Knowledge Representation using First-Order Logic

CHAPTER 8
HASSAN KHOSRAVI
SPRING2011

Outline

- What is First-Order Logic (FOL)?
  - Syntax and semantics
- Using FOL
- Wumpus world in FOL
- Knowledge engineering in FOL
- Required Reading:
  - All of Chapter 8

Pros and cons of propositional logic

- ☺ Propositional logic is declarative
  - -programming languages lack general mechanism for deriving facing from other facts
  - Update to data structure is domain specific
  - Knowledge and inference are separate
- ☺ Propositional logic allows partial/disjunctive/negated information
  - unlike most programming languages 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
  - Look, here comes superman.
- ☹ Propositional logic has limited expressive power
  - unlike natural language
  - E.g., cannot say "pits cause breezes in adjacent squares“
    - except by writing one sentence for each square

Wumpus World and propositional logic

- Find Pits in Wumpus world
  - $B_{x,y} \Leftrightarrow (P_{x,y+1} \vee P_{x,y-1} \vee P_{x+1,y} \vee P_{x-1,y})$ (Breeze next to Pit) 16 rules
- Find Wumpus
  - $S_{x,y} \Leftrightarrow (W_{x,y+1} \vee W_{x,y-1} \vee W_{x+1,y} \vee W_{x-1,y})$ (stench next to Wumpus) 16 rules
- At least one Wumpus in world
  - $W_{1,1} \vee W_{1,2} \vee \ldots \vee W_{4,4}$ (at least 1 Wumpus) 1 rule
- At most one Wumpus
  - $\neg W_{1,1} \vee \neg W_{1,2}$ (155 RULES)
- Keep track of location
  - $L_{x,y} \wedge FacingRight \wedge Forward \Rightarrow L_{x+1,y}$

First-Order Logic

- 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, …

Logics in General

- Ontological Commitment:
  - What exists in the world — TRUTH
  - PL : facts hold or do not hold.
  - FOL : objects with relations between them that hold or do not hold

Language	Ontological Commitment	Epistemological Commitment
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 $\in [0, 1]$
Fuzzy logic	degree of truth $\in [0, 1]$	known interval value

Syntax of FOL: Basic elements

- Constant Symbols:
  - Stand for objects
  - e.g., KingJohn, 2, UCI,...
- Predicate Symbols
  - Stand for relations
  - E.g., Brother(Richard, John), greater_than(3,2)...
- Function Symbols
  - Stand for functions
  - E.g., Sqrt(3), LeftLegOf(John),...

