Path Description Dependencies in Feature-Based DLs

Eva Feng[1], Alexander Borgida[2], Enrico Franconi[3], Peter F. Patel-Schneider[4], David Toman[1] and Grant Weddell[1]

[1]*Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada*
[2]*Department of Computer Science, Rutgers University, New Brunswick, US*
[3]*KRDB Research Centre for Knowledge and Data, Free University of Bozen-Bolzano, Italy*
[4]*No affiliation*

Abstract
All members of the FunDL family of description logics replace roles with partial functions (*features*) and have a concept constructor, called *path functional dependency* (PFD), for expressing extensions of functional dependencies, useful in object-relational data sources. In this paper, we consider generalizing PFDs to *path description dependencies* (PDD), by allowing inverse features in them, which result in paths being set-valued. We show that logical consequence for *partial-*$\mathcal{DLFDI}$, one of the most expressive dialects of the FunDL family, remains EXPTIME complete for coherent terminologies, when extended with PDDs. As a first application, we consider *referring expression types*, which are concept descriptions used to identify individuals in query answers. In the new dialect, such a type denotes a *set* of concept descriptions. As such, one must prove in advance that any interpretation of any member of the set will never have more than one element. We show that this "singularity condition" can be diagnosed as a collection of logical consequence questions wrt a TBox.

1. Introduction

The FunDL dialects of description logics (DLs) were designed primarily to support reasoning that is required in structured data integration and view-based query rewriting over object-relational data sources such as relational databases [1, 2] and more recently JSON stores [3]. Consequently, all such dialects replace roles with partial functions called *features* for a better alignment with the ubiquitous notion of an attribute or column in such data sources, and all have a *path functional dependency* (PFD) concept constructor for capturing equality-generating dependencies such as keys, uniqueness constraints and functional dependencies that are commonly part of schemata for such data sources.

A PFD generalizes the notion of a functional dependency by allowing *path functions* in place of attributes to express navigation over a sequence of features. For example, suppose for an accountant's personal database of clients, some are friends. An axiom expressing that the combination of the first name and the dial number of the phone identifies at most one friend

DL 2023: 36th International Workshop on Description Logics, September 2-4, 2023, Rhodes, Greece
ehfeng@uwaterloo.ca (E. Feng); borgida@cs.rutgers.edu (A. Borgida); franconi@inf.unibz.it (E. Franconi); pfpschneider@gmail.com (P. F. Patel-Schneider); david@uwaterloo.ca (D. Toman); gweddell@uwaterloo.ca (G. Weddell)

among all clients in the database can be expressed in a FunDL dialect as follows.

$$\text{CLIENT-FRIEND} \sqsubseteq (\text{CLIENT} : \mathit{fname}, \mathit{phone}.\mathit{dialnum} \rightarrow id) \tag{1}$$

Here, the axiom with the PFD concept constructor is used to express a key or uniqueness condition underlying a form of *identification vetting*. It can be paraphrased in English as: *client friends are clients that, among all clients, can be identified by a combination of their first name and the dial number of their phone.*

In this paper, we generalize the PFD concept constructor with a *path description dependency* (PDD) constructor that allows traversing features in both directions. For example, consider a generalization of the above example where the accountant's clients have possibly multiple phones and that those who are friends can still be identified by their first name and at least one of their phones when such exists. So, instead of a *phone* feature there would be a *PHONE* role relating clients with their possibly multiple phones. This can be done in description logics by encoding the primitive role *PHONE*, or indeed any primitive role, as the composition of an inverse feature with another feature, where the two features globally characterize the domain and range of the role (e.g., *phonedom* and *phoneran*). In our example, the role *PHONE* would be systematically represented as the composition (*phonedom*$^-$*.phoneran*). The above example can then be captured with a PDD by the following axiom.

$$\begin{aligned}&(\text{CLIENT-FRIEND} \sqcap \exists \mathit{phonedom}^-.\mathit{phoneran}.\top)\\ &\quad\sqsubseteq (\text{CLIENT} : \mathit{fname}, \mathit{phonedom}^-.\mathit{phoneran}.\mathit{dialnum} \rightarrow id)\end{aligned} \tag{2}$$

This axiom also expresses a key or uniqueness condition: *any pair of distinct clients for which at least one is also a friend with at least one phone will always disagree on either their first name or on all the dial numbers of their respective sets of phones.*

Furthermore, we may be interested in recording additional information for each pair of instances in the *PHONE* role, for example, the starting date of the telecom contract for a client and a phone. We can do that by reifying the *PHONE* role as the HAVING-PHONE concept, so that pairs of instances in the *PHONE* role have a one-to-one association with instances of the HAVING-PHONE concept.

$$\text{HAVING-PHONE} \sqsubseteq (\text{HAVING-PHONE} : \mathit{phonedom}, \mathit{phoneran} \rightarrow id)$$

This generalizes to arbitrary n-ary relations, which can be similarly treated in FunDL dialects [4].

Now consider where there is a need to view the set of phones for a person as a single instance which aggregates the phones, that is, as a so-called *plural entity* [5] about which additional information needs to be recorded, for example, a count of the number of such phones for a given plural entity. A role *PHONES* would relate an instance of the plural entity to the phones it aggregates. Exploiting the same encoding proposed above, the role *PHONES* would be systematically represented as the composition (*phonesdom*$^-$*.phonesran*). A feature *phonesgroup* then relates a client with its own phones, now seen as a single plural entity. A refinement of (2) accommodates this level of indirection.

$$\begin{aligned}&(\text{CLIENT-FRIEND} \sqcap \exists \mathit{phonesgroup}.\mathit{phonesdom}^-.\mathit{phoneran}.\top)\\ &\quad\sqsubseteq (\text{CLIENT} : \mathit{fname}, \mathit{phonesgroup}.\mathit{phonesdom}^-.\mathit{phonesran}.\mathit{dialnum} \rightarrow id)\end{aligned} \tag{3}$$

Note that the occurrences of inverse feature navigations, *phonesdom*$^{-}$, are now preceded by feature navigations of *phonesgroup*, and that this is not the case with subsumption (2) where inverse feature navigations happen at the start of path descriptions. Consequently, (2) can be rewritten as an axiom involving a *non-key* PFD in which inverse feature navigations are avoided, as in the following.

$$\texttt{HAVING-PHONE} \sqsubseteq (\texttt{HAVING-PHONE} : \mathit{phonedom.fname}, \mathit{phoneran.dialnum} \rightarrow \mathit{phonedom}) \quad (4)$$

However, this is not possible with subsumption (3) which requires the greater expressiveness of PDDs.

The remainder of the paper is organized as follows. The necessary background and definitions are given in Section 2 where we introduce the new FunDL dialect called *set-*$\mathcal{DLFDI}$ which replaces PFDs in the *partial-*$\mathcal{DLFDI}$ dialect of FunDL with PDDs. Our main result is presented in Section 3 in which we show that logical consequence for *set-*$\mathcal{DLFDI}$ is EXPTIME-complete when interpretations are assumed to satisfy a coherence condition. We also show that an alternative *set equality* semantics leads immediately to undecidability.

