

プログラミング・パラダイム (2012年度)・テスト問題用紙

(2012年 11月 29日 (木)・ 9:35 ~ 10:20)

解答上、その他の注意事項

- I. 問題は、問 I~II までである。
- II. 解答用紙の右上の欄に学籍番号・名前を記入すること。
- III. ノート・プリント・参考書などは持ち込み可である。
- IV. 携帯電話などの通信機能を持つものは 持ち込み不可 である。
- V. 問 I を解答するときのみ、ノート PC を使用して良い。ネットワークに接続して WWW を閲覧しても良いが、掲示板、チャット、メールなどで生身の人間と通信することは禁じる。
- VI. テストの配点は 50 点である。合格はレポートの得点を加えて、100 点満点中 60 点以上とする。

- (1) 引数として与えられる整数のリスト中の 5 の倍数の個数を返す関数

```
foo :: [Integer] -> Integer
```

を定義せよ。

例えば、foo [0,1,5,9] は 2 であり、foo [2,3,4,6] は 0 である。

この問では map, filter, foldl, foldr などのリストに関するライブラリ関数や内包表記を使わず、if ~ then ~ else ~ 式や算術演算子、論理演算子、比較演算子、パターンマッチング、再帰などを使って定義せよ。なお、Haskell で “余り” を表す算術演算子は、‘mod’ である。例えば 7 ‘mod’ 4 (あるいは mod 7 4) の値は 3 である。

- (2) 整数 n を引数として受け取り、正の整数の組 (i, j) で、 $0 \leq i \leq 2n$ かつ $0 \leq j \leq 2n$ かつ $(i-n)^2 + (j-n)^2 \leq n^2$ が成り立つものを列挙する関数

```
bar :: Integer -> [(Integer,Integer)]
```

を (リストの内包表記を用いて) 定義せよ。

例えば、bar 1 は [(0,1), (1,0), (1,1), (1,2), (2,1)] で、bar 2 は [(0,2), (1,1), (1,2), (1,3), (2,0), (2,1), (2,2), (2,3), (2,4), (3,1), (3,2), (3,3), (4,2)] となる。
(リストの要素の順番はこの通りでなくても良い。)

なお、 m から n まで (ただし $m \leq n$) の整数のリストは、[m..n] という式で得ることができる。例えば、[0..5] は [0,1,2,3,4,5] である。

II. (Haskell)

(12点×2)

次の例にならって、下の Haskell の式 (1)~(2) を評価した結果を書け。

例: `take 5 (from 1)` ⇒ 評価結果: `[1,2,3,4,5]`

ただし、`take` と `from` は講義プリントに定義されている通りの関数である。

```
from :: Integer -> [Integer]
from n = n : from (n+1)

take :: Integer -> [a] -> [a]
take 0 _      = []
take _ []     = []
take n (x:xs) = x : take (n-1) xs
```

(1) `takeWhile (< 50) (map (\ x -> x*x) (from 1))`

この問で使用されている関数 `map`, `takeWhile` の定義は次の通りである。

```
map :: (a -> b) -> [a] -> [b]
map f []      = []
map f (x:xs) = f x : map f xs

takeWhile :: (a -> Bool) -> [a] -> [a]
takeWhile _ []      = []
takeWhile p (x:xs) = if p x then x : takeWhile p xs else []
```

(2) `[(x,y) | x <- [1,2,3], y <- [2,4,6], 2*x /= y]`

(この問に関してはリスト内の順番のみの間違いは、減点はしない。なお、`/=`演算子は“not equal”である。(C言語の `!=`演算子に相当する。))

