CSE 473: First Order Logic

Luke Zettlemoyer

(With many slides from Dan Weld, Raj Rao, Mausam, Stuart Russell, Dieter Fox, Henry Kautz, Min-Yen Kan...)
Outline

• First-Order Logic
  – Definitions
  – Universal and Existential Quantifiers
  – Skolemization
  – Unification
  – Chaining and Resolution
Pros and Cons of Propositional Logic

• Propositional logic is *declarative*: pieces of syntax correspond to facts
• Propositional logic allows *partial/disjunctive/negated* information (unlike most data structures and databases)
• Propositional logic is *compositional*:
  – meaning of $B_{1,1} \land P_{1,2}$ derived from meanings of $B_{1,1}$ and $P_{1,2}$
• Propositional logic has very limited expressive power (unlike natural language)
  – E.g., cannot say “pits cause breezes in adjacent squares” except by writing one sentence for each square
Propositional logic: Deals with facts and propositions (can be true or false):

- $P_{1,1}$ -- “there is a pit in (1,1)”
- George_Monkey -- “George is a monkey”
- George_Curious -- “George is curious”
- 473student1_curious -- “student 1 is a curious”
- $(\text{George}_\text{Monkey} \land \neg 473\text{student1}_\text{Monkey}) \lor \ldots$
FOL Definitions

**Constants:** Name a specific object.

George, Monkey2, Larry, Luke ...

**Variables:** Refer to an object without naming it.

X, Y, ...

**Relations (predicates):** Properties of or relationships between objects.

Curious(.), PokesInTheEyes(.,.), SmarterThan(.,.)...

**Functions:** Mapping from objects to objects.

banana-of(.), grade-of(.), child-of(.,.)
Syntax of First Order Logic

Constants  \( \text{KingJohn}, 2, \text{UCB}, \ldots \)
Predicates  \( \text{Brother}, >, \ldots \)
Functions  \( \text{Sqrt, LeftLegOf, \ldots} \)
Variables  \( x, y, a, b, \ldots \)
Connectives  \( \land, \lor, \neg, \Rightarrow, \Leftarrow \)
Equality  \( = \)
Quantifiers  \( \forall, \exists \)

Atomic sentence  \( = \) \( \text{predicate(term}_1, \ldots, \text{term}_n) \)
or \( \text{term}_1 = \text{term}_2 \)

Term  \( = \) \( \text{function(term}_1, \ldots, \text{term}_n) \)
or \( \text{constant or variable} \)

Atomic Sentences:
E.g.,  \( \text{Brother(\text{KingJohn, RichardTheLionheart})} \)
\( > (\text{Length(LeftLegOf(Richard)), Length(LeftLegOf(\text{KingJohn})})) \)

Complex Sentences:
E.g.  \( \text{Sibling(\text{KingJohn, Richard})} \Rightarrow \text{Sibling(\text{Richard, KingJohn})} \)
\( > (1, 2) \lor \leq(1, 2) \)
\( > (1, 2) \land \neg > (1, 2) \)
Wumpus World

- **Performance measure**
  - Gold: +1000, death: -1000
  - -1 per step, -10 for using the arrow

- **Environment**
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to pit are breezy
  - Glitter iff gold is in the same square
  - Shooting kills wumpus if you are facing it
  - Shooting uses up the only arrow
  - Grabbing picks up gold if in same square
  - Releasing drops the gold in same square

- **Sensors:** Stench, Breeze, Glitter, Bump, Scream
- **Actuators:** Left turn, Right turn, Forward, Grab, Release, Shoot
Wumpus World

Properties of locations:
\[
\forall x, t \quad \text{At}(Agent, x, t) \land \text{Smelt}(t) \Rightarrow \text{Smelly}(x)
\]
\[
\forall x, t \quad \text{At}(Agent, x, t) \land \text{Breeze}(t) \Rightarrow \text{Breezy}(x)
\]

Diagnostic rule—infer cause from effect
\[
\forall y \quad \text{Breezy}(y) \Rightarrow \exists x \quad \text{Pit}(x) \land \text{Adjacent}(x, y)
\]

Causal rule—infer effect from cause
\[
\forall x, y \quad \text{Pit}(x) \land \text{Adjacent}(x, y) \Rightarrow \text{Breezy}(y)
\]

Neither of these is complete—e.g., the causal rule doesn’t say whether squares far away from pits can be breezy

Definition for the \textit{Breezy} predicate:
\[
\forall y \quad \text{Breezy}(y) \Leftrightarrow \left[ \exists x \quad \text{Pit}(x) \land \text{Adjacent}(x, y) \right]
\]
First Order Models

Sentences are true with respect to a model and an interpretation.

