On the Interaction between Inverse Features and Path-functional Dependencies in Description Logics

David Toman and Grant Weddell

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

Abstract

We investigate how inverse features can be added to a boolean complete description logic with path-functional dependencies in ways that avoid undecidability of the associated logical implication problem. In particular, we present two conditions that ensure the problem remains EXPTIME-complete. The first is syntactic in nature and limits the form that dependencies may have in argument terminologies. The second is a coherence condition on terminologies that is sufficiently weak to allow the transfer of relational and emerging object-oriented normalization techniques.

1 Introduction

For many applications, there is considerable incentive to enhance the modeling utility of feature based description logics (DLs) with an ability to capture richer varieties of uniqueness constraints such as keys and functional dependencies [DeHaan *et al.*, 2003; Khizder *et al.*, 2000; Stanchev and Weddell, 2003; Toman and Weddell, 2004b]. Unfortunately, in combination with feature or role inversion, the associated logical implication problem quickly becomes undecidable [Calvanese *et al.*, 2001]. We investigate conditions under which inverse features *can* be added to a boolean complete DL with *path-functional dependencies* (PFDs) [Weddell, 1989] without any consequent impact on the complexity of the associated logical implication problem.

Two conditions are presented that ensure this problem remains EXPTIME-complete. The first is syntactic in nature and imposes a prefix condition on PFDs that occur in argument terminologies. The condition complements and extends the results in [Calvanese *et al.*, 2001] which considered the problem of adding keys and functional dependencies to a DL with predicates in place of roles. The second is a coherence condition on terminologies that allows unrestricted use of PFDs, and is sufficiently weak to allow the formal specification of arbitrary relational or object-oriented schema, including those that fail to satisfy normalization conditions.

This latter observation is important since it enables an incremental development of terminologies that encode schema. One can begin, for example, with a "relational" terminology that fails to satisfy the conditions of Boyce-Codd Normal Form. (The approach used in [Calvanese *et al.*, 2001] is not generally capable of handling such anomalous cases.) Standard normalization algorithms and methodology can then employ reasoning services based on our results. Thus, our DL together with coherence is better equipped to enable the transfer of results in normalization and emerging object design theory for relational and object-oriented data models [Biskup *et al.*, 1996; Biskup and Polle, 2000].

1.1 Related Work

Our coherence condition derives from a similar condition proposed in [Biskup and Polle, 2003] to enable the development of a sound and complete axiomatization for an object-oriented data model, which essentially adds inclusion dependencies to a data model defined earlier in [Weddell, 1989]. The DL we consider in this paper is a further generalization; thus, our EXPTIME-completeness result *settles an open problem on the decidability of the implication problem for their model*.

In [Calvanese *et al.*, 2001], the authors consider a DL with (relational) functional dependencies together with a general form of keys called *identification constraints*. They show that their dialect is undecidable in the general case, but becomes decidable when unary functional dependencies are disallowed. We show undecidability in a simpler setting, in particular without the use of number restrictions. Our prefix condition on PFDs complements and extends their decidability result to more general PFDs, and our coherency condition serves as an alternative method for regaining decidability.

A form of key dependency with left-hand-side feature paths is considered for a DL coupled with various concrete domains [Lutz *et al.*, 2003; Lutz and Milicic, 2004]. The authors explore how the complexity of satisfaction is influenced by the selection of a concrete domain together with various syntactic restrictions on the key dependencies themselves. In contrast, we consider a DL that admits more general kinds of key constraints (and functional dependencies) for which identifying values can be defined on arbitrary domains.

The remainder of the paper is organized as follows. We begin by introducing the DL dialect $\mathcal{DLFAD}$ that will be the focus of the remainder of the paper. The dialect is *feature based* and therefore more functional in style as opposed to the more common *role based* derivatives of $\mathcal{ALC}$. As a consequence, it becomes straightforward to incorporate dependencies into the logic for capturing PFDs. In Section 3, we show that the combination of inverse features and arbitrary PFDs in $\mathcal{DLFAD}$ leads to the undecidability of its associated logical implication problem. Our main results then follow in Section 4 in which we consider two ways to recover decidability based on a *prefix restriction* condition for PFDs in argument terminologies and on a coherency condition for terminologies. Our summary comments follow in Section 5.

2 Preliminaries

Definition 1 (Description Logic $\mathcal{DLFAD}$) *Let F and C be sets of feature names and primitive concept names, respectively. A* path expression *is defined by the grammar "*$\mathsf{Pf} ::= f.\,\mathsf{Pf} \mid Id$*" for $f \in F$. We define derived* concept descriptions *by a second grammar on the left-hand-side of Figure 1. A concept description obtained by using the final production of this grammar is called a* path-functional dependency *(PFD).*

An inclusion dependency $\mathcal{C}$ *is an expression of the form $D \sqsubseteq E$. A* terminology $\mathcal{T}$ *consists of a finite set of inclusion dependencies.*

	Syntax	Semantics: "$(\cdot)^{\mathcal{I}}$"
D ::=	C	$(C)^{\mathcal{I}} \subseteq \Delta$
$\mid$	$D_1 \sqcap D_2$	$(D_1)^{\mathcal{I}} \cap (D_2)^{\mathcal{I}}$
$\mid$	$\neg D$	$\Delta \setminus (D)^{\mathcal{I}}$
$\mid$	$\forall f.D$	$\{x : (f)^{\mathcal{I}}(x) \in (D)^{\mathcal{I}}\}$
$\mid$	$\exists f^{-1}.D$	$\{(f)^{\mathcal{I}}(x) : x \in (D)^{\mathcal{I}}\}$
E ::=	D	
$\mid$	$E_1 \sqcap E_2$	$(E_1)^{\mathcal{I}} \cap (E_2)^{\mathcal{I}}$
$\mid$	$D : \mathsf{Pf}_1, ..., \mathsf{Pf}_k \rightarrow \mathsf{Pf}$	$\{x : \forall y \in (D)^{\mathcal{I}}. \bigwedge_{i=1}^{k} (\mathsf{Pf}_i)^{\mathcal{I}}(x) = (\mathsf{Pf}_i)^{\mathcal{I}}(y) \Rightarrow (\mathsf{Pf})^{\mathcal{I}}(x) = (\mathsf{Pf})^{\mathcal{I}}(y)\}$

Figure 1: Syntax and Semantics of $\mathcal{DLFAD}$.

