15-814 Fall 2003, Homework #6 selected solutions

Aleksey Kliger

2 December 2003

Please see the book's solutions to 15.2.5 and 15.3.6.

Problem 1 (26.2.2). *Give a couple of examples of pairs of types that are related by the subtype relation of full* $\mathsf{F}_{<:}$ *but are not subtypes in kernel* $\mathsf{F}_{<:}$*.*

Solution Recall that in kernel $\mathsf{F}_{<:}$, the bounds on the parameters of two quantified types $\forall X{<:}T_1.S_1$ and $\forall X{<:}T_2.S_2$ must be the same (*i.e.,* $T_1 = T_2$). So one easy example that is typable in full $\mathsf{F}_{<:}$ is

$$\dfrac{\begin{array}{c}\vdots\\ \vdash \{a{:}\mathsf{nat}, b : \mathsf{nat}\} <: \{a{:}\mathsf{nat}\}\end{array} \qquad \dfrac{}{X <: \{a{:}\mathsf{nat}, b{:}\mathsf{nat}\} \vdash T <: T}}{\vdash \forall X{<:}\{a{:}\mathsf{nat}\}.T <: \forall X{<:}\{a{:}\mathsf{nat}, b{:}\mathsf{nat}\}.T}$$

for any type T.

Problem 2. *Suppose* $\mathsf{Counter}$*,* c_1 *and* c_2 *are defined as follows:*

$$\mathsf{Counter} = \exists\alpha.\alpha \times (\alpha \to \mathsf{nat}) \times (\alpha \to \alpha)$$

$$c_1 = \{*\mathsf{nat}, \langle 0, \lambda x{:}\mathsf{nat}.x, \lambda x{:}\mathsf{nat}.\mathsf{succ}\ x\rangle\}\ \mathsf{as\ Counter}$$

$$c_2 = \{*\mathsf{nat}{\times}\mathsf{nat}, \langle\langle 0, 0\rangle, \lambda x{:}\mathsf{nat} \times \mathsf{nat}.x.1 + x.2, \lambda x{:}\mathsf{nat} \times \mathsf{nat}.\langle \mathsf{succ}\ x.2, x.1\rangle\rangle\}\ \mathsf{as\ Counter}$$

Show that $c_1 \approxeq c_2 : \mathsf{Counter}$

Solution Let $v_1 = \langle 0, \lambda x{:}\mathsf{nat}.x, \lambda x{:}\mathsf{nat}.\mathsf{succ}\ x\rangle$ and

$$v_2 = \langle\langle 0, 0\rangle, \lambda x{:}\mathsf{nat} \times \mathsf{nat}.x.1 + x.2, \lambda x{:}\mathsf{nat} \times \mathsf{nat}.\langle \mathsf{succ}\ x.2, x.1\rangle\rangle$$

respectively.

Let $Q = \alpha \times (\alpha \to \mathsf{nat}) \times (\alpha \to \alpha)$

Recall the definition of logical equivalence (given in Figure 1, extended to n-ary product types). By definition, it suffices to show that for some candiate $C = (\mathsf{nat}, \mathsf{nat} \times \mathsf{nat}, R)$, $v_1 \approxeq_{\mathsf{val}} v_2 : Q[C/\alpha]$, for some appropriate R. Indeed aside from pushing through the definition, the problem essentially boils down to finding the appropriate relation R.

