Structural Equality Generating Dependencies and Definite Descriptions

Extended Abstract of University of Waterloo Technical Report CS-2025-05 [1].

David Toman[1], Grant Weddell[1]

[1]Cheriton School of Computer Science, University of Waterloo, 200 University Ave W., Waterloo, ON N2L 3G1, Canada

Abstract

We introduce a very general variety of path description dependencies (PDDs) for an expressive dialect of the FunDL family of description logics called structural PDDs. In general, PDDs enable capturing equality generating dependencies for an ontology in a progressively more fine-grained manner, starting with equality implied by simple alignment of facts about entities through to new structural PDDs in which equality only follows according to a structured alignment of non-empty sets of facts about an entity. We show that logical consequence for this new FunDL dialect is decidable if a given ontology appeals to an exclusive use of structural PDDs, but that logical consequence becomes undecidable when more course grained varieties of PDDs are also allowed in the ontology. An extension to a referring expression type language for defining concepts in this description logic to serve as referring expressions that depend on structural identification is also presented and is tied to a diagnosis of a singularity condition for such concepts to logical consequence of PDDs for an ontology.

Keywords

referring expressions, path description dependencies, structural equality

1. Introduction

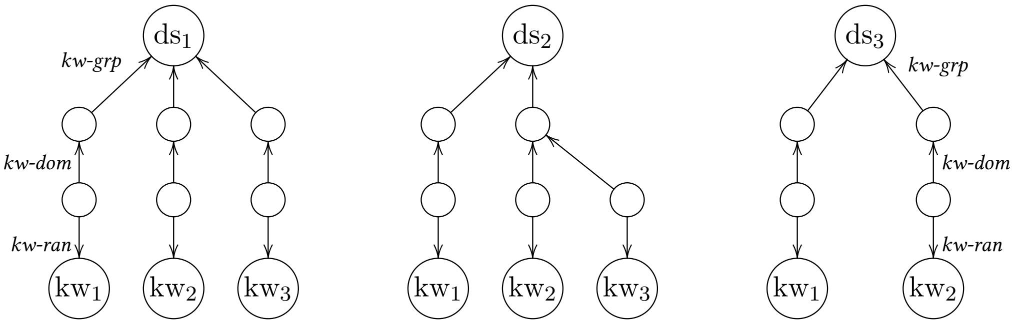
Structured data sources abound, and ontology based data access is all about querying such sources via an ontological understanding of their content. Here, effective ways to communicate answers to queries will depend critically on communicating references to underlying entities via referring expressions, also called definite descriptions [2]. Earlier work has introduced the notion of referring expression types, each of which will define a set of possible referring expressions [3, 4]. That such descriptions achieve unambiguous reference will turn depend on ontological knowledge of so-called equality generating dependencies, for example, knowing that a person will have a unique social insurance number, or that a room, when non-empty, will have a unique combination of people occupying the room.

In this paper, we introduce a more expressive variety of such dependencies when an ontological understanding is expressed in terms of a description logic, in particular, in terms of an expressive dialect of the FunDL family of description logics [5]. For a better alignment with common data sources such as relational databases, all such logics are feature-based instead of role-based, that is, consider facts to be captured with partial functions instead of more general binary relationships. Such logics have recently included a concept constructor called a *path description dependency* (PDD) in which component path descriptions can be annotated to define progressively richer conditions for equality generation [6, 7, 8]. There are two possible annotations that have been considered. Both relate to the respective non-empty sets of entities reachable by a path description: a “set intersection” annotation that is satisfied when there is at least one such entity in common, and a “set equality” annotation that is satisfied when the respective non-empty sets of reachable entities are the same. In this paper, we introduce a new more expressive “structural equality” annotation for path descriptions in which equality only follows according to a structured alignment of non-empty sets of facts about an entity.

