

# 平成19年度「プログラムの数理」期末試験

平成19年2月4日 8:30 - 10:00

工学部6号館63号室

以下の問いに答えよ。

1. 以下のリスト上の関数が単同型 (list homomorphism) であることを示せ。

- (a) リスト上の恒等関数  $id$   
(例えば、 $id [1, 3, 2] = [1, 3, 2]$  である。)
- (b) リストのリストを連結する関数  $concat$   
(例えば、 $concat [[1, 2, 3], [4, 5], [6]] = [1, 2, 3, 4, 5, 6]$  である。)
- (c) リストのすべての連結する先頭部分リストを長さの増加する順に並べたリストを返す関数  $inits$   
(例えば、 $inits [1, 2, 3] = [[], [1], [1, 2], [1, 2, 3]]$  である。)
- (d) リストの要素数が偶数であるかどうかを判定する関数  $evenlength$   
(例えば、 $evenlength [1, 2, 3] = False$ ,  $evenlength [1, 2, 3, 4] = True$  である。)

2. 関数  $remDups$  は、リストから隣接して重複している要素を取り除く。例えば、

```
remDups [1, 2, 2, 3, 3, 3, 1, 1] = [1, 2, 3, 1]
```

である。

- (a)  $remDups$  が一つの単同型関数で表現できないことを証明せよ。
  - (b)  $remDups = \pi_2 \cdot h$  を満たすような単同型関数  $h$  が存在することを示せ。
  - (c) 関数  $remDups$  を  $left\ reduction$  (または  $foldl$ ) を用いて定義せよ。
3. 関数  $choose\ k\ xs$  は長さ  $k$  であるような  $xs$  のすべての部分リストのリストを返す。例えば、

```
choose 3 [1, 2, 3, 4] = [[1, 2, 3], [1, 2, 4], [1, 3, 4], [2, 3, 4]]  
choose 3 [1, 2]      = []
```

である。Haskell を用いて関数  $choose$  の再帰的な定義を与えよ。

4. 次の関数を考える。

```
lnt = 1 / # * . asc < . tails
```

ここで、関数  $asc :: [Int] \rightarrow Bool$  は、数列が非減少であるかどうかを判定する。例えば、 $asc [1, 2, 2, 3, 4] = True$ ,  $asc [1, 2, 3, 2] = False$  である。

- (a) 関数  $lnt$  の型を与えよ。
- (b) 関数  $lnt$  が何を計算するかを例を用いて説明せよ。
- (c) 等式

```
lnt = 1 / . f * . tails
```

を満たすような関数  $f$  を導出せよ。

- (d) 関数  $lnt$  が次のように定義できることを証明せよ。

```
lnt = \pi_1 \cdot \odot f^{(0, \infty)}
```

where

```
(e_1, e_2) \odot a = if e_2 \le a then (e_1 + 1, a) else (1, a)
```

5. 漸成的アルゴリズム論に基づくプログラミング手法のポイントを例を用いて説明せよ。