

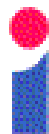


Unification and Generalised Modus Ponens

R&N: §9.1-9.2

Jacques Fleuriot

 School of
informatics
University of Edinburgh



Outline

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

Universal instantiation (UI)

- Every instantiation of a universally quantified formula α is entailed by it:

$$\frac{\forall v. \alpha}{\alpha\{v/g\}}$$

Contains no variables

for any **variable** v and **ground term** g

Example: $\forall x. King(x) \wedge Greedy(x) \Rightarrow Evil(x)$ yields:

$King(John) \wedge Greedy(John) \Rightarrow Evil(John)$

$King(Richard) \wedge Greedy(Richard) \Rightarrow Evil(Richard)$

$King(Father(John)) \wedge Greedy(Father(John)) \Rightarrow$
 $Evil(Father(John))$

etc...

Existential instantiation (EI)

- For any formula α , variable v , and some constant symbol k that does **not** appear elsewhere in the knowledge base:

$$\frac{\exists v. \alpha}{\alpha \{v/k\}}$$

Example. $\exists x. \text{Crown}(x) \wedge \text{OnHead}(x, \text{John})$ yields:

$$\text{Crown}(C_1) \wedge \text{OnHead}(C_1, \text{John})$$

provided C_1 is a new constant symbol, called a **Skolem constant**

Reduction to propositional inference

Suppose the KB contains just the following:

$\forall x. \text{King}(x) \wedge \text{Greedy}(x) \Rightarrow \text{Evil}(x)$

King(John)

Greedy(John)

Brother(Richard,John)

- Instantiating the universal sentence in all possible ways, we have:

King(John) \wedge Greedy(John) \Rightarrow Evil(John)

King(Richard) \wedge Greedy(Richard) \Rightarrow Evil(Richard)

King(John)

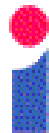
Greedy(John)

Brother(Richard,John)

Note: universal sentence can then be discarded

- The new KB is **propositionalized**: proposition symbols are

King(John), Greedy(John), Evil(John), King(Richard), etc.



Reduction contd.

- Every FOL KB can be propositionalized so as to **preserve** entailment
 - A ground sentence is entailed by new KB iff entailed by original KB
- Idea: propositionalize KB and query, apply DPLL (or some other complete propositional method), return result
- Problem: with function symbols, there are infinitely many ground terms,
 - e.g., *Father(Father(Father(John)))*

Reduction contd.

Theorem: Herbrand (1930). If a sentence α is entailed by a FOL KB, it is entailed by a finite subset of the propositionalized KB

Idea: For $n = 0$ to ∞ do
 create a propositional KB by instantiating with depth- n terms
 see if α is entailed by this KB

Problem: works if α is entailed, loops forever if α is not entailed

Theorem: Turing (1936), Church (1936). Entailment for FOL is **semi-decidable** (i.e. algorithms exist that say yes to every entailed sentence, but no algorithm exists that also says no to every non-entailed sentence.)

Problems with propositionalization

- Propositionalization seems to generate lots of irrelevant sentences.

- Example

- From:

$$\forall x. \text{King}(x) \wedge \text{Greedy}(x) \Rightarrow \text{Evil}(x)$$

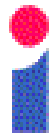
King(John)

$\forall y. \text{Greedy}(y)$

Brother(Richard,John)

- It seems obvious that *Evil(John)*, but propositionalization produces lots of facts such as *Greedy(Richard)* that are irrelevant

- With 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 $King(x)$ and $Greedy(x)$ match $King(John)$ and $Greedy(y)$
 - $\theta = \{x/John, y/John\}$ works
- $Unify(\alpha, \beta) = \theta$ iff $\alpha\theta = \beta\theta$

α	β	θ
Knows(John,x)	Knows(John,Jane)	
Knows(John,x)	Knows(y,OJ)	
Knows(John,x)	Knows(y,Mother(y))	
Knows(John,x)	Knows(x,Richard)	

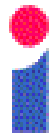
Unification

- We can get the inference immediately if we can find a substitution θ such that $King(x)$ and $Greedy(x)$ match $King(John)$ and $Greedy(y)$
 - $\theta = \{x/John, y/John\}$ works

- $Unify(\alpha, \beta) = \theta$ iff $\alpha\theta = \beta\theta$

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

- **Standardizing variables apart** eliminates overlap of variables, e.g., Change $Knows(x,Richard)$ to $Knows(z_{17},Richard)$ and then we succeed with $\theta = \{z_{17}/John, x/Richard\}$ for the last case



Unification

- To unify $Knows(John, x)$ and $Knows(y, z)$,
 $\theta = \{y/John, x/z\}$ or $\theta = \{y/John, x/John, z/John\}$
- The first unifier is **more general** than the second.
- FOL: There is a single **most general unifier** (MGU) that is unique up to renaming of variables.
 $MGU = \{y/John, x/z\}$
- Can be viewed as an **equation solving** problem.
 - *i.e. solve $Knows(John, x) \equiv Knows(y, z)$*

Example

What is the most general unifier, if any, of the following pairs of formulae?

- $\text{Loves}(\text{John}, x) \equiv \text{Loves}(y, \text{Mother}(y))$.
- $\text{Loves}(\text{John}, \text{Mother}(x)) \equiv \text{Loves}(y, y)$.

