Artificial Intelligence (CS303)

Lecture 11: First-Order Logic

Hints for this lecture

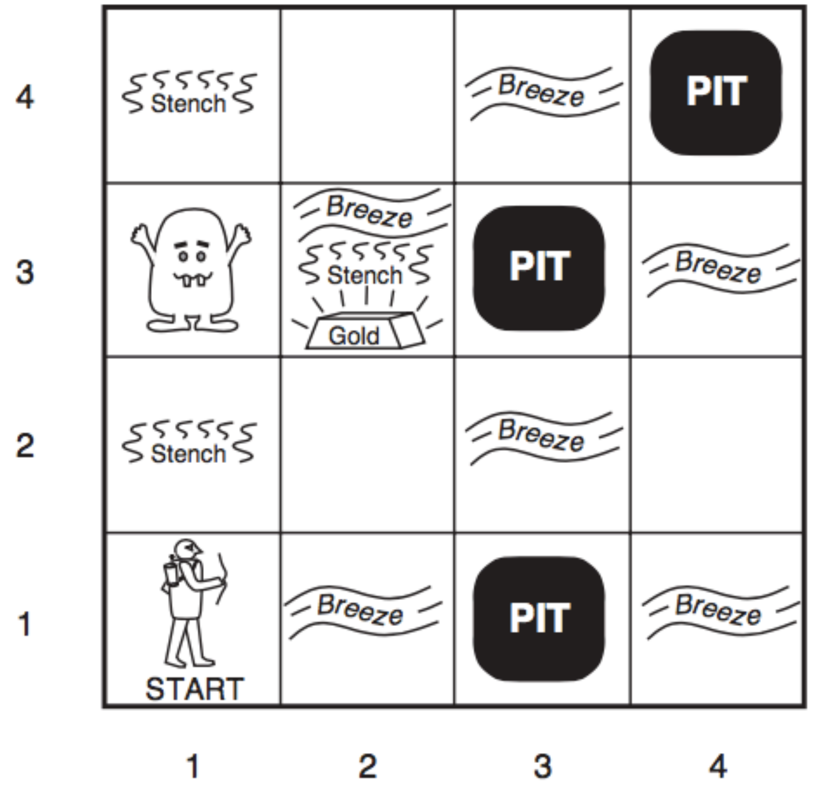
- **We prefer representations that are similar to natural language (at least sometimes).**

I. Definitions

Why need FOL

- A different knowledge representation that
  - might be easier to construct KB
  - or to inference
  - more natural to human thoughts

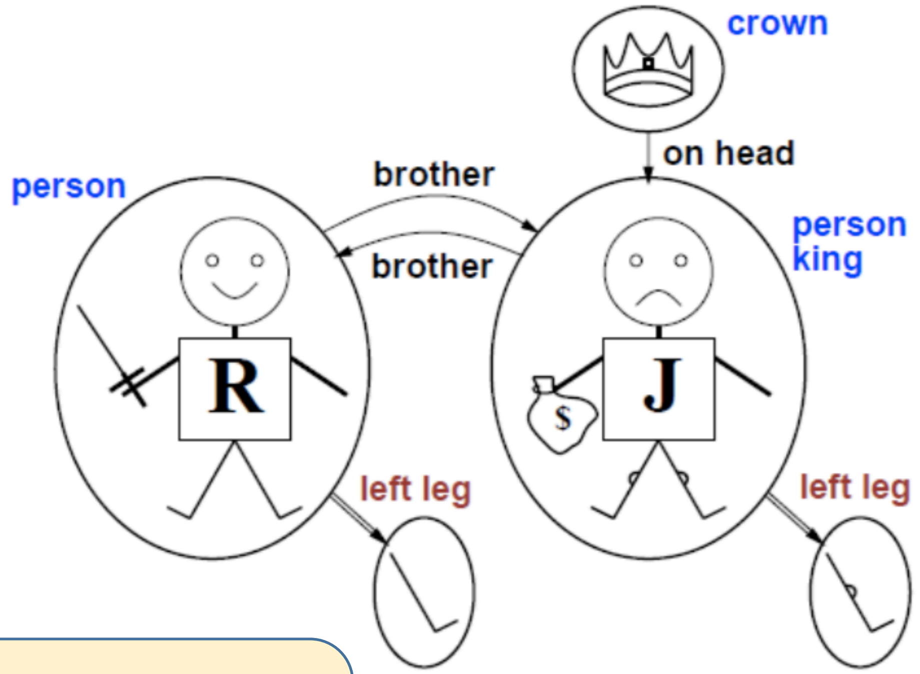
Boring to enumerate events for all squares



