Matching in Hybrid Terminologies

Sebastian Brandt

School of Computer Science, Manchester, UK
brandt@cs.manchester.ac.uk

Abstract. In the area of Description Logic (DL) based knowledge representation, hybrid terminologies have been proposed as a means to make non-standard inference services available to knowledge bases that contain general concept inclusion (GCI) axioms. Building on existing work on subsumption in hybrid terminologies, the present paper provides the first in-depth investigation of the non-standard inferences least-common subsumer, and matching in hybrid $\mathcal{EL}$-TBoxes; providing sound and complete algorithms for both inference services.

1 Motivation

In Description Logic (DL) based knowledge representation (KR), intensional knowledge of a given domain is represented by a terminology (TBox) that defines properties of concepts relevant to the domain [1]. A TBox usually comprises *definitions* of the form $A \equiv C$ by which a *concept name* A is assigned to a *concept description* C. Concept descriptions are terms built from atomic concepts by means of a set of constructors provided by the DL under consideration. TBoxes are interpreted with a model-theoretic *semantics* which allows to reason over the terminology in a formally well-defined way. Our DL of interest is $\mathcal{EL}$ which provides top concept ($\top$), conjunction ($\sqcap$), and existential restriction ($\exists r.C$).

General TBoxes additionally allow for *general concept inclusion (GCI)* axioms of the form $C \sqsubseteq D$, where both C and D may be complex concept descriptions. GCIs define implications ("D holds whenever C holds") relevant to the terminology as a whole. The utility of GCIs for practical KR applications has been examined in depth; see, e.g., [2,3,4]. In addition to constraining (admissible models of) terminologies further without explicitly changing all its definitions, using GCIs can lead to smaller, more readable TBoxes, and can facilitate the re-use of data in applications of different levels of detail. Consequently, GCIs are supported by most modern DL reasoners such as FaCT [5], Racer [6], Pellet [7], and Cel [8].

One of the most important reasoning services provided by such DL systems is *classification*, i.e., computing the subsumption hierarchy. Before DL systems can be deployed for reasoning over terminologies in an application area, however, the relevant TBoxes must be built-up and maintained. In order to support these knowledge engineering tasks, additional so-called 'non-standard' inference services have been proposed, most notably *least-common subsumer (lcs)* [9,10,11,12] and *matching* [13,14,15]. As discussed in [16], the lcs facilitates the build-up of

N. Dershowitz and A. Voronkov (Eds.): LPAR 2007, LNAI 4790, pp. 166–180, 2007.

DL knowledge bases in a 'bottom-up' fashion suitable for domain experts with limited KR background. Among other applications, matching can be used as a means of querying TBoxes for concepts of a certain structure [17]. This can be utilized to construct new concepts by retrieving and modifying structurally similar ones in the TBox.

Unfortunately, non-standard inferences are not straightforwardly available for general TBoxes: it has been shown in [18] that lcs need not always exist, even for cyclic $\mathcal{EL}$-TBoxes interpreted with descriptive semantics, the standard semantics for DL systems. This result carries over to general $\mathcal{EL}$-TBoxes and any extension of $\mathcal{EL}$. The same holds for matching which relies on the lcs.

In order to provide non-standard inferences in the presence of GCIs, so-called *hybrid TBoxes* have been proposed [19]. A hybrid $\mathcal{EL}$-TBox is a pair $(\mathcal{F}, \mathcal{T})$ of a general TBox $\mathcal{F}$ ('foundation') and a possibly cyclic TBox $\mathcal{T}$ ('terminology') defined over the same set of atomic concepts and roles. $\mathcal{F}$ serves as a foundation of $\mathcal{T}$ in that the GCIs in $\mathcal{F}$ define relationships between concepts used as atomic concept names in the definitions in $\mathcal{T}$. Hence, $\mathcal{F}$ lays a foundation of general implications constraining $\mathcal{T}$. The semantics of hybrid TBoxes is different from the usual descriptive semantics: while the foundation of a hybrid TBox is interpreted with descriptive semantics, the terminology is interpreted with so-called greatest-fixpoint (gfp) semantics to be introduced in detail in Section 2.

With respect to non-standard inferences for hybrid $\mathcal{EL}$-TBoxes, our point of departure is as follows: it has been sketched in [19] how an equivalence-preserving reduction from hybrid to cyclic $\mathcal{EL}$-TBoxes with gfp-semantics can be exploited to utilize the lcs defined for cyclic $\mathcal{EL}$-TBoxes with gfp-semantics in [18]. The lcs algorithm thus obtainable for hybrid $\mathcal{EL}$-TBoxes has not yet been studied, though. In case of matching, the above mentioned reduction appears useful as well, only that no matching algorithm for cyclic $\mathcal{EL}$-TBoxes with descriptive semantics exists as yet. The present paper closes both gaps before turning to matching in hybrid TBoxes: after introducing matching in cyclic $\mathcal{EL}$-TBoxes with gfp-semantics in Section 3 and the least-common subsumer for hybrid TBoxes in Section 4.1, our matching algorithm for hybrid TBoxes is presented in Section 4.2. It should be noted that matching problems have not yet been defined for cyclic or hybrid $\mathcal{EL}$ TBoxes. Hence, Sections 3 and 4 start by introducing the relevant notions for the cyclic and hybrid case, respectively. Given that hybrid TBoxes may be viewed as a rather exotic KR formalism, we conclude by discussing the utility of our results for common general $\mathcal{EL}$-TBoxes.

All details and complete proofs can be found in our technical report [20].

2 Formal Preliminaries

Concept descriptions are inductively defined with the help of a set of concept *constructors*, starting with arbitrary but fixed disjoint sets $\mathsf{N}_{\text{prim}} \uplus \mathsf{N}_{\text{def}} =: \mathsf{N}_{\text{con}}$ of *primitive concept names* (N_{prim}) and *defined concept names* (N_{def}), respectively, and a set N_{role} of *role names*. The DL $\mathcal{EL}$ provides the concept constructors top-concept ($\top$), conjunction ($\sqcap$), and existential restrictions ($\exists r.C$). Concept descriptions using only these constructors are called $\mathcal{EL}$-concept descriptions.

As usual, the semantics of concept descriptions is defined in terms of an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. The domain $\Delta^{\mathcal{I}}$ of $\mathcal{I}$ is a non-empty set and the interpretation function $\cdot^{\mathcal{I}}$ maps each concept name $P \in \mathsf{N}_{\text{prim}} \cup \mathsf{N}_{\text{def}}$ to a subset $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and each role name $r \in \mathsf{N}_{\text{role}}$ to a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The extension of $\cdot^{\mathcal{I}}$ to arbitrary $\mathcal{EL}$-concept descriptions is defined inductively as follows: $\top^{\mathcal{I}} := \Delta^{\mathcal{I}}$, $(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$, and

$$(\exists r.C)^{\mathcal{I}} := \{x \in \Delta^{\mathcal{I}} \mid \exists y\colon (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}.$$

The main purpose of DLs is to be used as underlying representation language for knowledge bases. Two common kinds of DL knowledge bases, TBoxes and general TBoxes, are defined as follows.

For every $A \in \mathsf{N}_{\text{def}}$ and every $\mathcal{EL}$-concept description C over N_{con} and N_{role}, $A \equiv C$ is a *definition of* A. Every finite set of definitions is an *$\mathcal{EL}$-terminology ($\mathcal{EL}$-TBox)* over N_{def}, N_{prim}, and N_{role} iff it contains at most one definition of A for every $A \in \mathsf{N}_{\text{def}}$. An $\mathcal{EL}$-TBox $\mathcal{T}$ is *acyclic* iff $\mathcal{T}$ is of the form $\{A_i \equiv C_i \mid 1 \leq i \leq n\}$ such that for every $i \in \{1, \ldots, n\}$, only defined names from $\{A_1, \ldots, A_{i-1}\}$ occur in C_i. For concept descriptions C, D over N_{con} and N_{role}, $C \sqsubseteq D$ is a *general concept inclusion (GCI) axiom*. Every finite set of GCIs is a *general $\mathcal{EL}$-TBox*. For every $\mathcal{EL}$-TBox $\mathcal{T}$, denote by $\mathsf{N}_{\text{con}}^{\mathcal{T}}$ ($\mathsf{N}_{\text{def}}^{\mathcal{T}}$, $\mathsf{N}_{\text{prim}}^{\mathcal{T}}$) and $\mathsf{N}_{\text{role}}^{\mathcal{T}}$ the sets of all (defined, primitive) concept names and role names, respectively, occurring in $\mathcal{T}$. For general $\mathcal{EL}$-TBoxes, only $\mathsf{N}_{\text{con}}^{\mathcal{T}}$ and $\mathsf{N}_{\text{role}}^{\mathcal{T}}$ apply. For the sake of brevity, we may write TBox instead of $\mathcal{EL}$-TBox.

