# Final Exam (Part 1) in Program Design and Data Structures (1DL201)

Teachers: Johannes Borgström, Dave Clarke

2018-01-11 / 08:00-13:00

#### Instructions

Read and follow these instructions carefully to increase your chance of getting good marks.

- This is a closed book exam. You may use a standard English dictionary. Otherwise, no notes, calculators, mobile phones, or other electronic devices are allowed. Cheating will not be tolerated.
- This is a multiple-choice exam. Each question has exactly **one** correct answer.
- You may keep these question sheets. **Only hand in the answer sheet.** Also read the instructions on the answer sheet before you start.
- Johannes or Dave will come to the exam hall around 10:00 to answer questions.

Good luck!

### Common Material

Some of the exam questions refer to the following function:

```
{- func a ls
    PRE: ?PRE?
    RETURNS: ?RETURNS?
    -}
func :: ?TYPE?
func = funci []
-- funci a b ls
-- VARIANT: ?VARIANT?
funci r _ [] = r
funci r a ((x,y):ls)
    | x == a = funci (y:r) a ls
    | otherwise = funci r a ls
```

## Questions

Please choose a single answer for each question. Read the questions carefully, and watch out for negations (**not**, **except**, etc.).

Question 1: What is the value of func 3 [(1, 'a'), (2, 'b'), (3, 'c')]?

Question 2: What is a type (?TYPE?) of func?

Question 3: What is the most appropriate precondition (?PRE?) for func a ls?

A ls is a list

C ls contains no duplicate elements

B ls contains at most one element matching (a,x)

D ls contains a pair (a,c)

E True

Question 4: What is the most appropriate description of the return value (?RETURNS?) for func a 1s?

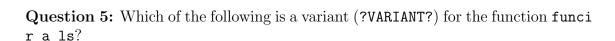
A a list of all pairs (x,y) in ls where x=a

C a list of all y such that (a,y) in 1s

D y:r

В [ъ]

E a list of all y such that (y,a) in 1s



D 2 \* length ls

Question 6: Which of the following modifications to the definition of func (or funci) will reverse the order of the list returned from func?

$$\boxed{\mathbf{C}}$$
 replace ((x,y):ls) by (ls++[(x,y)]).

$$E$$
 replace  $x==a$  by  $a==y$ .

Question 7: Which of the following expressions does not evaluate to 15?

Question 8: Consider the expression

Evaluating this expression will result in ...

C a type error.

E none of these.

D a run-time error.

Question 9: Consider the declaration

$$f x = let f x = x+1 in f (f x)$$

Which of the following is equivalent to the declaration above?

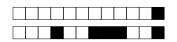
$$\boxed{A}$$
 f z = let g y = z+1 in g (g z)

$$\boxed{\mathrm{B}}$$
 f x = let f y = y+1 in f (f y)

$$\overline{|C|}$$
 f x = let g z = z+1 in g (g x)

$$\boxed{D}$$
 f y = let g y = y+1 in f (f y)

E None of these.



Question 10: Consider the function

zip (x:xs) (y:ys) = (x,y) : zip xs ys zip 
$$_{-}$$
 = []

Which of the following expressions is **not** a variant for zip xs ys?

A length ys

D length xs + length ys

B length xs

C length xs \* length ys

E abs (length xs - length ys)

Question 11: Which of the following evaluates to 11?

A foldr (\*) 0 [1,7,3]

D foldr (+) 1 [1,5,3]

B foldr (+) 0 [1,7,3]

C foldr (\*) 1 [1,7,3]

E foldr (:) [] [1,5,3]

Question 12: What is the type of bar, which is defined as follows:

bar f g a = f (g a) a

|A| (a -> b -> c) -> (a -> b) -> a -> c

 $\boxed{B}$  (a -> b -> c) -> (a -> b) -> a -> c

 $\boxed{C}$  ((a -> b -> c), (b -> a), b) -> c

 $\boxed{D}$  (a -> b -> c) -> b -> (a -> b) -> c

 $\boxed{\mathrm{E}}$  (a -> b -> c) -> (b -> a) -> a -> c

Question 13: Which of the following functions is different from the others — that is, which function gives different results when applied to the same arguments?

f [] = [] f (x:xs) | x > 2 = x + 10 : f xs A | otherwise = f xs

 $\boxed{\mathrm{B}}$  f = map (+10) . filter (>2)

|C| f xs = [x + 10 | x <- xs, x > 2]

 $\boxed{D}$  f = map (\x -> x + 10) . filter (\x -> 2 > x)

 $\boxed{\mathrm{E}}$  f = filter (>12) . map (+10)

**Question 14:** Type  $\tau$  is an *instance* of type  $\rho$  if  $\tau$  can be obtained from  $\rho$  by instantiating  $\rho$ 's type variables with other types (which may also be variables). Two types  $\tau$  and  $\rho$  are *related by instantiation* if  $\tau$  is an instance of  $\rho$  or  $\rho$  is an instance of  $\tau$ .

