

Last update: October 23, 2008

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

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

Syntax of FOL: Basic elements

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

Atomic sentences

Atomic sentence = $\textit{predicate}(\textit{term}_1, \dots, \textit{term}_n)$
or $\textit{term}_1 = \textit{term}_2$

Term = $\textit{function}(\textit{term}_1, \dots, \textit{term}_n)$
or $\textit{constant}$ or $\textit{variable}$

E.g.,

$\textit{Brother}(\textit{KingJohn}, \textit{RichardTheLionheart})$

$> (\textit{Length}(\textit{LeftLegOf}(\textit{Richard})), \textit{Length}(\textit{LeftLegOf}(\textit{KingJohn})))$

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.,

$$\textit{Sibling}(\textit{KingJohn}, \textit{Richard}) \Rightarrow \textit{Sibling}(\textit{Richard}, \textit{KingJohn})$$

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

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

Truth in first-order logic

Sentences are true with respect to a **domain** and an **interpretation**

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

Interpretation specifies referents for

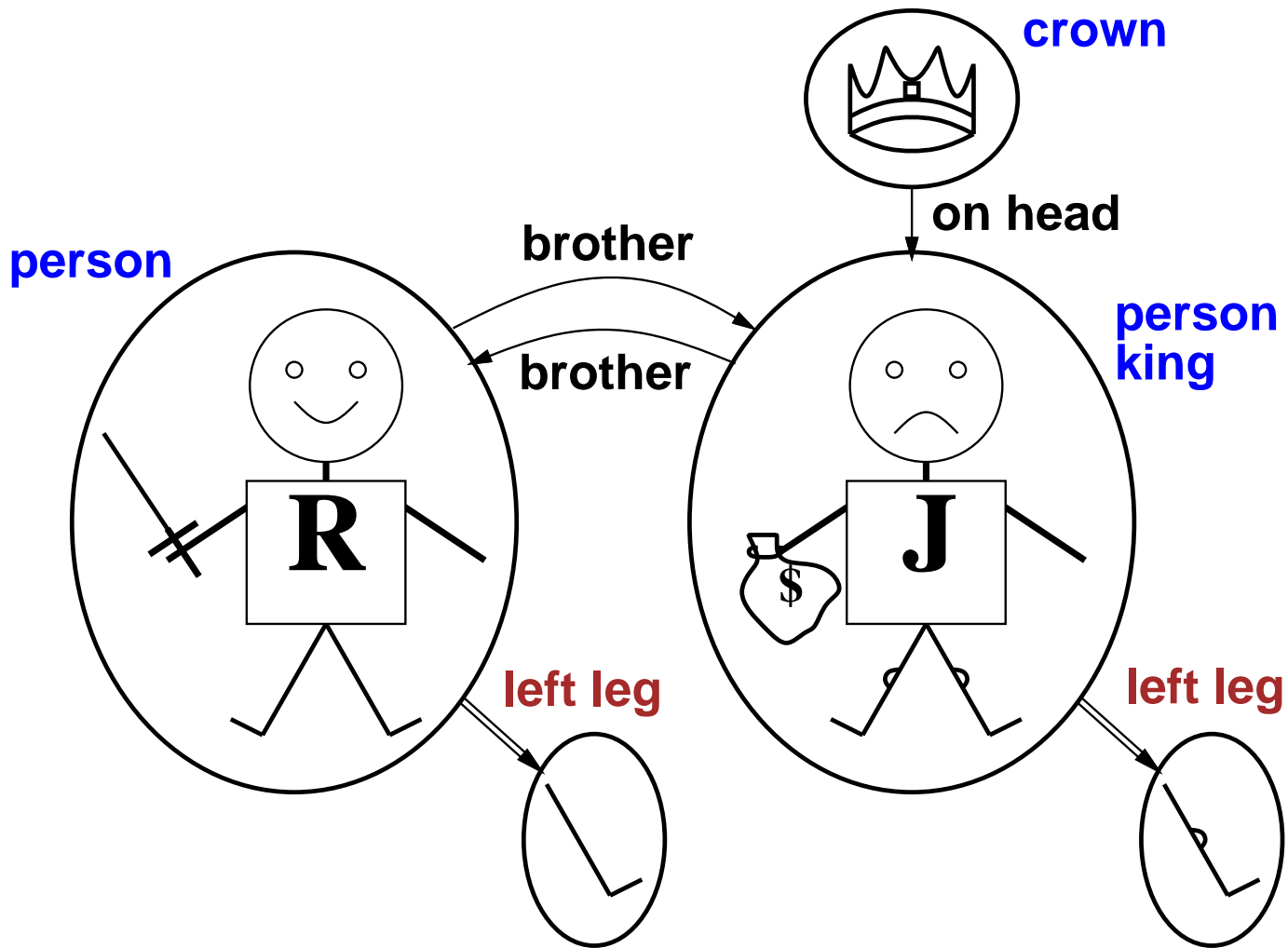
constant symbols \rightarrow **objects**