"Viewpoint" of FOL

- **The world is a "graph"**

Objects	people, houses, numbers, theories, Ronald McDonald, colors, baseball games, wars, centuries . . .
Relations	red, round, bogus, prime, multistoried, brother of, bigger than, inside, part of, has color, occurred after, owns, comes between,
Function	father of, best friend, third inning of, one more than, end of . . .



What is model in such a definition?

What is the advantages?

Syntax of FOL

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

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., $Brother(KingJohn, RichardTheLionheart)$
$> (Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))$

Propositional Logic

Propositional logic is the simplest logic—illustrates basic ideas

The proposition symbols P_1, P_2 etc are sentences

If S is a sentence, $\neg S$ is a sentence (negation)

If S_1 and S_2 are sentences, $S_1 \wedge S_2$ is a sentence (conjunction)

If S_1 and S_2 are sentences, $S_1 \vee S_2$ is a sentence (disjunction)

If S_1 and S_2 are sentences, $S_1 \Rightarrow S_2$ is a sentence (implication)

If S_1 and S_2 are sentences, $S_1 \Leftrightarrow S_2$ is a sentence (biconditional)

Syntax of FOL

Complex sentences are made from atomic sentences using connectives

$$\neg S, \quad S_1 \land S_2, \quad S_1 \lor S_2, \quad S_1 \Rightarrow S_2, \quad S_1 \Leftrightarrow S_2$$

E.g. $Sibling(KingJohn, Richard) \Rightarrow Sibling(Richard, KingJohn)$
$>(1,2) \lor \leq(1,2)$
$>(1,2) \land \neg{>}(1,2)$

Syntax of FOL

Sentences are true with respect to a model 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 $predicate(term_1, \ldots, term_n)$ is true
iff the objects referred to by $term_1, \ldots, term_n$
are in the relation referred to by $predicate$

Syntax of FOL - Universal/Existential Quantification and Equality

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

Everyone at Berkeley is smart:

$\forall x \ At(x, Berkeley) \Rightarrow Smart(x)$

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

$$\begin{array}{ll} & At(KingJohn, Berkeley) \Rightarrow Smart(KingJohn) \\ \wedge & At(Richard, Berkeley) \Rightarrow Smart(Richard) \\ \wedge & At(Berkeley, Berkeley) \Rightarrow Smart(Berkeley) \\ \wedge & \ldots \end{array}$$

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

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

$$\forall x \ At(x, Berkeley) \wedge Smart(x)$$

means "Everyone is at Berkeley and everyone is smart"

Syntax of FOL - Universal/Existential Quantification and Equality

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

Someone at Stanford is smart:

$\exists x \;\; At(x, Stanford) \land Smart(x)$

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

$$\begin{array}{ll} & At(KingJohn, Stanford) \land Smart(KingJohn) \\ \lor & At(Richard, Stanford) \land Smart(Richard) \\ \lor & At(Stanford, Stanford) \land Smart(Stanford) \\ \lor & \ldots \end{array}$$

Typically, $\land$ is the main connective with $\exists$.

