Detecting Conjunctive Query Differences between $\mathcal{ELH}^r$-Terminologies using Hypergraphs*

Michel Ludwig and Dirk Walther

TU Dresden, Theoretical Computer Science, Germany
Center for Advancing Electronics Dresden
{michel,dirk}@tcs.inf.tu-dresden.de

Abstract. We present a new method for detecting logical differences between $\mathcal{EL}$-terminologies extended with role inclusions, domain and range restrictions of roles using a hypergraph representation of ontologies. In this paper we consider differences given by pairs consisting of a conjunctive query and of an ABox formulated over a vocabulary of interest. We define a simulation notion between such hypergraph representations and we show that the existence of simulations coincides with the absence of a logical difference. To demonstrate the practical applicability of our approach, we evaluate a prototype implementation on large ontologies.

1 Introduction

The aim of this paper is to propose and investigate a novel and coherent approach to the logical difference problem as introduced in [4,6,7] using a hypergraph representation of ontologies. The logical difference is taken to be the set of queries relevant to an application domain that produce different answers when evaluated over ontologies that are to be compared. The language and signature of the queries can be adapted in such a way that exactly the differences of interest become visible, which can be independent of the syntactic representation of the ontologies. Three types of queries have been studied so far: concept subsumptions, instance and conjunctive queries. The logical difference problem involves reasoning tasks such as determining the existence of a difference and of a succinct representation of the entire set of queries that lead to different answers. Other relevant tasks include the construction of an example query that yields different answers from ontologies given a representation of the difference, as well as finding explanations, i.e. the axioms by which this query is entailed.

Our approach is based on representing ontologies as hypergraphs and computing simulations between them. Hypergraphs are a generalisation of graphs with many applications in computer science and discrete mathematics. In knowledge representation, hypergraphs have been used implicitly to define reachability-based modules of ontologies [10], explicitly to define locality-based modules [9]

* Supported by the German Research Foundation (DFG) via the Cluster of Excellence 'cfAED'.

and to perform efficient reasoning with ontologies [8]. We consider ontologies that can be translated into directed hypergraphs by taking the concept and role names that occur in them as nodes and treating the axioms as hyperedges. For instance, the axiom $A \sqsubseteq \exists r.B$ is translated into the hyperedge $(\{x_A\}, \{x_r, x_B\})$, and the axiom $A \equiv B_1 \sqcap B_2$ into the hyperedges $(\{x_A\}, \{x_{B_1}\})$, $(\{x_A\}, \{x_{B_2}\})$ and $(\{x_{B_1}, x_{B_2}\}, \{x_A\})$.

In this paper we consider the conjunctive query difference between ontologies formulated as terminologies in the description logic $\mathcal{ELH}^r$, i.e. the extension of $\mathcal{EL}$ with role inclusions and domain & range restrictions [1]. Given a signature Σ (i.e. a set of concept and role names), the *Σ-query difference* between TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ is the set $\mathsf{qDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ of pairs of the form $(\mathcal{A}, q(a))$, where $\mathcal{A}$ is an ABox and $q(\boldsymbol{x})$ a conjunctive query which both only use symbols from Σ, and $\boldsymbol{a}$ is a tuple of individual names from $\mathcal{A}$ such that $(\mathcal{T}_1, \mathcal{A}) \models q(\boldsymbol{a})$ and $(\mathcal{T}_2, \mathcal{A}) \not\models q(\boldsymbol{a})$ [4]. An extension of $\mathcal{ELH}^r$ has been introduced in [4] whose concept subsumptions taken as queries detect the same differences between general $\mathcal{ELH}^r$-TBoxes as conjunctive queries. Thus it is sufficient to consider concept subsumption queries over the extended language only. *Primitive witness theorems* state that for every concept subsumption in the difference between $\mathcal{ELH}^r$-terminologies, there are also *simpler* subsumptions of the form $A \sqsubseteq C$, $(\mathsf{dom}(r) \sqsubseteq C$, $\mathsf{ran}(r) \sqsubseteq C)$, or $D \sqsubseteq A$ that have an atomic concept (or a domain/range expression), called *witness*, either on the left-hand or the right-hand side. Checking for the existence of a logical difference is thus equivalent to searching for so-called *left-* and *right-hand witnesses*. In [4] distinct methods based on semantic notions are employed for each type of witness. The search for left-hand witnesses is performed by checking for simulations between canonical models, whereas two different approaches were suggested for right-hand witnesses: one is based on instance checking and the second one employs dynamic programming.

In this paper we develop an alternative approach for finding witnesses based on checking for the existence of certain simulations between hypergraph representations of ontologies. The detection of witnesses is performed by checking for the existence of *forward* and *backward simulations*. The existence of such simulations between hypergraphs characterises the fact that the corresponding ontologies cannot be distinguished from each other with conjunctive queries, i.e. no logical difference exists. Our approach is unifying in the sense that the existence of both types of witnesses can be characterised via graph-theoretic notions. We focus on backward simulations only as checking for forward simulations is similar to checking for simulations between canonical models [4].

The paper is organised as follows. After some preliminaries, we introduce a hypergraph representation of $\mathcal{ELH}^r$-terminologies and the notion of a backward simulation in hypergraphs. We show that the existence of backward simulations corresponds to the absence of right-hand witnesses. A prototype implementation of an algorithm that checks for the existence of both types of simulations demonstrates that witnesses can typically be found at least as quickly as with the previous tool CEX 2.5 [5]. Our prototype implementation, however, can handle large cyclic terminologies, which was not possible with CEX 2.5.

This paper uses results from [4, 6] and it extends the previously introduced approach on the concept subsumption difference between terminologies using hypergraphs but restricted to the logic $\mathcal{EL}$ [3].

2 Preliminaries

We start by briefly reviewing the description logic $\mathcal{EL}$ and its extensions $\mathcal{EL}^{\mathsf{ran}}$, $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$ with range restrictions, conjunctions of roles, and the universal role, as well as concept subsumptions based on $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$ and $\mathcal{EL}^{\mathsf{ran}}$. For a more detailed introduction to description logics, we refer to [2].

Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be countably infinite and disjoint sets of concept names and role names. The sets of $\mathcal{EL}$-*concepts* C, $\mathcal{EL}^{\mathsf{ran}}$-*concepts* D, $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$-*concepts* E and the sets of $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$-*inclusions* α and $\mathcal{EL}^{\mathsf{ran}}$-*inclusions* β are built according to the following grammar rules:

$$\begin{aligned}
C &::= \top \mid A \mid C \sqcap C \mid \exists r.C \\
D &::= \top \mid A \mid D \sqcap D \mid \exists r.D \mid \mathsf{ran}(r) \\
E &::= \top \mid A \mid E \sqcap E \mid \exists r_1 \sqcap \ldots \sqcap r_n.E \mid \exists u.E \\
\alpha &::= D \sqsubseteq E \mid r \sqsubseteq s \\
\beta &::= D \sqsubseteq C \mid r \sqsubseteq s
\end{aligned}$$

where $A \in \mathsf{N_C}$, $r, r_1, \ldots, r_n, s \in \mathsf{N_R}$, $n \geq 1$, and $u \notin \mathsf{N_R}$ is the universal role. Concept inclusions of the form $\mathsf{ran}(r) \sqsubseteq D$ are also called *range restrictions*, and those of the form $\mathsf{dom}(r) \sqsubseteq D$ are termed *domain restrictions*, where $\mathsf{dom}(r)$ stands for $\exists r.\top$. A *TBox* is a finite set of inclusions, which are also called *axioms*.

An $\mathcal{ELH}^r$-*terminology* $\mathcal{T}$ is a TBox in which every axiom is of the form $A \sqsubseteq C$, $A \equiv C$, $r \sqsubseteq s$, $\mathsf{ran}(r) \sqsubseteq C$ or $\mathsf{dom}(r) \sqsubseteq C$, where A is a concept name, C an $\mathcal{EL}$-concept and no concept name occurs more than once on the left-hand side of an axiom.[1]

The semantics is defined using interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where the domain $\Delta^{\mathcal{I}}$ is a non-empty set, and $\cdot^{\mathcal{I}}$ is a function mapping each concept name A to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, every role name r to a binary relation $r^{\mathcal{I}}$ over $\Delta^{\mathcal{I}}$, and $u^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The *extension* $C^{\mathcal{I}}$ of a concept C is defined inductively as: $(\top)^{\mathcal{I}} := \Delta^{\mathcal{I}}$, $(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$, $(\exists r_1 \sqcap \ldots \sqcap r_n.C)^{\mathcal{I}} := \{\, x \in \Delta^{\mathcal{I}} \mid \exists\, y \in C^{\mathcal{I}} : (x, y) \in \bigcap_{i=1}^{n} r_i^{\mathcal{I}} \,\}$, and $(\mathsf{ran}(r))^{\mathcal{I}} := \{y \in \Delta^{\mathcal{I}} \mid \exists x : (x, y) \in r^{\mathcal{I}}\}$.

An interpretation $\mathcal{I}$ *satisfies* a concept C, an inclusion $C \sqsubseteq D$ or $r \sqsubseteq s$ if, respectively, $C^{\mathcal{I}} \neq \emptyset$, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, or $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$. We write $\mathcal{I} \models \varphi$ if $\mathcal{I}$ satisfies the axiom φ. An interpretation $\mathcal{I}$ *satisfies* a TBox $\mathcal{T}$ if $\mathcal{I}$ satisfies all axioms in $\mathcal{T}$; in this case, we say that $\mathcal{I}$ is a *model* of $\mathcal{T}$. An axiom φ *follows* from a TBox $\mathcal{T}$, written $\mathcal{T} \models \varphi$, if for all models $\mathcal{I}$ of $\mathcal{T}$, we have that $\mathcal{I} \models \varphi$.

To simplify the presentation we assume that terminologies do not contain axioms of the form $A \equiv B$ or $A \equiv \top$ (after having removed multiple B or $\top$-conjuncts) for concept names A and B. A terminology is *acyclic* if it can

[1] A concept equation $A \equiv C$ stands for the inclusions $A \sqsubseteq C$ and $C \sqsubseteq A$.

be unfolded (i.e., the process of substituting concept names by their definitions terminates). An $\mathcal{ELH}^r$-terminology $\mathcal{T}$ is *normalised* iff it only contains axioms of the forms $r \sqsubseteq s$,

- $\varphi \sqsubseteq B_1 \sqcap \ldots \sqcap B_n$, $A \sqsubseteq \exists r.B$, $A \sqsubseteq \mathsf{dom}(r)$, and
- $A \equiv B_1 \sqcap \ldots \sqcap B_m$, $A \equiv \exists r.B$,

where $\varphi \in \{A, \mathsf{dom}(s), \mathsf{ran}(s)\}$, $n \geq 1$, $m \geq 2$, A, B, B_i are concept names, r, s roles names, and each conjunct B_i is non-conjunctive in $\mathcal{T}$, i.e. there does not exist an axiom of the form $B_i \equiv B'_1 \sqcap \ldots \sqcap B'_m \in \mathcal{T}$ for concept names $B'_1, \ldots, B'_m$ with $m \geq 2$ (otherwise B_i is said to be *conjunctive* in $\mathcal{T}$).

Every $\mathcal{ELH}^r$-terminology $\mathcal{T}$ can be normalised in polynomial time such that the resulting terminology is a conservative extension of $\mathcal{T}$ [4]. Note that axioms of the form $A \equiv \exists r.\top$ are rewritten into $A \sqsubseteq \mathsf{dom}(r)$ and $\mathsf{dom}(r) \sqsubseteq A$.

A signature Σ is a finite set of symbols from $\mathsf{N_C}$ and $\mathsf{N_R}$. The signature $\mathsf{sig}(\varphi)$ of a syntactic object φ is the set of concept and role names occurring in φ. Note that $\mathsf{sig}(\exists u.C) = \mathsf{sig}(C)$ for every concept C. The symbol Σ is used as a subscript to a set of concepts or inclusions to denote that its elements only use symbols from Σ, e.g., $\mathcal{EL}_\Sigma$, $\mathcal{EL}^{\mathsf{ran}}_\Sigma$, $\mathcal{ELH}^r_\Sigma$, etc.

The logical difference between two terminologies for $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$-inclusions as query language is defined as follows [4,6].

Definition 1 (Concept Inclusion Difference). *The* $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$-concept inclusion difference *between* $\mathcal{ELH}^r$*-terminologies* $\mathcal{T}_1$ *and* $\mathcal{T}_2$ *w.r.t. a signature* Σ *is the set* $\mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ *of all* $\mathcal{EL}^{\mathsf{ran},\sqcap,u}_\Sigma$*-inclusions* φ *such that* $\mathcal{T}_1 \models \varphi$ *and* $\mathcal{T}_2 \not\models \varphi$.

