Unification in Description Logics

Part II: Unification in the DL $\mathcal{EL}$

Oliver Fernández Gil

Chair of Automata Theory

TECHNISCHE UNIVERSITÄT DRESDEN

ESSLLI'19

Riga, August 2019

The lightweight DL $\mathcal{EL}$

- Fragment of $\mathcal{ALC}$:

$$C ::= \top \mid A \mid C \sqcap C \mid \exists r.C$$

The lightweight DL $\mathcal{EL}$

- Fragment of $\mathcal{ALC}$:

$$C ::= \top \mid A \mid C \sqcap C \mid \exists r.C$$

- Subsumption is polynomial, even w.r.t. general TBoxes [BBL05].

The lightweight DL $\mathcal{EL}$

- Fragment of $\mathcal{ALC}$:

$$C ::= \top \mid A \mid C \sqcap C \mid \exists r.C$$

- Subsumption is polynomial, even w.r.t. general TBoxes [BBL05].

- Underlies the OWL 2 EL profile and can be used to define large biomedical ontologies, such as SNOMED CT.

Characterization of equivalence/subsumption

Characterization of equivalence/subsumption

Characterization of equivalence

Characterization of equivalence/subsumption

Characterization of equivalence

$\mathcal{EL}$ concept descriptions C can be translated into an equivalent reduced form C^r:

Characterization of equivalence/subsumption

Characterization of equivalence

$\mathcal{EL}$ concept descriptions C can be translated into an equivalent reduced form C^r:

Apply the rewrite rule $C \sqcap D \rightarrow C$ if $C \sqsubseteq D$ (as long as possible).

Characterization of equivalence/subsumption

Characterization of equivalence

$\mathcal{EL}$ concept descriptions C can be translated into an equivalent reduced form C^r:

Apply the rewrite rule $C \sqcap D \to C$ if $C \sqsubseteq D$ (as long as possible).

Theorem 4 [Küs01]

In $\mathcal{EL}$, $C \equiv D$ iff $C^r = D^r$ (modulo associativity/commutativity of $\sqcap$).

Characterization of equivalence/subsumption

Characterization of equivalence

$\mathcal{EL}$ concept descriptions C can be translated into an equivalent reduced form C^r:

Apply the rewrite rule $C \sqcap D \to C$ if $C \sqsubseteq D$ (as long as possible).

Theorem 4 [Küs01]

In $\mathcal{EL}$, $C \equiv D$ iff $C^r = D^r$ (modulo associativity/commutativity of $\sqcap$).

Characterization of subsumption

Characterization of equivalence/subsumption

Characterization of equivalence

$\mathcal{EL}$ concept descriptions C can be translated into an equivalent reduced form C^r:

Apply the rewrite rule $C \sqcap D \rightarrow C$ if $C \sqsubseteq D$ (as long as possible).

Theorem 4 [Küs01]

In $\mathcal{EL}$, $C \equiv D$ iff $C^r = D^r$ (modulo associativity/commutativity of $\sqcap$).

Characterization of subsumption

Corollary 5 [BM10b]

Let C and D be

$$C = A_1 \sqcap \ldots \sqcap A_k \sqcap \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_m.C_m, \text{ and}$$
$$D = B_1 \sqcap \ldots \sqcap B_\ell \sqcap \exists s_1.D_1 \sqcap \ldots \sqcap \exists s_n.D_n, \text{ where}$$

$A_1, \ldots, A_k, B_1, \ldots, B_\ell \in \mathsf{N_C}$. Then $C \sqsubseteq D$ **iff** $\{B_1, \ldots, B_\ell\} \subseteq \{A_1, \ldots, A_k\}$ and for every j, $1 \leq j \leq n$, there exists an i, $1 \leq i \leq m$, such that $r_i = s_j$ and $C_i \sqsubseteq D_j$.

$\mathcal{EL}$ has unification type zero

An $\mathcal{EL}$ unification problem of type zero

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

$\mathcal{EL}$ has unification type zero

An $\mathcal{EL}$ unification problem of type zero

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

To show: every complete set of unifiers $\mathcal{M}$ of Γ is not minimal, i.e.,

$$\mathcal{M} \text{ contains } \sigma \neq \gamma \text{ such that } \sigma \preceq \gamma.$$

$\mathcal{EL}$ has unification type zero

An $\mathcal{EL}$ unification problem of type zero

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

To show: every complete set of unifiers $\mathcal{M}$ of Γ is not minimal, i.e.,

$$\mathcal{M} \text{ contains } \sigma \neq \gamma \text{ such that } \sigma \preceq \gamma.$$

Proof sketch.

- It is easy to find "a" solution for Γ:

$\mathcal{EL}$ has unification type zero

An $\mathcal{EL}$ unification problem of type zero

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

To show: every complete set of unifiers $\mathcal{M}$ of Γ is not minimal, i.e.,

$$\mathcal{M} \text{ contains } \sigma \neq \gamma \text{ such that } \sigma \preceq \gamma.$$

Proof sketch.

- It is easy to find "a" solution for Γ:

$$\begin{array}{lll} X \to \top & X \to \exists r.\top & X \to \exists r.A \\ Y \to Y & Y \to Y & Y \to A \end{array}$$

$\mathcal{EL}$ has unification type zero

An $\mathcal{EL}$ unification problem of type zero

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

To show: every complete set of unifiers $\mathcal{M}$ of Γ is not minimal, i.e.,

$$\mathcal{M} \text{ contains } \sigma \neq \gamma \text{ such that } \sigma \preceq \gamma.$$

Proof sketch.

- It is easy to find "a" solution for Γ:

$$\begin{array}{lll} X \rightarrow \top & X \rightarrow \exists r.\top & X \rightarrow \exists r.A \\ Y \rightarrow Y & Y \rightarrow Y & Y \rightarrow A \end{array}$$

- However, the green solution implies that $\mathcal{M}$ contains σ such that:

$$\sigma(X) \not\equiv \top \qquad \text{and} \qquad \sigma(X) \not\equiv \exists r.\top.$$

$\mathcal{EL}$ has unification type zero

- $\mathcal{M}$ contains σ of the form:

$$\sigma(X) = \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_n.C_n \quad (n > 0)$$
$$\sigma(Y) = D$$

$\mathcal{EL}$ has unification type zero

- $\mathcal{M}$ contains σ of the form:

$$\begin{aligned}\sigma(X) &= \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_n.C_n \quad (n > 0)\\ \sigma(Y) &= D\end{aligned}$$

- From σ, we build $\widehat{\sigma}$ using a new variable Z:

$$\widehat{\sigma}(X) := \sigma(X) \sqcap \exists r_1.Z \qquad \text{and} \qquad \widehat{\sigma}(Y) := \sigma(Y) \sqcap Z.$$

$\mathcal{EL}$ has unification type zero

- $\mathcal{M}$ contains σ of the form:

$$\sigma(X) = \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_n.C_n \quad (n > 0)$$
$$\sigma(Y) = D$$

- From σ, we build $\widehat{\sigma}$ using a new variable Z:

$$\widehat{\sigma}(X) := \sigma(X) \sqcap \exists r_1.Z \qquad \text{and} \qquad \widehat{\sigma}(Y) := \sigma(Y) \sqcap Z.$$

- One can prove the following about $\widehat{\sigma}$:

$\mathcal{EL}$ has unification type zero

- $\mathcal{M}$ contains σ of the form:

$$\sigma(X) = \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_n.C_n \quad (n > 0)$$
$$\sigma(Y) = D$$

- From σ, we build $\widehat{\sigma}$ using a new variable Z:

$$\widehat{\sigma}(X) := \sigma(X) \sqcap \exists r_1.Z \qquad \text{and} \qquad \widehat{\sigma}(Y) := \sigma(Y) \sqcap Z.$$

- One can prove the following about $\widehat{\sigma}$:
  - $\widehat{\sigma}$ is also a unifier of Γ (characterization of $\sqsubseteq$),

$\mathcal{EL}$ has unification type zero

- $\mathcal{M}$ contains σ of the form:

$$\sigma(X) = \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_n.C_n \quad (n > 0)$$
$$\sigma(Y) = D$$

- From σ, we build $\widehat{\sigma}$ using a new variable Z:

$$\widehat{\sigma}(X) := \sigma(X) \sqcap \exists r_1.Z \qquad \text{and} \qquad \widehat{\sigma}(Y) := \sigma(Y) \sqcap Z.$$

- One can prove the following about $\widehat{\sigma}$:
    - $\widehat{\sigma}$ is also a unifier of Γ (characterization of $\sqsubseteq$),
    - $\widehat{\sigma} \preceq \sigma$ ($Z \rightarrow C_1$ and $D \sqsubseteq C_1$),

$\mathcal{EL}$ has unification type zero

- $\mathcal{M}$ contains σ of the form:

$$\sigma(X) = \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_n.C_n \quad (n > 0)$$
$$\sigma(Y) = D$$

- From σ, we build $\widehat{\sigma}$ using a new variable Z:

$$\widehat{\sigma}(X) := \sigma(X) \sqcap \exists r_1.Z \qquad \text{and} \qquad \widehat{\sigma}(Y) := \sigma(Y) \sqcap Z.$$

- One can prove the following about $\widehat{\sigma}$:
  - $\widehat{\sigma}$ is also a unifier of Γ (characterization of $\sqsubseteq$),
  - $\widehat{\sigma} \preceq \sigma$ ($Z \to C_1$ and $D \sqsubseteq C_1$),
  - $\sigma \not\preceq \widehat{\sigma}$ (characterization of $\equiv$).

$\mathcal{EL}$ has unification type zero

- $\mathcal{M}$ contains σ of the form:

$$\sigma(X) = \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_n.C_n \quad (n > 0)$$
$$\sigma(Y) = D$$

- From σ, we build $\widehat{\sigma}$ using a new variable Z:

$$\widehat{\sigma}(X) := \sigma(X) \sqcap \exists r_1.Z \qquad \text{and} \qquad \widehat{\sigma}(Y) := \sigma(Y) \sqcap Z.$$

- One can prove the following about $\widehat{\sigma}$:
  - $\widehat{\sigma}$ is also a unifier of Γ (characterization of $\sqsubseteq$),
  - $\widehat{\sigma} \preceq \sigma$ ($Z \to C_1$ and $D \sqsubseteq C_1$),
  - $\sigma \neq \widehat{\sigma}$ (characterization of $\equiv$).

- $\widehat{\sigma}$ need not be in $\mathcal{M}$, but:

$\mathcal{EL}$ has unification type zero

- $\mathcal{M}$ contains σ of the form:

$$\sigma(X) = \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_n.C_n \quad (n > 0)$$
$$\sigma(Y) = D$$

- From σ, we build $\widehat{\sigma}$ using a new variable Z:

$$\widehat{\sigma}(X) := \sigma(X) \sqcap \exists r_1.Z \qquad \text{and} \qquad \widehat{\sigma}(Y) := \sigma(Y) \sqcap Z.$$

- One can prove the following about $\widehat{\sigma}$:
  - $\widehat{\sigma}$ is also a unifier of Γ (characterization of $\sqsubseteq$),
  - $\widehat{\sigma} \preceq \sigma$ ($Z \to C_1$ and $D \sqsubseteq C_1$),
  - $\sigma \not\preceq \widehat{\sigma}$ (characterization of $\equiv$).

- $\widehat{\sigma}$ need not be in $\mathcal{M}$, but:

there is $\tau \in \mathcal{M}$ s.t.
$$\tau \preceq \widehat{\sigma}$$

$\mathcal{EL}$ has unification type zero

- $\mathcal{M}$ contains σ of the form:

$$\sigma(X) = \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_n.C_n \quad (n > 0)$$
$$\sigma(Y) = D$$

- From σ, we build $\widehat{\sigma}$ using a new variable Z:

$$\widehat{\sigma}(X) := \sigma(X) \sqcap \exists r_1.Z \qquad \text{and} \qquad \widehat{\sigma}(Y) := \sigma(Y) \sqcap Z.$$

- One can prove the following about $\widehat{\sigma}$:
  - $\widehat{\sigma}$ is also a unifier of Γ (characterization of $\sqsubseteq$),
  - $\widehat{\sigma} \preceq \sigma$ ($Z \to C_1$ and $D \sqsubseteq C_1$),
  - $\sigma \not\equiv \widehat{\sigma}$ (characterization of $\equiv$).

- $\widehat{\sigma}$ need not be in $\mathcal{M}$, but:

there is $\tau \in \mathcal{M}$ s.t. $\tau \preceq \widehat{\sigma}$ $\quad \Rightarrow \quad$ $\tau \preceq \widehat{\sigma} \preceq \sigma$, $\tau \neq \sigma$

The decision problem. NP-hardness

The decision problem. NP-hardness

Idea: reduce the propositional satisfiability problem to $\mathcal{EL}$-unification.

The decision problem. NP-hardness

Idea: reduce the propositional satisfiability problem to $\mathcal{EL}$-unification.

SAT Problem

Instance:	A propositional formula φ in CNF: $\varphi = c_1 \wedge \ldots \wedge c_m$, where each c_i is a disjunction of literals.
Question:	Is there an assignment $t : \mathrm{Vars}(\varphi) \rightarrow \{\mathrm{t}, \mathrm{f}\}$ satisfying φ?

The decision problem. NP-hardness

Idea: reduce the propositional satisfiability problem to $\mathcal{EL}$-unification.

SAT Problem

Instance:	A propositional formula φ in CNF: $\varphi = c_1 \wedge \ldots \wedge c_m$, where each c_i is a disjunction of literals.
Question:	Is there an assignment $t : \mathsf{Vars}(\varphi) \to \{\mathsf{t}, \mathsf{f}\}$ satisfying φ?

