FOL (First-Order Logic)
BE4M36LUP

Jiří Vyskočil
CIIRRC CTU, Prague

2017
First-Order Logic

- propositional logic assumes the world contains facts
- first-order logic (like natural language or Prolog) assumes the world contains:
  - objects (people, houses, numbers, colors, animals, ...)
    - relations (being red, being prime, brother of, bigger than, ...)
    - functions (father of, successor of, one more than, plus, ...)
    - constants (13, \(\pi\), green, mouse, ...)


First-Order Language

- functions (sqrt, left_leg_of, ...)  
  - constants (2, Ø, ...)  
- variables (X, Y, A, B, ...)  
- predicates (odd, brother, >, ...)  
  - equality (=)  
- boolean connectives (¬, \( \rightarrow \), \( \vee \), &, ...)  
- quantifiers (\( \forall \), \( \exists \))
First-Order Language: TPTP

- CFG (Context Free Grammar) specification of TPTP (Thousands of Problems for Theorem Provers):

  \[\begin{align*}
  \text{variable} & ::= \text{a symbol starting with an upper case letter} \\
  \text{function\_symbol} & ::= \text{a symbol starting with a lower case letter} \\
  \text{term} & ::= \text{variable} \\
  & \quad \left| \text{function\_symbol} \quad \text{(constant)} \right. \\
  & \quad \left| \text{function\_symbol} '\left(\text{' term}_1',' \ldots',' \text{term}_n'\right)' \quad (n > 0) \right. \\
  \text{predicate\_symbol} & ::= \text{a symbol starting with a lower case letter} \\
  \text{atomic\_formula} & ::= \text{predicate\_symbol} \quad \text{(arity 0)} \\
  & \quad \left| \text{predicate\_symbol} '\left(\text{' term}_1',' \ldots',' \text{term}_n'\right)' \quad (n > 0) \right. \\
  \text{formula} & ::= \text{atomic\_formula} \\
  & \quad \left| '\left(\quad \text{' ~'} \quad \text{formula'}\right)' \quad \text{(\neg)} \right. \\
  & \quad \left| '\left(\quad \text{formula binary\_boolean\_connective formula'}\right)' \right. \\
  & \quad \left| '!\quad \text{[' variable ']}\quad \text{formula} \quad \text{(\forall)} \right. \\
  & \quad \left| '?\quad \text{[' variable ']}\quad \text{formula} \quad \text{(\exists)} \right. 
  \end{align*}\]
Truth in First-Order Logic

- formulae/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 $\rightarrow$ *objects*
  - predicate symbols $\rightarrow$ *relations*
  - function symbols $\rightarrow$ *functional relations*
- an atomic formula/sentence $\text{predicate}(\text{term}_1, \ldots, \text{term}_n)$ is *true* iff the *objects* referred to by $\text{term}_1, \ldots, \text{term}_n$ are in the *relation* referred to by predicate
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 $\forall Y \exists X$
  - $\exists X \forall Y \text{loves}(X, Y)$ “There is a person who loves everyone in the world”
  - $\forall X \exists Y \text{loves}(X, Y)$ “Everyone in the world is loved by at least one person”
- Quantifier duality: each can be expressed using the other
  - $\forall X \text{likes}(X, \text{icecream})$ is the same as $\neg \exists X \neg \text{likes}(X, \text{icecream})$
  - $\exists X \text{likes}(X, \text{broccoli})$ is the same as $\neg \forall X \neg \text{likes}(X, \text{broccoli})$
Properties of Quantifiers

- \( \neg \forall X \varphi(X) \) is the same as \( \exists X \neg \varphi(X) \)
- if \( X \) is not free in \( \psi \) then
  - \( \forall X (\psi \lor \varphi(X)) \) is the same as \( \psi \lor \forall X \varphi(X) \)
  - \( \forall X (\psi \land \varphi(X)) \) is the same as \( \psi \land \forall X \varphi(X) \)
  - \( \forall X (\psi \implies \varphi(X)) \) is the same as \( \psi \implies \forall X \varphi(X) \)
  - \( \forall X (\varphi(X) \implies \psi) \) is the same as \( (\exists X \varphi(X)) \implies \psi \)
Substitution

- a substitution is a total mapping $\sigma : V \rightarrow T$ from variables to terms
- the notation $\{x_1 \mapsto t_1, ..., x_k \mapsto t_k\}$ refers to a substitution mapping each variable $x_i$ to the corresponding term $t_i$, for $i = 1, ..., k$, and every other variable to itself; the $x_i$ must be pairwise distinct.
- applying a substitution to a term $t$ is written in postfix notation as $t\{x_1 \mapsto t_1, ..., x_k \mapsto t_k\}$ it means to simultaneously replace every occurrence of each $x_i$ in $t$ by $t_i$
- the result $t\sigma$ of applying a substitution $\sigma$ to a term $t$ is called an instance of that term $t$.
- a substitution $\sigma$ is called a renaming substitution if it is a permutation on the set of all variables
Resolution in First-Order Logic

resolution rule

\[
\frac{\triangle \lor L \quad \neg K \lor \Gamma}{\triangle \Theta \lor \Gamma \Theta}
\]

where:

- \(\triangle\) and \(\Gamma\) are disjunctions of arbitrary many literals
- clauses \((\triangle \lor L)\) and \((\neg K \lor \Gamma)\) have no common variables; otherwise we have to rename such variables in one of the clauses
- \(L\) and \(K\) are atomic formulae
- \(\Theta\) is a most general unifier (mgu) of \(L\) and \(K\)
Resolution in First-Order Logic

- factorization rule

\[
\begin{align*}
\Delta \lor L \lor K \\
\Delta \Theta \lor L\Theta
\end{align*}
\]

where:

- \( \Delta \) is disjunction of arbitrary many literals
- \( L \) and \( K \) are literals
- \( \Theta \) is a most general unifier (mgu) of \( L \) and \( K \)
Skolemization

- A formula of first-order logic is in *Skolem normal form* if it is in prenex normal form with only universal first-order quantifiers.
- A formula is in *prenex normal form* if it is written as a string of quantifiers (prefix) followed by a quantifier-free part.
- Every first-order formula may be converted into *Skolem normal form* while not changing its satisfiability via a process called *Skolemization*; the resulting formula is not necessarily equivalent to the original one, but is equisatisfiable with it: it is satisfiable iff the original one is satisfiable.