First-Order Logic

Chapter 8
(not 8.1)

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 $B_{1,1} \wedge P_{1,2}$ 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, …

Syntax of FOL: Basic elements

- Constants KingJohn, 2, Udel,...
- Predicates Brother, >,...
- Functions Sqrt, LeftLegOf,...
- Variables x, y, a, b,...
- Connectives $\neg, \Rightarrow, \wedge, \vee, \Leftrightarrow$
- Equality =
- Quantifiers $\forall, \exists$

Atomic sentences

Atomic sentence = *predicate* (*term*$_1$,...,*term*$_n$)

or *term*$_1$ = *term*$_2$

Term = *constant*
or *variable*
or *function* (*term*$_1$,...,*term*$_n$)

- E.g., *Brother(KingJohn,RichardTheLionheart)* , *(Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))*
  Married(mother(mother(mary)), father(mother(mary)))

Representing Some Sentences

- John likes Mary.
- Bill hit John.
- Florida is a state.
- The product of a and b is c.
- 2+3=7.
- Tweety is yellow.

Complex sentences

Complex sentences are made from atomic sentences using connectives

$$\neg S, S_1 \wedge S_2, S_1 \vee S_2, S_1 \Rightarrow S_2, S_1 \Leftrightarrow S_2,$$

E.g. *Sibling(KingJohn,Richard)* $\Rightarrow$ *Sibling(Richard,KingJohn)*

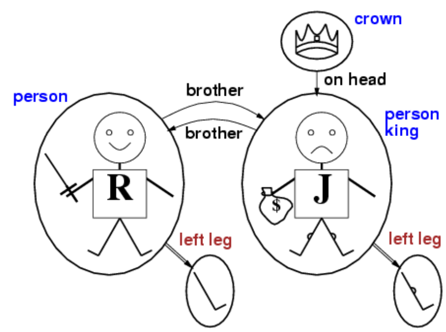
$>(1,2) \vee \leq (1,2)$

$>(1,2) \wedge \neg >(1,2)$

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
  - function symbols → functional relations
- An atomic sentence *predicate(term*$_1$*,...,term*$_n$*)* is true iff the objects referred to by *term*$_1$*,...,term*$_n$ are in the relation referred to by *predicate*

Models for FOL: Example



Universal quantification

- $\forall$<*variables*> <*sentence*>

Everyone at UDel is smart:
$\forall$x At(x,UDel) $\Rightarrow$ 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 *x*

At(KingJohn,UDel) $\Rightarrow$ Smart(KingJohn)
$\wedge$ At(Richard,UDel) $\Rightarrow$ Smart(Richard)
$\wedge$ At(NUS,UDel) $\Rightarrow$ Smart(NUS)
$\wedge$ ...

A common mistake to avoid

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

- Common mistake: using $\wedge$ as the main connective with $\forall$:
  $\forall$x At(x,UDel) $\wedge$ Smart(x)
  means "Everyone is at Udel and everyone is smart"

Existential quantification

- $\exists$*<variables> <sentence>*

Someone at Udel is smart:
$\exists x$ At(x,UDel) $\wedge$ 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*

  At(KingJohn,UDel) $\wedge$ Smart(KingJohn)
  $\vee$ At(Richard,UDel) $\wedge$ Smart(Richard)
  $\vee$ At(NUS,UDel) $\wedge$ Smart(NUS)
  $\vee$ ...

Another common mistake to avoid

- Typically, $\wedge$ is the main connective with $\exists$
- Common mistake: using $\Rightarrow$ as the main connective with $\exists$:

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

  is true if there is anyone who is not at UDel!

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 $\forall y\, \exists x$
- $\exists x\, \forall y$ 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) $\neg\exists x\, \neg$Likes(x,IceCream)
- $\exists x$ Likes(x,Broccoli) $\neg\forall x\, \neg$Likes(x,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) $\wedge$ $\exists$m,f $\neg$ (m = f) $\wedge$ Parent(m,x) $\wedge$ Parent(f,x) $\wedge$ Parent(m,y) $\wedge$ Parent(f,y)]

Using FOL

The kinship domain:

- Brothers are siblings
  $\forall x,y$ *Brother(x,y)* $\Leftrightarrow$ *Sibling(x,y)*
- One's mother is one's female parent
  $\forall m,c$ *Mother(c)* = m $\Leftrightarrow$ *(Female(m)* $\wedge$ *Parent(m,c))*
- "Sibling" is symmetric
  $\forall x,y$ *Sibling(x,y)* $\Leftrightarrow$ *Sibling(y,x)*

Using FOL

The set domain:

- $\forall s\ Set(s) \Leftrightarrow (s = \{\}) \vee (\exists x,s_2\ Set(s_2) \wedge s = \{x|s_2\})$
- $\neg\exists x,s\ \{x|s\} = \{\}$
- $\forall x,s\ x \in s \Leftrightarrow s = \{x|s\}$
- $\forall x,s\ x \in s \Leftrightarrow [\ \exists y,s_2\} (s = \{y|s_2\} \wedge (x = y \vee x \in s_2))]$
- $\forall s_1,s_2\ s_1 \subseteq s_2 \Leftrightarrow (\forall x\ x \in s_1 \Rightarrow x \in s_2)$
- $\forall s_1,s_2\ (s_1 = s_2) \Leftrightarrow (s_1 \subseteq s_2 \wedge s_2 \subseteq s_1)$
- $\forall x,s_1,s_2\ x \in (s_1 \cap s_2) \Leftrightarrow (x \in s_1 \wedge x \in s_2)$
- $\forall x,s_1,s_2\ x \in (s_1 \cup s_2) \Leftrightarrow (x \in s_1 \vee x \in s_2)$

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

Summary

- First-order logic:
  - objects and relations are semantic primitives
  - syntax: constants, functions, predicates, equality, quantifiers
- Increased expressive power: sufficient to define wumpus world