Decomposing Description Logic Ontologies

Boris Konev
Liverpool University
United Kingdom
konev@liverpool.ac.uk

Carsten Lutz
Universtiy of Bremen
Germany
clu@uni-bremen.de

Denis Ponomaryov
Institute of Informatics Systems
Novosibirsk, Russia
ponom@iis.nsk.su

Frank Wolter
Liverpool University
United Kingdom
wolter@liverpool.ac.uk

Abstract

Recent years have seen the advent of large and complex ontologies, most notably in the medical domain. As a consequence, structuring mechanisms for ontologies are nowadays viewed as an indispensible tool. A basic such mechanism is the automatic decomposition of the vocabulary of an ontology into independent parts. In this paper, we study decompositions that are syntax independent in the sense that the resulting partitioning depends only on the meaning of the vocabulary items, but not on the concrete syntactic form of the axioms in the ontology.

We present the first systematic investigation of decompositions of this type in the context of ontologies. Specifically, we focus on ontologies formulated in description logics and provide a variety of results that range from theorems stating the existence of unique finest decompositions to complexity results and algorithms computing decompositions. We also investigate the relationship between the existence of unique finite decompositions and a variant of the Craig interpolation property called parallel interpolation.

Introduction

The purpose of an ontology in knowledge representation is to fix the vocabulary of an application domain and to formally describe the meaning of this vocabulary using a logic-based language. This simple idea has proved to be rather successful, and consequently a considerable number of ontologies have been developed for various application domains. In broad domains such as medicine, ontologies used in practice can be extremely large; as an example, take the medical ontology SNOMED CT that covers almost half a million vocabulary items. Unsurprisingly, the design and maintenance of logical theories of this size poses serious challenges and it has long been a major goal of the KR community to provide support in the form of automated reasoning techniques.

Basic reasoning support for ontology design and maintenance aims to make explicit the structure of an ontology, for example by using classification (computing the subconcept/superconcept hierarchy). This is fundamental for an ontology designer who can easily lose track of the overall structure of an ontology—especially when it is constructed by multiple designers working in parallel as in the case of SNOMED CT. Making explicit the structure is also essential when an existing ontology has to be re-engineered due to changes in the modeled application domain or to customize it for a novel application—especially when the ontology was designed by somebody else.

In this paper, we consider a way of analyzing the structure of an ontology that aims at making explicit the *dependencies* among vocabulary items in the ontology. Our approach is based on *signature decompositions*, a partition of the signature of an ontology (i.e., of the symbols used to describe vocabulary items) into parts that are independent regarding their meaning. Similar kinds of structural analysis of an ontology have been advocated, e.g. in (d'Aquin et al. 2009). However, all existing approaches are *syntax-dependent* in the sense that two semantically equivalent, but syntactically different ontologies may yield different decompositions. Thus, the quality of the computed signature decomposition depends on the quality of the representation of the analyzed ontology (when the goal of the analysis may actually be to improve the quality of a poorly organized ontology).

Our aim is to establish the theoretical foundations for a purely semantic approach to signature decompositions that is not syntax-dependent in the above sense. Formally, the basic notion studied in this paper is the following: a partition $\Sigma_1, \dots, \Sigma_n$ of the signature of an ontology $\mathcal{T}$ formulated in an ontology language $\mathcal{L}$ is a *signature decomposition* of $\mathcal{T}$ in $\mathcal{L}$ if there are ontologies $\mathcal{T}_1, \dots, \mathcal{T}_n$ formulated in $\mathcal{L}$ such that (i) each $\mathcal{T}_i$ uses only symbols from Σ_i and (ii) the union $\mathcal{T}_1 \cup \cdots \cup \mathcal{T}_n$ is logically equivalent to $\mathcal{T}$. This notion has first been proposed by Parikh (1999) and Kourousias and Makinson (2007) in the context of propositional logic and belief revision. We emphasize that the ontologies $\mathcal{T}_1, \dots, \mathcal{T}_n$ used in the definition of signature decompositions need *not* be subsets of the original ontology $\mathcal{T}$. Moreover, as we are interested in decompositions of signatures, we only demand the existence of these ontologies, but do not insist they are explicitly computed. There is a close relationship between signature decompositions and approaches to modularization of ontologies that aim at a partition of the *axioms* (rather than signature) of an ontology into independent parts (Cuenca Grau et al. 2006; Amir and McIlraith 2005; Stuckenschmidt, Parent, and Spaccapietra 2009). Again, however, all existing approaches are syntax-dependent and aim at partitioning the existing axiomatization.

In many cases, the initial version of signature decompositions defined above can be expected to be too coarse to be informative. To see this, consider a description logic (DL) ontology $\mathcal{T}$ that consists of the axioms $\alpha = (\mathsf{Car} \sqsubseteq \exists \mathsf{has\_part}.\mathsf{Tire})$ and $\beta = (\mathsf{Ship} \sqsubseteq \exists \mathsf{has\_part}.\mathsf{Deck})$. It is not difficult to show that, due to the use of the role $\mathsf{has\_part}$, the only decomposition of $\mathcal{T}$ consists of only one set that contains the whole signature. From an ontology design perspective, though, the ontology $\mathcal{T}$ contains cars and ships as two separate subject areas that should not be 'merged' due to using the general-purpose role $\mathsf{has\_part}$ that, intuitively, does not belong to any specific subject area. From a logical viewpoint, $\mathsf{has\_part}$ behaves like a *logical* symbol much like the equality symbol or the symbol $\bot$ for contradiction. This example suggests to generalize the initial version of signature decompositon by adding a set of symbols Δ that do not induce dependencies and do not participate in the decomposition. Formally, a *signature Δ-decomposition* is defined just like a signature decomposition, except that each ontology $\mathcal{T}_i$ is allowed to use symbols from Σ_i *and* Δ. This generalization was first proposed by Ponomaryov (2008). In practice, it may not be easy to determine a suitable Δ. In fact, we do not expect signature decompositions to be a push-button technique, but rather envision an iterative and interactive process of understanding and improving the structure of an ontology, where the designer repeatedly chooses sets Δ and analyzes the impact on the resulting decomposition.

It is important to observe that the definition of a signature decomposition, both with and without the set Δ, depends on the language $\mathcal{L}$ used to formulate the ontologies $\mathcal{T}_1, \ldots, \mathcal{T}_n$ that realize the signature decomposition (henceforth called *realizations*). In principle, this is a point of concern as it may not be clear which language $\mathcal{L}$ is appropriate here; for example, when decomposing an ontology $\mathcal{T}$ given in a DL, one might expect more fine-grained decompositions if $\mathcal{L}$ is second-order logic (SO) compared to when $\mathcal{L}$ is again a DL. Therefore, the *first aim of this paper is to study in how far decompositions of DL ontologies depend on the language for the realizations*. Fortunately, it turns out that for many standard DLs, decompositions of TBoxes do not depend on whether one uses SO or the DL for realizations. The main tool for proving this and related results is establishing the *parallel interpolation property*, a type of interpolation that has not yet been investigated in the context of ontologies.

In general, one may expect that there can be many distinct and incomparable signature decompositions of a given ontology $\mathcal{T}$. This is another point of concern because facing a large number of incomparable decompositions is likely to be confusing rather than helpful for an ontology designer. Therefore and since finer decompositions are clearly more informative than coarser ones, one would ideally like to have a *unique* finest decomposition to work with. *Thus, the second aim of this paper is to investigate when unique finest decompositions exist*. Fortunately, we can use parallel interpolation to show that this is the case for many standard DLs.

Syntax	FO	$\mathcal{EL}$	$\mathcal{ALC}$	Short
$\top$	$x = x$	✓	✓	
$\bot$	$\neg(x = x)$		✓	
A	$A(x)$	✓	✓	
$\neg C$	$\neg C(x)$		✓	
$C \sqcap D$	$C(x) \wedge D(x)$	✓	✓	
$\exists r.C$	$\exists y\,(r(x,y) \wedge C(y))$	✓	✓	
$(\leqslant n\ r\ C)$	$\exists^{\geq n} y\,(r(x,y) \wedge C(y))$			$\mathcal{Q}$
$\{a\}$	$x = a$			$\mathcal{O}$
r^-	$r(y,x)$			$\mathcal{I}$
$C \sqsubseteq D$	$\forall x\,(C(x) \rightarrow D(x))$	✓	✓	
$r \sqsubseteq s$	$\forall xy\,(r(x,y) \rightarrow s(x,y))$			$\mathcal{H}$

Figure 1: Standard translation

Finally, we provide *a first analysis of the complexity of some computational problems related to signature decompositions in DL ontologies*. We show that for many expressive DLs, these problems are not harder than standard reasoning. Given that there is a very close connection between signature decompositions on the one hand, and computationally very expensive notions such as conservative extensions and uniform interpolation on the other hand, this result is rather surprising. We also show that in the lightweight description logic DL-Lite, signature decompositions can typically be computed in polynomial time. For the lightweight DL $\mathcal{EL}$, we establish the same result for some restricted, but natural cases.

Many proofs are omitted and can be found in the full version of this paper (Konev et al. 2010).

Preliminaries

Let $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$ be countably infinite and mutually disjoint sets of concept names (unary predicates), role names (binary predicates), and individual names. We use $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$ as the vocabulary for second-order logic (SO), first-order logic (FO), and a variety of DLs. More precisely, we consider SO (and FO) with equality, the predicates from $\mathsf{N_C} \cup \mathsf{N_R}$ and constants from $\mathsf{N_I}$.[1] Matching this vocabulary, second-order quantification is over set variables and binary relation variables. We use $\mathcal{T} \subseteq \mathrm{SO}$ and $\mathcal{T} \subseteq_{fin} \mathrm{SO}$ to denote that $\mathcal{T}$ is a set, respectively finite set, of SO-sentences; we write $\mathcal{T} \models \varphi$ if φ is an SO-sentence that is a consequence of $\mathcal{T}$. A set $\mathcal{T} \subseteq \mathrm{SO}$ is *satisfiable* iff $\mathcal{T}$ has a model. Two sets $\mathcal{T}_1 \subseteq \mathrm{SO}$ and $\mathcal{T}_2 \subseteq \mathrm{SO}$ are *equivalent*, in symbols $\mathcal{T}_1 \equiv \mathcal{T}_2$, if they have the same models or, equivalently, if $\mathcal{T}_1 \models \varphi$ for all $\varphi \in \mathcal{T}_2$ and vice versa. We sometimes write $\mathcal{T}_1 \models \mathcal{T}_2$ as shorthand for '$\mathcal{T}_1 \models \varphi$ for all $\varphi \in \mathcal{T}_2$'. The *signature* $\mathsf{sig}(\varphi)$ of an SO-formula is the set of all predicate and constant symbols (except equality) used in φ. This notion is lifted to sets of sentences in the obvious way. A *fragment* of second-order logic is simply a subset $\mathcal{L} \subseteq \mathrm{SO}$.

