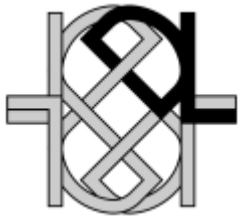
Unification in the Description Logic $\mathcal{EL}$

Franz Baader
Barbara Morawska

TU Dresden
Germany

Description Logics

research of the last 20 years



Phase 1:

- implementation of systems (Back, K-Rep, Loom, Meson, . . .)
- based on incomplete structural subsumption algorithms

Phase 2:

- development of tableau-based algorithms and complexity results
- first implementation of tableau-based systems (Kris, Crack)
- first formal investigation of optimization methods

Phase 3:

- tableau-based algorithms for very expressive DLs
- highly optimized tableau-based systems (FaCT, Racer)
- relationship to modal logic and decidable fragments of FOL

Phase 4:

- Web Ontology Language (OWL-DL) based on very expressive DL
- industrial-strength reasoners and ontology editors for OWL-DL
- investigation of light-weight DLs with tractable reasoning problems

The Description Logic $\mathcal{EL}$

conjunction $C \sqcap D$,

existential restriction $\exists r.C$,

top concept $\top$

$$Animal \sqcap \exists color.Green \sqcap \exists sits\_on.Leaf$$

DL with restricted expressive power

- no value restrictions $\forall r.C$
- $\mathcal{EL}$ has better algorithmic properties than $\mathcal{FL}_0$, the corresponding DL with value restrictions
- can represent large biomedical ontologies: SNOMED and the Gene Ontology

Dresden

Formal semantics and reasoning

An interpretation $\mathcal{I}$ has a domain $\Delta^{\mathcal{I}}$ and associates

- concepts C with sets $C^{\mathcal{I}}$, and
- roles r with binary relations $r^{\mathcal{I}}$.

The semantics of the constructors is defined through identities:

- $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$,
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$,
- $(\exists r.C)^{\mathcal{I}} = \{d \mid \exists e.(d,e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$.
- $(\forall r.C)^{\mathcal{I}} = \{d \mid \forall e.(d,e) \in r^{\mathcal{I}} \rightarrow e \in C^{\mathcal{I}}\}$.

The subsumption and the equivalence problem:

- C is subsumed by D ($C \sqsubseteq D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all interpretations $\mathcal{I}$
- C and D are equivalent ($C \equiv D$) iff $C \sqsubseteq D$ and $D \sqsubseteq C$

both problems are polynomial for $\mathcal{EL}$

Equational theory

point of view

The equivalence problem in $\mathcal{EL}$ is the same as
the word problem for the theory of semilattices with monotone operators

- $\sqcap$ is associative, commutative, and idempotent
- $\top$ is a unit for $\sqcap$

} ACIU

- $\exists r.(C \sqcap D) \sqcap \exists r.D \equiv \exists r.(C \sqcap D)$

Value restrictions satisfy the stronger identity

$$\forall r.(C \sqcap D) \equiv \forall r.C \sqcap \forall r.D$$

Unification in DLs

motivation

Avoid the insertion of redundant concepts into large ontologies like SNOMED:

equivalence test $C \equiv D$ not sufficient:

- different modellers may use different concept names:

  $Male$ versus $Masculine$

- different modellers may model on different levels of granularity:

  $Human \sqcap Male \sqcap \exists loves.Sports\_car$

  $\not\equiv$ versus

  $Man \sqcap \exists loves.(Car \sqcap Fast)$

can be made equivalent by applying the substitution:

$$Man \mapsto Human \sqcap Male$$

$$Sports\_car \mapsto Car \sqcap Fast$$

Dresden

Unification in DLs

definition

Partition the set of concept names into

- concept variables and
- concept constants

Substitutions can replace concept variables by concept terms.

Unification problem

$$\Gamma = \{C_1 \equiv^? D_1, \ldots, C_n \equiv^? D_n\}$$

Matching problem

no variables in $D_1, \ldots, D_n$

ground

Unifier of Γ

substitution σ with $\sigma(C_1) \equiv \sigma(D_1), \ldots, \sigma(C_n) \equiv \sigma(D_n)$

Decision problem

given a unification problem Γ, decide whether it has a unifier

Unification type

cardinality and existence of
minimal complete sets of unifiers

Instantiation preorder:

variables occurring in the unification problem

$$\sigma \leq \gamma \quad \text{iff} \quad \exists\lambda.\ \lambda(\sigma(X)) \equiv \gamma(X)$$

Minimal complete set of unifiers of Γ: set M of unifiers of Γ that is

Complete: for all unifiers θ of Γ there is $\sigma \in M$ with $\sigma \leq \theta$

Minimal: for all $\sigma, \theta \in M$ we have: $\sigma \leq \theta \longrightarrow \sigma = \theta$

Unification type: minimal complete sets of unifiers

- unitary, finitary, infinitary:
  always exist and have
  cardinality ≤ 1, finite cardinality, possibly infinite cardinality
- type zero: need not exist

Unification in DLs

previous results

Results for $\mathcal{FL}_0$ and $\mathcal{EL}$:

- Unification in $\mathcal{FL}_0$ is of type zero.
- Unification in $\mathcal{FL}_0$ is ExpTime-complete.

