

UNIVERSITY OF TORONTO
Faculty of Arts and Science
St. George Campus

DECEMBER 2003 EXAMINATION

CSC 324h F
Duration – 3 hours

Examination Aids: *1 handwritten sheet*

Student #: _____

Last Name: _____ First Name: _____

Do **not** turn this page until you have received the signal to start. In the meantime, please fill out the identification section above, and read the instructions below carefully.

This term test consists of 10 questions on 12 pages (including this one), printed on one side of the paper. When you receive the signal to start, please make sure that your copy of the examination is complete.

#1: _____/10

#2: _____/10

#3: _____/10

Answer each question directly on the examination paper, in the space provided, and use the reverse side of the pages for rough work. If you need more space for one of your solutions, use the reverse side of the page and indicate clearly the part of your work that should be marked.

#4: _____/10

#5: _____/15

#6: _____/15

In the programming questions, you do not have to write pre-conditions/post-conditions. Also, you do not have to comment your code but use **meaningful** variable names. You are allowed to use helper functions. You are not allowed to use any of the imperative features of Scheme or Prolog. Use recursion and patterns where possible.

#7: _____/20

#8: _____/22

#9: _____/18

Be aware that concise, well thought-out answers will be rewarded over long rambling ones. Also, unreadable answers will be given zero (0) so write legibly.

#10: _____/30

Total: _____/160

GOOD LUCK!

Question 1: Prolog unification [10 marks total]

For each of the following pairs of Prolog terms, indicate whether they unify, and if so, what variable bindings result from the unification:

(a) [1 mark] $f(X,Y)$ and $f(Z)$

(b) [3 marks] $p(b,f(X,X),c)$ and $p(U,f(U,V),V)$

(c) [3 marks] $f(X,h(b,X))$ and $f(g(P,a),h(b,g(Q,Q)))$

(d) [3 marks] $g(X,f(X,W),X)$ and $g(Y,Y,f(Z,Y))$

Question 2: ML inference [10 marks total]

(a) [3 marks] What is the type inferred by ML for the function sum2?

```
fun sum [] = 0
  | sum (x::xr) = x + sum xr;

fun sum2 [] = 0
  | sum2 (xs::xsr) = sum xs + sum2 xsr;
```

(b) [3 marks] What is the type inferred by ML for the function map2?

```
fun map f [] = []
  | map f (x::xr) = f x :: map f xr;

fun map2 g [] = []
  | map2 g (xs::xsr) = (map g xs) :: map2 g xsr;
```

(c) [4 marks] What is the type inferred by ML for the function foo?

```
fun foo([]) = []
  | foo([a]) = [a]
  | foo (a::b::rest) =
      if (a = b) then
        foo(b::rest)
      else
        a::foo(b::rest);
```

Question 3: Grammar [10 marks total]

Consider following BNF grammar. The start symbol is <Pedigree>

```
<Pedigree> → <Name>
<Pedigree> → <Name><Parents>
<Parents>  → <Person><Person>
<Person>  → () | (<Pedigree>)
<Name>    → <Letter> | <Letter><Name>
<Letter>  → A | ... | Z | a | ... | z
```

For each of the following, indicate whether the string belongs to the language recognized by this grammar or not and if it does belong to the language, give a derivation:

(a)[2 marks] Charles ()

(b)[2 marks] Charles (Elizabeth)

(c)[2 marks] Charles (() Philipp)

(d)[2 marks] Charles () (Philipp)

(e)[2 marks] Charles (Elizabeth) (Philipp)

Question 4: Scheme lists [10 marks total]

Alice travels a lot and keeps her travel plans in following form:

```
(define journeys
  '(
    ("June 5" "Stuttgart" "Ulm" "Augsburg" "Munich")
    ("June 6" "Munich" "Salzburg" "Vienna")
    ("June 7" "Vienna" "Bratislava" "Prague")
    ("June 8" "Prague")
    ("June 9" "Prague" "Dresden" "Leipzig")
  )
)
```

Define a Scheme function **get-dates-of-visits** that, given such a list and the name of a town, returns a list of dates on which Alice visits that town. *Sample invocations:*

```
]=> (get-dates-of-visits journeys "Vienna")
("June 6" "June 7")
]=> (get-dates-of-visits journeys "Prague")
("June 7" "June 8" "June 9")
```

Question 5: Scheme recursion [15 marks total]

(a) [2 marks] Here is one way to find the Nth Fibonacci number in Scheme:

```
(define (fib n)
  (cond ((= n 0) 0)
        ((= n 1) 1)
        (else
         (+ (fib (- n 1)) (fib (- n 2))))))
```

What's wrong with this definition?

(b) [13 marks] Write a different definition of **fib** that finds the nth Fibonacci number in linear time.

