Multi-Sorted Conjunctive Queries with Concrete and Temporal Domains

Stefan Borgwardt and Marcel Lippmann

LTCS-Report 15-20

Revised version from January 7th, 2016
Multi-Sorted Conjunctive Queries with Concrete and Temporal Domains

Stefan Borgwardt and Marcel Lippmann

Abstract

Ontology-based data access (OBDA) is becoming increasingly important for answering user queries over large datasets. In this context, mostly relatively inexpressive ontology and query languages (such as DL-Lite and conjunctive queries (CQs)) are considered, as they enjoy the property of first-order rewritability, which allows to make use of existing database techniques for optimized query execution.

In real-world datasets, one frequently encounters timestamps as well as concrete values obtained from measurements. In this paper, we extend DL-Lite and CQs by concrete domain predicates of arbitrary arity, and add a temporal dimension that allows us to reason over the evolution of the data. We correct and extend previous results about extensions of DL-Lite with concrete domains, and discuss restrictions we have to impose on our formalism in order to preserve first-order rewritability.

Contents

1 Introduction 2

2 Values, Ontologies, and Queries 3
   2.1 Concrete Domains 3
   2.2 A New Member of the DL-Lite Family 7
   2.3 Temporal Semantics 9
   2.4 Multi-sorted Conjunctive Queries 9

3 Abstract Canonical Models 12
   3.1 Initialization 13
   3.2 Completion 13
   3.3 Instantiation 15
   3.4 I_K is Canonical 17

4 Rewriting MCQs 20

5 Discussion and Future Work 27
1 Introduction

Ontology-based data access (OBDA) \cite{12,15} addresses three major requirements of users for answering queries over large datasets. Firstly, it adopts the so-called open-world assumption (OWA) which allows to describe incomplete data. In contrast to the closed-world assumption (CWA), under the OWA facts which are not present in the data are assumed to be unknown rather than false. Secondly, by means of ontologies it is possible to encode background knowledge about the data. This knowledge can be used to draw implicit conclusions about facts which are not present in the data. Finally, the use of ontologies allows the user to state queries using a vocabulary that is different from the actual relations used to describe the data. This is especially important for incorporating highly heterogeneous data from different data sources.

Recently, a lot of research has been conducted in this area to identify query languages and ontology languages that enjoy certain rewritability properties that allow to use existing database techniques for optimized query execution \cite{1,12,17,21,25}. For example, if a query language is first-order (FO) rewritable w.r.t. an ontology language, then a user query can be rewritten (using the ontology) such that the CWA can be adopted for answering the query w.r.t. the dataset. Various languages of the light-weight DL-Lite family of description logics (DLs) have the property that conjunctive queries (CQs) are FO rewritable w.r.t. ontologies formulated in them.

Real-world datasets, however, frequently contain timestamps as well as concrete data values obtained from measurements. Therefore, several authors have previously considered dialects of DL-Lite and CQs with concrete domains \cite{4,23,24,26,27} (see \cite{20} for a survey on DLs with concrete domains), and with temporal semantics \cite{2,3,10}.

In this work, we introduce a new member of this family that combines these efforts and is capable of handling concrete data values as well as temporal information. For this, we extend DL-Lite with concrete domains having predicates of arbitrary arity, and add a (linear and discrete) temporal dimension to reason about evolution and change of data. We propose a multi-sorted extension of CQs in order to access this information, and show that these queries are FO rewritable over ontologies of our new language. More precisely, in addition to concepts (e.g., CPU, Active) and roles (e.g., hasCore), representing unary and binary predicates on the application domain, respectively, concrete data values are handled by attributes like hasTemperature, which connect abstract domain elements (e.g., a CPU) to concrete values (e.g., its temperature).

However, like the standard Web Ontology Language OWL 2 \cite{22}, previous extensions of DL-Lite with concrete domains consider only unary predicates on data values, e.g., >50°C, which specifies a lower bound on the temperature. In contrast, our ontology language allows for predicates of arbitrary arity, qualified attribute restrictions of the form \exists hasTemperature.>0°C, saying that a temperature measurement exists and is within a certain range, and local attribute range constraints like CPU \subseteq \forall hasTemperature, hasTemperature.\exists, which expresses that any temperature measurements taken by independent sensors on a CPU should be equal, i.e., that hasTemperature is functional on the domain CPU. On the other hand, our temporal formalism is less expressive than previous proposals like \cite{2,3}—while our queries can explicitly refer to time points, the ontology can only express that certain concepts and roles (e.g., CPU and hasCore) are rigid, i.e., their interpretation does not change over time. However, we do not consider rigid attributes, since in our scenario attributes correspond to measurements that are refreshed at each time point.

To express queries like

\[(x, t) \leftarrow \text{CPU}(x) \land \text{hasTemperature}(x, v, t) \land \text{hasTemperature}(x, w, s) \land t \geq s \land t \leq s + 5 \land +10^\circ\text{C}(v, w),\]

which asks for CPUs \(x\) and time points \(t\) for which the temperature of \(x\) has increased by \(10^\circ\text{C}\) in a time interval of length at most \(5\), it is necessary to combine the capabilities of concrete domains
with predicates of higher arity and temporal semantics. It was already shown in [4,26,27] that for these kinds of queries the expressivity of the concrete domain has to be restricted in order to preserve FO rewritability. Apart from our FO rewritability result, we show a surprising equivalence between the property \((\text{infinitediff})\) introduced in [26,27] for this purpose, and the \emph{convexity} of the concrete domain, as defined several years earlier in [6] in the context of another DL, \(\mathcal{EL}^{++}\). This equivalence holds for unary predicates as well as for predicates of larger arity. To handle \(n\)-ary predicates with \(n > 1\), we require an additional property of the concrete domain, which, however, is satisfied by all examples of convex domains from [6]. At the same time, we point out an error in [27], namely that canonical models as claimed there need not exist (see Example 2.11); instead, we have to adopt an \emph{abstract} canonical model construction in order to show FO rewritability. Apart from correcting and extending the result of [27], we incorporate ideas from [3] to deal with the rigid symbols.

The paper is structured as follows. In the next section, we provide the fundamental notions such as concrete domains, the considered ontology and query languages, and their temporal semantics. Section 3 sets the theoretical base for FO rewritability in constructing an abstract version of a canonical model, and Section 4 gives details about the actual rewriting. We conclude with a summary and ideas for future work.

2 Values, Ontologies, and Queries

In this section, we introduce the basic notions of this paper. We recall concrete domains as a formal way of representing values in ontologies. Thereafter, we provide the formal definitions of a new member of the \(\mathcal{DL}-\text{Lite}\) family, and conclude this section with introducing the multi-sorted queries that we investigate.

2.1 Concrete Domains

Concrete domains are a well-known formalism for dealing with values in ontologies. In this paper, we make some assumptions about the structure of concrete domains, which are in accordance with most concrete domains described in the literature [19,22,27].

\textbf{Definition 2.1} (Concrete domain). A concrete domain \(\mathcal{D}\) over a signature \(\{\Pi_1/m_1, \Pi_2/m_2, \ldots\}\) of countably many predicate symbols \(\Pi_i\) with associated arities \(m_i\) consists of a set \(\Delta^\mathcal{D}\) and interpretations \(\Pi_i^\mathcal{D} \subseteq (\Delta^\mathcal{D})^{m_i}\) of the predicate symbols \(\Pi_i\), where we make the following assumptions:

- \(\mathcal{D}\) has a unary predicate \(\top\mathcal{D}\) that is interpreted as \(\Delta^\mathcal{D}\).
- \(\mathcal{D}\) has all unary predicates of the form \(\_ = d\) for \(d \in \Delta^\mathcal{D}\), whose interpretations are the singleton sets \(\{d\}\).
- \(\mathcal{D}\) has a binary predicate \(=\) that is interpreted as \(\{(d,d) \mid d \in \Delta^\mathcal{D}\}\).
- \(\Delta^\mathcal{D}\) contains a subset \(\Omega\) of countably infinitely many elements \(\omega_1, \omega_2, \ldots\), which represent \emph{untyped} constants that do not occur in the interpretation of any predicate symbols except \(\top\mathcal{D}, =_{\omega_1}, \text{ and } =\).

Concrete domains can be used to formulate constraints as follows. Let from now on \(\mathbb{N}_V\) be a countable set of variables.

\textbf{Definition 2.2} (Syntax and semantics of \(\mathcal{D}\)-formulas). A \(\mathcal{D}\)-formula \(\phi\) is a Boolean combination of \(\mathcal{D}\)-atoms of the form \(\Pi_i(v_1, \ldots, v_m)\), where \(\Pi_i\) is an \(n\)-ary predicate and \(v_1, \ldots, v_m \in \mathbb{N}_V\). The
set of variables occurring in $\phi$ is denoted by $\text{Var}(\phi)$. A $\mathcal{D}$-conjunction ($\mathcal{D}$-disjunction) is a
conjunction (disjunction) of $\mathcal{D}$-atoms.

Given a finite set $V \subseteq \mathbb{N}_V$, a variable assignment (for $V$) is a mapping $f : V \to \Delta^\mathcal{D}$. For a
$\mathcal{D}$-formula $\phi$ with $\text{Var}(\phi) \subseteq V$, the set $\text{sol}_V(\phi)$ contains all solutions for $\phi$, which are the variable assignments for $V$ under which $\phi$ is satisfied in $\mathcal{D}$ (using the standard notion of satisfaction in a relational structure). We simply write $\text{sol}(\phi)$ if $V = \text{Var}(\phi)$. A $\mathcal{D}$-formula is satisfiable if it has at least one solution.

Moreover, a $\mathcal{D}$-formula $\phi$ implies a $\mathcal{D}$-formula $\psi$ if $\text{sol}_V(\phi) \subseteq \text{sol}_V(\psi)$, where $V := \text{Var}(\phi) \cup \text{Var}(\psi)$.

A set of $\mathcal{D}$-formulas $\Gamma$ implies a $\mathcal{D}$-formula $\psi$ if $\bigwedge \Gamma$ implies $\psi$.

In the following, we allow $\mathcal{D}$-atoms to contain constants $d \in \Delta^\mathcal{D}$ in order to simplify some of the
constructions. This does not add any expressivity to $\mathcal{D}$-formulas, as constants can be rewritten using $=d$. For example, the $\mathcal{D}$-conjunction $\Pi(v_1, v_2, v_3) \land =d(v_3)$ has the same set of solutions as $\Pi(v_1, v_2, d)$.

We consider several properties of concrete domains, which are related to similar ones in [6,19,27].

Definition 2.3 (Decidable, Convex, Functional, Unary, Admissible). A concrete domain $\mathcal{D}$ is

- **decidable** if satisfiability of $\mathcal{D}$-conjunctions and implication of $\mathcal{D}$-atoms by $\mathcal{D}$-conjunctions are decidable;

- **convex** if, whenever a $\mathcal{D}$-conjunction implies a (non-empty) $\mathcal{D}$-disjunction, then it also implies one of its disjuncts;

- **functional** if, for any $m$-ary predicate $\Pi$, $d \in \Delta^\mathcal{D}$, and $i$, $1 \leq i \leq m$, the conjunction $\Pi(v_1, \ldots, v_m) \land =d(v_i)$ has at most one solution;

- **unary** if all predicates of $\mathcal{D}$ are unary (and hence $\mathcal{D}$ does not contain the binary equality predicate); and

- **admissible** if it is decidable, convex, and functional.

Note that we use the convexity notion from p-admissibility in [6], but do not require decidability in polynomial time since this property is used only for rewriting queries.

Example 2.4. The following concrete domains are p-admissible [6,7], and thus decidable and convex:

- The set $\mathbb{Q}$ of all rational numbers with the unary predicates $\top_\mathbb{Q}$, $=_q$, and $>_q$ (with the interpretation $\{x \mid x > q\}$), and binary predicates $=$ and $+q$ (with the interpretation $\{(x, y) \mid x = q + y\}$), for any $q \in \mathbb{Q}$.

- The set $\Sigma^*$ of all words over a fixed alphabet $\Sigma$ with the unary predicates $\top_\Sigma$ and $=_w$, and binary predicates $=$ and $\text{conc}_w$ (with the interpretation $\{(x, y) \mid x = w \cdot y\}$), for any $w \in \Sigma^*$.

Moreover, they are also functional, and hence admissible. In fact, there is a strong relationship between functionality and convexity, and the proofs of convexity in [7] heavily rely on the fact that these two concrete domains are functional.

We consider only a single admissible concrete domain $\mathcal{D}$ in this paper. However, one can easily combine several disjoint concrete domains such as the ones above.

Lemma 2.5. Let $\mathcal{D}_1, \ldots, \mathcal{D}_n$ be admissible concrete domains without untyped constants, but each with a binary predicate $\vDash$. Then the concrete domain $\mathcal{D}$ over the disjoint union of $\Delta^{\mathcal{D}_1}, \ldots, \Delta^{\mathcal{D}_n}, \Omega$, and with all predicates of $\mathcal{D}_1, \ldots, \mathcal{D}_n$, together with the unary predicates $\top_\mathcal{D}$ and $=_\omega$, $\omega \in \Omega$, and the extended binary predicate $\vDash$ is also admissible.
Thus, we also assume in the following that $\phi$ is a part of $\psi$. Convexity of implications between equality atoms also implies a disjunction of $\psi$ can be extended without affecting the validity of the implication that does not connect two such domains. This means that each atom is now uniquely associated to one of these domains. If a variable occurs in predicates from several of the component domains, then $\phi$ is obviously unsatisfiable (or we have guessed the wrong domains for the equality atoms). Otherwise, $\phi$ can be split into independent components for each of the component domains, and satisfiability can be checked independently. A conjunction of atoms over $\Omega$ is satisfiable if and only if no two variables connected by a (possibly empty) chain of equality predicates occur in two different $\omega$-atoms, which is decidable in polynomial time.

For implication of a $\mathcal{D}$-atom $\alpha$ by a $\mathcal{D}$-conjunction $\phi$, we can likewise assume that $\top_{\mathcal{D}}$ does not occur in the input. Furthermore, the implication is valid iff for all possible ways to assign equality atoms in $\mathcal{D}$ to the component domains we can find a component domain for $\alpha$ such that the resulting implication is valid. Hence, we can again assume that all predicates belong to a unique component domain. If $\phi$ is unsatisfiable (for example if a variable occurs in predicates from different domains), then the implication holds. Otherwise, if the variables in $\alpha$ occur in $\phi$ a predicate belonging to a different domain than the one in $\alpha$, then the implication does not hold. Finally, if all type checks have succeeded, then the implication only depends on those conjuncts in $\phi$ that match the domain of the predicate of $\alpha$. An implication between atoms over $\Omega$ can be decided by computing the congruence closure of all involved equality atoms and assigning elements of $\omega$ to some of the resulting equivalence classes, in polynomial time.

To prove convexity, consider a valid implication between a $\mathcal{D}$-conjunction $\phi$ and a $\mathcal{D}$-disjunction $\psi$. If $\phi$ is unsatisfiable, then it also implies all the disjuncts of $\psi$ individually, and hence we are done. If $\psi$ contains an atom of the form $\top_{\mathcal{D}}(v)$, then $\phi$ also implies this atom, and thus we assume in the following that $\top_{\mathcal{D}}$ does not occur in $\psi$. Further, if $\top_{\mathcal{D}}(v)$ occurs in $\phi$ and $v$ also occurs in at least one different atom in $\phi$, then we can remove the atom $\top_{\mathcal{D}}(v)$ from $\phi$ without affecting convexity. If $v$ occurs in $\phi$ only in an atom of the form $\top_{\mathcal{D}}(v)$, then the solutions of the finite disjunction $\psi$ must cover all possibilities of mapping $v$ to an untyped element $\omega \in \Omega$. Since there are infinitely many such elements and they only occur in the interpretation of the predicates $\top_{\omega}, \top_{\mathcal{D}}$, and $\top$, it must be the case that $v$ occurs in $\psi$ only in equality atoms of the form $\top_{\mathcal{D}}(v')$. But then either these equality atoms are not relevant for the validity of the implication (in which case we can remove them and the atom $\top_{\mathcal{D}}(v)$), or there must be one of them that is satisfied for infinitely many solutions that map $v$ to some element of $\Omega$. This can only be the case if this equality atom also occurs in $\phi$, and hence $\phi$ clearly implies this atom. Thus, we also assume in the following that $\top_{\mathcal{D}}$ does not occur in $\phi$.

