Informatics 2D · Agents and Reasoning · 2019/2020

Lecture 10 · First-Order Logic

Claudia Chirita

School of Informatics, University of Edinburgh

4th February 2020

Based on slides by: Jacques Fleuriot, Michael Rovatsos, Michael Herrmann, Vaishak Belle

Outline

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

Propositional logic as a language

Compared to languages in Computer Science

- serves as a basis for declarative languages
- allows partial/disjunctive/negated information
  - unlike most data structures and databases
- is compositional
  - e.g. the meaning of $B_{1,1} \land P_{1,2}$ is derived from that of $B_{1,1}$ and of $P_{1,2}$
  - unlike some instances of concurrent programming

Compared to natural languages

- meaning is context-independent
  - unlike natural languages, where meaning depends on context
- propositional logic has very limited expressive power
  - e.g. we can say *pits cause breezes in adjacent squares* only by writing one sentence for each square

First-order logic

Propositional logic deals with atomic facts (i.e. atomic, non-structured propositional symbols; usually finitely many).

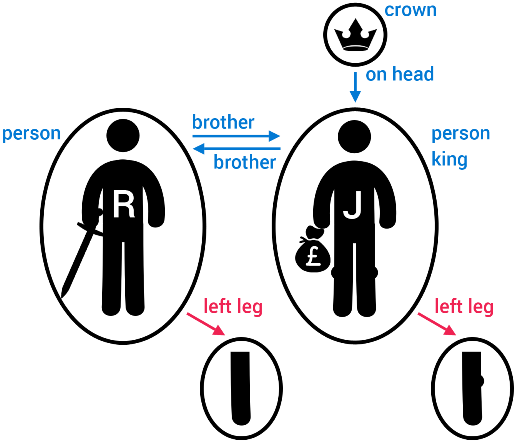
FOL brings structure to facts, which can be built from:

Objects: people, houses, numbers, colours, football games

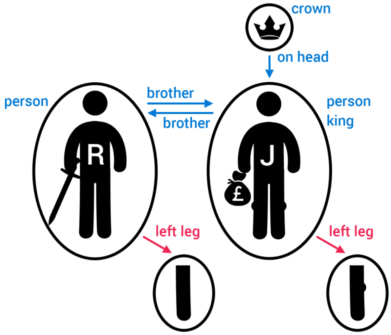
Functions: father of, best friend, one more than, plus

Relations: red, round, prime, brother of, bigger than, part of

Example · Of brothers and kings



Example · Of brothers and kings



$\text{Brother}(\text{KingJohn}, \text{RichardTheLionheart})$

$\text{Length}(\text{LeftLegOf}(\text{Richard})) > \text{Length}(\text{LeftLegOf}(\text{John}))$

Syntax · Signatures

A first-order signature is a pair (F, P)

- F – indexed family $(F_n)_{n\in\mathbb{N}}$ of sets of function symbols (operations)
- P – indexed family $(P_n)_{n\in\mathbb{N}}$ of sets of relation symbols (predicates)

For $\sigma \in F_n$ and $\pi \in P_n$, n is called arity.

Constant symbols are function symbols with arity zero.

Example

functions	$F_0 = \{\mathrm{Richard}, \mathrm{John}\}$, $F_1 = \{\mathrm{LeftLegOf}\}$
predicates	$P_1 = \{\mathrm{Crown}, \mathrm{King}, \mathrm{Person}\}$
	$P_2 = \{\mathrm{Brother}, \mathrm{OnHead}\}$

Syntax · Sentences

Terms: Least set T_F such that $\sigma(t_1, \dots, t_n) \in T_F$ for every $\sigma \in F_n$ and $t_1, \dots, t_n \in T_F$.

In particular, T_F contains all constants.

Variables: Every set of (F, P)-variables X determines an extended signature $(F \cup X, P)$ with the variables in X added to F_0 as new constants.

Sentences over a signature (F, P) are defined by the grammar

$$\begin{aligned} \varphi ::= {} & \pi(t_1, \dots, t_n) \mid t = t' && \text{atoms} \\ & \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \varphi \vee \varphi' \mid \varphi \to \varphi' \mid \varphi \leftrightarrow \varphi' && \text{boolean connectives} \\ & \mid \forall X.\varphi \mid \exists X.\varphi && \text{quantifiers} \end{aligned}$$

where $\pi \in P_n$ is a predicate symbol, $t, t', t_1, \dots, t_n$ are terms, and X is a set of variables.

