## syn.1 Satisfaction

sol:syn:sat:sec To define the satisfaction relation $\mathfrak{M}, s \vDash \varphi$ 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 diffence for variable assignments $s$: these now must not just provide values for the first-order variables, but also for the second-order variables. explanation

**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., $s(u_i^n)\colon |\mathfrak{M}|^n \to |\mathfrak{M}|$;

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. explanation

**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* $\mathrm{Val}_s^{\mathfrak{M}}(t)$ is defined as for first-order terms, plus the following clause:

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

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

**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 (Satisfaction).** For second-order formulas $\varphi$, the definition of satisfaction is like **??** with the addition of:

1. $\varphi \equiv X^n(t_1, \dots, t_n)$: $\mathfrak{M}, s \vDash \varphi$ iff $\langle \mathrm{Val}_s^{\mathfrak{M}}(t_1), \dots, \mathrm{Val}_s^{\mathfrak{M}}(t_n)\rangle \in s(X^n)$.
2. $\varphi \equiv \forall X\, \psi$: $\mathfrak{M}, s \vDash \varphi$ iff for every $X$-variant $s'$ of $s$, $\mathfrak{M}, s' \vDash \psi$.
3. $\varphi \equiv \exists X\, \psi$: $\mathfrak{M}, s \vDash \varphi$ iff there is an $X$-variant $s'$ of $s$ so that $\mathfrak{M}, s' \vDash \psi$.
4. $\varphi \equiv \forall u\, \psi$: $\mathfrak{M}, s \vDash \varphi$ iff for every $u$-variant $s'$ of $s$, $\mathfrak{M}, s' \vDash \psi$.
5. $\varphi \equiv \exists u\, \psi$: $\mathfrak{M}, s \vDash \varphi$ iff there is an $u$-variant $s'$ of $s$ so that $\mathfrak{M}, s' \vDash \psi$.

**Example syn.5.** 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 (in includes free variables $X$ and $Y$), it is only satisfied relative to a structure $\mathfrak{M}$ together with a variable assignment $s$.

$\mathfrak{M}, s \vDash \forall z\,(Xz \leftrightarrow \neg Yz)$ 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 $\mathfrak{M}, s_1 \vDash \forall z\,(X(z) \leftrightarrow \neg Y(z))$.

By contrast, if we have $s_2(X) = \{1, 2\}$ and $s_2(Y) = \{2, 3\}$, $\mathfrak{M}, s_2 \nvDash \forall z\,(X(z) \leftrightarrow \neg Y(z))$. That's because there is a $z$-variant $s_2'$ of $s_2$ with $s_2'(z) = 2$ where $\mathfrak{M}, s_2' \vDash X(z)$ (since $2 \in s_2'(X)$) but $\mathfrak{M}, s_2' \nvDash \neg Y(z)$ (since also $s_2'(z) \in s_2'(Y)$).

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

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

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

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

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