma 5.2. *Let ϕ be a CQ and $\mathcal{I}$ an (abstract) interpretation s.t. there exists a homomorphism h from ϕ into $\mathcal{I}$ and $x \in \mathbf{terms}(\phi)$. Then, there exists a homomorphism from $\phi \wedge X_1(x)$ into $\mathcal{I}$ iff $h(x) \in X_1^{\mathcal{I}}$, where X_1 is a concept that can occur on the left-hand side of a GCI.*

Algorithm 2: Computation of $rew(\Phi)$

Input: UCQ Φ, saturated TBox $\mathcal{T}$.
for $\phi \in \Phi$ **do**
  for $\phi_r \in \mathsf{reduce}(\phi)$ *and* ψ *such that* $\phi_r \overset{\mathbf{RR},\mathcal{T}}{\rightsquigarrow} \psi$ **do**
    if ψ *is satisfiable w.r.t.* $\mathcal{T}$ *and not redundant in* Φ **then**
      set $\Phi \coloneqq \{\psi' \in \Phi \mid \psi' \text{ is not redundant to } \psi\} \cup \{\psi\}$;
return Φ;

Proof. We distinguish the cases based on X_1.

- If $X_1 \in \mathsf{N_C}$,we have $\mathcal{I} \models X_1(h(x))$ iff $h(x) \in X_1^{\mathcal{I}}$, and consequently iff h is also a homomorphism from $\phi \wedge X_1(x)$ into $\mathcal{I}$.
- The case where $X_1 = \top$ is trivial.
- If X_1 is of the form $\exists R$, then $h(x) \in (\exists R)^{\mathcal{I}}$ iff $\langle h(x), e\rangle \in R^{\mathcal{I}}$ for some $e \in \Delta^{\mathcal{I}}$. Provided that $h(x) \in (\exists R)^{\mathcal{I}}$, we can thus extend h to a homomorphism h' from $\phi \wedge R(x,y)$ into $\mathcal{I}$ by setting $h'(y) = e$. On the other hand, if h can be extended to a homomorphism h' from $\phi \wedge R(x,y)$ into $\mathcal{I}$, we must have $h(x) \in (\exists R)^{\mathcal{I}}$.
- Assume $X_1 = \exists U_1, \ldots, U_n.\Pi$. If $h(x) \in X_1^{\mathcal{I}}$, then $\langle h(x), d_1\rangle \in U_1^{\mathcal{I}}$, ..., $\langle h(x), d_n\rangle \in U_n^{\mathcal{I}}$ s.t. $\langle d_1, \ldots, d_n\rangle \in \Pi^{\mathbb{R}_\sim}$, and consequently we can extend h to a homomorphism h' from $\phi \wedge U_1(x, y_1) \wedge \ldots \wedge U_n(x, y_n) \wedge \Pi(v_1, \ldots, v_n)$ into $\mathcal{I}$ by setting $h'(y_i) = d_i$ for all i, $1 \leq i \leq n$. For the other direction, assume such a homomorphism h' exists. Clearly, then also $h(x) \in X_1^{\mathcal{I}}$.
- If $X_1 = Y_1 \sqcap \ldots \sqcap Y_n$, by induction we obtain that for every $i \in [\![1, n]\!]$, there is a homomorphism h_i from $\phi \wedge Y_i(x)$ into $\mathcal{I}$. Note that by definition of $Y_i(x)$, the only variable shared by $Y_i(x)$ and ϕ is x, and by the constructions above, for all $y \in \mathbf{terms}(\phi)$, $h_i(y) = h(y)$. Hence, we can define the homomorphism h' by setting $h'(y) = h(y)$ for $y \in \mathbf{terms}(\phi)$ and $h'(y) = h_i(y)$ for $y \in \mathbf{terms}(Y_i(x)) \setminus \{x\}$. h' is a homomorphism from $\phi \wedge Y_1(x) \wedge \ldots \wedge Y_n(x)$ into $\mathcal{I}$. □

Lemma 5.3. *Let $\mathcal{K}$ be a KB, Φ a Boolean UCQ and $\phi \in rew(\Phi)$. Then, if $\mathcal{K} \models \phi$, also $\mathcal{K} \models \Phi$.*

Proof. Let $\mathcal{K}$ be a KB. The queries in $rew(\Phi)$ are obtained by the application of the rewriting rules, as well as by the function reduce. It is not hard to see that for every CQ ϕ and $\psi \in \mathsf{reduce}(\phi)$, since ψ is obtained from ϕ by applying a substitution, if $\mathcal{K} \models \phi$, then also $\mathcal{K} \models \psi$. To complete the proof, we need to show that for every CQ ϕ and ψ s.t. ψ is obtained from ϕ by applying a rewriting rule, we have that $\mathcal{K} \models \psi$ implies $\mathcal{K} \models \phi$, the theorem then follows by induction on the rule applications. Let $\phi \in rew(\Phi)$ and ψ be obtained from ϕ using a rewriting rule. To show that $\mathcal{K} \models \psi$ implies $\mathcal{K} \models \phi$, we show that for every model $\mathcal{I}$ of $\mathcal{K}$, $\mathcal{I} \models \psi$ implies $\mathcal{I} \models \phi$.

Let $\mathcal{I}$ be a model of $\mathcal{K}$, and let ϕ and ψ be two queries such that ψ is obtained from ϕ by applying a rewriting rule, and that $\mathcal{I} \models \psi$. There then exists a homomorphism h from ψ into $\mathcal{I}$.

We distinguish the cases based on which rewriting rule has been applied. It is not hard to see that the rules in Figure 3 in each case produce a logically equivalent query. We thus focus on the rules in Figure 2.

RR1 If $\psi = \psi' \wedge Y(\vec{t})$, by Lemma 5.2, then $h(\vec{t}) \in Y^{\mathcal{I}}$. Since $Y \sqsubseteq X' \in saturate(\mathcal{T})$ and $\mathcal{I} \models saturate(\mathcal{T})$, $Y^{\mathcal{I}} \subseteq (X')^{\mathcal{I}}$, and consequently, $h(\vec{t}) \in (X')^{\mathcal{I}}$. Furthermore, we have

$\models X' \sqsubseteq X$, and thus, $h(\vec{t}) \in X^{\mathcal{I}}$. By Lemma 5.2, there then exists a homomorphism from $X(\vec{t})$ into $\mathcal{I}$, and since $X(\vec{t})$ shares only the terms in $\vec{t}$ with the remaining query in ψ, this homomorphism is also a homomorphism from ϕ to $\mathcal{I}$.

RR2 If $\psi = \psi' \wedge C(x) \wedge U(x,y)$, then $h(x) \in C^{\mathcal{I}}$ and $\langle h(x), h(y)\rangle \in U^{\mathcal{I}}$. Since $\mathcal{I} \models C \sqsubseteq \forall U.\Pi_2$ and $\Pi_2 \models \Pi_1$, we must have $h(y) \in \Pi_1^{\mathbb{R}^\sim}$, so that h is a homomorphism from ϕ to $\mathcal{I}$.

RR3 This case is similar to the previous case.

RR4 Assume $\psi = \psi' \wedge C(x) \wedge =_d(y)$ and $C \sqsubseteq \exists U.=_d \in \mathit{saturate}(\mathcal{T})$. Then, $h(x) \in C^{\mathcal{I}}$ and $h(y) = d$. By the GCI, $h(y) = d$ is a U-successor of $h(x)$, and consequently, $\langle h(x), h(y)\rangle \in U^{\mathcal{I}}$.

RR5 This case corresponds to a special case of **RR1**, where we first apply the substitution $[y \mapsto x]$ on ϕ. Note that $\mathcal{I} \models \phi[y \mapsto x]$ implies $\mathcal{I} \models \phi$, via a homomorphism h' obtained from h by mapping x to $h(y)$.

RR6 This case follows from the semantics of $+_{d_2}$.

RR7 Assume $\mathcal{I} \models \psi$ via the homomorphism h, $C_1 \sqsubseteq \forall U_2, U_2.+_0 \in \mathit{saturate}(\mathcal{T})$ and $C_2 \sqsubseteq \exists U_1, U_2.+_{d_2} \in \mathit{saturate}(\mathcal{T})$. Since $h(x) \in (C_1 \sqcap C_2)^{\mathcal{I}}$, and $\mathcal{I} \models C_1 \sqsubseteq \forall U_2, U_2.+_0$, $h(x)$ has at most one U_2 successor, whose value is $h(v_2)$. Since furthermore $\mathcal{I} \models C_2 \sqsubseteq \exists U_1, U_2.+_{d_2}$, $h(x)$ has a U_1-successor whose value is d_2 less than $h(v_2)$, that is, $h(v_2) - d_2$. Since $\mathcal{I} \models \psi'$ via h, we also have $\langle h(v_1), h(v_2)\rangle \in (+_{d_2})^{\mathbb{R}^\sim}$, which means that $h(v_1) = h(v_2) - d_2$. It follows that $h(v_1)$ is the before-mentioned U_1-successor of $h(x)$, and $\langle h(x), h(v_1)\rangle \in U_1^{\mathcal{I}}$. It remains to show that $h(v_1) \in \varpi_{d_1}^{\mathbb{R}^\sim}$. Since $\varpi_{d_1+d_2}(v_2) \in \psi'$, we have $h(v_2) \in (\varpi_{d_1+d_2})^{\mathbb{R}^\sim}$. Note that ϖ is one of $=$, $<$ or $>$, and that $h(v_1) = h(v_2) - d_2$. Consequently, $h(v_1) \in (\varpi_{d_1})^{\mathbb{R}^\sim}$, and $\mathcal{I} \models \phi$.

RR8 This case is similar to **RR6**.

RR9 This case is similar to **RR7**.

We showed that for every model $\mathcal{I}$ of $\mathcal{K}$, $\mathcal{I} \models \psi$ implies $\mathcal{I} \models \phi$, provided that ψ is obtained from ϕ using one of our rewriting rules. It follows that $\mathcal{K} \models \psi$ implies $\mathcal{K} \models \phi$. Using our earlier observation regarding the function reduce it follows by induction on the rule applications that for every $\phi \in \mathit{rew}(\Phi)$, $\mathcal{K} \models \phi$ implies $\mathcal{K} \models \Phi$. □

Theorem 5.1 (Soundness of Rewriting)**.** *For every KB $\mathcal{K}$, UCQ Φ, $\phi \in \mathit{rew}(\Phi)$ and and answer $\mathbf{a}$ to Φ, if $\mathbf{a}$ is a certain answer to ϕ in $\mathcal{K}$, then it is a certain answer to Φ in $\mathcal{K}$.*

Proof. Note that i) no rewriting step changes the number of answer variables and ii) otherwise, answer variables are treated just as constants. As a consequence, for every CQ $\phi \in \mathit{rew}(\Phi)$ and every certain answer $\mathbf{a}$ to ϕ in $\mathcal{K}$, $\mathbf{a}(\phi) \in \mathbf{a}(\mathit{rew}(\Phi)$, where $\mathbf{a}(\phi)$ and $\mathbf{a}(\mathit{rew}(\Phi))$ are obtained from ϕ and Φ by replacing each answer variable x by $\mathbf{a}(x)$. This implies by Lemma 5.3 that $\mathcal{K} \models \mathbf{a}(\phi)$ implies $\mathcal{K} \models \mathbf{a}(\Phi)$, and thus that, if $\mathbf{a}$ is a certain answer to ϕ in $\mathcal{K}$, then $\mathbf{a}$ is a certain answer to Φ in $\mathcal{K}$. □

In order to prove termination of the algorithm, we need to show that the number of generated rewritings (CQs) is bounded. Actually, whether a rewriting rule depends on some axiom $\mathfrak{a} \in \mathcal{T}$ or not, it is in theory possible to generate an infinite number of CQs using our rewriting rules, unless we restrict the addition of new CQs appropriately. The reason behind this is that some of these rules may introduce an unbounded number of variables and atoms with new concrete domain predicates. We show, in Theorem 5.2, that queries which would trigger an unbounded application of rewriting rules are indeed either redundant or unsatisfiable. Thus, eliminating such queries prevents the generation of an unbounded number of CQs.

