CSE130 Discussion

Midterm Prep

2021-10-29

Agenda

- Lambda Calculus Reminder
- Haskell

Lambda Calculus

Tips:

1. Function application is LEFT associative!
2. Function abstraction is RIGHT associative!

```
\f -> f (\x -> x) (\g -> g)
```

Lambda Calc

Check the box next to **each** term that contains **exactly one** redex (i.e. there is one and only one way to apply a beta step to this term).

(A) `(\x -> x) (\x -> x)`   [ ]

(B) `\x -> x (\x -> x)`   [ ]

(C) `f (\x -> x) (\x -> x)`   [ ]

(D) `(\x -> x) f (\x -> x)`   [ ]

(E) `(\f x -> f (f x)) y z`   [ ]

Which are valid reductions?

```
(\x -> x) (\y -> apple y) (\z -> z)

(A) =b> (\x -> x) (apple (\z -> z))                     [ ]

(B) =b> (\y -> apple y) (\z -> z)                       [ ]

(C) =a> (\z -> z) (\y -> apple y) (\z -> z)             [ ]

(D) =a> (\x -> x) (\y -> orange y) (\z -> z)            [ ]

(E) =~> apple (\z -> z)                                 [ ]
```

Haskell

Go through an exam question live

Midterm WI19 Part II