Exam 2 Review

13 November 2020
Logistics
When?

Exam is released later today.

Due 11:59 p.m. on Friday.

Take it any time between now and then.
How long?

The exam is designed to be about one hour long, but you’re being given two hours to take it.
What can you use?

One 8.5×11-inch piece of paper with anything written on it.

(You may be asked to submit a copy of it.)
Strategy
Something > Nothing
Approach > Syntax
Format
Almost every problem will ask you to write data and/or function definitions.

No, you can’t use DrRacket.

This may feel weird, but it’s important to be able to think about a program without needing to run it.

Cf.: “whiteboard interviews”
Topics
Lists and recursion
List predicates (cons?, empty?)
List selectors (first, rest)
List constructors (cons / list)
Abstraction: Add a parameter
Abstraction: Add a function as input

Built-in list abstractions: foldr, map, filter, andmap, ormap

Scope of functions and values
local changes scope

A function that generates a function
lambda
Rapid-fire review
(first (list 5 10))

What value is the result?
(first (list 5 10))
What value is the result?

5
(first (list 5 10))

What data type is the result?
(first (list 5 10))

What data type is the result?

Number
(rest (list 5 10))

What value is the result?
(rest (list 5 10))

What value is the result?

(list 10)
(rest (list 5 10))

What data type is the result?
(rest (list 5 10))

What data type is the result?

List (of Numbers)
;; foo : ListOfSongs -> Natural

What data type does foo consume?
;;; foo : ListOfSongs -> Natural

What data type does foo consume?

ListOfSongs
;; foo : ListOfSongs → Natural

What data type does foo produce?
;;; foo : ListOfSongs → Natural

What data type does foo produce?

Natural
;;; foo : [List-of X] → Natural

What data type does foo consume?
;; foo : [List-of X] -> Natural

What data type does foo consume?

A List of X
;; foo : [List-of X] --> Natural
(foo 3 4 5 6)

Is this a valid way to call foo?
;; foo : [List-of X] -> Natural
(foo 3 4 5 6)

Is this a valid way to call foo?

No
;;; foo : [List-of X] → Natural
(foo (list 3 4 5 6))

Is this a valid way to call foo?
;;;; foo : [List-of X] → Natural
(foo (list 3 4 5 6))

Is this a valid way to call foo?

Yes!
;; foo : [List-of X] → Natural
(foo (list "a" "b"))

Is this a valid way to call foo?
;; foo : [List-of X] → Natural
(foo (list "a" "b"))

Is this a valid way to call foo?

Yes!
;; foo : [List-of X] → Natural
(foo '())

Is this a valid way to call foo?
;;; foo : [List-of X] → Natural
(foo '())

Is this a valid way to call foo?

Yes!
;; foo : [List-of X] → Natural
(foo #false)

Is this a valid way to call foo?
;; foo : [List-of X] → Natural
(foo #false)

Is this a valid way to call foo?

No!
(local [defn
    defn
    ...
  ]
  ???)

What's missing from a valid `local`?
(local [defn
    defn
    ...]
    ???)

What’s missing from a valid \texttt{local}?  

\textit{An expression (the body)}
By default, I can use a constant anywhere in my program, because constants have ______ scope.
By default, I can use a constant anywhere in my program, because constants have _____ scope.

`global`
Parameters have _____ scope.
Parameters have _____ scope.
The list abstraction **map** takes in a list and produces a _____.
The list abstraction \texttt{map} takes in a list and produces a \underline{list}. 
[X -> Boolean] represents a function that consumes a single value and produces a _____.
[X -> Boolean] represents a function that consumes a single value and produces a _____.

Boolean
This exam is going to go ____.
This exam is going to go _____.

great! :-}