Theorem 5.2 (Termination of rewriting)**.** *Algorithm* 2 *always terminates.*

Proof. We show that for any given input Φ, the set of queries that are generated by Algorithm 2 is bounded.

For convenience, we may sometimes assume that Rule $\mathbf{R_{init}}$-5 is applied silently, and identify concepts $\mathsf{Q}U_1, U_2.+_d$ with $\mathsf{Q}U_2, U_1.+_{-d}$.

In the following, we call a variable x occurring in a query ψ an atom of the form $A(x)$, $A \in \mathsf{N_C}$, $r(x, y)$, $r \in \mathsf{N_R}$ or in the first place in an atom of the form $U(x, v)$, $U \in \mathsf{N_A}$, *object variable in* ψ, and every other variable *concrete domain variable*. We show termination of the rewriting procedure by means of the following claims.

Claim 1. The number of object variables in a query $\psi \in \mathit{rew}(\Phi)$ is bounded.

Proof of claim. The only rule that may introduce a new object variable is **RR1**, where $Y = \exists R$ or $Y = \exists R^-$, in which case some sub-query $X(x)$ is replaced by $R(x, w)/R(w, x)$, where w does not occur anywhere else. The only rule applicable on this atom is again **RR1**, which then however would eliminate the occurrence either of the fresh variable w, or the occurrence of x, in the case where x is neither shared nor an answer variable. We obtain that any query $\psi \in \mathit{rew}(\Phi)$ may have at most one additional object variable compared to ϕ for each axiom of the forms $\exists R \sqsubseteq X$, $\exists R^- \sqsubseteq Y \in \mathit{saturate}(\mathcal{T})$, and each object variable occurring in ϕ. ■

Claim 2. The overall number of concrete domain predicates introduced by the rewriting procedure is bounded.

Proof of claim. We show that every sequence of rewriting rule applications that would lead to the introduction of an unbounded number of concrete domain predicates involves an unsatisfiable query. This query is removed by our rewriting procedure before further steps are applied, which contradicts that the sequence of rule applications is applied indefinitely.

The rules that introduce new concrete domain predicates to a query are **RC1**, **RC2**, **RC3**, **RC4** and **RC5**, as well as **RR6**, **RR7**, **RR8** and **RR9**. For rules **RC1**, **RC2**, **RC3**, **RC4** and **RC5**, we notice that they all produce logically equivalent queries with the same amount of concrete domain variables. If applying only these rules on some query ψ leads to the introduction of two atoms $+_d(v_1, v_2)$ and $+_{d'}(v_1, v_2)$, where $d \neq d'$, ψ must be unsatisfiable, as those atoms contradict each other and the resulting query is logically equivalent. The same holds for atoms of the forms $=_d(v)$ and $=_{d'}(v)$. Regarding atoms of the form $\varpi_d(v)$, we notice that those atoms are only introduced by Rule **RC5**, and one easily sees that, as a consequence of the previous observation, this rule can only introduce a finite number of such atoms for each concrete domain variable, provided that the original query is satisfiable. We can thus assign to every satisfiable query $\psi \in \mathit{rew}(\Phi)$ a finite set of queries produced using only the rules in Figure 3, as well as Rules **RR1**, **RR2**, **RR3**, **RR4** and **RR5**. For a given CQ $\psi \in \mathit{rew}(\Phi)$, denote this set by $\mathrm{rew}_1(\psi)$.

We now focus on those rules that may introduce new concrete domain predicates but depend on the TBox, that is, rules **RR6**, **RR7**, **RR8** and **RR9**. If these rules may create an unbounded number of concrete domain predicates, then there is some axiom $\mathfrak{a} \in \mathit{saturate}(\mathcal{T})$ that is used an unbounded number of times. We distinguish the two different cases based on the syntactical shape of that axiom $\mathfrak{a}$.

- Assume $\mathfrak{a}$ is of the form $C_1 \sqsubseteq \exists U_1, U_2.+_{d_1}$. This means the introduction of an unbounded number of concrete domain predicates is due to Rules **RR7** and **RR9**. Both rules require

an atom $U_1(x, y_1)$ in query, which is replaced by an atom $U_2(x, y_2)$. If $\mathfrak{a}$ is applied twice, we must thus have the following sequence of axioms in $saturate(\mathcal{T})$ that are used to produce the unbounded number of concrete domain predicates:

$$\begin{aligned} C_1 &\sqsubseteq \exists U_1, U_2.+_{d_1} & D_1 &\sqsubseteq \forall U_2, U_2.+_0 \\ C_2 &\sqsubseteq \exists U_2, U_3.+_{d_2} & D_2 &\sqsubseteq \forall U_3, U_3.+_0 \\ &\vdots \\ C_{n-1} &\sqsubseteq \exists U_{n-1}, U_n.+_{d_{n-1}} & D_{n-1} &\sqsubseteq \forall U_n, U_n.+_0 \\ C_n &\sqsubseteq \exists U_n, U_1.+_{d_n} & D_n &\sqsubseteq \forall U_1, U_1.+_0, \end{aligned}$$

where $d = \sum_{1 \leq i \leq n} d_n \neq 0$. (If $d = 0$, this sequence does not lead to the introduction of an unbounded number of concrete domain predicates, and we just get back the predicate with which we have started).

If we apply **RR7** or **RR9** with those axioms in order on a query ψ_1 containing $U_1(x, y)$, with y occurring in a unary atom (**RR7**) or a binary axiom (**RR9**), where in each step we add a new concrete domain variable bounded to x, we obtain a query ψ_n that contains $C_1(x)$, ..., $C_n(x)$, $D_1(x)$, ..., $D_n(x)$, as well as $U_1(x, v_{n+1})$.

Specifically, in any interpretation that allows for a homomorphism mapping x to some domain element e. e satisfies D_1, ..., D_n, which means it has at most one successor for each attribute U_1, ... U_n. Furthermore, e satisfies C_1, ..., C_n. By looking at the axioms above having those concepts on the left-hand side, and taking into consideration that there is at most one successor for each attribute, we obtain that e also satisfies $\exists U_1, U_1.+_d$. But since $d \neq 0$, this would imply that e has two U_1-successors, which contradicts the earlier observation that there can be only one such successor. We obtain that that ψ_n is unsatisfiable with $\mathcal{T}$, and would thus not be added to the current set of CQs. Now consider any sequence of rewriting rule applications that uses all these axioms in the same way, but in between might use additional rewriting steps. The resulting queries can all be obtained from ψ_n by just applying the rules in a different order. Since ψ_n is already unsatisfiable, by Theorem 5.1, these queries are unsatisfiable as well. As our rewriting procedure removes unsatisfiable queries, an unbounded sequence of rewriting steps as considered here will not happen with our algorithm.

- Assume $\mathfrak{a}$ is of the form $C_1 \sqsubseteq \forall U_1, U_2.+_{d_1}$. We can exclude Rules **RR7** and **RR9** in the application of the rewriting, since we already showed that they cannot lead to an unbounded sequence of inferences. We therefore only need to consider **RR6** and **RR8**. Each application of these rules shifts the occurrence of a concrete domain predicate to a new variable, for which it adds a new attribute atom. If **RR6** or **RR8** becomes applicable again, this must be on the new concrete domain predicate with the new attribute atom. We thus must have the following axioms in $saturate(\mathcal{T})$

$$\begin{aligned} C_1 &\sqsubseteq \forall U_1, U_2.+_{d_1} \\ C_2 &\sqsubseteq \forall U_2, U_3.+_{d_2} \\ &\vdots \\ C_n &\sqsubseteq \forall U_n, U_1.+_{d_n} \end{aligned}$$

Denote the result of applying **RR7** or **RR9** on some query ψ_1 with those axioms one after the other, each time on a concrete domain variable bound to the same object variable x, by ψ_n. Note that ψ_n contains $C_i(x)$ and $U_i(x, y_i)$ for each $i \in \{1, \ldots, n\}$.

If $d = \sum_{1 \leq i \leq n} d_n = 0$, then using these axioms in this order with **RR6** or **RR8** is harmless, as we end up with the same concrete domain predicate that we began with. Otherwise, we argue that ψ_n must be unsatisfiable with respect to $\mathcal{T}$.

We can combine all the above axioms step-wise using $\mathbf{R}_+$-2, obtaining the axiom

$$\prod_{1\leq i\leq n} C_n \sqcap \prod_{1\leq i\leq n} \exists U_i.\top_{\mathbb{R}_\sim} \sqsubseteq \forall U_1, U_1.+_{d_1+d},$$

and since $d \neq 0$, using $\mathbf{R}_\perp$-2 on this last axiom gives and $\mathfrak{a}_1$ gives us

$$\mathfrak{a}_\perp = \prod_{1\leq i\leq n} (C_i \sqcap \exists U_i).\top_{\mathbb{R}_\sim} \sqsubseteq \forall U_1, U_2.\perp$$

We note that the result of rewriting with the above axioms one after the other results in a query that contains $C_i(x)$ for each $i \in \{1,\dots,n\}$, as well as $U_i(x, v_i)$ for each $i \in \{1,\dots,n\}$. Thus, $\mathfrak{a}_\perp$ shows that ψ_n must be unsatisfiable. We can argue as before that any other sequence of rewriting rule applications on ψ_n that involves using **RR6** or **RR8** in the same way will also lead to an unsatisfiable query. We also obtain that this unbounded sequence of rule applications will not happen with our algorithm. ■

Claim 3. The number of domain predicate variables in a query $\psi \in \mathit{rew}(\Phi)$ bound to the same object variable is bounded

Proof of claim. The only rules that introduce new concrete domain predicaes are **RR1** and **RR6** to **RR9**. Rule **RR1** introduce at most one concrete domain variable per object variable and axiom, before the resulting axiom becomes redundant with respect to an earlier inferred CQ. Similarly, since rules **RR6** to **RR9** can only introduce a bounded number of concrete domain predicates, eventually a CQ will be inferred that is redundant to an earlier CQ. ■

As a result of Claim 1, Claim 2 and Claim 3, we establish that Algorithm 2 always generates a bounded set $\mathit{rew}(\Phi)$ of queries: the number of object variables in each query $\psi \in \mathit{rew}(\Phi)$ is bounded, each introduced concrete domain predicate is bound to some object variable, and the number of concrete domain predicate introduced per object variable is bounded. Our redundancy elimination further makes sure that we have no two queries in $\mathit{rew}(\Phi)$ that are the same modulo variable renaming. As a consequence, $\mathit{rew}(\Phi)$ is finite, and Algorithm 2 terminates. □

Furthermore, we can show that the obtained rewritings also yield a complete query answering procedure, which we show in the following subsection. It will be convenient to first focus on Boolean queries for these results.

5.1 Completeness of the Rewriting

We next show that our rewriting procedure is *complete* for Boolean queries Φ, that is, whenever $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle \models \Phi$, then $\langle \emptyset, \mathcal{A}\rangle \models \mathit{rew}(\Phi)$. Since $\mathit{rew}(\Phi)$ is contained in the union of all $\mathit{rew}(\phi)$ for which $\phi \in \Phi$, it suffices to focus on CQs ϕ. In the following of this subsection, we thus assume ϕ to be a fixed CQ.

The general idea is to define a kind of canonical model $\mathcal{I}$ of $\mathcal{K}$ such that for every CQ ψ, whenever $\mathcal{K} \models \psi$ then also $\mathcal{I} \models \psi$. This canonical model $\mathcal{I}$ is obtained based on a model of the ABox by extending it step-wise based on the TBox axioms. As a result, the canonical model construction yields a sequence $\mathcal{I}_0, \dots$ of interpretations. We then show completeness of the rewriting by showing that for every element $\mathcal{I}_i$ in the sequence, $\mathcal{I}_i \models \mathit{rew}(\phi)$ implies $\mathcal{I}_{i-1} \models \mathit{rew}(\phi)$.

A challenge with this approach is that we cannot really construct this sequence of models as classical interpretations. The reason is that the TBox may not fully specify the value of attribute-successors, which may only be restricted by predicates such as $>_d$ or $+_d$. Since classical interpretations need to assign a specific value to every attribute successor, we cannot always construct a classical interpretations that captures all query entailments. The solution to this problem is to use *abstract interpretations* instead of classical interpretations, in which concrete domain values are represented using variables, which are restricted using a set of constraints. These are introduced in the next subsection.