$t_1 \approxeq t_2 : Q$	whenever $t_1 \Downarrow$ iff $t_2 \Downarrow$ and for all v_1, v_2 if $t_i \to^* v_i$, then $v_1 \approxeq_{\mathsf{val}} v_2 : T$
$v_1 \approxeq_{\mathsf{val}} v_2 : b$	$v_1 = v_2$, where b is a base type (such as nat)
$v_1 \approxeq_{\mathsf{val}} v_2 : (T_1, T_2, R)$	whenever $(v_1, v_2) \in R$
$v_1 \approxeq_{\mathsf{val}} v_2 : Q_1 \times \cdots \times Q_n$	whenever for each $1 \leq i \leq n$, $v_1.i \approxeq v_2.i : Q_i$
$v_1 \approxeq_{\mathsf{val}} v_2 : Q_1 \to Q_2$	whenever for all $v_1' \in \mathsf{Val}(Left(Q_1))$ and $v_2' \in \mathsf{Val}(Right(Q_1))$, if $v_1' \approxeq_{\mathsf{val}} v_2' : Q_1$ then $v_1 v_1' \approxeq v_2 v_2' : Q_2$
$v_1 \approxeq_{\mathsf{val}} v_2 : \forall \alpha.Q$	whenever for all candidates $C = (T_1, T_2, R)$, $v_1[T_1] \approxeq v_2[T_2] : Q[C/\alpha]$
$v_1 \approxeq_{\mathsf{val}} v_2 : \exists \alpha.Q$	whenever $v_1 = \{*T_1, v_1'\}$ and $v_2 = \{*T_2, v_2'\}$ and there is some candidate $C = (T_1, T_2, R)$ such that $v_1' \approxeq_{\mathsf{val}} v_2 : Q[C/\alpha]$

Figure 1: Logical Equivalence

Informally, the second counter "works" by alternately incrementing either the first or second component of the pair, in the end when converting the counter to a natural number it adds up the components to get the total number of increments. So it is an invariant of the second counter that a counter's value is $i + j$ where $\langle i, j \rangle$ is the representation. So one possibility for the relation R is $\{(n, \langle i, j \rangle) | i + j \to^* n\}$[1].

Returning to the proof, it suffices to show (from the definition of logical equivalence for n-tuples) that $v_1.i \approxeq v_2.i : Q_i$ for $i = 1, 2, 3$ where $Q_1 = C, Q_2 = C \to \mathsf{nat}, Q_3 = C \to C$.

In each of the three cases, evidently $v_1.i \Downarrow$ iff $v_2.i \Downarrow$, and indeed we can take an evaluation step to get at the appropriate component of v_i.

So it suffices to show:

1. $0 \approxeq_{\mathsf{val}} \langle 0, 0 \rangle : C$
2. $\lambda x{:}\mathsf{nat}.x \approxeq_{\mathsf{val}} \lambda x{:}\mathsf{nat} \times \mathsf{nat}.x.1 + x.2 : C \to \mathsf{nat}$
3. $\lambda x{:}\mathsf{nat}.\mathsf{succ}\ x \approxeq_{\mathsf{val}} \lambda x{:}\mathsf{nat} \times \mathsf{nat}.\langle \mathsf{succ}\ x.2, x.1 \rangle : C \to C$

To show (1), by definition it suffices to show that $(0, \langle 0, 0 \rangle) \in R$. And since evidently $0 + 0 \to^* 0$, it holds.

To show (2), suffices to show that if $w_1, w_2 \in R$ (where $\vdash w_1 : \mathsf{nat}$, $\vdash w_2 : \mathsf{nat} \times \mathsf{nat}$), then $(\lambda x{:}\mathsf{nat}.x) w_1 \approxeq (\lambda x{:}\mathsf{nat} \times \mathsf{nat}.x.1 + x.2) w_2 : \mathsf{nat}$. By taking a few steps of evaluation, we see that it suffices to show $w_1 \approxeq w_2.1 + w_2.2 : \mathsf{nat}$. Now since w_2 has type $\mathsf{nat} \times \mathsf{nat}$, by canonical forms, $w_2 = \langle w_{21}, w_{22} \rangle$. So by some more evaluation, we see it suffices to show that $w_1 \approxeq w_{21} + w_{22} : \mathsf{nat}$.

