First-order logic

Chapter 8, Sections 1–3

Outline

◇ Why FOL?

◇ Syntax of FOL

◇ Semantics of FOL

Pros and cons of propositional logic

- 😊 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)
- ☹ Propositional logic has very limited expressive power (unlike natural language)
  E.g., cannot say "pits cause breezes in adjacent squares"
  except by writing one sentence for each square

First-order logic

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

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

Syntax of FOL: Basic elements

Constants	$John,\ 2,\ UCB, \ldots$
Predicates	$Brother,\ >, \ldots$
Functions	$Sqrt,\ LeftLeg, \ldots$
Variables	$x,\ y,\ a,\ b, \ldots$
Connectives	$\land\ \lor\ \neg\ \Rightarrow\ \Leftrightarrow$
Equality	$=$
Quantifiers	$\forall\ \exists$

Atomic sentences

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., $Brother(Richard, John)$
$Married(Father(Richard), Mother(John))$

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. $\neg Brother(LeftLeg(Richard), John)$
$Brother(Richard, John) \wedge Brother(John, Richard)$
$\neg King(Richard) \;\Rightarrow\; King(John)$

Note: The last one is equivalent to:

$$(\neg King(Richard)) \;\Rightarrow\; King(John)$$

and not:

$$\neg(King(Richard) \;\Rightarrow\; King(John))$$

Truth in first-order logic

Sentences are true with respect to a model and an interpretation

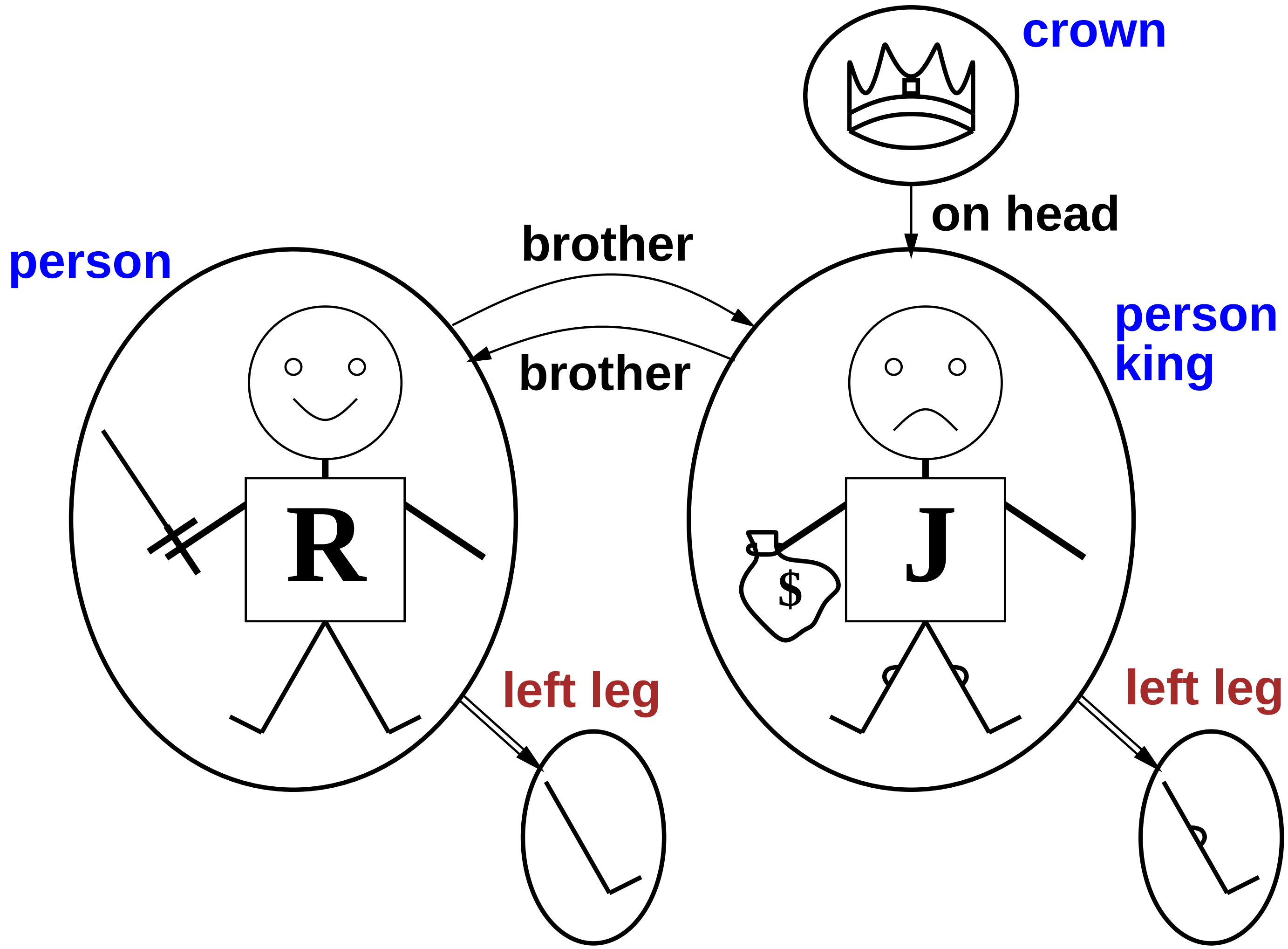
Model contains ≥ 1 objects (domain elements) and relations among them

