Last update: December 11, 2012

First-order logic

CMSC 421: Chapter 8

Motivation

- **Problem**: propositional logic has limited expressive power
  - E.g., can't say "pits cause breezes in adjacent squares"
    - Instead, must write one sentence for each square
- Need a logic that's more expressive
  - $\Rightarrow$ First Order Logic (FOL)

Outline

- Syntax and semantics of FOL
- Examples of sentences
- Wumpus world in FOL

Basic entities in FOL

◇ Propositional logic assumes world contains *facts* (statements that are true)

◇ First-order logic (like natural language) assumes the world contains

- *Objects*: people, houses, numbers, theories, colors, Testudo,
  baseball games, homework assignments, wars, centuries ...
- *Relations*: x is red, x is prime, x is at bat, x is y's brother,
  x's color is y, x owns y, x pwned y, x occurred after y,
  x is between y and z, x, y, and z sum to w, ...
- *Functions*: the sister of x and y, x's domain name server,
  the third inning of x, the product of x, y, and z,
  the end of x, ...

Logics in general

Language	What things it talks about	What it says about those things
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief
Fuzzy logic	facts + degree of truth	known interval value

Syntax of FOL: Basic elements

- Constant symbols $King\_John$, 2, $University\_of\_Maryland$, $\ldots$
- Predicate symbols $Brother$, $>$, $\ldots$
- Function symbols $Sqrt$, $Left\_leg\_of$, $\ldots$
- Variable symbols x, y, a, b, $\ldots$
- Connectives $\land$ $\lor$ $\neg$ $\Rightarrow$ $\Leftrightarrow$
- Equality $=$
- Quantifiers $\forall$ $\exists$
- Punctuation ( ) ,

Atomic sentences

♢ Atomic sentence = $predicate(term_1, \ldots, term_n)$
or $term_1 = term_2$

♢ Term = $function(term_1, \ldots, term_n)$
or $constant$
or $variable$

♢ E.g.,

- $Brother(King\_John, Richard)$
- $> (Length(Left\_leg\_of(Richard)), Length(Left\_leg\_of(King\_John)))$

Complex sentences

◊ Complex sentences are made from atomic sentences using connectives

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

◊ E.g.,

$$Sibling(King\_John, Richard) \Rightarrow Sibling(Richard, King\_John)$$

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

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

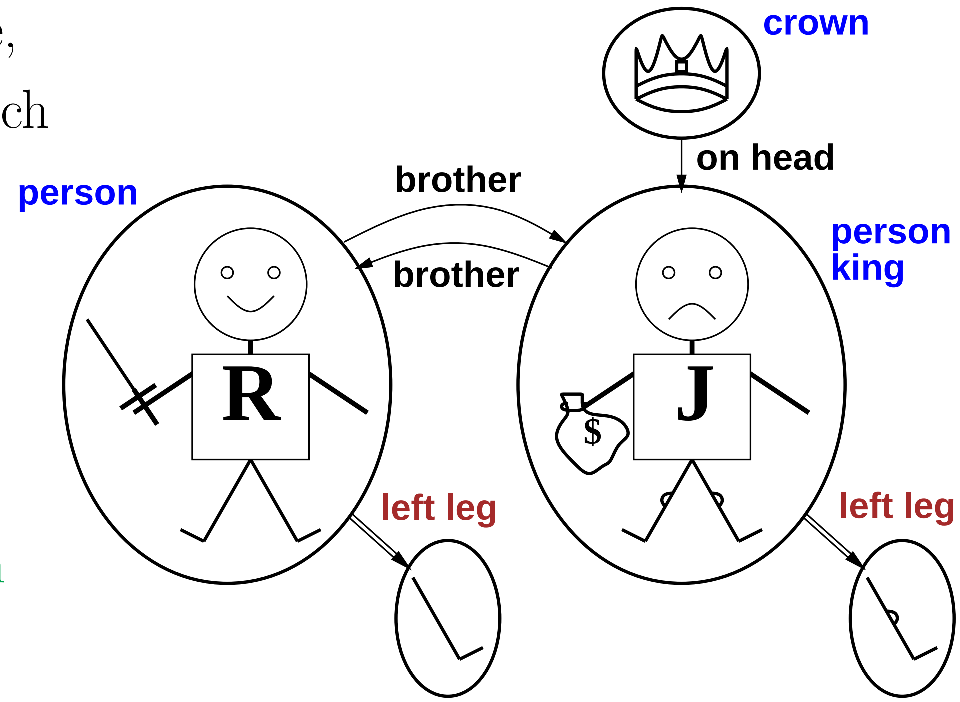
Truth in first-order logic

- In FOL, a *world* is a pair $M = (D, I)$, where
  - D is a *domain*:
    - nonempty set of objects (*domain elements*)
    - functions and relations among those objects
  - I is an *interpretation*: a function that maps
    - constant symbols → objects in the domain
    - predicate symbols → relations over objects in the domain
    - function symbols → functions over objects in the domain
- An atomic sentence $predicate(term_1, \ldots, term_n)$ is true in M iff
  - the objects that $term_1, \ldots, term_n$ refer to
    satisfy the relation that $predicate$ refers to
- As before, we say M *is a model of* a sentence α if α is true in M

