com.1  Identity

The construction of the term model given in the preceding section is enough to establish completeness for first-order logic for sets $\Gamma$ that do not contain $\equiv$. The term model satisfies every $\varphi \in \Gamma^*$ which does not contain $\equiv$ (and hence all $\varphi \in \Gamma$). It does not work, however, if $\equiv$ is present. The reason is that $\Gamma^*$ then may contain a sentence $t = t'$, but in the term model the value of any term is that term itself. Hence, if $t$ and $t'$ are different terms, their values in the term model—i.e., $t$ and $t'$, respectively—are different, and so $t = t'$ is false. We can fix this, however, using a construction known as “factoring.”

**Definition com.1.** Let $\Gamma^*$ be a consistent and complete set of sentences in $\mathcal{L}$. We define the relation $\approx$ on the set of closed terms of $\mathcal{L}$ by

$t \approx t' \iff t = t' \in \Gamma^*$

**Proposition com.2.** The relation $\approx$ has the following properties:

1. $\approx$ is reflexive.
2. $\approx$ is symmetric.
3. $\approx$ is transitive.
4. If $t \approx t'$, $f$ is a function symbol, and $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n$ are terms, then

$$f(t_1, \ldots, t_{i-1}, t, t_{i+1}, \ldots, t_n) \approx f(t_1, \ldots, t_{i-1}, t', t_{i+1}, \ldots, t_n).$$

5. If $t \approx t'$, $R$ is a predicate symbol, and $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n$ are terms, then

$$R(t_1, \ldots, t_{i-1}, t, t_{i+1}, \ldots, t_n) \in \Gamma^* \iff R(t_1, \ldots, t_{i-1}, t', t_{i+1}, \ldots, t_n) \in \Gamma^*.$$

**Proof.** Since $\Gamma^*$ is consistent and complete, $t = t' \in \Gamma^*$ iff $\Gamma^* \vdash t = t'$. Thus it is enough to show the following:

1. $\Gamma^* \vdash t = t$ for all terms $t$.
2. If $\Gamma^* \vdash t = t'$ then $\Gamma^* \vdash t' = t$.
3. If $\Gamma^* \vdash t = t'$ and $\Gamma^* \vdash t' = t''$, then $\Gamma^* \vdash t = t''$.
4. If $\Gamma^* \vdash t = t'$, then

$$\Gamma^* \vdash f(t_1, \ldots, t_{i-1}, t, t_{i+1}, \ldots, t_n) = f(t_1, \ldots, t_{i-1}, t', t_{i+1}, \ldots, t_n)$$

for every $n$-place function symbol $f$ and terms $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n$. 
5. If $I^* \vdash t = t'$ and $I^* \vdash R(t_1, \ldots, t_i-1, t, t_{i+1}, \ldots, t_n)$, then $I^* \vdash R(t_1, \ldots, t_i-1, t', t_{i+1}, \ldots, t_n)$ for every $n$-place predicate symbol $R$ and terms $t_1, \ldots, t_i-1, t_{i+1}, \ldots, t_n$.

$\square$

**Problem com.1.** Complete the proof of Proposition com.2.

**Definition com.3.** Suppose $I^*$ is a consistent and complete set in a language $L$, $t$ is a term, and $\approx$ as in the previous definition. Then:

$$\{t\}_\approx = \{t' : t' \in \text{Trm}(L), t \approx t'\}$$

and $\text{Trm}(L)/_\approx = \{\{t\}_\approx : t \in \text{Trm}(L)\}$.

**Definition com.4.** Let $\mathcal{M} = \mathcal{M}(I^*)$ be the term model for $I^*$ from ?? Then $\mathcal{M}/_\approx$ is the following structure:

1. $[\mathcal{M}/_\approx] = \text{Trm}(L)/_\approx$.
2. $c^{\mathcal{M}/_\approx} = [c]_\approx$
3. $f^{\mathcal{M}/_\approx}([t_1]_\approx, \ldots, [t_n]_\approx) = [f(t_1, \ldots, t_n)]_\approx$
4. $([t_1]_\approx, \ldots, [t_n]_\approx) \in R^{\mathcal{M}/_\approx}$ iff $\mathcal{M} \models R(t_1, \ldots, t_n)$, i.e., iff $R(t_1, \ldots, t_n) \in I^*$.

