MRI master class Computer Mathematics

For the course notes see `http://www.cs.kun.nl/ herman/lect2.ps.gz`

Exercises 2: Polymorphic λ-calculus

Recall the following definitions

$$\begin{array}{rclrcl} \bot & := & \forall\alpha.\alpha & \neg\sigma & := & \sigma\to\bot \\ \sigma\wedge\tau & := & \forall\alpha.(\sigma\to\tau\to\alpha)\to\alpha & \sigma\vee\tau & := & \forall\alpha.(\sigma\to\alpha)\to(\tau\to\alpha)\to\alpha \\ \exists\alpha.\sigma & := & \forall\beta.(\forall\alpha.\sigma\to\beta)\to\beta & & & \end{array}$$

1. Find inhabitants of the following types (both in the Curry and the Church variant of $\lambda 2$).

   (a) $\forall\alpha\forall\beta.\beta\to\alpha\to\beta$
   $\lambda x.\lambda y.x$
   $\lambda\alpha\lambda\beta.\lambda x{:}\beta.\lambda y{:}\alpha.x$

   (b) $\sigma\to\neg\neg\sigma$
   $\lambda x.\lambda y.yx$
   $\lambda x{:}\sigma.\lambda y{:}\neg\sigma.yx$

   (c) $\neg(\alpha\to\beta)\to\beta\to\alpha$
   $\lambda x.\lambda y.x(\lambda z.y)$
   $\lambda x{:}\neg(\alpha\to\beta).\lambda y{:}\beta.x(\lambda z{:}\alpha.y)\alpha$

2. Assign a type to the following terms in the Curry variant of $\lambda 2$.

   (a) $\lambda x.xx$.
   There are (countably) many possible answers; here are two, including the possible Church-variants, to indicate how these typings may arise: (a) $\bot\to\bot$ $(\lambda x{:}\bot.x(\bot\to\bot)x)$ or (b) $(\forall\alpha.\alpha\to\alpha)\to\sigma$ for arbitrary σ $((\lambda x{:}(\forall\alpha.\alpha\to\alpha).x(\sigma\to\sigma)(x\sigma))$.

   (b) $\lambda x.x(\lambda z.x)$.
   Two possible answers, including possible Church-variants: (a) $\bot\to\bot$ $(\lambda x{:}\bot.x(\sigma\to\bot)\to\bot(\lambda z{:}\sigma.x))$ or (b) $\forall\gamma(\forall\alpha\beta.\beta\to\alpha)\to\gamma$ (denote $(\forall\alpha\beta.\beta\to\alpha)$ as σ and let τ be an arbitrary type, then we have $\lambda\gamma.\lambda x{:}\sigma.x\gamma(\tau\to\sigma)(\lambda z{:}\tau.x)$.)

   NB: Yes, all normal forms can be typed in $\lambda 2$ by assigning $\bot$ to all the variables (free or bound) within the term. A nice additional exercise to do!

3. One can show the derivability of the logical rules for the conectives as defined above. Find closed terms of the following types.

   (a) $\sigma\to\tau\to(\sigma\wedge\tau)$ ($\wedge$-intro)
   $\lambda x{:}\sigma.\lambda y{:}\tau.\lambda\alpha.\lambda h{:}\sigma\to\tau\to\alpha.hxy$.

(b) $\sigma \to (\sigma \vee \tau)$ ($\vee$-intro)
$\lambda x{:}\sigma.\lambda\alpha.\lambda f{:}\sigma\to\alpha.\lambda g{:}\tau\to\alpha.fx.$

(c) $\forall\alpha.\sigma\to\neg\sigma\to\alpha$ ($\neg$-elim)
$\lambda\alpha.\lambda x{:}\sigma.\lambda y{:}\neg\sigma.yx\alpha$

(d) $\sigma[\tau/\alpha]\to\exists\alpha.\sigma$ ($\exists$-intro).
$\lambda x{:}\sigma[\tau/\alpha].\lambda\beta.\lambda h{:}\forall\alpha(\sigma\to\alpha).h\tau x$

4. Define the following functions in $\lambda 2$. (The functions Plus and Mult, given in the course notes and can be taken for granted.)

   (a) The faculty function on the natural numbers: Fac : Nat→Nat such that $\mathrm{Fac}\,0 = 1$ and $\mathrm{Fac}\,(S\,n) = (S\,n) * (\mathrm{Fac}\,n)$.
   Sorry, but I meant the exponentiation function: $\mathrm{Exp}\,m\,0 = 1$ and $\mathrm{Exp}\,m\,(S\,n) = m * (\mathrm{Exp}\,m\,n)$.
   $\mathrm{Exp} := \lambda m, n{:}\mathrm{Nat}.\mathrm{It}\,1\,(\lambda x{:}\mathrm{Nat}.\mathrm{Mult}\,m\,x)n$

   (b) The SumList function that sums all elements of a list of numbers: SumList : $\mathrm{List}_{\mathrm{Nat}}$→Nat such that $\mathrm{SumList}\,\mathrm{Nil} = 0$ and $\mathrm{SumList}\,(\mathrm{Cons}\,n\,l) = n{+}(\mathrm{SumList}\,l)$.
   SumList := It 0 Plus, where It is now the list-iteration.

5. Prove the Soundness of $[\![-]\!]_\rho$ in the proof of Strong Normalization (the Proposition on slide 19):

$$x_1 : \tau_1, \ldots, x_n : \tau_n \vdash M : \sigma \Rightarrow M[P_1/x_1, \ldots, P_n/x_n] \in [\![\sigma]\!]_\rho$$

   for all valuations ρ and $P_1 \in [\![\tau_1]\!]_\rho, \ldots, P_n \in [\![\tau_n]\!]_\rho$
   Ask me for details if you want to know.