Given φ, we build an $\mathcal{EL}$-unification problem Γ_φ such that:

φ is satisfiable **if, and only if**, Γ_φ has a unifier.

The decision problem. NP-hardness (the reduction)

The decision problem. NP-hardness (the reduction)

1. Encoding of propositional assignments:

   each variable x_i in φ $\longrightarrow$ concept variables $X_i, \bar{X}_i$

The decision problem. NP-hardness (the reduction)

1. Encoding of propositional assignments:

each variable x_i in φ	$\longrightarrow$	concept variables $X_i, \bar{X}_i$	t and f	$\longrightarrow$	concept constants A and B

The decision problem. NP-hardness (the reduction)

1. Encoding of propositional assignments:

each variable x_i in φ $\longrightarrow$ concept variables $X_i, \bar{X}_i$ t and f $\longrightarrow$ concept constants A and B

$$C := \exists r_1.A \sqcap \exists r_1.B \sqcap \ldots \sqcap \exists r_n.A \sqcap \exists r_n.B$$

$$\equiv^?$$

$$D := \exists r_1.X_1 \sqcap \exists r_1.\bar{X}_1 \sqcap \ldots \sqcap \exists r_n.X_n \sqcap \exists r_n.\bar{X}_n$$

The decision problem. NP-hardness (the reduction)

1. Encoding of propositional assignments:

each variable x_i in φ ⟶ concept variables $X_i, \bar{X}_i$

t and f ⟶ concept constants A and B

$$C := \exists r_1.A \sqcap \exists r_1.B \sqcap \ldots \sqcap \exists r_n.A \sqcap \exists r_n.B$$

$$\equiv^?$$

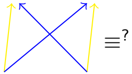
$$D := \exists r_1.X_1 \sqcap \exists r_1.\bar{X}_1 \sqcap \ldots \sqcap \exists r_n.X_n \sqcap \exists r_n.\bar{X}_n$$

The decision problem. NP-hardness (the reduction)

1. Encoding of propositional assignments:

each variable x_i in φ $\longrightarrow$ concept variables $X_i, \bar{X}_i$ t and f $\longrightarrow$ concept constants A and B

$$C := \exists r_1.A \sqcap \exists r_1.B \sqcap \ldots \sqcap \exists r_n.A \sqcap \exists r_n.B$$



$$D := \exists r_1.X_1 \sqcap \exists r_1.\bar{X}_1 \sqcap \ldots \sqcap \exists r_n.X_n \sqcap \exists r_n.\bar{X}_n$$

The decision problem. NP-hardness (the reduction)

1. Encoding of propositional assignments:

each variable x_i in φ $\longrightarrow$ concept variables $X_i, \bar{X}_i$

t and f $\longrightarrow$ concept constants A and B

$$C := \exists r_1.A \sqcap \exists r_1.B \sqcap \ldots \sqcap \exists r_n.A \sqcap \exists r_n.B$$

$$\equiv^?$$

$$D := \exists r_1.X_1 \sqcap \exists r_1.\bar{X}_1 \sqcap \ldots \sqcap \exists r_n.X_n \sqcap \exists r_n.\bar{X}_n$$

$\sigma(C) \equiv \sigma(D)$ **iff** $(\sigma(X_i) = A \wedge \sigma(\bar{X}_i) = B)$ or $(\sigma(X_i) = B \wedge \sigma(\bar{X}_i) = A)$

The decision problem. NP-hardness (the reduction)

1. Encoding of propositional assignments:

each variable x_i in φ $\longrightarrow$ concept variables $X_i, \bar{X}_i$ t and f $\longrightarrow$ concept constants A and B

$$C := \exists r_1.A \sqcap \exists r_1.B \sqcap \ldots \sqcap \exists r_n.A \sqcap \exists r_n.B$$

$$\equiv^?$$

$$D := \exists r_1.X_1 \sqcap \exists r_1.\bar{X}_1 \sqcap \ldots \sqcap \exists r_n.X_n \sqcap \exists r_n.\bar{X}_n$$

$\sigma(C) \equiv \sigma(D)$ **iff** $(\sigma(X_i) = A \wedge \sigma(\bar{X}_i) = B)$ or $(\sigma(X_i) = B \wedge \sigma(\bar{X}_i) = A)$

2. Simulate satisfiability of φ:

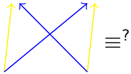
The decision problem. NP-hardness (the reduction)

1. Encoding of propositional assignments:

each variable x_i in φ $\longrightarrow$ concept variables $X_i, \bar{X}_i$

t and f $\longrightarrow$ concept constants A and B

$$C := \exists r_1.A \sqcap \exists r_1.B \sqcap \ldots \sqcap \exists r_n.A \sqcap \exists r_n.B$$



$$D := \exists r_1.X_1 \sqcap \exists r_1.\bar{X}_1 \sqcap \ldots \sqcap \exists r_n.X_n \sqcap \exists r_n.\bar{X}_n$$

$\sigma(C) \equiv \sigma(D)$ **iff** $(\sigma(X_i) = A \wedge \sigma(\bar{X}_i) = B)$ or $(\sigma(X_i) = B \wedge \sigma(\bar{X}_i) = A)$

2. Simulate satisfiability of φ:

each clause $c_j = \ell_{j_1} \vee \ldots \vee \ell_{j_q}$ $\longrightarrow$ concept pattern $P_j := Z_{j_1} \sqcap \ldots \sqcap Z_{j_q} \sqcap B$

The decision problem. NP-hardness (the reduction)

1. Encoding of propositional assignments:

each variable x_i in φ $\longrightarrow$ concept variables $X_i, \bar{X}_i$

t and f $\longrightarrow$ concept constants A and B

$$C := \exists r_1.A \sqcap \exists r_1.B \sqcap \ldots \sqcap \exists r_n.A \sqcap \exists r_n.B$$

$$\equiv^?$$

$$D := \exists r_1.X_1 \sqcap \exists r_1.\bar{X}_1 \sqcap \ldots \sqcap \exists r_n.X_n \sqcap \exists r_n.\bar{X}_n$$

$\sigma(C) \equiv \sigma(D)$ **iff** $(\sigma(X_i) = A \wedge \sigma(\bar{X}_i) = B)$ or $(\sigma(X_i) = B \wedge \sigma(\bar{X}_i) = A)$

2. Simulate satisfiability of φ:

each clause $c_j = \ell_{j_1} \vee \ldots \vee \ell_{j_q}$ $\longrightarrow$ concept pattern $P_j := Z_{j_1} \sqcap \ldots \sqcap Z_{j_q} \sqcap B$

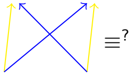
$Z_{j_p} = X_i$, if $\ell_{j_p} = x_i$
$\bar{X}_i$, if $\ell_{j_p} = \neg x_i$

The decision problem. NP-hardness (the reduction)

1. Encoding of propositional assignments:

each variable x_i in φ $\longrightarrow$ concept variables $X_i, \bar{X}_i$ t and f $\longrightarrow$ concept constants A and B

$$C := \exists r_1.A \sqcap \exists r_1.B \sqcap \ldots \sqcap \exists r_n.A \sqcap \exists r_n.B$$



$$D := \exists r_1.X_1 \sqcap \exists r_1.\bar{X}_1 \sqcap \ldots \sqcap \exists r_n.X_n \sqcap \exists r_n.\bar{X}_n$$

$$\sigma(C) \equiv \sigma(D) \textbf{ iff } (\sigma(X_i) = A \wedge \sigma(\bar{X}_i) = B) \text{ or } (\sigma(X_i) = B \wedge \sigma(\bar{X}_i) = A)$$

2. Simulate satisfiability of φ:

each clause $c_j = \ell_{j_1} \vee \ldots \vee \ell_{j_q}$ $\longrightarrow$ concept pattern $P_j := Z_{j_1} \sqcap \ldots \sqcap Z_{j_q} \sqcap B$

$Z_{j_p} = X_i$, if $\ell_{j_p} = x_i$
$\bar{X}_i$, if $\ell_{j_p} = \neg x_i$

match

$M := A \sqcap B$

The decision problem. NP-hardness (the reduction)

1. Encoding of propositional assignments:

each variable x_i in φ $\longrightarrow$ concept variables $X_i, \bar{X}_i$ t and f $\longrightarrow$ concept constants A and B

$$C := \exists r_1.A \sqcap \exists r_1.B \sqcap \ldots \sqcap \exists r_n.A \sqcap \exists r_n.B$$

$$\equiv^?$$

$$D := \exists r_1.X_1 \sqcap \exists r_1.\bar{X}_1 \sqcap \ldots \sqcap \exists r_n.X_n \sqcap \exists r_n.\bar{X}_n$$

$\sigma(C) \equiv \sigma(D)$ **iff** $(\sigma(X_i) = A \wedge \sigma(\bar{X}_i) = B)$ or $(\sigma(X_i) = B \wedge \sigma(\bar{X}_i) = A)$

2. Simulate satisfiability of φ:

each clause $c_j = \ell_{j_1} \vee \ldots \vee \ell_{j_q}$ $\longrightarrow$ concept pattern $P_j := Z_{j_1} \sqcap \ldots \sqcap Z_{j_q} \sqcap B$

$Z_{j_p} = X_i$, if $\ell_{j_p} = x_i$
$\bar{X}_i$, if $\ell_{j_p} = \neg x_i$

match

$$M := A \sqcap B$$

$\sigma(P_j) \equiv \sigma(M)$ **iff** $\sigma(Z_{j_p}) = A$ for at least one $1 \leq p \leq q$.

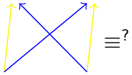
The decision problem. NP-hardness (the reduction)

1. Encoding of propositional assignments:

each variable x_i in φ $\longrightarrow$ concept variables $X_i, \bar{X}_i$

t and f $\longrightarrow$ concept constants A and B

$$C := \exists r_1.A \sqcap \exists r_1.B \sqcap \ldots \sqcap \exists r_n.A \sqcap \exists r_n.B$$



$$D := \exists r_1.X_1 \sqcap \exists r_1.\bar{X}_1 \sqcap \ldots \sqcap \exists r_n.X_n \sqcap \exists r_n.\bar{X}_n$$

$\sigma(C) \equiv \sigma(D)$ **iff** $(\sigma(X_i) = A \wedge \sigma(\bar{X}_i) = B)$ or $(\sigma(X_i) = B \wedge \sigma(\bar{X}_i) = A)$

2. Simulate satisfiability of φ:

each clause $c_j = \ell_{j_1} \vee \ldots \vee \ell_{j_q}$ $\longrightarrow$ concept pattern $P_j := Z_{j_1} \sqcap \ldots \sqcap Z_{j_q} \sqcap B$

$Z_{j_p} = X_i$, if $\ell_{j_p} = x_i$
$\bar{X}_i$, if $\ell_{j_p} = \neg x_i$

match |

$M := A \sqcap B$

B's are important!

$\sigma(P_j) \equiv \sigma(M)$ **iff** $\sigma(Z_{j_p}) = A$ for at least one $1 \leq p \leq q$.

The decision problem. NP-hardness (the reduction)

2. Simulate satisfiability of φ:

$$C_\varphi := \exists s_1.M \sqcap \ldots \sqcap \exists s_m.M$$

$$\equiv^?$$

$$P_\varphi := \exists s_1.P_1 \sqcap \ldots \sqcap \exists s_m.P_m$$

The decision problem. NP-hardness (the reduction)

2. Simulate satisfiability of φ:

$$C_\varphi := \exists s_1.M \sqcap \ldots \sqcap \exists s_m.M$$

$$\equiv^?$$

$$P_\varphi := \exists s_1.P_1 \sqcap \ldots \sqcap \exists s_m.P_m$$

φ is satisfiable **iff** $C \sqcap C_\varphi \equiv^? D \sqcap P_\varphi$ is unifiable

The decision problem. NP-hardness (the reduction)

2. Simulate satisfiability of φ:

$$C_\varphi := \exists s_1.M \sqcap \ldots \sqcap \exists s_m.M$$

$$\equiv^?$$

$$P_\varphi := \exists s_1.P_1 \sqcap \ldots \sqcap \exists s_m.P_m$$

φ is satisfiable **iff** $C \sqcap C_\varphi \equiv^? D \sqcap P_\varphi$ is unifiable

Theorem 6 [BK00]

$\mathcal{EL}$-unification is NP-hard. Even for the special case of matching!

The decision problem. Upper bound (pre-processing)

- Consider unification problems of the form:

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

The decision problem. Upper bound (pre-processing)

- Consider unification problems of the form:

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

- Restrict the attention to flat unification problems:

The decision problem. Upper bound (pre-processing)

- Consider unification problems of the form:
$$\Gamma = \{C_1 \sqsubseteq^? D_1, \ldots, C_n \sqsubseteq^? D_n\}$$
- Restrict the attention to flat unification problems:
  - Atom: $A \in \mathsf{N_C}$ or $\exists r.C$ Flat atom: A, $\exists r.A$ or $\exists r.\top$, where $A \in \mathsf{N_C}$.