Precedence: $\forall X, \exists X, \neg, \wedge, \vee, \to, \leftrightarrow$

Syntax · Sentences

Sentences over a signature (F, P) are defined by the grammar

$$\varphi ::= \pi(t_1, \dots, t_n) \mid t = t' \qquad \text{atoms}$$
$$\mid \neg\varphi \mid \varphi \wedge \varphi' \mid \varphi \vee \varphi' \mid \varphi \rightarrow \varphi' \mid \varphi \leftrightarrow \varphi' \qquad \text{boolean connectives}$$
$$\mid \forall X.\varphi \mid \exists X.\varphi \qquad \text{quantifiers}$$

where $\pi \in P_n$ is a predicate symbol, $t, t', t_1, \dots, t_n$ are terms, and X is a set of variables.

Precedence: $\forall X, \exists X, \neg, \wedge, \vee, \rightarrow, \leftrightarrow$

Example

$\text{Brother}(\text{John}, \text{Richard})$

$\text{Brother}(\text{John}, \text{Richard}) \wedge \text{Brother}(\text{Richard}, \text{John})$

$\neg\text{Brother}(\text{LeftLegOf}(\text{Richard}), \text{John})$

$\neg\text{King}(\text{Richard}) \rightarrow \text{King}(\text{John})$

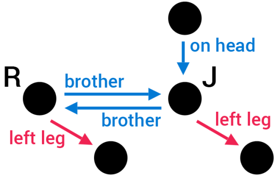
$\forall x.\text{King}(x) \rightarrow \text{Person}(x)$

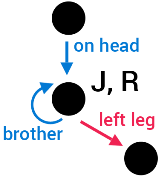
Semantics · Models

Given a signature (F, P), a model M consists of

- a non-empty set $|M|$, called the carrier set (domain) of M, whose elements are called objects
- a function $M_\sigma \colon |M|^n \to |M|$ for each operation symbol $\sigma \in F_n$
- a subset $M_\pi \subseteq |M|^n$ for each relation symbol $\pi \in P_n$

Examples





Satisfaction relation

The satisfaction relation links the syntax and the semantics.

- We write $M \vDash \varphi$ and read "M satisfies φ", for M a model and φ a sentence, both for the same signature (F, P).
- To make (F, P) explicit, we sometimes write $M \vDash_{(F,P)} \varphi$.
- The satisfaction relation is defined according to the structure of sentences (in the following slides), based on the evaluation of terms in models.

Evaluation of terms

- M_t denotes the interpretation of a term t in a model M.
- $M_{\sigma(t_1,\ldots,t_n)} = M_\sigma(M_{t_1}, \ldots, M_{t_n})$

e.g. $M_{\text{LeftLegOf(John)}} = M_{\text{LeftLegOf}}(M_{\text{John}})$

$= M_{\text{LeftLegOf}}($  $) =$ 

Satisfaction relation · $M \vDash \varphi$

Atoms

- $M \vDash t = t'$ iff $M_t = M_{t'}$
- $M \vDash \pi(t_1, \dots, t_n)$ iff $(M_{t_1}, \dots, M_{t_n}) \in M_\pi$

Boolean connectives

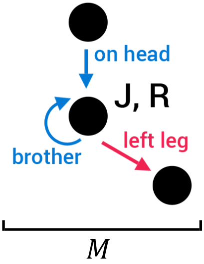
- $M \vDash \neg\varphi$ iff $M \nvDash \varphi$
- $M \vDash \varphi_1 \wedge \varphi_2$ iff $M \vDash \varphi_1$ and $M \vDash \varphi_2$
- $M \vDash \varphi_1 \vee \varphi_2$ iff $M \vDash \varphi_1$ or $M \vDash \varphi_2$
- $M \vDash \varphi_1 \rightarrow \varphi_2$ iff $M \vDash \varphi_2$ whenever $M \vDash \varphi_1$
- $M \vDash \varphi_1 \leftrightarrow \varphi_2$ iff $M \vDash \varphi_1 \rightarrow \varphi_2$ and $M \vDash \varphi_2 \rightarrow \varphi_1$

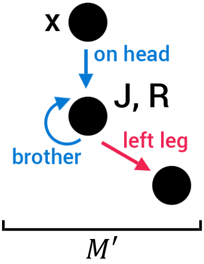
Satisfaction relation · $M \vDash \varphi$

Quantifiers

A model M' for $(F \cup X, P)$ is called an expansion of a model M for (F, P) if it interprets all symbols in F and in P the same as M. Expansions formalize assignments of elements from M to the variables in X.

Example





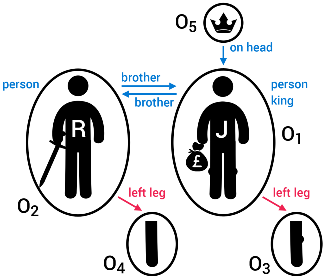
Satisfaction relation · $M \vDash \varphi$

Quantifiers

A model M' for $(F \cup X, P)$ is called an expansion of a model M for (F, P) if it interprets all symbols in F and in P the same as M. Expansions formalize assignments of elements from M to the variables in X.

