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

Barbara Morawska
Ahmedabad University, School of Engineering and Applied Science
barbara.morawska@ahduni.edu.in

Abstract

Unification in the Description Logic $\mathcal{FL}_\perp$ is shown to be in ExpTime. The idea is to reduce the problem to $\mathcal{FL}_0$-unification, which is ExpTime-complete. The exponential time algorithm for unification in $\mathcal{FL}_\perp$ consists of two steps: 1. $\perp$-elimination (flattening) and 2. $\mathcal{FL}_0$-unification

The Problem

Unification in Description Logics is a reasoning problem that asks for conditions under which some concept descriptions are equivalent.

The small description logic $\mathcal{FL}_0$ allows concept descriptions to be constructed from a set of concept names $\mathbf{N}$ and role names $\mathbf{R}$ with the following constructors: $\sqcap$ (conjunction), $\top$ (top constructor) and $\forall r$ (value restriction for a role r).

$\mathcal{FL}_\perp$ is constructed like $\mathcal{FL}_0$ with addition of $\perp$ (bottom constructor).

- The unification problem was first proposed and solved for $\mathcal{FL}_0$ in [Baader and Narendran, 2001]. It was shown that unification in $\mathcal{FL}_0$ is ExpTime complete.
- In [Baader and Küsters, 2001], the authors extended the result for the same logic with regular operations on role strings, $\mathcal{FL}_{reg}$.
- In in [Baader and Küsters, 2002] the same complexity bound was extended to $\mathcal{FL}_{\perp reg}$, which added inconsistency symbol to the constructors of $\mathcal{FL}_{reg}$.

At the first sight, the unification problem in $\mathcal{FL}_\perp$ seems to be solvable by an extension of a procedure for $\mathcal{FL}_0$-unification, similar to the extension of $\mathcal{FL}_{reg}$-unification for $\mathcal{FL}_{\perp reg}$. In [Baader and Küsters, 2002] the authors noticed that if one adds $\perp$ directly to $\mathcal{FL}_0$, their method of solving unification does not apply.

Properties of $\mathcal{FL}_\perp$ concept descriptions

Most of the properties hold also for $\mathcal{FL}_0$-concept descriptions.

1. Conjunction is associative, commutative, idempotent with $\top$ as the unit (ACUI-equational theory)
2. Value restriction behaves like homomorphism: $\forall r.(C_1 \sqcap C_2) \equiv \forall r.C_1 \sqcap \forall r.C_2$
3. Properties of $\top$: $C \sqcap \top \equiv C, \forall v.\top \equiv \top$.
4. Subsumption between concept descriptions: $C \sqsubseteq D$ iff $C \sqcap D \equiv C$.
5. Properties of $\perp$: $C \sqcap \perp \equiv \perp, \forall v_1.\perp \sqsubseteq \forall v_2.\perp$ iff v_1 is a prefix of v_2.
6. A **particle** is a concept description of the form $\forall r_1.\forall r_2.\dots.\forall r_m.A$ (in short $\forall r_1 r_2 \dots r_m.A$)
7. **Normal form**: C is a conjunction of particles (i.e. a set of particles).
8. Subsumption in $\mathcal{FL}_\perp$ is decidable in polynomial time.

Definition of the unification problem

- We assume a set of concept names $\mathbf{Var}$ (variables), disjoint from $\mathbf{N}$ (constants) and allow the variables to be substituted by $\mathcal{FL}_\perp$ concepts.
- Unification problem is given as a set of goal subsumptions between $\mathcal{FL}_\perp$ concepts in normal form: $\Gamma = \{C_1 \sqsubseteq^? D_1, \dots, C_n \sqsubseteq^? D_n\}$., where $D_1, \dots, D_n$ are particles.
- A **solution** or a unifier for the unification problem is an assignment of ground concepts to variables such that the goal subsumptions hold.

