Stefanie Shidoosh

LING 185A: HW #1

Part I

1. let x = 4 + 5 in (3 * x)

let x = 9 in (3 * x)	arithmetic
(3 * 9)	let reduction
27	arithmetic

2. (\x -> 3 * x) (4 + 5)

(\x -> 3 * x) 9	arithmetic
3 * 9	lambda reduction
27	arithmetic

3. ((\x -> (\y -> x + (3 * y))) 4) 1

(\y -> 4 + (3 * y)) 1	lambda reduction
4 + (3*1)	lambda reduction
4 + 3	arithmetic
7	arithmetic

4. let x = 4 in (let y = 1 in (x + (3 * y)))

(let y = 1 in (4 + (3 * y)))	let reduction
(4 + (3 * 1))	let reduction
4 + 3	arithmetic
7	arithmetic

5. let x = 4 in (let y = 1 + x in (x + (3 * y)))

(let y = 1 + 4 in (4 + (3 * y)))	let reduction
(let y = 5 in (4 + (3 * y)))	arithmetic
(4 + (3 * 5))	let reduction
(4 + 15)	arithmetic
19	arithmetic

6. ((\x -> (\y -> x + (3 * x))) 4) 1

(\y -> 4 + (3*4)) 1	lambda reduction
4 + (3*4)	lambda reduction
4 + 12	arithmetic
16	arithmetic

7. ((\x -> (\y -> y + (3 * y))) 4) 1

(\y -> y + (3 * y)) 1	lambda reduction
1 + (3 * 1)	lambda reduction
1 + 3	arithmetic
4	arithmetic

8. (\y -> y + ((\y -> 3*y) 4)) 5

(\y -> y + (3*4)) 5	lambda reduction
5 + (3*4)	lambda reduction
5 + 12	arithmetic
17	arithmetic

9. (\y -> ((\y -> 3*y) 4) + y) 5

(\y -> (3*4) + y) 5	lambda reduction
(3*4) + 5	lambda reduction
12 + 5	arithmetic
17	arithmetic

Stefanie Shidoosh

10. (\x -> x * (let x = 3*2 in (x + 7)) + x) 4

(\x -> x * (let x = 6 in (x + 7)) + x) 4	arithmetic
(\x -> x * ((6 + 7) + x) 4	let reduction
4 * (6 + 7) + 4	lambda reduction
4 * 13 + 4	arithmetic
52 + 4	arithmetic
56	arithmetic

11. g ((let x = 4 in (\y -> x + y)) 2)

g ((\y -> 4 + y) 2)	let reduction
g (4 + 2)	lambda reduction
g (6)	arithmetic
(\z -> z + 4) 6	substitution from file
6 + 4	lambda reduction
10	arithmetic

12. let x = 5 in (\z -> x * z)

(\z -> 5 * z)	let reduction

13. (\x -> (\z -> x * z)) 5

(\z -> 5 * z)	lambda reduction

14. f ((\fn -> fn Rock) (\x -> whatItBeats x))

f ((\x -> whatItBeats x) Rock)	lambda reduction
f (whatItBeats Rock)	lambda reduction
f ((\s -> case s of {Rock -> Scissors; Paper -> Rock; Scissors -> Paper}) Rock)	file substitution
f (case Rock of {Rock -> Scissors; Paper -> Rock; Scissors -> Paper})	lambda reduction
f (Scissors)	case reduction
(\s -> case s of {Rock -> 334; Paper -> 138; Scissors -> 99}) Scissors	file substitution
case Scissors of {Rock -> 334; Paper -> 138; Scissors -> 99}	lambda substitution
99	case reduction

15. ((\f -> (\x -> f (f x))) whatItBeats) Paper

(\x -> whatItBeats (whatItBeats x)) Paper	lambda reduction
(whatItBeats (whatItBeats Paper))	lambda reduction
(whatItBeats ((\s -> case s of {Rock -> Scissors; Paper -> Rock; Scissors -> Paper}) Paper))	file substitution
(whatItBeats (case Paper of {Rock -> Scissors; Paper -> Rock; Scissors -> Paper}))	lambda reduction
(whatItBeats (Rock))	case reduction
((\s -> case s of {Rock -> Scissors; Paper -> Rock; Scissors -> Paper}) Rock)	file substitution
case Rock of {Rock -> Scissors; Paper -> Rock; Scissors -> Paper}	lambda reduction
Scissors	case reduction