Description logics can be viewed as fragments of FO. DL *concepts* are formed starting from concept names by induc-

[1]This is only for uniformity with DLs. The results presented in this paper do not depend on the restricted arity.

tively applying concept constructors such as those shown in the upper part of Figure 1. The choice of different constructors gives rise to different DLs. In the figure, we have marked the constructors of the basic DLs $\mathcal{EL}$ and $\mathcal{ALC}$ and assigned to each additional constructor a letter that allows the systematic appellation of extended DLs. The extension $\mathcal{I}$ is with a role constructor for inverse roles, not a concept constructor. When $\mathcal{I}$ is present, inverse roles can be used inside existential restrictions, number restrictions $\mathcal{Q}$ and role hierarchies $\mathcal{H}$. For details, we refer the reader to (Baader et al. 2003).

To simplify notation, we identify models of SO (and, therefore, of FO and DLs) with interpretations $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consisting of a non-empty domain $\Delta^\mathcal{I}$ and a function $\cdot^\mathcal{I}$ that assigns a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ to each $A \in \mathsf{N_C}$, a relation $r^\mathcal{I}$ over $\Delta^\mathcal{I}$ to each $r \in \mathsf{N_C}$, and an element $a^\mathcal{I} \in \Delta^\mathcal{I}$ to each $a \in \mathsf{N_I}$. The extension $C^\mathcal{I} \subseteq \Delta^\mathcal{I}$ of a DL concept C is defined by the standard inductive translation of C into an FO-formula with one free variable x as shown in Figure 1.

A TBox (or *ontology*) is a finite set of concept inclusions (CIs) $C \sqsubseteq D$, where C, D are concepts. An interpretation *satisfies* a CI $C \sqsubseteq D$ (written $\mathcal{I} \models C \sqsubseteq D$) iff $C^\mathcal{I} \subseteq D^\mathcal{I}$ and a TBox $\mathcal{T}$ (written $\mathcal{I} \models \mathcal{T}$) if $\mathcal{I} \models C \sqsubseteq D$ for all $C \sqsubseteq D \in \mathcal{T}$. In the presence of role hierarchies (indicated by the letter $\mathcal{H}$), TBoxes can also include role inclusions $r \sqsubseteq s$ whose semantics can be found in Figure 1. We will typically not distinguish between DL concepts (resp. TBoxes) and their FO translations. In particular, we often regard DL TBoxes as finite sets of FO-sentences (and thus SO-sentences).

Signature Decomposition

We introduce and illustrate the basic notion of this paper and identify some of its fundamental properties.

Definition 1 (Signature Decomposition) *Let $\mathcal{T} \subseteq_{fin}$ SO, $\Delta \subseteq \mathsf{sig}(\mathcal{T})$ and $\mathcal{L}$ a fragment of SO. A partition $\Sigma_1, \ldots, \Sigma_n$ of $\mathsf{sig}(\mathcal{T}) \setminus \Delta$ is called a* signature Δ-decomposition of $\mathcal{T}$ in $\mathcal{L}$ *if there are $\mathcal{T}_1, \ldots, \mathcal{T}_n \subseteq \mathcal{L}$ such that*

- $\mathsf{sig}(\mathcal{T}_i) \subseteq \Sigma_i \cup \Delta$ *for* $1 \leq i \leq n$;
- $\mathcal{T}_1 \cup \cdots \cup \mathcal{T}_n \equiv \mathcal{T}$.

In this case, we say that $\mathcal{T}_1, \ldots, \mathcal{T}_n$ realize *the signature Δ-decomposition $\Sigma_1, \ldots, \Sigma_n$ in $\mathcal{L}$.*

For simplicity, we will often speak only of Δ-decompositions instead of signature Δ-decompositons. When $\Delta = \emptyset$, we simply drop it and speak only of (signature) decompositions. Note that in contrast to Kourousias and Makinson (2007), we consider only finitely axiomatized theories, which suffices for our purposes. Some proofs actually depend on this assumption.

For any $\mathcal{T}$ and Δ, there exists at least one Δ-decomposition, namely the *trivial decomposition* consisting only of the single set $\mathsf{sig}(\mathcal{T}) \setminus \Delta$. We call a partition $\Sigma_1, \ldots, \Sigma_n$ *finer* than a partition $\Pi_1, \ldots, \Pi_m$ if they are distinct and for every $i \leq m$ there exist $i_1, \ldots, i_k \leq n$ such that $\Pi_i = \bigcup_{\ell \leq k} \Sigma_{i_\ell}$.

Example 2 Let $\mathcal{T}$ be the TBox consisting of $\alpha_1 = (\mathsf{Ball} \sqsubseteq \mathsf{Physical\_Object})$, $\alpha_2 = (\mathsf{Table} \sqsubseteq \mathsf{Physical\_Object})$, $\alpha_3 = (\mathsf{Ball} \sqsubseteq \exists\mathsf{has\_colour}.\top)$, $\alpha_4 = (\mathsf{Table} \sqsubseteq \exists\mathsf{has\_colour}.\top)$, $\alpha_5 = (\mathsf{OrangeBall} \sqsubseteq \mathsf{Ball})$.

For any of $\Delta = \emptyset$, $\Delta = \{\mathsf{Physical\_object}\}$ and $\Delta = \{\mathsf{has\_colour}\}$, there are no non-trivial Δ-decompositions of $\mathcal{T}$ because, intuitively, Ball and Table are connected independently via both Phyical_object and has_colour. In many contexts, one would not regard this as a relevant dependency between the two terms. In fact, for $\Delta = \{\mathsf{Physical\_object}, \mathsf{has\_colour}\}$ the finest Δ-decomposition of $\mathcal{T}$ is $\{\mathsf{Ball}, \mathsf{OrangeBall}\}, \{\mathsf{Table}\}$, realized by $\{\alpha_1, \alpha_3, \alpha_5\}$ and $\{\alpha_2, \alpha_4\}$.

One way to extend $\mathcal{T}$ such that Ball and Table are separated already when choosing $\Delta = \{\mathsf{has\_colour}\}$ is to add $\alpha_6 = (\exists\mathsf{has\_colour}.\top \sqsubseteq \mathsf{Physical\_Object})$. In the resulting $\mathcal{T}'$, the axioms α_1, α_2 become redundant and the finest Δ-decomposition is $\{\mathsf{Physical\_Object}\}, \{\mathsf{Ball}, \mathsf{OrangeBall}\}, \{\mathsf{Table}\}$, realized by $\{\alpha_6\}$, $\{\alpha_3, \alpha_5\}$, $\{\alpha_4\}$.

Finally, note that OrangeBall and Ball cannot be separated in a non-trivial way because one would have to extend Δ by at least one of the two concepts.

Signature decompositions that can be obtained by analyzing the syntactic form of axioms are a special case of signature decompositions in the sense of Definition 1. The following example shows how such syntactic decompositions can be computed.

Example 3 (Syntactic decomposition) Let $\mathcal{T} \subseteq_{fin}$ SO and $\Delta \subseteq \mathsf{sig}(\mathcal{T})$. There always exists a (unique) finest Δ-decomposition $\Sigma_1, \ldots, \Sigma_n$ that is *realized by subsets $\mathcal{T}_1, \ldots, \mathcal{T}_n$ of $\mathcal{T}$*. We denote this Δ-decomposition by $\mathsf{sdeco}_\Delta(\mathcal{T})$ and call it the *syntactic Δ-decomposition of $\mathcal{T}$*. $\mathsf{sdeco}_\Delta(\mathcal{T})$ can be obtained as the partition of $\mathsf{sig}(\mathcal{T}) \setminus \Delta$ induced by the smallest equivalence relation on $\mathsf{sig}(\mathcal{T}) \setminus \Delta$ that contains all pairs (σ_1, σ_2) for which there exists $\alpha \in \mathcal{T}$ with $\{\sigma_1, \sigma_2\} \subseteq \mathsf{sig}(\alpha) \setminus \Delta$. In general, $\mathsf{sdeco}_\Delta(\mathcal{T})$ is of course not the finest decomposition possible. Note that $\mathsf{sdeco}_\Delta(\mathcal{T})$ can be computed in poly-time.

We now establish some basic properties of *decompositons in SO*, i.e., decompositions of ontologies based on realizations $\mathcal{T}_1, \ldots, \mathcal{T}_n$ that are formulated in SO. As announced in the introduction, decompositions in SO play a special role in this paper as they are easy to work with and turn out to be equivalent to decompositions in many standard DLs. To formulate SO decompositions succinctly, we write $\exists\sigma.\varphi$ to denote $\exists P.\varphi[P/\sigma]$, where either σ is a predicate and P a fresh predicate variable of the same arity as σ, or σ is an individual constant and P a fresh individual variable. Clearly, $\mathsf{sig}(\exists\sigma.\varphi) = \mathsf{sig}(\varphi) \setminus \{\sigma\}$. $\exists\Sigma.\varphi$ is shorthand for $\exists\sigma_1 \cdots \exists\sigma_n.\varphi$ if $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$.

Theorem 4 (Characterization) *Let $\mathcal{T} \subseteq_{fin}$ SO and $\Delta \subseteq \mathsf{sig}(\mathcal{T})$. A partition $\Sigma_1, \ldots, \Sigma_n$ of $\mathsf{sig}(\mathcal{T}) \setminus \Delta$ is a signature Δ-decomposition of $\mathcal{T}$ in SO iff*

$$\{\exists\overline{\overline{\Sigma_1}}. \bigwedge_{\varphi \in \mathcal{T}} \varphi, \cdots, \exists\overline{\overline{\Sigma_n}}. \bigwedge_{\varphi \in \mathcal{T}} \varphi\} \models \mathcal{T} \qquad (*)$$

where $\overline{\overline{\Sigma_i}} := \bigcup_{1 \leq j \leq n, j \neq i} \Sigma_j$.

Proof. "⇒". Assume that the partition $\Sigma_1, \ldots, \Sigma_n$ of $\mathsf{sig}(\mathcal{T}) \setminus \Delta$ is a Δ-decomposition of $\mathcal{T}$ in SO realized by $\mathcal{T}_1, \ldots, \mathcal{T}_n$. To show that $(*)$ holds, let $\mathcal{I}$ be a model of the left-hand side of $(*)$. Then $\mathcal{I}$ is a model of $\mathcal{T}_i$ for $1 \leq i \leq n$: since $\mathcal{I} \models \exists\overline{\overline{\Sigma_i}}.\bigwedge_{\varphi\in\mathcal{T}}\varphi$, there is a model $\mathcal{J}$ of $\mathcal{T}$ that agrees with $\mathcal{I}$ on the interpretation of all symbols from $\Sigma_i \cup \Delta$; since $\mathcal{T} \models \mathcal{T}_1 \cup \cdots \cup \mathcal{T}_n$, we have $\mathcal{J} \models \mathcal{T}_i$ and due to $\mathsf{sig}(\mathcal{T}_i) \subseteq \Sigma_i \cup \Delta$, it follows that $\mathcal{I} \models \mathcal{T}_i$ as stated. Thus $\mathcal{I} \models \mathcal{T}_1 \cup \cdots \cup \mathcal{T}_n$ and $\mathcal{T}_1 \cup \cdots \cup \mathcal{T}_n \models \mathcal{T}$ yields that $\mathcal{I}$ is a model of $\mathcal{T}$, as required.

