Inference in first-order logic:
Reducing first-order inference to propositional inference

AIMA
Chapter 9.1-9.2, 9.5
Outline

- Reducing first-order inference to propositional inference
- Unification
Universal instantiation (UI)

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

\[
\forall v \alpha \\
\text{Subst}([v/g], \alpha)
\]

for any variable \(v\) and ground term \(g\)

- E.g., \(\forall x \text{King}(x) \land \text{Greedy}(x) \Rightarrow \text{Evil}(x)\) yields any or all of:

\(v=x, \alpha = (\text{King}(x) \land \text{Greedy}(x) \Rightarrow \text{Evil}(x)\)

\((g=\text{John}): \text{King}(\text{John}) \land \text{Greedy}(\text{John}) \Rightarrow \text{Evil}(\text{John})\)

\((g=\text{Richard}): \text{King}(\text{Richard}) \land \text{Greedy}(\text{Richard}) \Rightarrow \text{Evil}(\text{Richard})\)

\((g=\text{GoldCrown}): \text{King}(\text{GoldCrown}) \land \text{Greedy}(\text{GoldCrown}) \Rightarrow \text{Evil}(\text{GoldCrown})\)

\[
\ldots
\]
Existential instantiation (EI) (simple case)

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

$$
\exists v \alpha \\
\text{Subst}\{\{v/k\}, \alpha\}
$$

- E.g., $\exists x \text{Crown}(x) \land \text{OnHead}(x,\text{John})$ yields:

$$
\text{Crown}(C1) \land \text{OnHead}(C1,\text{John})
$$

provided $C1$ is a new constant symbol, called a Skolem constant
Existential Instantiation *continued*

- UI can be applied several times to *add* new sentences
  - the new KB is logically equivalent to the old

- EI can be applied once to *replace* the existential sentence
  - the new KB is *not* equivalent to the old
  - but it is satisfiable iff the old KB was satisfiable
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, we have:

King(John) \land Greedy(John) \Rightarrow Evil(John)
King(Richard) \land Greedy(Richard) \Rightarrow Evil(Richard)
King(John)
Greedy(John)
Brother(Richard,John)

- The new KB is propositionalized: atomic proposition symbols are 
  King(John), Greedy(John), Evil(John), King(Richard), etc.
Reduction contd.

- **Claim:** 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 resolution, return result

- **Problem:** with function symbols, there are infinitely many ground terms,
  - e.g., $\text{Father}($Father($\text{Father}(\text{John}))$)
Problems with propositionalization

1. Propositionalization generates lots of irrelevant sentences.

2. With $p$ $k$-ary predicates and $n$ constants, there are $p \cdot n^k$ instantiations.
   - With function symbols, doing this naively can result in an unbounded number of sentences!
Substitution and Unification

- **Subst(θ, p)**: the result of substituting θ into sentence p
  - If θ = {x/Jane} (i.e. “substitute “Jane” for “x”), then
    \[ \text{Subst}(\theta, \text{Happy}(x)) = \text{Happy}(\text{Jane}) \]

- Unify algorithm: takes 2 sentences p and q and returns a unifier θ if one exists

  \[
  \text{Unify}(p, q) = \theta \quad \text{where} \quad \text{Subst}(\theta, p) = \text{Subst}(\theta, q)
  \]

- Example:
  - p = Knows(John,x)
  - q = Knows(John, Jane)
  \[
  \text{Unify}(p, q) = \{x/Jane\} 
  \]
Unification

More Examples:

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{Knows}(John, x)</td>
<td>\text{Knows}(John, 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, OJ)</td>
<td></td>
</tr>
</tbody>
</table>
## Unification

### More Examples:

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Knows(John,x)}$</td>
<td>$\text{Knows(John,Jane)}$</td>
<td>${x/\text{Jane}}$</td>
</tr>
<tr>
<td>$\text{Knows(John,x)}$</td>
<td>$\text{Knows(y,OJ)}$</td>
<td>${x/\text{OJ},y/\text{John}}$</td>
</tr>
<tr>
<td>$\text{Knows(John,x)}$</td>
<td>$\text{Knows(y,Mother(y))}$</td>
<td>${y/\text{John},x/\text{Mother(John)}}$</td>
</tr>
<tr>
<td>$\text{Knows(John,x)}$</td>
<td>$\text{Knows(x,OJ)}$</td>
<td>$\text{fail}$</td>
</tr>
</tbody>
</table>
Most General Unifier

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

  or

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

- The first unifier is more general than the second.

- There is a single most general unifier (MGU) that is unique up to renaming of variables.
  $\text{MGU} = \{y/\text{John}, x/z\}$
Generalized Modus Ponens (GMP)

\[
p_1', p_2', \ldots, p_n' \quad (p_1 \land p_2 \land \ldots \land p_n \implies q) \quad \frac{q \theta}{\text{where } p_i' \theta = p_i \theta \text{ for all } i}
\]

Example:

\[
\text{King(John), Greedy(y)} \quad (\text{King}(x) \land \text{Greedy}(x) \implies \text{Evil}(x))
\]

\[
\text{Evil(John)} \quad \theta = \{x/\text{John}, y/\text{John}\}
\]

- GMP used with KB of definite clauses (special case of Horn clause with exactly one positive literal)
- All variables assumed universally quantified
Resolution for FOL: brief summary

- Full first-order version:
\[
\frac{\ell_1 \lor \cdots \lor \ell_k, \quad m_1 \lor \cdots \lor m_n}{(\ell_1 \lor \cdots \lor \ell_i \lor \ell_{i+1} \lor \cdots \lor \ell_k \lor m_1 \lor \cdots \lor m_{j-1} \lor m_{j+1} \lor \cdots \lor m_n)\theta}
\]

where \(\text{Unify}(\ell_i, \neg m_j) = \theta\).

