UNIVERSITY OF BRISTOL

Winter 2024 Examination Period

SCHOOL OF COMPUTER SCIENCE

Third Year Examination for the Degrees
of
Bachelor of Science
Master of Engineering

Types and Lambda Calculus

TIME ALLOWED:
1 Hour

Answers to : Types and Lambda Calculus

Intended Learning Outcomes:

Credit will be given for partially correct answers. You may use any result from the course material, as long as it is labelled clearly. A reminder of key definitions is provided at the back of this section.

Q1. * (a) For each of the following terms, give its normal form.

i. $(\lambda xy.\, x)\ \underline{2}\ \underline{3}$

ii. $\mathsf{ifz}\ (\mathsf{pred}\ \underline{1})\ \underline{\mathsf{id}}\ \underline{\mathsf{const}}$

iii. $(\lambda x.\, x\ z\ \underline{5})\ (\lambda xy.\, y)$

iv. $(\lambda f.\, (\lambda x.\, f\ (x\ x))(\lambda x.\, f\ (x\ x)))\ (\lambda x.\, y)$

v. $\mathsf{ifz}\ \mathsf{Z}\ \underline{1}$

[10 marks]

Solution:

i. $\underline{2}$

ii. $\underline{\mathsf{id}}$

iii. $\underline{5}$

iv. y

v. $\mathsf{ifz}\ \mathsf{Z}\ \underline{1}$

* (b) For each of the following typing judgements, give a type derivation to justify it.

i. $\vdash \lambda x.\, \mathsf{pred}\ x : \mathsf{Nat} \to \mathsf{Nat}$

ii. $f : \mathsf{Nat} \to \mathsf{Nat}, g : \mathsf{Nat} \to \mathsf{Nat} \vdash \lambda x.\, g\ (f\ x) : \mathsf{Nat} \to \mathsf{Nat}$

iii. $\vdash \mathsf{fix}\ (\lambda x.\, x)\ \mathsf{S} : \mathsf{Nat} \to \mathsf{Nat}$

[10 marks]

Solution:

i.

$$\dfrac{\dfrac{\dfrac{}{x : \mathsf{Nat} \vdash \mathsf{pred} : \mathsf{Nat} \to \mathsf{Nat}} \qquad \dfrac{}{x : \mathsf{Nat} \vdash x : \mathsf{Nat}}}{x : \mathsf{Nat} \vdash \mathsf{pred}\ x : \mathsf{Nat}}}{\vdash \lambda x.\, \mathsf{pred}\ x : \mathsf{Nat} \to \mathsf{Nat}}$$

ii. With $\Gamma = \{f : \mathsf{Nat} \to \mathsf{Nat}, g : \mathsf{Nat} \to \mathsf{Nat}\}$:

$$\dfrac{\dfrac{\dfrac{}{\Gamma, x : \mathsf{Nat} \vdash f : \mathsf{Nat} \to \mathsf{Nat}} \qquad \dfrac{\dfrac{}{\Gamma, x : \mathsf{Nat} \vdash g : \mathsf{Nat} \to \mathsf{Nat}} \qquad \dfrac{}{\Gamma, x : \mathsf{Nat} \vdash x : \mathsf{Nat}}}{\Gamma, x : \mathsf{Nat} \vdash g\ x : \mathsf{Nat}}}{\Gamma, x : \mathsf{Nat} \vdash f\ (g\ x) : \mathsf{Nat}}}{\Gamma \vdash \lambda x.\, f\ (g\ x) : \mathsf{Nat} \to \mathsf{Nat}}$$

iii. With $A = \mathsf{Nat} \to \mathsf{Nat}$

$$\dfrac{\dfrac{\dfrac{}{\vdash \mathsf{fix} : (A \to A) \to A} \quad \dfrac{\dfrac{}{x : A \vdash x : A}}{\vdash \lambda x.\, x : A \to A}}{\vdash \mathsf{fix}\ (\lambda x.\, x) : A} \qquad \dfrac{}{\vdash \mathsf{S} : A}}{\vdash \mathsf{fix}\ (\lambda x.\, x)\ \mathsf{S} : A}$$

**(c) One of the following two types is inhabited by a closed, *pure* term and the other is not. Identify which is inhabited, and justify your answer by giving a closed, pure term that inhabits the type.

i. $a \to b$

ii. $(a \to b) \to ((c \to b) \to d) \to a \to d$

[5 marks]

Solution: Type (ii) has inhabitant $\lambda xyz.\, y(\lambda y.\, xz)$.

**(d) Let x be a variable and N a term. Prove the following by induction on M:

for all terms M, if $x \notin \mathsf{FV}(M)$ then $M[N/x] = M$.

[10 marks]

Solution: The proof is by induction on M.

