Deciding Subsumers of Least Fixpoint Concepts w.r.t. general $\mathcal{EL}$-TBoxes

Shasha Feng[1], Michel Ludwig[2(✉)], and Dirk Walther[2]

[1] Jilin University, Changchun, China
fengss@jlu.edu.cn
[2] Theoretical Computer Science, TU Dresden, Dresden, Germany
{michel,dirk}@tcs.inf.tu-dresden.de

Abstract. In this paper we provide a procedure for deciding subsumptions of the form $\mathcal{T} \models \mathcal{C} \sqsubseteq E$, where $\mathcal{C}$ is an $\mathcal{ELU}_\mu$-concept, E an $\mathcal{ELU}$-concept and $\mathcal{T}$ a general $\mathcal{EL}$-TBox. Deciding such subsumptions can be used for computing the logical difference between general $\mathcal{EL}$-TBoxes. Our procedure is based on checking for the existence of a certain simulation between hypergraph representations of the set of subsumees of $\mathcal{C}$ and of E w.r.t. $\mathcal{T}$, respectively. With the aim of keeping the procedure implementable, we provide a detailed construction of such hypergraphs deliberately avoiding the use of intricate automata-theoretic techniques.

1 Introduction

Description Logics (DLs) are popular KR languages [3]. Light-weight DLs such as $\mathcal{EL}$ with tractable reasoning problems in particular are commonly used ontology languages [1,2]. The notion of logical difference between TBoxes was introduced as a logic-based approach to ontology versioning [10]. Computing the logical difference between $\mathcal{EL}$-TBoxes can be reduced to fixpoint reasoning w.r.t. TBoxes in a hybrid μ-calculus [8,13]. This involves subsumptions of the form $\mathcal{T} \models \mathcal{C} \sqsubseteq \mathcal{D}$, where $\mathcal{C}$ is an $\mathcal{ELU}_\mu$-concept, i.e. $\mathcal{EL}$-concepts enriched with disjunction and the least fixpoint operator, $\mathcal{D}$ an $\mathcal{EL}_\nu$-concept, i.e. $\mathcal{EL}$-concepts enriched with the greatest fixpoint operator, and $\mathcal{T}$ an $\mathcal{EL}$-TBox. Such subsumptions can be reduced to finding an $\mathcal{ELU}$-concept E such that $\mathcal{T} \models \mathcal{C} \sqsubseteq E$ and $\mathcal{T} \models E \sqsubseteq \mathcal{D}$. Here E acts as an *interpolant* between the fixpoint concepts $\mathcal{C}$ and $\mathcal{D}$ w.r.t. $\mathcal{T}$. In this paper, we only focus on deciding the former type of subsumption, whose decision procedure is arguably more involved than the one for the latter type of subsumption. Deciding the existence of a suitable $\mathcal{ELU}$-concept E, however, will be handled in another paper. Unfortunately, the fact that the required fixpoint reasoning can be solved using automata theoretic techniques does not mean that one can immediately derive a practical algorithm from it [4,9,13]. We therefore aim to develop a procedure that can be implemented more easily by following our hypergraph-based approach to the logical difference problem as introduced in [6]

The second and third authors were supported by the German Research Foundation (DFG) within the Cluster of Excellence ‘Center for Advancing Electronics Dresden’.

S. Hölldobler et al. (Eds.): KI 2015, LNAI 9324, pp. 59–71, 2015.
DOI: 10.1007/978-3-319-24489-1_5

and further extended in [12]. The idea here is to solve the subsumption problem by checking for the existence of a certain simulation between hypergraphs that represent the set of subsumees of $\mathcal{C}$ and of E w.r.t. $\mathcal{T}$, respectively.

We proceed as follows. In the next section we start with reviewing the basic DL $\mathcal{EL}$ together with its extensions with disjunction and the least fixpoint operator. In Section 3 we present a three-step procedure for computing a normal form of $\mathcal{C}$ and of E w.r.t. $\mathcal{T}$ in detail, which is then used in Section 4 to decide the subsumption problem $\mathcal{T} \models \mathcal{C} \sqsubseteq E$. Finally in Section 5, we conclude the paper with a discussion on how the $\mathcal{ELU}$-concept E can be characterised as a most-consequence preserving subsumer, which is a more general version of the notion of least common subsumer [7].

2 Preliminaries

We start by briefly reviewing the lightweight description logic $\mathcal{EL}$ together with the extensions of $\mathcal{EL}$ that we consider in this paper.

Let $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_V}$ be mutually disjoint sets of concept names, role names, and variable names, respectively. We assume these sets to be countably infinite. We typically use A, B to denote concept names, r, s to indicate role names, and x, y to denote variable names.

The sets of *$\mathcal{EL}$-concepts* C, *$\mathcal{ELU}$-concepts* D, *$\mathcal{ELU}_V$-concepts* E, and *$\mathcal{ELU}_\mu$-concepts* $\mathcal{F}$ are built according to the following grammar rules:

$$C ::= \top \mid A \mid C \sqcap C \mid \exists r.C$$
$$D ::= \top \mid A \mid D \sqcap D \mid D \sqcup D \mid \exists r.D$$
$$E ::= \top \mid A \mid E \sqcap E \mid E \sqcup E \mid \exists r.E \mid x$$
$$\mathcal{F} ::= \top \mid A \mid \mathcal{F} \sqcap \mathcal{F} \mid \mathcal{F} \sqcup \mathcal{F} \mid \exists r.\mathcal{F} \mid x \mid \mu x.\mathcal{F}$$

where $A \in \mathsf{N_C}$, $r, s \in \mathsf{N_R}$, and $x \in \mathsf{N_V}$. We use calligraphic letters to denote concepts that may contain a least fixpoint operator. We denote with L the set of all L-concepts, where $L \in \{\mathcal{EL}, \mathcal{ELU}, \mathcal{ELU}_V, \mathcal{ELU}_\mu\}$. For an $\mathcal{ELU}_\mu$-concept $\mathcal{C}$, the set of *free variables* in $\mathcal{C}$, denoted by $\mathrm{FV}(\mathcal{C})$ is defined inductively as follows: $\mathrm{FV}(\top) = \emptyset$, $\mathrm{FV}(A) = \emptyset$, $\mathrm{FV}(\mathcal{D}_1 \sqcap \mathcal{D}_2) = \mathrm{FV}(\mathcal{D}_1) \cup \mathrm{FV}(\mathcal{D}_2)$, $\mathrm{FV}(\mathcal{D}_1 \sqcup \mathcal{D}_2) = \mathrm{FV}(\mathcal{D}_1) \cup \mathrm{FV}(\mathcal{D}_2)$, $\mathrm{FV}(\exists r.\mathcal{D}) = \mathrm{FV}(\mathcal{D})$, $\mathrm{FV}(x) = \{x\}$, $\mathrm{FV}(\mu x.\mathcal{D}) = \mathrm{FV}(\mathcal{D}) \setminus \{x\}$. An $\mathcal{ELU}_\mu$-concept $\mathcal{C}$ is *closed* if $\mathcal{C}$ does not contain free occurrences of variables, i.e. $\mathrm{FV}(\mathcal{C}) = \emptyset$. In the following we assume that every $\mathcal{ELU}_\mu$-concept $\mathcal{C}$ is *well-formed*, i.e. every subconcept of the form $\mu x.\mathcal{D}$ occurring in $\mathcal{C}$ binds a fresh variable x.

