Attribute Inversion in Description Logics with Path Functional Dependencies

David Toman and Grant Weddell

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

Abstract

We present a coherence condition for a boolean complete description logic with feature inversions and a very general form of uniqueness constraint that enables, among other things, the capture of unary functional dependencies. The condition is sufficiently weak to allow the transfer of relational and emerging object-oriented normalization techniques while still ensuring that the associated logical implication problem remains DEXPTIME-complete.

1 Introduction

For many applications, there is considerable incentive to enhance the modelling utility of a description logic (DL) with an ability to capture richer varieties of uniqueness constraints such as keys and functional dependencies [6, 9, 13, 15]. Unfortunately, in combination with role or attribute inversions, the associated logical implication problem quickly becomes undecidable [5]. We present a coherence condition for a boolean complete DL with feature inversions which allows unrestricted use of path functional dependencies [17]. The condition ensures that the associated logical implication problem remains DEXPTIME-complete, but 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. (Note that the approach used in [5] 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 is better equipped to enable the transfer of results in normalization and emerging object design theory for relational and object-oriented data models [2, 3]. We also show that relaxing this coherence condition leads to undecidability.

1.1 Related Work

Our coherence condition derives from a similar condition proposed in [4] to enable the development of a sound and complete axiomatization for an object-oriented data model, which essentially adds inclusion dependencies to an earlier data model [17]. The DL we consider in this paper is a further generalization; thus, our DEXPTIME-completeness result *settles an open problem on the decidability of the implication problem for their model.*

In [5], the authors consider a DL with (relational) functional dependencies together with a general form of keys called *identification constraints*. They show that this dialect is undecidable in the general case, but becomes decidable when unary functional dependencies are disallowed. 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 [12]. 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. 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. Definitions of the DL dialect $\mathcal{DLFAD}$ and the above-mentioned coherency condition are given next in Section 2. In Section 3, we show that failure to satisfy the coherency condition leads to undecidability of the implication problem for $\mathcal{DLFAD}$. For cases satisfying this condition, we show in Section 4 that the problem is DEXPTIME-complete. To do this, we first consider the problem for $\mathcal{DLFA}$, a fragment of $\mathcal{DLFAD}$ that excludes uniqueness constraints. Although $\mathcal{DLFA}$ is less expressive than, e.g., $\mathcal{DLR}$, we gain the opportunity of presenting encoding schemes for reductions in a more incremental fashion. Our summary comments follow in Section 5.

2 Preliminaries

Definition 1 (Description Logic $\mathcal{DLFAD}$) *Let F and C be sets of attribute 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 the grammar on the left-hand-side of Figure 1. A concept description*

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

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

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.*

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 attributes* f *to be total functions* $(f)^{\mathcal{I}} : \Delta \to \Delta$. *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* $\mathcal{T}$.

The coherency condition on which our decidability results depend is defined as follows. (Recall that a similar condition is introduced in [4] to ensure that an axiomatization for their data model is complete.)

Definition 2 (Coherent Terminology) *A terminology* $\mathcal{T}$ *is* coherent *if*

$$\mathcal{T} \models (D@f) \sqcap (E@f) \sqsubseteq (D \sqcap E)@f$$

holds for all descriptions D, E *and attributes* f.

Note that we can *syntactically guarantee* that $\mathcal{T}$ is coherent by adding the $(D@f) \sqcap (E@f) \sqsubseteq (D \sqcap E)@f$ assertions to $\mathcal{T}$ (for all descriptions D, E that appear in $\mathcal{T}$).

