TECHNISCHE UNIVERSITÄT DRESDEN

UNIVERSITÄT LEIPZIG

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

Franz Baader, Oliver Fernández Gil and Barbara Morawska

9th International Symposium, FroCoS

Nancy, September 20th, 2013

Outline

1. The Description Logic $\mathcal{EL}$.
2. Unification in $\mathcal{EL}$.
3. Hybrid Unification in $\mathcal{EL}$.
4. Hybrid Unification is NP-complete.
5. Goal-Oriented Unification Algorithm.
6. Conclusions.

The Description Logic $\mathcal{EL}$. Syntax and Semantics.

$\mathcal{EL}$ concept descriptions

built from finite sets:

$N_C := \{Head\_injury, Severe\}$

$N_R := \{status\}$

using concept constructors:

$\top$, $\sqcap$, $\exists r.C$

$Head\_injury \sqcap \exists status.Severe$

Semantics

An interpretation $\mathcal{I}$ is a pair $(\triangle^{\mathcal{I}}, .^{\mathcal{I}})$ that assigns:

- subsets $C^{\mathcal{I}}$ of $\triangle^{\mathcal{I}}$ to concept names C and,
- binary relations $r^{\mathcal{I}}$ on $\triangle^{\mathcal{I}}$ to role names r.

Semantics of the constructors:

$$\top^{\mathcal{I}} := \triangle^{\mathcal{I}}$$

$$(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$$

$$(\exists r.C)^{\mathcal{I}} := \{x \mid \exists y : (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$$

The Description Logic $\mathcal{EL}$. Syntax and Semantics.

$\mathcal{EL}$ concept descriptions

built from finite sets:
$N_C := \{Head\_injury, Severe\}$
$N_R := \{status\}$

using concept constructors:
$\top$, $\sqcap$, $\exists r.C$
$Head\_injury \sqcap \exists status.Severe$

Semantics

An interpretation $\mathcal{I}$ is a pair $(\triangle^{\mathcal{I}}, .^{\mathcal{I}})$ that assigns:

- subsets $C^{\mathcal{I}}$ of $\triangle^{\mathcal{I}}$ to concept names C and,
- binary relations $r^{\mathcal{I}}$ on $\triangle^{\mathcal{I}}$ to role names r.

Semantics of the constructors:

$$\top^{\mathcal{I}} := \triangle^{\mathcal{I}}$$

$$(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$$

$$(\exists r.C)^{\mathcal{I}} := \{x \mid \exists y : (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$$

The Description Logic $\mathcal{EL}$. Terminological Axioms.

Concept definitions

$A \equiv C$ for $A \in N_C$ and a concept description C.

A TBox $\mathcal{T}$ is a finite set of concept definitions:

$$\left\{ \begin{array}{l} \ldots \\ \mathsf{Severe\_injury} \equiv \mathsf{Injury} \sqcap \exists \mathsf{status}.\mathsf{Severe} \\ \ldots \end{array} \right\}$$

cyclic TBox: $\mathcal{T} = \{A \equiv \ldots \sqcap \exists r.B \sqcap \ldots, B \equiv \ldots \sqcap \exists r.A \sqcap \ldots\}$

General Concept Inclusions (GCIs)

$C \sqsubseteq D$ for concept descriptions C, D.

An ontology $\mathcal{O}$ is a finite set of GCIs:

$$\left\{ \begin{array}{l} \ldots \\ \mathsf{Severe\_injury} \sqsubseteq \mathsf{Injury} \sqcap \exists \mathsf{status}.\mathsf{Severe} \\ \mathsf{Injury} \sqcap \exists \mathsf{status}.\mathsf{Severe} \sqsubseteq \mathsf{Severe\_injury} \\ \ldots \end{array} \right\}$$

The Description Logic $\mathcal{EL}$. Terminological Axioms.

Concept definitions

$A \equiv C$ for $A \in N_C$ and a concept description C.

A TBox $\mathcal{T}$ is a finite set of concept definitions:

$$\left\{ \begin{array}{l} \ldots \\ \mathsf{Severe\_injury} \equiv \mathsf{Injury} \sqcap \exists \mathsf{status}.\mathsf{Severe} \\ \ldots \end{array} \right\}$$

cyclic TBox: $\mathcal{T} = \{A \equiv \ldots \sqcap \exists r.B \sqcap \ldots, B \equiv \ldots \sqcap \exists r.A \sqcap \ldots\}$

General Concept Inclusions (GCIs)

$C \sqsubseteq D$ for concept descriptions C, D.

An ontology $\mathcal{O}$ is a finite set of GCIs:

$$\left\{ \begin{array}{l} \ldots \\ \mathsf{Severe\_injury} \sqsubseteq \mathsf{Injury} \sqcap \exists \mathsf{status}.\mathsf{Severe} \\ \mathsf{Injury} \sqcap \exists \mathsf{status}.\mathsf{Severe} \sqsubseteq \mathsf{Severe\_injury} \\ \ldots \end{array} \right\}$$

The Description Logic $\mathcal{EL}$. Reasoning.

Subsumption problem

$C \sqsubseteq_{\mathcal{O}} D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{O}$.

Equivalence problem

$C \equiv_{\mathcal{O}} D$ iff $C \sqsubseteq_{\mathcal{O}} D$ and $D \sqsubseteq_{\mathcal{O}} C$.

Both problems are polynomial for $\mathcal{EL}$.

Unification in $\mathcal{EL}$. Example.

Unification can be used to detect redundancies in ontologies.

Two concept descriptions that are meant to represent the concept of a *patient with severe head injury*:

$$Patient \sqcap \exists finding.(Head\_injury \sqcap \exists status.Severe)$$

$$Patient \sqcap \exists finding.(Severe\_injury \sqcap \exists location.Head)$$

They are not equivalent, but can be made equivalent w.r.t. the following TBox:

$$\left\{ \begin{array}{l} \mathsf{Severe\_injury} \equiv \mathsf{Injury} \sqcap \exists \mathsf{status.Severe} \\ \mathsf{Head\_injury} \equiv \mathsf{Injury} \sqcap \exists \mathsf{location.Head} \end{array} \right\}$$

Unification in $\mathcal{EL}$. The decision problem.

Unification problem

N_C is partitioned into a set of defined concepts N_{def} and a set of primitive concepts N_{prim}.

Instance:

An ontology $\mathcal{O}$.

A finite set of subsumptions $\Gamma = \{C_1 \sqsubseteq^? D_1, \ldots, C_n \sqsubseteq^? D_n\}$.

Question:

Is there an acyclic TBox $\mathcal{T}$ such that:

$$C_1 \sqsubseteq_{\mathcal{O}\cup\mathcal{T}} D_1, \ldots, C_n \sqsubseteq_{\mathcal{O}\cup\mathcal{T}} D_n$$

Observations

$\mathcal{O}$ contains only concept names from N_{prim}.

The solution TBox $\mathcal{T}$ provides definitions for the concept names in N_{def}.

Unification in $\mathcal{EL}$. The decision problem.