Syntax of FOL: Basic elements

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

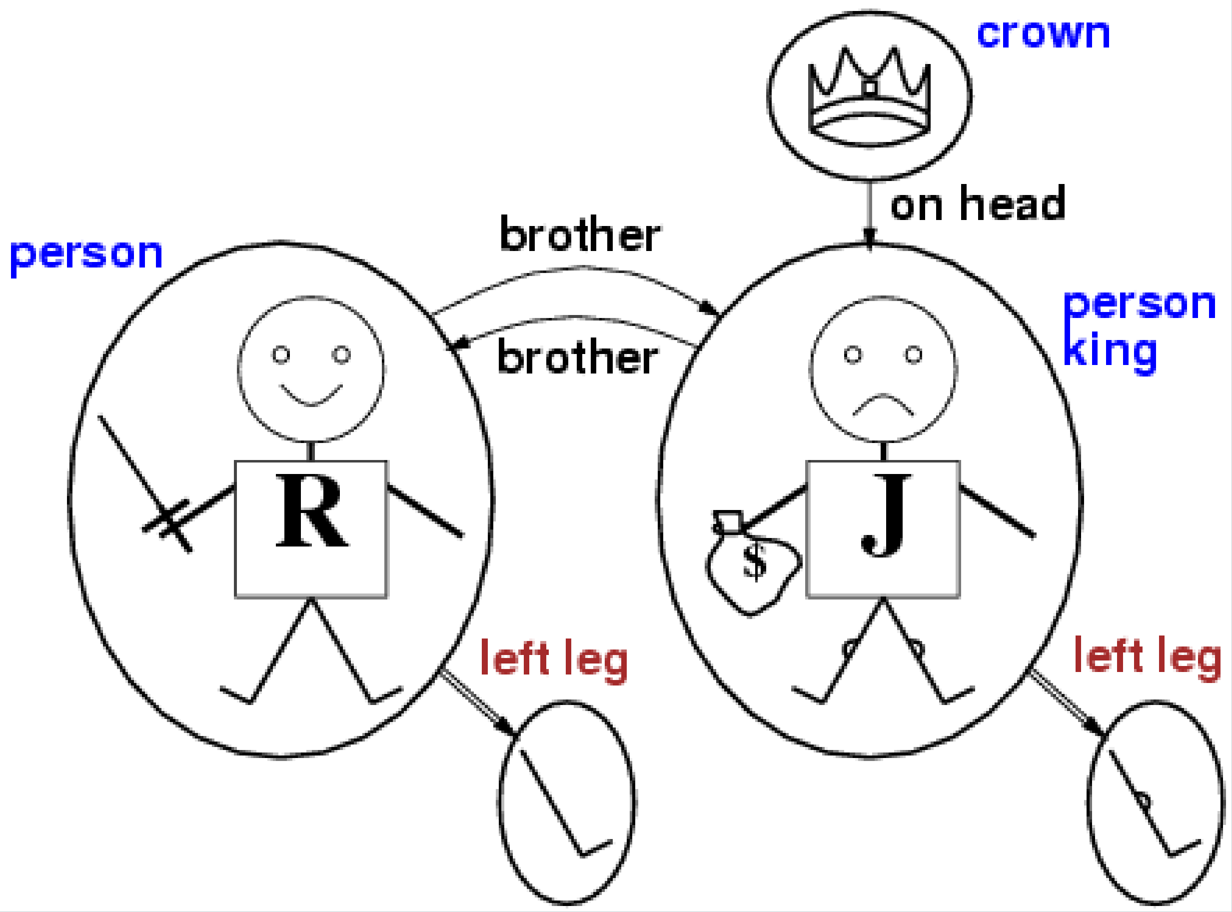
Relations

- Some relations are properties: they state
  some fact about a single object: Round(ball), Prime(7).

- n-ary relations state facts about two or more objects: Married(John,Mary), LargerThan(3,2).

- Some relations are functions: their value is another object: Plus(2,3), Father(Dan).

Models for FOL: Example



Terms

- Term = logical expression that refers to an object.
- There are 2 kinds of terms:
  - constant symbols: Table, Computer
  - function symbols: LeftLeg(Pete), Sqrt(3), Plus(2,3) etc

Atomic Sentences

- Atomic sentences state facts using terms and predicate symbols
  - P(x,y) interpreted as “x is P of y”

- Examples:

  LargerThan(2,3) is false.

  Brother_of(Mary,Pete) is false.

  Married(Father(Richard), Mother(John)) could be true or false

- Note: Functions do not state facts and form no sentence:
  - Brother(Pete) refers to John (his brother) and is neither true nor false.

- Brother_of(Pete,Brother(Pete)) is True.