The semantics *of expressions is defined with respect to a structure $(\Delta, \cdot^{\mathcal{I}})$, where Δ is a domain of "objects" and $(.)^{\mathcal{I}}$ an interpretation function that fixes the interpretations of primitive concepts C to be subsets of Δ and primitive features f to be total functions $(f)^{\mathcal{I}} : \Delta \rightarrow \Delta$. The interpretation is extended to path expressions, $(Id)^{\mathcal{I}} = \lambda x.x$, $(f.\,\mathsf{Pf})^{\mathcal{I}} = (\mathsf{Pf})^{\mathcal{I}} \circ (f)^{\mathcal{I}}$ and derived concept descriptions D and E as defined on the right-hand-side of Figure 1.*

An interpretation satisfies an inclusion dependency $D \sqsubseteq E$ *if $(D)^{\mathcal{I}} \subseteq (E)^{\mathcal{I}}$. The* logical implication problem *asks if $\mathcal{T} \models D \sqsubseteq E$ holds; that is, if $(D)^{\mathcal{I}} \subseteq (E)^{\mathcal{I}}$ for all interpretations that satisfy all constraints in $\mathcal{T}$.*

For the remainder of the paper, we use the following abbreviated notation: $\forall\,\mathsf{Pf}.\mathrm{D}$ is shorthand for $\forall f_1.\forall f_2.\ldots\forall f_k.\mathrm{D}$, and $\exists \mathsf{Pf}^{-1}.\mathrm{D}$ for $\exists {f_k}^{-1}.\ldots\exists {f_2}^{-1}.\exists {f_1}^{-1}.\mathrm{D}$ for $\mathsf{Pf} = f_1.f_2.\cdots.f_k.\,Id$. We also identify single feature names f with path descriptions $f.\,Id$ and allow concatenation of path descriptions, $\mathsf{Pf}\,\mathsf{Pf}'$ to denote their composition $\mathsf{Pf} \circ \mathsf{Pf}'$.

In addition, we classify constraints by the description on their right-hand side as *PFDs*, when the right-hand side is of the form $D : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \rightarrow \mathsf{Pf}$, and as *simple constraints* otherwise.

3 Undecidability of $\mathcal{DLFAD}$ Implication

We show a reduction of the unrestricted tiling problem to the $\mathcal{DLFAD}$ implication problem using a construction similar to that presented in [Calvanese *et al.*, 2001]. An unrestricted tiling problem U is a triple (T, H, V) where T is a finite set of tile types and $H, V \subseteq T \times T$ two binary relations. A *solution* to T is a mapping $t : \mathbf{N} \times \mathbf{N} \rightarrow T$ such that $(t(i,j), t(i+1,j)) \in H$ and $(t(i,j), t(i,j+1)) \in V$ for all $i \in \mathbf{N}$. This problem is Π^0_0-complete [Berger, 1966; van Emde Boas, 1997].

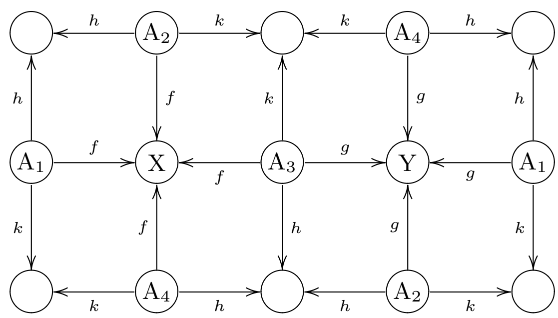

Figure 2: Defining a Grid.

The first step in the reduction is to establish an *integer grid*. This can be achieved, for example, as follows.