For example, consider where a document will have a style consisting of sets of sets of keywords, where each top level set is a group of keywords occurring in one of the document’s paragraphs. It will now be possible to identity document styles with exactly the same keywords that are also grouped in

DL 2025: 38th International Workshop on Description Logics, September 3–6, 2025, Opole, Poland
david@uwaterloo.ca (D. Toman); gweddell@uwaterloo.ca (G. Weddell)

exactly the same way. This is illustrated below in which three graphs define how keywords kw_1, kw_2 and kw_3 relate to three possible document styles ds_1, ds_2 and ds_3 via a path description of the form *kw-grp*$^-$.*kw-dom*$^-$.*kw-ran*:



Here, the path description consists of the three features *kw-grp*, *kw-dom* and *kw-ran*, and characterizes how the keywords can be "reached" from a document style by following a path of feature values: first the inverse of *kw-grp* to one of the document style's keyword groups, and then the inverse of *kw-dom* followed by *kw-ran* to the keywords in a group.

A non-empty set intersection annotation for this path would imply that all three graphs must describe the same document style while this would only hold for the left two graphs with the more fine-grained non-empty set equality annotation. But now, with our new structural equality annotation, the left two must also describe distinct document styles since the same keywords are now grouped by paragraphs in different ways.

2. Summary of Definitions and Results

We define a family of description logics *set-*$\mathcal{DLFDI}$ that are members of the FunDL dialects of description logic [5], We use standard symbols for and ways of interpreting *primitive features* and *concepts* as functions and sets of objects. The main novelty of the *set-*$\mathcal{DLFDI}$ family is allowing path descriptions Pd to participate in PDDs.

Definition 1 (Path Descriptions). *A* path description *is defined by the grammar*

$$\mathsf{Pd} ::= id \mid f.\,\mathsf{Pd} \mid f^-.\,\mathsf{Pd} \mid \mathrm{C}?.\,\mathsf{Pd},$$

for $f \in \mathsf{F}$*, where* f^- *is called the* inverse of f*,* C *a concept, and with the stipulation that substrings of the form* $f.f^-$ *and* $f^-.f$ *do not appear in any path description* Pd. □

In this paper we study the notion of *structural path description agreement* in PDDs.

Definition 2 (Structural Pd Agreement). *Let* Pd *be a path description,* $\mathcal{I}$ *an interpretation and* x *and* y *be two* $\triangle$ *elements. We say that* x *and* y structurally agree on Pd, $\mathsf{Pd}^{\simeq}(x, y)$*, when:*

$$\begin{array}{ll}
x = y & \text{if } \mathsf{Pd} = id, \\
\forall x_1, y_1.(x_1 = f^{\mathcal{I}}(x)) \wedge (y_1 = f^{\mathcal{I}}(y)) \rightarrow \mathsf{Pd}_1^{\simeq}(x_1, y_1) & \text{if } \mathsf{Pd} = f.\,\mathsf{Pd}_1, \\
\forall x_1.(f^{\mathcal{I}}(x_1) = x) \rightarrow \exists y_1.(f^{\mathcal{I}}(y_1) = y) \wedge \mathsf{Pd}_1^{\simeq}(x_1, y_1) & \\
\quad \wedge\, \forall y_1.(f^{\mathcal{I}}(y_1) = y) \rightarrow \exists x_1.(f^{\mathcal{I}}(x_1) = y) \wedge \mathsf{Pd}_1^{\simeq}(x_1, y_1) & \text{if } \mathsf{Pd} = f^-.\,\mathsf{Pd}_1, \\
x \in \mathrm{C}^{\mathcal{I}} \wedge y \in \mathrm{C}^{\mathcal{I}} \wedge \mathsf{Pd}_1^{\simeq}(x, y) & \text{if } \mathsf{Pd} = \mathrm{C}?.\,\mathsf{Pd}_1\,.
\end{array}$$

We introduce other notions of path equality (discussed in the introduction) in place of $\simeq$ in the definition of PDD below in Definition 4.