For convenience, we may assume in the following that Rules $\mathbf{R_{init}}$-5 and **RC2** are applied silently, and identify concepts of the form $\mathsf{Q}U_1, U_2.+_d$ with $\mathsf{Q}U_2, U_1.+_{-d}$, and atoms of the form $+_d(v_1, v_2)$ with $+_{-d}(v_2, v_1)$.

5.1.1 Abstract interpretations

Let $\mathsf{N_{dv}}$ be a countably infinite set of *domain variables* disjoint with $\mathsf{N_C}$, $\mathsf{N_R}$, $\mathsf{N_A}$, $\mathsf{N_I}$ and $\mathsf{N_V}$. A *concrete domain atom* is an expression of the form $\Pi(\vec{v})$, where $\Pi \in \mathcal{P}^{\mathbb{R}_\sim}$ is an n-ary concrete domain predicate and $\vec{v} \in (\mathsf{N_{dv}})^n$. Given a set Γ of concrete domain atoms, a *solution* is a mapping $\pi : \mathsf{N_{dv}} \to \mathbb{R}$ such that for every $\Pi(\vec{v}) \in \Gamma$, $\pi(\vec{v}) \in \Pi^{\mathbb{R}_\sim}$. A concrete domain atom α is entailed by a set Γ of concrete domain atoms, in symbols $\Gamma \models \alpha$, if every solution of Γ is also a solution of α

An abstract interpretation is now a triple $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I}, \Gamma^\mathcal{I} \rangle$, where $\Delta^\mathcal{I}$ and $\cdot^\mathcal{I}$ are defined the same as for interpretations, only that now $\cdot^\mathcal{I}$ maps attribute names $U \in \mathsf{N_A}$ to relations $U^\mathcal{I} \subseteq \Delta^\mathcal{I} \cup \mathsf{N_{dv}}$. The set $\Gamma^\mathcal{I}$ contains domain predicate atoms over the variables used in $\cdot^\mathcal{I}$. Given a solution π for $\Gamma^\mathcal{I}$, we obtain as *instance of* $\mathcal{I}$ the classical interpretation $\pi(\mathcal{I})$ in which every value $v \in \mathsf{N_{dv}}$ is replaced by $\pi(v)$. An abstract interpretation $\mathcal{I}$ is now an abstract model of a KB $\mathcal{K}$ if $\Gamma^\mathcal{I}$ has a solution and every instance of $\mathcal{I}$ is a model of $\mathcal{K}$. We then write $\mathcal{I} \models \mathcal{K}$. Similarly, for $e \in \Delta^\mathcal{I}$ and a concept C, we write $e \in C^\mathcal{I}$ if $e \in C^{\pi(\mathcal{I})}$ for every solution π of $\Gamma^\mathcal{I}$. A Boolean query ϕ is entailed by an abstract interpretation $\mathcal{I}$, in symbols $\mathcal{I} \models \phi$, if there exists a homomorphism $h : \mathbf{terms}(\phi) \to \Delta^\mathcal{I} \cup \mathsf{N_{dv}}$ s.t. for every solution π of $\mathcal{I}$ and corresponding instantiation $\pi(\mathcal{I})$ of $\mathcal{I}$, and for every atom $X(\vec{x}) \in \phi$, we have $\pi(\mathcal{I}) \models X(\vec{y})$, where $\vec{y}$ is obtained from $\vec{x}$ by replacing every constant by itself, every variable x s.t. $h(x) \in \Delta^\mathcal{I}$ by $h(x)$, and every variable x s.t. $h(x) \in \mathsf{N_{dv}}$ by $\pi(h(x))$. Given a concrete domain atom α, we further write $\mathcal{I} \models \alpha$ iff $\Gamma^\mathcal{I} \models \alpha$.

5.1.2 Canonical models

Given a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, we construct the abstract model $\mathcal{I}_\mathcal{K} = \langle \Delta_\mathcal{K}, \cdot^{\mathcal{I}_\mathcal{K}}, \Gamma_\mathcal{K} \rangle$, which may be infinite, as the limit of the sequence $(\mathcal{I}_i)_{i \geq 0}$, where $\mathcal{I}_i = \langle \Delta^{\mathcal{I}_i}, \cdot^{\mathcal{I}_i}, \Gamma_i \rangle$ for $i \geq 0$, which inductively defined in the following.

$\mathcal{I}_0$ is defined as a model of $\mathcal{A}$ by setting

- $\Delta^{\mathcal{I}_0} = \{a^{\mathcal{I}_0} \mid A(a) \in \mathcal{A}\} \cup \{a^{\mathcal{I}_0}, b^{\mathcal{I}_0} \mid r(a,b) \in \mathcal{A}\} \cup \{a^{\mathcal{I}_0} \mid U(a,d) \in \mathcal{A}\}$,
- for all $A \in \mathsf{N_C}$, $A^{\mathcal{I}_0} = \{a^{\mathcal{I}_0} \mid A(a) \in \mathcal{A}\}$,
- for all $r \in \mathsf{N_R}$, $r^{\mathcal{I}_0} = \{\langle a^{\mathcal{I}_0}, b^{\mathcal{I}_0} \rangle \mid r(a,b) \in \mathcal{A}\}$
- for all $U \in \mathsf{N_A}$, $U^{\mathcal{I}_0} = \{\langle a^{\mathcal{I}_0}, v_{a,U} \rangle \mid U(a,d) \in \mathcal{A}\}$,
- $\Gamma_0 = \{=_d(v_{a,U}) \mid (U(a,d) \in \mathcal{A}\}$

For $i > 0$, $\mathcal{I}_i$ is obtained from $\mathcal{I}_{i-1}$ don't-care non-deterministically, by picking a domain element or pair of domain elements $e \in \Delta^{\mathcal{I}_{i-1}} \cup (\Delta^{\mathcal{I}_{i-}} \times \Delta^{\mathcal{I}_{i-1}}) \cup (\Delta^{\mathcal{I}_{i-1}} \times \mathsf{N}_{\mathsf{dv}})$ and an axiom $X_1 \sqsubseteq X_2 \in \mathit{saturate}(\mathcal{T})$ s.t. $e \in (X_1^{\mathcal{I}_{i-1}} \setminus X_2^{\mathcal{I}_{i-1}})$. Intuitively, this means that e is a counter example for $\mathcal{I}_{i-1} \models X_1 \sqsubseteq X_2$.

If $X_2 = \bot$, $\mathcal{K}$ is unsatisfiable, and we stop the model construction. Otherwise, based on e and $X_1 \sqsubseteq X_2$, $\mathcal{I}_i$ is constructed as follows, where in each case, everything in $\mathcal{I}_i$ is as in $\mathcal{I}_{i-1}$ except for the changes mentioned.

CM1 If $X_2 \in \mathsf{N}_\mathsf{C} \cup \mathsf{N}_\mathsf{R}$, then $X_2^{\mathcal{I}_i} = X_2^{\mathcal{I}_{i-1}} \cup \{e\}$.

CM2 if $X_2 = r^-$, $r \in \mathsf{N}_\mathsf{R}$, and $e = \langle e_1, e_2 \rangle$ then $r^{\mathcal{I}_i} = r^{\mathcal{I}_{i-1}} \cup \{\langle e_2, e_1 \rangle\}$.

CM3 If $X_2 = \exists r$, $r \in \mathsf{N}_\mathsf{R}$, then $\Delta^{\mathcal{I}_i} = \Delta^{\mathcal{I}_{i-1}} \cup \{e'\}$ and $r^{\mathcal{I}_i} = r^{\mathcal{I}_{i-1}} \cup \{\langle e, e' \rangle\}$, where e' is fresh.

CM4 If $X_2 = \exists r^-$, $r \in \mathsf{N}_\mathsf{R}$, then $\Delta^{\mathcal{I}_i} = \Delta^{\mathcal{I}_{i-1}} \cup \{e'\}$ and $r^{\mathcal{I}_i} = r^{\mathcal{I}_{i-1}} \cup \{\langle e', e \rangle\}$, where e' is fresh.

CM5 If $X_2 = \exists U_1, \ldots, U_n.\Pi$, then $U_1^{\mathcal{I}_i} = U_1^{\mathcal{I}_{i-1}} \cup \{\langle e, v_1 \rangle\}$, …, $U_n^{\mathcal{I}_i} = U_n^{\mathcal{I}_{i-1}} \cup \{\langle e, v_n \rangle\}$, and $\Gamma_i = \Gamma_{i-1} \cup \{\Pi(v_1, \ldots, v_n)\}$, where v_1, …, $v_n \in \mathsf{N}_{\mathsf{dv}}$ are fresh.

CM6 If $X_2 = \forall U_1, \ldots, U_n.\Pi$, there exist $\langle e, v_1 \rangle \in U_1^{\mathcal{I}_{i-1}}$, …, $\langle e, v_n \rangle \in U_n^{\mathcal{I}_{i-1}}$ s.t. $\Gamma_{i-1} \not\models \Pi(v_1, \ldots, v_n)$, for which we set $\Gamma_i = \Gamma_{i-1} \cup \{\Pi(v_1, \ldots, v_n)\}$.

We further require that the above application is fair, in the sense that every applicable rule is eventually applied. $\mathcal{I}_\mathcal{K}$ is then defined as the limit of the constructed sequence.

We first show that, provided that $\mathcal{K}$ is satisfiable, then the construction succeeds and $\mathcal{I}_\mathcal{K}$ is indeed a canonical model of $\mathcal{K}$, in the sense that for every CQ ϕ, $\mathcal{I}_\mathcal{K} \models \phi$ iff $\mathcal{K} \models \phi$. This result is established in the following lemmas. If $\mathcal{K}$ is not satisfiable, then either the construction fails or $\Gamma_i \models \bot_{\mathbb{R}_\sim}(v)$ for some $v \in \mathsf{N}_{\mathsf{dv}}$.

First, we show that, if the model construction is successful, then $\mathcal{I}_\mathcal{K}$ is a model of $\mathcal{K}$.

Lemma 5.4. *Provided the model construction is successful, for every solution π of $\Gamma_\mathcal{K}$, $\pi(\mathcal{I}_\mathcal{K})$ is a model of $\mathcal{K}$.*

Proof. We show that $\mathcal{I}_\mathcal{K} \models \mathcal{K}$, that is, every solution π of $\Gamma_\mathcal{K}$ is a model of $\mathcal{K}$. Clearly, we have $\mathcal{I}_0 \models \mathcal{A}$. The construction explicitly takes care of axioms $\alpha \in \mathcal{T}$ and the application of rules is fair, so also $\mathcal{I}_\mathcal{K} \models \alpha$ for every $\alpha \in \mathcal{T}$. We obtain that $\mathcal{I}_\mathcal{K} \models \mathcal{K}$. □

Next, we show that $\mathcal{I}_\mathcal{K}$ is *canonical* in the sense that it covers all relevant common entailments of other models.

Lemma 5.5. *If $\mathcal{K}$ is satisfiable, then the model construction is successful and for every model $\mathcal{I}$ of $\mathcal{K}$, there is a solution π of $\Gamma_\mathcal{K}$ s.t. there is a homomorphism from $\pi(\mathcal{I}_\mathcal{K})$ to $\mathcal{I}$.*

Proof. Let $\mathcal{I}$ be a model of $\mathcal{K}$. We construct a homomorphism from $\mathcal{I}_\mathcal{K}$ into $\mathcal{I}$, and then construct a corresponding solution π of $\Gamma_\mathcal{K}$.

The idea is to just follow the construction steps for $\mathcal{I}_\mathcal{K}$, and map the domain elements and variables from $\mathcal{I}_\mathcal{K}$ to domain elements and constants in $\mathcal{I}$. Specifically, we show that for every $i \geq 0$, there is a homomorphism h_i from $\mathcal{I}_i$ into $\mathcal{I}$. We start with $\mathcal{I}_0$, for which we define $h_0(a^{\mathcal{I}_0}) = a^\mathcal{I}$ for every individual name a occuring in $\mathcal{K}$, and $h_0(v) = d$ for every variable v s.t. $=_d(v) \in \Gamma_0$. For $i > 0$, we distinguish the cases based on which rule has been used to construct $\mathcal{I}_i$ from $\mathcal{I}_{i-1}$. For the steps that do neither add a fresh domain element nor refer to a fresh domain variable, we set $h_i = h_{i-1}$. For the remaining steps, we distinguish the cases.