3 Undecidability for General $\mathcal{DLFAD}$ Terminologies

We show a reduction of the unrestricted tiling problem to the $\mathcal{DLFAD}$ implication problem using a construction similar to that presented in [5]. 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 $t : \mathbf{N} \times \mathbf{N} \to T$ 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 [1, 16]. 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, B, C and D, denoting cell edges.

$$\mathrm{A} \sqcap \mathrm{B} \sqsubseteq \bot, \quad \mathrm{A} \sqcap \mathrm{C} \sqsubseteq \bot, \quad \ldots, \quad \mathrm{C} \sqcap \mathrm{D} \sqsubseteq \bot$$

2. Grid cells are mapped to concepts X and Y that have four incoming f and g attributes, respectively.

$$\mathrm{X} \sqsubseteq (\mathrm{A}@f) \sqcap (\mathrm{B}@f) \sqcap (\mathrm{C}@f) \sqcap (\mathrm{D}@f),$$
$$\mathrm{Y} \sqsubseteq (\mathrm{A}@g) \sqcap (\mathrm{B}@g) \sqcap (\mathrm{C}@g) \sqcap (\mathrm{D}@g)$$

3. To ensure that squares are formed, add the following.

$$\begin{array}{llll} \mathrm{A} \sqsubseteq \mathrm{B} : f \to h, & \mathrm{B} \sqsubseteq \mathrm{C} : f \to i, & \mathrm{C} \sqsubseteq \mathrm{D} : f \to h, & \mathrm{D} \sqsubseteq \mathrm{A} : f \to i, \\ \mathrm{A} \sqsubseteq \mathrm{B} : h \to f, & \mathrm{B} \sqsubseteq \mathrm{C} : i \to f, & \mathrm{C} \sqsubseteq \mathrm{D} : h \to f, & \mathrm{D} \sqsubseteq \mathrm{A} : i \to f, \\ \mathrm{A} \sqsubseteq \mathrm{B} : g \to i, & \mathrm{B} \sqsubseteq \mathrm{C} : g \to h, & \mathrm{C} \sqsubseteq \mathrm{D} : g \to i, & \mathrm{D} \sqsubseteq \mathrm{A} : g \to h, \\ \mathrm{A} \sqsubseteq \mathrm{B} : i \to g, & \mathrm{B} \sqsubseteq \mathrm{C} : h \to g, & \mathrm{C} \sqsubseteq \mathrm{D} : i \to g, & \mathrm{D} \sqsubseteq \mathrm{A} : h \to g \end{array}$$

4. And to force squares to extend to the right and up, include the following.

$$\mathrm{A} \sqsubseteq \forall g.Y, \quad \mathrm{B} \sqsubseteq \forall g.Y, \quad \mathrm{C} \sqsubseteq \forall f.X, \quad \mathrm{D} \sqsubseteq \forall f.X$$

The *adjacency rules* from the instance U of the tiling problem are defined as follows:

$$\begin{array}{ll} A \sqcap \forall f.T_i \sqsubseteq \forall g.\bigsqcup_{(t_i,t_j)\in V} T_j, & C \sqcap \forall g.T_i \sqsubseteq \forall f.\bigsqcup_{(t_i,t_j)\in V} T_j \\ B \sqcap \forall f.T_i \sqsubseteq \forall g.\bigsqcup_{(t_i,t_j)\in H} T_j, & D \sqcap \forall g.T_i \sqsubseteq \forall f.\bigsqcup_{(t_i,t_j)\in H} T_j, \end{array}$$

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

Theorem 3 *A tiling problem U admits a solution iff $\mathcal{T}_U \not\models \mathrm{X} \sqcap (\bigsqcup_{t_i \in T} T_i) \sqsubseteq \bot$.*

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

4 Coherency implies Decidability

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. After we introduce the latter, we begin by defining reductions for the fragment $\mathcal{DLFA}$. An essentially incremental elaboration of these reductions is presented in the final subsection in which we establish our main result: decidability with coherency of the logical implication problem for $\mathcal{DLFAD}$.

