

int.1 Satisfaction

fol:int:sat:
sec

We can already skip ahead to the semantics of first-order logic once we know what **formulas** are: here, the basic definition is that of a **structure**. For our simple language, a **structure** \mathfrak{M} has just three components: a non-empty set $|\mathfrak{M}|$ called the *domain*, what a picks out in \mathfrak{M} , and what P is true of in \mathfrak{M} . The object picked out by a is denoted $a^{\mathfrak{M}}$ and the set of things P is true of by $P^{\mathfrak{M}}$. A **structure** \mathfrak{M} consists of just these three things: $|\mathfrak{M}|$, $a^{\mathfrak{M}} \in |\mathfrak{M}|$ and $P^{\mathfrak{M}} \subseteq |\mathfrak{M}|$. The general case will be more complicated, since there will be many **predicate symbols** and **constant symbols**, the **constant symbols** can have more than one place, and there will also be **function symbols**.

This is enough to give a definition of satisfaction for **formulas** that don't contain **variables**. The idea is to give an inductive definition that mirrors the way we have defined **formulas**. We specify when an atomic formula is satisfied in \mathfrak{M} , and then when, e.g., $\neg\varphi$ is satisfied in \mathfrak{M} on the basis of whether or not φ is satisfied in \mathfrak{M} . E.g., we could define:

1. $P(a)$ is satisfied in \mathfrak{M} iff $a^{\mathfrak{M}} \in P^{\mathfrak{M}}$.
2. $\neg\varphi$ is satisfied in \mathfrak{M} iff φ is not satisfied in \mathfrak{M} .
3. $(\varphi \wedge \psi)$ is satisfied in \mathfrak{M} iff φ is satisfied in \mathfrak{M} , and ψ is satisfied in \mathfrak{M} as well.

Let's say that $|\mathfrak{M}| = \{0, 1, 2\}$, $a^{\mathfrak{M}} = 1$, and $P^{\mathfrak{M}} = \{1, 2\}$. This definition would tell us that $P(a)$ is satisfied in \mathfrak{M} (since $a^{\mathfrak{M}} = 1 \in \{1, 2\} = P^{\mathfrak{M}}$). It tells us further that $\neg P(a)$ is not satisfied in \mathfrak{M} , and that in turn $\neg\neg P(a)$ is and $(\neg P(a) \wedge P(a))$ is not satisfied, and so on.

The trouble comes when we want to give a definition for the quantifiers: we'd like to say something like, " $\exists v_0 P(v_0)$ is satisfied iff $P(v_0)$ is satisfied." But the **structure** \mathfrak{M} doesn't tell us what to do about **variables**. What we actually want to say is that $P(v_0)$ is satisfied *for some value of* v_0 . To make this precise we need a way to assign **elements** of $|\mathfrak{M}|$ not just to a but also to v_0 . To this end, we introduce **variable assignments**. A **variable assignment** is simply a function s that maps **variables** to **elements** of $|\mathfrak{M}|$ (in our example, to one of 1, 2, or 3). Since we don't know beforehand which **variables** might appear in a **formula** we can't limit which **variables** s assigns values to. The simple solution is to require that s assigns values to *all* **variables** v_0, v_1, \dots . We'll just use only the ones we need.

Instead of defining satisfaction of **formulas** just relative to a **structure**, we'll define it relative to a **structure** \mathfrak{M} and a **variable assignment** s , and write $\mathfrak{M}, s \models \varphi$ for short. Our definition will now include an additional clause to deal with atomic **formulas** containing **variables**:

1. $\mathfrak{M}, s \models P(a)$ iff $a^{\mathfrak{M}} \in P^{\mathfrak{M}}$.
2. $\mathfrak{M}, s \models P(v_i)$ iff $s(v_i) \in P^{\mathfrak{M}}$.
3. $\mathfrak{M}, s \models \neg\varphi$ iff not $\mathfrak{M}, s \models \varphi$.

4. $\mathfrak{M}, s \models (\varphi \wedge \psi)$ iff $\mathfrak{M}, s \models \varphi$ and $\mathfrak{M}, s \models \psi$.

Ok, this solves one problem: we can now say when \mathfrak{M} satisfies $P(v_0)$ for the value $s(v_0)$. To get the definition right for $\exists v_0 P(v_0)$ we have to do one more thing: We want to have that $\mathfrak{M}, s \models \exists v_0 P(v_0)$ iff $\mathfrak{M}, s' \models P(v_0)$ for *some* way s' of assigning a value to v_0 . But the value assigned to v_0 does not necessarily have to be the value that $s(v_0)$ picks out. We'll introduce a notation for that: if $m \in |\mathfrak{M}|$, then we let $s[m/v_0]$ be the assignment that is just like s (for all **variables** other than v_0), except to v_0 it assigns m . Now our definition can be:

5. $\mathfrak{M}, s \models \exists v_i \varphi$ iff $\mathfrak{M}, s[m/v_i] \models \varphi$ for some $m \in |\mathfrak{M}|$.

Does it work out? Let's say we let $s(v_i) = 0$ for all $i \in \mathbb{N}$. $\mathfrak{M}, s \models \exists v_0 P(v_0)$ iff there is an $m \in |\mathfrak{M}|$ so that $\mathfrak{M}, s[m/v_0] \models P(v_0)$. And there is: we can choose $m = 1$ or $m = 2$. Note that this is true even if the value $s(v_0)$ assigned to v_0 by s itself—in this case, 0—doesn't do the job. We have $\mathfrak{M}, s[1/v_0] \models P(v_0)$ but not $\mathfrak{M}, s \models P(v_0)$.

If this looks confusing and cumbersome: it is. But the added complexity is required to give a precise, inductive definition of satisfaction for all **formulas**, and we need something like it to precisely define the semantic notions. There are other ways of doing it, but they are all equally (in)elegant.

Photo Credits

Bibliography