} [Baader, Narendran; 2001]

- Matching in $\mathcal{FL}_0$ is polynomial.
- Matching in $\mathcal{EL}$ is NP-complete. [Küster; 2001]

Results for $\mathcal{ALC}$ (closure of $\mathcal{FL}_0$ / $\mathcal{EL}$ under negation):

- Decision problem and unification type: open.
- Same for matching: unification can be reduced to matching.
- Undecidability results for small extensions. [Wolter, Zakharyaschev; 2008]

Dresden

Unification in $\mathcal{EL}$

our new results [Baader, Morawska; RTA'09]

Unification type

Unification in $\mathcal{EL}$ is of unification type zero.

Decision problem

Unification in $\mathcal{EL}$ is NP-complete.

Dresden

Unification in $\mathcal{EL}$

our new results [Baader, Morawska; RTA'09]

Unification type

Unification in $\mathcal{EL}$ is of unification type zero:

$$\{X \sqcap \exists r.Y \equiv^? \exists r.Y\}$$

does not have a minimal complete set of unifiers

Assume to the contrary that M is such a set.

- There is $\sigma \in M$ with $\sigma(X) \not\equiv \top$ and $\sigma(X) \not\equiv \exists r.\top$.
- If we define $\widehat{\sigma}$ as $\widehat{\sigma}(X) := \sigma(X) \sqcap \exists r.Z$ and $\widehat{\sigma}(Y) := \sigma(Y) \sqcap Z$ then $\widehat{\sigma} < \sigma$.
- There is $\sigma' \in M$ with $\sigma' \leq \widehat{\sigma} < \sigma$. *Contradicts minimality of M.*

Dresden

Unification in $\mathcal{EL}$

the decision problem

1. $\mathcal{EL}$-concept terms have a reduced form that is unique modulo AC [Küsters, 2001].

2. Define an appropriate well-founded order on substitutions:
   every solvable $\mathcal{EL}$-unification problem has a minimal reduced ground unifier.

3. Minimal reduced ground unifiers are local:
   built from "atoms" occurring in the (appropriately normalized) unification problem.

4. Guess and test algorithm:
   - guess a candidate for such a unifier
   - check whether it solves the unification problem
     using the P-time algorithm for equivalence

Reduced form

of $\mathcal{EL}$-concept terms [Küster; 2001]

A given $\mathcal{EL}$-concept term can be transformed into an equivalent reduced term by applying the following rules modulo associativity and commutativity of conjunction:

$C \sqcap \top \rightarrow C$	for all $\mathcal{EL}$-concept terms C
$A \sqcap A \rightarrow A$	for all concept names A
$\exists r.C \sqcap \exists r.D \rightarrow \exists r.C$	for all $\mathcal{EL}$-concept terms C, D with $C \sqsubseteq D$

Theorem

Let $\widehat{C}, \widehat{D}$ be reduced forms of C, D, respectively.

$$C \equiv D \quad \Leftrightarrow \quad \widehat{C} =_{AC} \widehat{D}$$

Dresden

Well-founded order

inverse subsumption order

There is no infinite sequence $C_0, C_1, C_2, C_3, \ldots$ of $\mathcal{EL}$-concept terms such that

$$C_0 \sqsubset C_1 \sqsubset C_2 \sqsubset C_3 \sqsubset \cdots \qquad C_i \sqsubseteq D_i, C_i \not\equiv D_i$$

Extend to substitutions:

$\sigma \succ \theta$

- consider the terms in the range of the substitution
- compare the multisets of these terms with multiset-extension of $\sqsubset$

well-founded

Theorem

Every solvable $\mathcal{EL}$-unification problem has a minimal reduced ground unifier.

Dresden

Normal form

of unification problems

Atom

- concept name, i.e., concept constant or concept variable, or
- existential restriction $\exists r.D$

Flat atom

- concept name, or
- existential restriction $\exists r.D$, where D is a concept name or $\top$

Every $\mathcal{EL}$-unification problem is equivalent to a flat $\mathcal{EL}$-unification problem, i.e, one that contains only equations of the form:

$$C_1 \sqcap \ldots \sqcap C_m \equiv^? D_1 \sqcap \ldots \sqcap D_n$$

where $C_1, \ldots, C_m, D_1, \ldots, D_n$ are flat atoms.