Descriptive semantics: an interpretation $\mathcal{I}$ is a *model of a general TBox $\mathcal{T}$* ($\mathcal{I} \models \mathcal{T}$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every GCI $C \sqsubseteq D \in \mathcal{T}$. Every (non-general) TBox can be viewed as a general TBox since every definition $A \equiv C$ is equivalent to the pair of GCIs $A \sqsubseteq C$, $C \sqsubseteq A$. This semantics is usually called *descriptive semantics* [22].

One of the most basic inference services provided by DL systems is computing the subsumption hierarchy. For concept descriptions C, D defined in a TBox $\mathcal{T}$, *C is subsumed by D w.r.t. $\mathcal{T}$* ($C \sqsubseteq_{\mathcal{T}} D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every model of $\mathcal{T}$. *C is equivalent to D w.r.t. $\mathcal{T}$* ($C \equiv_{\mathcal{T}} D$) iff $C \sqsubseteq_{\mathcal{T}} D$ and $D \sqsubseteq_{\mathcal{T}} C$. Explicit reference to the empty TBox may be omitted: if $\mathcal{T} = \emptyset$, write $C \sqsubseteq D$ instead of $C \sqsubseteq_{\mathcal{T}} D$, and analogously for equivalence.

Greatest-fixpoint semantics: for (non-general) TBoxes, we additionally introduce greatest-fixpoint semantics. We begin by interpreting only primitive concepts and roles occurring: for every TBox $\mathcal{T}$, a *primitive interpretation* $(\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ of $\mathcal{T}$ interprets all primitive concepts $P \in \mathsf{N}_{\text{prim}}$ by subsets of $\Delta^{\mathcal{J}}$ and all roles $r \in \mathsf{N}_{\text{role}}$ by binary relations on $\Delta^{\mathcal{J}}$. An Interpretation $\mathcal{I} := (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is *based on* $\mathcal{J}$ iff $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}}$ and $\cdot^{\mathcal{J}}$ and $\cdot^{\mathcal{I}}$ coincide on N_{role} and N_{prim}. The set of all interpretations based on $\mathcal{J}$ is denoted by $\text{Int}(\mathcal{J})$. On $\text{Int}(\mathcal{J})$, a binary relation $\preceq_{\mathcal{J}}$ is defined for all $\mathcal{I}_1, \mathcal{I}_2 \in \text{Int}(\mathcal{J})$ by $\mathcal{I}_1 \preceq_{\mathcal{J}} \mathcal{I}_2$ iff $A^{\mathcal{I}_1} \subseteq A^{\mathcal{I}_2}$ for all $A \in \mathsf{N}_{\text{def}}^{\mathcal{T}}$.

The pair $(\text{Int}(\mathcal{J}), \preceq_{\mathcal{J}})$ is a complete lattice, so that every subset of $\text{Int}(\mathcal{J})$ has a least upper bound (lub) and a greatest lower bound (glb) w.r.t. $\preceq_{\mathcal{J}}$. Hence, by

Tarski's fixpoint theorem [23], every monotonic function on $\mathrm{Int}(\mathcal{J})$ has a fixpoint. In particular, this applies to the function $O_{\mathcal{T},\mathcal{J}}$ defined by $O_{\mathcal{T},\mathcal{J}}\colon \mathrm{Int}(\mathcal{J}) \to \mathrm{Int}(\mathcal{J})$ with $\mathcal{I}_1 \mapsto \mathcal{I}_2$ iff $A^{\mathcal{I}_2} = C^{\mathcal{I}_1}$ for all $A \equiv C \in \mathcal{T}$.

As shown in [21], $O_{\mathcal{T},\mathcal{J}}$ is in fact a fixpoint operator on $\mathrm{Int}(\mathcal{J})$. Moreover, it holds that $\mathcal{I}$ is a fixpoint of $O_{\mathcal{T},\mathcal{J}}$ iff $\mathcal{I}$ is a model of $\mathcal{T}$. As a consequence, an interpretation $\mathcal{I}$ is called a *gfp-model* of $\mathcal{T}$ iff there is a primitive interpretation $\mathcal{J}$ such that $\mathcal{I} \in \mathrm{Int}(\mathcal{J})$ is the greatest fixpoint of $O_{\mathcal{T},\mathcal{J}}$.

As $(\mathrm{Int}(\mathcal{J}), \preceq_{\mathcal{J}})$ is a complete lattice, the gfp-model is uniquely determined for a given TBox $\mathcal{T}$ and a primitive interpretation $\mathcal{J}$. We may thus refer to *the* gfp-model $\mathrm{gfp}(\mathcal{T},\mathcal{J})$ for any given $\mathcal{T}$ and $\mathcal{J}$. With this preparation, we define gfp-subsumption by: for concept names A, B defined in $\mathcal{T}$, *A is subsumed by B w.r.t. gfp-semantics* ($A \sqsubseteq_{\mathrm{gfp},\mathcal{T}} B$) iff $A^{\mathcal{I}} \subseteq B^{\mathcal{I}}$ for all gfp-models $\mathcal{I}$ of $\mathcal{T}$.

Note that descriptive semantics considers a superset of the set of gfp-models, implying that descriptive subsumption entails gfp-subsumption. Hence, all subsumption relations w.r.t. $\sqsubseteq_{\mathcal{T}}$ also hold w.r.t. $\sqsubseteq_{\mathrm{gfp},\mathcal{T}}$. Moreover, both semantics coincide on acyclic TBoxes. For $\mathcal{EL}$, our DL of interest, least-fixpoint semantics is inappropriate w.r.t. cyclic TBoxes [21] and hence is not considered.

See [20] for details of how gfp-models can actually be computed.

Deciding subsumption w.r.t. cyclic $\mathcal{EL}$-TBoxes with gfp-semantics: a decision procedure for the subsumption problem w.r.t. cyclic $\mathcal{EL}$-TBoxes with descriptive semantics has been presented in [21]. We repeat the notions central to this procedure in so far as they are required for our matching algorithms w.r.t. cyclic and hybrid $\mathcal{EL}$-TBoxes.

An $\mathcal{EL}$-TBox $\mathcal{T}$ is *normalized* iff $A \equiv D \in \mathcal{T}$ implies that D is of the form $P_1 \sqcap \cdots \sqcap P_m \sqcap \exists r_1.B_1 \sqcap \ldots \exists r_\ell.B_\ell$, where for $m, \ell \geq 0$, $P_1, \ldots, P_m \in \mathsf{N}_{\mathrm{prim}}$ and $B_1, \ldots, B_\ell \in \mathsf{N}_{\mathrm{def}}$. If $m = \ell = 0$ then $D = \top$. The subsumption algorithm in [21] represents normalized $\mathcal{EL}$-TBoxes by means of *description graphs*. Given a normalized $\mathcal{EL}$ TBox $\mathcal{T}$, the $\mathcal{EL}$-description graph $\mathcal{G}_{\mathcal{T}} = (\mathsf{N}_{\mathrm{def}}^{\mathcal{T}}, E_{\mathcal{T}}, L_{\mathcal{T}})$ of $\mathcal{T}$ is defined as follows:

- the nodes of $\mathcal{G}_{\mathcal{T}}$ are the defined concepts of $\mathcal{T}$;
- if A is defined in $\mathcal{T}$ and $A \equiv P_1 \sqcap \cdots \sqcap P_m \sqcap \exists r_1.B_1 \sqcap \cdots \sqcap \exists r_\ell.B_\ell$ is its definition then $L_{\mathcal{T}}(A) := \{P_1, \ldots, P_m\}$, and A is the source of the edges $(A, r_1, B_1), \ldots, (A, r_\ell, B_\ell) \in E_{\mathcal{T}}$.