Binary relation Function

Complex Sentences

- We make complex sentences with connectives (just like in propositional logic).

property

$$\neg Brother(LeftLeg(Richard), John) \vee (Democrat(Bush))$$

binary relation

function

objects

connectives

More Examples

- Brother(Richard, John) $\wedge$ Brother(John, Richard)
- King(Richard) $\vee$ King(John)
- King(John) => $\neg$ King(Richard)
- LessThan(Plus(1,2) ,4) $\wedge$ GreaterThan(1,2)

(Semantics are the same as in propositional logic)

Variables

- Person(John) is true or false because we give it a single argument 'John'

- We can be much more flexible if we allow variables which can take on values in a domain. e.g., all persons x, all integers i, etc.
  - E.g., can state rules like Person(x) => HasHead(x)

    or Integer(i) => Integer(plus(i,1))

Universal Quantification ∀

- ∀ means "for all"
- Allows us to make statements about all objects that have certain properties
- Can now state general rules:

$\forall$ x King(x) => Person(x)

$\forall$ x Person(x) => HasHead(x)

$\forall$ i Integer(i) => Integer(plus(i,1))

Note that
$\forall$ x King(x) $\wedge$ Person(x) is not correct!
This would imply that all objects x are Kings and are People

$\forall$ x King(x) => Person(x) is the correct way to say this

Existential Quantification ∃

- ∃ x means "there exists an x such that…." (at least one object x)
- Allows us to make statements about some object without naming it
- Examples:

$\exists x \quad King(x)$

$\exists x \quad Lives\_in(John, Castle(x))$

$\exists i \quad Integer(i) \wedge GreaterThan(i,0)$

Note that $\wedge$ is the natural connective to use with $\exists$

(And => is the natural connective to use with $\forall$)

More examples

For all real x, x>2 implies x>3.

$$\forall x[(x > 2) \Rightarrow (x > 3)] \quad x \in R \quad (false)$$

$$\exists x[(x^2 = -1)] \quad x \in R \quad (false)$$

There exists some real x whose square is minus 1.

Fun with sentences

Brothers are siblings

Fun with sentences

Brothers are siblings

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

"Sibling" is symmetric

Fun with 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

Fun with 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

Fun with 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

$\forall\, x, y \;\; FirstCousin(x, y) \;\Leftrightarrow\; \exists\, p, ps \;\; Parent(p, x) \land Sibling(ps, p) \land Parent(ps, y)$

Combining Quantifiers

$\forall x \exists y$ Loves(x,y)

- For everyone ("all x") there is someone ("y") who loves them

$\exists y \forall x$ Loves(x,y)

- there is someone ("y") who loves everyone

Clearer with parentheses: $\exists y \; (\forall x \quad \text{Loves}(x,y) \;)$

Connections between Quantifiers

- Asserting that all x have property P is the same as asserting that does not exist any x that does't have the property P

$\forall x \ \text{Likes}(x, 271 \text{ class}) \Leftrightarrow \neg \exists x \ \neg \text{Likes}(x, 271 \text{ class})$

In effect:

- $\forall$ is a conjunction over the universe of objects
- $\exists$ is a disjunction over the universe of objects

Thus, DeMorgan's rules can be applied

De Morgan's Law for Quantifiers

De Morgan's Rule

$$P \wedge Q \equiv \neg(\neg P \vee \neg Q)$$

$$P \vee Q \equiv \neg(\neg P \wedge \neg Q)$$

$$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$$

$$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$$

Generalized De Morgan's Rule

$$\forall x P \equiv \neg \exists x (\neg P)$$

$$\exists x P \equiv \neg \forall x (\neg P)$$

$$\neg \forall x P \equiv \exists x (\neg P)$$

$$\neg \exists x P \equiv \forall x (\neg P)$$

Rule is simple: if you bring a negation inside a disjunction or a conjunction, always switch between them (or →and, and → or).

Using FOL

- We want to TELL things to the KB, e.g.

  TELL(KB, $\forall x, King(x) \Rightarrow Person(x)$)

  TELL(KB, *King(John)*)

  These sentences are assertions

- We also want to ASK things to the KB,

  ASK(KB, $\exists x, Person(x)$)

  these are queries or goals

  The KB should Person(x) is true: {x/John,x/Richard,...}

FOL Version of Wumpus World

- Typical percept sentence:
  Percept([Stench,Breeze,Glitter,None,None],5)

- Actions:
  Turn(Right), Turn(Left), Forward, Shoot, Grab, Release, Climb

- To determine best action, construct query:

  $\forall$ a BestAction(a,5)

- ASK solves this and returns {a/Grab}
  - And TELL about the action.