Definition 3 (Concepts, Subsumptions, and TBoxes). *A* $\{\simeq\}$*-*$\mathcal{DLFDI}$ *(a member of the set-*$\mathcal{DLFDI}$ *family)* concept description C *is constructed from primitive concepts using Boolean concept constructors* $\sqcap, \sqcup$*, and* $\neg$*, value restrictions on features* $\forall f.C$*, unqualified existential restrictions on features* $\exists f$ *and inverse features* $\exists f^-$*, and the* path description dependency (PDD) *of the form* $C : \mathsf{Pd}_1^{\simeq}, \ldots, \mathsf{Pd}_k^{\simeq} \rightarrow \mathsf{Pd}^{\simeq}$.

The semantics of all the derived concept descriptions C is defined in the standard way; for the PDD concept constructor the semantics is given by

$$(C : \mathsf{Pd}_1^{\simeq}, ..., \mathsf{Pd}_k^{\simeq} \to \mathsf{Pd}^{\simeq})^{\mathcal{I}} =$$
$$\{x \mid \forall y \in C^{\mathcal{I}} : \mathsf{Pd}^{\mathcal{I}}(\{x\}) \neq \emptyset \wedge \mathsf{Pd}^{\mathcal{I}}(\{y\}) \neq \emptyset \wedge (\bigwedge_{i=1}^{k} \mathsf{Pd}_i^{\simeq}(x, y)) \to \mathsf{Pd}^{\simeq}(x, y)\},$$

where, for a set $S \subseteq \triangle$, $\mathsf{Pd}^{\mathcal{I}}(S)$ is the set of $\triangle$ elements reachable from S in $\mathcal{I}$ via Pd*. A* subsumption *is an expression of the form $C_1 \sqsubseteq C_2$, where the C_i are concepts, and where PDDs occur only in C_2 but not within the scope of negation.[1] A* terminology (TBox) *$\mathcal{T}$ consists of a finite set of subsumptions, and a* posed question *$\mathcal{Q}$ is a single subsumption. The notions of satisfaction and entailment are standard.* □

The entailment in $\{\simeq\}$-$\mathcal{DLFDI}$ can be shown decidable via mapping to (unsatisfiability of) an Ackermann-prefix [11, 12] formula:

Theorem 1. *The entailment problem in $\{\simeq\}$-$\mathcal{DLFDI}$ is complete for EXPTIME.*

Alternative path-based Pd agreements, introduced in [7, 8], have been defined as follows:

Definition 4 (Alternative Path-Based PD Agreement(s))**.** *Let $\mathcal{I}$ be an interpretation and o_1 and o_2 be two $\triangle$ elements. We write $\mathsf{Pd}^{\cap}(o_1, o_2)$ to express $\mathsf{Pd}^{\mathcal{I}}(\{o_1\}) \cap \mathsf{Pd}^{\mathcal{I}}(\{o_2\}) \neq \emptyset$ (the set intersection agreement) and $\mathsf{Pd}^{\approx}(o_1, o_2)$ to express $\mathsf{Pd}^{\mathcal{I}}(\{o_1\}) = \mathsf{Pd}^{\mathcal{I}}(\{o_2\}) \neq \emptyset$ (the non-empty set agreement).* □

The following Theorem shows that *mixing* path agreement variants in PDDs/TBoxes leads to undecidability:

Theorem 2. *The entailment problems in $\{\simeq, \approx\}$-$\mathcal{DLFDI}$ and $\{\simeq, \cap\}$-$\mathcal{DLFDI}$ are undecidable.*

The members of the *set-$\mathcal{DLFDI}$* family are designed to serve as the underlying ontological languages that allow referring expressions [3] to be plural—a reference to an object now can be achieved by specifying a *set* of appropriately related objects (that have explicit identifiers).

