

TYPES AND λ -CALCULUS

Problem Sheet 3

Recall that a *closed* term has no free variables.

- * 1. For each of the following specifications, give an example of a *closed* term N in *normal form* that satisfies it (i.e do some reduction):
 - (a) $\underline{\text{id}} \ \underline{\text{id}} \triangleright^* N$
 - (b) $\underline{\text{sub}} \ \underline{\text{const}} \ \underline{\text{const}} \triangleright^* N$
 - (c) $\text{fix} (\lambda xy. y) \triangleright^* N$
 - (d) $(\lambda xy. yx) (\underline{\text{const}} \ \underline{\text{const}}) (\lambda x. xx) \triangleright^* N$

- * 2. Draw the reduction graph of the term $(\lambda xy. yy) ((\lambda z. zz) (\lambda z. zz)) (\lambda x. x)$. (This graph will have 4 vertices). What I mean by this is to draw a directed graph where:
 - The nodes are all N s.t. $(\lambda xy. yy) ((\lambda z. zz) (\lambda z. zz)) (\lambda x. x) \triangleright^* N$
 - There is an edge from node M to node N iff $M \triangleright N$

- ** 3. Give an example of a *closed* term M for each of the following properties:
 - (a) M is in normal form.
 - (b) M has exactly one reduct.
 - (c) M contains strictly fewer redexes than one of its reducts (here we mean “fewer in number”, the redexes may be quite different).
 - (d) A reduct of M contains a redex that did not occur anywhere in M .

- * 4. For each of the following reduction steps $M \triangleright N$, identify the redex P , the contraction Q , and the context $C[\]$ in which the contraction happens, i.e. such that $M = C[P]$ and $N = C[Q]$.

- (a) $\lambda x. \text{pred } (\text{pred } \underline{2}) \triangleright \lambda x. \text{pred } \underline{1}$
- (b) $\underline{\text{id}} (\underline{\text{const}} \underline{\text{div}} \underline{0}) \triangleright \underline{\text{id}} (\underline{\text{const}} (\underline{\text{id}} \underline{\text{div}}) \underline{0})$
- (c) $\underline{\text{const}} (\underline{\text{id}} \underline{\text{id}}) (\text{S } x) \triangleright (\lambda y. \underline{\text{id}} \underline{\text{id}}) (\text{S } x)$

** 5. Prove the following statement:

For all M, N and $C[\]$: if $M \triangleright N$ then $C[M] \triangleright C[N]$.

Note that “if $M \triangleright N$ then $C[M] \triangleright C[N]$ ” is subtly different from the definition of \triangleright which says that $C[P] \triangleright C[Q]$ whenever P is a redex and Q the contraction. Here, M and N can be any terms.

You do *not* need to use induction to prove it. You will need to work closely with the definition of \triangleright : on the one hand you will assume $M \triangleright N$ and want to know what you get out of it and, on the other hand, you will want to show $C[M] \triangleright C[N]$ and thus need to know what evidence is required to put into it.

Look again at the definition of \triangleright using contexts. In the definition, “just if” means the same as “iff”, so the definition of \triangleright can be seen as a pair of implications: one direction tells you what follows from $M \triangleright N$ when you have it as an assumption (forwards reasoning) and the other tells you what you need in order to deduce $M \triangleright N$ (backwards reasoning).

** 6.

(a) Complete the following proof by filling in (a):

For all $P, C[\]$, for all $n \in \mathbb{N}$: for all Q , if $P \triangleright^n Q$ then $C[P] \triangleright^n C[Q]$.

Proof. Let P be a term and $C[\]$ a context. We show that, for all $n \in \mathbb{N}$, for all Q , $P \triangleright^n Q$ implies $C[P] \triangleright^n C[Q]$ by induction on n :

- When $n = 0$, let Q be a term and suppose $P \triangleright^0 Q$. Then, by definition, $P = Q$ and hence $C[P] = C[Q]$. By definition, therefore $C[P] \triangleright^0 C[Q]$.
- When n is of shape $k + 1$, we can assume the induction hypothesis:
(IH) for all Q , $P \triangleright^k Q$ implies $C[P] \triangleright^k C[Q]$.

... (a) ...

□

(b) Deduce that (i.e. give a short proof of): For all $P, Q, C[\]$: if $P \triangleright^* Q$ then $C[P] \triangleright^* C[Q]$.

*** 7. Show that there is no term P that satisfies: for all M and N , $P(MN) \triangleright^* N$.
In other words, prove that we cannot write a PCF program that extracts the argument of an application.