An *$\mathcal{EL}$-TBox* $\mathcal{T}$ is a finite set of axioms, where an axiom can be a *concept inclusion* $C \sqsubseteq C'$, or a *concept equation* $C \equiv C'$, for $\mathcal{EL}$-concepts C, C'.

The semantics of $\mathcal{ELU}_\mu$ 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 $\Delta^\mathcal{I}$ and every role name r to a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. Interpretations are extended to concepts using a function $\cdot^{\mathcal{I},\xi}$ that is parameterised by an *assignment function* that maps each variable $x \in \mathsf{N_V}$ to a set $\xi(x) \subseteq \Delta^{\mathcal{I}}$. Given an interpretation $\mathcal{I}$ and an assignment ξ, the extension of an $\mathcal{ELU}_\mu$-concept is defined inductively as follows: $\top^{\mathcal{I},\xi} := \Delta^{\mathcal{I}}$, $x^{\mathcal{I},\xi} := \xi(x)$ for $x \in \mathsf{N_V}$, $(\mathcal{C}_1 \sqcap \mathcal{C}_2)^{\mathcal{I},\xi} := \mathcal{C}_1^{\mathcal{I}} \cap \mathcal{C}_2^{\mathcal{I}}$, $(\exists r.\mathcal{C})^{\mathcal{I},\xi} := \{ x \in \Delta^{\mathcal{I}} \mid \exists y \in \mathcal{C}^{\mathcal{I},\xi} : (x,y) \in r^{\mathcal{I}} \}$, and $(\mu x.\mathcal{C})^{\mathcal{I},\xi} = \bigcap \{ W \subseteq \Delta^{\mathcal{I}} \mid \mathcal{C}^{\mathcal{I},\xi[x \mapsto W]} \subseteq W \}$, where $\xi[x \mapsto W]$ denotes the assignment ξ modified by mapping x to W.

For $\mathcal{ELU}_\mu$-concepts $\mathcal{C}$ and $\mathcal{D}$, an interpretation $\mathcal{I}$ *satisfies* $\mathcal{C}$, an axiom $\mathcal{C} \sqsubseteq \mathcal{D}$ or $\mathcal{C} \equiv \mathcal{D}$ if, respectively, $\mathcal{C}^{\mathcal{I},\xi_\emptyset} \neq \emptyset$, $\mathcal{C}^{\mathcal{I},\xi_\emptyset} \subseteq \mathcal{D}^{\mathcal{I},\xi_\emptyset}$, or $\mathcal{C}^{\mathcal{I},\xi_\emptyset} = \mathcal{D}^{\mathcal{I},\xi_\emptyset}$, where $\xi_\emptyset(x) = \emptyset$ for every $x \in \mathsf{N_V}$. We write $\mathcal{I} \models \alpha$ iff $\mathcal{I}$ satisfies the axiom α. An interpretation $\mathcal{I}$ *satisfies* a TBox $\mathcal{T}$ iff $\mathcal{I}$ satisfies all axioms in $\mathcal{T}$; in this case, we say that $\mathcal{I}$ is a *model* of $\mathcal{T}$. An axiom α *follows* from a TBox $\mathcal{T}$, written $\mathcal{T} \models \alpha$, iff for all models $\mathcal{I}$ of $\mathcal{T}$, we have that $\mathcal{I} \models \alpha$. Deciding whether $\mathcal{T} \models C \sqsubseteq C'$, for two $\mathcal{EL}$-concepts C and C', can be done in polynomial time in the size of $\mathcal{T}$ and C, C' [1,5].

A signature Σ is a finite set of symbols from $\mathsf{N_C}$ and $\mathsf{N_R}$. The signature $\mathsf{sig}(\mathcal{C})$, $\mathsf{sig}(\alpha)$ or $\mathsf{sig}(\mathcal{T})$ of the concept $\mathcal{C}$, axiom α or TBox $\mathcal{T}$ is the set of concept and role names occurring in $\mathcal{C}$, α or $\mathcal{T}$, respectively. Analogously, $\mathrm{sub}(\mathcal{C})$, $\mathrm{sub}(\alpha)$, or $\mathrm{sub}(\mathcal{T})$ denotes the set of subconcepts occurring in $\mathcal{C}$, α or $\mathcal{T}$, respectively.

An $\mathcal{EL}$-TBox $\mathcal{T}$ is *normalised* if it only contains $\mathcal{EL}$-concept inclusions of the forms $\top \sqsubseteq B$, $A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B$, $A \sqsubseteq \exists r.B$, or $\exists r.A \sqsubseteq B$, where $A, A_i, B \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and $n \geq 1$. Every $\mathcal{EL}$-TBox $\mathcal{T}$ can be normalised in polynomial time in the size of $\mathcal{T}$ with a linear increase in the size of the normalised TBox w.r.t. $\mathcal{T}$ such that the resulting TBox is a conservative extension of $\mathcal{T}$ [10].

3 Normal Form Computation

Our aim is to check whether $\mathcal{T} \models \mathcal{C} \sqsubseteq D$ holds with the help of simulations, where $\mathcal{T}$ an $\mathcal{EL}$-TBox, $\mathcal{C}$ is an $\mathcal{ELU}_\mu$-concept, and D an $\mathcal{ELU}$-concept. Simulations are typically used to characterise properties between two graph structures, e.g. behavioural equivalence. To be able to apply simulations to our subsumption problem, we represent the unfoldings of $\mathcal{C}$ and the subsumees of D w.r.t. $\mathcal{T}$ in two separate hypergraphs. Intuitively, in such a hypergraph every node v together with its outgoing hyperedges represents a disjunction of the form $\bigsqcup_{i=1}^{m} A_i \sqcup \bigsqcup_{j=1}^{n} \exists r_j.C_j$ where the A_i and r_j are pairwise different, respectively. A hyperedge e in such a hypergraph is labelled with role names and it connects one source node with several target nodes. A hyperedge $e = (v_0, \{v_1, \ldots, v_\ell\})$ ($\ell \geq 1$) labelled with a role r represents an existential restriction $\exists r.\varphi$ where φ stands for the conjunction of the nodes $v_1, \ldots, v_\ell$. A crucial condition is that for every role r, a node has at most one outgoing hyperedge labelled with r as otherwise our simulation notion is not applicable. In this sense, the hypergraph can be seen to be *deterministic*. To obtain such a deterministic hypergraph, it is necessary to merge disjunctively connected existential restrictions involving the same role name. We therefore design the hypergraphs to be a conjunctive normal form representation of the set of respective subsumees as it becomes immediate to identify the existential restrictions that have to be merged. The hypergraph for an $\mathcal{ELU}$-concept is a tree, whereas the hypergraph for an $\mathcal{ELU}_\mu$-concept, or for an $\mathcal{ELU}$-concept w.r.t. a cyclic TBox, may contain cycles.

