Dismatching and Local Disunification in $\mathcal{EL}$
(Extended Abstract)

Franz Baader, Stefan Borgwardt, and Barbara Morawska*

Theoretical Computer Science, TU Dresden, Germany
{baader,stefborg,morawska}@tcs.inf.tu-dresden.de

Introduction. Description Logics (DLs) [4] are a family of logic formalisms, which are designed specially to represent the conceptual knowledge of an application domain. DLs allow users to define classes and relations of the domain using *concepts* and *roles*, to formulate constraints on the domain by means of terminological axioms and to deduce consequences such as subsumption (subclass) relationships following from the definitions and constraints. The DL $\mathcal{EL}$ allows to construct concepts from atomic concept names and role names using the constructors conjunction ($C \sqcap D$), existential restriction ($\exists r.C$ for a role name r), and the top concept ($\top$).

Unification in DLs was introduced in [8] to detect redundancies between concept definitions in so-called ontologies. For example, assume that one developer of a medical ontology defines the concept of a *patient with severe head injury* using the $\mathcal{EL}$-concepts

$$\mathsf{Patient} \sqcap \exists \mathsf{finding}.(\mathsf{Head\_injury} \sqcap \exists \mathsf{severity}.\mathsf{Severe}), \tag{1}$$

whereas another one represents it as

$$\mathsf{Patient} \sqcap \exists \mathsf{finding}.(\mathsf{Severe\_finding} \sqcap \mathsf{Injury} \sqcap \exists \mathsf{finding\_site}.\mathsf{Head}). \tag{2}$$

Formally, these expressions are not equivalent, but they are nevertheless meant to represent the same concept. They can obviously be made equivalent by treating the concept names $\mathsf{Head\_injury}$ and $\mathsf{Severe\_finding}$ as variables, and substituting them by $\mathsf{Injury} \sqcap \exists \mathsf{finding\_site}.\mathsf{Head}$ and $\exists \mathsf{severity}.\mathsf{Severe}$, respectively. In this case, we say that the concepts are unifiable, and call the substitution that makes them equivalent a *unifier*. In [7], we were able to show that unification in $\mathcal{EL}$ is NP-complete. The main idea underlying the proof of this result is to show that any solvable $\mathcal{EL}$-unification problem has a local unifier, i.e., a unifier built from a polynomial number of so-called atoms determined by the unification problem. This yields a brute-force NP-algorithm for unification, which guesses a local substitution and then checks (in polynomial time) whether it is a unifier.

Intuitively, a unifier proposes definitions for the concept names that are used as variables: in our example, we know that, if we define $\mathsf{Head\_injury}$ as $\mathsf{Injury} \sqcap \exists \mathsf{finding\_site}.\mathsf{Head}$ and $\mathsf{Severe\_finding}$ as $\exists \mathsf{severity}.\mathsf{Severe}$, then the two concepts (1) and (2) are equivalent w.r.t. these definitions. Of course, this example was constructed such that the unifier (which is local) provides sensible definitions for the concept names used as variables. In general, the existence of a unifier only says that there is a structural similarity between the two concepts. The developer that uses unification needs to inspect the unifier(s) to see whether the definitions it suggests really make sense. For example, the substitution that replaces $\mathsf{Head\_injury}$ by $\mathsf{Patient} \sqcap \mathsf{Injury} \sqcap \exists \mathsf{finding\_site}.\mathsf{Head}$ and $\mathsf{Severe\_finding}$ by $\mathsf{Patient} \sqcap \exists \mathsf{severity}.\mathsf{Severe}$ is also a local unifier, which however does not make sense. Unfortunately, even small unification problems like the one in our example can have too many local unifiers for manual inspection. We propose disunification to avoid local unifiers that do not make sense. In addition to positive constraints (requiring equivalence or subsumption between concepts), a disunification problem may also

*Supported by DFG under grant BA 1122/14-1

Table 1: Syntax and semantics of $\mathcal{EL}$

Concept	Syntax	Semantics
top	$\top$	$\top^{\mathcal{I}} := \Delta^{\mathcal{I}}$
conjunction	$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$(\exists r.C)^{\mathcal{I}} := \{x \mid \exists y.(x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$

contain negative constraints (preventing equivalence or subsumption between concepts). In our example, the nonsensical unifier can be avoided by adding the dissubsumption constraint

$$\mathsf{Head\_injury} \not\sqsubseteq^? \mathsf{Patient} \tag{3}$$

to the equivalence constraint (1) $\equiv^?$ (2).