predicate symbols \rightarrow **relations**

function symbols \rightarrow **functional relations**

An atomic sentence $\textit{predicate}(\textit{term}_1, \dots, \textit{term}_n)$ is true
iff the **objects** referred to by $\textit{term}_1, \dots, \textit{term}_n$
are in the **relation** referred to by $\textit{predicate}$

Models for FOL: Example



Truth example

Consider the interpretation in which

Richard → Richard the Lionheart

John → the evil King John

Brother → the brotherhood relation

Under this interpretation, $Brother(Richard, John)$ is true just in case Richard the Lionheart and the evil King John are in the brotherhood relation in the model

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

$\forall \langle \text{variables} \rangle \langle \text{sentence} \rangle$

Everyone at the University of Maryland is smart:

$\forall x \text{ At}(x, UMD) \Rightarrow \text{Smart}(x)$

$\forall x P$ 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 P

$(\text{At}(\text{KingJohn}, UMD) \Rightarrow \text{Smart}(\text{KingJohn}))$
 $\wedge (\text{At}(\text{Richard}, UMD) \Rightarrow \text{Smart}(\text{Richard}))$
 $\wedge (\text{At}(UMD, UMD) \Rightarrow \text{Smart}(UMD))$
 $\wedge \dots$

A common mistake to avoid

Common mistake with \forall :

using \wedge when you meant to use \Rightarrow

$$\forall x \text{ At}(x, UMD) \wedge \text{Smart}(x)$$

means “Everyone is at UMD and everyone is smart”

Probably you meant to say

$$\forall x \text{ At}(x, UMD) \Rightarrow \text{Smart}(x)$$

Everyone at UMD is smart.

Existential quantification

$\exists \langle \text{variables} \rangle \langle \text{sentence} \rangle$

Someone at UMD is smart:

$\exists x \text{ At}(x, \text{UMD}) \wedge \text{Smart}(x)$

$\exists x P$ 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

$(\text{At}(\text{KingJohn}, \text{UMD}) \wedge \text{Smart}(\text{KingJohn}))$
 $\vee (\text{At}(\text{Richard}, \text{UMD}) \wedge \text{Smart}(\text{Richard}))$
 $\vee (\text{At}(\text{UMD}, \text{UMD}) \wedge \text{Smart}(\text{UMD}))$
 $\vee \dots$

Another common mistake to avoid

A common mistake with \exists :

using \Rightarrow when you meant to use \wedge :

$$\exists x \text{ At}(x, \text{UMD}) \Rightarrow \text{Smart}(x)$$

This is equivalent to

$$\exists x \neg \text{At}(x, \text{UMD}) \vee \text{Smart}(x)$$

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:

$$\exists x \text{ At}(x, \text{UMD}) \wedge \text{Smart}(x)$$

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 $\forall y \exists x$

$\exists x \forall y \text{ Loves}(x, y)$

“There is a person who loves everyone in the world”

$\forall y \exists x \text{ Loves}(x, y)$

“Everyone in the world is loved by at least one person”

Quantifier duality: each can be expressed using the other

$\forall x \text{ Likes}(x, \text{IceCream}) \quad \neg \exists x \neg \text{Likes}(x, \text{IceCream})$

$\exists x \text{ Likes}(x, \text{Broccoli}) \quad \neg \forall x \neg \text{Likes}(x, \text{Broccoli})$