16. whatItBeats (case Paper of {Rock -> Paper; Paper -> Rock; Scissors -> Scissors})

whatItBeats (Rock)	case reduction
((\s -> case s of {Rock -> Scissors; Paper -> Rock; Scissors -> Paper}) Rock)	file substitution
case Rock of {Rock -> Scissors; Paper -> Rock; Scissors -> Paper}	lambda reduction
Scissors	case reduction

Stefanie Shidoosh

17. (case (Win Rock) of {Draw -> whatItBeats; Win z -> (\s -> Scissors)}) Paper

(\s -> Scissors) Paper	case reduction
Scissors	lambda reduction

18. case (Win (whatItBeats Rock)) of {Draw -> n; Win x -> (n + f x)}

case (Win ((\s -> case s of {Rock -> Scissors; Paper -> Rock; Scissors -> Paper}) Rock)) of {Draw -> n; Win x -> (n + f x)}	file substitution
case (Win (case Rock of {Rock -> Scissors; Paper -> Rock; Scissors -> Paper}) of {Draw -> n; Win x -> (n + f x)}	lambda reduction
case (Win Scissors) of {Draw -> n; Win x -> (n + f x)}	case reduction
(n + f Scissors)	case reduction
1 + f Scissors	file substitution
1 + (\s -> case s of {Rock -> 334; Paper -> 138; Scissors -> 99}) Scissors	file substitution
1 + (case Scissors of {Rock -> 334; Paper -> 138; Scissors -> 99})	lambda reduction
1 + 99	case reduction
100	arithmetic

19. let y = 2 in (case (Win (whatItBeats Rock)) of {Draw -> n; Win y -> (n + f y)} + y)

let y = 2 in (case (Win ((\s -> case s of {Rock -> Scissors; Paper -> Rock; Scissors -> Paper}) Rock)) of {Draw -> n; Win x -> (n + f y)+y}	file substitution
let y = 2 in (case (Win (case Rock of {Rock -> Scissors; Paper -> Rock; Scissors -> Paper}) of {Draw -> n; Win x -> (n + f y)+y}	lambda reduction
let y = 2 in case (Win Scissors) of {Draw -> n; Win y -> (n + f y)} + y)	case reduction
let y = 2 in n + f Scissors + y	case reduction
n + f Scissors + 2	let reduction
1 + f Scissors + 2	file substitution
1 + (\s -> case s of {Rock -> 334; Paper -> 138; Scissors -> 99}) Scissors + 2	file substitution
1 + (case Scissors of {Rock -> 334; Paper -> 138; Scissors -> 99}) + 2	lambda reduction
1 + 99 + 2	case reduction
100 + 2	arithmetic
102	arithmetic

Part II.

Construct an expression e, whose value depends on a shape represented by the variable s and a number represented by the variable x, such that: if the shape s is Rock, then e evaluates to the square of the number x; if the shape s is Paper, then e evaluates to the cube of the number x; and if the shape s is Scissors, then e evaluates to the number x.

e s x = case s of {Rock -> x * x; Paper -> x * x * x; Scissors -> x}

$[Rock/s][4/x]e \Rightarrow 16$ = e Rock 4

e Rock 4 = case Rock of {Rock -> 4 * 4; Paper -> 4 * 4 * 4; Scissors -> 4}

4 * 4	case reduction
16	arithmetic

$[Paper/s][4/x]e \Rightarrow 64$ = e Paper 4

e Paper 4 = case Paper of {Rock -> 4 * 4; Paper -> 4 * 4 * 4; Scissors -> 4}

4 * 4 * 4	case reduction
64	arithmetic

$[Scissors/s][4/x]e \Rightarrow 4$ = e Scissors 4

e Paper 4 = case Paper of {Rock -> 4 * 4; Paper -> 4 * 4 * 4; Scissors -> 4}

4	case reduction