"⇐" If $(*)$ holds, then $\mathcal{T}_i = \{\exists\overline{\overline{\Sigma_i}}.\bigwedge_{\varphi\in\mathcal{T}}\varphi\}$, $1 \leq i \leq n$, clearly realize $\Sigma_1, \ldots, \Sigma_n$. ❑

As a consequence of the proof of Theorem 4, for each decomposition $\Sigma_1, \ldots, \Sigma_n$ in SO, there exists a realization of the canonical (though rather uninformative) form $\mathcal{T}_i = \{\exists\overline{\overline{\Sigma_i}}.\bigwedge_{\varphi\in\mathcal{T}}\varphi\}$, $1 \leq i \leq n$. Clearly, this canonical form relies on second-order quantifiers and does not exist in (fragments of) FO. As a first application of Theorem 4, one can show that there always exists a unique finest Δ-decomposition in SO.

Theorem 5 (Unique Finest Decomposition) *Let $\mathcal{T} \subseteq_{fin}$ SO, $\Delta \subseteq \mathsf{sig}(\mathcal{T})$, and let $\Sigma_1, \ldots, \Sigma_n$ and $\Pi_1, \ldots, \Pi_m$ be Δ-decompositions of $\mathcal{T}$ in SO. Then the partition $\Sigma_i \cap \Pi_j$ for all i, j with $\Sigma_i \cap \Pi_j \neq \emptyset$ of $\mathsf{sig}(\mathcal{T}) \setminus \Delta$ is a Δ-decomposition of $\mathcal{T}$ in SO. Thus, there exists a unique finest Δ-decomposition of $\mathcal{T}$ in SO.*

In the following example, we compute the finest Δ-decomposition in SO of concept hierarchies.

Example 6 Let $\mathcal{T}$ be a concept hierarchy, i.e., a finite set of inclusions $A \sqsubseteq B$ between concept names A, B. A realization of the unique finest Δ-decomposition in SO of $\mathcal{T}$ is obtained by first adding to $\mathcal{T}$ all CIs $A \sqsubseteq B$ with $\mathcal{T} \models A \sqsubseteq B$ that contain at most one non-Δ symbol. Then remove from the resulting TBox all $A \sqsubseteq B$ with two non-Δ-symbols for which there exists $D \in \Delta$ with $A \sqsubseteq D, D \sqsubseteq B \in \mathcal{T}$, and denote by $\mathcal{T}'$ the resulting TBox. It can be shown that $\mathsf{sdeco}_\Delta(\mathcal{T}')$ is the unique finest Δ-decomposition of $\mathcal{T}$ in SO which, in this case, is realized using again a concept hierarchy and no second-order quantifiers.

Although there is always a unique finest decomposition in SO, the theories that realize this (finest) decomposition are generally not uniquely determined. To see this, consider the TBox $\mathcal{T} = \{\top \sqsubseteq A \sqcap B_1 \sqcap B_2\}$ and let $\Delta = \{A\}$. Then the unique finest Δ-decomposition $\{B_1\}, \{B_2\}$ is realized by both $\{\top \sqsubseteq A \sqcap B_1\}, \{\top \sqsubseteq B_2\}$ and $\{\top \sqsubseteq B_1\}, \{\top \sqsubseteq A \sqcap B_2\}$. Clearly, there are no two sets in these two realizations that are logically equivalent.

We now present a condition under which realizations are unique, for many fragments of SO. Say that $\mathcal{T}_1, \mathcal{T}_2 \subseteq \mathcal{L}$ are Δ-*inseparable w.r.t.* $\mathcal{L}$ if, and only if, $\mathcal{T}_1 \models \varphi$ iff $\mathcal{T}_2 \models \varphi$ for all φ in $\mathcal{L}$ such that $\mathsf{sig}(\varphi) \subseteq \Delta$. Clearly, if Δ contains the signatures $\mathsf{sig}(\mathcal{T}_1)$ and $\mathsf{sig}(\mathcal{T}_2)$, then $\mathcal{T}_1$ and $\mathcal{T}_2$ are Δ-inseparable w.r.t. $\mathcal{L}$ iff they are logically equivalent. Otherwise, Δ-inseparability is weaker than logical equivalence and is an extension of the notion of a *conservative extension* (for which, in addition to being Δ-inseparable, it is required that $\mathcal{T}_1 \subseteq \mathcal{T}_2$ and $\Delta = \mathsf{sig}(\mathcal{T}_1)$) that has been used to develop a formal framework for modular ontologies and module extraction (Konev et al. 2009; Lutz and Wolter 2010). Note that for the canonical realization $\mathcal{T}_i = \{\exists\overline{\overline{\Sigma_i}}.\bigwedge_{\varphi\in\mathcal{T}}\varphi\}$, $1 \leq i \leq n$, of Theorem 4 we have that $\mathcal{T}_i, \mathcal{T}_j$ are Δ-inseparable w.r.t. SO for all $1 \leq i, j \leq n$.

Definition 7 (Unique Decomposition Realizations) *Let $\mathcal{L}$ be a fragment of SO. We say that $\mathcal{L}$* has unique decomposition realizations *(UDR) if for all satisfiable $\mathcal{T} \subseteq_{fin} \mathcal{L}$ and all finite $\mathcal{L}$-realizations $\mathcal{T}_1, \ldots, \mathcal{T}_n$ and $\mathcal{T}'_1, \ldots, \mathcal{T}'_n$ of a Δ-decomposition of $\mathcal{T}$ such that*

- *$\mathcal{T}_i, \mathcal{T}_j$ are Δ-inseparable w.r.t. $\mathcal{L}$ for $i, j \leq n$ and*
- *$\mathcal{T}'_i, \mathcal{T}'_j$ are Δ-inseparable w.r.t. $\mathcal{L}$ for $i, j \leq n$,*

we have $\mathcal{T}_i \equiv \mathcal{T}'_i$ for all $i \leq n$.

UDR has interesting consequences. For example, if $\mathcal{T}_1, \ldots, \mathcal{T}_n$ satisfy the conditions of Definition 7 and $\mathcal{L}$ has UDR, then one can show that $\mathcal{T}$ is a conservative extension of each $\mathcal{T}_i$ (i.e., $\mathcal{T}_i \models \varphi$ iff $\mathcal{T} \models \varphi$ for all φ with $\mathsf{sig}(\varphi) \subseteq \mathsf{sig}(\mathcal{T}_i)$). Thus, realizations satisfy the basic conditions for logic-based ontology modules as proposed and discussed in (Cuenca Grau et al. 2006; 2008; Konev et al. 2009).

Theorem 8 *SO has UDR.*

One can show that the canonical realization provided by Theorem 4 satisfies the conditions of Definition 7 for SO. Therefore, by Theorem 8, all realizations of a given Δ-decomposition that satisfy the conditions of Definition 7 for SO are equivalent to its canonical realization.

In this section, we have seen that decompositions in SO have a variety of desirable properties. The aim of the next section is to investigate in how far these are also enjoyed by decompositions in DLs.

Signature decompositions and parallel interpolation in DLs

By definition, if $\mathcal{L}_1$ is a fragment of $\mathcal{L}_2$, then every Δ-decomposition of some $\mathcal{T}$ in $\mathcal{L}_1$ is a Δ-decomposition of $\mathcal{T}$ in $\mathcal{L}_2$. In particular, every Δ-decomposition of $\mathcal{T}$ in some fragment of SO is a Δ-decomposition of $\mathcal{T}$ in SO. In this section, we show that for many DLs the converse implication holds as well and that, therefore, DLs inherit many of the desirable properties of decompositions in SO.

Definition 9 ($\mathcal{L}$-decompositions = SO-decompositions) *Let $\mathcal{L}$ be a fragment of SO. We say that* $\mathcal{L}$-decompositions coincide with SO-decompositions *if for every $\mathcal{T} \subseteq_{fin} \mathcal{L}$ and every signature $\Delta \subseteq \mathsf{sig}(\mathcal{T})$, the Δ-decompositions of $\mathcal{T}$ in $\mathcal{L}$ coincide with the Δ-decompositions of $\mathcal{T}$ in SO.*

Before we provide methodologies for proving this property for a wide range of DLs, we provide a counterexample showing that $\mathcal{ALCO}$-decompositions do not coincide with SO-decompositions.

Example 10 Let $\Delta = \emptyset$ and $\mathcal{T}$ consist of the $\mathcal{ALCO}$-inclusions

$$\{a\} \sqsubseteq (\exists r.\neg\{a\}) \sqcap (\forall r.\neg\{a\}), \quad \top \sqsubseteq \{b\} \sqcup \{b'\},$$

$$\neg\{a\} \sqsubseteq (\exists r.\{a\}) \sqcap (\forall r.\{a\}), \quad \{a'\} \sqsubseteq \{a'\}.$$

By the CI $\top \sqsubseteq \{b\} \sqcup \{b'\}$, each model of $\mathcal{T}$ has at most two domain elements. Using the two CIs involving a it is, therefore, easy to see that $\mathcal{T}$ axiomatizes the class of two-element interpretations in which b, b' denote distinct elements and r is a symmetric and irreflexive relation that connects the two domain elements. In particular, $\mathcal{T}$ "says nothing" about a and a'. Thus, the finest Δ-decomposition in SO (and FO) of $\mathcal{T}$ is $\{a\}, \{a'\}, \{r\}, \{b, b'\}$. In contrast, one can show that there is no finer Δ-decomposition of $\mathcal{T}$ in $\mathcal{ALCO}$ than $\mathsf{sdeco}_\Delta(\mathcal{T})$ which coincides with $\{a, r\}, \{a'\}, \{b, b'\}$. Another Δ-decomposition of $\mathcal{T}$ in $\mathcal{ALCO}$, which is incompatible with $\mathsf{sdeco}_\Delta(\mathcal{T})$, is given by $\{a', r\}, \{a\}, \{b, b'\}$. It follows that $\mathcal{ALCO}$ TBoxes do not always have a unique finest Δ-decomposition in $\mathcal{ALCO}$.

We now introduce an interpolation property that is not only sufficient to prove that SO-decompositions coincide with $\mathcal{L}$-decompositions, but also implies UDR.

Definition 11 (Parallel Interpolation) *Let $\mathcal{L}$ be a fragment of SO, $(\mathcal{T}_1, \mathcal{T}_2)$ be two sets of SO-sentences, α an SO-sentence with $\mathcal{T}_1 \cup \mathcal{T}_2 \models \alpha$, and Δ a signature. A pair $(\mathcal{T}_1', \mathcal{T}_2')$ with $\mathcal{T}_i' \subseteq \mathcal{L}$ for $i = 1, 2$ is called a* Δ-parallel interpolant *of $(\mathcal{T}_1, \mathcal{T}_2)$ and α in $\mathcal{L}$ if the following conditions hold:*

- $\mathcal{T}_i \models \mathcal{T}_i'$ *for* $i = 1, 2$;
- $\mathsf{sig}(\mathcal{T}_i') \setminus \Delta \subseteq \mathsf{sig}(\mathcal{T}_i) \cap \mathsf{sig}(\alpha)$ *for* $i = 1, 2$;
- $\mathcal{T}_1' \cup \mathcal{T}_2' \models \alpha$.

$\mathcal{L}$ has the parallel interpolation property (PIP) if for all $\mathcal{T}_1, \mathcal{T}_2 \subseteq \mathcal{L}$, all $\alpha \in \mathcal{L}$, and all signatures Δ such that

1. $\mathsf{sig}(\mathcal{T}_1) \cap \mathsf{sig}(\mathcal{T}_2) \subseteq \Delta$,
2. $\mathcal{T}_1 \cup \mathcal{T}_2 \models \alpha$,
3. *$\mathcal{T}_1$ and $\mathcal{T}_2$ are Δ-inseparable w.r.t. $\mathcal{L}$,*

there exists a Δ-parallel interpolant of $(\mathcal{T}_1, \mathcal{T}_2)$ and α in $\mathcal{L}$.