The decision problem. Upper bound (pre-processing)

- Consider unification problems of the form:

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

- Restrict the attention to flat unification problems:
  - Atom: $A \in \mathsf{N_C}$ or $\exists r.C$ Flat atom: A, $\exists r.A$ or $\exists r.\top$, where $A \in \mathsf{N_C}$.
  - Flat unification problem: contains only subsumptions of the form:

$$C_1 \sqcap \ldots \sqcap C_m \sqsubseteq^? D, \text{ where } C_1, \ldots, C_m, D \text{ are flat } (m = 0 \Rightarrow \top \sqsubseteq^? D).$$

The decision problem. Upper bound (pre-processing)

- Consider unification problems of the form:

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

- Restrict the attention to flat unification problems:
  - Atom: $A \in \mathsf{N_C}$ or $\exists r.C$ Flat atom: A, $\exists r.A$ or $\exists r.\top$, where $A \in \mathsf{N_C}$.
  - Flat unification problem: contains only subsumptions of the form:

    $C_1 \sqcap \ldots \sqcap C_m \sqsubseteq^? D$, where $C_1, \ldots, C_m, D$ are flat $(m = 0 \Rightarrow \top \sqsubseteq^? D)$.
  - By introducing new variables, every Γ can be transformed into a flat Γ':

The decision problem. Upper bound (pre-processing)

- Consider unification problems of the form:

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

- Restrict the attention to flat unification problems:
  - Atom: $A \in N_C$ or $\exists r.C$ Flat atom: A, $\exists r.A$ or $\exists r.\top$, where $A \in N_C$.
  - Flat unification problem: contains only subsumptions of the form:

    $C_1 \sqcap \ldots \sqcap C_m \sqsubseteq^? D$, where $C_1, \ldots, C_m, D$ are flat $(m = 0 \Rightarrow \top \sqsubseteq^? D)$.
  - By introducing new variables, every Γ can be transformed into a flat Γ':

$A \sqcap \exists r.(B \sqcap \exists s.Y) \sqsubseteq^? X \sqcap \exists s.B$

The decision problem. Upper bound (pre-processing)

- Consider unification problems of the form:
$$\Gamma = \{C_1 \sqsubseteq^? D_1, \dots, C_n \sqsubseteq^? D_n\}$$
- Restrict the attention to flat unification problems:
  - Atom: $A \in \mathsf{N_C}$ or $\exists r.C$ Flat atom: A, $\exists r.A$ or $\exists r.\top$, where $A \in \mathsf{N_C}$.
  - Flat unification problem: contains only subsumptions of the form:
$$C_1 \sqcap \dots \sqcap C_m \sqsubseteq^? D, \text{ where } C_1, \dots, C_m, D \text{ are flat } (m = 0 \Rightarrow \top \sqsubseteq^? D).$$
  - By introducing new variables, every Γ can be transformed into a flat Γ':

$$A \sqcap \exists r.(B \sqcap \exists s.Y) \sqsubseteq^? X \sqcap \exists s.B$$

rule 1 ↓

$$A \sqcap \exists r.X' \sqsubseteq^? X \sqcap \exists s.B$$
$$X' \sqsubseteq^? B \sqcap \exists s.Y$$
$$B \sqcap \exists s.Y \sqsubseteq^? X'$$

The decision problem. Upper bound (pre-processing)

- Consider unification problems of the form:

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

- Restrict the attention to flat unification problems:
  - Atom: $A \in \mathsf{N_C}$ or $\exists r.C$ Flat atom: A, $\exists r.A$ or $\exists r.\top$, where $A \in \mathsf{N_C}$.
  - Flat unification problem: contains only subsumptions of the form:

    $$C_1 \sqcap \ldots \sqcap C_m \sqsubseteq^? D, \text{ where } C_1, \ldots, C_m, D \text{ are flat } (m = 0 \Rightarrow \top \sqsubseteq^? D).$$

  - By introducing new variables, every Γ can be transformed into a flat Γ':

$$A \sqcap \exists r.(B \sqcap \exists s.Y) \sqsubseteq^? X \sqcap \exists s.B$$

rule 1 ↓

$$A \sqcap \exists r.X' \sqsubseteq^? X \sqcap \exists s.B$$

$$X' \sqsubseteq^? B \sqcap \exists s.Y$$

$$B \sqcap \exists s.Y \sqsubseteq^? X'$$

rule 2 →

$$A \sqcap \exists r.X' \sqsubseteq^? X$$

$$A \sqcap \exists r.X' \sqsubseteq^? \exists s.B$$

The decision problem. Upper bound (pre-processing)

- Consider unification problems of the form:
$$\Gamma = \{C_1 \sqsubseteq^? D_1, \ldots, C_n \sqsubseteq^? D_n\}$$
- Restrict the attention to flat unification problems:
  - Atom: $A \in N_C$ or $\exists r.C$ Flat atom: A, $\exists r.A$ or $\exists r.\top$, where $A \in N_C$.
  - Flat unification problem: contains only subsumptions of the form:
  $$C_1 \sqcap \ldots \sqcap C_m \sqsubseteq^? D, \text{ where } C_1, \ldots, C_m, D \text{ are flat } (m = 0 \Rightarrow \top \sqsubseteq^? D).$$
  - By introducing new variables, every Γ can be transformed into a flat Γ':

$$A \sqcap \exists r.(B \sqcap \exists s.Y) \sqsubseteq^? X \sqcap \exists s.B$$

rule 1 ↓

$$A \sqcap \exists r.X' \sqsubseteq^? X \sqcap \exists s.B$$
$$X' \sqsubseteq^? B \sqcap \exists s.Y$$
$$B \sqcap \exists s.Y \sqsubseteq^? X'$$

rule 2:

$A \sqcap \exists r.X' \sqsubseteq^? X \sqcap \exists s.B$ → $A \sqcap \exists r.X' \sqsubseteq^? X$, $A \sqcap \exists r.X' \sqsubseteq^? \exists s.B$

$X' \sqsubseteq^? B \sqcap \exists s.Y$ → $X' \sqsubseteq^? B$, $X' \sqsubseteq^? \exists s.Y$

The decision problem. Upper bound (pre-processing)

- Consider unification problems of the form:

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

- Restrict the attention to flat unification problems:
  - Atom: $A \in N_C$ or $\exists r.C$ Flat atom: A, $\exists r.A$ or $\exists r.\top$, where $A \in N_C$.
  - Flat unification problem: contains only subsumptions of the form:

$$C_1 \sqcap \ldots \sqcap C_m \sqsubseteq^? D, \text{ where } C_1, \ldots, C_m, D \text{ are flat } (m = 0 \Rightarrow \top \sqsubseteq^? D).$$

  - By introducing new variables, every Γ can be transformed into a flat Γ':

$$A \sqcap \exists r.(B \sqcap \exists s.Y) \sqsubseteq^? X \sqcap \exists s.B$$

rule 1 ↓

$$A \sqcap \exists r.X' \sqsubseteq^? X \sqcap \exists s.B \quad \xrightarrow{\text{rule 2}} \quad A \sqcap \exists r.X' \sqsubseteq^? X, \quad A \sqcap \exists r.X' \sqsubseteq^? \exists s.B$$

$$X' \sqsubseteq^? B \sqcap \exists s.Y \quad \longrightarrow \quad X' \sqsubseteq^? B, \quad X' \sqsubseteq^? \exists s.Y$$

$$B \sqcap \exists s.Y \sqsubseteq^? X'$$

  - The transformation is polynomial (time and size of Γ' w.r.t. Γ); and

The decision problem. Upper bound (pre-processing)

- Consider unification problems of the form:
$$\Gamma = \{C_1 \sqsubseteq^? D_1, \ldots, C_n \sqsubseteq^? D_n\}$$
- Restrict the attention to flat unification problems:
  - Atom: $A \in N_C$ or $\exists r.C$ Flat atom: A, $\exists r.A$ or $\exists r.\top$, where $A \in N_C$.
  - Flat unification problem: contains only subsumptions of the form:
$$C_1 \sqcap \ldots \sqcap C_m \sqsubseteq^? D, \text{ where } C_1, \ldots, C_m, D \text{ are flat } (m = 0 \Rightarrow \top \sqsubseteq^? D).$$
  - By introducing new variables, every Γ can be transformed into a flat Γ':

$$A \sqcap \exists r.(B \sqcap \exists s.Y) \sqsubseteq^? X \sqcap \exists s.B$$

rule 1 ↓

$$A \sqcap \exists r.X' \sqsubseteq^? X \sqcap \exists s.B \quad \xrightarrow{\text{rule 2}} \quad A \sqcap \exists r.X' \sqsubseteq^? X, \quad A \sqcap \exists r.X' \sqsubseteq^? \exists s.B$$
$$X' \sqsubseteq^? B \sqcap \exists s.Y \quad \longrightarrow \quad X' \sqsubseteq^? B, \quad X' \sqsubseteq^? \exists s.Y$$
$$B \sqcap \exists s.Y \sqsubseteq^? X'$$

  - The transformation is polynomial (time and size of Γ' w.r.t. Γ); and

Γ has a unifier **iff** Γ' is unifiable.

The decision problem. Upper bound (local unifiers)

Idea of "in NP" upper bound:

Every unifiable flat $\mathcal{EL}$-unification problem has a local unifier.

The decision problem. Upper bound (local unifiers)

Idea of “in NP” upper bound:

Every unifiable flat $\mathcal{EL}$-unification problem has a local unifier.

Local unifiers

The decision problem. Upper bound (local unifiers)

Idea of "in NP" upper bound:

Every unifiable flat $\mathcal{EL}$-unification problem has a local unifier.

Local unifiers

Let Γ be a unification problem, N_v the variables in Γ and At_{nv} its non-variable atoms.

The decision problem. Upper bound (local unifiers)

Idea of "in NP" upper bound:

Every unifiable flat $\mathcal{EL}$-unification problem has a local unifier.

Local unifiers

Let Γ be a unification problem, N_v the variables in Γ and At_{nv} its non-variable atoms.

$$\Gamma := \{Y \sqsubseteq^? \exists r.A, X \sqsubseteq^? \exists r.Y \sqcap \exists s.Y, \exists s.X \sqsubseteq \exists s.B\}$$

The decision problem. Upper bound (local unifiers)

Idea of "in NP" upper bound:

Every unifiable flat $\mathcal{EL}$-unification problem has a local unifier.

Local unifiers

Let Γ be a unification problem, $\mathsf{N_v}$ the variables in Γ and $\mathsf{At_{nv}}$ its non-variable atoms.

$$\Gamma := \{Y \sqsubseteq^? \exists r.A, X \sqsubseteq^? \exists r.Y \sqcap \exists s.Y, \exists s.X \sqsubseteq \exists s.B\}$$

$$\mathsf{N_v} = \{X, Y\} \qquad \mathsf{At_{nv}} = \{A, B, \exists r.A, \exists s.B, \exists r.Y, \exists s.Y, \exists s.X\}$$

The decision problem. Upper bound (local unifiers)

Idea of "in NP" upper bound:

Every unifiable flat $\mathcal{EL}$-unification problem has a local unifier.

Local unifiers

Let Γ be a unification problem, $\mathsf{N_v}$ the variables in Γ and $\mathsf{At_{nv}}$ its non-variable atoms.

$$\Gamma := \{Y \sqsubseteq^? \exists r.A, X \sqsubseteq^? \exists r.Y \sqcap \exists s.Y, \exists s.X \sqsubseteq \exists s.B\}$$

$$\mathsf{N_v} = \{X, Y\} \qquad \mathsf{At_{nv}} = \{A, B, \exists r.A, \exists s.B, \exists r.Y, \exists s.Y, \exists s.X\}$$

- Consider assignments $S : \mathsf{N_v} \to \mathsf{At_{nv}}$

$S(X) := \{B, \exists r.Y, \exists s.Y\}$, $S(Y) := \{\exists r.A\}$ $\quad$ $S'(X) := \{B, \exists r.Y\}$, $S'(Y) := \{A, \exists s.X\}$

The decision problem. Upper bound (local unifiers)

Idea of "in NP" upper bound:

Every unifiable flat $\mathcal{EL}$-unification problem has a local unifier.

Local unifiers

Let Γ be a unification problem, $\mathsf{N_v}$ the variables in Γ and $\mathsf{At_{nv}}$ its non-variable atoms.

$$\Gamma := \{Y \sqsubseteq^? \exists r.A, X \sqsubseteq^? \exists r.Y \sqcap \exists s.Y, \exists s.X \sqsubseteq \exists s.B\}$$

$$\mathsf{N_v} = \{X, Y\} \qquad \mathsf{At_{nv}} = \{A, B, \exists r.A, \exists s.B, \exists r.Y, \exists s.Y, \exists s.X\}$$

- Consider assignments $S : \mathsf{N_v} \to \mathsf{At_{nv}}$

  $S(X) := \{B, \exists r.Y, \exists s.Y\}$, $S(Y) := \{\exists r.A\}$ $\qquad$ $S'(X) := \{B, \exists r.Y\}$, $S'(Y) := \{A, \exists s.X\}$

- Every assignment S induces the following relation $>_S$ on $\mathsf{N_v}$:

$$X >_S Y \text{ iff } Y \text{ occurs in } S(X).$$

The decision problem. Upper bound (local unifiers)

Idea of "in NP" upper bound:

Every unifiable flat $\mathcal{EL}$-unification problem has a local unifier.

Local unifiers

Let Γ be a unification problem, $\mathsf{N_v}$ the variables in Γ and $\mathsf{At_{nv}}$ its non-variable atoms.

$$\Gamma := \{Y \sqsubseteq^? \exists r.A, X \sqsubseteq^? \exists r.Y \sqcap \exists s.Y, \exists s.X \sqsubseteq \exists s.B\}$$

$$\mathsf{N_v} = \{X, Y\} \qquad \mathsf{At_{nv}} = \{A, B, \exists r.A, \exists s.B, \exists r.Y, \exists s.Y, \exists s.X\}$$

- Consider assignments $S : \mathsf{N_v} \to \mathsf{At_{nv}}$

  $S(X) := \{B, \exists r.Y, \exists s.Y\},\ S(Y) := \{\exists r.A\}$ $\quad$ $S'(X) := \{B, \exists r.Y\},\ S'(Y) := \{A, \exists s.X\}$

- Every assignment S induces the following relation $>_S$ on $\mathsf{N_v}$:

  $X >_S Y$ iff Y occurs in $S(X)$.

