Logic and theory of computation

4th lecture

08-10-2018

Syntax of first-order logic

FIRST-ORDER LOGIC

First-order logic is an extension of propositional logic that includes predicates interpreted as relations on a domain.

Symbols of first-order logic

Let $\mathcal{P}$, $\mathcal{F}$, $\mathcal{A}$ and $\mathcal{V}$ be countable sets of *predicate symbols, function symbols, constant symbols* and *variables*. Each predicate symbol $p^n \in \mathcal{P}$ and function symbol $f^n \in \mathcal{F}$ is associated with an arity, the number $n \geq 1$ of arguments that it takes. p^n is called an n-ary predicate (symbol), while f^n is called an n-ary function (symbol).

For $n = 1, 2$ we can use unary and binary respectively for n-ary.

Terms

Terms are defined recursively as follows:

Terms

- A variable or a constant is a term.
- If f^n is an n-ary function symbol $(n \geq 0)$ and $t_1, t_2, \ldots, t_n$ are terms, then $f^n(t_1, t_2, \ldots, t_n)$ is a term.

Note, that 0-ary functions and constants are basically the same. The superscript denoting the arity of the function will not be written since the arity can be inferred from the number of arguments.

Example:
Let $f, g \in \mathcal{F}$ be a binary and a unary function symbol, respectively. Let $a \in \mathcal{A}$ be a constant and $x, y \in \mathcal{V}$ be variables.
The following strings are terms:
$a, y, f(x, y), g(g(x)), f(g(f(x, y)), a)$.
The following strings are not: $f(f(x), x), g(x, x, x)$.

Formula

Atomic formula

An atomic formula is an n-ary predicate followed by a list of n arguments in parentheses $p(t_1, t_2, \ldots, t_n)$ where each argument t_i is a term.

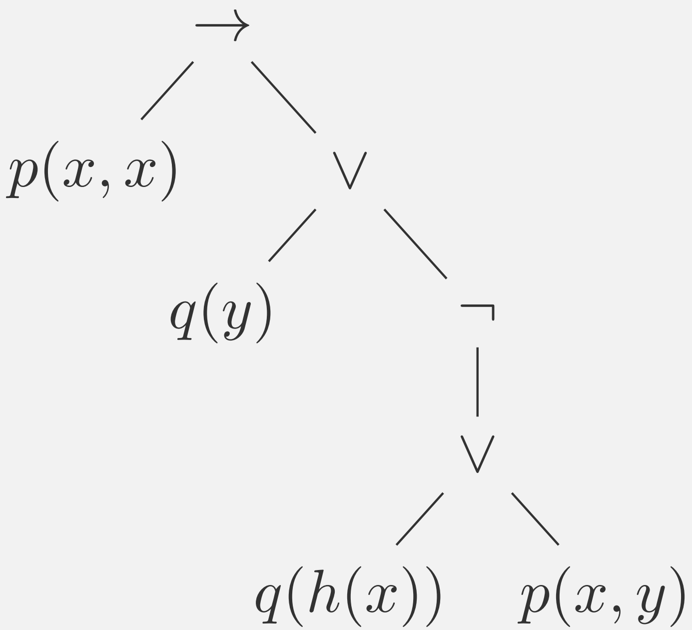
A formula in first-order logic is a tree defined recursively as follows.

Formula

- A formula is a leaf labeled by an atomic formula.
- A formula is a node labeled by $\neg$ with a single child that is a formula.
- A formula is a node labeled by a binary Boolean operator $(\wedge, \vee, \rightarrow)$ with two children both of which are formulas.
- A formula is a node labeled by $\forall x$ or $\exists x$ (for some variable x) with a single child that is a formula.

