2010年度プログラミング言語II（計算機言語I）定期試験

2011.01.31.

国島丈生

1. 型、パターン

1. Standard MLの式 (#"a", (true, [1, 2, 3], 95.0), ["string", "integer"]) の型を示せ。（10点）
2. 1. の式をパターン (x, (y, z, w), v::vs) に照合させるとき、各変数に束縛される値を示せ。（10点）

2. 関数

下記の関数をStandard MLで実装せよ。

1. リスト L から、2番目の要素を取り除いたリストを得る関数 remove_second(L): 'a list -> 'a list。 L の要素数が1以下の場合はnilを返すものとする。例えば remove_second([1, 3, 4, 2, 5]) の評価結果は [1, 4, 2, 5] となる。リストの長さを求める下記の関数 length は宣言なしに用いてよい。（20点）
```
fun length(nil) = 0
  | length(x::xs) = 1 + length(xs);
```
2. 2次元ベクトル (x, y) に対し、その長さ l は $l=\sqrt{x^2+y^2}$ で定義される。
   a) 2次元ベクトル(x, y) （x, yは実数）の長さを求める関数 vec_length: real * real -> real を実装せよ。実数の平方根を求める関数 sqrt: real -> real は宣言なしに用いてよい（例えば sqrt(4.0) の評価結果は 2.0 となる）。（5点）
   b) 2次元ベクトルのリスト $[(x_1, y_1), (x_2, y_2), ...]$ （ $x_1, y_1, x_2, y_2, ...$ は実数）に対し、各要素のベクトルの長さからなるリストを求める関数 vec_length_list: (real * real) list -> real list。例えば vec_length_list([(1.0, 0.0), (3.0, 4.0), (5.0, 12.0)]) の評価結果は [1.0, 5.0, 13.0] となる。前問の関数 vec_length を使用してよい。（15点）
3. リストの1番目の要素と2番目の要素、3番目の要素と4番目の要素、…をすべて入れ替える関数 swap: 'a list -> 'a list。例えば swap([1, 2, 3, 4, 5]) の評価結果は [2, 1, 4, 3, 5] となる。（20点）

3. 高階関数、データ型

下記の2問のうち、いずれかを選択して答えよ。両方解答してもよいが、その場合は点数の高かったもののみを合計点に加算する。（20点）

1. リスト $L=[x_1, x_2, \ldots, x_n]$ について、その要素 $x_i (1 \leq i \leq n)$ を、関数 $p(x_i)$ が true になるものと false になるものとに分割する高階関数 partition(p, L): ('a -> bool) * 'a list -> 'a list * 'a list を実装せよ。例えば、partition(fn x => x < 10, [1, 50, 3, 444, 20]) の結果は ([1, 3], [50, 444, 20]) となる。

2. 下記のデータ型 'a btree は 2 分木を表す。このとき、2 分木 T 中に、ラベル x をもつ節点が含まれるかを返す関数 find_in_btree(x, T): 'a * 'a btree -> bool を実装せよ。例えば find_in_btree(1, Node(4, Node(1, Empty, Empty), Node(2, Empty, Empty)) は true となる。（ T は 2 分探索木とは限らないことに注意せよ）

   datatype 'a btree = Empty | Node of 'a * 'a btree * 'a btree;

解答例

1. 型

1. char * (bool * int list * real) * string list
2. x: #”a”, y: true, z: [1, 2, 3], w: 95.0, v: “string”, vs: [“integer”]

2. 関数

1. fun remove_second(L) =
   if length(L) < 2 then nil else hd(L) :: tl(tl(L));
2.
   a) fun vec_length(x, y) = sqrt(x * x + y * y);
   b) fun vec_length_list(nil) = nil
      | vec_length_list((x, y)::zs) = vec_length(x, y)::vec_length_list(zs);
3. fun swap(nil) = nil
   | swap([x]) = [x]
   | swap(x::y::zs) = y::x::swap(zs);

2-2-a): sqrt 関数の引数が real 型でなければならないため、x, y も real 型と決まります。

2-2-a)で用いた関数 sqrt は、Standard ML の標準ライブラリに含まれています（関数 Math.sqrt）。したがって、実際に実行してみる場合は、sqrt を Math.sqrt に置き換えてください。

3. 高階関数、データ型

1. fun partition(p, nil) = (nil, nil)
   | partition(p, x::xs) =
   let
   val (ys, zs) = partition(p, xs)
   in
   if p(x) then (x::ys, zs)
   else (ys, x::zs)
   end;
2. fun find_in_btree(x, Empty) = false
   | find_in_btree(x, Node(a, l, r)) =
   x = a orelse find_in_btree(x, l) orelse find_in_btree(x, r);