Around Unification in $\mathcal{FL}_\bot$ – Three Related Problems (Extended Abstract)

Barbara Morawska[1], Sławomir Kost[1]

[1]Institute of Computer Science, University of Opole, Poland

Abstract

In this paper we present three results concerning the unification problem in the description logic $\mathcal{FL}_\bot$. The logic $\mathcal{FL}_\bot$ is a sub-Boolean logic that supports only conjunction, value restrictions, and the top and bottom constructors, without any form of negation. Subsumption in $\mathcal{FL}_\bot$ can be decided in polynomial time. Although we do not solve the unification problem itself, we establish three related findings. First, we show that unification in $\mathcal{FL}_\bot$ is of type nullary, a result inspired by a similar theorem for the modal logic K. Second, we reduce the unification problem in $\mathcal{FL}_\bot$ to the unification problem in $\mathcal{FL}_0$, equipped with a forward TBox. Third, we revisit the known result that the matching problem in $\mathcal{FL}_\bot$ can be solved in polynomial time and provide a new algorithm for it.

Keywords

description logic, unification type

1. Introduction

In this paper, we focus on a small description logic, $\mathcal{FL}_\bot$, which extends the constructors of its sister logic $\mathcal{FL}_0$ by adding the bottom concept. We present three results: the unification type of $\mathcal{FL}_\bot$ is nullary, inspired by a similar result for the modal logic K (see [1]); the unification problem in $\mathcal{FL}_\bot$ can be reduced to the one in $\mathcal{FL}_0$ with a special TBox, corresponding to [2]; and we present a simple-to-implement algorithm which solves the matching problem in $\mathcal{FL}_\bot$ in polynomial time.

2. The description logics $\mathcal{FL}_0$ and $\mathcal{FL}_\bot$

All notions in this chapter are introduced for $\mathcal{FL}_\bot$. To obtain their equivalents in $\mathcal{FL}_0$, simply omit $\bot$. In the description logic $\mathcal{FL}_\bot$, (complex) concepts are generated from two disjoint sets N_C and N_R, reffered to as concept names and role names, by the following grammar:
$C ::= \top \mid \bot \mid A \mid C \sqcap C \mid \forall r.C$, where $A \in N_C, r \in N_R$.

An interpretation of concepts in $\mathcal{FL}_\bot$ is a pair $I = (\Delta^I, \cdot^I)$, where Δ^I is a non-empty domain of elements and $\cdot^I$ is an interpreting function defined on concept names and role names as follow: $\top^I = \Delta^I$; $\bot^I = \emptyset$; $A^I \subseteq \Delta^I$, for any $A \in N_C$; $r^I \subseteq \Delta^I \times \Delta^I$, for any $r \in N_R$, and extended to all complex concepts in the usual way: $(C \sqcap D)^I = C^I \cap D^I$; $(\forall r.C)^I = \{d \in \Delta^I \mid \forall e \in \Delta^I[(d,e) \in r^I \to e \in C^I]\}$; $(\forall v.C)^I = (\forall r_1 \forall r_2 \ldots \forall r_n.C)^I$ where $v = r_1 \ldots r_n \in {N_R}^+$.

A concept may be reduced with the following reductions to an equivalent concept (interpreted by the same set in any interpretation): $C \sqcap \top, \top \sqcap C \rightsquigarrow C$; $C \sqcap \bot, \bot \sqcap C \rightsquigarrow \bot$; $\forall r.\top \rightsquigarrow \top$; $\forall r.(C \sqcap D) \rightsquigarrow \forall r.C \sqcap \forall r.D$. We call a concept C *reduced* iff none of the reduction rules applies.

For convenience, we will use the notation $\forall v.\alpha$ for the concept of the form: $\forall r_1(\forall r_2(\ldots(\forall r_n.\alpha)))$, where $v = r_1 \ldots r_n$ and α is either $\top$ or $\bot$ or a concept name A. A concept of this form is called a *particle*.

This research is part of the project No 2022/47/P/ST6/03196 within the POLONEZ BIS programme co-funded by the National Science Centre and the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 945339. For the purpose of Open Access, the author has applied a CC-BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.