If the set $\mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ is not empty, then it typically contains infinitely many concept inclusions. We make use of the *primitive witnesses theorems* from [4], which state that it is sufficient to consider the following inclusions to represent a difference: (δ_1) $r \sqsubseteq s$, (δ_2) $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$-inclusions of the forms $A \sqsubseteq E$, $\mathsf{dom}(r) \sqsubseteq E$ and $\mathsf{ran}(r) \sqsubseteq E$, and ($\delta_3$) $\mathcal{EL}^{\mathsf{ran}}$-inclusions the form $D \sqsubseteq A$.

The set of all $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$-concept inclusion difference witnesses is defined as

$$\mathsf{Wtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = (\mathsf{roleWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2), \mathsf{lhsWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2), \mathsf{rhsWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)),$$

where the set $\mathsf{roleWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ consists of all role inclusions in $\mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, and the sets $\mathsf{lhsWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) \subseteq (\mathsf{N_C} \cap \Sigma) \cup \{\, \mathsf{dom}(r) \mid r \in \Sigma \,\} \cup \{\, \mathsf{ran}(r) \mid r \in \Sigma \,\}$ and $\mathsf{rhsWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) \subseteq \mathsf{N_C} \cap \Sigma$ of *left-hand* and *right-hand concept difference witnesses* consist of the left-hand sides of the type-δ_2 inclusions in $\mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ and the right-hand sides of type-δ_3 inclusions in $\mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, respectively. Observe that the set $\mathsf{Wtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ is finite. Consequently, it can be seen as a succinct representation of the typically infinite set $\mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ in the sense that: $\mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$ iff $\mathsf{Wtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = (\emptyset, \emptyset, \emptyset)$ [4]. Thus, to decide the existence of concept inclusion differences, it is equivalent to decide non-emptiness of the three witness sets.

In this paper, we focus on right-hand witnesses in $\mathsf{rhsWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, i.e., only the inclusions of types δ_3 are relevant.

3 Logical Difference using Hypergraphs

Our approach for detecting logical differences is based on finding appropriate simulations between the hypergraph representations of terminologies. The hypergraph notion in this paper is such that the existence of certain simulations between the ontology hypergraphs of terminologies $\mathcal{T}_1$ and $\mathcal{T}_2$ coincides with $\mathsf{lhsWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$ and $\mathsf{rhsWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$. For every concept name $A \in \Sigma$ (or role name $r \in \Sigma$), we verify whether A (or $\mathsf{dom}(r), \mathsf{ran}(r)$) belongs to $\mathsf{lhsWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, or whether A is a member of $\mathsf{rhsWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$. For the former, we check for the existence of a *forward simulation*, and for the latter, for the existence of a *backward simulation* between the ontology hypergraphs of $\mathcal{T}_1$ and $\mathcal{T}_2$. In this paper we present backward simulations only. Checking for left-hand side witnesses and for witnesses in $\mathsf{roleWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ using ontology hypergraphs can be done similarly to [4].

We start with defining ontology hypergraphs and we subsequently introduce the notion of a backward simulation between such hypergraphs.

3.1 Ontology Hypergraphs

We introduce a hypergraph representation of $\mathcal{ELH}^r$-terminologies. A *directed hypergraph* is a tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a non-empty set of *nodes* (or *vertices*), and $\mathcal{E}$ is a set of *directed hyperedges* of the form $e = (S, S')$, where $S, S' \subseteq \mathcal{V}$. A node $v \in \mathcal{V}$ is *reachable* in $\mathcal{G}$ from a set of nodes $\mathcal{V}' \subseteq \mathcal{V}$ (written $\mathcal{V}' \geq_\mathcal{G} v$) if $v \in \mathcal{V}'$, or there is a hyperedge $e = (S, S') \in \mathcal{E}$ such that $v \in S'$ and every node in S is reachable from $\mathcal{V}'$.

We now show how to represent terminologies as hypergraphs.

Definition 2 (Ontology Hypergraph). *Let $\mathcal{T}$ be a normalised $\mathcal{ELH}^r$-terminology and let Σ be a signature. The* ontology hypergraph $\mathcal{G}^\Sigma_\mathcal{T}$ of $\mathcal{T}$ for Σ *is a directed hypergraph $\mathcal{G}^\Sigma_\mathcal{T} = (\mathcal{V}, \mathcal{E})$ defined as follows:*

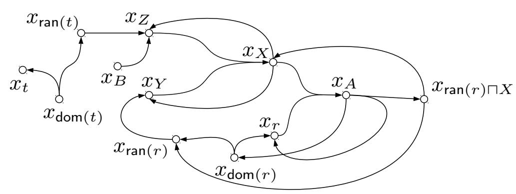
$$
\begin{aligned}
\mathcal{V} =& \{\, x_A \mid A \in \mathsf{N_C} \cap (\Sigma \cup \mathsf{sig}(\mathcal{T}))\,\}\\
&\cup \{\, x_r, x_{\mathsf{dom}(r)}, x_{\mathsf{ran}(r)} \mid r \in \mathsf{N_R} \cap (\Sigma \cup \mathsf{sig}(\mathcal{T}))\,\}, \textit{and}\\
\mathcal{E} =& \{(\{x_\ell\}, \{x_{B_i}\}) \mid \ell \bowtie B_1 \sqcap \ldots \sqcap B_n \in \mathcal{T}, \bowtie \in \{\sqsubseteq, \equiv\},\\
&\qquad \ell \in \mathsf{N_C} \cup \{\, \mathsf{dom}(s), \mathsf{ran}(s) \mid s \in \mathsf{N_R}\,\}\}\\
&\cup \{(\{x_A\}, \{x_{\mathsf{dom}(r)}\}) \mid A \sqsubseteq \mathsf{dom}(r) \in \mathcal{T} \textit{ or } A \bowtie \exists r.B \in \mathcal{T}, \bowtie \in \{\sqsubseteq, \equiv\}\}\\
&\cup \{(\{x_A\}, \{x_r, x_{\mathsf{ran}(r) \sqcap B}\}), (\{x_{\mathsf{ran}(r) \sqcap B}\}, \{x_B\}),\\
&\qquad (\{x_{\mathsf{ran}(r) \sqcap B}\}, \{x_{\mathsf{ran}(r)}\}) \mid A \bowtie \exists r.B \in \mathcal{T}, \bowtie \in \{\sqsubseteq, \equiv\}\}\\
&\cup \{\, (\{x_r, x_B\}, \{x_A\}) \mid A \equiv \exists r.B \in \mathcal{T}\,\}\\
&\cup \{\, (\{x_{B_1}, \ldots, x_{B_n}\}, \{x_A\}) \mid A \equiv B_1 \sqcap \ldots \sqcap B_n \in \mathcal{T}\,\}\\
&\cup \{\, (\{x_{\mathsf{dom}(r)}\}, \{x_r, x_{\mathsf{ran}(r)}\}) \mid r \in \mathsf{N_R} \cap (\Sigma \cup \mathsf{sig}(\mathcal{T})\,\}\\
&\cup \{(\{x_r\}, \{x_s\}), (\{x_{\mathsf{dom}(r)}\}, \{x_{\mathsf{dom}(s)}\}), (\{x_{\mathsf{ran}(r)}\}, \{x_{\mathsf{ran}(s)}\}) \mid r \sqsubseteq s \in \mathcal{T}\}
\end{aligned}
$$