  $X >_S Y$ $\qquad$ $X >_S Y, Y >_S X$

The decision problem. Upper bound (local unifiers)

The decision problem. Upper bound (local unifiers)

- An assignment S is called acyclic if ${>_S}^+$ is irreflexive

The decision problem. Upper bound (local unifiers)

- An assignment S is called acyclic if ${>_S}^+$ is irreflexive

$${>_S}^+ = \{(X, Y)\} \qquad \qquad \xcancel{{>_S}^+ = \{(X, Y), (Y, X), (X, X), (Y, Y)\}}$$

The decision problem. Upper bound (local unifiers)

- An assignment S is called acyclic if ${>_S}^+$ is irreflexive

$${>_S}^+ = \{(X, Y)\} \qquad \text{~~${>_S}^+ = \{(X,Y),(Y,X),(X,X),(Y,Y)\}$~~}$$

- Any acyclic assignment S induces a substitution σ_S:

The decision problem. Upper bound (local unifiers)

- An assignment S is called acyclic if ${>_S}^+$ is irreflexive

  ${>_S}^+ = \{(X, Y)\}$ ~~${>_S}^+ = \{(X, Y), (Y, X), (X, X), (Y, Y)\}$~~

- Any acyclic assignment S induces a substitution σ_S:
  $S(X) := \{B, \exists r.Y, \exists s.Y\}$, $S(Y) := \{\exists r.A\}$ use ${>_S}^+$

The decision problem. Upper bound (local unifiers)

- An assignment S is called acyclic if ${>_S}^+$ is irreflexive

$${>_S}^+ = \{(X, Y)\} \qquad \sout{{>_S}^+ = \{(X,Y),(Y,X),(X,X),(Y,Y)\}}$$

- Any acyclic assignment S induces a substitution σ_S:
  $S(X) := \{B, \exists r.Y, \exists s.Y\}$, $S(Y) := \{\exists r.A\}$ use ${>_S}^+$

$$Y \text{ is minimal} \longrightarrow \sigma_S(Y) := \bigsqcap_{D \in S(Y)} D = \exists r.A$$

The decision problem. Upper bound (local unifiers)

- An assignment S is called acyclic if ${>_S}^+$ is irreflexive

  ${>_S}^+ = \{(X, Y)\}$ ~~${>_S}^+ = \{(X, Y), (Y, X), (X, X), (Y, Y)\}$~~

- Any acyclic assignment S induces a substitution σ_S:
  $S(X) := \{B, \exists r.Y, \exists s.Y\}$, $S(Y) := \{\exists r.A\}$ use ${>_S}^+$

Y is minimal $\longrightarrow$ $\sigma_S(Y) := \bigsqcap_{D \in S(Y)} D = \exists r.A$

$\sigma_S(Y)$ is defined for all $X {>_S} Y$ $\longrightarrow$ $\sigma_S(X) := \bigsqcap_{D \in S(X)} \sigma_S(D) = B \sqcap \exists r.\exists r.A \sqcap \exists s.\exists r.A$

The decision problem. Upper bound (local unifiers)

- An assignment S is called acyclic if ${>_S}^+$ is irreflexive

$${>_S}^+ = \{(X,Y)\} \qquad \qquad {>_S}^+ = \{(X,Y),(Y,X),(X,X),(Y,Y)\}$$

- Any acyclic assignment S induces a substitution σ_S:

$S(X) := \{B, \exists r.Y, \exists s.Y\}$, $S(Y) := \{\exists r.A\}$ use ${>_S}^+$

Y is minimal $\longrightarrow$ $\sigma_S(Y) := \bigsqcap_{D \in S(Y)} D = \exists r.A$

$\sigma_S(Y)$ is defined for all $X {>_S} Y$ $\longrightarrow$ $\sigma_S(X) := \bigsqcap_{D \in S(X)} \sigma_S(D) = B \sqcap \exists r.\exists r.A \sqcap \exists s.\exists r.A$

- A substitution is local if there exists an acyclic assignment S s.t. $\sigma = \sigma_S$.

The decision problem. Upper bound (local unifiers)

- An assignment S is called acyclic if ${>_S}^+$ is irreflexive

$${>_S}^+ = \{(X, Y)\} \qquad \sout{{>_S}^+ = \{(X,Y),(Y,X),(X,X),(Y,Y)\}}$$

- Any acyclic assignment S induces a substitution σ_S:
  $S(X) := \{B, \exists r.Y, \exists s.Y\}$, $S(Y) := \{\exists r.A\}$ use ${>_S}^+$

$$Y \text{ is minimal} \longrightarrow \sigma_S(Y) := \bigsqcap_{D \in S(Y)} D = \exists r.A$$

$$\begin{array}{c}\sigma_S(Y) \text{ is defined} \\ \text{for all } X >_S Y\end{array} \longrightarrow \sigma_S(X) := \bigsqcap_{D \in S(X)} \sigma_S(D) = B \sqcap \exists r.\exists r.A \sqcap \exists s.\exists r.A$$

- A substitution is local if there exists an acyclic assignment S s.t. $\sigma = \sigma_S$.

σ_S is a local unifier of $\Gamma := \{Y \sqsubseteq^? \exists r.A, X \sqsubseteq^? \exists r.Y \sqcap \exists s.Y, \exists s.X \sqsubseteq \exists s.B\}$

The decision problem. Upper bound (local unifiers)

- An assignment S is called acyclic if ${>_S}^+$ is irreflexive

$${>_S}^+ = \{(X,Y)\} \qquad \xcancel{{>_S}^+ = \{(X,Y),(Y,X),(X,X),(Y,Y)\}}$$

- Any acyclic assignment S induces a substitution σ_S:

$S(X) := \{B, \exists r.Y, \exists s.Y\}$, $S(Y) := \{\exists r.A\}$ use ${>_S}^+$

Y is minimal $\longrightarrow$ $\sigma_S(Y) := \bigsqcap_{D \in S(Y)} D = \exists r.A$

$\sigma_S(Y)$ is defined for all $X >_S Y$ $\longrightarrow$ $\sigma_S(X) := \bigsqcap_{D \in S(X)} \sigma_S(D) = B \sqcap \exists r.\exists r.A \sqcap \exists s.\exists r.A$

- A substitution is local if there exists an acyclic assignment S s.t. $\sigma = \sigma_S$.

σ_S is a local unifier of $\Gamma := \{Y \sqsubseteq^? \exists r.A, X \sqsubseteq^? \exists r.Y \sqcap \exists s.Y, \exists s.X \sqsubseteq \exists s.B\}$

Theorem 7 [BM10b, BBM12b]

Let Γ be a flat unification problem. If Γ has a unifier, then it also has a local unifier.

The decision problem. Upper bound (local unifiers)

Proof of Theorem 7 (Sketch)

The decision problem. Upper bound (local unifiers)

Proof of Theorem 7 (Sketch)

Let Γ be a flat unification problem that has a unifier θ.

The decision problem. Upper bound (local unifiers)

Proof of Theorem 7 (Sketch)

Let Γ be a flat unification problem that has a unifier θ.

- Define the assignment S^θ as:

$$S^\theta(X) := \{D \in \mathsf{At}_{\mathsf{nv}} \mid \theta(X) \sqsubseteq \theta(D)\}, \text{ for all } X \in \mathsf{N_v}.$$

The decision problem. Upper bound (local unifiers)

Proof of Theorem 7 (Sketch)

Let Γ be a flat unification problem that has a unifier θ.

- Define the assignment S^θ as:

$$S^\theta(X) := \{D \in \mathsf{At}_{\mathsf{nv}} \mid \theta(X) \sqsubseteq \theta(D)\}, \text{ for all } X \in \mathsf{N_v}.$$

- S^θ is acyclic. If not, we have:

The decision problem. Upper bound (local unifiers)

Proof of Theorem 7 (Sketch)

Let Γ be a flat unification problem that has a unifier θ.

- Define the assignment S^θ as:

$$S^\theta(X) := \{D \in \mathsf{At}_{\mathsf{nv}} \mid \theta(X) \sqsubseteq \theta(D)\}, \text{ for all } X \in \mathsf{N_v}.$$

- S^θ is acyclic. If not, we have:

$$X_1 >_{S^\theta} X_2 >_{S^\theta} \ldots >_{S^\theta} X_n >_{S^\theta} X_1$$

The decision problem. Upper bound (local unifiers)

Proof of Theorem 7 (Sketch)

Let Γ be a flat unification problem that has a unifier θ.

- Define the assignment S^θ as:

$$S^\theta(X) := \{D \in \mathrm{At_{nv}} \mid \theta(X) \sqsubseteq \theta(D)\}, \text{ for all } X \in \mathrm{N_v}.$$

- S^θ is acyclic. If not, we have:

$$X_1 >_{S^\theta} X_2 >_{S^\theta} \ldots >_{S^\theta} X_n >_{S^\theta} X_1$$

$$\Rightarrow$$

$$\exists r_2.X_2 \in S^\theta(X_1), \exists r_3.X_3 \in S^\theta(X_2), \ldots, \exists r_1.X_1 \in S^\theta(X_n)$$

The decision problem. Upper bound (local unifiers)

Proof of Theorem 7 (Sketch)

Let Γ be a flat unification problem that has a unifier θ.

- Define the assignment S^θ as:

$$S^\theta(X) := \{D \in \mathsf{At}_{\mathsf{nv}} \mid \theta(X) \sqsubseteq \theta(D)\}, \text{ for all } X \in \mathsf{N_v}.$$

- S^θ is acyclic. If not, we have:

$$X_1 >_{S^\theta} X_2 >_{S^\theta} \ldots >_{S^\theta} X_n >_{S^\theta} X_1$$

$$\Rightarrow$$

$$\exists r_2.X_2 \in S^\theta(X_1), \exists r_3.X_3 \in S^\theta(X_2), \ldots, \exists r_1.X_1 \in S^\theta(X_n)$$

$$\Rightarrow$$

$$\theta(X_1) \sqsubseteq \exists r_2.\theta(X_2) \sqsubseteq \ldots \sqsubseteq \exists r_1 \ldots \exists r_n.\theta(X_1)$$

The decision problem. Upper bound (local unifiers)

Proof of Theorem 7 (Sketch)

Let Γ be a flat unification problem that has a unifier θ.

- Define the assignment S^θ as:
$$S^\theta(X) := \{D \in \mathsf{At}_{\mathsf{nv}} \mid \theta(X) \sqsubseteq \theta(D)\}, \text{ for all } X \in \mathsf{N_v}.$$
- S^θ is acyclic. If not, we have:
$$X_1 >_{S^\theta} X_2 >_{S^\theta} \ldots >_{S^\theta} X_n >_{S^\theta} X_1$$
$$\Rightarrow$$
$$\exists r_2.X_2 \in S^\theta(X_1), \exists r_3.X_3 \in S^\theta(X_2), \ldots, \exists r_1.X_1 \in S^\theta(X_n)$$
$$\Rightarrow$$
$$\theta(X_1) \sqsubseteq \exists r_2.\theta(X_2) \sqsubseteq \ldots \sqsubseteq \exists r_1 \ldots \exists r_n.\theta(X_1)$$

  Not true! Hence, σ_{S^θ} is a local substitution.

The decision problem. Upper bound (local unifiers)

Proof of Theorem 7 (Sketch)

Let Γ be a flat unification problem that has a unifier θ.

- Define the assignment S^θ as:

$$S^\theta(X) := \{D \in \mathsf{At}_{\mathsf{nv}} \mid \theta(X) \sqsubseteq \theta(D)\}, \text{ for all } X \in \mathsf{N_v}.$$

- S^θ is acyclic. If not, we have:

$$X_1 >_{S^\theta} X_2 >_{S^\theta} \ldots >_{S^\theta} X_n >_{S^\theta} X_1$$

$$\Rightarrow$$

$$\exists r_2.X_2 \in S^\theta(X_1), \exists r_3.X_3 \in S^\theta(X_2), \ldots, \exists r_1.X_1 \in S^\theta(X_n)$$

$$\Rightarrow$$

$$\theta(X_1) \sqsubseteq \exists r_2.\theta(X_2) \sqsubseteq \ldots \sqsubseteq \exists r_1 \ldots \exists r_n.\theta(X_1)$$

  Not true! Hence, σ_{S^θ} is a local substitution.

- It remains to show that σ_{S^θ} is a solution of Γ.

The decision problem. Upper bound (local unifiers)

σ_{S^θ} is a unifier of Γ

The decision problem. Upper bound (local unifiers)

σ_{S^θ} is a unifier of Γ

We prove a more general claim.

The decision problem. Upper bound (local unifiers)

σ_{S^θ} is a unifier of Γ

We prove a more general claim.

Lemma 8

Let $E_1, \ldots, E_m, D$ be atoms in Γ. Then,

$$\theta(E_1) \sqcap \ldots \sqcap \theta(E_m) \sqsubseteq \theta(D)$$

$$\Rightarrow$$

$$\sigma_{S^\theta}(E_1) \sqcap \ldots \sqcap \sigma_{S^\theta}(E_m) \sqsubseteq \sigma_{S^\theta}(D).$$

The decision problem. Upper bound (local unifiers)

σ_{S^θ} is a unifier of Γ

We prove a more general claim.

