CS-350, Nikhil Mehta, 190549

Q-1 : Idea: We will try to construct an example where 1st term take 3 steps and the other way takes 2 steps to reach to same term. As they do not reach to same term in 2 steps, it will not hold diamond property.

Example Take a term $(\lambda x : xx)((\lambda y. y) z)$

Tree :

$(\lambda x. xx)((\lambda y. y) z) \to t_1$

A

B

$((\lambda y. y) z)((\lambda y. y) z) \to t_2$

$(\lambda x. xx) z \to t_3$

$z(\lambda y. y) z$ or $((\lambda y. y) z) z \to t_4$

$z z \to t_4'$

We can see $t_2$ & $t_3$ does not reduce to same term, thus diamond property do not hold.

Question-2

In general: $M[x:=N][y:=P] \neq M[y:=P][x:=N]$

This can be shown if we find two examples s.t. in one associative property holds and in other it does not, proving that is no guarantee if it holds or not.

Example-1 : Associative substitution

$$M = (\lambda x.x)(\lambda y.y)$$
$$N = \lambda u.xu$$
$$P = \lambda v.yv$$

We can say that their are no free variable for $x$ and $y$, thus
$$M[x:=N][y:=P] = M[y:=P][x:=N]$$

Example-2 : Non-Associative substitution

Let us take an example in which both $x$ & $y$ are free variables in one term and are bound in second term

Let us take
$$M = \lambda z.xy(\lambda x.x)$$

Their will be no changes in $\lambda x.x$ term as it has $x$-bound.

We take $N$ which has $y$ in its substitution.

Let us take $N = (\lambda x.x)y$

We ~~substitute y~~ with new variable

$p =$

We substitute p such that it has ~~y~~ x as free variable.

$p = \cancel{(\lambda y. y)} x \; (\lambda y.y)x$

~~Ref~~

LHS

- $M[x:=N] = \lambda z.(\lambda x.x)yy\;(\lambda x.x)$

$M[x:=N][y:=M] = \cancel{\lambda z.(\lambda x.x)(\lambda y.y)}x$

$\lambda z\; xx \;(\lambda x.x)$

RHS

$M[y:=N] = \lambda z. x(\lambda y.y)x \;\lambda x.x$

$M[y:=N][x:=N] = \lambda z \; y y \; \lambda x.x$

Idea was to get LHS & RHS not equal through an example.

~~Sec~~

Camlin

Question : 3

We have to prove that :

$$M[x:=N][y:=P] \equiv M[y:=P][x:=N'] \text{ where } N' = N[y:=P]$$

given $x \neq y$ & $x \notin FV(P)$

To prove the above, we will try to use induction on the $\boxed{\text{structure}}$ of M, by making cases

- For the induction, we take the base case :

Base Case : M is a variable

3 Possibilities

Case-1 $M \equiv x$

$LHS = N$

$RHS = N' = N[y:=P] = N$

$LHS = RHS$

Case-2 $M \equiv y$ (Similar above

$LHS = P$

$RHS = P$

$LHS = RHS$

Case-3

$M \neq x, M \neq y$.

No substitution to make,

so $LHS = RHS$

Camlin

Induction step: On structure of M

C-1 When M is of the form $M_1 M_2$ where $M_1$ and $M_2$ are Lambda terms.

$$M[x:=N][y:=P] \equiv (M_1[x:=N][y:=P])(M_2[x:=N][y:=P])$$

$\therefore$ Using distribution property

Now, by induction hypothesis

$$= (M_1[y:=P][x:=N'])(M_2[y:=P][x:=N'])$$

$$= M[y:=P][x:=N'].$$

C-2 When M is of the form $\lambda w.M'$, assuming $w \neq y$ & $w \notin FV(P)$ or $FV(N)$ & $M'$ is a $\lambda$-term with smaller-structure than M.

By Induction Hypothesis.

$$M'[x:=N][y:=P] = M'[y:=P][x:=N']$$

So $$M[x:=N][y:=P] = \lambda w.\ M'[y:=P][x:=N']$$

$$= (\lambda w M')([y:=P][x:=N'])$$

$$= M[y:=P][x:=N']$$

Hence - Proved

Camlin

Question - 4 :

After taking some examples, we will observe that it seems to represent 'PRED'.

Eg. let $M = \lambda n.n\,(\lambda pz.z((succ\,(p\,true))(p\,true)))\,(\lambda z.z\,zero\,zero)\,false$

M is the given expression.

Let us also assume $M_1$ & $M_2$ as:

$$M_1 = \lambda pz.z(succ\,(p\,true))(p\,true)$$

$$M_2 = \lambda z.\,z\,zero\,zero$$

Now M becomes:

$$M = \lambda n.n\,M_1 M_2\,false$$

Example-1 M zero

$\cancel{M\,zero = zero\,M_1 M_2\,false}$

$\cancel{= M_2\,fa}$

$$M\,zero = zero\,M_1\,M_2\,false$$

$$= (\lambda mw.w)\,k_1\,k_2\,false$$

$$= k_2\,false$$

$$= false\,zero\,zero$$

$$= \underline{zero}$$

Example-2 M one

$$M\,one = (\lambda n.n\ M_1 M_2\ false)\ one = one\ M_1 M_2\ false$$

$$= K_1 K_2\ false$$

$$= (\lambda p z.z(succ(p\ true))(p\ true))\ K_2\ false$$

$$= (\lambda z.z\ one\ zero)\ false$$

$$= zero$$

$$pred(one) = zero$$

Example-3 M two =

$$M\,two = two\ M_1 M_2\ false$$

$$M\,two = K_1(K_1 K_2)\ false$$

$$M\,two = K_1(\lambda p z.z(succ(p\ true))(p\ true))(\lambda z.z\ zero\ zero)\ false$$

$$= K_1(\lambda z.z\ one\ zero)\ false$$

$$= (\lambda p z.z(succ(p\ true))(p\ true))(\lambda z.z\ one\ zero)\ false$$

$$= \lambda z.z\ two\ one = one$$

$$pred(two) = one$$

Camlin