Semantics of first-order logic

Takafumi Saikawa

November 21, 2016

Abstract

We introduce the standard semantics of first-order logic. This is a supplementary note to the Sicss Logics-2016 lecture notes.

1 Syntax

We briefly review the syntax of first-order logic.

1.1 Language

A (first-order) *language* is a collection of symbols. Each symbol is either a *function symbol* or a *predicate symbol*. We associate to each symbol a natural number, called *arity* of the symbol. A function/predicate symbol with arity n is called an *n-ary* function/predicate symbol. A nullary[1] function symbol is often called a *constant* symbol.

We often denote a language by $\mathcal{L}$.

1.2 Examples of languages

We enumerate several examples of languages.

1.2.1 Rings

Ring theory is an abstraction of the properties of integers and other similar structures with addition and multiplication. The language of ring theory is given as follows.

function symbols $\{\mathtt{0}, \mathtt{1}, -, +, \cdot\}$.

predicate symbols None.

arity $\mathtt{0}$ and $\mathtt{1}$ are nullary, $-$ unary, and $+$ and $\cdot$ binary.

[1] Adjectives 0-ary, 1-ary, and 2-ary are abbreviated as nullary, unary, and binary, respectively.

1.2.2 Ordered rings

Often a ring has an ordering over it. The ring of integers is one such[2] . We add a symbol to the language of ring theory in order to express the statements about ordering. The language of the theory of ordered rings is thus as follows.

function symbols Same as the language of ring theory.

predicate symbols $\{<\}$.

arity $<$ is binary.

1.2.3 Sets

Zermelo-Fraenkel set theory (ZF) is the standard axiomatization of the notion of sets. The language of ZF has only one symbol.

function symbols None.

predicate symbols $\{\in\}$.

arity $\in$ is binary.

1.3 Logical symbols

In addition to the symbols in a language, we use *logical* symbols to write logical formulas. We use the following four logical symbols in our system.

$$\neg, \wedge, \exists, = .$$

Logical symbols are used with every language in first-order logic. On the other hand, symbols in a language are not necessarily shared with another language, and therefore called *nonlogical* symbols. We assume that the set of nonlogical symbols is disjoint to that of logical symbols.

1.4 Terms and formulas

In this and the following subsections, we assume the knowledge of abstract syntax trees, namely how they are defined using Backus-Naur forms(BNFs) and the notion of subtrees.

Let $\mathcal{L}$ be a language and V be a countable set. We assume that V is disjoint to the set of logical or nonlogical symbols. We call the elements of V *variables*.

A *term* of $\mathcal{L}$ is a tree constructed from function symbols and variables. The definition is given by the following BNF. Note that we treat parentheses as parts of trees, not symbols.

$$\begin{aligned} \textbf{term} \ni t ::= {} & x \qquad \text{(for } x \text{ a variable)} \\ & \mid \mathtt{f}(t_1, \ldots, t_n) \qquad \text{(for } \mathtt{f} \text{ an } n\text{-ary function symbol).} \end{aligned}$$

[2] Think of a statement like $1 < 2$.

A *formula* of $\mathcal{L}$ is a tree constructed from logical symbols, predicate symbols, variables, and terms. The definition is as follows.

$$
\begin{aligned}
\textbf{formula} \ni \phi, \psi ::= \ & \mathtt{p}(t_1, \ldots, t_n) \qquad \text{(for } \mathtt{p} \text{ an n-ary predicate symbol)} \\
& \mid t_1 = t_2 \\
& \mid \neg(\phi) \\
& \mid (\phi) \wedge (\psi) \\
& \mid \exists x(\phi) \qquad \text{(for } x \text{ a variable).}
\end{aligned}
$$

In writing terms and formulas, we will often drop parentheses as long as it does not introduce ambiguity.

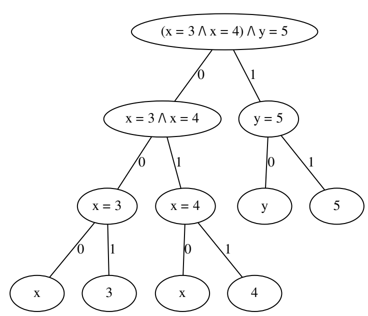
1.5 Occurrence

Given a tree, we distinguish a subtree and its occurrence. More precisely, an *occurrence* of a subtree is a list of natural numbers. The list indicates a path to the subtree from the root. If a number n appears as the k-th element of the list, it corresponds to the choice of n-th branch at a node of depth k.