As a first application to illustrate the utility of the new dialect, we show in Section 4 how a singularity condition for a so-called *referring expression type* (Re)[1] with respect to a TBox can be diagnosed as a collection of logical implication problems. An Re denotes a set of concept descriptions in *set-*$\mathcal{DLFDI}$ that should qualify as *referring concepts*, that is, as a means of referring to elements of the domain of an interpretation. The diagnosis of singularity for an Re ensures that any interpretation of any referring concept in the set will never denote more than one element. Summary comments and an outline of open problems and future work are then given in Section 5.

2. Background and Definitions

We now formally define the artifacts introduced in our introductory comments: a new member of the FunDL family of DLs called *set-*$\mathcal{DLFDI}$ and referring expression types. The latter is a pattern language for specifying a set of referring concepts in *set-*$\mathcal{DLFDI}$ which are intended to describe the existence of individuals with complex properties. The new dialect derives from an existing dialect called *partial-*$\mathcal{DLFDI}$ by allowing the general use of set valued path descriptions in place of path functions. Doing so in the case of the PFD concept constructor (common to all existing FunDL dialects) obtains our new PDD concept constructor.[2]

Definition 1 (Referring Concepts and TBoxes in *set-*$\mathcal{DLFDI}$)**.** *Let* F *and* PC *be sets of feature names and primitive concept names, respectively. A* path description *is defined by the grammar*

$$\mathsf{Pd} ::= id \mid f.\,\mathsf{Pd} \mid f^{-1}.\,\mathsf{Pd},$$

for $f \in \mathsf{F}$*. A* concept description *is defined by the grammar in the first column of Figure 1. A* referring concept *is a concept description parsed by the last four productions, and a* TBox concept

[1]Pronounced “are-ee”.

[2]The acronyms are respectively short for *description logic with set-valued path descriptions, dependencies and inverses* and *description logic with partial functions, dependencies and inverses.*

Syntax	Semantics: Defn of $(\cdot)^{\mathcal{I}}$	
$C ::= C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$	(disjunction)
$\mid \neg C$	$\triangle^{\mathcal{I}} \setminus C^{\mathcal{I}}$	(negation)
$\mid C : \mathsf{Pd}_1, ..., \mathsf{Pd}_k \to \mathsf{Pd}$	$\{x \mid \forall y \in C^{\mathcal{I}} : \mathsf{Pd}^{\mathcal{I}}(\{x\}) \neq \emptyset \wedge \mathsf{Pd}^{\mathcal{I}}(\{y\}) \neq \emptyset \wedge$ $(\bigwedge_{i=1}^{k} \mathsf{Pd}_i^{\mathcal{I}}(\{x\}) \cap \mathsf{Pd}_i^{\mathcal{I}}(\{y\}) \neq \emptyset) \to \mathsf{Pd}^{\mathcal{I}}(\{x\}) \cap \mathsf{Pd}^{\mathcal{I}}(\{y\})\} \neq \emptyset$	(PDD)
$\mid \forall \mathsf{Pd}.C$	$\{x \mid \mathsf{Pd}^{\mathcal{I}}(\{x\}) \subseteq C^{\mathcal{I}}\}$	(value restriction)
$\mid \exists \mathsf{Pd}.C$	$\{x \mid \mathsf{Pd}^{\mathcal{I}}(\{x\}) \cap C^{\mathcal{I}} \neq \emptyset\}$	(existential quantification)
$\mid \mathrm{A}$	$\mathrm{A}^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}}$	(primitive concept; $\mathrm{A} \in \mathsf{PC}$)
$\mid C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$	(conjunction)
$\mid \{a\}$	$\{a^{\mathcal{I}}\}$	(nominal)

Figure 1: Syntax and semantics of concept descriptions.

is a concept description parsed by all but the final production, that is, are concepts free of nominals. This includes the third production, a concept constructor called a path description dependency (PDD). *A* subsumption *is an expression of the form $C_1 \sqsubseteq C_2$, where the C_i are TBox concepts, and where PDDs occur only in C_2 but not within the scope of negation.*[3] *A* terminology (TBox) $\mathcal{T}$ *consists of a finite set of subsumptions, and a* posed question $\mathcal{Q}$ *is a single subsumption.*

Semantics is defined with respect to a structure $\mathcal{I} = (\triangle^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\triangle^{\mathcal{I}}$ is a domain of objects or entities and $\cdot^{\mathcal{I}}$ an interpretation function that fixes the interpretations of primitive concept names A to be subsets of $\triangle^{\mathcal{I}}$ and feature names f to be partial functions $f^{\mathcal{I}} : \triangle^{\mathcal{I}} \to \triangle^{\mathcal{I}}$. The interpretations of path descriptions Pd *are functions* $\mathsf{Pd}^{\mathcal{I}} : 2^{\triangle^{\mathcal{I}}} \to 2^{\triangle^{\mathcal{I}}}$ *over the powerset of $\triangle^{\mathcal{I}}$ defined as follows, where $S \subseteq \triangle^{\mathcal{I}}$:*

$$\mathsf{Pd}^{\mathcal{I}}(S) = \begin{cases} S & \textit{if } \mathsf{Pd} = \text{“}\mathit{id}\text{”}, \\ \mathsf{Pd}_1^{\mathcal{I}}(\{f^{\mathcal{I}}(x) \mid x \in S\}) & \textit{if } \mathsf{Pd} = \text{“}f.\,\mathsf{Pd}_1\text{”}, \\ \mathsf{Pd}_1^{\mathcal{I}}(\{x \mid f^{\mathcal{I}}(x) \in S\}) & \textit{if } \mathsf{Pd} = \text{“}f^{-1}.\,\mathsf{Pd}_1\text{”}. \end{cases}$$

The semantics of derived concept descriptions C is defined by the centre column of Figure 1.