[1] Here $+$ is the addition operation of the programming language. Another possibility is $R' = \{(n, \langle i, j \rangle) | i + j = n\}$ where $+$ is the mathematical operation of addition. The proof works for either relation, although it is somewhat shorter for R

However recall that $(w_1, \langle w_{21}, w_{22}\rangle) \in R$. So $w_{21} + w_{22} \to^* w_1$. So we have $w_1 \approxeq_{\mathsf{val}} w_1 : \mathsf{nat}$ which is true by definition of logical equivalence at base type.

To show (3), suffices to show that if $(w_1, w_2) \in R$, that $(\lambda x{:}\mathsf{nat}.\mathsf{succ}\ x)w_1 \approxeq (\lambda x{:}\mathsf{nat} \times \mathsf{nat}.\langle \mathsf{succ}\ x.2, x.1\rangle)w_2 : C$. By taking a few steps of evaluation, it suffices to show that $\mathsf{succ}\ w_1 \approxeq_{\mathsf{val}} \langle \mathsf{succ}\ w_{22}, w_{21}\rangle : C$ where $w_2 = \langle w_{21}, w_{22}\rangle$. By definition of logical equivalence at a candidate, it suffices to show that if $(\mathsf{succ}\ w_1, \langle \mathsf{succ}\ w_{22}, w_{21}\rangle) \in R$. That is, we wish to show that $\mathsf{succ}\ w_{22} + w_{21} \to^* \mathsf{succ}\ w_1$. However this follows easily from $(w_1, w_2) \in R$ by a lemma about natural numbers:

Lemma 2.1. *If* $v_1 + v_2 \to^* v$ *then* $\mathsf{succ}\ v_1 + v_2 \to^* \mathsf{succ}\ v$.

Proof. By induction on v_1, using the definition of $+$. □

Problem 3. *Suppose* $\vdash f : \forall\alpha.\alpha \to \alpha \to \alpha$, *and* $\vdash v_1 : T$ *and* $\vdash v_2 : T$. *Show that if* $f[T]v_1v_2 \to^* v$ *then either* $v = v_1$ *or* $v = v_2$.

Solution We will use parametricity and an appropriate candidate $C = (T, T, R)$ to show that this is the case. The choice of a particular R will be critical. One way to pick the appropriate R is to proceed with the proof while holding R abstract, and collect from the chain of logical inference a set of constraints on R. Then find an appropriate relation that makes those constraints true.

So, by parametricity, $f \approxeq_{\mathsf{val}} f : \forall\alpha.\alpha \to \alpha \to \alpha$. Therefore, from the definition of logical equivalence, for all candidates C, $f[Left(C)] \approxeq f[Right(C)] : C \to C \to C$. So pick $C = (T, T, R)$. Then since $f[T]v_1v_2 \to^* v$, it follows that $f[T] \to^* w$. Therefore, by definition of logical equivalence, $w \approxeq_{\mathsf{val}} w : C \to C \to C$. By definition of logical equivalence for values of arrow type, since $v_1 \approxeq_{\mathsf{val}} v_1 : C$[2], it follows that $wv_1 \approxeq wv_1 : C \to C$. Again since $f[T]v_1v_2 \to^* v$, it follows that $wv_1 \to^* w'$ and $w' \approxeq_{\mathsf{val}} w' : C \to C$. By definition of logical equivalence for values of arrow type, since $v_2 \approxeq_{\mathsf{val}} v_2 : C$[3], it follows that $w'v_2 \approxeq w'v_2 : C$. Finally, since $w'v_2 \to^* v$, it follows that $v \approxeq_{\mathsf{val}} v : C$. By definition of logical equivalence of values at a candidate, that means that $(v, v) \in R$. Therefore[4], $v = v_1$ or $v = v_2$.

So, spelling out the details from the footnotes, it turns out that we chose $R = \{(v_1, v_1), (v_2, v_2)\}$.

[2] Note that I haven't picked what R is yet, but I am adding the constraint that whatever it is, it has to at least include (v_1, v_1)

[3] ...and R must include at least (v_2, v_2)

[4] ... and those must be are the only things in the relation R