

# 非手続き型言語9回目課題 解答例

May 25, 2020

## 1 問題 1

資料演習問題 5.4.7 c) で空の文字列があっても動くようにする。例えば [ “abc” , “def” , “ba” , “aa” , ”” ] を入力しても動くようにするにはどうしたらいいかを解答せよ。

解答例:関数 filter が以下のように定義されているとする。

```
fun filter (P, []) = []  
|   filter (P, x::xs) = if P (x) then x::filter(P, xs) else filter(P, xs);
```

ここで例えば以下のようにすればよい。

```
filter( fn s => if s = "" then false  
else hd(explode(s)) = #"a", ["abc", "def", "ba", "aa", ""]);
```

## 2 資料演習問題 5.4.11

解答例:

```
fun reduceB(g,F,[]) = g  
|   reduceB(g,F,x::xs) = F(x, reduceB(g,F,xs));
```

注釈:関数 reduce と同じように例外を定義してリストが空リストの時に例外を発生させている解答もあるが、この関数は関数 foldr の導入を見据えたもので、定数 g というのは初期値 (デフォルト値) の役割をしている。だからリストが空リストの時は g を返すようにする関数を書いて欲しいというのが問題の意図。

## 3 資料演習問題 5.4.12

a) リストの長さ

```
reduceB(0, fn (x,y)=>y+1, [1,2,3]);
```

b) リストの接尾辞のリスト

```
reduceB([nil], fn (x,y)=>(x::hd(y))::y, [1,2,3]);
```

解説:接尾辞のリスト. 後ろから追加していく. 初期値は `[nil]` それに `[3]` を加えて `[2,3]` を加えて `[1,2,3]` を加えるということをするように考える.

解答例の関数は以下のように動く. `[1,2,3]` に対し初期値を `[nil]` とする

- 最初は  $y = [\text{nil}]$ ,  $x = 3$  でくるので  $(x::\text{hd}(y))$  は `[3]` で  $(x::\text{hd}(y))::y$  は `[[3], nil]`
- 次は  $y = [[3], \text{nil}]$ ,  $x = 2$  でくるので  $(x::\text{hd}(y))$  は `[2,3]` で  $(x::\text{hd}(y))::y$  は `[[2,3],[3],nil]`
- 次は  $y = [[2,3], [3], \text{nil}]$ ,  $x = 1$  でくるので  $(x::\text{hd}(y))$  は `[1,2,3]` で  $(x::\text{hd}(y))::y$  は `[[1,2,3],[2,3],[3],nil]`