Matching in the Description Logic $\mathcal{FL}_0$ with respect to General TBoxes* (Extended abstract)

Franz Baader, Oliver Fernández Gil, and Pavlos Marantidis

Theoretical Computer Science, TU Dresden
01062 Dresden, Germany
firstname.lastname@tu-dresden.de

Matching is the special case of unification where one of the expressions to be unified has no variables and thus remains unchanged under substitutions. In Description Logic (DL), matching of concept descriptions against concept patterns was originally introduced in [12] as a non-standard inference task that can be used to filter out the unimportant aspects of large concept descriptions appearing in knowledge bases of the system Classic [10]. Subsequently, matching (as well as the more general problem of unification) was also proposed as a tool for detecting redundancies in knowledge bases [9] and to support the integration of knowledge bases by prompting interschema assertions to the integrator [11].

All three applications have in common that one wants to search the knowledge base for concepts having a certain (not completely specified) form. This "form" can be expressed with the help of so-called *concept patterns*, i.e., concept descriptions containing variables (which stand for concept descriptions). For example, assume that we want to find concepts that are concerned with humans that share some characteristic with all their children. This can be expressed by the pattern $D := \mathsf{Human} \sqcap X \sqcap \forall \mathsf{has\text{-}child}.X$ where X is a variable standing for the common characteristic. The concept description $C := \mathsf{Human} \sqcap \mathsf{Tall} \sqcap \forall \mathsf{has\text{-}child}.\mathsf{Tall}$ matches this pattern in the sense that, if we replace the variable X by the concept description Tall, the pattern becomes *equivalent* to the concept description C. Thus, the substitution $\sigma := \{X \mapsto \mathsf{Tall}\}$ is a *matcher* of the matching problem $C \equiv^? D$ since $C \equiv \sigma(D)$.

Both matching and unification have been investigated in detail for the inexpressive DLs $\mathcal{FL}_0$ (with concept constructors top $\top$, conjunction $C \sqcap D$, value restriction $\forall r.C$) and $\mathcal{EL}$ (with concept constructors $\top$, $C \sqcap D$, existential restriction $\exists r.C$). Whereas in $\mathcal{EL}$ both matching [6] and unification [7] are NP-complete problems, the complexity of these problems differs significantly for $\mathcal{FL}_0$: matching is polynomial, but unification is ExpTime-complete [9]. These results were shown for the case without a background TBox, i.e., a finite set of *general concept inclusions (GCIs)*. For the DL $\mathcal{EL}$ it was proved in [8] that the presence of TBoxes does not change the complexity of the matching problem: it stays in NP. For unification in $\mathcal{EL}$ w.r.t. TBoxes, an NP upper bound could until now only be shown for a restricted form of TBoxes [2].

* Supported by DFG in the RTG 1763 (QuantLA) and grant BA 1122/20-1.

Matching in $\mathcal{FL}_0$ in the presence of TBoxes has not been investigated until now. In this paper, we close this gap by showing that it is an ExpTime-complete problem. Since already subsumption in $\mathcal{FL}_0$ w.r.t. TBoxes is ExpTime-complete [3], ExpTime-hardness of this problem is clear. The first main contribution of this paper is thus to show the ExpTime upper bound. We do this by first showing an ExpTime upper bound for the problem of testing whether an $\mathcal{FL}_0$ matching problem has a matcher in the extended logic $\mathcal{FL}_{reg}$ [1]. Basically, in $\mathcal{FL}_{reg}$ one can use regular languages to express infinite conjunctions of value restrictions. To prove the ExpTime upper bound we bring together the classic matching algorithm of [9] and a characterization of subsumption in the presence of general TBoxes that uses formal languages from [5]. In particular, the authors of [5] demonstrate how $\mathcal{FL}_0$ concept descriptions can be represented by tuples of regular languages, such that subsumption w.r.t. the TBox can be decided by comparing the corresponding tuples componentwise w.r.t. inclusion. For the case of the empty TBox, these languages are finite. The matching algorithm of [9] takes these languages and outputs other finite languages, which are used to construct $\mathcal{FL}_0$ concept descriptions that yield a candidate solution to the matching problem. It is then shown that the matching problem has a matcher iff this candidate solution is one. In the presence of a non-empty TBox, though, the languages obtained by applying the natural generalization of the algorithm in [9] may be infinite, and thus they cannot be used to construct an $\mathcal{FL}_0$ candidate solution. However, they can be used to construct an $\mathcal{FL}_{reg}$ candidate solution; it is *this* candidate that we check for being a solution. The proof that this procedure is in ExpTime depends on a fine-grained analysis of the complexity of subsumption of $\mathcal{FL}_{reg}$ concept descriptions w.r.t. an $\mathcal{FL}_0$ TBox, which uses automata on words and trees. The second step is then to show that an $\mathcal{FL}_0$ matching problem has an $\mathcal{FL}_0$ matcher iff it has an $\mathcal{FL}_{reg}$ matcher, a result obtained by using compactness of first-order logic.

