Principles of Programming Languages

28/01/22

Assignment 1

19CS30005, Aryan Agarwal

1. (a) $\lambda x.xz\ \lambda y.xy$
$= (\lambda x.((xz)(\lambda y.(xy))))$

This is the free variable (z)

(b) $(\lambda x.xz)\ \lambda y.w\ \lambda w\ wyzx$
$= ((\lambda x.(xz))(\lambda y.(w(\lambda w.(((wy)z)x)))))$

Here the free variables are – z (1st), w (1st), z (2nd) and x (2nd), shown by the '_',

(c) $\lambda x.xy\ \lambda x.yx$
$= (\lambda x.((xy)(\lambda x.(yx))))$

Here the free variables are – the two y's, shown by the '_'

2. To Prove: NOT (NOT FALSE) = FALSE

Proof: NOT (NOT FALSE)
$= \lambda x.((x\ \text{TRUE})\ \text{FALSE})\ (\text{NOT FALSE})$ [Replace 1st NOT with encoding]

$= ((\text{NOT FALSE})\ \text{TRUE})\ \text{FALSE}$ [β reduction: $x \rightarrow$ NOTFALSE]

$= ((\lambda x.((x\ \text{TRUE})\ \text{FALSE})\ \text{FALSE})\ \text{TRUE})\ \text{FALSE}$

[Replace NOT with encoding]

$= (((\text{FALSE TRUE})\text{FALSE})\text{TRUE})\text{FALSE}$

[β reduction: $x \to$ FALSE]

$= ((((\lambda x.\lambda y.y)\text{TRUE})\text{FALSE})\text{TRUE})\text{FALSE}$

[Replace FALSE with encoding]

$= (((\lambda v.v)\text{FALSE})\text{TRUE})\text{FALSE}$ [β reduction: $x \to$ TRUE]

$= ((\text{FALSE})\text{TRUE})\text{FALSE}$ [β reduction: $v \to$ FALSE]

$= ((\lambda x.\lambda y.y)\text{TRUE})\text{FALSE}$ [Replace FALSE with encoding]

$= (\lambda v.v)\text{FALSE}$ [β reduction: $x \to$ TRUE]

$= \text{FALSE}$

Hence, LHS = RHS

∴ Proved.

(b) To Prove: OR FALSE TRUE = TRUE

Proof:

OR FALSE FALSE

$= \lambda x.\lambda y.((x\ \text{TRUE})y)\ \text{FALSE TRUE}$

(Replace OR with encoding)

$= \lambda y.((\text{FALSE TRUE})y)\ \text{TRUE}$ [β reduction: $x \to$ FALSE]

$= (\text{FALSE TRUE})\ \text{TRUE}$ [β reduction: $y \to$ TRUE]

$= ((\lambda x.\lambda y.y)\ \text{TRUE})\ \text{TRUE}$ [Replace FALSE will encoding]

$= (\lambda y.y)\ \text{TRUE}$ [β reduction: $x \to$ TRUE]

$= \text{TRUE}$ [β reduction: $y \to$ TRUE]

Hence, LHS = RHS

∴ Proved

(c) add $\bar{5}\,\bar{1}$

$= (\lambda n.\lambda m.\lambda f.\lambda x.\,nf\,(mf\,x))\,(\lambda f.\lambda x.f x)\,(\lambda f.\lambda x.f x)$

[Replace all with encodings]

$= \lambda f.\lambda x.(\lambda f.\lambda x.f^5 x)\,f\,((\lambda f.\lambda x.f x)\,f x)$

[β reduction for n and m]

$= \lambda f.\lambda x.(\lambda v.\lambda u.v^5 u)\,f\,((\lambda f.\lambda x.f x)\,f x)$

$= \lambda f.\lambda x\,.(\lambda u.\,f^5 u)\,((\lambda v.\lambda u.\,v u)\,f x)$

$= \lambda f.\lambda x\,(\lambda u.f^5 u)\,(f x)$

$= \lambda f.\lambda x.\,(f^5\,(f x))$

$= \lambda f.\lambda x\,.(f^6 x)$

$= \bar{6}$

(d) To Prove: ~~If True~~

IF TRUE THEN x ~~[illegible]~~ ELSE $y = x$

[* The value will be $\underline{x}$, not y]

Proof:- IF TRUE THEN x ELSE y

$=$ TRUE $x\ y$ [Replace by encoding]

$= (\lambda x.\lambda y.x)\,xy$ [Replace by encoding]

$= (\lambda y.x)\,y$

$= x$

Hence, LHS = RHS

∴ Proved.

