Readings

• Chap. 8 of [Russell and Norvig, 2012]
Pros and cons of Propositional Logic

+ PL is **declarative**: pieces of syntax correspond to facts

+ PL 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 . . .
Syntax of FOL: Basic elements

Constants  \( KingJohn, 2, UCB, \ldots \)
Predicates  \( Brother, >, \ldots \)
Functions  \( Sqrt, LeftLegOf, \ldots \)
Variables  \( x, y, a, b, \ldots \)
Connectives  \( \land, \lor, \neg, \Rightarrow, \Leftrightarrow \)
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.,  $\text{Brother}(\text{KingJohn}, \text{RichardTheLionheart})$

$> (\text{Length}(\text{LeftLegOf}(\text{Richard})), \text{Length}(\text{LeftLegOf}(\text{KingJohn})))$
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. \( \textit{Sibling}(\textit{KingJohn}, \textit{Richard}) \Rightarrow \textit{Sibling}(\textit{Richard}, \textit{KingJohn}) \)
\[ > (1, 2) \lor \leq (1, 2) \]
\[ > (1, 2) \land \neg > (1, 2) \]
Language of FOL: Grammar

Sentence ::= AtomicS | ComplexS

AtomicS ::= True | False | RelationSymb(Term, . . .) | Term = Term

ComplexS ::= (Sentence) | Sentence Connective Sentence | ¬Sentence
         | Quantifier Sentence

Term ::= FunctionSymb(Term, . . .) | ConstantSymb | Variable

Connective ::= ∧ | ∨ | ⇒ | ⇔

Quantifier ::= ∀ Variable | ∃ Variable

Variable ::= a | b | · · · | x | y | · · ·

ConstantSymb ::= A | B | · · · | John | 0 | 1 | · · · | π | · · ·

FunctionSymb ::= F | G | · · · | Cosine | Height | FatherOf | + | · · ·

RelationSymb ::= P | Q | · · · | Red | Brother | Apple | > | · · ·
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
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.
Semantics of First-Order Logic

(A little) more formally:

An **interpretation** is a pair \((\mathcal{D}, \sigma)\) where

- \(\mathcal{D}\) is a set of objects, the universe (or *domain*);
- \(\sigma\) is mapping from variables to objects in \(\mathcal{D}\);
- \(C^\mathcal{D}\) is an object in \(\mathcal{D}\) for every constant symbol \(C\);
- \(F^\mathcal{D}\) is a function from \(\mathcal{D}^n\) to \(\mathcal{D}\) for every function symbol \(F\) of arity \(n\);
- \(R^\mathcal{D}\) is a relation over \(\mathcal{D}^n\) for every relation symbol \(R\) of arity \(n\);
An Interpretation $I$ in the Blocks World

Constant Symbols: $A, B, C, D, E, T$

Function Symbols: $Support$

Relation Symbols: $On, Above, Clear$

$A^H = A, B^H = B, C^H = C, D^H = D, E^H = E, T^H = T$

$Support^H = \{<A, T>, <B, A>, <C, B>, <D, C>, <E, D>\}$

$On^H = \{<A, T>, <B, A>, <C, B>, <D, C>, <E, D>\}$

$Above^H = \{<E, D>, <D, C>, ...\}$

$Clear^H = \{<E>\}$
Semantics of First-Order Logic

Let \((D, \sigma)\) be an interpretation and \(E\) an expression of FOL. We write \([E]_\sigma^D\) to denote the **meaning of \(E\) in the domain \(D\) under the variable assignment \(\sigma\)**.

The meaning \([t]_\sigma^D\) of a term \(t\) is an object of \(D\). It is inductively defined as follows.

\[
\begin{align*}
[x]_\sigma^D & := \sigma(x) \quad \text{for all variables} \ x \\
[C]_\sigma^D & := C^D \quad \text{for all constant symbols} \ C \\
[F(t_1, \ldots, t_n)]_\sigma^D & := F^D([t_1]_\sigma^D, \ldots, [t_n]_\sigma^D) \quad \text{for all function symbols} \ F \\
& \quad \text{of arity} \ n
\end{align*}
\]
Example

Consider the symbols $MotherOf$, $SchoolOf$, $Bill$ and the interpretation $(\mathcal{D}, \sigma)$ where

- $MotherOf^\mathcal{D}$ is a unary fn mapping people to their mother
- $FchildOf^\mathcal{D}$ is a binary fn mapping a couple to their first child

$$\sigma := \{x \mapsto \text{George W Bush}, \ y \mapsto \text{Barbara Bush}\}$$

What is the meaning of $\lceil MotherOf(x) \rceil$ according to $(\mathcal{D}, \sigma)$?

$$\lceil MotherOf(x) \rceil^\mathcal{D}_\sigma = \lceil MotherOf \rceil^\mathcal{D}_\sigma(\lceil x \rceil^\mathcal{D}_\sigma) = MotherOf^\mathcal{D}(\sigma(x)) = \text{Barbara Bush}$$
Semantics of First-Order Logic

The meaning $\llbracket \varphi \rrbracket^D_{\sigma}$ of a formula $\varphi$ is either True or False.