For example, in the formula

$$(x = 3 \wedge x = 4) \wedge y = 5,$$

the only occurrence of "$x = 3$" is $\langle 0, 0 \rangle$. There are two occurrences of "x": $\langle 0, 0, 0 \rangle$ and $\langle 0, 1, 0 \rangle$.



1.6 Binding

We say that an occurrence of a variable x is *bound* if it is in an occurrence of a subtree of the form $\exists x \phi$[3]. Otherwise, the occurrence is *free*. A variable which has a free/bound occurrence is called a *free/bound variable*, respectively.

Let α be a term or a formula. We denote by $\mathrm{FV}(\alpha)$ the set of all free variables in α. If $\mathrm{FV}(\alpha)$ is empty, α is said to be *closed*.

[3] That is, the occurrence of $\exists x \phi$ is an initial segment of the occurrence of x.

1.7 Substitution

Let ϕ be a formula and t a term. We obtain a new formula $\phi[x := t]$ by substituting t for all free occurrences of x in ϕ. In order to avoid "capturing" free variables, we introduce the following convention: when we write $\phi[x := t]$, we always assume that for any free variable y in t, ϕ does not contain a subformula of the form $\exists y \psi$ with x free in ψ. Violating this convention leads to a wrong substitution. For example, consider the formula $(\exists y, x < y)[x := y]$ as a statement about real numbers. $\exists y, x < y$ is intuitively true for any choice of x, but the result of the substitution $\exists y, y < y$ is just false. We see later that we can safely rename bound variables to address this sort of difficulties.

Remark. We have taken a detour from tradition in that terms, formulas and occurrences are defined in terms of trees. See Shoenfield [2] or Kunen [1] for the traditional approach where terms and formulas are defined as sequences of symbols.

1.8 Other connectives

Informal formulas in daily mathematics contain following symbols in addition to our logical symbols.

$$\Rightarrow, \vee, \forall.$$

We treat an informal formula with these symbols as an abbreviation of our formal formula.

$$\begin{aligned}
(\phi) \vee (\psi) &:= \neg((\neg(\phi)) \wedge (\neg(\psi))), \\
(\phi) \Rightarrow (\psi) &:= (\neg(\phi)) \vee (\psi), \\
\forall x(\phi) &:= \neg(\exists x(\neg(\phi))).
\end{aligned}$$

2 Semantics

The standard (Tarskian) semantics of first-order logic is defined using set theory. First we give a meaning to each symbol in a language, and then extend it to entire terms and formulas.

2.1 Structure

We take the meaning of function/predicate symbols simply to be set-theoretic functions/predicates.

Let $\mathcal{L}$ be a language. A *structure* $\mathfrak{A}$ for $\mathcal{L}$ consists of the following data.

- A nonempty underlying set[4] A.
- For each n-ary function symbol $\mathtt{f}$ of $\mathcal{L}$, a function $\mathtt{f}_{\mathfrak{A}} : A^n \to A$.
- For each n-ary predicate symbol $\mathtt{p}$, a subset $\mathtt{p}_{\mathfrak{A}}$ of A^n.

[4] Also called *universe* or *domain of discourse* in the literature.

2.2 Assignment

In order to interpret terms and formulas, we have to deal with free variables. Intuitively, free variables mean placeholders that are to be replaced by values in a structure. The following definition makes it precise.

Let $\mathcal{L}$ be a language, $\mathfrak{A}$ be a structure for $\mathcal{L}$ with the underlying set A, and V be the set of variables. An *assignment* is a function from a subset of V to A. If α is a term or a formula of $\mathcal{L}$, an *assignment for* α is an assignment whose domain contains $\mathrm{FV}(\alpha)$.

2.3 Interpretation of terms and formulas

The meaning of terms and formulas in a language is recursively determined once we fix a structure and an assignment of values to free variables.

Let $\mathcal{L}$ be a language and $\mathfrak{A}$ be a structure for $\mathcal{L}$ with the underlying set A. For a term t and an assignment σ for t, the interpretation $[\![t]\!]_{(\mathfrak{A},\sigma)}$ is an element of A defined as follows.

- If t is a variable x, then
$$[\![t]\!]_{(\mathfrak{A},\sigma)} := \sigma(x).$$
- If t is an $\mathtt{f}(u_1,\ldots,u_n)$ for a function symbol $\mathtt{f}$ and terms $u_1,\ldots,u_n$, then
$$[\![t]\!]_{(\mathfrak{A},\sigma)} := \mathtt{f}_{\mathfrak{A}}([\![u_1]\!]_{(\mathfrak{A},\sigma)},\ldots,[\![u_n]\!]_{(\mathfrak{A},\sigma)}).$$

For a formula ϕ and an assignment σ for ϕ, the interpretation $[\![\phi]\!]_{(\mathfrak{A},\sigma)}$ is an element of the set $\{\bot,\top\}$ of truth values.

