Inf2D 11: Unification and Generalised Modus Ponens

Cristina Alexandru

School of Informatics, University of Edinburgh

08/02/18

Slide Credits: Jacques Fleuriot, Michael Rovatsos, Michael Herrmann
Outline

- Reducing first-order inference to propositional inference
- Unification
- Generalized Modus Ponens
Every instantiation of a universally quantified formula $\alpha$ is entailed by it:

$$\forall v. \alpha$$

$$\alpha \{v/g\}$$

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

Example:

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

yields:

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

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

$$\text{King}(\text{Father}(\text{John})) \land \text{Greedy}(\text{Father}(\text{John})) \Rightarrow \text{Evil}(\text{Father}(\text{John}))$$

eetc...
Existential instantiation (EI)

For any formula $\alpha$, variable $v$, and some constant symbol $k$ that does not appear elsewhere in the knowledge base:

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

Example. $\exists x. Crown(x) \land OnHead(x, John)$ yields:

$$
Crown(C_1) \land OnHead(C_1, John)
$$

provided $C_1$ is a new constant symbol, called a Skolem constant
Reduction to propositional inference

Suppose the KB contains just the following:
\( \forall x. \text{King}(x) \land \text{Greedy}(x) \Rightarrow \text{Evil}(x) \)
King(John), Greedy(John), Brother(Richard,John)

Instantiating the universal sentence in all possible ways:
King(John) \land \text{Greedy}(John) \Rightarrow \text{Evil}(John)
King(Richard) \land \text{Greedy}(Richard) \Rightarrow \text{Evil}(Richard)
King(John), \text{Greedy}(John), \text{Brother}(Richard,John)

Note: universal sentence can then be discarded

The new KB is **propositionalised**: proposition symbols are
King(John), Greedy(John), Evil(John), King(Richard), etc.

Note: it’s not in KB as a fact
Every FOL KB can be propositionalised so as to **preserve** entailment

- A ground sentence is entailed by new KB iff entailed by original KB

**Idea:** propositionalise 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)))
Theorem: Herbrand (1930). If a sentence $\alpha$ is entailed by a FOL KB, it is entailed by a finite subset of the propositionalised KB

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

- Problem: works if $\alpha$ is entailed, loops forever if $\alpha$ 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 propositionalisation

- Propositionalisation seems to generate lots of irrelevant sentences.

Example

- From:
  \[ \forall x.\text{King}(x) \land \text{Greedy}(x) \Rightarrow \text{Evil}(x) \]
  \[ \text{King}(\text{John}) \]
  \[ \forall y.\text{Greedy}(y) \]
  \[ \text{Brother}(\text{Richard},\text{John}) \]

- It seems obvious that \(\text{Evil}(\text{John})\), but propositionalisation produces lots of facts such as \(\text{Greedy}(\text{Richard})\) that are irrelevant.

- With \(p\) \(k\)-ary predicates and \(n\) constants, there are \(p \cdot n^k\) instantiations.
We can get the inference immediately if we can find a substitution $\theta$ 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

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

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Knows(John,} x)$</td>
<td>$\text{Knows(John,} \text{Jane})$</td>
<td></td>
</tr>
<tr>
<td>$\text{Knows(John,} x)$</td>
<td>$\text{Knows(y,OJ)}$</td>
<td></td>
</tr>
<tr>
<td>$\text{Knows(John,} x)$</td>
<td>$\text{Knows(y,}\text{Mother(y)})$</td>
<td></td>
</tr>
<tr>
<td>$\text{Knows(John,} x)$</td>
<td>$\text{Knows(x, Richard)}$</td>
<td></td>
</tr>
</tbody>
</table>

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

- We can get the inference immediately if we can find a substitution $\theta$ 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

- Unification of two sentences means to find a substitution $\theta$ such that the sentences become identical.

- Symbolically, this substitution is produced by the *Unify* operator: $\text{Unify}(\alpha, \beta) = \theta$ iff $\alpha\theta \equiv \beta\theta$

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knows(John,x)</td>
<td>Knows(John,Jane)</td>
<td>${x/\text{Jane}}$</td>
</tr>
<tr>
<td>Knows(John,x)</td>
<td>Knows(y,OJ)</td>
<td>${x/\text{OJ}, y/\text{John}}$</td>
</tr>
<tr>
<td>Knows(John,x)</td>
<td>Knows(y,Mother(y))</td>
<td>${y/\text{John}, x/\text{Mother(John)}}$</td>
</tr>
<tr>
<td>Knows(John,x)</td>
<td>Knows(x, Richard)</td>
<td>[fail]</td>
</tr>
</tbody>
</table>
We cannot unify the sentences $Knows(John, x)$ and $Knows(x, Richard)$ if the two “$x$” are referring to the same variable.

If the two “$x$” are unrelated and just due to inconsiderate naming of an unknown, then we can rename one of them and avoid thus the clash of variables. This is called:

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

FOL: There is a single most general unifier (MGU) that is unique up to renaming of variables.