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 $predicate$

Models for FOL: Example



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 just in case Richard the Lionheart and the evil King John are in the brotherhood relation in the model

Models for FOL: Lots!

Entailment in propositional logic can be computed by enumerating models

We **can** enumerate the FOL models for a given KB vocabulary:

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 . . .

Computing entailment by enumerating FOL models is not easy!

Universal quantification

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

All kings are persons:
$\forall x \;\; King(x) \;\Rightarrow\; Person(x)$

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

Roughly speaking, equivalent to the conjunction of instantiations of P

$$
\begin{aligned}
& (King(Richard) \;\Rightarrow\; Person(Richard)) \\
\land\; & (King(John) \;\Rightarrow\; Person(John)) \\
\land\; & (King(TheCrown) \;\Rightarrow\; Person(TheCrown)) \\
\land\; & (King(LeftLeg(Richard)) \;\Rightarrow\; Person(LeftLeg(Richard))) \\
\land\; & \ldots
\end{aligned}
$$

A common mistake to avoid

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

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

$$\forall\, x \;\; King(John) \land Person(John)$$

means "Everything is both a king and a person"

Existential quantification

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

King John has a crown on his head:
$\exists x \ \ Crown(x) \land OnHead(x, John)$

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

Roughly speaking, equivalent to the disjunction of instantiations of P

$$
\begin{aligned}
& (Crown(Richard) \land OnHead(Richard, John)) \\
\lor \ & (Crown(John) \land OnHead(John, John)) \\
\lor \ & (Crown(TheCrown) \land OnHead(TheCrown, John)) \\
\lor \ & (Crown(LeftLeg(Richard)) \land OnHead(LeftLeg(Richard), John)) \\
\lor \ & \ldots
\end{aligned}
$$

Another common mistake to avoid

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

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

$$\exists\, x \;\; Crown(x) \;\Rightarrow\; OnHead(x, John)$$

is true if there is anything that is not a crown!

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

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

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

Fun with 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

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

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., to say that Richard has at least two brothers:

$$\exists\, x, y \;\; Brother(x, Richard) \land Brother(y, Richard) \land \neg(x = y)$$

Without the final $\neg(x = y)$, it would just say that
Richard has at least one brother.

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

$$\begin{aligned}
&\forall\, x, y \;\; Sibling(x, y) \;\Leftrightarrow \\
&\quad [\, \exists\, m, f \;\; \neg(x = y) \land \neg(m = f) \land \\
&\qquad\quad Parent(m, x) \land Parent(f, x) \land Parent(m, y) \land Parent(f, y)\,]
\end{aligned}$$

Example: The wumpus world

Squares are breezy near a pit:

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:

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

Summary

First-order logic:

– objects and relations are semantic primitives

– syntax: constants, functions, predicates, equality, quantifiers

Increased expressive power:

– sufficient to define the wumpus world