Matching with respect to general concept inclusions in the Description Logic $\mathcal{EL}$

Franz Baader and Barbara Morawska[*]
{baader,morawska}@tcs.inf.tu-dresden.de

Theoretical Computer Science, TU Dresden, Germany

Abstract Matching concept descriptions against concept patterns was introduced as a new inference task in Description Logics (DLs) almost 20 years ago, motivated by applications in the Classic system. For the DL $\mathcal{EL}$, it was shown in 2000 that the matching problem is NP-complete. It then took almost 10 years before this NP-completeness result could be extended from matching to unification in $\mathcal{EL}$. The next big challenge was then to further extend these results from matching and unification without a TBox to matching and unification w.r.t. a general TBox, i.e., a finite set of general concept inclusions. For unification, we could show some partial results for general TBoxes that satisfy a certain restriction on cyclic dependencies between concepts, but the general case is still open. For matching, we were able to solve the general case: we can show that matching in $\mathcal{EL}$ w.r.t. general TBoxes is NP-complete. We also determine some tractable variants of the matching problem.

1 Introduction

The DL $\mathcal{EL}$, which offers the constructors conjunction ($\sqcap$), existential restriction ($\exists r.C$), and the top concept ($\top$), has recently drawn considerable attention since, on the one hand, important inference problems such as the subsumption problem are polynomial in $\mathcal{EL}$, even in the presence of general concept inclusions (GCIs) [11]. On the other hand, though quite inexpressive, $\mathcal{EL}$ can be used to define biomedical ontologies, such as the large medical ontology SNOMED CT.[1]

Matching of concept descriptions against concept patterns is a non-standard inference task in Description Logics, which was originally motivated by applications of the Classic system [8]. In [10], Borgida and McGuinness proposed matching as a means to filter out the unimportant aspects of large concept descriptions appearing in knowledge bases of Classic. Subsequently, matching (as well as the more general problem of unification) was also proposed as a tool for detecting redundancies in knowledge bases [7] and to support the integration of knowledge bases by prompting possible interschema assertions to the integrator [9].

All three applications have in common that one wants to search the knowledge base for concepts having a certain (not completely specified) form. This

[*] Supported by DFG under grant BA 1122/14-2

[1] see http://www.ihtsdo.org/snomed-ct/

"form" can be expressed with the help of so-called *concept patterns*, i.e., concept descriptions containing variables (which stand for descriptions). For example, assume that we want to find concepts that are concerned with individuals having a son and a daughter sharing some characteristic. This can be expressed by the pattern $D := \exists \mathsf{has\text{-}child}.(\mathsf{Male} \sqcap X) \sqcap \exists \mathsf{has\text{-}child}.(\mathsf{Female} \sqcap X)$, where X is a variable standing for the common characteristic. The concept description $C := \exists \mathsf{has\text{-}child}.(\mathsf{Tall} \sqcap \mathsf{Male}) \sqcap \exists \mathsf{has\text{-}child}.(\mathsf{Tall} \sqcap \mathsf{Female})$ matches this pattern in the sense that, if we replace the variable X by the description Tall, the pattern becomes *equivalent* to the description. Thus, the substitution $\sigma := \{X \mapsto \mathsf{Tall}\}$ is a *matcher modulo equivalence* of the matching problem $C \equiv^? D$ since $C \equiv \sigma(D)$. The original paper by Borgida and McGuinness actually considered matching modulo subsumption rather than matching modulo equivalence: such a problem is of the form $C \sqsubseteq^? D$, and a matcher τ is a substitution τ satisfying $C \sqsubseteq \tau(D)$. Obviously, any matcher modulo equivalence is also a matcher modulo subsumption, but not vice versa. For example, the substitution $\sigma_\top := \{X \mapsto \top\}$ is a *matcher modulo subsumption* of the matching problem $C \sqsubseteq^? D$, but it is not a matcher modulo equivalence.

The first results on matching in DLs were concerned with sublanguages of the Classic description language, which does not allow for existential restrictions of the kind used in our example. A polynomial-time algorithm for computing matchers modulo subsumption for a rather expressive DL was introduced in [10]. The main drawback of this algorithm was that it required the concept patterns to be in structural normal form, and thus it was not able to handle arbitrary matching problems. In addition, the algorithm was incomplete, i.e., it did not always find a matcher, even if one existed. For the DL $\mathcal{ALN}$, a polynomial-time algorithm for matching modulo subsumption and equivalence was presented in [5]. This algorithm is complete and it applies to arbitrary patterns. In [4], matching in DLs with existential restrictions was investigated for the first time. In particular, it was shown that in $\mathcal{EL}$ the matching problem (i.e., the problem of deciding whether a given matching problem has a matcher or not) is polynomial for matching modulo subsumption, but NP-complete for matching modulo equivalence.

