Types and λ-calculus

Problem Sheet 7

** 1.

(a) Find an inhabitant of the type $(((a \to b) \to b) \to b) \to a \to b$.

(b) Give the corresponding proof of the corresponding formula.

Solution

(a) $\lambda x y. x(\lambda z. z y)$

(b) Suppose $(a \Rightarrow b) \Rightarrow b) \Rightarrow b$ (x) and assume a (y). We claim that $(a \Rightarrow b) \Rightarrow b$, the proof is as follows: suppose $a \Rightarrow b$ (z), then applying this to assumption (y) we get b. Returning to our original proof, from this and (x) we obtain b, as required.

** 2. Sketch an algorithm to decide the following problem and justify that it works:

Given:	two typable terms M and N
Decide:	if $M =_\beta N$

Solution

Given M and N we can simply explore all possible reduction sequences of each of them to obtain their normal forms M' and N' (respectively). This is guaranteed to terminate by Strong Normalisation. Then, by definition of $=_\beta$, it suffices to check if $M = N$. If they are identical terms then answer YES, otherwise answer NO.

** 3. A term M is a *fixed point combinator* just if, for all terms P, $M\ P =_\beta P\ (M\ P)$. In other words, M computes a fixed point of its argument.

Prove that no fixed point combinator is typable, i.e. if closed term M is a fixed point combinator, then M is not typable.

Hint: Try to arrive at a contradiction by obtaining a β-equality in which the two sides are distinct normal forms.

Solution

Suppose M is a fixed point combinator and suppose, for the purpose of obtaining a contradiction, that M is typable. It follows from Strong Normalisation that M has a normal form F. Since M is a fixed point combinator, for all terms P: $M\ P =_\beta P\ (M\ P)$. Since F is the normal form of M, $M \twoheadrightarrow_\beta F$ and it follows by equational reasoning (alternatively by the definition of $=_\beta$) that, for all terms P, $F\ P =_\beta P\ (F\ P)$. We use this with $P := x$, for some variable x, to give $F\ x =_\beta x\ (F\ x)$, call this equation (*). We distinguish two cases with respect to F:

- If F is an abstraction $\lambda y.Q$, then $F\ x =_\beta Q[x/y]$ and (*) is equivalent to $Q[x/y] =_\beta x\ Q[x/y]$. Since $\lambda y.Q$ was in normal form, so is $Q[x/y]$ and hence so is $x\ Q[x/y]$. However, we cannot have two distinct normal forms that are convertible (by definition of $=_\beta$ they must have a common redex), so this is a contradiction.
- If F is not an abstraction, then $F\ x$ is in normal form already and so is $x\ (F\ x)$, and we have the same contradiction as above.

** 4. Suppose we add a fixed point combinator fix to our lambda calculus as a new primitive. In other words, we extend the syntax of terms by the rule:

$$\frac{}{\mathsf{fix} \in \Lambda}$$

and we extend the type system by the following rule:

$$\frac{}{\Gamma \vdash \mathsf{fix} : (A \to A) \to A}\ (\mathsf{TFix})$$

This type makes sense since fix takes a function as input and returns a fixed point of the function. (We should also extend the definition of β-reduction, but it is not important to this question.)

(a) Show that every type in this extended system is inhabited.

(b) What is the consequence for the Curry-Howard correspondence?

Solution

(a) Every type A is inhabited by the term $\mathsf{fix}\ (\lambda x.\, x)$. We can construct the following derivation.

$$\dfrac{\dfrac{}{\vdash \mathsf{fix} : (A \to A) \to A}\,(\mathsf{TFix}) \qquad \dfrac{\dfrac{}{x : A \vdash x : A}\,(\mathsf{TVar})}{\vdash \lambda x.\, x : A \to A}\,(\mathsf{TAbs})}{\vdash \mathsf{fix}\ (\lambda x.\, x) : A}\,(\mathsf{TApp})$$

(b) Hence, by the Curry-Howard correspondence, this would lead to a proof system in which every formula was provable – i.e. an inconsistent logic!