Types and λ-calculus

Problem Sheet 6

Questions 1 and 3(c) will be marked.

* 1. Give a type derivation/proof tree for the judgements:

(a) $\vdash (\lambda x.\, x)\,\underline{2} : \mathsf{Nat}$

(b) $x : \mathsf{Nat},\, y : \mathsf{Nat} \vdash \mathsf{ifz}\ y\ x\ (\mathsf{pred}\ x) : \mathsf{Nat}$

(c) $\vdash \lambda xy.\, yxx : a \to (a \to a \to b) \to b$

(d) $\vdash \lambda xyz.\, y(xz) : (a \to b) \to (b \to c) \to a \to c$

Solution

(a) I have to split up the derivation to keep it on the page. First let δ be the following derivation of $\vdash \underline{2} : \mathsf{Nat}$.

$$\dfrac{\dfrac{}{\vdash \mathsf{S} : \mathsf{Nat} \to \mathsf{Nat}}\,(\mathsf{TCst}) \qquad \dfrac{\dfrac{}{\vdash \mathsf{S} : \mathsf{Nat} \to \mathsf{Nat}}\,(\mathsf{TCst}) \quad \dfrac{}{\vdash \underline{0} : \mathsf{Nat}}\,(\mathsf{TCst})}{\vdash \underline{1} : \mathsf{Nat}}\,(\mathsf{TApp})}{\vdash \underline{2} : \mathsf{Nat}}\,(\mathsf{TApp})$$

Then we extend this as follows:

$$\dfrac{\dfrac{\dfrac{}{x{:}\mathsf{Nat} \vdash x : \mathsf{Nat}}\,(\mathsf{TVar})}{\vdash (\lambda x.\, x) : \mathsf{Nat} \to \mathsf{Nat}}\,(\mathsf{TAbs}) \qquad \delta}{\vdash (\lambda x.\, x)\ \underline{2} : \mathsf{Nat}}\,(\mathsf{TApp})$$

(b) I have to split the derivation up so it fits onto the page. With $\Gamma = \{x : \mathsf{Nat},\, y : \mathsf{Nat}\}$. First, let δ be this derivation of $\Gamma \vdash \mathsf{ifz}\ y\ x : \mathsf{Nat} \to \mathsf{Nat}$:

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \mathsf{ifz} : \mathsf{Nat} \to \mathsf{Nat} \to \mathsf{Nat} \to \mathsf{Nat}}\,(\mathsf{TCst}) \quad \dfrac{}{\Gamma \vdash y : \mathsf{Nat}}\,(\mathsf{TVar})}{\Gamma \vdash \mathsf{ifz}\ y : \mathsf{Nat} \to \mathsf{Nat} \to \mathsf{Nat}}\,(\mathsf{TApp}) \qquad \dfrac{}{\Gamma \vdash x : \mathsf{Nat}}\,(\mathsf{TVar})}{\Gamma \vdash \mathsf{ifz}\ y\ x : \mathsf{Nat} \to \mathsf{Nat}}\,(\mathsf{TApp})$$

Then we extend this derivation δ as follows:

$$
\begin{prooftree}
\AxiomC{δ}
\AxiomC{}
\RightLabel{(TCst)}
\UnaryInfC{$\Gamma \vdash \mathsf{pred} : \mathsf{Nat} \to \mathsf{Nat}$}
\AxiomC{}
\RightLabel{(TVar)}
\UnaryInfC{$\Gamma \vdash x : \mathsf{Nat}$}
\RightLabel{(TApp)}
\BinaryInfC{$\Gamma \vdash \mathsf{pred}\ x : \mathsf{Nat}$}
\RightLabel{(TApp)}
\BinaryInfC{$\Gamma \vdash \mathsf{ifz}\ y\ x\ (\mathsf{pred}\ x) : \mathsf{Nat}$}
\end{prooftree}
$$

(c) With $\Gamma = \{x : a,\ y : a \to a \to b\}$:

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{(TVar)}
\UnaryInfC{$\Gamma \vdash y : a \to a \to b$}
\AxiomC{}
\RightLabel{(TVar)}
\UnaryInfC{$\Gamma \vdash x : a$}
\RightLabel{(TApp)}
\BinaryInfC{$\Gamma \vdash yx : a \to b$}
\AxiomC{}
\RightLabel{(TVar)}
\UnaryInfC{$\Gamma \vdash x : a$}
\RightLabel{(TApp)}
\BinaryInfC{$\Gamma \vdash yxx : b$}
\RightLabel{(TAbs)}
\UnaryInfC{$x : a \vdash \lambda y.\, yxx : (a \to a \to b) \to b$}
\RightLabel{(TAbs)}
\UnaryInfC{$\vdash \lambda xy.\, yxx : a \to (a \to a \to b) \to b$}
\end{prooftree}
$$