1. Introduce four disjoint concepts, A_1, A_2, A_3, and A_4, denoting cell edges.
$$\mathrm{A}_i \sqcap \mathrm{A}_j \sqsubseteq \bot \quad \text{for} \quad 0 < i < j \leq 4,$$
2. Grid cells are mapped to concepts X and Y that have four incoming f and g features, respectively.
$$\mathrm{X} \sqsubseteq \bigsqcap_{0<i\leq 4} \exists f^{-1}.\mathrm{A}_i \qquad \mathrm{Y} \sqsubseteq \bigsqcap_{0<i\leq 4} \exists g^{-1}.\mathrm{A}_i$$
3. To ensure that squares are formed, add the following.
$$\mathrm{A}_1 \sqsubseteq \mathrm{A}_2 : f \rightarrow h \sqcap \mathrm{A}_2 : h \rightarrow f \sqcap \mathrm{A}_2 : g \rightarrow k \sqcap \mathrm{A}_2 : k \rightarrow g$$
$$\mathrm{A}_2 \sqsubseteq \mathrm{A}_3 : f \rightarrow k \sqcap \mathrm{A}_3 : k \rightarrow f \sqcap \mathrm{A}_3 : g \rightarrow h \sqcap \mathrm{A}_3 : h \rightarrow g$$
$$\mathrm{A}_3 \sqsubseteq \mathrm{A}_4 : f \rightarrow h \sqcap \mathrm{A}_4 : h \rightarrow f \sqcap \mathrm{A}_4 : g \rightarrow k \sqcap \mathrm{A}_4 : k \rightarrow g$$
$$\mathrm{A}_4 \sqsubseteq \mathrm{A}_1 : f \rightarrow k \sqcap \mathrm{A}_1 : k \rightarrow f \sqcap \mathrm{A}_1 : g \rightarrow h \sqcap \mathrm{A}_1 : h \rightarrow g$$
4. And force squares to extend to the right and up by including the following.
$$\mathrm{A}_1 \sqsubseteq \forall f.\mathrm{X},\;\; \mathrm{A}_2 \sqsubseteq \forall g.\mathrm{Y},\;\; \mathrm{A}_3 \sqsubseteq \forall g.\mathrm{Y},\;\; \mathrm{A}_4 \sqsubseteq \forall f.\mathrm{X}$$

The accumulated effect of these inclusion dependencies on an interpretation is illustrated in Figure 2.

The adjacency rules for the instance U of the tiling problem can now be captured, e.g., as follows:

$$\mathrm{A}_1 \sqcap \forall g.T_i \sqsubseteq \forall f. \bigsqcup_{(t_i,t_j)\in H} T_j,\ \mathrm{A}_2 \sqcap \forall f.T_i \sqsubseteq \forall g. \bigsqcup_{(t_i,t_j)\in V} T_j,$$
$$\mathrm{A}_3 \sqcap \forall f.T_i \sqsubseteq \forall g. \bigsqcup_{(t_i,t_j)\in H} T_j,\ \mathrm{A}_4 \sqcap \forall g.T_i \sqsubseteq \forall f. \bigsqcup_{(t_i,t_j)\in V} T_j$$

where T_i corresponds to a tile type $t_i \in T$; we assume $T_i \sqcap T_j \sqsubseteq \perp$ for all $i < j$. The above constraints form a terminology $\mathcal{T}_U$ associated with an unrestricted tiling problem U.

Theorem 2 *A tiling problem U admits a solution iff $\mathcal{T}_U \not\models \mathrm{X} \sqcap (\bigsqcup_{t_i \in T} T_i) \sqsubseteq \perp$.*

Thus, the $\mathcal{DLFAD}$ implication problem is undecidable for unrestricted terminologies.

4 On Recovering Decidability

In this section, we present two ways to recover decidability for the $\mathcal{DLFAD}$ logical implication problem based on a syntactic prefix condition for PFDs occurring in argument terminologies and on a coherency condition for terminologies. Decidability in both cases is shown by exhibiting a reduction of logical implication problems in $\mathcal{DLFAD}$ to decidability problems of monadic sentences in the Ackermann prefix class.

Definition 3 (Monadic Ackerman Formulae) *Let P_i be monadic predicate symbols and x, y_i, z_i variables. A* monadic first-order formula in the Ackermann class *is a formula of the form $\exists z_1 \ldots \exists z_k \forall x \exists y_1 \ldots \exists y_l . \varphi$ where φ is a quantifier-free formula over the symbols P_i.*

Every formula with the Ackermann prefix can be converted to *Skolem normal form*: by replacing variables z_i by Skolem constants and y_i by unary Skolem functions not appearing in the original formula. This, together with standard boolean equivalences, yields a finite set of universally-quantified clauses containing at most one variable (x).

It is known that an Ackerman sentence has a model if and only if it has a Herbrand model; this allows us to use syntactic techniques for model construction. To establish the complexity bounds we use the following result for the satisfiability of Ackermann formulae:

Proposition 4 ([Fürer, 1981]) *The complexity of the satisfiability problem for Ackerman formulae is EXPTIME-complete.*

4.1 Prefix-restricted PFDs

Recall that the first of our conditions is syntactic and applies to the argument PFDs occurring in a terminology. This condition complements and extends the results in [Calvanese *et al.*, 2001]. However, because of "accidental common prefixes", it is not sufficient to follow the approach in [Calvanese *et al.*, 2001] of simply requiring unary PFDs to resemble keys. Non-unary PFDs can also cause trouble, as the following example illustrates.

Example 5 Consider the k-ary PFD

$$\mathrm{A}_1 \sqsubseteq \mathrm{A}_2 : f.a_1, \ldots, f.a_k \rightarrow h.$$

This PFD has a logical consequence $\mathrm{A}_1 \sqsubseteq \mathrm{A}_2 : f \rightarrow h$ and thus we can construct tiling similar to the one presented in the previous section.

The problem with the above k-ary PFD is that all the k paths in the precondition have the same prefix, f. To avoid this problem, it suffices to impose the following prefix condition.

Definition 6 (Prefix-restricted PFDs) *Let*

$$\mathrm{D} : \mathsf{Pf}\,\mathsf{Pf}_1, \ldots, \mathsf{Pf}\,\mathsf{Pf}_k \rightarrow \mathsf{Pf}'$$

be an arbitrary PFD where Pf is the maximal common prefix of the path expressions $\{\mathsf{Pf}\,\mathsf{Pf}_1, \ldots, \mathsf{Pf}\,\mathsf{Pf}_k\}$. The PFD is prefix-restricted *if either Pf' is a prefix of Pf or Pf is a prefix of Pf'.*

The above restriction allows us to recover an *almost* tree-model property for the logic. By using this restriction, we are able to construct special interpretations that satisfies all constraints in a given terminology but falsify a PFD constraint whenever an interpretation exists at all—essentially, the interpretation has the shape of two trees rooted by the two elements of the domain that provide an counterexample to the given PFD. To capture the effect of equalities implied by PFDs, the two trees are allowed to *share* nodes that are in the range of the same path function applied to the respective roots. In addition, we are able to bound the indegree branching in such an interpretation.

