Assignment 11

Exercises on lecture 11/chapter 11

19 November 2024

We will work on the following exercises during the next exercise class.

Exercise 11.1 — Give λ_{Y}-terms $\vdash * : \mathbf{N} \to \mathbf{N} \to \mathbf{N}$ and $\vdash \mathrm{fib} : \mathbf{N} \to \mathbf{N} \to \mathbf{N}$ that implement, respectively, the computation of multiplication and Fibonacci numbers.

Exercise 11.2 — Recall that the class of primitive recursive functions on the natural numbers consists of constant 0 maps, successor, projections, composition and primitive recursion. Except the last two, all the others are already built into λ_{Y}, and composition is straightforward to implement in λ_{Y}. A map $h \colon \mathbb{N} \times Y \to Z$ is said to be given by primitive recursion of functions $f \colon Y \to Z$ and $g \colon \mathbb{N} \times Y \times Z \to Z$ if the following two equations hold.

$$h(0, y) = f(y)$$
$$h(n+1, y) = g(n, y, h(n, y))$$

Given terms $t \colon A \to B$ and $s \colon \mathbf{N} \to A \to B \to B$, define primitive recursion as a term $\mathrm{PR}(t, s) \colon \mathbf{N} \to A \to B$ in λ_{Y}.

Exercise 11.3 — Define a λ_{Y}-term of type $\mathbf{N} \to \mathbf{N} \to \mathbf{N}$ that implements the Ackermann function. This is a function that cannot be implemented by just primitive recursion on natural numbers but requires you to use (primitive) recursion on function types.

Exercise 11.4 — The final piece to Turing-completeness, when combined with exercise 11.2, is the so-called minimisation operator or μ-recursion. A function $f \colon \mathbb{N} \to \mathbb{N}_{\bot}$ is said to be given by μ-recursion from a function $g \colon \mathbb{N} \to \mathbb{N}$, if the following holds.

$$f(n) = \begin{cases} \min\{k \in \mathbb{N} \mid k \geq n \text{ and } g(k) = 0\}, & \text{if there is a } k \geq n \text{ with } g(k) = 0 \\ \bot, & \text{otherwise} \end{cases}$$

For a term $t \colon \mathbf{N} \to \mathbf{N}$, give a term $\mathrm{Min}(t) \colon \mathbf{N} \to \mathbf{N}$ that implements minimisation in λ_{Y}.

Exercise 11.5 — Pick a term $t \colon \mathbf{N} \to \mathbf{N}$ and evaluate the term $\mathrm{Min}(t)$ that you constructed in exercise 11.4 on an input using the big-step semantics of λ_{Y}.

Problem 11.6 — The goal of this problem is to show that the product type of λ_{Y} is not strictly necessary by translating a program with product types in $\lambda_{\times}$ into one without in $\lambda_{\to}$. The idea is that a map $A \times B \to R$ is the same as a map $A \to B \to R$ by currying, but this forces that products should only ever occur on the left of an arrow. In order to then remove $A \times B$, we need to turn this type into one with where the product is on the left of an arrow. To this end, let R be a fixed result type and write A^* for the type $A \to R$. We then define a translation $A^\dagger$ of types as follows.

$$\begin{aligned} A^\dagger &= (A^u)^* \\ \mathbf{N}^u &= \mathbf{N}^* \\ (A \times B)^u &= A^\dagger \to B^\dagger \to R \\ (A \to B)^u &= (A^\dagger \to B^\dagger)^* \end{aligned}$$

Clearly, $A^\dagger$ has no product types left in it. For example, we get

$$(\mathbf{N} \times \mathbf{N})^\dagger = (\mathbf{N}^{**} \to \mathbf{N}^{**} \to R) \to R\,.$$

Given a context Γ, we define $\Gamma^\dagger$ to be element-wise translation:

$$(x_1 : A_1, \dots, x_n : A_n)^\dagger = x_1 : A_1^\dagger, \dots, x_n : A_n^\dagger$$

Your task for this problem is to translate a term $\Gamma \vdash t : A$ into a term $\Gamma^\dagger \vdash t^\dagger : A^\dagger$.
The type A^* can be understood as a kind of negation of A, if we see A as a proposition and R is the false proposition. Under this view, A^{**} is like a double negation of A. Analogously to intuitionistic logic, we have a term $\lambda x.\, \lambda f.\, f x : A \to A^{**}$ but there is not necessarily a term going the other direction.

Problem 11.7 — Define evaluation contexts E for λ_Y to be given by the following grammar.

$$E ::= - \mid E\,t \mid \mathbf{succ}\ E \mid \mathbf{pred}\ E \mid \mathbf{if}_0\ E\ \mathbf{then}\ s_1\ \mathbf{else}\ s_2 \mid \langle E, t\rangle \mid \langle t, E\rangle \mid \mathbf{fst}\ E \mid \mathbf{snd}\ E$$

a) Define a relation $\succ$ on λ_Y terms, such that the contextual closure $\longrightarrow$ given by

$$\frac{t \succ s}{E[t] \longrightarrow E[s]}$$

agrees with the big-step operational semantics in the following sense.

b) Recall from problem 4.6 that we denote by $\twoheadrightarrow$ the preorder closure $\mathrm{rt}(\longrightarrow)$ of $\longrightarrow$. Prove that if $t \Downarrow_A v$, then $t \twoheadrightarrow v$.