First-Order Logic

R&N: §8.1-8.3

Jacques Fleuriot

School of informatics
University of Edinburgh

Outline

- Why FOL?
- Syntax and semantics of FOL
- Using FOL
- Wumpus world 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:
  The meaning of $B_{1,1} \wedge P_{1,2}$ is derived from that 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)
  - for example, we 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, colours, football 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*, *UoE*,...
- Predicates *Brother*, >,...
- Functions *Sqrt*, *LeftLegOf*,...
- Variables *x*, *y*, *a*, *b*,...
- Connectives $\neg$, $\Rightarrow$, $\wedge$, $\vee$, $\Leftrightarrow$
- Equality =
- Quantifiers $\forall$, $\exists$

Atomic formulae

Atomic formula = *predicate* ($term_1$,...,$term_n$)
or $term_1$ = $term_2$

Term = *function* ($term_1$,...,$term_n$)
or *constant* or *variable*

Examples:

- *Brother*(*KingJohn*,*RichardTheLionheart*)
- >(*Length*(*LeftLegOf*(*Richard*)), *Length*(*LeftLegOf*(*KingJohn*)))

predicate functions constants

Complex formulae

Complex formulae are made from atomic formulae using connectives

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

Examples:

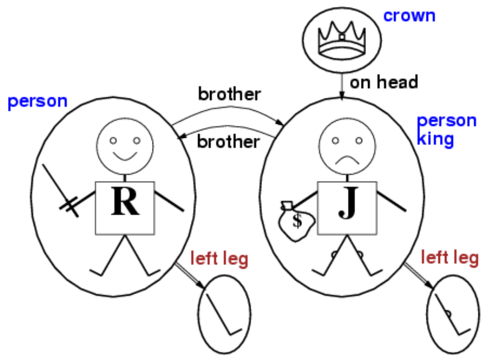
- *Sibling*(*KingJohn*,*Richard*) $\Rightarrow$ *Sibling*(*Richard*,*KingJohn*)
- $>(1,2) \vee \leq(1,2)$
- $>(1,2) \wedge \neg >(1,2)$

Semantics of first-order logic

- Formulae are mapped to an interpretation
  - An interpretation is called a model of a set of formulae when all the formulae are true in the interpretation.
- Interpretation contains objects (domain elements) and relations between them
- Mapping specifies referents for

  constant symbols $\mapsto$ objects
  predicate symbols $\mapsto$ relations
  function symbols $\mapsto$ functions
- An atomic formula *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*.

Interpretations for FOL: Example



Universal quantification

- ∀<*variables*>. <*formula*>
  - But will often write $\forall\, x,y.\ P$ for $\forall x.\ \forall y.$
  - Example: Everyone at UoE is smart: $\forall x.\ At(x, UoE) \Rightarrow Smart(x)$
- $\forall x.\ P$ is true in an interpretation m iff P is true with x being each possible object in the interpretation.
- Roughly speaking, equivalent to the conjunction of instantiations of P

$$
\begin{aligned}
& At(KingJohn, UoE) \Rightarrow Smart(KingJohn) \\
\wedge\ & At(Richard, UoE) \Rightarrow Smart(Richard) \\
\wedge\ & At(UoE, UoE) \Rightarrow Smart(UoE) \\
\wedge\ & \ldots
\end{aligned}
$$

A common mistake to avoid

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

$\forall x.\ At(x, UoE) \wedge Smart(x)$
means "Everyone is at UoE and everyone is smart"

Existential quantification

- ∃<*variables*>. <*formula*>
  - But will often write $\exists\, x,y.\ P$ for $\exists\ x.\ \exists\ y.\ P$
  - Example: Someone at UoE is smart: $\exists x.\ At(x, UoE) \wedge Smart(x)$
- $\exists x.\ P$ is true in an interpretation m iff P is true with x being some possible object in the model.
- Roughly speaking, equivalent to the disjunction of instantiations of P

$$
\begin{aligned}
& At(KingJohn, UoE) \wedge Smart(KingJohn) \\
\vee\ & At(Richard, UoE) \wedge Smart(Richard) \\
\vee\ & At(UoE, UoE) \wedge Smart(UoE) \\
\vee\ & \ldots
\end{aligned}
$$

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,UoE) $\Rightarrow$ Smart(x)

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

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) $\equiv$ $\neg\exists x.\neg$Likes(x,IceCream)
  - $\exists x.$ Likes(x,Broccoli) $\equiv$ $\neg\forall x.\neg$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.
- Example. 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

Example: The kinship domain:

- Brothers are siblings

  $\forall x,y.$ *Brother*$(x,y) \Rightarrow$ *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:

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

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*:

  `Tell`(KB, Percept([Smell,Breeze,None], 5))
  `Ask`(KB,∃a. 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 σ,
  – *S*σ denotes the result of "plugging" σ into *S*; e.g.,
    S = Smarter(x, y)
    σ = {x/Obama, y/Palin}
    *S*σ = Smarter(Obama, Palin)

- `Ask`(KB,S) returns some/all σ such that KB ⊨ Sσ

Knowledge base for the wumpus world

- Perception
  – $\forall t,s,b.$ Percept([*s*,*b*,Glitter],*t*) ⇒ Glitter(*t*)
- Reflex
  – $\forall t.$ Glitter(*t*) ⇒ BestAction(Grab,*t*)

Deducing hidden properties

- $\forall x,y,a,b.$ *Adjacent*([*x*,*y*],[a,b]) ⇔ [*a*,*b*] ∈ {[*x*+1,y], [*x*-1,y],[*x*,*y*+1],[*x*,*y*-1]}
- ∀s,t. *At*(Agent,s,t) ∧ Breeze(t) ⇒ Breezy(s)
- Squares are breezy near a pit:
  – Diagnostic rule: infer cause from effect
    ∀*s*. Breezy(*s*) ⇒ ∃*r*. Adjacent(*r*,*s*) ∧ Pit(*r*)
  – Causal rule: infer effect from cause
    ∀*r*. Pit(*r*) ⇒ [∀*s*. Adjacent(*r*,*s*) ⇒ Breezy(*s*)]

Summary

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