Artificial Intelligence: Methods and Applications

Lecture 3: Review of First-Order Logic (FOL)

Juan Carlos Nieves Sánchez
November 11, 2014
Outline

- Knowledge Engineering.
- Reducing first-order inference to propositional inference.
- Unification.
- Generalized Modus Ponens
- Forward and backward chaining
- Resolution
The knowledge-engineering process

1. Identify the task (identify the questions of interest).
2. Assemble the relevant knowledge (acquisition).
3. Decide on a vocabulary of predicates, functions, and constants (the ontology).
4. Encode general knowledge about the domain (write down axioms).
5. Encode a description of the specific problem instance.
6. Pose queries to the inference procedure and get answers (use the system).
7. Debug/maintain the knowledge base.
An n-ary relation $R$ is a subset of the Cartesian product $D_1 \times \cdots \times D_n$ where each $D_i (1 \leq i \leq n)$ is a set.

For example:

$PR(x)$ – the set of numbers $x$ which are prime: \{2,3,5,7,11,…\}.

$SQ(x,y)$ – the set of pairs $(x,y)$ such that $y$ is the square of $x$: \{(1,1), (2,4), (3,9),… \}.

Can you define the relation of sum of two integer numbers?
Predicates

A relation can be formalized as a boolean-valued function on n-tuples:

Let $D$ be a set. $R$ is an n-ary relation on the domain $D$ if $R$ is a relation on $D^n$.

Let $R$ be an n-ary relation on a domain $D$. The predicate $P$ associated with $R$ is:

$$P(d_1, \ldots, d_n) = T \text{ iff } \{d_1, \ldots, d_n\} \in R$$

For example:

- $SQ(2,1) = F$
- $SQ(2,2) = F$
- $SQ(2,3) = F$
- $SQ(2,4) = T$
The syntax of FOL

<table>
<thead>
<tr>
<th>Sentence →</th>
<th>AtomicSentence</th>
<th>ComplexSentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>AtomicSentence →</td>
<td>predicate</td>
<td>predicate(Term₁,...)</td>
</tr>
<tr>
<td>ComplexSentence →</td>
<td>(Sentence)</td>
<td>[Sentence]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Term →</th>
<th>Function(Term, ...)</th>
<th>Constant</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantifier →</td>
<td>∀</td>
<td>∃</td>
<td></td>
</tr>
<tr>
<td>Constant →</td>
<td>A</td>
<td>X1</td>
<td>John</td>
</tr>
<tr>
<td>Variable →</td>
<td>A</td>
<td>x</td>
<td>s</td>
</tr>
<tr>
<td>Predicate →</td>
<td>After</td>
<td>Loves</td>
<td>..</td>
</tr>
<tr>
<td>Function →</td>
<td>LeftLeg</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
An example of a FOL Theory

We build a knowledge base for natural numbers using the following vocabulary:

- The relation name $\text{Num}$
- The constant name $0$
- The function names $+$ and $S$

Assertions:

- $\text{Num}(0)$
- $\forall x \, \text{Num}(x) \Rightarrow \text{Num}(S(x))$
- $\forall x \, S(x) \neq 0$
- $\forall x, y \, x \neq y \Rightarrow S(x) \neq S(y)$
- $\forall x \, \text{Num}(x) \Rightarrow x + 0 = x$
- $\forall x \, \text{Num}(x) \land \text{Num}(y) \Rightarrow S(x) + y = S(x + y)$
Queries

Asking our knowledge base whether:

\(? S(S(S(0))) = S(S(0)) + S(0)\)

Should now in principle yield the answer true.

By the first two assertions, all three objects involved are numbers.

By the last assertion, the equality must hold. This means that

\( S(S(S(0))) = S(S(0)) + S(0) \)

is a theorem of our system. It can be derived from our axioms.
Another example of a knowledge base

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

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

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

Nono ... has some missiles:

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

How to solve queries automatically?

An enemy of America counts as “hostile”:

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

Missiles are weapons:

\[ \forall x \, \text{Missile}(x) \Rightarrow \text{Weapon}(x) \]

West, who is American:

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

The country Nono, an enemy of America:

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

Every instantiation of a universal 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

\[
\text{King}(\text{John}) \land \text{Greedy}(\text{John}) \Rightarrow \text{Evil}(\text{John}) \\
\text{King}(\text{Richard}) \land \text{Greedy}(\text{Richard}) \Rightarrow \text{Evil}(\text{Richard}) \\
\text{King}(\text{father}(\text{John})) \land \text{Greedy}(\text{father}(\text{John})) \Rightarrow \text{Evil}(\text{father}(\text{John}))
\] ...
Existential instantiation (EI)

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

$$\exists v \alpha \quad \text{SUBST}(\{v/g,K\})$$

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

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

such that $C_1$ is a new constant symbol, called Skolen constant.

