Lecture 9: Inference in First Order Logic
CS 580 (001) - Spring 2016

Amarda Shehu

Department of Computer Science
George Mason University, Fairfax, VA, USA

March 30, 2016
1. Outline of Today’s Class

2. Reducing First-order Inference to Propositional Inference

3. Unification

4. Generalized Modus Ponens

5. Forward and Backward Chaining
   - Forward Chaining
   - Backward Chaining

6. Logic Programming

7. Resolution
<table>
<thead>
<tr>
<th>Year</th>
<th>Figure</th>
<th>Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>450B.C.</td>
<td>Stoics</td>
<td>propositional logic, inference (maybe)</td>
</tr>
<tr>
<td>322B.C.</td>
<td>Aristotle</td>
<td>“syllogisms” (inference rules), quantifiers</td>
</tr>
<tr>
<td>1565</td>
<td>Cardano</td>
<td>probability theory (propositional logic + uncertainty)</td>
</tr>
<tr>
<td>1847</td>
<td>Boole</td>
<td>propositional logic (again)</td>
</tr>
<tr>
<td>1879</td>
<td>Frege</td>
<td>first-order logic</td>
</tr>
<tr>
<td>1922</td>
<td>Wittgenstein</td>
<td>proof by truth tables</td>
</tr>
<tr>
<td>1930</td>
<td>Gödel</td>
<td>∃ complete algorithm for FOL</td>
</tr>
<tr>
<td>1930</td>
<td>Herbrand</td>
<td>complete algorithm for FOL (reduce to propositional)</td>
</tr>
<tr>
<td>1931</td>
<td>Gödel</td>
<td>¬∃ complete algorithm for arithmetic</td>
</tr>
<tr>
<td>1960</td>
<td>Davis/Putnam</td>
<td>“practical” algorithm for propositional logic</td>
</tr>
<tr>
<td>1965</td>
<td>Robinson</td>
<td>“practical” algorithm for FOL—resolution</td>
</tr>
</tbody>
</table>
Universal Instantiation (UI)

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

\[ \forall v \alpha \quad \text{for any variable } v \text{ and ground term } g \]

E.g., \( \forall x \ King(x) \land Greedy(x) \implies Evil(x) \) yields

\[
\begin{align*}
\text{King}(John) \land \text{Greedy}(John) & \implies \text{Evil}(John) \\
\text{King}(Richard) \land \text{Greedy}(Richard) & \implies \text{Evil}(Richard) \\
\text{King}(\text{Father}(John)) \land \text{Greedy}(\text{Father}(John)) & \implies \text{Evil}(\text{Father}(John)) \\
& \vdots
\end{align*}
\]
Existential Instantiation (EI)

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

$$\exists v \alpha$$

$$\text{Subst}\left(\{v/k\}, \alpha\right)$$

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

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

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

Another example: from $\exists x d(x^y)/dy = x^y$ we obtain

$$d(e^y)/dy = e^y$$

provided $e$ is a new constant symbol
UI can be applied several times to **add** new sentences; the new KB is logically equivalent to the old.

El can be applied once to **replace** the existential sentence; the new KB is **not** equivalent to the old, but is satisfiable iff the old KB was satisfiable.
Suppose the KB contains 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, we have

\[ \text{King}(\text{John}) \land \text{Greedy}(\text{John}) \Rightarrow \text{Evil}(\text{John}) \]
\[ \text{King}(\text{Richard}) \land \text{Greedy}(\text{Richard}) \Rightarrow \text{Evil}(\text{Richard}) \]
\[ \text{King}(\text{John}) \]
\[ \text{Greedy}(\text{John}) \]
\[ \text{Brother}(\text{Richard}, \text{John}) \]

The new KB is propositionalized: proposition symbols are

\[ \text{King}(\text{John}), \text{Greedy}(\text{John}), \text{Evil}(\text{John}), \text{King}(\text{Richard}) \text{ etc.} \]
**Claim:** a ground sentence* is entailed by new KB iff entailed by original KB

**Claim:** every FOL KB can be propositionalized so as to preserve entailment

**Idea:** propositionalize KB and query, apply resolution, return result

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

**Theorem:** Herbrand (1930). If a sentence \(\alpha\) is entailed by an FOL KB, it is entailed by a **finite** subset of the propositional 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 if \(\alpha\) is not entailed

