11

Unification

Claudia Chirita

School of Informatics, University of Edinburgh

THE UNIVERSITY of EDINBURGH informatics

Based on slides by: Jacques Fleuriot, Michael Rovatsos, Michael Herrmann, Vaishak Belle

11.a

Reducing FOL inference to propositional

SUBSTITUTIONS

Let (F, P) be a FOL signature and X, Y sets of variables.

A **substitution** of variables from X with **terms over** Y is a function $\theta\colon X \to T_F(Y)$.

A substitution θ can be **extended** to $\tilde{\theta}\colon T_F(X) \to T_F(Y)$

$$\tilde{\theta}(\sigma(t_1, \dots, t_n) = \sigma(\tilde{\theta}(t_1), \dots, \tilde{\theta}(t_n))$$

for $\sigma \in F_n, t_1, \dots, t_n \in T_F(X)$. In particular, $\tilde{\theta}(\sigma) = \sigma$ for $\sigma \in F_0$.

$\{x_1/t_1, \dots, x_n/t_n\}$ is a notation for $\theta\colon X \to T_F(Y)$ where

Y is the set of all variables occuring in the terms t_i

$\theta(x_i) = t_i$, for $i = 1, \dots, n$, and $\theta(x) = x$ for $x \neq x_i$

SUBSTITUTIONS

Let (F, P) be a FOL signature and X, Y, Z sets of variables.

Applying substitutions to sentences

We denote by $\varphi\,\theta$ the result of applying the substitution $\theta\colon X \to T_F(Y)$ to the sentence φ:

$$\varphi\,\theta = \begin{cases} \pi(\tilde{\theta}(t_1), \ldots, \tilde{\theta}(t_n)) & \text{for } \varphi = \pi(t_1, \ldots, t_n) \\ \tilde{\theta}(t) = \tilde{\theta}(t') & \text{for } \varphi = (t = t') \\ \neg(\varphi_1\,\theta) & \text{for } \varphi = \neg\varphi_1 \\ (\varphi_1\,\theta) \wedge (\varphi_2\,\theta) & \text{for } \varphi = \varphi_1 \wedge \varphi_2 \\ \ldots & \\ \forall Z.(\varphi_1\,\theta_Z) & \text{for } \varphi = \forall Z.\varphi_1 \end{cases}$$

Let (F, P) be a FOL signature and X, Y, Z sets of variables.

Composing substitutions $\theta\colon X \to T_F(Y)$ and $\delta\colon Y \to T_F(Z)$

$\theta \mathbin{;} \delta\colon X \to T_F(Z)$, with $(\theta \mathbin{;} \delta)(x) = (\theta \mathbin{;} \tilde{\delta})(x)$.

The composition of substitutions is **associative**.

The composition of substitutions is not **commutative**, sometimes not even well defined.

UNIVERSAL INSTANTIATION

Every instantiation of a universally quantified sentence φ is entailed by it:

$$\frac{\forall x.\varphi}{\varphi\{x/t\}}$$

for any variable x and **ground term** t (without variables).

EXAMPLE

$\forall x.\mathrm{King}(x) \land \mathrm{Greedy}(x) \to \mathrm{Evil}(x)$

$$\mathrm{King}(\mathrm{John}) \land \mathrm{Greedy}(\mathrm{John}) \to \mathrm{Evil}(\mathrm{John})$$

$$\mathrm{King}(\mathrm{Richard}) \land \mathrm{Greedy}(\mathrm{Richard}) \to \mathrm{Evil}(\mathrm{Richard})$$

$$\mathrm{King}(\mathrm{Father}(\mathrm{John})) \land \mathrm{Greedy}(\mathrm{Father}(\mathrm{John})) \to \mathrm{Evil}(\mathrm{Father}(\mathrm{John}))$$

EXISTENTIAL INSTANTIATION

For any sentence φ, variable x, and some constant σ that does not appear elsewhere in the knowledge base:

$$\frac{\exists x.\varphi}{\varphi\{x/\sigma\}}$$

EXAMPLE

$\exists x.\mathrm{Crown}(x) \wedge \mathrm{OnHead}(x, \mathrm{John})$ yields

$$\mathrm{Crown}(C) \wedge \mathrm{OnHead}(C, \mathrm{John})$$

with C a new constant symbol, called a **Skolem constant**.

REDUCTION TO PROPOSITIONAL INFERENCE

Consider a KB containing just the following:

$\forall x.\text{King}(x) \land \text{Greedy}(x) \rightarrow \text{Evil}(x)$
$\text{King}(\text{John}), \text{Greedy}(\text{John}), \text{Brother}(\text{Richard}, \text{John})$

Instantiating the universal sentence in all possible ways
(using substitutions $\{x/\text{John}\}$ and $\{x/\text{Richard}\}$) we obtain:

$\text{King}(\text{John}) \land \text{Greedy}(\text{John}) \rightarrow \text{Evil}(\text{John})$
$\text{King}(\text{Richard}) \land \text{Greedy}(\text{Richard}) \rightarrow \text{Evil}(\text{Richard})$

The universal sentence can then be discarded.