(d) With $\Gamma = \{x : a \to b,\ y : b \to c,\ z : a\}$:

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{(Var)}
\UnaryInfC{$\Gamma \vdash y : b \to c$}
\AxiomC{}
\RightLabel{(Var)}
\UnaryInfC{$\Gamma \vdash x : a \to b$}
\AxiomC{}
\RightLabel{(Var)}
\UnaryInfC{$\Gamma \vdash z : a$}
\RightLabel{(App)}
\BinaryInfC{$\Gamma \vdash xz : b$}
\RightLabel{(App)}
\BinaryInfC{$\Gamma \vdash y(xz) : c$}
\RightLabel{(Abs)}
\UnaryInfC{$x : a \to b,\ y : b \to c \vdash \lambda z.\, y(xz) : a \to c$}
\RightLabel{(Abs)}
\UnaryInfC{$x : a \to b \vdash \lambda yz.\, y(xz) : (b \to c) \to a \to c$}
\RightLabel{(Abs)}
\UnaryInfC{$\vdash \lambda xyz.\, y(xz) : (a \to b) \to (b \to c) \to a \to c$}
\end{prooftree}
$$

** 2. Give terms M *in normal form* that satisfy each of the following (you are *not* required to justify them with a proof tree, but you may wish to so as to check your answer):

(a) $\vdash M : (a \to b) \to a \to b$
(b) $x : (a \to a) \to c \vdash M : c$
(c) $\vdash M : a \to b \to \mathsf{Nat}$

Solution

(a) e.g. $\lambda x.\, x$ or perhaps $\lambda xy.\, xy$
(b) e.g. $x\,(\lambda y.\, y)$
(c) e.g. $\lambda xy.\, \underline{2}$

** 3. Use inversion to prove that the following terms are not typable:

(a) $\underline{1}\,(\lambda x.x)$

(b) $\mathsf{pred}\,(\lambda x.x)$

(c) $\lambda xy.xy(yx)$

Solution

(a) Suppose $\underline{1}\,(\lambda x.x)$ is typable and we look for a contradiction. Then, by definition, there is a type A such that $\vdash \underline{1}\,(\lambda x.x) : A$. By inversion, there is a type B such that:

(i) $\vdash \underline{1} : B \to A$ and

(ii) $\vdash (\lambda x.x) : B$

By inversion on (i), we must have that there is a type C such that:

(a) $\vdash \mathsf{S} : C \to B \to A$ and

(b) $\vdash \underline{0} : C$

By inversion on (a), $\mathsf{S} : C \to B \to A$ is in $\mathbb{C}$, but this is impossible, because the only assignment to S in $\mathbb{C}$ is $\mathsf{Nat} \to \mathsf{Nat}$.

(b) Suppose $\mathsf{pred}\,(\lambda x.x)$ is typable and we look for a contradiction. Then, by definition, there is a type A and $\vdash \mathsf{pred}\,(\lambda x.x) : A$. By inversion, there is a type B and:

(i) $\vdash \mathsf{pred} : B \to A$

(ii) $\vdash \lambda x.x : B$

By inversion on (i), $\mathsf{pred} : B \to A$ must be in $\mathbb{C}$, from which, by definition of $\mathbb{C}$, we deduce that $B = A = \mathsf{Nat}$. This makes (ii) $\vdash \lambda x.x : \mathsf{Nat}$. By inversion on (ii), we have that there are types C_1 and C_2 such that $\mathsf{Nat} = C_1 \to C_2$, but this is impossible.

(c) Suppose that $\lambda xy.xy(yx)$ were typable. By definition of typability, there is some A such that $\vdash \lambda xy.xy(yx) : A$. By inversion twice, it must be that there are types B, C and D such that (1) $A = B \to C \to D$ and $x{:}B, y{:}C \vdash xy(yx) : D$. By inversion on this judgement, we have that there is some type E such that:

i $x{:}B, y{:}C \vdash xy : E \to D$

ii $x{:}B, y{:}C \vdash yx : E$

By inversion on the former, we have that there is some type F such that:

(a) $x{:}B, y{:}C \vdash x : F \to E \to D$

(b) $x{:}B, y{:}C \vdash y : F$

By inversion on these two judgments, we have that (2) $B = F \to E \to D$ and (3) $C = F$. By inversion on (ii) we get that there is a type G such that:

(A) $x{:}B, y{:}C \vdash y : G \to E$

(B) $x{:}B, y{:}C \vdash x : G$

By inversion on these two judgements, we get that (4) $C = G \to E$ and (5) $B = G$. Now, by combining equations (1)–(5) we obtain:

$$F = G \to E = B \to E = (F \to E \to D) \to E$$

but this is impossible because whatever type F is cannot include itself as a substring.

