

syn.1 Satisfaction

sol:syn:sat:
sec To define the satisfaction relation $\mathfrak{M}, s \models \varphi$ for second-order **formulas**, we have explanation to extend the definitions to cover second-order **variables**. The notion of a **structure** is the same for second-order logic as it is for first-order logic. There is only a difference for variable assignments s : these now must not just provide values for the first-order **variables**, but also for the second-order **variables**.

Definition syn.1 (Variable Assignment). A *variable assignment* s for a **structure** \mathfrak{M} is a function which maps each

1. object **variable** v_i to an element of $|\mathfrak{M}|$, i.e., $s(v_i) \in |\mathfrak{M}|$
2. n -place relation variable V_i^n to an n -place relation on $|\mathfrak{M}|$, i.e., $s(V_i^n) \subseteq |\mathfrak{M}|^n$;
3. n -place function variable u_i^n to an n -place function from $|\mathfrak{M}|$ to $|\mathfrak{M}|$, i.e., $s(u_i^n): |\mathfrak{M}|^n \rightarrow |\mathfrak{M}|$;

A **structure** assigns a **value** to each **constant symbol** and **function symbol**, explanation and a second-order variable assignment assigns objects and functions to each object and function variable. Together, they let us assign a value to every term.

Definition syn.2 (Value of a Term). If t is a term of the language \mathcal{L} , \mathfrak{M} is a **structure** for \mathcal{L} , and s is a **variable** assignment for \mathfrak{M} , the *value* $\text{Val}_s^{\mathfrak{M}}(t)$ is defined as for first-order terms, plus the following clause:

$$t \equiv u(t_1, \dots, t_n):$$

$$\text{Val}_s^{\mathfrak{M}}(t) = s(u)(\text{Val}_s^{\mathfrak{M}}(t_1), \dots, \text{Val}_s^{\mathfrak{M}}(t_n)).$$

Definition syn.3 (x -Variant). If s is a **variable** assignment for a **structure** \mathfrak{M} , then any **variable** assignment s' for \mathfrak{M} which differs from s at most in what it assigns to x is called an *x -variant* of s . If s' is an x -variant of s we write $s' \sim_x s$. (Similarly for second-order variables X or u .)

Definition syn.4. If s is a **variable** assignment for a **structure** \mathfrak{M} and $m \in |\mathfrak{M}|$, then the assignment $s[m/x]$ is the **variable** assignment defined by

$$s[m/y] = \begin{cases} m & \text{if } y \equiv x \\ s(y) & \text{otherwise,} \end{cases}$$

If X is an n -place relation **variable** and $M \subseteq |\mathfrak{M}|^n$, then $s[M/X]$ is the **variable** assignment defined by

$$s[M/y] = \begin{cases} M & \text{if } y \equiv X \\ s(y) & \text{otherwise.} \end{cases}$$

If u is an n -place function variable and $f: |\mathfrak{M}|^n \rightarrow |\mathfrak{M}|$, then $s[f/u]$ is the variable assignment defined by

$$s[f/y] = \begin{cases} f & \text{if } y \equiv u \\ s(y) & \text{otherwise.} \end{cases}$$

In each case, y may be any first- or second-order variable.

Definition syn.5 (Satisfaction). For second-order formulas φ , the definition of satisfaction is like ?? with the addition of:

1. $\varphi \equiv X^n(t_1, \dots, t_n)$: $\mathfrak{M}, s \models \varphi$ iff $\langle \text{Val}_s^{\mathfrak{M}}(t_1), \dots, \text{Val}_s^{\mathfrak{M}}(t_n) \rangle \in s(X^n)$.
2. $\varphi \equiv \forall X \psi$: $\mathfrak{M}, s \models \varphi$ iff for every $M \subseteq |\mathfrak{M}|^n$, $\mathfrak{M}, s[M/X] \models \psi$.
3. $\varphi \equiv \exists X \psi$: $\mathfrak{M}, s \models \varphi$ iff for at least one $M \subseteq |\mathfrak{M}|^n$ so that $\mathfrak{M}, s[M/X] \models \psi$.
4. $\varphi \equiv \forall u \psi$: $\mathfrak{M}, s \models \varphi$ iff for every $f: |\mathfrak{M}|^n \rightarrow |\mathfrak{M}|$, $\mathfrak{M}, s[f/u] \models \psi$.
5. $\varphi \equiv \exists u \psi$: $\mathfrak{M}, s \models \varphi$ iff for at least one $f: |\mathfrak{M}|^n \rightarrow |\mathfrak{M}|$ so that $\mathfrak{M}, s[f/u] \models \psi$.

