CS 476 HW6 -- Lambda Calculus

Sandeep Joshi

TOTAL POINTS

22.5 / 25

QUESTION 1

1 1 4.5 / 5

✓ + 1 pts a: %x. x x

✓ + 1 pts b: %x. %y. x

✓ + 1 pts c: takes two arguments

✓ + 1 pts c: applies the second to the first

✓ + 1 pts basic structure of lambda-terms

- 0.5 Point adjustment

1 This applies the first argument to the second

QUESTION 2

2 2 6 / 7

✓ + 2 pts a: each x bound to its binder

✓ + 2 pts b: inner x and outer x

✓ + 2 pts c: first x and y

+ 1 pts c: second x is also bound to the innermost binder

- 1 pts bound lambdas

+ 1 pts a: half right

2 Incorrect

QUESTION 3

3 3 6 / 7

✓ + 1 pts a: first step is (%z. z) (%z. z)

✓ + 1 pts a: second step is %z. z

✓ + 1 pts b: %y. %y. y

+ 2 pts c: discards argument

✓ + 1.5 pts d: correctly evaluates LHS

✓ + 0.5 pts d: correctly evaluates top-level application

- 0.5 pts forgot to remove binder in application

+ 0.5 pts dropped binders for unused arguments

+ 0.5 pts b: dropped binder

- 0.5 pts changed parenthesization

✓ + 1 pts c: confused x's

- 0.5 pts temporarily dropped argument

+ 1 pts d: arguments in reverse order

+ 1 pts c: applied the inner x first

3 Not quite! The x in the body is the inner x, not the outer x.

QUESTION 4

4 4 6 / 6

✓ + 1 pts a: int -> int -> int

✓ + 1 pts b: int -> (int -> int) -> int

✓ + 2 pts c: (int -> int) -> int

✓ + 1 pts c: correct type for function, but not application

✓ + 2 pts d: int -> int

✓ + 1 pts d: correct type for function, but not application

+ 0 pts graded

+ 0.5 pts b: dropped first argument

+ 0.5 pts c: arguments in wrong order

+ 0.5 pts a: two-argument function syntax

+ 0.5 pts b: missing return type

+ 0.5 pts b: arguments in wrong order

+ 0.5 pts d: extra arrow

+ 1 pts right argument types, wrong return types

1 1 4.5 / 5

✓ + 1 pts a: %x. x x

✓ + 1 pts b: %x. %y. x

✓ + 1 pts c: takes two arguments

✓ + 1 pts c: applies the second to the first

✓ + 1 pts basic structure of lambda-terms

- 0.5 Point adjustment

1 This applies the first argument to the second

2 2 6 / 7

✓ + 2 pts a: each x bound to its binder

✓ + 2 pts b: inner x and outer x

✓ + 2 pts c: first x and y

+ 1 pts c: second x is also bound to the innermost binder

- 1 pts bound lambdas

+ 1 pts a: half right

2 Incorrect

3 3 6 / 7

- ✓ + 1 pts a: first step is (%z. z) (%z. z)
- ✓ + 1 pts a: second step is %z. z
- ✓ + 1 pts b: %y. %y. y
- + 2 pts c: discards argument
- ✓ + 1.5 pts d: correctly evaluates LHS
- ✓ + 0.5 pts d: correctly evaluates top-level application
- - 0.5 pts forgot to remove binder in application
- + 0.5 pts dropped binders for unused arguments
- + 0.5 pts b: dropped binder
- - 0.5 pts changed parenthesization
- ✓ + 1 pts c: confused x's
- - 0.5 pts temporarily dropped argument
- + 1 pts d: arguments in reverse order
- + 1 pts c: applied the inner x first

3 Not quite! The x in the body is the inner x, not the outer x.

4 4 6 / 6

- ✓ + 1 pts a: int -> int -> int
- ✓ + 1 pts b: int -> (int -> int) -> int
- ✓ + 2 pts c: (int -> int) -> int
- ✓ + 1 pts c: correct type for function, but not application
- ✓ + 2 pts d: int -> int
- ✓ + 1 pts d: correct type for function, but not application
  - + 0 pts graded
  - + 0.5 pts b: dropped first argument
  - + 0.5 pts c: arguments in wrong order
  - + 0.5 pts a: two-argument function syntax
  - + 0.5 pts b: missing return type
  - + 0.5 pts b: arguments in wrong order
  - + 0.5 pts d: extra arrow
  - + 1 pts right argument types, wrong return types