- The two clauses are assumed to be “standardized apart” so that they share no variables.

- For example,
\[
\frac{\neg \text{Rich}(x) \lor \text{Unhappy}(x)}{\text{Rich}(Ken) \lor \text{Unhappy}(Ken) \lor \theta = \{x/\text{Ken}\}}
\]

- Apply resolution steps to CNF(KB \(\land \neg \alpha\)); complete for FOL
Conversion to CNF for FOL

- Everyone who loves all animals is loved by someone:
  \[ \forall x [\forall y \text{Animal}(y) \Rightarrow \text{Loves}(x,y)] \Rightarrow [\exists y \text{Loves}(y,x)] \]

1. Eliminate biconditionals and implications
   \[ \forall x [\neg \forall y \neg \text{Animal}(y) \lor \text{Loves}(x,y)] \lor [\exists y \text{Loves}(y,x)] \]

2. Move \( \neg \) inwards:
   \[ \neg \forall x p \equiv \exists x \neg p, \quad \neg \exists x p \equiv \forall x \neg p \]
   \[ \forall x [\exists y \neg (\neg \text{Animal}(y) \lor \text{Loves}(x,y))] \lor [\exists y \text{Loves}(y,x)] \]
   \[ \forall x [\exists y \neg \neg \text{Animal}(y) \land \neg \text{Loves}(x,y)] \lor [\exists y \text{Loves}(y,x)] \]
   \[ \forall x [\exists y \text{Animal}(y) \land \neg \text{Loves}(x,y)] \lor [\exists y \text{Loves}(y,x)] \]
Conversion to CNF contd.

3. Standardize variables: each quantifier should use a different one
   \[ \forall x \exists y \text{Animal}(y) \land \neg \text{Loves}(x,y)] \lor [\exists z \text{Loves}(z,x)] \\]

4. Skolemize: a more general form of existential instantiation.
   - Each existential variable is replaced by a Skolem function of the enclosing universally quantified variables:
     \[ \forall x [\text{Animal}(F(x)) \land \neg \text{Loves}(x,F(x))] \lor \text{Loves}(G(x),x) \\]

5. Drop universal quantifiers:
   \[ [\text{Animal}(F(x)) \land \neg \text{Loves}(x,F(x))] \lor \text{Loves}(G(x),x) \\]

6. Distribute \( \lor \) over \( \land \):
   \[ [\text{Animal}(F(x)) \lor \text{Loves}(G(x),x)] \land [\neg \text{Loves}(x,F(x)) \lor \text{Loves}(G(x),x)] \]
Example proof: Did Curiosity kill the cat?

- Jack owns a dog. Every dog owner is an animal lover. No animal lover kills an animal. Either Jack or Curiosity killed the cat, who is named Tuna. Did Curiosity kill the cat?

- The axioms can be represented as follows:
  A. $(\exists x) \text{Dog}(x) \land \text{Owns}(\text{Jack}, x)$
  B. $(\forall x) ((\exists y) \text{Dog}(y) \land \text{Owns}(x, y)) \Rightarrow \text{AnimalLover}(x)$
  C. $(\forall x) \text{AnimalLover}(x) \Rightarrow (\forall y) \text{Animal}(y) \Rightarrow \neg \text{Kills}(x, y)$
  D. $\text{Kills}(\text{Jack}, \text{Tuna}) \lor \text{Kills}(\text{Curiosity}, \text{Tuna})$
  E. $\text{Cat}(\text{Tuna})$
  F. $(\forall x) \text{Cat}(x) \Rightarrow \text{Animal}(x)$

(This and next slide adapted from CMSC421 – Fall 2006, Univ. of Maryland)
Example: Did Curiosity kill the cat?

1. Dog(spike)
2. Owns(Jack,spike)
3. \( \sim \text{Dog}(y) \lor \sim \text{Owns}(x, y) \lor \text{AnimalLover}(x) \)
4. \( \sim \text{AnimalLover}(x1) \lor \sim \text{Animal}(y1) \lor \sim \text{Kills}(x1,y1) \)
5. \( \text{Kills}(\text{Jack},\text{Tuna}) \lor \text{Kills}(\text{Curiosity},\text{Tuna}) \)
6. Cat(\text{Tuna})
7. \( \sim \text{Cat}(x2) \lor \text{Animal}(x2) \)

8. \( \sim \text{Kills}(\text{Curiosity},\text{Tuna}) \) negated goal
9. \( \text{Kills}(\text{Jack},\text{Tuna}) \) 5,8
10. \( \sim \text{AnimalLover}(\text{Jack}) \lor \sim \text{Animal}(\text{Tuna}) \) 9,4 x1/Jack,y1/Tuna
11. \( \sim \text{Dog}(y) \lor \sim \text{Owns}(\text{Jack},y) \lor \sim \text{Animal}(\text{Tuna}) \) 10,3 x/Jack
12. \( \sim \text{Owns}(\text{Jack},\text{spike}) \lor \sim \text{Animal}(\text{Tuna}) \) 11,1
13. \( \sim \text{Animal}(\text{Tuna}) \) 12,2
14. \( \sim \text{Cat}(\text{Tuna}) \) 13,7 x2/Tuna
15. False 14,6