

**2000年度計算機言語I定期試験**

以下、とくに断らない限り、実装はMLで行うこと。

**問題 1**

1. 次の式の型を示せ。

(a)  $((3.5, "a"), (2.5, "char")), \#"a"$

(b)  $(\#"a", \#"b"), [nil, [1, 2, 3]]$

2. 次の型に属する値の例を示せ。

(a)  $(int * char) list$

(b)  $string list * (int * (real * string)) * int$

**問題 2** 次の関数を再帰を用いて実装せよ。

1. 次の漸化式で与えられる数列(フィボナッチ数列)の第  $n$  項の値を求める関数  $fibonacci(n): int \rightarrow int$ 。

$$f(1) = 1, f(2) = 1, f(n) = f(n-1) + f(n-2) \quad (n \geq 3)$$

2. 文字列リストの  $n$  番目の値を求める関数  $nth(): string list * int \rightarrow string$ 。たとえば、 $nth(["a", "string", "int"], 2)$ の結果は"string"となる。

3. 整数のリストのリストを整数のリストに変換する関数  $flatten(): int list list \rightarrow int list$ 。たとえば、 $flatten([[1, 6, 8], [2, 4], nil, [3]])$ の結果は  $[1, 6, 8, 2, 4, 3]$ となる。

(裏面に続く)

問題3 2分探索木とは、ノードのラベルが以下に述べる属性に従う二分木である。 $x$  を二分探索木中のノード  $n$  のラベルとすると、 $n$  の左部分木中のすべてのラベル  $y$  について  $y < x$ 、 $n$  の右部分木中のすべてのラベル  $y$  について  $x < y$  が成り立つ。ただし、比較演算  $<$  は以下の性質を満たすものである。

- $x < y, y < z$  ならば  $x < z$  (推移性)
- $x \neq y$  ならば、 $x < y$  または  $y < x$  (比較可能性)
- いかなる  $x$  についても、 $x < x$  は成立しない (非反射性)

2分探索木に関する以下の問いに答えよ。実装には図1の関数を用いてかまわない。

1. 2分探索木に含まれるノードを昇順 (小さいほうから順) に整列したリストを得るには、木をどのような順序で探索すればよいか述べよ。
2. ノードのラベルが文字列であるような2分探索木を図1のように実装した。この実装では、データ型 `btree` はラベルとしてどのような型の値でも持つことができる。しかし関数 `insert()` は `string * string btree -> string btree` という型の関数だと判断される。`insert()` に関するMLの型推論の過程を説明せよ。
3. 1の整列を行う関数 `binTreeToList(): string btree -> string list` を実装せよ。
4. 文字列のリストの要素をすべて2分探索木に格納する関数 `listToBinTree(): string list -> string btree` を実装せよ。
5. 与えられた文字列のリストを2分探索木を用いて昇順に整列する関数 `binTreeSort(): string list -> string list` を実装せよ。

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

(* btree の例 *)
val t = Node("ML",
            Node("as",
                Node("a", Empty, Empty),
                Node("in", Empty, Empty)),
            Node("types", Empty, Empty));

fun lower(nil) = nil
  | lower(c::cs) = (Char.toLower c)::lower(cs);

fun lt(x, y) =
  implode(lower(explode(x))) < implode(lower(explode(y)));

fun insert(x, Empty) = Node(x, Empty, Empty)
  | insert(x, T as Node(y, left, right)) =
  if x=y then T
  else if lt(x, y) then Node(y, insert(x, left), right)
       else (* lt(y, x) *) Node(y, left, insert(x, right));
```

図1: MLによる2分探索木の実装