Any primitive interpretation $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ can be represented by an $\mathcal{EL}$-description graph as well, see [20] for details.

In preparation for the characterization of subsumption we need to introduce simulation relations on description graphs. Given two $\mathcal{EL}$-description graphs $\mathcal{G}_i = (V_i, E_i, L_i)$, $i = 1, 2$, the binary relation $Z \subseteq V_1 \times V_2$ is a *simulation relation* from $\mathcal{G}_1$ to $\mathcal{G}_2$ ($Z\colon \mathcal{G}_1 \rightleftharpoons \mathcal{G}_2$) iff (S1) $(v_1, v_2) \in Z$ implies $L_1(v_1) \subseteq L_2(v_2)$; and (S2) if $(v_1, v_2) \in Z$ and $(v_1, r, v_1') \in E_1$ then there exists a node $v_2' \in V_2$ such that $(v_1', v_2') \in Z$ and $(v_2, r, v_2') \in E_2$.

It has been shown in [21] that simulation relations are closed under concatenation. Moreover, one of the main results in [21] is a characterization of

gfp-subsumption w.r.t. cyclic $\mathcal{EL}$-TBoxes by simulation relations over description graphs. The following results provide the relevant characterizations.

Theorem 1. *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox and A, B be defined concepts in $\mathcal{T}$. Then $A \sqsubseteq_{\mathrm{gfp},\mathcal{T}} B$ iff there is a simulation relation $Z \colon \mathcal{G}_\mathcal{T} \rightleftharpoons \mathcal{G}_\mathcal{T}$ such that $(B, A) \in Z$.*

Since the description graph of a TBox is of polynomial size in the size of the TBox and since the existence of simulation relations with the required properties can be tested in polynomial time, subsumption w.r.t. cyclic $\mathcal{EL}$-TBoxes with gfp-semantics is decidable in polynomial time. [21].

The least-common subsumer w.r.t. cyclic $\mathcal{EL}$-TBoxes: a main preparatory step towards matching w.r.t. cyclic $\mathcal{EL}$-TBoxes is to introduce the lcs for cyclic $\mathcal{EL}$-TBoxes. The relevant definitions are due to [18].

Let $\mathcal{T}_1$ be a cyclic $\mathcal{EL}$-TBox and let $A, B \in \mathsf{N}_{\mathrm{def}}^{\mathcal{T}_1}$. Let $\mathcal{T}_2$ be a conservative extension of $\mathcal{T}_1$ with $E \in \mathsf{N}_{\mathrm{def}}^{\mathcal{T}_2} \setminus \mathsf{N}_{\mathrm{def}}^{\mathcal{T}_1}$. Then E is the *least common subsumer of A and B in $\mathcal{T}_1$ w.r.t. gfp-semantics (gfp-lcs)* iff the following conditions hold:

1. $A \sqsubseteq_{\mathrm{gfp},\mathcal{T}_2} E$ and $B \sqsubseteq_{\mathrm{gfp},\mathcal{T}_2} E$;
2. if $\mathcal{T}_3$ is a conservative extension of $\mathcal{T}_2$ and F a defined concept in $\mathcal{T}_3$ such that $A \sqsubseteq_{\mathrm{gfp},\mathcal{T}_3} F$ and $B \sqsubseteq_{\mathrm{gfp},\mathcal{T}_3} F$ then $E \sqsubseteq_{\mathrm{gfp},\mathcal{T}_3} F$.

In order to be able to actually compute the lcs, we need to compute the product of description graphs. Let $\mathcal{G}_i := (V_i, E_i, L_i)$, $i = 1, 2$ be two description graphs. Their *product* is the description graph $\mathcal{G}_1 \times \mathcal{G}_2 := (V, E, L)$ with $V := V_1 \times V_2$; $E := \{((v_1, v_2), r, (v_1', v_2')) \mid \forall i \in \{1, 2\} \colon (v_i, r, v_i') \in E_i\}$; and $L(v_1, v_2) := L_1(v_1) \cap L_2(v_2)$. For a description graph $\mathcal{G} = (V, E, L)$, the n-ary graph product is inductively defined in the obvious way, i.e., $\mathcal{G}^1 := \mathcal{G}$ and $\mathcal{G}^{n+1} := \mathcal{G}^n \times \mathcal{G}$.

In order to transform product graphs back to TBoxes, we define TBoxes induced by description graphs. Let $\mathcal{G} := (V, E, L)$ be a description graph. Then the *TBox of $\mathcal{G}$* is defined by

$$\mathsf{tbox}(\mathcal{G}) := \{A \equiv \bigsqcap_{P \in L(A)} P \sqcap \bigsqcap_{(A,r,B) \in E} \exists r.B \mid A \in V\}.$$

Two of the main results from [18] prove that the gfp-lcs w.r.t. cyclic $\mathcal{EL}$-TBoxes always exists and can in fact be computed by means of the graph product: for concept names A, B defined in $\mathcal{T}$, the concept (A, B) defined in $\mathcal{T} \cup \mathsf{tbox}(\mathcal{G}_\mathcal{T} \times \mathcal{G}_\mathcal{T})$ is the gfp-lcs of A and B w.r.t. $\mathcal{T}$. Hence, the gfp-lcs can be computed in polynomial time in the binary case can and in exponential time in the general case. We are now prepared to introduce hybrid TBox, our main TBox formalism of interest.

2.1 Hybrid TBoxes

Definition 1. *(Hybrid TBox) For every general $\mathcal{EL}$-TBox $\mathcal{F}$ over $\mathsf{N}_{\mathrm{prim}}$ and $\mathsf{N}_{\mathrm{role}}$, and every $\mathcal{EL}$-TBox $\mathcal{T}$ over $\mathsf{N}_{\mathrm{def}}$, $\mathsf{N}_{\mathrm{prim}}$, and $\mathsf{N}_{\mathrm{role}}$, the pair $(\mathcal{F}, \mathcal{T})$ is called a* hybrid $\mathcal{EL}$-TBox.

In order to simplify the presentation of our subsumption algorithm, we introduce a normal form for hybrid $\mathcal{EL}$-TBoxes. Analogous to the case of cyclic TBoxes, we normalize hybrid TBoxes in order to simplify the presentation of our proofs. See [20] for an example of what an actual hybrid $\mathcal{EL}$-TBox looks like and for details about normalization. The semantics of hybrid TBoxes can now be defined as follows.

Let $(\mathcal{F}, \mathcal{T})$ be a hybrid TBox over N_{prim}, N_{role}, and N_{def}. A primitive interpretation $\mathcal{J}$ is a *model of* $\mathcal{F}$ *(*$\mathcal{J} \models \mathcal{F}$*)* iff $C^{\mathcal{J}} \subseteq D^{\mathcal{J}}$ for all GCIs $C \sqsubseteq D$ in $\mathcal{F}$. A model $\mathcal{I} \in \text{Int}(\mathcal{J})$ is a *gfp-model* of $(\mathcal{F}, \mathcal{T})$ iff $\mathcal{J} \models \mathcal{F}$ and $\mathcal{I}$ is a gfp-model of $\mathcal{T}$.

Note that $\mathcal{F}$ ("foundation") is interpreted with descriptive semantics while $\mathcal{T}$ ("terminology") is interpreted with gfp-semantics. Note also that every gfp-model of $(\mathcal{F}, \mathcal{T})$ can be expressed as the greatest fixpoint $\text{gfp}(\mathcal{T}, \mathcal{J})$ for some primitive interpretation $\mathcal{J}$ with $\mathcal{J} \models \mathcal{F}$.