Lemma 8

Let $E_1, \ldots, E_m, D$ be atoms in Γ. Then,

$$\theta(E_1) \sqcap \ldots \sqcap \theta(E_m) \sqsubseteq \theta(D)$$

$$\Rightarrow$$

$$\sigma_{S^\theta}(E_1) \sqcap \ldots \sqcap \sigma_{S^\theta}(E_m) \sqsubseteq \sigma_{S^\theta}(D).$$

Proof technique. Use induction on:

$$\max\{\mathrm{rd}(\sigma_{S^\theta}(G)) \mid G \in \{E_1, \ldots, E_m, D\}\},$$

together with the characterization of subsumption.

The decision problem. Upper bound (local unifiers)

σ_{S^θ} is a unifier of Γ

We prove a more general claim.

Lemma 8

Let $E_1, \ldots, E_m, D$ be atoms in Γ. Then,

$$\theta(E_1) \sqcap \ldots \sqcap \theta(E_m) \sqsubseteq \theta(D)$$

$$\Rightarrow$$

$$\sigma_{S^\theta}(E_1) \sqcap \ldots \sqcap \sigma_{S^\theta}(E_m) \sqsubseteq \sigma_{S^\theta}(D).$$

Proof technique. Use induction on:

$$\max\{\mathrm{rd}(\sigma_{S^\theta}(G)) \mid G \in \{E_1, \ldots, E_m, D\}\},$$

together with the characterization of subsumption.

rd = role-depth of a concept description.

The decision problem. Upper bound (NP-algorithm)

Decision procedure

The decision problem. Upper bound (NP-algorithm)

Decision procedure

1. Guess an assignment S.
2. If S is cyclic FAIL.
3. If σ_S solves Γ then SUCCESS.

The decision problem. Upper bound (NP-algorithm)

Decision procedure

1. Guess an assignment S.
2. If S is cyclic FAIL.
3. If σ_S solves Γ then SUCCESS.

Soundness. It is obvious.

The decision problem. Upper bound (NP-algorithm)

Decision procedure

1. Guess an assignment S.
2. If S is cyclic FAIL.
3. If σ_S solves Γ then SUCCESS.

Soundness. It is obvious.

Completeness. If Γ is unifiable, then it has a local unifier θ.

The decision problem. Upper bound (NP-algorithm)

Decision procedure

1. Guess an assignment S.
2. If S is cyclic FAIL.
3. If σ_S solves Γ then SUCCESS.

Soundness. It is obvious.

Completeness. If Γ is unifiable, then it has a local unifier θ.

- S^θ is acyclic.

The decision problem. Upper bound (NP-algorithm)

Decision procedure

1. Guess an assignment S.
2. If S is cyclic FAIL.
3. If σ_S solves Γ then SUCCESS.

Soundness. It is obvious.

Completeness. If Γ is unifiable, then it has a local unifier θ.

- S^θ is acyclic.
- It is easy to show that $\theta(X) \equiv \sigma_{S_\theta}(X)$.

The decision problem. Upper bound (NP-algorithm)

Decision procedure

1. Guess an assignment S.
2. If S is cyclic FAIL.
3. If σ_S solves Γ then SUCCESS.

Soundness. It is obvious.

Completeness. If Γ is unifiable, then it has a local unifier θ.

- S^θ is acyclic.
- It is easy to show that $\theta(X) \equiv \sigma_{S_\theta}(X)$.

The procedure runs in non-deterministic polynomial time.

The decision problem. Upper bound (NP-algorithm)

Decision procedure

1. Guess an assignment S.
2. If S is cyclic FAIL.
3. If σ_S solves Γ then SUCCESS.

Soundness. It is obvious.

Completeness. If Γ is unifiable, then it has a local unifier θ.

- S^θ is acyclic.
- It is easy to show that $\theta(X) \equiv \sigma_{S_\theta}(X)$.

The procedure runs in non-deterministic polynomial time.

- An acyclic assignment S can be guessed in polynomial time.

The decision problem. Upper bound (NP-algorithm)

Decision procedure

1. Guess an assignment S.
2. If S is cyclic FAIL.
3. If σ_S solves Γ then SUCCESS.

Soundness. It is obvious.

Completeness. If Γ is unifiable, then it has a local unifier θ.

- S^θ is acyclic.
- It is easy to show that $\theta(X) \equiv \sigma_{S_\theta}(X)$.

The procedure runs in non-deterministic polynomial time.

- An acyclic assignment S can be guessed in polynomial time.
- How to check that σ_S is a unifier of Γ in polynomial time?

The decision problem. Upper bound (NP-algorithm)

Decision procedure

1. Guess an assignment S.
2. If S is cyclic FAIL.
3. If σ_S solves Γ then SUCCESS.

Soundness. It is obvious.

Completeness. If Γ is unifiable, then it has a local unifier θ.

- S^θ is acyclic.
- It is easy to show that $\theta(X) \equiv \sigma_{S_\theta}(X)$.

The procedure runs in non-deterministic polynomial time.

- An acyclic assignment S can be guessed in polynomial time.
- How to check that σ_S is a unifier of Γ in polynomial time?

σ_S can be of exponential size!

The decision problem. Upper bound (NP-algorithm)

Decision procedure

1. Guess an assignment S.
2. If S is cyclic FAIL.
3. If σ_S solves Γ then SUCCESS.

Soundness. It is obvious.

Completeness. If Γ is unifiable, then it has a local unifier θ.

- S^θ is acyclic.
- It is easy to show that $\theta(X) \equiv \sigma_{S_\theta}(X)$.

The procedure runs in non-deterministic polynomial time.

- An acyclic assignment S can be guessed in polynomial time.
- How to check that σ_S is a unifier of Γ in polynomial time?

σ_S can be of exponential size! $\xrightarrow{\text{solution}}$ acyclic TBox $\mathcal{T}_S$: $X \doteq \bigsqcap_{D \in S(X)} D$

The decision problem. Upper bound (NP-algorithm)

Decision procedure

1. Guess an assignment S.
2. If S is cyclic FAIL.
3. If σ_S solves Γ then SUCCESS.

Soundness. It is obvious.

Completeness. If Γ is unifiable, then it has a local unifier θ.

- S^θ is acyclic.
- It is easy to show that $\theta(X) \equiv \sigma_{S_\theta}(X)$.

The procedure runs in non-deterministic polynomial time.

- An acyclic assignment S can be guessed in polynomial time.
- How to check that σ_S is a unifier of Γ in polynomial time?

σ_S can be of exponential size! $\xrightarrow{\text{solution}}$ acyclic TBox $\mathcal{T}_S$: $X \doteq \bigsqcap_{D \in S(X)} D$ $\longrightarrow$ $\sigma_S(C) \sqsubseteq \sigma_S(D)$ iff $C \sqsubseteq_{\mathcal{T}_S} D$

The decision problem. Upper bound (NP-algorithm)

Decision procedure

1. Guess an assignment S.
2. If S is cyclic FAIL.
3. If σ_S solves Γ then SUCCESS.

Soundness. It is obvious.

Completeness. If Γ is unifiable, then it has a local unifier θ.

- S^θ is acyclic.
- It is easy to show that $\theta(X) \equiv \sigma_{S_\theta}(X)$.

The procedure runs in non-deterministic polynomial time.

- An acyclic assignment S can be guessed in polynomial time.
- How to check that σ_S is a unifier of Γ in polynomial time?

σ_S can be of exponential size! $\xrightarrow{\text{solution}}$ acyclic TBox $\mathcal{T}_S$: $X \doteq \bigsqcap_{D \in S(X)} D$ $\longrightarrow$ $\sigma_S(C) \sqsubseteq \sigma_S(D)$ iff $C \sqsubseteq_{\mathcal{T}_S} D$

Subsumption w.r.t. $\mathcal{T}_S$ can be checked in polynomial time.

The decision problem. More practical algorithms

Previous algorithm is a brutal "guess and then test".

The decision problem. More practical algorithms

Previous algorithm is a brutal "guess and then test".

- A direct implementation is very unlikely to perform well in practice.

The decision problem. More practical algorithms

Previous algorithm is a brutal "guess and then test".

- A direct implementation is very unlikely to perform well in practice.
- More practical algorithms are needed.

The decision problem. More practical algorithms

Previous algorithm is a brutal "guess and then test".

- A direct implementation is very unlikely to perform well in practice.
- More practical algorithms are needed.
  - Goal-oriented algorithm [BM10b].

The decision problem. More practical algorithms

Previous algorithm is a brutal "guess and then test".

- A direct implementation is very unlikely to perform well in practice.
- More practical algorithms are needed.
  - Goal-oriented algorithm [BM10b].
  - SAT Encoding.

More practical algorithms - SAT Encoding [BM10a]

More practical algorithms - SAT Encoding [BM10a]

Given a flat $\mathcal{EL}$-unification problem Γ, construct a propositional formula φ_Γ s.t.:

More practical algorithms - SAT Encoding [BM10a]

Given a flat $\mathcal{EL}$-unification problem Γ, construct a propositional formula φ_Γ s.t.:

- the size of φ_Γ is polynomial in the size of Γ, and

More practical algorithms - SAT Encoding [BM10a]

Given a flat $\mathcal{EL}$-unification problem Γ, construct a propositional formula φ_Γ s.t.:

- the size of φ_Γ is polynomial in the size of Γ, and
- Γ has a unifier **if, and only if**, φ_Γ is satisfiable.

More practical algorithms - SAT Encoding [BM10a]

Given a flat $\mathcal{EL}$-unification problem Γ, construct a propositional formula φ_Γ s.t.:

- the size of φ_Γ is polynomial in the size of Γ, and
- Γ has a unifier **if, and only if**, φ_Γ is satisfiable.

Idea.

$$\overset{\text{unif. problem}}{\Gamma = \{C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? D\}}$$

More practical algorithms - SAT Encoding [BM10a]

Given a flat $\mathcal{EL}$-unification problem Γ, construct a propositional formula φ_Γ s.t.:

- the size of φ_Γ is polynomial in the size of Γ, and
- Γ has a unifier **if, and only if**, φ_Γ is satisfiable.

Idea.

unif. problem

$\Gamma = \{C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? D\}$

prop. variables

$[A \not\sqsubseteq B] \quad \rightarrow \quad \sigma(A) \not\sqsubseteq \sigma(B)$

$[X > Y] \quad \rightarrow \quad$ prevents $\neg[X \not\sqsubseteq \exists r.X]$

More practical algorithms - SAT Encoding [BM10a]

Given a flat $\mathcal{EL}$-unification problem Γ, construct a propositional formula φ_Γ s.t.:

- the size of φ_Γ is polynomial in the size of Γ, and
- Γ has a unifier **if, and only if**, φ_Γ is satisfiable.

Idea.

unif. problem

$\Gamma = \{C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? D\}$

prop. variables

$[A \not\sqsubseteq B] \quad \rightarrow \quad \sigma(A) \not\sqsubseteq \sigma(B)$

$[X > Y] \quad \rightarrow \quad$ prevents $\neg[X \not\sqsubseteq \exists r.X]$

Γ has a solution: $[C_1 \not\sqsubseteq D] \wedge \ldots \wedge [C_n \not\sqsubseteq D] \rightarrow$

More practical algorithms - SAT Encoding [BM10a]

Given a flat $\mathcal{EL}$-unification problem Γ, construct a propositional formula φ_Γ s.t.:

- the size of φ_Γ is polynomial in the size of Γ, and
- Γ has a unifier **if, and only if**, φ_Γ is satisfiable.

Idea.

unif. problem	prop. variables
$\Gamma = \{C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? D\}$	$[A \not\sqsubseteq B] \quad \rightarrow \quad \sigma(A) \not\sqsubseteq \sigma(B)$
	$[X > Y] \quad \rightarrow \quad$ prevents $\neg[X \not\sqsubseteq \exists r.X]$

Γ has a solution: $[C_1 \not\sqsubseteq D] \wedge \ldots \wedge [C_n \not\sqsubseteq D] \rightarrow$

Subsumption properties in $\mathcal{EL}$: $\rightarrow [A \not\sqsubseteq B], \rightarrow [\exists r.A \not\sqsubseteq \exists s.B]$, transitivity, ...

More practical algorithms - SAT Encoding [BM10a]

Given a flat $\mathcal{EL}$-unification problem Γ, construct a propositional formula φ_Γ s.t.:

- the size of φ_Γ is polynomial in the size of Γ, and
- Γ has a unifier **if, and only if**, φ_Γ is satisfiable.

Idea.

unif. problem

$\Gamma = \{C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? D\}$

prop. variables

$[A \not\sqsubseteq B] \quad \rightarrow \quad \sigma(A) \not\sqsubseteq \sigma(B)$

$[X > Y] \quad \rightarrow \quad$ prevents $\neg[X \not\sqsubseteq \exists r.X]$

Γ has a solution: $[C_1 \not\sqsubseteq D] \wedge \ldots \wedge [C_n \not\sqsubseteq D] \rightarrow$

Subsumption properties in $\mathcal{EL}$: $\rightarrow [A \not\sqsubseteq B], \rightarrow [\exists r.A \not\sqsubseteq \exists s.B]$, transitivity, ...

Properties of $>$: $[X > X] \rightarrow, \rightarrow [X > Y] \vee [X \not\sqsubseteq \exists r.Y]$, transitivity

More practical algorithms - SAT Encoding [BM10a]

Given a flat $\mathcal{EL}$-unification problem Γ, construct a propositional formula φ_Γ s.t.:

- the size of φ_Γ is polynomial in the size of Γ, and
- Γ has a unifier **if, and only if**, φ_Γ is satisfiable.

