CMPS-112 · Programming Languages · Fall 2016 · Midterm Test 1 of 1

$Id: cmps112-2016q4-exam1.mm,v 1.71 2016-10-21 15:42:40-07 - - $

.PS

examboxes(5)

.PE

No books; No calculator; No computer; No email; No internet; No

notes; No phone. Do your scratch work elsewhere and enter only your

final answer into the spaces provided. Points will be deducted for

messy answers. Unreadable answers will be presumed incorrect.

.EQ

delim $$

.EN

1. Ocaml. Define a tail-recursive function evenlen which returns true

if its argument is a list of even length and false if there are an

odd number of elements in the list. Do not use a higher-order

function. [2pt]

# evenlen [];;

- : bool = true

# evenlen [1];;

- : bool = false

# evenlen [1;2];;

- : bool = true

2. Scheme. Define a tail-recursive function oddlen which returns #t

if the list has an odd number of elements, and #f if not. Do not

use a higher-order function. [2pt]

> (oddlen '())

#f

> (oddlen '(1))

#t

> (oddlen '(1 2))

#f

3. Scheme. Define the tail-recursive function foldl so that it may be

used in the following example. [2pt]

> (define (sum list) (foldl + 0 list))

> (sum '(1 2 3))

6

4. Ocaml. Define the function reverse to reverse an arbitrary list.

Use $ O ( n ) $ time and $ O ( 1 ) $ stack. You may code tail

recursion explicitly, or express the function as a one-liner using

List.fold\_left. [2pt]

val reverse : 'a list -> 'a list

# reverse [1;2;3;4];;

- : int list = [4; 3; 2; 1]

5. Scheme. Draw a picture of the following Scheme expression. For

each cons cell, draw a small box with two arrows coming out of it.

Each arrow should point at either another cell or an atom. Write

the Greek letter phi (\phi) to indicate a null pointer. [2pt]

((a) (b c) ((d e)))

6. Ocaml. Define length and sum for a list using a \beta-reduced

version of the definitions. Fill in the space with an appropriate

fun and another argument. [2pt]

# let foldl = List.fold\_left;;

val foldl : ('a -> 'b -> 'a) -> 'a -> 'b list -> 'a

# let length = foldl

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;;

val length : '\_a list -> int

# let sum = foldl \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;;

val sum : int list -> int

7. Ocaml. Define the function zipwith that takes a function and two

lists and returns a new list with the elements combined. Use

failwith if the lists are not of the same length. [2pt]

val zipwith : ('a -> 'b -> 'c) -> 'a list -> 'b list -> 'c list

# zipwith (+) [1;2;3;4] [5;6;7;8];;

- : int list = [6; 8; 10; 12]

# zipwith (+) [1;2;3;4] [5;6;7;8;9];;

Exception: Failure "zipwith".

# zipwith (fun a b -> a,b) [1;2;3] ['a';'b';'c'];;

- : (int \* char) list = [(1, 'a'); (2, 'b'); (3, 'c')]

# zipwith max [1;2;3;4] [4;3;2;1];;

- : int list = [4; 3; 3; 4]

8. Ocaml. In a strongly typed language:

(a) Every expression has exactly one type.

(b) When an expression is evaluated, exactly one of the following

general things might happen: [2pt]

(i)

(ii)

(iii)

(iv)

9. Scheme. The Collatz conjectures states that for any positive

integer $n$, if it is repeatedly replaced by $ n / 2 $ when even

and $ 3 n + 1 $ when odd, it eventually converges on the integer

$1$. Write a function that uses a tail-recursive inner function to

return a list of all integers starting from the argument and ending

with 1. The inner function produces the list in the reverse order,

but the outer function reverses the list. Some Scheme functions to

use: remainder, quotient, reverse, etc. [4pt]

> (collatz 4)

(4 2 1)

> (collatz 10)

(10 5 16 8 4 2 1)

> (collatz 20)

(20 10 5 16 8 4 2 1)

> (collatz 16)

(16 8 4 2 1)

> (collatz 17)

(17 52 26 13 40 20 10 5 16 8 4 2 1)

10. C or C++. Code a function to reverse a list. Do not allocate or

free any memory. Do not cause memory leak or use uninitialized

memory. Do not use recursion. Write a loop which does nothing but

manipulate pointers. Return a pointer to the first node in the

reversed list. [2pt]

|

typedef struct node node; | node\* reverse (node\* head) {

struct node { |

int value; |

node\* link; |

}; |

|

| }

11. Scheme. Define the functions map and filter. Do not use higher-

order functions.

