On the Utility of $\mathcal{CFDI}_{nc}^{\forall-}$

David Toman and Grant Weddell

Cheriton School of Computer Science
University of Waterloo, Canada
{david,gweddell}@cs.uwaterloo.ca

Abstract. We consider the description logic $\mathcal{CFDI}_{nc}^{\forall-}$, a feature-based dialect that allows capturing value restrictions, a variety of identification constraints, and unqualified feature inverses. We introduce PTIME algorithms for various reasoning tasks in this logic, such as knowledge base consistency and logical implication and discuss the necessity of restrictions over $\mathcal{CFDI}_{nc}^{\forall}$ to maintain tractability. We then show how $\mathcal{CFDI}_{nc}^{\forall-}$'s modeling capabilities make it suitable for capturing relational and object-relational data sources (including of n-ary relations) in a natural way. In addition, we show that $\mathcal{CFDI}_{nc}^{\forall-}$ can simulate reasoning in DL-Lite$_{\text{core}}^{\mathcal{F}}$. We also discuss an approach to capturing a limited variant of role hierarchies within $\mathcal{CFDI}_{nc}^{\forall-}$.

1 Introduction

We have been developing the $\mathcal{CFD}$ family of feature-based *description logic* (DL) dialects that are designed primarily to support efficient PTIME reasoning services about object relational data sources. The dialects are notable for their ability to support terminological cycles with universal restrictions over functional roles together with a rich variety of functional constraints such as keys and functional dependencies over functional role paths.

The dialect $\mathcal{CFD}$ was the first member of this family, initially proposed in [8]. In [16], the authors show that reasoning about logical consequence remains in PTIME when concept conjunction is allowed on left-hand-sides of inclusion dependencies, but that this is no longer the case should a variety of other concept constructors also be allowed. In particular, it was shown that adding inverse features in posed questions alone made reasoning about logical consequence in $\mathcal{CFD}$ intractable.

The dialect $\mathcal{CFD}_{nc}$ was subsequently introduced in which negation on right-hand-sides of inclusion dependencies replaced the ability to have conjunction on left-hand-sides [17]. This allowed the capture of so-called disjointness constraints, and also made it possible to support additional reasoning services in PTIME, notably conjunctive query answering. These results generalize to $\mathcal{CFD}_{nc}^{\forall}$ knowledge bases in which universal restrictions are also permitted on left-hand-sides of inclusion dependencies [18].

An earlier version of this paper has appeared as [19].

In this paper, we consider the dialect $\mathcal{CFDI}_{nc}^{\forall}$ which extends $\mathcal{CFD}_{nc}^{\forall}$ with an ability to have unqualified inverse features in inclusion dependencies, and also introduce a less general dialect $\mathcal{CFDI}_{nc}^{\forall-}$ in which a given $\mathcal{CFDI}_{nc}^{\forall}$ TBox is presumed to satisfy additional syntactic restrictions. The restrictions relate to combinations of value restrictions and inverses and to combinations of value restrictions and path functional dependencies. A Summary of our main results concerning reasoning in $\mathcal{CFDI}_{nc}^{\forall-}$, in particular PTIME algorithms for both logical consequence and for knowledge base consistency for $\mathcal{CFDI}_{nc}^{\forall-}$ knowledge bases.

For the remainder the paper, we give an overview of a number of applications of $\mathcal{CFDI}_{nc}^{\forall-}$, starting with how it can be used to address issues relating to relational data sources over database schema that can include arbitrary combinations of functional dependencies and unary inclusion dependencies. We also show how the task of evaluating instance queries over RDF data sources based on a DL-Lite$_{\text{core}}^{\mathcal{F}}$ entailment regime can be reduced to reasoning about $\mathcal{CFDI}_{nc}^{\forall-}$ knowledge base consistency. Note that the DL dialect DL-Lite$_{\text{core}}^{\mathcal{F}}$ is of particular relevance to the W3C OWL 2 QL profile. A discussion of related work and future directions then follow in Section 6.

2 The Description Logics $\mathcal{CFDI}_{nc}^{\forall}$ and $\mathcal{CFDI}_{nc}^{\forall-}$

All members of the $\mathcal{CFD}$ family of DLs are fragments of FOL with underlying signatures based on disjoint sets of unary predicate symbols called *primitive concepts*, constant symbols called *individuals* and unary function symbols called *features*. Note that incorporating features deviates from normal practice to use binary predicate symbols called *roles*. However, as we shall see, features make it easier to incorporate concept constructors suited to the capture of relational data sources that include various dependencies by a straightforward reification of n-ary predicates. Thus, e.g., a role R can be reified as a primitive concept R_C and two features $domR$ and $ranR$ in $\mathcal{CFDI}_{nc}^{\forall}$ or $\mathcal{CFDI}_{nc}^{\forall-}$, and an inclusion dependency $\mathrm{A} \sqsubseteq \forall R.\mathrm{B}$ can then be captured as an inclusion dependency $\forall domR.\mathrm{A} \sqsubseteq \forall ranR.\mathrm{B}$.

Definition 1 **($\mathcal{CFDI}_{nc}^{\forall}$ Knowledge Bases)** Let F, PC and IN be disjoint sets of (names of) features, primitive concepts and individuals, respectively. A *path function* Pf is a word in F^* with the usual convention that the empty word is denoted by *id* and concatenation by ".". *Concept descriptions* C and D are defined by the grammars on the left-hand-side of Figure 1 in which occurrences of "A" denote primitive concepts. A concept "$\mathrm{C} : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \rightarrow \mathsf{Pf}$" produced by the last production of the grammar for D is called a *path functional dependency* (PFD).

