Lesson 7: First-order logic, part II

Theme: The semantics of first-order logic.

1 Valuations

Recall that VAR is a set of variables. Let A be a structure.

- A \textit{valuation} in a structure A is a function \( \text{val} : \text{VAR} \to A \).

- For \( \bar{a} = (a_1, \ldots, a_n) \), where each \( a_i \in A \), and \( \bar{x} = (x_1, \ldots, x_n) \), where \( x_1, \ldots, x_n \) are all different variables, we write \( \text{val}[\bar{x} \mapsto \bar{a}] \) to denote the valuation \( \text{val}' \), where for every \( y \in \text{VAR} \),

\[
\text{val}'(y) = \begin{cases} 
\text{val}(y), & \text{if } y \notin \{x_1, \ldots, x_n\} \\
\bar{a}_i, & \text{if } y = x_i 
\end{cases}
\]

Sometimes we write \( [\bar{x} \mapsto \bar{a}] \) to denote a valuation \( \text{val} \) such that \( \text{val}(x_i) = a_i \).

2 Interpretations/models

An \textit{interpretation} is a pair \( (A, \text{val}) \), where A is a structure and \( \text{val} \) is a valuation. Quite often, interpretations are also called \textit{models}.

In an interpretation \( (A, \text{val}) \), each term \( t \) is associated with an element \( t^A[\text{val}] \) defined inductively as follows.

- \( x^A[\text{val}] = \text{val}(x) \), where \( x \in \text{VAR} \).
- \( c^A[\text{val}] = c^A \), where \( c \) is a constant symbol.
- \( f(t_1, \ldots, t_n)^A[\text{val}] = f^A(t_1^A[\text{val}], \ldots, t_n^A[\text{val}]) \).

\( t^A[\text{val}] \) reads the term \( t \) in structure \( A \) according to valuation \( \text{val} \).

As usual, when the structure \( A \) is clear from the context, we will simply write \( t[\text{val}] \), instead of \( t^A[\text{val}] \).

Given an FO formula \( \varphi \), and an interpretation \( (A, \text{val}) \), we define \( (A, \text{val}) \models \varphi \) (read: \( (A, \text{val}) \) is an interpretation/a model of \( \varphi \), or that \( \varphi \) holds in \( (A, \text{val}) \)) inductively as follows.

- \( (A, \text{val}) \models s \equiv t \), if and only if \( s^A[\text{val}] = t^A[\text{val}] \).
- \( (A, \text{val}) \models R(t_1, \ldots, t_n) \), if and only if \( (t_1^A[\text{val}], \ldots, t_n^A[\text{val}]) \in R^A \).
- \( (A, \text{val}) \models \neg \alpha \), if and only if it is not true that \( (A, \text{val}) \models \alpha \).
- \( (A, \text{val}) \models \alpha \land \beta \), if and only if \( (A, \text{val}) \models \alpha \) and \( (A, \text{val}) \models \beta \).
- \( (A, \text{val}) \models \alpha \lor \beta \), if and only if \( (A, \text{val}) \models \alpha \) or \( (A, \text{val}) \models \beta \).
- \( (A, \text{val}) \models \exists x \alpha \), if and only if there is \( a \in A \) such that \( (A, \text{val}[x \mapsto a]) \models \alpha \).
- \( (A, \text{val}) \models \forall x \alpha \), if and only if for every \( a \in A \), \( (A, \text{val}[x \mapsto a]) \models \alpha \).

We write \( (A, \text{val}) \not\models \varphi \), when it is not true that \( (A, \text{val}) \models \varphi \).

Note that whether \( (A, \text{val}) \models \varphi(x_1, \ldots, x_n) \) depends only on \( A \) (obviously!) and the images of \( x_1, \ldots, x_n \) under \( \text{val} \). In other words, the value \( \text{val}(y) \) does not matter for every \( y \notin \{x_1, \ldots, x_n\} \).

