

非手続き型言語 11 回目課題 解答例

June 1, 2020

1 演習問題 5.5.6

まず、部分並びの関数 `subsequence` を作成する。作成する方針は以下である。

1. まず二つの文字列を関数 `explode` を使って文字のリストに変換する。
2. 一つ目の文字のリストを `X`, 二つ目の文字のリストを `Y` とすると `X` の一つ目の文字が `Y` の中にでてくるまで調べる。
 - (a) 一つ目と一致する文字が `Y` の中に現れたら `X` の 2 文字目について繰り返す
 - (b) もし `Y` の方が先に `nil` になったら `false`
 - (c) もし `X` の方が先に `nil` になったら `X` の文字がすべて `Y` に出てきたということだから `true`

解答例 (他にもいろいろ書き方がある。結果としてきちんと部分並びを判定できていれば OK)

```
fun sq1(nil, _) = true
|   sq1(_, nil) = false
|   sq1(x::xs, y::ys) = if x=y then sq1(xs, ys)
                        else sq1 (x::xs, ys);
fun subsequence x y = sq1(explode x, explode y);
```

次に文字列のリスト ["ear", "part", "trap", "seat"] がある文字列の部分並びであるかどうかを判定するための関数リストを作る。演習 5.5.2 で作成した関数 `makeFnList` を使う

```
fun makeFnList F nil = nil
|   makeFnList F (x::xs) = F(x)::(makeFnList F xs);

val g = makeFnList(subsequence);
val G = g ["ear", "part", "trap", "seat"];
```

(参考)

```
val [ear, part, trap, seat] = g ["ear", "part", "trap", "seat"];
```

ということもできる.

最後にここで作成した関数リストを”separate” に対して適用する. 演習問題 5.5.1 で作成した関数 applyList を使う.

```
fun applyList nil _ = nil
  | applyList (F::Fs) a = F(a)::(applyList Fs a);

applyList G "separate";
```

2 実行例

```
- use "ML11answer.ml";
[opening ML11answer.ml]
ML11answer.ml:4.29 Warning: calling polyEqual
val sq1 = fn : ''a list * ''a list -> bool
val subsequence = fn : string -> string -> bool
val makeFnList = fn : ('a -> 'b) -> 'a list -> 'b list
val g = fn : string list -> (string -> bool) list
val G = [fn,fn,fn,fn] : (string -> bool) list
val applyList = fn : ('a -> 'b) list -> 'a -> 'b list
val it = () : unit
- applyList G "separate";
val it = [true,true,false,true] : bool list
```