CSE130 Discussion

Haskell: Simple Recursion

2021-10-15

Agenda

- Using Haskell
  - GHCi
  - Patterns
  - Recursion
- PA1 tips

Haskell

- Functional programming language
- NO side effects (e.g., write to file, DB, internet)
  - Unless you explicitly handle them

Function Application

Elsa:

```
ITE A B C = (((ITE A) B) C)
```

Haskell:

```
ite a b c = (((ite a) b) c)
```

LEFT ASSOCIATIVE!

Quiz

In Haskell, like Elsa, which is the correct parenthesization of:

```
fold add 0 empty
```

```
A.  (fold (add 0 empty))
B.  (fold (add (0 empty))
C.  (fold add) (0 empty)
D.  (((fold add) 0) empty)
E.  None of above
```

Quiz

In Haskell, like Elsa, which is the correct parenthesization of:

```
fold add 0 empty
```

A. `(fold (add 0 empty))`

B. `(fold (add (0 empty))`

C. `(fold add) (0 empty)`

D. `(((fold add) 0) empty)  -- left associative`

E. None of above

GHCI

Glorious Glasgow Haskell Compiler Interpreter

- $ make ghci

Issue?:

```
The following GHC options are incompatible with GHCi and have not been
passed to it: -threaded

Configuring GHCi with the following packages: hw1-haskell

/tmp/haskell-stack-ghci/74924166: createDirectory: permission denied
(Permission denied)
```

Fix: `$TMPDIR=$HOME/.tmp make ghci`

Practice #1: Write a function that computes x^n

```
power :: Int -> Int -> Int
```

Practice #2: Find the last element of a list.

```
myLast :: [a] -> a
```

Practice #3: Find the last but one element of a list.

```
lastButOne :: [a] -> a
```

Practice #4: Find the k'th element of a list. The first element starts at 0.

```
elementAt :: [a] -> Int -> a
```

Practice #5: Eliminate consecutive duplicates of list elements

```
dedup :: [a] -> [a]
```

Useful Resources

- Learn You a Haskell

  http://learnyouahaskell.com/

- Haskell wiki

  https://wiki.haskell.org/Learning_Haskell