**OBSERVATIONS:**

**UI** can be applied several times to add new sentences; the new logical theory is logical equivalent to the old.

**EI** can be applied once to replace the existence sentence; the new logical theory is not logical equivalent to the old, but is satisfiable iff the old logical theory was satisfiable.
Reduction to propositional inference

Suppose the KB contains just the following:

\[ \forall x \, \text{King}(x) \land \text{Greedy}(x) \Rightarrow \text{Evil}(x) \]

\[ \text{Kind}(\text{John}) \]
\[ \text{Greedy}(\text{John}) \]
\[ \text{Brother}(\text{Richard}, \text{John}) \]

Instantiating the universal sentence in all possible ways, we have

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

\[ \text{Kind}(\text{John}) \]
\[ \text{Greedy}(\text{John}) \]
\[ \text{Brother}(\text{Richard}, \text{John}) \]

The new KB is propositionalized. Some of the proposition symbols are

\[ \text{Kind}(\text{John}), \text{Greedy}(\text{John}), \text{Evil}(\text{John}), \ldots \]
Observations of the reduction

- **Claim:** A grounded sentence is entailed by the new knowledge base iff entailed by the original knowledge base.
- **Claim:** Every FOL knowledge base can be propositionalized so as to **preserve entailment**.
- **Idea:** propositionalize knowledge base and query, **apply resolution**, return result.
- **Problem:** with functions symbols, there are infinitely many grounded terms.

- **Theorem:** Helbrad (1930), If a sentence \( \alpha \) is entailed by an FOL knowledge base, it is entailed by a **finite subset** of the propositional knowledge base.
- **Idea:** For \( n = 0 \) to \( \infty \) do
  - create a propositional knowledge base by instatiating with depth-\( n \) terms
  - see if \( \alpha \) is entailed by this propositional knowledge base
- **Problem:** works if \( \alpha \) is entailed, loops if \( \alpha \) is not entailed
- **Theorem:** Turing (1936), Church(1936), entailment if FOL is **semidecidable**
Problems with propositionalization

Propositionalization seems to generate lots of irrelevant sentences. E.g., from

\[ \forall x \ King(x) \land Greedy \Rightarrow Evil(x) \]
\[ \begin{align*}
  & King(John) \\
  & \forall y \ Greedy(y) \\
  & Brother(Richard, John)
\end{align*} \]

it seems obvious that \textit{Evil(John)}, but propositionalization produces lots of facts such as \textit{Greedy(Richard)} that are irrelevant.

With \( p \) \( k \)-any predicates and \( n \) constants, there are \( p \ n^k \) instantiations.

With function symbols, it gets much much worse!
Unification

We can get the inference immediately if we can find a substitution $\theta$ such that $King(x)$ and $Gready(x)$ match $King(John)$ and $Gready(y)$

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

\[
Unify(\alpha, \beta) \text{ if } \alpha\theta = \beta\theta
\]

<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>${x/Jane}$</td>
</tr>
<tr>
<td>$Knows(John, x)$</td>
<td>$Knows(y, OJ)$</td>
<td>${x/OJ, \ y/John}$</td>
</tr>
<tr>
<td>$Knows(John, x)$</td>
<td>$Knows(y, Mother(y))$</td>
<td>${x/Mother(John), \ y/John}$</td>
</tr>
<tr>
<td>$Knows(John, x)$</td>
<td>$Knows(x, OJ)$</td>
<td>fail</td>
</tr>
</tbody>
</table>
Generalized Modus Ponens (GMP)

\[ p'_1, p'_2, \ldots, p'_n, (p_1 \land p_2 \land \ldots \land p_n \Rightarrow q) \]

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

\( p'_1 \) is \( \text{King}(John) \)  \( p_1 \) is \( \text{King}(x) \)
\( p'_2 \) is \( \text{Greedy}(y) \)  \( p_2 \) is \( \text{Greedy}(x) \)
\( \theta \) is \( \{x/John, \ y/John\} \)  \( q \) is \( \text{Evil}(x) \)
\( q\theta \) is \( \text{Evil}(John) \)

GMP used with knowledge bases of definite clauses (exactly one positive literal).