- If ϕ is a $\mathtt{p}(t_1,\ldots,t_n)$ for a predicate symbol $\mathtt{p}$ and terms $t_1,\ldots,t_n$, then
$$[\![\phi]\!]_{(\mathfrak{A},\sigma)} := \begin{cases} \top, \text{ if } \langle [\![t_1]\!]_{(\mathfrak{A},\sigma)},\ldots,[\![t_n]\!]_{(\mathfrak{A},\sigma)}\rangle \in \mathtt{p}_{\mathfrak{A}} \\ \bot, \text{ otherwise} \end{cases}.$$
- If ϕ is a $t_1 = t_2$, then
$$[\![\phi]\!]_{(\mathfrak{A},\sigma)} := \begin{cases} \top, \text{ if } [\![t_1]\!]_{(\mathfrak{A},\sigma)} = [\![t_2]\!]_{(\mathfrak{A},\sigma)} \\ \bot, \text{ otherwise} \end{cases}.$$
- If ϕ is a $\neg(\psi)$, then
$$[\![\phi]\!]_{(\mathfrak{A},\sigma)} := \begin{cases} \top, \text{ if } [\![\psi]\!]_{(\mathfrak{A},\sigma)} = \bot \\ \bot, \text{ otherwise} \end{cases}.$$
- If ϕ is a $(\psi_1) \wedge (\psi_2)$, then
$$[\![\phi]\!]_{(\mathfrak{A},\sigma)} := \begin{cases} \top, \text{ if } [\![\psi_1]\!]_{(\mathfrak{A},\sigma)} = [\![\psi_2]\!]_{(\mathfrak{A},\sigma)} = \top \\ \bot, \text{ otherwise} \end{cases}.$$

- If ϕ is a $\exists x(\psi)$, then

$$[\![\phi]\!]_{(\mathfrak{A},\sigma)} := \begin{cases} \top, \text{ if there is an element } a \in A \text{ such that } [\![\psi]\!]_{(\mathfrak{A},(\sigma\restriction(\mathrm{dom}(\sigma)\smallsetminus\{x\})\cup\{x\mapsto a\})} = \top \\ \bot, \text{ otherwise} \end{cases}.$$

The last case may need an explanation. We are interpreting the formula $\exists x(\psi)$ as "we can find an example a (out of our underlying set A) for the variable x, such that with a given assignment σ together with the additional assignment of a to x, ψ yields to be true." ($\sigma \restriction (\mathrm{dom}(\sigma) \smallsetminus \{x\}) \cup \{x \mapsto a\}$ is the new assignment with a mapping from x to a added.

Exercises

1. Prove that for any closed formula, its interpretation does not depend on the choice of assignment.
2. The definition of $[\![\exists x(\psi)]\!]_{(\mathfrak{A},\sigma)}$ is a bit complicated. What if we change it to

   $$[\![\exists x(\psi)]\!]_{(\mathfrak{A},\sigma)} := \begin{cases} \top, \text{ if there is an element } a \in A \text{ such that } [\![\psi]\!]_{(\mathfrak{A},(\sigma\cup\{x\mapsto a\})} = \top \\ \bot, \text{ otherwise} \end{cases},$$

   or further

   $$[\![\exists x(\psi)]\!]_{(\mathfrak{A},\sigma)} := \begin{cases} \top, \text{ if there is an assignment } \sigma' \text{ for } \psi \text{ such that } \sigma \subset \sigma' \text{ and } [\![\psi]\!]_{(\mathfrak{A},\sigma')} = \top \\ \bot, \text{ otherwise} \end{cases}.$$

   (a) Explain the the difference from the original definition.

   (b) If there is any problem with the new definitions, explain it.
3. Prove the following equalities.