The ontology hypergraph $\mathcal{G}^\Sigma_\mathcal{T} = (\mathcal{V}, \mathcal{E})$ contains a node x_ℓ for every signature symbol ℓ in Σ and $\mathcal{T}$.[2] Additionally, we represent concepts of the form $\mathsf{dom}(r)$

[2] Note that, differently to [3], the graph $\mathcal{G}^\Sigma_\mathcal{T}$ does not contain a node representing $\top$.

and $\mathsf{ran}(r)$ as nodes in the graph. We include a hyperedge $(\{x_{\mathsf{dom}(r)}\}, \{x_r, x_{\mathsf{ran}(r)}\})$ for every role name r in Σ and $\mathcal{T}$, which corresponds to the tautology $\mathsf{dom}(r) \sqsubseteq \exists r.\mathsf{ran}(r)$.[3] Recall that $\mathsf{dom}(r)$ equals $\exists r.\top$. The other hyperedges in $\mathcal{G}_{\mathcal{T}}^{\Sigma}$ represent the axioms in $\mathcal{T}$. Every hyperedge is directed and can be understood as an implication, i.e., $(\{x_{\ell_1}\}, \{x_{\ell_2}\})$ stands for $\mathcal{T} \models \ell_1 \sqsubseteq \ell_2$. The complex hyperedges are of the form $(\{x_A\}, \{x_r, x_B\})$ and $(\{x_r, x_B\}, \{x_A\})$ representing $\mathcal{T} \models A \sqsubseteq \exists r.B$ and $\mathcal{T} \models \exists r.B \sqsubseteq A$, and of the form $(\{x_{B_1}, ..., x_{B_n}\}, \{x_A\})$ representing $\mathcal{T} \models B_1 \sqcap \ldots \sqcap B_n \sqsubseteq A$.

Example 1. Let $\mathcal{T} = \{A \equiv \exists r.X,\ X \equiv Y \sqcap Z,\ B \sqsubseteq Z,\ \mathsf{ran}(r) \sqsubseteq Y\}$, and $\Sigma = \{A, B, r\}$. The ontology hypergraph $\mathcal{G}_{\mathcal{T}}^{\Sigma}$ of $\mathcal{T}$ for Σ can be depicted as follows.



In the following we introduce a relation $\to_{\mathcal{T}}$ on nodes $x_l, x_{l'}$ of an ontology hypergraph such that $x_l \to_{\mathcal{T}} x_{l'}$ holds iff $\mathcal{T} \models l \sqsubseteq l'$.

Definition 3. *Let $\mathcal{G}_{\mathcal{T}}^{\Sigma} = (\mathcal{V}, \mathcal{E})$ be the ontology hypergraph of a normalised $\mathcal{ELH}^r$-terminology $\mathcal{T}$ for a signature Σ. We define $\to_{\mathcal{T}} \subseteq \mathcal{V} \times \mathcal{V}$ to be the smallest reflexive and transitive relation closed under the following conditions:*

- $x \to_{\mathcal{T}} x'$ *if* $(\{x\}, \{x'\}) \in \mathcal{E}$;
- $x \to_{\mathcal{T}} x'$ *if* $(\{x\}, \{x_r, x_A\}) \in \mathcal{E}$, $(\{x_s, x_B\}, \{x'\}) \in \mathcal{E}$, $x_r \to_{\mathcal{T}} x_s$, *and* $x_A \to_{\mathcal{T}} x_B$;
- $x \to_{\mathcal{T}} x'$ *if* $(\{x_{B_1}, \ldots, x_{B_m}\}, \{x'\}) \in \mathcal{E}$ *and* $x \to_{\mathcal{T}} x_{B_i}$ *for every* $1 \leq i \leq m$.

It can be readily seen that the relation $\to_{\mathcal{T}}$ can be computed in polynomial time in the size of $\mathcal{T}$. Note that $\to_{\mathcal{T}}$ does not coincide with the usual reachability notion in a hypergraph.

3.2 Backward Simulation

We introduce backward simulations between ontology hypergraphs whose existence coincides with the absence of right-hand witnesses. The simulations are defined such that a node x_A in $\mathcal{G}_{\mathcal{T}_1}^{\Sigma}$ is simulated by a node $x_{A'}$ in $\mathcal{G}_{\mathcal{T}_2}^{\Sigma}$ iff A' is entailed in $\mathcal{T}_2$ by exactly the same $\mathcal{EL}_{\Sigma}^{\mathsf{ran}}$-concepts that entail A in $\mathcal{T}_1$. To define backward simulations we need to take all the axioms into account that cause Σ-concepts to entail a concept name. Axioms of the forms $A \equiv \exists r.B$, $A \equiv B_1 \sqcap \ldots \sqcap B_n$, and $\mathsf{ran}(r) \sqsubseteq A$ require special treatment, while it is more straightforward to deal with the other types of axioms. For the former, consider $\mathcal{T}_1 = \{A \equiv \exists r.X\}$, $\mathcal{T}_2 = \{A \sqsubseteq \exists r.\top\}$, and $\Sigma = \{A, r\}$. It holds that

[3] These hyperedges are relevant for the forward simulation only.

$\mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$. Observe that there does not exist an $\mathcal{EL}^{\mathsf{ran}}_\Sigma$-concept that entails A in $\mathcal{T}_1$ as the concept name X is not entailed by any Σ-concept in $\mathcal{T}$. Thus, the node x_A in $\mathcal{G}^\Sigma_{\mathcal{T}_1}$ should be simulated by the node x_A in $\mathcal{G}^\Sigma_{\mathcal{T}_2}$. To handle such cases, we want to characterise the entailment by an $\mathcal{EL}^{\mathsf{ran}}_\Sigma$-concept in terms of (a special) reachability notion in ontology hypergraphs.