A related normal form, later called *automaton normal form* [4], was introduced in [9] for the full modal μ-calculus with the difference that disjunctive normal form was used. In particular it is shown that every modal μ-calculus formula is equivalent to a formula in normal form. The transformation of a formula into automaton normal form is based on an involved, non-trivial construction using parity automata [4,9]. For our purposes, however, it was not immediate how to derive a practical algorithm from such a construction.

Our construction is essentially based on applying the following three equivalences as rewrite rules to transform $\mathcal{ELU}_\mu$-concepts into the desired format.

(i) $\mathcal{C} \sqcup (\mathcal{D}_1 \sqcap \mathcal{D}_2) \equiv (\mathcal{C} \sqcup \mathcal{D}_1) \sqcap (\mathcal{C} \sqcup \mathcal{D}_2)$
(ii) $(\exists r.\mathcal{C}) \sqcup (\exists r.\mathcal{D}) \equiv \exists r.(\mathcal{C} \sqcup \mathcal{D})$
(iii) $\mu x.\mathcal{C} \equiv \mathcal{C}[x \mapsto \mu x.\mathcal{C}]$

Equivalence (i) is used to transform every "existential level" of the $\mathcal{ELU}_\mu$-concept into conjunctive normal form, (ii) to regroup and merge existentials that involve the same role, and (iii) to unfold fixpoint variables. However, due to the unfolding of fixpoints, a straightforward rewriting of concepts using these equivalences may not terminate, and it is not clear how to formulate a termination condition based on the sequence of concept rewritings.

In the following sections we present a detailed construction of our normal form, and show how termination can be ensured. Our procedure consists of the following three steps:

(1) construct a finite labelled tree representing the successive applications of the equivalences (i)–(iii);
(2) transform the tree that was obtained in the previous step into an hypergraph by removing superfluous nodes, introducing hyperedges that represent existential restrictions over conjunctions, and possibly form cycles;
(3) simplify the hypergraph obtained in the previous step by pruning nodes that can safely be removed while preserving equivalence and that our simulation notion (Section 3.3) cannot handle correctly.

Before presenting the three steps, we introduce the following auxiliary notions. An $\mathcal{ELU}$-concept C is said to be *atomic* iff $C = \top$, $C = A \in \mathsf{N_C}$, or $C = \exists r.D$ for some $\mathcal{ELU}$-concept D. We denote with $\mathrm{Atoms}(S)$ the set of atomic concepts from a set S of $\mathcal{ELU}_\mu$-concepts.

Definition 1 (μ-Suppression). *Let $\mathcal{C}$ be an $\mathcal{ELU}_\mu$-concept. We define an $\mathcal{ELU}_V$-concept $\mathcal{C}^\dagger$ inductively as follows: $\top^\dagger := \top$, $A^\dagger := A$ for $A \in \mathsf{N_C}$, $x^\dagger := x$ for $x \in \mathsf{N_V}$, $(\mu x.\mathcal{D})^\dagger := x$, $(\mathcal{C}_1 \sqcap \mathcal{C}_2)^\dagger := (\mathcal{C}_1^\dagger) \sqcap (\mathcal{C}_2^\dagger)$, $(\mathcal{C}_1 \sqcup \mathcal{C}_2)^\dagger := (\mathcal{C}_1^\dagger) \sqcup (\mathcal{C}_2^\dagger)$, and $(\exists r.\mathcal{D})^\dagger := \exists r.(\mathcal{D}^\dagger)$ for $\mathcal{ELU}_\mu$-concepts $\mathcal{C}_1, \mathcal{C}_2, \mathcal{D}$.*

Example 1. Let $\mathcal{C} = (\exists s.\top) \sqcap (\exists r.\mu x.\mathcal{C}_1)$ for $\mathcal{C}_1 = A \sqcup \exists r.\mu y.\mathcal{C}_2$ and $\mathcal{C}_2 = B \sqcup \exists s.y \sqcup x$. Then $\mathcal{C}^\dagger = (\exists s.\top) \sqcap (\exists r.x)$, $(\mathcal{C}_1)^\dagger = A \sqcup \exists r.y$ and $(\mathcal{C}_2)^\dagger = B \sqcup \exists s.y \sqcup x$.

Definition 2 (Variable Expansion Function). *Let $\mathcal{C}$ be a closed $\mathcal{ELU}_\mu$-concept. A* variable expansion function *for $\mathcal{C}$ is a partial function $\xi_\mathcal{C} \colon \mathcal{ELU}_V \to 2^{\mathcal{ELU}_V}$ defined as follows: for every $x \in var(\mathcal{C})$,*

$$\xi_{\mathcal{C}}(x) := \{ \mathcal{D}^\dagger \mid \mu x.\mathcal{D} \in \text{sub}(\mathcal{C}) \}.$$

Note that since $\mathcal{C}$ is well-formed, $\xi_{\mathcal{C}}(x)$ is a singleton set for every $x \in \mathsf{dom}(\xi_{\mathcal{C}})$.

Example 2. Let $\mathcal{C}$ be defined as in Example 1. Then we obtain the following variable expansion function $\xi_{\mathcal{C}}$ for $\mathcal{C}$: $\xi_{\mathcal{C}} = \{x \mapsto A \sqcup \exists r.y, y \mapsto B \sqcup \exists s.y \sqcup x\}$.

Definition 3 (TBox Expansion Function). *Let $\mathcal{T}$ be a normalised $\mathcal{EL}$-TBox and let D be an $\mathcal{ELU}$-concept. A* TBox expansion function for $(D, \mathcal{T})$ *is a partial function $\xi_{(D,\mathcal{T})}\colon \mathcal{ELU}_V \rightarrow 2^{\mathcal{ELU}}$ defined as follows: for every $\varphi \in$* sub$(\mathcal{T}) \cup$ sub(D),

$$\xi_{(D,\mathcal{T})}(\varphi) := \{ \psi \in \text{sub}(\mathcal{T}) \cup \text{sub}(D) \mid \mathcal{T} \models \psi \sqsubseteq \varphi \}.$$

Example 3. Let $\mathcal{T} = \{A \sqsubseteq Z, \exists r.X \sqsubseteq Z, Z \sqsubseteq \exists r.Y, \exists r.Y \sqsubseteq X, B \sqsubseteq Y\}$ and $D = \top$. Then we obtain the following TBox expansion function for $\mathcal{T}$: $\xi_{(D,\mathcal{T})} = \{ \top \mapsto \{\top\}, A \mapsto \{A\}, B \mapsto \{B\}, Y \mapsto \{B, Y\}, Z \mapsto \{A, Z, \exists r.X\}, X \mapsto \{A, X, Z, \exists r.X, \exists r.Y\}, \exists r.X \mapsto \{\exists r.X\}, \exists r.Y \mapsto \{A, Z, \exists r.X, \exists r.Y\} \}$.