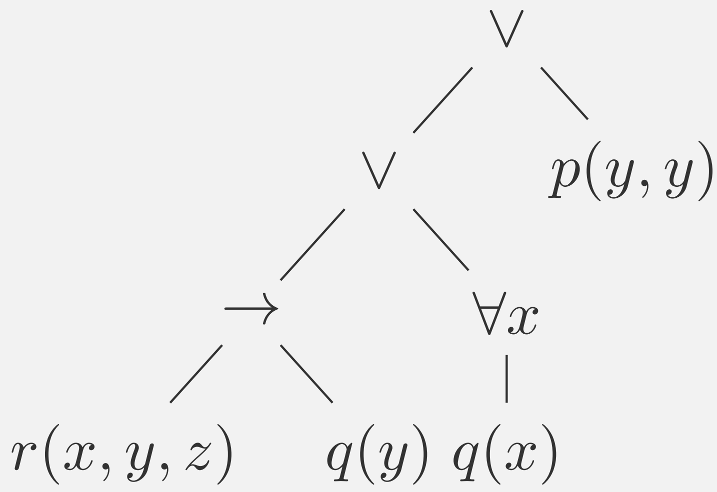
Example: formula

Let p be a binary, q be a unary and r be a 3-ary predicate symbol. Let h be a unary function symbol and let x, y, z be variables.



String representation:

$$(p(x,x) \to (q(y) \vee (\neg(q(h(x)) \vee p(x,y)))))$$



String representation:

$$(((r(x,y,z) \to q(y)) \vee (\forall x q(x))) \vee p(y,y))$$

Subformula, principal operator, leaving parantheses, scope

$\forall$ is the universal quantifier and is read *for all*. $\exists$ is the existential quantifier and is read *there exists*.

A formula of the form $\forall x A$ is called a *universal formula*. Similarly, a formula of the form $\exists x A$ is an *existential formula*.

Subformula and principal operator: same as in prop. logic.

Leaving parantheses: quantifiers are considered to have the same precedence as negation and a higher precedence than the binary operators, otherwise the same.

Scope

A universal or existential formula $\forall x A$ or $\exists x A$ is a *quantified formula*, x is called a *quantified variable* and its *scope* is the formula A.

It is not required that x actually appear in the scope of its quantification.

Free and bound variables

Free and bound variable

Let A be a formula. $x \in \mathcal{V}$ is a *free variable* of A iff x has a non-quantified occurance in A, such that x is not within the scope of a quantified variable x. A variable which is not free is called *bound*.

Example: $A = \forall x(\exists y(p(x, y) \to q(x)) \wedge q(y))$.
The scope of $\exists y$ is $p(x, y) \to q(x)$.
The scope of $\forall x$ is $\exists y(p(x, y) \to q(x)) \wedge q(y)$.
Both occurance of x in A is in the scope of $\forall x$. So x is a bound variable of A.
The second occurance of y is not in the scope of an $\exists y$ or a $\forall y$, so y is a free variable of A.

Closed formula

Closed formula

If a formula has no free variables, it is *closed*.

Closures of a non-closed formula

If $x_1, \ldots, x_n$ are all the free variables of A, the *universal closure* of A is $\forall x_1 \cdots \forall x_n A$ and the *existential closure* is $\exists x_1 \cdots \exists x_n A$.
$A(x_1, \ldots, x_n)$ indicates that the set of free variables of the formula A is a subset of $\{x_1, \ldots, x_n\}$.

Example: $A = \forall x(\exists y(p(x,y) \to q(x)) \wedge q(y))$.
y is a free variable of A, so $A = A(y)$ is not closed,
Existential closure of $A(y)$:
$\exists y A(y) = \exists y \forall x(\exists y(p(x,y) \to q(x)) \wedge q(y))$.
Universal closure of $A(y)$:
$\forall y A(y) = \forall y \forall x(\exists y(p(x,y) \to q(x)) \wedge q(y))$.

Semantics of first-order logic

Interpretation

Interpretation

Let U be a set of formulas such that $\{p_1, \ldots, p_k\}$ are all the predicate symbols, $\{f_1, \ldots, f_\ell\}$ are all the funtion symbols and $\{a_1, \ldots, a_m\}$ are all the constants appearing in U. An *interpretation* $\mathcal{I}$ for U is a 4-tuple:

$$(D, \{R_1, \ldots, R_k\}, \{F_1, \ldots, F_\ell\}, \{d_1, \ldots, d_m\}),$$

consisting of a non-empty set D called the *domain*, an assignment of an n_i-ary relation R_i on D to the n_i-ary predicate symbol p_i $(1 \leq i \leq k)$, an assignment of an n_j-ary function F_j on D to the n_j-ary function symbol f_j $(1 \leq j \leq \ell)$, and an assignment of an element $d_n \in D$ to the constant a_n $(1 \leq n \leq m)$.