To avoid clutter, we write \( (A, a_1, \ldots, a_n) \models \varphi(x_1, \ldots, x_n) \), to mean that \( (A, \text{val}) \models \varphi \), where \( \text{val} \) is a valuation function that maps each \( x_i \) to \( a_i \). In particular, if \( \alpha \) is a sentence, the valuation \( \text{val} \) is dispensable in the determination of \( (A, \text{val}) \models \alpha \). So, for a sentence \( \alpha \), we simply write \( A \models \alpha \).

A formula \( \varphi \) is \textit{satisfiable}, if \( \varphi \) has an interpretation/model.
3 Some examples

Example 7.1 Let $\mathcal{A} = (A, \text{plus}^A, 0^A)$ be the structure with signature $\{\text{plus}, 0\}$ defined as follows.

- $A = \{0, 1, 2, \ldots, 8\}$,
- plus is a binary function/operator, where $\text{plus}^A(x, y) = x + y \mod 9$,
- $0^A = 0$.

Here are some formulas that hold/not hold in $\mathcal{A}$.

- $\mathcal{A}, (x, y, z) \mapsto (3, 5, 8) \models \text{plus}(x, y) \approx z$. Can I say that $\mathcal{A} \models \text{plus}(3, 5) \approx 8$?
- $\mathcal{A}, (x, y) \mapsto (1, 2) \not\models \text{plus}(x, y) \approx 0$.
  This is equivalent to say that $\mathcal{A}, (x, y) \mapsto (1, 2) \models \neg(\text{plus}(x, y) \approx 0)$, or, $\mathcal{A}, (x, y) \mapsto (1, 2) \models \neg\text{plus}(x, y) \neq 0$.
- $\mathcal{A}, z \mapsto 0 \models \forall x \text{plus}(x, z) \approx x$.
- $\mathcal{A}, z \mapsto 1 \models \forall x \text{plus}(x, z) \neq x$. Can I say that $\mathcal{A} \models \forall x \text{plus}(x, 1) \neq x$?
- $\mathcal{A} \models \forall x \text{plus}(x, 0) \approx x$.
- $\mathcal{A} \models \forall x \exists y \text{plus}(x, y) \approx 0$.
- $\mathcal{A} \models \forall x (x \neq 0 \rightarrow (\exists y x \neq y \land \text{plus}(x, y) \approx 0))$.

Example 7.2 Let $\mathcal{B} = (B, E^B)$ be the following structure, where $\text{ar}(E) = 2$:

- $B = \{a_1, b_1, \ldots, a_n, b_n\}$,
- $E^B = \{(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)\}$.

The relation $E^B$ can be illustrated as follows.

```
  a1  b1
  |   |
  a2  b2
  |   |
   ..
  |   |
  an  bn
```

Here are some examples of formulas that hold/not hold in $\mathcal{B}$.

- $\mathcal{B}, (x, y) \mapsto (a_1, b_1) \models E(x, y)$. Can I say $\mathcal{B} \models E(a_1, b_1)$?
- $\mathcal{B}, (x, y) \mapsto (a_1, b_3) \not\models E(x, y)$.
- $\mathcal{B} \models \exists x \exists y E(x, y)$.
- $\mathcal{B} \not\models \exists x E(x, x)$, which can be rewritten as $\mathcal{B} \models \neg\exists x E(x, x)$
- $\mathcal{B} \models \forall x \exists y (E(x, y) \land \forall z (E(x, z) \rightarrow y \approx z))$. 

2
Example 7.3 Let \( Z = (\mathbb{Z}, \text{succ}^Z, \text{plus}^Z, 0^Z) \) be the structure with signature \( \{ \text{plus}, \text{succ}, 0 \} \) defined as follows.

- \( \mathbb{Z} = \{ \ldots, -2, -1, 0, 1, 2, \ldots \} \),
- \( \text{succ}^Z \) is a binary relation, where \((x, y) \in \text{succ}^Z \) if and only if \( y = x + 1 \),
- \( \text{plus}^Z \) is a binary operator, where \( \text{plus}^Z (x, y) = x + y \),
- \( 0^Z = 0 \).