The new KB is essentially **propositional** if we view the atomic sentences
$\text{King}(\text{John}), \text{Greedy}(\text{John}), \text{Evil}(\text{John}), \text{King}(\text{Richard}), \ldots$
as propositional symbols.

REDUCTION TO PROPOSITIONAL INFERENCE

Every first-order KB and query can be **propositionalized** such that entailment is **preserved**.

A ground sentence is entailed by the new KB iff it is entailed by the original KB.

IDEA

Propositionalise KB and query and apply DPLL (or some other complete propositional method).

PROBLEM

If the KB includes a function symbol, the set of possible ground-term substitutions is infinite.

> e.g. infinitely many nested terms such as
> $\text{Father}(\text{Father}(\text{Father}(\text{John})))$

HERBRAND'S THEOREM

Theorem (Herbrand, 1930). If a sentence φ is entailed by a first-order KB, then it is entailed by a finite subset of the propositionalised KB.

IDEA

for $n = 0$ to ∞ do
create a propositional KB by instantiating with depth-n terms
see if φ is entailed by this KB

PROBLEM

Works if φ is entailed, but loops forever if it is not entailed.

SEMIDECIDABILITY

Theorem (Turing, 1936. Church, 1936).

Entailment for first-order logic is **semidecidable**.

Algorithms exist that say yes to every entailed sentence, but no algorithm exists that also says no to every non-entailed sentence.

11.b

Unification

PROBLEMS WITH PROPOSITIONALISATION

Propositionalisation is inefficient; it generates irrelevant sentences.

EXAMPLE

The inference of $\mathsf{Evil}(\mathsf{John})$ from

$$\forall x. \mathsf{King}(x) \land \mathsf{Greedy}(x) \to \mathsf{Evil}(x)$$
$$\mathsf{King}(\mathsf{John})$$
$$\forall y. \mathsf{Greedy}(y)$$
$$\mathsf{Brother}(\mathsf{Richard}, \mathsf{John})$$

seems obvious, but propositionalisation produces irrelevant facts such as $\mathsf{Greedy}(\mathsf{Richard})$.

For p k-ary predicates and n constants, there are $p \cdot n^k$ instantiations.

UNIFICATION

We can get the inference immediately if we can find a substitution θ such that $\text{King}(x)$ and $\text{Greedy}(x)$ match $\text{King}(\text{John})$ and $\text{Greedy}(y)$.

$\theta = \{x/\text{John}, y/\text{John}\}$ works.

Intuitively, **unification** of two sentences means to find a substitution such that the sentences become identical under its application.

$\theta \in \text{Unify}(\alpha, \beta)$ iff $\alpha\theta = \beta\theta$.

α	β	θ
$\text{Knows}(\text{John}, x)$	$\text{Knows}(\text{John}, \text{Jane})$	$\{x/\text{Jane}\}$
$\text{Knows}(\text{John}, x)$	$\text{Knows}(y, \text{OJ})$	$\{x/\text{OJ}, y/\text{John}\}$
$\text{Knows}(\text{John}, x)$	$\text{Knows}(y, \text{Mother}(y))$	$\{y/\text{John}, x/\text{Mother}(\text{John})\}$
$\text{Knows}(\text{John}, x)$	$\text{Knows}(x, \text{Richard})$	$[\text{fail}]$

TERM UNIFICATION

An **equation** is a pair of terms (t, t') with $t, t' \in T_F(X)$.
We denote the equation (t, t') as $t \stackrel{?}{=} t'$.

A **unification problem** is a finite set of equations
$U = \{t_1 \stackrel{?}{=} t'_1, \dots, t_n \stackrel{?}{=} t'_n\}$

A **unifier** (solution) for U is a substitution $\theta \colon X \to T_F(Y)$
s.t. $\theta(t_i) = \theta(t'_i)$, for $i = 1, \dots, n$.
We denote by $\mathrm{Unify}(U)$ the set of unifiers for U.

If $\theta = \{x_1/t_1, \dots, x_n/t_n\}$ then
$U\{x_1/t_1, \dots, x_n/t_n\} = \{\theta(t) \stackrel{?}{=} \theta(t') \mid t \stackrel{?}{=} t' \in U\}$.

MOST GENERAL UNIFIER

EXAMPLE

To unify Knows(John, x) and Knows(y, z),
$\theta = \{y/\text{John}, x/z\}$ or $\theta = \{y/\text{John}, x/\text{John}, z/\text{John}\}$.