Definition 7 (Rank of Implication Problem) *Let $\mathcal{T}$ be a $\mathcal{DLFAD}$ terminology, $\mathcal{C}$ a constraint. We define $\mathrm{Rank}(\mathcal{T}, \mathcal{C})$ to be the number of occurrences of the $\exists f^{-1}.\mathrm{D}$ concept constructors in $\mathcal{T}$ and $\mathcal{C}$.*

The $\mathrm{Rank}(\mathcal{T}, \mathcal{C})$ limits the maximal number of different predecessors needed to satisfy all constraints in the terminology. The above observations provide the necessary tools for describing a single Herbrand interpretation with a fixed branching outdegree using monadic sentences that simulate the (special) interpretation showing that $\mathcal{T} \not\models \mathcal{C}$.

In the Herbrand interpretation, each term represents *two* elements, a left element and a right element of the original interpretation, unless these two elements are made equal by the effect of a PFD. We use the following unary predicates and function symbols:

Unary function symbols f (representing a feature f) and $f_1, \ldots, f_{\mathrm{Rank}(\mathcal{T},\mathcal{C})}$ (representing the possible inverses of f) for each feature $f \in F$. These function symbols are used, together with the constant 0 denoting the two roots in the original interpretation, to form terms. We overload the notation for path descriptions and use $\mathsf{Pf}(0)$ to denote terms as well.

Predicates $N^L(t)$ and $N^R(t)$ true for t representing elements that exist in the left and right parts of the original interpretation; these emulate partiality of the inverses.

Predicates $P^L_{\mathrm{D}}(t)$ and $P^R_{\mathrm{D}}(t)$ that are true for t that represent elements belonging to the description D in the respective parts of the original interpretation.

Predicates $E^{\mathsf{Pf}}(t)$ true for t whenever the two elements denoted by t agree on Pf in the original interpretation.

To ensure a finite number of assertions, we assume in the following that concept descriptions and features f are subconcepts of concepts or are features appearing in $\mathcal{T}$ and $\mathcal{C}$. We call the following collection of assertions $\Pi_{\mathcal{T}}$ (assertions with a superscript X stand for a pair of assertions, one with X substituted by L and one by R):

1. Totality of features: each object must have one outgoing feature f.

   $\forall x.N^X(x) \to N^X(f(x))$ for $x \neq f_i(y)$,
   $\forall x.N^X(f_i(x)) \to N^X(x)$

2. Functionality of features: Each element has at most one outgoing feature f.

   $\forall x.\neg(N^X(x) \wedge N^X(f(f_i(x)))$

3. Rules of equality: equalities propagate through function application, equal nodes have the same predecessors, must exist, and must belong to the same descriptions.

   $\forall x.(N^L(f(x)) \wedge N^R(f(x))) \to (E^{Id}(x) \to E^{Id}(f(x)))$
   $\forall x.(N^L(f_i(x)) \wedge N^R(f_i(x))) \to (E^{Id}(f_i(x)) \leftrightarrow E^{Id}(x))$
   $\forall x.\, E^{Id}(x) \to (N^L(x) \wedge N^R(x))$
   $\forall x.\, E^{Id}(x) \to (P^L_D(x) \leftrightarrow P^R_D(x))$

4. Concept formation—boolean constructors: enforce the excluded-middle law and the correct behavior of conjunction.

   $\forall x.N^X(x) \to (P^X_{\mathrm{D}}(x) \vee P^X_{\neg\mathrm{D}}(x))$
   $\forall x.N^X(x) \to \neg(P^X_{\mathrm{D}}(x) \wedge P^X_{\neg\mathrm{D}}(x))$
   $\forall x.N^X(x) \to (P^X_{\mathrm{D}_1 \sqcap \mathrm{D}_2}(x) \leftrightarrow (P^X_{\mathrm{D}_1}(x) \wedge P^X_{\mathrm{D}_2}(x)))$

5. Concept formation—value restrictions: assert value restrictions for pairs of neighboring nodes.

   $\forall x.(N^X(x) \wedge N^X(f(x))) \to (P^X_{\forall f.\mathrm{D}}(x) \leftrightarrow P^X_{\mathrm{D}}(f(x)))$
   $\forall x.(N^X(f_i(x)) \wedge N^X(x)) \to (P^X_{\forall f.\mathrm{D}}(f_i(x)) \leftrightarrow P^X_{\mathrm{D}}(x))$

6. Concept formation—existential restrictions: satisfy existential restrictions for inverses. Note that in the case of the left and right sides of the interpretations agreeing, the appropriate predecessor can be on either side of the interpretation (first two assertions).

   $$\forall x.\, E^{Id}(f(x)) \to (P^X_{\exists f^{-1}.\mathrm{D}}(f(x)) \leftrightarrow (N^L(x) \wedge P^L_{\mathrm{D}}(x)) \vee (N^R(x) \wedge P^R_{\mathrm{D}}(x)) \vee \bigvee N^X(f_i(f(x)) \wedge P^X_{\mathrm{D}}(f_i(f(x)))))$$
   $$\forall x.N^X(f(x)) \wedge \neg\, E^{Id}(f(x)) \to (P^X_{\exists f^{-1}.\mathrm{D}}(f(x)) \leftrightarrow (N^X(x) \wedge P^X_{\mathrm{D}}(x)) \vee \bigvee N^X(f_i(f(x)) \wedge P^X_{\mathrm{D}}(f_i(f(x)))))$$
   $$\forall x.N^X(x) \wedge x \neq f(y) \to (P^X_{\exists f^{-1}.\mathrm{D}}(x) \leftrightarrow \bigvee N^X(f_i(x) \wedge P^X_{\mathrm{D}}(f_i(x))))$$

7. Satisfaction of simple constraints in the terminology (GCDs): enforce simple subsumption constraints present in $\mathcal{T}$.