Definition 5 (Referring Expression Types)**.** *A* referring expression type *(Rt) is defined by the following grammar, where* A *is a primitive concept.*

$$Rt ::= \{?\} \mid \mathrm{A} \to Rt \mid \exists f.Rt \mid \exists f^{-}.Rt \mid Rt_1 \sqcap Rt_2 \mid Rt_1 \,;\, Rt_2$$

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

$$\begin{aligned}
\mathcal{L}(\{?\}) &= \{\{a\} \mid a \text{ is a constant symbol}\}\\
\mathcal{L}(\mathrm{A} \to Rt) &= \{\mathrm{A} \sqcap C \mid C \in \mathcal{L}(Rt)\}\\
\mathcal{L}(\exists f.Rt) &= \{\exists f.C \mid C \in \mathcal{L}(Rt)\}\\
\mathcal{L}(\exists f^{-}.Rt) &= \{(\exists f^{-}.C[\vec{a}/\vec{b}_1]) \sqcap \ldots \sqcap (\exists f^{-}.C[\vec{a}/\vec{b}_k]) \mid C \in \mathcal{L}(Rt)\}\\
\mathcal{L}(Rt_1 \sqcap Rt_2) &= \{C_1 \sqcap C_2 \mid C_1 \in \mathcal{L}(Rt_1) \wedge C_2 \in \mathcal{L}(Rt_2)\}\\
\mathcal{L}(Rt_1; Rt_2) &= \mathcal{L}(Rt_1) \cup \mathcal{L}(Rt_2)
\end{aligned}$$

where $C[\vec{a}/\vec{b}]$ is the concept C in which all nominals $\vec{a}$ in C have been replaced by $\vec{b}$; this replacement is over all possible distinct choices of $\vec{b}_1, \ldots, \vec{b}_k$ for $\vec{b}$ and all $k \in N$. Given a TBox $\mathcal{T}$ and referring expression type Rt, the singularity problem for Rt with respect to $\mathcal{T}$ *is to determine if $|C^{\mathcal{I}}| \leq 1$ for every $C \in \mathcal{L}(Rt)$ and every model $\mathcal{I}$ of $\mathcal{T}$.* □

Example 1. *Each of the three graphs in our introductory example are* parse trees *for concepts occurring in $\mathcal{L}(Rt)$ when Rt is "$\exists$kw-grp^{-}.$\exists$kw-dom^{-}.$\exists$kw-ran.$\{?\}$". For example, the middle graph would be the concept*

$$(\exists \textit{kw-grp}^{-}.\exists \textit{kw-dom}^{-}.\exists \textit{kw-ran}.\{kw_1\}) \sqcap$$
$$(\exists \textit{kw-grp}^{-}.(\exists \textit{kw-dom}^{-}.\exists \textit{kw-ran}.\{kw_2\} \sqcap \exists \textit{kw-dom}^{-}.\exists \textit{kw-ran}.\{kw_3\})).$$

To formulate our result we need to normalize the referring expression types.

Definition 6 (Normalized Referring Expression Types)**.** *We use $\mathsf{Norm}(Rt)$ to refer to an exhaustive application of the following rewrite rules:*

$$\begin{aligned}
\mathrm{A} \to (Rt_1; Rt_2) &\mapsto \mathrm{A} \to Rt_1; \mathrm{A} \to Rt_2\\
Rt \sqcap (Rt_1; Rt_2) &\mapsto Rt \sqcap Rt_1; Rt \sqcap Rt_2\\
(Rt_1; Rt_2) \sqcap Rt &\mapsto Rt_1 \sqcap Rt\,; Rt_2 \sqcap Rt
\end{aligned}
\qquad
\begin{aligned}
\exists f.(Rt_1; Rt_2) &\mapsto \exists f.Rt_1; \exists f.Rt_2\\
\exists f^{-}.(Rt_1; Rt_2) &\mapsto \exists f^{-}.Rt_1; \exists f^{-}.Rt_2
\end{aligned}$$
□

[1] Violating this latter condition leads immediately to undecidability [9, 10].

