

Frank Richter
Semantik II

Syntax and Semantics of a Higher Order Logic

Types All expressions of our higher-order language, $\mathcal{L}_{\text{Type}}$, will be typed. To keep the language simple, we will only use two basic types, e (for the basic entities in the domain) and t (for the truth values 0 and 1).

Definition 1 Types

Type is the smallest set such that

- i.) $e \in \text{Type}$,
- ii.) $t \in \text{Type}$,
- iii.) for each $\tau_1 \in \text{Type}$, for each $\tau_2 \in \text{Type}$, $\langle \tau_1, \tau_2 \rangle \in \text{Type}$.

Syntax The basic expressions of $\mathcal{L}_{\text{Type}}$ consist only of variables and constants. In contrast to first order logic there is no distinction between terms and formulae.

Definition 2 Basic Expressions

- i.) For each $\tau \in \text{Type}$, Var_τ is the smallest set such that for each $n \in \mathbb{N}_0$,
 $v_{n,\tau} \in \text{Var}_\tau$.
- ii.) For each $\tau \in \text{Type}$, Const_τ is the smallest set such that for each $n \in \mathbb{N}_0$,
 $c_{n,\tau} \in \text{Const}$.

We write Var for the set of all variables, $\bigcup_{\tau \in \text{Type}} \text{Var}_\tau$, and Const for the set of all constants, $\bigcup_{\tau \in \text{Type}} \text{Const}_\tau$.

The set of *basic expressions* of our language is the union of the set of variables and the set of constants.

Definition 3 Meaningful Expressions

The meaningful expressions of $\mathcal{L}_{\text{Type}}$ are the smallest family $(\text{ME}_\tau)_{\tau \in \text{Type}}$ such that

- i.) for each $\tau \in \text{Type}$, for each $n \in \mathbb{N}_0$, for each variable $v_{n,\tau} \in \text{Var}_\tau$,
 $v_{n,\tau} \in \text{ME}_\tau$;
- ii.) for each $\tau \in \text{Type}$, for each $n \in \mathbb{N}_0$, for each constant $c_{n,\tau} \in \text{Const}_\tau$,
 $c_{n,\tau} \in \text{ME}_\tau$;

- iii.) for each $\tau \in \text{Type}$, for each $\phi_\tau \in \text{ME}_\tau$, for each $\psi_\tau \in \text{ME}_\tau$,
 $(\phi_\tau \equiv \psi_\tau)_t \in \text{ME}_t$;
- iv.) for each $\phi_{\langle \tau_2, \tau_1 \rangle} \in \text{ME}_{\langle \tau_2, \tau_1 \rangle}$, for each $\psi_{\tau_2} \in \text{ME}_{\tau_2}$,
 $(\phi_{\langle \tau_2, \tau_1 \rangle}(\psi_{\tau_2}))_{\tau_1} \in \text{ME}_{\tau_1}$;
- v.) for each $\tau_1 \in \text{Type}$, for each $\tau_2 \in \text{Type}$, for each $n \in \mathbb{N}_0$, for each $v_{n, \tau_2} \in \text{Var}$, for each $\phi_{\tau_1} \in \text{ME}_{\tau_1}$,
 $(\lambda v_{n, \tau_2}. \phi_{\tau_1})_{\langle \tau_2, \tau_1 \rangle} \in \text{ME}_{\langle \tau_2, \tau_1 \rangle}$.

Semantics D_e is a set of entities, and $D_t = \{0, 1\}$. For each $\tau_1 \in \text{Type}$, for each $\tau_2 \in \text{Type}$, $D_{\langle \tau_1, \tau_2 \rangle} = D_{\tau_2}^{D_{\tau_1}}$ (the set of all functions from D_{τ_1} to D_{τ_2}). Let I be a function assigning a denotation to each non-logical constant, $c_{n, \tau}$, of $\mathcal{L}_{\text{Type}}$ from the set D_τ .

Let $M = \langle D_e, I \rangle$. We will call each M a *model*.

Let g be a function in $\bigcup_{\tau \in \text{Type}} \left(D_\tau^{\text{Var}_\tau} \right)$ which assigns an object (of the appropriate type) in the domain $\bigcup_{\tau \in \text{Type}} D_\tau$ to each variable in Var . We call each g an *assignment function*.

Assume that v is a variable of type τ and d is an element of D_τ . We will use the notation g^d for the assignment function g' which differs from the assignment function g in the following way:

For each $\tau \in \text{Type}$, for each $v \in \text{Var}_\tau$, for each $x \in \text{Var}_\tau$, for each $d \in D_\tau$,

$$g^d(x) = \begin{cases} d & \text{if } x = v, \text{ and} \\ g(x) & \text{otherwise.} \end{cases}$$

Definition 4 Denotation

Let M be a model and g an assignment function.

- i.) For each $\tau \in \text{Type}$, for each $n \in \mathbb{N}_0$, for each variable $v_{n, \tau} \in \text{Var}_\tau$,
 $\llbracket v_{n, \tau} \rrbracket^{M, g} = g(v_{n, \tau})$.
- ii.) For each $\tau \in \text{Type}$, for each $n \in \mathbb{N}_0$, for each constant $c_{n, \tau} \in \text{Const}_\tau$,
 $\llbracket c_{n, \tau} \rrbracket^{M, g} = I(c_{n, \tau})$.
- iii.) For each $\tau \in \text{Type}$, for each $\phi_\tau \in \text{ME}_\tau$, for each $\psi_\tau \in \text{ME}_\tau$,
 $\llbracket (\phi_\tau \equiv \psi_\tau)_t \rrbracket^{M, g} = 1$ iff $\llbracket \phi_\tau \rrbracket^{M, g} = \llbracket \psi_\tau \rrbracket^{M, g}$.
- iv.) For each $\phi_{\langle \tau_2, \tau_1 \rangle} \in \text{ME}_{\langle \tau_2, \tau_1 \rangle}$, for each $\psi_{\tau_2} \in \text{ME}_{\tau_2}$,
 $\llbracket (\phi_{\langle \tau_2, \tau_1 \rangle}(\psi_{\tau_2}))_{\tau_1} \rrbracket^{M, g} = \llbracket \phi_{\langle \tau_2, \tau_1 \rangle} \rrbracket^{M, g} \left(\llbracket \psi_{\tau_2} \rrbracket^{M, g} \right)$.
- v.) For each $\tau_1 \in \text{Type}$, for each $\tau_2 \in \text{Type}$, for each $n \in \mathbb{N}_0$, for each $v_{n, \tau_2} \in \text{Var}$, for each $\phi_{\tau_1} \in \text{ME}_{\tau_1}$,
 $\llbracket (\lambda v_{n, \tau_2}. \phi_{\tau_1})_{\langle \tau_2, \tau_1 \rangle} \rrbracket^{M, g}$ is that function h from D_{τ_2} to D_{τ_1} such that for each $o \in D_{\tau_2}$, $h(o) = \llbracket \phi_{\tau_1} \rrbracket^{M, g \overset{o}{v}_{n, \tau_2}}$.

Standard results tell us that the usual logical connectives and quantifiers can be defined using these five clauses.