First-Order Logic

Chapter 8
Outline

• Why FOL?
• Syntax and semantics of FOL
• Using FOL
• Wumpus world in FOL
• Knowledge engineering in FOL
Pros and cons of propositional logic

😊 Propositional logic is declarative

😊 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 the world contains facts,

• first-order logic (like natural language) assumes the world contains
  – **Objects**: people, houses, numbers, colors, baseball games, wars, …
  – **Relations**: red, round, prime, brother of, bigger than, part of, comes between, …
  – **Functions**: father of, best friend, one more than, plus, …
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>degree of belief</td>
</tr>
<tr>
<td>Fuzzy logic</td>
<td>facts + degree of truth</td>
<td>known interval value</td>
</tr>
</tbody>
</table>

- **Ontology:**
  - science or study of being, which problems are being investigated, the kinds of abstract entities that are to be admitted to a language system, things that exist in a system.

- **Epistemology:**
  - study of methods and grounds of knowledge, its limits, and its validity, theories of knowledge.
Syntax of FOL: Basic elements

• Constants  KingJohn, 2, NUS,...
• Predicates  Brother, >,...
• Functions  Sqrt, LeftLegOf,...
• Variables  x, y, a, b,...
• Connectives  \( \neg, \Rightarrow, \land, \lor, \equiv \)
• Equality  =
• Quantifiers  \( \forall, \exists \)
Atomic sentences

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

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

• E.g., \( \text{Brother(KingJohn,RichardTheLionheart)} > (\text{Length(LeftLegOf(Richard))}, \text{Length(LeftLegOf(KingJohn))}) \)
Complex sentences

• Complex sentences are made from atomic sentences using connectives

  \neg S, \ S_1 \land S_2, \ S_1 \lor S_2, \ S_1 \Rightarrow S_2, \ S_1 \Leftrightarrow S_2,

E.g. \textit{Sibling(KingJohn,Richard)} \Rightarrow \textit{Sibling(Richard,KingJohn)}

\>(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 ≥ 1 objects (domain elements) and relations among them.

- Interpretation specifies referents for:
  - Constant symbols → objects
  - Predicate symbols → relations
  - Function symbols → 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 $\text{predicate}$.
Models for FOL: Example

crown on head

person

left leg

brother

king

left leg
Truth Example

• Consider the interpretation in which:
  – Richard $\rightarrow$ Richard the Lionheart
  – John $\rightarrow$ the evil King John
  – Brother $\rightarrow$ the brotherhood relation

• Under this interpretation, Brother(Richard,John) is true when Richard the Lionheart & the evil King John are in the brotherhood relation in the model.
Models for FOL: Many!

• 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 $P_k$ relation on $n$ objects
      • For each constant symbol $C$ in vocabulary
        – For each choice of referent for $C$ from $n$ objects
          » ...  
  • Computing entailment by enumerating FOL models is not easy!
Universal quantification

• $\forall<variables> <sentence>$

Everyone at UWEE is smart:
$\forall x \text{At}(x,\text{UWEE}) \Rightarrow \text{Smart}(x)$

• $\forall x P$ 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 $P$
  
  $\text{At}(\text{KingJohn},\text{UWEE}) \Rightarrow \text{Smart}(\text{KingJohn})$
  $\wedge \text{At}(\text{Richard},\text{UWEE}) \Rightarrow \text{Smart}(\text{Richard})$
  $\wedge \text{At}(\text{UWEE},\text{UWEE}) \Rightarrow \text{Smart}(\text{UWEE})$
  $\wedge ...$
A common mistake to avoid

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

• Common mistake: using $\land$ as the main connective with $\forall$:
  
  $\forall x \text{ At}(x, \text{UWEE}) \land \text{Smart}(x)$

  means “Everyone is at NUS, and everyone is smart”
Existential quantification

• \( \exists \langle \text{variables} \rangle \ <\text{sentence}> \)

• Someone at WSU is smart:
  – \( \exists x \ \text{At}(x,\text{WSU}) \land \text{Smart}(x) \)

• \( \exists x \ \text{P} \) is true in a model \( m \) iff \( P \) is true with \( x \) being some possible object in the model

• Roughly speaking, equivalent to the disjunction of instantiations of \( P \)
  \begin{align*}
    & \text{At}(\text{KingJohn},\text{WSU}) \land \text{Smart}(\text{KingJohn}) \\
    & \lor \text{At}(\text{Richard},\text{WSU}) \land \text{Smart}(\text{Richard}) \\
    & \lor \text{At}(\text{WSU},\text{WSU}) \land \text{Smart}(\text{WSU}) \\
    & \lor \ldots
  \end{align*}
Another common mistake to avoid