Metadata and data in a $\mathcal{CFDI}_{nc}^{\forall}$ knowledge base $\mathcal{K}$ are respectively defined by a *TBox* $\mathcal{T}$ and an *ABox* $\mathcal{A}$. Assume $\mathrm{A} \in \mathsf{PC}$, C and D are arbitrary concepts given by the grammars in Figure 1, $\{\mathsf{Pf}_1, \mathsf{Pf}_2\} \subseteq \mathsf{F}^*$ and that $\{a, b\} \subseteq \mathsf{IN}$. Then $\mathcal{T}$ consists of a finite set of *inclusion dependencies* of the form $\mathrm{C} \sqsubseteq \mathrm{D}$, and $\mathcal{A}$

Syntax	Semantics: "$(\cdot)^{\mathcal{I}}$"
$\mathrm{C} ::= \mathrm{A}$	$\mathrm{A}^{\mathcal{I}} \subseteq \triangle$
$\mid \forall \mathsf{Pf}.\mathrm{C}$	$\{x \mid \mathsf{Pf}^{\mathcal{I}}(x) \in \mathrm{C}^{\mathcal{I}}\}$
$\mid \exists f^{-1}$	$\{x \mid \exists y \in \triangle : f^{\mathcal{I}}(y) = x\}$
$\mathrm{D} ::= \mathrm{C}$	$\mathrm{C}^{\mathcal{I}} \subseteq \triangle$
$\mid \neg \mathrm{C}$	$\triangle \setminus \mathrm{C}^{\mathcal{I}}$
$\mid \forall \mathsf{Pf}.\mathrm{D}$	$\{x \mid \mathsf{Pf}^{\mathcal{I}}(x) \in \mathrm{D}^{\mathcal{I}}\}$
$\mid \exists f^{-1}$	$\{x \mid \exists y \in \triangle : f^{\mathcal{I}}(y) = x\}$
$\mid \mathrm{C} : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \to \mathsf{Pf}$	$\{x \mid \forall y \in \mathrm{C}^{\mathcal{I}} : (\bigwedge_{i=1}^{k} \mathsf{Pf}_i^{\mathcal{I}}(x) = \mathsf{Pf}_i^{\mathcal{I}}(y)) \Rightarrow \mathsf{Pf}^{\mathcal{I}}(x) = \mathsf{Pf}^{\mathcal{I}}(y)\}$

Fig. 1. $\mathcal{CFDI}_{nc}^{\forall}$ Concepts.

consists of a finite set of facts in the form of *concept assertions* $\mathrm{A}(a)$, and *path function assertions* $\mathsf{Pf}_1(a) = \mathsf{Pf}_2(b)$. Any PFD occurring in $\mathcal{T}$ must also satisfy a *regularity* condition by adhering to one of the following two forms:

$$\mathrm{C} : \mathsf{Pf}\,.\,\mathsf{Pf}_1, \mathsf{Pf}_2, \ldots, \mathsf{Pf}_k \to \mathsf{Pf} \quad \text{or} \quad \mathrm{C} : \mathsf{Pf}\,.\,\mathsf{Pf}_1, \mathsf{Pf}_2, \ldots, \mathsf{Pf}_k \to \mathsf{Pf}\,.g. \qquad (1)$$

A PFD is a *key* if it adheres to the first of these forms.

Semantics is defined in the standard way with respect to an interpretation $\mathcal{I} = (\triangle, (\cdot)^{\mathcal{I}})$, where $\triangle$ is a domain of "objects" and $(\cdot)^{\mathcal{I}}$ an interpretation function that fixes the interpretation of primitive concepts A to be subsets of $\triangle$, features f to be total functions on $\triangle$, and individuals a to be elements of $\triangle$. The interpretation function is extended to path expressions by interpreting id, the empty word, as the identity function $\lambda x.x$, concatenation as function composition, and to derived concept descriptions C or D as defined in Figure 1.

An interpretation $\mathcal{I}$ satisfies an inclusion dependency $\mathrm{C} \sqsubseteq \mathrm{D}$ if $\mathrm{C}^{\mathcal{I}} \subseteq \mathrm{D}^{\mathcal{I}}$, a concept assertion $\mathrm{A}(a)$ if $a^{\mathcal{I}} \in \mathrm{A}^{\mathcal{I}}$, and a path function assertion $\mathsf{Pf}_1(a) = \mathsf{Pf}_2(b)$ if $\mathsf{Pf}_1^{\mathcal{I}}(a^{\mathcal{I}}) = \mathsf{Pf}_2^{\mathcal{I}}(b^{\mathcal{I}})$. $\mathcal{I}$ satisfies a knowledge base $\mathcal{K}$ if it satisfies each inclusion dependency and assertion in $\mathcal{K}$. □

Condition (1) on occurrences of the PFD concept constructor distinguish, e.g., PFDs of the form $\mathrm{C} : f \to id$ and $\mathrm{C} : f \to g$ from PFDs of the form $\mathrm{C} : f \to g.h$, and are necessary on $\mathcal{CFD}$ alone to avoid both intractability of reasoning about logical consequence [9] and undecidability of reasoning about KB consistency [15]. Conversely, and as usual, allowing conjunction (resp. disjunction) on the right-hand-sides (resp. left-hand-sides) of inclusion dependencies is a simple syntactic sugar.

Finally, note that $\mathcal{CFDI}_{nc}^{\forall}$ does not assume the unique name assumption, but that its ability to express disjointness enables mutual inequality between all pairs of n individuals to be captured by introducing $O(n)$ new atomic concepts, concepts assertions and inclusion dependencies in a straightforward way.

Lemma 2 ($\mathcal{CFDI}_{nc}^{\forall}$ KB Normal Form) For every KB $(\mathcal{T}, \mathcal{A})$, there is an equi-satisfiable KB $(\mathcal{T}', \mathcal{A}')$ in which subsumptions in $\mathcal{T}'$ adhere to the following forms:

$$\mathrm{A} \sqsubseteq \mathrm{B},\ \ \mathrm{A} \sqsubseteq \forall f.\mathrm{B},\ \ \forall f.\mathrm{A} \sqsubseteq \mathrm{B},\ \ \mathrm{A} \sqsubseteq \exists f^{-1},\ \text{or } \mathrm{A} \sqsubseteq \mathrm{A}' : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \to \mathsf{Pf},$$

where A and A′ are primitive concepts and B is a primitive concept or a negation of a primitive concept, and where ABox $\mathcal{A}'$ contains only assertions of the form $\mathrm{A}(a)$, $f(a) = b$, and $a = b$. □

Obtaining $\mathcal{T}'$ and $\mathcal{A}'$ from an arbitrary knowledge base $\mathcal{K}$ that are linear in the size of $\mathcal{K}$ is easily achieved by a straightforward conservative extension using auxiliary names for intermediate concept descriptions and individuals. For further details, see the definition of *simple concepts* in [13, 15].