Now, any disjunct $I(v_1, \ldots, v_m)$ where one of the variables $v_1, \ldots, v_m$ occurs in $\phi$ in a predicate from a different component domain can be removed from $\psi$ without changing the validity of the implication. Afterwards, it is easy to see that the implication can again be split up according to the membership of the predicates to the same concrete domain (where all equality atoms in $\psi$ belong to all domains, and all equality atoms in $\phi$ that cannot be uniquely assigned to a domain form their own separate conjunction), and it must be the case that one of the resulting implications is valid in the corresponding component. If this component is one of $\mathcal{D}_1, \ldots, \mathcal{D}_n$, then it follows from their admissibility and the fact that the conjunction on the left-hand side of a valid implication can be extended without affecting the validity of the implication that $\phi$ implies one of the disjuncts of $\psi$. Convexity of implications between equality atoms also follows from the convexity of the component domains (each of which contains a binary equality predicate). Otherwise, we know that a conjunction $\psi'$ using only the predicates $\top_{\omega}$ and $\top$ implies a disjunction $\psi'$ with the same property, where the former is a part of $\phi$ and the latter is a part of $\psi$. We can remove all equality atoms from $\psi'$ by identifying the variables occurring together in equality atoms. The remaining conjunction ensures that all solutions must map the
each variable $v$ occurring in $\psi'$ to a fixed element $\omega_v$ of $\Omega$. But then $\psi'$ must contain either an equality atom of the form $= (v, v)$, an atom of the form $= (v, v')$ with $\omega_v = \omega_{v'}$, or an atom of the form $= \omega_v (v)$. In each case, this atom is obviously implied by $\phi'$, and hence by $\phi$.  

In contrast to the concrete domains considered in [26,27], we consider predicates of arbitrary arity, not just unary ones. A unary concrete domain is always functional, and hence admissibility reduces to decidability and convexity in this case. However, instead of convexity, [26,27] impose a different restriction on concrete domains in order to be able to answer CQs containing unary concrete domain predicates using a first-order rewriting. Surprisingly, this condition turns out to be equivalent to convexity. We introduce here a variant suitable for arbitrary $n$-ary predicates:

\textbf{(infinitediff)} Let $\phi$ be a $\mathcal{D}$-conjunction, $\psi$ a $\mathcal{D}$-disjunction, and $V := \text{Var}(\phi) \cup \text{Var}(\psi)$. If $|\text{sol}_V(\phi)| > 1$ and $\text{sol}_V(\phi) \nsubseteq \text{sol}_V(\psi')$ for every $\mathcal{D}$-atom $\psi'$ in $\psi$, then the cardinality of $\text{sol}_V(\phi) \setminus \text{sol}_V(\psi)$ is infinite.

Actually, the condition $|\text{sol}_V(\phi)| > 1$ is missing in [26,27]. However, this weaker version of (infinitediff) still allows us to correct (and extend) the proofs from [26,27]. The presence of singleton solution sets would contradict the unmodified variant of (infinitediff), as soon as there exists a $\mathcal{D}$-atom with a disjoint solution set; but this is an unnecessary restriction, as conjunctions with only a single solution can actually be handled quite easily. [26,27] also consider two other restrictions, called \textbf{(infinite)} and \textbf{(opendomain)}, which, after careful inspection, turn out to be the special cases of (infinitediff) with $\psi = \text{false}$ and $\phi = \text{true}$, respectively.

A condition similar to (infinitediff), adapted from [26], can be found in [1]. However, the latter paper also considers only unary predicates and ignores the easy case of singleton solution sets: it requires that the difference between an arbitrary union and an arbitrary intersection of (interpretations of) predicates must be either empty or infinite. Additionally, a convexity condition is imposed on the concrete domain by requiring that the inclusion relationships between all predicates can be axiomatized by Horn rules. We now show that (infinitediff) is actually equivalent to the convexity of $\mathcal{D}$.

\textbf{Lemma 2.6.} A concrete domain $\mathcal{D}$ is convex iff it satisfies (infinitediff).

\textbf{Proof.} Consider an arbitrary $\mathcal{D}$-conjunction $\phi$ and a $\mathcal{D}$-disjunction $\psi$, and let $V := \text{Var}(\phi) \cup \text{Var}(\psi)$. 

($\Rightarrow$) Suppose that $|\text{sol}_V(\phi)| > 1$, $\text{sol}_V(\phi) \nsubseteq \text{sol}_V(\psi')$ for every atom $\psi'$ in $\psi$, and that $\text{sol}_V(\phi) \setminus \text{sol}_V(\psi)$ is finite. Let $S := \{f_1, \ldots, f_m\} := \text{sol}_V(\phi) \setminus \text{sol}_V(\psi)$ be the finitely many solutions of $\phi$ that do not satisfy $\psi$ ($m$ may be 0). Then we have

$$
\text{sol}_V(\phi) \subseteq \text{sol}_V(\psi) \cup S
$$

$$
= \text{sol}_V(\psi) \cup \bigcup_{i=1}^m \text{sol}_V \left( \bigwedge_{v \in V} \varphi = f_i(v)(v) \right)
$$

$$
= \text{sol}_V \left( \psi \lor \bigvee_{i=1}^m \left( \bigwedge_{v \in V} \varphi = f_i(v)(v) \right) \right)
$$

$$
= \text{sol}_V \left( \psi \lor \bigwedge_{(v_1, \ldots, v_m) \in V^m} \bigvee_{i=1}^m \varphi = f_i(v_i)(v_i) \right)
$$

$$
= \bigcap_{(v_1, \ldots, v_m) \in V^m} \text{sol}_V \left( \psi \lor \bigvee_{i=1}^m \varphi = f_i(v_i)(v_i) \right).
$$

Hence, $\phi$ implies $\psi \lor \bigvee_{i=1}^m \varphi = f_i(v_i)(v_i)$, for all $(v_1, \ldots, v_m) \in V^m$. Since $|\text{sol}_V(\phi)| > 1$, this disjunction cannot be the empty disjunction (false). Hence, by convexity of $\mathcal{D}$ and our assumption that $\phi$ does not imply any single atom of $\psi$, for every $(v_1, \ldots, v_m) \in V^m$ there must be an
Consider now the constant tuples $(v, \ldots, v) \in V^m$, for all $v \in V$. By the above property, we know that there is an index $i_v$ such that all solutions of $\phi$ map $v$ to $f_{i_v}(v)$. Hence, either $sol_V(\phi)$ is empty or it is a singleton set containing only the solution that maps all $v \in V$ to $f_{i_v}(v)$. This contradicts our assumption that $|sol_V(\phi)| > 1$.

(⇐) Assume that $sol_V(\phi) \subseteq sol_V(\psi)$ holds and $\psi$ contains at least one atom. If $sol_V(\phi)$ is empty, then clearly $sol_V(\phi) \subseteq sol_V(\psi')$ holds for all atoms $\psi'$ in $\psi$, and hence the claim is trivial. If $|sol_V(\phi)| = 1$, then the unique solution $f$ of $\phi$ must satisfy an atom $\psi'$ in $\psi$, and thus $sol_V(\phi) = \{ f \} \subseteq sol_V(\psi')$. Finally, consider the case that $|sol_V(\phi)| > 1$. Since $sol_V(\phi) \subseteq sol_V(\psi)$, we have $|sol_V(\phi) \setminus sol_V(\psi)| = 0$, and hence \text{(infinite}\text{diff)} implies the existence of an atom $\psi'$ of $\psi$ such that $sol_V(\phi) \subseteq sol_V(\psi')$. \hfill \square

Lemmas \ref{lem:unique_solution} and \ref{lem:unique_solution2} also hold in the absence of the equality predicate $=$. The reason that we include this predicate in our concrete domains is that we need it in the construction of the canonical model in Section \ref{sec:canonical_model}. However, as demonstrated on the examples above, it usually does not affect the admissibility of the concrete domain.

In \cite{25,26,27}, it is shown that answering conjunctive queries that can refer to concrete domain predicates is \text{co-NP}-hard in data complexity (and hence not \text{FO} rewritable) if the concrete domain is not convex, even if only unary predicates are used. Hence, this is a reasonable choice of restriction also for the case of predicates of larger arity, but there we additionally need the functionality of $D$.

2.2 A New Member of the DL-Lite Family

In the following, let $D$ be an admissible concrete domain. We introduce the logic $DL-Lite^{(HF)}(D)$, which is based on the logics $DL-Lite_{\text{core}}^{(HF)}$ and $DL-Lite_{A}$ of the DL-Lite family \cite{1,12,24}, and extends them by allowing $n$-ary concrete domain predicates $\Pi$ inside \text{qualified} attribute restrictions $\exists U_1, \ldots, U_n. \Pi$ as basic concepts, and (local) attribute range constraints of the form $B \subseteq \forall U_1, \ldots, U_n. \Pi$. In contrast, the languages in \cite{4,26,27} only allow unqualified attribute restrictions $\exists U. T_D$, and range constraints over unary predicates.

Let $N_\text{C}$, $N_\text{R}$, $N_\text{A}$, and $N_1$, be pairwise disjoint sets of \text{concept}, \text{role}, \text{attribute}, and \text{individual names}, respectively. A \text{role} is either a role name or an \text{inverse} role of the form $P^-$, where $P \in N_\text{R}$. A (\text{basic}) \text{concept} is either a concept name, an \text{existential restriction} of the form $\exists R$, where $R$ is a role, or an \text{attribute restriction} of the form $\exists \alpha_1, \ldots, \alpha_m. \Pi$, where $\alpha_1, \ldots, \alpha_m$ are attribute names or elements of $\Delta^\mathcal{D}$, and $\Pi$ is an $m$-ary predicate of $\mathcal{D}$.

An \text{interpretation} $\mathcal{I}$ consists of a non-empty \text{domain} $\Delta^\mathcal{I}$ (a.k.a. \text{object domain}) and an \text{interpretation function} $^\mathcal{I}$ that assigns

- to each individual name $a \in N_1$ an object $a^\mathcal{I} \in \Delta^\mathcal{I}$ such that for all $a, b \in N_1$ with $a \neq b$ we have $a^\mathcal{I} \neq b^\mathcal{I}$ (\text{unique name assumption (UNA)}),
- to each concept name $A \in N_\text{C}$ a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ of objects,
- to each role name $P \in N_\text{R}$ a binary relation $P^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$ between objects,
- to each attribute name $U \in N_\text{A}$ a binary relation $U^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{D}$ between objects and concrete values.

This function is extended to roles and concepts as follows:

- $(P^-)^\mathcal{I} \coloneqq \{(e_1, e_2) \mid (e_2, e_1) \in P^\mathcal{I}\}$;
- $(\exists R)^\mathcal{I} \coloneqq \{ e_1 \mid \exists e_2 : (e_1, e_2) \in R^\mathcal{I}\}$; and
A knowledge base (KB) is a finite set of

- inclusions $X_1 \subseteq X_2$,
- disjointness constraints $\text{disj}(X_1, X_2)$,
- functionality constraints $\text{funct}(R)$, and
- attribute range constraints $B \subseteq \forall \alpha_1, \ldots, \alpha_m \Pi$

where $X_1$ and $X_2$ are both either basic concepts, roles, or attribute names, $R$ is a role, $B$ is a basic concept, $\Pi$ is an $m$-ary predicate of $D$, and $\alpha_1, \ldots, \alpha_m$ are attribute names or concrete domain elements; and further any role name occurring in a functionality constraint is not allowed to occur on the right-hand side of an inclusion \[24\].

A KB is consistent if it has a model.

Note that we can simulate

- the top concept $\top$ by $\exists d. \top_d$, where $d$ is an arbitrary element of $\Delta^D$,
- global attribute range restrictions by inclusions of the form $\top \subseteq \forall \alpha_1, \ldots, \alpha_m \Pi$, and
- functionality constraints over attribute names by inclusions of the form $\top \subseteq \exists U, U.:=,$
and hence our language is an extension of those in [24,27]. In contrast to these previous proposals, we do not need the restriction that functional attribute names do not occur on the right-hand side of attribute inclusions. Since we extend DL-Lite\(^{(HF)}\) core, we can also simulate qualified existential restrictions (over non-functional roles) on the right-hand side of concept inclusions [1,24]: If \( R \) is not functional, i.e., the TBox does not contain the constraint \( \text{funct}(R) \), then \( A \sqsubseteq \exists R.B \) can be expressed by \( A \sqsubseteq \exists P B, P \sqsubseteq R, \) and \( \exists P^{-} \sqsubseteq B \), where \( P \) is a fresh role name.

### 2.3 Temporal Semantics

In this section, we recall the temporal semantics from [5,8]. We assume, as usual, that a subset of the DL symbols is designated as being rigid, i.e., their interpretation is not allowed to change over time. We use \( N_{RC} \subseteq N_{C} \) to denote the rigid concept names and \( N_{RR} \subseteq N_{R} \) to denote the rigid role names. We further call a basic concept or an inverse role rigid if all symbols appearing in it are rigid. All individual names and concrete domain predicates are implicitly viewed as rigid. Any symbol, concept, or role that is not rigid is called flexible. This in particular includes all attribute names.

We recall the notion of a temporal knowledge base [5], whose semantics is defined through infinite sequences of interpretations.

**Definition 2.7** (Temporal Semantics). A DL-LTL-structure \( \mathcal{J} = (I_{i})_{i \geq 0} \) is an infinite sequence of interpretations \( I_{i} = (\Delta_{I}, \cdot_{I}) \) over a fixed domain \( \Delta_{I} \) (constant domain assumption) such that \( \mathcal{J} \) respects the rigid names, i.e., we have \( X^{I_{i}} = X^{I_{j}} \) for all \( X \in N_{I} \cup N_{RC} \cup N_{RR} \) and all \( i, j \geq 0 \).

A temporal knowledge base (TKB) \( K = (\langle A_{i} \rangle_{0 \leq i \leq n}, T) \) is a pair consisting of a finite sequence of ABoxes \( A_{i} \), \( 0 \leq i \leq n \), and a TBox \( T \). We say that \( \mathcal{J} \) is a model of \( K \) (written \( \mathcal{J} \models K \)) if

- \( I_{i} \models A_{i} \) for all \( i, 0 \leq i \leq n \); and
- \( I_{i} \models T \) for all \( i \geq 0 \).

We call \( K \) consistent if it has a model.

Due to the fact that all DL-LTL-structures \( \mathcal{J} \) respect the rigid names, we may also write \( X^{I} \) instead of \( X^{I_{i}} \) for any \( X \in N_{I} \cup N_{RC} \cup N_{RR} \). We denote by \( N_{I}(K) \) the set of all individual names occurring in \( K \).

### 2.4 Multi-sorted Conjunctive Queries

In the following, let the set of variables \( N_{V} \) be partitioned into three sets: \( N_{OV} \) (object variables), \( N_{CV} \) (concrete domain variables, which we have already used), and \( N_{TV} \) (temporal variables). Elements of \( N_{I} \cup N_{OV} \) are called object terms, those of \( \Delta^{P} \cup N_{CV} \) are value terms, and those of \( N \cup N_{TV} \) are temporal terms.

**Definition 2.8** (MCQs). A multi-sorted CQ (MCQ) \( \phi \) is of the form \( (\vec{x}, \vec{v}, \vec{t}) \leftarrow \psi(\vec{y}, \vec{w}, \vec{s}), \) where

- \( \vec{x}, \vec{y} \) are vectors over \( N_{OV}; \)
- \( \vec{v}, \vec{w} \) are vectors over \( N_{CV}; \)
- \( \vec{t}, \vec{s} \) are vectors over \( N_{TV}; \)
- all variables occurring in \( (\vec{x}, \vec{v}, \vec{t}) \) also occur in \( (\vec{y}, \vec{w}, \vec{s}); \) and
• $\psi$ is a conjunction of atoms of the following forms, using exactly the variables in $(\vec{y}, \vec{w}, \vec{s})$:
  - $A(x, t)$ (concept atom),
  - $P(x, y, t)$ (role atom),
  - $U(x, v, t)$ (attribute atom),
  - $x = y$ (object equality atom),
  - $s \triangleright t + c$ (temporal comparison atom), or
  - $\Pi(v_1, \ldots, v_m)$ (value comparison atom),

where $A \in \mathbb{N}_C$, $P \in \mathbb{N}_R$, $U \in \mathbb{N}_A$, $x, y$ are object terms, $v, v_1, \ldots, v_m$ are value terms, $s, t$ are temporal terms, $c \in \mathbb{N}$, $\triangleright \in \{<, \leq, =, >\}$, and $\Pi$ is an $m$-ary predicate of $\mathbb{D}$.