Unification is a generalization of matching where both sides of the problem are patterns and thus the substitution needs to be applied to both sides. In [7] it was shown that the unification problem in the DL $\mathcal{FL}_0$, which offers the constructors conjunction ($\sqcap$), value restriction ($\forall r.C$), and the top concept ($\top$), is ExpTime-complete. In contrast, unification in $\mathcal{EL}$ is "only" NP-complete [6]. In the results for matching and unification mentioned until now, there was no TBox involved, i.e., equivalence and subsumption was considered with respect to the empty TBox. For unification in $\mathcal{EL}$, first attempts were made to take *general TBoxes*, i.e., finite sets of general concept inclusions (GCIs), into account. However, the results obtained so far, which are again NP-completeness results, are restricted to general TBoxes that satisfy a certain restriction on cyclic dependencies between concepts [2,3].

For matching, we were able to solve the general case: matching in $\mathcal{EL}$ w.r.t. general TBoxes is NP-complete. The matching problems considered in this paper are actually generalizations of matching modulo equivalence and matching modulo subsumption. For the special case of matching modulo subsumption, we show that the problem is tractable also in the presence of GCIs. The same is true for the dual problem where the pattern is on the side of the subsumee rather than on the side of the subsumer.

Due to space constraints, we cannot provide proofs of our results. They can be found in [1].

2 The Description Logics $\mathcal{EL}$

The expressiveness of a DL is determined both by the formalism for describing concepts (the concept description language) and the terminological formalism, which can be used to state additional constraints on the interpretation of concepts and roles in a so-called TBox.

The *concept description language* considered in this paper is called $\mathcal{EL}$. Starting with a finite set N_C of *concept names* and a finite set N_R of *role names*, *$\mathcal{EL}$-concept descriptions* are built from concept names using the constructors *conjunction* ($C \sqcap D$), *existential restriction* ($\exists r.C$ for every $r \in N_R$), and *top* ($\top$). Since in this paper we only consider $\mathcal{EL}$-concept descriptions, we will sometimes dispense with the prefix $\mathcal{EL}$.

On the *semantic side*, concept descriptions are interpreted as sets. To be more precise, an *interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consists of a non-empty domain $\Delta^\mathcal{I}$ and an interpretation function $\cdot^\mathcal{I}$ that maps concept names to subsets of $\Delta^\mathcal{I}$ and role names to binary relations over $\Delta^\mathcal{I}$. This function is inductively extended to concept descriptions as follows:

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

A *general concept inclusion axiom (GCI)* is of the form $C \sqsubseteq D$ for concept descriptions C, D. An interpretation $\mathcal{I}$ *satisfies* such an axiom $C \sqsubseteq D$ iff $C^\mathcal{I} \subseteq D^\mathcal{I}$. A *general $\mathcal{EL}$-TBox* is a finite set of GCIs. An interpretation is a *model* of a general $\mathcal{EL}$-TBox if it satisfies all its GCIs.

A concept description C is *subsumed* by a concept description D w.r.t. a general TBox $\mathcal{T}$ (written $C \sqsubseteq_\mathcal{T} D$) if every model of $\mathcal{T}$ satisfies the GCI $C \sqsubseteq D$. We say that C is *equivalent* to D w.r.t. $\mathcal{T}$ ($C \equiv_\mathcal{T} D$) if $C \sqsubseteq_\mathcal{T} D$ and $D \sqsubseteq_\mathcal{T} C$. If $\mathcal{T}$ is empty, we also write $C \sqsubseteq D$ and $C \equiv D$ instead of $C \sqsubseteq_\mathcal{T} D$ and $C \equiv_\mathcal{T} D$, respectively. As shown in [11], subsumption w.r.t. general $\mathcal{EL}$-TBoxes is decidable in polynomial time.

3 Matching in $\mathcal{EL}$

In addition to the set N_C of concept names (which must not be replaced by substitutions), we introduce a set N_V of concept variables (which may be replaced by substitutions). *Concept patterns* are now built from concept names and concept variables by applying the constructors of $\mathcal{EL}$. A *substitution* σ maps every concept variable to an $\mathcal{EL}$-concept description. It is extended to concept patterns in the usual way:

- $\sigma(A) := A$ for all $A \in N_C \cup \{\top\}$,
- $\sigma(C \sqcap D) := \sigma(C) \sqcap \sigma(D)$ and $\sigma(\exists r.C) := \exists r.\sigma(C)$.

An $\mathcal{EL}$-concept pattern C is *ground* if it does not contain variables, i.e., if it is a concept description. Obviously, a ground concept pattern is not modified by applying a substitution.

