To define the satisfaction relation $\mathcal{M}, s \models \varphi$ for second-order formulas, we have 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 $\mathcal{M}$ is a function which maps each

1. object variable $v_i$ to an element of $|\mathcal{M}|$, i.e., $s(v_i) \in |\mathcal{M}|$;
2. $n$-place relation variable $V^n_i$ to an $n$-place relation on $|\mathcal{M}|$, i.e., $s(V^n_i) \subseteq |\mathcal{M}|^n$;
3. $n$-place function variable $u^n_i$ to an $n$-place function from $|\mathcal{M}|$ to $|\mathcal{M}|$, i.e., $s(u^n_i): |\mathcal{M}|^n \to |\mathcal{M}|$.

A structure assigns a value to each constant symbol and function symbol, and a second-order variable 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 $\mathcal{L}$, $\mathcal{M}$ is a structure for $\mathcal{L}$, and $s$ is a variable assignment for $\mathcal{M}$, the value $\text{Val}_{\mathcal{M}}^s(t)$ is defined as for first-order terms, plus the following clause:

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

$$\text{Val}_{\mathcal{M}}^s(t) = s(u)(\text{Val}_{\mathcal{M}}^s(t_1), \ldots, \text{Val}_{\mathcal{M}}^s(t_n)).$$

**Definition syn.3 ($x$-Variant).** If $s$ is a variable assignment for a structure $\mathcal{M}$, then any variable assignment $s'$ for $\mathcal{M}$ 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 (Satisfaction).** For second-order formulas $\varphi$, the definition of satisfaction is like ?? with the addition of:

1. $\varphi \equiv X^n(t_1, \ldots, t_n): \quad \mathcal{M}, s \models \varphi$ iff $(\text{Val}_{\mathcal{M}}^s(t_1), \ldots, \text{Val}_{\mathcal{M}}^s(t_n)) \in s(X^n)$.
2. $\varphi \equiv \forall X \psi: \quad \mathcal{M}, s \models \varphi$ iff for every $X$-variant $s'$ of $s$, $\mathcal{M}, s' \models \psi$.
3. $\varphi \equiv \exists X \psi: \quad \mathcal{M}, s \models \varphi$ iff there is an $X$-variant $s'$ of $s$ so that $\mathcal{M}, s' \models \psi$.
4. $\varphi \equiv \forall u \psi: \quad \mathcal{M}, s \models \varphi$ iff for every $u$-variant $s'$ of $s$, $\mathcal{M}, s' \models \psi$.
5. $\varphi \equiv \exists u \psi: \quad \mathcal{M}, s \models \varphi$ iff there is an $u$-variant $s'$ of $s$ so that $\mathcal{M}, s' \models \psi$. 

satisfaction rev: 666b46f (2020-02-13) by OLP / CC–BY
Example syn.5. Consider the formula $\forall z (X(z) \leftrightarrow R(z))$. 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 $\forall z (P(z) \leftrightarrow R(z))$ 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 (in includes free variables $X$ and $Y$), it is only satisfied relative to a structure $\mathfrak{M}$ together with a variable assignment $s$.

$\mathfrak{M}, s \models \forall z (X(z) \leftrightarrow -Y(z))$ whenever the elements of $s(X)$ are not elements of $s(Y)$, and vice versa, i.e., iff $s(Y) = [\mathfrak{M}] \setminus s(X)$. So 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 $\mathfrak{M}, s_1 \models \forall z (X(z) \leftrightarrow -Y(z))$.

By contrast, if we have $s_2(X) = \{1, 2\}$ and $s_2(Y) = \{2, 3\}$, $\mathfrak{M}, s_2 \not\models \forall z (X(z) \leftrightarrow -Y(z))$. That’s because there is a $z$-variant $s'_2$ of $s_2$ with $s'_2(z) = 2$ where $\mathfrak{M}, s'_2 \models X(z)$ (since $2 \in s'_2(X)$) but $\mathfrak{M}, s'_2 \not\models -Y(z)$ (since also $s'_2(z) \in s'_2(Y)$).

Example syn.6. $\mathfrak{M}, s \models \exists Y (\exists y Y(y) \land \forall z (X(z) \leftrightarrow -Y(z)))$ if there is an $s' \sim_Y$ $s$ such that $\mathfrak{M}, s' \models (\exists y Y(y) \land \forall z (X(z) \leftrightarrow -Y(z)))$. And that is the case iff $s'(Y) \neq \emptyset$ (so that $\mathfrak{M}, s' \models \exists y Y(y)$) and, as in the previous example, $s'(Y) = [\mathfrak{M}] \setminus s'(X)$. In other words, $\mathfrak{M}, s \models \exists Y (\exists y Y(y) \land \forall z (X(z) \leftrightarrow -Y(z)))$ 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

$$\forall X \exists Y (\exists y Y(y) \land \forall z (X(z) \leftrightarrow -Y(z)))$$

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

$$\exists Y \forall X (\exists y Y(y) \land \forall z (X(z) \leftrightarrow -Y(z)))$$

is satisfied relative to any assignment $s$, since we can always find an $X$-variant $s'$ of $s$ with $s'(X) \neq [\mathfrak{M}]$.

Photo Credits

Bibliography