Solution

- $\text{Loves}(\text{John}, x) \equiv \text{Loves}(y, \text{Mother}(y))$.
 - $\{x/\text{Mother}(\text{John}), y/\text{John}\}$
- $\text{Loves}(\text{John}, \text{Mother}(x)) \equiv \text{Loves}(y, y)$.
 - Fails.

Finding the MGU

- Can be broken-down into a series of steps
 - Decomposition
 - Conflict
 - Eliminate
 - Delete
 - Switch
 - Coalesce
 - Occurs Check
- Other presentations of algorithm are possible (see R&N)

Decomposition

- *Given* $\text{Knows}(\text{John}, x) \equiv \text{Knows}(y, z)$.
- *Replace with* $\text{John} \equiv y$ and $x \equiv z$.
- *In general, given:*
$$f(s_1, \dots, s_n) \equiv f(t_1, \dots, t_n)$$
- *Replace with* $s_1 \equiv t_1, \dots, s_n \equiv t_n$.

Conflict

- *Given Knows(John, x) \equiv Greedy(y).*
- *Then fail.*
- *In general, given:*
$$f(s_1, \dots, s_m) \equiv g(t_1, \dots, t_n), \text{ where } f \neq g$$
- *Then **fail**.*

Eliminate

- *Given $\text{Knows}(\text{John}, x) \equiv \text{Knows}(y, z)$ and $z \equiv \text{Richard}$.*
- *Replace with*
 $\text{Knows}(\text{John}, x) \equiv \text{Knows}(y, \text{Richard})$ and $z \equiv \text{Richard}$.
- *In general, given: P and $x \equiv t$, where x occurs in P but not in t , and t is not a variable*
- *Replace with $P\{x/t\}$ and $x \equiv t$.*

Delete

- *Given $Greedy(John) \equiv Greedy(John)$.*
- *Remove this equation.*
- *In general, given P and $s \equiv s$*
- *Then replace with P .*

Switch

- Given $Knows(John, x) \equiv Knows(y, z)$ and $Richard \equiv z$.
- Replace with $Knows(John, x) \equiv Knows(y, z)$ and $z \equiv Richard$.
- In general, given: P and $s \equiv x$, where x , but not s , is a variable
- Replace with P and $x \equiv s$.

Coalesce

- *Given $\text{Knows}(\text{John}, x) \equiv \text{Knows}(y, z)$ and $y \equiv z$.*
- *Replace with $\text{Knows}(\text{John}, x) \equiv \text{Knows}(z, z)$ and $y \equiv z$.*
- *In general, given P and $x \equiv y$, where x and y are variables occurring in P .*
- *Replace with $P\{x/y\}$ and $x \equiv y$.*

Occurs Check

- *Given $x \equiv \text{Father}(x)$.*
- *Then fail, else eliminate will loop.*
 - $P(x)$ and $x \equiv \text{Father}(x) \mapsto P(\text{Father}(\text{Father}(\dots)))$.
- *In general, given $x \equiv s$,*
 - *where x occurs in s and s is not a variable*
- *Then **fail**.*

Example

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

↓ Decompose

$$\text{John} \equiv y \wedge x \equiv \text{Mother}(y)$$

↓ Switch

$$y \equiv \text{John} \wedge x \equiv \text{Mother}(y)$$

↓ Eliminate

$$y \equiv \text{John} \wedge x \equiv \text{Mother}(\text{John})$$

Generalized Modus Ponens (GMP)

$$\frac{p_1', p_2', \dots, p_n', (p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q)}{q\theta} \quad \text{where } p_i'\theta \equiv p_i \theta \text{ for all } i$$

Example:

p_1' is *King(John)*

p_1 is *King(x)*

p_2' is *Greedy(y)*

p_2 is *Greedy(x)*

θ is {x/John,y/John}

q is *Evil(x)*

$q\theta$ is *Evil(John)*

- GMP used with KB of **definite clauses** (exactly one positive literal)
- All variables assumed universally quantified

Soundness of GMP

- 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

- **Lemma:** For any sentence p , we have $p \vDash p\theta$ by UI

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. $p_1', \dots, p_n' \vDash p_1' \wedge \dots \wedge p_n' \vDash p_1'\theta \wedge \dots \wedge p_n'\theta = p_1\theta \wedge \dots \wedge p_n\theta$

since by definition of GMP $p_i'\theta = p_i\theta$ for all i

3. From 1 and 2, $q\theta$ follows by ordinary Modus Ponens

Example Knowledge Base

- The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.
- Prove that Colonel West is a criminal

Example Knowledge Base (contd)

... it is a crime for an American to sell weapons to hostile nations:

$American(x) \wedge Weapon(y) \wedge Sells(x,y,z) \wedge Hostile(z) \Rightarrow Criminal(x)$

Nono ... has some missiles, i.e., $\exists x Owns(Nono,x) \wedge Missile(x)$:

$Owns(Nono,M_1)$ and $Missile(M_1)$

... all of its missiles were sold to it by Colonel West

$Missile(x) \wedge Owns(Nono,x) \Rightarrow Sells(West,x,Nono)$

Missiles are weapons:

$Missile(x) \Rightarrow Weapon(x)$

An enemy of America counts as “hostile”:

$Enemy(x,America) \Rightarrow Hostile(x)$

West, who is American ...

$American(West)$

The country Nono, an enemy of America ...

$Enemy(Nono,America)$

Summary

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