Definition 4 (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 5 ([7]) *The complexity of the satisfiability problem for Ackerman formulae is DEXPTIME-complete.*

4.1 Decidability for Coherent $\mathcal{DLFA}$ Terminologies

We construct an Ackerman-class sentence whose satisfiability is equivalent to a given $\mathcal{DLFA}$ implication problem. In the simulation, $\mathcal{DLFA}$'s concept descriptions D are modeled by monadic predicates $P_D(x)$. The function symbols f and $\overline{f}$ are used in sentences to stand for the attribute f; the symbol $\overline{f}$ stands for the reverse of the attribute f in cases where an object is in the range of multiple attributes (this situation can be introduced, e.g., by the description $(\mathrm{D}_1@f_1) \sqcap (\mathrm{D}_2@f_2)$). This arrangement, in the case of coherent terminologies, allows us to represent $\mathcal{DLFA}$ interpretations as Herbrand interpretations (in the extended language). The construction proceeds in two steps. First, the structural properties of $\mathcal{DLFA}$ are encoded using the following assertions.

- Node existence assertions:

$$\forall x.N(x) \leftrightarrow N(f_i(x)) \text{ for } x \neq \overline{f}_i(y), \quad \forall x.N(\overline{f}_i(x)) \leftrightarrow N(x)$$

- Functionality of attributes and Coherence:

$$\forall x.\neg(N(x) \wedge N(f(\overline{f}(x)))), \qquad \forall x.\neg(N(x) \wedge N(\overline{f}(f(x)))$$

- Concept formation assertions for boolean constructors:

$$\forall x.N(x) \rightarrow (P_D(x) \vee P_{\neg D}(x)), \quad \forall x.\neg(P_D(x) \wedge P_{\neg D}(x)),$$
$$\forall x.N(x) \rightarrow (P_{D_1 \sqcap D_2}(x) \leftrightarrow (P_{D_1}(x) \wedge P_{D_2}(x)))$$

- Concept formation assertions for attribute constructors:

$$\begin{array}{rcl} \forall x.N(x) & \rightarrow & (P_{\forall f_i.D}(x) \leftrightarrow P_D(f_i(x))) \ \text{ for } \ x \neq \ \overline{f}_i(y) \\ \forall x.N(\overline{f}_i(x)) & \rightarrow & (P_{\forall f_i.D}(\overline{f}_i(x)) \leftrightarrow P_D(x)) \\ \forall x.N(x) & \rightarrow & (P_{D@f_i}(x) \rightarrow N(\overline{f}_i(x))) \ \text{ for } \ x \neq \ f_i(y) \\ \forall x.N(x) & \rightarrow & (P_{D@f_i}(x) \leftrightarrow P_D(\overline{f}_i(x))) \ \text{ for } \ x \neq \ f_i(y) \\ \forall x.N(\overline{f}_i(x)) & \rightarrow & (P_{D@f_i}(f_i(x)) \leftrightarrow P_D(x)) \end{array}$$

The collection of the assertions above is denoted $\Pi_{\mathcal{DLFA}}$. This set captures the structural relationships between $\mathcal{DLFA}$ concepts. Although this set is infinite in general, the set of concepts appearing in a particular implication problem, $\mathcal{T} \models \mathcal{C}$, is finite. Hence, one can restrict the set of assertions $\Pi_{\mathcal{DLFA}}$ to a finite subset $\Pi^{\mathcal{T},\mathcal{C}}_{\mathcal{DLFA}}$ that contains only the predicates that define concepts in $\mathcal{T} \cup \{\mathcal{C}\}$. (In the rest of the paper we omit the superscript whenever clear from the context.) To complete the translation of a $\mathcal{DLFA}$ implication problem, what remains is the translation of the inclusion constraints.

Definition 6 *Let $\mathcal{T}$ and $\mathcal{C} \equiv D \sqsubseteq E$ be a $\mathcal{DLFA}$ terminology and an inclusion constraint, respectively. We define*

- $\Pi_{\mathcal{T}} = \{N(x) \rightarrow (\forall x.P_D(x) \rightarrow P_E(x)) : D \sqsubseteq E \in \mathcal{T}\}$ *and*
- $\Pi_{\mathcal{C}} = \{N(0), P_D(0), P_{\neg E}(0)\}$,