All the variables assumed universally quantified
Example knowledge base

The law says that is a crime for an American to sell weapons to a 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 Est 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: \( \exists x \text{Owns}(\text{Nono}, x) \land \text{Missile}(x) \). If we apply Skolemization, we get \( \text{Owns}(\text{Nono}, M1) \land \text{Missile}(M1) \).

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

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

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

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

West, who is American: \( \text{American}(\text{West}) \)

The country Nono, an enemy of America: \( \text{Enemy}(\text{Nono, America}) \)
Forward chaining inference

Intentional knowledge base (rules):

- $\text{American}(x) \land \text{Weapon}(y) \land \text{Sells}(x, y, z) \land \text{Hostile}(z) \Rightarrow \text{Criminal}(x)$
- $\text{Missile}(x) \land \text{Owms}(\text{Nono}, x) \Rightarrow \text{Sells}(\text{West}, x, \text{Nono})$
- $\text{Enemigy}(x, \text{America}) \Rightarrow \text{Hostile}(x)$
- $\text{Missile}(x) \Rightarrow \text{Weapon}(x)$

Extentional knowledge base (facts):

- $\text{American}(\text{West})$
- $\text{Missile}(M1)$
- $\text{Owns}(\text{Nono}, M1)$
- $\text{Enemy}(\text{Nono}, \text{America})$

Criminal(West)

Weapon(M1)  Sells(West, M1, Nono)  Hostile(Nono)
Properties of forward chaining

- Sound and complete for first-order definite clauses (proof similar to propositional proof).
- Datalog = first-order definite clauses + no function. Forward chaining terminates for Datalog in Poly interactions.
- May not terminate in general if $\alpha$ is not entailed.
- Forward chaining is widely used in deductive databases.
- Matching conjunctive premises against know facts is NP-hard.

How can we avoid matching between conjunctive premises and know facts?

Backward chaining inference
Backward chaining inference

Review of First-Order Logic
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 by classical logic programming interpreters!
Resolution: a short review

\[
\begin{array}{c}
l_1 \lor \cdots \lor l_k \\
\hline
\hline
l_1 \lor \cdots \lor l_{i-1} \lor l_{i+1} \lor \cdots l_k \lor m_1 \lor \cdots \lor m_{j-1} \lor m_{j+1} \lor \cdots \lor m_n \theta
\end{array}
\]

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

For example:

\[
\begin{array}{c}
\neg \text{Rich}(x) \lor \text{Unhappy}(x) \\
\hline
\text{Rich(Ken)}
\end{array}
\]

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

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

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

Intentional knowledge base (rules):

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

- \( \text{Missile}(x) \land \text{Owms}(\text{Nono}, x) \Rightarrow \text{Sells}(\text{West}, x, \text{Nono}) \)
  \[ \neg \text{Missile}(x) \lor \neg \text{Owms}(\text{Nono}, x) \lor \text{Sells}(\text{West}, x, \text{Nono}) \]

- \( \text{Enemigy}(x, \text{America}) \Rightarrow \text{Hostile}(x) \)
  \[ \neg \text{Enemigy}(x, \text{America}) \lor \text{Hostile}(x) \]

- \( \text{Missile}(x) \Rightarrow \text{Weapon}(x) \)
  \[ \neg \text{Missile}(x) \lor \text{Weapon}(x) \]
Resolution example

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

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

\[\neg Missile(x) \lor Weapon(x)\]

\[\neg Missile(West)\]

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

\[\neg Missle(Nono,MI) \lor \neg Owns(Nono,MI) \lor \neg Hostile(Nono)\]

\[\neg Owns(Nono,MI) \lor \neg Hostile(Nono)\]

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

\[\neg Hostile(Nono)\]

\[\neg Enemy(Nono,America)\]

\[\neg Enemy(Nono,America)\]
Observations

Given a logical theory $T$ and a formula $\alpha$, if $T$ entails $\alpha$, then $T \land \neg \alpha$ is unsatisfiable.

If a set of sentences is unsatisfiable, then resolution will always be able to derive a contradiction.

- Resolution is one of the most powerful tools for implementing automated reasoning.
- There are several extensions of the resolution rule in order to implement inference in other non-classical logics, e.g. Possibilistic Logic.
Sources of this Lecture

• Some of these slides are based on the slides of the book: Artificial Intelligence: A Modern Approach, by Stuart Russell and Peter Norvig.