Note that we have defined $f^{\mathcal{M}/_\approx}$ and $R^{\mathcal{M}/_\approx}$ for elements of $\text{Trm}(L)/_\approx$ by referring to them as $[t]_\approx$, i.e., via representatives $t \in [t]_\approx$. We have to make sure that these definitions do not depend on the choice of these representatives, i.e., that for some other choices $t'$ which determine the same equivalence classes $([t]_\approx = [t']_\approx)$, the definitions yield the same result. For instance, if $R$ is a one-place predicate symbol, the last clause of the definition says that $[t]_\approx \in R^{\mathcal{M}/_\approx}$ iff $\mathcal{M} \models R(t)$. If for some other term $t'$ with $t \approx t'$, $\mathcal{M} \not\models R(t)$, then the definition would require $[t']_\approx \notin R^{\mathcal{M}/_\approx}$. If $t \approx t'$, then $[t]_\approx = [t']_\approx$, but we can’t have both $[t]_\approx \in R^{\mathcal{M}/_\approx}$ and $[t]_\approx \notin R^{\mathcal{M}/_\approx}$. However, Proposition com.2 guarantees that this cannot happen.

**Proposition com.5.** $\mathcal{M}/_\approx$ is well defined, i.e., if $t_1, \ldots, t_n, t'_1, \ldots, t'_n$ are terms, and $t_i \approx t'_i$ then

1. $[f(t_1, \ldots, t_n)]_\approx = [f(t'_1, \ldots, t'_n)]_\approx$, i.e.,

$$f(t_1, \ldots, t_n) \approx f(t'_1, \ldots, t'_n)$$

and

2. $\mathcal{M} \models R(t_1, \ldots, t_n)$ iff $\mathcal{M} \models R(t'_1, \ldots, t'_n)$, i.e.,

$$R(t_1, \ldots, t_n) \in I^* \iff R(t'_1, \ldots, t'_n) \in I^*.$$

**Proof.** Follows from Proposition com.2 by induction on $n$. $\square$
As in the case of the term model, before proving the truth lemma we need the following lemma.

**Lemma com.6.** Let $\mathfrak{M} = \mathfrak{M}(\Gamma^*)$, then $\text{Val}^\mathfrak{M}/\approx (t) = [t]_\approx$.

*Proof.* The proof is similar to that of ??.

**Problem com.2.** Complete the proof of Lemma com.6.

**Lemma com.7.** $\mathfrak{M}/\approx \models \varphi$ iff $\varphi \in \Gamma^*$ for all sentences $\varphi$.

*Proof.* By induction on $\varphi$, just as in the proof of ?? . The only case that needs additional attention is when $\varphi \equiv t = t'$.

\[
\mathfrak{M}/\approx \models t = t' \iff [t]_\approx = [t']_\approx \quad (\text{by definition of } \mathfrak{M}/\approx )
\]
\[
\quad \text{iff } t \approx t' \quad (\text{by definition of } [t]_\approx )
\]
\[
\quad \text{iff } t = t' \in \Gamma^* \quad (\text{by definition of } \approx ).
\]

Note that while $\mathfrak{M}(\Gamma^*)$ is always enumerable and infinite, $\mathfrak{M}/\approx$ may be finite, since it may turn out that there are only finitely many classes $[t]_\approx$. This is to be expected, since $\Gamma$ may contain sentences which require any structure in which they are true to be finite. For instance, $\forall x \forall y x = y$ is a consistent sentence, but is satisfied only in structures with a domain that contains exactly one element.

**Photo Credits**

**Bibliography**