Started on	Thursday, 31 March 2022, 9:10 AM
State	Finished
Completed on	Thursday, 31 March 2022, 9:17 AM
Time taken	7 mins 24 secs
Marks	3.00/4.00
Grade	**7.50** out of 10.00 (**75**%)

Question 1

Correct

Mark 1.00 out of 1.00

True or False? Let binding can be defined as a derived form by using a lambda abstraction, but only if the type of the binding is explicitly specified in the let. In other words:

- **let x = t1 in t2** cannot be a derived form in simply typed lambda calculus
- **let x:T = t1 in t2** can be a derived form in simply typed lambda calculus

Select one:

- True
- False ✔

The correct answer is 'False'.

Question 2

Correct

Mark 1.00 out of 1.00

Consider the following lambda term with records. Is it well-typed? If yes, what is the type? If not, what is the type error?

let f = (λx:{a:Nat,b:Bool}. **{**a = x.b, b = x.a**}**) **in** f **{**a = 1, b = true**}**

- a. Ill-typed. x is not a record
- b. Ill-typed. x.b is expected to have type Nat, but actually has type Bool
- c. Well-typed. {a:Nat,b:Bool}
- d. Well-typed. {a:Bool,b:Nat} ✔
- e. Ill-typed. x.a is expected to have type Nat, but actually has type Bool
- f. Ill-typed. x.a is expected to have type Bool, but actually has type Nat
- g. Ill-typed. Cannot apply f to **{**a = 1, b = true**}**, as it does not have the right type.
- h. Ill-typed. x.b is expected to have type Bool, but actually has type Nat

Your answer is correct.

The correct answer is:
Well-typed. {a:Bool,b:Nat}

Question 3

Correct

Mark 1.00 out of 1.00

Consider the following lambda term with records. Is it well-typed? If yes, what is the type? If not, what is the type error?

let f = (λx:{a:Nat,b:Nat}. **{**a = x.b, b = x.a**}**) **in** f **{**a = 1, b = true**}**

- a. Ill-typed. x.b is expected to have type Bool, but actually has type Nat
- b. Ill-typed. x.a is expected to have type Nat, but actually has type Bool
- c. Ill-typed. x.a is expected to have type Bool, but actually has type Nat
- d. Ill-typed. x.b is expected to have type Nat, but actually has type Bool
- e. Ill-typed. x is not a record
- f. Well-typed. {a:Nat,b:Bool}
- g. Ill-typed. Cannot apply f to **{**a = 1, b = true**}**, as it does not have the right type. ✔
- h. Well-typed. {a:Bool,b:Nat}

Your answer is correct.

The correct answer is:
Ill-typed. Cannot apply f to **{**a = 1, b = true**}**, as it does not have the right type.

Question 4

Incorrect

Mark 0.00 out of 1.00

Assuming call-by-value evaluation strategy with standard one-step reduction semantics, how many steps does it take to evaluate the following term with pairs?

(λx:Nat × Nat. {x.2, x.1}) {pred (succ 0), if true then {0, false} else {1, true}}

Select one:

- a. 6
- b. 2
- c. 3 ✖
- d. 5
- e. 1
- f. 9
- g. 10
- h. 0 (this term is already a value)
- i. This term will get stuck during evaluation.
- j. 4
- k. 8
- l. 7
- m. Infinitely many. This term will never compute to a value.

Your answer is incorrect.

The correct answers are:
5,

6,

Infinitely many. This term will never compute to a value.,

This term will get stuck during evaluation.

◄ Quiz 2 — Mar 30 from 10:50 to 11:00 (10 minutes)

Jump to...

Quiz 4 — Apr 6 from 10:50 to 11:00 (10 minutes) ►

Data retention summary
Get the mobile app