Hereon, we identify $\neg\forall\,\mathsf{Pf}\,.A$ with $\forall\,\mathsf{Pf}\,.\neg A$, and say that $\forall\,\mathsf{Pf}\,.\mathrm{A}$ and $\forall\,\mathsf{Pf}\,.\neg\mathrm{A}$ are *complementary* for $\mathsf{Pf} \in \mathsf{F}^*$. Also, when a particular KB $(\mathcal{T}, \mathcal{A})$ is considered, we assume the sets PC and F contain symbols that appear in $\mathcal{T}$ and $\mathcal{A}$ only.

Unfortunately, use unqualified inverse features make reasoning about logical consequence over an arbitrary $\mathcal{CFDI}_{nc}^{\forall}$ KB $\mathcal{K}$ intractable [19]. To recover PTIME reasoning for both logical implication and KB consistency, $\mathcal{K}$ will need to satisfy additional syntactic restrictions.

Definition 3 ($\mathcal{CFDI}_{nc}^{\forall -}$ Knowledge Bases) A $\mathcal{CFDI}_{nc}^{\forall -}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is a $\mathcal{CFDI}_{nc}^{\forall}$ KB in normal form that satisfies the following two conditions.

1. (*inverse feature and value restriction interaction*) If $\{\mathrm{A} \sqsubseteq \exists f^{-1}, \forall f.\mathrm{A}' \sqsubseteq \mathrm{B}\} \subseteq \mathcal{T}$ then (a) $\mathrm{A} \sqsubseteq \mathrm{A}' \in \mathcal{T}$, (b) $\mathrm{A}' \sqsubseteq \mathrm{A} \in \mathcal{T}$ or (c) $\mathrm{A} \sqsubseteq \neg\mathrm{A}' \in \mathcal{T}$.

2. (*inverse feature and PFD interaction*) Any PFD occurring in $\mathcal{T}$ must also satisfy a *stronger* regularity condition by adhering to one of the following two forms:

$$\mathrm{C} : \mathsf{Pf}\,.\,\mathsf{Pf}_1, \mathsf{Pf}_2, \ldots, \mathsf{Pf}_k \to \mathsf{Pf} \quad \text{or} \quad \mathrm{C} : \mathsf{Pf}\,.f, \mathsf{Pf}_2, \ldots, \mathsf{Pf}_k \to \mathsf{Pf}\,.g. \qquad (2)$$

Relaxing either of the conditions leads to EXPTIME and PSPACE completeness, respectively [19]. Note also, that the additional condition (2) imposed on PFDs applies only to non-key PFDs. Overall, however, such restrictions do not seem to impact the modeling utility of $\mathcal{CFDI}_{nc}^{\forall}$ in relation to keys and functional constraints. Indeed, we show that arbitrary functional dependencies in relational schema are easily captured.

3 $\mathcal{CFDI}_{nc}^{\forall -}$ TBoxes and Concept Satisfiability

It is easy to see that every $\mathcal{CFDI}_{nc}^{\forall -}$ TBox $\mathcal{T}$ is consistent (by setting all primitive concepts to be interpreted as the empty set). To test for *(primitive) concept satisfiability* we use the following construction:

Definition 4 (TBox Closure) Let $\mathcal{T}$ be a $\mathcal{CFDI}_{nc}^{\forall -}$ TBox in normal form. We define $\mathsf{Clos}(\mathcal{T})$ to be the least set of subsumptions that contains $\mathcal{T}$ and is closed under the following five inference rules:

1. $C_1 \sqsubseteq C_1 \in \mathsf{Clos}(\mathcal{T})$;
2. If $\{C_1 \sqsubseteq C_2, C_2 \sqsubseteq C_3\} \subseteq \mathsf{Clos}(\mathcal{T})$, then $C_1 \sqsubseteq C_3 \in \mathsf{Clos}(\mathcal{T})$;
3. If $\{C_1 \sqsubseteq D_1, C_2 \sqsubseteq D_2\} \subseteq \mathsf{Clos}(\mathcal{T})$ and D_1 and D_2 are complementary, then $C_1 \sqsubseteq \neg C_2 \in \mathsf{Clos}(\mathcal{T})$;
4. If $\mathrm{A} \sqsubseteq \mathrm{B} \in \mathsf{Clos}(\mathcal{T})$, then $\forall f.\mathrm{A} \sqsubseteq \forall f.\mathrm{B} \in \mathsf{Clos}(\mathcal{T})$; and
5. If $\{\mathrm{A} \sqsubseteq \exists f^{-1}, \forall f.\mathrm{A}' \sqsubseteq \forall f.\mathrm{B}, \mathrm{A} \sqsubseteq \mathrm{A}'\} \subseteq \mathsf{Clos}(\mathcal{T})$, then $\mathrm{A} \sqsubseteq \mathrm{B} \in \mathsf{Clos}(\mathcal{T})$,

where A is a primitive concept, B is a primitive concept or its negation, and where C_1, C_2, D_1, and D_2 are subconcepts of concepts in $\mathcal{T}$ or their negations. □

Note that $\mathsf{Clos}(\mathcal{T})$ is at most quadratic in $|\mathcal{T}|$. It is also easy to verify that each inclusion added to $\mathsf{Clos}(\mathcal{T})$ by the inferences (1-4) in Definition 4 is logically implied by $\mathcal{T}$. Also, a variant of the closure rule (5) is not needed for the case $\mathrm{A}' \sqsubseteq \mathrm{A}$ since we also have $\mathrm{A}' \sqsubseteq \exists f^{-1}$, nor it is needed in the case $\mathrm{A}' \sqsubseteq \neg\mathrm{A}$ since, in this case, the value restriction in the rule is satisfied vacuously.