In order to complete the semantics of hybrid TBoxes, we still have to introduce an appropriate notion of subsumption: Let A, B be defined concepts in $\mathcal{T}$. Then A *is subsumed by* B *w.r.t.* $(\mathcal{F}, \mathcal{T})$ *(*$A \sqsubseteq_{\text{gfp},\mathcal{F},\mathcal{T}} B$*)* iff $A^{\mathcal{I}} \subseteq B^{\mathcal{I}}$ for all gfp-models $\mathcal{I}$ of $(\mathcal{F}, \mathcal{T})$.

Hybrid TBoxes generalize cyclic TBoxes with gfp-semantics in the sense that every cyclic $\mathcal{EL}$-TBox $\mathcal{T}$ can be viewed as a hybrid TBox with an empty foundation. Thus, gfp-subsumption w.r.t. $\mathcal{T}$ coincides with subsumption w.r.t. the hybrid TBox $(\emptyset, \mathcal{T})$. Also note that every general TBox $\mathcal{T}'$ can be seen as a hybrid TBox $(\mathcal{T}', \emptyset)$. In this case, a descriptive subsumption $P \sqsubseteq_{\mathcal{T}'} Q$ holds iff A_P is subsumed by A_Q w.r.t. the normalized instance of $(\mathcal{T}', \emptyset)$.

Deciding subsumption w.r.t. hybrid $\mathcal{EL}$-TBoxes: in order to decide subsumption of concepts defined in a $\mathcal{EL}$-hybrid TBox, an equivalence preserving reduction from hybrid to cyclic $\mathcal{EL}$-TBoxes with gfp-semantics has been proposed in [19]. After the reduction, subsumption can be decided as described above.

The idea underlying the reduction is to use the *descriptive* subsumption relations induced by the GCIs in $\mathcal{F}$ to extend the definitions in $\mathcal{T}$ accordingly. To this end, we view the union of $\mathcal{F}$ and $\mathcal{T}$ as a general TBox and ask for all descriptive implications in $\mathcal{T}$ directly involving names from $\mathcal{F}$. These implications are then added to the definitions in $\mathcal{T}$. This notion is formalized as follows: for a given normalized hybrid $\mathcal{EL}$-TBox $(\mathcal{F}, \mathcal{T})$, the $\mathcal{F}$*-completion* $f(\mathcal{T})$ extends the definitions in $\mathcal{T}$ to $f(\mathcal{T}) := \{A \equiv C \sqcap f(A) \mid A \equiv C \in \mathcal{T}\}$, where for every $A \in \mathsf{N}_{\text{def}}^{\mathcal{T}}$, the concept description $f(A)$ is defined as follows.

$$f(A) := \bigsqcap_{P \in \{P' \in \mathsf{N}_{\text{prim}}^{\mathcal{F}} \mid A \sqsubseteq_{\mathcal{F} \cup \mathcal{T}} P'\}} P \sqcap \bigsqcap_{r \in \mathsf{N}_{\text{role}}^{\mathcal{T}}} \bigsqcap_{Q \in \{Q' \in \mathsf{N}_{\text{prim}}^{\mathcal{F}} \mid A \sqsubseteq_{\mathcal{F} \cup \mathcal{T}} \exists r.Q'\}} \exists r.A_Q \ .$$

Note that $f(\mathcal{T})$ is still a normalized $\mathcal{EL}$-TBox. To preserve normalization, $f(A)$ adds $\exists r.A_Q$ instead of $\exists r.Q$ whenever A implies $\exists r.Q$. It has been shown in [19] that the above reduction yields a cyclic TBox equivalent to the original hybrid one in the following sense.

Theorem 2. *Let* $(\mathcal{F}, \mathcal{T})$ *be a normalized hybrid* $\mathcal{EL}$*-TBox and* $A, B \in \mathsf{N}_{\text{def}}^{\mathcal{T}}$*. Then,* $A \sqsubseteq_{\text{gfp},\mathcal{F},\mathcal{T}} B$ *iff* $A \sqsubseteq_{\text{gfp},f(\mathcal{T})} B$*.*

It has been shown in [19] that subsumption w.r.t. hybrid $\mathcal{EL}$-TBoxes can be decided in polynomial time in the size of the hybrid TBox.

In the following sections, we introduce matching problems w.r.t. cyclic and hybrid TBoxes and present appropriate matching algorithms for both cases.

3 Matching w.r.t. Cyclic $\mathcal{EL}$-TBoxes

Our first step towards defining matching w.r.t. cyclic TBoxes is to extend concept descriptions to concept patterns by admitting concept variables.

Denote by N_{var} a finite set of *variables* pairwise disjoint to N_{con} and N_{role}. The set of *concept patterns* over N_{con}, N_{role}, and N_{var} is inductively defined as follows: every $\mathcal{EL}$-concept description over N_{con} and N_{role} is a concept pattern; every variable $X \in \mathsf{N}_{\text{var}}$ is a concept pattern; and if $r \in \mathsf{N}_{\text{role}}$ and D_1, D_2 are concept patterns then so are $D_1 \sqcap D_2$ and $\exists r.D_1$.

Trivially, every concept description is a concept pattern. The following definition similarly extends cyclic TBoxes to pattern TBoxes in which the right-hand side of a definition may be a concept pattern.

Definition 2. *(Pattern TBox) An* $\mathcal{EL}$-pattern TBox $\mathcal{T}$ *is a finite set of definitions of the form* $A \equiv C$*, where* $A \in \mathsf{N}_{\text{def}}$ *and* C *is a concept pattern over* N_{prim}*,* N_{def}*,* N_{role}*, and* N_{var}*. A is called* defined in $\mathcal{T}$ *and may occur on the left-hand side of no other definition in* $\mathcal{T}$*. Denote by* $\mathsf{N}_{\text{var}}^{\mathcal{T}}$ *the set of all variables occurring in* $\mathcal{T}$*.*

Note that variables do not occur on left-hand sides of definitions. Denote by $\mathsf{N}_{\text{var}}^{\mathcal{T}}(A)$ the set of variables in $\mathcal{T}$ 'reachable' from A. Matching problems over cyclic TBoxes can now be defined as follows.

Definition 3. *(Matching problem) Let* $\mathcal{T}$ *be an* $\mathcal{EL}$*-pattern TBox with* $A, B \in \mathsf{N}_{\text{def}}^{\mathcal{T}}$*. Moreover, let* $\mathsf{N}_{\text{var}}^{\mathcal{T}}(A) = \emptyset$*. Then* $A \equiv^{?}_{\text{gfp},\mathcal{T}} B$ *is an* $\mathcal{EL}$-matching problem modulo equivalence w.r.t. $\mathcal{T}$ with gfp-semantics.

Throughout this section, we shall refer to '$\mathcal{EL}$-matching problem modulo equivalence with gfp-semantics' by '$\mathcal{EL}$-matching problem'. In order to define solutions to matching problems appropriately, some preparation is necessary. The following definition introduces conservative extensions for pattern TBoxes.

Definition 4. *(Conservative extension) Let* $\mathcal{T}_1$ *be an* $\mathcal{EL}$*-pattern TBox over* N_{prim}*,* N_{def}*,* N_{role}*, and* N_{var}*. Then an* $\mathcal{EL}$*-pattern TBox* $\mathcal{T}_2$ *is a* conservative extension *of* $\mathcal{T}_1$ *iff* $\mathsf{N}_{\text{prim}}^{\mathcal{T}_2} = \mathsf{N}_{\text{prim}}^{\mathcal{T}_1}$*,* $\mathsf{N}_{\text{role}}^{\mathcal{T}_2} = \mathsf{N}_{\text{role}}^{\mathcal{T}_1}$*,* $\mathsf{N}_{\text{var}}^{\mathcal{T}_1} \supseteq \mathsf{N}_{\text{var}}^{\mathcal{T}_2}$*, and* $\mathcal{T}_1 \subseteq \mathcal{T}_2$*.*

The above definition coincides on ordinary TBoxes with the definition of conservative extensions from [18]. Moreover, since $\mathcal{T}_2$ is a pattern TBox, $\mathsf{N}_{\text{def}}^{\mathcal{T}_1}$ and $\mathsf{N}_{\text{def}}^{\mathcal{T}_2 \setminus \mathcal{T}_1}$ are disjoint. In contrast to concept matching (as, e.g., in [24]), we do not use substitutions to instantiate variables. Instead, we simply extend pattern TBoxes by appropriate definitions for the occurring variables. This leads to the notion of *instantiation*.