3.1 Step 1

We start with a tableau-like procedure to produce a finite labelled tree, called *concept expansion tree*. The tree is iteratively constructed using four expansion rules. For an $\mathcal{ELU}_\mu$-concept $\mathcal{C}$ we start from a root node labelled with the $\mathcal{ELU}_V$-concept $\mathcal{C}^\dagger$ using the variable expansion function $\xi_{\mathcal{C}}$ in the expansion rules, whereas for an $\mathcal{ELU}$-concept D and an $\mathcal{EL}$-TBox $\mathcal{T}$ the root node is labelled with $D^\dagger = D$ and the TBox expansion function $\xi_{(D,\mathcal{T})}$ is used instead. The tree structure and the expansion rules are defined as follows.

Definition 4 (Concept Expansion Tree). *Let $\mathcal{C}$ be a closed $\mathcal{ELU}_\mu$-concept, and let $\xi\colon \mathcal{ELU}_V \rightarrow 2^{\mathcal{ELU}_V}$ be a TBox or variable expansion function. A* concept expansion tree for $\mathcal{C}$ w.r.t. ξ *is a finite labelled tree $T = (\mathcal{V}, \mathcal{E}, \mathcal{L})$, where $\mathcal{V}$ is a finite, non-empty set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges and $\mathcal{L}$ is a labelling function mapping every node $v \in \mathcal{V}$ to a subset $\mathcal{L}(v)$ of* sub$(\mathcal{C}) \cup$ sub$(\mathsf{ran}(\xi))$. *We say that a node $v \in \mathcal{V}$ is* blocked *iff there exists an ancestor v' of v in T such that $\mathcal{L}(v) = \mathcal{L}(v')$.*

A concept expansion tree $T = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ for $\mathcal{C}$ w.r.t. ξ is initialised with a single node v_0 with $\mathcal{L}(v_0) = \{\mathcal{C}^\dagger\}$ and T is expanded using the following rules which are only applied on leaf nodes $v \in \mathcal{V}$ that are not blocked, and the rule (Exists) is only applied when no other rule is applicable.

(Disj) *if $C_1 \sqcup \ldots \sqcup C_n \in \mathcal{L}(v)$ and $\{C_1, \ldots, C_n\} \not\subseteq \mathcal{L}(v)$, add the node v' with $\mathcal{L}(v') = \mathcal{L}(v) \cup \{C_1, \ldots, C_n\}$ as a child of v;*

(Conj) *if $C \sqcap D \in \mathcal{L}(v)$ and $\{C, D\} \cap \mathcal{L}(v) = \emptyset$, add the two nodes v_1, v_2 with $\mathcal{L}(v_1) = \mathcal{L}(v) \cup \{C\}$, $\mathcal{L}(v_2) = \mathcal{L}(v) \cup \{D\}$ as children of v;*

(Expansion) *if $\varphi \in \mathcal{L}(v)$, $\varphi \in \mathsf{dom}(\xi)$, and $\xi(\varphi) \not\subseteq \mathcal{L}(v)$, add the node v' with $\mathcal{L}(v') = \mathcal{L}(v) \cup \xi(\varphi)$ as a child of v;*

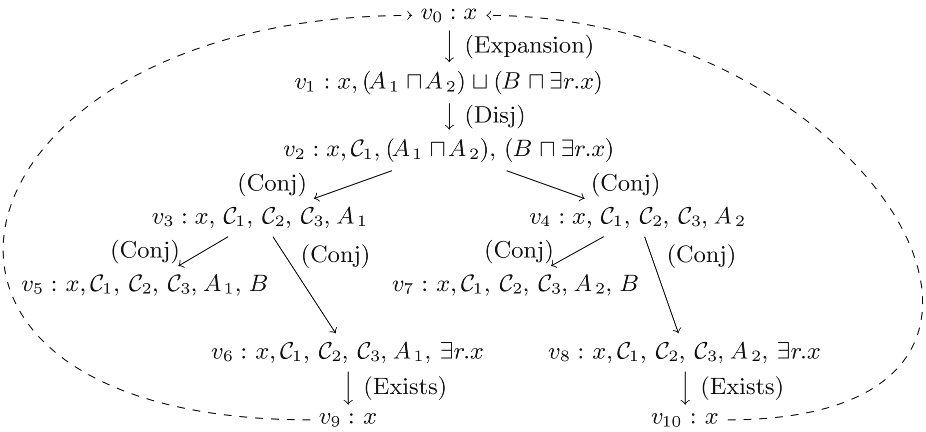


Fig. 1. Fully-expanded concept expansion tree for Example 4

(Exists) *if* $Atoms(\mathcal{L}(v)) = \{A_1, \ldots, A_m\} \cup \{\exists r_1.C_1, \ldots, \exists r_n.C_n\}$, *then for every* $r \in \{r_1, \ldots, r_n\}$, *add the node* v_r *with* $\mathcal{L}(v_r) = \{\, C_i \mid r_i = r, 1 \leq i \leq n \,\}$ *as a child of* v.

A concept tree is said to be fully expanded *iff none of the expansion rules is applicable.*

The rule (Disj) is responsible for splitting disjunctions and adding the disjuncts to the respective node label. The rule (Conj) splits conjunctions and distributes the conjuncts over new successor nodes. The (Expansion) rule handles the expansion of fixpoint variables and TBox entailments. The rule (Exists) adds one (and only one) successor for every role occurring in some existential restriction contained in the node label. In this sense, the resulting expansion tree can be seen to be deterministic. The rule (Exists) ensures that all the subconcepts of existential restrictions over the same role are included disjunctively into the successor node dedicated to that role. For instance, if $\{\exists r.C_1, \exists r.C_2\} \subseteq \mathcal{L}(v)$ for some node v, then the concepts C_1, C_2 will be added to the label of the successor node of v for r. We assume that the rule (Exists) has the least priority among all expansion rules, i.e. (Exists) is only applied when no other rule is applicable. During the expansion process, every rule is applied on leaf nodes that are not blocked. A node is blocked if there exists an ancestor node with the same label.

Example 4. Let $\mathcal{C} = \mu x.\mathcal{C}_1$ where $\mathcal{C}_1 = \mathcal{C}_2 \sqcup \mathcal{C}_3$ and $\mathcal{C}_2 = A_1 \sqcap A_2$, $\mathcal{C}_3 = B \sqcap \exists r.x$.