Given $\mathsf{Clos}(\mathcal{T})$, an object o, and a primitive concept A, we define the following family of subsets of PC indexed by paths of features (and their inverses), starting from o, as follows:

1. $S_o = \{\mathrm{B} \mid \mathrm{A} \sqsubseteq \mathrm{B} \in \mathsf{Clos}(\mathcal{T})\}$;
2. $S_{f(x)} = \{\mathrm{B} \mid \mathrm{A} \sqsubseteq \forall f.\mathrm{B} \in \mathsf{Clos}(\mathcal{T})$ and $\mathrm{A} \in S_x\}$, when $f \in \mathsf{F}$ and x not of the form "$f^-(y)$"; and
3. $S_{f^-(x)} = \{\mathrm{B} \mid \forall f.\mathrm{A} \sqsubseteq \mathrm{B}$ and $\mathrm{A} \in S_x\}$, when $\mathrm{A}' \sqsubseteq \exists f^{-1} \in \mathsf{Clos}(\mathcal{T})$, $\mathrm{A}' \in S_x$, and x not of the form "$f(y)$".

We say that S_x is *defined* if it conforms to one of the three above cases, and that it is *consistent* if, whenever $\{\mathrm{A}, \mathrm{A}'\} \subseteq S_x$, $\mathrm{A} \sqsubseteq \neg\mathrm{A}' \notin \mathsf{Clos}(\mathcal{T})$.

Theorem 5 (Primitive Concept Satisfiability) Let $\mathcal{T}$ be a $\mathcal{CFDI}_{nc}^{\forall-}$ TBox in normal form and A a primitive concept description. Then A is satisfiable with respect to T if and only if $\mathrm{A} \sqsubseteq \neg\mathrm{A} \notin \mathsf{Clos}(\mathcal{T})$.

Proof (sketch): We build a model of $\mathcal{T}$ in which $o \in \mathrm{A}^{\mathcal{I}}$ for some $o \in \triangle$ as follows:

- $\triangle = \{x \mid S_x \text{ is defined}\}$;
- $f^{\mathcal{I}} = \{(x, f(x)) \mid S_{f(x)} \text{ is defined}\} \cup \{(f^-(x), x) \mid S_{f^-(x)} \text{ is defined}\}$; and
- $\mathrm{A}^{\mathcal{I}} = \{x \mid S_x \text{ is defined}, \mathrm{A} \in S_x\}$.

It is easy to see that, due to closure rules in Definition 4, all the defined sets S_x must be consistent. Otherwise, A ($\in S_0$) must be inconsistent, implying in turn that $\mathrm{A} \sqsubseteq \neg\mathrm{A} \in \mathsf{Clos}(\mathcal{T})$, a contradiction. Hence, $\mathcal{I} = (\triangle, .^{\mathcal{I}})$ is a model of $\mathcal{T}$ (it satisfies all dependencies in $\mathsf{Clos}(\mathcal{T})$) such that $o \in \mathrm{A}^{\mathcal{I}}$. □

Note that the model witnessing satisfiability of A does not contain any identical path agreements (beyond the trivial $id = id$) and hence vacuously satisfies all PFDs in $\mathcal{T}$.

The above theorem can be used to check satisfiability of complex (non-PFD) concepts; e.g., satisfiability of $\forall \mathsf{Pf} . B$ w.r.t. $\mathcal{T}$ can be tested by checking satisfiability of a new primitive concept A w.r.t. $\mathcal{T} \cup \{\mathrm{A} \sqsubseteq \forall \mathsf{Pf} . B\}$. It also provides a technique for checking satisfiability of finite conjunctions of primitive concepts with respect to $\mathcal{T}$:

Corollary 6 Let $\mathcal{T}$ be a $\mathcal{CFDI}_{nc}^{\forall -}$ TBox in normal form and $\mathrm{A}_1, \ldots, \mathrm{A}_\mathrm{k}$ primitive concepts. Then $\mathrm{A}_1 \sqcap \ldots \sqcap \mathrm{A}_\mathrm{k}$ is satisfiable with respect to $\mathcal{T}$ if and only if A is satisfiable with respect to $\mathcal{T} \cup \{\mathrm{A} \sqsubseteq \mathrm{A}_1, \ldots, \mathrm{A} \sqsubseteq \mathrm{A}_\mathrm{k}\}$, for A a fresh primitive concept. □

4 Knowledge Base Consistency and Logical Implication

We start with the problem of determining if a given $\mathcal{CFDI}_{nc}^{\forall -}$ knowledge base is consistent. This is resolved in a straightforward way with the following notion of an *interesting* path function and the subsequent definition of an ABox completion procedure.

Definition 7 Let $\mathcal{T}$ be a $\mathcal{CFDI}_{nc}^{\forall -}$ TBox. We say that a path function Pf is *interesting in* $\mathcal{T}$ if it is a common prefix of all Pf_i in a PFD $\mathrm{A} \sqsubseteq \mathrm{B} : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \rightarrow \mathsf{Pf} \in \mathcal{T}$. □

Definition 8 Let $(\mathcal{T}, \mathcal{A})$ be a $\mathcal{CFDI}_{nc}^{\forall -}$ knowledge base. We define an ABox $\mathsf{completion}_{\mathcal{T}}(\mathcal{A})$ to be the least ABox $\mathcal{A}'$ such that $\mathcal{A} \subseteq \mathcal{A}'$ and $\mathcal{A}'$ is closed under the rules in Figure 2. □

Note that since $\mathcal{A}$ is in normal form, individuals can only be declared to be members of primitive concepts. Thus, a $\mathcal{CFDI}_{nc}^{\forall -}$ ABox alone cannot lead to inconsistency. Only when combined with a TBox does it become possible that certain conjunctions of primitive concepts must interpret as empty in every model, thus leading to KB inconsistency. This observation combined with Corollary 6 yields the following theorem:

Theorem 9 ($\mathcal{CFDI}_{nc}^{\forall -}$ KB consistency) Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a $\mathcal{CFDI}_{nc}^{\forall -}$ KB (in normal form). Then $\mathcal{K}$ is consistent if and only if $\{\mathrm{A} \mid \mathrm{A}(a) \in \mathsf{completion}_{\mathcal{T}}(\mathcal{A})\}$ is satisfiable with respect to $\mathsf{Clos}(\mathcal{T})$ for all objects a in $\mathcal{A}$. □