The main reason for unification in $\mathcal{EL}$ to be decidable in NP is locality: if the problem has a unifier then it has a local unifier. It turns out that disunification in $\mathcal{EL}$ is not local in this sense. Decidability and complexity of disunification in $\mathcal{EL}$ remains an open problem, but we provide partial solutions that are of interest in practice. On the one hand, we consider *dismatching problems*, i.e., disunification problems where the negative constraints are dissubsumptions $C \not\sqsubseteq^? D$ for which either C or D is ground. The dissubsumption (3) from above actually satisfies this restriction since $\mathsf{Patient}$ is not a variable. We prove that (general) solvability of dismatching problems can be reduced to *local disunification*, i.e., the question whether a given disunification problem has a *local* solution, which shows that dismatching in $\mathcal{EL}$ is NP-complete. On the other hand, we develop two specialized algorithms to solve local disunification problems that extend the ones for unification [6, 7]: a goal-oriented algorithm that reduces the amount of nondeterministic guesses necessary to find a local solution, as well as a translation to SAT. The reason we extend both algorithms is that, in the case of unification, they have proved to complement each other well in first evaluations [1]: the goal-oriented algorithm needs less memory and finds minimal solutions faster, while the SAT reduction generates larger data structures, but outperforms the goal-oriented algorithm on unsolvable problems.

Disunification in DLs is closely related to unification and admissibility in modal logics [9, 12–17], as well as (dis)unification modulo equational theories [7, 8, 10, 11]. In the following, we shortly describe the ideas behind our work. More details can be found in [2, 3].

Preliminaries. An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty domain $\Delta^{\mathcal{I}}$ and an interpretation function that maps concept names to subsets of $\Delta^{\mathcal{I}}$ and role names to binary relations over $\Delta^{\mathcal{I}}$. This function is extended to concepts as shown in Table 1. A concept C is *subsumed* by a concept D (written $C \sqsubseteq D$) if for every interpretation $\mathcal{I}$ it holds that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. The two concepts C and D are *equivalent* (written $C \equiv D$) if $C \sqsubseteq D$ and $D \sqsubseteq C$. Since conjunction is interpreted as intersection, we can treat $\sqcap$ as a commutative and associative operator, and thus dispense with parentheses in nested conjunctions. An *atom* is a concept name or an existential restriction. Hence, every concept term C is a conjunction of atoms or $\top$. We call the atoms in this conjunction the *top-level atoms* of C. Obviously, C is equivalent to the conjunction of its top-level atoms, where the empty conjunction corresponds to $\top$. An atom is *flat* if it is a of the form A or $\exists r.A$ for a concept name A.

Subsumption in $\mathcal{EL}$ is decidable in polynomial time [5] and can be checked by recursively comparing the top-level atoms of the two concept terms: for two atoms C, D, we have $C \sqsubseteq D$ iff $C = D$ is a concept name or $C = \exists r.C'$, $D = \exists r.D'$, and $C' \sqsubseteq D'$; if C, D are concepts, then $C \sqsubseteq D$ iff for every top-level atom D' of D there is a top-level atom C' of C such that $C' \sqsubseteq D'$.

This means that $C \not\sqsubseteq D$ (i.e., C is not subsumed by D) iff there is a top-level atom D' of D such that for all top-level atoms C' of C we have $C' \not\sqsubseteq D'$. By further analyzing the structure of atoms, we obtain the following.