- For **CM3**, we note that $h_{i-1}(e)$ must have an r-successor e'', and we just set $h_i(e') = e''$.
- For **CM4**, we note that $h_{i-1}(e)$ must have an r-predecessor e'', and we just set $h_i(e') = e''$.
- For **CM5**, we note that for every U-attribute successor d of $h_{i-1}(e)$, there must exist some $\langle h_{i-1}(e), d\rangle \in U^{\mathcal{I}}$, and we set $h_i(v_{e,U}) = d$.

Note that the case where $X_2 = \bot$, which would terminate the canonical model construction, can never happen, since $\mathcal{I}$ is a model of $\mathcal{T}$. Thus, we already obtain that the canonical model construction must be successful. The mapping h' is now defined as the limit of the sequence h_0, h_1, ...

We define the solution π of $\Gamma_{\mathcal{K}}$ by setting $\pi(v) = h'(v)$ for all variables v occurring in $\mathcal{I}_{\mathcal{K}}$. For every $\Pi(v_1, \ldots, v_n) \in \Gamma_{\mathcal{K}}$, we must have $\langle h'(v_1), \ldots, h'(v_n)\rangle \in \Pi^{\mathbb{R}_\sim}$, and consequently, π is a solution of $\Gamma_{\mathcal{K}}$. The homomorphism h from $\pi(\mathcal{I}_{\mathcal{K}})$ to $\mathcal{I}$ is now obtained from h' by setting $h(e) = h'(e)$ for all $e \in \Delta^{\mathcal{I}_{\mathcal{K}}}$, an $h(v) = \pi(v)$ for all $v \in \mathsf{N_V}$. It is standard to verify that for every $X \in \mathsf{N_C} \cup \mathsf{N_R} \cup \mathsf{N_A}$ and every $e \in X^{\pi(\mathcal{I}_{\mathcal{K}})}$, we also have $h(e) \in X^{\mathcal{I}}$, which means that h is a homomorphism from $\pi(\mathcal{I}_{\mathcal{K}})$ into $\mathcal{I}$. □

Note that as a corollary of the previous lemmas, we observe that if $\mathcal{K}$ is unsatisfiable, the construction must fail or result in an abstract interpretation without a solution.

Corollary 5.1. *The construction fails or results in an abstract interpretation without a solution iff $\mathcal{K}$ is unsatisfiable.*

We can use this corollary to establish the following theorem, using a slight modification of the canonical model construction.

Theorem 5.3. *Satisfiability of DL-Lite$_\sqcap(\mathbb{R}_\sim)$ KBs is decidable.*

Proof. By Corollary 5.1, satisfiability of a DL-Lite$_\sqcap(\mathbb{R}_\sim)$ KB $\mathcal{K}$ can be decided by establishing whether a canonical model for $\mathcal{K}$ with a solution can be constructed. While our construction of a canonical model may not terminate, for satisfiability the following adaptation is sufficient: whenever we apply **CM3** or **CM4**, we check whether it was used before to add a domain element $e_{\exists R}$ satisfying the concept $\exists R$. If we did, we reuse that domain element. This way, we limit the number of added domain elements polynomially. Furthermore, the resulting abstract interpretation can always be unravelled to an unbounded one as it would be generated by the original construction. For **CM5** and **CM6**, we need to test the entailment of an atom α from the current Γ_i. For this, we build a system of inequalities of the predicates in Γ_i, together with an inequality representing the negation of α. We then only need to determine whether this system of inequalities has a solution, which is decidable. The same technique is used to determine whether the resulting abstract interpretation has a solution. □

Corollary 5.2. *Satisfiability of CQs w.r.t. to a TBox $\mathcal{T}$ is decidable.*

Proof. Given a CQ ϕ, we build a minimal ABox $\mathcal{A}_\phi$ satisfying all atoms in ϕ, and test the satisfiability of $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}_\phi\rangle$ using Theorem 5.3. □

The technique used in the preceding proofs to test for satisfiability of a KB cannot be used to test for entailment of queries, since the resulting abstract interpretation cannot be homomorphically embedded into every model of the KB. This is why for the following results, we need to make use of the canonical models as we defined them in the beginning of this subsection.

Lemma 5.6. *For every query ϕ, $\mathcal{K} \models \phi$ iff $\mathcal{I}_{\mathcal{K}} \models \phi$.*

Proof. Assume that for some solution π of $\Gamma_{\mathcal{K}}$, we have $\pi(\mathcal{I}_{\mathcal{K}}) \not\models q$. By Lemma 5.4, $\pi(\mathcal{I}_{\mathcal{K}})$ is a model of $\mathcal{K}$, so that also $\mathcal{K} \not\models q$. For the other direction, Assume $\mathcal{I}_{\mathcal{K}} \models \phi$. Then, for every solution π of $\Gamma_{\mathcal{K}}$, we have $\pi(\mathcal{I}_{\mathcal{K}}) \models \phi$, that is, there exists a homomorphism g_π from ϕ into $\pi(\mathcal{I}_{\mathcal{K}})$. By Lemma 5.5, for every model $\mathcal{I}$ of $\mathcal{K}$ there is a homomorphism h from one such $\pi(\mathcal{I}_{\mathcal{K}})$ to $\mathcal{I}$, which means $g_\pi \circ h$ is a homomorphism from ϕ into $\mathcal{I}$, and $\mathcal{I} \models \phi$. It follows that $\mathcal{K} \models \phi$ □

Before we finish this subsection, we give some auxiliary lemmas about canonical models that will be helpful when we show the completeness of the rewriting procedure.

We start with some general observations

Lemma 5.7. *The following properties hold true for every $i \geq 0$:*

1. *For every variable v occurring in Γ_i but not in Γ_0, there exists a unique attribute $U \in \mathsf{N_A}$ and domain element $e \in \Delta^{\mathcal{I}_i}$ s.t. $\langle e, v\rangle \in U^{\mathcal{I}}$ and $C \sqsubseteq \exists U.\Pi \in \mathit{saturate}(\mathcal{T})$, where $e \in C^{\mathcal{I}_{i-1}}$.*
2. *If a variable v occurs in multiple atoms in Γ_i, exactly one of those atoms was added due to **CM5**, while the others were added due to **CM6**.*
3. *If Γ_i contains a chain of atoms of the following form:*

$$+_{d_1}(v_0, v_1), +_{d_2}(v_1, v_2), \ldots, +_{d_n}(v_{n-1}, v_n),$$

then there exists a domain element $e \in \Delta^{\mathcal{I}_i}$ attributes U_0, $U_n \in \mathsf{N_A}$ s.t.t $\langle e, v_0\rangle \in U_0$, $\langle e, v_n\rangle \in U_n$ and $\mathit{saturate}(\mathcal{T})$ contains an axiom of the form $C \sqsubseteq \exists U_0, U_n.+_d$, where $d = \sum_{1 \leq i \leq n} d_i$ and $e \in C^{\mathcal{I}_{i-1}}$.

Proof. We prove the observations one after the other.

1. New variables are only added due to Step **CM5**, which never reuses an existing variable, and no step associates an existing variable to another attribute. If v was added due to an axiom of the form $C \sqsubseteq \exists U_1, U_2.\Pi$, due Rule $\mathbf{R_{init}}$-1, we also have $C \sqsubseteq \exists.\top_{\mathbb{R}_\sim}$,$C \sqsubseteq U_2.\top_{\mathbb{R}_\sim} \in \mathit{saturate}(\mathcal{T})$.
2. New variables are added to Γ_i due to Step **CM5**, and the only Step that reuses variables is Step **CM6**.
3. By the first observation, there exists a domain element e s.t. we can associate to every variable v_i an attribute U_i s.t. $\langle e, v_i\rangle \in U_1^{\mathcal{I}_i}$. Every binary atom $+_{d_i}(v_{i-1}, v_i)$ in the sequence must have been added due to Step **CM5** or due to Step **CM6**. It follows that for every j, $0 \leq j \leq n$, one of the following holds:
   (a) $\mathcal{T}$ contains an axiom of the form $C_j \sqsubseteq \exists U_j, U_{j+1}.+_{d_j}$, where $e \in C^{\mathcal{I}_{i-1}}$, or
   (b) $\mathcal{T}$ contains an axiom of the form $C_j \sqsubseteq \forall U_j, U_{j+1}.+_{d_j}$, where $e \in C^{\mathcal{I}_{i-1}}$, $\langle e, v_j\rangle \in (U'_j)^{\mathcal{I}_{i-1}}$ and $\langle e, v_{j+1}\rangle \in (U_{j+1})^{\mathcal{I}_{i-1}}$.

   First assume that for all j, Case 2 applies. By applying Rule $\mathbf{R}_+$-2 iteratively, to obtain

$$\bigsqcap_{C' \in \mathcal{C}} C' \sqsubseteq \forall U_1, U_n.+_d \in \mathit{saturate}(\mathcal{T}),$$

   where $C_j \in \mathcal{C}$ for all j, $0 \leq j \leq n$, $\exists U_j.\top_{\mathbb{R}_\sim} \in \mathcal{C}$ for all j, $1 \leq j \leq n-1$, and $\mathcal{C}$ contains no further concepts. By Observation 1, there exist axioms $C'_0 \sqsubseteq \exists U_0.\Pi$,

$C'_n \sqsubseteq \exists U_n.\Pi' \in \mathit{saturate}(\mathcal{T})$, where $e \in (C'_0 \sqcap C'_n)^{\mathcal{I}_{i-1}}$. We can thus use $\mathbf{R_{init}}$-4 to obtain the axiom $C \sqsubseteq \exists U_1, U_n.{+_d} \in \mathit{saturate}(\mathcal{T})$ as required.

Now assume that for some j, Case 1 is responsible for ${+_{d_j}}(v'_j, v'_{j+1})$. By Observation 2, if $j > 0$, Case 2 must hold for $j-1$, and if $j < n$, Case 2 must hold for $j+1$. By applying Rules $\mathbf{R}_+$-1 – $\mathbf{R}_+$-3 iteratively, where $\mathbf{R}_+$-2 is used to get the required premise for $\mathbf{R}_+$-1, we obtain

$$\prod_{C \in \mathcal{C}} C \sqsubseteq \exists U_1, U_n.{+_d} \in \mathit{saturate}(\mathcal{T}),$$

where for all j, $0 \leq j \leq n$, $C_j, \exists U_j.\top_{\mathbb{R}_\sim} \in \mathcal{C}$, and $\mathcal{C}$ contains no further concepts. To show that indeed $e \in C^{\mathcal{I}_{i-}}$, we note that, if successors v_1 and v_n where added in the current step, there must be an axiom $C \sqsubseteq \exists U_1, U_n.{+_d}$ s.t. $e \in C^{\mathcal{I}_{i-1}}$. Otherwise, they must have been added in previous steps, together with the other successors v_2, …, v_{n-1}, so that we obtain $e \in (\prod_{C \in \mathcal{C}} C)^{\mathcal{I}_{i-1}}$. □

Lemma 5.8. *If for some $i > 0$, $v_1, v_2 \in \mathsf{N_{dv}}$ and $d \in \mathbb{R}$, $\Gamma_i \models {+_d}(v_1, v_2)$ and for all $d' \in \mathbb{R}$, $\Gamma_i \not\models {=_{d'}}(v_1)$ and $\Gamma_i \not\models {=_{d'}}(v_2)$, then $\mathit{saturate}(\mathcal{T})$ contains an axiom of the form $C \sqsubseteq \exists U_1, U_2.{+_d}$ s.t. for some domain element $e \in \Delta^{\mathcal{I}_i}$, we have $e \in C^{\mathcal{I}_{i-1}}$, $\langle e, v_1\rangle \in U_1^{\mathcal{I}_i}$, and $\langle e, v_2\rangle \in U_2^{\mathcal{I}_i}$.*