- $M \vDash_{(F,P)} \forall X.\varphi$ iff $M' \vDash_{(F \cup X,P)} \varphi$
  for all expansions M' along the inclusion $(F, P) \subseteq (F \cup X, P)$
- $M \vDash_{(F,P)} \exists X.\varphi$ iff there exists an expansion M' along the inclusion $(F, P) \subseteq (F \cup X, P)$ such that $M' \vDash_{(F \cup X,P)} \varphi$

Satisfaction relation · Example



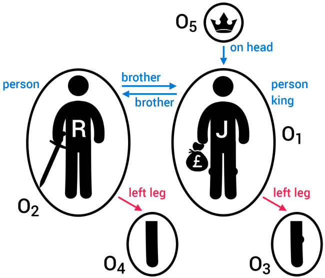
True or False?

$\text{Brother}(\text{John}, \text{Richard}) \land \text{Brother}(\text{Richard}, \text{John})$

$\neg\text{Brother}(\text{LeftLegOf}(\text{Richard}), \text{John})$

$\neg\text{King}(\text{Richard}) \rightarrow \text{King}(\text{John})$

Satisfaction relation · Example



True or False?

$\forall x.\text{King}(x) \rightarrow \text{Person}(x)$

$x \mapsto O_1$ (i.e. $M'_x = O_1$) O_1 (John) is a king → O_1 is a person.

$x \mapsto O_2$ O_2 (Richard) is a king → O_2 is a person.

$x \mapsto O_3$ O_3 (John's left leg) is a king → O_3 is a person.

$x \mapsto O_4$ O_4 (Richard's left leg) is a king → O_4 is a person.

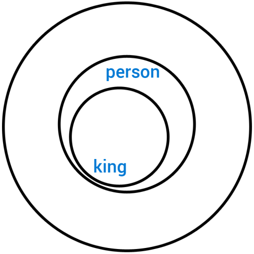
$x \mapsto O_5$ O_5 (crown) is a king → O_5 is a person.

Expressivity · Quantifiers

$\forall x.\text{King}(x) \rightarrow \text{Person}(x)$

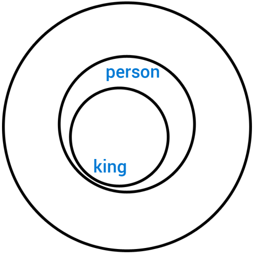
Expressivity · Quantifiers

$\forall x.\text{King}(x) \rightarrow \text{Person}(x)$



Expressivity · Quantifiers

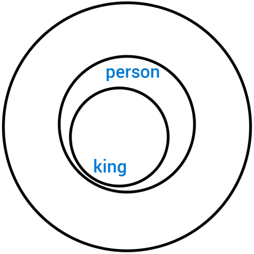
$\forall x.\text{King}(x) \rightarrow \text{Person}(x)$



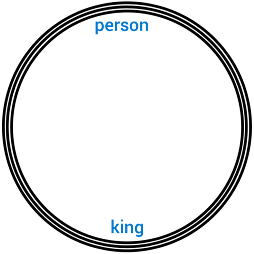
$\forall x.\text{King}(x) \wedge \text{Person}(x)$

Expressivity · Quantifiers

$\forall x.\text{King}(x) \rightarrow \text{Person}(x)$



$\forall x.\text{King}(x) \land \text{Person}(x)$

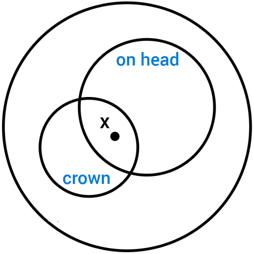


Expressivity · Quantifiers

$\exists x.\text{Crown}(x) \land \text{OnHead}(x, \text{John})$

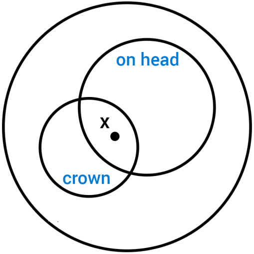
Expressivity · Quantifiers

$\exists x.\text{Crown}(x) \wedge \text{OnHead}(x, \text{John})$



Expressivity · Quantifiers

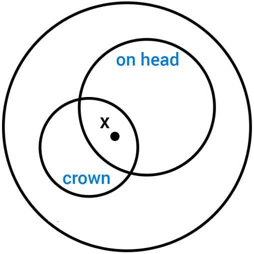
$\exists x.\text{Crown}(x) \wedge \text{OnHead}(x, \text{John})$



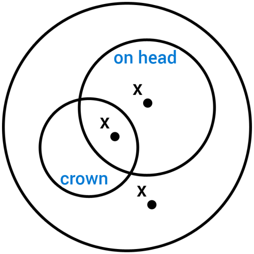
$\exists x.\text{Crown}(x) \rightarrow \text{OnHead}(x, \text{John})$

Expressivity · Quantifiers

$\exists x.\text{Crown}(x) \wedge \text{OnHead}(x, \text{John})$



$\exists x.\text{Crown}(x) \rightarrow \text{OnHead}(x, \text{John})$



Expressivity · Quantifiers

The order of quantifiers

$\exists X.\forall Y.\varphi$ is not the same thing as $\forall Y.\exists X.\varphi$

$\exists x.\forall y.\mathrm{Loves}(x, y)$
There is a person who loves everyone in the world.