The set of answer variables (or distinguished variables) of $\phi$, denoted by $\text{FVar}(\phi)$, contains exactly the variables occurring in $(\vec{x}, \vec{v}, \vec{t})$. The remaining variables in $(\vec{y}, \vec{w}, \vec{s})$ are called existentially quantified (or nondistinguished). As for assertions, we use $P^-(x, y, t)$ as an abbreviation for $P(y, x, t)$. An MCQ is called Boolean if it does not have any answer variables. We write $\alpha \in \phi$ to denote that $\alpha$ is an atom occurring in the MCQ $\phi$. The set $\text{terms}(\phi)$ contains all elements of $\mathbb{N}_I, \Delta^D, \mathbb{N}$, and $\mathbb{N}_V$ that occur in $\phi$, and $\Delta^D(\phi)$ denotes the set of all concrete domain values that occur in $\phi$, and we similarly define $\mathbb{N}_I(\phi), \mathbb{N}_V(\phi)$, et cetera.

A DL-LTL-structure $\mathcal{I} = (\mathcal{I}_i)_{i \geq 0}$ with $\mathcal{I}_i = (\Delta^3, \mathcal{I}_i)$ satisfies (or is a model of) a Boolean MCQ $\phi$ (written $\mathcal{I} \models \phi$) if there is a homomorphism $\pi : \text{terms}(\phi) \rightarrow \Delta^3 \cup \Delta^D \cup \mathbb{N}$ of $\phi$ into $\mathcal{I}$ such that

- $\pi$ maps all object variables into $\Delta^3$, all concrete domain variables into $\Delta^D$, and all temporal variables into $\mathbb{N}$;
- $\pi(a) = a^3$ for all $a \in \mathbb{N}_I \cap \text{terms}(\phi)$;
- $\pi(d) = d$ for all $d \in \Delta^D \cap \text{terms}(\phi)$;
- $\pi(c) = c$ for all $c \in \mathbb{N} \cap \text{terms}(\phi)$;
- $\pi(x) \in A^{\pi(\phi)}$ for all concept atoms $A(x, t) \in \phi$;
- $(\pi(x), \pi(y)) \in P^{\pi(\phi)}$ for all role atoms $P(x, y, t) \in \phi$;
- $(\pi(x), \pi(v)) \in U^{\pi(\phi)}$ for attribute atoms $U(x, v, t) \in \phi$;
- $\pi(x) = \pi(y)$ for all object equality atoms $x = y \in \phi$;
- $\pi(s) \triangleright t + c$ for all temporal comparison atoms $s \triangleright t + c \in \phi$; and
- $(\pi(v_1), \ldots, \pi(v_m)) \in \Pi^D$ for all value comparison atoms $\Pi(v_1, \ldots, v_m) \in \phi$.

A KB $\mathcal{K}$ entails a Boolean MCQ $\phi$ (written $\mathcal{K} \models \phi$) if every model $\mathcal{I}$ of $\mathcal{K}$ is also a model of $\phi$. Given a TKB $\mathcal{K} = \langle (\mathcal{A}_i)_{0 \leq i \leq n}, T \rangle$, the active object domain $\text{adom}(\mathcal{K})$ of $\mathcal{K}$ is the set of individual names occurring in $\mathcal{K}$, the active concrete domain $\text{adom}(\mathcal{K})$ of $\mathcal{K}$ is the set of concrete domain elements occurring in $\mathcal{K}$, and the active temporal domain $\text{atdom}(\mathcal{K})$ of $\mathcal{K}$ is the set of time points occurring in $\mathcal{K}$, namely $\{0, \ldots, n\}$. The active domain $\text{adom}(\mathcal{K})$ of $\mathcal{K}$ is the union of these three sets. A possible answer to an MCQ $\phi$ w.r.t. $\text{adom}(\mathcal{K})$ is a mapping $\alpha : \text{FVar}(\phi) \rightarrow \text{adom}(\mathcal{K})$ that maps all object variables into $\text{adom}(\mathcal{K})$, all concrete domain variables into $\text{adom}(\mathcal{K})$, and all temporal variables into $\text{atdom}(\mathcal{K})$. A certain answer to $\phi$ is a $\mathcal{K}$-answer tuple of the form $\langle \vec{x}, \vec{v}, \vec{t} \rangle$ where $\alpha$ is a possible answer for which $\mathcal{K}$ entails the Boolean MCQ $\alpha(\phi) : () \leftarrow \psi(\vec{y}, \vec{w}, \vec{s})$ w.r.t. $\mathcal{K}$. The set of all certain answers to $\phi$ w.r.t. $\mathcal{K}$ is denoted by $\text{cert}(\phi, \mathcal{K})$. Similarly, for a DL-LTL-structure $\mathcal{I}$, we denote by $\text{ans}_{\text{adom}(\mathcal{K})}(\phi, \mathcal{I})$ the set of all tuples $\alpha(\vec{x}, \vec{v}, \vec{t})$, where $\alpha$ is a possible answer to $\phi$ w.r.t. $\text{adom}(\mathcal{K})$ such that $\mathcal{I} \models \alpha(\phi)$. We usually omit the subscript $\text{adom}(\mathcal{K})$ since it is clear from the context.
We consider the variable assignment
which encodes the formula
yields the unary predicates
for a single canonical model
not exist a single canonical model
true.
This restriction, which is also imposed on the queries in \[27\], is used to facilitate the FO rewriting. The following result shows that in the presence of predicates of larger arity it is really necessary.

**Lemma 2.10.** Entailment of Boolean MCQs is \(\text{co-NP}\)-hard in data complexity, even for atemporal semantics.

**Proof.** Since we consider only the atemporal case, we need to find a KB \((A, \mathcal{T})\) and an MCQ \(\phi\) in which we omit all temporal arguments for simplicity. Our argument is inspired by similar \(\text{co-NP}\)-hardness proofs that are based on a reduction from satisfiability of propositional 2+2-CNF formulas \[14,15,26\]. The main insight is that non-safe MCQs can simulate non-convex concrete domain predicates via projection, and hence the arguments of \[26\] apply. In particular, the projection of the binary predicates \(\text{conc}_w\) over \(\Sigma^*\) (see Example \[2,4\]) to the first component yields the unary predicates \(\text{pref}_w\) that match all words with prefix \(w\), which were shown to be non-convex (in the presence of the unary predicates \(\Sigma^-\) and \(\varepsilon\)) in \[7\].

We use here the concrete domain over \(\{\varepsilon\}^*\) with the predicates \(\top_{(a)}, \varepsilon\) and \(\text{conc}_a\). We consider a propositional 2+2-CNF formula \(f = c_1 \land \cdots \land c_n\) over the propositional variables \(p_1, \ldots, p_m\), where each clause \(c_i\) is of the form \(p^{(1)}_i \lor p^{(2)}_i \lor \neg p^{(3)}_i \lor \neg p^{(4)}_i\) for \(p^{(j)}_i \in \{p_1, \ldots, p_m, \text{true}, \text{false}\}\). For the reduction, we abuse the notation and treat \(c_1, \ldots, c_n\) and \(p_1, \ldots, p_m, \text{true}, \text{false}\) as individual names. We further use a concept name \(A\), role names \(P, Q, N_1, N_2\), and an attribute name \(U\). We consider the TBox \(\mathcal{T} := \{A \sqsubseteq \exists U. \top_{(a)}\}\) and the ABox

\[
\mathcal{A}_f := \{A(p_1), \ldots, A(p_m), U(\text{true}, a), U(\text{false}, \varepsilon)\} \cup \{P(c_i, p^{(1)}_i), P(c_i, p^{(2)}_i), N_1(c_i, p^{(3)}_i), N_2(c_i, p^{(4)}_i) \mid 1 \leq i \leq n\},
\]

which encodes the formula \(f\). Intuitively, the truth value assignments will be given by the values of the attribute \(U\) at \(p_1, \ldots, p_m\), where \(\varepsilon\) represents false, and every other word \(a^n, n \geq 1\), indicates true.

The following Boolean MCQ checks whether there exists an unsatisfied clause:

\[
\phi := () \leftarrow P_1(x_c, x_1) \land P_2(x_c, x_2) \land N_1(x_c, x_3) \land N_2(x_c, x_4) \land U(x_1, v_1) \land U(x_2, v_2) \land U(x_3, v_3) \land U(x_4, v_4) \land \varepsilon(v_1) \land \varepsilon(v_2) \land \text{conc}_a(v_3, v'_3) \land \text{conc}_a(v_4, v'_4).
\]

Note that this query is not safe since \(v'_3\) and \(v'_4\) do not occur in any attribute atoms.

If \(f\) is satisfiable by a variable assignment \(\eta\), then we can define an atemporal interpretation \(\mathcal{I}\) by \(\Delta^\mathcal{I} := \{c_1, \ldots, c_n, p_1, \ldots, p_m, \text{true}, \text{false}\}\) and \(\mathcal{U}^\mathcal{I} := \{(p, a) \mid \eta(p) = \text{true}\} \cup \{(p, \varepsilon) \mid \eta(p) = \text{false}\}\); the interpretation of the other symbols is obvious. This interpretation clearly satisfies \(\mathcal{A}_f\) and \(\mathcal{T}\), but not \(\phi\). Conversely, assume that \(f\) is unsatisfiable, and let \(\mathcal{I}\) be any model of \(\mathcal{A}_f\) and \(\mathcal{T}\). We consider the variable assignment \(\eta\) defined by \(\eta(p) := \text{false}\) if \((p, \varepsilon) \in \mathcal{U}^\mathcal{I}\), and \(\eta(p) := \text{true}\), otherwise. Then there must exist a clause \(c_i\) such that \((p^{(1)}_i, \varepsilon), (p^{(2)}_i, \varepsilon), (p^{(3)}_i, a^{n_3}), (p^{(4)}_i, a^{n_4}) \in \mathcal{U}^\mathcal{I}\) holds for some natural numbers \(n_3 \geq 1\) and \(n_4 \geq 1\). But then \(\phi\) can be satisfied in \(\mathcal{I}\) by mapping \(v'_3\) to \(a^{n_3-1}\) and \(v'_4\) to \(a^{n_4-1}\).

Since satisfiability of 2+2-CNF formulas is \(\text{NP}\)-hard \([16]\) and neither TBox nor query depend on the input formula \(f\), our entailment problem is \(\text{co-NP}\)-hard in data complexity. \(\Box\)

We now present the promised example that, contrary to what is claimed in \[26,27\], there may not exist a single canonical model \(\mathcal{I}\) for a TBK \(\mathcal{K}\) that correctly answers all possible queries \(\phi\) over \(\mathcal{K}\) (in the usual sense that \(\text{ans}(\phi, \mathcal{I}) = \text{cert}(\phi, \mathcal{K})\)).
Example 2.11. We use the concrete domain \( D \) constructed from \( Q \) and \( \Omega \) with the predicates \( \top_D \), \( \top_Q \), and \( >_d \), \( d \in \Delta^D \). This domain is unary and convex and does not contain any predicates interpreted as singleton sets, and hence satisfies the requirements of [27]. Further consider the (atemporal) KB \( K = \{ \{ A(a) \} \} \). Let \( I \) be the canonical model for this KB from [27], or indeed any fixed model of \( K \). We show that there is an MCQ (without temporal terms) that is satisfied by \( I \), but not entailed by \( K \). Since \( I \) satisfies \( K \), we have \( (a^T, d) \in U^I \) for some \( d \in Q \). But then the (safe) query \( \exists v. U(a, v) \wedge >_{d-1}(v) \) satisfies our requirements.

Since this example is expressible in the language of [27], it shows that a canonical model as claimed in [27] cannot exist. In the following construction, we first build an abstract canonical model, and when instantiating it we take into account the predicates occurring in the query.

3 Abstract Canonical Models

In the following, we adapt the known construction of the canonical model from [23,24] to our modified definition of concrete domains and extend it to the temporal case, similar to a construction in [3]. In contrast to previous constructions, however, our canonical structures are not actual interpretations in the sense introduced in Section 2.3, but rather abstract interpretations that are allowed to use concrete domain variables in place of values.

Formally, an abstract DL-LTL-structure \( J = (I_i)_{i \geq 0} \) consisting of the abstract interpretations \( I_i = (\Delta^3, \mathcal{E}_i, (\Gamma_{e,i})_{e \in \Delta^2}) \) with the constraint sets \( \Gamma_{e,i}, e \in \Delta^2, i \geq 0 \), is defined in the same way as an ordinary one, with the exception that each attribute name \( U \) is interpreted by a binary relation \( U^J_i \subseteq \Delta^2 \times (\Delta^D \cup N_{CV}) \), and the sets \( \Gamma_{e,i} \) contain constraints on the values of the variables occurring in \( U^J_i \), i.e., \( D \)-atoms of the form \( \Pi(v_1, \ldots, v_m) \), where each term \( v_m \), \( 1 \leq o \leq m \), is either a constant from \( \Delta^D \) or there exists an attribute name \( U \) such that \( (e, v_0) \in U^J_i \). Additionally, we make the following modifications to those parts of the previous definitions that are concerned with \( D \):

• The interpretation of attribute restrictions and the satisfaction of attribute range constraints at a domain element \( e \) of \( I_i \) are lifted to variables by replacing the expression \( "(d_1, \ldots, d_m) \in \Pi_D" \) by \( \Gamma_{e,i} \) implies \( \Pi(d_1, \ldots, d_m) " \).

• \( I_i \) satisfies a disjointness constraint \( \exists\exists \neg(U_1, U_2) \) for two attribute names \( U_1, U_2 \) if for all pairs \( (e, v_1) \in U^J_1 \) and \( (e, v_2) \in U^J_2 \) it holds that \( \Gamma_{e,i} \) does not imply \( \neg(1, v_2) \); this in particular rules out the case that \( v_1 = v_2 \).

• Instead of homomorphisms, we consider (abstract) homomorphisms \( \pi \) of a Boolean MCQ \( \phi \) into \( J \), which are defined similarly as before, with the exception that concrete domain variables may also be mapped to the concrete domain variables occurring in \( J \), and the satisfaction conditions is modified as follows:

  • For an attribute atom \( U(x, v, t) \in \phi \), we require that \( (\pi(x), w) \in U^J_{\pi(v)} \) holds for some concrete domain term \( w \) for which \( =(w, \pi(v)) \) is implied by the union of all constraint sets \( \Gamma_{e,i} \) that contain variables from the set \( N_{CV} \cap \{w, \pi(v)\} \) (note that this includes the case that \( w = \pi(v) \), for example if \( v \) is a constant value).

  • For a value comparison atom \( \Pi(v_1, \ldots, v_m) \in \phi \), the atom \( \Pi(\pi(v_1), \ldots, \pi(v_m)) \) must be implied by the union of all constraint sets that contain variables from the set \( N_{CV} \cap \{\pi(v_1), \ldots, \pi(v_m)\} \).

All other notions like satisfaction of knowledge bases are defined as for DL-LTL-structures, and we use \( |=_\pi \) instead of \( |= \) to differentiate the satisfaction/entailment relations, and similarly write \( \text{ans}_\pi \) instead of \( \text{ans} \). We denote by \( \text{terms}(\Gamma_{e,i}) \) the set of all variables and constants occurring in
such a set of $\mathcal{D}$-atoms, by $\text{Var}(\mathcal{J})$ the set of all variables occurring in $\mathcal{J}$, either in the interpretation of attribute names or in the constraint sets, and similarly define $\text{Var}(\Gamma_{e,i})$ and $\text{terms}(\mathcal{J})$. This definition allows us to express restrictions on the allowed combinations of attribute values at a domain element $e$, without having to explicitly give a variable assignment that satisfies these restrictions. If an abstract DL-LTL-structure does not contain any variables, then it behaves like an ordinary DL-LTL-structure since the implication of a ground atom $\Pi(d_1,\ldots,d_m)$ by an arbitrary set of $\mathcal{D}$-formulas is equivalent to the fact that $(d_1,\ldots,d_m) \in \Pi^{\mathcal{D}}$.