If $U = \{A\}$, we say that $\mathcal{I}$ is an interpretation for A.

Interpretation – examples

Here are three interpretations for the formula $\forall x p(a, x)$:

$\mathcal{I}_1 = (\mathbb{N}, \{\leq\}, \{\}, \{0\})$,
$\mathcal{I}_2 = (\mathbb{N}, \{\leq\}, \{\}, \{1\})$,
$\mathcal{I}_3 = (\mathbb{Z}, \{\leq\}, \{\}, \{0\})$.
The domain is either $\mathbb{N}$, the set of natural numbers, or $\mathbb{Z}$, the set of integers.
The binary relation $\leq$ (less-than-or-equal-to) is assigned to the binary predicate p and either 0 or 1 is assigned to the constant a.

The formula can also be interpreted over strings:
$\mathcal{I}_4 = (\mathcal{S}, \{\sqsubseteq\}, \{\}, \varepsilon)$.
The domain $\mathcal{S}$ is a set of strings, $\sqsubseteq$ is the binary relation such that $(s_1, s_2) \in \sqsubseteq$ iff s_1 is a substring of s_2 , and ε is the empty string of length 0.
Note, that no function was needed in the interpretations.

Evaluating terms

Assignment

Let $\mathcal{I}$ be an interpretation for a formula A. An *assignment* $\sigma_{\mathcal{I}} : \mathcal{V} \to D$ is a function which maps every free variable $v \in V$ to an element $d \in D$, where D is the domain of $\mathcal{I}$.

In a given interpretation $\mathcal{I}$ we may write σ for $\sigma_{\mathcal{I}}$.
Let us define $\mathcal{D}_{\mathcal{I},\sigma}(t)$, the value of a term t in the interpretation $\mathcal{I}$ and assignment σ.

Evaluating terms

- for a constant $a \in \mathcal{A}$ that is interpreted for $d \in D$ let $\mathcal{D}_{\mathcal{I},\sigma}(a) = d$,
- for a variable $v \in \mathcal{V}$ let $\mathcal{D}_{\mathcal{I},\sigma}(v) = \sigma(v)$,
- for a term $f(t_1, \ldots, t_n)$ where f is interpreted for F let $\mathcal{D}_{\mathcal{I},\sigma}(f(t_1, \ldots, t_n)) = F(\mathcal{D}_{\mathcal{I},\sigma}(t_1), \ldots \mathcal{D}_{\mathcal{I},\sigma}(t_n))$.

Evaluating terms – example

Example:

Let $t = f(f(x, g(a)), g(y))$ be a term. Consider the interpretations

$\mathcal{I}_5 = (\mathbb{N}, \{\}, \{+, next\}, \{0\})$,

$\mathcal{I}_6 = (\{0,1\}, \{\}, \{+_{\text{mod } 2}, next_{\text{mod } 2}\}, \{0\})$,
where $next(x)$ assigns the next number to x, e.g., 13 for 12.

Let $\sigma(x) = 7, \sigma(y) = 5$. Then $\mathcal{D}_{\mathcal{I}_5,\sigma}(t) =$14.

Let $\sigma'(x) = 1, \sigma'(y) = 0$. Then $\mathcal{D}_{\mathcal{I}_6,\sigma'}(t) =$1.

Note, that the result is always an element of the respective domain.

For an assignment σ for an interpretation $\mathcal{I}$, variable x and $d \in D$ let $\sigma[x \leftarrow d]$ denote the assignment that is the same as σ except that x is mapped to d.