Question 6: Scheme high-order [15 marks total]

(a) [10 marks] Define a function **ncompose** in ML. **ncompose** takes a pair of a function, *f*, and a number, $n \geq 0$ as input, and returns a function that applies *f* to its argument *n* times. *Sample invocation* :

```
> fun add1 x = x+1;  
> val add5 = ncompose(add1,5);  
> add5 10;  
- val it = 15: int
```

(b) [5 marks] What type will ML infer for **ncompose**?

Question 7: ML recursive types [20 marks total]

(a) [2 marks] Define an ML datatype, **tree2**, for binary trees in which each node and each edge stores a weight, i.e., a real number (positive, negative, or zero). Each node may have 0, 1 or 2 children.

(b) [1 mark] Draw such a tree and show how it is represented in ML using your datatype. Your tree should include nodes with 0, 1 and 2 children.

(c) [8 marks] Define an ML function, **minNum**, of type `tree2 -> real`, that returns the value of the smallest real number stored in a binary tree.

(d) [9 marks] Define a path in a binary tree to be a sequence of nodes and edges from the root of the tree to a leaf. The weight of the path is the sum of the weights of all the nodes and all the edges on the path. Define an ML function, **minPath**, of type `tree2->real`, that returns the weight of a lightest path in a binary tree.

Question 8: ML references [22 marks total]

(a) [2 marks] Define a polymorphic ML type, 'a llist, for linked lists. You should use references so you can insert (or remove) an element into the middle of a list without copying any of the list structure.

(b) [1 mark] How would you represent the list ["abc", "def", "123"] using your datatype?

(c) [9 marks] Define an ML function, **remove**, of type $\text{int} * ('a \text{ llist ref}) \rightarrow 'a$, that removes and returns the Nth element of a list, for $N \geq 1$. Raise an exception if the list does not have at least N elements. (*Hint*: use a recursive helper function.)

(d) [10 marks] Define an ML function, **insert**, of type $\text{int} * 'a * ('a \text{ llist ref}) \rightarrow \text{unit}$, that inserts an element into a list immediately after the Nth element, for $N \geq 0$. Raise an exception if the list does not have at least N elements. (*Hint*: use a recursive helper function.)

Question 9: ML exception handling [18 marks total]

(a) [1 mark] Define a type called **employee** for employee records, where each record has three fields: id, name and salary, of type int, string and real, respectively.

(b) [5 marks] Define a function, **retrieve(I,L)**, that returns the record of the employee whose id is I from a list of employee records, L. If no such employee is in the list, then raise an exception.

(c) [1 mark] What is the type of retrieve?

(d) (10 marks) Define a function called **retrieveAll(List1,List2)**. Here, List1 is a list of employee id's, and List2 is a list of employee records. For each id in List1, the retrieveAll function uses the retrieve function defined above to retrieve a record from List2. It also forms a list of all the retrieved records. In addition, it catches all exceptions raised by the retrieve function, and forms a list of all id numbers that caused exceptions. Finally, it returns a pair of lists, the list of retrieved employee records, and the list of exception-causing id numbers.

(e) [1 mark] What is the type of retrieveAll?

Question 10 : Prolog [30 marks total]

In chess, a queen piece can move any number of squares up or down in its column, any number of squares left or right in its row, or diagonally --- one row up or down for every column left or right.

The N-Queens Puzzle is solved by placing N queens on an N x N chessboard so that no queen can take any other queen in a single move. This means that no two queens are in a single row, a single column, or along any diagonal extent (not just between the corners of the chessboard).

Write a Prolog program that solves the N-queens puzzle for N=4. We will not mark your solution according to efficiency --- only correctness.

Hint 1: Because any solution must have exactly one queen in every column and exactly one queen in every row, the positions of all 4 queens on the 4 x 4 chessboard can be represented by any permutation of the numbers 1, 2, 3, and 4. In any such permutation, we understand that the first number provides the column number of the queen in the first row, the second number provides the column number of the queen in the second row, etc. The permutation 3-1-4-2, for example, corresponds to the solution:

```

__ Q _ 3
Q _ _ _ 1
_ _ _ Q 4
_ Q _ _ 2

```

Hint 2: Not every such permutation is a solution --- you must still check that no two queens lie along the same diagonal. What arithmetic relationship exists among two pairs of coordinates, (R_i,C_i) and (R_j,C_j), when they corresponding pieces lie along the same upper-left-to-lower-right diagonal? What relationship exists along the same lower-left-to-upper-right diagonal?

Hint 3: To generate all possible permutations, you will find this predicate helpful:

```

select(X,[X|Rest],Rest).
select(X,[H|T],[H|Rest]) :- select(X,T,Rest).

```

select/3 non-deterministically selects an element from a list, and returns the rest.

END OF EXAMINATION