Locality

of minimal reduced ground unifiers

main technical result

The following holds for every pair of

- flat $\mathcal{EL}$-unification problem Γ and
- minimal reduced ground unifier γ of Γ.

If X is a concept variable occurring in Γ, then

- $\gamma(X) \equiv \top$ or
- there are non-variable atoms $D_1, \ldots, D_n$ ($n \geq 1$) of Γ such that $\gamma(X) \equiv \gamma(D_1) \sqcap \ldots \sqcap \gamma(D_n)$.

Dresden

Non-deterministic polynomial-time algorithm for deciding solvability of a given flat $\mathcal{EL}$-unification problem Γ:

1. For every variable X occurring in Γ, guess a finite, possibly empty, set S_X of non-variable atoms of Γ.

2. We say that the variable X directly depends on the variable Y if Y occurs in an atom of S_X. Let depends on be the transitive closure of directly depends on. If there is a variable that depends on itself, then the algorithm returns "fail." Otherwise, there exists a strict linear order $>$ on the variables occurring in Γ such that $X > Y$ if X depends on Y.

3. We define the substitution σ along the linear order $>$:

   - If X is the least variable w.r.t. $>$, then S_X does not contain any variables. We define $\sigma(X)$ to be the conjunction of the elements of S_X, where the empty conjunction is $\top$.
   - Assume that $\sigma(Y)$ is defined for all variables $Y < X$. Then S_X only contains variables Y for which $\sigma(Y)$ is already defined. If S_X is empty, then we define $\sigma(X) := \top$. Otherwise, let $S_X = \{D_1, \ldots, D_n\}$. We define $\sigma(X) := \sigma(D_1) \sqcap \ldots \sqcap \sigma(D_n)$.

4. Test whether the substitution σ computed in the previous step is a unifier of Γ. If this is the case, then return σ; otherwise, return "fail."

structure sharing needed

Dresden

Practical algorithms

for $\mathcal{EL}$-unification

The NP-algorithm presented until now is not usable in practice

- brutal "guess and then test" algorithm
- many non-deterministic choices even for very simple unification problems

We have developed two more practical algorithms:

- transformation-based algorithm that makes non-deterministic choices "only if necessary"
- translation into SAT, where the non-determinism is then dealt with by a highly optimzed SAT solver (MiniSat)

Correctness proofs require the locality result!

Transformation-based algorithm

for $\mathcal{EL}$-unification

Works on a data structure consisting of

- a flat unification problem
- a current substitution,
  which is induced by an acyclic collection of sets S_X — *current assignment*

Applies three types of transformation rules:

- Eager-Assignment: deterministic rule that is applied eagerly
- Decomposition and Extension:
  non-deterministic rules that try to solve an unsolved atom — *non-variable atom occurring on one side of an equation, but not the other side*

A run of the non-deterministic transformation algorithm

- Fails: if a rule application makes the current assignment cyclic, or there is an unsolved atom to which neither Decomposition nor Extension applies.
- Succeeds: if there are no unsolved atoms and the current assignment is acyclic.

Transformation rules

illustrated on an example

$$Z \equiv^? \exists r.A$$
$$Z \sqcap X \equiv^? A \sqcap \exists r.Y$$

$S_X = S_Y = S_Z = \emptyset$

↓ Eager-Assignment

$$Z \sqcap \exists r.A \equiv^? \exists r.A$$
$$Z \sqcap \exists r.A \sqcap X \equiv^? A \sqcap \exists r.Y$$

$S_X = S_Y = \emptyset, S_Z = \{\exists r.A\}$

Z finished

↓ Extension

$$Z \sqcap \exists r.A \equiv^? \exists r.A$$
$$Z \sqcap \exists r.A \sqcap X \sqcap A \equiv^? A \sqcap \exists r.Y$$

$S_X = \{A\}, S_Y = \emptyset, S_Z = \{\exists r.A\}$

↓ Decomposition

$$Z \sqcap \exists r.A \equiv^? \exists r.A$$
$$Z \sqcap \exists r.A \sqcap X \sqcap A \equiv^? A \sqcap \exists r.Y \sqcap \exists r.A$$
$$Y \sqcap A \equiv^? Y$$

$S_X = \{A\}, S_Y = \emptyset, S_Z = \{\exists r.A\}$

Transformation rules

illustrated on an example

$$Z \sqcap \exists r.A \equiv^? \exists r.A$$
$$Z \sqcap \exists r.A \sqcap X \sqcap A \equiv^? A \sqcap \exists r.Y \sqcap \exists r.A$$
$$Y \sqcap A \equiv^? Y$$