The main reason for studying parallel interpolation is the following result.

Theorem 12 *Let $\mathcal{L}$ be a fragment of SO with the PIP. Then*

1. *$\mathcal{L}$-decompositions coincide with SO-decompositions.*
2. *$\mathcal{L}$ has UDR.*

In particular, every $\mathcal{T} \subseteq_{fin} \mathcal{L}$ has a unique finest Δ-decomposition in $\mathcal{L}$.

Proof. (Sketch for Point 1) Assume that Σ_1, Σ_2 is a Δ-decomposition in SO of $\mathcal{T}$. It follows from Theorem 4 that $\{\exists \Sigma_2. \bigwedge_{\varphi \in \mathcal{T}} \varphi, \exists \Sigma_1. \bigwedge_{\varphi \in \mathcal{T}} \varphi\} \models \mathcal{T}$. Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be the subsets of $\mathcal{L}$ obtained from $\mathcal{T}$ by replacing all predicates in Σ_2 and Σ_1, respectively, by fresh predicates. Then $\mathcal{S}_1 \cup \mathcal{S}_2 \models \mathcal{T}$ and the componentwise union of the Δ-parallel interpolants of $(\mathcal{S}_1, \mathcal{S}_2)$ and α in $\mathcal{L}$, $\alpha \in \mathcal{T}$, realizes Σ_1, Σ_2 in $\mathcal{L}$. ❑

The proof shows that an algorithm computing Δ-parallel interpolants in $\mathcal{L}$ can be directly employed to construct a realization in $\mathcal{L}$ of a given Δ-decomposition. As the focus of this paper is on signature decompositions rather than realizations, we concentrate on proving the PIP and leave the computation of Δ-parallel interpolants for future work.

In FO, it is easy to prove the equivalence of the PIP and the standard Craig interpolation property (Parikh 1999; Kourousias and Makinson 2007). Unfortunately, this is not the case for DLs because the proof uses the fact that FO-sentences are closed under Boolean operations and this typically does not hold for DLs (e.g., there does not exist a TBox $\mathcal{T}$ in $\mathcal{ALC}$ that is equivalent to $\neg(\top \sqsubseteq A)$). This also implies that recent results on the existence and computation of Craig interpolants in DL using tableaux are not directly applicable (Seylan, Franconi, and de Bruijn 2009). Nevertheless, it turns our that many DLs have the PIP:

Theorem 13 *The following DLs have the PIP: $\mathcal{EL}$, $\mathcal{ELH}$, $\mathcal{ALC}$, $\mathcal{ALCI}$, $\mathcal{ALCQ}$, $\mathcal{ALCQI}$.*

With the exception of $\mathcal{ELH}$, for which a proof is given in the full version of this paper, Theorem 13 is proved by employing known results regarding the interpolation and Robinson joint consistency properties of DLs. Namely, let $\mathcal{L}$ be a set of sentences in FO. We say that $\mathcal{L}$ has the *Robinson Joint Consistency Property (RJCP)* if the following holds for all $\mathcal{T}_1, \mathcal{T}_2 \subseteq \mathcal{L}$ and all signatures Δ: if $\mathsf{sig}(\mathcal{T}_1) \cap \mathsf{sig}(\mathcal{T}_2) \subseteq \Delta$ and $\mathcal{T}_1$ and $\mathcal{T}_2$ are Δ-inseparable w.r.t. $\mathcal{L}$, then

$$\mathcal{T}_1 \cup \mathcal{T}_2 \models \alpha \quad \Leftrightarrow \quad \mathcal{T}_1 \models \alpha$$

for all sentences α in $\mathcal{L}$ with $\mathsf{sig}(\alpha) \subseteq \mathsf{sig}(\mathcal{T}_1)$. We say that $\mathcal{L}$ has the *Boolean Craig Interpolation Property (BCIP)* if for all $\mathcal{T} \subseteq \mathcal{L}$ and all Boolean combinations φ of $\mathcal{L}$-sentences the following holds: if $\mathcal{T} \models \varphi$, then there exists a Boolean combination ψ of $\mathcal{L}$-sentences with $\mathsf{sig}(\psi) \subseteq \mathsf{sig}(\mathcal{T}) \cap \mathsf{sig}(\varphi)$ such that $\mathcal{T} \models \psi$ and $\psi \models \varphi$. Finally, we say that $\mathcal{L}$ has the *disjoint union property* if the following holds for all $\mathcal{T} \subseteq \mathcal{L}$: for all families $\mathcal{I}_i$, $i \in I$, of interpretations the following conditions are equivalent:

- all $\mathcal{I}_i$, $i \in I$, are models of $\mathcal{T}$;
- the disjoint union of all $\mathcal{I}_i$, $i \in I$, is a model of $\mathcal{T}$.

Note that $\mathcal{EL}$, $\mathcal{ELH}$, $\mathcal{ALC}$, $\mathcal{ALCQI}$ and all standard dialects of DL-Lite have the disjoint union property. Examples of DLs without the disjoint union property are DLs with nominals or the universal role. Now one can prove the following equivalences.

Theorem 14 *Let $\mathcal{L}$ be a fragment of FO with the disjoint union property. Then the following conditions are equivalent:*

- $\mathcal{L}$ *has the PIP;*
- $\mathcal{L}$ *has RJCP;*
- $\mathcal{L}$ *has the BCIP.*

We come to the proof of Theorem 13: the PIP of $\mathcal{EL}$ follows from Theorem 14 and its RJCP proved in (Lutz and Wolter 2010). The PIP of $\mathcal{ALC}$, $\mathcal{ALCQ}$, $\mathcal{ALCI}$, and $\mathcal{ALCQI}$ follows from Theorem 14 and their BCIP proved in (Konev et al. 2009). It remains to apply Theorem 14.

It is interesting to observe that the addition of role inclusions to $\mathcal{EL}$ preserves the PIP. This is true for DL-Lite (see the analysis below) as well, but expressive DLs with role inclusions typically do not have the PIP:

Example 15 $\mathcal{ALCH}$ does not have the PIP. Let $\Delta = \{r_1, r_2\}$, $\alpha = \forall r_1.A \sqsubseteq \exists r_2.A$, $\mathcal{T}_1 = \{\top \sqsubseteq \exists r_1.\top \sqcap \exists r_2.\top\}$, and $\mathcal{T}_2 = \{s \sqsubseteq r_1, s \sqsubseteq r_2, \top \sqsubseteq \exists s.\top\}$. Then $\mathcal{T}_1 \cup \mathcal{T}_2 \models \alpha$ but there does not exist a Δ-parallel interpolant of $(\mathcal{T}_1, \mathcal{T}_2)$ and α in $\mathcal{ALCH}$. We note that it remains an open problem whether $\mathcal{ALCH}$-decompositions coincide with SO-decomposition.

We now show how the PIP can be restored for expressive DLs with role inclusions and/or nominals by including into Δ all role and individual names. To obtain the PIP in the presence of nominals we take, in addition, the @-operator from hybrid logic (Areces and ten Cate 2006) (an alternative approach to restoring the PIP is to admit Boolean TBoxes or, equivalently, the universal role). Given a DL $\mathcal{L}$, we denote by $\mathcal{L}$@ the DL obtained from $\mathcal{L}$ by adding the @-operator as a new concept constructor: if a is an individual name and C an $\mathcal{L}$@-concept, then $@_a C$ is an $\mathcal{L}$@ concept. In every interpretation $\mathcal{I}$, $(@_a C)^{\mathcal{I}} = \Delta^{\mathcal{I}}$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and $(@_a C)^{\mathcal{I}} = \emptyset$ otherwise. The following theorem can now be proved by extending results and techniques introduced in (ten Cate 2005; ten Cate et al. 2006).

Theorem 16 *Assume $\mathcal{L} \in \{\mathcal{ALCH}, \mathcal{ALCHI}, \mathcal{ALCO}@, \mathcal{ALCHO}@, \mathcal{ALCHIO}@\}$. Then Δ-parallel interpolants exist in $\mathcal{L}$ for every $(\mathcal{T}_1, \mathcal{T}_2)$ in $\mathcal{L}$ and $\mathcal{L}$-inclusion α such that 1.–3. from Definition 11 hold and Δ contains all role and individual names in $\mathcal{T}_1, \mathcal{T}_2, \alpha$.*

In particular, for every $\mathcal{T}$ in $\mathcal{L}$ and Δ containing all role and individual names in $\mathcal{T}$, Δ-decompositions of $\mathcal{T}$ in SO coincide with Δ-decompositions of $\mathcal{T}$ in $\mathcal{L}$.

Computing decompositions in expressive DLs

We now exploit the results of the previous two sections to analyze the computational complexity of the problem of computing, given $\mathcal{T} \subseteq_{fin} \mathcal{L}$ and $\Delta \subseteq \mathsf{sig}(\mathcal{T})$, the finest Δ-decomposition of $\mathcal{T}$ in $\mathcal{L}$. We confine ourselves to languages $\mathcal{L}$ in which SO-decompositions coincide with $\mathcal{L}$-decompositions and, therefore, can assume that unique finest decompositions always exist and coincide with the finest Δ-decomposition in SO. In this section, we prove tight complexity bounds for a range of expressive DLs; in the next section, we consider lightweight DLs.

It will be convenient to reformulate the problem of computing the finest Δ-decomposition as a decision problem. Say that a signature Σ (concept C, CI α) is *Δ-decomposable w.r.t. a TBox $\mathcal{T}$* iff there exists a Δ-decomposition $\Sigma_1, \ldots, \Sigma_n$ of $\mathcal{T}$ such that $\Sigma \not\subseteq \Sigma_i \cup \Delta$ ($\mathsf{sig}(C) \not\subseteq \Sigma_i \cup \Delta$, $\mathsf{sig}(\alpha) \not\subseteq \Sigma_i \cup \Delta$) for all $i \leq n$. *Deciding Δ-decomposability in $\mathcal{L}$* means, given a TBox $\mathcal{T}$ in $\mathcal{L}$, $\Delta \subseteq \mathsf{sig}(\mathcal{T})$, and $\sigma_1, \sigma_2 \in \mathsf{sig}(\mathcal{T})$, to check whether σ_1 and σ_2 are Δ-decomposable w.r.t. $\mathcal{T}$. Δ-decomposability may be viewed as the decision problem associated with computing the finest Δ-decomposition of $\mathcal{T}$: it is not difficult to see that the finest Δ-decomposition of $\mathcal{T}$ coincides with the partition of $\mathsf{sig}(\mathcal{T}) \setminus \Delta$ induced by the equivalence relation $\sim$ defined by setting $\sigma_1 \sim \sigma_2$ iff $\{\sigma_1, \sigma_2\}$ are Δ-indecomposable w.r.t. $\mathcal{T}$.