Here are some formulas that hold/not hold in \( Z \).

- \( Z, (x, y, z) \mapsto (3, 5, 8) \models \text{plus}(x, y) \approx z \).
- \( Z, (x, y) \mapsto (1, 2) \not\models \text{plus}(x, y) \approx 0 \).
- \( Z, z \mapsto 0 \models \forall x \text{ plus}(x, z) \approx x \).
- \( Z, z \mapsto 1 \models \forall x \text{ plus}(x, z) \not\approx x \).
- \( Z \models \forall x \text{ plus}(x, 0) \approx x \).
- \( Z \models \forall x \exists y \text{ plus}(x, y) \approx 0 \).
- \( Z \models \forall x \exists y \text{ succ}(x, y) \land x \not\approx y \).
- \( Z \models \forall x \exists y \text{ succ}(x, y) \land \left( \forall z (\text{succ}(x, z) \rightarrow y \approx z) \right) \).
- \( Z \models \forall x \forall y \forall z \forall w \left( (\text{succ}(x, z) \land \text{succ}(w, y)) \rightarrow \text{plus}(x, y) \approx \text{plus}(z, w) \right) \).

4 Two little theorems

Theorem 7.4 Let \( h : A \rightarrow B \) be an isomorphism. Then, for every formula \( \varphi(\bar{x}) \),

\[(A, \bar{a}) \models \varphi(\bar{x}) \quad \text{if and only if} \quad (B, h(\bar{a})) \models \varphi(\bar{x}) \]

(Recall that \( \bar{x} \) and \( \bar{a} \) stands for a vector of variables and elements, respectively, which we tacitly assume to be of the same length.)

A \( \forall \)-sentence (read: a universal sentence) is a sentence of the form:

\[
\forall x_1 \cdots \forall x_n \varphi, \quad (1)
\]

where \( \varphi \) is quantifier free. Likewise, an \( \exists \)-sentence (read: an existential sentence) is a sentence of the form:

\[
\exists x_1 \cdots \exists x_n \varphi, \quad (2)
\]

where \( \varphi \) is quantifier free. As usual, we will simply write \( \forall \bar{x} \varphi \) or \( \exists \bar{x} \varphi \), instead of Eq. (1) and (2), respectively.

Theorem 7.5 Let \( A \subseteq B \).

- For every \( \forall \)-sentence \( \psi \), if \( B \models \psi \), then \( A \models \psi \).
- For every \( \exists \)-sentence \( \psi \), if \( A \models \psi \), then \( B \models \psi \).
Exercise set 1

In the following $E, R, T, S$ are relational symbols, $f, g$ are function symbols and $c, c_1, c_2, \ldots$ are constant symbols.

(1) Determine the quantifier rank of each of the following formulas.

\[
\begin{align*}
\beta_1 & := \forall x \exists y (z \neq y \land R(x, y)) \\
\beta_2 & := \forall x (x \neq y \land \exists y R(x, y)) \\
\beta_3 & := \left( \forall z (\exists z z \neq y) \right) \land f(z) \approx z \\
\beta_4 & := \forall z \left( z \approx y \land \exists z (f(z) \approx g(z)) \right) \\
\beta_5 & := \exists y \forall x (R(z, g(z, y)) \land T(y) \rightarrow \exists z \forall y x \approx f(x, g(y, z))) \\
\beta_6 & := x \neq f(c, z) \land \forall z \forall x \left( R(x, c, c, y) \land f(x, z) \approx c \land \exists y (f(x, y) \land g(z, y)) \right)
\end{align*}
\]

(2) Determine the free variables of each of the formulas above.

(3) Determine the result of each of the following substitutions.