DL 2025: 38th International Workshop on Description Logics, September 3–6, 2025, Opole, Poland

barbara.morawska@uni.opole.pl (B. Morawska); skost@uni.opole.pl (S. Kost)

0000-0003-4724-7206 (B. Morawska); 0000-0003-1898-9489 (S. Kost)

The word v over N_R is called *the role word of* the particle $\forall v.\alpha$. For role words v, v', by $v \leq v'$ we denote that v is a prefix of v'.

It is easy to see that any concept is equivalent to a conjunction of particles, $C = \forall v_1.\alpha_1 \sqcap \cdots \sqcap \forall v_n.\alpha_n$, where $v_1, \ldots, v_n$ are possibly empty words over N_R. In fact because of properties of conjunction, we identify a reduced concept with a set of particles in such a conjunction.

Let C be an $\mathcal{FL}_\bot$-reduced concept. We define $rd(C)$ (role depth) and $size(C)$ (size) recursively: if $C = A$ or $C = \top$ or $C = \bot$, then $rd(C) = size(C) = 0$; if $C = D \sqcap E$, then $rd(C) = max(\{rd(D), rd(E)\})$ and $size(C) = size(D) + size(E)$; if $C = \forall r.C'$, $rd(C) = rd(C') + 1$ and $size(C) = size(C') + 1$.

Subsumption between concepts $C \sqsubseteq D$ obtains iff for all interpretations I, $C^I \subseteq D^I$. *Equivalence*: $C \equiv D$ iff $C \sqsubseteq D$ and $D \sqsubseteq C$. For any concept C, we have $\bot \sqsubseteq C$ and $C \sqsubseteq \top$. In $\mathcal{FL}_\bot$, let C and $D = \{P_1, \ldots, P_n\}$ be reduced concepts. Then $C \sqsubseteq D$ iif for every $P \in D$, one of the following holds: (1) $P \in C$, (2) $P = \forall v.\alpha$, where α is a concept name or $\bot$, and there exists $\forall v'.\bot \in C$ such that $v' \leq v$.

3. Unification problem in $\mathcal{FL}_\bot$

In order to define a unification problem, we partition the set of concept names N_C into two disjoint sets: variables (Var) and constants ($Cons$). A variable is thus a concept name that may be substituted by any concept while a constant cannot be substituted.

A *substitution* is a mapping from Var to the set of all $\mathcal{FL}_\bot$-concepts. It is extended to all concepts in the usual way. The *unification problem* (*unification problem*) is defined by its input $\Gamma = \{C_1 \sqsubseteq^? D_1, \ldots, C_n \sqsubseteq^? D_n\}$; and the output is "yes" if there is a substitution that makes these subsumptions true, or "no" otherwise. Without loss of generality, we can assume that $D_1, \ldots, D_n$ are particles. A substitution σ is a *unifier* for the unification problem $\Gamma = \{C_1 \sqsubseteq^? P_1, \ldots, C_n \sqsubseteq^? P_n\}$ iff $\sigma(C_1) \sqsubseteq \sigma(P_1), \ldots, \sigma(C_n) \sqsubseteq \sigma(P_n)$. In this case, we say that the problem is *unifiable*.

Let Γ be an unification problem with the set of variables V and unifiers σ, γ. We say that σ is *more general* than γ (or γ is *less general* than σ), if there is a substitution τ such that $\gamma(X) \equiv \tau(\sigma(X))$, for all $X \in V$. If a unifier is more general than any other unifier, we call it a *most general unifier* (an mgu) of Γ.

A set Π of unifiers of a given unification problem Γ is called a *complete set of unifiers* if every unifier of Γ is less general than some element of Π. For a given unification problem Γ we define four *unification types* (from "best" to "worst") based on the existence and cardinality of its complete set. The problem has unification type: *unitary* if there exists complete set of unifiers consisting of one unifier σ; *finitary* if it has finite compete set of unifiers, but has no most general unifier; *infinitary* if it has an infinite minimal complete set of unifiers; *nullary* (or *zero*) if it has no minimal complete set of unifiers. The unification type of a logic ($\mathcal{FL}_\bot$ in our case) is the worst unification type of its unifiable problems.

4. Type nullary result

In this section, we sketch a prove that $\mathcal{FL}_\bot$ has nullary unification type by showing that the unification problem $\Gamma = \{X \sqsubseteq^? \forall r.X\}$ has no minimal complete set of unifiers. To this end, we introduce the set U of substitutions consisting of:

$\sigma_0(X) = \bot$; $\sigma_n(X) = X \sqcap \forall r.X \sqcap \ldots \sqcap \forall r^{n-1}.X \sqcap \forall r^n.\bot$, for $n \geq 1$; $\sigma_\top(X) = \top$.

