

非手続き型言語8回目課題 解答例

May 22, 2020

1 問題 1

関数 trap を使って x^3 と x^4 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 を求める問題. (本当に「最小」のものを求めるのが大変なときは自分が計算したものの中で一番それに近い n とその時の積分値を答える.)

スライドにある以下の関数 trap を自分の処理系で定義する.

```
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;
```

そして例えば以下のようにして結果を見る. $f(x) = x^3$ の場合. 理論値は 0.25.

```
trap(0.0, 1.0, 1000, fn x => x * x * (x:real));
trap(0.0, 1.0, 10000, fn x => x * x * (x:real));
trap(0.0, 1.0, 100000, fn x => x * x * (x:real));
trap(0.0, 1.0, 1000000, fn x => x * x * (x:real));
```

$f(x) = x^4$ の場合. 理論値は 0.2.

```
trap(0.0, 1.0, 1000, fn x => x * x * x * (x:real));
trap(0.0, 1.0, 10000, fn x => x * x * x * (x:real));
trap(0.0, 1.0, 100000, fn x => x * x * x * (x:real));
trap(0.0, 1.0, 1000000, fn x => x * x * x * (x:real));
```

(結果はマシンによって違うのでここでは省略. 自分のマシンでやってみてください.)

2 資料演習問題 5.4.2

いろいろな解答例がありうるが、例えばこれ.

```
fun simpson(a,b,n,F) =  
  let  
    val delta = (b-a)/(2.0*real(n));  
    val keisu = delta/3.0;  
    fun simpsonsub(ai) = 4.0*F(ai) + 2.0*F(ai+delta);  
    fun simpsonsub2(_,0) = 0.0  
      | simpsonsub2(ai,m) =  
          simpsonsub(ai) + simpsonsub2(ai+delta+delta,m-1);  
  in  
    if n <= 0 orelse b-a <= 0.0 then 0.0  
    else  
      keisu*(F(a)+simpsonsub2(a+delta,n)-F(b))  
  end;
```

これを使って

```
simpson(0.0,1.0,10,fn (x:real)=>x*x);  
simpson(0.0,1.0,10,fn (x:real)=>x*x*x);
```

のようにして積分値が正しくなくなる最小の n を探す.

3 資料演習問題 5.4.3 b)

すでに演習問題 5.4.2 の解がこの条件を満たしていました. 代わりに a) の方の解答を示しますので、問題の意図を汲み取ってください.

4 資料演習問題 5.4.3 a)

以下の関数 trap を

```
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;
```

delta を再帰呼び出しのたびに計算しないようにしたもの.

```

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);
      fun trap1(_,0) = 0.0
        | trap1(x,i) = delta*(F(x)+F(x+delta))/2.0
          + trap1(x+delta,i-1)
    in
      trap1(a,n)
    end;

```

5 資料演習問題 5.4.5

```

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

```

```

(* a *)
val L=[1.0, 6.5, ~3.3, 1.2];
map(fn x=> if x > 0.0 then x else 0.0, L);

```

```

(* b *)
val L2=[1, 6, ~3];
map(fn x=> x+1, L2);

```

```

(* c *)
val S=["a", "#N", "#f", "#r"];
map(fn c=> if c>= #"a" andalso c<= #"z" then chr(ord(c)-32) else c, S);
(* 32 がわからなかったら*)
map(fn c=> if c>= #"a" andalso c<= #"z" then chr(ord(c)-ord(#"a")+ord(#"A")) else c, S);

```

```

(* d *)
val S2=["abcde", "Nonmonoton", "fa", "rest"];
map(fn s=> if size(s) > 5 then substring(s,0,5) else s, S2);

```