Example syn.6. Consider the formula $\forall z (X(z) \leftrightarrow \neg Y(z))$. It contains no second-order quantifiers, but does contain the second-order variables X and Y (here understood to be one-place). The corresponding first-order sentence $\forall z (P(z) \leftrightarrow \neg R(z))$ says that whatever falls under the interpretation of P does not fall under the interpretation of R and vice versa. In a structure, the interpretation of a predicate symbol P is given by the interpretation $P^{\mathfrak{M}}$. But for second-order variables like X and Y , the interpretation is provided, not by the structure itself, but by a variable assignment. Since the second-order formula is not a sentence (it includes free variables X and Y), it is only satisfied relative to a structure \mathfrak{M} together with a variable assignment s .

$\mathfrak{M}, s \models \forall z (X(z) \leftrightarrow \neg Y(z))$ whenever the elements of $s(X)$ are not elements of $s(Y)$, and vice versa, i.e., iff $s(Y) = |\mathfrak{M}| \setminus s(X)$. For instance, take $|\mathfrak{M}| = \{1, 2, 3\}$. Since no predicate symbols, function symbols, or constant symbols are involved, the domain of \mathfrak{M} is all that is relevant. Now for $s_1(X) = \{1, 2\}$ and $s_1(Y) = \{3\}$, we have $\mathfrak{M}, s_1 \models \forall z (X(z) \leftrightarrow \neg Y(z))$.

By contrast, if we have $s_2(X) = \{1, 2\}$ and $s_2(Y) = \{2, 3\}$, $\mathfrak{M}, s_2 \not\models \forall z (X(z) \leftrightarrow \neg Y(z))$. That's because $\mathfrak{M}, s_2[2/z] \models X(z)$ (since $2 \in s_2[2/z](X)$) but $\mathfrak{M}, s_2[2/z] \not\models \neg Y(z)$ (since also $2 \in s_2[2/z](Y)$).

Example syn.7. $\mathfrak{M}, s \models \exists Y (\exists y Y(y) \wedge \forall z (X(z) \leftrightarrow \neg Y(z)))$ if there is an $N \subseteq |\mathfrak{M}|$ such that $\mathfrak{M}, s[N/Y] \models (\exists y Y(y) \wedge \forall z (X(z) \leftrightarrow \neg Y(z)))$. And that is the case for any $N \neq \emptyset$ (so that $\mathfrak{M}, s[N/Y] \models \exists y Y(y)$) and, as in the previous example, $M = |\mathfrak{M}| \setminus s(X)$. In other words, $\mathfrak{M}, s \models \exists Y (\exists y Y(y) \wedge \forall z (X(z) \leftrightarrow \neg Y(z)))$ iff $|\mathfrak{M}| \setminus s(X)$ is non-empty, i.e., $s(X) \neq |\mathfrak{M}|$. So, the formula is satisfied, e.g., if $|\mathfrak{M}| = \{1, 2, 3\}$ and $s(X) = \{1, 2\}$, but not if $s(X) = \{1, 2, 3\} = |\mathfrak{M}|$.

Since the **formula** is not satisfied whenever $s(X) = |\mathfrak{M}|$, the **sentence**

$$\forall X \exists Y (\exists y Y(y) \wedge \forall z (X(z) \leftrightarrow \neg Y(z)))$$

is never satisfied: For any **structure** \mathfrak{M} , the assignment $s(X) = |\mathfrak{M}|$ will make the **sentence** false. On the other hand, the sentence

$$\exists X \exists Y (\exists y Y(y) \wedge \forall z (X(z) \leftrightarrow \neg Y(z)))$$

is satisfied relative to any assignment s , since we can always find $M \subseteq |\mathfrak{M}|$ but $M \neq |\mathfrak{M}|$ (e.g., $M = \emptyset$).

Example syn.8. The second-order **sentence** $\forall X \forall y X(y)$ says that every 1-place relation, i.e., every property, holds of every object. That is clearly never true, since in every \mathfrak{M} , for a variable assignment s with $s(X) = \emptyset$, and $s(y) = a \in |\mathfrak{M}|$ we have $\mathfrak{M}, s \not\models X(y)$. This means that $\varphi \rightarrow \forall X \forall y X(y)$ is equivalent in second-order logic to $\neg\varphi$, that is: $\mathfrak{M} \models \varphi \rightarrow \forall X \forall y X(y)$ iff $\mathfrak{M} \models \neg\varphi$. In other words, in second-order logic we can define \neg using \forall and \rightarrow .

Problem syn.1. Show that in second-order logic \forall and \rightarrow can define the other connectives:

1. Prove that in second-order logic $\varphi \wedge \psi$ is equivalent to $\forall X (\varphi \rightarrow (\psi \rightarrow \forall x X(x)) \rightarrow \forall x X(x))$.
2. Find a second-order formula using only \forall and \rightarrow equivalent to $\varphi \vee \psi$.

Photo Credits

Bibliography