

2006 年度「計算機言語 I」定期試験問題

担当教員: 国島丈生

2007-02-05

以下の問はすべて Standard ML を用いて考えよ。

1 型と式 (40 点)

1. 次の式を評価した値とその型を示せ。誤った式の場合はその理由を述べよ。(各 5 点)

(a) $(1, 3)::[(2*3, 4 \text{ div } 2), (6 \text{ mod } 3, 1)]$

(b) $\text{tl}(\text{hd}([\text{hd}([2,3], \text{nil}), [4]]))$

(c) $[\#1(1, 2, 3), \#3(1,2,3), \#4(1,2,3)]$

2. 次の型に属する式の例を示せ。(各 5 点)

(a) `bool list list`

(b) `(bool * string) * (bool * string)`

(c) `(int * string) list * int * real list`

3. 次の関数の型を示し、どのような型推論を経て決定できるのか述べよ。(10 点)

```
fun test(c, e) = if c then e else 0;
```

2 関数 (50 点)

次の関数を実装せよ。

1. 要素数 2 以上のリスト L について、 L の先頭・2 番目の要素からなるリストを返す関数 `hdd`:

`'a list → 'a list`。例えば `hdd([3,6,2,4])` は `[3,6]` となる。再帰関数である必要はない。また、 L の要素数が 1 以下であった場合のエラー処理は考えなくてよい。

2. 整数のリスト L について、奇数であるような要素からなるリストを得る再帰関数 `odd`: `int`

`list → int list`。例えば `odd([4,3, 7])` は `[3, 7]` となる。

3. 文字列 s と整数 n を引数とし、 s を n 回連結した文字列を得る再帰関数 `pow`: `string * int`

`→ string`。例えば `pow("abc", 3)` は `"abcabcabc"` となる。

4. 平面座標値のリストを入力とし、その x 座標の和、 y 座標の和を組として返す再帰関数

`sumPoints`: `(int * int) list → (int * int)`。例えば `sumPoints([(1,2), (3,4), (5,6)])` は `(9, 12)`

となる。

5. 次のデータ型で2分木を表すこととする。整数をラベルとする2分木について、すべてのラベルを2倍して得られる2分木を返す再帰関数 `double: int btree → int btree`。例えば `double(Node(2, Node(4, Empty, Empty), Empty))` は `Node(4, Node(8, Empty, Empty), Empty)` となる。

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

3 高階関数 (10 点)

整数のリスト L について、要素の絶対値の最大値を求める関数 `maxabs: int list → int` を、下記の高階関数を用いて実装せよ。なお、整数の絶対値を得る関数 `abs: int → int` は定義なしに用いてよい。

```
fun map(f, nil) = nil  
  | map(f, x::xs) = f(x) :: map(f, xs);  
  
exception empty;  
fun reduce(f, nil) = raise empty  
  | reduce(f, [x]) = x  
  | reduce(f, x::xs) = f(x, reduce(f, xs));
```