**Theorem:** Turing (1936), Church (1936). Entailment in FOL is **semidecidable**
Propositionalization seems to generate lots of irrelevant sentences.
E.g., from
\[
\forall x \ King(x) \land \ Greedy(x) \implies Evil(x)
\]
\[
\begin{align*}
\text{King}(\text{John}) \\
\forall y \ Greedy(y) \\
\text{Brother}(\text{Richard}, \text{John})
\end{align*}
\]
it seems obvious that \text{Evil}(\text{John}), but propositionalization 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

With function symbols, it gets much much worse!
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

$\text{UNIFY}(\alpha, \beta) = \theta$ if $\alpha\theta = \beta\theta$

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Knows}(\text{John}, x)$</td>
<td>$\text{Knows}(\text{John}, \text{Jane})$</td>
<td>${x/\text{Jane}}$</td>
</tr>
</tbody>
</table>
| $\text{Knows}(\text{John}, x)$ | $\text{Knows}(y, \text{OJ})$ | }
We can get the inference immediately if we can find a substitution $\theta$ such that $\text{King}(x)$ and $\text{Greedy}(x)$ match $\text{King}(John)$ and $\text{Greedy}(y)$.

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

$\text{UNIFY}(\alpha, \beta) = \theta$ if $\alpha\theta = \beta\theta$

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

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</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>${x/OJ, y/\text{John}}$</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, OJ)</td>
<td></td>
</tr>
</tbody>
</table>
We can get the inference immediately if we can find a substitution $\theta$ such that $\textit{King}(x)$ and $\textit{Greedy}(x)$ match $\textit{King}(\textit{John})$ and $\textit{Greedy}(y)$

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

$\text{UNIFY} (\alpha, \beta) = \theta$ if $\alpha \theta = \beta \theta$

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

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

$$\text{UNIFY}(\alpha, \beta) = \theta \text{ if } \alpha\theta = \beta\theta$$

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

Standardizing apart eliminates overlap of variables, e.g.,
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

$\text{UNIFY}(\alpha, \beta) = \theta$ if $\alpha\theta = \beta\theta$

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

**Standardizing apart** eliminates overlap of variables, e.g., $\text{Knows}(z_{17}, \text{OJ})$
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

$\text{UNIFY}(\alpha, \beta) = \theta$ if $\alpha\theta = \beta\theta$

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

Standardizing apart eliminates overlap of variables, e.g., $\text{Knows}(z_{17}, \text{OJ})$
Generalized Modus Ponens (GMP)

\[
p_1', \ p_2', \ \ldots, \ p_n', \ (p_1 \land p_2 \land \ldots \land p_n \Rightarrow q) \quad \Rightarrow \quad q^\theta
\]

where \( p_i^\theta = p_i^\theta \) for all \( i \)

\[\begin{align*}
p_1' \ &\text{is } \text{King}(\text{John}) & p_1 \ &\text{is } \text{King}(x) \\
p_2' \ &\text{is } \text{Greedy}(y) & p_2 \ &\text{is } \text{Greedy}(x) \\
\theta \ &\text{is } \{x/\text{John}, \ y/\text{John}\} & q \ &\text{is } \text{Evil}(x) \\
q^\theta \ &\text{is } \text{Evil}(\text{John})
\end{align*}\]

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

All variables assumed universally quantified
Soundness of GMP

Need to show that:

\[ p_1', \ldots, p_n', (p_1 \land \ldots \land p_n \Rightarrow q) \models q_\theta \]

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

**Lemma:** For any definite clause \( p \), we have \( p \models p_\theta \) by UI

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

2. \( p_1', \ldots, p_n' \models p_1' \land \ldots \land p_n' \models p_1' \theta \land \ldots \land p_n' \theta \)

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 Col. 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.,

\[ \exists x \text{Owns}(\text{Nono}, x) \land \text{Missile}(x) \]

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

\[ \forall x \text{Missile}(x) \land \text{Owns}(\text{Nono}, x) \Rightarrow \text{Sells}(\text{West}, x, \text{Nono}) \]

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...