Model contains $\geq 1$ objects (domain elements) and relations among them.

Interpretation specifies referents for:
- constant symbols $\rightarrow$ objects
- predicate symbols $\rightarrow$ relations
- function symbols $\rightarrow$ functional relations

An atomic sentence $\text{predicate}(\text{term}_1, \ldots, \text{term}_n)$ is true iff the objects referred to by $\text{term}_1, \ldots, \text{term}_n$ are in the relation referred to by $\text{predicate}$.
Example: A World of Kings and Legs

• Syntactic elements:

  Constants: Richard, John, RsLeftLeg, ...

  Functions: leftleg(.), onheadof(.), ...

  Relations: On(.,.) IsKing(.), IsPerson(.), ...
All Possible Models

We **can** enumerate the FOL models for a given KB vocabulary:

For each number of domain elements $n$ from 1 to $\infty$
  - For each $k$-ary predicate $P_k$ in the vocabulary
    - For each possible $k$-ary relation on $n$ objects
      - For each constant symbol $C$ in the vocabulary
        - For each choice of referent for $C$ from $n$ objects . . .

**Lesson:** Computing entailment by enumerating models will be challenging!
More Definitions

• Logical connectives: and, or, not, ⇒, ⇔

• Quantifiers:
  – ∀ For all (Universal quantifier)
  – ∃ There exists (Existential quantifier)

• Examples
  – George is a monkey and he is curious
    \[ \text{Monkey}(\text{George}) \land \text{Curious}(\text{George}) \]
  – All monkeys are curious
    \[ \forall m: \text{Monkey}(m) \Rightarrow \text{Curious}(m) \]
  – There is a curious monkey
    \[ \exists m: \text{Monkey}(m) \land \text{Curious}(m) \]
Quantifier / Connective Interaction

\( \forall x: \ M(x) \land C(x) \)

\( M(x) \equiv \text{“x is a monkey”} \)

\( C(x) \equiv \text{“x is curious”} \)

“Everything is a curious monkey”

\( \forall x: \ M(x) \implies C(x) \)

“All monkeys are curious”

\( \exists x: \ M(x) \land C(x) \)

“There exists a curious monkey”

\( \exists x: \ M(x) \implies C(x) \)

“There exists an object that is either a curious monkey, or not a monkey at all”
Nested Quantifiers: Order matters!

\[ \forall x \exists y \ P(x,y) \neq \exists y \ \forall x \ P(x,y) \]

• Example

Every monkey has a tail

\[ \forall m \ \exists t \ \text{has}(m,t) \]

Every monkey shares a tail!

\[ \exists t \ \forall m \ \text{has}(m,t) \]

Try:

Everybody loves somebody vs. Someone is loved by everyone

\[ \forall x \ \exists y \ \text{loves}(x, y) \]

\[ \exists y \ \forall x \ \text{loves}(x, y) \]
Fun With Sentences

• Brothers are siblings.
  \[ \forall x, y \; \text{Brother}(x, y) \Rightarrow \text{Sibling}(x, y) \].

• “Sibling” is symmetric.
  \[ \forall x, y \; \text{Sibling}(x, y) \iff \text{Sibling}(y, x) \].

• One’s mother is one’s female parent.
  \[ \forall x, y \; \text{Mother}(x, y) \iff (\text{Female}(x) \land \text{Parent}(x, y)) \].

