First-order logic

Chapter 8

First-order logic

Whereas propositional logic assumes world contains **facts**,
first-order logic (like natural language) assumes the world contains:

- Objects: people, houses, numbers colors, . . .
- Relations: red, round, prime. . .,
  brother of, bigger than, inside, part of, has color, occurred after, owns, comes between, . . .
- Functions: father of, best friend, one more than, end of . . .

Syntax of FOL: Basic elements

Constants	$Ali, John, Sabanci,\ 2\ldots$
Variables	$x,\ y,\ a,\ b,\ldots$
Functions	$Sqrt,\ FatherOf,\ldots$
Predicates	$Brother, GoodStudent,\ldots$
Connectives	$\wedge\ \ \vee\ \ \neg\ \ \Rightarrow\ \ \ \Leftrightarrow$
Equality	$=$
Quantifiers	$\forall\ \exists$

Terms and Atomic Sentences

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

Constants:

Ali

Ayse

Variables:

x

y

Function:

Father(Ayse)

Functions vs Predicates

Function:

Father(Aye) - indicates a term

Predicate:

Father(Ahmet,Ayse) - indicates a predicate that has a True or False value

Atomic sentences

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

Married(Ali, Ayse)
Father(Ayse) $=$ Ahmet
Married(Father(Ayse), Mother(Ayse))

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., $Color(object) = Red$

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. $Sibling(Ali, Ayse) \;\Rightarrow\; Sibling(Ayse, Ali)$
$>(x, 0) \wedge <(x, 3)$

Convention:

When we use $P(x, y)$, we will mean "x is a P of y" or "x P y".
In other words we say Father(fathersname, childsname) rather than the other way around.

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 → objects
  predicate symbols → relations
  function symbols → functional relations

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

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.

Universal and Existential Quantification

Once we have objects, FOL lets us express properties of entire collections of objects.

Universal quantification

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

Everyone at Sabanci is smart:
$\forall x \ \ At(x, Sabanci) \Rightarrow Smart(x)$

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

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

$$\begin{aligned} & (Student(Ahmet, Sabanci) \Rightarrow Smart(Ahmet)) \\ \wedge \ & (Student(Mehmet, Sabanci) \Rightarrow Smart(Mehmet)) \\ \wedge \ & (Student(Ayse, Sabanci) \Rightarrow Smart(Ayse)) \\ \wedge \ & \ldots \end{aligned}$$

Typically: $\Rightarrow$ is the main connective with $\forall$.

A common mistake to avoid

Typically: $\Rightarrow$ is the main connective with $\forall$.

Common mistake: using $\land$ as the main connective with $\forall$:

$$\forall x \;\; Student(x, Sabanci) \land Smart(x)$$

means "Everyone is at Sabanci and everyone is smart"

Existential quantification

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

Someone at Koc is smart:
$\exists x \;\; Student(x, Koc) \wedge Smart(x)$

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

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

$$
\begin{aligned}
& (Student(Kemal, Koc) \wedge Smart(Kemal)) \\
\vee\; & (Student(Esra, Koc) \wedge Smart(Esra)) \\
\vee\; & (Student(Funda, Koc) \wedge Smart(Funda)) \\
\vee\; & \ldots
\end{aligned}
$$

Another common mistake to avoid

Typically:, $\wedge$ is the main connective with $\exists$

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

$$\exists\, x\;\; Student(x, Koc) \Rightarrow Smart(x)$$

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

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??)

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 $\forall y \ \exists x$

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

...

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

...

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 $\forall y\ \exists x$

$\exists x\ \forall y\ Loves(x,y)$
"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" "For everyone, there is someone who loves them"

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 $\forall y \;\; \exists x$

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

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

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

Natural Language to Logic

Brothers are siblings (use Brother and Sibling predicates)

Sibling relationship is symmetric (use Sibling predicate)

One's mother is one's female parent (use Mother, Parent and Female predicates) - Note predicate versus Function ...

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

Natural Language to Logic

Brothers are siblings (use Brother and Sibling predicates)

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

Sibling relationship is symmetric (use Sibling predicate)

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

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

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

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 $Parent$:

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

Assumptions

$Brother(John, Richard) \wedge Brother(Geoffrey, Richard)$.

$\diamondsuit$ Unique names assumption: Every constant symbol refer to a distinct object

$\diamondsuit$ Domain closure: Each model contains no more domain elements than those named by a constant symbol.

$\diamondsuit$ Closed world assumption: atomic sentences not known to be true are false

Under what is called **database semantics** (above assumptions), the above sentence conveys our knowledge that "Richard has two brothers".

Note:This is different than the standard semantics of FOL.

Higher-Order Logic

FOL: one can quantify over objects (first order entities that actually exist in the world)

Higher Order Logic: quantify over relations and functions

e.g. $\forall\, x, y \;\; (x = y) \;\Leftrightarrow\; (\forall\, P \;\; P(x) \;\Leftrightarrow\; P(y))$

Using FOL

$Tell(KB, King(John))$
$Tell(KB, Person(Richard))$
$Tell(\forall x \;\; King(x) \;\Leftrightarrow\; Person(x))$.