It is easy to verify that constructing $\mathsf{Clos}(\mathcal{T})$ and $\mathsf{completion}_{\mathcal{T}}(\mathcal{A})$ is polynomial in $|\mathcal{K}|$, and that testing for consistency implicitly contains Horn-SAT due to the presence of PFDs. Thus, we have the following:

Corollary 10 Consistency of $\mathcal{CFDI}_{nc}^{\forall -}$ knowledge bases is complete for PTIME. □

ABox Equality Rules:

1. If $\{a = b, b = c\} \subseteq \mathcal{A}$, then $a = c \in \mathcal{A}$.
2. If $\{f(a) = b, b = c\} \subseteq \mathcal{A}$, then $f(a) = c \in \mathcal{A}$.
3. If $\{a = b, f(b) = c\} \subseteq \mathcal{A}$, then $f(a) = c \in \mathcal{A}$.
4. If $\{f(a) = b, f(a) = c\} \subseteq \mathcal{A}$, then $b = c \in \mathcal{A}$.
5. If $\{a = b, \mathrm{A}(a)\} \subseteq \mathcal{A}$, then $\mathrm{A}(b) \in \mathcal{A}$.

ABox–TBox Interactions:

6. If $\mathrm{A}(a) \in \mathcal{A}$ and $\mathrm{A} \sqsubseteq \mathrm{B} \in \mathsf{Clos}(\mathcal{T})$, then $\mathrm{B}(a) \in \mathcal{A}$.
7. If $\{\mathrm{A}(a), f(a) = b\} \subseteq \mathcal{A}$ and $\mathrm{A} \sqsubseteq \forall f.\mathrm{B} \in \mathsf{Clos}(\mathcal{T})$, then $\mathrm{B}(b) \in \mathcal{A}$.
8. If $\{\mathrm{A}(a), f(b) = a\} \subseteq \mathcal{A}$ and $\forall f.\mathrm{A} \sqsubseteq \mathrm{B} \in \mathsf{Clos}(\mathcal{T})$, then $\mathrm{B}(b) \in \mathcal{A}$.

ABox–Inverse Interactions:

9. If $\mathsf{Pf} = f_1 f_2 \cdots f_k$ is interesting in $\mathcal{T}$, $\mathrm{A}_0(a_0) \in \mathcal{A}$, a_0 is an object in the original ABox $\mathcal{A}$, and $\{\mathrm{A}_{\mathrm{i}-1} \sqsubseteq \exists {f_i}^{-1}, \forall f_i.\mathrm{A}_{i-1} \sqsubseteq \mathrm{A}_i\} \subseteq \mathsf{Clos}(\mathcal{T})$ for $0 < i \leq k$, then $\{\mathrm{A}_i(a_i), f_i(a_{i-1}) = a_i\} \subseteq \mathcal{A}$.

ABox–PFD Interactions:

10. If $\{A(a), B(b)\} \subseteq \mathcal{A}$, $\{\mathsf{Pf}'_i(a) = c_i, \mathsf{Pf}'_i(b) = c_i\} \subseteq \mathcal{A}$ for $0 < i \leq k$, and $\mathrm{A} \sqsubseteq \mathrm{B} : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \rightarrow \mathsf{Pf} \in \mathcal{T}$ such that Pf'_i is a prefix of Pf_i, then
    (a) $\{\mathsf{Pf}'(a) = c, \mathsf{Pf}'(b) = c\} \subseteq \mathcal{A}$ for Pf' a prefix of Pf,
    (b) If $\{\mathsf{Pf}(a) = c, \mathsf{Pf}(b) = d\} \subseteq \mathcal{A}$, then $c = d \in \mathcal{A}$, or
    (c) If Pf is of the form $\mathsf{Pf}''.f$ and $\{\mathsf{Pf}''(a) = c, \mathsf{Pf}''(b) = d\} \subseteq \mathcal{A}$, then $f(c) = e$ and $f(d) = e$ to $\mathcal{A}$ for a new individual e.

Fig. 2. ABox Completion Rules.

It is also straightforward to reduce logical implication for a $\mathcal{CFDI}_{nc}^{\forall-}$ TBox $\mathcal{T}$ to knowledge base consistency. Indeed, subsumptions between literals are directly present in $\mathsf{Clos}(\mathcal{T})$. Logical implication involving more general concept descriptions, such as PFDs, is reduced to knowledge base (in)consistency by encoding a counterexample as an ABox.

Theorem 11 Logical consequence for $\mathcal{CFDI}_{nc}^{\forall-}$ terminologies is complete for PTIME. $\square$

5 Applications

We now introduce two major applications for $\mathcal{CFDI}_{nc}^{\forall-}$: capture of relational (and object-relational) database schemas and its ability to fully simulate DL-Lite$_{\text{core}}^{\mathcal{F}}$. We also show how *role hierarchies* can be partially accommodated by $\mathcal{CFDI}_{nc}^{\forall-}$.

5.1 Relational Data Sources and BCNF

There are a number of applications of $\mathcal{CFDI}_{nc}^{\forall-}$ in addressing issues that surface with relational data sources. To illustrate, we show how a relational schema (S, Σ) with relation symbols S and with functional dependencies and unary foreign keys Σ can be easily mapped to a $\mathcal{CFDI}_{nc}^{\forall-}$ terminology $\mathcal{T}_{(S,\Sigma)}$, and then exhibit a straightforward reduction of so-called *Boyce-Codd normal form* (BCNF) diagnosis to logical consequence over $\mathcal{T}_{(S,\Sigma)}$.

