

Problem 1.

Given

$0: \lambda f. \lambda x. x$
 $\text{succ}: \lambda n. \lambda f. \lambda x. (f ((n f) x))$
 n : if n is a natural number then its semantics is the result of n applications of succ on 0 .
 $\text{true}: \lambda x. \lambda y. x$
 $\text{false}: \lambda x. \lambda y. y$
 $\text{second}: \lambda x. \lambda y. \lambda z. y$
 $g: \lambda n. ((n \text{ second}) \text{ false})$

What is the result of

- (a) $(g \ 0)$ when n is 0 .
- (b) $(g \ n)$ when n results from some application succ on 0 .
- (c) What mathematical/logical operation is computed by g .

Solution**Part (a)**

$(g \ 0)$
 $(g \ \lambda f. \lambda x. x)$
 $(\lambda n. ((n \text{ second}) \text{ false}) \ \lambda f. \lambda x. x)$
 $((\lambda f. \lambda x. x \text{ second}) \text{ false})$
 $(\lambda x. x \text{ false})$
 false

Part (b)

First consider what is happening when n applications of succ are taken on 0 .

Consider $n = 1$ application of succ on 0 .

$(\text{succ } 0)$
 $(\lambda n. \lambda f. \lambda x. (f ((n f) x)) \ \lambda f. \lambda x. x)$
 $\lambda f. \lambda x. (f ((\lambda f. \lambda x. x f) x))$
 $\lambda f. \lambda x. (f (\lambda x. x x))$
 $\lambda f. \lambda x. (f x)$

Now consider $n = 2$ applications of succ on 0 .

$(\text{succ } (\text{succ } 0))$
 $(\text{succ } \lambda f. \lambda x. (f x))$
 $(\lambda n. \lambda f. \lambda x. (f ((n f) x)) \ \lambda f. \lambda x. (f x))$
 $\lambda f. \lambda x. (f ((\lambda f. \lambda x. (f x) f) x))$
 $\lambda f. \lambda x. (f (\lambda x. (f x) x))$
 $\lambda f. \lambda x. (f (f x))$

As discussed in class, this pattern will continue.

That is, n applications of `succ` on 0 can be written as the following.

$\lambda f. \lambda x. (f^1 (\dots (f^n x)))$

Note the superscript denotes the occurrence index for the lambda function f .

Now consider $(g\ n)$

$(g\ n)$

$(g\ (\text{succ}_1 (\dots (\text{succ}_n\ 0))))$

$(\lambda n. ((n\ \text{second})\ \text{false})\ (\text{succ}_1 (\dots (\text{succ}_n\ 0))))$

$((\text{succ}_1 (\dots (\text{succ}_n\ 0)))\ \text{second})\ \text{false})$

$((\lambda f. \lambda x. (f^1 (\dots (f^n\ x))))\ \text{second})\ \text{false})$

$((\lambda f. \lambda x. (f^1 (\dots (f^n\ x))))\ \lambda x. \lambda y. \lambda z. y)\ \text{false})$

As these are lambda expressions, subscripts denote both the number of occurrences as well as the uniqueness of each lambda variable.

$(\lambda x. (\lambda x_1. \lambda y_1. \lambda z_1. y_1 (\dots (\lambda x_n. \lambda y_n. \lambda z_n. y_n\ x))))\ \text{false})$

$(\lambda x_1. \lambda y_1. \lambda z_1. y_1 (\dots (\lambda x_n. \lambda y_n. \lambda z_n. y_n\ \text{false})))$

$(\lambda x_1. \lambda y_1. \lambda z_1. y_1 (\dots (\lambda x_{n-1}. \lambda y_{n-1}. \lambda z_{n-1}. y_{n-1}\ (\lambda x_n. \lambda y_n. \lambda z_n. y_n\ \text{false}))))$

$(\lambda x_1. \lambda y_1. \lambda z_1. y_1 (\dots (\lambda x_{n-1}. \lambda y_{n-1}. \lambda z_{n-1}. y_{n-1}\ \lambda y_n. \lambda z_n. y_n)))$

Note that $\lambda y. \lambda z. y$ is defined as 'true'.

$(\lambda x_1. \lambda y_1. \lambda z_1. y_1 (\dots (\lambda x_{n-1}. \lambda y_{n-1}. \lambda z_{n-1}. y_{n-1}\ \text{true})))$

If $n = 1$ we would be done here and have a result of 'true'

(as the $1 \dots n-1$ terms would not exist).

If $n > 1$, we apply another beta reduction

$(\lambda x_1. \lambda y_1. \lambda z_1. y_1 (\dots (\lambda x_{n-2}. \lambda y_{n-2}. \lambda z_{n-2}. y_{n-2}\ \lambda y_{n-1}. \lambda z_{n-1}. y_{n-1})))$

$(\lambda x_1. \lambda y_1. \lambda z_1. y_1 (\dots (\lambda x_{n-2}. \lambda y_{n-2}. \lambda z_{n-2}. y_{n-2}\ \text{true})))$

$(\lambda x_1. \lambda y_1. \lambda z_1. y_1 (\dots \lambda y_{n-2}. \lambda z_{n-2}. y_{n-2}))$

$(\lambda x_1. \lambda y_1. \lambda z_1. y_1 (\dots (\lambda x_{n-3}. \lambda y_{n-3}. \lambda z_{n-3}. y_{n-3}\ \text{true})))$

Again note that if $n = 2$, we would be done here and have a result of 'true'.

If $n > 3$ this trend would continue recursively until resolving to true.

Part (c)

As g resolves to false when $n = 0$, and true otherwise

it then computes $n > 0$ or $n \neq 0$ on the domain of natural numbers.

Problem 2.

Consider the following λ -expression:

$$Y : \lambda t. (\lambda x. (t (x x)) \lambda x. (t (x x)))$$

Prove/disprove that $(Y t)$ after application of several β -reductions results in $(t (Y t))$.

Solution

Setup

We are given $(Y t)$, apply the definition of Y to this given lambda expression.

$$(\lambda t. (\lambda x. (t (x x)) \lambda x. (t (x x))) t)$$

Denote $Q: \lambda x. (t (x x))$

Note that $(Q Q)$ after a single β -reduction results in $(t (Q Q))$

Thus we are starting with $(t (Q Q))$.

Claim

let $n \in \mathbb{N}$

If n β -reductions are taken of $(t (Q Q))$ denoted R

then $R \neq (t (Y t))$

Proof by Induction on n

Base Case ($n = 0$)

This is true as $Q \neq Y \wedge Q \neq t$

Thus $(Q Q) \neq (Y t)$

Therefore $(t (Q Q)) \neq (t (Y t))$

Induction Hypothesis

Assume $(t_0 (t_1 (\dots (t_n (Q Q))))) \neq (t (Y t))$

Induction Step

We have $(t_0 (t_1 (\dots (t_n (Q Q))))) \neq (t (Y t))$

Apply a single β -reduction to the left hand side producing

$$(t_0 (t_1 (\dots (t_n (t_{n+1} (Q Q)))))$$

Note that $(t_0 (t_1 (\dots (t_n (t_{n+1} (X))))) \neq (t (X))$ for any X

as $(t_n (\dots))$ is not a lambda expression that can be β -reduced.

Therefore $(t_0 (t_1 (\dots (t_n (t_{n+1} (Q Q))))) \neq (t (Y t)) \quad \square$