Definition 5. *(Instantiation) Let $\mathcal{T}_1$ be an $\mathcal{EL}$-pattern TBox over N_{prim}, N_{def}, N_{role}, and N_{var}. Let $\mathcal{T}_2$ be a conservative extension of $\mathcal{T}_1$. For every $X \in \mathsf{N}_{\text{var}}^{\mathcal{T}_1}$, let D_X be a concept pattern over N_{prim}, N_{def}, N_{role}, and $\mathsf{N}_{\text{var}}^{\mathcal{T}_1}$. Then $\mathcal{T}_3 := \mathcal{T}_2 \cup \{X \equiv D_X \mid X \in \mathsf{N}_{\text{var}}^{\mathcal{T}_1}\}$ is an* instantiation *of $\mathcal{T}_1$.*

Intuitively, an instantiation turns variables into defined concepts, and thus turns a pattern TBox into an ordinary TBox. Using these notions, it is particularly simple to define solutions to matching problems.

Definition 6. *(Matcher) Let $A \equiv^?_{\text{gfp},\mathcal{T}} B$ be an $\mathcal{EL}$-matching problem and let $\mathcal{T}'$ be an instantiation of $\mathcal{T}$. Then $\mathcal{T}'$ is a* matcher of $A \equiv^?_{\text{gfp},\mathcal{T}} B$ *iff $A \equiv_{\text{gfp},\mathcal{T}'} B$.*

Hence, a matcher to $A \equiv^?_{\text{gfp},\mathcal{T}} B$ extends the pattern TBox $\mathcal{T}$ by definitions for all variables reachable from B such that A and B become equivalent. Clearly, we may restrict ourselves to matching problems over names because it holds for every concept description C and every concept pattern D defined over a pattern TBox $\mathcal{T}$ that the matching problem $C \equiv^?_{\text{gfp},\mathcal{T}} D$ can be simulated by $A \equiv^?_{\text{gfp},\mathcal{T}\cup\{A\equiv C, B\equiv D\}} B$ with A, B fresh defined names.

We are now ready to show how to solve matching problems w.r.t. cyclic $\mathcal{EL}$-TBoxes as defined above.

3.1 Solving Matching Problems w.r.t. Cyclic $\mathcal{EL}$-TBoxes

By treating variables as primitive concepts, pattern TBoxes can, syntactically, be regarded as ordinary TBoxes. This allows us to define normalized pattern TBoxes analogously to normalized cyclic TBoxes, and to transform pattern TBoxes into description graphs and vice versa. Similarly, we adopt the notion of a product TBox. For an $\mathcal{EL}$-pattern TBox and $n \in \mathbb{N}$, let $\mathcal{T}^n := \mathsf{tbox}(\mathcal{G}_{\mathcal{T}}^n)$. In order to extend the notion of simulation relations to graphs of pattern TBoxes, variables are simply ignored. We can now define our matching algorithm w.r.t. cyclic $\mathcal{EL}$-TBoxes as follows.

Definition 7. *(*match*) Let $\mathcal{T}$ be a normalized $\mathcal{EL}$-pattern TBox and let $A \equiv^?_{\text{gfp},\mathcal{T}} B$ be an $\mathcal{EL}$-matching problem. For every simulation relation $Z \colon \mathcal{G}_{\mathcal{T}} \rightleftharpoons \mathcal{G}_{\mathcal{T}}$ and for every $X \in \mathsf{N}_{\text{var}}^{\mathcal{T}}$, define*

$$Z(X) := \{A' \in \mathsf{N}_{\text{def}} \mid \exists B' \in \mathsf{N}_{\text{def}} \colon (B', A') \in Z \wedge X \in L_{\mathcal{T}}(B')\}.$$

Then, $\mathsf{match}(A \equiv^?_{\text{gfp},\mathcal{T}} B)$ is defined as shown in Figure 1.

Upon input $A \equiv^?_{\text{gfp},\mathcal{T}} B$, our matching algorithm match returns all instantiations $\mathcal{T}_Z$ for which, firstly, Z is a simulation relation on $\mathcal{G}_{\mathcal{T}}$ with $(B, A) \in Z$; and secondly, A subsumes B w.r.t. $\mathcal{T}_Z$ interpreted with gfp-semantics.

For a given Z, $\mathcal{T}_Z$ is defined as an instantiation of a conservative extension of $\mathcal{T}$. We discuss the conservative extension first and the additional definitions for variables afterwards. For every variable $X \in \mathsf{N}_{\text{var}}^{\mathcal{T}}$, $\mathcal{T}$ is extended by the $|Z(X)|$-ary graph product of $\mathcal{T}$. For every X, the set $Z(X)$ contains all 'destination' vertices onto which vertices in $\mathcal{G}_{\mathcal{T}}$ labeled by X are mapped. Hence, whenever

Input: matching problem $\mathcal{P} := A \equiv^{?}_{\mathrm{gfp},\mathcal{T}} B$ with normalized $\mathcal{EL}$-pattern TBox $\mathcal{T}$
Output: set of matchers of $\mathcal{P}$

Return $\{\mathcal{T}_Z \mid Z\colon \mathcal{G}_\mathcal{T} \rightleftharpoons \mathcal{G}_\mathcal{T} \wedge (B, A) \in Z \wedge A \sqsupseteq_{\mathrm{gfp},\mathcal{T}_Z} B\}$,
where, for every $Z\colon \mathcal{G}_\mathcal{T} \rightleftharpoons \mathcal{G}_\mathcal{T}$, $\mathcal{T}_Z$ is defined by:

$$\begin{aligned}\mathcal{T}_Z := \mathcal{T} \cup & \bigcup_{i \in \{|Z(X)| \mid X \in \mathsf{N}^\mathcal{T}_{\mathrm{var}}(B)\} \setminus \{1\}} (\mathcal{T}[X/\top \mid X \in \mathsf{N}^\mathcal{T}_{\mathrm{var}}])^i \\ & \cup \{X \equiv (A_1, \ldots, A_n) \mid X \in \mathsf{N}^\mathcal{T}_{\mathrm{var}}(B) \\ & \qquad\qquad \wedge Z(X) = \{A_1, \ldots, A_n\} \wedge |Z(X)| = n\} \\ & \cup \{X \equiv \top \mid X \notin \mathsf{N}^\mathcal{T}_{\mathrm{var}}(B)\}.\end{aligned}$$

Fig. 1. The algorithm match for cyclic $\mathcal{EL}$-TBoxes

Z maps vertices labeled by X onto n different vertices then $\mathcal{T}$ is extended by the n-ary graph product of $\mathcal{T}$. More precisely, the graph product is computed after removing variables from $\mathcal{T}$. Note that this removal is only done for convenience to simplify the notation in our proofs and not necessary for correctness or completeness of the algorithm.

As a result, the relevant conservative extension of $\mathcal{T}$ for every X contains a definition of the lcs over all destination vertices of vertices labeled by X: if $Z(X)$ contains n pairwise distinct destination vertices $\{A_1, \ldots, A_n\}$ then the relevant lcs is the vertex $(A_1, \ldots, A_n)$ in the n-ary product of $\mathcal{T}$.

As the second line of the definition of $\mathcal{T}_Z$ shows, X is finally assigned the lcs over all destinations of X: $X \equiv (A_1, \ldots, A_n)$. Note that the condition $|Z(X)| = n$ only ensures pairwise distinctness of the vertices $A_1, \ldots, A_n$. Without this condition, X might be assigned to vertices not existing in the relevant extension. Note also that variables unreachable from B are assigned $\top$.