$S_X = \{A\}, S_Y = \emptyset, S_Z = \{\exists r.A\}$

↓ Extension

$$Z \sqcap \exists r.A \equiv^? \exists r.A$$
$$Z \sqcap \exists r.A \sqcap X \sqcap A \equiv^? A \sqcap \exists r.Y \sqcap \exists r.A$$
$$Y \sqcap A \equiv^? Y \sqcap A$$

$S_X = \{A\}, S_Y = \{A\}, S_Z = \{\exists r.A\}$

↓ Decomposition

$$Z \sqcap \exists r.A \equiv^? \exists r.A$$
$$Z \sqcap \exists r.A \sqcap \exists r.Y \sqcap X \sqcap A \equiv^? A \sqcap \exists r.Y \sqcap \exists r.A$$
$$Y \sqcap A \equiv^? Y \sqcap A$$
$$Y \sqcap A \equiv^? A$$

$S_X = \{A\}, S_Y = \{A\}, S_Z = \{\exists r.A\}$

no unsolved atoms

unifier $\{X \mapsto A, Y \mapsto A, Z \mapsto \exists r.A\}$

Dresden

Transformation rules

illustrated on an example

$$Z \sqcap \exists r.A \equiv^? \exists r.A$$
$$Z \sqcap \exists r.A \sqcap X \sqcap A \equiv^? A \sqcap \exists r.Y \sqcap \exists r.A$$
$$Y \sqcap A \equiv^? Y$$

$S_X = \{A\}, S_Y = \emptyset, S_Z = \{\exists r.A\}$

↓ Extension

$$Z \sqcap \exists r.A \equiv^? \exists r.A$$
$$Z \sqcap \exists r.A \sqcap X \sqcap A \equiv^? A \sqcap \exists r.Y \sqcap \exists r.A$$
$$Y \sqcap A \equiv^? Y \sqcap A$$

$S_X = \{A\}, S_Y = \{A\}, S_Z = \{\exists r.A\}$

↓ Extension *alternative non-deterministic choice*

$$Z \sqcap \exists r.A \equiv^? \exists r.A$$
$$Z \sqcap \exists r.A \sqcap X \sqcap \exists r.Y \sqcap A \equiv^? A \sqcap \exists r.Y \sqcap \exists r.A$$
$$Y \sqcap A \equiv^? Y \sqcap A$$

$S_X = \{A, \exists r.Y\}, S_Y = \{A\}, S_Z = \{\exists r.A\}$

no unsolved atoms

unifier $\{X \mapsto A \sqcap \exists r.A, Y \mapsto A, Z \mapsto \exists r.A\}$

Dresden

Translation into SAT

for $\mathcal{EL}$-unification

Uses two types of propositional variables:

- $[A \not\sqsubseteq B]$ for atoms A, B of the flat unification problem: guess non-subsumptions that hold after applying the unifier
- $[X > Y]$ for variables X, Y of the flat unification problem: guess the "depends on" relation and prevent cycles in it

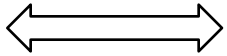
Creates propositional clauses:

- Horn clauses that encode the equations in the spirit of the Kapur&Narendran translation of ACIU-unification into HornSAT.
- Horn clauses that encode properties of (non-)subsumption in $\mathcal{EL}$ and the fact that $>$ is a strict order.
- Non-Horn clauses that encode transitivity of subsumption and properties of the "depends on" relation.

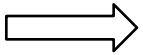
Translation into SAT

correctness

The created set of clauses is satisfiable



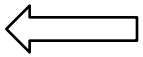
The unification problem has a minimal reduced ground unifier



A satisfying valuation of the clauses yields the acyclic assignment

$$S_X := \{C \mid [X \not\sqsubseteq C] = \mathsf{false} \wedge C \text{ non-variable atom}\}$$

and the induced substitution is a unifier.



A minimal reduced ground unifier σ defines a satisfying valuation of the clauses:

- $[C \not\sqsubseteq D] = \mathsf{true}$ iff $\sigma(C) \not\sqsubseteq \sigma(D)$
- $[X > Y] = \mathsf{true}$ iff $\sigma(X) \sqsubseteq \exists r_1. \cdots \exists r_n.\sigma(Y)$ for $n \geq 1$ roles $r_1, \ldots, r_n$

Dresden

Conclusion

We have shown that

- $\mathcal{EL}$-unification is of unification type zero
- $\mathcal{EL}$-unification is NP-complete
- more practical decision procedures than the brutal "guess and then test" algorithm exist

Future work:

- test the more practical algorithms on medical ontologies like SNOMED CT
- extension to equivalence modulo general inclusion axioms
- extension by other concept constructors