Artificial Intelligence
First-Order Logic

Dr. Debasis Das
Department of Computer Science and Engineering
Indian Institute of Technology Jodhpur

The FOL language

FOL is a language—we define the syntax, the semantics, and give examples.

The limitation of propositional logic

- Propositional logic has nice properties:
  - 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)

- Limitation:
  - Propositional logic has very limited expressive power, unlike natural language. E.g., we cannot express “pits cause breezes in adjacent squares” except by writing one sentence for each square

First-order logic

- Whereas propositional logic assumes that a 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 . . ., brother of, bigger than, inside, part of, has color, occurred after, owns, comes between, . . .
  - *Functions*: father of, best friend, third inning of, one more than, end of . . .

FOL syntax elements

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

FOL syntax grammar

$$
\begin{array}{lcl}
\langle\text{sentence}\rangle & \rightarrow & \langle\text{atomic sentence}\rangle \\
 & & \mid \langle\text{complex sentence}\rangle \\
 & & \mid [\forall \mid \exists]\ \langle\text{variable}\rangle\ \ \langle\text{sentence}\rangle \\
 & & \\
\langle\text{atomic sentence}\rangle & \rightarrow & \text{predicate}(\langle\text{term}\rangle,\ldots) \\
 & & \mid \langle\text{term}\rangle{=}\langle\text{term}\rangle \\
 & & \\
\langle\text{term}\rangle & \rightarrow & \text{function}(\langle\text{term}\rangle,\ldots) \\
 & & \mid \text{constant} \\
 & & \mid \text{variable} \\
 & & \\
\langle\text{complex sentence}\rangle & \rightarrow & \neg\ \langle\text{sentence}\rangle \\
 & & \mid (\langle\text{sentence}\rangle\ [\wedge \mid \vee \mid\ \Rightarrow\ \mid\ \Leftrightarrow\ ]\ \langle\text{sentence}\rangle
\end{array}
$$

Quantifiers

- Universal quantification

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

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

Example: "Everyone at Berkeley is smart:" $\forall x \;\; At(x, Berkeley) \Rightarrow Smart(x)$

- Existential quantification

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

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

Example: "Someone at Stanford is smart:" $\exists x \;\; At(x, Stanford) \land Smart(x)$

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 $\forall y\ \exists x$

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

  $\forall x\ Likes(x, IceCream) \quad \equiv \quad \neg\exists x\ \neg Likes(x, IceCream)$

  $\exists x\ Likes(x, Broccoli) \quad \equiv \quad \neg\forall x\ \neg Likes(x, Broccoli)$

Truth in first-order logic

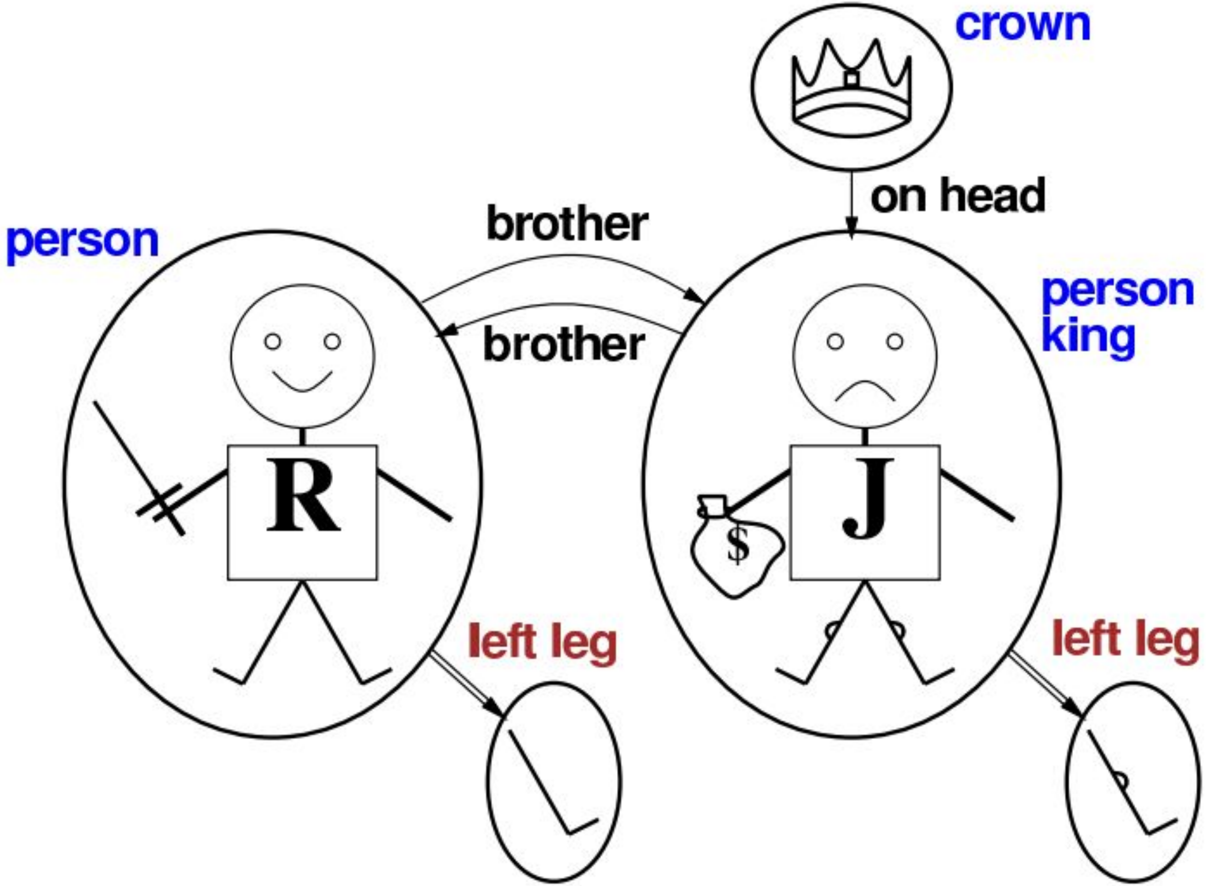
- Sentences are true with respect to a model and an interpretation
- A model contains ≥ 1 objects and relations among them
- An interpretation specifies referents for

  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 $predicate$

Models for FOL: Example



Models for FOL: Lots!

- Entailment in propositional logic can be computed by enumerating models
- We can also enumerate the FOL models for a given KB:
  - 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 $\ldots$
- Enumerating FOL models is very inefficient

Example sentences

- "Brothers are siblings"

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

- "Sibling" is symmetric

Example sentences

- "Brothers are siblings"

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

- "Sibling" is symmetric

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

- "One's mother is one's female parent"

Example sentences

- "Brothers are siblings"

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

- "Sibling" is symmetric

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

- "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"

Example sentences

- "Brothers are siblings"

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

- "Sibling" is symmetric

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

- "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"

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

Thanks