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

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

Nono ... has some missiles
i.e., \( \exists x \ \text{Owns}(\text{Nono}, x) \land \text{Missile}(x) \):

\[
\exists x \ \text{Owns}(\text{Nono}, x) \land \text{Missile}(x)
\]
... it is a crime for an American to sell weapons to hostile nations:

\[ American(x) \land Weapon(y) \land Sells(x, y, z) \land Hostile(z) \implies Criminal(x) \]

Nono . . . has some missiles

i.e., \( \exists x \) Owns(Nono, x) \land Missile(x):

\( Owns(Nono, M_1) \) and \( Missile(M_1) \)
... it is a crime for an American to sell weapons to hostile nations:
\[\text{American}(x) \land \text{Weapon}(y) \land \text{Sells}(x, y, z) \land \text{Hostile}(z) \implies \text{Criminal}(x)\]

Nono ... has some missiles
i.e., \(\exists x \ \text{Owns}(\text{Nono}, x) \land \text{Missile}(x)\):
\[\text{Owns}(\text{Nono}, M_1) \ \text{and} \ \text{Missile}(M_1)\]

... all of its missiles were sold to it by Colonel West
... it is a crime for an American to sell weapons to hostile nations:
\[\text{American}(x) \land \text{Weapon}(y) \land \text{Sells}(x, y, z) \land \text{Hostile}(z) \implies \text{Criminal}(x)\]

Nono . . . has some missiles
e.g., \[\exists x \; \text{Owns}(\text{Nono}, x) \land \text{Missile}(x):\]
\[\text{Owns}(\text{Nono}, M_1) \text{ and } \text{Missile}(M_1)\]

... all of its missiles were sold to it by Colonel West
\[\forall x \; \text{Missile}(x) \land \text{Owns}(\text{Nono}, x) \implies \text{Sells}(\text{West}, x, \text{Nono})\]
... it is a crime for an American to sell weapons to hostile nations:
\[\text{American}(x) \land \text{Weapon}(y) \land \text{Sells}(x, y, z) \land \text{Hostile}(z) \implies \text{Criminal}(x)\]

Nono ... has some missiles
i.e., \(\exists x \text{ Owns}(Nono, x) \land \text{Missile}(x)\):
\[\text{Owns}(Nono, M_1) \text{ and } \text{Missile}(M_1)\]

... all of its missiles were sold to it by Colonel West
\[\forall x \text{Missile}(x) \land \text{Owns}(Nono, x) \implies \text{Sells}(West, x, Nono)\]

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

Nono ... has some missiles
i.e., \( \exists x \ \text{Owns}(\text{Nono}, x) \land \text{Missile}(x) \):
\[ \text{Owns}(\text{Nono}, M_1) \text{ and } \text{Missile}(M_1) \]

... all of its missiles were sold to it by Colonel West
\[ \forall x \ \text{Missile}(x) \land \text{Owns}(\text{Nono}, x) \implies \text{Sells}(\text{West}, x, \text{Nono}) \]

Missiles are weapons:
\[ \text{Missile}(x) \implies \text{Weapon}(x) \]
Example Knowledge Base

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

Nono ... has some missiles
i.e., \(\exists x \ \text{Owns}(\text{Nono}, x) \land \text{Missile}(x)\):
\[\text{Owns}(\text{Nono}, M_1) \land \text{Missile}(M_1)\]

... all of its missiles were sold to it by Colonel West
\[\forall x \ \text{Missile}(x) \land \text{Owns}(\text{Nono}, x) \implies \text{Sells}(\text{West}, x, \text{Nono})\]

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

An enemy of America counts as “hostile”:
it is a crime for an American to sell weapons to hostile nations:

$\text{American}(x) \land \text{Weapon}(y) \land \text{Sells}(x, y, z) \land \text{Hostile}(z) \implies \text{Criminal}(x)$

Nono . . . has some missiles

i.e., $\exists x \ \text{Owns}(\text{Nono}, x) \land \text{Missile}(x)$:

$\text{Owns}(\text{Nono}, M_1) \land \text{Missile}(M_1)$

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

$\forall x \ \text{Missile}(x) \land \text{Owns}(\text{Nono}, x) \implies \text{Sells}(\text{West}, x, \text{Nono})$

Missiles are weapons:

$\text{Missile}(x) \implies \text{Weapon}(x)$

An enemy of America counts as “hostile”:

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

Nono . . . has some missiles
i.e., \( \exists x \ \text{Owns}(\text{Nono}, x) \land \text{Missile}(x): \)
\[ \text{Owns}(\text{Nono}, M_1) \text{ and } \text{Missile}(M_1) \]

. . . all of its missiles were sold to it by Colonel West
\[ \forall x \text{Missile}(x) \land \text{Owns}(\text{Nono}, x) \implies \text{Sells}(\text{West}, x, \text{Nono}) \]

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

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

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

Nono ... has some missiles
i.e., \[ \exists x \, \text{Owns}(\text{Nono}, x) \land \text{Missile}(x) \]
\[ \text{Owns}(\text{Nono}, M_1) \text{ and } \text{Missile}(M_1) \]