First the mapping: each $R(A_1 : D_1, \ldots, A_k : D_k)$ in S (i.e., a relation of arity k) is reified by mapping to the following inclusion dependencies in $\mathcal{T}_{(S,\Sigma)}$:

$$\mathrm{C}_R \sqsubseteq \mathrm{C}_R : a_{R.A_1}, \ldots, a_{R.A_k} \to id \text{ and}$$
$$\mathrm{C}_R \sqsubseteq \forall a_{R.A_i}.D_i, \text{ for each } 0 < i \leq k,$$

where (a) C_R is a primitive concept for which an interpretation will be the *tuple objects* that correspond to tuples in R, (b) $a_{R.A_i}$ are features that yield values of fields in such tuples, and (c) D_i are primitive concepts standing for the *domains* of the features. In addition, for each pair of $R, R' \in S$, add to $\mathcal{T}_{(S,\Sigma)}$

$$\mathrm{C}_R \sqsubseteq \neg\mathrm{C}_{R'}.$$

Each functional dependency $R : A_{i_1}, \ldots, A_{i_k} \to A_{i_0}$ in Σ is then mapped to an inclusion dependency:

$$\mathrm{C}_R \sqsubseteq \mathrm{C}_R : a_{R.A_{i_1}}, \ldots, a_{R.A_{i_k}} \to a_{R.A_{i_0}},$$

and each unary inclusion dependency $R[A] \subseteq R'[A']$ in Σ to three inclusion dependencies, where A is a fresh primitive concept unique to $R[A] \subseteq R'[A']$:

$$\mathrm{C}_R \sqsubseteq \forall a_{R.A}.\mathrm{A},$$
$$\mathrm{A} \sqsubseteq \exists a_{R'.A'}{}^{-1}, \text{ and}$$
$$\forall a_{R'.A'}.\mathrm{A} \sqsubseteq \mathrm{C}_{R'}.$$

BCNF diagnosis then translates to logical consequence in $\mathcal{CFDI}_{nc}^{\forall-}$ in a straightforward fashion:

Theorem 12 (Diagnosing BCNF for Relational Data Sources) Each relation R in (S, Σ) is in BCNF iff there is no set of features $\{a_{R.A_{i_0}}, \ldots, a_{R.A_{i_k}}\}$ in $\mathcal{T}_{(S,\Sigma)}$ such that

$$\mathcal{T}_{(S,\Sigma)} \models \mathrm{C}_R \sqsubseteq \mathrm{C}_R : a_{R.A_{i_1}}, \ldots, a_{R.A_{i_k}} \to a_{R.A_0}$$

but where

$$\mathcal{T}_{(S,\Sigma)} \not\models \mathrm{C}_R \sqsubseteq \mathrm{C}_R : a_{R.A_{i_1}}, \ldots, a_{R.A_{i_k}} \to id\,.$$

□

Note that this easily generalizes to the object-relational setting where the domain D_i of an attribute may now refer directly to R_i-objects, and where a generalization of binary decompositions called *pivoting* is the means of repairing violations of BCNF [3, 4].

An application in query optimization over relational data sources relates to SQL `distinct`-keyword elimination, that is, detecting where operations in query plans to remove duplicates can be safely eliminated [8]. Such rewrites can be reduced to knowledge based consistency problems in $\mathcal{CFDI}_{nc}^{\forall -}$ by using an ABox to encode simple selection conditions in SQL queries [7].

5.2 Encoding of DL-Lite

Another application of $\mathcal{CFDI}_{nc}^{\forall -}$ in a different setting relates to the problem of evaluating *basic graph patterns* in SPARQL with a presumed entailment regime defined by DL-Lite$_{\text{core}}^{\mathcal{F}}$, a DL dialect that is related to the W3C OWL 2 QL profile. Such tasks reduce fundamentally to instance checking problems which reduce, in turn, to knowledge base consistency problems in the standard way.

Our reduction is based on mapping a given DL-Lite$_{\text{core}}^{\mathcal{F}}$ knowledge base $\mathcal{K}$ to a $\mathcal{CFDI}_{nc}^{\forall -}$ knowledge base $M_{\mathcal{K}}$ as follows: for each role P in $\mathcal{K}$, we reify P by introducing a new primitive concept C_P and by adding the following key PFD to $M_{\mathcal{K}}$:

$$\mathrm{C}_P \sqsubseteq \mathrm{C}_P : domP, ranP \to id\,.$$

The following rules define the mapping of each inclusion dependency in $\mathcal{K}$ (in normal form [2]) and each ABox assertion in $\mathcal{K}$ to corresponding dependencies and assertions in $M_{\mathcal{K}}$:

$$\begin{array}{lll}
\mathrm{A}_1 \sqsubseteq \mathrm{A}_2 & \mapsto & \{\mathrm{A}_1 \sqsubseteq \mathrm{A}_2\}, \\
\mathrm{A}_1 \sqsubseteq \neg\mathrm{A}_2 & \mapsto & \{\mathrm{A}_1 \sqsubseteq \neg\mathrm{A}_2\}, \\
\mathrm{A}_1 \sqsubseteq \exists P & \mapsto & \{\mathrm{A}_1 \sqsubseteq \exists domP^{-1}, \forall domP.\mathrm{A}_1 \sqsubseteq \mathrm{C}_P\}, \\
\mathrm{A}_1 \sqsubseteq \exists P^- & \mapsto & \{\mathrm{A}_1 \sqsubseteq \exists ranP^{-1}, \forall ranP.\mathrm{A}_1 \sqsubseteq \mathrm{C}_P\}, \\
\exists P \sqsubseteq \mathrm{A}_1 & \mapsto & \{\mathrm{C}_P \sqsubseteq \forall domP.\mathrm{A}_1\}, \\
\exists P^- \sqsubseteq \mathrm{A}_1 & \mapsto & \{\mathrm{C}_P \sqsubseteq \forall ranP.\mathrm{A}_1\}, \\
(\mathsf{func}\ P) & \mapsto & \{\mathrm{C}_P \sqsubseteq \mathrm{C}_P : domP \to id\}, \\
(\mathsf{func}\ P^-) & \mapsto & \{\mathrm{C}_P \sqsubseteq \mathrm{C}_P : ranP \to id\}, \\
a : \mathrm{A} & \mapsto & \{a : \mathrm{A}\} \text{ and} \\
P(a,b) & \mapsto & \{c_{a,b}^P : \mathrm{C}_P, domP(c_{a,b}^P) = a, ranP(c_{a,b}^P) = b\}.
\end{array}$$

This mapping yields the following as a straightforward consequence:

Theorem 13 (DL-Lite$_{\text{core}}^{\mathcal{F}}$ Reasoning) Let $\mathcal{K}$ be a DL-Lite$_{\text{core}}^{\mathcal{F}}$ KB. Then knowledge base consistency, logical implication, and instance checking with respect to $\mathcal{K}$ can be reduced to reasoning about KB consistency with respect to $M_{\mathcal{K}}$. □

On Role Hierarchies The reduction above is only defined for DL-Lite$_{\text{core}}^{\mathcal{F}}$. Indeed, it is well known that an (unrestricted) combination *functionality* with *role hierarchies*, e.g., DL-Lite$_{\text{core}}^{\mathcal{HF}}$, leads to intractability [2]. On the other hand, the ability to reify roles seems to allow to capture a limited version of *role hierarchies*[1].

