First-order logic

Chapter 8
Outline

◊ Why FOL?
◊ Syntax and semantics of FOL
◊ Fun with sentences
◊ Wumpus world in FOL
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 $B_{1,1} \land P_{1,2}$ is derived from meaning of $B_{1,1}$ and of $P_{1,2}$

Meaning in propositional logic is **context-independent** (unlike natural language, where meaning depends on context)

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
First-order logic

Whereas propositional logic assumes world contains facts, first-order logic (like natural language) assumes the world contains

- **Objects**: people, houses, numbers, theories, Ronald McDonald, colors, baseball games, wars, centuries . . .
- **Relations**: red, round, bogus, prime, multistoried . . ., brother of, bigger than, inside, part of, has color, occurred after, owns, comes between, . . .
- **Functions**: father of, best friend, third inning of, one more than, end of . . .
## Logics in general

<table>
<thead>
<tr>
<th>Language</th>
<th>Ontological Commitment</th>
<th>Epistemological Commitment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propositional logic</td>
<td>facts</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>First-order logic</td>
<td>facts, objects, relations</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>Temporal logic</td>
<td>facts, objects, relations, times</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>Probability theory</td>
<td>facts</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>Fuzzy logic</td>
<td>facts + degree of truth</td>
<td>known interval value</td>
</tr>
<tr>
<td>Syntax of FOL: Basic elements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Constants</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>KingJohn</em>, 2, <em>UCB</em>,…</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Predicates</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Brother</em>, &gt;, …</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Functions</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Sqrt</em>, <em>LeftLegOf</em>, …</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Variables</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>x</em>, <em>y</em>, <em>a</em>, <em>b</em>,…</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Connectives</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>∧, ∨, ¬, ⇒, ⇔</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Equality</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>=</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Quantifiers</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>∀, ∃</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Atomic sentences

Atomic sentence  =  \textit{predicate}(\textit{term}_1, \ldots, \textit{term}_n)

or  \textit{term}_1 = \textit{term}_2

Term  =  \textit{function}(\textit{term}_1, \ldots, \textit{term}_n)

or  constant or variable

E.g.,  \textit{Brother}(\textit{King John}, \textit{Richard The Lionheart})

> (\textit{Length}(\textit{Left Leg Of}(\textit{Richard})), \textit{Length}(\textit{Left Leg Of}(\textit{King John})))
Complex sentences

Complex sentences are made from atomic sentences using connectives

\[ \neg S, \quad S_1 \land S_2, \quad S_1 \lor S_2, \quad S_1 \Rightarrow S_2, \quad S_1 \Leftrightarrow S_2 \]

E.g.  \( Sibling(King\text{John}, Richard) \Rightarrow Sibling(Richard, King\text{John}) \)
\[ >(1, 2) \lor \leq (1, 2) \]
\[ >(1, 2) \land \neg>(1, 2) \]
Truth in first-order logic

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 $\text{predicate}(term_1, \ldots, term_n)$ is true iff the objects referred to by $term_1, \ldots, term_n$ are in the relation referred to by $\text{predicate}$.
Models for FOL: Example
Consider the interpretation in which

Richard → Richard the Lionheart
John → the evil King John
Brother → the brotherhood relation

Under this interpretation, \textit{Brother}(Richard, John) is true just in case Richard the Lionheart and the evil King John are in the brotherhood relation in the model.
Models for FOL: Lots!

Entailment in propositional logic can be computed by enumerating 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 . . .

Computing entailment by enumerating FOL models is not easy!
Universal quantification

∀ (variables) (sentence)

Everyone at Berkeley is smart:
∀ x  At(x, Berkeley) ⇒ Smart(x)

∀ x  P  is true in a model m  iff  P  is true with x being each possible object in the model

Roughly speaking, equivalent to the conjunction of instantiations of P

(At(KingJohn, Berkeley) ⇒ Smart(KingJohn))
∧ (At(Richard, Berkeley) ⇒ Smart(Richard))
∧ (At(Berkeley, Berkeley) ⇒ Smart(Berkeley))
∧  ...
A common mistake to avoid

Typically, $\Rightarrow$ is the main connective with $\forall$

Common mistake: using $\land$ as the main connective with $\forall$:

$$\forall x \ At(x, \text{Berkeley}) \land Smart(x)$$

means “Everyone is at Berkeley and everyone is smart”
**Existential quantification**

\[ \exists \langle \text{variables} \rangle \langle \text{sentence} \rangle \]

