

平成21年度プログラミング言語II(計算機言語I) 定期試験

以下の問はすべて Standard ML を用いて答えよ。

1. 型と式

次の型を持つ式の例をそれぞれ1つ挙げよ。(20点)

1. `bool list * (int * (int * int)) list`
2. `string * (int -> int)`

2. 環境

次に示す式を順に実行したとする。このとき、最後の式 `f(4) + g(6)` の評価値は何になるか。答のみ示せばよい。(10点)

```
val a = 2;
fun f(b) = a * b;
val b = 3;
fun g(a) = a + b;
f(4) + g(6);
```

3. 関数

次の関数を実装せよ。

1. 平面座標のリストが与えられるとき、その最初の点のy座標を求める関数 `firsty: (real * real) list -> real`。各座標値は実数であるものとする。また空リストの場合は考えなくてよい。例えば、`firsty([(32.1, 2.5), (22.7, 4.3), (0.0, 0.0)])` を評価すると 2.5 になる。(10点)
2. 指定された要素がリスト中に何回出現するかを求める関数 `count: 'a * 'a list -> int`。例えば `count("#a", ["#b", "#z", "#a", "#a", "#p"])` を評価すると 2 になる。高階関数を用いずに実装すること。(20点)
3. 2. の関数 `count` を、次に示す高階関数 `filter` を用いて実装せよ。このとき、`filter`、およびリストの要素数を求める次の関数 `length` は定義なしに用いてよい。(20点)

```
fun filter(f, nil) = nil
  | filter(f, x::xs) = if f(x) then x::filter(f, xs)
                      else filter(f, xs);

fun length(nil) = 0
  | length(x::xs) = 1 + length(xs);
```

4. データ型

整数の集合を次のデータ型で表現することを考えよう。

```
datatype e = Intg of int | Intv of (int * int);  
type intset = e list;
```

ここで $\text{Intg}(n)$ は集合 $\{n\}$ (n のみからなる集合)を、 $\text{Intv}(m, n)$ は集合 $\{m, m+1, \dots, n-1, n\}$ ($m \leq n$)をそれぞれ表す。

`intset`型の値(すなわち `Intg` や `Intv` のリスト)は、リストの要素の和集合を表すものとする。例えば、`[Intg(1), Intv(3, 5), Intg(7)]` は $\{1\} \cup \{3, 4, 5\} \cup \{7\} = \{1, 3, 4, 5, 7\}$ を表す。

このとき、ある整数が与えられた`intset`中に含まれるかを判定する関数 `inv: int * intset -> bool` を実装せよ。例えば `inv(4, [Intg(1), Intv(3, 5), Intg(7)])` を評価すると `true` になる。(20点)

平成21年度プログラミング言語II(計算機言語I)定期試験解答例

1. 型と式

1. ([true, false], [(1, (2, 3)), (4, (5, 6))])
2. ("string", fn x => x+1)

2. 環境

17

$f(b) = a * b = 2 * b$ (val a = 2より)、 $g(a) = a + b = a + 3$ (val b = 3より)なので、 $f(4) + g(6) = 2 * 4 + (6 + 3) = 17$ となります。

3. 関数

1.

```
fun firstly(L:((real * real) list)) = #2(hd(L));
```

解答例では、問題文に合わせるように引数の型を指定しました。

2.

```
fun count(x, nil) = 0
  | count(x, y::ys) = if x = y then 1 + count(x, ys) else count(x, ys);
```

3.

```
fun count(x, L) = length(filter(fn y => x = y, L));
```

4. データ型

```
fun inv(x, nil) = false
  | inv(x, Intg(n)::xs) = (x = n) orelse inv(x, xs)
  | inv(x, Intv(m, n)::xs) = (m <= x andalso x <= n) orelse inv(x, xs);
```