For the rest of this paper, let $\mathcal{D}$ be an admissible concrete domain and $\mathcal{K} = \{(A_i)_{0 \leq i \leq n}, \mathcal{T}\}$ be a TKB formulated in $\text{DL-Lite}_{\text{core}}^{\mathcal{D}, \mathcal{F}}(\mathcal{D})$. We now describe the construction of the (abstract) canonical model $\mathcal{I}_k$ for $\mathcal{K}$, which is separated into three phases: Initialization, Completion, and Instantiation. We first construct a sequence of abstract DL-LTL-structures $\mathcal{I}_k^{(\ell)} = (I_{k,i}^{(\ell)})_{i \geq 0}$ with $I_{k,i}^{(\ell)} = (\Delta_{k,i}^{(\ell)}, \tau_{k,i}^{(\ell)}, \Pi_{k,i}^{(\ell)}, \Gamma_{k,i}^{(\ell)})$, $i \geq 0$, $\ell \geq 0$, starting from an initial DL-LTL-structure $\mathcal{I}_k^{(0)}$ that is based on the sequence of ABoxes $(A_i)_{i \geq 0}$ (see Section 3.1). After the construction of this abstract part of the model is finished, the constraint sets $\Gamma_{e,i}^{(0)}$ will help us to find a “canonical” instantiation of the variables with concrete domain values (see Section 3.3). However, in most of our proofs we will work directly with the abstract canonical model.

### 3.1 Initialization

For $X \in N_C \cup N_R \cup N_A$ and $0 \leq j \leq n$, we define $X^{A_j} := \{e \mid X(e) \in A_j\}$, and, for $i \geq 0$,

$$X^{K,i} := \begin{cases} X^{A_i} & \text{if } 0 \leq i \leq n \text{ and } X \text{ is flexible}, \\ \emptyset & \text{if } X \text{ is rigid}, \\ \bigcup_{0 \leq j \leq n} X^{A_j} & \text{otherwise}. \end{cases}$$

We now define the initial DL-LTL-structure $\mathcal{I}_k^{(0)}$ as follows, for all $i \geq 0$:

- $\Delta_{k}^{(0)} := N_i(\mathcal{K})$,
- $\tau_{k}^{(0)} := a$ and $\Gamma_{a,i}^{(0)} := \emptyset$ for all $a \in N_i(\mathcal{K})$, and
- $X_{k}^{(0)} := X^{K,i}$ for all $X \in N_C \cup N_R \cup N_A$.

### 3.2 Completion

For $\ell \geq 0$, the abstract DL-LTL-structure $\mathcal{I}_k^{(\ell+1)}$ is obtained from $\mathcal{I}_k^{(\ell)}$ by applying one of the following completion rules, and then closing the interpretation of the affected symbol(s) under the rigidity constraints; that is, if $X_{k}^{(\ell+1)} \neq X_{k}^{(\ell)}$ and $X$ is rigid, then we set $X_{k}^{(\ell+1)} := X_{k}^{(\ell+1)}$ for all $j \geq 0$. The completion rules are as follows, for each inclusion $X_1 \subseteq X_2 \in \mathcal{T}$, every $i \geq 0$, and every $e \in X_{1}^{(\ell)} \setminus X_{2}^{(\ell)}$:

- **(CR1)** If $X_2 \in N_C \cup N_R \cup N_A$, then $X_{2}^{(\ell+1)} := X_{2}^{(\ell)} \cup \{e\}$.

- **(CR2)** In the case that $X_2 = \exists P$ with $P \in N_R$, we set $\Delta_{k}^{(\ell+1)} := \Delta_{k}^{(\ell)} \cup \{e_P\}$ and $\tau_{k}^{(\ell+1)} := \tau_{k}^{(\ell)} \cup \{(e_P, e)\}$, where $e_P$ is a fresh element.

- **(CR3)** In the case that $X_2 = \exists P^-$ with $P \in N_R$, we set $\Delta_{k}^{(\ell+1)} := \Delta_{k}^{(\ell)} \cup \{e_{P^-}\}$ and $\tau_{k}^{(\ell+1)} := \tau_{k}^{(\ell)} \cup \{(e_{P^-}, e)\}$, where $e_{P^-}$ is a fresh element.
Assume now that we show below that the image of \( \mathcal{E} \) with the conditions.

\[ (E4) \]

\[ (E5) \]

The interpretation of all other symbols under \( \mathcal{T}^{(e,i)}_{K,\bar{a}} \) is the same as under \( \mathcal{T}^{(i)}_{K,\bar{a}} \), and all other sets \( \mathcal{E}^{(e,i)} \) are defined to be \( \mathcal{E}^{(i)} \). Likewise, the other time points are not affected by the completion rules (except if \( X_2 \) is rigid).

For attribute range restrictions, we need the following additional completion rule, which is similar to \( \text{(CR4)} \) but only applies if the attribute values already exist:

\[ \text{(CR5)} \]

Let \( B \subseteq \forall a_1, \ldots, a_m \Pi \in \mathcal{T} \) and \( e = (e_1, e_2) \). If there are terms \( v_o, 1 \leq o \leq m, \) with \( (e, v_o) \in \mathcal{E}^{(i)} \), and \( \Pi^{(i)} \) does not imply \( \Pi(v_1, \ldots, v_m) \), then set \( \Pi^{(e,i)} := \Pi^{(i)} \cup \{ \Pi(v_1, \ldots, v_m) \} \).

The abstract canonical model \( \mathcal{J}_K = (\mathcal{I}_{K,\bar{a}})_{i \geq 0} \) with \( \mathcal{I}_{K,\bar{a}} = (\Delta^3, \mathcal{T}^{(i)}_{K,\bar{a}}, (\Pi^{(i)})_{i \in \Delta^3}) \) is defined as the limit of this inductive procedure, i.e., it is obtained by applying the completion rules starting with \( \mathcal{I}^{(0)}_{K,\bar{a}} \) in a fair manner, meaning that each applicable completion rule is applied at some point. This is possible since the temporal dimension is countable, and the set of all symbols relevant for this construction (concept names, predicates, etc.) is finite. Note that different domain elements and time points do not share any variables (neither in the interpretations of attributes nor in the sets \( \Pi^{(i)} \)) since attribute names cannot be rigid.

This interpretation can be embedded into every model of \( K \) in the following sense.

**Lemma 3.1.** Let \( \mathcal{J} = (\mathcal{J}_i)_{i \geq 0} \) with \( \mathcal{J}_i = (\Delta^3, \mathcal{T}^{(i)}_{K,\bar{a}}, (\Pi^{(i)})_{i \in \Delta^3}) \), then there is a function \( f_\mathcal{J} : \Delta^3 \rightarrow \Delta^3 \) and a variable assignment \( f_\mathcal{J} : \text{Var}(\mathcal{J}_K) \rightarrow \Delta^3 \) such that, for all \( A \in \text{NC}, P \in \text{NR}, U \in \text{NA}, a \in \text{N}(K), e, e' \in \Delta^3, \) predicate \( \Pi, \) terms \( v_1, \ldots, v_m, \) and \( i \geq 0, \)

\[ (E1) \]
\[ (E2) \]
\[ (E3) \]
\[ (E4) \]
\[ (E5) \]

**Proof.** We construct \( f_\mathcal{J} \) and \( f_\mathcal{J} \) by induction on the construction of \( \mathcal{J}_K \). We start by setting \( f_\mathcal{J}(a) := a^3 \) for all \( a \in \text{N}(K) \) and keeping \( f_\mathcal{J} \) undefined everywhere. The conditions \( (E1)-(E5) \) are thus satisfied for the initial interpretation \( \mathcal{J}^{(0)}_K \) and the (empty) sets \( \Gamma^{(0)}_{\mathcal{J}_i} \) since each \( \mathcal{J}_i \) is a model of \( \mathcal{A}_i, 0 \leq i \leq n, \) and \( \mathcal{J} \) respects the rigid names.

Assume now that \( \mathcal{J}^{(e,i)}_K, f_\mathcal{J}, \) and \( f_\mathcal{J} \) have already been partially constructed such that \( (E1)-(E5) \) are satisfied. We consider a single application of a completion rule to \( \mathcal{J}^{(e,i)}_K \). If the rule affects a rigid symbol \( X \) by setting \( X^{(e,i)}_{K,\bar{a}} := X^{(e,i)}_{K,\bar{a}} \cup \{ e \} \) for some domain element or tuple \( e \), then we show below that the image of \( e \) under \( f_\mathcal{J} \) and \( f_\mathcal{J} \) belongs to \( X^{(i)} \). Since \( \mathcal{J} \) respects the rigid names, it must also belong to \( X^{(i)} \), for all \( j \geq 0, \) and hence by the induction hypothesis and the fact that \( \mathcal{J}^{(e,i)}_K \) already respects the rigid names, we can set \( X^{(e,i)}_{K,\bar{a}} := X^{(e,i)}_{K,\bar{a}} \) without violating the conditions.

14
Assume that the rule application was triggered by an inclusion \(X_1 \subseteq X_2 \in \mathcal{T}\) and an element \(e \in X_1^{\mathcal{T}^{(i)}} \setminus X_2^{\mathcal{T}^{(i)}}\), where \(i \geq 0\). By the induction hypothesis, we have that the image of \(e\) under \(f_\delta\) and \(f_\delta\) belongs to \(X_1^{\mathcal{T}^{(i)}}\); we show this only for the case that \(X_1 = \exists \alpha_1, \ldots, \alpha_m \Pi\), and hence \(e\) is an element of \(\Delta^{\mathcal{T}^{(i)}}\) (the other cases can be handled by similar arguments). In that case, we know that there are terms \(v_o\) with \((e, v_o) \in \alpha_o^{\mathcal{T}^{(i)}}\), \(1 \leq o \leq m\), such that \(\Gamma^{(i)}_{e, i}\) implies \(\Pi(v_1, \ldots, v_m)\). By the induction hypothesis, we know that \((f_\delta(e), f_\delta(v_o)) \in \alpha_o^{\mathcal{T}^{(i)}}\) and that \(f_\delta\) solves all atoms in \(\Gamma^{(i)}_{e, i}\). By the above implication, it also solves \(\Pi(v_1, \ldots, v_m)\), and hence \(f_\delta(e)\) satisfies \(\exists \alpha_1, \ldots, \alpha_m \Pi\) in \(\mathcal{J}_i\).

We now make a case distinction on the type of rule that was applied.

**(CR1)** Consider the case where \(X_2 \in \mathcal{N}_\mathcal{C}\), and thus \(X_1\) is a basic concept. Since \(f_\delta(e) \in X_1^{\mathcal{T}^{(i)}}\) and \(\mathcal{J}_i \models \mathcal{T}\), we have \(f_\delta(e) \in X_2^{\mathcal{T}^{(i)}}\). This means that adding \(e\) to \(X_2^{\mathcal{T}^{(i)}}\) does not violate (E2).

The cases \(X_2 \in \mathcal{N}_\mathcal{F}\) and \(X_2 \in \mathcal{N}_\mathcal{A}\) can be treated similarly.

**(CR2)** Since \(X_1\) must be a basic concept, we again know that \(f_\delta(e) \in X_1^{\mathcal{T}^{(i)}} \subseteq X_2^{\mathcal{T}^{(i)}} = (\exists P)^{\mathcal{T}^{(i)}}\). Hence, there must exist an element \(e' \in \Delta^{\mathcal{T}}\) such that \((f_\delta(e), e') \in P^{\mathcal{T}}\). We can thus define \(f_\delta(e_P) := e'\) for the fresh element \(e_P\) introduced in the rule, in order to satisfy (E3).

**(CR3)** This rule can be treated similarly.

**(CR4)** We again have \(f_\delta(e) \in X_1^{\mathcal{T}^{(i)}} \subseteq X_2^{\mathcal{T}^{(i)}} = (\exists \alpha_1, \ldots, \alpha_m \Pi)^{\mathcal{T}^{(i)}}\). This implies that there are \(d_o\) with \((f_\delta(e), d_o) \in \alpha_o^{\mathcal{T}^{(i)}}, 1 \leq o \leq m\), such that \((d_1, \ldots, d_m) \in \Pi^{\mathcal{D}}\). We define \(f_\delta(v_o) := d_o\) for all the fresh variables \(v_o\) introduced by the completion rule, and hence (E4) and (E5) remain satisfied.

**(CR5)** We have \((f_\delta(e_1), f_\delta(e_2)) \in X_1^{\mathcal{T}^{(i)}} \subseteq X_2^{\mathcal{T}^{(i)}} = (P^-)^{\mathcal{T}^{(i)}}\), where \(e = (e_1, e_2)\). This implies \((f_\delta(e_2), f_\delta(e_1)) \in P^{\mathcal{T}^{(i)}}\), and thus \((e_2, e_1)\) can be added to \(P^{\mathcal{T}^{(i)}}\) without violating (E3).

**(CR6)** If this rule was applied to an attribute range constraint \(B \subseteq \forall \alpha_1, \ldots, \alpha_m \Pi \in \mathcal{T}\), \(e \in \Delta^{\mathcal{T}^{(i)}}\), and terms \(v_1, \ldots, v_m\), then we know as above that \(f_\delta(e) \in B^{\mathcal{T}^{(i)}}\). By the induction hypothesis, we also have \((f_\delta(e), f_\delta(v_o)) \in \alpha_o^{\mathcal{T}^{(i)}}, 1 \leq o \leq m\). Since \(\mathcal{J}_i \models \mathcal{T}\), we obtain \((f_\delta(v_1), \ldots, f_\delta(v_m)) \in \Pi^{\mathcal{D}}\). Hence, we can add \(\Pi(v_1, \ldots, v_m)\) to \(\Gamma^{(i)}_{e, i}\) without violating (E5).

In the following, we call a mapping \(f_\delta : \text{Var}(\mathcal{J}_\mathcal{K}) \to \Delta^{\mathcal{D}}\) having property (E5) an instantiation of \(\mathcal{J}_\mathcal{K}\). Given such an instantiation \(f_\delta\), we denote by \(f(\mathcal{J}_\mathcal{K})\) the DL-LTL-structure resulting from replacing all variables in \(\mathcal{J}_\mathcal{K}\) according to \(f_\delta\) and discarding the constraint sets.

### 3.3 Instantiation

In some of the results, we need to work with an actual interpretation, which is obtained from \(\mathcal{J}_\mathcal{K}\) by a “canonical” instantiation. As shown in Example 2.11 this instantiation needs to take into account the query. Let hence \(\phi\) be an MCQ over the signature of \(\mathcal{K}\). We specify a set \(\mathfrak{K}\) of concrete domain predicates, which intuitively describe constellations of concrete domain values that should be avoided in the following construction (if possible).

We first define the set \(\mathfrak{K}_\mathcal{K}\) of all relevant concrete domain predicates in \(\mathcal{K}\) as

\[
\mathfrak{K}_\mathcal{K} := \{ \Pi \mid \exists \alpha_1, \ldots, \alpha_m \Pi \text{ or } \forall \alpha_1, \ldots, \alpha_m \Pi \text{ occurs in } \mathcal{T}\} \cup \{\forall d \mid d \in \Delta^{\mathcal{D}}(\mathcal{K})\}.
\]

Similarly, we define

\[
\mathfrak{K}_\phi := \{ \Pi \mid \Pi(v_1, \ldots, v_m) \text{ occurs in } \phi\} \cup \{\forall d \mid d \in \Delta^{\mathcal{D}}(\phi)\}.
\]
In addition, we consider the following predicates:

\[ \mathcal{R}_\Gamma := \{ =_{c} \mid =_{d}(v) \text{ is implied by a satisfiable } \Gamma, c \in \Delta^3_{\kappa}, i \geq 0, v \in \mathcal{N}_{CV} \}. \]

To see that this set is finite, observe that the number of concrete domain terms in each set \( \Gamma \) is bounded by the size of \( \mathcal{T} \), and that the number of predicates occurring in these sets is bounded by \( |\mathcal{R}_\kappa| \). Hence, modulo renaming of variables, there are only finitely many possible sets \( \Gamma \) of \( \mathcal{D} \)-atoms over a given number of variables and the given predicates and constants. Furthermore, each satisfiable set of this form can imply at most one \( \mathcal{D} \)-atom of the form \( =_{d}(v) \) for each of its variables. We further consider the predicates in

\[ \mathcal{R}_{\Gamma,0} := \{ =_{d} \mid =_{d}(v) \text{ is implied by a satisfiable } \Pi(v_1, \ldots, v_m) \wedge =_{d'}(v'), \Pi \in \mathcal{R}_\kappa \cup \mathcal{R}_\phi, \]

\[ v_1, \ldots, v_m \in \mathcal{N}_{CV} \cup \Delta^3_{\kappa} \cup \Delta^3_{\phi}, v' \in \{ v_1, \ldots, v_m \}, =_{d'} \in \mathcal{R}_{\Gamma,0} \}, \]

which is also finite by similar arguments as above and due to the fact that \( \mathcal{D} \) is functional. Finally, we set \( \mathcal{R} := \mathcal{R}_\kappa \cup \mathcal{R}_\phi \cup \mathcal{R}_{\Gamma,0} \cup \mathcal{R}_{\Gamma,1} \cup \{ = \}. \)

We now fix an arbitrary enumeration \((e_1, i_1), (e_2, i_2), \ldots \) of \( \Delta^3_{\kappa} \times \mathbb{N} \), and define \( \Gamma_j := \Gamma_{e_j, i_j} \).