• Typically, $\land$ is the main connective with $\exists$

• Common mistake: using $\Rightarrow$ as the main connective with $\exists$:

$$\exists x \ \text{At}(x, \text{WSU}) \Rightarrow \text{Smart}(x)$$

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

• ∀x ∀y is the same as ∀y ∀x
• ∃x ∃y is the same as ∃y ∃x

• ∃x ∀y is not the same as ∀y ∃x
• ∃x ∀y Loves(x,y)
  – “There is a person who loves everyone in the world”
• ∀y ∃x Loves(x,y)
  – “Everyone in the world is loved by at least one person”

• Quantifier duality: each can be expressed using the other
• ∀x Likes(x,IceCream) \rightarrow \neg\exists x \neg\text{Likes}(x,\text{IceCream})
• ∃x Likes(x,Broccoli) \rightarrow \forall x \neg\text{Likes}(x,\text{Broccoli})
Using FOL

The kinship domain:

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

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

• One's mother is one's female parent
  \[ \forall m, c \, \text{Mother}(c) = m \iff (\text{Female}(m) \land \text{Parent}(m, c)) \]

• 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., definition of Sibling in terms of Parent:
  \[
  \forall x, y \ Sibling(x, y) \iff \neg(x = y) \land \exists m, f \neg (m = f) \land \\
  \quad \land \ \text{Parent}(m, x) \land \text{Parent}(f, x) \land \text{Parent}(m, y) \land \text{Parent}(f, y)
  \]
Using FOL

The set domain:
- \( \forall s \ Set(s) \iff (s = \{\} \lor (\exists x,s_2 \ Set(s_2) \land s = \{x|s_2\})) \)
- \( \neg \exists x,s \ (x|s) = \{\} \)
- \( \forall x,s \ x \in s \iff s = \{x|s\} \)
- \( \forall x,s \ x \in s \iff [\exists y,s_2 \ (s = \{y|s_2\} \land (x = y \lor x \in s_2))] \)
- \( \forall s_1,s_2 \ s_1 \subseteq s_2 \iff (\forall x \ x \in s_1 \Rightarrow x \in s_2) \)
- \( \forall s_1,s_2 \ (s_1 = s_2) \iff (s_1 \subseteq s_2 \land s_2 \subseteq s_1) \)
- \( \forall x,s_1,s_2 \ x \in (s_1 \cap s_2) \iff (x \in s_1 \land x \in s_2) \)
- \( \forall x,s_1,s_2 \ x \in (s_1 \cup s_2) \iff (x \in s_1 \lor x \in s_2) \)
Interacting with FOL KBs

• Suppose a wumpus-world agent is using an FOL KB and perceives a smell and a breeze (but no glitter) at t=5:

  \textbf{Tell}(\text{KB}, \text{Percept}([\text{Smell}, \text{Breeze}, \text{None}],5))

  \textbf{Ask}(\text{KB}, \exists a \text{ BestAction}(a,5))

• I.e., does the KB entail some best action at t=5?

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

• Given a sentence \( S \) and a substitution \( \sigma \),

  \( S\sigma \) denotes the result of plugging \( \sigma \) into \( S \); e.g.,

  \( S = \text{Smarter}(x,y) \)

  \( \sigma = \{x/\text{Hillary}, y/\text{Bill}\} \)

  \( S\sigma = \text{Smarter}(\text{Hillary}, \text{Bill}) \)

• \textbf{Ask}(\text{KB},S) returns some/all \( \sigma \) such that KB \( \vdash \sigma \)
Knowledge base for the wumpus world

• **Perception**
  - $\forall b, g, t \text{ Percept}([\text{Smell}, b, g], t) \Rightarrow \text{Smelt}(t)$
  - $\forall s, b, t \text{ Percept}([s, b, \text{Glitter}], t) \Rightarrow \text{AtGold}(t)$

• **Reflex**
  - $\forall t \text{ AtGold}(t) \Rightarrow \text{BestAction}(\text{Grab}, t)$

• **Reflex with internal state**
  - $\forall t \text{ AtGold}(t) \land \neg \text{Holding}(\text{Gold}, t) \Rightarrow \text{BestAction}(\text{Grab}, t)$
  - “Holding” can’t be observed, so we need to keep track of state change.
Deducing hidden properties

• $\forall x,y,a,b \; \text{Adjacent}([x,y],[a,b]) \iff [a,b] \in \{[x+1,y], [x-1,y], [x,y+1], [x,y-1]\}$

Properties of squares:
• $\forall s,t \; \text{At}(\text{Agent},s,t) \land \text{Breeze}(t) \Rightarrow \text{Breezy}(s)$