The fully expanded concept expansion tree $T = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ for $\mathcal{C}$ is shown in Figure 1. The nodes together with their labels are represented in the form '$v : \mathcal{L}(v)$' for $v \in \mathcal{V}$. Edges are represented as arrows between nodes. Arrows are additionally labelled by the expansion rule that was applied. Blocked nodes are indicated using dashed arrows.

$$v_0 : x$$
$$\downarrow \text{(Expansion)}$$
$$v_1 : x, A \sqcup (\exists r^2.x) \sqcup (\exists r^3.x)$$
$$\downarrow \text{(Disj)}$$
$$v_2 : x, A \sqcup (\exists r^2.x) \sqcup (\exists r^3.x), A, (\exists r^2.x), (\exists r^3.x)$$
$$\downarrow \text{(Exists)}$$
$$v_3 : (\exists r.x), (\exists r^2.x)$$
$$\downarrow \text{(Exists)}$$
$$v_4 : x, (\exists r.x)$$
$$\downarrow \text{(Expansion)}$$
$$v_5 : x, (\exists r.x), A \sqcup (\exists r^2.x) \sqcup (\exists r^3.x)$$
$$\downarrow \text{(Disj)}$$
$$v_6 : x, (\exists r.x), A \sqcup (\exists r^2.x) \sqcup (\exists r^3.x), A, (\exists r^2.x), (\exists r^3.x)$$
$$\downarrow \text{(Exists)}$$
$$v_7 : x, (\exists r.x), (\exists r^2.x)$$
$$\downarrow \text{(Expansion)}$$
$$v_8 : x, (\exists r.x), (\exists r^2.x), A \sqcup (\exists r^2.x) \sqcup (\exists r^3.x)$$
$$\downarrow \text{(Disj)}$$
$$v_9 : x, (\exists r.x), (\exists r^2.x), A \sqcup (\exists r^2.x) \sqcup (\exists r^3.x), A, (\exists r^3.x)$$

Fig. 2. Fully-expanded concept expansion tree for Example 5

Example 5. Let $\mathcal{D} = \mu x.(A \sqcup (\exists r.\exists r.x) \sqcup (\exists r.\exists r \exists r.x))$. The fully expanded concept expansion tree for $\mathcal{D}$ is depicted in Figure 2.

3.2 Step 2

In the second step of our normalisation procedure the concept expansion tree that was obtained in the previous step is transformed into an *expansion hypergraph.*

Definition 5 (Expansion Hypergraph). *Let S be a finite set of atomic $\mathcal{ELU}_V$-concepts. An* expansion hypergraph over *S is a finite labelled directed hypergraph $(\mathcal{V}, \mathcal{E}, \mathcal{L}, \mathcal{R})$ with a dedicated set $\mathcal{R}$ of root nodes, where*

- *$\mathcal{V}$ is a finite, non-empty set of nodes;*
- *$\mathcal{E} \subseteq \mathcal{V} \times 2^{\mathcal{V}}$ is a set of directed hyperedges;*
- *$\mathcal{L} \colon \mathcal{V} \cup \mathcal{E} \to 2^S \cup 2^{\mathsf{N_R}}$ is a labelling function, mapping nodes $v \in \mathcal{V}$ to subsets $\mathcal{L}(v) \subseteq S$, and mapping edges $e \in \mathcal{E}$ to sets of role names $\mathcal{L}(e) \subseteq \mathsf{sig}(S) \cap \mathsf{N_R}$;*
- *$\mathcal{R} \subseteq \mathcal{V}$ is a non-empty set of root nodes,*

such that if $\mathcal{L}(v) = \mathcal{L}(v')$ for some $v, v' \in \mathcal{V}$, then $v = v'$.

Nodes in such hypergraphs are labelled with sets of atomic concepts and sets of roles occurring in the outermost existential restrictions of such concepts are the labels of hyperedges. Expansion hypergraphs have a dedicated set of root nodes, indicating a starting point for concept unfoldings, which will be defined later. Note also that in expansion hypergraphs all the nodes have different labels, which ensures that only finitely many expansion hypergraphs over S exist.

We now describe how a fully expanded concept expansion tree can be transformed into an expansion hypergraph over the set of atomic concepts occurring in the node labels of the concept expansion tree.

Definition 6 (Concept Expansion Tree Transformation). *Let $T = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ a fully expanded concept expansion tree for a closed $\mathcal{ELU}_\mu$-concept $\mathcal{C}$ w.r.t. a TBox or variable expansion function ξ with root node v_0. Let $\mathcal{V}_\exists \subseteq \mathcal{V}$ be the set of nodes on which the (Exists)-rule was applied. For every $v \in \mathcal{V}_\exists$ let $succ_r(v) \in \mathcal{V}$ such that $(v, succ_r(v)) \in \mathcal{E}$ and $\mathcal{L}(succ_r(v)) = \{ C \mid \exists r.C \in \mathcal{L}(v) \}$.*

First, given $v \in \mathcal{V}$, let $\chi(v) \subseteq \mathcal{V}$ be the smallest set closed under the following conditions:

- *if $v \in \mathcal{V}_\exists$, let $\chi(v) := \{v\}$;*
- *if v is a leaf that is not blocked, let $\chi(v) := \{v\}$;*
- *if v is blocked by an ancestor v', let $\chi(v) := \chi(v')$;*
- *otherwise, let $\chi(v) := \bigcup \{ \chi(v_i) \mid (v, v_i) \in \mathcal{E} \}$.*

We now define the expansion hypergraph $\mathcal{G}'_T = (\mathcal{V}', \mathcal{E}', \mathcal{L}', \mathcal{R}')$ for T as follows:

- $\mathcal{V}' = \chi(v_0) \cup \bigcup \{ \chi(succ_r(v)) \mid v \in \mathcal{V}_\exists, \exists r.C \in \mathcal{L}(v) \}$;
- $\mathcal{E}' = \{ (v, \chi(succ_r(v))) \mid \exists r.C \in \mathcal{L}(v) \}$;
- $\mathcal{L}' = \{ (v, Atoms(\mathcal{L}(v))) \mid v \in \mathcal{V}' \} \cup$
  $\{ (e, M) \mid e = (v, \chi(v')), M = \{ r \mid v' = succ_r(v)) \} \}$;
- $\mathcal{R}' = \chi(v_0)$.

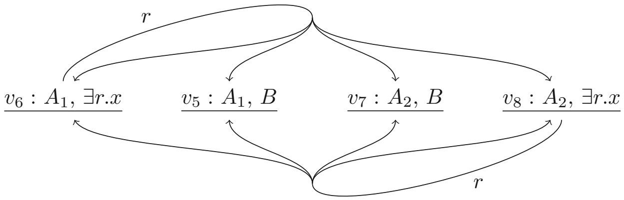
We obtain the required hypergraph $\mathcal{G}_T$ from $\mathcal{G}'_T$ by taking the quotient of $\mathcal{G}'_T$ w.r.t. equal node labels, i.e. all nodes $v_1, v_2 \in \mathcal{V}'$ such that $\mathcal{L}'(v_1) = \mathcal{L}'(v_2)$ are unified.

The expansion hypergraph of a closed $\mathcal{ELU}_\mu$-concept $\mathcal{C}$ is the expansion hypergraph $\mathcal{G}_{T_\mathcal{C}}$ for a fully expanded concept expansion tree $T_\mathcal{C}$ for $\mathcal{C}$ w.r.t. the variable expansion function $\xi_\mathcal{C}$ for $\mathcal{C}$. Similarly, the expansion hypergraph of an $\mathcal{ELU}$-concept D w.r.t. a TBox $\mathcal{T}$ is the expansion hypergraph $\mathcal{G}_{T_{(D,\mathcal{T})}}$ for a fully expanded concept expansion tree $T_{(D,\mathcal{T})}$ for D w.r.t. the TBox expansion function $\xi_{(D,\mathcal{T})}$.

