First-order logic

Chapter 7

Syntax of FOL: Basic elements

- Constants: KingJohn, 2, UCB,...
- Predicates: Brother, >, ...
- Functions: Sqrt, LeftLegOf, ...
- Variables: x, y, a, b, ...
- Connectives: \( \land, \lor, \lnot, \Rightarrow, \Leftrightarrow \)
- Equality: =
- Quantifiers: \( \forall, \exists \)

Atomic sentences

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

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

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

Complex sentences

Complex sentences are made from atomic sentences using connectives

\( \lnot S, S \land S_2, S_1 \lor S_2, S_1 \Rightarrow S_2, S_1 \Leftrightarrow S_2 \)

E.g., Sibling(KingJohn, Richard) \( \Rightarrow \) Sibling(Richard, KingJohn)

\( \lnot (1, 2) \lor \lnot (2, 2) \)

\( (1, 2) \land \lnot (1, 2) \)

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 \( \text{predicate}(\text{term}_1, \ldots, \text{term}_n) \) is true if the objects referred to by \( \text{term}_1, \ldots, \text{term}_n \) are in the relation referred to by \( \text{predicate} \)
Models for FOL: Example

objects

relations: sets of tuples of objects

\{ \langle \text{John}, \text{Berkeley} \rangle, \langle \text{Richard}, \text{Berkeley} \rangle, \ldots \} \]

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

\{ \langle \text{John}, \text{Berkeley} \rangle, \langle \text{Richard}, \text{Berkeley} \rangle, \ldots \} \]

Universal quantification

\( \forall \text{variables} \) (sentence)

Everyone at Berkeley is smart:

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

\( \forall x \, P \) is equivalent to the conjunction of instantiations of \( P \)

\( \text{At}(\text{KingJohn}, \text{Berkeley}) \Rightarrow \text{Smart}(\text{KingJohn}) \)

\( \land \text{At}(\text{Richard}, \text{Berkeley}) \Rightarrow \text{Smart}(\text{Richard}) \)

\( \land \text{At}(\text{Berkeley}, \text{Berkeley}) \Rightarrow \text{Smart}(\text{Berkeley}) \)

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

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

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

means “Everyone is at Berkeley and everyone is smart”

Existential quantification

\( \exists \text{variables} \) (sentence)

Someone at Stanford is smart:

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

\( \exists x \, P \) is equivalent to the disjunction of instantiations of \( P \)

\( \text{At}(\text{KingJohn}, \text{Stanford}) \land \text{Smart}(\text{KingJohn}) \)

\( \lor \text{At}(\text{Richard}, \text{Stanford}) \land \text{Smart}(\text{Richard}) \)

\( \lor \text{At}(\text{Stanford}, \text{Stanford}) \land \text{Smart}(\text{Stanford}) \)

\( \lor \ldots \)

Typically, \( \lor \) is the main connective with \( \exists \).

Common mistake: using \( \land \) 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 \( \forall y \, \forall x \, (\text{why?}) \)

\( \exists x \, \exists y \) is the same as \( \exists y \, \exists x \, (\text{why?}) \)

\( \exists x \, \forall y \) 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
Equality

\[ \text{term}_1 = \text{term}_2 \text{ is true under a given interpretation if and only if } \text{term}_1 \text{ and } \text{term}_2 \text{ refer to the same object} \]

E.g., \( 1 = 2 \) and \( \forall x \ (\text{Sqrt}(x), \text{Sqrt}(x)) = x \) are satisfiable

\( 2 = 2 \) is valid

E.g., definition of (full) Sibling in terms of Parent:

\[ \forall x, y \ \text{Sibling}(x, y) \iff \neg(x = y) \land \exists m, f \ (m = f) \land \text{Parent}(m, x) \land \text{Parent}(f, x) \land \text{Parent}(m, y) \land \text{Parent}(f, y) \]

Knowledge base for the wumpus world

"Perception"

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

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

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

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

\[ \forall t \ \text{AtGold}(t) \land \neg \text{Holding}(\text{Gold}, t) \Rightarrow \text{Action}(\text{Grasp}, t) \]

\( \text{Holding}(\text{Gold}, t) \) cannot be observed

\[ \Rightarrow \text{keeping track of change is essential} \]

Deducing hidden properties

Properties of locations:

\[ \forall l, t \ \text{At}(\text{Agent}, l, t) \land \text{Smell}(t) \Rightarrow \text{Smell}(l) \]

\[ \forall l, t \ \text{At}(\text{Agent}, l, t) \land \text{Breeze}(t) \Rightarrow \text{Breezy}(l) \]

Squares are breezy near a pit:

Diagnostic rule—infer cause from effect

\[ \forall y \ \text{Breezy}(y) \Rightarrow \exists x \ \text{Pit}(x) \land \text{Adjacent}(x, y) \]

Causal rule—infer effect from cause

\[ \forall x, y \ \text{Pit}(x) \land \text{Adjacent}(x, y) \Rightarrow \text{Breezy}(y) \]

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:

\[ \forall y \ \text{Breezy}(y) \iff \exists x \ \text{Pit}(x) \land \text{Adjacent}(x, y) \]

Keeping track of change

Facts hold in situations, rather than eternally

E.g., \( \text{Holding}(\text{Gold}, \text{Now}) \) rather than just \( \text{Holding}(\text{Gold}) \)

Situation calculus is one way to represent change in FOL:

Add a situation argument to each non-temporal predicate

E.g., \( \text{Now in } \text{Holding}(\text{Gold}, \text{Now}) \) denotes a situation

Situations are connected by the Result function

\( \text{Result}(a, s) \) is the situation that results from doing \( a \) is \( s \)

Describing actions I

"Effect" axiom—describe changes due to action

\[ \forall s \ \text{AtGold}(s) \Rightarrow \text{Holding}(\text{Gold}, \text{Result}(\text{Grasp}, s)) \]

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

\[ \forall s \ \text{HaveArrow}(s) \Rightarrow \text{HaveArrow}(\text{Result}(\text{Grasp}, s)) \]

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

\[ (a) \text{ representation—avoid frame axioms} \]

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

Qualification problem: true descriptions of real actions require endless caveats—what if the 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 the 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:
\(\forall a, s \quad \text{Holding}(Gold, Result(a, s)) \Leftrightarrow\)
\[\{[a = \text{Grab} \land \text{AtGold}(s)] \lor \text{Holding}(Gold, s) \land a \neq \text{Release}\}\]

Making plans: A better way

Represent plans as action sequences \([a_1; a_2; \ldots; a_n]\)

\(\text{PlanResult}(p, s)\) is the result of executing \(p\) in \(s\)

Then the query \(\text{Ask}(KB, \exists p \quad \text{Holding}(Gold, \text{PlanResult}(p, S_0)))\)

has the solution \(\{p/\text{Forward}, \text{Grab}\}\)

Definition of \(\text{PlanResult}\) in terms of \(\text{Result}\):
\(\forall s \quad \text{PlanResult}([], s) = s\)
\(\forall a, p, s \quad \text{PlanResult}([a[p], s] = \text{PlanResult}(p, \text{Result}(a, s))\)

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

Making plans

Initial condition in KB:
\(\text{At}(\text{Agent}, [1, 1], S_0)\)
\(\text{At}(\text{Gold}, [1, 2], S_0)\)

Query: \(\text{Ask}(KB, \exists s \quad \text{Holding}(\text{Gold}, s))\)
i.e., in what situation will I be holding the gold?

Answer: \(\{s/\text{Result}([\text{Grab, Result(F}orward, S_0)])\}\)
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

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