We iteratively build a partial mapping \( f_{\phi}^{(j)} : \mathcal{N}_{CV} \rightarrow \Delta^3_{\phi}, j \geq 0 \), that specifies how to replace the variables in \( \Gamma_1, \ldots, \Gamma_j \) by actual values. Initially, \( f_{\phi}^{(0)} \) is undefined everywhere. We now assume that we have already constructed a partial mapping \( f_{\phi}^{(j)}, j \geq 0 \), and try to find a valuation for the variables in the next constraint set \( \Gamma_{j+1} \). We construct the following sets of positive and negative constraints, respectively:

\[ \text{Pos}_{j+1} := \{ \Pi(v_1, \ldots, v_m) \mid \Pi \in \mathcal{R}, v_1, \ldots, v_m \in \text{terms}(\Gamma_{j+1}) \cup \Delta^3_{\kappa} \cup \Delta^3_{\phi} \cup f_{\phi}^{(j)}(\mathcal{N}_{CV}) \} \cup \{ =_{d}(v) \mid d \in f_{\phi}^{(j)}(\mathcal{N}_{CV}), v \in \text{terms}(\Gamma_{j+1}) \}. \]

\[ \text{Neg}_{j+1} := \{ \Pi(v_1, \ldots, v_m) \mid \Pi \in \mathcal{R}, v_1, \ldots, v_m \in \text{terms}(\Gamma_{j+1}) \cup \Delta^3_{\kappa} \cup \Delta^3_{\phi} \cup f_{\phi}^{(j)}(\mathcal{N}_{CV}) \} \cup \{ =_{d}(v) \mid d \in f_{\phi}^{(j)}(\mathcal{N}_{CV}), v \in \text{terms}(\Gamma_{j+1}) \}. \]

Now we can try to solve the positive constraints while respecting the negative constraints. We construct the restricted set

\[ \text{Neg}^{-}_{j+1} := \{ \Pi(v_1, \ldots, v_m) \in \text{Neg}_{j+1} \mid \text{Pos}_{j+1} \text{ does not imply } \Pi(v_1, \ldots, v_m) \}. \]

Both \( \text{Pos}_{j+1} \) and \( \text{Neg}^{-}_{j+1} \) are finite since \( \Gamma_{j+1} \) is finite and only finitely many values of \( f_{\phi}^{(j)} \) have already been defined. We now consider the set

\[ \text{sol}_{j+1} := \text{sol}(\bigwedge \text{Pos}_{j+1}) \setminus \text{sol}(\bigvee \text{Neg}^{-}_{j+1}). \]

Due to \textit{(infdiff)} there are only three options for the cardinality of this set of solutions:

- If \( |\text{sol}_{j+1}| = 1 \), then we have no choice but to replace the variables of \( \Gamma_{j+1} \) according to the single variable assignment \( f \in \text{sol}_{j+1} \), i.e., we set \( f^{(j+1)}_{\phi}(v) := f(v) \) for all \( v \in \text{Var}(\text{Pos}_{j+1}) \).
  
  Note that any variable \( v \) that already had a value under \( f_{\phi}^{(j)} \) is restricted by the atom \( =_{d}(v) \) in \( \text{Pos}_{j+1} \) with \( d = f_{\phi}^{(j)}(v) \), and hence we have \( f^{(j+1)}_{\phi}(v) = f_{\phi}^{(j)}(v) \).

- If \( \text{sol}_{j+1} \) is empty, then we can choose arbitrary concrete domain elements to replace the variables since this indicates that \( \mathcal{K} \) is inconsistent. Due to our construction of \( \text{Neg}^{-}_{j+1} \), this can only be the case if \( \text{Pos}_{j+1} \) is already unsatisfiable.

- Otherwise, \( \text{sol}_{j+1} \) must contain infinitely many elements. We choose one such element and obtain \( f_{\phi}^{(j+1)} \) as in the first case.
The variable assignment resulting from this infinite construction is denoted by $f_\phi$. This construction ensures that all necessary concrete domain restrictions are satisfied and that no unnecessary ones from $\mathcal{R}$ are satisfied.

**Lemma 3.2.** If there is an instantiation of $\mathcal{I}_K$, then all sets $\text{sol}_j$ considered for the construction of $f_\phi$ are non-empty, and hence $f_\phi$ is also an instantiation of $\mathcal{I}_K$.

**Proof.** Let $f$ be an instantiation of $\mathcal{I}_K$. We show the claim by induction on the construction of $f_\phi^{(j)}$, $j \geq 0$. Assume that it holds for some $j > 0$, and consider $f_\phi^{(j+1)}$. Since different sets $\Gamma_j$ do not share variables and $f$ solves $\Gamma_j+1$ by setting $f'(v) := f_\phi^{(j)}(v)$ for all variables $v$ for which $f_\phi^{(j)}(v)$ is defined, and $f'(v) := f(v)$ for all $v \in N_F(\Gamma_j+1)$. Since we excluded in $\text{Neg}_{j+1}$ all atoms from $\text{Neg}_{j+1}$ that are implied by $\text{Pos}_{j+1}$, by (infiniteiff) we know that $\text{sol}_{j+1}$ cannot be empty: if there is only the one solution $f'$, then it cannot be contained in $\text{sol}_t(\bigvee \text{Neg}_{j+1})$, and if there is more than one solution of $\text{Pos}_{j+1}$, there must even be infinitely many that are not also solutions of $\bigvee \text{Neg}_{j+1}$.

We need another technical result that allows us to replace $\text{Pos}_j$ by the constraint sets $\Gamma_1, \ldots, \Gamma_j$ in some circumstances.

**Lemma 3.3.** Assume that there is an instantiation of $\mathcal{I}_K$ and let $v_1, \ldots, v_m \in \text{terms}(\mathcal{I}_K) \cup \Delta^D(\phi)$ and $\Pi \in \mathcal{R}_f$. If $\Pi(v_1, \ldots, v_m)$ is satisfied by $f_\phi^{(j)}$, $j \geq 0$, then this atom is implied by $\Gamma_1 \cup \cdots \cup \Gamma_j$.

**Proof.** From the assumptions it immediately follows that the atom is contained in $\text{Neg}_j$, and thus implied by $\text{Pos}_j$. We now show the claim by induction on $j$. If $j = 0$, we know that $v_1, \ldots, v_m$ must be constants occurring in $\mathcal{K}$, and hence the claim is trivial. Let now $j > 0$. If the atom does not contain any variables from $\Gamma_j$, then it is also contained in $\text{Neg}_{j-1}$, and the claim follows by the induction hypothesis. If this atom contains only variables from $\Gamma_j$, then it is also implied by $\Gamma_j$, which again yields the claim.

It remains to consider the case that $\Pi(v_1, \ldots, v_m)$ contains both variables from $\Gamma_j$ as well as variables from some constraint sets that were considered earlier in the construction. Let $d_i := f_\phi^{(j)}(v_i)$, $1 \leq i \leq m$, and $v_{j_1}, \ldots, v_{j_n}$ be the variables among $v_1, \ldots, v_m$ that do not occur in $\Gamma_j$. Then $\Gamma_j \wedge \bigwedge_{i=1}^n =_{d_i} (v_{j_i})$ implies $\Pi(v_1, \ldots, v_m) \wedge \bigwedge_{i=1}^n =_{d_i} (v_{j_i})$. By the functionality of $D$ and Lemma 3.2, we know that the former conjunction has exactly one solution for the variables $v_1, \ldots, v_m$, and hence it implies at least one atom of the form $=_{d_i} (v_i)$, $1 \leq \ell \leq m$, where $v_i \in \text{Var}(\Gamma_j)$. But this atom must already be implied by $\Gamma_j$ since this set does not share variables with the other constraint sets. This means that $=_{d_i} \in \mathcal{R}_{\Gamma,0}$.

Consider now the conjunction $\Pi(v_1, \ldots, v_m) \wedge =_{d_i} (v_{\ell})$. Functionality of $D$ implies that the only solution of this formula maps each $v_{\ell}$ to $d_{\ell}$, $1 \leq \ell \leq n$, and hence we have $=_{d_{\ell}} \in \mathcal{R}_{\Gamma,1}$ for all $\ell$, $1 \leq \ell \leq n$. But then each atom $=_{d_{\ell}} (v_{j_{\ell}})$, $1 \leq \ell \leq n$, must be implied by some $\Gamma_{j', j' < j}$, that contains the variable $v_{j_{\ell}}$, due to the construction of $\text{Neg}_{j'}$. This shows that $\Gamma_1 \cup \cdots \cup \Gamma_j$ implies $\Pi(v_1, \ldots, v_m) \wedge \bigwedge_{i=1}^n =_{d_i} (v_{j_i})$, which in turn implies $\Pi(v_1, \ldots, v_m)$, as required.

### 3.4 $\mathcal{I}_K$ is Canonical

We now show that $\mathcal{I}_K$ behaves like a canonical model of $\mathcal{K}$.

**Lemma 3.4.** If $\mathcal{K}$ is consistent, then $\mathcal{I}_K \models_\alpha \mathcal{K}$.

**Proof.** Let $\mathcal{I}$ be a model of $\mathcal{K}$, and $f_\phi$ be the corresponding instantiation of $\mathcal{I}_K$ that exists by Lemma 3.1] $\mathcal{I}_K$ clearly satisfies the ABoxes of $\mathcal{K}$ due to the construction of $\mathcal{I}_K^{(0)}$. Furthermore, all inclusions and attribute range restrictions are satisfied due to the completion rules. Additionally, $\mathcal{I}_K$ respects the rigid names by construction.
Assume now that some \( \mathcal{I}_{K,i} \), \( i \geq 0 \), violates a functionality constraint \( \text{funct}(R) \in \mathcal{T} \). Then, either (i) \( \mathcal{I}_{K,i}^{(0)} \) already violates the constraint, (ii) this violation is the direct result of an application of a completion rule to some \( \mathcal{I}_{K,i}^{(\ell)} \), \( \ell \geq 0 \), or (iii) it is due to the completion of \( \mathcal{I}_{K,i}^{(\ell+1)} \) w.r.t. the rigid names after a rule application. In case (i), there must be \( a_1, a_2, a_3 \in N_i \) such that \( a_2 \neq a_3 \) and \( (a_1, a_2), (a_1, a_3) \in R^{(e)} \subseteq \mathcal{R}^{(e)}_i \). By Lemma 3.1 and the UNA, \( \mathcal{I} \) must violate \( \text{funct}(R) \) at time point \( i \), which contradicts our assumption that it is a model of \( K \). Case (ii) cannot be caused by \( \text{[CR1]} \) or \( \text{[CR5]} \) since these rules cannot modify the interpretation of a functional role (recall that such roles are not allowed to occur on the right-hand side of an inclusion). Furthermore \( \text{[CR4]} \) and \( \text{[CR6]} \) do not affect the interpretation of roles. But for any of the remaining two rules this would mean that the element \( e \) that triggered the rule was already present in \( (\exists R)^{\mathcal{I}_{K,j}} \) before the application of the rule, which contradicts the fact that these completion rules only add new elements if the existential restriction was not already satisfied. Finally, for case (iii), assume that \( \mathcal{I}_{K,i}^{(\ell+1)} \) violates the constraint because \( R \) is rigid and a new \( R \)-successor of \( e \) was added to \( \mathcal{I}_{K,j}^{(\ell+1)} \) for some \( j \neq i \) by a completion rule. Then we know that \( e \in (\exists R)^{\mathcal{I}_{K,j}} = (\exists R)^{\mathcal{I}_{K,i}} \) since \( \mathcal{I}^{(i)} \) respects the rigid names, which yields a contradiction by the same argument as before.

Suppose now that \( \mathcal{I}_{K,i} \) violates a concept or role disjointness constraint \( \text{disj}(X_1, X_2) \) in \( \mathcal{T} \), i.e., there is an \( e \in \Delta^{\mathcal{I}_{K,i}} \) such that \( e \in X_1^{\mathcal{I}_{K,i}} \cap X_2^{\mathcal{I}_{K,i}} \). By Lemma 3.1, \( \mathcal{I} \) must violate the constraint at \( i \), which again yields a contradiction. Finally, consider an attribute disjointness constraint \( \text{disj}(U_1, U_2) \in \mathcal{T} \) and assume that \( (e, v_1) \in U_1^{\mathcal{I}_{K,i}}, (e, v_2) \in U_2^{\mathcal{I}_{K,i}}, \) and \( (v_1, v_2) \) is implied by \( \Gamma_{e,i} \). By \( \text{[ES]} \) the pair \( (e, d) \) with \( d = f_{\phi}(v_1) = f_{\phi}(v_2) \) must belong to the interpretation of both \( U_1 \) and \( U_2 \) under \( \mathcal{I} \) at \( i \), which contradicts the fact that \( \mathcal{I} \) satisfies the disjointness constraint.

For the converse of this lemma, we prove an even stronger result, namely that \( f_{\phi}(\mathcal{I}_{K}) \) can be seen as canonical models of \( K \), for every MCQ \( \phi \).

**Lemma 3.5.** Let \( \phi \) be an arbitrary MCQ over the signature of \( K \). If there is an instantiation of \( \mathcal{I}_{K} \) and \( \mathcal{I}_{K} \models \phi \), then \( f_{\phi}(\mathcal{I}_{K}) \models K \).

**Proof.** By Lemma 3.2, \( f_{\phi} \) is an instantiation of \( \mathcal{I}_{K} \). It is easy to see \( f_{\phi}(\mathcal{I}_{K}) \) satisfies all ABoxes \( A_i, \ 0 \leq i \leq \sum, \) due to the construction of \( \mathcal{I}_{K}^{(0)} \), which does not contain any variables and is a substructure of \( \mathcal{I}_{K} \). Furthermore, inclusions and disjointness constraints over roles and functionality constraints are obviously not affected by the instantiation. Additionally, \( f_{\phi}(\mathcal{I}_{K}) \) respects the rigid names due to the construction of \( \mathcal{I}_{K} \).

We now show that we have \( B^{\mathcal{I}_{K,i}} = B^{f_{\phi}(\mathcal{I}_{K,i})} \) for all basic concepts \( B \). The claim for concept names and existential restrictions is immediate from the fact that they do not involve the concrete domain. Consider now an attribute restriction \( \exists a_1, \ldots, a_m. e \in \Delta^{\mathcal{I}_{K,i}}, \) and \( (e, d_o) \in \alpha_o^{f_{\phi}(\mathcal{I}_{K,i})}, \) \( 1 \leq a \leq m, \) such that \( (d_1, \ldots, d_m) \in \Pi^{\mathcal{I}_{K,i}} \). By construction, there are terms \( v_o \) with \( (e, v_o) \in \alpha_o^{\mathcal{I}_{K,i}}, \) and \( f_{\phi}(v_o) = d_o, \) \( 1 \leq a \leq m, \) where, if \( a_o \) is a constant, then \( v_o = d_o = a_o. \) Since \( \Pi \) is rigid for \( \mathcal{I}_{K,i} \) and \( f_{\phi} \) satisfies \( \Pi(v_1, \ldots, v_m) \), this atom must be implied by the set \( \text{Pos}_i \) with \( (e_j, i_j) = (e, i). \) Since \( v_1, \ldots, v_m \) only contains variables from \( \Gamma_{e,i} \), the atom \( \Pi(v_1, \ldots, v_m) \) is already implied by \( \Gamma_{e,i}. \) This shows that \( e \in (\exists a_1, \ldots, a_m. \Pi)^{\mathcal{I}_{K,i}} \). Conversely assume that this holds due to some terms \( v_1, \ldots, v_m \) occurring in \( \Gamma_{e,i} \) or \( \Delta^{\mathcal{I}_{K,i}} \). Then we have \( (f_{\phi}(v_1), \ldots, f_{\phi}(v_m)) \in \Pi^{\mathcal{I}_{K,i}} \) since \( f_{\phi} \) satisfies \( \Gamma_{e,i}, \) which proves the other direction of the inclusion.

Since the behavior of the basic concepts is not changed by the instantiation \( f_{\phi} \), we can immediately infer that all inclusions and disjointness constraints on concepts remain satisfied in \( f_{\phi}(\mathcal{I}_{K,i}) \).