Idea.

unif. problem

$\Gamma = \{C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? D\}$

prop. variables

$[A \not\sqsubseteq B] \quad \rightarrow \quad \sigma(A) \not\sqsubseteq \sigma(B)$

$[X > Y] \quad \rightarrow \quad$ prevents $\neg[X \not\sqsubseteq \exists r.X]$

Γ has a solution: $[C_1 \not\sqsubseteq D] \wedge \ldots \wedge [C_n \not\sqsubseteq D] \rightarrow$

Subsumption properties in $\mathcal{EL}$: $\rightarrow [A \not\sqsubseteq B], \rightarrow [\exists r.A \not\sqsubseteq \exists s.B]$, transitivity, ...

Properties of $>$: $[X > X] \rightarrow, \rightarrow [X > Y] \vee [X \not\sqsubseteq \exists r.Y]$, transitivity

Advantages

More practical algorithms - SAT Encoding [BM10a]

Given a flat $\mathcal{EL}$-unification problem Γ, construct a propositional formula φ_Γ s.t.:

- the size of φ_Γ is polynomial in the size of Γ, and
- Γ has a unifier **if, and only if**, φ_Γ is satisfiable.

Idea.

unif. problem

$\Gamma = \{C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? D\}$

prop. variables

$[A \not\sqsubseteq B] \quad \rightarrow \quad \sigma(A) \not\sqsubseteq \sigma(B)$

$[X > Y] \quad \rightarrow \quad$ prevents $\neg[X \not\sqsubseteq \exists r.X]$

Γ has a solution: $[C_1 \not\sqsubseteq D] \wedge \ldots \wedge [C_n \not\sqsubseteq D] \rightarrow$

Subsumption properties in $\mathcal{EL}$: $\rightarrow [A \not\sqsubseteq B], \rightarrow [\exists r.A \not\sqsubseteq \exists s.B]$, transitivity, ...

Properties of $>$: $[X > X] \rightarrow, \rightarrow [X > Y] \vee [X \not\sqsubseteq \exists r.Y]$, transitivity

Advantages

- a new in NP proof (maybe much simpler).

More practical algorithms - SAT Encoding [BM10a]

Given a flat $\mathcal{EL}$-unification problem Γ, construct a propositional formula φ_Γ s.t.:

- the size of φ_Γ is polynomial in the size of Γ, and
- Γ has a unifier **if, and only if**, φ_Γ is satisfiable.

Idea.

unif. problem

$\Gamma = \{C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? D\}$

prop. variables

$[A \not\sqsubseteq B] \quad \rightarrow \quad \sigma(A) \not\sqsubseteq \sigma(B)$

$[X > Y] \quad \rightarrow \quad$ prevents $\neg[X \not\sqsubseteq \exists r.X]$

Γ has a solution: $[C_1 \not\sqsubseteq D] \wedge \ldots \wedge [C_n \not\sqsubseteq D] \rightarrow$

Subsumption properties in $\mathcal{EL}$: $\rightarrow [A \not\sqsubseteq B], \rightarrow [\exists r.A \not\sqsubseteq \exists s.B]$, transitivity, ...

Properties of $>$: $[X > X] \rightarrow, \rightarrow [X > Y] \vee [X \not\sqsubseteq \exists r.Y]$, transitivity

Advantages

- a new in NP proof (maybe much simpler).
- allows to employ highly optimized SAT solvers to implement an $\mathcal{EL}$-unification algorithm.

Computing minimal $\mathcal{EL}$-unifiers [BBM12a]

Computing minimal $\mathcal{EL}$-unifiers [BBM12a]

All the existent algorithms have in common:

Computing minimal $\mathcal{EL}$-unifiers [BBM12a]

All the existent algorithms have in common:

- they compute all minimal unifiers (this is not $\preceq$):

Computing minimal $\mathcal{EL}$-unifiers [BBM12a]

All the existent algorithms have in common:

- they compute all minimal unifiers (this is not $\preceq$):

Let $\mathcal{X}$ be a set of variables, we define

$$\sigma \geq_{\mathcal{X}} \theta \textbf{ iff } \sigma(X) \sqsubseteq \theta(X) \text{ holds for all } X \in \mathcal{X}.$$

A unifier σ is $\mathcal{X}$-minimal if no other unifier is strictly smaller.

Computing minimal $\mathcal{EL}$-unifiers [BBM12a]

All the existent algorithms have in common:

- they compute all minimal unifiers (this is not $\preceq$):

Let $\mathcal{X}$ be a set of variables, we define

$$\sigma \geq_{\mathcal{X}} \theta \textbf{ iff } \sigma(X) \sqsubseteq \theta(X) \text{ holds for all } X \in \mathcal{X}.$$

A unifier σ is $\mathcal{X}$-minimal if no other unifier is strictly smaller.

- they also compute local unifiers that are not minimal.

Computing minimal $\mathcal{EL}$-unifiers [BBM12a]

All the existent algorithms have in common:

- they compute all minimal unifiers (this is not $\preceq$):

Let $\mathcal{X}$ be a set of variables, we define

$$\sigma \geq_{\mathcal{X}} \theta \textbf{ iff } \sigma(X) \sqsubseteq \theta(X) \text{ holds for all } X \in \mathcal{X}.$$

A unifier σ is $\mathcal{X}$-minimal if no other unifier is strictly smaller.

- they also compute local unifiers that are not minimal.

- very small unification problems can have hundreds of local unifiers.

Computing minimal $\mathcal{EL}$-unifiers [BBM12a]

All the existent algorithms have in common:

- they compute all minimal unifiers (this is not $\preceq$):

Let $\mathcal{X}$ be a set of variables, we define

$$\sigma \geq_{\mathcal{X}} \theta \textbf{ iff } \sigma(X) \sqsubseteq \theta(X) \text{ holds for all } X \in \mathcal{X}.$$

A unifier σ is $\mathcal{X}$-minimal if no other unifier is strictly smaller.

- they also compute local unifiers that are not minimal.

- very small unification problems can have hundreds of local unifiers.

Minimal unifiers represent a significantly smaller (in many cases) subset of the set of local unifiers.

Computing minimal $\mathcal{EL}$-unifiers [BBM12a]

All the existent algorithms have in common:

- they compute all minimal unifiers (this is not $\preceq$):

> Let $\mathcal{X}$ be a set of variables, we define
>
> $$\sigma \geq_{\mathcal{X}} \theta \textbf{ iff } \sigma(X) \sqsubseteq \theta(X) \text{ holds for all } X \in \mathcal{X}.$$
>
> A unifier σ is $\mathcal{X}$-minimal if no other unifier is strictly smaller.

- they also compute local unifiers that are not minimal.

- very small unification problems can have hundreds of local unifiers.

Minimal unifiers represent a significantly smaller (in many cases) subset of the set of local unifiers.

Is it possible to have an NP-algorithm that computes only (and all) $\mathcal{X}$-minimal unifiers?

Computing minimal $\mathcal{EL}$-unifiers [BBM12a]

All the existent algorithms have in common:

- they compute all minimal unifiers (this is not $\preceq$):

Let $\mathcal{X}$ be a set of variables, we define

$$\sigma \geq_{\mathcal{X}} \theta \textbf{ iff } \sigma(X) \sqsubseteq \theta(X) \text{ holds for all } X \in \mathcal{X}.$$

A unifier σ is $\mathcal{X}$-minimal if no other unifier is strictly smaller.

- they also compute local unifiers that are not minimal.
- very small unification problems can have hundreds of local unifiers.

Minimal unifiers represent a significantly smaller (in many cases) subset of the set of local unifiers.

Is it possible to have an NP-algorithm that computes only (and all) $\mathcal{X}$-minimal unifiers?

- Answer: seems unlikely!

Computing minimal $\mathcal{EL}$-unifiers [BBM12a]

The minimal unifier containment problem:

Instance:	A flat $\mathcal{EL}$-unification problem Γ, a set $\mathcal{X} \subseteq \mathsf{N_v}$, a concept constant $A \in \mathsf{N_C}$ and a concept variable $X \in \mathcal{X}$.
Question:	Is there an $\mathcal{X}$-minimal unifier σ of Γ s.t. $\sigma(X) \sqsubseteq A$?

Computing minimal $\mathcal{EL}$-unifiers [BBM12a]

The minimal unifier containment problem:

Instance:	A flat $\mathcal{EL}$-unification problem Γ, a set $\mathcal{X} \subseteq \mathsf{N_v}$, a concept constant $A \in \mathsf{N_C}$ and a concept variable $X \in \mathcal{X}$.
Question:	Is there an $\mathcal{X}$-minimal unifier σ of Γ s.t. $\sigma(X) \sqsubseteq A$?

This problem is Σ_2^p-complete (Th. 4.1 [BBM12a]).

Computing minimal $\mathcal{EL}$-unifiers [BBM12a]

The minimal unifier containment problem:

Instance:	A flat $\mathcal{EL}$-unification problem Γ, a set $\mathcal{X} \subseteq \mathsf{N_v}$, a concept constant $A \in \mathsf{N_C}$ and a concept variable $X \in \mathcal{X}$.
Question:	Is there an $\mathcal{X}$-minimal unifier σ of Γ s.t. $\sigma(X) \sqsubseteq A$?

This problem is Σ_2^p-complete (Th. 4.1 [BBM12a]).

Assume there is an NP-algorithm $\mathcal{A}$ that computes exactly the set of $\mathcal{X}$-minimal unifiers:

Computing minimal $\mathcal{EL}$-unifiers [BBM12a]

The minimal unifier containment problem:

Instance:	A flat $\mathcal{EL}$-unification problem Γ, a set $\mathcal{X} \subseteq N_v$, a concept constant $A \in N_C$ and a concept variable $X \in \mathcal{X}$.
Question:	Is there an $\mathcal{X}$-minimal unifier σ of Γ s.t. $\sigma(X) \sqsubseteq A$?

This problem is Σ_2^p-complete (Th. 4.1 [BBM12a]).

Assume there is an NP-algorithm $\mathcal{A}$ that computes exactly the set of $\mathcal{X}$-minimal unifiers:

- Extend $\mathcal{A}$ with checking, for successful paths, whether $\sigma(X) \sqsubseteq A$.

Computing minimal $\mathcal{EL}$-unifiers [BBM12a]

The minimal unifier containment problem:

Instance:	A flat $\mathcal{EL}$-unification problem Γ, a set $\mathcal{X} \subseteq N_v$, a concept constant $A \in N_C$ and a concept variable $X \in \mathcal{X}$.
Question:	Is there an $\mathcal{X}$-minimal unifier σ of Γ s.t. $\sigma(X) \sqsubseteq A$?

This problem is Σ_2^p-complete (Th. 4.1 [BBM12a]).

Assume there is an NP-algorithm $\mathcal{A}$ that computes exactly the set of $\mathcal{X}$-minimal unifiers:

- Extend $\mathcal{A}$ with checking, for successful paths, whether $\sigma(X) \sqsubseteq A$.
- The minimal unifier containment problem would be decidable in NP.

Computing minimal $\mathcal{EL}$-unifiers [BBM12a]

The minimal unifier containment problem:

Instance:	A flat $\mathcal{EL}$-unification problem Γ, a set $\mathcal{X} \subseteq N_v$, a concept constant $A \in N_C$ and a concept variable $X \in \mathcal{X}$.
Question:	Is there an $\mathcal{X}$-minimal unifier σ of Γ s.t. $\sigma(X) \sqsubseteq A$?

This problem is Σ_2^p-complete (Th. 4.1 [BBM12a]).

Assume there is an NP-algorithm $\mathcal{A}$ that computes exactly the set of $\mathcal{X}$-minimal unifiers:

- Extend $\mathcal{A}$ with checking, for successful paths, whether $\sigma(X) \sqsubseteq A$.
- The minimal unifier containment problem would be decidable in NP.
- NP=Σ_2^p $\Rightarrow$ the polynomial hierarchy collapses.

Unification Modulo $\mathcal{EL}$-TBoxes

Extension to general TBoxes - locality is lost

Extension to general TBoxes - locality is lost

- The presence of a non-empty TBoxes adds new information

Extension to general TBoxes - locality is lost

- The presence of a non-empty TBoxes adds new information

$\Gamma = \{X \sqsubseteq^? \exists r.X\}$ has solutions w.r.t. $\mathcal{T} = \{A \sqsubseteq \exists r.A\}$

Extension to general TBoxes - locality is lost

- The presence of a non-empty TBoxes adds new information

$$\Gamma = \{X \sqsubseteq^? \exists r.X\} \text{ has solutions w.r.t. } \mathcal{T} = \{A \sqsubseteq \exists r.A\}$$

$$\downarrow$$

$$\sigma(X) = A, \sigma(X) = \exists r.A, \ldots$$

Extension to general TBoxes - locality is lost

- The presence of a non-empty TBoxes adds new information

$$\Gamma = \{X \sqsubseteq^? \exists r.X\} \text{ has solutions w.r.t. } \mathcal{T} = \{A \sqsubseteq \exists r.A\}$$

$$\downarrow$$

$$\sigma(X) = A, \sigma(X) = \exists r.A, \ldots$$

- The previous notion of locality does not work.

