Restricted Unification in the Description Logic $\mathcal{FL}_\bot$

Franz Baader and Oliver Fernández Gil

Theoretical Computer Science, TU Dresden, Dresden, Germany
franz.baader@tu-dresden.de, oliver.fernandez@tu-dresden.de

In a previous paper, we have investigated restricted unification in the Description Logic (DL) $\mathcal{FL}_0$. Here we extend this investigation to the DL $\mathcal{FL}_\bot$, which is obtained from $\mathcal{FL}_0$ by adding the bottom concept. We show that restricted unification in $\mathcal{FL}_\bot$ is decidable and provide some upper and lower bounds for the complexity. This is particularly interesting since the decidability status of unrestricted unification in $\mathcal{FL}_\bot$ appears to be still open. We also show an ExpTime lower bound for the unrestricted problem.

1 Introduction

Unification of concept patterns has been proposed as an inference service in Description Logic (DL) that can, for example, be used to detect redundancies in ontologies. 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 [4], where it was shown, by a reduction to solvability of linear language equations over finite languages, that the decision problem is in ExpTime. These languages are built over the alphabet Σ of all role names. ExpTime-hardness was proved by a reduction from the intersection emptiness problem of deterministic top-down tree automata, which is known to be ExpTime complete [5]. For the DL $\mathcal{FL}_\bot$, which extends $\mathcal{FL}_0$ with the bottom concept ($\bot$), unification can still be reduced to solving language equations. These equations are, however, more complicated, and thus the approach for solving linear equations using tree automata developed in [4] does not apply. The decidability status of unification in $\mathcal{FL}_\bot$ appears to be still open, though there have been some claims of decidability results. In the present paper, we will not tackle the decidability problem, but will show that ExpTime-hardness also holds for unification in $\mathcal{FL}_\bot$.

The DL $\mathcal{FL}_{reg}$ is obtained from $\mathcal{FL}_0$ by allowing for regular role expressions. Unification in this DL is also ExpTime-complete [2]. The upper bound can again be shown by a reduction to solvability of linear language equations, but now the solutions may also contain infinite (or, equivalently, regular) languages. Interestingly, when adding $\bot$ to this DL, unification in the obtained DL $\mathcal{FL}_{\bot reg}$ remains ExpTime-complete [3]. Intuitively, the reason for this is that solvable linear language equations over infinite languages have a unique maximal solution, which is regular [2]. This observation is used in [3] to reduce solvability of the more involved systems of language equations induced by $\mathcal{FL}_{\bot reg}$ unification problems to the simpler ones obtained from $\mathcal{FL}_{reg}$ unification problems.

In [1], 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 below a certain bound k. On the other hand, we consider a *semantic restriction* where, when defining the semantics of concepts, only interpretations are taken into account for which the length of role paths is bounded by a given number k. In the restricted setting, the complexity of unification in $\mathcal{FL}_0$ 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, both for the syntactic and the semantic restriction.

In the present paper, we investigate the complexity of unification in $\mathcal{FL}_\bot$ in the restricted setting. For the syntactically restricted case, we use the fact that only words of length at most k can occur in solutions of the corresponding language equations to compute a greatest solution by starting with the greatest substitution satisfying the syntactic restriction, and then successively removing words violating the condition that the left-hand side should be equal to the right-hand side. While this leads to a decision procedure, the upper bounds obtained this way (ExpTime for unary coding of k and 2ExpTime for binary coding) are higher than the respective lower bounds (PSpace and ExpTime). For the semantically restricted case, a similar approach yielding the same upper bounds could be used. However, in this case we can obtain better upper bounds by adapting the approach of [3] for unification in $\mathcal{FL}_{\bot reg}$ to this setting. Basically, the idea is that the rôle played by the language Σ^* for $\mathcal{FL}_{\bot reg}$ is now taken on by the language $\Sigma^{\leq k}$. This way, we obtain the upper bounds PSpace and ExpTime for unary and binary coding of k, respectively.