One can easily check that $\sigma_\alpha(X) \sqsubseteq \sigma_\alpha(\forall r.X)$, for each $\alpha \in \mathbb{N} \cup \{\top\}$.

It can also be shown that the set U is complete for Γ. Let σ be a unifier for Γ not equal to $\sigma_\top$ and let $\sigma_n \in U$ where $n = rd(\sigma(X))$. Then $\sigma(X) \equiv \sigma(\sigma_n(X))$.

At this point we know that U is a complete set of unifiers of Γ. To complete the argument, we observe that there is no minimal complete set of unifiers for Γ. It can be easily shown that: σ_{n+1} is more general that σ_n, but σ_n is not more general than σ_{n+1}, for each $n \geq 0$. Using a proof by contradiction we obtain the result:

Theorem 1. *The type of the unification problem Γ is nullary.*

5. Reduction from $\mathcal{FL}_\bot$ to $\mathcal{FL}_0$ with a TBox

A $\mathcal{FL}_0$ TBox (TBox for short) is a finite set of $\mathcal{FL}_0$-subsumptions. A model of a TBox $\mathcal{T}$ is an interpretation I such that $E^I \subseteq F^I$ for all $E \sqsubseteq F \in \mathcal{T}$. Let C and D be concepts. We say that C is subsumed by D w.r.t. a TBox $\mathcal{T}$ (written $C \sqsubseteq_{\mathcal{T}} D$) if $C^I \subseteq D^I$ for each model I of $\mathcal{T}$. We say that σ is a unifier of a unification problem Γ w.r.t. a TBox $\mathcal{T}$ if $\sigma(C) \sqsubseteq_{\mathcal{T}} \sigma(D)$ for each $C \sqsubseteq D \in \Gamma$.

Let C be an $\mathcal{FL}_\bot$ concept, and B be a constant, that does not appear in C. By C_B we denote the $\mathcal{FL}_0$-concept obtained from C by replacing all occurrences of $\bot$ with the constant B. For $s = C \sqsubseteq D$, $s_B = C_B \sqsubseteq D_B$. Given a finite set Γ of $\mathcal{FL}_\bot$-subsumptions, we define the corresponding set Γ_B of $\mathcal{FL}_0$-subsumptions by $\Gamma_B = \{s_B \mid s \in \Gamma\}$. For a given finite set of subsumptions Γ, $N_C(\Gamma)$ is the set of all concept names occuring in Γ, $N_R(\Gamma)$ is the set of all role names occuring in Γ. For a given signature $\Sigma =< S_C, S_R >$, where S_C is a finite subset of N_C and S_R is a finite subset of N_R, we define the following TBox: $\mathcal{T}_B^\Sigma = \{B \sqsubseteq A \mid \text{ for every } A \in S_C\} \cup \{B \sqsubseteq \forall r.B \mid \text{ for every } r \in S_R\}$. To simplify notation, we henceforth denote $\mathcal{T}_B^{<N_C(\{s\}),N_R(\{s\})>}$ as $\mathcal{T}_B^s$, and express $< N_C(\Gamma), N_R(\Gamma) >$ as $\Sigma(\Gamma)$.

The following theorem is similar to Lemma 2.2 in [2], which considers subsumptions between concept names:

Theorem 2. *An $\mathcal{FL}_\bot$-subsumption s of the form $C \sqsubseteq D$ obtains iff $C_B \sqsubseteq_{\mathcal{T}_B^s} D_B$.*

If σ is a unifier of an $\mathcal{FL}_\bot$ unification problem Γ of the minimal size where size of σ is sum of $\{size(\sigma(X))$ where X is in domain of $\sigma\}$, then the signature of σ is contained in $\Sigma(\Gamma)$. Therefore:

Theorem 3. *Let Γ be a unification problem in $\mathcal{FL}_\bot$. Then Γ has an $\mathcal{FL}_\bot$-unifier iff Γ_B has an $\mathcal{FL}_0$-unifier w.r.t. the TBox $\mathcal{T}_B^{\Sigma(\Gamma)}$.*

We showed that the unification problem in $\mathcal{FL}_\bot$ can be reduced to a unification problem in $\mathcal{FL}_0$ with a TBox. This does not give us a solution for the unification in $\mathcal{FL}_\bot$, since unification in $\mathcal{FL}_0$ with a TBox is not solved. However, it show that the unification problem in $\mathcal{FL}_0$ with a TBox is more difficult than unification in $\mathcal{FL}_\bot$.