Theorem 17 (Complexity of Δ-decomposability)
In $\mathcal{ALC}$, $\mathcal{ALCI}$, $\mathcal{ALCQ}$, or $\mathcal{ALCQI}$, Δ-decomposability is ExpTime*-complete.*

Proof. We start with the upper bound. Assume a TBox $\mathcal{T}$ in $\mathcal{L}$, a signature $\Delta \subseteq \mathsf{sig}(\mathcal{T})$, and $\sigma_1, \sigma_2 \in \mathsf{sig}(\mathcal{T}) \setminus \Delta$ are given. Enumerate all (exponentially many) partitions Σ_1, Σ_2 of $\mathsf{sig}(\mathcal{T}) \setminus \Delta$ such that $\sigma_1 \in \Sigma_1$ and $\sigma_2 \in \Sigma_2$. Then σ_1, σ_2 are Δ-decomposable w.r.t. $\mathcal{T}$ if, and only if, at least one these partitions is a Δ-decomposition of $\mathcal{T}$. It is thus sufficient to show that the latter problem can be decided in ExpTime. Assume Σ_1, Σ_2 is given. By Theorem 4, Σ_1, Σ_2 is a Δ-decomposition of $\mathcal{T}$ in SO (and, therefore, by the PIP, in $\mathcal{L}$) if, and only if,

$$\{\exists \Sigma_2. \bigwedge_{C \sqsubseteq D \in \mathcal{T}} C \sqsubseteq D, \exists \Sigma_1. \bigwedge_{C \sqsubseteq D \in \mathcal{T}} C \sqsubseteq D\} \models \mathcal{T}.$$

By introducing fresh predicates for the existentially quantified variables, this condition can be checked using standard subsumption checking w.r.t. $\mathcal{L}$-TBoxes, thus in ExpTime (Baader et al. 2003). For the ExpTime-lower bound, observe that a TBox $\mathcal{T}$ is unsatisfiable iff A, B are Δ-decomposable w.r.t. $\mathcal{T} \cup \{A \sqsubseteq B\}$, where A, B are concept names that do not occur in $\mathcal{T}$. Checking unsatisfiability of TBoxes in $\mathcal{L}$ is ExpTime-hard (Baader et al. 2003). ❏

Clearly, this proof does not provide a practical method for computing finest decompositions. For expressive DLs we leave this as future work. Theorem 17 can be generalized in various directions. In the proof, we did not use any specific properties of $\mathcal{L}$, except that SO-decompositions coincide with $\mathcal{L}$-decompositions. Thus the same proof can be used to show that for any such language $\mathcal{L}$ in which subsumption is at least ExpTime-hard, checking Δ-decomposability of two symbols is of the same complexity as subsumption. Together with Theorem 16, we also obtain the following result.

Theorem 18 *In $\mathcal{ALCH}$, $\mathcal{ALCHI}$, $\mathcal{ALCO}@$, $\mathcal{ALCHO}@$, and $\mathcal{ALCHIO}@$, Δ-decomposability with Δ containing all role and invidual names from the input TBox is* ExpTime*-complete.*

For languages $\mathcal{L}$ in which reasoning is strictly less complex than EXPTIME, the proof does not necessarily work because the enumeration step for the signature partitions requires exponential time already. In particular, we cannot use the proof to establish tractability of Δ-decomposability for DLs such as DL-Lite and $\mathcal{EL}$ in which subsumption is tractable.

Decomposition in DL-Lite

Our aim in this section is to establish the PIP and prove tractability of computing the finest Δ-decomposition for members of the DL-Lite family of description logics (Calvanese et al. 2009). We start by investigating the basic language DL-Lite$_{core}$ and then move via DL-Lite$_{horn}$ and full DL-Lite$_{horn}$ to DL-Lite$^{\mathcal{H}}_{horn}$, the extension of DL-Lite$_{horn}$ with role hierarchies. Using the techniques introduced in this section, it is rather straightforward to extend the results presented here to other DL-Lite dialects such as DL-Lite$_{\mathcal{R}}$, DL-Lite$_{\mathcal{F}}$, and DL-Lite$^{\mathcal{N}}_{horn}$ (Calvanese et al. 2006; Artale et al. 2009). The algorithms in this and the subsequent section work by first converting the input TBox $\mathcal{T}$ into an equivalent TBox $\mathcal{T}'$ in which every CI is Δ-indecomposable w.r.t. $\mathcal{T}$. It is not hard to show that, then, $\mathsf{sdeco}_\Delta(\mathcal{T}')$ coincides with the finest Δ-decomposition of $\mathcal{T}'$, and thus of $\mathcal{T}$. In contrast to the "non-constructive" second-order approach underlying the proof of Theorem 17, this also allows to compute a realization $\mathcal{T}_1, \dots, \mathcal{T}_n$ formulated in the same language as the input TBox $\mathcal{T}$.

Recall that *basic DL-Lite concepts* B are defined as

$$B \ ::= \ \top \ | \ \bot \ | \ A \ | \ \exists r \ | \exists r^-,$$

where A ranges over $\mathsf{N_C}$ and r over $\mathsf{N_R}$. DL-Lite$_{core}$-inclusions take the form $B_1 \sqsubseteq B_2$ and $B_1 \sqsubseteq \neg B_2$, where B_1, B_2 are basic DL-Lite concepts. A DL-Lite$_{core}$-TBox is a finite set of DL-Lite$_{core}$-inclusions.

Theorem 19 *DL-Lite$_{core}$ has the PIP. For DL-Lite$_{core}$-TBoxes $\mathcal{T}$ and signatures Δ, one can compute in polynomial time a realization in DL-Lite$_{core}$ of the finest Δ-decomposition $\mathcal{T}$.*

Proof. The algorithm is rather straightforward and almost identical to the algorithm for concept hierarchies in Example 6. First add to $\mathcal{T}$ all DL-Lite$_{core}$ CIs $B_1 \sqsubseteq B_2$ with $\mathcal{T} \models B_1 \sqsubseteq B_2$ and containing not more than one non-Δ-symbol. Now remove from the resulting TBox all $B_1 \sqsubseteq B_2$ containing two non-Δ-symbols for which there exists a concept D that is either a basic DL-Lite concept or its negation and such that $\mathsf{sig}(D) \subseteq \Delta$ and $\mathcal{T} \models B_1 \sqsubseteq D$ and $\mathcal{T} \models D \sqsubseteq B_2$. For the resulting TBox $\mathcal{T}'$, one can show that $\mathsf{sdeco}_\Delta(\mathcal{T}')$ coincides with the finest Δ-decomposition of $\mathcal{T}$. The correctness of this algorithm and the PIP are proved in the full version, but can also be derived from results for more expressive DL-Lite dialects given below. ❑

The construction above can easily be generalized to DL-Lite dialects admitting no conjunctions on the left-hand side of CIs such as DL-Lite$_{\mathcal{R}}$, DL-Lite$_{\mathcal{F}}$, and the dialect underpinning OWL2-QL.

The construction of realizations of finest Δ-decompositions becomes more involved if axioms with conjunctions on the left hand side of CIs are admitted. To illustrate our approach, we provide an example.

Example 20 Let $\Delta = \{D_1, D_2\}$ and

$$\mathcal{T} = \{A_1 \sqcap A_2 \sqsubseteq B, A_1 \sqsubseteq D_1, A_2 \sqsubseteq D_2, D_1 \sqcap D_2 \sqsubseteq A_1\}.$$

In the spirit of the proof of Theorem 19, let us try to replace CIs in $\mathcal{T}$ to make $\mathsf{sdeco}_\Delta(\mathcal{T})$ as fine-grained as possible. Since no CI except $\alpha_0 = (A_1 \sqcap A_2 \sqsubseteq B)$ contains more than one non-Δ-symbol, all CIs distinct from α_0 are Δ-indecomposable w.r.t. $\mathcal{T}$ and replacing them is of no help. So the only CI we attempt to replace is α_0. Intuitively, α_0 is Δ-decomposable w.r.t. $\mathcal{T}$ because $A_1 \sqcap A_2$ is equivalent to a concept not using A_1 in $\mathcal{T} \setminus \{\alpha_0\}$. More precisely,

$$\mathcal{T} \setminus \{\alpha_0\} \models (A_1 \sqcap A_2) \equiv (D_1 \sqcap A_2).$$

Thus we can replace in $\mathcal{T}$ the CI α_0 by $D_1 \sqcap A_2 \sqsubseteq B$. The resulting TBox realizes the partition $\{A_2, B\}, \{A_1\}$ which can be shown to be the finest Δ-decomposition of $\mathcal{T}$. Note that we could have used $D_1 \sqcap D_2 \sqcap A_2$ instead of $D_1 \sqcap A_2$.

Example 20 suggests to extend the algorithm in the proof of Theorem 19 as follows: for each CI $C_0 \sqsubseteq B_0$ in a TBox $\mathcal{T}$ under consideration, we check whether C_0 can be replaced by a concept C_0' with $\mathsf{sig}(C_0') \setminus \Delta \subsetneq \mathsf{sig}(C_0) \setminus \Delta$ such that

$$\mathcal{T} \equiv (\mathcal{T} \setminus \{C_0 \sqsubseteq B_0\}) \cup \{C_0' \sqsubseteq B_0\}.$$

When searching for such a C_0', it turns out to be sufficient to consider concepts C_0' that are equivalent to C_0 w.r.t. $\mathcal{T} \setminus \{C_0 \sqsubseteq B_0\}$. In other words, it is sufficient to search for an *explicit definition*

$$C_0 \equiv C_0'$$

of C_0 that follows from $\mathcal{T} \setminus \{C_0 \sqsubseteq B_0\}$ and in which C_0' is a concept using less non-Δ-symbols than C_0. If one adopts this approach, it remains to find a polytime algorithm searching for explicit definitions of a concept C_0 within a signature Σ. In the case of DL-Lite, one can employ the following greedy algorithm: for a finite signature Σ, let $\mathsf{Cons}_{\mathcal{T},\Sigma}(C_0)$ consist of all basic DL-Lite concepts D with $\mathsf{sig}(D) \subseteq \Sigma$ such that $\mathcal{T} \models C_0 \sqsubseteq D$. This set is finite (in fact, of linear size in the size of Σ) because there are only linearly many basic DL-Lite concepts over any finite signature. It can also be computed in polynomial time. Thus, we can form the conjunction over all concepts in $\mathsf{Cons}_{\mathcal{T},\Sigma}(C_0)$, which, for simplicity, we denote by $\mathsf{Cons}_{\mathcal{T},\Sigma}(C_0)$ as well. In Example 20, one obtains

$$\mathsf{Cons}_{\mathcal{T}\setminus\{\alpha\},\{D_1,D_2,A_2\}}(A_1 \sqcap A_2) = D_1 \sqcap D_2 \sqcap A_2.$$

By definition, $\mathsf{Cons}_{\mathcal{T},\Sigma}(C_0)$ is the *most specific Σ-concept* subsuming C_0 w.r.t. $\mathcal{T}$. Thus, we obtain that there exists an explicit definition C_0' of C_0 w.r.t. $\mathcal{T} \setminus \{C_0 \sqsubseteq B_0\}$ using symbols in Σ only if, and only if,

$$\mathcal{T} \setminus \{C_0 \sqsubseteq B_0\} \models \mathsf{Cons}_{\mathcal{T}\setminus\{C_0 \sqsubseteq B_0\},\Sigma}(C_0) \sqsubseteq C_0,$$

and, if this happens to be the case, then $\mathsf{Cons}_{\mathcal{T}\setminus\{C_0 \sqsubseteq B_0\},\Sigma}(C_0)$ is such a definition. Finally, to test whether there is some Σ containing less non-Δ-symbols than $\mathsf{sig}(C_0)$ with this property, one can go through all

Input: Propositional DL-Lite$_{\text{horn}}$ TBox $\mathcal{T}$ and signature $\Delta \subseteq \mathsf{sig}(\mathcal{T})$.