... all of its missiles were sold to it by Colonel West
\[ \forall x \, \text{Missile}(x) \land \text{Owns}(\text{Nono}, x) \implies \text{Sells}(\text{West}, x, \text{Nono}) \]

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

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

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

\[ \text{American}(x) \land \text{Weapon}(y) \land \text{Sells}(x, y, z) \land \text{Hostile}(z) \implies \text{Criminal}(x) \]

Nono . . . has some missiles
i.e., \( \exists x \ \text{Owns}(\text{Nono}, x) \land \text{Missile}(x) \):

\[ \text{Owns}(\text{Nono}, M_1) \text{ and } \text{Missile}(M_1) \]

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

\[ \forall x \ \text{Missile}(x) \land \text{Owns}(\text{Nono}, x) \implies \text{Sells}(\text{West}, x, \text{Nono}) \]

Missiles are weapons:

\[ \text{Missile}(x) \implies \text{Weapon}(x) \]

An enemy of America counts as “hostile”:

\[ \text{Enemy}(x, \text{America}) \implies \text{Hostile}(x) \]

West, who is American . . .

\[ \text{American}(\text{West}) \]

The country Nono, an enemy of America . . .

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

Nono ... has some missiles
i.e., \[ \exists x \ \text{Owns}(\text{Nono}, x) \land \text{Missile}(x) : \]
\[ \text{Owns}(\text{Nono}, M_1) \land \text{Missile}(M_1) \]

... all of its missiles were sold to it by Colonel West
\[ \forall x \ \text{Missile}(x) \land \text{Owns}(\text{Nono}, x) \implies \text{Sells}(\text{West}, x, \text{Nono}) \]

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

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

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

The country Nono, an enemy of America ... 
\[ \text{Enemy}(\text{Nono}, \text{America}) \]
function FOL-FC-Ask\((KB, \alpha)\) returns a substitution or false

repeat until new is empty

new ← {} 

for each sentence \( r \) in \( KB \) do 

\( (p_1 \land \ldots \land p_n \implies q) \leftarrow \text{STANDARDIZE-APART}(r) \)

for each \( \theta \) such that \((p_1 \land \ldots \land p_n)\theta = (p'_1 \land \ldots \land p'_n)\theta\)

for some \( p'_1, \ldots, p'_n \) in \( KB \)

\( q' \leftarrow \text{SUBST}(\theta, q) \)

if \( q' \) is not a renaming of a sentence already in \( KB \) or new then do

add \( q' \) to \( new \)

\( \phi \leftarrow \text{UNIFY}(q', \alpha) \)

if \( \phi \) is not fail then return \( \phi \)

add \( new \) to \( KB \)

return false
Forward Chaining Proof

\[
\begin{align*}
\text{American(West)} & \quad \text{Missile(M1)} & \quad \text{Owns(Nono,M1)} & \quad \text{Enemy(Nono, America)} \\
\end{align*}
\]
Forward Chaining Proof

American(West) \rightarrow Missile(M1) \rightarrow Weapon(M1) \rightarrow Sells(West,M1,Nono) \rightarrow Owns(Nono,M1) \rightarrow Hostile(Nono) \rightarrow Enemy(Nono,America)
Forward Chaining Proof

The diagram represents a forward chaining proof with the following steps:

1. **Criminal(West)**
2. **Weapon(M1)**
3. **Sells(West,M1,Nono)**
4. **Hostile(Nono)**
5. **American(West)**
6. **Missile(M1)**
7. **Owns(Nono,M1)**
8. **Enemy(Nono,America)**

Each node in the diagram represents a fact or a proposition. The proof starts with the goal **Criminal(West)** and proceeds by chaining forward through the relationships represented in the diagram.
Properties of Forward Chaining

Sound and complete for first-order definite clauses
(proof similar to propositional proof)

\[ \text{Datalog} = \text{first-order definite clauses } + \text{no functions} \text{ (e.g., crime KB)} \]

FC terminates for Datalog in poly iterations: at most \( p \cdot n^k \) literals

May not terminate in general if \( \alpha \) is not entailed

This is unavoidable: entailment with definite clauses is semidecidable
Efficiency of Forward Chaining

Simple observation: no need to match a rule on iteration $k$
if a premise wasn’t added on iteration $k - 1$
$\implies$ match each rule whose premise contains a newly added literal