   $\forall x.P^X_{\mathrm{D}_1}(x) \to P^X_{\mathrm{D}_2}(x)$ for $\mathrm{D}_1 \sqsubseteq \mathrm{D}_2 \in \mathcal{T}$

8. Satisfaction of prefix-restricted PFDs by inverses: disallow violations of prefix-restricted PFDs due to existence of multiple inverse features agreeing on a node.

   $$\forall x.\neg(N^L(x) \wedge N^R(x) \wedge \neg\, E^{Id}(x) \wedge E^{Id}(f(x)) \wedge P^L_{\exists \mathsf{Pf}\,\mathsf{Pf}'^{-1}.D_1}(x)) \wedge P^R_{\exists \mathsf{Pf}\,\mathsf{Pf}'^{-1}.D_2}(x))$$
   $$\forall x.\neg(N^L(x) \wedge N^R(x) \wedge \neg\, E^{Id}(x) \wedge E^{Id}(f(x)) \wedge P^L_{\exists \mathsf{Pf}\,\mathsf{Pf}'^{-1}.D_2}(x)) \wedge P^R_{\exists \mathsf{Pf}\,\mathsf{Pf}'^{-1}.D_1}(x))$$
   $$\forall x.\neg(N^X(x) \wedge P^X_{\exists \mathsf{Pf}\,\mathsf{Pf}'^{-1}.D_1}(x) \wedge N^X(f_i(f(x))) \wedge P^X_{\exists \mathsf{Pf}\,\mathsf{Pf}'^{-1}.D_2}(f_i(f(x))))$$
   $$\forall x.\neg(N^X(x) \wedge P^X_{\exists \mathsf{Pf}\,\mathsf{Pf}'^{-1}.D_2}(x) \wedge N^X(f_i(f(x))) \wedge P^X_{\exists \mathsf{Pf}\,\mathsf{Pf}'^{-1}.D_1}(f_i(f(x))))$$
   $$\forall x.\neg(N^X(f_i(x)) \wedge P^X_{\exists \mathsf{Pf}\,\mathsf{Pf}'^{-1}.D_1}(f_i(x)) \wedge N^X(f_j(x)) \wedge P^X_{\exists \mathsf{Pf}\,\mathsf{Pf}'^{-1}.D_2}(f_j(x)))$$

   for all $\mathrm{D}_1 \sqsubseteq \mathrm{D}_2 : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \to \mathsf{Pf} \in \mathcal{T}$ such that $\mathsf{Pf}\,\mathsf{Pf}'\, f\, \mathsf{Pf}''$ is the common prefix of $\mathsf{Pf}_1, \ldots, \mathsf{Pf}_k$.

9. Path agreements: extend the simple equality to assertions of two nodes agreeing on a path; this is necessary to avoid exponential blowup in the length of the path descriptions in the PFDs.

   $$\forall x.(N^L(x) \wedge N^R(x) \wedge x \neq f_i(y)) \to (E^{f.\mathsf{Pf}}(x) \leftrightarrow E^{\mathsf{Pf}}(f(x)))$$
   $$\forall x.(N^L(f_i(x)) \wedge N^R(f_i(x))) \to (E^{f.\mathsf{Pf}}(f_i(x)) \leftrightarrow E^{\mathsf{Pf}}(x))$$

   where Pf ranges over all prefixes of path descriptions in $\mathcal{T}$'s PFDs.

10. Prefix-restricted PFDs: enforce PFDs between the left and right parts of the interpretation.

    $$\forall x.P^L_{\exists \mathsf{Pf}'^{-1}.\mathrm{D}_1}(x) \wedge P^R_{\exists \mathsf{Pf}'^{-1}.\mathrm{D}_2}(x) \wedge E^{\mathsf{Pf}_1}(x) \wedge \ldots \wedge E^{\mathsf{Pf}_k}(x) \to E^{\mathsf{Pf}}(x)$$
    $$\forall x.P^R_{\exists \mathsf{Pf}'^{-1}.\mathrm{D}_1}(x) \wedge P^L_{\exists \mathsf{Pf}'^{-1}.\mathrm{D}_2}(x) \wedge E^{\mathsf{Pf}_1}(x) \wedge \ldots \wedge E^{\mathsf{Pf}_k}(x) \to E^{\mathsf{Pf}}(x)$$

    for all $\mathrm{D}_1 \sqsubseteq \mathrm{D}_2 : \mathsf{Pf}'\,\mathsf{Pf}_1, \ldots, \mathsf{Pf}'\,\mathsf{Pf}_k \to \mathsf{Pf}'\,\mathsf{Pf} \in \mathcal{T}$.

