Universität Bielefeld

Artificial Intelligence Methods

Marc Erich Latoschik
AI & VR Lab
Artificial Intelligence Group
University of Bielefeld

Universität Bielefeld

First-Order Logic

*see (Russel & Norvig, 2004) Chapter 8

Outline

- Why FOL?
- Syntax and semantics of FOL
- Using FOL
- Wumpus world in FOL
- Knowledge engineering in FOL

Pros and cons of propositional logic

☺ Propositional logic is declarative

☺ 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

Logic in general

- Ontological Commitment: What exists in the world — **TRUTH**
- Epistemoligical Commitment: What an agent believes about facts — **BELIEF**

Language	Ontological Commitment	Epistemological Commitment
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief $\in [0, 1]$
Fuzzy logic	degree of truth $\in [0, 1]$	known interval value

First-order logic

- Whereas propositional logic assumes the world contains facts,
- first-order logic (like natural language) assumes the world contains
  - **Objects**: people, houses, numbers, colors, baseball games, wars, …
  - **Relations**: red, round, prime, brother of, bigger than, part of, comes between, …
  - **Functions**: father of, best friend, one more than, plus, …

Basic FOL elements:

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

FOL Syntax

Term:
function ($term_1$,...,$term_n$)
or *constant* or *variable*

Atomic sentence:
predicate ($term_1$,...,$term_n$)
or $term_1$ = $term_2$

E.g.,
- *Brother(KingJohn,RichardTheLionheart)*
- *> (Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))*

Complex sentences:
made from atomic sentences
using connectives
$\neg S$, $S_1 \wedge S_2$, $S_1 \vee S_2$, $S_1 \Rightarrow S_2$, $S_1 \Leftrightarrow S_2$,

E.g.
- *Sibling(KingJohn,Richard)* $\Rightarrow$ *Sibling(Richard,KingJohn)*
- $>(1,2) \vee \leq (1,2)$
- $>(1,2) \wedge \neg >(1,2)$

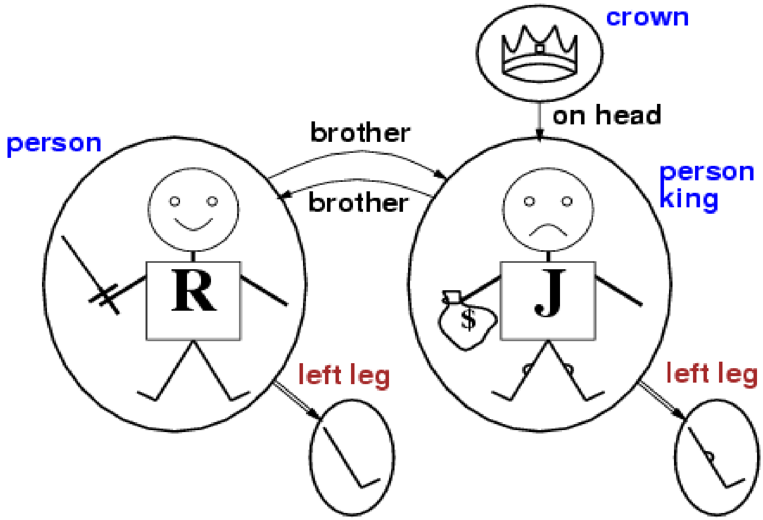
$$
\begin{aligned}
\textit{Sentence} &\rightarrow \textit{AtomicSentence} \\
&\mid (\textit{Sentence Connective Sentence}) \\
&\mid \textit{Quantifier Variable}, \ldots\ \textit{Sentence} \\
&\mid \neg\, \textit{Sentence} \\
\textit{AtomicSentence} &\rightarrow \textit{Predicate}(\textit{Term}, \ldots) \mid \textit{Term} = \textit{Term} \\
\textit{Term} &\rightarrow \textit{Function}(\textit{Term}, \ldots) \\
&\mid \textit{Constant} \\
&\mid \textit{Variable} \\
\textit{Connective} &\rightarrow \Rightarrow \mid \wedge \mid \vee \mid \Leftrightarrow \\
\textit{Quantifier} &\rightarrow \forall \mid \exists \\
\textit{Constant} &\rightarrow A \mid X_1 \mid \textit{John} \mid \ldots \\
\textit{Variable} &\rightarrow a \mid x \mid s \mid \ldots \\
\textit{Predicate} &\rightarrow \textit{Before} \mid \textit{HasColor} \mid \textit{Raining} \mid \ldots \\
\textit{Function} &\rightarrow \textit{Mother} \mid \textit{LeftLeg}
\end{aligned}
$$