2 Unification in $\mathcal{FL}_0$ and $\mathcal{FL}_\bot$

Concept descriptions of $\mathcal{FL}_0$ are built from finite sets of concept and role names using the constructors conjunction ($\sqcap$), value restriction ($\forall r.C$), and top concept ($\top$). In $\mathcal{FL}_\bot$, the additional constructor bottom concept ($\bot$) is available. The *semantics* of $\mathcal{FL}_0$ and $\mathcal{FL}_\bot$ are defined as usual, based on the notion of an interpretation, which has a non-empty domain, and assigns subsets of the domain to concept descriptions, according to the semantics of the constructors (see [4] and [3] for details). Two concept descriptions C, D are *equivalent* (written $C \equiv D$) if they are interpreted as the same set in each interpretation. It is well-known that equivalence of $\mathcal{FL}_0$ concept descriptions can be decided in polynomial time [6]. The same can be shown for $\mathcal{FL}_\bot$, by using the characterization of equivalence given in [3].

Concept patterns are like concept descriptions, but may additionally contain concept variables in place of concept names. *Substitutions* replace concept variables by concept descriptions. Given two concept patterns C, D, the substitution σ is a *unifier* of the *unification problem* $C \overset{?}{\equiv} D$ if $\sigma(C) \equiv \sigma(D)$. A unification problem is *unifiable* (or *solvable*) if it has a unifier. Simultaneous unifiability of several unification problems by the same unifier can be reduced to unifiability of a single unification problem (see Lemma 3.3 in [4]). In the following, we assume that Σ denotes the set of role names occurring in C, D, $\mathsf{N_C}$ the set of concept names occurring in C, D, and $\mathsf{N_V}$ the set of variables occurring in C, D.

In [4] it is shown that unifiability of $C \overset{?}{\equiv} D$ in $\mathcal{FL}_0$ can be reduced to solving systems of *linear language equations with one-sided concatenation*, i.e., equations of the following form:

$$S_0 \cup S_1 \cdot X_1 \cup \ldots \cup S_n \cdot X_n \;=\; T_0 \cup T_1 \cdot X_1 \cup \ldots \cup T_n \cdot X_n, \tag{1}$$

where $S_0, \ldots, S_n, T_0, \ldots, T_n$ are finite languages of words over the finite alphabet Σ. A *solution* of such an equation assigns finite languages over Σ to the variables $X_1, \ldots, X_n$. Conversely, any system of such equations can be transformed into an equi-solvable unification problem. As shown in [4], deciding solvability of such systems of equations (and thus of unification problems in $\mathcal{FL}_0$) is an ExpTime-complete problem. The upper bound is proved by a reduction to the emptiness problem of automata on finite trees, whereas the lower bound is demonstrated by a reduction of the intersection emptiness problem for deterministic top-down tree automata.

In a unification problem $C \overset{?}{\equiv} D$ in $\mathcal{FL}_\bot$, not only the concept patterns C, D may contain the bottom concept, but also the concepts replacing the variables. As shown in [3], such a problem can be translated into an equi-solvable system of language equations of the following form:

$$\begin{aligned} E_\bot &:= S_\bot \cdot \Sigma^* \cup \bigcup_{X \in \mathsf{N_V}} S_X \cdot X_\bot \cdot \Sigma^* = T_\bot \cdot \Sigma^* \cup \bigcup_{X \in \mathsf{N_V}} T_X \cdot X_\bot \cdot \Sigma^*, \\ E_A &:= E_\bot^\ell \cup S_A \cup \bigcup_{X \in \mathsf{N_V}} S_X \cdot X_A = E_\bot^r \cup T_A \cup \bigcup_{X \in \mathsf{N_V}} T_X \cdot X_A. \end{aligned} \tag{2}$$

There is one equation E_A for each concept name $A \in \mathsf{N_C}$ and an equation $E_\bot$ for bottom. The sets $S_\bot, T_\bot$, S_A, T_A, and S_X, T_X are finite sets of words over Σ, whereas the terms $E_\bot^\ell$ and $E_\bot^r$ denote the left-hand side and the right-hand side, respectively, of the equation $E_\bot$.

There are two reasons why testing unifiability in $\mathcal{FL}_\bot$ is more problematic than in $\mathcal{FL}_0$. First, for $\mathcal{FL}_0$, one obtains one equation of the form (1) for each concept name, which can be solved independently since these equations do not share variables. In the system (2), the variables $X_\bot$ occur in each equation. Second, the concatenation with Σ^* from the right in $E_\bot$ means that the equations are no longer ones with one-sided concatenation. This makes it hard to impossible to find a reduction to the emptiness problem of automata on finite trees. In the next two sections we will show that things become easier in the restricted setting.

