

TYPES AND λ -CALCULUS

Problem Sheet 3

This week, questions 2, 8 and 9 will be marked.

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

- * 1. Perform one step of reduction for each of the following terms:
- (a) const pred pred
 - (b) sub const
 - (c) $(\lambda x. x x)(\lambda x. x x)$
 - (d) const (pred pred)

Solution _____

- (a) $(\lambda y. \text{pred}) \text{ pred}$
- (b) $\lambda y z. \text{const } z (y z)$
- (c) $(\lambda x. x x)(\lambda x. x x)$
- (d) $\lambda y. \text{pred pred}$ or const wrong

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

Solution _____

- (a) $P = \text{pred } \underline{2}, Q = \underline{1}, C[\] = \lambda x. \text{pred } [\]$

- (b) $P = \underline{\text{div}}, Q = \underline{\text{id}} \underline{\text{div}}, C[] = \underline{\text{id}} (\underline{\text{const}} [] \underline{0})$
(c) $P = \underline{\text{const}} (\underline{\text{id}} \underline{\text{id}}), Q = \lambda y. \underline{\text{id}} \underline{\text{id}}, C[] = [] (S \ x)$

* 3. Let us define the Booleans as follows:

$$\underline{\text{false}} = \underline{0}$$

$$\underline{\text{true}} = \underline{1}$$

Define Boolean conjunction as a term and, disjunction as a term or and negation as a term not.

Solution _____

Define:

$$\underline{\text{not}} = \lambda x. \text{ifz } x \ \underline{\text{true}} \ \underline{\text{false}}$$

$$\underline{\text{and}} = \lambda x y. \text{ifz } x \ \underline{\text{false}} \ (\text{ifz } y \ \underline{\text{false}} \ \underline{\text{true}})$$

$$\underline{\text{or}} = \lambda x y. \underline{\text{not}} (\underline{\text{and}} \ x \ y)$$

* 4. Define terms curry and uncurry with the following behaviour:

$$\underline{\text{curry}} \ M \ N \ P \triangleright^* M (N, P)$$

$$\underline{\text{uncurry}} \ M (N, P) \triangleright^* M \ N \ P$$

Solution _____

$$\underline{\text{curry}} := \lambda f x y. f \ (x, y)$$

$$\underline{\text{uncurry}} := \lambda f p. f \ (\underline{\text{proj}}_1^2 p) (\underline{\text{proj}}_2^2 p)$$

* 5. 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) $\underline{id} \ \underline{id} \triangleright^* N$
- (b) $\underline{sub} \ \underline{const} \ \underline{const} \triangleright^* N$
- (c) $\text{fix} (\lambda xy. y) \triangleright^* N$
- (d) $(\lambda xy. yx) (\underline{const} \ \underline{const}) (\lambda x. xx) \triangleright^* N$

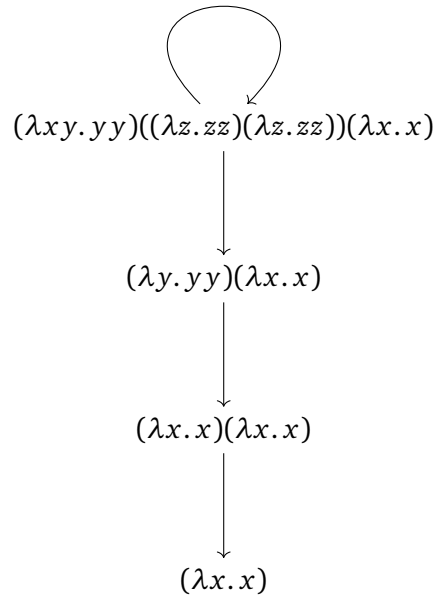
Solution _____

- (a) \underline{id}
- (b) \underline{id}
- (c) \underline{id}
- (d) \underline{const}

- * 6. 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 $M \triangleright N$

Solution _____



- ** 7. 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 .

Solution

There are lots of possible answers, here is one set:

- (a) $\underline{\text{id}}$
- (b) $\text{pred } \underline{2}$
- (c) $(\lambda x. xxx) (\underline{\text{id}} \underline{1})$
- (d) $(\lambda xy. xy) \underline{\text{id}}$

- ** 8. 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).

Solution

Let M, N be terms and let $C[\]$ be an evaluation context. Suppose $M \triangleright N$. Then it follows from the definition of \triangleright that there must exist an evaluation context $C'[\]$ and a redex/contraction pair P, Q such that $M = C'[P]$ and $N = C'[Q]$. So, $M \triangleright N$ is really:

$$C'[P] \triangleright C'[Q]$$

Now, by these equations, $C[M] = C[C'[P]]$ and $C[N] = C[C'[Q]]$ (*). Our aim is to prove that $C[M] \triangleright C[N]$ so, by definition of \triangleright , we have to find a context $C''[\]$ and a redex/contraction pair P', Q' such that $C[M] = C''[P']$ and $C''[Q'] = C[N]$. By (*) we can take $C''[\] = C[C'[\]]$ and $P' = P$ and $Q' = Q$.

** 9.

(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 $C[P] \triangleright^* C[Q]$.

Solution

- (a) Now, let Q be a term and suppose $P \triangleright^{k+1} Q$. Then, by definition, there is some U such that (i) $P \triangleright^k U$ and (ii) $U \triangleright Q$. We can use (IH) on (i) to obtain $C[P] \triangleright^k C[U]$ (*). We can use the statement of the previous question on (ii) to obtain $C[U] \triangleright C[Q]$ (**). By definition of \triangleright^{k+1} , we can use (*) and (**) to obtain $C[P] \triangleright^{k+1} C[Q]$.
- (b) Let P and Q be terms and $C[]$ a context. Suppose $P \triangleright^* Q$. By definition, there is some n such that $P \triangleright^n Q$. It follows from the previous result that, therefore, $C[P] \triangleright^n C[Q]$. Hence, by definition of \triangleright^* , $C[P] \triangleright^* C[Q]$.

- ** 10. 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.

Solution

Suppose such a P exists and we look to obtain a contradiction. Then we know that this P satisfies $\forall MN. P(MN) \triangleright^* N$, let's call this (*). Then, instantiating M by const (id id) and N by const in (*), we have:

$$P(\text{const}(\text{id id}) \text{const}) \triangleright^* \text{const}$$

However, by simply performing the reductions, we also have

$P(\text{const}(\text{id id}) \text{const}) \triangleright^* P(\text{id id})$. By instantiating (*) with $M = \text{id}$ and $N = \text{id}$ we get $P(\text{id id}) \triangleright^* \text{id}$. So, overall, we also have:

$$P(\text{const}(\text{id id}) \text{const}) \triangleright^* \text{id}$$

However, with the other reduction sequence inset above, this contradicts the unique normal forms theorem, since id and const are distinct normal forms of $P(\text{const}(\text{id id}) \text{const})$, so we have our contradiction.