Someone at Stanford is smart:
\[ \exists x \ At(x, \text{Stanford}) \land \text{Smart}(x) \]

\[ \exists x \ P \quad \text{is true in a model } m \quad \text{iff } \ P \quad \text{is true with } x \quad \text{being some possible object in the model} \]

**Roughly** speaking, equivalent to the disjunction of instantiations of \( P \)

\[
(At(King\,John, \text{Stanford}) \land \text{Smart}(King\,John))
\lor (At(Richard, \text{Stanford}) \land \text{Smart}(Richard))
\lor (At(Stanford, \text{Stanford}) \land \text{Smart}(Stanford))
\lor \ldots
\]
Another common mistake to avoid

Typically, \( \land \) is the main connective with \( \exists \).

Common mistake: using \( \Rightarrow \) as the main connective with \( \exists \):

\[ \exists x \, \text{At}(x, \text{Stanford}) \Rightarrow \text{Smart}(x) \]

is true if there is anyone who is not at Stanford!
Properties of quantifiers

\( \forall x \ \forall y \) is the same as \( \forall y \ \forall x \) (why??)

\( \exists x \ \exists y \) is the same as \( \exists y \ \exists x \) (why??)

\( \exists x \ \forall y \) is not the same as \( \forall y \ \exists x \)

\( \exists x \ \forall y \ Loves(x, y) \)
“There is a person who loves everyone in the world”

\( \forall y \ \exists x \ Loves(x, y) \)
“Everyone in the world is loved by at least one person”

Quantifier duality: each can be expressed using the other

\( \forall x \ Likes(x, IceCream) \quad \neg \exists x \ \neg Likes(x, IceCream) \)

\( \exists x \ Likes(x, Broccoli) \quad \neg \forall x \ \neg Likes(x, Broccoli) \)
Fun with sentences

Brothers are siblings
Fun with sentences

Brothers are siblings

\( \forall x, y \  Brother(x, y) \Rightarrow Sibling(x, y). \)

“Sibling” is symmetric
Fun with sentences

Brothers are siblings

∀ x, y Brother(x, y) ⇒ Sibling(x, y).

“Sibling” is symmetric

∀ x, y Sibling(x, y) ⇔ Sibling(y, x).

One’s mother is one’s female parent
Fun with sentences

Brothers are siblings

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

“Sibling” is symmetric

\[ \forall x, y \; \text{Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x). \]

One’s mother is one’s female parent

\[ \forall x, y \; \text{Mother}(x, y) \Leftrightarrow (\text{Female}(x) \land \text{Parent}(x, y)). \]

A first cousin is a child of a parent’s sibling
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

\[ \forall x, y \; \text{FirstCousin}(x, y) \iff \exists p, ps \; \text{Parent}(p, x) \land \text{Sibling}(ps, p) \land \text{Parent}(ps, y) \]
Equality

\( term_1 = term_2 \) is true under a given interpretation if and only if \( term_1 \) and \( term_2 \) refer to the same object.

E.g., \( 1 = 2 \) and \( \forall x \ (Sqrt(x), Sqrt(x)) = x \) are satisfiable.
\( 2 = 2 \) is valid.

E.g., definition of (full) \textit{Sibling} in terms of \textit{Parent}:
\[
\forall x, y \ Sibling(x, y) \iff [\neg (x = y) \land \exists m, f \neg (m = f) \land \\
Parent(m, x) \land Parent(f, x) \land Parent(m, y) \land Parent(f, y)]
\]
Suppose a wumpus-world agent is using an FOL KB and perceives a smell and a breeze (but no glitter) at $t = 5$:

$\text{Tell}(KB, \text{Percept}([\text{Smell}, \text{Breeze}, \text{None}], 5))$
$\text{Ask}(KB, \exists a \ \text{Action}(a, 5))$

I.e., does $KB$ entail any particular actions at $t = 5$?

Answer: Yes, $\{a/\text{Shoot}\}$ ← substitution (binding list)

Given a sentence $S$ and a substitution $\sigma$,
$S\sigma$ denotes the result of plugging $\sigma$ into $S$; e.g.,
$S = \text{Smarter}(x, y)$
$\sigma = \{x/\text{Hillary}, y/\text{Bill}\}$
$S\sigma = \text{Smarter}(\text{Hillary}, \text{Bill})$

$\text{Ask}(KB, S)$ returns some/all $\sigma$ such that $KB \models S\sigma$