Proof. Assume that for some $i > 0$ and $v_1, v_2 \in \mathsf{N_{dv}}$, i) $\Gamma_i \models {+_d}(v_1, v_2)$, and ii) $\Gamma_i \not\models {=_{d_1}}(v_1), {=_{d_2}}(v_2)$ for all $d_1, d_2 \in \mathbb{R}$. By ii), $\Gamma_i \models {+_d}(v_1, v_2)$ can only hold due to atoms of the form ${+_{d'}}(v'_1, v'_2) \in \Gamma_i$. Specifically, there must be a sequence (possibly of length 1), of such atoms:

$$+_{d'_0}(v'_0, v'_1), +_{d'_1}(v'_1, v'_2), \ldots, +_{d_n}(v'_n, v'_{n+1}),$$

s.t. $\sum_{0 \leq i \leq n} d_i = d$. From here, the lemma is a direct consequence of Lemma 5.7.3. □

Lemma 5.9. *Assume for some $i > 0$, $\Gamma_i \models \varpi_d(v)$. Then, one of the following holds for some $e \in \Delta$:*

1. *$U(a, d') \in \mathcal{A}$, and $a^{\mathcal{I}_0} = e$, where ${=_{d'}} \models \varpi_d$,*
2. *$C \sqsubseteq \exists U.\Pi \in \mathit{saturate}(\mathcal{T})$, $e \in C^{\mathcal{I}_{i-1}}$ and $\langle e, v\rangle \in U^{\mathcal{I}_i}$, where $\Pi \models \varpi_d$,*
3. *$U_0(a, d_1) \in \mathcal{A}$, $a^{\mathcal{I}_0} = e$, $C_1 \sqsubseteq \forall U_0, U_1.{+_{d_2}}$, $C_2 \sqsubseteq \exists U_1.\Pi \in \mathit{saturate}(\mathcal{T})$, $\langle e, v\rangle \in U_1^{\mathcal{I}_i}$ and $e \in (C_1 \sqcap C_2)^{\mathcal{I}_{i-1}}$, where ${=_{d_1+d_2}} \models \varpi_d$, or*
4. *$U_0(a, d_1) \in \mathcal{A}$, $a^{\mathcal{I}_0} = e$, $C_1 \sqsubseteq \forall U_0, U_1.{+_{d_2}}$, $C_2 \sqsubseteq \exists U_1, U_2.{+_{d_3}} \in \mathit{saturate}(\mathcal{T})$, $\langle e, v\rangle \in U_2^{\mathcal{I}_i}$ and $e \in (C_1 \sqcap C_2)^{\mathcal{I}_{i-1}}$, where ${=_{d_1+d_2+d_3}} \models \varpi_d$.*

Proof. Assume Case 1 does not hold. We have to show that then, one of the remaining cases holds. Since $\Gamma_i \models \varpi_d(v)$, there must exist some $\varpi'_{d_0}(v_0) \in \Gamma_i$, together with a (possibly empty) sequence of atoms

$$+_{d_1}(v_0, v_1), +_{d_2}(v_1, v_2), \ldots, +_{d_n}(v_{n-1}, v_n) \in \Gamma_i,$$

s.t. $\sum_{0 \leq j \leq n} d_j \in (\varpi_d)^{\mathbb{R}_\sim}$, $\langle e, v_j\rangle \in U_j^{\mathcal{I}_i}$ for all j, $0 \leq j \leq n$, $v_n = v$ and $U_n = U$. Furthermore, unless the sequence is empty, we have $\langle e, v_j\rangle \in U_j^{\mathcal{I}_{i-1}}$ for all j, $0 \leq j \leq n$, since the successors must have been added in a previous step. In case where the sequence is empty, Case 2 immediately follows, which is why we assume in the following that $\langle e, v_j\rangle \in U_j^{\mathcal{I}_{i-1}}$ for all j, $0 \leq j \leq n$.

Each atom in the above sequence of atoms was added due to Step **CM5** or Step **CM6**, and furthermore, by Lemma 5.7.2, if for some j, ${+_{d_j}}(v_{j-1}, v_j)$ was added due to **CM5**, then if $j > 1$, ${+_{d_{j-1}}}(v_{j-2}, v_{j-1})$ was added due to **CM6**, and if $j < n$, ${+_{d_{j+1}}}(v_j, v_{j+1})$ was also added due to **CM6**.

We show the case for $\varpi'_{d_0} = (=_{d_0})$. The case for the other unary predicates is analogous. For $=_{d_0}(v_0) \in \Gamma_i$, there are two possible reasons.

1. $=_{d_0}(v_0) \notin \Gamma_0$. We can assume w.l.o.g. that all other variables were added by **CM5**, (otherwise, we can simpy pick a shorter sequence of binary atoms in Γ_i. This means that by Lemma 5.7.1 that for every j, $1 \leq j \leq n$, we have an axiom $C'_j \sqsubseteq \exists U_j.\Pi_j \in \mathit{saturate}(\mathcal{T})$. In particular, this also implies that $e \in (C'_j)^{\mathcal{I}_{i-1}}$ for all j, $1 \leq j \leq n$.

   $=_{d_0}(v_0)$ must have been added due to Step **CM5**, or Step **CM6**. We argue that in both cases, we have some axiom $C_0 \sqsubseteq \exists U_0.=_{d_0} \in \mathit{saturate}(\mathcal{T})$, $e \in C_0{}^{\mathcal{I}_{i-1}}$. In case $=_{d_0}(v_0)$ was added due to **CM5**, this immediately follows. Otherwise, there exists some axiom $\mathfrak{a}_1 = C'_0 \sqsubseteq \forall U_0.=_{d_0} \in \mathit{saturate}(\mathcal{T})$ and $\langle e, v_0\rangle \in U_0^{\mathcal{I}}$. By Lemma 5.7.1, there also exists an axiom $C''_0 \sqsubseteq \exists U_0.\Pi \in \mathit{saturate}(\mathcal{T})$. Due to Rule $\mathbf{R}_{\mathbf{init}}$-1, in both cases we have $\mathfrak{a}_2 = C''_0 \sqsubseteq \exists U_0.\Pi \in \mathit{saturate}(\mathcal{T})$. Applying Rule $\mathbf{R}_{\mathbf{init}}$-3 on $\mathfrak{a}_1$ and $\mathfrak{a}_2$ gives $C_0 \sqsubseteq \exists U_0.=_{d_0} \in \mathit{saturate}(\mathcal{T})$, where in this case, $C_0 = C'_0 \sqcap C''_0$. Note that in both cases, $e \in C_0^{\mathcal{I}_{i-1}}$.

   We illustrate the sequence of rule applications that leads to an axiom as in Case 2 in the lemma. Since v_0 is shared and $=_{d_0}(v_0) \in \Gamma_i$, by Lemma 5.7.2, $+_{d_1}(v_0, v_1)$ was added by Step **CM6**, which means we have $C_1 \sqsubseteq \forall U_0, U_1.+_{d_1} \in \mathit{saturate}(\mathcal{T})$ and $e \in (C_1)^{\mathcal{I}_{i-1}}$. Applying Rule $\mathbf{R}_{\varpi}$-1 on this axiom and $C'_1 \sqsubseteq \exists U_1.\Pi_1$, we obtain $C_0 \sqcap C_1 \sqcap C'_1 \sqsubseteq \exists U_1.=_{d_0+d_1}$.

   For $+_{d_2}(v_1, v_2) \in \Gamma_i$, we have two possibilities. a) The atom was added due Step **CM6**. We then have $C_2 \sqsubseteq \forall U_1, U_2.+_{d_2}$ and $e \in C_2^{\mathcal{I}_{i-1}}$. By Lemma 5.7.1, we also have $C'_2 \sqsubseteq \exists U_2.\Pi \in \mathit{saturate}(\mathcal{T})$. Applying Rule $\mathbf{R}_{\varpi}$-1 then gives $C_0 \sqcap C_1 \sqcap C'_1 \sqcap C_2 \sqcap C'_2 \sqsubseteq \exists U_2.=_{d_0+d_1+d_2}$. b) The atom was added due Step **CM5**. We then have $\mathfrak{a}_1 = C_2 \sqsubseteq \exists U_1, U_2.+_{d_2}$ and $e \in C_2^{\mathcal{I}_i}$. Applying $\mathbf{R}_{+}$-2 on $C_1 \sqsubseteq \forall U_0, U_1.+_{d_0}$ gives $\mathfrak{a}_2 = C_1 \sqcap \exists U_0.\top_{\mathbb{R}_\sim} \sqsubseteq \forall U_1, U_1.+_0$. By Lemma 5.7.2, we have $C_3 \sqsubseteq \forall U_2, U_3.+_{d_3} \in \mathit{saturate}(\mathcal{T})$, and applying $\mathbf{R}_{+}$-2 on this axiom gives us $\mathfrak{a}_3 = C_3 \sqcap \exists U_3.\top_{\mathbb{R}_\sim} \sqsubseteq \forall U_2, U_2.+_0$. We apply $\mathbf{R}_{\mathbf{init}}$-6 on $\mathfrak{a}_1$, $\mathfrak{a}_2$ and $\mathfrak{a}_3$ to obtain an axiom of the form $C'''_3 \sqsubseteq \forall U_1, U_2.+_{d_2}$, which means we can continue as in a) to obtain an axiom of the form $C'''_3 \sqsubseteq \exists U_2.=_{d_0+d_1+d_2}$.

   We can repeat this procedure incrementally to establish $C \sqsubseteq \exists U_n.=_d$ s.t. $e \in C^{\mathcal{I}_{i-1}}$. Recall that $U_n = U$, so that this axiom is in the form as in Case 2 in the lemma.

2. $=_{d_0}(v_0) \in \Gamma_0$. Then, $U(a, d_0) \in \mathcal{A}$. We can then use the same construction as in the previous case to get a situation as in Case 3 or 4 in the Lemma. □

Lemma 5.10. *Assume that for some $i > 0$, and $v_1, v_2 \in \mathsf{N}_{\mathsf{dv}}$ and $d_1, d_2 \in \mathbb{R}$, we have $\Gamma_i \models \varpi_{d_1}(v_1), +_{d_2}(v_1, v_2)$, where $v_1 \neq v_2$, and for no $d_3 \in \mathbb{R}$, $\Gamma_i \models =_{d_3}(v_1)$. Then, $\langle e, v_1\rangle \in U_1^{\mathcal{I}_i}$, $\langle e, v_2\rangle \in U_2^{\mathcal{I}_i}$ and $e \in C^{\mathcal{I}_i}$, s.t. either $C \sqsubseteq \forall U_1.\varpi_{d'_1} \in \mathit{saturate}(\mathcal{T})$ or $C \sqsubseteq \forall U_2.\varpi_{d'_1+d_2}$, where $\varpi_{d'_1} \models \varpi_{d_1}$.*

Proof. Assume the premise of the lemma holds. Since $\Gamma_i \models \varpi_{d_1}(v_1)$ but $\Gamma_i \not\models =_{d_3}(v_1)$ for any $d_3 \in \mathbb{R}$, there must exist a sequence of the following form in Γ_i:

$$+_{d'_1}(v'_1, v'_2), \ldots +_{d'_{n-1}}(v'_{n-1}, v'_n), \varpi_{d'_n}(v'_n) \in \Gamma_i,$$

where n may be 2, $v_1 = v_1$ and $v_2 = v'_k$ for some $k \in \{2, \ldots, n\}$, $\sum_{1 \leq j \leq k} d'_i = d_2$ and $\varpi_{d_n - \sum_{1 \leq k < n} d_k} \models \varpi_{d_1}$, and for some $e \in \Delta^{\mathcal{I}_i}$ and U_j, $j \in \{1, \ldots, n\}$, $\langle e, v_j\rangle \in U'_j$. All atoms in this sequence must have been added due to Step **CM5** and **CM6**. **CM5** always adds new variables, and these expressions share a variable. Therefore, if for some $j \in \{1, \ldots, n\}$, the jth element in the sequence was added due to **CM5**, the $(j-1)$th element—if it exists—must have been added due to **CM6**, and the same holds for the $(j+1)$th element.

Before we continue, we make one further observation regarding all variables $v'_1, \ldots, v'_n$ occurring in this sequence. For all j, $1 \leq j \leq n$, we have $\Gamma_i \models \varpi_d(v'_j)$ for some $d \in \mathbb{R}$, so that Lemma 5.9

applies. Moreover, for no j we have $\Gamma_i \models =_d(v'_j)$ for any $d \in \mathbb{R}$, since otherwise the premise of the lemma would be contradicted. Inspection of the cases in Lemma 5.9 thus reveals that only one of those cases applies, namely Case 2. Thus, in the following, when we refer to Lemma 5.9, we always refer to this case.

