

非手続き型言語5回目課題 解答例

May 11, 2020

1 局所環境を使った関数 rev

```
fun rev(L) =  
  let  
    fun rev1(M, []) = M  
    | rev1(M, x::xs) = rev1(x::M, xs)  
  in  
    rev1([],L)  
  end;
```

2 演習問題 3.5.2

ここでは局所環境を使わない解答例を示すが、局所環境を用いた方がさらによい。

```
fun cat(nil,M) = M  
| cat(x::xs,M) = x::cat(xs,M);  
  
fun cycle1(L, M, 0) = cat(L, rev(M))  
| cycle1(x::xs, M, i) = cycle1(xs, x::M, i-1);  
  
fun cycle(L,i) = cycle1(L, nil, i);
```

関数 rev を使わない別解 (問題文のヒントはこちらのことを言っていると思われる).

```
fun split (nil, ys, i) = (nil, ys)  
| split (x::xs, ys, 0) = (x::xs, ys)  
| split (x::xs, ys, i) =  
  let  
    val (M, N) = split(xs, ys, i-1)  
  in  
    (M, x::N)  
  end;  
  
fun cat(nil,M) = M
```

```

|   cat(x::xs,M) = x::cat(xs,M);

fun cycle2(L, i) =
  let
    val (M, N) = split(L, nil, i)
  in
    cat(M, N)
  end;

```

3 1 から n までの総和を求める末尾再帰の関数

```

fun sum(n) =
  let
    fun f1(0, s) = s
      | f1(n, s) = f1(n-1,s+n)
    in
      f1(n,0)
    end;

```

4 実行結果

```

- use "ML5answer.ml";
[opening ML5answer.ml]
val rev = fn : 'a list -> 'a list
val cat = fn : 'a list * 'a list -> 'a list
ML5answer.ml:15.5-16.48 Warning: match nonexhaustive
      (L,M,0) => ...
      (x :: xs,M,i) => ...

val cycle1 = fn : 'a list * 'a list * int -> 'a list
val cycle = fn : 'a list * int -> 'a list
val split = fn : 'a list * 'a list * int -> 'a list * 'a list
val cat = fn : 'a list * 'a list -> 'a list
val cycle2 = fn : 'a list * int -> 'a list
val sum = fn : int -> int
val it = () : unit
- rev ([1,2,3,4,5]);
val it = [5,4,3,2,1] : int list
- cycle ([1,2,3,4,5], 4);
val it = [5,1,2,3,4] : int list
- cycle2 ([1,2,3,4,5], 4);
val it = [5,1,2,3,4] : int list
- sum (10);
val it = 55 : int

```