Now we can ask questions to the knowledgebase using ASK:

$Ask(KB, King(John))$ returns True.

$Ask(KB, Person(John))$ should also return True.

$Ask(KB, \exists x \quad Person(x))$ is True, but not very useful if we don't know who that person is.

$AskVars(KB, Person(x))$ returns a **substitution** in the form of
$\{x/John\}, \{x/Richard\}$.

Substitution in FOL

$AskVars$ is useful in KB that can be written in Horn form in which every way of making a query true will bind the variables to specific values.

Consider a sentence in FOL which is *not* a Horn close:

$King(John) \vee King(Richard)$.

Then $Ask(KB, \exists x \;\; King(x))$ should return True, but there is no binding to x.

KB Design: Axioms, Definitions, Theorems

What to put in a KB?

Axioms: basic facts about a domain (basic predicates)

$\forall\, x, y \;\; Sibling(x, y) \;\Leftrightarrow\; [\neg(x\!=\!y)] \land \exists\, p \;\; (Parent(p, x) \land Parent(p, y))$.

Theorems: facts entailed by axioms (e.g. the theorem below is entailed by definition of the Sibling predicate given above)

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

Theorems are useful to encode important conclusions, to prevent the need to make deductions from scratch.

KB Design: Axioms, Definitions, Theorems

Not all axioms are definitions.

$\forall\, x \;\; Person(x) \;\Leftrightarrow\; \ldots$ (difficult!)

Instead:

$\forall\, x \;\; Person(x) \;\Rightarrow\; \ldots$

$\forall\, x \;\; \ldots \;\Rightarrow\; Person(x)$

Knowledge base for the wumpus world - agent and percepts

Percept:

$Percept([Smell, Breeze, Glitter, none, none], 5)$

where the percept is represented in a list.

Actions: Turn(Right), Turn(Left), Forward,...

To ask for best actions:

$ASKVARS(\exists\, a \;\; BestAction(a, 5))$ which would return a binding list such as $\{$a/Grab$\}$.

Knowledge base for the wumpus world - agent and percepts

Raw percept data implies certain facts about the current state

$\forall\, b, g, t \;\; Percept([Smell, b, g], t) \;\Rightarrow\; Smelt(t)$

$\forall\, s, b, t \;\; Percept([s, b, Glitter], t) \;\Rightarrow\; AtGold(t)$

Simple reflex behavior can be implemented as:

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

Knowledge base for the wumpus world - environment

Agent's location changes over time, so we will use:

$At(Agent, s, t)$

Properties of locations (notice time or location dependence):
$\forall\, s, t\;\; At(Agent, s, t) \wedge Smelt(t) \;\Rightarrow\; Smelly(s)$
$\forall\, s, t\;\; At(Agent, s, t) \wedge Breeze(t) \;\Rightarrow\; Breezy(s)$

Knowledge base for the wumpus world - environment

Now with the power of FOL, we can represent adjacent squares as:

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

Then, we could have a rule such as:

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

Knowledge base for the wumpus world - environment

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:

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

Interacting with FOL KBs

Suppose a wumpus-world agent is using an FOL KB
and perceives a glitter and a breeze at $t = 5$:

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

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

Answer: $Yes,\ \{a/Grab\}$ $\leftarrow$ 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

Simple reflex behaviour can be implemented by quantified implication sentences: Reflex: $\forall\, t \;\; AtGold(t) \;\Rightarrow\; Action(Grab, t)$

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

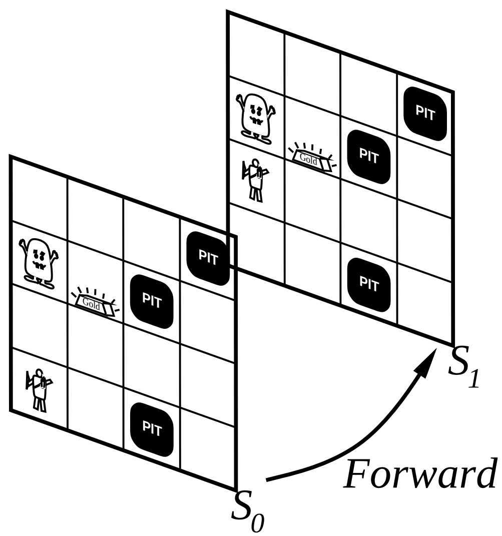
$Holding(Gold, t)$ cannot be observed
$\Rightarrow$ keeping track of change is essential

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

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

"Frame" axiom—describe **non-changes** due to action

$\forall s \;\; HaveArrow(s) \;\Rightarrow\; HaveArrow(Result(Grab, s))$

Frame problem: find 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):

P true afterwards $\Leftrightarrow$ [an action made P true
$\lor$ P true already and no action made P false]

For holding the gold:

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