Consider now an attribute inclusion \( U_1 \sqsubseteq U_2 \in \mathcal{T} \). Since every pair \( (e, v) \in U_1^{\mathcal{I}_{K,i}} \) is contained in \( U_2^{\mathcal{I}_{K,i}} \), we obtain the same relation after instantiation. For an attribute disjointness constraint
We now show that the canonical model $\mathcal{K}$ satisfies all concept, role, object equality, and temporal comparison atoms. By construction, there must be terms $v_1, v_2$ such that $(e, v_1) \in U_1^{\mathcal{F}(\mathcal{K}, \cdot)}$, $(e, v_2) \in U_2^{\mathcal{F}(\mathcal{K}, \cdot)}$, and $f_\phi(v_1) = f_\phi(v_2) = d$. Since the atom $\equiv(v_1, v_2)$ is contained in $\mathsf{Neg}$, with $(e, i) = (e, i)$, it must be implied by $\mathsf{Pos}$, and hence by $\Gamma_1 = \Gamma_{e,i}$. This contradicts our assumption that $\mathcal{K}$ satisfies $\mathcal{T}$. Finally, for an attribute range restriction $B \subseteq \forall \alpha_1, \ldots, \alpha_n \Phi \in \mathcal{T}$, consider any $e \in B^{\mathcal{F}(\mathcal{K}, \cdot)} = B^{\mathcal{K}}$, and $(e, d_0) \in \alpha_0^{\mathcal{F}(\mathcal{K}, \cdot)}$, $1 \leq o \leq m$, which means that there are terms $v_o$ with $(e, v_o) \in \alpha_o^{\mathcal{K}}$, and $f_\phi(v_o) = d_o$, $1 \leq o \leq m$. Since $\mathcal{K}$ satisfies $\mathcal{T}$, we know that $\Gamma_{e,i}$ implies $\Pi(v_1, \ldots, v_m)$. Since $f_\phi$ solves this set, we obtain $(d_1, \ldots, d_m) \in \Pi^D$, as required.

Let now $\phi$ be a safe MCQ. We assume in the following that all concrete domain elements in $\phi$ also occur in $\mathcal{K}$, and hence in $\mathsf{acdom}(\mathcal{K})$. This is without loss of generality, since we could add an attribute assertion $U(a, d)$ to $\mathcal{A}_0$, for every $d \in \Delta^D$ occurring in $\phi$, where $U \in \mathcal{N}_A$ and $a \in \mathcal{N}_I$ do not occur in $\phi$ or $\mathcal{K}$. Likewise, we assume that the elements of $\mathcal{N}_1(\phi)$ also occur in $\mathsf{acdom}(\mathcal{K})$. In essence, we only want to query information about named individuals for which we have data available in our input ABoxes.

We now show that the canonical model $\mathcal{K}$ yields the correct answers when evaluating MCQs over $\mathcal{K}$.

**Lemma 3.6.** If $\mathcal{K}$ is consistent, then we have $\mathsf{cert}(\phi, \mathcal{K}) = \mathsf{ans}_d(\phi, \mathcal{K})$.

**Proof.** By the definition of certain answers, it suffices to verify that, for every *Boolean* MCQ $\phi$, it holds that $\mathcal{K} \models \phi$ iff $\mathcal{K} \models \phi$.

For the “if”-direction, let $\pi : \mathsf{terms}(\phi) \to \Delta^{\mathcal{K}} \cup \Delta^D \cup \mathsf{Var}(\mathcal{K}) \cup \mathbb{N}$ be a homomorphism of $\phi$ into $\mathcal{K}$, and $\mathfrak{I} = (\mathfrak{I}_i)_{i \geq 0}$ be a model of $\mathcal{K}$ with $\mathfrak{I}_i = (\Delta^3, \mathcal{J}_i)$, $i \geq 0$. By Lemma 3.1, there are two functions $f_\pi : \Delta^{\mathcal{K}} \to \Delta^D$ and $f_\varphi : \mathsf{Var}(\mathcal{K}) \to \Delta^D$ that embed $\mathcal{K}$ into $\mathfrak{I}$. We now define the function $\pi' : \mathsf{terms}(\phi) \to \Delta^3 \cup \Delta^D \cup \mathbb{N}$ by setting $\pi'(x) := f_\varphi(\pi(x))$ for all object terms $x$, $\pi'(v) := f_\pi(\pi(v))$ for all value terms $v$, and $\pi'(t) := \pi(t)$ for all temporal terms $t$. It is straightforward to check that all object variables are mapped into $\Delta^3$, all concrete domain variables are mapped into $\Delta^D$, all temporal variables are mapped into $\mathbb{N}$, each $a \in \mathcal{N}_1(\phi)$ is mapped to $a^3$, and all elements of $\Delta^D \cup \mathbb{N}$ are mapped to themselves.

We now verify that $\pi'$ is indeed a homomorphism of $\phi$ into $\mathfrak{I}$. For any concept atom $\mathcal{A}(x, t) \in \phi$, we have $\pi(x) \in A^{\mathcal{K}}(\mathfrak{I})$ by assumption, and hence $\pi'(x) \in A^{\mathfrak{I}}(\mathfrak{I})$ by (E2). Similar arguments apply for role atoms, attribute atoms, object equality atoms, and temporal comparison atoms. Finally, consider a value comparison atom $\Pi(v_1, \ldots, v_m) \in \phi$, where $\Pi$ is an $m$-ary predicate of $\mathcal{D}_1$. Since a finite union of sets $\Gamma_{e,i}$ implies $\Pi(\pi(v_1), \ldots, \pi(v_m))$, and $f_\varphi$ solves all such sets by (E5) we know that $\pi'(v_1), \ldots, \pi'(v_m)) = (f_\varphi(\pi(v_1)), \ldots, f_\varphi(\pi(v_m))) \in \Pi^D$, as required.

For the “only if”-direction, assume that $\mathcal{K}$ is consistent and $\mathcal{K} \models \phi$. By Lemmas 3.1, 3.2 and 3.4, we know that $f_\pi$ is an instantiation of $\mathcal{K}$ and $f_\phi(\mathcal{K}) \models \mathcal{K}$, and hence $f_\phi(\mathcal{K}) \models \phi$. Let $\pi$ be a homomorphism of $\phi$ into $f_\phi(\mathcal{K})$. We define a homomorphism $\pi'$ of $\phi$ into $f_\phi(\mathcal{K})$. For an attribute atom $U(x, v, t) \in \phi$ that is satisfied by $\pi$ in $f_\phi(\mathcal{K})$, consider first the case that $v$ is a constant. Then we know that $(\pi(x), v) \in U^{\mathcal{F}(\mathcal{K}, \cdot)}$. If this tuple also occurs in $U^{\mathcal{K}}$, then we are done. Otherwise, there must be a variable $w$ such that $(\pi(x), w) \in U^{\mathcal{F}(\mathcal{K}, \cdot)}$ and $f_\phi(w) = v$. Since $v_w \in \mathcal{K}_x \cup \mathcal{K}_v$, by Lemma 5.3 the atom $\equiv(v, w)$ is implied by $\Gamma_{\pi(x), \pi(w)}$. Hence, the atom is also satisfied under $\pi'$ in the abstract interpretation $\mathcal{K}$. If $v$ is a (nondistinguished) variable, then let $U'(x', v', t')$ be the attribute atom that was chosen to define $\pi'(v)$, i.e., we have

$$\text{disj}(U_1, U_2) \in \mathcal{T},$$

assume that there is a pair $(e, d) \in U_1^{\mathcal{F}(\mathcal{K}, \cdot)} \cap U_2^{\mathcal{F}(\mathcal{K}, \cdot)}$. By construction,
\((\pi(x'), w') \in U^{\tilde{\mathcal{K}}, \pi(w'), \pi(v)} = f_\phi(w'), \pi'(v) = w'.\) Similarly, we know that there exists a term \(w\) such that \((\pi(x), w) \in U^{\tilde{\mathcal{K}}, \pi(w)}\) and \(\pi(v) = f_\phi(w)\). This means that \(f_\phi(w) = f_\phi(w')\), and hence by Lemma 3.3 and the fact that different constraint sets in \(\mathcal{K}\) do not share variables, the atom \(=(w, w')\) is implied by \(\Gamma_\pi(x), \pi(v) \cup \Gamma_\pi(x'), \pi(v')\). Since \(\pi'(v) = w'\), the mapping \(\pi'\) satisfies the attribute atom \(U(x, v, t)\).

Finally, consider any \(\Pi(v_1, \ldots , v_m) \in \phi\). For every variable \(v_o\), \(1 \leq o \leq m\), there is an attribute atom \(\alpha_o(x_o, v_o, t_o)\) and a term \(v'_o \in \text{terms}(\Gamma_\pi(x_o), \pi'(t_o))\) as above. If \(v_o\) is a constant, then we set \(\alpha_o := v'_o := v_o\). We thus have \(\{f_\phi(v'_1), \ldots, f_\phi(v'_m)\} = (\pi(v_1), \ldots , \pi(v_m))\) \(\in \Pi^D\) and \(\pi'(v_o) = v'_o\), \(1 \leq o \leq m\). By Lemma 3.3, we obtain that \(\Pi(v'_1, \ldots, v'_m)\) is implied by all sets \(\Gamma_\pi(x_o), \pi'(t_o)\) for which \(v_o\) is a variable. This shows that \(\pi'\) satisfies \(\Pi(v_1, \ldots, v_m)\) according to the modified definition of homomorphisms for abstract interpretations. 

\section{Rewriting MCQs}

For a practical algorithm, it is necessary to reduce query answering to a finite structure. However, for this purpose the MCQ needs to be rewritten. We extend the UCQ rewriting from \cite{12,23} and for the temporal components we incorporate ideas from \cite{3}. We use a simple, unoptimized rewriting algorithm in order to simplify the proof of correctness.

In the following, we introduce several operators that are applied to sets of MCQs, starting with the singleton set \(\{\phi\}\), in order to obtain all relevant MCQs that imply \(\phi\). In such sets, we regard MCQs as equal if they are equivalent modulo a renaming of the nondistinguished variables.

\textbf{Nonessential variables.} Following \cite{12,23,24}, we say that an existentially quantified variable \(y\) of \(\phi\) is nonessential if it occurs only once in \(\phi\). For a given set \(\Phi\) of MCQs, we define the set \(\text{anon}(\Phi)\) of MCQs resulting from \(\Phi\) by replacing all nonessential variables by the special symbol \(\_\). The set \(\text{inst}(\Phi)\) reverses this process, i.e., it contains all MCQs in which each separate occurrence of \(\_\) in \(\phi\) is replaced by a fresh existentially quantified variable of the appropriate type (determined by the position in the atom it occurs in). For an existential restriction \(\exists R\), an element \(x \in N_{TV} \cup N \cup \{\_\}\), and \(t \in N_{TV} \cup N\), we define the query atom

\[(\exists R)(x, t) := \begin{cases} P(x, \_\), t & \text{if } R = P \in N_R, \\ P(\_, x, t) & \text{if } R = P^- \text{ with } P \in N_R. \end{cases}\]

Similarly, for an attribute restriction \(B = \exists \alpha_1, \ldots , \alpha_m I\), we define the set of atoms \(B(x, t)\) as follows. Let \(v_1, \ldots , v_m\) be value terms such that \(v_m = \alpha \) whenever \(\alpha \in \Delta^D\), \(1 \leq o \leq m\), and if \(\alpha_o \in N_A\), then \(v_o\) is a nondistinguished variable. Then \(B(x, t)\) contains \(\Pi(v_1, \ldots , v_m)\) and the atoms \(\alpha_o(x, v_o, t)\) for all \(\alpha, 1 \leq o \leq m\), with \(\alpha \in N_A\). We say that \(B(x, t)\) occurs in an MCQ \(\phi\) if a nonempty subset of its atoms occurs in \(\phi\) and all concrete domain variables occurring in them do not occur elsewhere in \(\phi\). Similarly, we can add such an atom \(\exists \alpha_1, \ldots , \alpha_m I(x, t)\) to an MCQ \(\phi\) by introducing fresh nondistinguished variables \(v_o\) as required and adding the atoms listed above to \(\phi\).

\textbf{Backward chaining.} For two MCQs \(\phi, \phi'\), we write \(\phi \rightarrow_{T, D} \phi'\) if one of the following cases applies:

- \textbf{Inclusions:}
  - There exists \(X_1 \sqsubseteq X_2 \in T\) such that \(X_2(x, t)\) occurs in \(\phi\), and
  - \(\phi'\) is obtained from \(\phi\) by replacing this atom (or atoms) with \(X_1(x, t)\),
where \( \bar{x} \) denotes a vector of terms matching the type of \( X_2 \), i.e.,

- \( \bar{x} = x_1 \in N_I \cup N_{OV} \cup \{ \_ \} \) if \( X_2 \) is a basic concept,
- \( \bar{x} \) is a pair of such terms if \( X_2 \) is a role, and
- \( \bar{x} = (x_1, v_2) \) consists of \( x_1 \in N_I \cup N_{OV} \cup \{ \_ \} \) and \( v_2 \in (\Delta^D \setminus \Omega) \cup N_{CV} \cup \{ \_ \} \) if \( X_2 \) is an attribute name.

- **Attribute range constraints:**
  - There exists \( B \subseteq \forall \alpha_1, \ldots, \alpha_m. \Pi \in T \) such that \( \Pi(v_1, \ldots, v_m) \) occurs in \( \phi \), and
  - \( \phi' \) is obtained from \( \phi \) by replacing \( \Pi(v_1, \ldots, v_m) \) with \( B(x, t) \) and, for all \( o, 1 \leq o \leq m \), the atom \( =_{\alpha_o}(v_o) \) if \( \alpha_o \in \Delta^D \), and \( o(x, v_o, t) \) if \( \alpha_o \in N_A \), where \( x \) and \( t \) are fresh nondistinguished variables.

- **Concrete domain implications:**
  - The atom \( \Pi(v_1, \ldots, v_m) \) occurs in \( \phi \),
  - there exists a \( T \)-conjunction \( \psi \) using only the predicates from \( \mathfrak{R}_K \cup \mathfrak{R}_p \), variables from \( \{v_1, \ldots, v_m\} \) or fresh variables, and constants from \( \Delta^D(K) \), such that \( \psi \) implies \( \Pi(v_1, \ldots, v_m) \),
  - \( \phi' \) is obtained from \( \phi \) by replacing \( \Pi(v_1, \ldots, v_m) \) with all atoms of \( \psi \), and, for each fresh variable \( v \) an attribute atom \( U(x, v, t) \), where \( U \) is an attribute name occurring in \( K \), \( x \) is a fresh object variable, and \( t \) is a fresh temporal variable.

This condition also applies to the atom \( T_D(v) \) for \( v \in N_{CV}(\phi) \), which does not have to be explicitly present in \( \phi \) (implicitly, this atom is always satisfied).

- **Variable splitting:**
  - The atom \( =_d(v) \) occurs in \( \phi \), and
  - \( \phi' \) is obtained from \( \phi \) by replacing another occurrence of \( v \) in \( \phi \) by a fresh nondistinguished variable \( v' \) and adding the atom \( =_d(v') \).

These cases correspond to the completion rules and the modified satisfaction condition for concrete domain atoms in abstract interpretations. We now define the set

\[
\text{infer}(\Phi) := \{ \phi' \mid \phi \in \Phi, \ \phi \rightarrow_{T,D} \phi' \}
\]
as the result of applying all the replacements formalized by \( \rightarrow_{T,D} \) in all possible ways to all MCQs in \( \Phi \).

**Substitutions.** A substitution w.r.t. an MCQ \( \phi \) is a function \( \sigma : N_V(\phi) \rightarrow \text{terms}(\phi) \) with the property that all variables are mapped to terms of the corresponding type, e.g., elements of \( N_{OV} \) are mapped to \( N_I \cup N_{OV} \). Such a substitution allows us to unify any variable of \( \phi \) with any (compatible) term occurring in \( \phi \). We denote by \( \text{subst}(\phi) \) the set of all such substitutions, and by \( \sigma(\phi) \) the MCQ that is obtained from \( \phi \) by replacing all variables according to \( \sigma \) and subsequently removing duplicate atoms. The substitution also has to be applied to the tuple of answer variables, and hence the resulting tuple may contain constants and multiple occurrences of the same variable. This does not affect the semantics, if we use the convention that constants are not affected by applying a possible answer. We now define

\[
\text{merge}(\Phi) := \{ \sigma(\phi) \mid \phi \in \Phi, \ \sigma \in \text{subst}(\phi) \}.
\]

If \( \sigma(\phi) \) contains an obvious unsatisfiability, e.g., \( t < t \), then we could simply discard it. However, this optimization is not important for the correctness of our procedure.
Rigid atoms. The operator rigid likewise applies to sets of MCQs and, in each MCQ \( \phi \), replaces an arbitrary atom \( X(\bar{x}, t) \) with a rigid concept or role \( X(X(\bar{x}, t')) \), where \( t' \) is a fresh existentially quantified temporal variable. This reflects the fact that the satisfaction of rigid atoms does not depend on a specific time point.