Unification problem

N_C is partitioned into a set of defined concepts N_{def} and a set of primitive concepts N_{prim}.

Instance:

An ontology $\mathcal{O}$.

A finite set of subsumptions $\Gamma = \{C_1 \sqsubseteq^? D_1, \ldots, C_n \sqsubseteq^? D_n\}$.

Question:

Is there an acyclic TBox $\mathcal{T}$ such that:

$$C_1 \sqsubseteq_{\mathcal{O}\cup\mathcal{T}} D_1, \ldots, C_n \sqsubseteq_{\mathcal{O}\cup\mathcal{T}} D_n$$

Observations

$\mathcal{O}$ contains only concept names from N_{prim}.

The solution TBox $\mathcal{T}$ provides definitions for the concept names in N_{def}.

Unification in $\mathcal{EL}$. Previous Results.

[Baader and Morawska 2009]

Unification in $\mathcal{EL}$ without background ontology is NP-complete

An $\mathcal{EL}$-unification problem that is solvable w.r.t. the empty ontology has a solution that is a local acyclic TBox.

Locality

Atoms of Γ (At): $C, \exists r.D$ Non-variable atoms (At_{nv}): $At \setminus N_{def}$

A TBox $\mathcal{T}$ is local if each concept definition in $\mathcal{T}$ is of the following form:

$$X \equiv D_1 \sqcap \ldots \sqcap D_n$$

where $D_i \in At_{nv}$ for all $i, 1 \leq i \leq D_n$.

NP-decision procedure

Guesses a local acyclic TBox and then checks whether it is a unifier.

Unification in $\mathcal{EL}$. Previous Results.

[Baader and Morawska 2009]

Unification in $\mathcal{EL}$ without background ontology is NP-complete

An $\mathcal{EL}$-unification problem that is solvable w.r.t. the empty ontology has a solution that is a local acyclic TBox.

Locality

Atoms of Γ (At): $C, \exists r.D$ Non-variable atoms (At_{nv}): $At \setminus N_{def}$

A TBox $\mathcal{T}$ is local if each concept definition in $\mathcal{T}$ is of the following form:

$$X \equiv D_1 \sqcap \ldots \sqcap D_n$$

where $D_i \in At_{nv}$ for all $i, 1 \leq i \leq D_n$.

NP-decision procedure

Guesses a local acyclic TBox and then checks whether it is a unifier.

Unification in $\mathcal{EL}$. Previous Results.

[Baader and Morawska 2009]

Unification in $\mathcal{EL}$ without background ontology is NP-complete

An $\mathcal{EL}$-unification problem that is solvable w.r.t. the empty ontology has a solution that is a local acyclic TBox.

Locality

Atoms of Γ (At): $C, \exists r.D$ Non-variable atoms (At_{nv}): $At \setminus N_{def}$

A TBox $\mathcal{T}$ is local if each concept definition in $\mathcal{T}$ is of the following form:

$$X \equiv D_1 \sqcap \ldots \sqcap D_n$$

where $D_i \in At_{nv}$ for all $i, 1 \leq i \leq D_n$.

NP-decision procedure

Guesses a local acyclic TBox and then checks whether it is a unifier.

Unification in $\mathcal{EL}$. Previous Results.

Extending unification to non-empty ontologies

- The notion of locality does not work.
- Unification problems with solution, but no local unifiers.
- The algorithm is complete only for cycle restricted ontologies [Baader, Borgwardt, Morawska 2011].

$$\mathcal{O} \not\models C \sqsubseteq \exists w.C$$

A different approach

Extend what is accepted as a solution to the unification problem:

- Allow the TBox $\mathcal{T}$ to be cyclic.
- Use greatest fixpoint semantics to interpret defined concepts in $\mathcal{T}$.

Unification in $\mathcal{EL}$. Previous Results.

Extending unification to non-empty ontologies

- The notion of locality does not work.
- Unification problems with solution, but no local unifiers.
- The algorithm is complete only for cycle restricted ontologies [Baader, Borgwardt, Morawska 2011].

$$\mathcal{O} \not\models C \sqsubseteq \exists w.C$$

A different approach

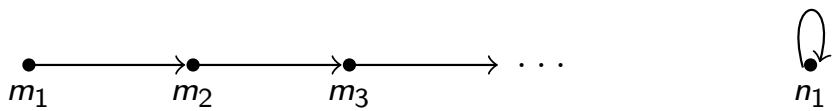
Extend what is accepted as a solution to the unification problem:

- Allow the TBox $\mathcal{T}$ to be cyclic.
- Use greatest fixpoint semantics to interpret defined concepts in $\mathcal{T}$.

Hybrid Unification. Greatest fixpoint semantics [Nebel 1991].

Example

- $N_{prim} := \{Node\}$ and $N_{def} := \{INode\}$
- $\mathcal{T} := \{INode \equiv Node \sqcap \exists edge.INode\}$
- $\mathcal{J}$ is an interpretation of *Node* and *edge*:



- How to extend $\mathcal{J}$ to a model $\mathcal{I}$ of $\mathcal{T}$?

classical models: $\{m_1, m_2, \ldots\} \cup \{n_1\}$, $\{m_1, m_2, \ldots\}$, $\{n_1\}$ or $\emptyset$.

gfp model: $\{m_1, m_2, \ldots\} \cup \{n_1\}$.

Formally,

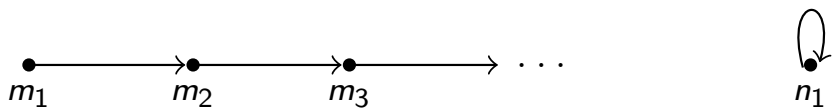
For two extensions $\mathcal{I}_1, \mathcal{I}_2$ of $\mathcal{J}$: $\mathcal{I}_1 \prec_{\mathcal{J}} \mathcal{I}_2$ iff $X^{\mathcal{I}_1} \subseteq X^{\mathcal{I}_2}$ for each $X \in N_{def}$.

The gfp model is the greatest element of $\prec_{\mathcal{J}}$ that models $\mathcal{T}$.

Hybrid Unification. Greatest fixpoint semantics [Nebel 1991].

Example

- $N_{prim} := \{Node\}$ and $N_{def} := \{INode\}$
- $\mathcal{T} := \{INode \equiv Node \sqcap \exists edge.INode\}$
- $\mathcal{J}$ is an interpretation of *Node* and *edge*:



- How to extend $\mathcal{J}$ to a model $\mathcal{I}$ of $\mathcal{T}$?

classical models: $\{m_1, m_2, \ldots\} \cup \{n_1\}$, $\{m_1, m_2, \ldots\}$, $\{n_1\}$ or $\emptyset$.

gfp model: $\{m_1, m_2, \ldots\} \cup \{n_1\}$.

Formally,

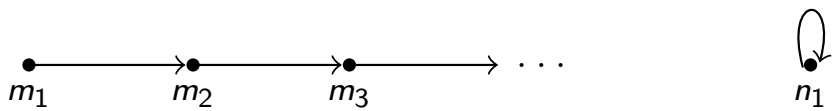
For two extensions $\mathcal{I}_1, \mathcal{I}_2$ of $\mathcal{J}$: $\mathcal{I}_1 \prec_{\mathcal{J}} \mathcal{I}_2$ iff $X^{\mathcal{I}_1} \subseteq X^{\mathcal{I}_2}$ for each $X \in N_{def}$.