The above assertions simulate the constraints implied by $\mathcal{T}$. To capture the violation of the constraint $\mathcal{C}$ we set $\Pi_{\mathcal{C}} = \{N^L(0), P^L_{\mathrm{D}_1}(0), N^R(0), \neg P^R_{\mathrm{D}_2}(0), E^{Id}(0)\}$ for $\mathcal{C}$ ordinary and $\Pi_{\mathcal{C}} = \{N^L(0), N^R(0), P^L_{\mathrm{D}_1}(0), P^R_{\mathrm{D}_2}(0), E^{\mathsf{Pf}_1}(0), \ldots, E^{\mathsf{Pf}_k}(0), \neg\, E^{\mathsf{Pf}}(0)\}$ for $\mathcal{C}$ a PFD.

Theorem 8 *Let $\mathcal{T}$ be a $\mathcal{DLFAD}$ terminology and $\mathcal{C}$ a constraint. Then $\mathcal{T} \models \mathcal{C}$ if and only if $\Pi_{\mathcal{T}} \cup \Pi_{\mathcal{C}}$ is not satisfiable.*

Proof (sketch): Consider a model $\mathcal{M}$ of $\Pi_{\mathcal{T}} \cup \Pi_{\mathcal{C}}$. We construct an interpretation $\mathcal{I} = (\Delta, (.)^{\mathcal{I}})$ as follows: Let $t^0(0)$, $t^L(0)$, and $t^R(0)$ be distinct values for each term $t(0)$.

$$\Delta = \{t^0(0) \mid \mathcal{M} \models E^{Id}(t(0))\} \cup \\ \{t^L(0) \mid \mathcal{M} \models N^L(t(0)) \wedge \neg E^{Id}(t(0))\} \cup \\ \{t^R(0) \mid \mathcal{M} \models N^R(t(0)) \wedge \neg E^{Id}(t(0))\}$$

$$(f)^{\mathcal{I}} = \{(t^X(0), f(t^X(0))) \mid t^X(0), f(t^X(0)) \in \Delta\} \cup \\ \{(f_i(t^X(0)), t^X(0)) \mid t^X(0), f_i(t^X(0)) \in \Delta\} \cup \\ \{(t^R(0), f(t^0(0)) \mid t^R(0), f(t^0(0)) \in \Delta\} \cup \\ \{(t^L(0), f(t^0(0)) \mid t^L(0), f(t^0(0)) \in \Delta\}$$

for $X \in \{0, L, R\}$,

$$(\mathrm{D})^{\mathcal{I}} = \{t^0(0) \mid \mathcal{M} \models N^L(t(0)) \wedge P^L_{\mathrm{D}}(t(0))\} \cup \\ \{t^L(0) \mid \mathcal{M} \models N^L(t(0)) \wedge P^L_{\mathrm{D}}(t(0))\} \cup \\ \{t^R(0) \mid \mathcal{M} \models N^R(t(0)) \wedge P^R_{\mathrm{D}}(t(0))\}$$

It is easy to verify that this interpretation satisfies all constraints in $\mathcal{T}$ and violates $\mathcal{C}$. Thus $\mathcal{T} \not\models \mathcal{C}$.

Conversely, given an interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \not\models \mathcal{C}$, we construct a model of $\Pi_{\mathcal{T}} \cup \Pi_{\mathcal{C}}$ as follows.

For $\mathcal{C} = \mathrm{D}_1 \sqsubseteq \mathrm{D}_2 : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \to \mathsf{Pf}$ a PFD, there must be elements $o_1, o_2 \in \Delta$ such that $o_1 \in (\mathrm{D}_1)^{\mathcal{I}}$ and $o_2 \in (\mathrm{D}_1)^{\mathcal{I}}$ such that $(\mathsf{Pf}_i)^{\mathcal{I}}(o_1) = (\mathsf{Pf}_i)^{\mathcal{I}}(o_2)$ for $0 < i \leq k$ but $(\mathsf{Pf})^{\mathcal{I}}(o_1) \neq (\mathsf{Pf})^{\mathcal{I}}(o_2)$.

We first define a part of the interpretation that consists of successors of o_1 and o_2:

$$\{N^L(\mathsf{Pf}(0)), P^L_{\mathrm{D}}(\mathsf{Pf}(0)) \mid (\mathsf{Pf})^{\mathcal{I}}(o_1) \in (\mathrm{D})^{\mathcal{I}}\} \cup \\ \{N^R(\mathsf{Pf}(0)), P^R_{\mathrm{D}}(\mathsf{Pf}(0)) \mid (\mathsf{Pf})^{\mathcal{I}}(o_2) \in (\mathrm{D})^{\mathcal{I}}\} \cup \\ \{E^{\mathsf{Pf}'}(\mathsf{Pf}(0)) \mid (\mathsf{Pf}\,\mathsf{Pf}')^{\mathcal{I}}(o_1) = (\mathsf{Pf}\,\mathsf{Pf}')^{\mathcal{I}}(o_2)\} \quad \subseteq \mathcal{M}$$

This part of $\mathcal{M}$ violates $\mathcal{C}$ and satisfies all constraints in $\mathcal{T}$ with the exception of the $\exists f^{-1}.D$. We extend $\mathcal{M}$ to satisfy the existential restrictions as follows:

for $\mathsf{Pf}(0)$ such that $E^{Id}(\mathsf{Pf}(0)) \notin \mathcal{M}$ and $N^L(\mathsf{Pf}(0))$. We chose $l \leq \mathrm{Rank}(\mathcal{T}, \mathcal{C})$ predecessors of $(\mathsf{Pf})^{\mathcal{I}}(o_1)$ that satisfy all implied existential restrictions. and identify these with the terms $f_1(\mathsf{Pf}(0)), \ldots, f_l(\mathsf{Pf}(0))$ and set

$$N^L(f_i(\mathsf{Pf}(0))), P^L_{\mathrm{D}(f_i(\mathsf{Pf}(0)))}, P^L_{\exists f^{-1}.\mathrm{D_i}}(\mathsf{Pf}(0)) \in \mathcal{M}$$

for all the chosen $\mathrm{D_i}$ predecessors of $(\mathsf{Pf})^{\mathcal{I}}(o_1)$ for $0 < i \leq l$. For all the terms $f_i(\mathsf{Pf}(0))$ we extract a tree interpretation from $\mathcal{I}$. Similarly, we extend $\mathcal{M}$ on the right side.

for $\mathsf{Pf}(0)$ such that $E^{Id}(\mathsf{Pf}(0)) \in \mathcal{M}$ we follow the same steps as above for the left part only.