The three clauses $\Pi_{\mathcal{C}}$ represent the skolemized version of $\neg\forall x.P_D(x) \rightarrow P_E(x)$; 0 is the Skolem constant for x. As usual, a model "containing" $\Pi_{\mathcal{C}}$ is a counterexample for $\mathcal{C}$. To show the correspondence formally we need the following definition and lemma:

Definition 7 *An interpretation $(\Delta, (.)^{\mathcal{I}})$ is* coherent *if $(f_i)^{\mathcal{I}}(x) = (f_i)^{\mathcal{I}}(y) \rightarrow x = y$ for all $x, y \in \Delta$ and f_i an attribute name.*

Lemma 8 *Let $\mathcal{T}$ be a coherent terminology, $\mathcal{C}$ a subsumption 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) $(f_i)^{\mathcal{I}}(x) = (f_i)^{\mathcal{I}}(y)$. 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 $(f_i)^{\mathcal{I}}(x) \in ((D_1 \sqcap \neg D_2)@f_i \sqcap D_2@f_i)^{\mathcal{I}}$, 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. □

Theorem 9 *Let $\mathcal{T}$ and $\mathcal{C}$ be a terminology and inclusion dependency in $\mathcal{DLFA}$, respectively. Then $\mathcal{T} \models \mathcal{C} \iff \Pi_{\mathcal{DLFA}} \cup \Pi_{\mathcal{T}} \cup \Pi_{\mathcal{C}}$ is not satisfiable.*

Proof: (sketch) Consider a Herbrand model $\mathcal{M}$ such that $\mathcal{M} \models \Pi_{\mathcal{DLFA}} \cup \Pi_{\mathcal{T}} \cup \Pi_{\mathcal{C}}$. We construct an interpretation $\mathcal{I}_{\mathcal{M}} = (\Delta, (.)^{\mathcal{I}})$ where:

- $\Delta = \{x : \mathcal{M} \models N(x)\}$,
- $(D)^{\mathcal{I}} = \{x : \mathcal{M} \models P_D(x)\}$, and $(f)^{\mathcal{I}} = \{(x, y) : y = f(x) \text{ or } \overline{f}(y) = x\}$.

It is easy to verify (by cases analysis) that $\mathcal{I}_{\mathcal{M}} \models \mathcal{T}$ but $\mathcal{I}_{\mathcal{M}} \not\models \mathcal{C}$.

For the other direction we take any coherent interpretation $\mathcal{I}$, such that $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \not\models \mathcal{C}$. This interpretation must exist by Lemma 8 whenever $\mathcal{T} \not\models \mathcal{C}$. Let $o \in \Delta$ be an object that falsifies $\mathcal{C}$ in $\mathcal{I}$. We construct a Herbrand universe as the set of all terms that correspond to undirected paths of attributes originating in o; we use $\overline{f}_i$ for every attribute f traversed "backward" along this path. Each of these terms, t_x, due to the coherence condition, corresponds to exactly one element of $x \in \Delta$. On top of this universe we define a Herbrand model $\mathcal{M}_{\mathcal{I}} = \{P_D(t_x) : x \in (D)^{\mathcal{I}}\} \cup \{N(t_x)\}$ The remainder is verification of $\mathcal{M}_{\mathcal{I}} \models \Pi_{\mathcal{DLFA}} \cup \Pi_{\mathcal{T}} \cup \Pi_{\mathcal{C}}$ by cases analysis. □

The translation therefore provides a DEXPTIME decision procedure by appealing to Proposition 5. Completeness follows from DEXPTIME-hardness of the implication problem for the $\{D_1 \sqcap D_2, \forall f.D\}$ fragment [14, 15].

Corollary 10 *The implication problem for $\mathcal{DLFA}$ is DEXPTIME-complete.*

4.2 Decidability for Coherent $\mathcal{DLFAD}$ Terminologies

For each implication problem $\mathcal{T} \models \mathcal{C}$, we define a satisfiability problem $\Pi_{\mathcal{DLFAD}} \cup \Pi_{\mathcal{T}} \cup \Pi_{\mathcal{C}}$. There are two cases to consider depending on $\mathcal{C}$.