The gfp model is the greatest element of $\prec_{\mathcal{J}}$ that models $\mathcal{T}$.

Hybrid Unification. Greatest fixpoint semantics [Nebel 1991].

Example

- $N_{prim} := \{Node\}$ and $N_{def} := \{INode\}$
- $\mathcal{T} := \{INode \equiv Node \sqcap \exists edge.INode\}$
- $\mathcal{J}$ is an interpretation of *Node* and *edge*:



- How to extend $\mathcal{J}$ to a model $\mathcal{I}$ of $\mathcal{T}$?

classical models: $\{m_1, m_2, \ldots\} \cup \{n_1\}$, $\{m_1, m_2, \ldots\}$, $\{n_1\}$ or $\emptyset$.

gfp model: $\{m_1, m_2, \ldots\} \cup \{n_1\}$.

Formally,

For two extensions $\mathcal{I}_1, \mathcal{I}_2$ of $\mathcal{J}$: $\mathcal{I}_1 \prec_{\mathcal{J}} \mathcal{I}_2$ iff $X^{\mathcal{I}_1} \subseteq X^{\mathcal{I}_2}$ for each $X \in N_{def}$.

The gfp model is the greatest element of $\prec_{\mathcal{J}}$ that models $\mathcal{T}$.

Hybrid Unification. Hybrid $\mathcal{EL}$-Ontologies.

Hybrid $\mathcal{EL}$-ontology [Brandt and Model 2005]

A hybrid $\mathcal{EL}$-ontology is a pair $(\mathcal{O}, \mathcal{T})$ where $\mathcal{O}$ is an $\mathcal{EL}$-ontology and $\mathcal{T}$ is a (cyclic) TBox.

An interpretation $\mathcal{I}$ is hybrid model of a hybrid ontology $(\mathcal{O}, \mathcal{T})$ iff:

- $\mathcal{I}$ is an extension of a model $\mathcal{J}$ of $\mathcal{O}$.
- $\mathcal{I}$ is a gfp model of $\mathcal{T}$.

Subsumption w.r.t. hybrid ontologies

C is subsumed by D w.r.t. $(\mathcal{O}, \mathcal{T})$ ($C \sqsubseteq_{gfp, \mathcal{O} \cup \mathcal{T}} D$) iff:

$$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}, \text{ for every hybrid model } \mathcal{I} \text{ of } (\mathcal{O}, \mathcal{T})$$

Subsumption is decidable in polynomial time.[Brandt and Model 2005, Novakovic 2007]

Hybrid Unification. Hybrid $\mathcal{EL}$-Ontologies.

Hybrid $\mathcal{EL}$-ontology [Brandt and Model 2005]

A hybrid $\mathcal{EL}$-ontology is a pair $(\mathcal{O}, \mathcal{T})$ where $\mathcal{O}$ is an $\mathcal{EL}$-ontology and $\mathcal{T}$ is a (cyclic) TBox.

An interpretation $\mathcal{I}$ is hybrid model of a hybrid ontology $(\mathcal{O}, \mathcal{T})$ iff:

- $\mathcal{I}$ is an extension of a model $\mathcal{J}$ of $\mathcal{O}$.
- $\mathcal{I}$ is a gfp model of $\mathcal{T}$.

Subsumption w.r.t. hybrid ontologies

C is subsumed by D w.r.t. $(\mathcal{O}, \mathcal{T})$ ($C \sqsubseteq_{gfp, \mathcal{O} \cup \mathcal{T}} D$) iff:

$$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}, \text{ for every hybrid model } \mathcal{I} \text{ of } (\mathcal{O}, \mathcal{T})$$

Subsumption is decidable in polynomial time.[Brandt and Model 2005, Novakovic 2007]

Hybrid Unification. The decision problem.

Hybrid unification problem

Instance:

An ontology $\mathcal{O}$.

A finite set of subsumptions $\Gamma = \{C_1 \sqsubseteq^? D_1, \ldots, C_n \sqsubseteq^? D_n\}$.

Question:

Is there a (cyclic) TBox $\mathcal{T}$ such that:

$$C_1 \sqsubseteq_{gfp,\mathcal{O}\cup\mathcal{T}} D_1, \ldots, C_n \sqsubseteq_{gfp,\mathcal{O}\cup\mathcal{T}} D_n$$

We call such a TBox $\mathcal{T}$ a hybrid unifier of Γ w.r.t. $\mathcal{O}$.

Remark

Acyclic hybrid unifiers are classical unifiers.

Hybrid Unification. The decision problem.

Hybrid unification problem

Instance:

An ontology $\mathcal{O}$.

A finite set of subsumptions $\Gamma = \{C_1 \sqsubseteq^? D_1, \ldots, C_n \sqsubseteq^? D_n\}$.

Question:

Is there a (cyclic) TBox $\mathcal{T}$ such that:

$$C_1 \sqsubseteq_{gfp,\mathcal{O}\cup\mathcal{T}} D_1, \ldots, C_n \sqsubseteq_{gfp,\mathcal{O}\cup\mathcal{T}} D_n$$

We call such a TBox $\mathcal{T}$ a hybrid unifier of Γ w.r.t. $\mathcal{O}$.

Remark

Acyclic hybrid unifiers are classical unifiers.

Hybrid Unification. Example.

Ontology

$$\left\{ \begin{array}{l} \mathsf{Human} \sqsubseteq \exists \mathsf{parent}.\mathsf{Human} \\ \mathsf{Horse} \sqsubseteq \exists \mathsf{parent}.\mathsf{Horse} \end{array} \right\}$$

Unification problem

$$\Gamma := \{\mathit{Human} \sqsubseteq^? X, \mathit{Horse} \sqsubseteq^? X, X \sqsubseteq^? \exists \mathit{parent}.X\}$$

Γ has a hybrid unifier,

$$\mathcal{T} := \{X \equiv \exists \mathit{parent}.X\}$$

but it does not have a classical unifier.

Hybrid Unification. Example.

Ontology

$$\left\{ \begin{array}{l} \mathsf{Human} \sqsubseteq \exists \mathsf{parent}.\mathsf{Human} \\ \mathsf{Horse} \sqsubseteq \exists \mathsf{parent}.\mathsf{Horse} \end{array} \right\}$$

Unification problem

$$\Gamma := \{\mathit{Human} \sqsubseteq^? X, \mathit{Horse} \sqsubseteq^? X, X \sqsubseteq^? \exists \mathit{parent}.X\}$$

Γ has a hybrid unifier,

$$\mathcal{T} := \{X \equiv \exists \mathit{parent}.X\}$$

but it does not have a classical unifier.

Hybrid Unification is NP-complete. NP-membership.

Proposition

Γ *has a hybrid unifier w.r.t.* $\mathcal{O}$ *iff* Γ *has a local hybrid unifier w.r.t.* $\mathcal{O}$.