   (a)

   $$[\![(\phi) \Rightarrow (\psi)]\!]_{(\mathfrak{A},\sigma)} = \begin{cases} \top, \text{ if } [\![\phi]\!]_{(\mathfrak{A},\sigma)} = \bot \text{ or } [\![\psi]\!]_{(\mathfrak{A},\sigma)} = \top \\ \bot, \text{ otherwise} \end{cases}.$$

   (b)

   $$[\![(\phi) \vee (\psi)]\!]_{(\mathfrak{A},\sigma)} = \begin{cases} \top, \text{ if } [\![\phi]\!]_{(\mathfrak{A},\sigma)} = \top \text{ or } [\![\psi]\!]_{(\mathfrak{A},\sigma)} = \top \\ \bot, \text{ otherwise} \end{cases}.$$

   (c)

   $$[\![\forall x(\phi)]\!]_{(\mathfrak{A},\sigma)} = \begin{cases} \top, \text{ if } [\![\phi]\!]_{(\mathfrak{A},(\sigma\restriction(\mathrm{dom}(\sigma)\smallsetminus\{x\})\cup\{x\mapsto a\})} = \top \text{ holds for every element } a \in A \\ \bot, \text{ otherwise} \end{cases}.$$
4. Let us identify $\bot$ and $\top$ with natural numbers 0 and 1, respectively. Then prove the following equalities.

(a)

$$[\![(\phi) \Rightarrow (\psi)]\!]_{(\mathfrak{A},\sigma)} = \begin{cases} 1, \text{ if } [\![\phi]\!]_{(\mathfrak{A},\sigma)} \leq [\![\psi]\!]_{(\mathfrak{A},\sigma)} \\ 0, \text{ otherwise} \end{cases}.$$

(b)

$$[\![(\phi) \wedge (\psi)]\!]_{(\mathfrak{A},\sigma)} = \min\{[\![\phi]\!]_{(\mathfrak{A},\sigma)}, [\![\psi]\!]_{(\mathfrak{A},\sigma)}\}.$$

(c)

$$[\![(\phi) \vee (\psi)]\!]_{(\mathfrak{A},\sigma)} = \max\{[\![\phi]\!]_{(\mathfrak{A},\sigma)}, [\![\psi]\!]_{(\mathfrak{A},\sigma)}\}.$$

(d)

$$[\![\forall x(\phi)]\!]_{(\mathfrak{A},\sigma)} = \min_{a \in A}\{[\![\phi]\!]_{(\mathfrak{A},(\sigma \restriction (\mathrm{dom}(\sigma) \setminus \{x\}) \cup \{x \mapsto a\})}\}.$$

(e)

$$[\![\exists x(\phi)]\!]_{(\mathfrak{A},\sigma)} = \max_{a \in A}\{[\![\phi]\!]_{(\mathfrak{A},(\sigma \restriction (\mathrm{dom}(\sigma) \setminus \{x\}) \cup \{x \mapsto a\})}\}.$$

2.4 Satisfaction and model

Several semantical notions are introduced in this and the next subsection.

Let $\mathcal{L}$ be a language, $\mathfrak{A}$ be a structure for $\mathcal{L}$, and ϕ be a formula of $\mathcal{L}$. If $[\![\phi]\!]_{(\mathfrak{A},\sigma)} = \top$ for any assignment σ for ϕ, we say that

- $\mathfrak{A}$ *models* ϕ, or
- $\mathfrak{A}$ *satisfies* ϕ, or
- ϕ is *valid in* $\mathfrak{A}$,
- (maybe more are found in the literature)

and denote it by $\mathfrak{A} \vDash \phi$.

Similarly for a set Γ of formulas, we say that $\mathfrak{A}$ is a *model* of Γ if for every formula $\psi \in \Gamma$, $\mathfrak{A} \vDash \psi$ holds; and we denote it by $\mathfrak{A} \vDash \Gamma$. We say that Γ is *satisfiable* if there is a model of Γ.

Exercises

1. Prove that $\mathfrak{A} \vDash \phi$ iff $\mathfrak{A} \vDash \forall x(\phi)$ for any $\mathfrak{A}$, ϕ, and x.

2.5 Semantic consequence

Let $\mathcal{L}$ be a language, $\mathfrak{A}$ be a structure for $\mathcal{L}$. For a formula ϕ and a set Γ of formulas, we say that ϕ is a *semantic consequence* or a *logical consequence* of Γ if for every structure $\mathfrak{A}$ such that $\mathfrak{A} \vDash \Gamma$, $\mathfrak{A} \vDash \phi$ holds. We denote this relation by $\Gamma \vDash \phi$.

Remark. Be careful that we have overloaded the notation $\vDash$ for quite a different uses. $\mathfrak{A} \vDash \phi$ states the relation between a structure and a formula. On the other hand, $\Gamma \vDash \phi$ states the relation between a set of formulas and a formula.

References

[1] Kunen, Kenneth. The Foundations of Mathematics. College Publications, 2009.

[2] Shoenfield, Joseph R. Mathematical Logic. Addison-Wesley, 1967.