Matching itself can be expensive

**Database indexing** allows $O(1)$ retrieval of known facts

e.g., query $Missile(x)$ retrieves $Missile(M_1)$

Matching conjunctive premises against known facts is NP-hard

Forward chaining is widely used in *deductive databases*
Hard Matching Example

\[
\begin{align*}
\text{Diff}(\text{wa}, \text{nt}) & \land \text{Diff}(\text{wa}, \text{sa}) \land \\
\text{Diff}(\text{nt}, \text{q}) & \land \text{Diff}(\text{nt}, \text{sa}) \land \\
\text{Diff}(\text{q}, \text{nsw}) & \land \text{Diff}(\text{q}, \text{sa}) \land \\
\text{Diff}(\text{nsw}, \text{v}) & \land \text{Diff}(\text{nsw}, \text{sa}) \land \\
\text{Diff}(\text{v}, \text{sa}) & \\
\implies & \text{Colorable()} \\
\end{align*}
\]

\[
\begin{align*}
\text{Diff}(\text{Red}, \text{Blue}) & \quad \text{Diff}(\text{Red}, \text{Green}) \\
\text{Diff}(\text{Green}, \text{Red}) & \quad \text{Diff}(\text{Green}, \text{Blue}) \\
\text{Diff}(\text{Blue}, \text{Red}) & \quad \text{Diff}(\text{Blue}, \text{Green}) \\
\end{align*}
\]

\textbf{Colorable()} \text{ is inferred iff the CSP has a solution}

CSPs include 3SAT as a special case, hence matching is NP-hard
function FOL-BC-Ask(KB, goals, θ) returns a set of substitutions

inputs: KB, a knowledge base
goals, a list of conjuncts forming a query (θ already applied)
θ, the current substitution, initially the empty substitution {}

local variables: answers, a set of substitutions, initially empty

if goals is empty then return {θ}

q′ ← Subst(θ, First(goals))

for each sentence r in KB

where Standardize-Apart(r) = (p₁ ∧ ... ∧ pₙ ⇒ q) and θ′ ← Unify(q, q′) succeeds

new_goals ← [p₁, ..., pₙ | Rest(goals)]

answers ← FOL-BC-Ask(KB, new_goals, Compose(θ′, θ)) ∪ answers

return answers
Backward Chaining Example

Criminal(West)
Backward Chaining Example

Criminal(West)

{\text{x/West}}

American(x)  Weapon(y)  Sells(x,y,z)  Hostile(z)
Backward Chaining Example

```
American(West)
{ }
Weapon(y)
Sells(x,y,z)
Hostile(z)
```

```
Criminal(West)
{x/West}
```
Backward Chaining Example

![Diagram of Backward Chaining Example](image-url)
Backward Chaining Example

Criminal(West)

{ x/West, y/M1, z/Nono }

American(West)

{ }

Weapon(y)

Sells(West,M1,z)

{ z/Nono }

Hostile(z)

Missile(y)

Missile(M1)

Owns(Nono,M1)

{ y/M1 }
Backward Chaining Example

Criminal(West)

\{x/West, y/M1, z/Nono\}

American(West)

\{ \}

Weapon(y)

\{ z/Nono \}

Sells(West,M1,z)

\{ \}

Hostile(Nono)

\{ \}

Missile(y)

\{ y/M1 \}

Missile(M1)

\{ \}

Owns(Nono,M1)

\{ \}

Enemy(Nono,America)

\{ \}
Properties of Backward Chaining

Depth-first recursive proof search: space is linear in size of proof

Incomplete due to infinite loops
⇒ fix by checking current goal against every goal on stack

Inefficient due to repeated subgoals (both success and failure)
⇒ fix using caching of previous results (extra space!)

Widely used (without improvements!) for logic programming
Sound bite: computation as inference on logical KBs

<table>
<thead>
<tr>
<th>Logic programming</th>
<th>Ordinary programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Identify problem</td>
<td>Identify problem</td>
</tr>
<tr>
<td>2. Assemble information</td>
<td>Assemble information</td>
</tr>
<tr>
<td>3. Tea break</td>
<td>Figure out solution</td>
</tr>
<tr>
<td>4. Encode information in KB</td>
<td>Program solution</td>
</tr>
<tr>
<td>5. Encode problem instance as facts</td>
<td>Encode problem instance as data</td>
</tr>
<tr>
<td>6. Ask queries</td>
<td>Apply program to data</td>
</tr>
<tr>
<td>7. Find false facts</td>
<td>Debug procedural errors</td>
</tr>
</tbody>
</table>

Should be easier to debug $Capital(NewYork, US)$ than $x := x + 2$!
Basis: backward chaining with Horn clauses + bells & whistles
Widely used in Europe, Japan (basis of 5th Generation project)
Compilation techniques ⇒ approaching a billion LIPS

Program = set of clauses = head :- literal₁, ... literalₙ.

criminal(X) :- american(X), weapon(Y), sells(X,Y,Z), hostile(Z).