An interpretation $\mathcal{I}$ satisfies a subsumption $C_1 \sqsubseteq C_2$ *if $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$ and is a* model of $\mathcal{T}$, *written $\mathcal{I} \models \mathcal{T}$, if it satisfies all subsumptions in $\mathcal{T}$. Given a terminolgy $\mathcal{T}$ and posed question $\mathcal{Q}$, the* logical consequence problem *asks if $\mathcal{Q}$ is satisfied in all models of $\mathcal{T}$, written $\mathcal{T} \models \mathcal{Q}$.* □

As suggested by subsumptions (2), (3) and (4) above, PDDs adopt a *set intersection* semantics. Consequently, any subsumption that is symmetric and a simple key of the form “$C \sqsubseteq (C : \mathsf{Pd}_1, ..., \mathsf{Pd}_k \to \mathit{id})$” can be captured as an *identification constraint* (IdC) proposed in [7]. IdCs are a variant of equality-generating dependencies that, unlike in our case, correspond to an additional kind of assertion in a TBox, separate from subsumptions, that have the form “(id C $\mathsf{Pd}_1, ..., \mathsf{Pd}_k$)”. Note that path descriptions in IdCs also allow the occurrence of *test relations* “C?” denoting the identity role on the interpretation of C. A more general asymmetric version of test relations of the form “(C_1, C_2)” could be added as an additional possibility for

[3] Violating this latter condition leads immediately to undecidability [6].

path descriptions in PDDs as syntactic sugar. To suggest the simple mapping involved, consider a PDD using such an asymmetric test relation:

$$C_3 \sqsubseteq (C_4 : \mathsf{Pd}_1 .(C_1, C_2). \mathsf{Pd}_2 \rightarrow id).$$

This would be syntactic shorthand for the PDD

$$(C_3 \sqcap \forall \mathsf{Pd}_1.C_1) \sqsubseteq ((C_4 \sqcap \forall \mathsf{Pd}_1.C_2) : \mathsf{Pd}_1 . \mathsf{Pd}_2 \rightarrow id).$$

The formal definition of a referring expression type Re in the context of *set-$\mathcal{DLFDI}$* now follows. Such types were first introduced in [8] by abstracting constants and by admitting a production to express preference among WFFs free in one variable as a means of entity reference in first order knowledge bases. They have been adapted for FunDL dialects in [3] as part of a proposed semantics for JSON values as FunDL knowledge bases.

Definition 2 (Referring Expression Types). *A* referring expression type (Re) *is defined by the following grammar, where* A *is a primitive concept:*

$$Re ::= \mathrm{A} \mid \{?\} \mid \exists \mathsf{Pd}.Re \mid Re_1 \sqcap Re_2 \mid Re_1 \,;Re_2$$

The language of referring concepts inhabiting Re, $\mathcal{L}(Re)$, is given as follows:

$$\begin{aligned} \mathcal{L}(\mathrm{A}) &= \{\mathrm{A}\} \\ \mathcal{L}(\{?\}) &= \{\{a\} \mid a \text{ is a constant symbol}\} \\ \mathcal{L}(\exists \mathsf{Pd}.Re) &= \{\exists \mathsf{Pd}.C \mid C \in \mathcal{L}(Re)\} \\ \mathcal{L}(Re_1 \sqcap Re_2) &= \{C_1 \sqcap C_2 \mid C_1 \in \mathcal{L}(Re_1) \text{ and } C_2 \in \mathcal{L}(Re_2)\} \\ \mathcal{L}(Re_1; Re_2)) &= \mathcal{L}(Re_1) \cup \mathcal{L}(Re_2) \end{aligned}$$

Given a TBox $\mathcal{T}$ and referring expression type Re, the singularity problem for Re with respect to $\mathcal{T}$ *is to determine if* $|C^{\mathcal{I}}| \leq 1$ *for any* $C \in \mathcal{L}(Re)$ *and any interpretation* $\mathcal{I}$. □

For example, a referring expression type denoting possible referring concepts for CLIENT entities might be given as the following:

$$\begin{aligned} &(\text{CLIENT-FRIEND} \sqcap \exists \textit{phonesgroup.phonedom}^{-}.\textit{phoneran}.\top \\ &\qquad \sqcap \exists \textit{fname}.\{?\} \sqcap \exists \textit{phonesgroup.phonedom}^{-}.\textit{phoneran.dialnum}.\{?\}) \;; \\ &(\text{CLIENT} \sqcap \exists \textit{fname}.\{?\} \sqcap \exists \textit{lname}.\{?\}) \end{aligned} \tag{5}$$

Here, the use of ";" indicates a preference for using the first name and a phone number for referring to a client who is also a friend with at least one phone number, and by using a combination of the first name and last name otherwise (see [3] for further details). Here also, ensuring that a TBox $\mathcal{T}$ logically implies both the following subsumption as well as (3) above will be needed in any diagnosis of singularity for (5).

$$\text{CLIENT} \sqsubseteq (\text{CLIENT} : \textit{fname}, \textit{lname} \rightarrow id) \tag{6}$$

3. Decidability of Logical Consequence for *set*-$\mathcal{DLFDI}$

To simplify the following constructions, we assume that both the TBox $\mathcal{T}$ and the posed question $\mathcal{Q}$ are in normal forms defined as follows:

Definition 3 (Normal Form TBoxes and Posed Questions). *Let* A, B, *and* C *denote primitive concepts,* $\bot(= \mathrm{A} \sqcap \neg\mathrm{A})$, *or* $\top(= \mathrm{A} \sqcup \neg\mathrm{A})$. *A TBox in* normal form *consists of eight possible subsumptions:* $\mathrm{A} \sqsubseteq \mathrm{C}$, $\mathrm{A} \sqcap \mathrm{B} \sqsubseteq \mathrm{C}$, $\mathrm{A} \sqsubseteq \mathrm{B} \sqcup \mathrm{C}$, $\mathrm{A} \sqsubseteq \forall f.\mathrm{B}$, $\mathrm{A} \sqsubseteq \exists f.\mathrm{B}$, $\mathrm{A} \sqsubseteq \forall f^{-}.\mathrm{B}$, $\mathrm{A} \sqsubseteq \exists f^{-}.\mathrm{B}$, *and* $\mathrm{A} \sqsubseteq \mathrm{B} : \mathsf{Pd}_1, \ldots, \mathsf{Pd}_k \rightarrow \mathsf{Pd}$. *TBoxes that do not mention a PDD, that is, do not have a subsumption of the form* $\mathrm{A} \sqsubseteq \mathrm{B} : \mathsf{Pd}_1, \ldots, \mathsf{Pd}_k \rightarrow \mathsf{Pd}$, *are called* simple.

A posed question in normal form has the form $\mathrm{A} \sqsubseteq \mathrm{B} : \mathsf{Pd}_1, \ldots, \mathsf{Pd}_k \rightarrow \mathsf{Pd}$ *or is a simple subsumption* $\mathrm{A} \sqsubseteq \mathrm{B}$. □

It is easy to see that for every TBox and posed question there is a conservative extension of the TBox that preserves entailment.

Logical consequence for *partial*-$\mathcal{DLFDI}$ was shown in [9] to be undecidable, which will therefore also be the case with *set*-$\mathcal{DLFDI}$. One way to regain decidability suggested in [9] is to adopt a *coherency* condition on a TBox $\mathcal{T}$. The condition requires that $\mathcal{T}$ implicitly contains a set of subsumptions of the following form, where C ranges over TBox concepts in the language of $\mathcal{T}$, and where f^{-} explicitly appears in a concept $\exists f^{-}.\mathrm{D}$ in the normalized TBox $\mathcal{T}$ (we call such features *interesting*):

$$\exists f^{-}.C \sqsubseteq \forall f^{-}.C \qquad (*)$$

Note that, while we assume these subsumptions are part of every TBox, they are not part of the input to our decision procedure since the coherency condition is an integral part of the procedure itself. For the remainder of the paper we assume the coherency condition for every *set*-$\mathcal{DLFDI}$ TBox.

