Last update: March 4, 2010

First-order logic

CMSC 421: Chapter 8 and Section 10.3

Pros and cons of propositional logic

+ Propositional logic is **declarative**: pieces of syntax correspond to facts

+ 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
  E.g., cannot say "pits cause breezes in adjacent squares"
  except by writing 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

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 . . .,
  is the brother of, is bigger than, is inside, is part of, has color, occurred after, owns, comes between, . . .
- *Functions*: father of, best friend, third inning of, one more than, end of . . .

Logics in general

Language	Ontological commitment* (what it talks about)	Epistemological commitment* (what it says about truth)
Prop. 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

*To philosophers, these mean roughly the following:

ontological commitment $\approx$ our assumptions about what things exist

epistemological commitment $\approx$ what we can know about those things

Syntax of FOL: Basic elements

Constant symbols	$KingJohn,\ 2,\ UniversityofMaryland, \ldots$
Predicate symbols	$Brother,\ >, \ldots$
Function symbols	$Sqrt,\ LeftLegOf, \ldots$
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.,

$Brother(KingJohn, RichardTheLionheart)$

$> (Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))$

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.,

$$Sibling(KingJohn, Richard) \;\Rightarrow\; Sibling(Richard, KingJohn)$$

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

$$>(1,2) \;\wedge\; \neg{>}(1,2)$$

Truth in first-order logic

In FOL, a *model* is a pair $M = (D, I)$, where D is a *domain* and I is an *interpretation*

D contains ≥ 1 objects (*domain elements*)
and relations among them

I specifies referents for
constant symbols $\rightarrow$ objects in the domain
predicate symbols $\rightarrow$ relations over objects in the domain
function symbols $\rightarrow$ functional relations over objects in the domain

Recall that mathematically, a *relation* is a set of ordered n-tuples

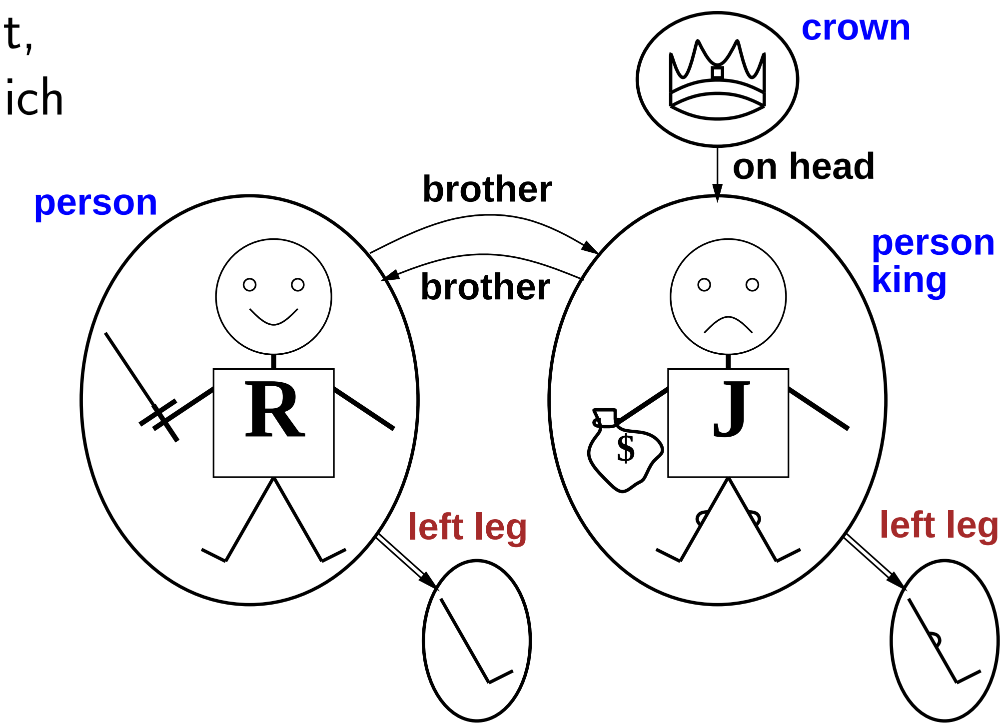
An atomic sentence $predicate(term_1, \ldots, term_n)$ is true in M
iff the objects referred to by $term_1, \ldots, term_n$
are in the relation referred to by $predicate$

Like 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 at right,
and 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 M iff the pair consisting of
Richard the Lionheart and the evil King John
is in the brotherhood relation

Models for FOL: Lots!