Examples of sentences

Brothers are siblings

Examples of sentences

Brothers are siblings

$\forall x, y \text{ Brother}(x, y) \Rightarrow \text{Sibling}(x, y).$

“Sibling” is symmetric

Examples of sentences

Brothers are siblings

$$\forall x, y \text{ Brother}(x, y) \Rightarrow \text{Sibling}(x, y).$$

“Sibling” is symmetric

$$\forall x, y \text{ Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x).$$

One's mother is one's female parent

Examples of sentences

Brothers are siblings

$$\forall x, y \text{ Brother}(x, y) \Rightarrow \text{Sibling}(x, y).$$

“Sibling” is symmetric

$$\forall x, y \text{ Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x).$$

One's mother is one's female parent

$$\forall x, y \text{ Mother}(x, y) \Leftrightarrow (\text{Female}(x) \wedge \text{Parent}(x, y)).$$

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

Examples of sentences

Brothers are siblings

$$\forall x, y \text{ Brother}(x, y) \Rightarrow \text{Sibling}(x, y).$$

“Sibling” is symmetric

$$\forall x, y \text{ Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x).$$

One's mother is one's female parent

$$\forall x, y \text{ Mother}(x, y) \Leftrightarrow (\text{Female}(x) \wedge \text{Parent}(x, y)).$$

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

$$\forall x, y \text{ FirstCousin}(x, y) \Leftrightarrow \exists p, ps \text{ Parent}(p, x) \wedge \text{Sibling}(ps, p) \wedge \text{Parent}(ps, y)$$

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
 $2 = 2$ is valid

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

$$\forall x, y \text{ Sibling}(x, y) \Leftrightarrow [\neg(x = y) \wedge \exists m, f \neg(m = f) \wedge \\ \text{Parent}(m, x) \wedge \text{Parent}(f, x) \wedge \text{Parent}(m, y) \wedge \text{Parent}(f, y)]$$

Substitutions

Substitution: a set of variable bindings

Suppose we want to talk about 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 $\sigma = \{x = 1, y = f(z)\}$

Given a sentence S and a substitution σ ,

$S\sigma$ (postfix notation) is the result of applying σ to S

$S = \text{GreaterThan}(x, y)$

$\sigma = \{x = 1, y = f(z)\}$

$S\sigma = \text{GreaterThan}(1, f(z))$

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

$S = \text{GreaterThan}(x, y)$

$\sigma = \{x = 2, y = g(x)\}$

$S\sigma = \text{GreaterThan}(2, g(x))$

Interacting with FOL KBs

Suppose a wumpus-world agent is using an FOL KB that contains axioms for how the Wumpus world works

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: $Yes, \{a = Shoot\}$ ← substitution

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

Knowledge base for the wumpus world

“Perception”

$\forall b, g, t \text{ Percept}([Smell, b, g], t) \Rightarrow Smelled(t)$

$\forall s, b, t \text{ Percept}([s, b, Glitter], t) \Rightarrow AtGold(t)$

Reflex: $\forall t \text{ AtGold}(t) \Rightarrow \text{Action}(Grab, t)$

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

$\forall t \text{ AtGold}(t) \wedge \neg Holding(Gold, t) \Rightarrow \text{Action}(Grab, t)$

$Holding(Gold, t)$ cannot be observed

\Rightarrow keeping track of change is essential

Deducing hidden properties

Properties of locations:

$$\forall x, t \text{ At}(\text{Agent}, x, t) \wedge \text{Smelled}(t) \Rightarrow \text{Smelly}(x)$$

$$\forall x, t \text{ At}(\text{Agent}, x, t) \wedge \text{Breeze}(t) \Rightarrow \text{Breezy}(x)$$

Squares are breezy near a pit:

Diagnostic rule—infer cause from effect

$$\forall y \text{ Breezy}(y) \Rightarrow \exists x \text{ Pit}(x) \wedge \text{Adjacent}(x, y)$$

Causal rule—infer effect from cause

$$\forall x, y \text{ Pit}(x) \wedge \text{Adjacent}(x, y) \Rightarrow \text{Breezy}(y)$$

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:

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

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

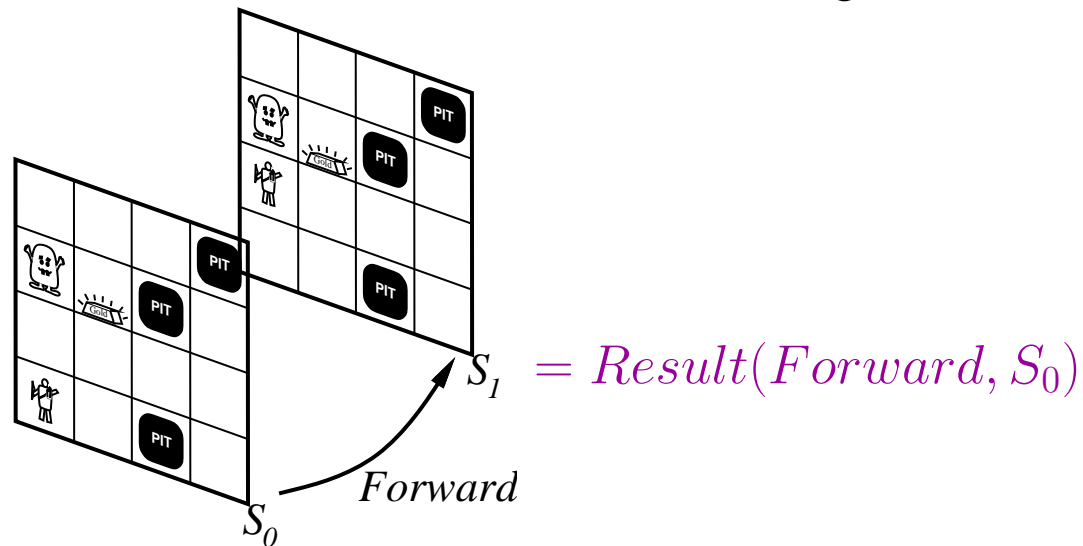
Situation calculus is one way to represent change in FOL:

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*



Describing actions I

“Effect” axiom—describe changes due to action

$$\forall s \text{ AtGold}(s) \Rightarrow \text{Holding}(\text{Gold}, \text{Result}(\text{Grab}, s))$$

Is this the only effect?

Describing actions I

“Effect” axiom—describe changes due to action

$$\forall s \text{ AtGold}(s) \Rightarrow \text{Holding}(\text{Gold}, \text{Result}(\text{Grab}, s))$$

“Frame” axiom—describe **non-changes** due to action

$$\forall s \text{ HaveArrow}(s) \Rightarrow \text{HaveArrow}(\text{Result}(\text{Grab}, s))$$

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 :

$$\text{"}P \text{ true afterwards"} \Leftrightarrow [\text{"an action made } P \text{ true"} \vee \text{"}P \text{ true already and no action made } P \text{ false"}]$$

$$\text{"}P \text{ true in } Result(a, s)\text{"} \Leftrightarrow [a = a_1 \vee a = a_2 \vee \dots \vee a = a_m \vee (\text{"}P \text{ true in } s" \wedge a \neq b_1 \wedge a \neq b_2 \wedge \dots \wedge a \neq b_n)]$$

where a_1, \dots, a_m are all of the actions that can make P true
and b_1, \dots, b_n are all of the actions that can make P false

For holding the gold:

$$\forall a \forall s \text{ Holding}(\text{Gold}, \text{Result}(a, s)) \Leftrightarrow [(a = \text{Grab} \wedge \text{AtGold}(s)) \vee (\text{Holding}(\text{Gold}, s) \wedge a \neq \text{Release})]$$

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

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

$[]$ = the empty list, like `NIL` in Lisp

$[a, b, c]$ = list of a , b , and c , like `'(a b c)` in Lisp

$[a|y]$ = list whose head is a and tail is y , like `(cons a y)` in Lisp

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

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

$\forall s \text{ PlanResult}([], s) = s$

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

Then this query

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

has this solution:

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

Planning

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

We just discussed a highly restricted case called **classical planning**:

Find sequence of actions to achieve goal, subject to a bunch of assumptions

- ◇ time is a sequence of instants $0, 1, 2, \dots, n$
- ◇ 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 classical planning (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 in one week