Apply exhaustively the following transformation rule to each $\alpha = C \sqsubseteq B \in \mathcal{T}$ such that $|\mathsf{sig}(\alpha) \setminus \Delta| \geq 2$.

If $\mathcal{T} \setminus \{\alpha\} \models \alpha$
Then
    $\mathcal{T} := \mathcal{T} \setminus \{\alpha\}$.
Else
    If $\mathsf{sig}(C) \not\subseteq \Delta$, $\mathsf{sig}(B) \not\subseteq \Delta$, and $\mathcal{T} \models \mathsf{Cons}_{\mathcal{T},\Delta}(C) \sqsubseteq B$
    Then
        $\mathcal{T} := (\mathcal{T} \setminus \{\alpha\})$;
        $\mathcal{T} := \mathcal{T} \cup \{\mathsf{Cons}_{\mathcal{T},\Delta}(C) \sqsubseteq B\} \cup \bigcup_{B' \in \mathsf{Cons}_{\mathcal{T},\Delta}(C)} \{C \sqsubseteq B'\}$
    If for some $X \in \mathsf{sig}(C) \setminus \Delta$
        $\mathcal{T} \setminus \{\alpha\} \models \mathsf{Cons}_{\mathcal{T}\setminus\{\alpha\},(\mathsf{sig}(C)\setminus\{X\})\cup\Delta}(C) \sqsubseteq C$
    Then
        $\mathcal{T} := (\mathcal{T} \setminus \{\alpha\}) \cup \{\mathsf{Cons}_{\mathcal{T}\setminus\{\alpha\},(\mathsf{sig}(C)\setminus\{X\})\cup\Delta}(C) \sqsubseteq B\}$

Figure 2: Procedure $\mathsf{Rewrite}_{\text{PropDL-Lite}_{\text{horn}}}$

$\Sigma := (\Delta \cup \mathsf{sig}(C_0)) \setminus \{X\}$ for $X \in \mathsf{sig}(C_0) \setminus \Delta$. Since the finest decomposition is unique, the order in which we go through such Σ's does not matter.

We now present the algorithm implementing this approach in detail. Recall that a DL-Lite$_{horn}$-inclusion takes the form $B_1 \sqcap \cdots \sqcap B_m \sqsubseteq B$, where the $B_1, \ldots, B_m$ and B are basic DL-Lite concepts. We first consider *propositional DL-Lite$_{horn}$*, i.e., DL-Lite$_{horn}$-inclusions and TBoxes not containing any roles. Of course propositional DL-lite$_{horn}$ is nothing else but propositional Horn-logic. We first observe that DL-Lite$_{\text{horn}}$ and propositional DL-Lite$_{\text{horn}}$ have the PIP; so it does not make any difference whether we consider signature decompositions realized in DL-Lite$_{horn}$ or in SO:

Lemma 21 *DL-Lite$_{\text{horn}}$ and propositional DL-Lite$_{\text{horn}}$ have the PIP.*

Proof. It is shown in (Kontchakov, Wolter, and Zakharyaschev 2010) that DL-Lite$_{\text{horn}}$ has the RJCP. By Theorem 14, DL-Lite$_{\text{horn}}$ has the PIP. The same proof works for propositional DL-Lite$_{\text{horn}}$. ❑

Theorem 22 *For any propositional DL-Lite$_{\text{horn}}$ TBox $\mathcal{T}$, the algorithm in Figure 2 runs in poly-time and outputs an equivalent TBox $\mathcal{T}'$ in which every CI is Δ-indecomposable w.r.t. $\mathcal{T}$. Thus,* $\mathsf{sdeco}_\Delta(\mathcal{T}')$ *coincides with the finest Δ-decomposition of $\mathcal{T}$.*

Proof. We provide a sketch of the correctness proof; a detailed proof can be found in the full version. Denote by $\mathcal{T}$ the output of the algorithm in Figure 2. It can be verified that this TBox is equivalent to the original TBox. Moreover it has the following properties:

(Red) For every $\alpha \in \mathcal{T}$ with $|\mathsf{sig}(\alpha) \setminus \Delta| \geq 2$, we have $\mathcal{T} \setminus \{\alpha\} \not\models \alpha$;

(Def) If for some $\alpha = (C \sqsubseteq B) \in \mathcal{T}$ and $\Sigma \subseteq \mathsf{sig}(C) \setminus \Delta$ we have
$$\mathcal{T} \setminus \{\alpha\} \models \mathsf{Cons}_{\mathcal{T}\setminus\{\alpha\},\Sigma\cup\Delta}(C) \sqsubseteq C,$$
then $\Sigma = \mathsf{sig}(C) \setminus \Delta$.

(Int) For any $C \sqsubseteq B \in \mathcal{T}$ such that $\mathsf{sig}(C) \not\subseteq \Delta$ and $\mathsf{sig}(B) \not\subseteq \Delta$, we have
$$\mathcal{T} \not\models \mathsf{Cons}_{\mathcal{T},\Delta}(C) \sqsubseteq B.$$

Thus, it is sufficient to prove the following

Claim. If a TBox $\mathcal{T}$ has properties (Red), (Def), and (Int), then every CI in $\mathcal{T}$ is Δ-indecomposable w.r.t. $\mathcal{T}$.

Proof of Claim. Suppose that some $\alpha = C \sqsubseteq B \in \mathcal{T}$ is Δ-decomposable w.r.t. $\mathcal{T}$. Then either

(a) there exists a signature Δ-decomposition Σ_1, Σ_2 of $\mathcal{T}$ such that $\mathsf{sig}(C) \cap \Sigma_i \neq \emptyset$ for $i = 1, 2$ or

(b) there exists a signature Δ-decomposition Σ_1, Σ_2 of $\mathcal{T}$ such that $\mathsf{sig}(C) \cap \Sigma_2 \neq \emptyset$, $\mathsf{sig}(C) \subseteq \Sigma_2 \cup \Delta$, and $\mathsf{sig}(B) \subseteq \Sigma_1$.

We show that, in both cases, a contradiction can be derived. We use the following notation for renaming symbols within concepts, CIs, and TBoxes. Let D be a concept. By D_{Σ_1} we denote the concept obtained from D by replacing every occurrence of a symbol $x \in \Sigma_2$ with a fresh symbol x'. By D_{Σ_2} we denote the concept obtained from D by replacing every occurrence of a symbol $x \in \Sigma_1$ with a fresh symbol $x^\dagger$. The CIs α_{Σ_1}, α_{Σ_2} and TBoxes $\mathcal{T}_{\Sigma_1}$, $\mathcal{T}_{\Sigma_2}$ are defined in the same way. Recall from Theorem 4 that Σ_1, Σ_2 is a Δ-decomposition of $\mathcal{T}$ if, and only if,
$$\mathcal{T}_{\Sigma_1} \cup \mathcal{T}_{\Sigma_2} \models \alpha$$
for all $\alpha \in \mathcal{T}$.

Consider now Case (a). By (Red), we have $\mathcal{T} \setminus \{\alpha\} \not\models \alpha$. Therefore,
$$(\mathcal{T} \setminus \{\alpha\})_{\Sigma_1} \cup (\mathcal{T} \setminus \{\alpha\})_{\Sigma_2} \not\models \alpha. \tag{1}$$
On the other hand,
$$\mathcal{T}_{\Sigma_1} \cup \mathcal{T}_{\Sigma_2} \models \alpha,$$
since Σ_1, Σ_2 is a Δ-decomposition of $\mathcal{T}$. Thus, there exists $i \in \{1, 2\}$ such that
$$\mathcal{T}_{\Sigma_1} \cup \mathcal{T}_{\Sigma_2} \models C \sqsubseteq C_{\Sigma_i} \tag{2}$$
because otherwise, by (1) we would find a (propositional) model $\mathcal{I}$

- satisfying $(\mathcal{T} \setminus \{\alpha\})_{\Sigma_1} \cup (\mathcal{T} \setminus \{\alpha\})_{\Sigma_2}$ and C;
- and refuting C_{Σ_1}, C_{Σ_2}, and B.

For such an $\mathcal{I}$ we would have $\mathcal{I} \models \alpha_{\Sigma_1}$ and $\mathcal{I} \models \alpha_{\Sigma_2}$ and, therefore, $\mathcal{I} \models \mathcal{T}_{\Sigma_1} \cup \mathcal{T}_{\Sigma_2}$ but $\mathcal{I} \not\models C \sqsubseteq B$, which is a contradiction.

Now one can show (the proof is non-trivial and given in the full paper) that (2) implies
$$\mathcal{T} \setminus \{\alpha\} \models \mathsf{Cons}_{\mathcal{T}\setminus\{\alpha\},\Delta\cup(\Sigma_i\cap\mathsf{sig}(C)))}(C) \sqsubseteq C,$$
which contradicts (Def).

In Case (b), one can show (the proof is non-trivial) that
$$\mathcal{T} \models \mathsf{Cons}_{\mathcal{T},\Delta}(C) \sqsubseteq B,$$
which contradicts (Int). ❑

Input: DL-Lite$_{\text{horn}}$ TBox $\mathcal{T}$ and signature $\Delta \subseteq \mathsf{sig}(\mathcal{T})$.
1. **Let** $\mathcal{T}_{\text{Aux}} := \{\exists r \sqsubseteq \bot \mid \mathcal{T} \models \exists r \sqsubseteq \bot\}$
2. **Let** $\mathcal{T}^P_{\text{Res}} := \mathsf{Rewrite}_{\text{PropDL-Lite}_{\text{horn}}}(\mathcal{T}^P \cup \mathcal{T}^P_{\text{Aux}}, \Delta^P)$
3. Let $\mathcal{T}'$ be the result of replacing in $\mathcal{T}^P_{\text{Res}}$ expressions of the form $P_{\exists r}$, for $r \in \mathsf{N_R}$, with $\exists r$ and $P_{\exists r^-}$ with $\exists r^-$.
4. **Return** $\mathcal{T}'$

Figure 3: Procedure $\mathsf{Rewrite}_{\text{DL-Lite}_{\text{horn}}}$

We now consider DL-Lite$_{\text{horn}}$. The following lemma shows that reasoning in DL-Lite$_{\text{horn}}$ can be reduced to reasoning in propositional DL-Lite$_{\text{horn}}$. Its proof is similar to the ones of results in (Artale et al. 2009) relating DL-Lite dialects and fragments of first-order logic.