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

How to enumerate possible worlds in FOL?

For each number of domain elements n from 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:
$\forall x \;\; At(x, UMD) \;\Rightarrow\; Smart(x)$

$\forall x \;\; P$ is true in a model m iff P is true with x being **each** possible object in the model

Roughly equivalent to the conjunction of instantiations of P

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

A common mistake to avoid

Common mistake with $\forall$:

using $\wedge$ when you meant to use $\Rightarrow$

$$\forall x \;\; At(x, UMD) \wedge Smart(x)$$

means "Everyone is at UMD and everyone is smart"

Probably you meant to say

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

Everyone at UMD is smart.

Existential quantification

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

Someone at UMD is smart:
$\exists x \;\; At(x, UMD) \land Smart(x)$

$\exists x \;\; P$ is true in a model m iff P is true with x being **some** possible object in the model

Roughly equivalent to the disjunction of instantiations of P

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

Another common mistake to avoid

A common mistake with $\exists$:
using $\Rightarrow$ when you meant to use $\wedge$:

$$\exists\, x \;\; At(x, UMD) \;\Rightarrow\; Smart(x)$$

This is equivalent to

$$\exists\, x \;\; \neg At(x, UMD) \vee Smart(x)$$

There's someone who either is smart or isn't at UMD.

That's true if there's anyone who is not at UMD.

Probably you meant to say this instead:

$$\exists\, x \;\; At(x, UMD) \wedge Smart(x)$$

There's someone who is at UMD and is smart.

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

$\forall x\ \ Likes(x, IceCream) \qquad \neg\exists x\ \ \neg Likes(x, IceCream)$

$\exists x\ \ Likes(x, Broccoli) \qquad \neg\forall x\ \ \neg Likes(x, Broccoli)$

Examples of sentences

Brothers are siblings

Examples of sentences

Brothers are siblings

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

"Sibling" is symmetric

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

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

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

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

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

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

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

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

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., $1 = 2$ 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)

E.g., definition of $Sibling$ in terms of $Parent$:

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

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 try to write $\sigma = \{x \leftarrow 1, y \leftarrow f(z)\}$

Given a sentence S and a substitution σ,
$S\sigma$ (postfix notation) is the result of applying σ to S

$$S = \; GreaterThan(x, y)$$
$$\sigma = \{x \leftarrow 1, y \leftarrow f(z)\}$$
$$S\sigma = \; GreaterThan(1, f(z))$$

The substitutions are performed simultaneously like `let`,
not sequentially like `let*`

$$S = \; GreaterThan(x, y)$$
$$\sigma = \{x \leftarrow 2, y \leftarrow g(x)\}$$
$$S\sigma = \; GreaterThan(2, g(x))$$

Interacting with FOL KBs

Suppose an agent has an FOL KB of
axioms for how the Wumpus world works

A model of the KB consists of a domain and interpretation (e.g., an actual Wumpus World) that makes every sentence in the KB true

Suppose we have a way to do inference in the FOL KB (see next chapter)

Suppose the agent perceives a smell and a breeze (but no glitter) at $t = 5$:

$Tell(KB, Percept([Smell, Breeze, None], 5))$
$Ask(KB, \exists\, a\ \ Action(a, 5))$

I.e., does KB entail any particular actions at $t = 5$?