Note that since $\mathcal{I}$ satisfies $\mathcal{T}$ all (translations of) ordinary constraints are satisfied and no pair of nodes on the left/right sides can violate a prefix-restricted PFD thus satisfying constraints generated from PFDs (8-10 above).

For $\mathcal{C} = \mathrm{D}_1 \sqsubseteq \mathrm{D}_2$ a simple constraint, there must be an element $o \in \Delta$ such that $o \in (\mathrm{D}_1)^{\mathcal{I}}$ and $o \notin (\mathrm{D}_2)^{\mathcal{I}}$. We define $\mathcal{M}$ using the same process as above noting that the two roots o_1 and o_2 are already equal in $\mathcal{I}$. □

As a side-effect of the above construction, we can now transform an arbitrary interpretation that satisfies $\mathcal{T}$ and violates $\mathcal{C}$ to an almost-tree interpretation informally referred to above in order to motivate the monadic assertions in $\Pi_{\mathcal{T}}$ and $\Pi_{\mathcal{C}}$.

The translation, in itself polynomial, therefore provides an EXPTIME decision procedure by appealing to Proposition 4. Completeness follows from EXPTIME-hardness of the implication problem for the $\{D_1 \sqcap D_2, \forall f.D\}$ fragment [Toman and Weddell, 2001; 2004b].

Corollary 9 *The implication problem for $\mathcal{DLFAD}$ with prefix-restricted PFDs is EXPTIME-complete.*

Using similar techniques, the $\exists f^{-1}.\mathrm{D}$ concept constructor can be generalized to more general *number restrictions* requiring lower and upper bounds (coded in binary) on the number of f predecessors while still maintaining the complexity bound.

4.2 Coherent Terminologies

The second of our conditions for recovering decidability is to impose a coherency condition on terminologies themselves. The main advantage of this approach is that we thereby regain the ability for unrestricted use of PFDs in terminologies. The disadvantage is roughly that there is a "single use" restriction on using feature inversions in terminologies.

Definition 10 (Coherent Terminology) *A terminology $\mathcal{T}$ is* coherent *if*

$$\mathcal{T} \models (\exists f^{-1}.D) \sqcap (\exists f^{-1}.E) \sqsubseteq \exists f^{-1}.(D \sqcap E)$$

for all descriptions D, E that appear as subconcepts of concepts that appear in $\mathcal{T}$, or their negations.

Note that we can *syntactically guarantee* that $\mathcal{T}$ is coherent by adding the $(\exists f^{-1}.D) \sqcap (\exists f^{-1}.E) \sqsubseteq \exists f^{-1}.(D \sqcap E)$ assertions to $\mathcal{T}$ for all descriptions D, E that appear in $\mathcal{T}$. This restriction allows us to construct interpretations of non-PFD descriptions that have the following property:

Definition 11 *An interpretation $(\Delta, (.)^{\mathcal{I}})$ is* coherent *if, for any $f \in F$, description D and $x, y \in \Delta$, $y \in (D)^{\mathcal{I}}$ if $x \in (D)^{\mathcal{I}}$ and $(f)^{\mathcal{I}}(x) = (f)^{\mathcal{I}}(y)$.*

Lemma 12 *Let $\mathcal{T}$ be a coherent terminology, $\mathcal{C}$ a simple constraint, and $\mathcal{I}$ an interpretation such that $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \not\models \mathcal{C}$. Then there is a coherent interpretation $\mathcal{I}'$ such that $\mathcal{I}' \models \mathcal{T}$ and $\mathcal{I}' \not\models \mathcal{C}$.*

Proof (sketch): Consider distinct $x, y \in \Delta_{\mathcal{I}}$ such that (i) $x \in (D_1)^{\mathcal{I}}$, (ii) $y \in (D_2)^{\mathcal{I}}$, and (iii) $(f)^{\mathcal{I}}(x) = (f)^{\mathcal{I}}(y)$. Then, since $\mathcal{T}$ is coherent, $x \in (D_1 \sqcap D_2)^{\mathcal{I}}$. For, $x \in (D_1 \sqcap \neg D_2)^{\mathcal{I}}$ leads to $(f_i)^{\mathcal{I}}(x) \in (\exists f_i{}^{-1}.(D_1 \sqcap \neg D_2) \sqcap \exists f_i{}^{-1}.D_2)^{\mathcal{I}}$, a contradiction. Thus, as models of $\mathcal{DLFA}$ have the tree model property, we can remove the farther of x or y and all its descendants, where the distance is measured from the node falsifying $\mathcal{C}$ in $\mathcal{I}$. The resulting interpretation still satisfies $\mathcal{T}$ and falsifies $\mathcal{C}$. Repeating this process yields a coherent interpretation. □

By restricting logical implication problems for $\mathcal{DLFAD}$ to cases in which terminologies are coherent, it becomes possible to apply reductions to satisfiability problems for Ackerman formulae.

