Restricted Unification in the DL $\mathcal{FL}_0$ (Extended Abstract)*

Franz Baader[1], Oliver Fernández Gil[1], and Maryam Rostamigiv[2]

[1] Theoretical Computer Science, TU Dresden, Dresden, Germany
franz.baader@tu-dresden.de, oliver.fernandez@tu-dresden.de
[2] Département d'Informatique, Paul Sabatier University, Toulouse, France
Maryam.Rostamigiv@irit.fr

1 Unification in $\mathcal{FL}_0$

Unification of concept patterns has been proposed as an inference service in Description Logics that can, for example, be used to detect redundancies in ontologies. Basically, a concept pattern is a concept description where certain concept names are viewed as variables. Given two such patterns C, D, the question is now whether they can be made equivalent by applying a substitution that replaces the variables with (possibly complex) patterns. A substitution that achieves this is called a unifier of C and D. In the context of detecting redundancies, the variables are concepts represented as concept names by one ontologist, but may be defined in more detail (i.e., by a complex description) by another one.

For the DL $\mathcal{FL}_0$, which has the concept constructors conjunction ($\sqcap$), value restriction ($\forall r.C$), and top concept ($\top$), unification was investigated in detail in [6]. It was shown there that unification in $\mathcal{FL}_0$ corresponds to unification modulo the equational theory *ACUIh* since (modulo equivalence) conjunction is associative (A), commutative (C), idempotent (I) and has top as a unit (U), and value restrictions behave like homomorphisms for conjunction and top (h). For this equational theory, it had already been shown in [1] that it has unification type zero, which means that a solvable unification problem need not have a minimal complete set of unifiers, and thus in particular not a finite one. From the DL point of view, the decision problem is, however, more interesting than the unification type. Since *ACUIh* is a commutative/monoidal theory [1,11], solvability of *ACUIh* unification problems (and thus of unification problems in $\mathcal{FL}_0$) can be reduced to solvability of systems of linear equations in a certain semiring, which for the case of *ACUIh* consists of finite languages over a finite alphabet, with union as semiring addition and concatenation as semiring multiplication [6]. By a reduction to the emptiness problem for root-to-frontier tree automata (RFAs), it was then shown in [6] that solvability of the language equations corresponding to an $\mathcal{FL}_0$ unification problem can be decided in exponential time. In addition, ExpTime-hardness of this problem was proved in [6] by a reduction from the intersection emptiness problem for deterministic RFAs (DRFAs) [13].

2 Restricted Unification in $\mathcal{FL}_0$

In [3], we investigate two kinds of restrictions on unification in $\mathcal{FL}_0$. On the one hand, we *syntactically restrict the role depth* (i.e., the maximal nesting of value restrictions) in the concepts obtained by applying a unifier to be bounded by a natural number $k \geq 1$. This restriction was motivated by a similar restriction used in research on least common subsumers (lcs) [12], where imposing a bound on the role depth guarantees existence of the lcs also in the presence of a (possibly cyclic) terminology. Also note that such a restriction was used in [9] for the equational theory ACh, for which unification is known to be undecidable [10]. It is shown in [9] that the problem becomes decidable if a bound on the maximal nesting of applications of homomorphisms is imposed. On the other hand, we consider a *semantic restriction* where, when defining the semantics of concepts, only interpretations for which the length of role paths is bounded by a given number k are considered. A similar restriction (for $k = 1$) was employed in [7] to improve the unification type for the modal logic $\mathbf{K}$ from type zero [8] to unitary or finitary for $\mathbf{K} + \Box\Box\bot$.

3 Results

Regarding the unification type of $\mathcal{FL}_0$, we show in [3] that both the syntactic and the semantic restriction ensures that it improves from type zero to unitary for unification without constants and finitary for unification with constants.

Theorem 1 ([3]). *Syntactically and semantically k-restricted unification in $\mathcal{FL}_0$ is unitary for unification without constants and finitary for unification with constants.*

This means that, in the restricted setting, finite complete sets of unifiers always exist, i.e., for a given pair C, D of concept patterns there always is a finite set of unifiers such that every unifier of C and D is an instance of a unifier in this set. If all the concept names occurring in C, D are variables then we call this a unification problem without constants. The theorem says that any solvable unification problem without constants has a most general unifier, i.e., a single unifier that has all unifiers as instances.

Regarding the decision problem, we can show that the complexity depends on whether the bound k is assumed to be encoded in unary or binary. For binary encoding of k, the complexity stays ExpTime, whereas for unary coding it drops from ExpTime to PSpace. This is again the case both for the syntactic and the semantic restriction.

Theorem 2 ([3]). *Given an integer $k \geq 1$ and $\mathcal{FL}_0$ concept patterns C, D as input, the problem of deciding whether C and D have a syntactically (semantically) k-restricted unifier or not is ExpTime-complete (in ExpTime) if the number k is assumed to be encoded in binary, and PSpace-complete if k is assumed to be encoded in unary.*

The complexity upper bounds can be obtained by adapting the tree automata constructions employed in [6] for solving the language equations induced by $\mathcal{FL}_0$ unification problems. Basically, one needs to add an appropriate counter to the states of the automata.

The ExpTime lower bound for binary coding in the syntactically restricted case is proved by a reduction from unrestricted unification in $\mathcal{FL}_0$. The PSpace lower bound for the case of unary coding is shown using a k-restricted variant of Seidl's ExpTime hardness result [13] for the intersection emptiness problem for DRFAs. The k-restricted intersection emptiness problem for DRFAs asks whether a given finite collection of DRFAs accepts a common tree of depth at most k.