Proof.

Assume that $\mathcal{T}$ is a hybrid unifier of Γ w.r.t. $\mathcal{O}$. Define $\mathcal{S}_X$ for each $X \in N_{def}$ as:

$$\mathcal{S}_X = \{D \in At_{nv} \mid X \sqsubseteq_{gfp,\mathcal{O}\cup\mathcal{T}} D\}$$

A local TBox $\mathcal{T}'$ is defined as:

$$\mathcal{T}' = \{X \equiv \prod_{D \in S_X} D \mid X \in N_{def}\}$$

Show that $\mathcal{T}'$ is also a hybrid unifier of Γ w.r.t. $\mathcal{O}$. □

Theorem

Hybrid unification w.r.t. to an arbitrary $\mathcal{EL}$*-ontology is in NP.*

Hybrid Unification is NP-complete. NP-membership.

Proposition

Γ *has a hybrid unifier w.r.t.* $\mathcal{O}$ *iff* Γ *has a local hybrid unifier w.r.t.* $\mathcal{O}$.

Proof.

Assume that $\mathcal{T}$ is a hybrid unifier of Γ w.r.t. $\mathcal{O}$. Define $\mathcal{S}_X$ for each $X \in N_{def}$ as:

$$\mathcal{S}_X = \{D \in At_{nv} \mid X \sqsubseteq_{gfp,\mathcal{O}\cup\mathcal{T}} D\}$$

A local TBox $\mathcal{T}'$ is defined as:

$$\mathcal{T}' = \{X \equiv \bigsqcap_{D\in S_X} D \mid X \in N_{def}\}$$

Show that $\mathcal{T}'$ is also a hybrid unifier of Γ w.r.t. $\mathcal{O}$. □

Theorem

Hybrid unification w.r.t. to an arbitrary $\mathcal{EL}$*-ontology is in NP.*

Hybrid Unification is NP-complete. NP-hardness.

$\mathcal{EL}$-matching problem modulo equivalence

$C \equiv^? D$, where C and D are concept descriptions s.t. C is ground.

Solution

A substitution σ such that $C \equiv \sigma(D)$?

Matching in $\mathcal{EL}$ is NP-Complete. [Küster 2001]

Reduction to the hybrid unification problem

Given $C \equiv^? D$, define $\Gamma = \{C \sqsubseteq^? D, D \sqsubseteq^? C\}$ and $\mathcal{O} = \emptyset$.

Proposition

$C \equiv^? D$ has a solution iff Γ has a hybrid unifier w.r.t. $\mathcal{O} = \emptyset$.

Theorem

Hybrid unification in $\mathcal{EL}$ is NP-hard.

Hybrid Unification is NP-complete. NP-hardness.

$\mathcal{EL}$-matching problem modulo equivalence

$C \equiv^? D$, where C and D are concept descriptions s.t. C is ground.

Solution

A substitution σ such that $C \equiv \sigma(D)$?

Matching in $\mathcal{EL}$ is NP-Complete. [Küster 2001]

Reduction to the hybrid unification problem

Given $C \equiv^? D$, define $\Gamma = \{C \sqsubseteq^? D, D \sqsubseteq^? C\}$ and $\mathcal{O} = \emptyset$.

Proposition

$C \equiv^? D$ has a solution iff Γ has a hybrid unifier w.r.t. $\mathcal{O} = \emptyset$.

Theorem

Hybrid unification in $\mathcal{EL}$ is NP-hard.

Hybrid Unification is NP-complete. NP-hardness.

$\mathcal{EL}$-matching problem modulo equivalence

$C \equiv^? D$, where C and D are concept descriptions s.t. C is ground.

Solution

A substitution σ such that $C \equiv \sigma(D)$?

Matching in $\mathcal{EL}$ is NP-Complete. [Küster 2001]

Reduction to the hybrid unification problem

Given $C \equiv^? D$, define $\Gamma = \{C \sqsubseteq^? D, D \sqsubseteq^? C\}$ and $\mathcal{O} = \emptyset$.

Proposition

$C \equiv^? D$ has a solution iff Γ has a hybrid unifier w.r.t. $\mathcal{O} = \emptyset$.

Theorem

Hybrid unification in $\mathcal{EL}$ is NP-hard.

Proof Calculus **HC** [Novakovic 2007].

A sequent is of the form $C \sqsubseteq_n D$ where C and D are sub-descriptions in $(\mathcal{O}, \mathcal{T})$ and $n \geq 0$.

Rules of **HC**.

$$\frac{}{C \sqsubseteq_n C}\ (\text{Ax}) \qquad \frac{}{C \sqsubseteq_n \top}\ (\text{Top}) \qquad \frac{}{C \sqsubseteq_0 D}\ (\text{Start})$$

$$\frac{C \sqsubseteq_n E}{C \sqcap D \sqsubseteq_n E}\ (\text{AndL1}) \qquad \frac{D \sqsubseteq_n E}{C \sqcap D \sqsubseteq_n E}\ (\text{AndL2}) \qquad \frac{C \sqsubseteq_n D \quad C \sqsubseteq_n E}{C \sqsubseteq_n D \sqcap E}\ (\text{AndR})$$

$$\frac{C \sqsubseteq_n D}{\exists r.C \sqsubseteq_n \exists r.D}\ (\text{Ex})$$

$$\frac{C \sqsubseteq_n D}{X \sqsubseteq_n D}\ (\text{DefL}) \qquad \frac{D \sqsubseteq_n C}{D \sqsubseteq_{n+1} X}\ (\text{DefR}) \qquad \text{for } X \equiv C \in \mathcal{T}$$

$$\frac{C \sqsubseteq_n E \qquad F \sqsubseteq_n D}{C \sqsubseteq_n D}\ (\text{GCI}) \qquad \text{for } E \sqsubseteq F \in \mathcal{O}$$

$C \sqsubseteq_{gfp,\mathcal{O}\cup\mathcal{T}} D$ iff $C \sqsubseteq_n D$ for all $n \geq 0$ ($C \sqsubseteq_\infty D$).

There exists ℓ such that $\sqsubseteq_\ell = \sqsubseteq_{\ell+1} = \ldots$

Proof Calculus **HC** [Novakovic 2007].

A sequent is of the form $C \sqsubseteq_n D$ where C and D are sub-descriptions in $(\mathcal{O}, \mathcal{T})$ and $n \geq 0$.

