CMPS-112 · Comparative Programming Languages · Fall 2014 · Final1Eofm1

$Id: cmps112-2014q4-exam3.mm,v 1.115 2014-12-11 13:23:31-08 - - $

.PS

examboxes(5)

.PE

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

notes; No phone. Neatness counts! Do your scratch work elsewhere

and enter only your final answer into the spaces provided.

.EQ

delim $$

.EN

1. Ocaml. Define the function reverse which reverses a list. Use an

inner function that is tail recursive. Do not use a higher-order

function. [2pt]

val reverse : 'a list -> 'a list

2. Ocaml. Define the functions reverse and sum using fold\_left.

(Fill in the blanks.) [3pt]

# List.fold\_left;;

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

# let reverse = List.fold\_left

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

val reverse : '\_a list -> '\_a list

# let sum = List.fold\_left

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

val sum : int list -> int

# let length = List.fold\_left

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

val length : '\_a list -> int

3. Ocaml. The Collatz conjecture states that for any positive integer

$n$, if it is replaced by $ n / 2 $ when even and $ 3 n + 1 $ when

odd, that the sequence eventually converges on $1$. Write a

function which 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 reverse order, but the

result is reversed by the outer function. [3pt]

# collatz;;

- : int -> int list

# collatz 1;;

- : int list = [1]

# collatz 2;;

- : int list = [2; 1]

# collatz 3;;

- : int list = [3; 10; 5; 16; 8; 4; 2; 1]

4. Scheme. Write a function take in Scheme which will make a copy of

the first $n$ elements of a list. If there are fewer than $n$

elements in the list, it returns the complete list. If $n <= 0$ it

returns the empty list. [2pt]

(define (take n list)

5. Ocaml, or any functional language. Rules about type checking.

(i) Every expression has exactly one type.

(ii) When an expression is evaluated, exactly one of four general

things may happen. List them. [2pt]

(a)

(b)

(c)

(d)

6. \lambda-calculus. For both applicative and normal order, perform

\beta-reduction on the following expression. [2pt]

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

|Applicative order | Normal order |

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

|$ ( lambda x . ~ ~ \* ~ x ~ x ) ~ ( + ~ 2 ~ 3 ) $ | $ ( lambda x . ~ ~ \* ~ x ~ x ) ~ ( + ~ 2 ~ 3 ) $ |

| | |

| | |

| | |

| | |

| | |

| | |

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

7. Ocaml. Given the function at the left, fill in the table at the

right giving the Ocaml types for each item. [2pt]

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

let fac n = | fac | |

if n < 0 +------+--------------------------------+

then failwith "fac n | n < 0" | n | |

else let rec fac' n' m' = +------+--------------------------------+

if n' = 0 | 0 | |

then m' +------+--------------------------------+

else fac' (n' - 1) (n' \* m') | fac' | |

in fac' n 1 +------+--------------------------------+

;; | n' | |

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

| m' | |

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

| - | |

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

| \* | |

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

8. Prolog. Given the graph shown here, write edge facts to describe

it. Write a rule adjacent which uses edge to determine if two

nodes are adjacent. [2pt]

.PS 0.75i

A: circle "a"

B: circle "b" at A+(-4\*circlerad,-4\*circlerad)

C: circle "c" at A+(0,-4\*circlerad)

D: circle "d" at A+(4\*circlerad,-4\*circlerad)

line from A to B chop

line from A to C chop

line from A to D chop

line from B to C chop

line from C to D chop

.PE

9. Scheme. Define the functions map and filter. [2pt]

> (map (lambda (n) (+ 1 n)) '(3 4 5))

(4 5 6)

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

(5 6 7)

10. Perl. Write a script in Perl which will iterate over all of the

input files given on the command line and print the contents of all

of the files to the standard output. If no files are specified,

copy the standard input. You are limited to one statement only.

Hint: Use <>. [1pt]

#!/usr/bin/perl

11. Prolog. Given facts like the ones presented at the left, define

the rules father and mother where the first argument is the parent

and the second argument is the child. [2pt]

parents(henry\_vii,elizabeth\_of\_york,henry\_viii).

parents(henry\_viii,catherine\_of\_aragon,mary\_i).

parents(henry\_viii,anne\_boleyn,elizabeth\_i).

parents(henry\_viii,jane\_seymour,edward\_vi).

| ?- father(X,henry\_viii).

X = henry\_vii

| ?- father(henry\_viii,X).

X = mary\_i

X = elizabeth\_i

X = edward\_vi

12. Define a function range with two integer arguments and which

returns a list of all arguments in order including the two

arguments. If the first number is larger than the second, return

an empty list. [3pt]

(a) Ocaml.

# range 2 7;;

- : int list = [2; 3; 4; 5; 6; 7]

# range 7 2;;

- : int list = []

(b) Scheme.

> (range 2 7)

(2 3 4 5 6 7)

> (range 7 2)

()

(c) Perl.

print "[@{[range(2,7)]}]\n";

[2 3 4 5 6 7]

print "[@{[range(7,2)]}]\n";

[]

13. Write a function that takes two lists as arguments and which

returns a single list where each element is a list of corresponding

pairs. If the lists are of different lengths, trailing elements of

the longer list are ignored.