Proposition 3 ([3]). *The k-restricted intersection emptiness problem for DRFAs is PSpace-complete if the number k is represented in unary.*

4 Conclusion

We have investigated in [3] both a semantically and a syntactically k-restricted variant of unification in $\mathcal{FL}_0$. These restrictions lead to a considerable improvement of the unification type from the worst possible type to unitary/finitary for unification without/with constants. For the complexity of the decision problem, we only obtain an improvement if k is assumed to be encoded in unary.

While these results are mainly of (complexity) theoretic interest, they could also have a practical impact. In fact, in our experiments with the system UEL, which implements several unification algorithms for the DL $\mathcal{EL}$ [5], we have observed that the algorithms usually yield many different unifiers, and it is hard to choose one that is appropriate for the application at hand (e.g., when generating new concepts using unification [2]). For this reason, we added additional constraints to the unification problem to ensure that the generated concepts are of a similar shape as the concepts already present in the ontology [2]. It makes sense also to use a restriction on the role depth as such an additional constraint since the role depth of the (unfolded) concepts occurring in real-world ontologies is usually rather small. This claim is supported by our experiments with the medical ontology SNOMED CT,[1] which has a maximal role depth of 10, and the acyclic ontologies in Bioportal 2017,[2] where a large majority also has a role depth of at most 10.

Acknowledgements

Franz Baader was partially supported by DFG TRR 248 (cpec, grant 389792660), Oliver Fernández Gil by DFG in project number 335448072, and Maryam Rostamigiv by a DAAD Short-Term Grant, 2021 (57552336). The authors should like to thank Patrick Koopmann for determining the maximal role depth of concepts in ontologies from Bioportal 2017 and in SNOMED CT.

[1] `https://www.snomed.org/`

[2] `https://zenodo.org/record/439510`

References

1. Franz Baader. Unification in commutative theories. *J. Symbolic Computation*, 8(5):479–497, 1989.
2. Franz Baader, Stefan Borgwardt, and Barbara Morawska. Constructing SNOMED CT concepts via disunification. LTCS-Report 17-07, Chair for Automata Theory, Institute for Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany, 2017. `https://lat.inf.tu-dresden.de/research/reports/2017/BaBM-LTCS-17-07.pdf`.
3. Franz Baader, Oliver Fernández Gil, and Maryam Rostamigiv. Restricted unification in the description logic $\mathcal{FL}_0$. In Boris Konev and Giles Reger, editors, *Proc. of the 13th International Symposium on Frontiers of Combining Systems (FroCoS 2021)*, volume 12941 of *Lecture Notes in Computer Science*. Springer, 2021. To appear. A long version of this paper containing detailed proofs has been published as technical report [4].
4. Franz Baader, Oliver Fernández Gil, and Maryam Rostamigiv. Restricted unification in the DL $\mathcal{FL}_0$ (extended version). LTCS-Report 21-02, Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany, 2021. `https://lat.inf.tu-dresden.de/research/reports/2021/BaGiRo21.pdf`.
5. Franz Baader, Julian Mendez, and Barbara Morawska. UEL: Unification solver for the description logic $\mathcal{EL}$ – system description. In *Proc. of the 6th International Joint Conference on Automated Reasoning (IJCAR 2012)*, volume 7364 of *Lecture Notes in Artificial Intelligence*, pages 45–51. Springer, 2012.
6. Franz Baader and Paliath Narendran. Unification of concept terms in description logics. *J. Symbolic Computation*, 31(3):277–305, 2001.
7. Philippe Balbiani, Cigdem Gencer, Maryam Rostamigiv, and Tinko Tinchev. About the unification type of $\mathbf{K} + \Box\Box\bot$. In *Proc. of the 34th International Workshop on Unification (UNIF 2020)*, pages 4:1–4:6. RISC-Linz, 2020.
8. Emil Jerabek. Blending margins: The modal logic K has nullary unification type. *J. Logic and Computation*, 25(5):1231–1240, 2015.
9. Ajay Kumar Eeralla and Christopher Lynch. Bounded ACh unification. *Math. Struct. Comput. Sci.*, 30(6):664–682, 2020.
10. Paliath Narendran. Solving linear equations over polynomial semirings. In *Proc. of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS 1996)*, pages 466–472. IEEE Computer Society, 1996.
11. Werner Nutt. Unification in monoidal theories. In Mark E. Stickel, editor, *Proc. of the 10th Int. Conf. on Automated Deduction (CADE 1990)*, volume 449 of *Lecture Notes in Artificial Intelligence*, pages 618–632, Kaiserslautern, Germany, 1990. Springer.
12. Rafael Peñaloza and Anni-Yasmin Turhan. A practical approach for computing generalization inferences in $\mathcal{EL}$. In Grigoris Antoniou, Marko Grobelnik, Elena Paslaru Bontas Simperl, Bijan Parsia, Dimitris Plexousakis, Pieter De Leenheer, and Jeff Z. Pan, editors, *Proc. of the 8th Extended Semantic Web Conference (ESWC 2011)*, volume 6643 of *Lecture Notes in Computer Science*, pages 410–423. Springer, 2011.
13. Helmut Seidl. Haskell overloading is DEXPTIME-complete. *Information Processing Letters*, 52:57–60, 1994.