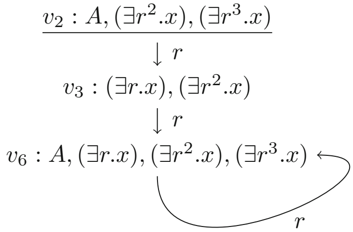
For a node $v \in \mathcal{V}$, the set $\chi(v)$ consists of leaf nodes and nodes on which the (Exists)-rule was applied that are reachable from v in T without walking along an edge that was produced by the (Exists)-rule. In this way the set $\chi(v)$ can be seen as representing the conjunctive normal form of the concepts in $\mathcal{L}(v)$.

The function χ is used to define the node set and the set of root nodes of the resulting expansion hypergraph. Note that not necessarily all nodes of T are nodes in the expansion hypergraph for T. The hyperedges connect a node with a set of nodes. Every node is labelled with a set of atomic concepts, and every hyperedge with a set of roles.

Example 6. Let $\mathcal{C}$ and T be defined as in Example 4. The expansion hypergraph for T is shown below. The root nodes are underlined. We have: $\chi(v_0) = \chi(v_1) = \chi(v_2) = \chi(v_{10}) = \{v_5, v_6, v_7, v_8\}$, $\chi(v_3) = \{v_5, v_6\}$, and $\chi(v_4) = \{v_7, v_8\}$.



Example 7. Let $\mathcal{D}$ and T be defined as in Example 5. The expansion hypergraph for T is shown below. Note that only v_2 is a root node.



We now define how to obtain the $\mathcal{EL}$-concepts that are represented by an expansion hypergraph.

Definition 7 ($\mathcal{EL}$-Concept Unfoldings of an Expansion Hypergraph). *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L}, \mathcal{R})$ be an expansion hypergraph over a finite set S of atomic $\mathcal{ELU}_V$-concepts. First, let $Unfold_{\mathcal{G}} \subseteq \mathcal{V} \times \mathcal{EL}$ be the smallest set closed under the following conditions:*

- *if $v \in \mathcal{V}$, $\varphi = A \in \mathcal{L}(v)$ or $\varphi = \top \in \mathcal{L}(v)$, then $(v, \varphi) \in Unfold_{\mathcal{G}}$;*
- *if $e = (v, \{v_1, \ldots, v_n\}) \in \mathcal{E}$, $r \in \mathcal{L}(e)$, $(v_i, C_i) \in Unfold_{\mathcal{G}}$ for every $1 \le i \le n$, then $(v, \exists r. \bigsqcap_{i=1}^{n} C_i) \in Unfold_{\mathcal{G}}$.*

We set $Unfold(\mathcal{G}) = \{ \bigsqcap_{v \in \mathcal{R}} C_v \mid (v, C_v) \in Unfold_{\mathcal{G}} \}$.

Example 8. Let $\mathcal{G}_{\mathcal{C}}$ be the expansion hypergraph for $\mathcal{C}$, and let $\mathcal{G}_{\mathcal{D}}$ be the expansion hypergraph for $\mathcal{D}$, where $\mathcal{C}$, $\mathcal{D}$ are defined as in Examples 4 and 5. Then $\text{Unfold}(\mathcal{G}_{\mathcal{C}}) = \{A_1 \sqcap A_2, A_1 \sqcap A_2 \sqcap B, A_1 \sqcap \exists r.(A_1 \sqcap A_2), \ldots\}$ and $\text{Unfold}(\mathcal{G}_{\mathcal{D}}) = \{A, \exists r^2.A, \exists r^3.A, \exists r^4.A, \ldots\}$.

3.3 Step 3

The last step of our normalisation procedure removes superfluous nodes from the expansion hypergraph obtained in the previous step to ensure the correctness of our simulation check. Here, a node is superfluous if it does not yield an $\mathcal{EL}$-concept unfolding (cf. Definition 7).

Definition 8 (Simplifying Expansion Hypergraphs). *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L}, \mathcal{R})$ be an expansion hypergraph over a finite set S of atomic $\mathcal{ELU}_V$-concepts.*

Let $\mathcal{V}_{red} \subseteq \mathcal{V}$ be the smallest set closed under the following conditions:

- *for every $v \in \mathcal{V}$ with $\mathcal{L}(v) \cap \mathsf{N_C} \neq \emptyset$ or $\top \in \mathcal{L}(v)$, we have $v \in \mathcal{V}_{red}$;*
- *if $v_1, \ldots, v_n \in \mathcal{V}_{red}$ and $(v, \{v_1, \ldots, v_n\}) \in \mathcal{E}$, then $v \in \mathcal{V}_{red}$.*

The simplified *expansion graph of $\mathcal{G}$ is the expansion graph $\mathcal{G}'$ such that $\mathcal{G}' = (\{v\}, \emptyset, \{v \mapsto \emptyset\}, \{v\})$ if $\mathcal{V}_{red} \cap \mathcal{R} = \emptyset$; and otherwise, $\mathcal{G}' = (\mathcal{V}', \mathcal{E}', \mathcal{L}', \mathcal{R}')$ where*

- *$\mathcal{V}' = \mathcal{V}_{red}$;*
- *$\mathcal{E}' = \{ (v, \{v_1, \ldots, v_n\}) \mid v \in \mathcal{V}', \{v_1, \ldots, v_n\}) \subseteq \mathcal{V}' \}$;*
- *$\mathcal{L}' = \{ (v, \mathcal{L}_v(v)) \mid v \in \mathcal{V}' \} \cup \{ (e, \mathcal{L}_e(e)) \mid e \in \mathcal{V}' \}$; and*
- *$\mathcal{R}' = \mathcal{R} \cap \mathcal{V}'$.*

Intuitively, an expansion hypergraph is simplified by starting from nodes containing concept names or $\top$ in their labels and by collecting the nodes and hyperedges encountered while following hyperedges backwards. Note that we only walk an edge $(v_0, \{v_1, \ldots, v_n\})$ to the node v_0 when all the nodes v_i have been visited already (i.e. they are contained in $\mathcal{V}_{\text{red}}$). This condition corresponds to the intuition that for building a concept for v_0 it must be possible to construct at least one concept for every node v_i.

Example 9. Let $\mathcal{C} = A \sqcup \mu x.(\exists r.x))$. We obtain the following concept expansion hypergraph for $\mathcal{C}$. The concept expansion hypergraph for $\mathcal{C}$ and its simplification are respectively shown on the left-hand and right-hand side below.

We can now state the correctness property of our normal form transformation, i.e. all the $\mathcal{EL}$-concepts that are subsumed by the initial closed $\mathcal{ELU}_\mu$-concept $\mathcal{C}$ or by an $\mathcal{ELU}$-concept D w.r.t. a TBox $\mathcal{T}$ are preserved.

In the following we write $S_1 \equiv S_2$, for two sets S_1, S_2 of $\mathcal{EL}$-concepts, to denote that for every $C_1 \in S_1$ there exists $C_2 \in S_2$ with $\models C_1 \sqsubseteq C_2$, and that for every $D_2 \in S_2$ there exists $D_1 \in S_1$ with $\models D_2 \sqsubseteq D_1$.