Lemma 4. *Let* $\mathcal{T}$ *be a set-*$\mathcal{DLFDI}$ *TBox and* $\mathcal{Q} = \mathrm{A} \sqsubseteq \mathrm{B}$ *a posed question. For any counterexample* $\mathcal{J}$ *to* $\mathcal{T} \models \mathcal{Q}$*, there is also a tree-shaped counterexample* $\mathcal{I}$ *for which: (a)* $\mathcal{I} \models \mathcal{T}$*, (b)* $\mathcal{I}$ *is rooted by a witness* $o \in (\mathrm{A} \sqcap \neg\mathrm{B})^{\mathcal{I}}$*, and (c) for any* $f \in \mathsf{F}$*, there is no element of* $\triangle^{\mathcal{I}}$ *that has two or more incoming features* f*.*

Proof (sketch): The tree-shaped model is constructed by unravelling the original counterexample model, starting from the witness o, as follows. For every $f \in \mathsf{F}$: (i) all f successors of an object are added to $\mathcal{I}$ and subsequently unravelled, and (ii) exactly one of the interesting f-predecessors is also added and unravelled (note that the single f-predecessor can be the parent of the current object). It follows from $(*)$ that, in such an unravelling, all simple subsumptions in $\mathcal{T}$ must be satisfied (by inspection) and also that, since *no object in* $\triangle^{\mathcal{I}}$ *that has two or more incoming features* f *for any* $f \in \mathsf{F}$, all PDDs in $\mathcal{T}$ hold vacuously. □

Note that the Pd paths starting from the witness o uniquely identify objects in $\triangle^{\mathcal{I}}$.

Lemma 5. *Let* $\mathcal{T}$ *be a set-*$\mathcal{DLFDI}$ *TBox and* $\mathcal{Q} = \mathrm{A} \sqsubseteq \mathrm{B} : \mathsf{Pd}_1, \ldots, \mathsf{Pd}_k \rightarrow \mathsf{Pd}_0$ *a posed question. For any counterexample* $\mathcal{J}$ *to* $\mathcal{T} \models \mathcal{Q}$*, there is also a counterexample* $\mathcal{I}$*, called a* two-tree counterexample*, constructed as follows:*

1. *let* $\mathcal{I}_1$ *be an unravelling of* $\mathcal{J}$ *from an object* $o_1 \in \mathrm{A}^{\mathcal{J}}$*;*

2. *let $\mathcal{I}_2$ be an unravelling of $\mathcal{J}$ from an object $o_2 \in \mathrm{B}^{\mathcal{J}}$; and*
3. *let $\mathcal{I}$ be a disjoint union of $\mathcal{I}_1$ and $\mathcal{I}_2$ modified to agree on a path* Pd *starting in o_1 and o_2, respectively, if and only if* $\mathsf{Pd}^{\mathcal{J}}(\{o_1\}) \cap \mathsf{Pd}^{\mathcal{J}}(\{o_2\}) \neq \emptyset$.

Proof (sketch): The unravellings $\mathcal{I}_1$ and $\mathcal{I}_2$ are as in Lemma 4. The domain $\triangle^{\mathcal{I}}$ is then the disjoint union of the domains of $\mathcal{I}_1$ and $\mathcal{I}_2$ factored by the equivalence relation generated by the path agreements. Note that it is sufficient to only consider path agreements that are *symmetric* with respect to o_1 and o_2; this is a consequence of $(*)$ and of the syntactic structure of the PDD concept constructor. □

The properties established by Lemmata 4 and 5 allow one to reduce the existence of an posed question-appropriate counterexample as test of satisfiability of a formula in the Ackermann class prefix $\exists^*\forall\exists^*$ [10]. We use the Skolemized version of the problem that replaces the inner existential variables by appropriate unary function symbols and the outer ones by constants.

The main intuition is that we use a single term to denote *the (possibly two) corresponding objects* in the unravellings $\mathcal{I}_1$ from o_1 and $\mathcal{I}_2$ from o_2, if they exist. We introduce the following function symbols and unary predicates to facilitate this construction:

- function symbols $\mathsf{f}(x)$ and $\bar{\mathsf{f}}(x)$ for every feature f appearing in $\mathcal{T}$, standing for f and (inverse of) f^-, respectively;
- unary predicates $\mathsf{P}^L_{\mathrm{A}}(x)$ and $\mathsf{P}^R_{\mathrm{A}}(x)$ whose interpretation will consist of individuals corresponding to individuals in $\mathrm{A}^{\mathcal{I}}$ for every primitive concept in $\mathcal{T}$;
- unary predicates $\mathsf{N}^L(x)$ and $\mathsf{N}^R(x)$ whose interpretation will consist of individuals that *exist* in $\mathcal{I}$ (note that *set-$\mathcal{DLFDI}$* uses *partial features* while the Skolem functions are total);
- unary predicates $\mathsf{Eq}^{\mathsf{Pd}}(x)$ that assert equality between the two paths originating in the corresponding objects in the two unravellings.

The encoding itself relies on the following auxiliary axioms (and auxiliary predicate definitions), where X ranges over $\{L, R\}$:

Features and inverse features: There cannot be objects with the following f–$\bar{\mathsf{f}}$ patterns:

$$\forall x.\neg\mathsf{N}^X(\mathsf{f}.\bar{\mathsf{f}}(x)) \text{ and } \forall x.\neg\mathsf{N}^X(\bar{\mathsf{f}}.\mathsf{f}(x)).$$

We have such an axiom for every $f \in \mathsf{F}$. This restriction corresponds to functionality of features and to disallowing two or more incoming f's in Lemma 4;

Equalities and paths: The following axioms describe how equalities (based on intersection) propagate along PDs:

$$\begin{array}{l}
\forall x.(\mathsf{N}^L(x) \wedge \mathsf{N}^L(\mathsf{f}(x)) \wedge \mathsf{N}^R(x) \wedge \mathsf{N}^R(\mathsf{f}(x))) \rightarrow (\mathsf{Eq}^{\mathsf{Pd}}(\mathsf{f}(x)) \leftrightarrow \mathsf{Eq}^{f.\,\mathsf{Pd}}(x)) \\
\forall x.(\mathsf{N}^L(x) \wedge \mathsf{N}^L(\bar{\mathsf{f}}(x)) \wedge \mathsf{N}^R(x) \wedge \mathsf{N}^R(\bar{\mathsf{f}}(x))) \rightarrow (\mathsf{Eq}^{\mathsf{Pd}}(x) \leftrightarrow \mathsf{Eq}^{f.\,\mathsf{Pd}}(\bar{\mathsf{f}}(x))) \\
\forall x.(\mathsf{N}^L(x) \wedge \mathsf{N}^L(\mathsf{f}(x)) \wedge \mathsf{N}^R(x) \wedge \mathsf{N}^R(\mathsf{f}(x))) \rightarrow (\mathsf{Eq}^{\mathsf{Pd}}(x) \leftrightarrow \mathsf{Eq}^{f^-.\,\mathsf{Pd}}(\mathsf{f}(x))) \\
\forall x.(\mathsf{N}^L(x) \wedge \mathsf{N}^L(\bar{\mathsf{f}}(x)) \wedge \mathsf{N}^R(x) \wedge \mathsf{N}^R(\bar{\mathsf{f}}(x))) \rightarrow (\mathsf{Eq}^{\mathsf{Pd}}(\bar{\mathsf{f}}(x)) \leftrightarrow \mathsf{Eq}^{f^-.\,\mathsf{Pd}}(x))
\end{array}$$