Syntax for FOL in BNF

Truth in first-order logic

- Sentences are true with respect to a **model** and an **interpretation**
- Model contains objects (**domain elements**) and relations among them
- Interpretation specifies referents for
  constant symbols → **objects**
  predicate symbols → **relations**
  function symbols → **functional relations**
- An atomic sentence *predicate*($term_1$,...,$term_n$) is true
  iff the **objects** referred to by $term_1$,...,$term_n$
  are in the **relation** referred to by *predicate*

Models for FOL: Example



Models for FOL

- We can enumerate the 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 the models will not be easy even for a finite number of elements in given domain!
- It is worse for an infinite number, e.g., for the domains of integers or real values.
  - Infinite number of possible models!
  - Infinite number of possible interpretations!

Universal quantification

$\forall$***<variables> <sentence>***

E.g., everyone at UNIBI is smart: $\forall x\ At(x,UNIBI) \Rightarrow Smart(x)$

- $\forall x\ P$ is true in a model *m* iff *P* is true with *x* being each possible object in the model
- Roughly speaking, equivalent to the **conjunction** of **instantiations** of *P*

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

A common mistake to avoid:

- Typically, $\Rightarrow$ is the main connective with $\forall$
- Common mistake: using $\wedge$ as the main connective with $\forall$:

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

…means “Everyone is at UNIBI and everyone is smart”

Existential quantification

$\exists$***<variables> <sentence>***

E.g., someone at UNIBI is smart: $\exists x\ At(x,UNIBI) \wedge Smart(x)$

- $\exists x\ P$ is true in a model *m* iff *P* is true with *x* being some possible object in the model
- Roughly speaking, equivalent to the **disjunction** of **instantiations** of *P*

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

Another common mistake to avoid:

- Typically, $\wedge$ is the main connective with $\exists$
- Common mistake: using $\Rightarrow$ as the main connective with $\exists$:

$$\exists x\ At(x,UNIBI) \Rightarrow Smart(x)$$

…is true if there is anyone who is not at UNIBI!

Quantifiers/Equality

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) $\neg\exists x\ \neg$Likes(x,IceCream)
- $\exists x$ Likes(x,Broccoli) $\neg\forall x\ \neg$Likes(x,Broccoli)

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., definition of *Sibling* in terms of *Parent*:

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

Using FOL

The kinship domain:

1. Domain objects: humans
2. Two unary predicates *Male* and *Female, b*inary predicates for kinship relations
3. Functions for *Mother* and *Father* since they are unique for an individual.

- One's mother is one's female parent:
  $\forall m,c\ Mother(c) = m \Leftrightarrow (Female(m) \wedge Parent(m,c))$
- Husband is one’s male spouse:
  $\forall w,h\ Husband\ (w,h) \Leftrightarrow (Male(h) \wedge Spouse\ (h,w))$
- Male and female are disjoint categories:
  $\forall x\ Male(x) \Leftrightarrow \neg Female(x)$
- Parent and child are inverse relations:
  $\forall p,c\ Parent(p,c) \Leftrightarrow Child(c,p)$
- “Sibling” is symmetric
  $\forall x,y\ Sibling(x,y) \Leftrightarrow Sibling(y,x)$
- A grandparent is a parent of one’s parents:
  $\forall g,c\ Grandparent(g,c) \Leftrightarrow \exists p\ Parent(g,p) \wedge Parent(p,c)$
- A sibling is another child of one’s parents:
  $\forall x,y\ Sibling(x,y) \Leftrightarrow x \neq y \wedge \exists\ p\ Parent(p,x) \wedge Parent(p,y)$