

ソフトウェアサイエンス実験 S8 課題 2-4

200911434 青木大祐

平成 24 年 10 月 18 日

以下に課題 2-4 のソースコードを示す。

```
1 (* 式の型 *)
2 type exp =
3   | IntLit of int
4   | Plus of exp * exp
5   | Times of exp * exp
6   | BoolLit of bool      (* 追加分; 真理値リテラル, つまり trueや false *)
7   | If of exp * exp * exp (* 追加分; if-then-else式 *)
8   | Eq of exp * exp      (* 追加分; e1 = e2 *)
9   | Greater of exp * exp ;;
10
11 (* 値の型 *)
12 type value =
13   | IntVal of int      (* 整数の値 *)
14   | BoolVal of bool    (* 真理値の値 *);;
15
16 (* eval2 : exp -> value *)
17 let rec eval2 e =
18   match e with
19   | IntLit(n) -> IntVal(n)
20   | Plus(e1,e2) ->
21     begin
22       match (eval2 e1, eval2 e2) with
23       | (IntVal(n1),IntVal(n2)) -> IntVal(n1+n2)
24       | _ -> failwith "integer values expected"
25     end
26   | Times(e1,e2) ->
27     begin
28       match (eval2 e1, eval2 e2) with
29       | (IntVal(n1),IntVal(n2)) -> IntVal(n1*n2)
30       | _ -> failwith "integer values expected"
31     end
32   | Eq(e1,e2) ->
33     begin
34       match (eval2 e1, eval2 e2) with
35       | (IntVal(n1),IntVal(n2)) -> BoolVal(n1=n2)
36       | (BoolVal(b1),BoolVal(b2)) -> BoolVal(b1=b2)
37       | _ -> failwith "wrong value"
38     end
39   | BoolLit(b) -> BoolVal(b)
40   | If(e1,e2,e3) ->
41     begin
42       match (eval2 e1) with
43       | BoolVal(true) -> eval2 e2
44       | BoolVal(false) -> eval2 e3
45       | _ -> failwith "wrong value"
46     end
47   | Greater(e1, e2) ->
48     begin
49       match (eval2 e1, eval2 e2) with
50       | (IntVal(n1), IntVal(n2)) -> BoolVal(n1 > n2)
51       | _ -> failwith "wrong value"
52     end
53   | _ -> failwith "unknown expression e";;
54
55 print_string "2.4.1 テスト";;
56 let _ = eval2 (IntLit 1);;
57 let _ = eval2 (IntLit 11);;
58 let _ = eval2 (Plus (IntLit 1, Plus (IntLit 2, IntLit 11)));;
59 let _ = eval2 (Times (IntLit 1, Plus (IntLit 2, IntLit 11)));;
60 let _ = eval2 (If (Eq(IntLit 2, IntLit 11),
61   Times(IntLit 1, IntLit 2),
62   Times(IntLit 1, Plus(IntLit 2,IntLit 3))));;
63 let _ = eval2 (Eq (IntLit 1, IntLit 1));;
64 let _ = eval2 (Eq (IntLit 1, IntLit 2));;
65 let _ = eval2 (Eq (BoolLit true, BoolLit true));;
66 let _ = eval2 (Eq (BoolLit true, BoolLit false));;
67
68 print_string "2.4.2 適当な例でエラーを起こす";;
69 eval2 (Plus (IntLit 10, BoolLit(true)));;
70 eval2 (If ((IntLit 1), (IntLit 2), (IntLit 3)));;
71
72 print_string "2.4.3 整数と真理値をeqで比較する";;
73 eval2( Eq(IntLit 1, BoolLit true));;
74
75 print_string "2.4.4 Greaterを実装する";;
76 eval2 (Greater (IntLit 10, IntLit 50));;
77 eval2 (Greater (IntLit 10, IntLit 5));;
```

2.4.1 テスト

以下にテストコードと、その実行結果を示す。

```
1 print_string "2.4.1 テスト";;
2 let _ = eval2 (IntLit 1);;
3 let _ = eval2 (IntLit 11);;
4 let _ = eval2 (Plus (IntLit 1, Plus (IntLit 2, IntLit 11)));;
5 let _ = eval2 (Times (IntLit 1, Plus (IntLit 2, IntLit 11)));;
6 let _ = eval2 (If (Eq(IntLit 2, IntLit 11),
7                 Times(IntLit 1, IntLit 2),
8                 Times(IntLit 1, Plus(IntLit 2, IntLit 3))));;
9 let _ = eval2 (Eq (IntLit 1, IntLit 1));;
10 let _ = eval2 (Eq (IntLit 1, IntLit 2));;
11 let _ = eval2 (Eq (BoolLit true, BoolLit true));;
12 let _ = eval2 (Eq (BoolLit true, BoolLit false));;
```

```
1 # 2.4.1 テスト- : unit = ()
2 # - : value = IntVal 1
3 # - : value = IntVal 11
4 # - : value = IntVal 14
5 # - : value = IntVal 13
6 # - : value = IntVal 5
7 # - : value = BoolVal true
8 # - : value = BoolVal false
9 # - : value = BoolVal true
10 # - : value = BoolVal false
```

正しく計算できていることがわかる。

2.4.2 エラーを起こす適当な例

```
1 print_string "2.4.2 適当な例でエラーを起こす";;
2 eval2 (Plus (IntLit 10, BoolLit(true)));;
3 eval2 (If ((IntLit 1), (IntLit 2), (IntLit 3)));;
```

このコードを実行すると、以下のように例外が発生する。

```
1 # 2.4.2 適当な例でエラーを起こす- : unit = ()
2 # Exception: Failure "integer values expected".
3 # Exception: Failure "wrong value".
```

どちらも引数として適合しない型の値を渡しているため、パターンマッチに用意されている例外が発生する。

2.4.3 整数と真理値の比較

```
1 print_string "2.4.3 整数と真理値をeqで比較する";;
2 eval2( Eq(IntLit 1, BoolLit true));;
```

実行結果は以下のとおり。

```
1 # 2.4.3 整数と真理値をeqで比較する - : unit = ()
2 # Exception: Failure "wrong value".
```

35 行目、36 行目のどちらにもマッチしないので、45 行目の例外が発生する。

2.4.4 Greater の実装

```
1 print_string "2.4.4 Greaterを実装する";;  
2 eval2 (Greater (IntLit 10, IntLit 50));;  
3 eval2 (Greater (IntLit 10, IntLit 5));;
```

実行結果は以下のとおり。

```
1 # 2.4.4 Greaterを実装する - : unit = ()  
2 # - : value = BoolVal false  
3 # - : value = BoolVal true
```

正しく計算できている。