Case 1: $\mathcal{C}$ is a $\mathcal{DLFA}$ inclusion dependency.

Lemma 11 *Let $\mathcal{T} \models \mathcal{C}$ be a $\mathcal{DLFAD}$ implication problem in which $\mathcal{T}$ is coherent and for which $\mathcal{C}$ is a $\mathcal{DLFA}$ inclusion dependency. Let $\mathcal{T}'$ be the largest subset of $\mathcal{T}$ that is also a $\mathcal{DLFA}$ terminology. Then $\mathcal{T}' \models \mathcal{C}$ if and only if $\mathcal{T} \models \mathcal{C}$.*

Proof: Assume $\mathcal{T}' \not\models \mathcal{C}$. Then by Lemma 8 there must be a coherent interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{T}'$ but $\mathcal{I} \not\models \mathcal{C}$. However, since $\mathcal{I}$ is coherent, it also satisfies $\mathcal{T}$. The other direction is immediate as $\mathcal{T}' \subseteq \mathcal{T}$. □

Thus, in this case, we can use Theorem 9 to decide the implication problem.

Case 2: $\mathcal{C} = D_1 \sqsubseteq D_2 : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \to \mathsf{Pf}$. To falsify such an inclusion dependency, *two* objects (one in D_1 and another in D_2) that satisfy the preconditions of the dependency but fail to satisfy the conclusion are needed. We therefore construct two copies of the interpretation for the $\mathcal{DLFA}$ constraints in $\mathcal{T}$ in a fashion analogous to [8, 18]. However, as Herbrand terms are essentially the same in the two copies, it is sufficient to distinguish them by renaming the predicate symbols [10]. In addition, we need to model the "rules" of equality and their interaction with concept descriptions. The structural rules for $\mathcal{DLFAD}$ are thus defined as follows:

$$\Pi_{\mathcal{DLFAD}} = \Pi^L_{\mathcal{DLFA}} \cup \Pi^R_{\mathcal{DLFA}} \cup \left\{ \begin{array}{l} \forall x.E(x) \to E(f_i(x)) \\ \forall x.(N(\overline{f}_i(x) \wedge E(\overline{f}_i(x))) \to E(x) \\ \forall x.(N(x) \wedge E(x)) \to (P^L_D(x) \leftrightarrow P^R_D(x)) \end{array} \right\},$$

where Π^L is the set of assertions Π in which every predicate P_D is renamed to P^L_D (similarly for Π^R). In addition, we use the following notation: we say that $\overline{\mathsf{Pf}}$ is a *reverse prefix of* Pf iff $\overline{\mathsf{Pf}}$ is a prefix of Pf in which each f_i was replaced by $\overline{f}_i$ and the order of the attributes was reversed. We also define

$$\lfloor \mathsf{Pf} \rfloor = \left\{ \begin{array}{ll} \lfloor \mathsf{Pf}_1 \circ \mathsf{Pf}_2 \rfloor & \text{for } \mathsf{Pf} = \mathsf{Pf}_1 \circ \overline{f}_i \circ f_i \circ \mathsf{Pf}_2 \\ \mathsf{Pf} & \text{otherwise} \end{array} \right.$$

We construct a set of assertions for a given terminology $\mathcal{T}$. Let $\mathcal{T}'$ denote the subsumption constraints in $\mathcal{T}$ without PFDs, and $\mathcal{T}'' = \mathcal{T} - \mathcal{T}'$. Then we define

