Mathematical Logic and Computability

Lecture 2: Syntax of First-Order Logic

Yunpyo An

Ulsan National Institute of Science and Technology

Mathematical Logic and Computability
Sep 17, 2023

Alphabets

Terms and Formulas

Induction in the Calculi of Terms and Formulas

Free Variables

Alphabets

Definition

An *alphabet* $\mathbb{A}$ is a finite set of **symbols**.

The finite sequences of symbols from an alphabet $\mathbb{A}$ are called *strings* or *words*. The $\mathbb{A}^\star$ denotes the set of all strings over $\mathbb{A}$.

Alphabets

Example

Suppose that $\mathbb{A} = \{a, b, c\}$ is an alphabet.

Then, $\mathbb{A}^\star$ is the set of all strings over $\mathbb{A}$, i.e.,

$$\begin{aligned}\mathbb{A}^\star = \{\Box, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc,\\ aaa, aab, aac, aba, abb, abc, aca, acb, acc, \ldots\}\end{aligned}$$

where $\Box$ is the *empty string*.

Alphabets

Lemma

For a nonempty set M the followings are equivalent:

1. *M is at most countable. (i.e., M is finite or countably infinite.)*
2. *There is an injective map* $\beta : M \to \mathbb{N}$.
3. *There is a surjective map* $\alpha : \mathbb{N} \to M$.

Alphabets of First-Order Logic

Definition

The alphabet of a first-order language concists of the following symbols:

1. Variables: $v_0, v_1, v_2, \ldots$
2. $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ (logical connectives)
3. $\forall, \exists$ (quantifiers)
4. $\equiv$ (equality)
5. (,) (parentheses)
6. - For every $n \geq 1$, a (possible empty) set of n-ary function symbols
   - For every $n \geq 1$, a (possible empty) set of n-ary relation symbols
   - a (possible empty) set of constant symbols

Let $\mathbb{A}$ be the set of symbols 1 to 5. Let S be the set of symbols 6.
The set S determines the first-order language, for convenience, we denote $\mathbb{A}_S$ as the alphabet of the first-order language.

Alphabets of First Order Logic

For example, the alphabet of the first-order language of group theory is defined by symbol set $S_{\mathrm{gr}} := \{\cdot, e\}$,
and the equivalence relations is defined by symbol set $S_{\mathrm{eq}} = \{R\}$.

Question: What is the alphabet of the first-order language of group theory with equivalence relations?

Question: Assume that S is a countable set. Is $\mathbb{A}_S$ countable?

Terms and Formulas

Ok, we define the alphabets of first-order logic.
We already know that not all strings over an alphabet are interpretable.
But, we don't define the meaning (semantics) of first-order logic yet.
Also, we already know that not all strings over an alphabet are meaningful.

Now, we define the *syntax* of first-order logic. It is the grammar of first-order logic.

Terms

Definition

S-terms (T^S) are precisely those strings in $\mathbb{A}_S^{\star}$ that can be obtained by applying the following rules:

1. Every variable is an S-term.
2. Every constant symbol is an S-term.
3. If the strings $t_1, \cdots, t_n$ are S-terms and f is an n-ary function symbol, then $f(t_1, \cdots, t_n)$ is an S-term.

Terms

If f is a unary and g a binary function symbol and the symbol set $S = \{f, g, cR\}$, then

$$gv_0fgv_4c$$

Question: Is it S-term?

Answer: Yes, it is S-term. We can derive it by the rules of S-terms.

Terms

Definition

S-formulas (L^S) are precisely those strings in $\mathbb{A}_S^\star$ that can be obtained by applying the following rules:

1. If t_1 and t_2 are S-terms, then $t_1 \equiv t_2$ is an S-formula.
2. If $t_1, \cdots t_n$ are S-terms and R is an n-ary relation symbol, then $R(t_1, \cdots, t_n)$ is an S-formula.
3. If ϕ is an S-formula, then $\neg\phi$ is an S-formula.
4. If ϕ and ρ are S-formulas, then $(\phi \wedge \rho)$, $(\phi \vee \rho)$, $(\phi \rightarrow \rho)$, and $(\phi \leftrightarrow \rho)$ are S-formulas.
5. If ϕ is an S-formula and x is variable, then $\forall x\phi$ and $\exists x\phi$ are S-formulas.

Terms and Formulas

We might wonder why we need to define the syntax of first-order logic.

The syntax of first-order logic is construction rule of strings in first-order logic.
The terms and formulas are the different types of strings in first-order logic.

We borrow the idea from next lecture 'semantics' of first-order logic, the terms are the 'objects' and the formulas are the 'properties' of the objects.
Note that, we don't define the meaning of the terms and formulas yet. Keep in mind it.

Induction in the Calculi of Terms and Formulas

Let S be the set of symbols and $Z \subseteq \mathbb{A}_S^\star$ be the set of strings over the alphabet $\mathbb{A}_S$.

We want to construct that from symbol set S to the terms T^S 5 and formulas L^S 6.
We have the rules to construct the terms and formulas!

