

主専攻実験[S-8] 関数プログラミング

発展課題 7-6,7-7

情報科学類 202113564 三村潤之介

関数 unify に、いくつかの型に関する式を与え、正しく単一化できていることを確かめよ。

資料の例を実行した結果を以下に示す。

```
# unify tyenv1;; (* [(TVar("'a"),TBool)] *)
- : (tyvar * ty) list = [("'a", TBool)]
# unify tyenv2;; (* [(TInt,TBool)] *)
Exception: Failure "unification failed".
# unify tyenv3;; (* [(TVar("'a"),TVar("'b"))]* )
- : (tyvar * ty) list = [("'a", TVar "'b")]
# unify tyenv4;; (* [(TArrow(TVar("'a"),TVar("'b")),
                        TArrow(TVar("'b"), TVar("'c")))]*)
- : (tyvar * ty) list = [("'b", TVar "'c"); ("a", TVar "'c")]
# unify tyenv5;; (* [(TVar("'a"),TArrow(TVar("'b"),TVar("'a")))] *)
Exception: Failure "unification failed".
```

型推論器に、いくつかの式を与え、正しく型推論できていることを確かめよ。

実行した結果を以下に示す。

```
#tinf2top
(Fun("f",Fun("x",App(Var("f"),App(Var("f"),Var("x"))))));;
- : (string * ty) list * ty * tysubst * int = ([],TArrow (TArrow
(TVar "'a3", TVar "'a3"), TArrow (TVar "'a3", TVar "'a3")), [(("'a1",
TVar "'a3"); ("a2", TVar "'a3"); ("a0", TArrow (TVar "'a3", TVar
"'a3"))], 4)

# tinf2top (Fun("x",Fun("y",Fun("z", App(App(Var("x"),Var("z")),
App(Var("y"),Var("z"))))));;
- : (string * ty) list * ty * tysubst * int =
([],
TArrow (TArrow (TVar "'a2", TArrow (TVar "'a4", TVar "'a5")),
TArrow (TArrow (TVar "'a2", TVar "'a4"), TArrow (TVar "'a2", TVar
"'a5"))),
[(("'a3", TArrow (TVar "'a4", TVar "'a5"));
("'a1", TArrow (TVar "'a2", TVar "'a4"));
("'a0", TArrow (TVar "'a2", TArrow (TVar "'a4", TVar "'a5")))],
```

上記の実装では BoolLit と If のケースが落ちている。これらを実装して型推論器 tinf2 を完成させよ。

BoolLit と If を以下のように実装した。

```
...
| BoolLit(_) -> (te, TBool, theta0, n)
| If(e1,e2,e3) ->
  let (te1, t1, theta1, n1) = tinf2 te e1 n in
  let (te2, t2, theta2, n2) = tinf2 te1 e2 n1 in
  let (te3, t3, theta3, n3) = tinf2 te2 e3 n2 in
  let t11 = subst_ty theta2 t1 in
  let t12 = subst_ty theta3 t11 in
  let theta4 = unify [(t12,TBool)] in
  let t21 = subst_ty theta4 t2 in
  let t22 = subst_ty theta3 t21 in
  let t31 = subst_ty theta4 t3 in
  let theta5 = unify [(t22,t31)] in
  let theta6 = compose_subst theta5 (compose_subst theta4 (compose_subst
theta3 (compose_subst theta2 theta1))) in
  let te4 = subst_tyenv theta6 te3 in
  (te4, t22, theta6, n3)
...
```

実行結果は以下ようになった。資料に示されていた If 文を対象とした。

```
# tinf2top (If(BoolLit(true), IntLit(1), IntLit(100))));;
- : (string * ty) list * ty * tysubst * int = ([], TInt, [], 0)

# tinf2top (If(Var("x"),Plus(Var("y"), Var("z")), Var("z")));;
- : (string * ty) list * ty * tysubst * int =
[("z", TInt); ("y", TInt); ("x", TBool)], TInt,
[("a0", TBool); ("a2", TInt); ("a1", TInt)], 3)
```

この型推論器を拡張して、ミニ OCaml 言語のすべての式に対応するようにせよ。(let などにも対応させよ。)

Let 文を以下のように実装した。

```
| Let(x,e1,e2) ->
  let (te1, t1, theta1, n1) = tinf2 te e1 n in
  let te2 = (ext (te1) (x) t1) in
  tinf2 te2 e2 n1
```

実行結果は以下ようになった。適当な Let 文を用意し、実行した。

```
# tinf2top Let ("x", IntLit 2, If (BoolLit true, Var "y", Var
"x"));;
- : (string * ty) list * ty * tysubst * int =
([("y", TInt); ("x", TInt)], TVar "'a0", [("a0", TInt)], 1)
```