非手続き型言語　8回目課題

09430509 今田将也

関数trapを使ってxxxと$xxx*x$ の0から1までの積分値がそれぞれ0.25, 0.2 となる最小のnを求めよ.

```
fun trap (a,b,n,F)=
    if n <= 0 orelse b-a <= 0.0 then 0.0
    else
        let
            val delta = (b-a) / real(n)
        in
            delta * (F(a)+F(a + delta))/ 2.0
                + trap(a+delta, b, n-1, F)
        end
;

trap(0.0, 1.0, 545000, fn x =>x*x*x);
trap(0.0, 1.0, 550000, fn x =>x*x*x*x);
```

出力

```
- use "trap";
[opening trap]
val trap = fn : real * real * int * (real -> real) -> real
val it = 0.25 : real
val it = 0.2 : real
val it = () : unit
```

資料演習問題5.4.2

わかりませんでした.

```
fun simpson(a,b,n,F)=
if n <= 0 orelse b-a <= 0.0 then 0.0
else
    let
        val h = (b-a) / (2.0 * real(n))
        val keisu = h / 3.0
        fun simpsonsub (ai) = 4.0 * F(ai) + 2.0 * F(ai+h)
        fun simpsonx2 (_,0) = 0.0
        |   simpsonx2 (ai,m) = simpsonsub(ai) + simpsonx2(ai+h+h,m-1)
    in
        keisu * (F(a) + simpsonx2(a+h, n) - F(b))
```

```
        end
    ;

    simpson (0.0, 1.0, 1, fn x=>x*x*x*x);(* 0.2 のはず*)
```

出力

```
- use "kadai542";
[opening kadai542]
val simpson = fn : real * real * int * (real -> real) -> real
val it = 0.208333333333 : real
val it = () : unit
```

資料演習問題5.4.3 b)

わかりませんでした.

```
fun simpson(a,b,n,F)=
if n <= 0 orelse b-a <= 0.0 then 0.0
else
    let
        val h = (b-a) / (2.0 * real(n))
        val keisu = h / 3.0
        fun simpsonsub (x) = 4.0 * F(x) + 2.0 * F(x+h)
        fun simpsonx2 (_,0) = 0.0
        |   simpsonx2 (x,m) = simpsonsub(x) + simpsonx2(x+h+h,m-1)
    in
        keisu * (F(a) + simpsonx2(a+h, n) - F(b))
    end
;

simpson (0.0, 1.0, 1, fn x=>x*x*x);
```

出力

```
- use "kadai543";
[opening kadai543]
val simpson = fn : real * real * int * (real -> real) -> real
val it = 0.25 : real
val it = () : unit
```

資料演習問題5.4.5

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

(* a. 実数のリス中の負の数を0で置き換え、非負の数はそのままにする *)
simpleMap(fn x=> if x < 0.0 then 0.0 else x, [4.4, ~4.4, 9.9]);

(* b. 整数のリストの全ての要素に1を加える *)
simpleMap(fn x=> x + 1, [1, ~2, 3, 4]);

(* c. 小文字だけ大文字に変換する *)
simpleMap(fn c => if c >= #"a" andalso c <= #"z" then chr(ord(c)+ord(#"A")-
ord(#"a")) else c, [#"I",#"m",#"A",#"d",#"a"]);

(* d. 文字列のリストの各要素の5文字より長い部分を切り捨てる *)
simpleMap(fn s => if size(s)>5 then substring(s, 0, 5) else s, ["Imada", "MASAYA",
"SMLSASAKURA"]);
```

出力

```
- use "simplemap";
[opening simplemap]
val simpleMap = fn : ('a -> 'b) * 'a list -> 'b list
val it = [4.4,0.0,9.9] : real list
val it = [2,~1,4,5] : int list
val it = [#"I",#"M",#"A",#"D",#"A"] : char list
val it = ["Imada","MASAY","SMLSA"] : string list
val it = () : unit
```