- If $+_{d'_1}(v'_1, v'_2)$ was added due to **CM6**, we have $C_1 \sqsubseteq \forall U'_1, U'_2.+_{d'_1} \in \mathit{saturate}(\mathcal{T})$. Applying $\mathbf{R}_+$-2 on this axiom gives $C_1 \sqcap \exists U'_2.\top_{\mathbb{R}_\sim} \sqsubseteq \forall U'_1, U'_1.+_0 \in \mathit{saturate}(\mathcal{T})$. By Lemma 5.9, we further have an axiom $C_2 \sqsubseteq \exists U'_1.\varpi_{d''_1}$, where $\varpi_{d''_1}\top \models \varpi_{d_1}$. Applying $\mathbf{R}_{\mathbf{init}}$-2 on the last two mentioned axioms gives $C_1 \sqcap C_2 \sqcap \exists U'_2.\top_{\mathbb{R}_\sim} \sqsubseteq \forall U'_1.\varpi_{d'_1}$. We have $e \in (C_1 \sqcap C_2 \sqcap \exists U'_2)^{\mathcal{I}_i}$, and thus this axiom is of the form as postulated by the lemma.
- Assume $v_2 = v_n$. If $\varpi_{d'_n}(v'_n)$ was added due **CM6**, we are done. Otherwise, it was added due **CM5**, and by Lemma 5.7.2, $+_{d'_{n-1}}(v'_{n-1}, v'_n)$ was added due **CM6**, and $C_1 \sqsubseteq \forall U'_{n-1}, U'_n.+_{d'_{n-1}} \in \mathit{saturate}(\mathcal{T})$. We apply $\mathbf{R}_+$-2 to obtain $\mathfrak{a}_1 = C_1 \sqcap \exists U'_{n-1}.\top_{\mathbb{R}_\sim} \sqsubseteq \forall U'_n, U'_n.+_0 \in \mathit{saturate}(\mathcal{T})$. Since $\varpi_{d'_n}(v'_m)$ was added due **CM5**, we have $\mathfrak{a}_2 = C_2 \sqsubseteq \exists U'_n.\varpi_{d'_n} \in \mathit{saturate}(\mathcal{T})$. Applying $\mathbf{R}_{\mathbf{init}}$-2 on $\mathfrak{a}_1$ and $\mathfrak{a}_2$ gives $\mathfrak{a}_3 = C_1 \sqcap \exists U'_{n-1}.\top_{\mathbb{R}_\sim} \sqcap C_2 \sqsubseteq \forall \varpi_{d'_n}(v'_n) \in \mathit{saturate}(\mathcal{T})$. We have $e \in (C_1 \sqcap \exists U'_{n-1}.\top_{\mathbb{R}_\sim} \sqcap C_2)^{\mathcal{I}_i}$. Furthermore, $\varpi_{d'_n} \models \varpi_{d_1+d_2}$. Thus, $\mathfrak{a}_3$ is of the form as postulated by the lemma.
- Assume $v_2 = v_k$ for some $k < n$, and $+_{d'_k}(v'_k, v'_{k+1})$ was added due **CM6**. We then have $C_1 \sqsubseteq \forall U'_k, U'_{k+1}.+_{d'_k} \in \mathit{saturate}(\mathcal{T})$. Applying $\mathbf{R}_{\mathbf{init}}$-2 gives $\mathfrak{a}_1 = C_1 \sqcap \exists U'_{k+1}.\top_{\mathbb{R}_\sim} \sqsubseteq \forall U'_k, U'_k.+_0 \in \mathit{saturate}(\mathcal{T})$. Since $\Gamma_i \models \varpi_{d_1+d_2}(v'_k)$, by Lemma 5.9, we have $\mathfrak{a}_2 = C_2 \sqsubseteq \exists U'_k.\varpi_{d_3}$, where $\varpi_{d_3} \models \varpi_{d_1+d_2}$. We apply $\mathbf{R}_{\mathbf{init}}$-2 on $\mathfrak{a}_1$ and $\mathfrak{a}_2$, and obtain $C_1 \sqcap \exists U'_{k+1}.\top_{\mathbb{R}_\sim} \sqcap C_2 \sqsubseteq \forall U'_k.\varpi_{d_3}$, which is of the form as postulated by the lemma.
- Assume $v_2 = v_k$ for some $k < n$, and $+_{d'_k}(v'_k, v'_{k+1})$ was added due **CM5**. We then have that $+_{d'_{k-1}}(v'_{k-1}, v'_k)$ was added due **CM6**, which means that $C_1 \sqsubseteq \forall U'_{k-1}, U'_k.+_{d'_{k-1}} \in \mathit{saturate}(\mathcal{T})$. Applying $\mathbf{R}_+$-2 gives $C_1 \sqcap \exists U'_{k-1}.\top_{\mathbb{R}_\sim} \sqsubseteq \forall U'_k, U'_k.+_0 \in \mathit{saturate}(\mathcal{T})$. From here, we can continue as in the previous point. □

Lemma 5.11. *Assume that for some $i > 0$, distinct $v_1, v_2, v_3 \in \mathsf{N_V}$ and $d_1, d_2 \in \mathbb{R}$, $\Gamma_i \models +_{d_1}(v_1, v_2), +_{d_2}(v_2, v_3)$, while for no $d \in \mathbb{R}$, $\Gamma_i \models =_d(v_1)$. Then, for some $e \in \Delta^{\mathcal{I}_i}$, concept C and $U \in \mathsf{N_A}$, we have $e \in C^{\mathcal{I}_i}$, $\langle e, v_2\rangle \in U^{\mathcal{I}_i}$ and $C \sqsubseteq \forall U, U.+_0$.*

Proof. Assume the premises hold. Then, Γ_i must contain two atoms $+_{d'_1}(v'_1, v_2)$ and $+_{d'_2}(v_2, v'_3)$. By Lemma 5.7.2, at least one of them was added by Step **CM6**. By Lemma 5.7.1, there exists an attribute name $U \in \mathsf{N_A}$ and a domain element $d \in \Delta^{\mathcal{I}_i}$ s.t. $\langle e, v_2\rangle \in U^{\mathcal{I}_i}$. We may thus assume that either $C_1 \sqsubseteq \forall U_1, U.+_{d'_1} \in \mathit{saturate}(\mathcal{T})$ or $C_1 \sqsubseteq \forall U, U_1.+_{d'_2} \in \mathit{saturate}(\mathcal{T})$. In both cases, we can apply $\mathbf{R}_+$-2 to infer $C_1 \sqcap \exists U_1.\top_{\mathbb{R}_\sim} \sqsubseteq \forall U, U.+_0$, which is of the form as desired by the lemma. □

5.1.3 Completeness of the Rewriting

Before we prove completeness of the rewriting procedure, we need some auxiliary lemmas regarding the set of rewritten queries.

Lemma 5.12. *If for some (abstract) interpretation $\mathcal{I}$ and query ψ, $\mathcal{I} \models \psi$, then there exists a query $\psi' \in \mathit{rew}(\psi)$ with $\mathcal{I} \models \psi'$ and $\mathbf{terms}(\psi') \subseteq \mathbf{terms}(\psi)$ s.t. every variable v occurring in ψ' in a concrete domain predicate also occurs in an atom of the form $U(x, v)$.*

Proof. Assume $\mathcal{I} \models \psi$. We first show that we can transform ψ into a query in which every variable occurs in maximally one unary predicate, and every pair of variables occurs in maximally one binary domain predicate. For the latter, it suffices to observe that ψ cannot contain two

atoms $+_{d_1}(v_1, v_2)$ and $+_{d_2}(v_1, v_2)$ such that $d_1 \neq d_2$, since then the query would be unsatisfiable and we would not have $\mathcal{I} \models \psi$. If a variable v_1 occurs in two unary atoms $\Pi_1(v_1)$ and $\Pi_2(v_2)$, we note that Π_1 and Π_2 cannot contradict each other either, since otherwise we again would not have $\mathcal{I} \models \psi$. But if Π_1 and Π_2 do not contradict each other, for our choice of unary predicates it holds that either $\Pi_1 \models \Pi_2$ or $\Pi_2 \models \Pi_1$, so that we may employ Rewriting Rule **RC6** to eliminate either $\Pi_1(v_1)$ or $\Pi_2(v)$. We thus assume from here on that every variable occurs in at most one unary atom, and every pair of variables occurs in at most one binary atom.

Assume now that ψ contains a variable v for which it contains no atom of the form $U(x, v)$. We show that then, there exists a query $\psi' \in \mathit{rew}(\psi)$ with $\mathbf{terms}(\psi') \subseteq \mathbf{terms}(\psi)$ and $v \notin \mathbf{terms}(\psi')$, and s.t. for every interpretation $\mathcal{I}$, $\mathcal{I} \models \psi$ iff $\mathcal{I} \models \psi'$. By applying the argument repeatedly, we obtain the result stated in the lemma.

If v occurs only in a unary atom of the form $\Pi(v)$, this directly follows from Rewriting Rule **RC8**, which simply removes $\Pi(v)$. Since v occurs nowhere else, $\mathcal{I} \models \psi$ iff $\mathcal{I} \models \psi'$.

Assume v occurs in a binary atom of the form $+_{d_1}(v, v_1)$. (By Rewriting Rule **RC2**, we do not have to consider the case where it is of the form $+_d(v_1, v)$.) Furthermore, v must have at least one more occurrence in ψ, since otherwise, by Rule **RC10**, we obtain a query in which $+_{d_1}(v, v_1)$ is removed. We distinguish the cases based on this other occurrence.

- If v also occurs in an atom of the form $=_{d_2}(v)$, we can apply Rule **RC4** and obtain a query in which $+_{d_1}(v, v_1)$ is replaced by $=_{d_2 - d_1}(v_1)$.
- If v also occurs in an atom of the form $+_{d_2}(v_2, v)$, we can apply Rule **RC1** and obtain a query in which $+_{d_2}(v_2, v)$ is replaced by $+_{d_1 + d_2}(v_2, v_1)$.
- If v also occurs in an atom of the form $+_{d_2}(v, v_2)$, we first apply Rule **RC2**, and are then in the same situation as in the previous item.

By considering these cases, we can step-wise reduce the number of occurrences of v in binary atoms, until it either occurs only in unary atoms, or only in a single binary atom and no unary atom. In both cases, the remaining occurrence of v can be removed using respectively Rule **RC8** and **RC10**.

We can thus eliminate any occurrence of a variable that does not occur in an atom of the form $U(x, v)$. By applying this method iteratively, we obtain a query as required in the lemma. □

We are now ready to prove completeness of the rewriting.

Theorem 5.4. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a consistent KB and ϕ be a Boolean query. If $\mathcal{K} \models q$, then $\langle \emptyset, \mathcal{A} \rangle \models \mathit{rew}(\phi)$.*

Proof. Assume $\mathcal{K}$ is consistent and $\mathcal{K} \models \phi$. We show that then, $\mathcal{A} \models \mathit{rew}(\phi)$. By Lemma 5.6, $\mathcal{K} \models \phi$ implies $\mathcal{I}_\mathcal{K} \models \phi$, which means there exists some homomorphism h from ϕ into $\mathcal{I}_\mathcal{K}$. Since h can only refer to finitely many domain elements and variables in $\mathcal{I}_\mathcal{K}$, there must exist some $j > 0$ s.t. $\mathcal{I}_j \models \phi$. Since $\phi \in \mathit{rew}(\phi)$, this implies $\mathcal{I}_j \models \mathit{rew}(\phi)$ for this $j > 0$. We show that this also implies $\mathcal{I}_0 \models \mathit{rew}(\phi)$, and thus $\langle \emptyset, \mathcal{A} \rangle \models \mathit{rew}(\phi)$ by showing that for every $i > 0$, $\mathcal{I}_i \models \mathit{rew}(\phi)$ implies $\mathcal{I}_{i-1} \models \mathit{rew}(\phi)$.

