First-Order Logic

AIMA, Chapter 8
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
Outline

- Syntax and semantics of FOL
- Using FOL
- Wumpus world in FOL
First-order logic: Ontological commitments

- Propositional logic assumes the world contains \textit{facts}

- First-order logic (like natural language) assumes the world contains
  - \textit{Objects}: people, houses, numbers, colors, baseball games, wars, centuries…
  - \textit{n-ary Relations}: red, round, prime, brother of, bigger than, part of, comes between, bogus …
  - \textit{Functions}: father of, best friend, third inning of, 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>Probability theory</td>
<td>facts</td>
<td>degree of belief</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>
</tbody>
</table>
Syntax of FOL: Basic elements

- **Constants**  
  *KingJohn, 2, Penn,* ...

- **Predicates**  
  *Brother(x,y), King(x), Loves(x,y),* ...

- **Functions**  
  *Sqrt, LeftLegOf,* ...

- **Variables**  
  *x, y, a, b,* ...

- **Connectives**  
  *¬, ⇒, ∧, ∨, ⇔*

- **Equality**  
  *=*

- **Quantifiers**  
  *∀, ∃*
Atomic sentences

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

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

- For example:
  - \(\text{Brother(KingJohn, RichardTheLionheart)}\)
  - \(\text{Loves}(x, 2)\)
Complex sentences

- Complex sentences are made from atomic sentences using connectives
  \[ \neg S, S_1 \land S_2, S_1 \lor S_2, S_1 \Rightarrow S_2, S_1 \Leftrightarrow S_2, \]

For example

\[ \text{Sibling}(\text{KingJohn}, \text{Richard}) \Rightarrow \text{Sibling}(\text{Richard}, \text{KingJohn}) \]
\[ \text{Loves}(\text{KingJohn}, \text{KingJohn}) \]
Truth in first-order logic

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

- Model contains objects (domain elements) and relations among them.

- Interpretation specifies referents for
  - constant symbols → objects
  - predicate symbols → 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 $\text{predicate}$

- What about sentences with variables? Later.....
Models for FOL: Example
Truth example

Consider the interpretation in which

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

Under this interpretation, **Brother(Richard,John)** is true just in case **Richard the Lionheart** and **the evil King John** are in **the brotherhood relation** in the model.
Universal quantification

- $\forall$<variables> <sentence>

Everyone at Penn is smart:

$\forall x \text{At}(x,\text{Penn}) \Rightarrow \text{Smart}(x)$

- $\forall x P$ 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 $P$
  
  $\text{At}(\text{KingJohn},\text{Penn}) \Rightarrow \text{Smart}(\text{KingJohn})$
  
  $\wedge \text{At}(\text{Richard},\text{Penn}) \Rightarrow \text{Smart}(\text{Richard})$
  
  $\wedge \text{At}(\text{Penn},\text{Penn}) \Rightarrow \text{Smart}(\text{Penn})$
  
  $\wedge \ldots$
A common mistake to avoid

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

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

$$\forall x \ At(x,Penn) \land Smart(x)$$

means

“Everyone is at Penn and everyone is smart”
Existential quantification

• $\exists <\text{variables}> <\text{sentence}>$

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

• $\exists x P$ 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 $P$:

  $\text{At}(\text{KingJohn},\text{Penn}) \land \text{Smart}(\text{KingJohn})$

  $\lor \text{At}(\text{Richard},\text{Penn}) \land \text{Smart}(\text{Richard})$

  $\lor \text{At}(\text{Penn},\text{Penn}) \land \text{Smart}(\text{Penn})$
Another common mistake to avoid

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

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

$\exists x\ At(x, Penn) \Rightarrow Smart(x)$

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

- \( \forall x \forall y \) is the same as \( \forall y \forall x \)
- \( \exists x \exists y \) is the same as \( \exists y \exists x \)
- \( \exists x \forall y \) is not the same as \( \forall y \exists x \)
- \( \exists x \forall y \text{ Loves}(x,y) \)
  - “There is a person who loves everyone in the world”
- \( \forall y \exists x \text{ Loves}(x,y) \)
  - “Everyone in the world is loved by at least one person”

- **Quantifier duality**: each can be expressed using the other

\[
\forall x \text{ Likes}(x,\text{IceCream}) \iff \exists x \neg \text{Likest}(x,\text{IceCream})
\]
\[
\exists x \text{ Likes}(x,\text{Broccoli}) \iff \forall x \neg \text{Likes}(x,\text{Broccoli})
\]
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*:

$$\forall x,y \ Sibling(x,y) \Leftrightarrow \neg (x = y) \land \exists m,f \neg (m = f) \land Parent(m,x) \land Parent(f,x) \land Parent(m,y) \land Parent(f,y)$$
Using FOL for the wumpus world

“Perception”
- \( \forall s,b,t \ Percept([s,b,Glitter],t) \Rightarrow AtGold(t) \)
- \( \forall b,g,t \ Percept([Stench],t) \Rightarrow Stench(t) \)

Reflex
- \( \forall t \ AtGold(t) \Rightarrow Action(Grab,t) \)

Reflex with internal state: do we have gold already?
- \( \forall t \ AtGold(t) \land \neg Holding(Gold,t) \Rightarrow Action(Grab,t) \)

\( Holding(Gold,t) \) cannot be observed
- keeping track of change is essential
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:

\[ \forall s,t \text{ At(Agent},s,t) \land \text{Breezy}(t) \Rightarrow \text{Breezy}(s) \]

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
  \[ \forall r \text{ Pit}(r) \Rightarrow [\forall s \text{ Adjacent}(r,s) \Rightarrow \text{Breezy}(s) ] \]
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 $t = 5$:

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

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

- Answer: Yes, \{a/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 $KB \models S\sigma$ `