主専攻実験[S-8] 関数プログラミング

課題7-2

情報科学類 202113564 三村潤之介

# 課題7-2

・関数tcheck1 に，いくつかの式を与え，正しく型検査できていることを 確かめよ．(たとえば，If(BoolLit(true), IntLit(1), IntLit(100)) を与え たり，If(BoolLit(true), Plus(IntLit(1), BoolLit(false)), IntLit(100)) を与えたりして，どういう結果になるか確認せよ．)

　例として挙げられている２つを検査することを考える。Ifについて、第一引数がTBoolでかつ、第二引数、第三引数が同じ型であれば、その型を全体の型として返す。違う型であれば、型エラーとする。

実際に実行した様子を下に示す。

# tcheck1(If(BoolLit(true), IntLit(1), IntLit(100)));;

- : ty = Tint

#　tcheck1(If(BoolLit(true),Plus(IntLit(1),BoolLit(false)), 　IntLit(100)) );;

Exception: Failure "type error in Plus".

　挙げられた例について、前者は、第一引数はTBool、第二引数と第三引数はともにTIntであるのでTintが返され、後者は、第二引数について、Plusの第二引数がTBoolであるので、Plusにて型エラーになっている。

・上記の定義で扱っている式の範囲を拡張し，(e1 = e2) (等しさの比較) に対する型も推論できるようにせよ． ただし、(e1 = e2) が整合的な型を持つのは、 e1 と e2 が同じ型(両方とも整数型か、両方とも真理値型)のときであり、 どちらのケースでも (e1=e2)全体は真理値型である。

“Eq”として比較を追加する。まず、exp式にEqの定義を追加する。

type exp =

…

| Eq of exp \* exp

Eqは２つのexp式を引数とするため上記のような記述となる。

次に、型検査器にEqの定義を追加する。

let rec tcheck1 e =

match e with

　 …

| Eq(e1,e2) ->

begin

match (tcheck1 e1, tcheck1 e2) with

| (TBool,TBool) -> TBool

| (TInt,TInt) -> TBool

| \_ -> failwith "type error in EQ"

end

与えられる２つの引数について、型検査した結果がどちらも同じ型であればTBoolとし、そうでなければ、型エラーとする。これを、match文により、上記のように実装する。

以下にOCamlで例を与えたときの実行結果を示す。

# tcheck1( Eq(BoolLit(false), BoolLit(true)) );;

- : ty = TBool

# tcheck1( Eq(BoolLit(false), IntLit(20)) );;

Exception: Failure "type error in EQ".

上の例では、どちらもTBoolで同じ型であるので、TBoolが返されている。下の例では、第二引数がTintであるので、エラーが返されている。