In order to get an impression how the above matching algorithm works, see our example in [20].

We can show that the above algorithm is sound and complete and that the set of all matchers of a given matching problem can be computed in exponential time, see [20] for details. More precisely, we show that our matching algorithm is *s-complete*. Intuitively, this means that the set of matchers computed by the algorithm contains all 'interesting' solutions which contain as much information about the input matching problem as possible; see [20] for details.

In addition to that, we obtain that our matching algorithm for cyclic $\mathcal{EL}$-TBoxes with greatest-fixpoint semantics generalizes the $\mathcal{EL}$-matching algorithm w.r.t. the empty TBox presented in [24]. This immediately implies several complexity lower bounds: Firstly, deciding the solvability of matching problems modulo equivalence w.r.t. cyclic $\mathcal{EL}$-TBoxes is NP-hard. Secondly, the minimal matchers to matching problems w.r.t. cyclic $\mathcal{EL}$-TBoxes can be of exponential size in the input TBox. Moreover, the number of minimal matchers can also be exponential in the input TBox. Any algorithm solving matching problems

w.r.t. cyclic $\mathcal{EL}$-TBoxes is therefore necessarily worst-case exponential. In this sense, our algorithm is worst-case optimal. It is open whether deciding the solvability of matching problems modulo equivalence w.r.t. cyclic $\mathcal{EL}$-TBoxes with gfp-semantics is in NP.

4 Matching w.r.t. Hybrid TBoxes

The main ingredient of the matching algorithm presented in the previous section has been the gfp-lcs w.r.t. cyclic $\mathcal{EL}$-TBoxes with gfp-semantics from [18]. Our aim now is to extend the algorithm from cyclic to hybrid TBoxes. We begin by extending the notion of a pattern TBox from Definition 2 to hybrid TBoxes.

Definition 8. *(Hybrid pattern TBox) A hybrid* $\mathcal{EL}$-pattern TBox $\mathcal{T}$ *is a pair* $(\mathcal{F}, \mathcal{T})$ *of a general* $\mathcal{EL}$*-TBox* $\mathcal{F}$ *defined over* N_{prim} *and* N_{role}*, and an* $\mathcal{EL}$*-pattern TBox defined over* N_{def}*,* N_{prim}*, and* N_{role}*.*

Hence, hybrid pattern TBoxes extend ordinary pattern TBoxes by adding a 'foundation' general TBox. Conservative extensions and instantiations of hybrid pattern TBoxes are defined analogous to their counterpart cyclic TBoxes, i.e., they affect only $\mathcal{T}$ and leave $\mathcal{F}$ unchanged. We can now immediately extend the notion of matching problems to hybrid pattern TBoxes.

Definition 9. *(Matching problem) Let* $(\mathcal{F}, \mathcal{T})$ *be a hybrid* $\mathcal{EL}$*-pattern TBox with* $A, B \in \mathsf{N}_{\text{def}}^{\mathcal{T}}$*. Moreover, let* $\mathsf{N}_{\text{var}}^{\mathcal{T}}(A) = \emptyset$*. Then* $A \equiv^{?}_{\text{gfp},\mathcal{F},\mathcal{T}} B$ *is a* hybrid $\mathcal{EL}$-matching problem modulo equivalence w.r.t. $(\mathcal{F}, \mathcal{T})$*.*

Note that, despite the restriction of A to defined concept names from $\mathcal{T}$, concept patterns can also be matched against concept names defined in $\mathcal{F}$. For instance, in order to match a concept pattern B defined in $\mathcal{T}$ against some $P \in \mathsf{N}_{\text{con}}^{\mathcal{T}}$ from $\mathcal{F}$, it suffices to extend $\mathcal{T}$ by a definition of the form $A_P \equiv P$, with A_P a fresh concept name, and solve the matching problem $A_P \equiv^{?}_{\text{gfp},\mathcal{F},\mathcal{T}} B$. Clearly, one can also define concept patterns using only names from $\mathcal{F}$.

Solutions to hybrid $\mathcal{EL}$-matching problems can now be defined analogous to matchers for matching problems w.r.t. cyclic TBoxes.

Definition 10. *(Matcher) Let* $A \equiv^{?}_{\text{gfp},\mathcal{F},\mathcal{T}} B$ *be a hybrid* $\mathcal{EL}$*-matching problem and let* $(\mathcal{F}, \mathcal{T}')$ *be an instantiation of* $(\mathcal{F}, \mathcal{T})$*. Then* $(\mathcal{F}, \mathcal{T}')$ *is a* matcher of $A \equiv^{?}_{\text{gfp},\mathcal{F},\mathcal{T}} B$ *iff* $A \equiv_{\text{gfp},\mathcal{F},\mathcal{T}'} B$*.*

In preparation to solving matching problems w.r.t. hybrid TBoxes, we extend the lcs algorithm to hybrid TBoxes in the following section. In Section 4.2, the actual matching algorithm for hybrid TBoxes is presented.

4.1 The Least-Common Subsumer w.r.t. Hybrid $\mathcal{EL}$-TBoxes

Our aim is to extend the lcs w.r.t. cyclic $\mathcal{EL}$-TBoxes to hybrid $\mathcal{EL}$-TBoxes. To this end, we begin by extending the notion of conservative extensions of $\mathcal{EL}$-TBoxes from cyclic to hybrid TBoxes. A hybrid TBox $(\mathcal{F}, \mathcal{T}_2)$ is a conservative extension

of $(\mathcal{F}, \mathcal{T}_1)$ iff $\mathcal{T}_2$ is a conservative extension of $\mathcal{T}_1$ in the sense of Definition 4. Hence, a conservative extension of $(\mathcal{F}, \mathcal{T})$ is obtained by fixing $\mathcal{F}$ and extending $\mathcal{T}$ in the usual way. We can now define the lcs w.r.t. hybrid TBoxes analogously to the case of cyclic ones.

Definition 11. *(Hybrid lcs) Let $(\mathcal{F}, \mathcal{T}_1)$ be a hybrid TBox and $A, B \in \mathsf{N}_{\mathrm{def}}^{\mathcal{T}_1}$. Let $(\mathcal{F}, \mathcal{T}_2)$ be a conservative extension of $(\mathcal{F}, \mathcal{T}_1)$ with $C \in \mathsf{N}_{\mathrm{def}}^{\mathcal{T}_2}$. Then, C in $(\mathcal{F}, \mathcal{T}_2)$ is the* hybrid least-common subsumer (lcs) of A, B in $(\mathcal{F}, \mathcal{T}_1)$ *iff the following conditions hold.*

1. $A \sqsubseteq_{\mathrm{gfp},\mathcal{F},\mathcal{T}_2} C$ *and* $B \sqsubseteq_{\mathrm{gfp},\mathcal{F},\mathcal{T}_2} C$*; and*
2. *If $(\mathcal{F}, \mathcal{T}_3)$ is a conservative extension of $(\mathcal{F}, \mathcal{T}_2)$ and $D \in \mathsf{N}_{\mathrm{def}}^{\mathcal{T}_3}$ such that $A \sqsubseteq_{\mathrm{gfp},\mathcal{F},\mathcal{T}_3} D$ and $B \sqsubseteq_{\mathrm{gfp},\mathcal{F},\mathcal{T}_3} D$ then $C \sqsubseteq_{\mathrm{gfp},\mathcal{F},\mathcal{T}_3} D$.*

In order to compute the lcs w.r.t. hybrid $\mathcal{EL}$-TBoxes, we again utilize the reduction from hybrid to cyclic TBoxes from [19] and the usual gfp-lcs algorithm for cyclic $\mathcal{EL}$-TBoxes from [18]. We show in [20] that the hybrid lcs algorithm thus obtained in fact yields the correct results: (A, B) in $(\mathcal{F}, f(\mathcal{T}) \cup f(\mathcal{T})^2)$ is the hybrid lcs of any concepts A, B defined in a given hybrid TBox $(\mathcal{F}, \mathcal{T})$.