6. Matching in $\mathcal{FL}_\bot$ is polynomial

The matching problem is a special kind of a unification problem $C \equiv^? D$, where C contains no variables. In [3], it was shown that, with respect to general TBoxes, matching is ExpTime-complete in $\mathcal{FL}_0$, whereas for a restricted form of TBoxes, namely *forward TBoxes*, the complexity drops to PSpace. We can transfer this result to $\mathcal{FL}_\bot$ via Theorem 3, obtaining that matching in $\mathcal{FL}_\bot$ is in PSpace. In [4] (see Theorem 3.8) it was shown that matching in $\mathcal{FL}_\bot$ is polynomial. Here, we present another simple-to-implement algorithm which solves the matching problem in $\mathcal{FL}_\bot$ in polynomial time.

Algorithm 1 Matching

Input: $C \equiv^? D$, where C does not contain variables, $D = E \sqcap \forall v_1.X_1 \sqcap \cdots \sqcap \forall v_n.X_n$, where E does not contain variables, $X_1, \ldots, X_n$ are (not necessarily different) variables, and $v_1, \ldots, v_n$ are words over N_R.
Output: True if there is a matcher, False otherwise.

procedure MATCHING($C \equiv^? D$)
  if $C \not\sqsubseteq E$ **then**
    return False
  else
    for all $\forall v.A \in C$ such that $\forall v.A \notin E$ and there is no $\forall v'.\bot \in E$ where $v' \leq v$ **do**
      Find $\forall v_i.X_i$ such that $v_i \leq v$ (v_i is a prefix of v)
      if no $\forall v_i.X_i$ is found **then**
        return False
  return True

One can see that the algorithm must terminate in time polynomial in the size of the problem. In order to justify the correctness of Algorithm 1 we define a special substitution $\hat{\sigma}$. For every X occurring in

D, $\hat{\sigma}(X) := \bigsqcap\{\forall u.\alpha \mid \forall v.X \in D$ and $\forall vu.\alpha \in C$ where α is a constant or $\bot\}$. Next we prove that a matching problem $C \equiv^? D$ has a unifier iff the substitution $\hat{\sigma}$ is a unifier. The correctness follows from the fact that the algorithm computes the substitution $\hat{\sigma}$.

7. Conclusions

We have presented three results related to the unification problem in $\mathcal{FL}_\bot$. The unification type of $\mathcal{FL}_\bot$ turns out to be nullary. Hence, $\mathcal{FL}_\bot$ has the same type as the description logics $\mathcal{EL}$, $\mathcal{FL}_0$, and $\mathcal{ALC}$. The second result, reduction of the unification problem in $\mathcal{FL}_\bot$ to unification in $\mathcal{FL}_0$ modulo a TBox $\mathcal{T}_B^\Sigma$ implies that the unification problem in $\mathcal{FL}_\bot$ is easier than the one in $\mathcal{FL}_0$ with a TBox. It is even easier than the unification in $\mathcal{FL}_0$ with a forward TBox. As the third result, we have presented a simple algorithm that solves matching in polynomial time.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT based on GPT-4o in order to: Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] E. Jerábek, Blending margins: the modal logic K has nullary unification type, J. Log. Comput. 25 (2015) 1231–1240. doi:10.1093/LOGCOM/EXT055.

[2] F. Baader, R. Küsters, A. Borgida, D. L. McGuinness, Efficient tbox reasoning with value restrictions using the $\mathcal{FL}$0wer reasoner., Theory and Practice of Logic Programming 22 (2022) 162–192.

[3] F. Baader, O. Fernández Gil, P. Marantidis, Matching in the description logic FL0 with respect to general tboxes (extended abstract), in: M. Simkus, G. Weddell (Eds.), Proceedings of the 32nd International Workshop on Description Logics (DL'19), volume 2373 of *CEUR Workshop Proceedings*, CEUR-WS, 2019.

[4] F. Baader, R. Küsters, A. Borgida, D. L. McGuinness, Matching in description logics, Journal of Logic and Computation 9 (1999) 411–447.