The second main contribution of this paper is to show that the complexity of the matching problem can be lowered from ExpTime to PSpace if one considers TBoxes of a restricted form. Namely, we consider TBoxes containing GCIs where the role depth on the left-hand side of a GCI is not larger than the role depth on the right-hand side. We first introduce an algorithm for checking subsumption in the presence of such TBoxes. The main idea is the following: C is subsumed by D if D "syntactically" occurs in C, or if there exists an intermediate concept E that subsumes C and we can prove it is subsumed by D in a single step using one GCI. This way, a sequence of GCIs is constructed that proves subsumption between C and D. Even though this sequence may be of exponential length, choosing the intermediate concept in a clever way (which is possible because of the special form of the TBox) ensures that the concept to be checked is always of polynomial size, and hence it is possible to perform the check in PSpace. We then derive the matching algorithm in a similar fashion: again, we are looking for the sequence of GCIs proving subsumption, but at every point we check whether it is possible to extend the substitution in order to shorten the search.

The paper containing these results was published at LPAR-22 [4].

References

1. Franz Baader. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91)*, 1991.
2. Franz Baader, Stefan Borgwardt, and Barbara Morawska. Extending unification in $\mathcal{EL}$ towards general TBoxes. In *Proc. of the 13th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2012)*, pages 568–572. AAAI Press/The MIT Press, 2012.
3. Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, pages 364–369, Edinburgh (UK), 2005. Morgan Kaufmann, Los Altos.
4. Franz Baader, Oliver Fernández Gil, and Pavlos Marantidis. Matching in the description logic $\mathcal{FL}_0$ with respect to general tboxes. In Gilles Barthe, Geoff Sutcliffe, and Margus Veanes, editors, *Proc. of the 22nd Int. Conf. on Logic for Programming, Artificial Intelligence and Reasoning (LPAR-22)*, volume 57 of *EPiC Series in Computing*, pages 76–94. EasyChair, 2018.
5. Franz Baader, Oliver Fernández Gil, and Maximilian Pensel. Standard and non-standard inferences in the description logic $\mathcal{FL}_0$ using tree automata. In Daniel Lee, Alexander Steen, and Toby Walsh, editors, *GCAI 2018, 4th Global Conference on Artificial Intelligence*, volume 55 of *EPiC Series in Computing*, pages 1–14. EasyChair, 2018.
6. Franz Baader and Ralf Küsters. Matching in description logics with existential restrictions. In *Proc. of the 7th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2000)*, pages 261–272, 2000.
7. Franz Baader and Barbara Morawska. Unification in the description logic $\mathcal{EL}$. *Logical Methods in Computer Science*, 6(3), 2010.
8. Franz Baader and Barbara Morawska. Matching with respect to general concept inclusions in the description logic $\mathcal{EL}$. In Carsten Lutz and Michael Thielscher, editors, *Proc. of the 37th German Annual Conf. on Artificial Intelligence (KI'14)*, volume 8736 of *Lecture Notes in Computer Science*, pages 135–146. Springer, 2014.
9. Franz Baader and Paliath Narendran. Unification of Concept Terms in Description Logics. *J. Symb. Comput.*, 31(3):277–305, 2001.
10. Alexander Borgida, Ronald J. Brachman, Deborah L. McGuinness, and Lori Alperin Resnick. CLASSIC: A structural data model for objects. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 59–67, 1989.
11. Alexander Borgida and Ralf Küsters. What's not in a name? Initial explorations of a structural approach to integrating large concept knowledge-bases. Technical Report DCS-TR-391, Rutgers University, 1999.
12. Alexander Borgida and Deborah L. McGuinness. Asking queries about frames. In *Proc. of the 5th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'96)*, pages 340–349, 1996.