Squares are breezy near a pit:
– **Diagnostic** rule---infer cause from effect
  \[ \forall s \; \text{Breezy}(s) \Rightarrow [\exists \{r\} \; \text{Adjacent}(r,s) \land \text{Pit}(r)] \]
– **Causal** rule---infer effect from cause
  \[ \forall r \; \text{Pit}(r) \Rightarrow [\forall s \; \text{Adjacent}(r,s) \Rightarrow \text{Breezy}(s) ] \]
Knowledge engineering in FOL

1. Identify the task
2. Assemble the relevant knowledge
3. Decide on a vocabulary of predicates, functions, and constants
4. Encode general knowledge about the domain
5. Encode a description of the specific problem instance
6. Pose queries to the inference procedure and get answers
7. Debug the knowledge base
The electronic circuits domain

One-bit full adder
The electronic circuits domain

1. Identify the task
   – Does the circuit actually add properly? (circuit verification)

2. Assemble the relevant knowledge
   – Composed of wires and gates; Types of gates (AND, OR, XOR, NOT)
   – Irrelevant: size, shape, color, cost of gates

3. Decide on a vocabulary (ontology, what exists)
   – Alternatives:
     Type($X_1$) = XOR
     Type($X_1$, XOR)
     XOR($X_1$)
The electronic circuits domain

4. Encode general knowledge of the domain (epistemology)
   - \( \forall t_1, t_2 \) Connected\((t_1, t_2) \) \( \Rightarrow \) Signal\((t_1) = \) Signal\((t_2) \)
   - \( \forall t \) Signal\((t) = 1 \) \( \lor \) Signal\((t) = 0 \)
   - \( 1 \neq 0 \)
   - \( \forall t_1, t_2 \) Connected\((t_1, t_2) \) \( \Rightarrow \) Connected\((t_2, t_1) \)
   - \( \forall g \) Type\((g) = \text{OR} \) \( \Rightarrow \) Signal\((\text{Out}(1,g)) = 1 \iff \exists n \) Signal\((\text{In}(n,g)) = 1 \)
   - \( \forall g \) Type\((g) = \text{AND} \) \( \Rightarrow \) Signal\((\text{Out}(1,g)) = 0 \iff \exists n \) Signal\((\text{In}(n,g)) = 0 \)
   - \( \forall g \) Type\((g) = \text{XOR} \) \( \Rightarrow \) Signal\((\text{Out}(1,g)) = 1 \iff \)
   \hspace{1em} \text{Signal}(\text{In}(1,g)) \neq \text{Signal}(\text{In}(2,g))
   - \( \forall g \) Type\((g) = \text{NOT} \) \( \Rightarrow \) Signal\((\text{Out}(1,g)) \neq \text{Signal}(\text{In}(1,g)) \)
The electronic circuits domain

5. Encode the specific problem instance
   Type($X_1$) = XOR       Type($X_2$) = XOR
   Type($A_1$) = AND       Type($A_2$) = AND
   Type($O_1$) = OR

   Connected(Out(1,$X_1$),In(1,$X_2$))  Connected(Out(1,$X_1$),In(1,$A_2$))
   Connected(Out(1,$X_1$),In(2,$A_2$))  Connected(Out(1,$A_2$),In(1,$O_1$))
   Connected(Out(1,$A_2$),In(1,$O_1$))  Connected(Out(1,$A_1$),In(2,$O_1$))
   Connected(Out(1,$A_1$),In(2,$O_1$))  Connected(Out(1,$X_2$),Out(1,$C_1$))
   Connected(Out(1,$X_2$),Out(1,$C_1$))  Connected(Out(1,$O_1$),Out(2,$C_1$))
   Connected(Out(1,$O_1$),Out(2,$C_1$))  Connected(In(3,$C_1$),In(2,$X_1$))
   Connected(In(3,$C_1$),In(2,$X_2$))  Connected(In(3,$C_1$),In(1,$A_2$))
The electronic circuits domain

6. Pose queries to the inference procedure

What are the possible sets of values of all the terminals for the adder circuit?

\[
\exists i_1,i_2,i_3,o_1,o_2 \ \text{Signal(In(1,C_1))} = i_1 \land \text{Signal(In(2,C_1))} = i_2 \land \text{Signal(In(3,C_1))} = i_3 \land \text{Signal(Out(1,C_1))} = o_1 \land \text{Signal(Out(2,C_1))} = o_2
\]

7. Debug the knowledge base

May have omitted assertions like \(1 \neq 0\)
Summary

• First-order logic:
  – objects and relations are semantic primitives
  – syntax: constants, functions, predicates, equality, quantifiers

• Increased expressive power: sufficient to define wumpus world