For a signature Σ, let $\Sigma^{\mathsf{Ran}} = \Sigma \cup \Sigma^{\mathsf{dom}} \cup \Sigma^{\mathsf{ran}}$, where $\Sigma^{\mathsf{dom}} = \{\, \mathsf{dom}(t) \mid t \in \mathsf{N_R} \cap \Sigma \,\}$ and $\Sigma^{\mathsf{ran}} = \{\, \mathsf{ran}(t) \mid t \in \mathsf{N_R} \cap \Sigma \,\}$ are the sets consisting of concepts of the form $\mathsf{dom}(t)$ and $\mathsf{ran}(t)$ for every role name t in Σ, respectively.

Definition 4 (Σ^{Ran}-Reachability). *Let $\mathcal{G}^\Sigma_\mathcal{T} = (\mathcal{V}, \mathcal{E})$ be the ontology hypergraph of a normalised $\mathcal{ELH}^r$-terminology $\mathcal{T}$ for a signature Σ. We set $\mathcal{V}^\Sigma_{\mathsf{Ran}} = \{\, x \in \mathcal{V} \mid x_\sigma \to_\mathcal{T} x$ for some $\sigma \in \Sigma^{\mathsf{Ran}} \,\}$ to be the set of nodes in $\mathcal{G}^\Sigma_\mathcal{T}$ that are reachable via $\to_\mathcal{T}$ from a node labelled with elements from Σ^{Ran}. We say that a node $v \in \mathcal{V}$ is* Σ^{Ran}-reachable in $\mathcal{G}^\Sigma_\mathcal{T}$ *iff $\mathcal{V}^\Sigma_{\mathsf{Ran}} \geq_{\mathcal{G}^\Sigma_\mathcal{T}} v$.*

It can readily be seen that all Σ^{Ran}-reachable nodes can be identified in polynomial time w.r.t. the size of $\mathcal{T}$.

Example 2. Let $\mathcal{T} = \{A \equiv \exists r.X,\ X \equiv Y \sqcap Z,\ B \sqsubseteq Z,\ \mathsf{ran}(r) \sqsubseteq Y\}$ (cf. Ex. 1) and let $\Sigma = \{B, r\}$. Then all the nodes are Σ^{Ran}-reachable in $\mathcal{G}^\Sigma_\mathcal{T}$ and we have that $\mathcal{T} \models \exists r.B \sqsubseteq A$.

The following example demonstrates that the relation $\to_\mathcal{T}$ is not sufficient to characterise entailment by $\mathcal{EL}^{\mathsf{ran}}_\Sigma$-concepts in every case.

Example 3. Let $\mathcal{T} = \{A \equiv \exists r.X,\ X \equiv \exists r.B\}$ and let $\Sigma = \{A, B, r\}$. The nodes x_A and x_B are Σ^{Ran}-reachable in $\mathcal{G}^\Sigma_\mathcal{T}$, $\mathcal{T} \models \exists r.\exists r.B \sqsubseteq A$, but $x_B \not\to_\mathcal{T} x_A$.

We now state the properties of Σ^{Ran}-reachable nodes that we obtain.

Lemma 1. *Let $\mathcal{T}$ be a normalised $\mathcal{ELH}^r$-terminology and let Σ be a signature. Then the following statements hold:*

(i) $x_A \in \mathcal{V}$ is Σ^{Ran}-reachable in $\mathcal{G}^\Sigma_\mathcal{T}$ iff there is an $\mathcal{EL}^{\mathsf{ran}}_\Sigma$-concept D such that $\mathcal{T} \models D \sqsubseteq A$;
(ii) $x_s \in \mathcal{V}$ is Σ^{Ran}-reachable in $\mathcal{G}^\Sigma_\mathcal{T}$ iff there is $s' \in \mathsf{N_R} \cap \Sigma$ such that $\mathcal{T} \models s' \sqsubseteq s$.

For axioms of the form $A \equiv B_1 \sqcap \ldots \sqcap B_n$, we introduce the following notion which associates with every node x_A in a hypergraph $\mathcal{G}_\mathcal{T}$ a set of concept names non-conj(x_A) that are essential to entail A in $\mathcal{T}$ (cf. [4]).

Definition 5 (Non-Conjunctive). *Let $\mathcal{G}^\Sigma_\mathcal{T} = (\mathcal{V}, \mathcal{E})$ be the ontology hypergraph of a normalised $\mathcal{ELH}^r$-terminology $\mathcal{T}$ for a signature Σ. For $x_A \in \mathcal{V}$, let non-conj(x_A) be defined as:*

- *if $(\{x_{B_1}, \ldots, x_{B_m}\}, \{x_A\}) \in \mathcal{E}$ with $m \geq 2$ (i.e. $A \equiv B_1 \sqcap \ldots \sqcap B_m \in \mathcal{T}$), we define*

$$\textit{non-conj}_\mathcal{T}(x_A) = \{x_{B_1}, \ldots, x_{B_m}\};$$

- *otherwise, let non-conj$_\mathcal{T}(x_A) = \{x_A\}$.*

Note that for any concept name A in a normalised $\mathcal{ELH}^r$-terminology $\mathcal{T}$ the concept names $B_1, \ldots, B_m$ with $\text{non-conj}_{\mathcal{T}}(x_A) = \{x_{B_1}, \ldots, x_{B_m}\}$ are non-conjunctive in $\mathcal{T}$.

We need to take special care of axioms of the form $\mathsf{ran}(r) \sqsubseteq X$ as they might cause non-obvious entailments. Let $\mathcal{T} = \{X \equiv B_1 \sqcap B_2,\ A \equiv \exists r.X\}$ and $\Sigma = \{A, B_1, B_2, r\}$. Then the Σ-concept $\exists r.(B_1 \sqcap B_2)$ entails A in $\mathcal{T}$. If we add the axiom $\mathsf{ran}(r) \sqsubseteq B_1$ to $\mathcal{T}$, then already the Σ-concept $\exists r.B_2$ (of smaller signature) is sufficient to entail A in $\mathcal{T}$. Intuitively, the conjunct B_1 of X is already covered by $\mathsf{ran}(r)$ in the presence of the axiom $\mathsf{ran}(r) \sqsubseteq B_1$ (as $\mathcal{T} \models \mathsf{ran}(r) \sqcap B_2 \sqsubseteq X$). To define backward simulations for axioms of the form $A \equiv \exists r.X$, all axioms of the form $\mathsf{ran}(r) \sqsubseteq Y$ need to be taken into account.