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

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

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

Syntax of FOL - Universal/Existential Quantification and 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

$Father(John) = Henry$ implies $Father(John)$ and $Henry$ refers to the same object

used with negation to insist two terms are not the same object

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

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

II. Inference with FOL

Inference with FOL

- **naïve idea: reduce to propositional logic**

Every instantiation of a universally quantified sentence is entailed by it:

$$\frac{\forall v \;\; \alpha}{\textsc{Subst}(\{v/g\}, \alpha)}$$

for any variable v and ground term g

Universal Instantiation

E.g., $\forall x \;\; King(x) \land Greedy(x) \;\Rightarrow\; Evil(x)$ yields

$$\begin{array}{l} King(John) \land Greedy(John) \;\Rightarrow\; Evil(John) \\ King(Richard) \land Greedy(Richard) \;\Rightarrow\; Evil(Richard) \\ King(Father(John)) \land Greedy(Father(John)) \;\Rightarrow\; Evil(Father(John)) \\ \vdots \end{array}$$

UI can be applied several times to **add** new sentences; the new **KB** is logically equivalent to the old one

Inference with FOL

- **naïve idea: reduce to propositional logic**

Existential Instantiation

For any sentence α, variable v, and constant symbol k
that does not appear elsewhere in the knowledge base:

$$\frac{\exists v \;\; \alpha}{\text{SUBST}(\{v/k\}, \alpha)}$$

E.g., $\exists x \;\; Crown(x) \land OnHead(x, John)$ yields

$$Crown(C_1) \land OnHead(C_1, John)$$

provided C_1 is a new constant symbol, called a Skolem constant

Another example: from $\exists x \;\; d(x^y)/dy = x^y$ we obtain

$$d(e^y)/dy = e^y$$

provided e is a new constant symbol

EI can be applied once times to **replace** the existential sentence; the new **KB** is not equivalent to the old one, but is satisfiable is the old KB is satisfiable.

Example: Reduction to Propositional Logic

Suppose the KB contains just the following:

$$\forall x \ \ King(x) \land Greedy(x) \ \Rightarrow \ Evil(x)$$
$$King(John)$$
$$Greedy(John)$$
$$Brother(Richard; John)$$

Instantiating the universal sentence in all possible ways, we have

$$King(John) \land Greedy(John) \ \Rightarrow \ Evil(John)$$
$$King(Richard) \land Greedy(Richard) \ \Rightarrow \ Evil(Richard)$$
$$King(John)$$
$$Greedy(John)$$
$$Brother(Richard; John)$$

The new KB is **propositionalized**: proposition symbols are

$$King(John); Greedy(John); Evil(John); King(Richard) etc.$$

Problem with Reduction

Claim: a ground sentence $\star$ is entailed by new KB iff entailed by original KB

Claim: every FOL KB can be propositionalized so as to preserve entailment

Idea: propositionalize KB and query, apply resolution, return result

Problem: with function symbols, there are infinitely many ground terms, e.g., $Father(Father(Father(John)))$

Theorem: Herbrand (1930). If a sentence α is entailed by an FOL KB, it is entailed by a finite subset of the propositional KB

Idea: $For\ n = 0$ to ∞ do
  create a propositional KB by instantiating with depth-n terms
  see if α is entailed by this KB

Problem: works if α is entailed, loops if α is not entailed

Theorem: Turing (1936), Church (1936), entailment in FOL is **semidecidable**

Representation unlikely changes the complexity, this is because FOL expresses a more complicated world.

Problem with Reduction

Propositionalization seems to generate lots of irrelevant sentences.
E.g., from

$$\forall x \;\; King(x) \wedge Greedy(x) \;\Rightarrow\; Evil(x)$$
$$King(John)$$
$$\forall y \;\; Greedy(y)$$
$$Brother(Richard; John)$$

it seems obvious that $Evil(John)$, but propositionalization produces lots of facts such as $Greedy(Richard)$ that are irrelevant

With p k-ary predicates and n constants, there are $p \cdot n^k$ instantiations

With function symbols, it gets much much worse!

Better Ideas to Inference with FOL

- **Unification**
  - **Resolution**
  - **Chaining Algorithms**

To be continued