

宣言型プログラム論

200911434 青木大祐

平成 24 年 9 月 18 日

問題3.1

```
1 (* 図形のデータ型 *)
2 type figure =
3   | Circle of float
4   | Square of float
5   | Rectangle of float * float;;
6
7 (* 面積を求める関数 *)
8 let area (x:figure) =
9   match x with
10    | Circle x -> x *. x *. 3.14
11    | Square x -> x *. x
12    | Rectangle ( x, y ) -> x *. y;;
13
14 print_float (area (Circle 2.));;
15 print_float (area (Square 2.));;
16 print_float (area (Rectangle(2.0, 3.0)));;
```

```
1 type figure = Circle of float | Square of float | Rectangle of float * float
2 # val area : figure -> float = <fun>
3 # 12.56- : unit = ()
4 # 4.- : unit = ()
5 # 6.- : unit = ()
6 #
```

正しく計算できている。

問題3.2

```
1 (* 木の定義 *)
2 type 'a tree =
3   | Lf
4   | Br of 'a * 'a tree * 'a tree;;
5
6 (* 木の深さを求める関数 *)
7 let rec depth (t:'a tree) =
8   match t with
9   | Lf -> 0
10  | Br (v, t1, t2) -> 1 + max (depth t1) (depth t2);;ラベル
11
12 (*の完全二分木を返す関数X *)
13 let rec comptree (n:int) x =
14   if n = 0 then Lf else Br (x, (comptree (n - 1) x), (comptree (n - 1) x));;
15
16 let sample = Br(2, Br(4, Br(5, Lf, Lf), Lf), Br(1, Lf, Lf));;
17 depth sample;;
18 comptree 5 "a";;
```

```
1 type 'a tree = Lf | Br of 'a * 'a tree * 'a tree
2 # val depth : 'a tree -> int = <fun>
3 # val comptree : int -> 'a -> 'a tree = <fun>
4 # val sample : int tree = Br (2, Br (4, Br (5, Lf, Lf), Lf), Br (1, Lf, Lf))
5 # - : int = 3
6 # - : string tree =
7 Br ("a",
8   Br ("a",
9     Br ("a", Br ("a", Lf, Lf), Br ("a", Lf, Lf)),
10    Br ("a", Br ("a", Lf, Lf), Br ("a", Lf, Lf))),
11   Br ("a", Br ("a", Br ("a", Lf, Lf), Br ("a", Lf, Lf)),
12    Br ("a", Br ("a", Lf, Lf), Br ("a", Lf, Lf)))),
13  Br ("a",
14    Br ("a", Br ("a", Br ("a", Lf, Lf), Br ("a", Lf, Lf)),
15     Br ("a", Br ("a", Lf, Lf), Br ("a", Lf, Lf))),
16    Br ("a", Br ("a", Br ("a", Lf, Lf), Br ("a", Lf, Lf)),
17     Br ("a", Br ("a", Lf, Lf), Br ("a", Lf, Lf))))
```