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 3 日

以下に課題 2-4 のソースコードを示す。

```
(* 式の型 *)
type exp =
  | IntLit of int
  | Plus of exp * exp
  | Times of exp * exp
  | BoolLit of bool        (* 追加分; 真理値リテラル，つまり trueや false  *)
  | If of exp * exp * exp  (* 追加分; if-then-else式 *)
  | Eq of exp * exp        (* 追加分; e1 = e2 *)
  | Greater of exp * exp ;;

(* 値の型 *)
type value =
  | IntVal  of int          (* 整数の値 *)
  | BoolVal of bool         (* 真理値の値 *);;

(* eval2 : exp -> value *)
let rec eval2 e =
  match e with
  | IntLit(n)  -> IntVal(n)
  | Plus(e1,e2) ->
      begin
        match (eval2 e1, eval2 e2) with
    | (IntVal(n1),IntVal(n2)) -> IntVal(n1+n2)
          | _ -> failwith "integer values expected"
      end
  | Times(e1,e2) ->
      begin
        match (eval2 e1, eval2 e2) with
    | (IntVal(n1),IntVal(n2)) -> IntVal(n1*n2)
          | _ -> failwith "integer values expected"
      end
  | Eq(e1,e2) ->
      begin
  match (eval2 e1, eval2 e2) with
    | (IntVal(n1),IntVal(n2)) -> BoolVal(n1=n2)
    | (BoolVal(b1),BoolVal(b2)) -> BoolVal(b1=b2)
    | _ -> failwith "wrong value"
      end
  | BoolLit(b) -> BoolVal(b)
  | If(e1,e2,e3) ->
      begin
  match (eval2 e1) with
    | BoolVal(true) -> eval2 e2
    | BoolVal(false) -> eval2 e3
    | _ -> failwith "wrong value"
      end
  | Greater (e1, e2) ->
      begin
        match (eval2 e1, eval2 e2) with
          | (IntVal(n1), IntVal(n2)) -> BoolVal(n1 > n2)
          | _ -> failwith "wrong value"
      end
  | _ -> failwith "unknown expression e";;

print_string "2.4.1 テスト";;
let _ = eval2 (IntLit 1);;
let _ = eval2 (IntLit 11);;
let _ = eval2 (Plus (IntLit 1, Plus (IntLit 2, IntLit 11)));;
let _ = eval2 (Times (IntLit 1, Plus (IntLit 2, IntLit 11)));;
let _ = eval2 (If (Eq(IntLit 2, IntLit 11),
                   Times(IntLit 1, IntLit 2),
                   Times(IntLit 1, Plus(IntLit 2,IntLit 3))));;
let _ = eval2 (Eq (IntLit 1, IntLit 1));;
let _ = eval2 (Eq (IntLit 1, IntLit 2));;
let _ = eval2 (Eq (BoolLit true, BoolLit true));;
let _ = eval2 (Eq (BoolLit true, BoolLit false));;

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

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

print_string "2.4.5 Greaterを実装する";;
eval2 (Greater (IntLit 10, IntLit 50));;
eval2 (Greater (IntLit 10, IntLit 5));;
```

2.4.1　テスト

```
式の型追加分真理値リテラルつまりや追加分式追加分値の型整数の値真理値の値テスト
let _ = eval2 (IntLit 1);;
let _ = eval2 (IntLit 11);;
let _ = eval2 (Plus (IntLit 1, Plus (IntLit 2, IntLit 11)));;
let _ = eval2 (Times (IntLit 1, Plus (IntLit 2, IntLit 11)));;
let _ = eval2 (If (Eq(IntLit 2, IntLit 11),
                   Times(IntLit 1, IntLit 2),
                   Times(IntLit 1, Plus(IntLit 2,IntLit 3))));;
let _ = eval2 (Eq (IntLit 1, IntLit 1));;
let _ = eval2 (Eq (IntLit 1, IntLit 2));;
let _ = eval2 (Eq (BoolLit true, BoolLit true));;
let _ = eval2 (Eq (BoolLit true, BoolLit false));;

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

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

print_string "2.4.5 Greaterを実装する";;
eval2 (Greater (IntLit 10, IntLit 50));;
eval2 (Greater (IntLit 10, IntLit 5));;
```

以下にテストコードと、その実行結果を示す。

```
#     2.4.1 テスト- : unit = ()
# - : value = IntVal 1
# - : value = IntVal 11
# - : value = IntVal 14
# - : value = IntVal 13
#     - : value = IntVal 5
# - : value = BoolVal true
# - : value = BoolVal false
# - : value = BoolVal true
# - : value = BoolVal false
```

正しく計算できていることがわかる。