計算機科学実験3HSW レポート3

橘大佑
1029-31-6811
2019 年度入学

2021/07/10

工学部情報学科計算機科学コース
1029-31-6811
橘 大佑

SW レポート3

計算機科学実験及び演習 3
2021 年 7 月 10 日

Exercise 4.2.1

- プログラムの設計方針
  すべての場合について型推論アルゴリズムを完成させた。また、インタプリタに変更を加え、型推論ができるようにした。

- 実装の詳細な説明
  syntax.ml に miniML の型を表す OCaml の値の型 ty と ty 型の値のための pretty printer となる pp_ty を記述した。そして、cui.ml に型環境を REPL で保持する read_eval_print と initial_env のための型環境を作った。また、main.ml に initial_tyenv を REPL の最初の呼び出しで渡した。加えて、typing.ml を変更して T-Int と T-Plus のケースにならってすべての場合について型推論アルゴリズムを完成させた。ty_prim には、 積については演算対象がどちらも整数のときに整数、大小比較については演算対象がどちらも整数のときに真偽値、論理積については演算対象がどちらも真偽値のときに真偽値、論理和については演算対象がどちらも真偽値のときに真偽値、そうでない場合はそれぞれエラーを返すように 4 つの二項演算子について記述した。また、ty_exp には IfExp に関する記述を追加した。条件式が真偽値で、true と false の場合の型が等しい場合はその型を返し、両者の型が異なる場合や、そもそも条件式が真偽値でない場合にはエラーを返すようにした。そして、LetExp には let x = e1 in e2 という式において式 e1 が型 $\tau 1$ を持ち，式 e2 が Γ を x:$\tau 1$ というエントリで拡張して得られる型環境 Γ,x:$\tau 1$ の下で型 $\tau 2$ を持つならば，式 let x=e1 in e2 は全体として $\tau 2$ を持つという判断を導いた。最後に、ty_decl に Decl に関する記述を追加した。

Exercise 4.3.1

- プログラムの設計方針
  pp_ty と freevar_ty を完成させた。

- 実装の詳細な説明
  型のための pretty printer である pp_ty と ty に現れる自由な型変数の識別子（つまり，tyvar 型の集合）を返す関数である freevar_ty と MiniML 上の型を受け取ってその文字列表現を返す関数である string_ty の 3 つを実装した。pp_ty では整数値、真偽値であるときはそれぞれ"int","string" を表示し、TyVar 型であるときは char_of_int 関数を用いて 0 のときは" 'a "、1 のときは" 'b "$\cdots$ と表示されるようにした。TyFun (t1, t2) の場合は pp_ty を再帰的に適用して「t1 を表現する文字列」->「t2 を表現する文字列」と出力するように記述した。string_of_ty では整数値、真偽値であるときはそれぞれ"int","string"を表示し、TyVar 型であるときは string_of_int 関数を用いて 0 のときは" 'a0 "、1 のときは" 'a1 "$\cdots$ と表示されるようにした。これによって型として valid な記号を使った型を付けることができた。TyFun (t1, t2) の場合は -> が右結合であることに注意して、ty1 が TyFun 型であるときのみ (->) と括弧をつけるようにした。

Exercise 4.3.2

- プログラムの設計方針
  型変数と型のペアのリストで表現される、型代入に関する型、関数 subst_typ を typing.ml 中に実装した。

- 実装の詳細な説明
  subst_type では末尾の型代入から、自身より前の型代入に適用していき、それにより新しく出来た型代入を型変数に適用した。

Exercise 4.3.3

- プログラムの設計方針
  単一化アルゴリズムを (ty * ty) list -> subst 型の関数 unify として実装した。

- 実装の詳細な説明
  型代入を等式制約に適用する関数 subst_eqs と subst_eqs を用いて単一化を行うプログラム unify を typing.ml に記述した。

$$
\begin{aligned}
Unify(\emptyset) &= \emptyset \\
Unify(X' \uplus \{\tau = \tau\}) &= Unify(X') \\
Unify(X' \uplus \{\tau_{11} \to \tau_{12} = \tau_{21} \to \tau_{22}\}) &= Unify(\{\tau_{11} = \tau_{21}, \tau_{12} = \tau_{21}\} \uplus X') \\
Unify(X' \uplus \{\alpha = \tau\}) \quad (if \tau \neq \alpha) &= \begin{cases} Unify([\alpha \mapsto \tau]X') \circ [\alpha \mapsto \tau] & (\alpha \notin FTV(\tau)) \\ Error & (Otherwise) \end{cases} \\
Unify(X' \uplus \{\tau = \alpha\}) \quad (if \tau \neq \alpha) &= \begin{cases} Unify([\alpha \mapsto \tau]X') \circ [\alpha \mapsto \tau] & (\alpha \notin FTV(\tau)) \\ Error & (Otherwise) \end{cases} \\
Unify(X \uplus \tau_1 = \tau_2) &= Error \quad (Otherwise)
\end{aligned}
$$