Truth value of a formula of first-order logic

Truth value of a formula of first-order logic

Let A be a formula, $\mathcal{I}$ an interpretation and $\sigma_{\mathcal{I}}$ an assignment. $v_{\mathcal{I},\sigma}(A)$, the truth value of A under $\mathcal{I}$ and $\sigma_{\mathcal{I}}$, is defined by recursion on the structure of A (we simplified the notation by writing $v_{\mathcal{I},\sigma}$ for $v_{\mathcal{I},\sigma}$):

- Let $A = p_k(t_1, \ldots, t_n)$ be an atomic formula where each t_i is a term. $v_{\mathcal{I},\sigma}(A) = T$ iff $(\mathcal{D}_{\mathcal{I},\sigma}(t_1), \ldots, \mathcal{D}_{\mathcal{I},\sigma}(t_n)) \in R_k$ where R_k is the relation assigned by $\mathcal{I}_A$ to p_k.
- $v_{\mathcal{I},\sigma}(\neg A_1) = T$ iff $v_{\mathcal{I},\sigma}(A_1) = F$.
- $v_{\mathcal{I},\sigma}(A_1 \vee A_2) = T$ iff $v_{\mathcal{I},\sigma}(A_1) = T$ or $v_{\mathcal{I},\sigma}(A_2) = T$, and similarly for the other Boolean operators.
- $v_{\mathcal{I},\sigma}(\forall x A_1) = T$ iff $v_{\mathcal{I},\sigma[x\leftarrow d]}(A_1) = T$ for all $d \in D$.
- $v_{\mathcal{I},\sigma}(\exists x A_1) = T$ iff $v_{\mathcal{I},\sigma[x\leftarrow d]}(A_1) = T$ for some $d \in D$.

Truth value of a formula – examples

$\mathcal{I}_1 = (\mathbb{N}, \{\leq\}, \{\}, \{0\}),$
$\mathcal{I}_2 = (\mathbb{N}, \{\leq\}, \{\}, \{1\}),$
$\mathcal{I}_3 = (\mathbb{Z}, \{\leq\}, \{\}, \{0\}).$
$\mathcal{I}_4 = (\mathcal{S}, \{\sqsubseteq\}, \{\}, \varepsilon).$

Example 1 : Let $\sigma(x) = 7,\ \sigma(y) = 3$
$v_{\mathcal{I}_1,\sigma}(p(a,x) \rightarrow p(x,x)) = T \rightarrow T = T.$
$v_{\mathcal{I}_1,\sigma}(\neg p(x,y) \rightarrow p(x,x) \wedge p(y,a)) = \neg F \rightarrow T \wedge F = T \rightarrow F = F.$

Example 2 : $A = \forall x p(a,x)$:
$v_{\mathcal{I}_1,\sigma}(A) =$T $\quad \forall x \in \mathbb{N} : 0 \leq x$
$v_{\mathcal{I}_2,\sigma}(A) =$F $\quad \forall x \in \mathbb{N} : 1 \leq x$
$v_{\mathcal{I}_3,\sigma}(A) =$F $\quad \forall x \in \mathbb{Z} : 0 \leq x$
$v_{\mathcal{I}_4,\sigma}(A) =$T $\quad \forall x \in \mathcal{S} : \varepsilon \sqsubseteq x$

Truth value of a closed formula

Theorem

Let A be a closed formula and let $\mathcal{I}$ be an interpretation for A. Then $v_{\mathcal{I},\sigma}(A)$ does not depend on σ.

Theorem

Let $A = A(x_1, \ldots, x_n)$ be a (non-closed) formula with free variables $x_1, \ldots, x_n$, and let $\mathcal{I}$ be an interpretation. Then:

- $v_{\mathcal{I},\sigma}(A) = T$ for some assignment σ iff $v_{\mathcal{I}}(\exists x_1 \cdots \exists x_n A) = T$.
- $v_{\mathcal{I},\sigma}(A) = T$ for all assignments σ iff $v_{\mathcal{I}}(\forall x_1 \cdots \forall x_n A) = T$.

Semantic properties of formulas

ONLY for closed formulas

Let A be a closed formula of first-order logic.

- A is *true* in $\mathcal{I}$ or $\mathcal{I}$ is a model for A iff $v_{\mathcal{I}}(A) = T$.
  Notation: $\mathcal{I} \models A$.
- A is *valid* if for all interpretations $\mathcal{I}$, $\mathcal{I} \models A$. Notation: $\models A$.
- A is *satisfiable* if for some interpretation $\mathcal{I}$, $\mathcal{I} \models A$.
- A is *unsatisfiable* if it is not satisfiable.
- A is *falsifiable* if it is not valid.