(a) Scheme. [2pt]

> (pairthem '(1 2 3 4) '(a b c d e))

((1 a) (2 b) (3 c) (4 d))

> (pairthem '(1 2 3 4 5) '(a b))

((1 a) (2 b))

(b) Ocaml. [2pt]

# pairthem [1;2;3] ['a';'b';'c';'d'];;

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

# pairthem [1;2;3;4;5] ['a';'b'];;

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

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 not to be avoided in a pure functional language?

(A) goto

(B) lambda

(C) loops

(D) variables

2. Which languages do not have parametric polymorphism?

(A) C++

(B) Java

(C) Ocaml

(D) Smalltalk

3. What is the expected running time of fold left and fold right on a

list of length $n$?

(A) fold left $ O ( 1 ) $; fold right $ O ( 1 ) $

(B) fold left $ O ( 1 ) $; fold right $ O ( n ) $

(C) fold left $ O ( n ) $; fold right $ O ( 1 ) $

(D) fold left $ O ( n ) $; fold right $ O ( n ) $

4. What is the required amount of stack space for fold left and fold

right on a list of length $n$?

(A) fold left $ O ( 1 ) $; fold right $ O ( 1 ) $

(B) fold left $ O ( 1 ) $; fold right $ O ( n ) $

(C) fold left $ O ( n ) $; fold right $ O ( 1 ) $

(D) fold left $ O ( n ) $; fold right $ O ( n ) $

5. If guess is a predicate that searches a database to return one of

its elements, and verify checks to see if the thing found is good,

then we may define the predicate find, which returns a valid entry

from the database as:

(A) find(X) :- guess(X), verify(X).

(B) find(X) :- guess(X).

find(X) :- verify(X).

(C) find(X) :- verify(X), guess(X).

(D) guess(X) :- find(X), verify(X).

6. What is the type of List.map?

(A) ('a -> 'b -> 'a) -> 'a -> 'b list -> 'a

(B) ('a -> 'b -> 'b) -> 'a list -> 'b -> 'b

(C) ('a -> 'b) -> 'a list -> 'b list

(D) ('a -> bool) -> 'a list -> 'a list

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

(A) The message +4 is sent to the object 3.

(B) The message 3 is sent to the function +, the result of which

is a function to which the message 4 is sent.

(C) The same as (+)3 4.

(D) The tuple message (3,4) is sent to the object +.

8. What is the type of the Ocaml function

let g () = 3;;

(A) int \* unit

(B) int -> unit

(C) unit \* int

(D) unit -> int

9. Unification is a part of the static type checking algorithm used

by compilers of?

(A) C and C++

(B) Java and Smalltalk

(C) ML and Ocaml

(D) Perl and Python

10. When a garbage collector forms the closure of the root set, it

identifies all \_\_\_ objects on the heap.

(A) dead

(B) live

(C) reachable

(D) unreachable

11. A C++ compiler does object-oriented dynamic dispatch via:

(A) default memory allocator

(B) help-allocated closure

(C) inferential type indicator

(D) virtual function table

12. ``Structured Programming with goto Statements''

(A) Edsger Dijkstra

(B) C.A.R. Hoare

(C) Donald Knuth

(D) Niklaus Wirth

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. Which language is not completely strongly typed?

(A) C++

(B) Perl

(C) Scheme

(D) Smalltalk

2. What is a comment in Ocaml?

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

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

(C) //...

(D) ;;...

3. What kind of garbage collection fails on cyclic data structures?

(A) copying with semispaces

(B) generational

(C) mark and sweep

(D) reference counting

4. In Ocaml, what is the type of (/.)?

(A) float \* float \* float

(B) float \* float -> float

(C) float -> float \* float

(D) float -> float -> float

5. In Java or C++, which statement can cause control to pass from the

current function to the calling function, or perhaps the caller of

the caller, or perhaps even all the way back to the main function?

(A) break

(B) continue

(C) return

(D) throw

6. A static (access) link is:

(A) a pointer to the instruction which called the current

function.

(B) a pointer to the next free byte of storage on the heap.

(C) a pointer to the stack frame in which the current function is

nested.

(D) a pointer to the stack frame of the caller of the current

function.

7. Which is a fully curried lazy functional language with

overloading?

(A) Haskell

(B) Ocaml

(C) Scheme

(D) Smalltalk

8. Given the Smalltalk statement a:=[:x|x+1].

What expression would return the number 4?

(A) 3 to: a.

(B) a 3.

(C) a at: 3.

(D) a value: 3.

9. In Perl, what will print the current date and time?

(A) print "date";

(B) print 'date';

(C) print (date);

(D) print `date`;

10. What is not false in Perl?

(A) 0

(B) 0.0/0.0

(C) ""

(D) undef

11. In Smalltalk, what is 9?

(A) (4 + 5) value.

(B) [4 + 5] value.

(C) "4 + 5" value.

(D) {4 + 5} value.

12. ``Go To Statement Considered Harmful''

(A) Edsger Dijkstra

(B) C.A.R. Hoare

(C) Donald Knuth

(D) Niklaus Wirth