PART II.

Predicate logic

FIRST ORDER LOGIC

Logic in computer science

Seminar: INGK401-K5; INHK401; INJK401-K4
University of Debrecen, Faculty of Informatics
kadek.tamas@inf.unideb.hu

Alphabets

- Logical symbols:
  - logical connectives $\neg$, $\wedge$, $\vee$, $\supset$,
  - quantifiers: $\forall$, $\exists$,
  - individuum variables: $x, y, z, \ldots$ (lowercase english letters).
- Separator symbols: circle brackets and the comma.
- Non-logical symbols: $L = \langle Srt, Pr, Fn, Cnst \rangle$ where
  - Srt is a nonempty set of sorts (or types),
  - Pr is a set of predicate symbols,
  - Fn is a set of function symbols.
  - $Cnst$ is a set of constant symbols.

Signature

Each individuum variable belongs to a type.
The **signature** is a triple of function $\langle \nu_1, \nu_2, \nu_3 \rangle$ where

- ν_1 assigns a list of type to each $P \in Pr$,
- ν_2 assigns a list of type to each $f \in Fn$,
- ν_3 assigns a type to each $c \in Cnst$.

π type terms

1. x, if x is a π type variable,
2. c, if $c \in Cnst$ and $\nu_3(c) = (\pi)$,
3. $f(t_1, t_2, \ldots, t_n)$, if $f \in Fn$ and $\nu_2(f) = (\pi_1, \pi_2, \ldots, \pi_n, \pi)$ and $t_1, t_2, \ldots, t_n$ are $\pi_1, \pi_2, \ldots, \pi_n$ type terms,
4. A string of symbols is a term if and only if it can be obtained by starting with variables (1) or constants (2) repeatedly applying the inductive steps (3), and it must terminates after a finite number of steps.

Formulas

- $P(t_1, t_2, \ldots, t_n)$ is an atomic formula, if $P \in Pr$ and $\nu_1(P) = (\pi_1, \pi_2, \ldots, \pi_n)$ and $t_1, t_2, \ldots, t_n$ are $\pi_1, \pi_2, \ldots, \pi_n$ type terms,
- $\neg A$, if A is an arbitrary formula,
- $(A \circ B)$, if A and B are an arbitrary formulas and $\circ \in \{\wedge, \vee, \supset\}$,
- $\exists x A$, if A is an arbitrary formula and x is a variable,
- $\forall x A$, if A is an arbitrary formula and x is a variable,
- A string of symbols is a formula if and only if it can be obtained by starting with atomic formulas, repeatedly applying the inductive steps, and it must terminates after a finite number of steps.

Syntactical properties

Immediate subterms:

1. A constant or a variable has no immediate subterm.
2. The immediate subterms of $f(t_1, t_2, \ldots, t_n)$ are the t_1, t_2, ..., t_n terms.

Let T be a term. The **set of subterms** of T is the

- smallest set
- that contains T,
- and contains, with each member, the immediate subterms of that member.

Syntactical properties

The **functional degree** of a T term is $\tilde{d}(T)$:

1. $\tilde{d}(T) = 0$, if T is a variable or a constatnt, otherwise
2. $\tilde{d}(f(t_1, t_2, \ldots, t_n)) = \tilde{d}(t_1) + \tilde{d}(t_2) + \ldots + \tilde{d}(t_n) + 1$.

Syntactical properties

Immediate subformulas:

1. An atomic formula has no immediate subformulas.
2. The only immediate subformula of $\neg A$ is A.
3. The immediate subformulas of $(A \circ B)$ are A and B, where $\circ \in \{\wedge, \vee, \supset\}$.
4. The immediate subformulas of QxA is A, where $Q \in \{\forall, \exists\}$ and x is a variable.

Let A be a formula. The **set of subformulas** of A is the

- smallest set
- that contains A,
- and contains, with each member, the immediate subformulas of that member.

Syntactical properties

The **logical degree** of an A formula is $d(A)$:

1. $d(A) = 0$, if A is an atomic formula,
2. $d(\neg A) = d(A) + 1$, where A is an arbitrary formula,
3. $d(A \circ B) = d(A) + d(B) + 1$, where A and B are arbitrary formulas, and $\circ \in \{\wedge, \vee, \supset\}$.
4. $d(QxA) = d(A) + 1$, where A is an arbitrary formula, x is a variable, and $Q \in \{\forall, \exists\}$.