A set of closed formulas $U = \{A_1, \ldots\}$ is *(simultaneously) satisfiable* iff there exists an interpretation $\mathcal{I}$ such that $v_{\mathcal{I}}(A_i) = T$ for all i. The satisfying interpretation is a model of U.

U is *valid* iff for every interpretation $\mathcal{I}$, $v_{\mathcal{I}}(A_i) = T$ for all i.

Logical equivalence and consequence

Logical equivalence

A_1 is logically equivalent to A_2 iff $v_{\mathcal{I}}(A_1) = v_{\mathcal{I}}(A_2)$ for all interpretations $\mathcal{I}$ for $\{A_1, A_2\}$. Notation: $A_1 \equiv A_2$.

Logical consequence

Let A be a closed formula and U a set of closed formulas. A is a logical consequence of U iff for all interpretations $\mathcal{I}$ for $U \cup \{A\}$, $v_{\mathcal{I}}(A_i) = T$ for all $A_i \in U$ implies $v_{\mathcal{I}}(A) = T$. Notation: $U \models A$.

Similarly to propositional logic:

Theorem

Let $U = \{A_1, \ldots, A_n\}$ and A be a formula.

$$\begin{aligned} U \models A \quad & \Leftrightarrow \quad \models A_1 \land \cdots \land A_n \to A \\ & \Leftrightarrow \quad A_1 \land \cdots \land A_n \land \neg A \text{ is unsatisfiable.} \end{aligned}$$

Laws of first-order logic

- $\forall x \forall y A \equiv \forall y \forall x A$,
- $\exists x \exists y A \equiv \exists y \exists x A$,
- $\neg \exists x A \equiv \forall x \neg A$,
- $\neg \forall x A \equiv \exists x \neg A$,
- $\forall x A \wedge \forall x B \equiv \forall x (A \wedge B)$
- $\exists x A \vee \exists x B \equiv \exists x (A \vee B)$.

- $\models \exists x \forall y A(x, y) \to \forall y \exists x A(x, y)$
- $\models \forall x A(x) \vee \forall x B(x) \to \forall x (A(x) \vee B(x))$,
- $\models \exists x (A(x) \wedge B(x)) \to \exists x A(x) \wedge \exists x B(x)$.

Laws of first-order logic II.

If x is not free in B

- $\exists x A(x) \vee B \equiv \exists x(A(x) \vee B)$,
- $\forall x A(x) \vee B \equiv \forall x(A(x) \vee B)$,
- $B \vee \exists x A(x) \equiv \exists x(B \vee A(x))$,
- $B \vee \forall x A(x) \equiv \forall x(B \vee A(x))$,
- $\exists x A(x) \wedge B \equiv \exists x(A(x) \wedge B)$,
- $\forall x A(x) \wedge B \equiv \forall x(A(x) \wedge B)$,
- $B \wedge \exists x A(x) \equiv \exists x(B \wedge A(x))$,
- $B \wedge \forall x A(x) \equiv \forall x(B \wedge A(x))$.

There are further laws in the book.

An example for proving equivalence by definition

Let us prove that $\forall x A(x) \equiv \neg \exists x \neg A(x)$.

For an arbitrary interpretation $\mathcal{I}$ and assignment σ

$v_{\mathcal{I},\sigma}(\forall x A(x)) = T$

$\Leftrightarrow v_{\mathcal{I},\sigma[x \leftarrow d]} A(x) = T$ for all $d \in D$.

$\Leftrightarrow v_{\mathcal{I},\sigma[x \leftarrow d]} \neg A(x) = F$ for all $d \in D$.

$\Leftrightarrow$ there is no $d \in D$, such that $v_{\mathcal{I},\sigma[x \leftarrow d]} \neg A(x) = T$.

$\Leftrightarrow v_{\mathcal{I},\sigma}(\exists x \neg A(x)) = F$.

$\Leftrightarrow v_{\mathcal{I},\sigma}(\neg \exists x \neg A(x)) = T$.