(a) map [1pt]

> (map (lambda (n) (+ 1 n)) '(3 6 9))

(4 7 10)

(b) filter [2pt]

> (filter (lambda (n) (< n 4)) '(1 2 3 4 5 6 7))

(1 2 3)

Multiple choice. To the left of each question, write the letter that

indicates your answer. Write Z if you don't want to risk a wrong

answer. Wrong answers are worth negative points. [6pt]

+--------------------------+------+------+------------+

|number of | |× 1 = | $= a$ |

|correct answers | | | |

+--------------------------+------+------+------------+

|number of | |× ½ = | $= b$ |

|wrong answers | | | |

+--------------------------+------+------+------------+

|number of | |× 0 = | 0 |

|missing answers | | | |

+--------------------------+------+------+------------+

|column total | 6 | | $= c$ |

|$ c = max ( a - b , 0 ) $ | | | |

+--------------------------+------+------+------------+

1. ``Go To Statement Considered Harmful'', Communications of the ACM,

Vol. 11, No. 3, March 1968.

(A) John Backus

(B) Edsger Dijkstra

(C) Donald Knuth

(D) John McCarthy

2. Static type inference is a major feature of:

(A) C

(B) Java

(C) Ocaml

(D) Scheme

3. In smalltalk code is executed by:

(A) calling functions which are static members of classes.

(B) making use of the standard template library.

(C) sending messages to objects.

(D) using higher-order functions.

4. In the expression (lambda (x) (+ x y))

(A) x is bound and y is bound.

(B) x is bound and y is free.

(C) x is free and y is bound.

(D) x is free and y is free.

5. The make language can be referred to as:

(A) a functional language.

(B) a ``little'' language.

(C) a logic language.

(D) an object-oriented language.

6. In Smalltalk, the expression 3+4. means:

(A) The message + is sent to the number 3, the result of which is

a function that accepts the message 4.

(B) The message +4 is sent to the number 3.

(C) The message 3+ is sent to the number 4.

(D) The messages 3 and 4 are sent to the operator +.

Multiple choice. To the left of each question, write the letter that

indicates your answer. Write Z if you don't want to risk a wrong

answer. Wrong answers are worth negative points. [12pt]

+--------------------------+------+------+------------+

|number of | |× 1 = | $= a$ |

|correct answers | | | |

+--------------------------+------+------+------------+

|number of | |× ½ = | $= b$ |

|wrong answers | | | |

+--------------------------+------+------+------------+

|number of | |× 0 = | 0 |

|missing answers | | | |

+--------------------------+------+------+------------+

|column total | 12 | | $= c$ |

|$ c = max ( a - b , 0 ) $ | | | |

+--------------------------+------+------+------------+

1. What is the Scheme value of:

(caddr '(1 2 3 4 5))

(A) (3 4 5)

(B) (4 5)

(C) 2

(D) 3

2. How much stack space does the following function use?

let rec f n = match n with

| 0 -> 0

| 1 -> 1

| n -> f (n - 1) + f (n - 2)

(A) $ O ( 1 ) $

(B) $ O ( log n ) $

(C) $ O ( n ) $

(D) $ O ( 2 sup n ) $

3. What is the Ocaml type signature of:

(/);;

(A) - : int \* int \* int

(B) - : int \* int -> int

(C) - : int -> int \* int

(D) - : int -> int -> int

4. Which function can be written in a tail recursive purely

functional manner?

(A) filter

(B) fold\_left

(C) fold\_right

(D) map

5. Lisp (McCarthy) and Scheme (Steele and Sussman), in general form,

are based on a form of mathematics first formulated by Alonzo

Church.

(A) \lambda-calculus

(B) \mu-calculus

(C) \pi-calculus

(D) \psi-calculus

6. Which line is a comment in Scheme?

(A) (\*...\*)

(B) /\*...\*/

(C) //...

(D) ;;...

7. What feature of imperative languages is typically missing from

functional languages?

(A) conditionals

(B) functions

(C) loops

(D) recursion

8. What is the signature of the Ocaml function List.hd (equivalent to

car)?

(A) 'a -> 'a list

(B) 'a -> 'a

(C) 'a list -> 'a list

(D) 'a list -> 'a

9. Given:

# List.map ((+)3) [1;2;3];;

- : int list = [4; 5; 6]

what is the type of List.map ((+)3)?

(A) int -> int

(B) int -> int list

(C) int list -> int

(D) int list -> int list

10. In both Java and C++, what keyword is used to restrict access to a

class itself but allow access to classes derived from it?

(A) inheritance

(B) private

(C) protected

(D) public

11. Backus-Naur format describes what about a language?

(A) environment

(B) linkage

(C) semantics

(D) syntax

12. What language was designed by John Kemeny and Thomas Kurtz in

1965?

(A) BASIC

(B) COBOL

(C) FORTRAN

(D) LISP