Equalities and functionality: The following axioms simulate the functionality of features:

$$\forall x.\mathsf{N}^L(x) \land \mathsf{N}^L(\mathsf{f}(x)) \land \mathsf{N}^R(x) \land \mathsf{N}^R(\mathsf{f}(x)) \land \mathsf{Eq}^{id}(x) \to \mathsf{Eq}^{id}(\mathsf{f}(x))$$
$$\forall x.\mathsf{N}^L(x) \land \mathsf{N}^L(\bar{\mathsf{f}}(x)) \land \mathsf{N}^R(x) \land \mathsf{N}^R(\bar{\mathsf{f}}(x)) \land \mathsf{Eq}^{id}(\bar{\mathsf{f}}(x)) \to \mathsf{Eq}^{id}(x)$$

Equalities and concept membership: The following axioms simulate the effects of equality on concept membership:

$$\forall x.\mathsf{N}^L(x) \land \mathsf{N}^R(x) \land \mathsf{Eq}^{id}(x) \to (\mathsf{P}^L_{\mathrm{C}}(x) \leftrightarrow \mathsf{P}^R_{\mathrm{C}}(x))$$

where C ranges over all primitive concepts in $\mathcal{T} \cup \mathcal{Q}$.

Path existence: The $\mathsf{N}^X_{\mathsf{Pd}}(x)$ predicate asserts that all objects on the path Pd starting in x exist (on side X):

$$\begin{array}{l}
\mathsf{N}^X_{id}(x) \to \mathsf{N}^X(x)\\
\mathsf{N}^X_{f.\,\mathsf{Pd}}(x) \to (\mathsf{N}^X(x) \land \mathsf{N}^X_{\mathsf{Pd}}(\mathsf{f}(x)))\\
\mathsf{N}^X_{f^-.\,\mathsf{Pd}}(x) \to (\mathsf{N}^X(x) \land \mathsf{N}^X_{\mathsf{Pd}}(\bar{\mathsf{f}}(x)))
\end{array}$$

We call the above collection of axioms Π. With the above preparation, we are ready to map subsumptions in normalized TBoxes to the following axioms:

Simple TBox subsumptions:

$$\begin{array}{lll}
\mathrm{A} \sqsubseteq \mathrm{C} & \mapsto & \forall x.(\mathsf{N}^X(x) \land \mathsf{P}^X_{\mathrm{A}}(x)) \to \mathsf{P}^X_{\mathrm{C}}(x)\\
\mathrm{A} \sqcap \mathrm{B} \sqsubseteq \mathrm{C} & \mapsto & \forall x.(\mathsf{N}^X(x) \land \mathsf{P}^X_{\mathrm{A}}(x) \land \mathsf{P}^X_{\mathrm{B}}(x))) \to \mathsf{P}^X_{\mathrm{C}}(x)\\
\mathrm{A} \sqsubseteq \mathrm{B} \sqcup \mathrm{C} & \mapsto & \forall x.(\mathsf{N}^X(x) \land \mathsf{P}^X_{\mathrm{A}}(x)) \to (\mathsf{P}^X_{\mathrm{B}}(x) \lor \mathsf{P}^X_{\mathrm{C}}(x))\\
\mathrm{A} \sqsubseteq \forall f.\mathrm{B} & \mapsto & \forall x.(\mathsf{N}^X(x) \land \mathsf{N}^X(\mathsf{f}(x)) \land \mathsf{P}^X_{\mathrm{A}}(x)) \to \mathsf{P}^X_{\mathrm{B}}(\mathsf{f}(x)),\\
 & & \forall x.(\mathsf{N}^X(\bar{\mathsf{f}}(x)) \land \mathsf{N}^X(x) \land \mathsf{P}^X_{\mathrm{A}}(\bar{\mathsf{f}}(x)) \to \mathsf{P}^X_{\mathrm{B}}(x)\\
\mathrm{A} \sqsubseteq \exists f.\mathrm{B} & \mapsto & \forall x.(\mathsf{N}^X(x) \land \mathsf{P}^X_{\mathrm{A}}(x)) \to (\mathsf{N}^X(\mathsf{f}(x)) \land \mathsf{P}^X_{\mathrm{B}}(\mathsf{f}(x))), x \neq \bar{\mathsf{f}}(y)\\
 & & \forall x.(\mathsf{N}^X(\bar{\mathsf{f}}(x)) \land \mathsf{P}^X_{\mathrm{A}}(\bar{\mathsf{f}}(x)) \to (\mathsf{N}^X(x) \land \mathsf{P}^X_{\mathrm{B}}(x))\\
\mathrm{A} \sqsubseteq \forall f^-.\mathrm{B} & \mapsto & \forall x.(\mathsf{N}^X(x) \land \mathsf{N}^X(\mathsf{f}(x)) \land \mathsf{P}^X_{\mathrm{A}}(\mathsf{f}(x))) \to \mathsf{P}^X_{\mathrm{B}}(x),\\
 & & \forall x.(\mathsf{N}^X(\bar{\mathsf{f}}(x)) \land \mathsf{N}^X(x) \land \mathsf{P}^X_{\mathrm{A}}(x)) \to \mathsf{P}^X_{\mathrm{B}}(\bar{\mathsf{f}}(x))\\
\mathrm{A} \sqsubseteq \exists f^-.\mathrm{B} & \mapsto & \forall x.(\mathsf{N}^X(\mathsf{f}(x)) \land \mathsf{P}^X_{\mathrm{A}}(\mathsf{f}(x))) \to (\mathsf{N}^X(x) \land \mathsf{P}^X_{\mathrm{B}}(x)),\\
 & & \forall x.(\mathsf{N}^X(x) \land \mathsf{P}^X_{\mathrm{A}}(x)) \to (\mathsf{N}^X(\bar{\mathsf{f}}(x)) \land \mathsf{P}^X_{\mathrm{B}}(\bar{\mathsf{f}}(x))), x \neq \mathsf{f}(y)
\end{array}$$

Note that since a single feature f can be *coded* using both f and (inverse of) $\bar{\mathsf{f}}$, we need two axioms for both the value and existential restrictions. Also, for the existential restrictions of the form $\exists f.B$ and $\exists f^-.B$ we need to ensure that $\bar{\mathsf{f}}$ is not used when x is of the form $\mathsf{f}(y)$ (and vice versa as that would create a superfluous inverse). Since the Ackermann prefix class disallows explicit equalities, we need to simulate this inequality by explicitly listing all the possible terms for x that do not start with f. This is always possible since there are only finitely many features in $\mathcal{T}$.