Syntactical properties

The **free-variable occurrences** in a formula:

1. If A is an atomic formula, then all the variable occurrences in A are free-variable occurrences.
2. The free-variable occurrences in $\neg A$ are free-variable occurrences in A.
3. The free-variable occurrences in $(A \circ B)$ are the free-variable occurrences in A and the free-variable occurrences in B, where $\circ \in \{\wedge, \vee, \supset\}$.
4. The free-variable occurrences in $\forall x A$ and $\exists x A$ are free-variable occurrences in A, except for occurrences of x.

A variable occurrence is called **bound** if it is not free.

A **closed formula** (**sentence**) is a formula with no free-variable occurrences.

Bound variable renaming, clean formula

$\left[A_y^x\right]$ denotes, that we replace each free occurrence of the x variable in the A formula with an y variable, where x and y are in the same type.

Regular bound variable renaming
If y has no free occurrences in the A formula, and there is no occurrences of x within the scope of a quantifier bounding y, then the following renamings are regular:

$$\begin{aligned} \forall x A \quad &\Rightarrow \forall y \left[A_y^x\right] \\ \exists x A \quad &\Rightarrow \exists y \left[A_y^x\right] \end{aligned}$$

A formula called **clean**, if

- a variable has just free or just bound occurrences, not both, and
- each different quantifier bounds different variable.

Semantics

Let $L = \langle Srt, Pr, Fn, Cnst \rangle$ be a first-order language. The **interpretation** of L is an ordered 4-tuple of functions:

$$I = \langle I_{Srt}, I_{Pr}, I_{Fn}, I_{Cnst} \rangle$$

- The I_{Srt} function assigns a non empty set to each $\pi \in Srt$ sort.

$$I_{Srt}(\pi) = D_\pi \quad \text{where} \quad D_\pi \neq \emptyset$$

- the I_{Pr} function assigns a P^I logical function to each $P \in Pr$ prediacte symbol. If $\nu_1(P) = (\pi_1, \pi_2, \ldots \pi_n)$ then

$$P^I : D_{\pi_1} \times D_{\pi_2} \times \ldots \times D_{\pi_n} \to \{true, false\}.$$

- The I_{Fn} function assigns an f^I mathematical function to each $f \in Fn$ function symbol. If $\nu_2(f) = (\pi_1, \pi_2, \ldots \pi_n, \pi)$ then

$$f^I : D_{\pi_1} \times D_{\pi_2} \times \ldots \times D_{\pi_n} \to D_\pi.$$

- the I_{Cnst} function assigns a c^I object to each $c \in Cnst$ costatnt symbol. If $\nu_3(c) = (\pi)$ then $c^I \in D_\pi$.

A **varible substitution** is a κ function which assigns an object to each variable. If the type of the x variable is π, then $\kappa(x) \in D_\pi$.

A κ' variable substitution called x **variant** of the κ variable substitution, if $\kappa(y) = \kappa'(y)$ is true for all y variable different than x.

Term valuation

We denote the **term valuation** of a t term by a given I interpretation and κ variable substitution as $|t|^{I,\kappa}$. It is defined as a recursive function:

- $|x|^{I,\kappa} = \kappa(x)$ where x is a variable,
- $|c|^{I,\kappa} = I_{Cnst}(c)$ where $c \in Cnst$,
- $|f(t_1, t_2, \ldots, t_n)|^{I,\kappa} = f^I(|t_1|^{I,\kappa}, |t_2|^{I,\kappa}, \ldots, |t_n|^{I,\kappa})$ where $I_{Fn}(f) = f^I$.