Definition 1. *Let $\mathcal{T}$ be a general $\mathcal{EL}$-TBox.*[2] *An $\mathcal{EL}$-*matching problem *w.r.t. $\mathcal{T}$ is a finite set $\Gamma = \{C_1 \sqsubseteq^? D_1, \ldots, C_n \sqsubseteq^? D_n\}$ of subsumptions between $\mathcal{EL}$-concept patterns, where for each $i, 1 \leq i \leq n$, C_i or D_i is ground. A substitution σ is a* matcher *of Γ w.r.t. $\mathcal{T}$ if σ* solves *all the subsumptions in Γ, i.e. if $\sigma(C_1) \sqsubseteq_\mathcal{T} \sigma(D_1), \ldots, \sigma(C_n) \sqsubseteq_\mathcal{T} \sigma(D_n)$. We say that Γ is* matchable *w.r.t. $\mathcal{T}$ if it has a matcher.*

Matching problems modulo equivalence and subsumption are special cases of the matching problems introduced above:

- The $\mathcal{EL}$-matching problem Γ is a *matching problem modulo equivalence* if $C \sqsubseteq D \in \Gamma$ implies $D \sqsubseteq C \in \Gamma$. This coincides with the notion of matching modulo equivalence considered in [5,4], but extended to a non-empty general TBox.
- The $\mathcal{EL}$-matching problem Γ is a *left-ground matching problem modulo subsumption* if $C \sqsubseteq D \in \Gamma$ implies that C is ground. This coincides with the notion of matching modulo subsumption considered in [5,4], but again extended to a non-empty general TBox.
- The $\mathcal{EL}$-matching problem Γ is a *right-ground matching problem modulo subsumption* if $C \sqsubseteq D \in \Gamma$ implies that D is ground. To the best of our knowledge, this notion of matching has not been investigated before.

The general case of matching, as introduced in Definition 1, and thus also matching modulo equivalence, is NP-complete, whereas the two notions of matching modulo subsumption are tractable, even in the presence of GCIs.

Theorem 2. *Let Γ be an $\mathcal{EL}$-matching problem and $\mathcal{T}$ a general $\mathcal{EL}$-TBox. Deciding whether Γ has a matcher w.r.t. $\mathcal{T}$ is*

1. *polynomial if Γ is a left-groud or a right-ground matching problem modulo subsumption;*
2. *NP-complete in the general case.*

[2] Note that the GCIs in $\mathcal{T}$ are built using concept descriptions, and thus do not contain variables.

A detailed proof of this theorem can be found in [1]. Basically, the results for the case of matching modulo subsumption are proved as follows: in each case we define a specific substitution, and show that the matching problem has a matcher iff this substitution is a matcher. NP-hardness for the general case follows from the known NP-hardness result for matching modulo equivalence without a TBox. The NP-upper bound can be shown by introducing a goal-oriented matching algorithm that uses nondeterministic rules to transform a given matching problem into a solved form by a polynomial number of rule applications.

References

1. Baader, F., , Morawska, B.: Matching with respect to general concept inclusions in the description logic $\mathcal{EL}$. LTCS-Report 14-03, Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany (2014), see http://lat.inf.tu-dresden.de/research/reports.html.
2. Baader, F., Borgwardt, S., Morawska, B.: Extending unification in $\mathcal{EL}$ towards general TBoxes. In: Proc. of the 13th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2012). pp. 568–572. AAAI Press/The MIT Press (2012)
3. Baader, F., Borgwardt, S., Morawska, B.: A goal-oriented algorithm for unification in $\mathcal{ELH}_{R^+}$ w.r.t. cycle-restricted ontologies. In: Thielscher, M., Zhang, D. (eds.) Pro. of 25th Australasian Joint Conf. on Artificial Intelligence (AI'12). Lecture Notes in Artificial Intelligence, vol. 7691, pp. 493–504. Springer-Verlag (2012)
4. Baader, F., Küsters, R.: Matching in description logics with existential restrictions. In: Proc. of the 7th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2000). pp. 261–272 (2000)
5. Baader, F., Küsters, R., Borgida, A., McGuinness, D.L.: Matching in description logics. J. of Logic and Computation 9(3), 411–447 (1999)
6. Baader, F., Morawska, B.: Unification in the description logic $\mathcal{EL}$. Logical Methods in Computer Science 6(3) (2010)
7. Baader, F., Narendran, P.: Unification of concept terms in description logics. J. of Symbolic Computation 31(3), 277–305 (2001)
8. Borgida, A., Brachman, R.J., McGuinness, D.L., Alperin Resnick, L.: CLASSIC: A structural data model for objects. In: Proc. of the ACM SIGMOD Int. Conf. on Management of Data. pp. 59–67 (1989)
9. Borgida, A., Küsters, R.: What's not in a name? Initial explorations of a structural approach to integrating large concept knowledge-bases. Tech. Rep. DCS-TR-391, Rutgers University (1999)
10. Borgida, A., McGuinness, D.L.: Asking queries about frames. In: Proc. of the 5th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'96). pp. 340–349 (1996)
11. Brandt, S.: Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In: de Mántaras, R.L., Saitta, L. (eds.) Proc. of the 16th Eur. Conf. on Artificial Intelligence (ECAI 2004). pp. 298–302 (2004)