The definition of Norm is an adaptation of referring expression type normalization in [3] with the following consequences: (1) $\mathcal{L}(Rt) = \mathcal{L}(\mathsf{Norm}(Rt))$, and (2) all *preference operators* (";") are at the top level of $\mathsf{Norm}(Rt)$. We call the maximal ";"-free parts of $\mathsf{Norm}(Rt)$ *preference-free components.* The following auxiliary function will be used to formulate subsumptions in *set-$\mathcal{DLFDI}$* to statically test for singularity of each preference free component.

$$
\begin{aligned}
\mathsf{Pds}(\{?\}) &= \{(id)^{\simeq}\} \\
\mathsf{Pds}(\mathrm{A} \to Rt) &= \{(\mathrm{A}?.\,\mathsf{Pd})^{\simeq} \mid (\mathsf{Pd})^{\simeq} \in \mathsf{Pds}(Rt)\} \\
\mathsf{Pds}(\exists f.Rt) &= \{(f.\,\mathsf{Pd}')^{\simeq} \mid (\mathsf{Pd}')^{\simeq} \in \mathsf{Pds}(Rt)\} \\
\mathsf{Pds}(\exists f^{-}.Rt) &= \{(f^{-}.\,\mathsf{Pd}')^{\simeq} \mid (\mathsf{Pd}')^{\simeq} \in \mathsf{Pds}(Rt)\} \\
\mathsf{Pds}(Rt_1 \sqcap Rt_2) &= \mathsf{Pds}(Rt_1) \cup \mathsf{Pds}(Rt_2)
\end{aligned}
$$

The function extracts a set of path descriptions adorned with "$\simeq$" leading to nominals from the preference-free referring expression type. The singularity test is now as follows:

Theorem 3. *Let $\mathcal{T}$ be a TBox in set-$\mathcal{DLFDI}$ and Rt a referring expression type. Then all referring concepts in $\mathcal{L}(Rt)$ are singular with respect to $\mathcal{T}$ if and only if $\mathcal{T} \models \top \sqsubseteq \top : \mathsf{Pds}(Rt') \to id$ holds for every preference-free component Rt' of $\mathsf{Norm}(Rt)$.* $\square$

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] D. Toman, G. E. Weddell, Structural Equality Generating Dependencies in Definite Descriptions, Technical Report CS-2025-05, Cheriton School of Computer Science, University of Waterloo, 2025.

[2] B. Russell, On denoting, Mind 14 (1905) 479–493. URL: http://www.jstor.org/stable/2248381.

[3] 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.

[4] A. Borgida, E. Franconi, D. Toman, G. E. Weddell, Understanding document data sources using ontologies with referring expressions, in: AI 2022: Advances in Artificial Intelligence, volume 13728 of *LNCS*, Springer, 2022, pp. 367–380.

[5] 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.

[6] E. Feng, A. Borgida, E. Franconi, P. F. Patel-Schneider, D. Toman, G. E. Weddell, Path Description Dependencies in Feature-Based DLs, in: Description Logics, volume 3515 of *CEUR Workshop Proceedings*, 2023.

[7] E. Feng, D. Toman, G. E. Weddell, On mixed semantics of path description dependencies in FunDL, in: Description Logics, volume in press of *CEUR Workshop Proceedings*, 2024.

[8] E. Feng, E. Franconi, P. F. Patel-Schneider, D. Toman, G. E. Weddell, Equality generating dependencies in description logics via path agreements, in: AI (2), volume 15443 of *Lecture Notes in Computer Science*, 2024, pp. 214–227.

[9] 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.

[10] D. Toman, G. E. Weddell, On Keys and Functional Dependencies as First-Class Citizens in Description Logics, J. Aut. Reasoning 40 (2008) 117–132.

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

[12] M. Fürer, Alternation and the Ackermann Case of the Decision Problem, L'Enseignement Math. 27 (1981) 137–162.