** 4. The following property is called *Weakening*:

For all Γ, Γ' and A: if $\Gamma \vdash M : A$ and $\Gamma \subseteq \Gamma'$ then $\Gamma' \vdash M : A$.

We can prove Weakening by induction on M.

Proof. The proof is by induction on M.

- When M is a variable x ... (a)

- When M is a constant c, let A be a type, Γ and Γ' be type environments such that $\Gamma \subseteq \Gamma'$ and suppose $\Gamma \vdash c : A$. By inversion, it follows that $c{:}A \in \mathbb{C}$. Therefore, the side condition is fulfilled to use (TCst) to also justify $\Gamma' \vdash c : A$ (this rule does not place any requirements on the environment).

- When M is an application PQ, assume the induction hypotheses:

  (IH1) For all Γ'' and Γ''' and A', if $\Gamma'' \subseteq \Gamma'''$ and $\Gamma'' \vdash P : A'$ then $\Gamma''' \vdash P : A'$.

  (IH2) For all Γ'' and Γ''' and A', if $\Gamma'' \subseteq \Gamma'''$ and $\Gamma'' \vdash Q : A'$ then $\Gamma''' \vdash Q : A'$.

  Let A be a type, Γ and Γ' be environments such that $\Gamma \subseteq \Gamma'$. Then suppose $\Gamma \vdash PQ : A$. By inversion, there must be a type B such that $\Gamma \vdash P : B \to A$ and $\Gamma \vdash Q : B$. It follows from (IH1) with $\Gamma'' := \Gamma$ and $\Gamma''' := \Gamma'$ and $A' := B \to A$ that $\Gamma' \vdash P : B \to A$. It follows from (IH2) with $\Gamma'' := \Gamma$, $\Gamma''' := \Gamma'$ and $A' := B$ that $\Gamma' \vdash Q : B$. Therefore, by (TApp), $\Gamma' \vdash PQ : A$.

- When M is an abstraction $\lambda x.P$ ... (b)

$\square$

Complete the remaining two cases.

Solution

(a) When M is a variable x, let A be a type, Γ and Γ' be type environments such that $\Gamma \subseteq \Gamma'$ and suppose $\Gamma \vdash x : A$. By inversion, it follows that $x : A \in \Gamma$. Since Γ' contains all the typings of Γ, also $x : A \in \Gamma'$. Hence, by (TVar), $\Gamma' \vdash x : A$.

(b) When M is an abstraction $\lambda x.\, P$, assume the induction hypothesis:

(IH) For all Γ'' and Γ''' and A', if $\Gamma'' \subseteq \Gamma'''$ and $\Gamma'' \vdash P : A'$ then $\Gamma''' \vdash P : A'$.

Let A be a type, Γ and Γ' be type environments such that $\Gamma \subseteq \Gamma'$. Then suppose $\Gamma \vdash \lambda x.\, P : A$. By the variable convention we can assume that x does not occur in Γ or Γ'. By inversion, it follows that there are types B and C such that $A = B \to C$ and $\Gamma, x{:}B \vdash P : C$. Then, it follows from the induction hypothesis with $\Gamma'' = \Gamma \cup \{x{:}B\}$ and $\Gamma''' = \Gamma' \cup \{x : B\}$ and $A' := C$, that $\Gamma', x{:}B \vdash P : C$. It follows by (TAbs) that, therefore, $\Gamma' \vdash \lambda x.\, P : B \to C$, as required.

*** 5. Find terms M and N such that:

(i) M is not typable

(ii) N is typable

(iii) $M \rhd N$

Solution

Let $M := (\lambda y.\, \underline{0})\ (\lambda x.\, x x)$ and $N := \underline{0}$. Clearly requirements (ii) and (iii) are satisfied. M is not typable because it has an untypable term as a subterm. It is possible to prove by induction that if M is typable then it cannot have an untypable subterm, but you can also "see" this if you consider that each rule of the type system has a conclusion of shape $\Gamma \vdash M : A$ and judgements as premises whose subjects are exactly the immediate subterms of M. Hence, every subterm of M will eventually be the conclusion of a rule in the derivation of $\Gamma \vdash M : A$. Hence, it must be that every closed subterm is typable.