The rewriting. We now define the operator

\[
\text{step}(\Phi) := \Phi \cup \text{rigid}(\Phi) \cup \text{merge}(\Phi) \cup \text{inst}(\text{anom}(\Phi))
\]

and the set \( \text{PerfectRef}_{T,D}(\phi) \) as the result of exhaustively applying \( \text{step} \) to \( \{ \phi \} \), i.e., until we have reached a fixed-point \([12]\). This set is finite since there can be only finitely many combinations of atoms over the signature of \( K \). It is easy to verify the following observation.

**Proposition 4.1.** If \( \phi \) is safe, then every \( \phi' \in \text{PerfectRef}_{T,D}(\phi) \) is also safe, and moreover satisfies \( \mathcal{R}_{\mathcal{X}} \subseteq \mathcal{R}_{K} \cup \mathcal{R}_{\phi} \).

Since our transformation has to deal with implications over \( D \), the proof of the following result holds different challenges than similar ones in \([3,12,27]\) for separate extensions of \( \text{DL-Lite} \) with concrete domains and temporal semantics (but our temporal formalism is less expressive than the one in \([3]\)). For the following lemma, we consider the “finite” \( \text{DL-LTL} \)-structure \( \mathcal{I}_{A} = (\mathcal{I}_{A,(i)})_{i \geq 0} \) that is defined similarly to \( \mathcal{I}^{(0)} \), but uses \( X^{[A,i]} := \{ e \mid i \leq n, X(e) \in A \} \) to interpret all concept names, role names, or attribute names \( X \). This substructure of \( \mathcal{I}^{(0)} \) can be seen as finite since it does not satisfy anything at time points after \( n \), but it does not necessarily respect the rigid names.

**Lemma 4.2.** If \( K \) is consistent, then we have

\[
\text{ans}_{\mathcal{A}}(\phi, \mathcal{I}_{K}) = \bigcup_{\phi' \in \text{PerfectRef}_{T,D}(\phi)} \text{ans}(\phi', \mathcal{I}_{A})
\]

**Proof.** For the \( \supseteq \)-direction, let \( a \) be a possible answer with \( \mathcal{J}_{A} \models a(\phi') \) for some element \( \phi' \in \text{PerfectRef}_{T,D}(\phi) \). Since \( \mathcal{J}_{K} \) is an extension of \( \mathcal{I}_{A} \), this implies that \( \mathcal{J}_{K} \models a(\phi') \). We know that \( \phi' \) must be the result of a finite number of applications of the operators \( \text{anom}, \text{infer}, \text{inst}, \text{merge} \), and \( \text{rigid} \) to \( \phi \). Hence, it suffices to show that for each of these operations every answer tuple of the result in \( \mathcal{J}_{K} \) is also an answer tuple of the original query in \( \mathcal{J}_{A} \). For the purposes of this proof, we can treat each occurrence of the symbol \( \_ \) as a unique nondistinguished variable, and hence the claim for \( \text{anom} \) and \( \text{inst} \) is trivial.

Consider now an application of \( \text{merge} \), in which a substitution was applied to an intermediate MCQ \( \phi'' \), resulting in the MCQ \( \sigma(\phi'') \). Clearly, a homomorphism \( \pi \) of \( a(\sigma(\phi'')) \) into \( \mathcal{J}_{K} \) yields a possible answer \( a' \) and a homomorphism \( \pi' \) of \( a'(\phi'') \) into \( \mathcal{J}_{K} \), by setting \( a'(x) := a(\sigma(x)) \) for all answer variables \( x \in \text{FVar}(\phi'') \), \( \pi'(x) := \pi(\sigma(x)) \) for all \( x \in \text{terms}(a'(\phi'')) \setminus \text{FVar}(\sigma(\phi'')) \), and \( \pi'(x) := a(\sigma(x)) \) for all \( x \in \text{terms}(a'(\phi'')) \cap \text{FVar}(\sigma(\phi'')) \). Recall that individual names, concrete domain elements, and temporal constants are not affected by \( \sigma \), and variables must be replaced by terms of the same type. Moreover, we obtain the same answer tuple due to our definition of \( a' \).

Assume now that rigid was applied to \( \phi'' \) in order to obtain \( \phi''' \) by equipping a rigid atom \( A(x,t) \) with a fresh nondistinguished temporal variable \( t' \), and let \( \pi \) be a homomorphism of \( a(\phi''') \) into \( \mathcal{J}_{K} \). We discuss only the case of a rigid concept atom; the case of rigid role atoms can be handled using similar arguments. For all terms \( x \in \text{terms}(a(\phi''')) \), we define \( \pi'(x) := \pi(x) \). If \( a(t) \) is not present in \( \phi''' \), we additionally set \( \pi'(a(t)) \) to an arbitrary element of \( \mathcal{N} \). To show that \( \pi' \) is a homomorphism of \( a(\phi'') \) into \( \mathcal{J}_{K} \), it clearly suffices to consider the atom \( a(A(x,t)) \). Since \( A \) is rigid and \( \mathcal{J}_{K} \) respects the rigid names, we obtain \( \pi'(a(x)) = \pi(a(x)) \in \mathcal{A}^{T_{K},\pi(x')} = \mathcal{A}^{T_{K},\pi'(a(t))} \), exactly as required.
Finally, consider the case of an MCQ $\phi''$ that was used to obtain $\phi'''$ with $\phi'' \rightarrow_{T,D} \phi'''$ via the operator infer. We consider the cases of the definition of $\rightarrow_{T,D}$:

- Assume that there exists an inclusion $X_1 \subseteq X_2 \in T$ such that (part of) $X_2(\vec{x}, t) \in \phi''$ for appropriate terms $\vec{x}, t$, was replaced by $X_1(\vec{x}, t)$ in order to obtain $\phi'''$. Again, let $\pi$ be a homomorphism of $a(\phi''')$ into $J_{K_C}$. A homomorphism $\pi'$ of $a(\phi''')$ into $J_{K_C}$ can be constructed using the completion rules (CR1), (CR5). We consider here only the case of an inclusion of the form $\exists \alpha_1, \ldots, \alpha_m \Pi \subseteq \exists \alpha'_1, \ldots, \alpha'_m \Pi'$; the other kinds of inclusions can be treated using similar, but simpler, arguments. By assumption, $\phi'''$ contains a subset of $\{\alpha'_1(\vec{x}, v'_1, t), \ldots, \alpha'_{k_2}(\vec{x}, v'_k, t), \Pi'(v'_1, \ldots, v'_k)\}$, where the variables $v'_i$ do not occur in any other atoms of $\phi'''$. In $\phi''$, these atoms were replaced by $\Pi(v_1, \ldots, v_m)$ and appropriate attribute atoms $\alpha_\omega(\vec{x}, v_\omega, t)$, in which all concrete domain variables are fresh.

Hence, terms $\text{terms}(a(\phi'''))$ and terms $\text{terms}(a(\phi'''))$ differ only in the terms $v'_1, \ldots, v'_k$ and $v_1, \ldots, v_m$.

Moreover, we know that $(\pi(a(x)), w_\omega) \in \alpha_\omega^z_{K_C, \pi(a(x))}$, $1 \leq \omega \leq m$, and $(w_\omega, \pi(v_\omega))$ and $\Pi(\pi(v_1), \ldots, \pi(v_m))$ are implied by the relevant constraint sets $\Gamma_{\pi, t}$. Since the constraint sets do not share variables, we infer that $\Gamma_{\pi(a(x)), \pi(a(t))}$ implies $\Pi(w_1, \ldots, w_m)$. Thus, we obtain $\pi(a(x)) \in \exists \alpha_1, \ldots, \alpha_m \Pi^r_{K_C, \pi(a(x))}$ for some finite $\ell \geq 0$. By (CR4) there must exist terms $\phi'_\omega$ such that $\pi(a(x)) \in (\alpha_\omega^z_{K_C, \pi(a(x))})$, $1 \leq \omega \leq k$, and $\Pi(\pi(v_1), \Pi(\pi(v_k)))$ is implied by $\Gamma_{\pi(a(x)), \pi(a(t))}$. Hence, by defining $\pi'(v'_\omega) := v'_\omega$, $1 \leq \omega \leq k$, and $\pi'(z) := \pi(z)$ for all $z \in \text{terms}(a(\phi''')) \setminus \{v_1, \ldots, v_m\}$, we can satisfy the atoms that were replaced in $\phi'''$; all other atoms remain satisfied since the variables among $v'_1, \ldots, v'_k$ do not occur in them.

- Consider $B \subseteq \forall \alpha_1, \ldots, \alpha_m \Pi \in T$ such that $\Pi(v_1, \ldots, v_m)$ occurs in $\phi'''$, and in $\phi''$ this atom was replaced by $B(x, t)$ and $\alpha_\omega(\vec{x}, v_\omega, t)$ or $=_{\alpha_\omega}(v_\omega)$, depending on the type of $\alpha_\omega$, $1 \leq \omega \leq m$. Let $\pi$ be a homomorphism of $a(\phi''')$ into $J_{K_C}$. Hence, we have

- $\pi(a(x)) \in B_{K_C, \pi(a(x))}$,
- $w_\omega := \pi(a(v_\omega)) = \alpha_\omega$ whenever $\alpha_\omega \in \Delta_D$, and
- for all attribute names $\alpha_\omega$, there exist $w_\omega$ such that $\pi(a(x)), w_\omega \in \alpha_\omega^z_{K_C, \pi(a(x))}$ and $(w_\omega, \pi(a(v_\omega)))$ is implied by the relevant constraint sets.

By (CR6) we obtain that $\Gamma_{\pi(a(x)), \pi(a(t))}$ implies $\Pi(w_1, \ldots, w_m)$. Hence, the atom $\Pi(\pi(a(v_1)), \ldots, \pi(a(v_m)))$ is implied by the relevant constraint sets, which shows that $\pi$ is also a homomorphism of $a(\phi''')$ into $J_{K_C}$.

- Consider now an atom $\Pi(\pi(v_1), \ldots, \pi(v_m))$ and a $D$-conjunction $\psi$ using variables from $\{v_1, \ldots, v_m\}$ and additional fresh variables, such that $\psi$ implies $\Pi(v_1, \ldots, v_m)$, and the latter atom was replaced in $\phi'''$ by the atoms of $\psi$ and additional attribute atoms to obtain $\phi''$. Let $\pi$ be a homomorphism of $a(\phi''')$ into $J_{K_C}$, which means that $\pi(a(\psi))$ is implied a union of finitely many constraint sets $\Gamma_{\pi, t}$. If a variable $\omega$, $1 \leq \omega \leq m$, does not occur in $\psi$, then its value is irrelevant for the implication, and moreover it must still occur in $\phi''$ since $\phi'''$ is safe. Hence, the same constraint sets imply $\Pi(\pi(a(v_1)), \ldots, \pi(a(v_m)))$. Since constraint sets do not share variables, this implication depends only on those constraint sets containing the variables among $\pi(a(v_1)), \ldots, \pi(a(v_m))$, and hence $\pi$ satisfies the atom $\Pi(\pi(a(v_1)), \ldots, \pi(a(v_m)))$ (as well as all other atoms) in $a(\phi'')$.

- Finally, assume that $=_{\psi}(v)$ occurs in $\phi''$, and in $\phi'''$ we replaced one other occurrence of $v$ by a fresh nondistinguished variable $v'$ and added $=_{\psi}(v')$. Let $\pi$ be a homomorphism of $a(\phi''')$ into $J_{K_C}$. Hence, we have $\pi(a(v)) = \pi(a(v')) = d$, which means that $v$ and $v'$ can be used interchangeably without changing the satisfaction of the query. Thus, $\pi$ is also a homomorphism of $a(\phi'')$ into $J_{K_C}$.
For the $\subseteq$-direction, we assume that $\mathcal{J}_K \models_a a(\phi)$ for a possible answer $a$, and first show
that there exists a $\phi' \in \text{PerfectRef}_{\mathcal{T}, \mathcal{D}}(\phi)$ with $\mathcal{J}_{K(0)}^{(0)} \models_a a(\phi')$ such that $a(\phi)$ and $a(\phi')$ yield the same answer tuple. For this purpose, let $\ell \geq 0$ be the smallest index for which there is a $\phi' \in \text{PerfectRef}_{\mathcal{T}, \mathcal{D}}(\phi)$, a possible answer $a'$ with the above property, and a homomorphism of $\pi(\phi')$ into $\mathcal{J}_K^{(\ell)}$. Such an index must exist since $\mathcal{J}_K \models_a a(\phi)$, $\phi \in \text{PerfectRef}_{\mathcal{T}, \mathcal{D}}(\phi)$, and due to the fairness requirement in the construction of $\mathcal{J}_K$. If $\ell = 0$, then we have proven the claim; we now consider the case that $\ell > 0$ and show that it is impossible. Let $\phi' \in \text{PerfectRef}_{\mathcal{T}, \mathcal{D}}(\phi)$ and $\pi$ be a homomorphism of $\pi(\phi')$ into $\mathcal{J}_K^{(\ell)}$. Since $\ell$ is minimal, we know that the last completion rule applied to obtain $\mathcal{J}_K^{(\ell)}$ from $\mathcal{J}_K^{(\ell-1)}$ was necessary to satisfy an atom of $\pi(\phi')$ under $\pi$. We make a case distinction on the type of the rule that was applied.

- If $\text{(CR1)}$ was applied, then there is an inclusion $X_1 \subseteq X_2 \in \mathcal{T}$ with $X_2 \in \mathcal{N}_C \cup \mathcal{N}_R \cup \mathcal{N}_A$, and a tuple $e \in X_2^{(\ell-1)}$ that was added to $X_2^{(\ell-1)}$. We consider first the case that $X_2 \in \mathcal{N}_C \cup \mathcal{N}_R$. Since we assumed that the rule application is necessary to satisfy $\pi(\phi')$ via $\pi$, there must be at least one atom $X_2(\bar{x}, t) \in \phi'$ such that $\pi(\pi(\phi')) = e$ and either
  (i) $X_2$ is flexible and $\pi(\pi(\phi')) = i$ or
  (ii) $X_2$ is rigid and $\pi(\pi(\phi'))$ is arbitrary.

  In case (i), we can replace each such atom $X_2(\bar{x}, t)$ by $X_1(\bar{x}, t)$, according to $\mathcal{T}_\mathcal{D}$. The resulting MCQ is also an element of $\text{PerfectRef}_{\mathcal{T}, \mathcal{D}}(\phi)$ and is satisfied by $a'$ and $\pi$ in $\mathcal{J}_K^{(\ell-1)}$, which contradicts our assumption on the minimality of $\ell$.

  In case (ii), we first replace each tuple $X_2(\bar{x}, t)$ using the operator rigid by $X_2(\bar{x}, t')$, where $t'$ is a fresh variable. The resulting MCQ $\phi'' \in \text{PerfectRef}_{\mathcal{T}, \mathcal{D}}(\phi)$ is such that $\pi(\phi'')$ can be satisfied by mapping all old terms according to $\pi$, and the new variables $t'\to i$. Since each $\bar{x}$ from above still satisfies $\pi'(\pi'(\bar{x})) = e$, we can now proceed as in case (ii).

  Finally, consider the case that $X_2 \in \mathcal{N}_A$, and hence also $X_1 \in \mathcal{N}_A$. By our assumption, the new tuple $(e, w) \in X_2^{(\ell-1)}$ was necessary to satisfy at least one atomic atom $X_2(x, v, t) \in \phi'$ under $\pi$, i.e., we have $\pi(\pi(\phi')) = e$ and $\pi(\pi(\phi')) = i$ and the relevant constraint sets among $\Gamma_{e', e}$ imply $\pi(\pi(\phi')) = (w, \pi(\pi(\phi'))))$. Since $(e, w) \in X_2^{(\ell-1)}$ and $\Gamma_{e', e}^{(\ell-1)} = \Gamma_{e', e}^{(\ell)}$, we can apply $\mathcal{T}_\mathcal{D}$ to replace $X_2(x, v, t)$ with $X_1(x, v, t)$ to obtain an element of $\text{PerfectRef}_{\mathcal{T}, \mathcal{D}}(\phi)$ that is satisfied by $a'$ and $\pi$ in $\mathcal{J}_K^{(\ell-1)}$, which again yields a contradiction.