- When M is a variable y, proceed as follows. Suppose (i) $x \notin \mathsf{FV}(M)$. Then there are two cases:
  - If $x = y$, we have $\mathsf{FV}(M) = \{x\}$, but this contradicts (i) and so the conclusion follows vacuously.
  - Otherwise, by definition, $M[N/x] = y[N/x] = y = M$.
- When M is a constant c, we suppose $x \notin \mathsf{FV}(M)$ and, by definition, $M[N/x] = c[N/x] = c = M$.
- When M is an application PQ, we assume the induction hypotheses:

  IH1 if $x \notin \mathsf{FV}(P)$ then $P[N/x] = P$

  IH2 if $x \notin \mathsf{FV}(Q)$ then $Q[N/x] = Q$

  Suppose (i) $x \notin \mathsf{FV}(M)$. Then, by definition, $x \notin \mathsf{FV}(P)$ and $x \notin \mathsf{FV}(Q)$. Hence, it follows from (IH1) and (IH2) that $P[N/x] = P$ and $Q[N/x] = Q$. By definition, therefore, $M[N/x] = (PQ)[N/x] = P[N/x]Q[N/x] = PQ = M$.
- When M is an abstraction $\lambda y.\, P$, we assume the induction hypothesis:

1. if $x \notin \mathsf{FV}(P)$ then $P[N/x] = P$

Suppose $x \notin \mathsf{FV}(M)$. By definition, either $x = y$ or $x \notin \mathsf{FV}(P)$, but, by the variable convention, we may assume $x \neq y$. Hence, we can apply the induction hypothesis to obtain $P[N/x] = P$ and it follows by definition that $M[N/x] = (\lambda y.\, P)[N/x] = \lambda y.\, P[N/x] = \lambda y.\, P = M$.

*** (e) Fix a one-hole context $C[]$ containing no free variables, and a closed term M. Give a detailed proof that, if $C[M]$ is typable, then $C[\underline{\mathsf{div}}]$ is typable.

[10 marks]

Solution: First observe that $\vdash \underline{\mathsf{div}} : A$ for any type A and any environment Γ:

$$\dfrac{\dfrac{}{\Gamma \vdash \mathsf{fix} : (A \to A) \to A} \qquad \dfrac{\dfrac{}{\Gamma, x : A \vdash x : A}}{\Gamma \vdash \lambda x.\, x : A \to A}}{\Gamma \vdash \mathsf{fix}\ (\lambda x.\, x) : A}$$

We prove that, for all A and all Γ, $\Gamma \vdash C[M] : A$ implies $\Gamma \vdash C[\underline{\mathsf{div}}] : A$, by induction on $C[]$.

- When C is just a hole $[]$, assume $\Gamma \vdash C[M] : A$. By definition $C[\underline{\mathsf{div}}] = \underline{\mathsf{div}}$, for which we have $\Gamma \vdash \underline{\mathsf{div}} : A$, by the above.

- When C is of shape $P\ D[]$, we assume the induction hypothesis. Let A be a type and Γ an environment. Assume that $\Gamma \vdash C[M] : A$. Then, by inversion, there is some type B such that $\Gamma \vdash P : B \to A$ and $\Gamma \vdash D[M] : B$. Then it follows from the induction hypothesis that $\Gamma \vdash D[\mathsf{fix}] : B$ and so the result follows by the definition of the type system.

- When C is of shape $D[]\ Q$, we assume the induction hypothesis. Let A be a type and Γ an environment. Suppose $\Gamma \vdash C[M] : A$. Then, by inversion, there is some type B such that $\Gamma \vdash D[M] : B \to A$ and $\Gamma \vdash Q : B$. It follows from the induction hypothesis that $\Gamma \vdash D[\underline{\mathsf{div}}] : B \to A$ and so the result follows by definition.

- When C is of shape $\lambda x.\, D[]$, we assume the induction hypothsis. Let A be a type and Γ an environment. Suppose $\Gamma \vdash C[M] : A$. Then, by inversion, A has shape $B \to C$ and $\Gamma, x : B \vdash D[M] : C$. Therefore, it follows from the induction hypothesis that $\Gamma, x : B \vdash D[\underline{\mathsf{div}}] : C$ and the result follows by definition.

*** (f) Suppose we add the following additional redex-contraction pair to the definition of redexes:

$$\lambda x.\, M\ x \ /\ M \qquad \text{when } x \notin \mathsf{FV}(M)$$

Prove that the induced notion of reduction is not confluent.

[5 marks]

Solution: Suppose the augmented notion of reduction remains confluent. A counterexample is $\lambda x.\, \mathsf{fix}\, x$. Using the augmented notion of reduction, we have $\lambda x.\, \mathsf{fix}\, x \rhd^* \mathsf{fix}$ and $\lambda x.\, \mathsf{fix}\ x \rhd^* \lambda x.\, x\ (\mathsf{fix}\ x)$. However, fix is a normal form, so it follows from confluence that $\lambda x.\, x\ (\mathsf{fix}\ x) \rhd^* \mathsf{fix}$. This is impossible, because for every $n \in \mathbb{N}$, the only term P that satisfies $\lambda x.\, x\ (\mathsf{fix}\ x) \rhd^n P$, is $\lambda x.\, x^{n+1}\ (\mathsf{fix}\ x)$.