Equalities and PFDs: To simulate the effects of PDDs, it is sufficient to record their effects between the left and the right sides of the counterexample as follows:

$$\begin{array}{l}
\mathrm{A} \sqsubseteq \mathrm{B} : \mathsf{Pd}_1, \ldots, \mathsf{Pd}_k \to \mathsf{Pd} \;\mapsto\\
\qquad \forall x.(\mathsf{N}^L(x) \land \mathsf{N}^R(x) \land \mathsf{P}^L_{\mathrm{A}} \land \mathsf{P}^R_{\mathrm{B}} \land \bigwedge_{i=1}^k \mathsf{Eq}^{\mathsf{Pd}_i}(x)) \to \mathsf{Eq}^{\mathsf{Pd}}(x)\\
\qquad \forall x.(\mathsf{N}^L(x) \land \mathsf{N}^R(x) \land \mathsf{P}^R_{\mathrm{A}} \land \mathsf{P}^L_{\mathrm{B}} \land \bigwedge_{i=1}^k \mathsf{Eq}^{\mathsf{Pd}_i}(x)) \to \mathsf{Eq}^{\mathsf{Pd}}(x)
\end{array}$$

We call the set of these axioms $\Pi_{\mathcal{T}}$. The posed questions then map to the following assertions about witnesses of non-entailment:

Posed questions: The (ground) axioms represent a counterexample to the posed question; the constant 0 stands for the o_1 (and o_2) objects in Lemmata 4 and 5.

$$\begin{array}{lcl}
\mathrm{A} \sqsubseteq \mathrm{B} & \mapsto & \mathsf{N}^L(0), \mathsf{P}^L_{\mathrm{A}}(0), \neg\mathsf{P}^L_{\mathrm{B}}(0) \\
\mathrm{A} \sqsubseteq \mathrm{B} : \mathsf{Pd}_1, \ldots, \mathsf{Pd}_k \to \mathsf{Pd} & \mapsto & \mathsf{P}^L_{\mathrm{A}}(0), \mathsf{P}^R_{\mathrm{B}}(0), \\
 & & \mathsf{N}^L_{\mathsf{Pd}_1}(0), \ldots, \mathsf{N}^L_{\mathsf{Pd}_k}(0), \mathsf{N}^L_{\mathsf{Pd}}(0), \\
 & & \mathsf{N}^R_{\mathsf{Pd}_1}(0), \ldots, \mathsf{N}^R_{\mathsf{Pd}_k}(0), \mathsf{N}^R_{\mathsf{Pd}}(0), \\
 & & \mathsf{Eq}^{\mathsf{Pd}_1}(0), \ldots, \mathsf{Eq}^{\mathsf{Pd}_k}(0), \neg\mathsf{Eq}^{\mathsf{Pd}}(0)
\end{array}$$

We call the set of these axioms $\Pi_{\neg\mathcal{Q}}$.

To show correctness of the above construction, we use the following auxiliary definition that maps feature paths in models of *set-$\mathcal{DLFDI}$* to paths in models of the corresponding Ackermann formula (and back).

$$\mathsf{pm}(id) = 0, \mathsf{pm}(f.\,\mathsf{Pd}) = \mathsf{f}(\mathsf{pm}(\mathsf{Pd})), \text{ and } \mathsf{pm}(f^-.\,\mathsf{Pd}) = \bar{\mathsf{f}}(\mathsf{pm}(\mathsf{Pd}))$$

Theorem 6. *Let $\mathcal{T}$ and $\mathcal{Q}$ be a set-$\mathcal{DLFDI}$ TBox and a posed question, respectively, both in normal form. Then $\mathcal{T} \models \mathcal{Q}$ if and only if $\Pi \cup \Pi_{\mathcal{T}} \cup \Pi_{\neg\mathcal{Q}}$ is unsatisfiable.*

Proof (sketch): For posed question $\mathcal{Q}$ of the form $\mathrm{A} \sqsubseteq \mathrm{B}$ and a tree-shaped counter-example $\mathcal{I}$ starting from o (see Lemma 4) we create a model of $\Pi \cup \Pi_{\mathcal{T}} \cup \Pi_{\neg\mathcal{Q}}$ as follows: we assert

- $\mathsf{N}^L(\mathsf{pm}(\mathsf{Pd}))$ for every path $o.\,\mathsf{Pd}^{\mathcal{I}}$ that exists in $\mathcal{I}$, and
- $\mathsf{P}^L_{\mathrm{A}}(\mathsf{pm}(\mathsf{Pd}))$ whenever $o.\,\mathsf{Pd}^{\mathcal{I}} \in \mathrm{A}^{\mathcal{I}}$

in $\mathcal{I}$ and verify that all formulæ in $\Pi \cup \Pi_{\mathcal{T}} \cup \Pi_{\neg\mathcal{Q}}$ are true.

Similarly, given a model of $\Pi \cup \Pi_{\mathcal{T}} \cup \Pi_{\neg\mathcal{Q}}$ we construct $\mathcal{I}$ as follows:

- $\triangle^{\mathcal{I}} = \{o_{\mathsf{Pd}} \mid \mathsf{N}^L(\mathsf{pm}(\mathsf{Pd}))$ holds$\}$;
- $\mathrm{A}^{\mathcal{I}} = \{o_{\mathsf{Pd}} \mid \mathsf{N}^L(\mathsf{pm}(\mathsf{Pd})) \wedge \mathsf{P}^L_{\mathrm{A}}(\mathsf{pm}(\mathsf{Pd}))$ holds$\}$; and
- $f^{\mathcal{I}} = \{(o_{\mathsf{Pd}}, o_{f.\,\mathsf{Pd}}) \mid \mathsf{N}^L(\mathsf{f}.\,\mathsf{pm}(\mathsf{Pd}))\} \cup \{(o_{\mathsf{Pd}}, o_{f^-.\,\mathsf{Pd}}) \mid \mathsf{N}^L(\bar{\mathsf{f}}.\,\mathsf{pm}(\mathsf{Pd}))\}$,

and verify that $\mathcal{I}$ is a model of $\mathcal{T}$ and o_{id} falsifies $\mathcal{Q}$.