Induction in the Calculi of Terms and Formulas

We can construct the terms and formulas by inductions. Let's the induction begin!

Assume that $\zeta_1, \cdots, \zeta_n$ all belong to Z. Then also ζ belongs to Z writing as following

$$\frac{\zeta_1, \cdots, \zeta_n}{\zeta} \tag{1}$$

We call this rule as *inference rule*.

It allows $n = 0$, the first sort of rules in 5 and 6 is "premise-free" rules.

Induction in the Calculi of Terms and Formulas

The calculus **C** (rule) of terms T^S as follows:

$$\frac{\square}{x} \tag{2}$$

$$\frac{\square}{c}, c \in S \tag{3}$$

$$\frac{t_1, \cdots, t_n}{f(t_1, \cdots, t_n)}, f \in S, f \text{ is } n\text{-ary function symbol} \tag{4}$$

We can prove that some strings are terms by using the inference rules it is called *induction over calculus* **C**.

Induction in the Calculi of Terms and Formulas

The calculus **C** (rule) of formulas L^S as follows:

$$\frac{t_1, t_2}{t_1 \equiv t_2} \tag{5}$$

$$\frac{t_1, \cdots, t_n}{R(t_1, \cdots, t_n)}, R \in S, \quad R \text{ is } n\text{-ary relation symbol} \tag{6}$$

$$\frac{\phi}{\neg\phi}, \tag{7}$$

$$\frac{\phi, \rho}{(\phi \star \rho)}, \quad \text{where} \star = \vee, \wedge, \rightarrow, \leftrightarrow \tag{8}$$

$$\frac{\phi}{\forall x \phi}, \tag{9}$$

$$\frac{\phi}{\exists x \phi}, \tag{10}$$

Induction in the Calculi of Terms and Formulas

Definition

The function var, which associates with each S-terms the set of variables ocurring in it:

$$\begin{aligned} \mathrm{var}(x) &:= \{x\} \\ \mathrm{var}(c) &:= \emptyset \\ \mathrm{var}(f(t_1, \cdots, t_n)) &:= \mathrm{var} t_1 \cup \cdots \cup \mathrm{var} t_n \end{aligned}$$

Induction in the Calculi of Terms and Formulas

Definition

The function SF, which assigns to each formula the set of its subformulas as following:

$$
\begin{aligned}
\mathrm{SF}(t_1 \equiv t_2) &:= \{t_1 \equiv t_2\} \\
\mathrm{SF}(R(t_1, \cdots, t_n)) &:= \{R(t_1, \cdots, t_n)\} \\
\mathrm{SF}(\neg\phi) &:= \{\neg\phi\} \cup \mathrm{SF}(\phi) \\
\mathrm{SF}((\phi \star \rho)) &:= \{(\phi \star \rho)\} \cup \mathrm{SF}(\phi) \cup \mathrm{SF}(\rho) \qquad \text{where} \star = \vee, \wedge, \rightarrow, \leftrightarrow \\
\mathrm{SF}(\forall x\phi) &:= \{\forall x\phi\} \cup \mathrm{SF}(\phi) \\
\mathrm{SF}(\exists x\phi) &:= \{\exists x\phi\} \cup \mathrm{SF}(\phi)
\end{aligned}
$$

Free variables

Consider the following formula:

$$\phi := \exists x(Ryz \land \forall y(\neg y \equiv x \lor Ryz))$$

Question: Can we subtitute y to other alphabet (such as a, b not y and z)?
Question: Is it good sentence?
We may consider about y is in the scope of $\forall y$. Also, we can consider the scope of x. But we need to define (or assign) yz when interprete it. Now, let's define free variables.

Free variables

Definition

The function free, which assigns to each formula the set of its free variables as following:

$$
\begin{aligned}
\mathrm{free}(t_1 \equiv t_2) &:= \mathrm{var}(t_1) \cup \mathrm{var}(t_2) \\
\mathrm{free}(R(t_1, \cdots, t_n)) &:= \mathrm{var}(t_1) \cup \cdots \cup \mathrm{var}(t_n) \\
\mathrm{free}(\neg\phi) &:= \mathrm{free}(\phi) \\
\mathrm{free}((\phi \star \rho)) &:= \mathrm{free}(\phi) \cup \mathrm{free}(\rho) && \text{where} \star = \vee, \wedge, \rightarrow, \leftrightarrow \\
\mathrm{free}(\forall x\phi) &:= \mathrm{free}(\phi) \setminus \{x\} \\
\mathrm{free}(\exists x\phi) &:= \mathrm{free}(\phi) \setminus \{x\}
\end{aligned}
$$

When the formula without free variables, we call it *closed formula* or *sentence*. Furthermore, we denote L_n^S the set of S-formulas in which the variables occuring free among the n variables.

Summary

For summary of this lecture, we have the following definitions:

- **Terms** T^S 5
- **Formulas** L^S 6
- **Induction over calculus C** 2 and 5
- **Subformulas** 8
- **Free variables** 9