- MGU = \( \{ y/\text{John}, x/z \} \)

Can be viewed as an equation solving problem.
- i.e. solve \( \text{Knows}(\text{John},x) \equiv \text{Knows}(y,z) \)
What is the most general unifier, if any, of the following pairs of formulae?

- Loves(John, x) ≡ Loves(y, Mother(y)).
- Loves(John, Mother(x)) ≡ Loves(y, y).
Solution

\[ \text{Loves}(\text{John}, x) \equiv \text{Loves}(y, \text{Mother}(y)). \]

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

\[ \text{Loves}(\text{John}, \text{Mother}(x)) \equiv \text{Loves}(y, y). \]

\[ \text{Fails} \]
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: Knows(John, x) ≡ Knows(y, z).
Replace with John ≡ y and x ≡ z.

In general, given

\[ f(s_1, \ldots, s_n) \equiv f(t_1, \ldots, t_n) \]

Replace with \( s_1 \equiv t_1, \ldots, s_n \equiv t_n \).
Given: \( \text{Knows}(\text{John}, x) \equiv \text{Greedy}(y) \).

Then fail.

In general, given:

\[ f(s_1, \ldots, s_m) \equiv g(t_1, \ldots, t_n), \text{ where } f \neq g \]

Then fail.
Eliminate

- Given: Knows(John, x) ≡ Knows(y, z) and z ≡ Richard.
  Replace with Knows(John, x) ≡ Knows(y, Richard) and z ≡ Richard.

- In general, given: P and x ≡ t, where x occurs in P but not in t, and t is not a variable
  Replace with P \{x/t\} and x ≡ t.
Given: Greedy(John) ≡ Greedy(John).
Remove this equation.

In general, given \( P \) and \( s ≡ s \)
Then replace with \( P \).
Given: Knows(John, x) ≡ Knows(y, z) and Richard ≡ z.
Replace with Knows(John, x) ≡ Knows(y, z) and z ≡ Richard.

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

Then fail, else eliminate will loop.

- $P(x)$ and $x \equiv \text{Father}(x) \iff P(\text{Father}(\text{Father}(\ldots))))$.

In general, given $x \equiv s$, where $x$ occurs in $s$ and $s$ is not a variable

- Then fail.
Example

Loves(John, x) ≡ Loves(y, Mother(y))

\[\Downarrow\] Decompose

John ≡ y \land x ≡ Mother(y)

\[\Downarrow\] Switch

y ≡ John \land x ≡ Mother(y)

\[\Downarrow\] Eliminate

y ≡ John \land x ≡ Mother(John)
Generalized Modus Ponens (GMP)

\[
p_1', p_2', \ldots, p_n' (p_1 \land p_2 \land \cdots \land p_n \Rightarrow q) \quad \text{when } p_i'\theta \equiv p_i\theta \, \forall \, i
\]

Example:

- \( p_1' \) is King (John) \hspace{1cm} p_1 \) is King(\( x \))
- \( p_2' \) is Greedy(\( y \)) \hspace{1cm} p_2 \) is Greedy(\( x \))
- \( \theta \) is (\( x/\text{John}, y/\text{John} \)) \hspace{1cm} 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

\[ p'_1, \ldots, p'_n, (p_1 \land \cdots \land p_n \Rightarrow q) \models q\theta \]

provided that \( p'_i\theta = p_i\theta \) for all \( i \)

- Lemma: For any sentence \( p \), we have \( p \models p\theta \) by UI

1. \((p_1 \ldots p_n \Rightarrow q) \models (p_1 \ldots p_n \Rightarrow q)\theta = (p_1\theta \ldots p_n\theta \Rightarrow q\theta)\)

2. \( p'_1, \ldots, p'_n \models p'_1 \ldots p'_n \models p'_1\theta \ldots p'_n\theta = p_1\theta \ldots p_n\theta \)

   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
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
... it is a crime for an American to sell weapons to hostile nations:
\[
\text{American}(x) \land \text{Weapon}(y) \land \\
\text{Sells}(x,y,z) \land \text{Hostile}(z) \Rightarrow \text{Criminal}(x)
\]

Nono ... has some missiles, i.e., \( \exists x \) Owns(Nono,\( x \)) \land Missile(\( x \)):
\[
\text{Owns}(\text{Nono},M_1) \text{ and Missile}(M_1)
\]
... all of its missiles were sold to it by Colonel West
\[
\text{Missile}(x) \land \text{Owns}(\text{Nono},x) \Rightarrow \text{Sells}(\text{West},x,\text{Nono})
\]

Missiles are weapons:
\[
\text{Missile}(x) \Rightarrow \text{Weapon}(x)
\]

An enemy of America counts as “hostile”:
\[
\text{Enemy}(x,\text{America}) \Rightarrow \text{Hostile}(x)
\]

West, who is American ...
\[
\text{American}(\text{West})
\]

The country Nono, an enemy of America ...
\[
\text{Enemy}(\text{Nono},\text{America})
\]
Summary

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