For a posed question $\mathrm{A} \sqsubseteq \mathrm{B} : \mathsf{Pd}_1, \ldots, \mathsf{Pd}_k \to \mathsf{Pd}_0$ and a two-tree counterexample $\mathcal{I}$ starting in o_1 and o_2 (see Lemma 5) we create a model of $\Pi \cup \Pi_{\mathcal{T}} \cup \Pi_{\neg\mathcal{Q}}$ as follows: we assert

- $\mathsf{N}^L(\mathsf{pm}(\mathsf{Pd}))$ for every path $o_1.\,\mathsf{Pd}^{\mathcal{I}}$ that exists in $\mathcal{I}$;
- $\mathsf{P}^L_{\mathrm{A}}(\mathsf{pm}(\mathsf{Pd}))$ whenever $o_1.\,\mathsf{Pd}^{\mathcal{I}} \in \mathrm{A}^{\mathcal{I}}$;
- $\mathsf{N}^R(\mathsf{pm}(\mathsf{Pd}))$ for every path $o_2.\,\mathsf{Pd}^{\mathcal{I}}$ that exists in $\mathcal{I}$;
- $\mathsf{P}^R_{\mathrm{A}}(\mathsf{pm}(\mathsf{Pd}))$ whenever $o_2.\,\mathsf{Pd}^{\mathcal{I}} \in \mathrm{A}^{\mathcal{I}}$; and
- $\mathsf{Eq}^{id}(\mathsf{pm}(\mathsf{Pd}))$ whenever $\mathsf{Pd}^{\mathcal{I}}(o_1) \cap \mathsf{Pd}^{\mathcal{I}}(o_2) \neq \emptyset$.

We extend this to the remaining auxiliary predicates $\mathsf{Eq}^{\mathsf{Pd}}$ and $\mathsf{N}^X_{\mathsf{Pd}}$ in a natural way and verify that this yields the required model of $\Pi \cup \Pi_{\mathcal{T}} \cup \Pi_{\neg\mathcal{Q}}$. The other direction constructs a two-tree model of $\mathcal{T}$ that falsifies $\mathcal{Q}$ using the construction for $\mathrm{A} \sqsubseteq \mathrm{B}$ twice (once for L starting in o^L_{id} and once for R starting in o^R_{id}) and equating paths $o^L_{id}.\,\mathsf{Pd}$ and $o^R_{id}.\,\mathsf{Pd}$ for which $\mathsf{Eq}^{\mathsf{Pd}}(0)$ holds in the model of $\Pi \cup \Pi_{\mathcal{T}} \cup \Pi_{\neg\mathcal{Q}}$. That yields a two-tree model of $\mathcal{T}$ in which o^L_{id} and o^R_{id} witness the violation of $\mathcal{Q}$. □

Corollary 7. *Let $\mathcal{T}$ and $\mathcal{Q}$ be a set-$\mathcal{DLFDI}$ TBox and a posed question, respectively. Then $\mathcal{T} \models \mathcal{Q}$ is decidable and complete for EXPTIME.*

Proof (sketch): It is easy to see from our construction that the size $|\Pi \cup \Pi_{\mathcal{T}} \cup \Pi_{\neg \mathcal{Q}}|$ is polynomial in $|\mathcal{T}| + |\mathcal{Q}|$. The result then follows from EXPTIME bound for satisfiability of Ackermann prefix formulæ [11] and closure of EXPTIME under complement. □

3.1. Set Equality Semantics and Undecidability

Here, we consider where a *set semantics* for PDDs is assumed, and briefly outline how this leads to undecidability for the logical consequence problem. In particular, now assume the semantics for a PDD concept $C : \mathsf{Pd}_1, ..., \mathsf{Pd}_k \to \mathsf{Pd}$ is given as follows:

$$\{x \mid \forall y \in C^{\mathcal{I}} : (\textstyle\bigwedge_{i=1}^{k} \mathsf{Pd}_i^{\mathcal{I}}(\{x\}) = \mathsf{Pd}_i^{\mathcal{I}}(\{y\})) \to \mathsf{Pd}^{\mathcal{I}}(\{x\}) = \mathsf{Pd}^{\mathcal{I}}(\{y\})\}$$

Theorem 8. *Let $\mathcal{T}$ be a set-$\mathcal{DLFDI}$ TBox in normal form and $\mathcal{Q}$ a posed question. Then, under the set equality semantics for PDDs, $\mathcal{T} \models \mathcal{Q}$ is undecidable.*

Proof (sketch): The subsumptions $\mathrm{A} \sqsubseteq \mathrm{A} : f^{-} \to id$ together with $\top \sqsubseteq \forall f.\bot$ and $\top \sqsubseteq \exists g.\mathrm{A}$ make the primitive concept A to behave as a nominal (i.e., contains exactly one object in every model of the above subsumptions). Subsequently asserting, for three such "nominals", A, B, and C, that $\mathrm{A} \sqsubseteq \forall h.\mathrm{B} \sqcap \forall k.\mathrm{C}$ and $\mathrm{B} \sqsubseteq \forall k.\mathrm{C}$ postulates the existence of a triangle in every model which allows us to apply the construction in [12, Section 5], yielding undecidability using a tiling argument. □

4. Referring Expression Types and Singularity Diagnosis

We now show how the test for singularity of an Re for a given TBox $\mathcal{T}$ presented in [3] is easily adapted for *set-$\mathcal{DLFDI}$*. This is achieved by appeal to a sequence of logical consequence problems for subsumptions expressing dependencies with PDDs that are induced by a given Re. The subsumptions derive from the following normalization.

Definition 9 (Normalized Referring Expression Types). *We write* $\mathsf{Norm}(Re)$ *to refer to an exhaustive application of the following rewrite rules:*

$$\begin{aligned} Re \sqcap (Re_1; Re_2) &\mapsto Re \sqcap Re_1; Re \sqcap Re_2 \\ (Re_1; Re_2) \sqcap Re &\mapsto Re_1 \sqcap Re; Re_2 \sqcap Re \\ \exists \mathsf{Pd}.(Re_1; Re_2) &\mapsto \exists \mathsf{Pd}.Re_1; \exists \mathsf{Pd}.Re_2 \end{aligned}$$

□

The definition of Norm is an adaptation of referring expression type normalization in [8] with the following consequences: (1) $\mathcal{L}(Re) = \mathcal{L}(\mathsf{Norm}(Re))$, and (2) all *preference operators* (";") are at the top level of $\mathsf{Norm}(Re)$. We call the maximal ";"-free parts of $\mathsf{Norm}(Re)$ *preference-free components*. The following auxiliary functions will then be used in formulating subsumptions with PDDs to statically test for singularity of each preference free component.

$$\begin{aligned} \mathsf{Pfs}(\{?\}) &= \{id\} & \mathsf{Con}(\{?\}) &= \top \\ \mathsf{Pfs}(\mathrm{A}) &= \{\,\} & \mathsf{Con}(\mathrm{A}) &= \mathrm{A} \\ \mathsf{Pfs}(\exists \mathsf{Pd}.Re) &= \{\mathsf{Pd} \,.\, \mathsf{Pd} \mid \mathsf{Pd} \in \mathsf{Pfs}(Re)\} & \mathsf{Con}(\exists \mathsf{Pd}.Re) &= \exists \mathsf{Pd}.\mathsf{Con}(Re) \\ \mathsf{Pfs}(Re_1 \sqcap Re_2) &= \mathsf{Pfs}(Re_1) \cup \mathsf{Pfs}(Re_2) & \mathsf{Con}(Re_1 \sqcap Re_2) &= \mathsf{Con}(Re_1) \sqcap \mathsf{Con}(Re_2) \end{aligned}$$