Here, we do not address the problem of determining a complexity upper bound, but at least establish an ExpTime lower bound for unification in $\mathcal{FL}_\bot$, which was not known until now.

Theorem 1. *Deciding unifiability of $\mathcal{FL}_\bot$ unification problems is ExpTime-hard.*

Note that ExpTime-hardness of unification for $\mathcal{FL}_\bot$ does not follow immediately from ExpTime-hardness for $\mathcal{FL}_0$. In fact, even if we consider a unification problem $C \overset{?}{\equiv} D$ for $\mathcal{FL}_0$ concept descriptions C, D, this problem may have a unifier in $\mathcal{FL}_\bot$, though it does not have one in $\mathcal{FL}_0$. The reason is that there are more $\mathcal{FL}_\bot$ substitutions than $\mathcal{FL}_0$ substitutions. To overcome this problem, we consider the proof of ExpTime-hardness for unification in $\mathcal{FL}_0$. For every instance of the intersection emptiness problem for deterministic top-down tree automata, this reduction constructs a system of equations of the form (1) (one for each automaton), which is solvable iff the intersection of the languages accepted by the automata in the instance is non-empty. This system can then be translated into an equi-solvable $\mathcal{FL}_0$ unification problem. A close analysis of this problem reveals that the $\mathcal{FL}_0$ unification problems obtained this way cannot have $\mathcal{FL}_\bot$ solutions containing the bottom concept. Thus, they are solvable in $\mathcal{FL}_\bot$ iff they are solvable in $\mathcal{FL}_0$.

3 Syntactically Restricted Unification in $\mathcal{FL}_\bot$

In the syntactically restricted case, equivalence $\equiv$ between concepts is replaced by syntactically k-restricted equivalence $\equiv_{syn}^k$ [1]. Two $\mathcal{FL}_\bot$ concepts C, D satisfy $C \equiv_{syn}^k D$ if $C \equiv D$ and the role depth of C and D (i.e., the maximal nesting of value restrictions) is bounded by k. For an integer $k \geq 1$, a *syntactically k-restricted $\mathcal{FL}_\bot$ unification problem* is of the form $C \overset{?}{\equiv}{}_{syn}^k D$, where C and D are $\mathcal{FL}_\bot$ concept patterns. A *unifier* of this equation is a substitution such that $\sigma(C) \equiv_{syn}^k \sigma(D)$.

Since avoiding nesting of value restrictions of depth $> k$ corresponds to avoiding words of length $> k$, syntactically restricted unification in $\mathcal{FL}_\bot$ can be reduced to checking whether the equations in (2) have a solution θ such that, for all $X \in \mathsf{N_V}$ and all $A \in \mathsf{N_C}$, the following holds:

$$S_X \cdot \theta(X_\bot) \cup T_X \cdot \theta(X_\bot) \subseteq \Sigma^{\leq k} \quad \text{and} \quad S_X \cdot \theta(X_A) \cup T_X \cdot \theta(X_A) \subseteq \Sigma^{\leq k}, \tag{3}$$

where $\Sigma^{\leq k}$ consists of all words over Σ of length at most k.

We can now try to construct a solution of (2) satisfying (3) as follows. We start with an assignment θ that maps every variable to $\Sigma^{\leq k}$, and then remove from each set $\theta(X)$ for the variables X occurring in (2) those elements that lead to a violation of (3). After that, we iteratedly remove words that violate (2). For instance, given a variable $X_\bot$, we remove from $\theta(X_\bot)$ the set $\{w \in \theta(X_\bot) \mid S_X \cdot \{w\} \not\subseteq \theta(E^r_\bot) \text{ or } T_X \cdot \{w\} \not\subseteq \theta(E^\ell_\bot)\}$. Since we start with sets $\theta(X)$ of cardinality $|\Sigma^{\leq k}|$ for polynomially many variables X, and in each step remove at least one element from such a set, this removal process terminates. We can show that (2) has a solution satisfying (3) iff the final assignment reached by this process solves (2). The cardinality of $\Sigma^{\leq k}$ is exponential in the size of the unary representation of k (which is k), but doubly exponential in the size of the binary representation of k (which is $\log k$). This yields the upper bounds stated in the following theorem. The lower bounds can be shown similarly to how they are proved for the syntactically restricted case for $\mathcal{FL}_0$ in [1], but additionally using the idea employed in the proof of Theorem 1.