Rules of **HC**.

$$\frac{}{C \sqsubseteq_n C}\ (\text{Ax}) \qquad \frac{}{C \sqsubseteq_n \top}\ (\text{Top}) \qquad \frac{}{C \sqsubseteq_0 D}\ (\text{Start})$$

$$\frac{C \sqsubseteq_n E}{C \sqcap D \sqsubseteq_n E}\ (\text{AndL1}) \qquad \frac{D \sqsubseteq_n E}{C \sqcap D \sqsubseteq_n E}\ (\text{AndL2}) \qquad \frac{C \sqsubseteq_n D \quad C \sqsubseteq_n E}{C \sqsubseteq_n D \sqcap E}\ (\text{AndR})$$

$$\frac{C \sqsubseteq_n D}{\exists r.C \sqsubseteq_n \exists r.D}\ (\text{Ex})$$

$$\frac{C \sqsubseteq_n D}{X \sqsubseteq_n D}\ (\text{DefL}) \qquad \frac{D \sqsubseteq_n C}{D \sqsubseteq_{n+1} X}\ (\text{DefR}) \qquad \text{for } X \equiv C \in \mathcal{T}$$

$$\frac{C \sqsubseteq_n E \qquad F \sqsubseteq_n D}{C \sqsubseteq_n D}\ (\text{GCI}) \qquad \text{for } E \sqsubseteq F \in \mathcal{O}$$

$C \sqsubseteq_{gfp,\mathcal{O}\cup\mathcal{T}} D$ iff $C \sqsubseteq_n D$ for all $n \geq 0$ ($C \sqsubseteq_\infty D$).

There exists ℓ such that $\sqsubseteq_\ell = \sqsubseteq_{\ell+1} = \ldots$

Proof Calculus **HC** [Novakovic 2007].

A sequent is of the form $C \sqsubseteq_n D$ where C and D are sub-descriptions in $(\mathcal{O}, \mathcal{T})$ and $n \geq 0$.

Rules of **HC**.

$$\frac{}{C \sqsubseteq_n C}\ (\text{Ax}) \qquad \frac{}{C \sqsubseteq_n \top}\ (\text{Top}) \qquad \frac{}{C \sqsubseteq_0 D}\ (\text{Start})$$

$$\frac{C \sqsubseteq_n E}{C \sqcap D \sqsubseteq_n E}\ (\text{AndL1}) \qquad \frac{D \sqsubseteq_n E}{C \sqcap D \sqsubseteq_n E}\ (\text{AndL2}) \qquad \frac{C \sqsubseteq_n D \quad C \sqsubseteq_n E}{C \sqsubseteq_n D \sqcap E}\ (\text{AndR})$$

$$\frac{C \sqsubseteq_n D}{\exists r.C \sqsubseteq_n \exists r.D}\ (\text{Ex})$$

$$\frac{C \sqsubseteq_n D}{X \sqsubseteq_n D}\ (\text{DefL}) \qquad \frac{D \sqsubseteq_n C}{D \sqsubseteq_{n+1} X}\ (\text{DefR}) \qquad \text{for } X \equiv C \in \mathcal{T}$$

$$\frac{C \sqsubseteq_n E \qquad F \sqsubseteq_n D}{C \sqsubseteq_n D}\ (\text{GCI}) \qquad \text{for } E \sqsubseteq F \in \mathcal{O}$$

$C \sqsubseteq_{gfp, \mathcal{O} \cup \mathcal{T}} D$ iff $C \sqsubseteq_n D$ for all $n \geq 0$ ($C \sqsubseteq_\infty D$).

There exists ℓ such that $\sqsubseteq_\ell = \sqsubseteq_{\ell+1} = \ldots$

Proof Calculus **HC** [Novakovic 2007].

A sequent is of the form $C \sqsubseteq_n D$ where C and D are sub-descriptions in $(\mathcal{O}, \mathcal{T})$ and $n \geq 0$.

Rules of **HC**.

$$\frac{}{C \sqsubseteq_n C}\ (\text{Ax}) \qquad \frac{}{C \sqsubseteq_n \top}\ (\text{Top}) \qquad \frac{}{C \sqsubseteq_0 D}\ (\text{Start})$$

$$\frac{C \sqsubseteq_n E}{C \sqcap D \sqsubseteq_n E}\ (\text{AndL1}) \qquad \frac{D \sqsubseteq_n E}{C \sqcap D \sqsubseteq_n E}\ (\text{AndL2}) \qquad \frac{C \sqsubseteq_n D \quad C \sqsubseteq_n E}{C \sqsubseteq_n D \sqcap E}\ (\text{AndR})$$

$$\frac{C \sqsubseteq_n D}{\exists r.C \sqsubseteq_n \exists r.D}\ (\text{Ex})$$

$$\frac{C \sqsubseteq_n D}{X \sqsubseteq_n D}\ (\text{DefL}) \qquad \frac{D \sqsubseteq_n C}{D \sqsubseteq_{n+1} X}\ (\text{DefR}) \qquad \text{for } X \equiv C \in \mathcal{T}$$

$$\frac{C \sqsubseteq_n E \qquad F \sqsubseteq_n D}{C \sqsubseteq_n D}\ (\text{GCI}) \qquad \text{for } E \sqsubseteq F \in \mathcal{O}$$

$C \sqsubseteq_{gfp,\mathcal{O}\cup\mathcal{T}} D$ iff $C \sqsubseteq_n D$ for all $n \geq 0$ ($C \sqsubseteq_\infty D$).

There exists ℓ such that $\sqsubseteq_\ell = \sqsubseteq_{\ell+1} = \ldots$

Goal-Oriented Unification Algorithm.

Main idea:

Try to obtain proof trees for $C_i \sqsubseteq^?_\ell D_i$ while building $\mathcal{T}$.

Applies two types of rules:

- *eager* rules (are always applied first) and *nondeterministic* rules.
- rules represent the possible ways to derive $C_i \sqsubseteq_\ell D_i$.
- A non-failing application of a rule does the following:
  - it solves exactly one unsolved sequent,
  - it may introduce new proof obligations, i.e.: a sequent $C' \sqsubseteq^?_\ell D'$,
  - it may extend the current assignment S.

Estimation of the value for ℓ? we need to show $C_i \sqsubseteq_\infty D_i$.

- locality $\Rightarrow$ maximal size of $\mathcal{T}$ is bounded w.r.t. the size of $\mathcal{O}$ and Γ.
- ℓ is polynomial on the size of $\mathcal{O}$ and Γ.

Run of the algorithm:

- Fails: if a rule application fails or there is an unsolved *sequent* to which no rule applies.
- Succeeds: if there are no unsolved *sequents*.

Goal-Oriented Unification Algorithm.

Main idea:

Try to obtain proof trees for $C_i \sqsubseteq^?_\ell D_i$ while building $\mathcal{T}$.

Applies two types of rules:

- *eager* rules (are always applied first) and *nondeterministic* rules.
- rules represent the possible ways to derive $C_i \sqsubseteq_\ell D_i$.
- A non-failing application of a rule does the following:
  - it solves exactly one unsolved sequent,
  - it may introduce new proof obligations, i.e.: a sequent $C' \sqsubseteq^?_\ell D'$,
  - it may extend the current assignment $\mathcal{S}$.

Estimation of the value for ℓ? we need to show $C_i \sqsubseteq_\infty D_i$.

- locality $\Rightarrow$ maximal size of $\mathcal{T}$ is bounded w.r.t. the size of $\mathcal{O}$ and Γ.
- ℓ is polynomial on the size of $\mathcal{O}$ and Γ.

Run of the algorithm:

- Fails: if a rule application fails or there is an unsolved *sequent* to which no rule applies.
- Succeeds: if there are no unsolved *sequents*.