As the lcs of arbitrary arity can be reduced to the binary lcs, the above results immediately carry over to the n-ary lcs. As the reduction from hybrid to cyclic $\mathcal{EL}$-TBoxes can be computed in polynomial time and as the lcs algorithm for cyclic $\mathcal{EL}$-TBoxes with gfp-semantics has already been studied [18], we find that the lcs of concepts defined in a hybrid TBox $(\mathcal{F}, \mathcal{T})$ always exists and (in the binary case) can be computed in polynomial time in the size of $(\mathcal{F}, \mathcal{T})$. Moreover, the lcs of arbitrary arity w.r.t. hybrid $\mathcal{EL}$-TBoxes can be computed in exponential time in the size of the input and is of exponential size in the size of the input in the worst-case. In particular, our lcs algorithm is worst-case optimal.

4.2 Solving Matching Problems w.r.t. Hybrid $\mathcal{EL}$-TBoxes

We are now prepared to introduce our matching algorithm for hybrid TBoxes.

Definition 12. *($\mathsf{match}_{\mathrm{hy}}$) Let $(\mathcal{F}, \mathcal{T})$ be a normalized hybrid $\mathcal{EL}$-TBox and let $A \equiv^{?}_{\mathrm{gfp},\mathcal{F},\mathcal{T}} B$ be a hybrid $\mathcal{EL}$-matching problem. Then define*

$$\mathsf{match}_{\mathrm{hy}}(A \equiv_{\mathrm{gfp},\mathcal{F},\mathcal{T}} B) := \{(\mathcal{F}, (\mathcal{T}' \setminus f(\mathcal{T})) \cup \mathcal{T}) \mid \mathcal{T}' \in \mathsf{match}(A \equiv_{\mathrm{gfp},f(\mathcal{T})} B)\}.$$

In the above definition, $f(\mathcal{T})$ denotes the $\mathcal{F}$-completion of $\mathcal{T}$ from Section 2.1 and match the matching algorithm for cyclic $\mathcal{EL}$-TBoxes from Definition 7. Hence, the algorithm $\mathsf{match}_{\mathrm{hy}}$ proceeds in three main steps. Firstly, the input hybrid pattern TBox $(\mathcal{F}, \mathcal{T})$ is translated into an equivalent[1] cyclic pattern TBox $f(\mathcal{T})$. Secondly, for the translated matching problem $A \equiv_{\mathrm{gfp},f(\mathcal{T})} B$, the algorithm match computes all minimal solutions and returns them in the form of instantiations $\mathcal{T}'$ of $f(\mathcal{T})$. Thirdly, the solution is returned as a set of instantiations of *hybrid* pattern TBoxes. How exactly these hybrid instantiations are defined deserves a closer look.

[1] Treating variables as atomic concepts.

As every instantiation $\mathcal{T}'$ returned by the algorithm match is a conservative extension of $f(\mathcal{T})$ and not $\mathcal{T}$, $\mathcal{T}'$ already completely specifies a solution to the initial hybrid matching problem. Or, in other words, $\mathcal{F}$ becomes redundant. As we are interested in *hybrid* instantiations of $(\mathcal{F}, \mathcal{T})$, and not of $(\mathcal{F}, f(\mathcal{T}))$, we modify every $\mathcal{T}'$ by removing $f(\mathcal{T})$ and replacing it by the original TBox $\mathcal{T}$, i.e., compute $(\mathcal{T}' \setminus f(\mathcal{T})) \cup \mathcal{T}$. This modification preserves equivalence as a direct consequence of the correctness of the $\mathcal{F}$-completion shown in [19]. Together with the correctness of our hybrid lcs algorithm, we immediately obtain soundness and completeness of the hybrid matching algorithm.

Corollary 1. *Let $(\mathcal{F}, \mathcal{T})$ be a normalized hybrid $\mathcal{EL}$-TBox and let $A \equiv_{\mathrm{gfp},\mathcal{F},\mathcal{T}} B$ be an $\mathcal{EL}$-matching problem w.r.t. $(\mathcal{F}, \mathcal{T})$. Then, $\mathsf{match}_{\mathrm{hy}}(A \equiv_{\mathrm{gfp},\mathcal{F},\mathcal{T}} B)$ computes an s-complete set of matchers to $A \equiv_{\mathrm{gfp},\mathcal{F},\mathcal{T}} B$.*

The complexity results obtained in the previous section together with the fact that $f(\mathcal{T})$ can be computed in polynomial time in the size of $(\mathcal{F}, \mathcal{T})$ [19] imply the following complexity results: Deciding the solvability of matching problems modulo subsumption w.r.t. hybrid $\mathcal{EL}$-TBoxes is tractable. Deciding the solvability of matching problems modulo equivalence w.r.t. hybrid $\mathcal{EL}$-TBoxes is NP-hard. The solutions to a matching problem w.r.t. hybrid $\mathcal{EL}$-TBoxes can be exponential in number and of exponential size in the input matching problem. They can be computed by a deterministic exponential-time algorithm. The computation algorithm is worst-case optimal. See [20] for details.

It is open whether deciding the solvability of matching problems modulo equivalence w.r.t. hybrid $\mathcal{EL}$-TBoxes is in NP. Note that that additional rewriting might be desirable in order to present the solutions of $\mathsf{match}_{\mathrm{hy}}$ more succinctly: $\mathcal{T}'$ can contain the n-ary product of $f(\mathcal{T})$ which might contain information already implied by $\mathcal{F}$.

5 Conclusion and Outlook

In the present paper, we have proposed the notion of matching problems in cyclic $\mathcal{EL}$-TBoxes with gfp-semantics and have devised a sound and s-complete exponential time algorithm for that case. Using an existing reduction from hybrid $\mathcal{EL}$-TBoxes to cyclic ones, we have shown that the lcs w.r.t. hybrid $\mathcal{EL}$-TBoxes always exists and have devised a sound and complete exponential time algorithm to compute it. Utilizing both the reduction and the result on the hybrid lcs, we could devise a sound and complete exponential time matching algorithm for matching problems w.r.t. hybrid TBoxes. All computation algorithms are worst-case optimal. Optimality of the relevant algorithms for the decision problem, i.e., existence of a matcher, remains an open problem.

Apart from the fact that reasoning over $\mathcal{EL}$-TBoxes has an attractive computational complexity, ontologies based on $\mathcal{EL}$-TBoxes are of some significance to the life sciences. For instance, the widely used medical terminology SNOMED [27] corresponds to an $\mathcal{EL}$-Tbox [28]. Similarly, the Gene Ontology [29] can be represented by an $\mathcal{EL}$-TBox with one transitive role, and large parts of the medical

knowledge base GALEN [30] can be expressed by a general $\mathcal{EL}$-TBox with transitive roles. Similarly, the widely used International Classification for Nursing Practice (ICNP) [31] corresponds to a general $\mathcal{EL}$-TBox.

Matching in general $\mathcal{EL}$-TBoxes: the apparent popularity of 'common' general $\mathcal{EL}$-TBoxes motivates the question to which extent the above results have any potential to be used for that KR formalism.

It has been shown in [18] that the least-common subsumer w.r.t. cyclic $\mathcal{EL}$-TBoxes with descriptive semantics need not exist[2], a result that carries over to general $\mathcal{EL}$-TBoxes. Moreover, as every lcs can be expressed as a minimal solution to some matching problem, minimal matchers need not always exist likewise.

On the other hand, we have pointed out in Section 2.1 that every general $\mathcal{EL}$-TBox $\mathcal{T}$ can be viewed as a hybrid TBox $(\mathcal{T}, \emptyset)$ with empty terminology. Hence, we can define matching problems in general TBoxes (with descriptive semantics) and use our hybrid matching algorithm to compute a set of solutions S with gfp-semantics. As descriptive subsumption entails gfp-subsumption, every 'descriptive' solution to the matching problem is obtained by a gfp-matching algorithm. All matchers w.r.t. descriptive semantics can thus be computed by first computing S with our hybrid matching algorithm and then removing every matcher from S that is not valid w.r.t. descriptive semantics.