Theorem 1. *Let $\mathcal{T}$ be a normalised $\mathcal{EL}$-TBox, let $\mathcal{C}$ be a closed $\mathcal{ELU}_\mu$-concept, and let D be an $\mathcal{ELU}$-concept. Then the expansion hypergraph $\mathcal{G}(\mathcal{C})$ for $\mathcal{C}$ and the expansion hypergraph $\mathcal{G}_\mathcal{T}(D)$ for D w.r.t. $\mathcal{T}$ can be computed in exponential time w.r.t. the size of $\mathcal{C}$, or D and $\mathcal{T}$, respectively. Moreover, the following two statements hold:*

(i) Unfold($\mathcal{G}(\mathcal{C})$) $\equiv \{ E \in \mathcal{EL} \mid \emptyset \models E \sqsubseteq \mathcal{C} \}$;
(ii) Unfold($\mathcal{G}_\mathcal{T}(D)$) $\equiv \{ E \in \mathcal{EL} \mid \mathcal{T} \models E \sqsubseteq D \}$.

4 Simulation

We are now ready to characterise the subsumption $\mathcal{T} \models \mathcal{C} \sqsubseteq D$ in terms of simulations between the respective expansion hypergraphs. In this way we obtain a practical decision procedure for the subsumption $\mathcal{T} \models \mathcal{C} \sqsubseteq D$.

Definition 9 (Expansion Graph Simulation). *Let $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1, \mathcal{L}_1, \mathcal{R}_1)$, $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2, \mathcal{L}_2, \mathcal{R}_2)$ be two expansion graphs.*

We say that $\mathcal{G}_1$ can be simulated *by $\mathcal{G}_2$, written $\mathcal{G}_1 \hookrightarrow \mathcal{G}_2$, iff there exists a binary relation $S \subseteq \mathcal{V}_1 \times \mathcal{V}_2$ which fulfills the following conditions:*

(i) if $(v_1, v_2) \in S$ and $\top \notin \mathcal{L}_2(v_2)$, then $\mathcal{L}_1(v_1) \cap (\mathsf{N_C} \cup \{\top\}) \subseteq \mathcal{L}_2(v_2)$;
(ii) if $(v_1, v_2) \in S$, $\top \notin \mathcal{L}_2(v_2)$, and $e_1 = (v_1, H_1) \in \mathcal{E}_1$, then for every $r \in \mathcal{L}_1(e_1)$ there exists $e_2 = (v_2, H_2) \in \mathcal{E}_2$ such that $r \in \mathcal{L}_2(e_2)$ and for every $v_2' \in H_2$ there exists $v_1' \in H_1$ with $(v_1, v_1') \in S$; and
(iii) for every $v_2 \in \mathcal{R}_2$ there exists $v_1 \in \mathcal{R}_1$ such that $(v_1, v_2) \in S$.

The simulation Condition (i) ensures that all the concept names contained in the label of v_1 must also be present in the label of v_2 (if the label of v_2 does not contain $\top$). Condition (i) is a *local* condition in the sense that it does not depend on other nodes to be contained in the simulation. Condition (ii) propagates the simulation conditions along hyperedges (if the label of v_2 does not contain $\top$), and in contrast to Condition (i) it imposes constraints on other nodes. Condition (iii) enforces that the root nodes are present in the simulation.

For a more detailed explanation of the simulation conditions, we refer the reader to [6,12], where a similar simulation notion between hypergraphs and its connection to reasoning has been established.

We note that without Step 3 in our normal form transformation it would be impossible to establish for $\mathcal{C} = \mu x.(\exists r.x)$ and $D = A$ that $\mathcal{T} \models \mathcal{C} \sqsubseteq A$ holds (as $\mathcal{C}^{\mathcal{I},\emptyset} = \emptyset$ in every interpretation $\mathcal{I}$). Condition (ii) would require the hypergraph $\mathcal{G}_\mathcal{T}(D)$ to contain edges labelled with r, which it does not have.

Example 10. Let $\mathcal{T} = \{\exists s.X \sqsubseteq Y, \exists r.Y \sqsubseteq X, A \sqsubseteq X\}$ and let $\mathcal{C} = B \sqcap \mu x.(A \sqcup \exists r.\exists s.x)$. Then $\mathcal{T} \models \mathcal{C} \sqsubseteq A \sqcup \exists r.\exists s.X$. The simplified expansion hypergraph $\mathcal{G}(\mathcal{C})$ for $\mathcal{C}$ and the simplified expansion hypergraph $\mathcal{G}_\mathcal{T}(\mathcal{D})$ for $\mathcal{D}$ w.r.t. $\mathcal{T}$ are shown in Figure 3.

We have that $S = \{(v_1, v_0'), (v_2, v_1'), (v_1, v_2'), (v_2, v_3')\}$ is a simulation between $\mathcal{G}(\mathcal{C})$ and $\mathcal{G}_\mathcal{T}(D)$.

We can now state our main result, linking the existence of a simulation between simplified expansion hypergraphs with subsumption.

Theorem 2. *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox, let $\mathcal{T}'$ be its normalisation, let $\mathcal{C}$ be a closed $\mathcal{ELU}_\mu$-concept, and let D be an $\mathcal{ELU}$-concept. Additionally, let $\mathcal{G}(\mathcal{C})$ be the simplified expansion hypergraph for $\mathcal{C}$ and let $\mathcal{G}_\mathcal{T}(\mathcal{D})$ be the simplified expansion hypergraph for $\mathcal{D}$ w.r.t. $\mathcal{T}'$. Then the following two statements are equivalent:*

(i) $\mathcal{T}' \models \mathcal{C} \sqsubseteq D$;
(ii) $\mathcal{G}(\mathcal{C}) \hookrightarrow \mathcal{G}_{\mathcal{T}'}(D)$.

The subsumption $\mathcal{T} \models \mathcal{C} \sqsubseteq D$ can be decided in exponential time in the size of $\mathcal{T}$, $\mathcal{C}$, and D.

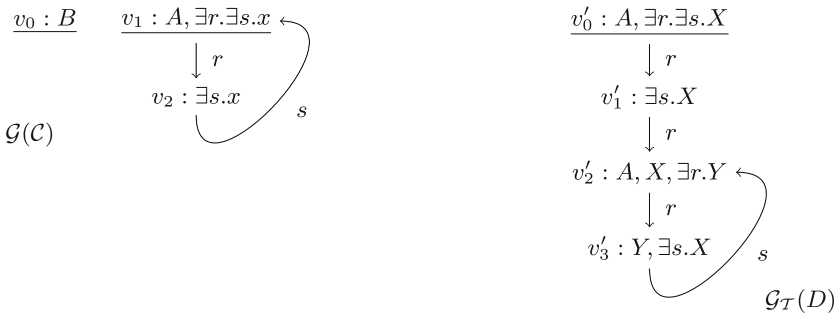


Fig. 3. Simplified expansion hypergraphs for Example 10

5 Conclusion and Discussion

We have provided a procedure for deciding subsumptions of the form $\mathcal{T} \models \mathcal{C} \sqsubseteq D$, where $\mathcal{C}$ is an $\mathcal{ELU}_\mu$-concept, D an $\mathcal{ELU}$-concept and $\mathcal{T}$ an $\mathcal{EL}$-TBox. Our procedure is based on checking for the existence of a certain simulation between hypergraph representations of the unfoldings of $\mathcal{C}$ and the subsumees of D w.r.t. $\mathcal{T}$, respectively. We have presented in detail how the hypergraphs can be built in three steps from $\mathcal{C}$ and D, $\mathcal{T}$, not relying on automata-theoretic techniques.