Efficient unification by open coding
Efficient retrieval of matching clauses by direct linking
Depth-first, left-to-right backward chaining

Built-in predicates for arithmetic etc., e.g., X is Y*Z+3
Closed-world assumption (“negation as failure”)

e.g., given alive(X) :- not dead(X).

alive(joe) succeeds if dead(joe) fails
Depth-first search from a start state X:

\[ \text{dfs}(X) :- \text{goal}(X). \]
\[ \text{dfs}(X) :- \text{successor}(X,S), \text{dfs}(S). \]

No need to loop over S: \text{successor} succeeds for each

Appending two lists to produce a third:

\[ \text{append}([],Y,Y). \]
\[ \text{append}([X|L],Y,[X|Z]) :- \text{append}(L,Y,Z). \]

query: \text{append}(A,B,[1,2]) ?

answers:
A=[] B=[1,2]
A=[1,2] B=[]
Full first-order version:

\[
\ell_1 \lor \cdots \lor \ell_k, \quad m_1 \lor \cdots \lor m_n
\]

\[
(\ell_1 \lor \cdots \lor \ell_{i-1} \lor \ell_{i+1} \lor \cdots \lor \ell_k \lor m_1 \lor \cdots \lor m_{j-1} \lor m_{j+1} \lor \cdots \lor m_n)\theta
\]

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

For example,

\[
\neg \text{Rich}(x) \lor \text{Unhappy}(x)
\]

\[
\text{Rich}(Ken) \quad \text{Unhappy}(Ken)
\]

with \( \theta = \{x/\text{Ken}\} \)

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

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

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:

\[ \neg \forall x, p \equiv \exists x \neg p, \quad \neg \exists x, p \equiv \forall x \neg p: \]

\[ \forall x [\exists y \neg (\neg \text{Animal}(y) \lor \text{Loves}(x, y))] \lor [\exists y \text{Loves}(y, x)] \]
\[ \forall x [\exists y \neg \neg \text{Animal}(y) \land \neg \text{Loves}(x, y)] \lor [\exists y \text{Loves}(y, x)] \]
\[ \forall x [\exists y \text{Animal}(y) \land \neg \text{Loves}(x, y)] \lor [\exists y \text{Loves}(y, x)] \]
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 \( \land \) over \( \lor \):

\[ [\text{Animal}(F(x)) \lor \text{Loves}(G(x), x)] \land [\neg \text{Loves}(x, F(x)) \lor \text{Loves}(G(x), x)] \]
Resolution Proof: Definite Clauses

\[ \neg \text{American}(x) \lor \neg \text{Weapon}(y) \lor \neg \text{Sells}(x,y,z) \lor \neg \text{Hostile}(z) \lor \text{Criminal}(x) \]

\[ \neg \text{Criminal}(\text{West}) \]

\[ \neg \text{American}(\text{West}) \lor \neg \text{Weapon}(y) \lor \neg \text{Sells}(\text{West},y,z) \lor \neg \text{Hostile}(z) \]

\[ \neg \text{Missile}(x) \lor \text{Weapon}(x) \]

\[ \neg \text{Missile}(y) \lor \neg \text{Sells}(\text{West},y,z) \lor \neg \text{Hostile}(z) \]

\[ \neg \text{Missile}(x) \lor \neg \text{Owns}(\text{Nono},x) \lor \text{Sells}(\text{West},x,\text{Nono}) \]

\[ \neg \text{Sells}(\text{West},\text{M1},z) \lor \neg \text{Hostile}(z) \]

\[ \neg \text{Missile}(\text{M1}) \lor \neg \text{Owns}(\text{Nono},\text{M1}) \lor \neg \text{Hostile}(\text{Nono}) \]

\[ \neg \text{Owns}(\text{Nono},\text{M1}) \lor \neg \text{Hostile}(\text{Nono}) \]

\[ \neg \text{Enemy}(x,\text{America}) \lor \text{Hostile}(x) \]

\[ \neg \text{Hostile}(\text{Nono}) \]

\[ \neg \text{Enemy}(\text{Nono},\text{America}) \]

\[ \text{Enemy}(\text{Nono},\text{America}) \]