Given a DL-Lite$_{\text{horn}}$ concept C (CI $C \sqsubseteq B$ or TBox $\mathcal{T}$, respectively), we consider a propositional DL-Lite$_{\text{horn}}$ concept C^P (propositional CI $C^P \sqsubseteq B^P$ or propositional TBox $\mathcal{T}^P$) obtained by replacing every occurrence of an expression of the form $\exists r$ (resp. $\exists r^-$) with its *surrogate*, a fresh concept name $P_{\exists r}$ (resp. $P_{\exists r^-}$). We assume that surrogates do not occur in the given DL-Lite$_{\text{horn}}$ concept (CI, TBox, respectively). Let Σ be a signature. We define its propositional counterpart as

$$\Sigma^P = \{A \mid A \in \Sigma, A \in \mathsf{N_C}\} \cup \{P_{\exists r}, P_{\exists r^-} \mid r \in \Sigma, r \in \mathsf{N_R}\}.$$

The following is readily checked.

Lemma 23 *Let $C \sqsubseteq B$ be a DL-Lite$_{horn}$ CI, and $\mathcal{T}$ a satisfiable DL-Lite$_{horn}$ TBox such that for all roles r, if $\mathcal{T} \models \exists r \sqsubseteq \bot$, then $\exists r \sqsubseteq \bot \in \mathcal{T}$. Then $\mathcal{T} \models C \sqsubseteq B$ if, and only if, $\mathcal{T}^P \models C^P \sqsubseteq B^P$.*

Using Lemma 23 one can now prove the correctness of the algorithm given in Figure 3.

Theorem 24 *The algorithm $\mathsf{Rewrite}_{\text{DL-Lite}_{horn}}$ given in Fig. 3 transforms a given DL-Lite$_{horn}$ TBox into an equivalent DL-Lite$_{horn}$ TBox in which every CI is Δ-indecomposable.*

Finally, we consider DL-Lite$^{\mathcal{H}}_{\text{horn}}$, the extension of DL-Lite$_{\text{horn}}$ with role inclusions $r \sqsubseteq s$. This time, we employ a reduction of DL-Lite$^{\mathcal{H}}_{\text{horn}}$ to DL-Lite$_{\text{horn}}$.

Lemma 25 *Let $\mathcal{T}$ be a DL-Lite$^{\mathcal{H}}_{horn}$ TBox and Δ a signature. Let $\mathcal{T}^0$ be the set of CIs in $\mathcal{T}$ and set*

$$\mathcal{T}' = \mathcal{T}^0 \cup \{\exists r \sqsubseteq \exists s, \exists r^- \sqsubseteq \exists s^- \mid r \sqsubseteq s \in \mathcal{T}\}.$$

Then $\mathcal{T} \models \alpha$ if, and only if, $\mathcal{T}' \models \alpha$ for all CIs α in DL-Lite$_{horn}$.

Using this reduction, one can show that DL-Lite$^{\mathcal{H}}_{\text{horn}}$ has the PIP and one can prove the correctness of the algorithm given in Figure 4.

Theorem 26 *The algorithm $\mathsf{Rewrite}_{\text{DL-Lite}^{\mathcal{H}}_{horn}}$ given in Fig. 4 transforms a given DL-Lite$^{\mathcal{H}}_{horn}$ TBox into an equivalent DL-Lite$^{\mathcal{H}}_{horn}$ TBox in which every inclusion is Δ-indecomposable.*

Input: DL-Lite$^{\mathcal{H}}_{\text{horn}}$ TBox $\mathcal{T}$ and signature $\Delta \subseteq \mathsf{sig}(\mathcal{T})$.
1. **Let** $\mathcal{T}_C$ be the the set of CI in $\mathcal{T}$
2. **Let** $\mathcal{T}_R$ be the the set of RI in $\mathcal{T}$
3. $\mathcal{T}_C := \mathcal{T}_C \cup \{\exists r \sqsubseteq \exists s \mid \mathcal{T} \models \exists r \sqsubseteq \exists s\} \cup \{\exists r \sqsubseteq \bot \mid \mathcal{T} \models \exists r \sqsubseteq \bot\}$
4. $\mathcal{T}_C := \mathsf{Rewrite}_{\text{DL-Lite}_{\text{horn}}}(\mathcal{T}_C, \Delta)$
5. **For all** $r \sqsubseteq s \in \mathcal{T}_R$ **do**
6. **If** $\mathcal{T}_C \models \exists r \sqsubseteq \bot$
7. **Then** $\mathcal{T}_R := \mathcal{T}_R \setminus \{r \sqsubseteq s\}$
8. **Else if** $\mathcal{T}_R \models r \sqsubseteq t$ and $\mathcal{T}_R \models t \sqsubseteq s$ for some $t \in \Delta$
9. **Then**
10. $\mathcal{T}_R := (\mathcal{T}_R \setminus \{r \sqsubseteq s\}) \cup \{r \sqsubseteq t\} \cup \{t \sqsubseteq s\}$
11. **Return** $(\mathcal{T}_C \cup \mathcal{T}_R)$

Figure 4: Procedure $\mathsf{Rewrite}_{\text{DL-Lite}^{\mathcal{H}}_{\text{horn}}}$

Decomposition in $\mathcal{EL}$

We have seen already (Theorem 13) that $\mathcal{EL}$ and $\mathcal{ELH}$ have the PIP. In this section, we focus on computing the finest Δ-decompositions in $\mathcal{EL}$. In contrast to DL-Lite, we have partial results only. Call an $\mathcal{EL}$-TBox $\mathcal{T}$ *role-acyclic* if there does not exist an $\mathcal{EL}$-concept C and role names $r_1, \ldots, r_n$ with $n \geq 1$ such that $\mathcal{T} \models C \sqsubseteq \exists r_1. \cdots \exists r_n.C$ Note that acyclic terminologies such as SNOMED CT satisfy this condition.

Theorem 27 *Let*

1. *$\Delta = \emptyset$ and $\mathcal{T}$ be an arbitrary $\mathcal{EL}$-TBox; or*
2. *Δ arbitrary and $\mathcal{T}$ be a role-acyclic TBox.*

Then the finest Δ-decomposition of $\mathcal{T}$ can be computed in polynomial time.

It remains an open problem whether this results holds for arbitrary $\mathcal{EL}$-TBoxes. We nowgive a sketch of the main ideas behind the proof for Point 2. First, using results from (Lutz and Wolter 2010), one can transform any given $\mathcal{EL}$-TBox $\mathcal{T}_0$ and signature Δ_0 into a new TBox $\mathcal{T}$ and signature Δ (which is, modulo fresh definitions $A \equiv C$, equivalent to $\mathcal{T}_0$) such that the finest Δ-decomposition of $\mathcal{T}$ can be transformed in linear time into the finest Δ_0-decomposition of $\mathcal{T}_0$ and such that:

(Dec) if $C \sqsubseteq D \in \mathcal{T}$, then D is Δ-indecomposable w.r.t. $\mathcal{T}$.

The full version describes how $\mathcal{T}$ and Δ can be computed. Given $\mathcal{T}$ and Δ satisfying (Dec), we want to proceed in the same way as for DL-Lite: a CI $\alpha = (C \sqsubseteq D) \in \mathcal{T}$ should be simplified to $C' \sqsubseteq D$ if C' is an explicit definition of C relative to $\mathcal{T} \setminus \{\alpha\}$ using less non-Δ-symbols than C. This simplification will again rely on sets of concepts $\mathsf{Cons}_{\mathcal{T},\Sigma}(C)$ consisting of all $\mathcal{EL}$-concepts D such that $\mathcal{T} \models C \sqsubseteq D$ and $\mathsf{sig}(D) \subseteq \Sigma$. However, there are two additional difficulties compared to DL-Lite: first, we do not currently know whether this approach is complete for arbitrary $\mathcal{EL}$-TBoxes. For this reason, our procedure is restricted to role-acyclic TBoxes. Second, even for role-acyclic TBoxes, explicit definitions can be of exponential size. Even worse and as shown by the following example, this problem actually manifests itself in realizations of finest Δ-decompositions, which can also be of exponential size.

Example 28 Let $\mathcal{T}$ consist of $A_i \equiv \exists r_i.A_{i+1} \sqcap \exists s_i.A_{i+1}$, for $0 \leq i < n$, and $A_n \equiv \top$. For

$$\Delta = \{r_0, \ldots, r_{n-1}, s_0, \ldots, s_{n-1}\},$$

the finest Δ-decomposition of $\mathcal{T}$ is $\{A_0\}, \ldots, \{A_n\}$ because we can define a realization $\mathcal{T}_0, \ldots, \mathcal{T}_n$ by setting, inductively,

$$\mathcal{T}_n = \{A_n \equiv \top\},$$
$$\mathcal{T}_i = \{A_i \equiv \exists r_i.C_{i+1} \sqcap \exists s_i.C_{i+1}\},$$

where

$$C_n = \bot, \quad C_i = \exists r_i.C_{i+1} \sqcap \exists s_i.C_{i+1}.$$

This realization is of exponential size and that there does not exist any smaller realization of $\{A_0\}, \ldots, \{A_n\}$ using $\mathcal{EL}$-TBoxes: the smallest explicit definition of A_0 that does not use the symbols $\{A_1, \ldots, A_n\}$ corresponds to the concept representing the binary tree with edges s_i and r_i and is, therefore, of exponential size.

To resolve this problem, we consider realizations not in $\mathcal{EL}$ but in the extension $\mathcal{EL}^{\nu+}$ of $\mathcal{EL}$ by greatest fixpoints introduced and investigated in (Lutz, Piro, and Wolter 2010). In this language, explicit definitions are always of polynomial size, they can be computed in polynomial time, and, importantly, reasoning is still tractable. It will be convenient for us to use a syntactic variant, $\mathcal{EL}^{st}$, of $\mathcal{EL}^{\nu+}$ using *simulation quantifiers* instead of greatest fixpoints. To define $\mathcal{EL}^{st}$, let $\mathcal{I}_1$ and $\mathcal{I}_2$ be interpretations, $d_1 \in \Delta^{\mathcal{I}_1}$, $d_2 \in \Delta^{\mathcal{I}_2}$ and Σ a signature. A relation $S \subseteq \Delta^{\mathcal{I}_1} \times \Delta^{\mathcal{I}_2}$ containing (d_1, d_2) is a Σ-*simulation* from $(\mathcal{I}_1, d_1)$ to $(\mathcal{I}_2, d_2)$ if

- for all concept names $A \in \Sigma$ and all $(e_1, e_2) \in S$, if $e_1 \in A^{\mathcal{I}_1}$, then $e_2 \in A^{\mathcal{I}_2}$;
- for all role names $r \in \Sigma$, all $(e_1, e_2) \in S$, and all $e_1' \in \Delta^{\mathcal{I}_1}$ with $(e_1, e_1') \in r^{\mathcal{I}_1}$, there exists $e_2' \in \Delta^{\mathcal{I}_2}$ such that $(e_2, e_2') \in r^{\mathcal{I}_2}$ and $(e_1', e_2') \in S$.