We will therefore define the notion of backward simulation using an additional parameter $\zeta \in \mathcal{C}^\Sigma = \{\epsilon\} \cup (\mathsf{N_R} \cap \Sigma)$, called the *role context*. Such a parameter ζ stands for an expression of the form $\mathsf{ran}(\zeta)$ in which a node $x \in \mathcal{V}_1$ should be simulated by a node $x' \in \mathcal{V}_2$. We treat ϵ as a special role name and set $\mathsf{ran}(\epsilon) = \top$.

Additionally, we say that a node $y \in \mathcal{V}$ in an ontology hypergraph $\mathcal{G}^\Sigma_\mathcal{T} = (\mathcal{V}, \mathcal{E})$ is *relevant for a node x in $\mathcal{T}$ w.r.t. a set of node labels $\mathcal{L}$* used in $\mathcal{G}^\Sigma_\mathcal{T}$ if $y \in \text{non-conj}_{\mathcal{T}}(x)$ and $x_\ell \not\to_{\mathcal{T}} y$ for every $\ell \in \mathcal{L}$.

We now give the definition of *backward simulations* as subsets of $\mathcal{V}_1 \times \mathcal{V}_2 \times \mathcal{C}^\Sigma$.

Definition 6 (Backward Simulation). *Let $\mathcal{G}^\Sigma_{\mathcal{T}_1} = (\mathcal{V}_1, \mathcal{E}_1)$, $\mathcal{G}^\Sigma_{\mathcal{T}_2} = (\mathcal{V}_2, \mathcal{E}_2)$ be the ontology hypergraphs of normalised $\mathcal{ELH}^r$-terminologies $\mathcal{T}_1$ and $\mathcal{T}_2$ for a signature Σ. A relation $\hookleftarrow\ \subseteq \mathcal{V}_1 \times \mathcal{V}_2 \times \mathcal{C}^\Sigma$ is a* backward Σ-simulation *if the following conditions are satisfied:*

- *(i_b) if $(x, x', \zeta) \in\ \hookleftarrow$, then for every $\sigma \in \Sigma^{\mathsf{Ran}}$: $x_\sigma \to_{\mathcal{T}_1} x$ implies $x_\sigma \to_{\mathcal{T}_2} y'$ for every $y' \in \mathcal{V}_2$ relevant for x' in $\mathcal{T}_2$ w.r.t. $\{\mathsf{ran}(\zeta)\}$;*
- *(ii_b) if $(x, x', \zeta) \in\ \hookleftarrow$, $(\{x_r, y\}, \{x\}) \in \mathcal{E}_1$, and y is Σ^{Ran}-reachable in $\mathcal{G}^\Sigma_{\mathcal{T}_1}$, then for every $s \in \Sigma$ such that $x_s \to_{\mathcal{T}_1} x_r$, and for every $y' \in \mathcal{V}_2$ relevant for x' in $\mathcal{T}_2$ w.r.t. $\{\mathsf{ran}(\zeta), \mathsf{dom}(s)\}$, there exists $(\{x_{r'}, z'\}, \{y'\}) \in \mathcal{E}_2$ with $x_s \to_{\mathcal{T}_2} x_{r'}$ and $(y, z', s) \in\ \hookleftarrow$;*
- *(iii_b) if $(x, x', \zeta) \in\ \hookleftarrow$ and $(\{x_{X_1}, \ldots, x_{X_n}\}, \{x\}) \in \mathcal{E}_1$, then for every $y' \in \mathcal{V}_2$ relevant for x' w.r.t. $\{\mathsf{ran}(\zeta)\}$ there exists $y \in \mathcal{V}_1$ relevant for x in $\mathcal{T}_1$ w.r.t. $\{\mathsf{ran}(\zeta)\}$ with $(y, y', \epsilon) \in\ \hookleftarrow$.*

We write $\mathcal{G}^\Sigma_{\mathcal{T}_1} \hookleftarrow \mathcal{G}^\Sigma_{\mathcal{T}_2}$ iff there exists a backward Σ-simulation $\hookleftarrow\ \subseteq \mathcal{V}_1 \times \mathcal{V}_2 \times \mathcal{C}^\Sigma$ such that $(x_A, x_A, \epsilon) \in\ \hookleftarrow$ for every $A \in \mathsf{N_C} \cap \Sigma$.

Members of a backward simulation $\hookleftarrow$ are called *simulation triples*.

For a node x in $\mathcal{G}^\Sigma_{\mathcal{T}_1}$ to be backward simulated by x' in $\mathcal{G}^\Sigma_{\mathcal{T}_2}$, Condition ($i_b$) enforces that appropriate Σ-concept names B or concepts of the form $\mathsf{ran}(s)$, $\mathsf{dom}(s)$ with $s \in \Sigma$ that entail x in $\mathcal{T}_1$ must also entail x' in $\mathcal{T}_2$. Condition (ii_b) applies to nodes $x_A \in \mathcal{G}^\Sigma_{\mathcal{T}_1}$ for which there exists an axiom $A \equiv \exists r.X$ in $\mathcal{T}_1$ and propagates the simulation to the successor node x_X by taking into account possible entailments regarding domain or range restrictions in $\mathcal{T}_2$. Condition (iii_b) handles axioms of the form $A \equiv B_1 \sqcap \ldots \sqcap B_n$ in $\mathcal{T}_1$. We have to match every conjunct y' that is relevant for x' in $\mathcal{T}_2$ with some conjunct y relevant for x in $\mathcal{T}_1$ (possibly leaving some conjuncts y unmatched) since, intuitively speaking, some conjuncts in the definition of A in $\mathcal{T}_1$ can be ignored to preserve logical entailment. For instance, let $\mathcal{T}_1 = \{A \equiv B_1 \sqcap B_2\}$, $\mathcal{T}_2 = \{B_1 \sqsubseteq A\}$ and $\Sigma = \{A, B_1, B_2\}$. Then $\mathsf{rhsWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$ and, in particular, $\mathcal{T}_2 \models B_1 \sqcap B_2 \sqsubseteq A$ holds as well. Note that the simulation between conjuncts is propagated in the context ϵ only as all the conjuncts that are entailed by $\mathsf{ran}(\zeta)$ have been filtered out already.