Goal-Oriented Unification Algorithm.

Main idea:

Try to obtain proof trees for $C_i \sqsubseteq^?_\ell D_i$ while building $\mathcal{T}$.

Applies two types of rules:

- *eager* rules (are always applied first) and *nondeterministic* rules.
- rules represent the possible ways to derive $C_i \sqsubseteq_\ell D_i$.
- A non-failing application of a rule does the following:
  - it solves exactly one unsolved sequent,
  - it may introduce new proof obligations, i.e.: a sequent $C' \sqsubseteq^?_\ell D'$,
  - it may extend the current assignment $\mathcal{S}$.

Estimation of the value for ℓ? we need to show $C_i \sqsubseteq_\infty D_i$.

- locality $\Rightarrow$ maximal size of $\mathcal{T}$ is bounded w.r.t. the size of $\mathcal{O}$ and Γ.
- ℓ is polynomial on the size of $\mathcal{O}$ and Γ.

Run of the algorithm:

- Fails: if a rule application fails or there is an unsolved *sequent* to which no rule applies.
- Succeeds: if there are no unsolved *sequents*.

Goal-Oriented Unification Algorithm.

Main idea:

Try to obtain proof trees for $C_i \sqsubseteq^?_\ell D_i$ while building $\mathcal{T}$.

Applies two types of rules:

- *eager* rules (are always applied first) and *nondeterministic* rules.
- rules represent the possible ways to derive $C_i \sqsubseteq_\ell D_i$.
- A non-failing application of a rule does the following:
  - it solves exactly one unsolved sequent,
  - it may introduce new proof obligations, i.e.: a sequent $C' \sqsubseteq^?_\ell D'$,
  - it may extend the current assignment $\mathcal{S}$.

Estimation of the value for ℓ? we need to show $C_i \sqsubseteq_\infty D_i$.

- locality $\Rightarrow$ maximal size of $\mathcal{T}$ is bounded w.r.t. the size of $\mathcal{O}$ and Γ.
- ℓ is polynomial on the size of $\mathcal{O}$ and Γ.

Run of the algorithm:

- Fails: if a rule application fails or there is an unsolved *sequent* to which no rule applies.
- Succeeds: if there are no unsolved *sequents*.

Goal-Oriented Unification Algorithm.

Main idea:

Try to obtain proof trees for $C_i \sqsubseteq^?_\ell D_i$ while building $\mathcal{T}$.

Applies two types of rules:

- *eager* rules (are always applied first) and *nondeterministic* rules.
- rules represent the possible ways to derive $C_i \sqsubseteq_\ell D_i$.
- A non-failing application of a rule does the following:
  - it solves exactly one unsolved sequent,
  - it may introduce new proof obligations, i.e.: a sequent $C' \sqsubseteq^?_\ell D'$,
  - it may extend the current assignment $\mathcal{S}$.

Estimation of the value for ℓ? we need to show $C_i \sqsubseteq_\infty D_i$.

- locality $\Rightarrow$ maximal size of $\mathcal{T}$ is bounded w.r.t. the size of $\mathcal{O}$ and Γ.
- ℓ is polynomial on the size of $\mathcal{O}$ and Γ.

Run of the algorithm:

- Fails: if a rule application fails or there is an unsolved *sequent* to which no rule applies.
- Succeeds: if there are no unsolved *sequents*.

Goal-Oriented Unification Algorithm.

Main idea:

Try to obtain proof trees for $C_i \sqsubseteq^?_\ell D_i$ while building $\mathcal{T}$.

Applies two types of rules:

- *eager* rules (are always applied first) and *nondeterministic* rules.
- rules represent the possible ways to derive $C_i \sqsubseteq_\ell D_i$.
- A non-failing application of a rule does the following:
  - it solves exactly one unsolved sequent,
  - it may introduce new proof obligations, i.e.: a sequent $C' \sqsubseteq^?_\ell D'$,
  - it may extend the current assignment $\mathcal{S}$.

Estimation of the value for ℓ? we need to show $C_i \sqsubseteq_\infty D_i$.

- locality $\Rightarrow$ maximal size of $\mathcal{T}$ is bounded w.r.t. the size of $\mathcal{O}$ and Γ.
- ℓ is polynomial on the size of $\mathcal{O}$ and Γ.

Run of the algorithm:

- Fails: if a rule application fails or there is an unsolved *sequent* to which no rule applies.
- Succeeds: if there are no unsolved *sequents*.

Goal-Oriented Unification Algorithm.

Main idea:

Try to obtain proof trees for $C_i \sqsubseteq^?_\ell D_i$ while building $\mathcal{T}$.

Applies two types of rules:

- *eager* rules (are always applied first) and *nondeterministic* rules.
- rules represent the possible ways to derive $C_i \sqsubseteq_\ell D_i$.
- A non-failing application of a rule does the following:
  - it solves exactly one unsolved sequent,
  - it may introduce new proof obligations, i.e.: a sequent $C' \sqsubseteq^?_\ell D'$,
  - it may extend the current assignment $\mathcal{S}$.

Estimation of the value for ℓ? we need to show $C_i \sqsubseteq_\infty D_i$.

- locality $\Rightarrow$ maximal size of $\mathcal{T}$ is bounded w.r.t. the size of $\mathcal{O}$ and Γ.
- ℓ is polynomial on the size of $\mathcal{O}$ and Γ.

Run of the algorithm:

- Fails: if a rule application fails or there is an unsolved *sequent* to which no rule applies.
- Succeeds: if there are no unsolved *sequents*.

Goal-Oriented Unification Algorithm.

Main idea:

Try to obtain proof trees for $C_i \sqsubseteq^?_\ell D_i$ while building $\mathcal{T}$.

Applies two types of rules:

- *eager* rules (are always applied first) and *nondeterministic* rules.
- rules represent the possible ways to derive $C_i \sqsubseteq_\ell D_i$.
- A non-failing application of a rule does the following:
  - it solves exactly one unsolved sequent,
  - it may introduce new proof obligations, i.e.: a sequent $C' \sqsubseteq^?_\ell D'$,
  - it may extend the current assignment $\mathcal{S}$.

Estimation of the value for ℓ? we need to show $C_i \sqsubseteq_\infty D_i$.

- locality $\Rightarrow$ maximal size of $\mathcal{T}$ is bounded w.r.t. the size of $\mathcal{O}$ and Γ.
- ℓ is polynomial on the size of $\mathcal{O}$ and Γ.

Run of the algorithm:

- Fails: if a rule application fails or there is an unsolved *sequent* to which no rule applies.
- Succeeds: if there are no unsolved *sequents*.

Goal-Oriented Unification Algorithm. Rules.