• A first cousin is a child of a parent’s sibling.
  \[ \forall x, y \; \text{FirstCousin}(x, y) \iff \exists p, ps \; \text{Parent}(p, x) \land \text{Sibling}(ps, p) \land \text{Parent}(ps, y) \].
## Propositional Logic vs. First Order Logic

<table>
<thead>
<tr>
<th><strong>Ontology</strong></th>
<th>Facts (P, Q,…)</th>
<th>Objects, Properties, Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Syntax</strong></td>
<td>Atomic sentences Connectives</td>
<td>Variables &amp; quantification Sentences have structure: terms father-of(mother-of(X))</td>
</tr>
<tr>
<td><strong>Semantics</strong></td>
<td>Truth Tables</td>
<td>Interpretations &amp; Models (Much more complicated)</td>
</tr>
<tr>
<td><strong>Inference Algorithm</strong></td>
<td>DPLL, WalkSAT Fast in practice</td>
<td>Unification Forward, Backward chaining Prolog, theorem proving</td>
</tr>
<tr>
<td><strong>Complexity</strong></td>
<td>NP-Complete</td>
<td>Semi-decidable May run forever if $\text{KB} \not\models \alpha$</td>
</tr>
</tbody>
</table>
FOL Reasoning: Outline

- Basics of FOL reasoning
- Classes of FOL reasoning methods
  - Compilation to propositional logic
  - Forward & Backward Chaining
  - Resolution
# FOL Reasoning: Brief History

<table>
<thead>
<tr>
<th>Year</th>
<th>Person</th>
<th>Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>450 B.C.</td>
<td>Stoics</td>
<td>propositional logic, inference (maybe)</td>
</tr>
<tr>
<td>322 B.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>
Basics: Universal Instantiation

• Universally quantified sentence:
  – \( \forall x: \text{Monkey}(x) \Rightarrow \text{Curious}(x) \)

• Intuitively, \( x \) can be anything:
  – \( \text{Monkey}(\text{George}) \Rightarrow \text{Curious}(\text{George}) \)
  – \( \text{Monkey}(\text{473Student1}) \Rightarrow \text{Curious}(\text{473Student1}) \)
  – \( \text{Monkey}(\text{DadOf}(\text{George})) \Rightarrow \text{Curious}(\text{DadOf}(\text{George})) \)

• Formally:

  \[
  \begin{array}{c}
  \forall x \ S \\
  \underline{\text{Subst}\{\{x/p\}, S\}} \\
  \end{array}
  \]

  \[
  \begin{array}{c}
  \forall x \ \text{Monkey}(x) \Rightarrow \text{Curious}(x) \\
  \underline{\text{Subst}\{\{x/p\}, S\}} \\
  \text{Monkey}(\text{George}) \Rightarrow \text{Curious}(\text{George}) \\
  \end{array}
  \]

  \( x \) is replaced with \( p \) in \( S \), and the quantifier removed
  \( x \) is replaced with George in \( S \), and the quantifier removed
Basics: Existential Instantiation

• Existentially quantified sentence:
  \( \exists x: \text{Monkey}(x) \land \lnot \text{Curious}(x) \)

• Can we conclude:
  \( \text{Monkey}(\text{George}) \land \lnot \text{Curious}(\text{George}) \) ???
  No! \( S \) might not be true for George!

• Use a \textit{Skolem Constant} and draw the conclusion:
  \( \text{Monkey}(K) \land \lnot \text{Curious}(K) \)

• Formally:
  \[
  \begin{align*}
  \exists x & \ S \\
  \text{Subst}\{{x/K}, S\} \\
  K & \text{ is called a Skolem constant}
  \end{align*}
  \]

• Existential instantiation changes the KB, but still entails the same set of formulas!
Reduction to Propositional Inference

Suppose the KB contains just the following:

\[
\forall x \ King(x) \land Greedy(x) \Rightarrow Evil(x)
\]

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 propositionally: proposition symbols are

King(John), Greedy(John), Evil(John), King(Richard) etc.
Reduction to Propositional Inference

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., $Father(Father(Father(John)))$

Theorem: Herbrand (1930). If a sentence $\alpha$ is entailed by an FOL KB,
   it is entailed by a \textbf{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 \textbf{semidecidable}
Problems with Propositionalization

Propositionalization seems to generate lots of irrelevant sentences. E.g., 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 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!
Motivation for Unification

• What if we want to use modus ponens?
  Propositional Logic:
  \[
  \begin{align*}
  a \land b, & \quad a \land b \Rightarrow c \\
  \hline
  \Rightarrow & \\
  c
  \end{align*}
  \]