It is inductively defined as follows.

- $\llbracket t_1 = t_2 \rrbracket^D_{\sigma} := True$ iff $\llbracket t_1 \rrbracket^D_{\sigma}$ is the same as $\llbracket t_2 \rrbracket^D_{\sigma}$
- $\llbracket R(t_1, \ldots, t_n) \rrbracket^D_{\sigma} := True$ iff $\langle \llbracket t_1 \rrbracket^D_{\sigma}, \ldots, \llbracket t_n \rrbracket^D_{\sigma} \rangle \in R^D$
- $\llbracket \neg \varphi \rrbracket^D_{\sigma} := True/False$ iff $\llbracket \varphi \rrbracket^D_{\sigma} = False/True$
- $\llbracket \varphi_1 \lor \varphi_2 \rrbracket^D_{\sigma} := True$ iff $\llbracket \varphi_1 \rrbracket^D_{\sigma} = True$ or $\llbracket \varphi_2 \rrbracket^D_{\sigma} = True$
- $\llbracket \exists x \varphi \rrbracket^D_{\sigma} := True$ iff $\llbracket \varphi \rrbracket^D_{\sigma'} = True$ for some $\sigma'$ the same as $\sigma$ except for $x$
Semantics of First-Order Logic

The meaning of formulas built with the other logical symbols can be defined by reduction to the previous symbols.

\[
\begin{align*}
\llbracket \varphi_1 \land \varphi_2 \rrbracket^D_{\sigma} & := \llbracket \neg(\neg \varphi_1 \lor \neg \varphi_2) \rrbracket^D_{\sigma} \\
\llbracket \varphi_1 \rightarrow \varphi_2 \rrbracket^D_{\sigma} & := \llbracket \neg \varphi_1 \lor \varphi_2 \rrbracket^D_{\sigma} \\
\llbracket \varphi_1 \leftrightarrow \varphi_2 \rrbracket^D_{\sigma} & := \llbracket (\varphi_1 \rightarrow \varphi_2) \land (\varphi_2 \rightarrow \varphi_1) \rrbracket^D_{\sigma} \\
\llbracket \forall x \varphi \rrbracket^D_{\sigma} & := \llbracket \neg \exists x \neg \varphi \rrbracket^D_{\sigma}
\end{align*}
\]

If a sentence is closed (no free variables), its meaning does not depend on the variable assignment (although it may depend on the domain):

\[
\llbracket \forall x \exists y R(x, y) \rrbracket^D_{\sigma} = \llbracket \forall x \exists y R(x, y) \rrbracket^D_{\sigma'}, \quad \text{for any } \sigma, \sigma'
\]
Models, Validity, etc. for Sentences

An interpretation \((\mathcal{D}, \sigma)\) satisfies a sentence \(\varphi\), or is a model for \(\varphi\), if \([\varphi]^{\mathcal{D}}_{\sigma} = True\).

A sentence is satisfiable if it has at least one model.

Examples: \(\forall x \ x \geq y, \ P(x)\)

A sentence is unsatisfiable if it has no models.

Examples: \(P(x) \land \neg P(x), \ \neg(x = x)\)

A sentence \(\varphi\) is valid if every interpretation is a model for \(\varphi\).

Examples: \(P(x) \Rightarrow P(x), \ x = x\)

\(\varphi\) is valid/unsatisfiable iff \(\neg\varphi\) is unsatisfiable/valid.
Models, Validity, etc. for Sets of Sentences

An interpretation \((\mathcal{D}, \sigma)\) satisfies a set \(\Gamma\) of sentences, or is a model for \(\Gamma\), if it is a model for every sentence in \(\Gamma\).

A set \(\Gamma\) of sentences is satisfiable if it has at least one model.

Ex: \(\{\forall x \ x \geq 0, \ \forall x \ x + 1 > x\}\)

\(\Gamma\) is unsatisfiable, or inconsistent, if it has no models.