Truth example

◇ Suppose $M = (D, I)$, where

- D is the domain shown here,
- I is an interpretation in which
  - ◇ $Richard \rightarrow$ Richard the lionheart
  - ◇ $John \rightarrow$ the evil King John
  - ◇ $Brother \rightarrow$ the brotherhood relation



◇ $Brother(Richard, John)$ is true in world M iff

- the pair (Richard the lionheart, the evil King John) satisfies the brotherhood relation

Models for FOL: Lots!

◇ Entailment in propositional logic can be computed by enumerating all of the possible worlds (i.e., model checking)

◇ How to enumerate possible worlds in FOL?

- For each number of domain elements $n = 1$ to ∞

  For each k-ary predicate P_k in the vocabulary

  For each possible k-ary relation on n objects

  For each constant symbol C in the vocabulary

  For each choice of referent for C from n objects ...

◇ Computing entailment in this way is not easy!

Universal quantification

$\forall \langle variables \rangle \; \langle sentence \rangle$

◇ Everyone at the University of Maryland is smart:

- $\forall x \;\; At(x, UMD) \Rightarrow Smart(x)$

◇ $\forall x \;\; P(x)$ is true in a world m iff

- $P(x)$ is true for **every** possible object x in m

◇ Roughly equivalent to the conjunction of all instantiations of $P(x)$

$$
\begin{aligned}
& (At(King\_John, UMD) \Rightarrow Smart(King\_John)) \\
\land \; & (At(Richard, UMD) \Rightarrow Smart(Richard)) \\
\land \; & (At(UMD, UMD) \Rightarrow Smart(UMD)) \\
\land \; & \ldots
\end{aligned}
$$

Existential quantification

$\exists \langle variables \rangle \ \langle sentence \rangle$

◇ Someone at the University of Maryland is smart:

- $\exists x \ At(x, UMD) \land Smart(x)$

◇ $\exists x \ P(x)$ is true in a world m iff

- $P(x)$ is true for **at least one** object x in m

◇ Roughly equivalent to the disjunction of all instantiations of $P(x)$

$$
\begin{aligned}
& (At(King\_John, UMD) \land Smart(King\_John)) \\
\lor \ & (At(Richard, UMD) \land Smart(Richard)) \\
\lor \ & (At(UMD, UMD) \land Smart(UMD)) \\
\lor \ & \ldots
\end{aligned}
$$

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
  - $\forall x\ Likes(x, IceCream)$ is the same as $\neg\exists x\ \neg Likes(x, IceCream)$
  - $\exists x\ Likes(x, Broccoli)$ is the same as $\neg\forall x\ \neg Likes(x, Broccoli)$

Examples of sentences

◊ Brothers are siblings

- $\forall\, x, y \;\; Brother(x, y) \;\Rightarrow\; Sibling(x, y)$

◊ "Sibling" is symmetric

- $\forall\, x, y \;\; Sibling(x, y) \;\Leftrightarrow\; Sibling(y, x)$

◊ One's mother is one's female parent

- $\forall\, x, y \;\; Mother(x, y) \;\Leftrightarrow\; (Female(x) \land Parent(x, y))$

◊ A first cousin is a child of a parent's sibling

- $\forall\, x, y \;\; FirstCousin(x, y) \;\Leftrightarrow$
  $\exists\, px, py \;\; Parent(px, x) \land Sibling(px, py) \land Parent(py, y)$

Equality

◇ $term_1 = term_2$ is true under a given interpretation iff $term_1$ and $term_2$ refer to the same object

◇ Examples

- $1 = 2$ and $\forall\, x \;\; \times(Sqrt(x), Sqrt(x)) = x$ are *satisfiable* (true under at least one interpretation)
- $2 = 2$ is *valid* (true in every interpretation)

◇ Definition of $Sibling$ in terms of $Parent$:

$$\forall\, x, y \;\; Sibling(x, y) \;\Leftrightarrow\; \big[\neg(x\!=\!y) \land \exists\, m \;\; \exists\, f \;\; \neg(m\!=\!f) \land Parent(m, x) \land Parent(f, x) \land Parent(m, y) \land Parent(f, y)\big]$$