Example 1. $\Gamma = \{X \sqsubseteq^? A,\ X \sqsubseteq^? \forall r.X\}$ The unification problem Γ cannot be solved in $\mathcal{FL}_0$, but it can in $\mathcal{FL}_\perp$. It is easy to see that X cannot be substituted by $\top$. A substitution for X may contain A or be $\perp$. The next subsumption creates a cycle, which can be solved either by $\top$ or a $\perp$-particle. Hence X has to be $\perp$.

Example 2. $\Gamma^* = \{X \sqsubseteq^? A,\ \forall r.X \sqsubseteq^? X\}$ The unification problem Γ^* cannot be solved in $\mathcal{FL}_\perp$, but it can in $\mathcal{FL}_{reg}$. X cannot be substituted by $\top$. A substitution for X may contain A or be $\perp$. The next subsumption creates a cycle, which can only be solved by $\top$. Hence there is no solution in $\mathcal{FL}_\perp$. The solution in $\mathcal{FL}_{reg}$ is $[X \mapsto \forall r^*.\perp]$

Properties of unifiers

Let γ be a unifier of Γ.

1. No redundant particles. If P is a redundant particle in $\gamma(X)$ it can be replaced by $\top$.
2. Minimal depth for $\perp$

   Each $\perp$-particle in $\gamma(\Gamma)$ is connected by a series of *solving relations* to $\perp$ in Γ or a $\perp$ in $\gamma(X)$, for a variable X. X is then called an *anchor variable*.

   For X to be an anchor variable, there must be a goal subsumption of the form: $C \sqcap \forall v.X \sqsubseteq^? \forall v'.Z$ in the unification problem.
3. Existance of $\mathcal{FL}_0$-unifier

   Let Γ be an $\mathcal{FL}_\perp$ unification problem such that $\perp$ does not appear as a symbol in Γ. If γ is a $\mathcal{FL}_\perp$ solution of Γ and there are no cycles among $\perp$-particles in $\gamma(\Gamma)$, then there is also an $\mathcal{FL}_0$ solution for Γ.

The Algorithm

1. $\perp$-elimination (flattening)

Implicit rule At each step of the following procedure, we implicitly apply the following rule that removes trivially solved equations or fails:

- if there is a goal subsumption $C \sqsubseteq^? P$ such that $\perp \in C$ or $P \in C$ or $P \equiv \top$, then remove this subsumption from the current unification problem.
- if there is a goal subsumption $C \sqsubseteq^? P$ such that $\forall v.\top \in C$, then delete the particle from C,
- **fail** at once if $C \sqsubseteq^? \perp$ is in the goal and $\perp \notin C$.

Step 1. In the first step we guess which variables in the goal contain bottom and we replace them with $\perp$. We keep the partial solution for the eliminated variables as a set of assignments ($[X \mapsto \perp]$).

Step 2. Consider the remaining subsumptions.

Let $s = C_1 \sqcap \dots \sqcap C_n \sqsubseteq^? D$:

1. If $D = \forall r.D'$, then replace s with s^{-r}, e.g. $\forall r.Y \sqcap B \sqcap X \sqsubseteq^? \forall r.\perp$ is replaced by $\forall r.Y \sqcap X \sqsubseteq^? \perp$
2. If D is a constant, then we replace s with s^D, e.g. $\forall r.Y \sqcap B \sqcap X \sqsubseteq^? A$ is replaced by $X \sqsubseteq^? A$
3. If D is a variable, s contains a particle of the form $\forall r.C' \in C$, since D is guessed not to contain $\perp$, we *split* D.
   (a) for each $r \in \mathbf{R}$, we add s^{-r}.
   (b) we guess for each constant A, if it should be in the substitution for D and if this is the case, we add the following goal subsumptions: $D \sqsubseteq^? A$ and $C_1^A \sqcap \dots \sqcap C_n^A \sqsubseteq^? A$,

   e.g. $\forall r.Y \sqcap B \sqcap X \sqsubseteq^? Z$ is replaced by $Y \sqcap B \sqcap X^r \sqsubseteq^? Z^r$, $X \sqsubseteq^? A$ and $Z \sqsubseteq^? A$

Note on decomposition variables

X^r, Z^r in the above examples are the so called decomposition variables. Their meaning is expressed in the following property: for any solution γ and any ground particle Q,
$\forall r.Q \in \gamma(X)$ if and only if $Q \in \gamma(X^r)$.