• In First-Order Logic?
  \[
  \forall x \text{ Monkey}(x) \Rightarrow \text{Curious}(x) \quad \text{Monkey}(\text{George})
  \]

• Must "unify" \( x \) with George:
  Need to substitute \( \{x/\text{George}\} \) in \( \text{Monkey}(x) \)
  \( \Rightarrow \text{Curious}(x) \) to infer \( \text{Curious}(\text{George}) \)
Unification Examples

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}(\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(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)

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

\[ \frac{q}{q^\theta} \]

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

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

GMP used with KB of definite clauses (\textbf{exactly} one positive literal)
All variables assumed universally quantified
Knowledge Base Example

• Knowledge: 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.

• Goal: Prove that Col. West is a criminal – *Criminal(West)*

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

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

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

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

\[
\text{Missile}(x) \Rightarrow \text{Weapon}(x) \quad \text{Enemy}(x, \text{America}) \Rightarrow \text{Hostile}(x)
\]
American(x) ∧ Weapon(y) ∧ Sells(x, y, z) ∧ Hostile(z) ⇒ Criminal(x)

Owns(Nono, M1)  Missile(M1)

Missile(x) ∧ Owns(Nono, x) ⇒ Sells(West, x, Nono)

American(West)  Enemy(Nono, America)

Missile(x) ⇒ Weapon(x)  Enemy(x, America) ⇒ Hostile(x)
First-order Resolution

Full first-order version:

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

\[
\frac{(\ell_1 \lor \cdots \lor \ell_{i-1} \lor \ell_{i+1} \lor \cdots \lor \ell_k \lor m_1 \lor \cdots \lor m_{j-1} \lor m_{j+1} \lor \cdots \lor m_n)\theta}{Unify(\ell_i, \neg m_j) = \theta}
\]

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

For example,

\[
\neg Rich(x) \lor Unhappy(x) \\
Rich(Ken)
\]

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

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

Apply resolution steps to \( CNF(KB \land \neg \alpha) \); complete for FOL
Conversion to CNF (Part 1)

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

1. Eliminate biconditionals and implications

\[
\forall x \ [\neg \forall y \ \neg Animal(y) \lor Loves(x, y)] \lor [\exists y \ Loves(y, x)]
\]

2. Move \( \neg \) inwards: \( \neg \forall x, p \equiv \exists x \ \neg p \), \( \neg \exists x, p \equiv \forall x \ \neg p \):

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

\[ \forall x \ [\exists y \ Animal(y) \land \neg Loves(x, y)] \lor [\exists z \ Loves(z, x)] \]

4. Skolemize: a more general form of existential instantiation. Each existential variable is replaced by a Skolem function of the enclosing universally quantified variables:

\[ \forall x \ [Animal(F(x)) \land \neg Loves(x, F(x))] \lor Loves(G(x), x) \]

5. Drop universal quantifiers:

\[ [Animal(F(x)) \land \neg Loves(x, F(x))] \lor Loves(G(x), x) \]

6. Distribute \( \land \) over \( \lor \):

\[ [Animal(F(x)) \lor Loves(G(x), x)] \land [\neg Loves(x, F(x)) \lor Loves(G(x), x)] \]
A Resolution Proof

¬ American(x) ∨ ¬ Weapon(y) ∨ ¬ Sells(x,y,z) ∨ ¬ Hostile(z) ∨ Criminal(x)
¬ Criminal(West)

¬ American(West) ∨ ¬ Weapon(y) ∨ ¬ Sells(West,y,z) ∨ ¬ Hostile(z)

¬ Missle(x) ∨ Weapon(x)
¬ Missle(y) ∨ ¬ Sells(West,y,z) ∨ ¬ Hostile(z)

¬ Missle(x) ∨ ¬ Owns(Nono,x) ∨ Sells(West,x,Nono)
¬ Missle(M1) ∨ ¬ Sells(West,M1,z) ∨ ¬ Hostile(z)

¬ Missle(M1) ∨ ¬ Owns(Nono,M1) ∨ ¬ Hostile(Nono)
¬ Owns(Nono,M1) ∨ ¬ Hostile(Nono)

¬ Enemy(x,America) ∨ Hostile(x)
¬ Enemy(Nono,America)
¬ Hostile(Nono)