Extension to general TBoxes - locality is lost

- The presence of a non-empty TBoxes adds new information

$$\Gamma = \{X \sqsubseteq^? \exists r.X\} \text{ has solutions w.r.t. } \mathcal{T} = \{A \sqsubseteq \exists r.A\}$$

$$\downarrow$$

$$\sigma(X) = A, \sigma(X) = \exists r.A, \ldots$$

- The previous notion of locality does not work.

$\mathcal{T} = \{B \sqsubseteq \exists s.D, D \sqsubseteq B\}$

Extension to general TBoxes - locality is lost

- The presence of a non-empty TBoxes adds new information

$$\Gamma = \{X \sqsubseteq^? \exists r.X\} \text{ has solutions w.r.t. } \mathcal{T} = \{A \sqsubseteq \exists r.A\}$$

$$\downarrow$$

$$\sigma(X) = A, \sigma(X) = \exists r.A, \ldots$$

- The previous notion of locality does not work.

$$\mathcal{T} = \{B \sqsubseteq \exists s.D, D \sqsubseteq B\} \qquad \Gamma = \{A_1 \sqcap B \equiv^? Y_1, A_2 \sqcap B \equiv^? Y_2, \exists s.Y_1 \sqsubseteq^? X$$
$$\exists s.Y_2 \sqsubseteq^? X, X \sqsubseteq^? \exists s.X\}$$

Extension to general TBoxes - locality is lost

- The presence of a non-empty TBoxes adds new information

$$\Gamma = \{X \sqsubseteq^? \exists r.X\} \text{ has solutions w.r.t. } \mathcal{T} = \{A \sqsubseteq \exists r.A\}$$

$$\downarrow$$

$$\sigma(X) = A, \sigma(X) = \exists r.A, \ldots$$

- The previous notion of locality does not work.

$$\mathcal{T} = \{B \sqsubseteq \exists s.D, D \sqsubseteq B\} \qquad \Gamma = \{A_1 \sqcap B \equiv^? Y_1, A_2 \sqcap B \equiv^? Y_2, \exists s.Y_1 \sqsubseteq^? X$$
$$\exists s.Y_2 \sqsubseteq^? X, X \sqsubseteq^? \exists s.X\}$$

A unifier $\theta = \{Y_1 \mapsto A_1 \sqcap B, Y_2 \mapsto A_2 \sqcap B, X \mapsto \exists s.B\}$

Extension to general TBoxes - locality is lost

- The presence of a non-empty TBoxes adds new information

$$\Gamma = \{X \sqsubseteq^? \exists r.X\} \text{ has solutions w.r.t. } \mathcal{T} = \{A \sqsubseteq \exists r.A\}$$

$$\downarrow$$

$$\sigma(X) = A, \sigma(X) = \exists r.A, \ldots$$

- The previous notion of locality does not work.

$$\mathcal{T} = \{B \sqsubseteq \exists s.D, D \sqsubseteq B\} \qquad \Gamma = \{A_1 \sqcap B \equiv^? Y_1, A_2 \sqcap B \equiv^? Y_2, \exists s.Y_1 \sqsubseteq^? X$$
$$\exists s.Y_2 \sqsubseteq^? X, X \sqsubseteq^? \exists s.X\}$$

A unifier $\theta = \{Y_1 \mapsto A_1 \sqcap B, Y_2 \mapsto A_2 \sqcap B, X \mapsto \exists s.B\}$

The construction of σ_{S^θ} w.r.t. $\sqsubseteq_{\mathcal{T}}$ yields a cyclic assignment.

Extension to general TBoxes - locality is lost

- The presence of a non-empty TBoxes adds new information

$$\Gamma = \{X \sqsubseteq^? \exists r.X\} \text{ has solutions w.r.t. } \mathcal{T} = \{A \sqsubseteq \exists r.A\}$$

$$\downarrow$$

$$\sigma(X) = A, \sigma(X) = \exists r.A, \ldots$$

- The previous notion of locality does not work.

$$\mathcal{T} = \{B \sqsubseteq \exists s.D, D \sqsubseteq B\} \qquad \Gamma = \{A_1 \sqcap B \equiv^? Y_1, A_2 \sqcap B \equiv^? Y_2, \exists s.Y_1 \sqsubseteq^? X$$
$$\exists s.Y_2 \sqsubseteq^? X, X \sqsubseteq^? \exists s.X\}$$

A unifier $\theta = \{Y_1 \mapsto A_1 \sqcap B, Y_2 \mapsto A_2 \sqcap B, X \mapsto \exists s.B\}$

The construction of σ_{S^θ} w.r.t. $\sqsubseteq_\mathcal{T}$ yields a cyclic assignment.

$$\theta(X) = \exists s.B \sqsubseteq_\mathcal{T} \exists s.\exists s.B = \theta(\exists s.X) \quad \Rightarrow \quad \exists s.X \in S^\theta(X)$$

Extension to general TBoxes - locality is lost

- The presence of a non-empty TBoxes adds new information

$$\Gamma = \{X \sqsubseteq^? \exists r.X\} \text{ has solutions w.r.t. } \mathcal{T} = \{A \sqsubseteq \exists r.A\}$$

$$\downarrow$$

$$\sigma(X) = A, \sigma(X) = \exists r.A, \ldots$$

- The previous notion of locality does not work.

$$\mathcal{T} = \{B \sqsubseteq \exists s.D, D \sqsubseteq B\} \qquad \Gamma = \{A_1 \sqcap B \equiv^? Y_1, A_2 \sqcap B \equiv^? Y_2, \exists s.Y_1 \sqsubseteq^? X, \exists s.Y_2 \sqsubseteq^? X, X \sqsubseteq^? \exists s.X\}$$

A unifier $\theta = \{Y_1 \mapsto A_1 \sqcap B, Y_2 \mapsto A_2 \sqcap B, X \mapsto \exists s.B\}$

The construction of σ_{S^θ} w.r.t. $\sqsubseteq_\mathcal{T}$ yields a cyclic assignment.

$$\theta(X) = \exists s.B \sqsubseteq_\mathcal{T} \exists s.\exists s.B = \theta(\exists s.X) \quad \Rightarrow \quad \exists s.X \in S^\theta(X)$$

One can show that Γ does not have local unifiers!

Extension to general TBoxes - locality is lost

- The presence of a non-empty TBoxes adds new information

$$\Gamma = \{X \sqsubseteq^? \exists r.X\} \text{ has solutions w.r.t. } \mathcal{T} = \{A \sqsubseteq \exists r.A\}$$

$$\downarrow$$

$$\sigma(X) = A, \sigma(X) = \exists r.A, \ldots$$

- The previous notion of locality does not work.

$$\mathcal{T} = \{B \sqsubseteq \exists s.D, D \sqsubseteq B\} \qquad \Gamma = \{A_1 \sqcap B \equiv^? Y_1, A_2 \sqcap B \equiv^? Y_2, \exists s.Y_1 \sqsubseteq^? X$$
$$\exists s.Y_2 \sqsubseteq^? X, X \sqsubseteq^? \exists s.X\}$$

A unifier $\theta = \{Y_1 \mapsto A_1 \sqcap B, Y_2 \mapsto A_2 \sqcap B, X \mapsto \exists s.B\}$

The construction of σ_{S^θ} w.r.t. $\sqsubseteq_\mathcal{T}$ yields a cyclic assignment.

$$\theta(X) = \exists s.B \sqsubseteq_\mathcal{T} \exists s.\exists s.B = \theta(\exists s.X) \quad \Rightarrow \quad \exists s.X \in S^\theta(X)$$

One can show that Γ does not have local unifiers!

- Decidability of $\mathcal{EL}$-unification w.r.t. general TBoxes remains an open problem.

Extension to general TBoxes - locality is lost

- The presence of a non-empty TBoxes adds new information

$$\Gamma = \{X \sqsubseteq^? \exists r.X\} \text{ has solutions w.r.t. } \mathcal{T} = \{A \sqsubseteq \exists r.A\}$$

$$\downarrow$$

$$\sigma(X) = A, \sigma(X) = \exists r.A, \ldots$$

- The previous notion of locality does not work.

$$\mathcal{T} = \{B \sqsubseteq \exists s.D, D \sqsubseteq B\} \qquad \Gamma = \{A_1 \sqcap B \equiv^? Y_1, A_2 \sqcap B \equiv^? Y_2, \exists s.Y_1 \sqsubseteq^? X$$
$$\exists s.Y_2 \sqsubseteq^? X, X \sqsubseteq^? \exists s.X\}$$

A unifier $\theta = \{Y_1 \mapsto A_1 \sqcap B, Y_2 \mapsto A_2 \sqcap B, X \mapsto \exists s.B\}$

The construction of σ_{S^θ} w.r.t. $\sqsubseteq_{\mathcal{T}}$ yields a cyclic assignment.

$$\theta(X) = \exists s.B \sqsubseteq_{\mathcal{T}} \exists s.\exists s.B = \theta(\exists s.X) \quad \Rightarrow \quad \exists s.X \in S^\theta(X)$$

One can show that Γ does not have local unifiers!

- Decidability of $\mathcal{EL}$-unification w.r.t. general TBoxes remains an open problem.
- What can we do?

Extension to general TBoxes - locality is lost

- The presence of a non-empty TBoxes adds new information

$$\Gamma = \{X \sqsubseteq^? \exists r.X\} \text{ has solutions w.r.t. } \mathcal{T} = \{A \sqsubseteq \exists r.A\}$$

$$\downarrow$$

$$\sigma(X) = A, \sigma(X) = \exists r.A, \ldots$$

- The previous notion of locality does not work.

$$\mathcal{T} = \{B \sqsubseteq \exists s.D, D \sqsubseteq B\} \qquad \Gamma = \{A_1 \sqcap B \equiv^? Y_1, A_2 \sqcap B \equiv^? Y_2, \exists s.Y_1 \sqsubseteq^? X$$
$$\exists s.Y_2 \sqsubseteq^? X, X \sqsubseteq^? \exists s.X\}$$

A unifier $\theta = \{Y_1 \mapsto A_1 \sqcap B, Y_2 \mapsto A_2 \sqcap B, X \mapsto \exists s.B\}$

The construction of σ_{S^θ} w.r.t. $\sqsubseteq_\mathcal{T}$ yields a cyclic assignment.

$$\theta(X) = \exists s.B \sqsubseteq_\mathcal{T} \exists s.\exists s.B = \theta(\exists s.X) \quad \Rightarrow \quad \exists s.X \in S^\theta(X)$$

One can show that Γ does not have local unifiers!

- Decidability of $\mathcal{EL}$-unification w.r.t. general TBoxes remains an open problem.
- What can we do? Consider a restricted class of TBoxes.

Restricting the TBox

Cycle-restricted TBoxes

A general TBox $\mathcal{T}$ is called cycle-restricted

iff

there is no word $w \in {\mathsf{N_R}}^+$ and $\mathcal{EL}$ concept C such that $C \sqsubseteq_\mathcal{T} \exists w.C$.

Restricting the TBox

Cycle-restricted TBoxes

A general TBox $\mathcal{T}$ is called cycle-restricted

iff

there is no word $w \in {\mathsf{N_R}}^+$ and $\mathcal{EL}$ concept C such that $C \sqsubseteq_{\mathcal{T}} \exists w.C$.

Some remarks:

Restricting the TBox

Cycle-restricted TBoxes

A general TBox $\mathcal{T}$ is called cycle-restricted

iff

there is no word $w \in {\mathsf{N_R}}^+$ and $\mathcal{EL}$ concept C such that $C \sqsubseteq_{\mathcal{T}} \exists w.C$.

Some remarks:

- cycles $\exists w.C \sqsubseteq_{\mathcal{T}} C$ are allowed.

Restricting the TBox

Cycle-restricted TBoxes

A general TBox $\mathcal{T}$ is called cycle-restricted

iff

there is no word $w \in {\mathsf{N_R}}^+$ and $\mathcal{EL}$ concept C such that $C \sqsubseteq_{\mathcal{T}} \exists w.C$.

Some remarks:

- cycles $\exists w.C \sqsubseteq_{\mathcal{T}} C$ are allowed.
- cycle-restrictedness can be checked in polynomial time.

Restricting the TBox

Cycle-restricted TBoxes

A general TBox $\mathcal{T}$ is called cycle-restricted

iff

there is no word $w \in {\mathsf{N_R}}^+$ and $\mathcal{EL}$ concept C such that $C \sqsubseteq_{\mathcal{T}} \exists w.C$.

Some remarks:

- cycles $\exists w.C \sqsubseteq_{\mathcal{T}} C$ are allowed.
- cycle-restrictedness can be checked in polynomial time.

Locality is regained under cycle-restrictedness

Restricting the TBox

Cycle-restricted TBoxes

A general TBox $\mathcal{T}$ is called cycle-restricted
iff
there is no word $w \in {\mathsf{N_R}}^+$ and $\mathcal{EL}$ concept C such that $C \sqsubseteq_{\mathcal{T}} \exists w.C$.

Some remarks:

- cycles $\exists w.C \sqsubseteq_{\mathcal{T}} C$ are allowed.
- cycle-restrictedness can be checked in polynomial time.

Locality is regained under cycle-restrictedness

- Define the assignment S^{θ} as:

$$S^{\theta}(X) := \{D \in \mathsf{At_{nv}} \mid \theta(X) \sqsubseteq_{\mathcal{T}} \theta(D)\}, \text{ for all } X \in \mathsf{N_v}.$$

Restricting the TBox

Cycle-restricted TBoxes

A general TBox $\mathcal{T}$ is called cycle-restricted
iff
there is no word $w \in \mathsf{N_R}^+$ and $\mathcal{EL}$ concept C such that $C \sqsubseteq_{\mathcal{T}} \exists w.C$.