Answer: $Yes,\ \{a \leftarrow Shoot\}$ $\leftarrow$ *substitution*

$Ask(KB, S)$ returns some/all σ such that $KB \models S\sigma$

Knowledge base for the wumpus world

"Perception"
$\forall\, b, g, t \;\; Percept([Smell, b, g], t) \;\Rightarrow\; Smelled(t)$
$\forall\, s, b, t \;\; Percept([s, b, Glitter], t) \;\Rightarrow\; AtGold(t)$

Reflex: $\forall\, t \;\; AtGold(t) \;\Rightarrow\; Action(Grab, t)$

Reflex with internal state: do we have the gold already?
$\forall\, t \;\; AtGold(t) \wedge \neg Holding(Gold, t) \;\Rightarrow\; Action(Grab, t)$

$Holding(Gold, t)$ cannot be observed
$\Rightarrow$ keeping track of change is essential

Deducing hidden properties

Properties of locations:
$\forall x, t \;\; At(Agent, x, t) \wedge Smelled(t) \;\Rightarrow\; Smelly(x)$
$\forall x, t \;\; At(Agent, x, t) \wedge Breeze(t) \;\Rightarrow\; Breezy(x)$

Squares are breezy near a pit:

Diagnostic rule—infer cause from effect

$$\forall y \;\; Breezy(y) \;\Rightarrow\; \exists x \;\; Pit(x) \wedge Adjacent(x, y)$$

Causal rule—infer effect from cause

$$\forall x, y \;\; Pit(x) \wedge Adjacent(x, y) \;\Rightarrow\; Breezy(y)$$

Neither of these is complete—e.g., the causal rule doesn't say whether squares far away from pits can be breezy

Definition for the $Breezy$ predicate:

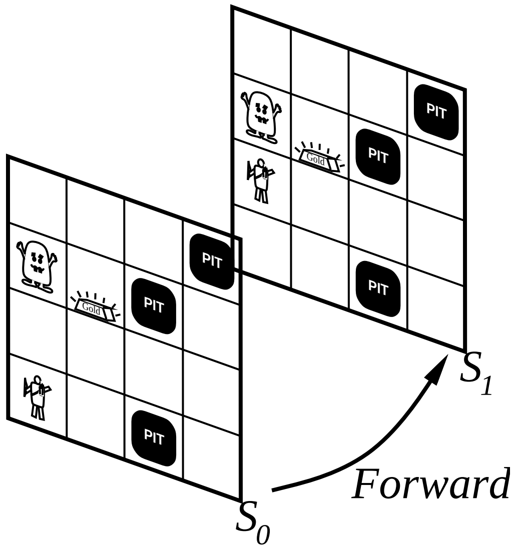
$$\forall y \;\; Breezy(y) \;\Leftrightarrow\; [\exists x \;\; Pit(x) \wedge Adjacent(x, y)]$$

Keeping track of change

Facts hold in *situations*, rather than eternally
E.g., truth of $Holding(Gold)$ depends on whether we've grabbed the gold

Situation calculus is one way to represent change in FOL:
Adds a situation argument to each non-eternal predicate
e.g., $Holding(Gold, s)$ to mean we're holding the gold in situation s

Situations are connected by the $Result$ function
$Result(a, s)$ is the situation that results from doing a in s



$= Result(Forward, S_0)$

Describing actions I

"Effect" axiom—describe changes due to action

$\forall\, s \;\; AtGold(s) \;\Rightarrow\; Holding(Gold, Result(Grab, s))$

Is this the only effect?

Describing actions I

"Effect" axiom—describe changes due to action
$\forall s \;\; AtGold(s) \;\Rightarrow\; Holding(Gold, Result(Grab, s))$

"Frame" axiom—describe **non-changes** due to action
$\forall s \;\; HaveArrow(s) \;\Rightarrow\; HaveArrow(Result(Grab, s))$

Frame problem: find a way to handle non-change

Qualification problem: true descriptions of real actions require endless caveats—what if gold is slippery or nailed down or . . .

Ramification problem: real actions have many secondary consequences—what about the dust on the gold, wear and tear on gloves, . . .