Example 4. Let $\mathcal{T}_1 = \{A \equiv \exists r.X,\ X \equiv Y \sqcap Z,\ B \sqsubseteq Z,\ \mathsf{ran}(r) \sqsubseteq Y\}$ (cf. Ex. 1), $\mathcal{T}_2 = \{A \equiv X \sqcap Y,\ X \equiv \exists r.B,\ \mathsf{dom}(s) \sqsubseteq Y,\ r \sqsubseteq s\}$, and $\Sigma = \{A, B, r\}$.

It can be readily seen that the nodes x_B, x_Y, x_Z, and x_X are Σ^{Ran}-reachable in $\mathcal{G}^\Sigma_{\mathcal{T}_1}$. As only $x_B \rightarrow_{\mathcal{T}_1} x_B$, we have that the node x_B in $\mathcal{G}^\Sigma_{\mathcal{T}_1}$ can be simulated by the node x_B in $\mathcal{G}^\Sigma_{\mathcal{T}_2}$ in the contexts ϵ and r. Similarly, as only $x_B \rightarrow_{\mathcal{T}_1} x_Z$, the node x_Z in $\mathcal{G}^\Sigma_{\mathcal{T}_1}$ can be simulated by the node x_B in $\mathcal{G}^\Sigma_{\mathcal{T}_2}$ in the contexts ϵ and r. Hence, as $\text{non-conj}_{\mathcal{T}_2}(x_B) = \{x_B\}$ and as x_Z is relevant for x_X in $\mathcal{T}_1$ w.r.t. $\{\mathsf{ran}(r)\}$, we have that x_X in $\mathcal{G}^\Sigma_{\mathcal{T}_1}$ can be simulated by x_B in $\mathcal{G}^\Sigma_{\mathcal{T}_2}$ in the context r. Finally, as $\text{non-conj}_{\mathcal{T}_2}(x_A) = \{x_X, x_Y\}$ and as only x_X is relevant for x_A in $\mathcal{G}^\Sigma_{\mathcal{T}_2}$ (due to $x_{\mathsf{dom}(r)} \rightarrow_{\mathcal{T}_2} x_Y$), we can conclude that the node x_A in $\mathcal{G}^\Sigma_{\mathcal{T}_1}$ can be simulated by x_A in $\mathcal{G}^\Sigma_{\mathcal{T}_2}$ in the contexts ϵ and r. Overall,

$$S = \{\, (x_A, x_A, \zeta) \mid \zeta \in \{\epsilon, r\}\,\} \cup \{\, (x_B, x_B, \zeta) \mid \zeta \in \{\epsilon, r\}\,\} \\ \cup \{\, (x_Z, x_B, \zeta) \mid \zeta \in \{\epsilon, r\}\,\} \cup \{(x_X, x_B, r)\}$$

is a backward Σ-simulation between $\mathcal{G}^\Sigma_{\mathcal{T}_1}$ and $\mathcal{G}^\Sigma_{\mathcal{T}_2}$ such that $(x_A, x_A, \epsilon) \in S$.

Example 5. Let $\mathcal{T}_1$, $\mathcal{T}_2$, and Σ be defined as in Ex. 4. Now let $\mathcal{T}_1' = \mathcal{T}_1 \cup \{\mathsf{ran}(t) \sqsubseteq Z\}$ and $\Sigma' = \Sigma \cup \{t\}$. We observe that $x_{\mathsf{ran}(t)} \rightarrow_{\mathcal{T}_2} x'$ does not hold for any node $x' \in \mathcal{G}^{\Sigma'}_{\mathcal{T}_2}$, i.e., the node x_Z in $\mathcal{G}^{\Sigma'}_{\mathcal{T}_1'}$ cannot be simulated by any node in $\mathcal{G}^{\Sigma'}_{\mathcal{T}_2}$ (in any context) as Condition (i_b) cannot be fulfilled. Hence, the node x_X in $\mathcal{G}^{\Sigma'}_{\mathcal{T}_1'}$ cannot be simulated by x_B in $\mathcal{G}^{\Sigma'}_{\mathcal{T}_1'}$ in the context r as Condition (iii_b) is violated. Thus, there cannot exist a backward Σ-simulation such that x_A in $\mathcal{G}^{\Sigma'}_{\mathcal{T}_1'}$ is simulated by x_A in $\mathcal{G}^{\Sigma'}_{\mathcal{T}_2}$ in the context ϵ as Condition (ii_b) cannot be fulfilled.

We can now show that the existence of a backward simulation coincides with the absence of right-hand witnesses and that one can check in polynomial time whether backward simulations exist.

Theorem 1. *Let $\mathcal{T}_1, \mathcal{T}_2$ be normalised $\mathcal{ELH}^r$-terminologies, and let Σ be a signature. Then it holds that $\mathsf{rhsWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$ iff $\mathcal{G}^\Sigma_{\mathcal{T}_1} \hookleftarrow \mathcal{G}^\Sigma_{\mathcal{T}_2}$.*

Theorem 2. *Let $\mathcal{T}_1$, $\mathcal{T}_2$ be normalised $\mathcal{ELH}^r$-terminologies and let Σ be signature. Then it can be checked in polynomial time whether $\mathcal{G}^\Sigma_{\mathcal{T}_1} \hookleftarrow \mathcal{G}^\Sigma_{\mathcal{T}_2}$ holds.*

So far we focused on finding concept names A contained in $\mathsf{rhsWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, which, together with the sets $\mathsf{roleWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ and $\mathsf{lhsWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, is sufficient to decide the existence of a logical difference between $\mathcal{T}_1$ and $\mathcal{T}_2$. However, in practical applications users may require concrete concept inclusions $D \sqsubseteq A$

$\mathcal{T}_1$	$\mathcal{T}_2$	Time (s) - CEX 2.5	with ex.	Time (s) - Prototype	with ex.
SM09a	SM09b	340.40	580.44	211.08	214.20
SM09b	SM10a	495.22	639.46	295.24	302.53
SM09b	SM09a	477.34	599.07	324.56	328.81
SM10a	SM09b	444.71	608.44	229.15	235.61

Table 1. Experimental Results Obtained for SNOMED CT

(or $A \sqsubseteq E$) in $\mathsf{Diff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ that correspond to a witness A. We note that such *example concept inclusions* (and also example conjunctive queries) can be constructed recursively from triples for which the simulation conditions failed.