Some remarks:

- cycles $\exists w.C \sqsubseteq_{\mathcal{T}} C$ are allowed.
- cycle-restrictedness can be checked in polynomial time.

Locality is regained under cycle-restrictedness

- Define the assignment S^θ as:

$$S^\theta(X) := \{D \in \mathsf{At_{nv}} \mid \theta(X) \sqsubseteq_{\mathcal{T}} \theta(D)\}, \text{ for all } X \in \mathsf{N_v}.$$

- S^θ must be acyclic, for otherwise:

$$\theta(X_1) \sqsubseteq_{\mathcal{T}} \exists r_2.\theta(X_2) \sqsubseteq_{\mathcal{T}} \ldots \sqsubseteq_{\mathcal{T}} \exists r_1 \ldots \exists r_n.\theta(X_1)$$

Restricting the TBox

Cycle-restricted TBoxes

A general TBox $\mathcal{T}$ is called cycle-restricted
iff
there is no word $w \in {N_R}^+$ and $\mathcal{EL}$ concept C such that $C \sqsubseteq_\mathcal{T} \exists w.C$.

Some remarks:

- cycles $\exists w.C \sqsubseteq_\mathcal{T} C$ are allowed.
- cycle-restrictedness can be checked in polynomial time.

Locality is regained under cycle-restrictedness

- Define the assignment S^θ as:
$$S^\theta(X) := \{D \in \mathrm{At}_{\mathrm{nv}} \mid \theta(X) \sqsubseteq_\mathcal{T} \theta(D)\}, \text{ for all } X \in \mathrm{N}_\mathrm{v}.$$
- S^θ must be acyclic, for otherwise:
$$\theta(X_1) \sqsubseteq_\mathcal{T} \exists r_2.\theta(X_2) \sqsubseteq_\mathcal{T} \ldots \sqsubseteq_\mathcal{T} \exists r_1 \ldots \exists r_n.\theta(X_1)$$
- σ_{S^θ} is a local substitution.

Restricting the TBox

Theorem 9 [BBM12b]

Let $\mathcal{T}$ be a flat cycle-restricted TBox and Γ a flat unification problem. If Γ has a unifier w.r.t. $\mathcal{T}$, then it also has a local unifier w.r.t. $\mathcal{T}$.

Restricting the TBox

Theorem 9 [BBM12b]

Let $\mathcal{T}$ be a flat cycle-restricted TBox and Γ a flat unification problem. If Γ has a unifier w.r.t. $\mathcal{T}$, then it also has a local unifier w.r.t. $\mathcal{T}$.

The proof is similar as for $\mathcal{T} = \emptyset$, i.e., we proof the corresponding more general claim:

Restricting the TBox

Theorem 9 [BBM12b]

Let $\mathcal{T}$ be a flat cycle-restricted TBox and Γ a flat unification problem. If Γ has a unifier w.r.t. $\mathcal{T}$, then it also has a local unifier w.r.t. $\mathcal{T}$.

The proof is similar as for $\mathcal{T} = \emptyset$, i.e., we proof the corresponding more general claim:

Lemma 10

Let $E_1, \ldots, E_m, D$ be atoms in Γ. Then,

$$\theta(E_1) \sqcap \ldots \sqcap \theta(E_m) \sqsubseteq_{\mathcal{T}} \theta(D)$$

$$\Rightarrow$$

$$\sigma_{S^\theta}(E_1) \sqcap \ldots \sqcap \sigma_{S^\theta}(E_m) \sqsubseteq_{\mathcal{T}} \sigma_{S^\theta}(D).$$

Restricting the TBox

Theorem 9 [BBM12b]

Let $\mathcal{T}$ be a flat cycle-restricted TBox and Γ a flat unification problem. If Γ has a unifier w.r.t. $\mathcal{T}$, then it also has a local unifier w.r.t. $\mathcal{T}$.

The proof is similar as for $\mathcal{T} = \emptyset$, i.e., we proof the corresponding more general claim:

Lemma 10

Let $E_1, \ldots, E_m, D$ be atoms in Γ. Then,

$$\theta(E_1) \sqcap \ldots \sqcap \theta(E_m) \sqsubseteq_{\mathcal{T}} \theta(D)$$

$$\Rightarrow$$

$$\sigma_{S^\theta}(E_1) \sqcap \ldots \sqcap \sigma_{S^\theta}(E_m) \sqsubseteq_{\mathcal{T}} \sigma_{S^\theta}(D).$$

To take into account $\mathcal{T}$:

Restricting the TBox

Theorem 9 [BBM12b]

Let $\mathcal{T}$ be a flat cycle-restricted TBox and Γ a flat unification problem. If Γ has a unifier w.r.t. $\mathcal{T}$, then it also has a local unifier w.r.t. $\mathcal{T}$.

The proof is similar as for $\mathcal{T} = \emptyset$, i.e., we proof the corresponding more general claim:

Lemma 10

Let $E_1, \ldots, E_m, D$ be atoms in Γ. Then,

$$\theta(E_1) \sqcap \ldots \sqcap \theta(E_m) \sqsubseteq_{\mathcal{T}} \theta(D)$$

$$\Rightarrow$$

$$\sigma_{S^\theta}(E_1) \sqcap \ldots \sqcap \sigma_{S^\theta}(E_m) \sqsubseteq_{\mathcal{T}} \sigma_{S^\theta}(D).$$

To take into account $\mathcal{T}$:

- A new characterization of subsumption is developed.

Restricting the TBox

Theorem 9 [BBM12b]

Let $\mathcal{T}$ be a flat cycle-restricted TBox and Γ a flat unification problem. If Γ has a unifier w.r.t. $\mathcal{T}$, then it also has a local unifier w.r.t. $\mathcal{T}$.

The proof is similar as for $\mathcal{T} = \emptyset$, i.e., we proof the corresponding more general claim:

Lemma 10

Let $E_1, \ldots, E_m, D$ be atoms in Γ. Then,

$$\theta(E_1) \sqcap \ldots \sqcap \theta(E_m) \sqsubseteq_{\mathcal{T}} \theta(D)$$

$$\Rightarrow$$

$$\sigma_{S^\theta}(E_1) \sqcap \ldots \sqcap \sigma_{S^\theta}(E_m) \sqsubseteq_{\mathcal{T}} \sigma_{S^\theta}(D).$$

To take into account $\mathcal{T}$:

- A new characterization of subsumption is developed.
- A slightly modified induction hypothesis is used.

Summary - Unification in the DL $\mathcal{EL}$

Summary - Unification in the DL $\mathcal{EL}$

Investigated in detail for the case where $\mathcal{T} = \emptyset$

Summary - Unification in the DL $\mathcal{EL}$

Investigated in detail for the case where $\mathcal{T} = \emptyset$

- The problem is NP-complete. NP-hardness holds even for the special case of matching.

Summary - Unification in the DL $\mathcal{EL}$

Investigated in detail for the case where $\mathcal{T} = \emptyset$

- The problem is NP-complete. NP-hardness holds even for the special case of matching.
- Development of practical algorithms: goal oriented algorithm and SAT encoding.

Summary - Unification in the DL $\mathcal{EL}$

Investigated in detail for the case where $\mathcal{T} = \emptyset$

- The problem is NP-complete. NP-hardness holds even for the special case of matching.
- Development of practical algorithms: goal oriented algorithm and SAT encoding.
- Unification Solver UEL: based on the SAT encoding [BMM12].

Summary - Unification in the DL $\mathcal{EL}$

Investigated in detail for the case where $\mathcal{T} = \emptyset$

- The problem is NP-complete. NP-hardness holds even for the special case of matching.
- Development of practical algorithms: goal oriented algorithm and SAT encoding.
- Unification Solver UEL: based on the SAT encoding [BMM12].
- Computing minimal $\mathcal{EL}$-unifiers is hard.

Summary - Unification in the DL $\mathcal{EL}$

Investigated in detail for the case where $\mathcal{T} = \emptyset$

- The problem is NP-complete. NP-hardness holds even for the special case of matching.
- Development of practical algorithms: goal oriented algorithm and SAT encoding.
- Unification Solver UEL: based on the SAT encoding [BMM12].
- Computing minimal $\mathcal{EL}$-unifiers is hard.

Unification modulo an arbitrary TBox

Summary - Unification in the DL $\mathcal{EL}$

Investigated in detail for the case where $\mathcal{T} = \emptyset$

- The problem is NP-complete. NP-hardness holds even for the special case of matching.
- Development of practical algorithms: goal oriented algorithm and SAT encoding.
- Unification Solver UEL: based on the SAT encoding [BMM12].
- Computing minimal $\mathcal{EL}$-unifiers is hard.

Unification modulo an arbitrary TBox

- Decidability is an open problem.

Summary - Unification in the DL $\mathcal{EL}$

Investigated in detail for the case where $\mathcal{T} = \emptyset$

- The problem is NP-complete. NP-hardness holds even for the special case of matching.
- Development of practical algorithms: goal oriented algorithm and SAT encoding.
- Unification Solver UEL: based on the SAT encoding [BMM12].
- Computing minimal $\mathcal{EL}$-unifiers is hard.

Unification modulo an arbitrary TBox

- Decidability is an open problem.
- Positive results exist only if the TBox or the unification problem are restricted:

Summary - Unification in the DL $\mathcal{EL}$

Investigated in detail for the case where $\mathcal{T} = \emptyset$

- The problem is NP-complete. NP-hardness holds even for the special case of matching.
- Development of practical algorithms: goal oriented algorithm and SAT encoding.
- Unification Solver UEL: based on the SAT encoding [BMM12].
- Computing minimal $\mathcal{EL}$-unifiers is hard.

Unification modulo an arbitrary TBox

- Decidability is an open problem.
- Positive results exist only if the TBox or the unification problem are restricted:
  - Matching w.r.t. a general TBox is NP-complete [BM14].

Summary - Unification in the DL $\mathcal{EL}$

Investigated in detail for the case where $\mathcal{T} = \emptyset$

- The problem is NP-complete. NP-hardness holds even for the special case of matching.
- Development of practical algorithms: goal oriented algorithm and SAT encoding.
- Unification Solver UEL: based on the SAT encoding [BMM12].
- Computing minimal $\mathcal{EL}$-unifiers is hard.

Unification modulo an arbitrary TBox

- Decidability is an open problem.
- Positive results exist only if the TBox or the unification problem are restricted:
  - Matching w.r.t. a general TBox is NP-complete [BM14].
  - Unification w.r.t. cycle-restricted TBoxes is NP-complete.

Summary - Unification in the DL $\mathcal{EL}$

Investigated in detail for the case where $\mathcal{T} = \emptyset$

- The problem is NP-complete. NP-hardness holds even for the special case of matching.
- Development of practical algorithms: goal oriented algorithm and SAT encoding.
- Unification Solver UEL: based on the SAT encoding [BMM12].
- Computing minimal $\mathcal{EL}$-unifiers is hard.

Unification modulo an arbitrary TBox

- Decidability is an open problem.
- Positive results exist only if the TBox or the unification problem are restricted:
    - Matching w.r.t. a general TBox is NP-complete [BM14].
    - Unification w.r.t. cycle-restricted TBoxes is NP-complete.
        - Goal oriented algorithm and SAT encoding.

References I

Franz Baader, Sebastian Brandt, and Carsten Lutz.
Pushing the $\mathcal{EL}$ envelope.
In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, pages 364–369, Edinburgh (UK), 2005. Morgan Kaufmann, Los Altos.

Franz Baader, Stefan Borgwardt, and Barbara Morawska.
Computing minimal el-unifiers is hard.
In *Advances in Modal Logic 9, papers from the ninth conference on "Advances in Modal Logic," held in Copenhagen, Denmark, 22-25 August 2012*, pages 18–35. College Publications, 2012.

Franz Baader, Stefan Borgwardt, and Barbara Morawska.
Extending unification in $\mathcal{EL}$ towards general TBoxes.
In *Proc. of the 13th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2012)*, pages 568–572. AAAI Press/The MIT Press, 2012.

Franz Baader and Ralf Küsters.
Matching in description logics with existential restrictions.
In *Proc. of the 7th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2000)*, pages 261–272, 2000.

References II

Franz Baader and Barbara Morawska.
SAT encoding of unification in *EL*.
In *Logic for Programming, Artificial Intelligence, and Reasoning - 17th International Conference, LPAR-17, Yogyakarta, Indonesia, October 10-15, 2010. Proceedings*, volume 6397 of *Lecture Notes in Computer Science*, pages 97–111. Springer, 2010.

Franz Baader and Barbara Morawska.
Unification in the description logic $\mathcal{EL}$.
Logical Methods in Computer Science, 6(3), 2010.

Franz Baader and Barbara Morawska.
Matching with respect to general concept inclusions in the description logic *EL*.
In *KI 2014: Advances in Artificial Intelligence - 37th Annual German Conference on AI, Stuttgart, Germany, September 22-26, 2014. Proceedings*, volume 8736 of *Lecture Notes in Computer Science*, pages 135–146. Springer, 2014.

References III

Franz Baader, Julian Mendez, and Barbara Morawska.
UEL: unification solver for the description logic $\mathcal{EL}$ - system description.
In *Automated Reasoning - 6th International Joint Conference, IJCAR 2012, Manchester, UK, June 26-29, 2012. Proceedings*, volume 7364 of *Lecture Notes in Computer Science*, pages 45–51. Springer, 2012.

Ralf Küsters.
Non-Standard Inferences in Description Logics, volume 2100 of *Lecture Notes in Computer Science*.
Springer, 2001.