We plan to apply our procedure for solving the logical difference problem between $\mathcal{EL}$-TBoxes and for checking the existence of uniform interpolants of $\mathcal{EL}$-TBoxes. In this context an evaluation of the procedure will be required.

Applying our procedure for solving $\mathcal{T} \models \mathcal{C} \sqsubseteq D$ to the logical difference problem between $\mathcal{EL}$-TBoxes requires finding a suitable $\mathcal{ELU}$-concept D, which acts as an interpolant between $\mathcal{C}$ and an $\mathcal{EL}_\nu$-concept w.r.t. $\mathcal{T}$ (cf. Section 1). The notion of the *least common subsumer* (lcs) appears to lend itself as a candidate for such concepts D [7]. For our purposes we would consider the following notion. An $\mathcal{ELU}$-concept D is the *least common subsumer* of an $\mathcal{ELU}_\mu$-concept $\mathcal{C}$ w.r.t. an $\mathcal{EL}$-TBox $\mathcal{T}$ (written $\mathsf{lcs}_{\mathcal{T}}(\mathcal{C})$) if it satisfies the following two conditions: (i) $\mathcal{T} \models \mathcal{C} \sqsubseteq D$, and (ii) $\mathcal{T} \models D \sqsubseteq E$, for all $\mathcal{ELU}$-concepts E with $\mathcal{T} \models \mathcal{C} \sqsubseteq E$. Intuitively, $\mathcal{C}$ stands for a disjunction of infinitely many concept descriptions which are to be approximated by an $\mathcal{ELU}$-concept. However, such an lcs does not always exist. For instance, let $\mathcal{C} = \mu x.(A \sqcup \exists r.x)$ and $\mathcal{T}' = \{A \sqsubseteq Y, \exists r.Y \sqsubseteq Y\}$. For $\varphi_0 = Y$, $\varphi_1 = A \sqcup \exists r.Y$, $\varphi_2 = A \sqcup \exists r.(A \sqcup \exists r.Y)$, etc., it holds that $\mathcal{T}' \models \mathcal{C} \sqsubseteq \varphi_i$ and $\mathcal{T}' \models \varphi_{i+1} \sqsubseteq \varphi_i$ for $i \geq 0$. Then, for every $\mathcal{ELU}$-concept E with $\mathcal{T}' \models \mathcal{C} \sqsubseteq E$ we have that $\mathcal{T}' \not\models E \sqsubseteq \varphi_i$ for some $i \geq 0$.

As an alternative to an lcs of a least fixpoint concepts w.r.t. a background TBox, we say that an $\mathcal{ELU}$-concept D is a *most-consequence preserving subsumer (mcps)* of an $\mathcal{ELU}_\mu$-concept $\mathcal{C}$ for $\mathcal{EL}$-consequences w.r.t. an $\mathcal{EL}$-TBox $\mathcal{T}$ iff the following conditions hold: (i) $\mathcal{T} \models \mathcal{C} \sqsubseteq D$, and (ii) there does not exist an $\mathcal{ELU}$-concept E with $\mathcal{T} \models \mathcal{C} \sqsubseteq E$ and $\{ F \in \mathcal{EL} \mid \mathcal{T} \models D \sqsubseteq F \} \subsetneq \{ F \in \mathcal{EL} \mid \mathcal{T} \models E \sqsubseteq F \}$. Continuing the example above, every φ_i is an mcps of $\mathcal{C}$ w.r.t. $\mathcal{T}$ as $\{ F \in \mathcal{EL} \mid \mathcal{T}' \models \varphi_i \sqsubseteq F \} = \{Y\}$ and $\{ F \in \mathcal{EL} \mid \mathcal{T}' \models \mathcal{C} \sqsubseteq F \} = \{Y\}$.

We plan to investigate the notion of an mcps further and possibly apply it to finding interpolants of least and greatest fixpoint concepts w.r.t. TBoxes.

References

1. Baader, F., Brandt, S., Lutz, C.: Pushing the EL envelope. In: Proceedings of IJCAI-05: the 19th International Joint Conference on Artificial Intelligence, pp. 364–369. Morgan-Kaufmann Publishers (2005)
2. Baader, F., Brandt, S., Lutz, C.: Pushing the EL envelope further. In: Proceedings of the OWLED 2008 DC Workshop on OWL: Experiences and Directions (2008)
3. 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, 2nd edition (2010)
4. Bradfield, J., Stirling, C.: Modal μ-calculi. In: Blackburn, P., van Benthem, J., Wolter, F. (eds.) Handbook of Modal Logic, Studies in Logic and Practical Reasoning, vol. 3, pp. 721–756. Elsevier (2007)
5. Brandt, S.: Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and–what else? In: Proceedings of ECAI-04: the 16th European Conference on Artificial Intelligence, pp. 298–302. IOS Press (2004)
6. Ecke, A., Ludwig, M., Walther, D.: The concept difference for EL-terminologies using hypergraphs. In: Proceedings of DChanges-13: the International Workshop on (Document) Changes: Modeling, Detection, Storage and Visualization. CEUR-WS, vol. 1008 (2013)
7. Baader, F., Küsters, R., Molitor, R.: Computing least common subsumers in description logics with existential restrictions. In: Proceedings of IJCAI-99: the 16th International Joint Conference on Artificial Intelligence, pp. 96–101. Morgan-Kaufmann Publishers (1999)
8. Feng, S., Ludwig, M., Walther, D.: The logical difference for EL: from terminologies towards TBoxes. In: Proceedings of IWOST-15: the 1st International Workshop on Semantic Technologies. CEUR-WS (2015)
9. Janin, D., Walukiewicz, I.: Automata for the modal mu-calculus and related results. In: Wiedermann, J., Hájek, P. (eds.) Mathematical Foundations of Computer Science 1995. LNCS, vol. 969, pp. 552–562. Springer, Heidelberg (1995)
10. Konev, B., Ludwig, M., Walther, D., Wolter, F.: The logical difference for the lightweight description logic EL. Journal of Artificial Intelligence Research (JAIR) **44**, 633–708 (2012)
11. Kozen, D.: Results on the propositional μ-calculus. Theoretical Computer Science **27**(3), 333–354 (1983)
12. Ludwig, M., Walther, D.: The logical difference for ELHr-terminologies using hypergraphs. In: Proceedings of ECAI-14: the 21st European Conference on Artificial Intelligence. Frontiers in Artificial Intelligence and Applications, vol. 263, pp. 555–560. IOS Press (2014)
13. Sattler, U., Vardi, M.Y.: The hybrid mgr-calculus. In: Goré, R.P., Leitsch, A., Nipkow, T. (eds.) IJCAR 2001. LNCS (LNAI), vol. 2083, p. 76. Springer, Heidelberg (2001)