Eager Ground Solving:

Applies to ground sequents: $A \sqsubseteq_n^? B \longrightarrow$ fails if $A \not\sqsubseteq_{\mathcal{O}} B$

Decomposition and *Extension*:

$$\exists r.X_1 \sqcap \exists r.X_2 \sqsubseteq_n^? \exists r.C \quad \begin{cases} \nearrow & X_1 \sqsubseteq_n^? C \longrightarrow \mathcal{S}_{X_1} = \{C\} \\ \searrow & X_2 \sqsubseteq_n^? C \longrightarrow \mathcal{S}_{X_2} = \{C\} \end{cases}$$

Mutation (using GCIs):

$$\exists r.X \sqsubseteq_n^? B \quad \mathcal{O} = \{\exists r.A \sqsubseteq B\} \quad \begin{cases} \swarrow & \exists r.X \sqsubseteq_n^? \exists r.A \\ \searrow & B \sqsubseteq_n^? B \end{cases}$$

Goal-Oriented Unification Algorithm. Rules.

Eager Ground Solving:

Applies to ground sequents: $A \sqsubseteq_n^? B \longrightarrow$ fails if $A \not\sqsubseteq_{\mathcal{O}} B$

Decomposition and *Extension*:

$$\exists r.X_1 \sqcap \exists r.X_2 \sqsubseteq_n^? \exists r.C \quad \begin{cases} \nearrow & X_1 \sqsubseteq_n^? C \longrightarrow \mathcal{S}_{X_1} = \{C\} \\ \searrow & X_2 \sqsubseteq_n^? C \longrightarrow \mathcal{S}_{X_2} = \{C\} \end{cases}$$

Mutation (using GCIs):

$$\exists r.X \sqsubseteq_n^? B \quad \mathcal{O} = \{\exists r.A \sqsubseteq B\} \quad \begin{cases} \swarrow & \exists r.X \sqsubseteq_n^? \exists r.A \\ \searrow & B \sqsubseteq_n^? B \end{cases}$$

Goal-Oriented Unification Algorithm. Rules.

Eager Ground Solving:

Applies to ground sequents: $A \sqsubseteq_n^? B \longrightarrow$ fails if $A \not\sqsubseteq_{\mathcal{O}} B$

Decomposition and *Extension*:

$$\exists r.X_1 \sqcap \exists r.X_2 \sqsubseteq_n^? \exists r.C \quad \begin{cases} \nearrow & X_1 \sqsubseteq_n^? C \longrightarrow \mathcal{S}_{X_1} = \{C\} \\ \searrow & X_2 \sqsubseteq_n^? C \longrightarrow \mathcal{S}_{X_2} = \{C\} \end{cases}$$

Mutation (using GCIs):

$$\exists r.X \sqsubseteq_n^? B \quad \begin{cases} \swarrow & \\ & \mathcal{O} = \{\exists r.A \sqsubseteq B\} \\ \searrow & \end{cases}$$

$$\exists r.X \sqsubseteq_n^? \exists r.A \qquad B \sqsubseteq_n^? B$$

Goal-Oriented Unification Algorithm. Rules.

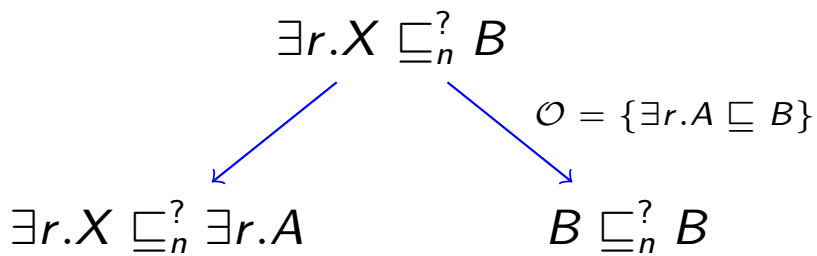
Eager Ground Solving:

Applies to ground sequents: $A \sqsubseteq_n^? B \longrightarrow$ fails if $A \not\sqsubseteq_{\mathcal{O}} B$

Decomposition and *Extension*:

$$\exists r.X_1 \sqcap \exists r.X_2 \sqsubseteq_n^? \exists r.C \quad \begin{cases} \nearrow & X_1 \sqsubseteq_n^? C \longrightarrow \mathcal{S}_{X_1} = \{C\} \\ \searrow & X_2 \sqsubseteq_n^? C \longrightarrow \mathcal{S}_{X_2} = \{C\} \end{cases}$$

Mutation (using GCIs):



Goal-Oriented Unification Algorithm. Example 1.

$$\mathcal{O} = \{\mathit{Horse} \sqsubseteq \exists \mathit{parent}.\mathit{Horse}, \mathit{Human} \sqsubseteq \exists \mathit{parent}.\mathit{Human}\}$$

$$\Gamma = \{\mathit{Human} \sqsubseteq_\ell X, \mathit{Horse} \sqsubseteq_\ell X, X \sqsubseteq_\ell \exists \mathit{parent}.X\}$$

$X \sqsubseteq_\ell \exists \mathit{parent}.X$ $\quad$ $S_X = \emptyset$ $\quad$ Mutation ($\mathit{Human} \sqsubseteq \exists \mathit{par} \ldots$)

Goal-Oriented Unification Algorithm. Example 1.

$$\mathcal{O} = \{Horse \sqsubseteq \exists parent.Horse, Human \sqsubseteq \exists parent.Human\}$$

$$\Gamma = \{Human \sqsubseteq_\ell X, Horse \sqsubseteq_\ell X, X \sqsubseteq_\ell \exists parent.X\}$$

$X \sqsubseteq_\ell \exists parent.X$ $\quad S_X = \emptyset$ $\quad$ Mutation ($Human \sqsubseteq \exists par \ldots$)

$X \sqsubseteq_\ell Human$

$\exists parent.Human \sqsubseteq_\ell \exists parent.X$

Goal-Oriented Unification Algorithm. Example 1.

$$\mathcal{O} = \{Horse \sqsubseteq \exists parent.Horse, Human \sqsubseteq \exists parent.Human\}$$

$$\Gamma = \{Human \sqsubseteq_\ell X, Horse \sqsubseteq_\ell X, X \sqsubseteq_\ell \exists parent.X\}$$

$X \sqsubseteq_\ell \exists parent.X$	$S_X = \emptyset$	Mutation ($Human \sqsubseteq \exists par \ldots$)
$X \sqsubseteq_\ell Human$		
$\exists parent.Human \sqsubseteq_\ell \exists parent.X$		
$X \sqsubseteq_\ell Human$	$S_X = \{Human\}$	Extension

Goal-Oriented Unification Algorithm. Example 1.

$$\mathcal{O} = \{Horse \sqsubseteq \exists parent.Horse, Human \sqsubseteq \exists parent.Human\}$$

$$\Gamma = \{Human \sqsubseteq_\ell X, Horse \sqsubseteq_\ell X, X \sqsubseteq_\ell \exists parent.X\}$$

$X \sqsubseteq_\ell \exists parent.X$	$S_X = \emptyset$	Mutation ($Human \sqsubseteq \exists par \ldots$)
$X \sqsubseteq_\ell Human$		
$\exists parent.Human \sqsubseteq_\ell \exists parent.X$		
$X \sqsubseteq_\ell Human$	$S_X = \{Human\}$	Extension
$Horse \sqsubseteq_{\ell-1} Human$		Expansion

Goal-Oriented Unification Algorithm. Example 1.

$$\mathcal{O} = \{Horse \sqsubseteq \exists parent.Horse, Human \sqsubseteq \exists parent.Human\}$$

$$\Gamma = \{Human \sqsubseteq_\ell X, Horse \sqsubseteq_\ell X, X \sqsubseteq_\ell \exists parent.X\}$$

$X \sqsubseteq_\ell \exists parent.X$	$S_X = \emptyset$	Mutation ($Human \sqsubseteq \exists par \ldots$)
$X \sqsubseteq_\ell Human$		
$\exists parent.Human \sqsubseteq_\ell \exists parent.X$		
$X \sqsubseteq_\ell Human$	$S_X = \{Human\}$	Extension
$Horse \sqsubseteq_{\ell-1} Human$		Expansion
$Horse \sqsubseteq_{\ell-1} Human$		Eager Ground Solving

A failing run, $Horse \not\sqsubseteq_{\mathcal{O}} Human$.