Lemma 1. *Let C, D be two atoms. Then $C \not\sqsubseteq D$ iff either (i) C or D is a concept name and $C \neq D$; (ii) $D = \exists r.D'$, $C = \exists s.C'$, and $r \neq s$; or (iii) $D = \exists r.D'$, $C = \exists r.C'$, and $C' \not\sqsubseteq D'$.*

For the purposes of (dis)unification, we designate certain concept names as *variables*, while all others are *constants*. We consider *(basic) disunification problems*, which are conjunctions of *subsumptions* $C \sqsubseteq^? D$ and *dissubsumptions* $C \not\sqsubseteq^? D$ between concepts C, D.[1] A substitution σ *solves* a disunification problem Γ if the (dis)subsumptions of Γ become valid when applying σ on both sides. We here restrict substitutions to a finite signature of *constants* and role names—it would not make sense for the new definitions to extend the vocabulary of the ontologies under consideration, nor to define variables in terms of other variables.

We now consider a *flat* disunification problem Γ, i.e. one that contains only (dis)subsumptions where both sides are conjunctions of flat atoms. We denote by At the set of all such atoms that occur in Γ, by Var the set of variables occurring in Γ, and by $\mathsf{At}_{\mathsf{nv}} := \mathsf{At} \setminus \mathsf{Var}$ the set of *non-variable atoms* of Γ. Let $S\colon \mathsf{Var} \to 2^{\mathsf{At}_{\mathsf{nv}}}$ be an *assignment*, i.e. a function that assigns to each variable $X \in \mathsf{Var}$ a set $S_X \subseteq \mathsf{At}_{\mathsf{nv}}$. The relation $>_S$ on Var is defined as the transitive closure of $\{(X, Y) \in \mathsf{Var}^2 \mid Y \text{ occurs in an atom of } S_X\}$. If $>_S$ is irreflexive, then S is called *acyclic*. In this case, we can define the substitution σ_S inductively along $>_S$ as follows: if X is minimal, then $\sigma_S(X) := \bigsqcap_{D \in S_X} D$; otherwise, assume that $\sigma_S(Y)$ is defined for all $Y \in \mathsf{Var}$ with $X > Y$, and define $\sigma_S(X) := \bigsqcap_{D \in S_X} \sigma_S(D)$. Substitutions of this form are called *local*.

Unification in $\mathcal{EL}$ is *local*: each problem Γ can be transformed into an equivalent flat problem that has a local solution iff Γ is solvable, and hence (general) solvability of unification problems in $\mathcal{EL}$ is in NP [7]. However, disunification in $\mathcal{EL}$ is *not local* in this sense: consider

$$\Gamma := \{X \sqsubseteq^? B,\ A \sqcap B \sqcap C \sqsubseteq^? X,\ \exists r.X \sqsubseteq^? Y,\ \top \not\sqsubseteq^? Y,\ Y \not\sqsubseteq^? \exists r.B\} \tag{4}$$

with variables X, Y and constants A, B, C. If we set $\sigma(X) := A \sqcap B \sqcap C$ and $\sigma(Y) := \exists r.(A \sqcap C)$, then σ is a solution of Γ that is not local. This is because $\exists r.(A \sqcap C)$ is not a substitution of any non-variable atom in Γ. Assume now that Γ has a local solution γ. Since γ must solve the first dissubsumption, $\gamma(Y)$ cannot be $\top$, and due to the third subsumption, none of the constants A, B, C can be a conjunct of $\gamma(Y)$. The remaining atoms $\exists r.\gamma(X)$ and $\exists r.B$ are ruled out by the last dissubsumption since both $\gamma(X)$ and B are subsumed by B. This shows that Γ cannot have a local solution, although it is solvable. We call a *local disunification problem* a disunification problem asking for local solutions only. Thus (4) defined as a *local* disunification problem does not have solutions.

Obviously, deciding the existence of a local solution for a flat disunification problem is decidable in NP: We can guess an assignment S, and check it for acyclicity and whether σ_S solves the disunification problem in polynomial time. The corresponding complexity lower bound follows from NP-hardness of (local) solvability of unification problems in $\mathcal{EL}$ [7].

Dismatching in $\mathcal{EL}$ is NP-complete. A *dismatching* problem Γ is a disunification problem where one side of each dissubsumption is ground. Notice that Γ defined in (4) is in fact a dismatching problem. We show that such a problem can be polynomially reduced to a flat disunification problem that has a local solution iff Γ is solvable. This shows that deciding solvability of dismatching problems in $\mathcal{EL}$ is in NP. For a detailed description of the algorithm, see [3, Algorithm 8].

