CSCI-GA-2110: Programming Languages, Fall 2024

PSet 2 - Written

Aditeya Baral
N19186654

1 Encoding Let into Lambda

Task 1.1 Demonstrate this encoding by showing how you would translate the following two examples into a version that uses single-argument lambdas instead of `let`.

1. `(let ([x e1] [y e2] [z e3]) e)`

```
((lambda (z) ((lambda (y) ((lambda (x) e) e1)) e2)) e3)

; Example 1: No dependencies
((lambda (z) ((lambda (y) ((lambda (x) (+ x (+ y z))) 2)) 3)) 4)
; x = 2, y = 3, z = 4 -> 2 + 3 + 4 = 9

; Example 2: With dependencies
((lambda (z) ((lambda (y) ((lambda (x) (+ x (+ y z))) 2)) x)) y)
; x = 2, y = x, z = y -> error: x: free variable while typechecking in: x
```

2. `((let* ([x e1] [y e2] [z e3]) e))`

```
((lambda (x) ((lambda (y) ((lambda (z) e) e3)) e2)) e1)

; Example 1: No dependencies
((lambda (x) ((lambda (y) ((lambda (z) (+ x (+ y z))) 4)) 3)) 2)
; x = 2, y = 3, z = 4 -> 2 + 3 + 4 = 9

; Example 2: With dependencies
((lambda (x) ((lambda (y) ((lambda (z) (+ x (+ y z))) y)) x)) 2)
; x = 2, y = x, z = y -> 2 + 2 + 2 = 6
```

2 Lexical Addresses

Task 2.1 Convert the following Racket program into the lexical address form, similarly to the example above.

```
(lambda (x y z)
    (z (lambda (x y) (z x)) y))
```

```
(lambda 3
    ((: 0 2) (lambda 2 ((: 1 2) (: 0 0))) (: 0 1)))
```

Task 2.2 Consider the following program written in lexical address form. Write down an equivalent program written in standard Racket form with names for variables.

```
(lambda 1
(lambda 1
  (: 1 0)))
```

```
(lambda (x)
  (lambda (y)
    x))
```