Predicate Logic

Lunjin Lu

Topics

- Syntax
- Semantics
- Clausal Form
- Resolution
- Substitutions

Formalization

- Predicate logic is a formalization of declarative sentences

1. Every mother loves her children
2. Mary is a mother and Tom is Mary’s child
3. Mary loves Tom

(i) $\forall X(\forall Y(((mother(X) \wedge child\_of(Y,X)) \rightarrow loves(X,Y)))$

(ii) $mother(mary) \wedge child\_of(tom,mary)$

(iii) $loves(mary,tom)$

Alphabet ***A***

- Variables X ranging over ***V***
- Function symbols f ranging over Σ
- Predicate symbols p ranging over Π
- Logical connectives: $\neg \wedge \vee \rightarrow \leftrightarrow$
- Quantifiers: $\forall \; \exists$
- Auxiliary symbols (,)

Terms and Formulae

- Terms t ranging over ***T***

  $t ::= X \mid f(t_1,\ldots,t_n)$

- Formulae F ranging over ***Wff***

  $F ::= p(t_1,\ldots,t_n) \mid (\neg F_1) \mid (F_1 \wedge F_2)$
  $\mid (F_1 \vee F_2) \mid (F_1 \rightarrow F_2) \mid (F_1 \leftrightarrow F_2)$
  $\mid (\forall X\, F_1) \mid (\exists X\, F_1)$

- Closed formulae do not have ***free*** variables

Interpretation

- An interpretation *J* of an alphabet ***A*** is a non-empty domain *D* (|*J*|) and a mapping that associate
  - an n-ary function symbol f in ***A*** with a function f_J: $D^n \rightarrow D$
  - an n-ary predicate symbol p in ***A*** with a relation $p_J \subseteq D^n$

Semantics of Terms

- A valuation φ is a mapping from V to |J|.
- The meaning φ_J of t with respect to J and φ
  - $\varphi_J(X) = \varphi(X)$
  - $\varphi_J(f(t_1,\ldots,t_n)) = f_J(\varphi_J(t_1),\ldots, \varphi_J(t_n))$

Logical Consequences

- An interpretation J is a model of a set of closed formulae P iff every formula in P is true in J
- A formula F is a logical consequence of a set of formulae iff F is true in ***every*** model of P

Semantics of Formulae

- $J \models_{\varphi} p(t_1,\ldots,t_n)$ iff $< \varphi_J(t_1),\ldots, \varphi_J(t_n)>$ is in p_J
- $J \models_{\varphi} (\neg F)$ iff $J \not\models_{\varphi} F$
- $J \models_{\varphi} (F_1 \wedge F_2)$ iff $J \models_{\varphi} F_1$ and $J \models_{\varphi} F_2$
- $J \models_{\varphi} (F_1 \vee F_2)$ iff $J \models_{\varphi} F_1$ or $J \models_{\varphi} F_2$
- $J \models_{\varphi} (F_1 \rightarrow F_2)$ iff $J \models_{\varphi} F_2$ whenever $J \models_{\varphi} F_1$
- $J \models_{\varphi} (F_1 \leftrightarrow F_2)$ iff $J \models_{\varphi} (F_1 \rightarrow F_2)$ and $J \models_{\varphi} (F_2 \rightarrow F_1)$
- $J \models_{\varphi} (\forall X\ F_1)$ iff $J \models_{\varphi[X\rightarrow t]} F_1$ for **every** t in |J|
- $J \models_{\varphi} (\exists X\ F_1)$ iff $J \models_{\varphi[X\rightarrow t]} F_1$ for **some** t in |J|

- Write J |= F if F is a closed formula

Clausal forms

- A literal L is an atom or negation of an atom
- A clause is of the form $\forall\ (L_1 \vee L_2 \vee \ldots L_k)$ with all variable universally quantified
- Any first predicate logic formula **is logically equivalent to** a conjunction of clauses

Modus Pollens

- To prove a closed formula F is a logical consequence of a set P of closed formulae
- It is equivalent to proving that $\{\neg F\} \cup P$ is un-satisfiable.

Resolution

- If $\{\neg F\} \cup P$ is in clausal form then it can be done mechanically by repeated applying resolution to derive an empty clause (false).
- Let $\forall\ (L_1 \vee L_2 \vee \ldots L_n)$ and $\forall\ (K_1 \vee K_2 \vee \ldots K_m)$ be clauses that do not share variables, $L_i = p(\underline{t})$ and $K_j = \neg p(\underline{s})$ such that $\theta(\underline{t}) = \theta(\underline{s})$ then $\theta(\bigvee_{o \neq i} L_o \vee \bigvee_{o \neq j} K_o)$ is a logical consequence of $L_1 \vee L_2 \vee \ldots L_n$ and $K_1 \vee K_2 \vee \ldots K_m$