Theorem 13 *Let $\mathcal{T}$ be a coherent $\mathcal{DLFAD}$ terminology. Then the implication problem $\mathcal{T} \models \mathcal{C}$ is decidable and EXPTIME-complete.*

Proof (sketch): We use a reduction similar to the one for prefix-restricted terminologies. However, as the terminology is coherent, in the left and right parts of the interpretation objects can never have more then one incoming feature with the same name and thus a single *inverse* f_1 for $f \in F$ is always sufficient to model this situation using the monadic formulas. Also, the left and right sides of the interpretation vacuously satisfy all PFDs (and thus the assertions in item 8 can be dropped). The interaction between PFDs and inverses is now completely captured by the auxiliary E^{Pf} predicates. For details see [Toman and Weddell, 2004a]. □

Note that we cannot consider coherent interpretations only as then *all* PFDs would be vacuously satisfied—no two nodes could possibly agree on a common path. This would make all PFDs trivial *coherent* consequences (i.e., when only coherent interpretations are considered). Our coherency restriction on terminologies is weaker: it only postulates that we can avoid multiple f predecessors if we wish to do so.

5 Summary and Future Work

We have presented a pair of conditions, one syntactic and one semantic, under which it becomes possible to combine feature inversion with path-functional dependencies in a boolean complete description logic while still ensuring that the associated logical implication problem is EXPTIME-complete; the problem is shown to be undecidable otherwise. The second of these conditions resolves an open issue on decidability of an analogous implication problem in [Biskup and Polle, 2003].

A natural extension of the description logic presented here allows *regular languages* (L) to replace path expressions, yielding the $\forall L.D$, $\exists L.D$, $\exists L^{-1}.D$, and $D : L \to L'$ constructors, and developing a decision procedure using the approach in [Toman and Weddell, 2004b]. One of the main applications of such an extension we envision is describing data structures for purposes of query optimization, extending [Liu *et al.*, 2002] to inductive data types.

Another direction of research considers weaker restrictions on $\mathcal{DLFAD}$ terminologies that still guarantee decidability, e.g., relaxing our *coherence* condition with respect to the unary PFDs actually present in a terminology.

References

[Berger, 1966] R. Berger. The undecidability of the dominoe problem. *Mem. Amer. Math. Soc.*, 66:1–72, 1966.

[Biskup and Polle, 2000] 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.

[Biskup and Polle, 2003] Joachim Biskup and Torsten Polle. Adding inclusion dependencies to an object-oriented data model with uniqueness constraints. *Acta Informatica*, 39:391–449, 2003.

[Biskup *et al.*, 1996] Joachim Biskup, Ralf Menzel, Torsten Polle, and Yehoshua Sagiv. Decomposition of Relationships through Pivoting. In *Conceptual Modeling - ER'96*, pages 28–41, 1996.

[Calvanese *et al.*, 2001] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. Identification Constraints and Functional Dependencies in Description Logics. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 155–160, 2001.

[DeHaan *et al.*, 2003] David DeHaan, David Toman, and Grant E. Weddell. Rewriting Aggregate Queries using Description Logics. In *Description Logics 2003*, pages 103–112. CEUR-WS vol.81, 2003.

[Fürer, 1981] Martin Fürer. Alternation and the Ackermann Case of the Decision Problem. *L'Enseignement Math.*, 27:137–162, 1981.

[Khizder *et al.*, 2000] Vitaliy L. Khizder, David Toman, and Grant E. Weddell. Reasoning about Duplicate Elimination with Description Logic. In *Rules and Objects in Databases (DOOD, part of CL'00)*, pages 1017–1032, 2000.

[Liu *et al.*, 2002] Huizhu Liu, David Toman, and Grant E. Weddell. Fine Grained Information Integration with Description Logic. In *Description Logics 2002*, pages 1–12. CEUR-WS vol.53, 2002.

[Lutz and Milicic, 2004] C. Lutz and M. Milicic. Description Logics with Concrete Domains and Functional Dependencies. In *Proc. European Conference on Artificial Intelligence (ECAI)*, pages 378–382, 2004.

[Lutz *et al.*, 2003] Carsten Lutz, Carlos Areces, Ian Horrocks, and Ulrike Sattler. Keys, Nominals, and Concrete Domains. In *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 349–354, 2003.

[Stanchev and Weddell, 2003] Lubomir Stanchev and Grant E. Weddell. Index Selection for Embedded Control Applications using Description Logics. In *Description Logics 2003*, pages 9–18. CEUR-WS vol.81, 2003.

[Toman and Weddell, 2001] David Toman and Grant E. 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.

[Toman and Weddell, 2004a] David Toman and Grant E. Weddell. Attribute Inversion in Description Logics with Path Functional Dependencies. In *Description Logics 2004*, pages 178–187. CEUR-WS vol.104, 2004.

[Toman and Weddell, 2004b] David Toman and Grant E. Weddell. On Reasoning about Structural Equality in XML: A Description Logic Approach. *Theoretical Computer Science*, 2004. doi:10.1016/j.tcs.2004.10.036.

[van Emde Boas, 1997] P. van Emde Boas. The convenience of tilings. In *Complexity, Logic, and Recursion Theory*, volume 187 of *Lecture notes in pure and applied mathematics*, pages 331–363. Marcel Dekker Inc., 1997.

[Weddell, 1989] Grant Weddell. A Theory of Functional Dependencies for Object Oriented Data Models. In *International Conference on Deductive and Object-Oriented Databases*, pages 165–184, 1989.