The first unifier is **more general** than the second.

A unifier $\theta \in \text{Unify}(U)$ is **more general** than $\delta \in \text{Unify}(U)$ if there is a substitution τ s.t. $\delta = \theta \,;\, \tau$.

A unifier $\theta \in \text{Unify}(U)$ is a **most general unifier** (mgu) if for any $\delta \in \text{Unify}(U)$ there is a substitution τ s.t. $\delta = \theta \,;\, \tau$.

There is a single most general unifier that is unique up to renaming of variables.

EXAMPLE

$\text{mgu}(\{\text{John} \stackrel{?}{=} y, x \stackrel{?}{=} z\}) = \{y/\text{John}, x/z\}$

QUESTION TIME!

What is the most general unifier of the following equations?

$$\mathrm{Loves}(\mathrm{John}, x) \stackrel{?}{=} \mathrm{Loves}(y, \mathrm{Mother}(y))$$

$$\mathrm{Loves}(\mathrm{John}, \mathrm{Mother}(x)) \stackrel{?}{=} \mathrm{Loves}(y, y)$$

ANSWER TIME!

$\text{Loves}(\text{John}, x) \overset{?}{=} \text{Loves}(y, \text{Mother}(y))$

$\{x/\text{Mother}(\text{John}), y/\text{John}\}$

$\text{Loves}(\text{John}, \text{Mother}(x)) \overset{?}{=} \text{Loves}(y, y)$

Fail

UNIFICATION

Let $R = \{x_1 \stackrel{?}{=} t_1, \ldots, x_n \stackrel{?}{=} t_n\}$ be a unification problem with variables from X, and Y the set of variables occurring in t_i.

We say that R is **solved** if $x_i \neq x_j$ for $i \neq j$ and $x_i \notin Y$.

Any solved problem R defines a **substitution** θ_R

$\theta_R = \{x_1/t_1, \ldots, x_n/t_n\}$

$\theta_R \in \mathsf{Unify}(R)$

The following algorithm transforms a non-ground unification problem U into another non-ground unification problem R. If $R = \emptyset$, then U has no unifiers. Otherwise, R is solved, and the substitution θ_R determined by R is an mgu for U.

What happens if U is ground?

UNIFICATION ALGORITHM

Input $\mathcal{U} = \{t_1 \stackrel{?}{=} t'_1, \dots, t_n \stackrel{?}{=} t'_n\}$ a non-ground unification problem

Initialise $R = \mathcal{U}$

Execute non-deterministically the steps:

Delete: $R \cup \{t \stackrel{?}{=} t\} \Rightarrow R$ if t is ground

Switch: $R \cup \{t \stackrel{?}{=} x\} \Rightarrow R \cup \{x \stackrel{?}{=} t\}$ if x is a variable, and t is not

Decomposition:

$R \cup \{f(t_1, \dots, t_n) \stackrel{?}{=} f(t'_1, \dots, t'_n)\} \Rightarrow R \cup \{t_1 \stackrel{?}{=} t'_1, \dots, t_n \stackrel{?}{=} t'_n\}$

Conflict: $R \cup \{f(t_1, \dots, t_n) \stackrel{?}{=} g(t'_1, \dots, t'_k)\} \Rightarrow \emptyset$ if $f \neq g$

Eliminate: $R \cup \{x \stackrel{?}{=} t\} \Rightarrow \{x \stackrel{?}{=} t\} \cup R\{x/t\}$ if x is a variable that occurs in R but not in t, and t is not a variable

Occurs check: $R \cup \{x \stackrel{?}{=} t\} \Rightarrow \emptyset$ if x is a variable that occurs in t and $t \neq x$

Coalesce: $R \cup \{x \stackrel{?}{=} y\} \Rightarrow \{x \stackrel{?}{=} y\} \cup R\{x/y\}$ if x and y are variables occurring in R

Output if $R = \emptyset$, then there are no solutions for problem $\mathcal{U}$

if $R \neq \emptyset$, then R is an mgu for $\mathcal{U}$

$$U = R = \{\text{Loves}(\text{John}, x) \stackrel{?}{=} \text{Loves}(y, \text{Mother}(y))\}$$

$\Downarrow$ **Decompose**

$$R = \{\text{John} \stackrel{?}{=} y,\ x \stackrel{?}{=} \text{Mother}(y)\}$$

$\Downarrow$ **Switch**

$$R = \{y \stackrel{?}{=} \text{John},\ x \stackrel{?}{=} \text{Mother}(y)\}$$

$\Downarrow$ **Eliminate**

$$R = \{y \stackrel{?}{=} \text{John},\ x \stackrel{?}{=} \text{Mother}(\text{John})\}$$

SUMMARY

Rules for quantifiers

Reducing FOL to PL

Unification as equation solving