$\forall y.\exists x.\mathrm{Loves}(x, y)$
Everyone in the world is loved by someone.

Duality

$\varphi \wedge \varphi' \equiv \neg(\neg\varphi \vee \neg\varphi')$ and $\varphi \vee \varphi' \equiv \neg(\neg\varphi \wedge \neg\varphi')$

$\forall X.\varphi \equiv \neg\exists X.\neg\varphi$

$\forall x.\mathrm{Likes}(x, \mathrm{IceCream}) \equiv \neg\exists x.\neg\mathrm{Likes}(x, \mathrm{IceCream})$

$\exists X.\varphi \equiv \neg\forall X.\neg\varphi$

$\exists x.\mathrm{Likes}(x, \mathrm{Broccoli}) \equiv \neg\forall x.\neg\mathrm{Likes}(x, \mathrm{Broccoli})$

Using FOL · Kinship domain

Axioms: definitions, theorems

One's mother is one's female parent.

$\forall m, c. m = \text{Mother}(c) \leftrightarrow (\text{Female}(m) \land \text{Parent}(m, c))$

Parent and child are inverse relations.

$\forall p, c. \text{Parent}(p, c) \leftrightarrow \text{Child}(c, p)$

A sibling is another child of one's parents.

$\forall x, y. \text{Sibling}(x, y) \leftrightarrow x \neq y \land \exists p. \text{Parent}(p, x) \land \text{Parent}(p, y)$

Brothers are siblings.

$\forall x, y. \text{Brother}(x, y) \rightarrow \text{Sibling}(x, y)$

The sibling relation is symmetric.

$\forall x, y. \text{Sibling}(x, y) \leftrightarrow \text{Sibling}(y, x)$

Interacting with FOL KBs

$\textsc{Tell}/\textsc{Ask}$ interface

Assertions

$\textsc{Tell}(\text{KB}, \text{King}(\text{John}))$
$\textsc{Tell}(\text{KB}, \text{Person}(\text{Richard}))$
$\textsc{Tell}(\text{KB}, \forall x.\text{King}(x) \to \text{Person}(x))$

Queries (goals)

$\textsc{Ask}(\text{KB}, \text{Person}(\text{John}))$	*true*
$\textsc{Ask}(\text{KB}, \exists x.\text{Person}(x))$	*true*
$\textsc{Ask}(\text{KB}, \text{Person}(x))$	$\{x/\text{John}\}, \{x/\text{Richard}\}$

Idea
$\textsc{Ask}(\text{KB}, \varphi)$ returns all substitutions θ such that $\text{KB} \vDash \theta(\varphi)$.

Example · Wumpus world

Suppose a wumpus-world agent is using an FOL KB and perceives a smell and a breeze (but no glitter) at $t = 5$.

$\textsc{Tell}(\text{KB}, \text{Percept}([\text{Smell}, \text{Breeze}, \text{None}], 5))$

Does the KB entail some best action at $t = 5$?

$\textsc{Ask}(\text{KB}, \exists a.\text{BestAction}(a, 5))$

Answer: *true*, $\{a/\text{Shoot}\}$

Perception

$\forall t, s, b.\text{Percept}([s, b, \text{Glitter}], t) \rightarrow \text{Glitter}(t)$

Reflex

$\forall t.\text{Glitter}(t) \rightarrow \text{BestAction}(\text{Grab}, t)$

Example · Wumpus world

The environment

$\forall x, y, a, b.\text{Adjacent}([x, y]) \leftrightarrow [a, b] \in \{[x+1, y], [x-1, y],$
$[x, y+1], [x, y-1]\}$

$\forall s, t.\text{At}(\text{Agent}, s, t) \land \text{Breeze}(t) \rightarrow \text{Breezy}(s)$

Squares are breezy near a pit.

Diagnostic rule: infer cause from effect

$\forall s.\text{Breezy}(s) \rightarrow \exists r.\text{Adjacent}(r, s) \land \text{Pit}(r)$

Causal rule: infer effect from cause

$\forall r.\text{Pit}(r) \rightarrow (\forall s.\text{Adjacent}(r, s) \rightarrow \text{Breezy}(s))$

Summary

First-order logic:

- Objects and relations are semantic primitives.
- Syntax: constants, functions, predicates, quantifiers.
- Increased expressive power – sufficient to define the Wumpus world.