Knowledge Base for Wumpus World

- Perception
  - $\forall s,g,t$ Percept([s, Breeze,g],t) $\Rightarrow$ Breeze(t)
  - $\forall s,b,t$ Percept([s,b,Glitter],t) $\Rightarrow$ Glitter(t)
- Reflex
  - $\forall t$ Glitter(t) $\Rightarrow$ BestAction(Grab,t)
- Reflex with internal state
  - $\forall t$ Glitter(t) $\wedge \neg$Holding(Gold,t) $\Rightarrow$ BestAction(Grab,t)

Holding(Gold,t) can not be observed: keep track of change.

Deducing hidden properties

Environment definition:

$\forall$x,y,a,b *Adjacent*([x,y],[a,b]) $\Leftrightarrow$ [a,b] $\in$ {[x+1,y], [x-,y],[x,y+1],[x,y-1]}

Properties of locations:

$\forall$s,t *At*(Agent,s,t) $\wedge$ Breeze(t) $\Rightarrow$ Breezy(s)

Location *s* and time *t*

Squares are breezy near a pit:

- Diagnostic rule---infer cause from effect

  $\forall$s Breezy(s) $\Leftrightarrow$ $\exists$ r Adjacent(r,s) $\wedge$ Pit(r)

- Causal rule---infer effect from cause (model based reasoning)

  $\forall$r Pit(r) $\Rightarrow$ [$\forall$s Adjacent(r,s) $\Rightarrow$ Breezy(s)]

Set Theory in First-Order Logic

Can we define set theory using FOL?

- individual sets, union, intersection, etc

Answer is yes.

Basics:

- empty set = constant = { } and elements x, y …
- unary predicate Set(S), true for sets

- binary predicates:

member(x,s)	$x \in S$	(true if x is a member of the set x)
subset(s_1,s_2)	$S_1 \subseteq S_2$	(true if s1 is a subset of s2)

Set Theory in First-Order Logic

- binary functions:

Intersect(s_1,s_2) $s_1 \cap s_2$

Union(s_1,s_2) $s_1 \cup s_2$

Adjoin(x,s) adding x to set s $\{x|s\}$

The only sets are the empty set and sets made by adjoining an element to a set

$\forall s\ Set(s) \Leftrightarrow (s = \{\}\ ) \vee (\exists x,s_2\ Set(s_2) \wedge s = Adjoin(x, s_2))$

The empty set has no elements adjoined to it

$\neg\exists x,s\ \ Adjoin(x, s) = \{\}$

A Possible Set of FOL Axioms for Set Theory

Adjoining an element already in the set has no effect

$\forall x,s \ \ member(x,s) \Leftrightarrow s = Adjoin(x, s)$

A set is a subset of another set iff all the first set's members are members of the 2nd set

$\forall s_1,s_2 \ \ subset(s_1,s_2) \Leftrightarrow (\forall x \ \ member(x ,s_1) \Rightarrow member(x , s_2)$

Two sets are equal iff each is a subset of the other

$\forall s_1,s_2 \ (s_1 = s_2) \Leftrightarrow (subset(s_1,s_2) \wedge subset(s_2 , s_1))$

A Possible Set of FOL Axioms for Set Theory

An object is in the intersection of 2 sets only if a member of both

$\forall x,s_1,s_2\ x \in \text{intersect}(s_1 , s_2) \Leftrightarrow (\text{member}(x ,s_1) \wedge \text{member}(x ,s_2))$

An object is in the union of 2 sets only if a member of either

$\forall x,s_1,s_2\ x \in \text{union}(s_1 , s_2) \Leftrightarrow (\text{member}(x ,s_1) \vee \text{member}(x ,s_2))$