

First-order logic

CHAPTER 7

Outline

- ◇ Syntax and semantics of FOL
- ◇ Fun with sentences
- ◇ Wumpus world in FOL

Syntax of FOL: Basic elements

Constants *KingJohn, 2, UCB, ...*

Predicates *Brother, >, ...*

Functions *Sqrt, LeftLegOf, ...*

Variables *x, y, a, b, ...*

Connectives $\wedge \vee \neg \Rightarrow \Leftrightarrow$

Equality $=$

Quantifiers $\forall \exists$

Atomic sentences

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

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

E.g., $\text{Brother}(\text{KingJohn}, \text{RichardTheLionheart})$
 $> (\text{Length}(\text{LeftLegOf}(\text{Richard})), \text{Length}(\text{LeftLegOf}(\text{KingJohn})))$

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

Sentences are true with respect to a model and an interpretation

Model contains objects and relations among them

Interpretation specifies referents for

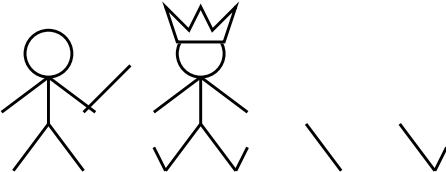
constant symbols \rightarrow objects

predicate symbols \rightarrow relations

function symbols \rightarrow functional relations

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

Models for FOL: Example

objects 

relations: sets of tuples of objects

$\{ \langle \text{king}, \text{queen} \rangle, \langle \text{queen}, \text{king} \rangle, \dots \}$

functional relations: all tuples of objects + "value" object

$\{ \langle \text{king}, \text{queen} \rangle, \langle \text{queen}, \text{king} \rangle, \dots \}$

Universal quantification

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

Everyone at Berkeley is smart:

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

$\forall x \text{ } P$ is equivalent to the conjunction of instantiations of P

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

Typically, \Rightarrow is the main connective with \forall .

Common mistake: using \wedge as the main connective with \forall :

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

means “Everyone is at Berkeley and everyone is smart”

Existential quantification

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

Someone at Stanford is smart:

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

$\exists x P$ is equivalent to the disjunction of instantiations of P

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

Typically, \wedge is the main connective with \exists .

Common mistake: using \Rightarrow as the main connective with \exists :

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

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

Properties of quantifiers

$\forall x \forall y$ is the same as $\forall y \forall x$ (why??)

$\exists x \exists y$ is the same as $\exists y \exists x$ (why??)

$\exists x \forall y$ 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

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

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

Fun with sentences

Brothers are siblings

.

“Sibling” is reflexive

.

One’s mother is one’s female parent

.

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

.

.

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

.

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

.

$$\forall x, y \text{ } \textit{Mother}(x, y) \Leftrightarrow (\textit{Female}(x) \textit{and} \textit{Parent}(x, y))$$

.

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

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 (full) *Sibling* in terms of *Parent*:

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

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($KB, \exists a \text{ Action}(a, 5)$)

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

Answer: *Yes*, $\{a/Shoot\} \leftarrow \underline{\text{substitution}}$ (binding list)

Given a sentence S and a substitution σ ,
 $S\sigma$ denotes the result of plugging σ into S ; e.g.,
 $S = Smarter(x, y)$
 $\sigma = \{x/Hillary, y/Bill\}$
 $S\sigma = Smarter(Hillary, Bill)$

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

Knowledge base for the wumpus world

“Perception”

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

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

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:

$$\forall l, t \quad At(Agent, l, t) \wedge Smelt(t) \Rightarrow Smelly(l)$$

$$\forall l, t \quad At(Agent, l, t) \wedge Breeze(t) \Rightarrow Breezy(l)$$

Squares are breezy near a pit:

Diagnostic rule—infer cause from effect

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

Causal rule—infer effect from cause

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

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:

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

Keeping track of change

Facts hold in situations, rather than eternally

E.g., $Holding(Gold, Now)$ rather than just $Holding(Gold)$

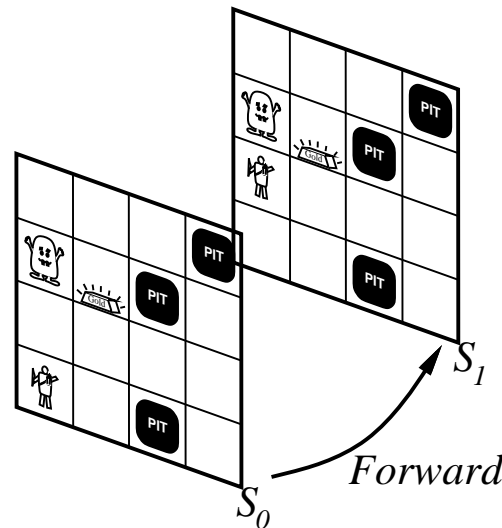
Situation calculus is one way to represent change in FOL:

Adds a situation argument to each non-eternal predicate

E.g., Now in $Holding(Gold, Now)$ denotes a situation

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

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

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

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

Frame problem: find an elegant way to handle non-change

(a) representation—avoid frame axioms

(b) inference—avoid repeated “copy-overs” to keep track of state

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, ...

Describing actions II

Successor-state axioms solve the representational frame problem

Each axiom is “about” a predicate (not an action per se):

$$\begin{aligned} P \text{ true afterwards} \quad \Leftrightarrow \quad & [\text{an action made } P \text{ true} \\ & \vee \quad P \text{ true already and no action made } P \text{ false}] \end{aligned}$$

For holding the gold:

$$\begin{aligned} \forall a, s \quad & Holding(Gold, Result(a, s)) \Leftrightarrow \\ & [(a = Grab \wedge AtGold(s)) \\ & \vee (Holding(Gold, s) \wedge a \neq Release)] \end{aligned}$$

Making plans

Initial condition in KB:

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

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

Query: $ASK(KB, \exists s \text{ Holding}(Gold, s))$

i.e., in what situation will I be holding the gold?

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

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

This assumes that the agent is interested in plans starting at S_0 and that S_0 is the only situation described in the KB

Making plans: A better way

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

$PlanResult(p, s)$ is the result of executing p in s

Then the query $ASK(KB, \exists p \text{ Holding}(Gold, PlanResult(p, S_0)))$ has the solution $\{p/[Forward, Grab]\}$

Definition of $PlanResult$ in terms of $Result$:

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

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

Planning systems are special-purpose reasoners designed to do this type of inference more efficiently than a general-purpose reasoner

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 planning as inference on a situation calculus KB