Example 14 Consider roles R and S and the corresponding primitive concepts C_R and C_S, respectively. In contrast to the development in the previous section, we assume that the domains and ranges of the reified roles are captured by the feature *dom* and *ran* (common to both the reified roles). Then we can capture subsumption and disjointness of these roles as follows:

$$\begin{array}{lll} R \sqsubseteq S & \mapsto & \mathrm{C}_R \sqsubseteq \mathrm{C}_S, \mathrm{C}_R \sqsubseteq \mathrm{C}_S : \mathit{dom}, \mathit{ran} \rightarrow \mathit{id}, \\ R \sqcap S \sqsubseteq \bot & \mapsto & \mathrm{C}_R \sqsubseteq \neg \mathrm{C}_S, \mathrm{C}_R \sqsubseteq \mathrm{C}_S : \mathit{dom}, \mathit{ran} \rightarrow \mathit{id}, \end{array}$$

assuming that the reified role R (and analogously S) also satisfies the key constraint $\mathrm{C}_R \sqsubseteq \mathrm{C}_R : \mathit{dom}, \mathit{ran} \rightarrow \mathit{id}$. Role typing is achieved in a way analogous to DL-Lite$^{\mathcal{F}}_{\text{core}}$.

Note, however, that such a reduction does *not* lend itself to capturing role hierarchies between roles and *inverses* of roles: this is due to fixing the (names of the) features *dom* and *ran*. Moreover, the condition introduced in Definition 3(1), that governs the interactions between inverse features and value restrictions, introduces additional interactions that interfere with (simulating) role hierarchies, in particular in cases when *mandatory participation* constraints are present.

Example 15 Consider roles R_1 and R_2 and the corresponding primitive concepts C_{R_1} and C_{R_2}, respectively, and associated constraints that declare typing for the roles,

$$\begin{array}{l} \mathrm{C}_{R_1} \sqsubseteq \forall \mathit{dom}.\mathrm{A}_1, \mathrm{C}_{R_1} \sqsubseteq \forall \mathit{ran}.\mathrm{B}_1, \mathrm{C}_{R_1} \sqsubseteq \mathrm{C}_{R_1} : \mathit{dom}, \mathit{ran} \rightarrow \mathit{id} \\ \mathrm{C}_{R_2} \sqsubseteq \forall \mathit{dom}.\mathrm{A}_2, \mathrm{C}_{R_2} \sqsubseteq \forall \mathit{ran}.\mathrm{B}_2, \mathrm{C}_{R_2} \sqsubseteq \mathrm{C}_{R_2} : \mathit{dom}, \mathit{ran} \rightarrow \mathit{id} \end{array}$$

originating, e.g., from an ER diagram postulating that entity sets A_i and B_i participate in a relationship R_i (for $i = 1, 2$). Now consider a situation where the participation of A_i in R_i is *mandatory* (expressed, e.g., as $\mathrm{A}_i \sqsubseteq \exists R_i$ in DL-Lite). This leads to the following constraints:

$$\mathrm{A}_1 \sqsubseteq \exists \mathit{dom}^{-1}, \forall \mathit{dom}.\mathrm{A}_1 \sqsubseteq \mathrm{C}_{R_1} \text{ and } \mathrm{A}_2 \sqsubseteq \exists \mathit{dom}^{-1}, \forall \mathit{dom}.\mathrm{A}_2 \sqsubseteq \mathrm{C}_{R_2}.$$

Condition (1) in Definition 3 then requires that one of

$$\mathrm{A}_1 \sqsubseteq \mathrm{A}_2, \mathrm{A}_2 \sqsubseteq \mathrm{A}_1, \text{ or } \mathrm{A}_1 \sqsubseteq \neg \mathrm{A}_2$$

are present in the TBox. The first (and second) conditions imply that $\mathrm{C}_{R_1} \sqsubseteq \mathrm{C}_{R_2}$ ($\mathrm{C}_{R_2} \sqsubseteq \mathrm{C}_{R_1}$, respectively). The third condition states that the domains of (the reified versions of) R_1 and R_2 are disjoint, hence the roles themselves must also be disjoint. Hence, in the presence of $\mathrm{C}_{R_1} \sqsubseteq \mathrm{C}_{R_2} : \mathit{dom}, \mathit{ran} \rightarrow \mathit{id}$, the concepts C_{R_1} and C_{R_2} must also be disjoint.

In this setting role hierarchies can be mapped to $\mathcal{CFDI}^{\forall}_{nc}$ are as follows:

[1] Unlike DL-Lite$^{(\mathcal{HF})}_{\text{core}}$, that restricts the applicability of functional constraints in the presence of role hierarchies, we study what forms role hierarchies can be captured while retaining the ability to specify arbitrary keys and functional dependencies.

1. only primitive roles are supported,
2. for each pair of roles participating in the same role hierarchy, either one of the roles is a super-role of the other, or the roles' domains/ranges are disjoint.

The first restriction originates in the way (binary) roles are reified—by assigning canonically-named features. This prevents modeling constraints such as $R \sqsubseteq R^-$ (which would seem to require simple equational constraints for feature renaming). The second condition is essential to maintaining tractability of reasoning [19]. Note, however, that no such restriction is needed for roles that do *not* participate in the same role hierarchy; this is achieved by appropriate choice of names for the features *dom* and *ran* similarly to the development in Section 5.2.