The pure decision problem for general TBoxes might be even more interesting for our hybrid matching algorithm. As pointed out in [17], matching can be utilized as a retrieval mechanism over TBoxes in a straightforward way. The user specifies a concept pattern with the syntactic structure he has in mind. The matching algorithm is then used to retrieve all concepts in the TBox for which a matcher exists. The fact that variables in concept patterns are named, in contrast to, e.g., wildcards ('$*$') known from standard database queries, allows us to search the TBox for concepts with very specific structural properties.

In the application scenario sketched above, two ways of dealing with 'descriptive' results suggest themselves. The first option is to solve the full computation problem in the background and return only those concepts for which the matcher is also valid with descriptive semantics. Queries of the above kind, however, are motivated by structural properties of concepts defined in the TBox. Therefore, a viable second option might be to just present all solutions retrieved with gfp-semantics.

In order to substantiate the claim that the above query mechanism driven by our hybrid matching algorithm is useful for the task of knowledge engineering, we plan to implement our matching algorithm as a plugin to the widely used ontology editor PROTÉGÉ [32]. One way to achieve this might be to integrate the query functionality into the system SONIC [33], a plug-in specifically designed for the purpose to bring non-standard inferences to users of PROTÉGÉ.

[2] Nevertheless, the existence of the lcs under these circumstances is decidable, see [26].

Acknowledgements

We would like to thank Hongkai Liu for his valuable contribution to the results for matching in cyclic $\mathcal{EL}$-TBoxes.

References

1. Nardi, D., Brachmann, R.J.: An introduction to description logics. In: The Description Logic Handbook: Theory, Implementation, and Applications, pp. 1–40. Cambridge University Press, Cambridge (2003)
2. Rector, A., Nowlan, W., Glowinski, A.: Goals for concept representation in the GALEN project. In: Proc. of SCAMC, Washington, USA, pp. 414–418 (1993)
3. Rector, A.: Medical informatics. In: The Description Logic Handbook: Theory, Implementation, and Applications, pp. 406–426. Cambridge University Press, Cambridge (2003)
4. Horrocks, I., Rector, A.L., Goble, C.A.: A description logic based schema for the classification of medical data. In: Proc. of KRDB 1996 (1996)
5. Horrocks, I.: Using an expressive description logic: FaCT or fiction? In: Proc. of KR 1998, pp. 636–645. Morgan-Kaufmann Publishers, San Francisco (1998)
6. Haarslev, V., Möller, R.: Racer system description. In: Goré, R.P., Leitsch, A., Nipkow, T. (eds.) IJCAR 2001. LNCS (LNAI), vol. 2083, pp. 701–712. Springer, Heidelberg (2001)
7. Sirin, E., Parsia, B.: Pellet: An OWL DL reasoner. In: Proc. of DL 2004. CEUR-WS (2004) Proceedings (2004), http://CEUR-WS.org/Vol-104/
8. Baader, F., Lutz, C., Suntisrivaraporn, B.: CEL—a polynomial-time reasoner for life science ontologies. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, pp. 287–291. Springer, Heidelberg (2006)
9. Cohen, W.W., Borgida, A., Hirsh, H.: Computing least common subsumers in description logics. In: Proc. of AAAI 1992, pp. 754–760. The MIT Press, USA (1992)
10. Cohen, W.W., Hirsh, H.: The learnability of description logics with equality constraints. Machine Learning 17(2/3), 169–199 (1994) Special Issue for COLT 1992.
11. Frazier, M., Pitt, L.: CLASSIC learning. Machine Learning 25, 151–193 (1996) Was in COLT 1994.
12. Baader, F., Küsters, R.: Computing the least common subsumer and the most specific concept in the presence of cyclic $\mathcal{ALN}$-concept descriptions. In: Herzog, O. (ed.) KI 1998. LNCS (LNAI), vol. 1504, pp. 129–140. Springer, Heidelberg (1998)
13. McGuinness, D.: Explaining Reasoning in Description Logics. Ph.D. dissertation, Department of Computer Science, Rutgers University, USA (1996)
14. Borgida, A., McGuinness, D.L.: Asking queries about frames. In: Proc. of KR 1996, pp. 340–349. Morgan-Kaufmann Publishers, San Francisco (1996)
15. Baader, F., Küsters, R., Borgida, A., McGuinness, D.: Matching in description logics. Journal of Logic and Computation 9(3), 411–447 (1999)
16. Baader, F., Küsters, R., Molitor, R.: Computing least common subsumers in description logics with existential restrictions. In: Proc. of IJCAI 1999, pp. 96–101. Morgan-Kaufmann Publishers, San Francisco (1999)
17. Brandt, S., Turhan, A.Y.: Using non-standard inferences in description logics — what does it buy me? In: Proc. of KIDLWS 2001. CEUR-WS (September 2001) Proceedings online available from http://CEUR-WS.org/Vol-44/

18. Baader, F.: Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles. In: Proc. of IJCAI 2003, pp. 319–324. Morgan-Kaufmann Publishers, San Francisco (2003)
19. Brandt, S., Model, J.: Subsumption in $\mathcal{EL}$ w.r.t. hybrid TBoxes. In: Furbach, U. (ed.) KI 2005. LNCS (LNAI), vol. 3698, pp. 34–48. Springer, Heidelberg (2005)
20. Brandt, S.: Matching and general concept inclusion axioms. Technical report (2007), See `http://personalpages.manchester.ac.uk/staff/Sebastian-philipp.Brandt/tr0707.pdf`
21. Baader, F.: Terminological cycles in a description logic with existential restrictions. In: Proc. of IJCAI 2003, pp. 325–330. Morgan-Kaufmann Publishers, San Francisco (2003)
22. Nebel, B.: Terminological cycles: Semantics and computational properties. In: Proc. of Principles of Semantic Networks, pp. 331–361. Morgan Kaufmann, San Francisco (1991)
23. Tarski, A.: A lattice-theoretic fixpoint theorem and its applications. Pacific Journal of Mathematics 5(2), 285–309 (1955)
24. Baader, F., Küsters, R.: Matching in description logics with existential restrictions. In: Proc. of KR 2000, pp. 261–272. Morgan-Kaufmann Publishers, San Francisco (2000)
25. Küsters, R.: Non-Standard Inferences in Description Logics. In: Küsters, R. (ed.) Non-Standard Inferences in Description Logics. LNCS (LNAI), vol. 2100, Springer, Heidelberg (2001)
26. Baader, F.: A graph-theoretic generalization of the least common subsumer and the most specific concept in the description logic $\mathcal{EL}$. In: Hromkovič, J., Nagl, M., Westfechtel, B. (eds.) WG 2004. LNCS, vol. 3353, pp. 177–188. Springer, Heidelberg (2004)
27. Côté, R., Rothwell, D., Palotay, J., Beckett, R., Brochu, L.: The systematized nomenclature of human and veterinary medicine. Technical report, SNOMED International, Northfield, IL (1993)
28. Spackman, K.: Normal forms for description logic expressions of clinical concepts in SNOMED RT. Journal of the American Medical Informatics Association (Symposium Supplement) (2001)
29. Consortium, T.G.O.: Gene Ontology: Tool for the unification of biology. Nature Genetics 25, 25–29 (2000)
30. Rector, A., Bechhofer, S., Goble, C.A., Horrocks, I., Nowlan, W.A., Solomon, W.D.: The GRAIL concept modelling language for medical terminology. Artificial Intelligence in Medicine 9, 139–171 (1997)
31. International council of Nurses, Geneva, CH. See `http://www.icn.ch/icnp.html`
32. Horridge, M., Tsarkov, D., Redmond, T.: Supporting early adoption of OWL 1.1 with Protege-OWL and FaCT++. In: Proc. of OWL-ED 2006 (2006)
33. Turhan, A.Y.: Pushing the SONIC border—SONIC 1.0. In: Proc. of FTP 2005. Technical Report, University of Koblenz (2005)