The Wumpus world is so simple that it's easy to specify all the qualifications and ramifications of each action

But we still need to handle the frame problem

Describing actions II

One way to handle the frame problem is to use *successor-state axioms*

One axiom for each predicate P:

$$\text{“}P\text{ true afterwards”} \Leftrightarrow [\text{“an action made } P \text{ true”} \lor \text{“}P\text{ true already and no action made } P\text{ false”}]$$

$$\text{“}P\text{ true in } Result(a,s)\text{”} \Leftrightarrow [\,a = a_1 \lor a = a_2 \lor \ldots \lor a = a_m \lor (\text{“}P\text{ true in } s\text{”} \land a \neq b_1 \land a \neq b_2 \land \ldots \land b \neq b_n)]$$

where $a_1, \ldots, a_m$ are all of the actions (plus preconditions if needed) that can make P true

and $b_1, \ldots, b_n$ are all of the actions that can make P false

For holding the gold:

$$\forall a \;\; \forall s \;\; Holding(Gold, Result(a,s)) \Leftrightarrow [(a = Grab \land AtGold(s)) \lor (Holding(Gold,s) \land a \neq Release)]$$

Making plans

Initial condition in KB:

$At(Agent, [1,1], S_0)$
$At(Gold, [1,2], S_0)$

Query: $Ask(KB, \exists\, s \;\; Holding(Gold, s))$

i.e., is there a situation in which I'll be holding the gold?

Answer: $\{s \leftarrow Result(Grab, Result(Forward, S_0))\}$

i.e., go forward and then grab the gold

This notation is awkward. How to improve it?

Better Notation

Extend FOL syntax to use the following notation for lists
This notation comes from the Prolog programming language

$$\begin{aligned} [\,] &= \text{the empty list, like } \texttt{NIL} \text{ in Lisp} \\ [a,b,c] &= \text{list of } a,\ b, \text{ and } c, \text{ like } \texttt{'(}a\ \ b\ \ c\texttt{)} \text{ in Lisp} \\ [a|y] &= \text{list whose head is } a \text{ and tail is } y, \text{ like } \texttt{(cons}\ \ a\ \ y\texttt{)} \text{ in Lisp} \end{aligned}$$

Represent plans as action sequences $[a_1, a_2, \ldots, a_n]$

Define $PlanResult(p, s)$ to be the result of executing p in s:

$\forall\, s\ \ PlanResult([\,], s) = s$

$\forall\, a, p, s\ \ PlanResult([a|p], s) = PlanResult(p, Result(a, s))$

Then this query

$Ask(KB, \exists\, p\ \ Holding(Gold, PlanResult(p, S_0)))$

has this solution:

$\{p \leftarrow [Forward, Grab]\}$

Planning

Planning: find a set of actions to achieve goal or perform task

We just discussed a special case called *classical planning*:
Find sequence of actions to achieve goal, subject to a bunch of assumptions

- ◇ time is a sequence of instants $0, 1, 2, \ldots, n$
- ◇ at each instant i, a state of the world $s_i = Result(a_i, s_{i-1})$
- ◇ we know the initial state s_0 and goal condition g
- ◇ actions have no duration
- ◇ no change ever occurs in the world, except for our agent's actions
- ◇ actions have deterministic outcomes
- ◇ we can infer exactly what each action will do

  ...

Can do classical planning using the situation calculus, but it's very inefficient

There are many *planning algorithms* for doing it more efficiently
  See Chapter 11

Summary

First-order logic:
– objects and relations are semantic primitives
– syntax: constants, functions, predicates, equality, quantifiers

Increased expressive power: sufficient to define wumpus world

Situation calculus:
– conventions for describing actions and change in FOL
– can formulate and solve planning problems (inefficiently)
as inference on a situation calculus KB

Homework assignment

Problems 7.5, 8.4, 8.5, 8.13

10 points each, 40 points total

Due on Thursday, March 25 (after spring break)