[1] A *unification problem* contains only subsumptions.

Based on Lemma 1, we designed a non-deterministic algorithm which applies transformation rules reducing dissubsumptions whenever possible. For example, we replace the first dissubsumption in (4), $\top \not\sqsubseteq^? Y$, with $Y \sqsubseteq^? \exists r.Z$. The rule we applied here is the following:

Solving Left-Ground Dissubsumptions:

Condition: This rule applies to $\mathfrak{s} = C_1 \sqcap \cdots \sqcap C_n \not\sqsubseteq^? X$ if X is a variable and $C_1, \ldots, C_n$ are ground atoms.
Action: Choose one of the following options:

- Choose a constant $A \in \Sigma$ and replace $\mathfrak{s}$ by $X \sqsubseteq^? A$. If $C_1 \sqcap \cdots \sqcap C_n \sqsubseteq A$, then *fail*.
- Choose a role $r \in \Sigma$, introduce a new variable Z, replace $\mathfrak{s}$ by $X \sqsubseteq^? \exists r.Z$, $C_1 \not\sqsubseteq^? \exists r.Z$, ..., $C_n \not\sqsubseteq^? \exists r.Z$, and immediately apply Atomic Decomposition to each of these dissubsumptions.

Notice that the rule involves a *don't know* nondeterministic choice. According to the rule, we can choose a constant or create a new existential restriction with a fresh variable, and use it in the new subsumption and dissubsumptions. In our example the left hand side of the dissubsumption $\top \not\sqsubseteq^? Y$ is empty, hence only a subsumption is produced. In effect, we obtain from (4) a new flat disunification problem:

$$\Gamma' := \{X \sqsubseteq^? B,\ A \sqcap B \sqcap C \sqsubseteq^? X,\ \exists r.X \sqsubseteq^? Y,\ Y \sqsubseteq^? \exists r.Z,\ Y \not\sqsubseteq^? \exists r.B\} \tag{5}$$

The main idea of the reduction of dismatching in $\mathcal{EL}$ to local disunification is to increase the set of *local* atoms in new subsumptions and dissubsumptions added to the original problem so as to be able to solve the dissubsumptions using a local substitution. In our example, such a new atom is $\exists r.Z$. In general disunification problems, this idea does not work, because we may be forced to add new variables and atoms *ad infinitum*. But in the case of dismatching, where one side of a dissubsumption is always ground, the number of new variables is restricted by the number of ground subconcepts in the ground sides of dissubsumptions. Hence the process terminates in polynomial time with a flat disunification problem whose local solution is also a solution for the original dismatching problem. Thus we can state the main result of this paper.

Theorem 2. *Deciding solvability of dismatching problems in $\mathcal{EL}$ is* NP*-complete.*

Goal-directed algorithm and SAT reduction to find local solutions. The brute-force NP-algorithm for checking local solvability of flat disunification problems is hardly practical. For this reason, we extended the rule-based algorithm from [7] and the SAT reduction from [6] by additional rules and propositional clauses, respectively, to deal with dissubsumptions.

For a given disunification problem, both algorithms attempt to define an acyclic assignment S of non-variable atoms to variables. In the case of the rule-based algorithm, the problem is transformed with every rule application. Rules apply to *unsolved* subsumptions or dissubsumptions, and this can cause extension of the assignment S and possibly the generation of new (dis)subsumptions. For example, solving Γ' in (5) would require adding $\exists r.Z$ to S_Y and this would trigger the addition of $\exists r.Z \not\sqsubseteq^? \exists r.B$ to Γ'. This dissubsumption would be immediately decomposed to $Z \not\sqsubseteq^? B$. Finally, since Z would have an empty set of non-variable atoms assigned ($S_Z = \emptyset$), we obtain the solution $\{X \mapsto B, Y \mapsto \exists r.\top, Z \mapsto \top\}$, which is a local solution of Γ' in (5) and a non-local solution of Γ in (4).