For "if" direction, for each X^r, we add the *increasing goal subsumption*: $X \sqsubseteq^? \forall r.X^r$.

The "only if" direction, must be enforced by a unification procedure. This is the so called *decreasing rule*: if $\forall r.C \in \gamma(X)$, then $C \in \gamma(X^r)$.

$\perp$-elimination fails or returns a unification problem Γ' which contains flat goal subsumptions and increasing subsumptions. $\perp$ does not occur in Γ', but there may be decomposition variables which have been guessed to be $\perp$.

2. Applying an $\mathcal{FL}_0$-unification procedure on the remaining goal

- There cannot be any cycles in $\gamma'(\Gamma')$, where γ' is a solution extended for decomposition variables. A subsumption of the form $C \sqcap \forall v.X^r \sqsubseteq^? \forall v'.Z$ for X^r guessed to be $\perp$, is impossible in Γ'. (Properties of unifiers: 2)
- Hence there is an $\mathcal{FL}_0$-unifier of Γ' obeying the decreasing rule. (Properties of unifiers: 3).

Therefore we use an $\mathcal{FL}_0$-unification procedure to detect such a unifier. This procedure must enforce the decreasing rule and it cannot increase the number of $\perp$-variables which were guessed in the $\perp$-elimination step. For this purpose we can use the procedure in [Morawska, 2020].

Examples

Consider the first example: $\Gamma = \{X \sqsubseteq^? A,\ X \sqsubseteq^? \forall r.X\}$. If we guess X to be $\perp$, both subsumptions will be removed by the implicit rule, as solved.

Consider the second example: $\Gamma^* = \{X \sqsubseteq^? A, \forall r.X \sqsubseteq^? X\}$. If we guess X to be $\perp$, the first subsumption is removed by the implicit rule, the second fails at once. If we do not guess X to be $\perp$, then the second subsumption will be split and we obtain the new goal: $\Gamma' = \{X \sqsubseteq^? A, X \sqsubseteq X^r, \top \sqsubseteq^? A, X \sqsubseteq \forall r.X^r\}$ which is passed to the $\mathcal{FL}_0$-unification. The $\mathcal{FL}_0$-unification procedure will fail by detecting that $\top \sqsubseteq^? A$ does not hold.

Conclusions

Theorem. (main result) For an $\mathcal{FL}_\perp$ unification problem Γ is decidable in ExpTime.

Proof. The $\perp$-elimination takes at most non-deterministic polynomial time before it terminates either with failure or with Γ'. The $\mathcal{FL}_0$-procedure runs in ExpTime in the worst case. □

Forthcoming Research

To get the exact complexity of the problem, one can try to modify the ExpTime-hardness argument from [Baader and Narendran, 2001] to apply to the unification in $\mathcal{FL}_\perp$. One can also extend the $\mathcal{FL}_\perp$ unification procedure to show that it is also working in the case of $\mathcal{FL}_\perp$ modulo a flat TBox.

References

[Baader and Küsters, 2001] Baader, F. and Küsters, R. (2001). Unification in a description logic with transitive closure of roles. In *Working Notes of DL2001*.

[Baader and Küsters, 2002] Baader, F. and Küsters, R. (2002). Unification in a description logic with inconsistency and transitive closure of roles. In *Proceedings of DL2002*.

[Baader and Narendran, 2001] Baader, F. and Narendran, P. (2001). Unification of concept terms in description logics. *Journal of Symbolic Computation*, 31(3):277–305.

[Morawska, 2020] Morawska, B. (2020). Unification in $\mathcal{FL}_0$ modulo a Flat TBox. In *Proceedings of DL2020*.

Acknowledgements

The author would like to thank the anonymous referees for their detailed critique which allowed the paper to become much clearer and more precise.