

2004年度 計算機言語I 追試験問題

国島丈生

2005-02-14

1. 次の式の型を示せ。ただし、式として間違っている場合はその理由を述べよ。(各5点)

- (a) `[[], #'c"], ([5,9], #'z")]`
- (b) `#2((6,10), (5.0, true), (7, #'a"))`
- (c) `hd(tl(["string", "integer"]))`
- (d) `if 1<2 then nil else [6.0,7.1]`

2. 次の関数の型を示し、型推論の過程を述べよ。(各10点)

```
fun f(0, y) = y
|   f(x, y) = f(x-1, x*y);
```

3. 次の関数を ML で実装せよ。(各10点)

- (a) リストの末尾2つの要素からなるリストを得る関数 `suffixTwo`。ただしリストを逆順に並び替える関数 `reverse` は定義することなく用いてよい。
- (b) 組のリスト $[(x_1, y_1), \dots, (x_n, y_n)]$ から、リストの組 $[[x_1, \dots, x_n], [y_1, \dots, y_n]]$ を得る関数 `unzip`: `('a * 'b) list → 'a list * 'b list`。たとえば `unzip([(1, "a"), (2, "b")]) = ([1,2], ["a","b"])` となる。
- (c) 次のデータ型で2分木を表現するとき、2分木に含まれる節のラベルを帰りがけ順に並べたリストを得る関数 `postorder`: `'a btree → 'a list`。

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

- (d) リストのリストの要素数を求める関数 `nestnumber`: `'a list list → int`。たとえば `nestnumber([[1,2], nil, [3,4,5]]) = 5` となる。
- (e) リスト L 中の要素がすべて述語 P が満たすかどうか判定する関数 `all`: `('a → bool) * 'a list → bool`。たとえば、引数 x が正なら `true` を返す関数 `positive(x)` について、`all(positive, [1, 2, 1, 3, 5]) = true` となる。ただし、`all(P, nil)` は P に関わらず `true` とする。

(裏に続く)

4. 一般の木では、すべての節点について、子の数は制限されない。そのため、子節点をリストで表現することにより、一般の木を次のようなデータ型で書くことができる。

```
datatype 'l tree = Node of 'l * 'l tree list;
```

- (a) `real tree` の型を持つ式の例を示せ。(10 点)
- (b) `real tree` の節点のラベルの総和を求める関数 `sum: real tree → real` を実装せよ。(10 点)