4 Experiments

To investigate the practical applicability of our simulation-based approach for detecting right-hand witnesses, we implemented a prototype tool in OCaml that is based on the CEX 2.5 tool [5]. We then conducted a brief experimental evaluation involving large fragments of three versions of SNOMED CT (the first and second international release from 2009 as well as the first international release from 2010) and 119 versions of NCI[4] which appeared between October 2003 and January 2014. The considered fragments of SNOMED CT each contain about 280 000 concepts names and 62 role names. The aim of our experiments was to compare the performance of our prototype implementation against the CEX 2.5 tool, which can detect logical differences between acyclic terminologies only. We instructed both tools to compute the set $\mathsf{Wtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ for various versions $\mathcal{T}_1$ and $\mathcal{T}_2$ of SNOMED CT and NCI. All the experiments were conducted on PCs equipped with an Intel Core i5-2500 CPU running at 3.30GHz, and all the computation times we report on are the average of three executions.

In our experiments involving SNOMED CT we used signatures composed of the intersection of the concept names in the two versions that were compared, together with the same 31 role names (including "RoleGroup") that were chosen at random initially (and which occur in every version). The results that we obtained are shown in Table 1. The first two columns indicate which versions were used as ontologies $\mathcal{T}_1$ and $\mathcal{T}_2$. The next two columns then show the computation times (CPU time) required by CEX 2.5, with column four depicting the computation times if additionally examples illustrating the witnesses were computed. The last two columns then indicate the computation times of our prototype tool. The times required when additionally examples were computed are shown in the last column. One can see that in all the cases our prototype tool required less time to compute difference witnesses (also together with example inclusions) than CEX 2.5.

[4] More precisely, we first extracted the $\mathcal{ELH}^r$-fragment of the NCI versions by removing up to 8% of the axioms which were not expressed in this fragment.

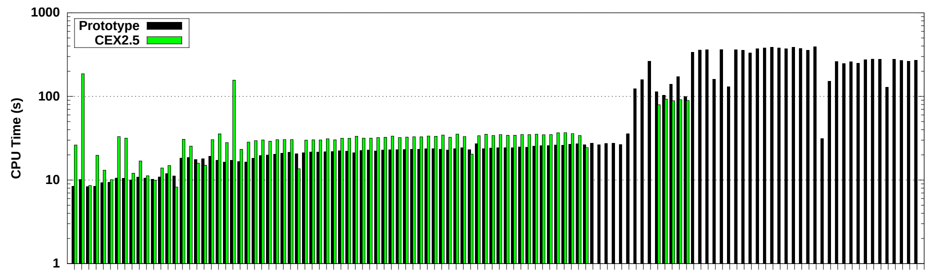


Fig. 1. Experimental Results Obtained for NCI

For each considered version α of NCI, we computed conjunctive query witnesses for $\mathcal{T}_1 = \text{NCI}_\alpha$ and $\mathcal{T}_2 = \text{NCI}_{\alpha+1}$ on signatures $\Sigma = \mathsf{sig}(\text{NCI}_\alpha) \cap \mathsf{sig}(\text{NCI}_{\alpha+1})$, where $\alpha + 1$ denotes the successor version of α, together with corresponding examples. The results that we obtained are depicted in Fig. 1. The computations are sorted chronologically along the x-axis according to the publication date of version NCI_α. Each pair of bars represents the computation times required by our prototype tool and by CEX 2.5, respectively, for one comparison. In the cases where only one bar is shown, the ontology $\mathcal{T}_1 = \text{NCI}_\alpha$ was cyclic and CEX 2.5 could not be used.

Generally speaking, both tools required longer computation times on more recent NCI versions than on older releases, which could be explained by the fact that the size of NCI versions increased with every new release. In the comparisons before version 10.03h our prototype tool could typically compute the witnesses and example inclusions faster than CEX 2.5. However, on later versions our new tool then required slightly longer computation times. One can also see that overall it took the longest time to compute witnesses for cyclic versions of NCI.

Finally, we note that in our experiments all the computations required at most 2.85 GiB of main memory.

5 Conclusion

We presented a unifying approach to solving the logical difference problem for possibly cyclic $\mathcal{ELH}^r$-terminologies. We showed that the existence of *backward simulations* in hypergraph representations of terminologies corresponds to the absence of right-hand witnesses (an analogous correspondence exists between forward simulations and left-hand witnesses). We also demonstrated the applicability of the hypergraph approach using a prototype implementation. The experiments showed that in most cases our prototype tool outperformed the previous tool, CEX 2.5, for computing the logical difference. Moreover, our prototype tool could be successfully applied on fairly large *cyclic* terminologies, whereas previous approaches only worked for acyclic (or rather small cyclic) terminologies.

We plan to further improve our prototype implementation. Moreover, extensions of our techniques to DL-Lite, general $\mathcal{ELH}^r$-TBoxes, or even Horn-$\mathcal{SHIQ}$ ontologies could be investigated.

References

1. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope further. In: Proceedings of OWLED'08 (2008)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The description logic handbook: theory, implementation, and applications. Cambridge University Press (2007)
3. Ecke, A., Ludwig, M., Walther, D.: The concept difference for $\mathcal{EL}$-terminologies using hypergraphs. In: Proceedings of DChanges'13. CEUR-WS.org (2013)
4. Konev, B., Ludwig, M., Walther, D., Wolter, F.: The logical difference for the lightweight description logic $\mathcal{EL}$. JAIR 44, 633–708 (2012)
5. Konev, B., Ludwig, M., Wolter, F.: Logical difference computation with CEX2.5. In: Proceedings of IJCAR'12. pp. 371–377. Springer (2012)
6. Konev, B., Walther, D., Wolter, F.: The logical difference problem for description logic terminologies. In: Proceedings of IJCAR'08. pp. 259–274. Springer (2008)
7. Kontchakov, R., Wolter, F., Zakharyaschev, M.: Logic-based ontology comparison and module extraction, with an application to DL-Lite. Artificial Intelligence 174(15), 1093–1141 (Oct 2010)
8. Lembo, D., Santarelli, V., Savo, D.F.: Graph-based ontology classification in OWL 2 QL. In: Proceedings of ESWC 2013. LNCS, vol. 7882, pp. 320–334. Springer (2013)
9. Nortje, R., Britz, A., Meyer, T.: Module-theoretic properties of reachability modules for SRIQ. In: Proceedings of DL'13. pp. 868–884. CEUR-WS.org (2013)
10. Suntisrivaraporn, B.: Polytime reasoning support for design and maintenance of large-scale biomedical ontologies. Ph.D. thesis, TU Dresden, Germany (2009)