The functions extract a set of paths leading to nominals and a FunDL concept from the preference-free referring expression type. Altogether, the singularity test in [8, Theorem 20] will now apply:

Theorem 10. *Let $\mathcal{T}$ be a TBox and Re a referring expression type. Then all referring concepts in $\mathcal{L}(Re)$ are singular with respect to $\mathcal{T}$ if $\mathcal{T} \models \mathsf{Con}(Re') \sqsubseteq (\mathsf{Con}(Re') : \mathsf{Pfs}(Re') \rightarrow id)$ for every preference-free component Re' of $\mathsf{Norm}(Re)$.*

For example, a test for singularity of (5) with respect to TBox $\mathcal{T}$ would require the following two subsumptions to be logical consequences of $\mathcal{T}$ (and note that subsumptions (3) and (6) above also being logical consequences would be important factors in establishing this):

$$\mathcal{T} \models C_1 \sqsubseteq (C_1 : \mathit{fname}, \mathit{lname} \rightarrow id)$$
$$\mathcal{T} \models C_2 \sqsubseteq (C_2 : \mathit{fname}, \mathit{phonesgroup}.\mathit{phonedom}^{-}.\mathit{phoneran}.\mathit{dialnum} \rightarrow id)$$

where $C_1 = \texttt{CLIENT} \sqcap \exists \mathit{fname}.\top \sqcap \exists \mathit{lname}.\top$ and $C_2 = \texttt{CLIENT-FRIEND} \sqcap \exists \mathit{fname}.\top \sqcap \exists \mathit{phonesgroup}.\mathit{phonedom}^{-}.\mathit{phoneran}.\top \sqcap \exists \mathit{phonesgroup}.\mathit{phonedom}^{-}.\mathit{phoneran}.\mathit{dialnum}.\top$.

5. Summary and Future Work

The primary incentive for *set-$\mathcal{DLFDI}$* stems from an outline of future work in [3] which recognized the need for plural entities in formally capturing JSON array values and for the extension to referring expression types to accommodate references to such entities. Subsumption (4) and Re (5) in our introduction and definitions sections are derived from a case considered in this earlier work.

The diagnosis of singularity for referring expression types is only one of a number of possibilities. Other criteria are also important and remain avenues for further work on *set-$\mathcal{DLFDI}$*, in particular that concern the more general issues of *identification resolution* in query answering for backend data sources such as SQL engines, as explored in [13]. One example considered in [8] concerns *strong identification* (in contrast to singularity, an issue in *weak identification*): to determine if any pair of *distinct* referring concepts in $\mathcal{L}(Re)$ must refer to distinct domain elements, that is, where reasoning about *inequality* is also important. This becomes a more complicated issue in *set-$\mathcal{DLFDI}$*.

There are also a number of open problems and research issues that remain for *set-$\mathcal{DLFDI}$*. We end with a couple that are more pressing:

- On alternative semantics for PDDs: we have outlined how a set equality semantics leads to undecidability in Subsection 3.1. However, an alternative *non-empty set equality* semantics seems to be straightforward, with an analogous proof of decidability to our non-empty intersection semantics. The case is far less clear, however, for a "mixed mode" in which a PDD could involve both.
- On undecidability of *knowledge base* (KB) consistency (deriving from [12, Section 5]): as with PFDs, it is necessary to impose restrictions on the use of PDDs for *set-$\mathcal{DLFDI}$* KBs that have ABoxes as well as TBoxes to ensure decidability of KB consistency. This has been explored extensively for Horn fragments of FunDL dialects and even to fragments with KB consistency in PTIME [14]. Obtaining analogous results for *set-$\mathcal{DLFDI}$* is desirable.

References

[1] S. McIntyre, D. Toman, G. E. Weddell, FunDL - A family of feature-based description logics, with applications in querying structured data sources, in: Description Logic, Theory Combination, and All That - Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday, 2019, pp. 404–430.

[2] D. Toman, G. E. Weddell, Using feature-based description logics to avoid duplicate elimination in object-relational query languages, Künstliche Intell. 34 (2020) 355–363. URL: https://doi.org/10.1007/s13218-020-00666-7. doi:10.1007/s13218-020-00666-7.

[3] A. Borgida, E. Franconi, D. Toman, G. E. Weddell, Understanding document data sources using ontologies with referring expressions, in: H. Aziz, D. Corrêa, T. French (Eds.), AI 2022: Advances in Artificial Intelligence - 35th Australasian Joint Conference, AI 2022, Perth, WA, Australia, December 5-8, 2022, Proceedings, volume 13728 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 367–380.

[4] A. Artale, E. Franconi, R. Peñaloza, F. Sportelli, A decidable very expressive description logic for databases, in: ISWC 2017 - 16th International Semantic Web Conference, volume 10587, Springer, 2017, pp. 37–52.

[5] E. Franconi, A treatment of plurals and plural quantifications based on a theory of collections, Minds Mach. 3 (1993) 453–474.

[6] D. Toman, G. Weddell, On Keys and Functional Dependencies as First-Class Citizens in Description Logics, in: Proc. of Int. Joint Conf. on Automated Reasoning (IJCAR), 2006, pp. 647–661.

[7] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Path-Based Identification Constraints in Description Logics, in: KR, 2008, pp. 231–241.

[8] A. Borgida, D. Toman, G. Weddell, On referring expressions in query answering over first order knowledge bases, in: Proc. Principles of Knowledge Representation and Reasoning, KR 2016, 2016, pp. 319–328.

[9] D. Toman, G. E. Weddell, On the interaction between inverse features and path-functional dependencies in description logics, in: Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI), 2005, pp. 603–608.

[10] W. Ackermann, Uber die Erfullbarkeit gewisser Zahlausdrucke, Mathematische Annalen 100 (1928) 638–649.

[11] M. Fürer, Alternation and the Ackermann case of the decision problem, L'Enseignement Mathematique (1981).

[12] D. Toman, G. E. Weddell, On keys and functional dependencies as first-class citizens in description logics, J. Aut. Reasoning 40 (2008) 117–132.

[13] A. Borgida, D. Toman, G. E. Weddell, On referring expressions in information systems derived from conceptual modelling, in: I. Comyn-Wattiau, K. Tanaka, I. Song, S. Yamamoto, M. Saeki (Eds.), Conceptual Modeling - 35th International Conference, ER 2016, volume 9974 of *Lecture Notes in Computer Science*, 2016, pp. 183–197.

[14] S. McIntyre, A. Borgida, D. Toman, G. E. Weddell, On limited conjunctions and partial features in parameter-tractable feature logics, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, 2019, pp. 2995–3002.