Key Definitions for Types and Lambda Calculus

Terms

$$
\begin{array}{rrcl}
\text{(Terms)} & M, N & ::= & x \mid c \mid (\lambda x.\, M) \mid (M N) \\
\text{(Constants)} & c & ::= & \mathsf{fix} \mid \mathsf{Z} \mid \mathsf{S} \mid \mathsf{pred} \mid \mathsf{ifz}
\end{array}
$$

A term is said to be *pure* just if it contains no constants. Abbreviations:

$$
\begin{array}{rcl}
\underline{n} & = & \mathsf{S}^n\ \mathsf{Z} \\
\underline{\mathsf{id}} & = & \lambda x.\, x \\
\underline{\mathsf{const}} & = & \lambda x y.\, x \\
\underline{\mathsf{sub}} & = & \lambda x y z.\, x z (y z) \\
\underline{\mathsf{div}} & = & \mathsf{fix}\ \underline{\mathsf{id}}
\end{array}
$$

Free Variables

$$
\begin{array}{rcl}
\mathsf{FV}(x) & = & \{x\} \\
\mathsf{FV}(c) & = & \emptyset \\
\mathsf{FV}(PQ) & = & \mathsf{FV}(P) \cup \mathsf{FV}(Q) \\
\mathsf{FV}(\lambda x.\, N) & = & \mathsf{FV}(N) \setminus \{x\}
\end{array}
$$

Substitution

$$
\begin{array}{rcll}
c[N/x] & = & c & \\
y[N/x] & = & y & \text{if } x \neq y \\
y[N/x] & = & N & \text{if } x = y \\
(PQ)[N/x] & = & P[N/x]Q[N/x] & \\
(\lambda y.\, P)[N/x] & = & \lambda y.\, P & \text{if } y = x \\
(\lambda y.\, P)[N/x] & = & \lambda y.\, P[N/x] & \text{if } y \neq x \text{ and } y \notin \mathsf{FV}(N)
\end{array}
$$

Redexes

$$
\begin{array}{r}
\mathsf{pred}\ \mathsf{Z} \ / \ \mathsf{Z} \\
\mathsf{pred}\ (\mathsf{S}\ N) \ / \ N \\
\mathsf{ifz}\ \mathsf{Z}\ N\ P \ / \ N \\
\mathsf{ifz}\ (\mathsf{S}\ M)\ N\ P \ / \ P \\
(\lambda x.\, M)\ N \ / \ M[N/x] \\
\mathsf{fix}\ M \ / \ M\ (\mathsf{fix}\ M)
\end{array}
$$

One Step

$$C[] ::= [] \mid M\ C[] \mid C[]\ N \mid \lambda x.\, C[]$$

Define $M \rhd N$ just if there is a context $C[]$ and a redex/contraction pair P / Q such that $M = C[P]$ and $N = C[Q]$.

- If $M \rhd^* N$ then the term N is said to be a **reduct** of M.
- If $M \rhd^+ N$ then the term N is said to be a **proper reduct** of M.
- A term M without proper reduct is a **normal form**.
- A term M that can reduce to normal form **has a normal form** or is **normalisable**.
- A term M that has no infinite reduction sequences is said to be **strongly normalisable**.

Reduction and Conversion

- $P \rhd^0 Q$ just if $P = Q$.
- $P \rhd^{k+1} Q$ just if there is some U such that $P \rhd^k U$ and $U \rhd Q$.

Define $M \rhd^* N$ just if there is some n such that $M \rhd^n N$.

We write $M \approx N$ just if there is a term P such that $M \rhd^* P$ and $N \rhd^* P$.

Type Assignment

$$(\text{Types}) \quad A, B \quad ::= \mathsf{Nat} \mid a \mid (A \to B)$$

Let $\mathbb{C}$ be the following collection of type assignments:

$$\begin{aligned} &\{\mathsf{Z} : \mathsf{Nat}\} \cup \{\mathsf{S} : \mathsf{Nat} \to \mathsf{Nat}\} \cup \{\mathsf{pred} : \mathsf{Nat} \to \mathsf{Nat}\} \\ &\qquad \cup \{\mathsf{ifz} : \mathsf{Nat} \to A \to A \to A \mid A \in \mathbb{T}\} \\ &\qquad \cup \{\mathsf{fix} : (A \to A) \to A \mid A \in \mathbb{T}\} \end{aligned}$$

The typing rules are:

$$x{:}A \in \Gamma \, \frac{}{\Gamma \vdash x : A} \, (\mathrm{TVar}) \qquad c{:}A \in \mathbb{C} \, \frac{}{\Gamma \vdash c : A} \, (\mathrm{TCst})$$

$$\frac{\Gamma \vdash M : B \to A \quad \Gamma \vdash N : B}{\Gamma \vdash MN : A} \, (\mathrm{TApp}) \qquad x \notin \mathsf{dom}\, \Gamma \, \frac{\Gamma \cup \{x : B\} \vdash M : A}{\Gamma \vdash \lambda x.\, M : B \to A} \, (\mathrm{TAbs})$$

We say that a closed term M is **typable** just if there is some type A such that $\vdash M : A$ is derivable in the type system. If $\vdash M : A$, then M is said to be an **inhabitant** of A. The **pure-term inhabitation problem**, is the problem of, given a type A, determining if there a closed, *pure* term M such that $\vdash M : A$.