- If $\text{(CR4)}$ was applied, then there is an inclusion $B \subseteq \exists_{O_1}, \ldots, O_m$ in $\mathcal{T}$, an element $e \in B^{(\ell-1)}$, and variables $v_1, \ldots, v_m$ such that $(e, v_0)$ is added to $\alpha_{O^{(\ell-1)}}$, $1 \leq o \leq m$, and $\Pi(v_1, \ldots, v_m)$ is added to $\Delta_{e, i}$. By our assumption, there must be at least one atom of the form
  (a) $\alpha_o(x, v', t)$ such that $\pi(\pi(\phi')) = e$, $\pi(\pi(\phi')) = i$, and $=(\pi(\pi(\phi'))), v_0\rangle$ is implied by a union of constraint sets $\Gamma_{e', e}^{(\ell)}$; or
  (b) $\Pi'(v_1, \ldots, v'_k)$ such that $\Pi'(\pi(\pi(\phi'))), \ldots, \pi(\pi(\phi')))$ is implied by some constraint sets $\Gamma_{e', e}^{(\ell)}$.

  We first replace the atoms of type $\text{(B)}$ as follows.

  For one such atom $\Pi'(v_1, \ldots, v'_k)$, let $\Gamma$ be a minimal subset of the union of all sets $\Gamma_{e', e}^{(\ell)}$ that still implies $\Pi'(\pi(\pi(\phi'))), \ldots, \pi(\pi(\phi')))$. By the definition of $\pi$, we need to consider only those constraint sets containing the variables among $\pi(\pi(\phi'))$, $\ldots, \pi(\pi(\phi'))$, and hence $\Gamma$ is finite. We consider the set $\pi^{-1}(\Gamma)$ that is obtained from $\Gamma$ by replacing every variable of the form $\pi(\pi(\phi'))$, $1 \leq o \leq k$, by $\pi(\pi(\phi'))$. Then we have that $\Gamma' = \pi^{-1}(\Gamma) \cup \{\pi(\pi(\phi')) | 1 \leq o \leq k, \pi(\pi(\phi')) = d \in \Delta^{(0)}\}$ implies $\Pi'(v_1, \ldots, v'_k)$. Note that $\Gamma'$ only contains predicates from $\mathcal{N}_K$: by Proposition 4.1, we know that $\phi'$ is safe, and hence if a variable $v'_o$ is mapped to a constant $\pi(\pi(\phi')) = d$, then either

24
\(\pi(a'(v'_k)) = a'(v'_o) \in \text{acdom}(K)\), or there is an attribute atom \(U(x, v'_o, t)\) in \(\phi'\), i.e., we have \((\pi(a'(x)), d) \in U^\mathcal{I}(t)\), and thus \(d\) must occur in \(K\) due to the construction of \(\mathcal{X}^\mathcal{I}\). Furthermore, \(\Gamma'\) uses only variables that already occur in \(\phi'\), ones from \(\text{Var}(\mathcal{X}^{\mathcal{I}}_{\mathcal{K}})\), or some of the fresh variables, and for the latter two cases we assume without loss of generality that they do not occur in \(\phi'\). Finally, \(\Gamma'\) can only use constants from \(\Delta^\mathcal{D}(K)\). We now replace \(\Pi'(v'_1, \ldots, v'_k)\) in \(\phi'\) according to \(\rightarrow_{T, \mathcal{D}}\) by the atoms of \(\Gamma'\), where for each “new” variable \(v\) in this set we know that it must occur in some \((e', v) \in U^\mathcal{I}(t')\), and hence we can add the atom \(U(x, v, t)\) to the query. These atoms can be satisfied by mapping \(v\) to itself, \(x\) to \(e'\), and \(t\) to \(t'\), and the atoms in \(\Gamma'\) are then also satisfied since they directly occur in the relevant sets \(\Gamma_{e', e}^{\mathcal{I}}\) or are already satisfied by \(\pi'\) and \(\pi\). Moreover, the only new attribute atoms that are not already satisfied in \(\mathcal{X}^{\mathcal{I}}_{\mathcal{K}}\) are new ones of the form \([a]\) above, which involve the fresh variables \(v_o\) introduced by the completion rule.

By replacing all atoms of type \([b]\) in this way, we obtain another query \(\phi''\) from \(\text{PerfectRef}_{T, \mathcal{D}}(\phi)\) and a homomorphism \(\pi'\) of \(a'(\phi'')\) into \(\mathcal{X}^{\mathcal{I}}_{\mathcal{K}}\). The only atoms in \(\phi''\) that are not already satisfied by \(a'\) and \(\pi'\) in \(\mathcal{X}^{\mathcal{I}}_{\mathcal{K}}\) are those of the form \(\pi^{-1}(\Pi(v_1, \ldots, v_m))\) introduced by the above replacements, and the ones of the form \([a]\) (possibly more than in \(\phi'\)). However, each of the atoms of the first kind is mapped by \(a'\) and \(\pi'\) to the same atom, namely \(\Pi(v_1, \ldots, v_m)\), and hence we can apply a substitution \(\sigma\) that unifies these atoms. Note that all these atoms agree on the constants \(\alpha_o \in \Delta^\mathcal{D}, 1 \leq o \leq m\), of the attribute restriction \(\exists a_{o1}, \ldots, a_{om}\Pi\) under consideration.

Similarly, each atom \(\alpha_o(x, v'_o, t)\) of type \([a]\) is mapped by \(a'\) and \(\pi'\) to some \(\alpha_o(e, w_o, i)\) for which \(=(v_o, w_o)\) is implied by some constraint sets \(\Gamma_{e', e}^{\mathcal{I}}\), and hence we can similarly ensure by an application of \text{merge} that there is at most one such atom for each fresh variable \(v_o\), and moreover that they use the same object variable \(x\) and the same temporal variable \(t\). If the resulting merged atom \(\alpha_o(x, v'_o, t)\) is such that \(v'_o\) is a constant or a distinguished variable, then we have \(w_o = a'(v'_o) \in \Delta^\mathcal{D}\) and \(=w_o(v_o)\) is implied by the constraint sets. Since \(v_o\) is a fresh variable, this atom is already implied by \(\Pi(v_1, \ldots, v_m)\).

We can now eliminate the occurrence of \(v'_o\) in \(\alpha_o(x, v'_o, t)\) by first introducing a new atom \(=w_o(v'_o)\) (since it implies \(\top_{\mathcal{D}}(v'_o)\)), and then splitting the variable in order to obtain atoms \(\alpha_o(x, v'_o, t), =w_o(v'_o),\) and \(=w_o(v'_o)\). We can then eliminate the third atom by replacing it with \(\Pi(v'_o, \ldots, v'_m)\) using some fresh nondistinguished variables. The latter atom is of type \([b]\) and hence can be merged as described above. Now, all atoms \(\alpha_o(x, v'_o, t)\) of type \([a]\) must be such that \(v'_o\) is nondistinguished. This variable can further be merged with the \(o\)-th term of the single remaining atom of type \([b]\) since both can be mapped to \(v_o\). All these operations can be done using \text{infer} and \text{merge}, and hence we stay inside \(\text{PerfectRef}_{T, \mathcal{D}}(\phi)\). It is straightforward to adapt the possible answer \(a'\) of \(\phi''\) into a new possible answer \(a''\) of the resulting query that yields the same answer tuple.

The result of this second step is another element \(\phi'''\) of \(\text{PerfectRef}_{T, \mathcal{D}}(\phi)\) that is satisfied in \(\mathcal{X}^{\mathcal{I}}_{\mathcal{K}}\) via \(a''\) and \(\pi'\), and the only attributes that are not already satisfied \(\mathcal{X}^{\mathcal{I}}_{\mathcal{K}}\) form a subset of \(\exists a_{o1}, \ldots, a_{om}\Pi(x, t)\) (note that we have eliminated all constants and nondistinguished variables \(v'_o\) above). This implies that no other terms can be mapped to the new variables \(v_o\) introduced by the completion rule, and hence the concrete domain variables in these atoms do not occur elsewhere in \(\phi'''\). This means that we can finally apply the operator \text{infer} in order to obtain another element of \(\text{PerfectRef}_{T, \mathcal{D}}(\phi)\) that is satisfied in \(\mathcal{X}^{\mathcal{I}}_{\mathcal{K}}\) via \(a''\) and \(\pi'\), which yields a contradiction.

- If \([\text{CR6}]\) was applied, then there is an attribute range constraint \(B \subseteq \forall a_{o1}, \ldots, a_{om}\Pi \in T\) and an element \(e \in B^{\mathcal{I}_{\mathcal{K}}_{\mathcal{K}}}, (e, v_o) \in a^{\mathcal{I}_{\mathcal{K}}_{\mathcal{K}}}, 1 \leq o \leq m,\) and \(\Pi(v_1, \ldots, v_m)\) was added to \(\Gamma_{e, i}\) in order to obtain \(\Gamma_{e, i}^{\mathcal{I}}\). By our assumption, this atom is necessary to imply one or more atoms of the form \(\Pi'\left(\pi(a'(v'_1)), \ldots, \pi(a'(v'_k))\right)\) for value comparison atoms.
This replacement does not affect the satisfaction of any other atoms since the images of all other variables under $a\cdot$ and $\pi$ remain the same. $\Box$

Finally, we need to show that we can even obtain an element $\phi''$ of $\text{PerfectRef}_{T,D}(\phi)$ and a possible answer $a''$ such that $a''(\phi'')$ is satisfied in the finite structure $\mathcal{J}_A$. If an atom $X(\vec{x},t)$ of $\phi'$ is satisfied by $a'$ and a homomorphism $\pi$ into $\mathcal{J}_K^{(0)}$, then either it involves a flexible name, and hence is also satisfied by the same mappings in $\mathcal{J}_A$, or it involves a rigid name $X$, and then the definition of $\mathcal{J}_K^{(0)}$ yields that there exists a time point $i$ at which it is satisfied in $\mathcal{J}_A$. We can hence replace this atom using the operator rigid by $X(\vec{x},t')$, where $t'$ is a fresh nondistinguished temporal variable, and then extend $\pi$ by mapping $t'$ to $i$, in order to satisfy the new atom in $\mathcal{J}_A$. This replacement does not affect the satisfaction of any other atoms since the images of all other variables under $a'$ and $\pi$ remain the same.

Lemmas 3.4, 3.5, 3.6 and 4.2 suggest the following procedure for answering an MCQ $\phi$ over $\mathcal{K}$: Check that $\mathcal{K}$ is consistent, compute $\text{PerfectRef}_{T,D}(\phi)$, and then answer the union of all MCQs $\mathcal{J}_K$ in this set over the finite structure $\mathcal{J}_A$, which can be seen as the minimal Herbrand model of the sequence $(\mathcal{A}_i)_{0 \leq i \leq n}$.

By Lemmas 3.4 and 3.5, the first problem is equivalent to checking whether $\mathcal{J}_K$ has an instantiation and $\mathcal{J}_K \models_a \mathcal{K}$ holds. For the first part, observe that there can be only finitely many constraint sets $\Gamma_{e,t}$ in $\mathcal{J}_K$ (modulo variable renaming), and hence we can use the following modified construction of $\mathcal{J}_K$:

- We consider only the time points $0, \ldots, n+1$. This suffices since all time points after $n$ generate equivalent constraint sets due to the fact that they only depend on the behavior of the rigid symbols at time points $0, \ldots, n$.

- We check after each rule application whether all constraint sets are satisfiable, which possible due to admissibility of $D$.

- If we construct a constraint $\Gamma_{e,t}$ set that is equivalent to $\Gamma_{e',t'}$, where $e'$ has been created before $e$ in the construction, we block the creation of new role successors for $e$ at time point $i$. The behavior of these successors would be equivalent to the behavior of the successors of $e'$ at $t'$. This condition is similar to the blocking condition of tableau algorithms for DLs $\Box$ and guarantees the termination of this procedure.

If this is successful, we know that there is an instantiation of $\mathcal{J}_K$, and it remains to verify that $\mathcal{J}_K \models_a \mathcal{K}$ holds. The proof of Lemma 3.4 suggests the following procedure for checking this for the TKB $\mathcal{K} = \langle (\mathcal{A}_i)_{i \geq 0}, \mathcal{T} \rangle$ (see also $\Box$). If functionality constraints are violated by $\mathcal{J}_K$, then this must be the case already in $\mathcal{J}_K^{(0)}$, which can be checked in polynomial time. For the disjointness constraints, we restrict $\mathcal{K}$ to the TKB $\mathcal{K}' = \langle (\mathcal{A}_i)_{i \geq 0}, \mathcal{T}' \rangle$, where $\mathcal{T}'$ is obtained from $\mathcal{T}$ by dropping all functionality and disjointness constraints. We know that $\mathcal{K}'$ is consistent since its canonical model is the same as that of $\mathcal{K}$, and hence has an instantiation, and the only other sources of inconsistencies have been removed (see the proof of Lemma 3.4). We
then answer a union of Boolean MCQs $\psi$ over $K'$: For each $\text{disj}(X_1, X_2) \in \mathcal{T}$, we include an MCQ $() \leftarrow X_1(\vec{x}, t) \land X_2(\vec{x}, t)$ in $\psi$, where $\vec{x}$ is a vector of variables of the appropriate types. By Lemmas 3.6 and 4.2 we can answer this union by rewriting each MCQ with $\text{PerfectRef}_{T', \mathcal{D}}$, and evaluating the result over $\mathcal{I}_A$. Then we know that $K$ is consistent iff the answer to this rewritten query is negative.

This brings us to the only missing piece of this approach, namely the question of how to evaluate a union of MCQs over $\mathcal{I}_A$. But this is a finite model checking problem for a first-order formula. In particular, the ABoxes $(\mathcal{A}_i)_{i \geq 0}$ can be represented in a (time-stamped) database which contains one table for each DL symbol $A$. Such a table contains an entry for an element or tuple $e$ and time point $i$ iff $A(e)$ occurs in $\mathcal{A}_i$. Furthermore, the temporal comparison atoms in MCQs can be handled by either materializing the relevant relations $t \preceq s + c$ over the active temporal domain $\{0, \ldots, n\}$ in the database, or by assuming that such relations are in-built predicates of the database management system (DBMS). The first approach, which is also adopted by [3], requires that we restrict the language a priori to a finite set of such predicates. With the second approach, the complexity of our approach depends on the particular implementation inside the DBMS. Similar arguments can be applied to the predicates of $\mathcal{D}$, where the first approach is followed by [4], and the second by [27].

5 Discussion and Future Work

We have shown FO rewritability of MCQs over ontologies formulated in $DL-Lite^{(HF)}_{\text{core}}(\mathcal{D})$, which is the first such result for a query language that can refer to both concrete data values and time points. It is easy to see that this holds already under atemporal semantics, i.e., when only one time point is considered, and hence we have repaired and extended the result of [27]. Additionally, we pointed out a surprising equivalence between their condition $(\text{infinitediff})$ and the convexity condition from [6].

However, the size of our rewriting is still quite large, and hence it is worth investigating alternative target languages for the rewriting, e.g., non-recursive Datalog [25], or the so-called combined approach, in which also the ABoxes may be changed [21]. Additionally, instead of explicitly considering all possible implications between relevant concrete domain predicates in the rewriting, they could be handled by integrating a dedicated solving engine for the concrete domain. Ideally, these implications could be axiomatized by Datalog rules, or another formalism that can be integrated with a database query language.

There are many directions in which our results could be extended. An easy modification would be to allow concrete domain variables and predicates in ABoxes, similar to the languages proposed in [19]; these could be handled by including them in the initialization phase of the canonical model construction. If we restrict MCQs to have exactly one temporal answer variable, we could incorporate them in the LTL-like query language of [10]; since we have constructed a canonical DL-LTL-structure respecting the rigid names, the major drawback of that paper, namely that a sequence of independent atemporal canonical models does not necessarily respect the rigid names, could be overcome. Additionally, we could try to extend the ontology language by Horn inclusions [4,11], local identification constraints (keys) [13], or (rewritable) inconsistency-tolerant semantics [18], all of which are important features for modeling real-world applications.

Acknowledgments

This work was partially supported by the DFG in the CRC 912 (HAEC). The authors wish to thank Carsten Lutz for sharing his ideas on convex concrete domains and canonical models.
References