Theorem 2. *Let $k \geq 1$ and C, D two $\mathcal{FL}_\bot$ concept patterns. Deciding whether the syntactically k-restricted unification problem $C \overset{?}{\equiv}{}^k_{syn} D$ has a unifier or not is PSpace-hard and in ExpTime if k is encoded in unary, and ExpTime-hard and in 2ExpTime if k is encoded in binary.*

4 Semantically Restricted Unification in $\mathcal{FL}_\bot$

For an integer $n \geq 1$ and a given interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, a role path of length n is a sequence $d_0, r_1, d_1, \ldots, d_{n-1}, r_n, d_n$, where $d_0, \ldots, d_n$ are elements of $\Delta^\mathcal{I}$, $r_1, \ldots, r_n$ are role names, and $(d_{i-1}, d_i) \in r_i^\mathcal{I}$ holds for all $i = 1, \ldots, n$. The interpretation $\mathcal{I}$ is called k-restricted if it does not contain any role paths of length $> k$. Semantically k-restricted equivalence is then defined as follows: $C \equiv^k_{sem} D$ if $C^\mathcal{I} = D^\mathcal{I}$ for all k-restricted interpretations $\mathcal{I}$. A *semantically k-restricted unification problem* is of the form $C \overset{?}{\equiv}{}^k_{sem} D$, where C and D are $\mathcal{FL}_\bot$ concept patterns. A unifier of this equation is a substitution σ such that $\sigma(C) \equiv^k_{sem} \sigma(D)$.

The effect of this restriction is that value restrictions of a nesting depth $> k$ are equivalent to $\top$, and thus can be removed. On the language equation side this means that equality of the left- and right-hand side of an equation needs to hold only for words of length up to k. For semantically k-restricted unification in $\mathcal{FL}_0$ it was shown in [1] that unifiability can be reduced to testing whether, for language equations of the form (1), there is an assignment θ of finite languages to the variables satisfying

$$(S_0 \cup S_1 \cdot \theta(X_1) \cup \ldots \cup S_n \cdot \theta(X_n)) \cap \Sigma^{\leq k} \;=\; (T_0 \cup T_1 \cdot \theta(X_1) \cup \ldots \cup T_n \cdot \theta(X_n)) \cap \Sigma^{\leq k}. \quad (4)$$

For semantically k-restricted unification in $\mathcal{FL}_\bot$, the intersection with $\Sigma^{\leq k}$ needs to be applied to both sides of the equations in the system of the form (2). In addition, since we are then only interested in words of length up to k, right-concatenation with the language Σ^* can be replaced by right-concatenation with $\Sigma^{\leq k}$. Thus, semantically k-restricted unification in $\mathcal{FL}_\bot$ can be reduced to deciding whether there is an assignment θ of finite languages satisfying

$$(S_\bot \cdot \Sigma^{\leq k} \cup \bigcup_{X \in \mathsf{N_V}} S_X \cdot \theta(X_\bot) \cdot \Sigma^{\leq k}) \cap \Sigma^{\leq k} = (T_\bot \cdot \Sigma^{\leq k} \cup \bigcup_{X \in \mathsf{N_V}} T_X \cdot \theta(X_\bot) \cdot \Sigma^{\leq k}) \cap \Sigma^{\leq k}, \quad (5)$$

$$(\theta(E^\ell_{\bot,k}) \cup S_A \cup \bigcup_{X \in \mathsf{N_V}} S_X \cdot \theta(X_A)) \cap \Sigma^{\leq k} = (\theta(E^r_{\bot,k}) \cup T_A \cup \bigcup_{X \in \mathsf{N_V}} T_X \cdot \theta(X_A)) \cap \Sigma^{\leq k}. \quad (6)$$