Substitutions

- *Substitution*: a set of variable bindings
- Consider a substitution σ that assigns $x = 1$, $y = f(z)$
  - A logician would write $\sigma = \{1/x, f(z)/y\}$
  - Russell and Norvig write $\sigma = \{x/1, y/f(z)\}$
  - To try to avoid ambiguity, I'll write $\sigma = \{x = 1, y = f(z)\}$
- Given a sentence S and a substitution σ,
  - $S\sigma$ (postfix notation) is the result of applying σ to S
    - $S = \ GreaterThan(x, y)$
    - $\sigma = \{x = 1, y = f(z)\}$
    - $S\sigma = \ GreaterThan(1, f(z))$
  - Applied simultaneously, like `(x,y)=(1,f(z))` in Python
    - $S = \ GreaterThan(x, y)$
    - $\sigma = \{y = g(x), x = 2\}$
    - $S\sigma = \ GreaterThan(2, g(x))$, not $GreaterThan(2, g(2))$

Numbers

- The book gives axioms for natural numbers and addition
  - $Natnum(0)$
  - $\forall n \;\; Natnum(n) \;\Rightarrow\; natNum(S(n))$
  - $\forall n \;\; 0 \neq S(n)$
  - $\forall m, n \;\; m \neq n \;\Rightarrow\; S(m) \neq S(n)$
  - $\forall m \;\; NatNum(m) \;\Rightarrow\; +(0, m) = m$
  - $\forall m, n \;\; NatNum(m) \wedge NatNum(n) \;\Rightarrow\; +(S(m), n) = S(+(m, n))$
- If we introduce infix notation and rewrite $S(n)$ as $n + 1$, we can write
  - $\forall m \;\; NatNum(m) \;\Rightarrow\; 0 + m = m$
  - $\forall m, n \;\; NatNum(m) \wedge NatNum(n) \;\Rightarrow\; (m + 1) + n = (m + n) + 1$
- Useful for proving mathematical theorems, but inefficient computationally
  - In practical implementations, one would just compute it directly
  - Problem if the expression is only partly instantiated

Sets and lists

- The book also has axioms for finite sets, finite lists
  - These are more practical
  - There's a programming language called Prolog that does things like this

Interacting with FOL KBs

- Suppose KB is a first-order-logic KB of axioms for the Wumpus world
  - A model of KB consists of a domain (an actual Wumpus World) and interpretation that makes every sentence in KB true
- $Tell(KB, S)$ adds S to KB
- $Ask(KB, S)$ returns some/all σ such that $KB \models S\sigma$
- Suppose we have a way to do inference in KB (see next chapter)
  - Suppose that time $t = 5$, the agent perceives stench, breeze, glitter, no bump, no scream
- $Tell(KB, Percept([Stench, Breeze, Glitter, None, None], 5))$
- $Ask(KB, \exists\, a \;\; Action(a, 5))$
  - I.e., does KB entail an action at $t = 5$?
- Answer: $Yes,\ \{a = Grab\}$ $\quad = substitution$

Knowledge base for the wumpus world

◇ Perception

- $\forall\, t, s, g, m, c \;\; Percept([s, Breeze, g, m, c], t) \;\Rightarrow\; Breeze(t)$
- $\forall\, t, s, b, m, c \;\; Percept([s, b, Glitter, m, c], t) \;\Rightarrow\; Glitter(t)$

◇ Reflex actions (if we want a simple reflex agent)

- $\forall\, t \;\; Glitter(t) \;\Rightarrow\; BestAction(Grab, t)$

◇ Reflex with internal state: do we have the gold already?

- $\forall\, t \;\; Glitter(t) \land \neg Holding(Gold, t) \;\Rightarrow\; BestAction(Grab, t)$

◇ Example of a successor state axiom:

- $\forall\, t \;\; HaveArrow(t+1) \;\Leftrightarrow\; [HaveArrow(t) \land \neg Action(Shoot, t)]$

Reasoning about locations

◇ Adjacency:

- $\forall\, x, y, a, b \;\; Adjacent([x, y], [a, b]) \Leftrightarrow$
  $x = a \land (y = b - 1 \lor y = b + 1)) \lor (y = b \land (x = a - 1 \lor x = a + 1))$

◇ Agent can only be in one location:

- $\forall\, x, s_1, s_2, t \;\; At(x, s_1, t) \land At(x, s_2, t) \;\Rightarrow\; s_1 = s_2$

◇ Properties of locations:

- $\forall\, x, t \;\; At(Agent, x, t) \land Breeze(t) \;\Rightarrow\; Breezy(x)$
- $\ldots$

◇ Axiom to infer whether a pit is nearby:

  - ⋄ $\forall\, s \;\; Breezy(s) \;\Leftrightarrow\; [Adjacent(r, s) \land Pit(r)]$

Summary

- First-order logic:
  - objects and relations are semantic primitives
  - syntax: constants, functions, predicates, equality, quantifiers
- Increased expressive power compared to propositional logic