Realizability

Samuel Mimram

22 November 2021

Second-order logic. We suppose fixed a set $\mathcal{T}$ of *generators* of given arity and write $\mathcal{T}^*$ for the set of generated terms. The syntax of second-order formulas is

$$A \qquad ::= \qquad X(a_1, \ldots, a_n) \quad | \quad A \Rightarrow B \quad | \quad \forall x.A \quad | \quad \forall X.A$$

where $a_i \in \mathcal{T}^*$, each second-order formula X having a fixed arity n. We consider typing rules which extend those of simply-typed λ-calculus by

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall x.A} \qquad \frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall X.A} \qquad \frac{\Gamma \vdash t : \forall x.A}{\Gamma \vdash t : A[a/x]} \qquad \frac{\Gamma \vdash t : \forall X.A}{\Gamma \vdash t : A[B/X]}$$

where, in the first two rules, we suppose x and X not free in Γ respectively.

1. Show that identity can be given the type $\forall X.X \Rightarrow X$.
2. Recall the elimination rule for $\vee$. How can we encode this operator into our logic?
3. Similarly, provide an encoding of the operators $\wedge$, $\perp$, $\neg$ and existential quantifications.

Realizability. We write Λ for the set of λ-*terms* and Π for the set of *stacks*, which are sequences $t_1 \cdot t_2 \cdots t_n$ of λ-terms. *Processes* are elements (t, π) of $\Lambda \times \Pi$, often written $t \star \pi$. The *reduction* relation $\succ$ between processes is given by the following two rules:

$$tu \star \pi \succ t \star u \cdot \pi$$
$$\lambda x.t \star u \cdot \pi \succ t[u/x] \star \pi$$

An element of $\mathcal{P}(\Pi)$ is called a *truth value*. Suppose fixed a set $\perp\!\!\!\perp$ of processes closed under anti-reduction. We define an interpretation $[\![A]\!] \in \mathcal{P}(\Pi)$ by induction on the formula A by

$$[\![A \Rightarrow B]\!] = \{t \cdot \pi \mid t \in |A|, \pi \in [\![B]\!]\} \qquad [\![\forall x.A]\!] = \bigcup_{a \in \mathcal{T}^*} [\![A[a/x]]\!] \qquad [\![\forall X.A]\!] = \bigcup_{V \in \mathcal{P}(\Pi)} [\![A[V/X]]\!]$$

where

$$|A| = \{t \in \Lambda \mid \forall \pi \in [\![A]\!], t \star \pi \in \perp\!\!\!\perp\}$$

denotes the set of *realizers* of the formula A. Above, we have supposed fixed an interpretation of the first- and second-order free variables (by abuse of notation, given $V \in \mathcal{P}(\Pi)$, we still write V for a variable whose interpretation is V). We write $t \Vdash A$ when $t \in |A|$ and say that t *realizes* A.

4. What are $[\![\perp]\!]$ and $|\perp|$?

Identity-like terms. Our goal is now to characterize the behavior of terms of type $\forall X.X \Rightarrow X$.

5. Give examples of terms which are of type $\forall X.X \Rightarrow X$.

6. Show that $(\lambda x.x) \star u \cdot \pi \succ u \star \pi$.

A term $t \in \Lambda$ is *identity-like* when $t \star u \cdot \pi \succ u \star \pi$ for every $u \in \Lambda$ and $\pi \in \Pi$.

7. Show that if t is identity-like then $t \Vdash \forall X.X \Rightarrow X$.

We admit the *adequation lemma*: if $x_1 : A_1, \ldots, x_n : A_n \vdash t : A$ is derivable and $\forall i, t_i \Vdash A_i$ then $t[t_1/x_1, \ldots, t_n/x_n] \Vdash A$.

8. Show the converse to previous question, i.e. $\vdash \theta : \forall X.X \Rightarrow X$ implies that t is identity-like (hint: use a suitably chosen $\perp\!\!\!\perp$).

9. Give an example of an identity-like term which is not the identity, and even non-typable.

Booleans.

10. Suppose that our signatures contains constants 0 and 1. Define a predicate $\mathrm{Bool}(x)$, which encodes the fact that x is a boolean.

11. Show that $\vdash \theta : \mathrm{Bool}(0)$ implies $\theta \star t \cdot u \cdot \pi \succ t \star \pi$ (and similarly for $\vdash \theta : \mathrm{Bool}(1)$).

The adequation lemma.

12. Show that $t \Vdash A \Rightarrow B$ and $u \Vdash A$ implies $tu \Vdash B$.

13. Show that if for every $u \in \Lambda$, $u \Vdash A$ implies $t[u/x] \Vdash B$, then $\lambda x.t \Vdash A \Rightarrow B$.

14. Prove the adequation lemma.