To check for the existence of an assignment satisfying (5) and (6), we follow the ideas used in [3] to solve (2) for the case of infinite languages. The reduction given in [3] can be adapted to show that deciding the existence of a finite assignment θ satisfying (5) and (6) can be reduced to checking whether, for language equations of the form (1), there is a finite assignment θ satisfying (4). More precisely, we can show the following two results (which are analoga of the results in Lemmata 5 and 7 in [3]):

- An assignment of finite languages satisfying (5) exists iff there is an assignment θ of finite languages such that
$$(S_\bot \cdot \Sigma^{\leq k} \cup \bigcup_{X \in \mathsf{N_V}} S_X \cdot \theta(X_\bot)) \cap \Sigma^{\leq k} = (T_\bot \cdot \Sigma^{\leq k} \cup \bigcup_{X \in \mathsf{N_V}} T_X \cdot \theta(X_\bot)) \cap \Sigma^{\leq k}. \qquad (7)$$

- An assignment of finite languages satisfying (5) and (6) exists iff there is an assignment θ of finite languages satisfying (5) such that the following holds for all $A \in \mathsf{N_C}$:
$$(S_\bot \cdot \Sigma^{\leq k} \cup S_A \cup \bigcup_{X \in \mathsf{N_V}} S_X \cdot \theta(X_A)) \cap \Sigma^{\leq k} = (T_\bot \cdot \Sigma^{\leq k} \cup T_A \cup \bigcup_{X \in \mathsf{N_V}} T_X \cdot \theta(X_A)) \cap \Sigma^{\leq k}. \quad (8)$$

This shows that the problem of finding an assignment of finite languages satisfying (5) and (6) can be reduced to finding an assignment of finite languages satisfying (7) and (8). Since the solvability conditions (7) and (8) are exactly the ones dealt with in [1] for semantically k-restricted unification in $\mathcal{FL}_0$, we can use the approach for solving them developed there. For the case of unary coding of k, this yields a PSpace upper bound since the right-concatenation of $S_\bot$ and $T_\bot$ with $\Sigma^{\leq k}$ can be realized using a system of equations of size linear in k. For binary coding of k, this does not work since the equation system expressing the right-concatenation with $\Sigma^{\leq k}$ would then have exponential size. Instead, one needs to look more closely into the automaton construction used in [1] to show the ExpTime upper bound for the binary cases, and then show that it can be adapted to deal also with the right-concatenation with $\Sigma^{\leq k}$ without causing an exponential blow-up. A PSpace lower bound for the unary case can be proved by a reduction from semantically k-restricted unification in $\mathcal{FL}_0$. For the binary case, like in $\mathcal{FL}_0$, it remains open whether the ExpTime upper bound is tight.

Theorem 3. *Let $k \geq 1$ and C, D two $\mathcal{FL}_\bot$ concept patterns. Deciding whether the k-semantically restricted unification problem $C \overset{?}{\equiv}{}^{k}_{sem} D$ has a unifier or not is PSpace-complete if k is encoded in unary, and in ExpTime if k is encoded in binary.*

References

[1] Franz Baader, Oliver Fernández Gil, and Maryam Rostamigiv. Restricted unification in the DL $\mathcal{FL}_0$. In *Frontiers of Combining Systems - 13th International Symposium, FroCoS 2021, Proceedings*, volume 12941 of *LNCS*. Springer, 2021.

[2] Franz Baader and Ralf Küsters. Unification in a description logic with transitive closure of roles. In *Logic for Programming, Artificial Intelligence, and Reasoning, 8th International Conference, LPAR 2001, Proceedings*, volume 2250 of *LNCS*, Springer, 2001.

[3] Franz Baader and Ralf Küsters. Unification in a description logic with inconsistency and transitive closure of roles. In *Proceedings of the 2002 International Workshop on Description Logics (DL2002)*, volume 53 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2002.

[4] Franz Baader and Paliath Narendran. Unification of concept terms in description logics. *J. Symb. Comput.*, 31(3):277–305, 2001.

[5] Helmut Seidl. Haskell overloading is DEXPTIME-complete. *Inf. Process. Lett.*, 52(2):57–60, 1994.

[6] Hector J. Levesque and Ron J. Brachman. Expressiveness and Tractability in Knowledge Representation and Reasoning. *Computational Intelligence*, 3:78–93, 1987.