Assume $i > 0$ and $\mathcal{I}_i \models \mathit{rew}(\phi)$. There then exists some $\psi \in \mathit{rew}(\phi)$ s.t. $\mathcal{I}_i \models \psi$. If $\mathcal{I}_{i-1} \models \psi$, we are done. Otherwise, let h be the homomorphism from ψ into $\mathcal{I}_i$. We distinguish the cases based on the operation performed to obtain $\mathcal{I}_i$ from $\mathcal{I}_{i-1}$. Note that in each case, $e \in X_1^{\mathcal{I}_{i-1}} \setminus X_2^{\mathcal{I}_{i-1}}$, where $X_1 \sqsubseteq X_2 \in \mathcal{T}$ and $e \in \Delta^{\mathcal{I}_{i-1}} \cup (\Delta^{\mathcal{I}_{i-}} \times \Delta^{\mathcal{I}_{i-1}}) \cup (\Delta^{\mathcal{I}_{i-1}} \times \mathsf{N}_{\mathsf{dv}})$

CM1 $X_2 \in \mathsf{N_C} \cup \mathsf{N_R}$. We then have $X_2^{\mathcal{I}_i} = X_2^{\mathcal{I}_{i-1}} \cup \{e\}$. Since $\mathcal{I}_{i-1} \not\models \psi$, ψ must contain an atom of the form $X_2(x)$, and $h(x) = e$. By Rewriting Rule **RR1**, there exists a CQ $\psi' \in \mathit{rew}(q)$ that is obtained by replacing $X_2(x)$ by $X_1(x)$. By Lemma 5.2, we obtain that $\psi \wedge X_1(x)$ is entailed by $\mathcal{I}_{i-1}$, and consequently, that ψ' is entailed by $\mathcal{I}_{i-1}$.

CM2 $X_2 = R^-$, $R \in \mathsf{N_R}$, and $e = \langle d_1, d_2 \rangle$. We then have $R^{\mathcal{I}_i} = R^{\mathcal{I}_{i-1}} \cup \{\langle d_2, d_1 \rangle\}$, and furthermore $X_1 \in \mathsf{N_R}$ or $X_1 = S^-$. Since $\mathcal{I}_{i-1} \not\models \psi$, we must have $R(x, y) \in \psi$. If $X_1 \in \mathsf{N_R}$, due to Rewriting Rule **RR1** there exists a rewriting $\psi' \in \mathit{rew}(q)$ where $R(x, y)$ is replaced by $X_1(y, x)$. If $X_1 = S^-$, there exists a rewriting $\psi' \in \mathit{rew}(q)$ where $R(x, y)$ is replaced by $S(x, y)$ (using **RR1**). In both cases, the homomorphism h from ψ into $\mathcal{I}_i$ is also a homomorphism from ψ' into $\mathcal{I}_{i-1}$.

CM3 $X_2 = \exists R$, $R \in \mathsf{N_R}$. We then have $\Delta^{\mathcal{I}_i} = \Delta^{\mathcal{I}_{i-1}} \cup \{e'\}$ and $R^{\mathcal{I}_i} = R^{\mathcal{I}_{i-1}} \cup \{\langle e, e' \rangle\}$, where e' is fresh. Since $\mathcal{I}_{i-1} \not\models \psi$, we must have $R(x, y) \in \psi$, where $h(x) = e$ and $h(y) = e'$. Since e' occurs nowhere else in $\mathcal{I}_i$, the only possible occurrence of y in ψ is in the atom $R(x, y)$, which means Rewriting Rule **RR1** can be applied to replace $R(x, y)$ based on $X_1 \sqsubseteq \exists R$. That means that there exists a query $\psi' \in \mathit{rew}(q)$ in which $R(x, y)$ has been replaced by $X_1(x)$. It follows now similar as for **CM1** from Lemma 5.2 that $\mathcal{I}_{i-1} \models \psi'$.

CM4 $X_2 = \exists R^-$, $r \in \mathsf{N_R}$. We then have $\Delta^{\mathcal{I}_i} = \Delta^{\mathcal{I}_{i-1}} \cup \{e'\}$ and $R^{\mathcal{I}_i} = R^{\mathcal{I}_{i-1}} \cup \{\langle e', e \rangle\}$, where e' is fresh. This case is similar as **CM3**.

CM5+CM6 $X_2 = \mathsf{Q}U_1, \ldots, U_n.\Pi$. This is the more challenging case. We first show that we can make some further assumptions on ψ and the homomorphism h. First, by Lemma 5.12, we can assume that every variable occurring in a concrete domain predicate also occurs in an atom of the form $U(x, y) \in \psi$. Second, we can assume that for every such variable y s.t. for no $d \in \mathbb{R}$, $\Gamma_i \not\models =_d(h(y))$, there is exactly one such atom $U(x, y)$: if there are two, we apply Rewriting Rule **RC7** to make them different. Clearly, the resulting query is still entailed by $\mathcal{I}_i$. Specifically, we have the following assumption.

* For every y occurring in a concrete domain predicate, there exists $U(x, y) \in \psi$. If $\Gamma_i \not\models =_d(h(y))$ for all $d \in \mathbb{R}$, then there exists exactly one such atom in ψ.

We show that for some $\psi' \in \mathit{rew}(\psi)$, $\mathcal{I}_{i-1} \models \psi'$. Since $\mathcal{I}_{i-1} \not\models \psi$, there must exist some atom $\alpha \in \psi$ s.t. $\mathcal{I}_{i-1} \not\models h(\alpha)$. We distinguish the cases based on the syntactical shape of α, and show that each of these atoms can be replaced using our rewriting rules, so that the resulting query is entailed by $\mathcal{I}_{i-1}$.

(a) α is of the form $U(x, y)$. If y occurs in an atom with a concrete domain predicate, we first apply one of the following cases to eliminate this occurrence. We may further assume that $U(x, y)$ is the only occurrence of y in ψ', due to a similar argument as for Condition (*). Since $\mathcal{I}_i \models U(h(x), h(y))$, we have $\langle h(x), h(y) \rangle \in U^{\mathcal{I}_{i-1}}$, and by Lemma 5.7.1, there must exist some axiom $C \sqsubseteq \exists U.\Pi \in \mathit{saturate}(\mathcal{T})$ s.t. $h(x) \in C^{\mathcal{I}_{i-1}}$. Notice that $\psi \wedge U(x, y) = \psi \wedge (\exists U.\top_{\mathbb{R}_\sim})(x)$ and $\Pi \models \top_{\mathbb{R}_\sim}$. Therefore, Rewriting Rule **RR1** applies, resulting in a query ψ' in which $U(x, y)$ is replaced by $C(x)$. Since $h(x) \in C^{\mathcal{I}_{i-1}}$, it follows from Lemma 5.2 that $\mathcal{I}_{i-1} \models C(h(x))$.

(b) α is of the form $\varpi_d(y)$. We then have $\Gamma_i \models \varpi_d(h(y))$, which means that Lemma 5.9 applies.

We distinguish between the different cases in that lemma.

i. $U(a, d') \in \mathcal{A}$, and $a^{\mathcal{I}_0} = e$, where $=_{d'} \models \varpi_d$. Then $\Gamma_0 \models \varpi_d(h(y))$, which contradicts $\mathcal{I}_{i-1} \not\models \varpi_d(h(y))$.

ii. $C \sqsubseteq \exists U.\Pi \in \mathit{saturate}(\mathcal{T})$, $e \in C^{\mathcal{I}_{i-1}}$ and $\langle e, v \rangle \in U^{\mathcal{I}_i}$, where $\Pi(v) \models \varpi_d(v)$
Note that we must have $e = h(x)$, where x occurs in ψ in the atom $U(x, y)$. If y is not shared, we can apply Rewriting Rule **RR1** to replace $U(x, y)$, $\varpi_d(y)$ with $C(x)$, and we are done. Otherwise, assume y is shared.

Otherwise, first assume $\Pi = (=_d)$. Applying Rewriting Rule **RR4** on ψ results in a query $\psi_1 \in \mathit{rew}(\phi)$ in which $U(x,y)$ is replaced by $C(x)$. We repeat this rewriting step until there are no atoms of the form $U'(x',y)$ left in the current query, so that y only occurs in concrete domain predicate atoms. By Lemma 5.12, we can eliminate these, and obtain a query in which y does not occur.

Now assume $\mathit{saturate}(\mathcal{T})$ contains no axiom of the form $C \sqsubseteq \exists U.=_{d_1}$. Lemma 5.9 thus implies that $\Gamma_i \not\models =_{d_1}(h(y))$ for all $d_1 \in \mathbb{R}$. Since we can always apply Rule **RC6**, we may assume that y is only shared by a binary predicate $+_{d_2}(y,y_2) \in \psi$. Since thus $\Gamma_i \models +_{d_1}(h(y),h(y_2))$, Lemma 5.10 applies, which means $\langle e, h(y))\rangle \in U_2^{\mathcal{I}_i}$ and $e \in C_1^{\mathcal{I}_i}$, where either i) $C_1 \sqsubseteq \forall U.\varpi_{d''} \in \mathit{saturate}(\mathcal{T})$, or ii) $C_1 \sqsubseteq \forall U_2.\varpi_{d''+d_2}$, where in both cases, $\varpi_{d''} \models \varpi_d$. If i) applies, we can apply Rewriting Rule **RR2** to replace $\varpi_d(y)$ by $C_1(x)$, $U(x)$, and we note that $\mathcal{I}_i \models C_1(h(x))$, $U(h(x))$. If ii) applies, we first apply Rewriting Rule **RC5** to replace $\varpi_d(y)$ by $\varpi_{d+d_2}(y)$, and then Rewriting Rule **RR2** to replace $\varpi_{d+d_2}(y)$ by $C_1(x)$, $U_2(x,y_2)$. In all cases, we obtain a query still entailed in which the respective concrete domain predicate has been removed.

iii. $U_0(a,d_1) \in \mathcal{A}$, $a^{\mathcal{I}_0} = e$, $C_1 \sqsubseteq \forall U_0, U_1.+_{d_2}$, $C_2 \sqsubseteq \exists U_1.\Pi \in \mathit{saturate}(\mathcal{T})$ and $e \in (C_1 \sqcap C_2)^{\mathcal{I}_{i-1}}$, where $=_{d_1+d_2} \models \varpi_d$. Note that in our case, $U_1 = U$ and $h(x) = e$. We can apply Rewriting Rule **RR6** on ψ to replace $\varpi_d(y)$ by $U_0(x,y')$, $\varpi'_{d-d_2}(y')$. We note that these atoms are entailed by $\mathcal{I}_{i-1}$ with a homomorphism obtained from h by mapping y' to d_1.

iv. $U_0(a,d_1) \in \mathcal{A}$, $a^{\mathcal{I}_0} = e$, $\mathfrak{a}_1 = C_1 \sqsubseteq \forall U_0, U_1.+_{d_2}$, $\mathfrak{a}_2 = C_2 \sqsubseteq \exists U_1, U_2.+_{d_3} \in \mathit{saturate}(\mathcal{T})$ and $e \in (C_1 \sqcap C_2)^{\mathcal{I}_i}$, where $\varpi_{d_1+d_2+d_3} \models \varpi_d$. Note that again $h(x) = e$ and $U_2 = U$.

We first apply Rule $\mathbf{R}_+$-2 on $\mathfrak{a}_1$ and obtain

$$\mathfrak{a}_3 = C_1 \sqcap \exists U_0.\top_{\mathbb{R}_\sim} \sqsubseteq \forall U_1, U_1.+_0.$$

Using Rewriting Rule **RR7** on ψ with $\mathfrak{a}_2$ and $\mathfrak{a}_3$, we obtain a query in which $U_2(x,y)$ and $\varpi_d(y)$ are replaced by $(C_1 \sqcap C_2)(x)$, $U_0(x,y_1)$, $U_1(x,y_2)$ $+_{d_3}(y_2,y)$ and $\varpi_{d-d_3}(y_1)$.

