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

This paper contains *one* question. All answers will be used for assessment. The maximum for this section is *50 marks*.

Other Instructions:

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 paper.

TURN OVER ONLY WHEN TOLD TO START WRITING

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

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. ifz Z $\underline{1}$

[10 marks]

* (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]

** (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]

** (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]

*** (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 $C[\underline{\text{div}}]$ is typable.

[10 marks]

*** (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]

Key Definitions for Types and Lambda Calculus

Terms

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

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

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

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

- If $M \ \triangleright^*\ N$ then the term N is said to be a **reduct** of M.
- If $M \ \triangleright^+\ N$ 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

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

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

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

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:

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

The typing rules are:

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

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

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 $\vdash M : A$.