Goal-Oriented Unification Algorithm. Example 2.

$$\mathcal{O} = \{Horse \sqsubseteq \exists parent.Horse, Human \sqsubseteq \exists parent.Human\}$$

$$\Gamma = \{Human \sqsubseteq_\ell X, Horse \sqsubseteq_\ell X, X \sqsubseteq_\ell \exists parent.X\}$$

$X \sqsubseteq_\ell \exists parent.X$ $\qquad S_X = \{\exists parent.X\}$ $\qquad$ Extension

Goal-Oriented Unification Algorithm. Example 2.

$$\mathcal{O} = \{Horse \sqsubseteq \exists parent.Horse, Human \sqsubseteq \exists parent.Human\}$$

$$\Gamma = \{Human \sqsubseteq_\ell X, Horse \sqsubseteq_\ell X, X \sqsubseteq_\ell \exists parent.X\}$$

$X \sqsubseteq_\ell \exists parent.X$	$S_X = \{\exists parent.X\}$	Extension
$Human \sqsubseteq_{\ell-1} \exists parent.X$		Expansion
$Horse \sqsubseteq_{\ell-1} \exists parent.X$		

Goal-Oriented Unification Algorithm. Example 2.

$$\mathcal{O} = \{Horse \sqsubseteq \exists parent.Horse, Human \sqsubseteq \exists parent.Human\}$$

$$\Gamma = \{Human \sqsubseteq_\ell X, Horse \sqsubseteq_\ell X, X \sqsubseteq_\ell \exists parent.X\}$$

$X \sqsubseteq_\ell \exists parent.X$	$S_X = \{\exists parent.X\}$	Extension
$Human \sqsubseteq_{\ell-1} \exists parent.X$		Expansion
$Horse \sqsubseteq_{\ell-1} \exists parent.X$		
$Human \sqsubseteq_{\ell-1} \exists parent.X$		Mutation

Goal-Oriented Unification Algorithm. Example 2.

$$\mathcal{O} = \{Horse \sqsubseteq \exists parent.Horse, Human \sqsubseteq \exists parent.Human\}$$

$$\Gamma = \{Human \sqsubseteq_\ell X, Horse \sqsubseteq_\ell X, X \sqsubseteq_\ell \exists parent.X\}$$

$X \sqsubseteq_\ell \exists parent.X$	$S_X = \{\exists parent.X\}$	Extension
$Human \sqsubseteq_{\ell-1} \exists parent.X$		Expansion
$Horse \sqsubseteq_{\ell-1} \exists parent.X$		
$Human \sqsubseteq_{\ell-1} \exists parent.X$		Mutation
$Human \sqsubseteq_{\ell-1} Human$		
$\exists parent.Human \sqsubseteq_{\ell-1} \exists parent.X$		

Goal-Oriented Unification Algorithm. Example 2.

$$\mathcal{O} = \{Horse \sqsubseteq \exists parent.Horse, Human \sqsubseteq \exists parent.Human\}$$

$$\Gamma = \{Human \sqsubseteq_\ell X, Horse \sqsubseteq_\ell X, X \sqsubseteq_\ell \exists parent.X\}$$

$X \sqsubseteq_\ell \exists parent.X$	$S_X = \{\exists parent.X\}$	Extension
$Human \sqsubseteq_{\ell-1} \exists parent.X$		Expansion
$Horse \sqsubseteq_{\ell-1} \exists parent.X$		
$Human \sqsubseteq_{\ell-1} \exists parent.X$		Mutation
$Human \sqsubseteq_{\ell-1} Human$		
$\exists parent.Human \sqsubseteq_{\ell-1} \exists parent.X$		
$\exists parent.Human \sqsubseteq_{\ell-1} \exists parent.X$		Decomposition

Goal-Oriented Unification Algorithm. Example 2.

$$\mathcal{O} = \{Horse \sqsubseteq \exists parent.Horse, Human \sqsubseteq \exists parent.Human\}$$

$$\Gamma = \{Human \sqsubseteq_\ell X, Horse \sqsubseteq_\ell X, X \sqsubseteq_\ell \exists parent.X\}$$

$X \sqsubseteq_\ell \exists parent.X$	$S_X = \{\exists parent.X\}$	Extension
$Human \sqsubseteq_{\ell-1} \exists parent.X$		Expansion
$Horse \sqsubseteq_{\ell-1} \exists parent.X$		
$Human \sqsubseteq_{\ell-1} \exists parent.X$		Mutation
$Human \sqsubseteq_{\ell-1} Human$		
$\exists parent.Human \sqsubseteq_{\ell-1} \exists parent.X$		
$\exists parent.Human \sqsubseteq_{\ell-1} \exists parent.X$		Decomposition
$Human \sqsubseteq_{\ell-1} X$		
...		
$Human \sqsubseteq_0 X$		

$\mathcal{T} = \{X \equiv \exists parent.X\}$ is a hybrid unifier of Γ w.r.t. $\mathcal{O}$.

Goal-Oriented Unification Algorithm. Results.

Theorem

The algorithm is sound and complete.

Theorem

The algorithm is an NP-decision procedure for hybrid unification in $\mathcal{EL}$.

Conclusions.

Our Results...

- Definition of the hybrid unification problem in $\mathcal{EL}$ w.r.t. arbitrary ontologies.
- Hybrid $\mathcal{EL}$-unification is NP-complete.
- A sound and complete goal-oriented unification algorithm.

Future Work...

- Implementation of the algorithm. Optimizations (estimation of ℓ?).
- Decidability and complexity of classical $\mathcal{EL}$-unification w.r.t. arbitrary ontologies.

Conclusions.

Our Results...

- Definition of the hybrid unification problem in $\mathcal{EL}$ w.r.t. arbitrary ontologies.
- Hybrid $\mathcal{EL}$-unification is NP-complete.
- A sound and complete goal-oriented unification algorithm.

Future Work...

- Implementation of the algorithm. Optimizations (estimation of ℓ?).
- Decidability and complexity of classical $\mathcal{EL}$-unification w.r.t. arbitrary ontologies.

Thanks for your attention!