We can furthermore apply Rewriting Rule **RR6** with $\mathfrak{a}_1$ to replace $\varpi_{d-d_3}(y')$ by $C_1(x)$, $U_0(x,y_1)$ and $\varpi_{d-d_3-d_2}(y_1)$. Note that $=_{d_1} \models \varpi_{d-d_3-d_2}$, and that in complete, we replaced $\varpi_d(y_2)$ with $C_1(x)$, $C_2(x)$, $U_0(x,y_1)$, $\varpi_{d-d_3-d_2}(y_1)$, $U_1(x,y_2)$, $+_{d_3}(y_2,y)$. All but the last of these atoms are entailed in $\mathcal{I}_{i-1}$ by a homomorphism that is appropriately extended from h. The atom $+_{d_3}(y_2,y)$ can be eliminated using Lemma 5.12, or using the next case.

(c) α is of the form $+_d(y_1,y_2)$. We then have $\Gamma_i \models +_d(h(y_1),h(y_2))$.

First assume $\Gamma_i \not\models =_{d_1}(h(y_1))$, $=_{d_2}(h(y_2))$ for any values d_1 and d_2. By Condition (*), we then have that ψ contains exactly one atom $U_1(x,y_1)$ and exactly one atom $U_2(x,y_2)$. Furthermore, Lemma 5.8 applies, which means that $\mathfrak{a}_1 = C \sqsubseteq \exists U_1, U_2.+_d \in \mathit{saturate}(\mathcal{T})$ s.t. for some domain element $e \in \Delta^{\mathcal{I}_i}$, we have $e \in C^{\mathcal{I}_{i-1}}$, $\langle e, v_1\rangle \in U_1^{\mathcal{I}_i}$, and $\langle e, v_2\rangle \in U_2^{\mathcal{I}_i}$. If neither y_1 nor y_2 occur in any other concrete domain predicate, we can apply Rewriting Rule **RR5** to replace $U_1(x,y_1)$, $U_2(x,y_2)$ and $+_{d_1}(y_1,y_2)$ by $C(x)$, and we note that $\mathcal{I}_{i-1} \models C(h(x))$. Otherwise, assume y_1 does occur in another concrete domain predicate. We can eliminate all unary concrete domain predicates as in the previous cases, and thus only need to consider the case where y_1 or y_2 is shared by a binary concrete domain predicate. For y_1, this means $+_{d_2}(y_0,y_1) \in \phi$. By Lemma 5.11, we then have $\mathfrak{a}_2 = C_1 \sqsubseteq \forall U_1, U_1.+_0$, where $h(x) \in C_1^{\mathcal{I}_i}$. We can apply Rewriting Rule **RR9** with $\mathfrak{a}_1$ and $\mathfrak{a}_2$ to replace $U_2(x,y_1)$, $+_{d_2}(y_0,y_1)$ by $(C \sqcap C_1)(x)$, $U_1(x,y_3)$, $+_{d_2+d}(y_0,y_3)$, $+_d(y_1,y_3)$. $U_2(x,y_1)$ was the only attribute atom containing y_1, we can apply Lemma 5.12 to eliminate all occur-

rences of y_1. In the same way, we can proceed to remove any shared occurrences of y_2.

Now assume that $\Gamma_i \models =_{d_1}(h(y_1))$ for some $d_1 \in \mathbb{R}$. Because $\Gamma_i \models +_d(h(y_1), h(y_2))$, this also implies $\Gamma_i \models =_{d_2}(h(y_2))$, where $d_2 = d_1 + d$. Since by assumption, $\Gamma_{i-1} \not\models +_d(h(y_1), h(y_2))$, we must have one of $\Gamma_{i-1} \not\models =_{d_1}(h(y_1))$ or $\Gamma_{i-1} \not\models =_{d_2}(h(y_2))$. Without loss of generality, we assume $\Gamma_{i-1} \not\models =_{d_1}(h(y_1))$. Since $\Gamma_i \models =_{d_1}(h(y_1))$, Lemma 5.9 applies, which in our case means we are in one of the following cases:

i. $U_1(a, d_1) \in \mathcal{A}$, and $a^{\mathcal{I}_0} = e$. This case is impossible since $\Gamma_{i-1} \not\models =_{d_1}(h(y_1))$.

ii. $C \sqsubseteq \exists U_1.=_{d_1} \in \mathit{saturate}(\mathcal{T})$, $e \in C^{\mathcal{I}_i}$ and $\langle e, v\rangle \in U_1^{\mathcal{I}_i}$. We first apply **RR4** to replace $U_1(x, y_1)$ with $C(x)$, $=_{d_1}(y_1)$, and then **RC4** to replace $+_d(y_1, y_2)$ by $=_{d_2}(y_2)$. Since y_1 occurs only in $=_{d_1}(y_1)$, we can drop this atom using **RC8**, and obtain a query $\psi' \in \mathit{rew}(\phi)$ that is obtained from ψ by replacing $+_d(y_1, y_2)$ with $C(x)$, $=_{d_2}(y_2)$. If $\mathcal{I}_{i-1} \not\models =_{d_2}(h(y_2))$ we continue with this atom as above.

iii. $U_0(a, d_3) \in \mathcal{A}$, $a^{\mathcal{I}_0} = e$, $C_1 \sqsubseteq \forall U_0, U_1.+_{d_4}$, $C_2 \sqsubseteq \exists U_1.\Pi \in \mathit{saturate}(\mathcal{T})$, $e \in (C_1 \sqcap C_2)^{\mathcal{I}_i}$ and $\langle e, v\rangle \in U_1^{\mathcal{I}_1}$, where $d_1 = d_3 + d_4$. We can apply Rewriting Rule **RR8** to replace $U_1(x, y_1)$, $+_d(y_1, y_2)$ with $U_1(x, y_1)$, $C_1(x)$, $U_0(x, y_3)$, $+_{d-d_3}(y_3, y_2)$. Extend h to the homomorphism h' by setting $h'(y_3) = d_3$. We have $\mathcal{I}_{i-1} \models C_1(h'(x))$, $U_0(h'(x), h(y_3'))$, $+_{d-d_3}(y_3, y_2)$. If $U(x, y_1)$ was added in the current step of the model construction, we can eliminate it as in the previous case, after we eliminated remaining occurrences of y_1.

iv. $U_0'(a, d_2) \in \mathcal{A}$, $a^{\mathcal{I}_0} = e$, $\mathfrak{a}_1 = C_1 \sqsubseteq \forall U_0', U_1'.+_{d_3}$, $\mathfrak{a}_2 = C_2 \sqsubseteq \exists U_1', U_1.+_{d_4} \in \mathit{saturate}(\mathcal{T})$, $e \in (C_1 \sqcap C_2)^{\mathcal{I}_i}$ and $\langle e, v\rangle \in U_1^{\mathcal{I}_1}$, where $d_1 = d_2 + d_3 + d_4$. We apply Rule $\mathbf{R}_+$-2 on $\mathfrak{a}_1$ to obtain

$$\mathfrak{a}_3 = C_1 \sqcap \exists U_0'.\top_{\mathbb{R}_\sim} \sqsubseteq \forall U_1', U_1'.+_0.$$

We can now use Rewriting Rule **RR9** with $\mathfrak{a}_3$ and $\mathfrak{a}_2$ to obtain a query in which $U_1(x, y_1)$ and $+_d(y_1, y_2)$ are replaced by

$$U_0'(x, y') \wedge U_1'(x, y_3) \wedge +_{d+d_4}(y_3, y_2) \wedge +_{d_4}(y_3, y_1).$$

Extend h to a homomorphism for h' by mapping $h'(y') = d_2$ and $h'(y_3) = v$. Again, the new atoms in ψ' are entailed by $\mathcal{I}_{i-1}$.

Applying these cases exhaustively for each atom $\alpha \in \psi$ s.t. $\mathcal{I}_{i-1} \not\models h(\alpha)$, we obtain the desired query $\psi' \in \mathit{rew}(\phi)$ s.t. $\mathcal{I}_{i-1} \models \psi'$.

We showed that for all $\psi \in \mathit{rew}(\phi)$ and $i > 0$, $\mathcal{I}_i \models \psi$ implies $\mathcal{I}_{i-1} \models \psi'$ for some $\psi' \in \mathit{rew}(\phi)$. As a consequence, $\mathcal{I}_\mathcal{K} \models \phi$ implies $\mathcal{I}_0 \models \mathit{rew}(\phi)$, which in turn implies $\langle , \mathcal{A}\rangle \models \mathit{rew}(\phi)$. □

Theorem 5.5 (Soundness and completeness of rewriting)**.** *Let $\mathcal{T}$ be a saturated TBox and $\mathcal{A}$ be an ABox s.t. $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ is satisfiable. Then, for any UCQ Φ, and any answer $\mathbf{a}$ to Φ, $\mathbf{a}$ is a certain answer to Φ in $\mathcal{K}$ iff $\mathbf{a}$ is a certain answer to $\mathit{rew}(\Phi)$ in $\langle \emptyset, \mathcal{A}\rangle$.*

Proof. Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ be a consistent KB. It follows from Theorems 5.1 and 5.4, that, if Φ is a Boolean UCQ, we have $\mathcal{K} \models \Phi$ iff $\langle \emptyset, \mathcal{A}\rangle \models \mathit{rew}(\Phi)$. Now assume Φ is not Boolean, and let $\mathsf{ans}(b)$ e an answer for Φ. Inspection of the rewriting procedure reveals that $\mathit{rew}(\mathsf{ans}(()\phi)) = \mathsf{ans}(\mathit{rew}(\phi))$: we note that rules treat answer variables in the same way as constants, and no rule removes an answer variable from any CQ. □

6 Conclusion

In this report, we have considered extensions of DL-Lite with concrete domains over $\mathbb{R}$. In this setting, we have presented a query rewriting approach that handles not only unary concrete

domain predicates, but also binary ones. The key idea is that entailments that are caused by the TBox and the functionality of the cr-rewritable concrete domains employed here, are computed beforehand and are added to the TBox during saturation. Although this step can be costly, it only needs to be performed once (for all queries) and can be done "off-line", before query execution. Furthermore, we have shown that our approach yields a sound, complete, and terminating query rewriting procedure. This procedure is, in contrast to earlier approaches [2], data-independent. The latter is crucial for gaining practicality on larger data sets, and our method allows for a more goal-oriented computation of rewritings than the other approaches. In order to support this claim, we are currently working on an implementation of our approach, where the resulting UCQ is translated into a union of SQL queries which then can be answered using any RDBMS. We plan to use the rewritings not only on classical data sets, but also on probabilistic data in which concrete domain values are characterised by continuous probability distributions. OBQA for this setting has been theoretically investigated in [**?**].

As for future work, there are various directions we are considering. First, we would like to investigate the possibility of extending the concrete domains with more binary predicates, e.g. multiplication, while preserving first order rewritability of CQs. On the other hand, we would like to study query rewriting w.r.t. with other concrete domains, for instance those over strings. We believe that the rules we have introduced in this paper could serve as a basis for a more generalised procedure supporting those domains.

References

[1] Alessandro Artale, Vladislav Ryzhikov, and Roman Kontchakov. *DL-Lite* with Attributes and Datatypes. In *ECAI 2012: 20th European Conference on Artificial Intelligence (ECAI'12)*, 2012.

[2] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. Query rewriting for *DL-Lite* with n-ary concrete domains. In *Proc. of the 26th International Joint Conference on AI (IJCAI'17)*, 2017.

[3] Franz Baader and Philipp Hanschke. A Scheme for Integrating Concrete Domains into Concept Languages. In *Proc. of the 12th International Joint Conference on AI (IJCAI'91)*, 1991.

[4] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable Reasoning and Efficient Query Answering in Description Logics: The *DL-Lite* Family. *Journal of Automated Reasoning*, 2007.

[5] André Hernich, Julio Lemos, and Frank Wolter. Query Answering in *DL-Lite* with Datatypes: A Non-Uniform Approach. In *Proc. of the 31st AAAI Conference on AI (AAAI'17)*, 2017.

[6] Magdalena Ortiz. Ontology Based Query Answering: The Story So Far. In *Proc. of the 7th Alberto Mendelzon International Workshop on Foundations of Data Management (AMW'13)*, 2013.

[7] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. *Journal on Data Semantics*, 2008.

[8] Ognjen Savkovic and Diego Calvanese. Introducing Datatypes in *DL-Lite*. In *ECAI 2012 - 20th European Conference on Artificial Intelligence (ECAI'12)*, 2012.