Which of the following types is **not** related to any of the others by instantiation?

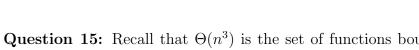
A a -> a

D (a, a) -> (a, b)

 $\boxed{\mathrm{B}}$  (a, b) -> (a, b)

 $\boxed{\mathrm{C}}$  (Int, a) -> (Int, a)

E (a -> a) -> (a -> a)



Question 15: Recall that  $\Theta(n^3)$  is the set of functions bounded both above and below by  $n^3$ , modulo a constant factor, etc. Which of the following is equal to  $\Theta(n^3)$ ?

$$\bigcirc$$
  $O(n^3) \setminus O(n^2)$ 

$$\boxed{\mathsf{D}} \ (O(n^3) \setminus O(n^2)) \cap (\Omega(n^2) \setminus \Omega(n^3))$$

$$\square$$
  $O(n^3) \cap \Omega(n^3)$ 

|E| All of the above

Question 16: Consider the following function

foo f 
$$l@(\_:as) = f l : foo f as$$

Assuming that the run-time cost of some function g is given by  $T_g(n)$ , where n is the size of its input, which of the following recurrences describes the run-time cost of foo g?

$$\boxed{\mathbf{A}} \ T(n) = \left\{ \begin{array}{ll} T_g(0) & \text{if } n = 0 \\ \Theta(1) + T(n-1) + \Theta(n) & \text{if } n \geq 1 \end{array} \right.$$

$$\boxed{\mathbf{B}} \ T(n) = \begin{cases} T_g(0) & \text{if } n = 0 \\ T_g(n) + T(n-1) + \Theta(n) & \text{if } n > 1 \end{cases}$$

$$\boxed{D} T(n) = \begin{cases}
T_g(0) & \text{if } n = 1 \\
T_g(n) + T(n-1) + \Theta(n) & \text{if } n > 1
\end{cases}$$

$$\boxed{E} T(n) = \begin{cases}
T_g(0) & \text{if } n = 0 \\
T_g(n) + T(n-1) + \Theta(1) & \text{if } n \ge 1
\end{cases}$$

$$\boxed{\mathbf{E}} \ T(n) = \begin{cases} T_g(0) & \text{if } n = 0\\ T_g(n) + T(n-1) + \Theta(1) & \text{if } n > 1 \end{cases}$$

**Question 17:** Which of the following recurrences has the closed form T(n) = 5n+7?

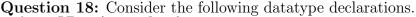
$$\boxed{\mathbf{A}} \ T(n) = \left\{ \begin{array}{ll} 7 & \text{if } n = 0 \\ T(n-1) + 5 & \text{if } n \ge 1 \end{array} \right.$$

$$\boxed{\mathbf{B}} \ T(n) = \begin{cases} 5 & \text{if } n = 0\\ 2T(n-1) + 7 & \text{if } n \ge 1 \end{cases}$$

$$\boxed{C} T(n) = \begin{cases} 5 & \text{if } n = 0 \\ 7T(n) + 1 & \text{if } n \ge 1 \end{cases}$$

$$\boxed{D} T(n) = \begin{cases} 7 & \text{if } n = 0\\ 5T(n-1) & \text{if } n \ge 1 \end{cases}$$

$$\boxed{\mathbf{E}} \ T(n) = \left\{ \begin{array}{ll} 3 & \text{if } n = 0 \\ 5T(n-1) + 4 & \text{if } n \ge 1 \end{array} \right.$$



data BTreeA a = Leaf a

| Node (BTreeA a) a (BTreeA a)

data BTreeB a = Leaf a

| Node (BTreeB a) (BTreeB a)

data BTreeC a = Leaf

| Node (BTreeC a) a (BTreeC a)

data BTreeD a = Leaf a

| Node a (BTreeD a) (BTreeD a)

data BTreeE a = Leaf a

| Node (BTreeE a) (BTreeE a) a

Which of them can represent trees containing an arbitrary non-negative number of data items?

A BTreeA

B BTreeB

|C| BTreeC

D BTreeD

E BTreeE

**Question 19:** Which of the binary tree datatypes defined above admits insertion of a single new data item in constant time (i.e., O(1))?

With insertion is meant an operation insert :: a -> BTreeX a -> BTreeX a, where insert x t returns a tree containing x and the data items in t (similar to (:) :: a -> [a] -> [a]).

A BTreeB and BTreeC

D BTreeB only

B BTreeA, BTreeD, and BTreeE

C BTreeC only

E All except BTreeB

#### Question 20:

Which of the datatype declarations above **cannot** be used with the standard binary search tree invariant and search algorithm as seen in class?

A BTreeA

B BTreeB

C BTreeC

D BTreeD

E BTreeE