- $z/f(z, z, x)$ in $\beta_1$.
- $y/g(c, c)$ in $\beta_2$.
- $z/f(x, y, z)$ in $\beta_3$.
- $y/z$ in $\beta_4$.
- $z/f(c, z, x)$ in $\beta_5$.
- $(x, y, z)/(x, x, x)$ in $\beta_6$.

Which substitutions are collision-free?

Exercise set 2: The notion of congruence

In this exercise, we will study the notion of congruence on structures. Let $Z$ be set, and $\sim$ be an equivalence relation on $Z$. For a positive integer $n$, define a binary relation $\sim^n$ on $Z^n$ as follows.

\[(a_1, \ldots, a_n) \sim^n (b_1, \ldots, b_n) \text{ if and only if } a_i \sim b_i, \text{ for each } i \in \{1, \ldots, n\}.\]

(4) Prove that $\sim^n$ is an equivalence relation.

The relation $\sim^n$ is called the extension of $\sim$ to $Z^n$.

(5) Prove that $[\bar{a}]_\sim^n = [a_1]_\sim \times [a_2]_\sim \times \cdots \times [a_n]_\sim$, where $\bar{a} = (a_1, \ldots, a_n)$.

When it is clear from the context, we will simply use the same symbol $\sim$, instead of $\sim^n$. That is, we will write $\bar{a} \sim \bar{b}$ to mean the extension of $\sim$ to $Z^n$, instead of $\bar{a} \sim^n \bar{b}$.

Let $\mathcal{A}$ be an $L$-structure. A congruence in $\mathcal{A}$ is an equivalence relation $\sim$ on $A$ such that for every function symbol $f \in L$, the following holds.

\[\text{If } \bar{a} \sim \bar{b}, \text{ then } f(\bar{a}) \sim f(\bar{b}).\]

(6) Let $\sim$ be a congruence in an $L$-structure $\mathcal{A}$. The factor of $\mathcal{A}$ modulo $\sim$ is a structure $\mathcal{B}$ such that

- $B = A/\sim = \{ [a]_\sim \mid a \in A \}$,
Consider a sentence

\[ \exists x \exists y \exists z \varphi \]

Pick \( n \) "new" constant symbols \( c_1, \ldots, c_n \notin L \). Show that \( \varphi[(x_1, \ldots, x_n)/(c_1, \ldots, c_n)] \) and \( \psi \) are equi-satisfiable.

Consider a sentence \( \psi \) over a vocabulary \( L \) of the form:

\[ \forall x_1 \cdots \forall x_n \exists y \varphi \]

Pick a "new" arity \( n \) function symbol \( f \notin L \). Show that \( \forall x_1 \cdots \forall x_n \varphi[y/f(x_1, \ldots, x_n)] \) and \( \psi \) are equi-satisfiable.

Consider a sentence \( \psi \) over a vocabulary \( L \) of the form:

\[ \forall x_1 \cdots \forall x_n \exists y_1 \cdots \exists y_m \varphi, \]

where \( \varphi \) does not start with existential quantifiers. Prove that there is a sentence of the form:

\[ \psi' := \forall x_1 \cdots \forall x_n \varphi' \]

such that \( \varphi' \) does not start with existential quantifiers, and \( \psi \) and \( \psi' \) are equi-satisfiable.

(Skolem normal form) Consider a sentence \( \psi \) over a vocabulary \( L \) of the form:

\[ \psi := Q_1 x_1 \cdots Q_n x_n \varphi, \quad (3) \]

where each \( Q_i \) is a quantifier (either \( \forall \) or \( \exists \)), and \( \varphi \) is quantifier free. Prove that there is \( \forall \)-sentence \( \psi' \) (over different vocabulary \( L' \)) such that \( \psi \) and \( \psi' \) are equi-satisfiable.

Note 1: The \( \forall \)-sentence \( \psi' \) is called the Skolem normal form of \( \psi \).

Note 2: Formulas of the form \( (3) \) are often called formulas in Prenex Normal Form (PNF). We will show later on that every formula can be converted into PNF.