The relationship between $\mathcal{EL}$ and simulations has been investigated and employed extensively (Lutz and Wolter 2010; Lutz, Piro, and Wolter 2010). One important connection is that whenever there is a Σ-simulation from $(\mathcal{I}_1, d_1)$ to $(\mathcal{I}_2, d_2)$, then d_2 is an instance of any Σ-concept of which d_1 is an instance; the converse holds for all interpretations of finite outdegree. We now define $\mathcal{EL}^{st}$-concepts, CIs, and TBoxes by simultaneous induction as follows, see (Lutz, Piro, and Wolter 2010):

- every $\mathcal{EL}$-concept (CI, TBox) is an $\mathcal{EL}^{st}$-concept (CI, TBox);
- if C is an $\mathcal{EL}^{st}$-concept, $\mathcal{T}$ is an $\mathcal{EL}^{st}$-TBox, and Σ a signature, then $\exists^{sim}\Sigma.(\mathcal{T}, C)$ is an $\mathcal{EL}^{st}$-concept;
- if C and D are $\mathcal{EL}^{st}$-concepts, then $C \sqsubseteq D$ is an $\mathcal{EL}^{st}$-CI; a finite set of $\mathcal{EL}^{st}$ CIs is an $\mathcal{EL}^{st}$-TBox.

The semantics of simulation operators is defined as follows. For any interpretation $\mathcal{I}$ and $d \in \Delta^{\mathcal{I}}$, let $d \in (\exists^{sim}\Sigma.(\mathcal{T}, C))^{\mathcal{I}}$ iff there exists a model $\mathcal{J}$ of $\mathcal{T}$ with a $d' \in C^{\mathcal{J}}$ such that there is a Γ-simulation from $(\mathcal{J}, d')$ to $(\mathcal{I}, d)$, where $\Gamma = (\mathsf{sig}(\mathcal{T}) \cup \mathsf{sig}(C)) \setminus \Sigma$.

Example 29 Consider the TBox $\mathcal{T}$ from Example 28. Then

$$\mathcal{T} \models C_i \equiv \exists^{sim}\{A_0, \ldots, A_n\}(\mathcal{T}, A_i),$$

for all $i \leq n$. Thus, one can realize $\{A_0\}, \ldots, \{A_n\}$ using the TBoxes $\mathcal{T}_i = \{A_i \equiv \exists^{sim}\{A_0, \ldots, A_n\}(\mathcal{T}, A_i)\}$.

Now consider the sets of concepts $\mathsf{Cons}_{\mathcal{T},\Sigma}(C)$. In contrast to the DL-Lite case, these sets can clearly be infinite. In the case of role-acyclic TBoxes, though, one can show that there always is a *finite* set of $\mathcal{EL}$-concepts equivalent to $\mathsf{Cons}_{\mathcal{T},\Sigma}(C)$. To avoid exponential size as in Example 28, we now show how to use simulation quantifiers to give a succinct representation of this finite set.

Even for arbitrary TBoxes, it is possible to prove that the concept $\exists^{sim}\Gamma.(\mathcal{T}, C)$, where $\Gamma = \mathsf{sig}(\mathcal{T}, C) \setminus \Sigma$, represents $\mathsf{Cons}_{\mathcal{T},\Sigma}(C)$ in the sense that

- $\mathcal{T} \models C \sqsubseteq \exists^{sim}\Gamma.(\mathcal{T}, C)$ and
- $\mathcal{T} \models \exists^{sim}\Gamma.(\mathcal{T}, C) \sqsubseteq D$ for all $D \in \mathsf{Cons}_{\mathcal{T},\Sigma}(C)$.

Since for role-acyclic TBoxes $\mathsf{Cons}_{\mathcal{T},\Sigma}(C)$ is equivalent to a finite set of $\mathcal{EL}$-concepts, we thus obtain the following.

Proposition 30 *Let $\mathcal{T}$ be a role-acyclic TBox and C an $\mathcal{EL}$-concept. For $\Gamma = \mathsf{sig}(\mathcal{T}, C) \setminus \Sigma$, the concept $\exists^{sim}\Gamma.(\mathcal{T}, C)$ is equivalent to the conjunction over all concepts in* $\mathsf{Cons}_{\mathcal{T},\Sigma}(C)$.

It follows that we can use the linear size concept $\exists^{sim}\Gamma.(\mathcal{T}, C)$ in place of $\mathsf{Cons}_{\mathcal{T},\Sigma}(C)$. The algorithm presented in Figure 5 is now almost a copy of the transformation algorithm for propositional DL-Lite$_{\text{horn}}$ in Figure 2. As reasoning in $\mathcal{EL}^{st}$ is still tractable (Lutz, Piro, and Wolter 2010), this algorithm runs in polynomial time. A detailed (and rather involved) proof of the following result is given in the full paper.

Theorem 31 *The algorithm $\mathsf{Rewrite}_{\mathcal{EL}}$ given in Fig. 5 transforms a given role-acyclic $\mathcal{EL}$-TBox satisfying (Dec) into an equivalent $\mathcal{EL}^{st}$-TBox in which every CI is Δ-indecomposable.*

Conclusion

We have established the theoretical foundations for a syntax-independent approach to signature decomposition in ontologies. Our investigation has been inspired by previous work in propositional logic, belief revision, and abstract logical calculi (Parikh 1999; Kourousias and Makinson 2007; Ponomaryov 2008). Of course, a semantic approach leads to reasoning services of higher complexity than purely syntactic approaches. Still, the results are quite promising: for many lightweight DLs, the main reasoning problem is still tractable and for expressive DLs it is not harder than subsumption checking. This shows that signature decomposition is computationally much simpler than semantically complete approaches to other modularization tasks such as module extraction, conservative extensions, and forgetting/uniform interpolation (Konev et al. 2009;

Input: $\mathcal{EL}$ TBox $\mathcal{T}$ satisfying (Dec) and signature $\Delta \subseteq \mathsf{sig}(\mathcal{T})$.

Apply exhaustively the following transformation rule to each $\alpha = C \sqsubseteq B \in \mathcal{T}$ such that $|\mathsf{sig}(\alpha) \setminus \Delta| \geq 2$.

1. **If** $\mathcal{T} \setminus \{\alpha\} \models \alpha$
2. **Then**
3. $\quad \mathcal{T} := \mathcal{T} \setminus \{\alpha\}$.
4. **Else**
5. $\quad$ **If** $\mathsf{sig}(C) \not\subseteq \Delta$, $\mathsf{sig}(D) \not\subseteq \Delta$, and $\mathcal{T} \models \exists^{sim}(\mathsf{sig}(\mathcal{T}) \setminus \Delta).(\mathcal{T}, C) \sqsubseteq D$
6. $\quad$ **Then**
7. $\quad \mathcal{T} := (\mathcal{T} \setminus \{\alpha\})$;
8. $\quad \mathcal{T} := \mathcal{T} \cup \{C \sqsubseteq \exists^{sim}(\mathsf{sig}(\mathcal{T}) \setminus \Delta).(\mathcal{T}, C)\} \cup \{\exists^{sim}(\mathsf{sig}(\mathcal{T}) \setminus \Delta).(\mathcal{T}, C) \sqsubseteq D\}$
9. $\quad$ **If** for $X \in \mathsf{sig}(C) \setminus \Delta$ and $\Gamma = \{X\} \cup \mathsf{sig}(\mathcal{T}) \setminus (\Delta \cup \mathsf{sig}(C))$
10. $\quad \mathcal{T} \setminus \{\alpha\} \models \exists^{sim}\Gamma.(\mathcal{T} \setminus \{\alpha\}, C) \sqsubseteq C$
11. $\quad$ **Then**
12. $\quad \mathcal{T} := (\mathcal{T} \setminus \{\alpha\}) \cup \{\exists^{sim}\Gamma.(\mathcal{T} \setminus \{\alpha\}, C) \sqsubseteq D\}$

Figure 5: Procedure $\mathsf{Rewrite}_{\mathcal{EL}}$

Lutz, Walther, and Wolter 2007; Cuenca Grau et al. 2008). Future work will include decomposition experiments with existing ontologies and the development of guidelines to determine meaningful Δ's.

References

Amir, E., and McIlraith, S. A. 2005. Partition-based logical reasoning for first-order and propositional theories. *Artificial Intelligence* 162(1-2):49–88.

Areces, C., and ten Cate, B. 2006. Hybrid logics. In Blackburn, P.; van Benthem, J.; and Wolter, F., eds., *Handbook of Modal Logic*. Elsevier.

Artale, A.; Calvanese, D.; Kontchakov, R.; and Zakharyaschev, M. 2009. The DL-Lite family and relations. *Journal of Artificial Intelligence Research* 36:1–69.

Baader, F.; Calvanes, D.; McGuiness, D.; Nardi, D.; and Patel-Schneider, P. 2003. *The Description Logic Handbook: Theory, implementation and applications*. Cambridge University Press.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2006. Data complexity of query answering in description logics. In *Proceedings of KR 2006*, 260–270.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; Poggi, A.; Rodriguez-Muro, M.; and Rosati, R. 2009. Ontologies and databases: The DL-Lite approach. In *Reasoning Web 2009*. Springer. 255–356.

Cuenca Grau, B.; Parsia, B.; Sirin, E.; and Kalyanpur, A. 2006. Modularity and web ontologies. In *Proceedings of KR 2006*, 198–209.

Cuenca Grau, B.; Horrocks, I.; Kazakov, Y.; and Sattler, U. 2008. Modular reuse of ontologies: Theory and practice. *Journal of Artificial Intelligence Research* 31:273–318.

d'Aquin, M.; Schlicht, A.; Stuckenschmidt, H.; and Sabou, M. 2009. Criteria and evaluation for ontology modularization techniques. In *Modular Ontologies*, volume 5445 of *Lecture Notes in Computer Science*. Springer. 67–89.

Konev, B.; Lutz, C.; Walther, D.; and Wolter, F. 2009. Formal properties of modularisation. In *Modular Ontologies*, volume 5445 of *Lecture Notes in Computer Science*. Springer. 25–66.

Konev, B.; Lutz, C.; Ponomaryov, D.; and Wolter, F. 2010. Decomposing description logic ontologies. Technical report, University of Liverpool.

Kontchakov, R.; Wolter, F.; and Zakharyaschev, M. 2010. Logic-based ontology comparison and module extraction, with an application to DL-Lite. To appear in Journal of Artificial Intelligence.

Kourousias, G., and Makinson, D. 2007. Parallel interpolation, splitting, and relevance in belief change. *Journal of Symbolic Logic* 72(3):994–1002.

Lutz, C., and Wolter, F. 2010. Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. *Journal of Symbolic Computation* 45(2):194–228.

Lutz, C.; Piro, R.; and Wolter, F. 2010. EL-concepts go second-order: Greatest fixpoints and simulation quantifiers. Submitted.

Lutz, C.; Walther, D.; and Wolter, F. 2007. Conservative extensions in expressive description logics. In *Proc. of IJCAI*, 453–458.

Parikh, R. 1999. Beliefs, belief revision, and splitting languages. In *Logic, Language and Information*. CSLI Publications. 266–278.

Ponomaryov, D. 2008. On decomposibility in logical calculi. *Bulletin of the Novosibirsk Computing Center* 28:111–120.

Seylan, I.; Franconi, E.; and de Bruijn, J. 2009. Effective query rewriting with ontologies over dboxes. In *Proceedings of IJCAI 2009*, 923–925.

Stuckenschmidt, H.; Parent, C.; and Spaccapietra, S., eds. 2009. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, volume 5445 of *Lecture Notes in Computer Science*. Springer.

ten Cate, B.; Conradie, W.; Marx, M.; and Venema, Y. 2006. Definitorially complete description logics. In *Proceedings of KR 2006*, 79–89.

ten Cate, B. 2005. *Model theory for extended modal languages*. Ph.D. Dissertation, University of Amsterdam.