Formula valuation

We denote the **formula valuation** of an F formula by a given I interpretation and κ variable substitution as $|F|^{I,\kappa}$. It is defined as a recursive function:

- $|P(t_1, t_2, \ldots, t_n)|^{I,\kappa} = P^I(|t_1|^{I,\kappa}, |t_2|^{I,\kappa}, \ldots, |t_n|^{I,\kappa})$ if $I_{Pr}(P) = P^I$
- $|\neg A|^{I,\kappa} = \dot{\neg}\, |A|^{I,\kappa}$
- $|(A \supset B)|^{I,\kappa} = |A|^{I,\kappa} \dot{\supset} |B|^{I,\kappa}$
- $|(A \wedge B)|^{I,\kappa} = |A|^{I,\kappa} \dot{\wedge} |B|^{I,\kappa}$
- $|(A \vee B)|^{I,\kappa} = |A|^{I,\kappa} \dot{\vee} |B|^{I,\kappa}$
- $|\exists x A|^{I,\kappa} = \begin{cases} true & \text{if } |A|^{I,\kappa'} = true \text{ for at least one } \kappa' \text{ } x\text{-variant of } \kappa \\ false & \text{otherwise} \end{cases}$
- $|\forall x A|^{I,\kappa} = \begin{cases} true & \text{if } |A|^{I,\kappa'} = true \text{ for all } \kappa' \text{ } x\text{-variant of } \kappa \\ false & \text{otherwise} \end{cases}$

Semantical properties

An A formula of the L first order language is called **first-order tautology**, when $|A|^{\mathcal{I},\kappa} = true$ for all possible $\mathcal{I}$ interpretation of the L language and all possible κ variable substiturion of the $\mathcal{I}$ interpretation. We shortly denote this by $\models A$.

An A formula of the L first order language is called **first-order contradiction**, when $|A|^{\mathcal{I},\kappa} = false$ for all possible $\mathcal{I}$ interpretation of the L language and all possible κ variable substiturion of the $\mathcal{I}$ interpretation. We shortly denote this by $\models A$.

Let A and B formulas of the same L first order language. A and B are **logically equivalent**, if

$$|A|^{I,\kappa} = |B|^{I,\kappa}$$

for all possible $\mathcal{I}$ interpretation of the L language and all possible κ variable substiturion of the $\mathcal{I}$ interpretation.

First order consequence

Let $A_1, A_2, \dots, A_n$ $(n \geqslant 1)$ and B formulas of the same L first order language. We say that B is a **first-order consequence** of $A_1, A_2, \dots, A_n$, if B is true in every interpretation for every valuation, where every formula of $A_1, A_2, \dots, A_n$ is true. We denote this as $A_1, A_2, \dots, A_n \vDash B$.

$$A_1, A_2, \dots, A_n \vDash B \quad \text{if and only if} \quad \vDash A_1, A_2, \dots, A_n \supset B$$

$$A_1, A_2, \dots, A_n \vDash B \quad \text{if and only if} \quad \Dashv A_1, A_2, \dots, A_n \wedge \neg B$$

Prenex form

A formula is in a prenex form, if it is in the form of

$$Q_1 x_1 Q_2 x_2 \ldots Q_n x_n A \quad (n \geqslant 0)$$

where $Q_i \in \{\exists, \forall\}$ and x_i is a variable and A is a formula without quantifiers.

- It is a prenex conjunctive normal form (PCNF), if A is a conjunctive normal form,
- it is a prenex disjunctive normal form (PDNF), if A is a disjunctive normal form.

Algorithm to create prenex forms

1. Clean the formula,
2. use the following rules:

$$\neg\forall x A \sim \exists x \neg A \qquad \neg\exists x A \sim \forall x \neg A$$

$$\forall x A \vee B \sim \forall x(A \vee B) \qquad A \vee \forall x B \sim \forall x(A \vee B)$$

$$\forall x A \wedge B \sim \forall x(A \wedge B) \qquad A \wedge \forall x B \sim \forall x(A \wedge B)$$

$$\exists x A \vee B \sim \exists x(A \vee B) \qquad A \vee \exists x B \sim \exists x(A \vee B)$$

$$\exists x A \wedge B \sim \exists x(A \wedge B) \qquad A \wedge \exists x B \sim \exists x(A \wedge B)$$

$$\forall x A \supset B \sim \exists x(A \supset B) \qquad A \supset \forall x B \sim \forall x(A \supset B)$$

$$\exists x A \supset B \sim \forall x(A \supset B) \qquad A \supset \exists x B \sim \exists x(A \supset B)$$