(e) I. add is commutative

To Prove: add $\bar{m}\,\bar{n}$ = add $\bar{n}\,\bar{m}$

add $\bar{m}\,\bar{n}$

$= (\lambda u.\lambda v.\lambda f.\lambda x.\ uf(vfx))\ (\lambda f.\lambda x.\ f^m x)(\lambda f.\lambda x.\ f^n x)$

$= \lambda f.\lambda x.(\lambda f.\lambda x.\ f^m x)\ f((\lambda f.\lambda x.\ f^n x)\ fx)$

$= \lambda f.\lambda x.\ (\lambda x.\ f^m x)\ (f^n x)$

$= \lambda f.\lambda x.\ (f^m(f^n x))$

$= \lambda f.\lambda x.\ (f^{(m+n)} x)$

$= (\overline{m+n})$

add $\bar{n}\,\bar{m}$

$= (\lambda u.\lambda v.\lambda f.\lambda x.\ uf(vfx))\ (\lambda f.\lambda x.\ f^n x)(\lambda f.\lambda x.\ f^m x)$

$= \lambda f.\lambda x.(\lambda f.\lambda x.\ fx)\ f((\lambda f.\lambda x.\ f^m x)\ fx)$

$= \lambda f.\lambda x.\ (\lambda x.f^n x)\ (f^m x)$

$= \lambda f.\lambda x.(f^n(f^m x))$

$= \lambda f.\lambda x(f^{(n+m)} x)$

$= \lambda f.\lambda x\ (f^{(m+n)} x)$

$= (\overline{m+n})$

$\therefore$ add $\bar{m}\,\bar{n}$ = add $\bar{n}\,\bar{m}$

$\therefore$ Proved

II mul is commutative

To Prove: mul $\bar{m}\bar{n}$ = mul $\bar{n}\bar{m}$

mul $\bar{m}\bar{n}$

$= \lambda u.\lambda v.\lambda w.(u(v w))\ \bar{m}\ \bar{n}$

$= \lambda w.(\bar{m}(\bar{n}\ w))$

$= \lambda w.((\lambda f.\lambda x.f^m x)((\lambda f.\lambda x.f^n x)w))$

$= \lambda w.((\lambda f.\lambda x.f^m x)(\lambda x.w^n x))$

$= \lambda w.((\lambda x.(\lambda x.w^n x)^m x))$

$= \lambda w.((\lambda x.\underbrace{w^n(w^n(w^n(\ldots)))\ldots)}_{m\ \text{times}}\ x)$

$= \lambda w.\lambda x.w^{n*m}x$

$= (\overline{n*m})$

mul $\bar{n}\bar{m}$

$= \lambda u.\lambda v.\lambda w.(u(v w))\ \bar{n}\ \bar{m}$

$= \lambda w.(\bar{n}(\bar{m}\ w))$

$= \lambda w.((\lambda f.\lambda x.f^n x)((\lambda f.\lambda x.f^m x)w))$

$= \lambda w.((\lambda f.\lambda x.f^n x)(\lambda x.w^m x))$

$= \lambda w.(\lambda x.(\lambda x.w^m x)^n x)$

$= \lambda w(\lambda x.\underbrace{(w^m(w^m(\ldots)\ldots)}_{n\ \text{times}}\ x)$

$= \lambda w.\lambda x.w^{m*n}x$

$= \lambda w.\lambda x.w^{n*m}x$

$= (\overline{n*m})$

$\therefore$ mul $\bar{m}\,\bar{n}$ = mul $\bar{n}\,\bar{m}$

Hence, Proved.

---- 3 a is at last ----

3 b We name the function to rotate a list N places left as rotLeft. It takes two arguments : n (places to shift) and l (the list)

```
→ (defun rotLeft (n l)
      (append (nthcdr n l)
          (butlast l (-( length l) n)) ))
```

c We define the Ackerman function i.e. ack with two parameters m and n.

```
→ ( define ack
    (lambda (m n)
        (if (= m 0)
            (+ n 1)
            (if (= n 0)
                (ack (- m 1) 1)
                (ack (- m 1) (ack m (- n 1))))))))
```

d We name the function sumOddSquares, with no parameters

```
sumOddSquares :: Int
sumOddSquares = sum( [x * x | x <- [1,3, .. 99]])
```

* As 99 is the largest odd with square less than 10 000.

It returns 166650

(e) For this, we create a function leastDivisor with two parameters:
m: the input and n: the divisor. To run the function, we initiate n=2 in main.

→ leastDivisor :: Int → Int → Int

```
leastDivisor m n
        | mod m n == 0 = n
        | otherwise    = leastDivisor m (n+1)

main = do
      putStrLn "Enter number: "
      input1 <- getLine
      let n = (read input1 :: Int)
      let x = leastDivisor n 2
      print x
```

3) (a) $$! :: Bool → Bool → Bool

```
True $$! True = True
True $$! False = False
False $$! True = False
False $$! False = False
```