One can, however, model object participation in sibling roles participating in a role hierarchy using *delegation* [1], leading to a more complex translation of role assertions to $\mathcal{CFDI}_{nc}^{\forall -}$:

Example 16 Consider roles R_1, R_2, and S involved in a role hierarchy $R_1 \sqsubseteq S$ and $R_2 \sqsubseteq S$. To assert that A objects must participate in both the roles R_i, for $i \in \{1, 2\}$, we first explicitly establish the domains of the roles (the same applies for ranges of roles),

$$\mathrm{DR}_i \sqsubseteq \exists dom^{-1}, \forall dom.\mathrm{DR}_i \sqsubseteq \mathrm{C}_{R_i},\ \text{and}\ \mathrm{C}_{R_i} \sqsubseteq \forall \mathrm{DR}_i..$$

Then, instead of asserting $\mathrm{A} \sqsubseteq \mathrm{DR}_i$ (which immediately leads to inconsistency due to our PTIME restrictions on roles) we assert $\mathrm{A} \sqsubseteq \forall f_{R_i}.\mathrm{DR}_i$ where the f_{R_i} images of an A object are the *delegates* used to participate in the roles R_i.

Last, the $\mathcal{CFDI}_{nc}^{\forall}$-based approach to role hierarchies can easily be extended to handling hierarchies of non-homogeneous relationships (again, via reification and appropriate naming of features) that originate, e.g., from relating the aggregation constructs via inheritance in the EER model [10, 11].

6 Related Work and Future Directions

Toman and Weddell have also proposed the $\mathcal{DLF}$ family of feature-based Boolean-complete DL dialects obtained by allowing arbitrary use of negation in concepts [12]. In particular, they have shown that allowing inverse features in such dialects makes reasoning about logical consequence undecidable [14]. They have also shown that allowing negation on left-hand-sides of inclusion dependencies in $\mathcal{CFDI}_{nc}^{\forall}$ leads to intractability, but that PTIME algorithms exist for reasoning about logical consequence and knowledge base consistency if a number of additional conditions are satisfied [20].

A variety of path based identification constraints have been proposed [5] together with analogous applications in relational schema diagnosis [6], although $\mathcal{CFDI}_{nc}^{\forall -}$ seems to provide a more natural and transparent approach to this problem.

References

1. M. Aksit, J.W. Dijkstra, and A. Tripathi. Atomic delegation: object-oriented transactions. *Software, IEEE*, 8(2):84–92, March 1991.
2. Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The DL-Lite family and relations. *J. AI Research*, 36:1–69, 2009.
3. Joachim Biskup, Ralf Menzel, Torsten Polle, and Yehoshua Sagiv. Decomposition of Relationships through Pivoting. In *Conceptual Modeling*, pages 28–41, 1996.
4. Joachim Biskup and Torsten Polle. Decomposition of Database Classes under Path Functional Dependencies and Onto Constraints. In *Foundations of Information and Knowledge Systems*, pages 31–49, 2000.
5. Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Path-based identification constraints in description logics. In Gerhard Brewka and Jérôme Lang, editors, *KR*, pages 231–241, 2008.
6. Diego Calvanese, Wolfgang Fischl, Reinhard Pichler, Emanuel Sallinger, and Mantas Simkus. Capturing Relational Schemas and Functional Dependencies in RDFS. In *to appear in Proc. Nat. Conf. on Artificial Intelligence (AAAI)*, 2014.
7. Vitaliy Khizder, David Toman, and Grant E. Weddell. Adding ABoxes to a Description Logic with Uniqueness Constraints via Path Agreements. In *Proc. International Workshop on Description Logics DL2007*, pages 339–346, 2007.
8. Vitaliy L. Khizder, David Toman, and Grant Weddell. Reasoning about Duplicate Elimination with Description Logic. In *Rules and Objects in Databases (DOOD, part of CL'00)*, pages 1017–1032, 2000.
9. Vitaliy L. Khizder, David Toman, and Grant Weddell. On Decidability and Complexity of Description Logics with Uniqueness Constraints. In *Int. Conf. on Database Theory ICDT'01*, pages 54–67, 2001.
10. Il-Yeol Song and Peter P. Chen. Entity relationship model. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*, volume 1, pages 1003–1009. Springer, 2009.
11. Bernhard Thalheim. Extended entity relationship model. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*, volume 1, pages 1083–1091. Springer, 2009.
12. David Toman and Grant Weddell. On Attributes, Roles, and Dependencies in Description Logics and the Ackermann Case of the Decision Problem. In *Description Logics 2001*, pages 76–85. CEUR-WS vol.49, 2001.
13. David Toman and Grant Weddell. On Keys and Functional Dependencies as First-Class Citizens in Description Logics. In *Proc. of Int. Joint Conf. on Automated Reasoning (IJCAR)*, pages 647–661, 2006.
14. David Toman and Grant E. Weddell. On the interaction between inverse features and path-functional dependencies in description logics. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 603–608, 2005.
15. David Toman and Grant E. Weddell. On keys and functional dependencies as first-class citizens in description logics. *J. Aut. Reasoning*, 40(2-3):117–132, 2008.
16. David Toman and Grant E. Weddell. Applications and extensions of PTIME description logics with functional constraints. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 948–954, 2009.
17. David Toman and Grant E. Weddell. Conjunctive Query Answering in $\mathcal{CFD}_{nc}$: A PTIME Description Logic with Functional Constraints and Disjointness. In *Australasian Conference on Artificial Intelligence*, pages 350–361, 2013.

18. David Toman and Grant E. Weddell. Answering Queries over $\mathcal{CFD}_{nc}^{\forall}$ Knowledge Bases. Technical Report CS-2014-14, Cheriton School of Computer Science, University of Waterloo, 2014.
19. David Toman and Grant E. Weddell. On adding inverse features to the description logic $\mathcal{CFD}_{nc}^{\forall}$. In *PRICAI 2014: Trends in Artificial Intelligence - 13th Pacific Rim International Conference on Artificial Intelligence, Gold Coast, QLD, Australia, December 1-5, 2014.*, pages 587–599, 2014.
20. David Toman and Grant E. Weddell. Pushing the $\mathcal{CFD}_{nc}$ Envelope. In *International Workshop on Description Logics DL2014*, 2014.