In the case of the algorithm reducing a local disunification problem to SAT, we use (among others) the propositional variables $[C \sqsubseteq D]$ for all $C, D \in \mathsf{At}$. The intuition is that a satisfying valuation of the propositional problem induces a solution σ such that $\sigma(C) \sqsubseteq \sigma(D)$ holds whenever $[C \sqsubseteq D]$ is true under the valuation. Properties of (dis)subsumptions in $\mathcal{EL}$ and the (dis)subsumptions of the problem are encoded with the help of these variables as propositional clauses. For example, the only dissubsumption in Γ' in (5) is encoded as $[Y \sqsubseteq \exists r.B] \rightarrow$. The subsumptions with more than one atom on the left-hand side (like the second subsumption in Γ') require non-Horn clauses. We prove that the set of clauses obtained by this translation of a disunification problem is satisfiable iff there is a local solution of the problem. For a detailed definition of the algorithms and more explanations, see [2].

The SAT reduction has been implemented in our prototype system UEL.[2] First experiments show that dismatching is helpful for reducing the number and size of unifiers. The performance of the solver for dismatching problems is comparable to the one for pure unification problems.

References

[1] Baader, F., Borgwardt, S., Mendez, J.A., Morawska, B.: UEL: Unification solver for $\mathcal{EL}$. In: Proc. of the 25th Int. Workshop on Description Logics (DL'12). CEUR-WS, vol. 846, pp. 26–36 (2012)

[2] Baader, F., Borgwardt, S., Morawska, B.: Dismatching and local disunification in $\mathcal{EL}$. LTCS-Report 15-03, TU Dresden (2015), see http://lat.inf.tu-dresden.de/research/reports.html.

[3] Baader, F., Borgwardt, S., Morawska, B.: Dismatching and local disunification in $\mathcal{EL}$. In: Proc. of the 26th Int. Conf. on Rewriting Techniques and Applications (RTA'15). LIPIcs, vol. 36. Dagstuhl Publishing (2015), to appear.

[4] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)

[5] Baader, F., Küsters, R., Molitor, R.: Computing least common subsumers in description logics with existential restrictions. In: Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI'99). pp. 96–101. Morgan Kaufmann (1999)

[6] Baader, F., Morawska, B.: SAT encoding of unification in $\mathcal{EL}$. In: Proc. of the 17th Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'10). LNCS, vol. 6397, pp. 97–111. Springer (2010)

[7] Baader, F., Morawska, B.: Unification in the description logic $\mathcal{EL}$. Log. Meth. Comput. Sci. 6(3) (2010)

[8] Baader, F., Narendran, P.: Unification of concept terms in description logics. J. Symb. Comput. 31(3), 277–305 (2001)

[9] Babenyshev, S., Rybakov, V.V., Schmidt, R., Tishkovsky, D.: A tableau method for checking rule admissibility in $S4$. In: Proc. of the 6th Workshop on Methods for Modalities (M4M-6) (2009)

[10] Buntine, W.L., Bürckert, H.J.: On solving equations and disequations. J. of the ACM 41(4), 591–629 (1994)

[11] Comon, H.: Disunification: A survey. In: Lassez, J.L., Plotkin, G. (eds.) Computational Logic: Essays in Honor of Alan Robinson, pp. 322–359. MIT Press, Cambridge, MA (1991)

[12] Ghilardi, S.: Unification through projectivity. J. Logic Comput. 7(6), 733–752 (1997)

[13] Ghilardi, S.: Unification in intuitionistic logic. J. Symb. Logic 64(2), 859–880 (1999)

[14] Iemhoff, R., Metcalfe, G.: Proof theory for admissible rules. Ann. Pure Appl. Logic 159(1-2), 171–186 (2009)

[15] Rybakov, V.V.: Admissibility of logical inference rules, Studies in Logic and the Foundations of Mathematics, vol. 136. North-Holland Publishing Co., Amsterdam (1997)

[16] Rybakov, V.V.: Multi-modal and temporal logics with universal formula - reduction of admissibility to validity and unification. J. Logic Comput. 18(4), 509–519 (2008)

[17] Wolter, F., Zakharyaschev, M.: Undecidability of the unification and admissibility problems for modal and description logics. ACM T. Comput. Log. 9(4), 25:1–25:20 (2008)

[2] version 1.3.0, available at `http://uel.sourceforge.net/`