Informatics 2D · Agents and Reasoning · 2019/2020

Lecture 11 · Unification and Generalised Modus Ponens

Claudia Chirita

School of Informatics, University of Edinburgh

6th February 2020

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

Outline

- Reducing first-order inference to propositional inference
- Unification
- Generalized Modus Ponens

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:

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

Substitutions · Composition

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

Composing substitutions $\theta \colon X \to T_F(Y)$ and $\delta \colon Y \to T_F(Z)$:
$\theta \,;\, \delta \colon X \to T_F(Z)$, with $(\theta \,;\, \delta)(x) = (\theta \,;\, \tilde{\delta})(x)$.

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

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

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

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

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

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

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

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

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

Reduction to propositional inference

Consider a KB containing just the following:

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

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

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

The universal sentence can then be discarded.

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

Eg. infinitely many nested terms such as
$\mathrm{Father}(\mathrm{Father}(\mathrm{Father}(\mathrm{John})))$

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, 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.

Problems with propositionalisation

Propositionalisation is inefficient; it generates irrelevant sentences.

Example

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

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

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

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 $\mathrm{King}(x)$ and $\mathrm{Greedy}(x)$ match $\mathrm{King}(\mathrm{John})$ and $\mathrm{Greedy}(y)$.

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

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

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

α	β	θ
$\mathrm{Knows}(\mathrm{John}, x)$	$\mathrm{Knows}(\mathrm{John}, \mathrm{Jane})$	$\{x/\mathrm{Jane}\}$
$\mathrm{Knows}(\mathrm{John}, x)$	$\mathrm{Knows}(y, \mathrm{OJ})$	$\{x/\mathrm{OJ}, y/\mathrm{John}\}$
$\mathrm{Knows}(\mathrm{John}, x)$	$\mathrm{Knows}(y, \mathrm{Mother}(y))$	$\{y/\mathrm{John}, x/\mathrm{Mother}(\mathrm{John})\}$
$\mathrm{Knows}(\mathrm{John}, x)$	$\mathrm{Knows}(x, \mathrm{Richard})$	[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 $\theta : X \to T_F(Y)$
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 $\mathrm{Knows}(\mathrm{John}, x)$ and $\mathrm{Knows}(y, z)$,
$\theta = \{y/\mathrm{John}, x/z\}$ or $\theta = \{y/\mathrm{John}, x/\mathrm{John}, z/\mathrm{John}\}$.

The first unifier is more general than the second.

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

A unifier $\theta \in \mathrm{Unify}(U)$ is a most general unifier (mgu) if for any $\delta \in \mathrm{Unify}(U)$ 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

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

Example

What is the most general unifier of the following equations?

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

Example · Solution

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

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

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

  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

$\theta_R = \{x_1/t_1, \ldots, x_n/t_n\}$
$\theta_R \in \mathrm{Unify}(R)$

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: $U = \{t_1 \overset{?}{=} t'_1, \dots, t_n \overset{?}{=} t'_n\}$ a non-ground unification problem
Initialise: $R = U$

Execute non-deterministically the steps:

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

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

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

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

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

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

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

Output: if $R = \emptyset$, then there are no solutions for problem U
if $R \neq \emptyset$, then R is an mgu for U

Example

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

$\Downarrow$ 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

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

Generalized Modus Ponens (GMP)

For the atomic sentences $p_1, \dots, p_n, p'_1, \dots, p'_n, q$, and a unifier θ s.t. $p'_i\theta = p_i\theta$ for all i, we have the inference rule:

$$\frac{p'_1, p'_2, \dots, p'_n \quad (p_1 \wedge p_2 \wedge \dots \wedge p_n \rightarrow q)}{q\theta}$$

GMP is used with KB of definite clauses (one positive literal).

All variables are assumed universally quantified.

Example

p'_1 is $\mathrm{King(John)}$ p'_2 is $\mathrm{Greedy}(y)$

p_1 is $\mathrm{King}(x)$ p_2 is $\mathrm{Greedy}(x)$ q is $\mathrm{Evil}(x)$

θ is $(x/\mathrm{John}, y/\mathrm{John})$

$q\theta$ is $\mathrm{Evil(John)}$

GMP · Soundness

We need to show that $p'_1, \dots, p'_n, (p_1 \wedge \dots \wedge p_n \rightarrow q) \vDash q\theta$, provided that $p'_i\theta = p_i\theta$, for all i and θ a unifier.

Proof.

For any sentence p, we have that $p \vDash p\theta$ by the Universal Instantiation rule. Using this, we have:

1. $(p_1 \wedge \dots \wedge p_n \rightarrow q) \vDash (p_1 \wedge \dots \wedge p_n \rightarrow q)\theta = (p_1\theta \wedge \dots \wedge p_n\theta \rightarrow q\theta)$
2. $$\begin{aligned} p'_1, \dots, p'_n \vDash p'_1 \wedge \dots \wedge p'_n \vDash (p'_1 \wedge \dots \wedge p'_n)\theta &= p'_1\theta \wedge \dots \wedge p'_n\theta \\ &= p_1\theta \wedge \dots \wedge p_n\theta \end{aligned}$$
   because by the definition of generalized modus ponens we have that $p'_i\theta = p_i\theta$, for all i.
3. From the previous two steps, and by applying modus ponens, $q\theta$ follows.

Example · Winnie-the-Pooh

It is known in The Hundred-Acre Wood that if someone who is very fond of food gives a treat to one of their friends, they must be really generous.

Eeyore, the sad donkey, has some hunny that he has received for his birthday from Winnie-the-Pooh, who, as we know, is very fond of food.

Prove that Winnie-the-Pooh is generous.

Example · Winnie-the-Pooh

It is an act of generosity for someone very fond of food to share treats with his friends.

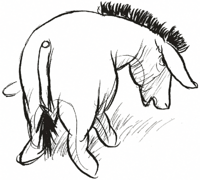
$\text{VeryFondOfFood}(x) \land \text{Treat}(y) \land \text{Friend}(z) \land \text{Gives}(x, y, z) \rightarrow \text{Generous}(x)$

Eeyore has some hunny.

$\exists x.\text{Owns}(\text{Eeyore}, x) \land \text{Hunny}(x)$

He must have received the hunny from Winnie-the-Pooh.

$\text{Hunny}(x) \land \text{Owns}(\text{Eeyore}, x) \rightarrow \text{Gives}(\text{Pooh}, x, \text{Eeyore})$



Example · Winnie-the-Pooh

Hunny is a treat.

$\text{Hunny}(x) \rightarrow \text{Treat}(x)$

Residents of The Hundred-Acre Wood are friends.

$\text{Resident}(x, \text{HundredAcreWood}) \rightarrow \text{Friend}(x)$

Eeyore is a resident of The Hundred-Acre Wood.

$\text{Resident}(\text{Eeyore}, \text{HundredAcreWood})$

Pooh is very fond of food.

$\text{VeryFondOfFood}(\text{Pooh})$

Summary

- Rules for quantifiers
- Reducing FOL to PL
- Unification as equation solving
- Generalized modus ponens