Ex: \(\{P(x), \neg P(x)\}\)

As in Propositional Logic, \(\Gamma\) entails a sentence \(\varphi\) (\(\Gamma \models \varphi\)), if every model of \(\Gamma\) is also a model of \(\varphi\).

Ex: \(\{\forall x \ P(x) \Rightarrow Q(x), \ P(A_{10})\} \models Q(A_{10})\)

Note: Again, \(\Gamma \models \varphi\) iff \(\Gamma \land \neg \varphi\) is unsatisfiable.
Possible Interpretations Semantics

Sentences can be seen as *constraints* on the set $S$ of all possible interpretations.

A sentence *denotes* all the possible interpretations that satisfy it (the models of $\varphi$):

If $\varphi_1$ denotes a set of interpretations $S_1$ and $\varphi_2$ denotes a set $S_2$, then

- $\varphi_1 \lor \varphi_2$ denotes $S_1 \cup S_2$,
- $\varphi_1 \land \varphi_2$ denotes $S_1 \cap S_2$,
- $\neg \varphi_1$ denotes $S \setminus S_1$,
- $\varphi_1 \models \varphi_2$ iff $S_1 \subseteq S_2$.

A sentence denotes either no interpretations or an infinite number of them!

Valid sentences do not tell us anything about the world. They are satisfied by every possible interpretation!
Models for FOL: Lots!

We *can* enumerate the models for a given FOL sentence:

For each number of universe elements $n$ from 1 to $\infty$
   For each $k$-ary predicate $P_k$ in the sentence
      For each possible $k$-ary relation on $n$ objects
         For each constant symbol $C$ in the sentence
            For each one of $n$ objects mapped to $C$
               ...

Enumerating models is not going to be easy!
Universal quantification

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

Everyone at Berkeley is smart:
\[ \forall x \ At(x, Berkeley) \Rightarrow Smart(x) \]

\[ \forall x \ P \] 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 \( P \)

\[
(At(KingJohn, Berkeley) \Rightarrow Smart(KingJohn)) \\
\land (At(Richard, Berkeley) \Rightarrow Smart(Richard)) \\
\land (At(Berkeley, Berkeley) \Rightarrow Smart(Berkeley)) \\
\land \ldots
\]
A common mistake to avoid

Typically, ⇒ is the main connective with ∀

Common mistake: using ∧ as the main connective with ∀:

\[ \forall x \ At(x, Berkeley) \land Smart(x) \]

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

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

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

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

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

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

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

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

∀x ∀y \text{ is the same as } ∀y ∀x \quad \text{(why?)}

∃x ∃y \text{ is the same as } ∃y ∃x \quad \text{(why?)}

∃x ∀y \text{ is not the same as } ∀y ∃x

∃x ∀y \text{ Loves}(x, y)

“There is a person who loves everyone in the world”

∀y ∃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

∀x \text{ Likes}(x, \text{IceCream}) \quad \neg ∃x \neg \text{Likes}(x, \text{IceCream})

∃x \text{ Likes}(x, \text{Broccoli}) \quad \neg ∀x \neg \text{Likes}(x, \text{Broccoli})
Brothers are siblings
Fun with sentences

Brothers are siblings

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

“Sibling” is symmetric
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
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
∀ x, y  Mother(x, y) ⇔ (Female(x) ∧ Parent(x, y)).

A first cousin is a child of a parent’s sibling
Fun with sentences

Brothers are siblings
\[ \forall x, y \quad Brother(x, y) \Rightarrow Sibling(x, y). \]

“Sibling” is symmetric
\[ \forall x, y \quad Sibling(x, y) \Leftrightarrow Sibling(y, x). \]

One’s mother is one’s female parent
\[ \forall x, y \quad Mother(x, y) \Leftrightarrow (Female(x) \land Parent(x, y)). \]

A first cousin is a child of a parent’s sibling
\[ \forall x, y \quad FirstCousin(x, y) \Leftrightarrow \exists p, ps \quad Parent(p, x) \land Sibling(ps, p) \land Parent(ps, y). \]
Equality

\( \text{term}_1 = \text{term}_2 \) is true under a given interpretation if and only if \( \text{term}_1 \) and \( \text{term}_2 \) refer to the same object

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

E.g., definition of (full) Sibling in terms of Parent:

\[
\forall x, y \ Sibling(x, y) \iff \neg(x = y) \land \exists m, f \ \neg(m = f) \land
\neg Parent(m, x) \land Parent(f, x) \land Parent(m, y) \land Parent(f, y)
\]