$$\begin{array}{rcl} \Pi_{\mathcal{T}} & = & \Pi^L_{\mathcal{T}'} \cup \Pi^R_{\mathcal{T}'} \\ & \cup & \left\{ \begin{array}{l} N^L(\overline{\mathsf{Pf}}(x)) \wedge N^R(\overline{\mathsf{Pf}}(x)) \wedge \\ ((P^L_{D_1}(\overline{\mathsf{Pf}}(x)) \wedge P^R_{D_2}(\overline{\mathsf{Pf}}(x))) \vee (P^R_{D_1}(\overline{\mathsf{Pf}}(x)) \wedge P^L_{D_2}(\overline{\mathsf{Pf}}(x)))) \wedge \\ \hfill \bigwedge_{i=1}^k E(\lfloor \overline{\mathsf{Pf}} \circ \mathsf{Pf}_i \rfloor(x) \to E(\lfloor \overline{\mathsf{Pf}} \circ \mathsf{Pf}_0 \rfloor(x) \\ \hfill \text{where } \mathrm{D}_1 \sqsubseteq D_2 : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \to \mathsf{Pf}_0 \in \mathcal{T}'' \\ \hfill \text{and } \overline{\mathsf{Pf}} \text{ a reverse prefix of } \mathsf{Pf}_i \end{array} \right\} \end{array}$$

and, for $\mathcal{C} \equiv \mathrm{D}_1 \sqsubseteq D_2 : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \to \mathsf{Pf}_0$ we define

$$\Pi_{\mathcal{C}} = \{N(0), P^L_{D_1}(0), P^R_{D_2}(0), E(\mathsf{Pf}_1(0)), \ldots, E(\mathsf{Pf}_k(0)), \neg E(\mathsf{Pf}_0(0))\}$$

Theorem 12 *Let $\mathcal{T} \models \mathcal{C}$ be a $\mathcal{DLFAD}$ implication problem in which $\mathcal{C}$ is a PFD. Then $\mathcal{T} \models \mathcal{C}$ if and only if $\Pi_{\mathcal{DLFAD}} \cup \Pi_{\mathcal{T}} \cup \Pi_{\mathcal{C}}$ is not satisfiable.*

Proof: (sketch) The proof proceeds analogously to the proof of Theorem 9 by explicitly constructing a counterexample interpretation from a Herbrand model, and vice versa. The crux lies in observing that, in addition to proper simulation of $\mathcal{DLFA}$ descriptions, a path agreement (i.e., a precondition or a consequent of a PFD) holds in a $\mathcal{DLFAD}$ interpretation if and only if a corresponding $E(t(0))$ atom appears in the Herbrand model—this fact hinges on introducing the $\overline{f}_i$ function symbols and on representing a single PFD by *multiple* formulae. $\square$

5 Summary and Future Work

We have defined a coherence condition for a boolean complete description logic with feature inversions and arbitrary path functional dependencies that ensures the associated logical implication problem is DEXPTIME-complete; the problem is undecidable otherwise. This resolves an open issue on decidability of an analogous implication problem in [4].

A natural extension of the description logic presented here allows *regular languages* (L) to replace path expressions, yielding the $\forall L.D$, $\exists L.D$, $D@L$, and $D : L \to L'$ constructors, and developing a decision procedure using the approach in [15]. One of the main applications of such an extension we envision is describing data structures for purposes of query optimization, extending [11] 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

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

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

[3] 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.

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

[5] 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 2001)*, pages 155–160, 2001.

[6] 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.

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

[8] Minoru Ito and Grant Weddell. Implication Problems for Functional Constraints on Databases Supporting Complex Objects. *Journal of Computer and System Sciences*, 49(3):726–768, 1994.

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

[10] Vitaliy L. Khizder, David Toman, and Grant E. Weddell. On Decidability and Complexity of Description Logics with Uniqueness Constraints. In *International Conference on Database Theory ICDT'01*, pages 54–67, 2001.

[11] 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.

[12] 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 2003)*, pages 349–354, 2003.

[13] 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.

[14] 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.

[15] David Toman and Grant E. Weddell. On Reasoning about Structural Equality in XML: A Description Logic Approach. *Theoretical Computer Science*, 2004. ICDT 2003 special issue, in press.

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

[17] 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.

[18] Grant E. Weddell. Reasoning about Functional Dependencies Generalized for Semantic Data Models. *TODS*, 17(1):32–64, 1992.