上が単一化アルゴリズムの定義である。1 式では制約集合が空なら空の代入が返される。2 式では $\tau = \tau$ という制約は単に削除される。3 式では関数型同士の間の等式制約は分解される。4,5 式では $\alpha = \tau$ という制約の単一化を行っている。6 式ではどれにも当てはまらない場合はエラーとなる。これを (ty * ty) の形式に当てはめていく。2 式 は TyInt, TyInt や TyBool, TyBool のパターン、また同一変数のペアが当てはまる。3 式 は TyFun (ty11, ty12), TyFun (ty21, ty22) が当てはまる。4,5 式は TyVar var1, TyVar var2 で、var1 と var2 が異なる場合と、TyVar とそれ以外のペアが当てはまる。

Exercise 4.3.4

- プログラムの設計方針
  単一化アルゴリズムにおいて，オカーチェックの条件 $\alpha \notin FTV(T)$ を説明した。

- 実装の詳細な説明
  a と τ が異なり、FTV(τ) に α が含まれている場合としては以下のような場合が考える。

  (TyVar a, TyFun (TyVar a, TyInt))

  もし単一化アルゴリズムにおいて上記の条件がない場合、a の型が TyFun (TyVar a, TyInt) となるが、 a が自身の型を内包するため型が無限に展開されるため、$\alpha \notin FTV(T)$ という条件によって、自身の型を含み無限展開される再帰的な型を生成しないようにしている。

Exercise 4.3.5

工学部情報学科計算機科学コース
1029-31-6811
橘 大佑

計算機科学実験及び演習 3

SW レポート 3

2021 年 7 月 10 日

- プログラムの設計方針
  他の型付け規則に関しても同様に型推論の手続きを与えた。そして、以下の typing.ml に加えるべき変更の解説を参考にして，型推論アルゴリズムの実装を完成させた。

- 実装の詳細な説明
  - id
    - Γ から id の型変数を探して返す
  - 整数値
    - 空の型代入 と int を出力として返す
  - 真偽値
    - 空の型代入 と bool を出力として返す.
  - e1 * e2
    - Γ,e1 を入力として型推論を行い，$\theta 1$，$\tau 1$ を得る
    - Γ,e2 を入力として型推論を行い，$\theta 2$，$\tau 2$ を得る
    - 型代入 $\theta 1$,$\theta 2$ を $\alpha = \tau$ という形の方程式の集まりとみなして，$\theta 1 \cup \theta 2 \cup \tau 1 = \text{int},(\tau 2,\text{int})$ を単一化し，型代入 $\theta 3$ を得る
    - $\theta 3$ と int を出力として返す
  - e1 < e2
    - Γ,e1 を入力として型推論を行い，$\theta 1$，$\tau 1$ を得る
    - Γ,e2 を入力として型推論を行い，$\theta 2$，$\tau 2$ を得る
    - 型代入 $\theta 1$,$\theta 2$ を $\alpha = \tau$ という形の方程式の集まりとみなして，$\theta 1 \cup \theta 2 \cup \tau 1 = \text{int},(\tau 2,\text{int})$ を単一化し，型代入 $\theta 3$ を得る
    - $\theta 3$ と bool を出力として返す
  - e1 && e2
    - Γ,e1 を入力として型推論を行い，$\theta 1$，$\tau 1$ を得る
    - Γ,e2 を入力として型推論を行い，$\theta 2$，$\tau 2$ を得る
    - 型代入 $\theta 1$,$\theta 2$ を $\alpha = \tau$ という形の方程式の集まりとみなして，$\theta 1 \cup \theta 2 \cup \tau 1 = \text{bool},(\tau 2,\text{bool})$ を単一化し，型代入 $\theta 3$ を得る
    - $\theta 3$ と bool を出力として返す
  - e1 || e2
    - Γ,e1 を入力として型推論を行い，$\theta 1$，$\tau 1$ を得る
    - Γ,e2 を入力として型推論を行い，$\theta 2$，$\tau 2$ を得る
    - 型代入 $\theta 1$,$\theta 2$ を $\alpha = \tau$ という形の方程式の集まりとみなして，$\theta 1 \cup \theta 2 \cup \tau 1 = \text{bool},(\tau 2,\text{bool})$ を単一化し，型代入 $\theta 3$ を得る
    - $\theta 3$ と bool を出力として返す

- if e1 then e2 else e3
  - Γ,e1 を入力として型推論を行い，$\theta 1$，$\tau 1$ を得る
  - Γ,e2 を入力として型推論を行い，$\theta 2$，$\tau 2$ を得る
  - Γ,e3 を入力として型推論を行い，$\theta 3$，$\tau 3$ を得る
  - 型代入 $\theta 1$,$\theta 2$,$\theta 3$ を $\alpha = \tau$ という形の方程式の集まりとみなして，
    $\theta 1 \cup \theta 2 \cup \tau 1 = \{(\tau 1, bool)\} \cup \theta 1 \cup \theta 2 \cup \theta 3 \cup \{(\tau 2, \tau 3)\}$ を単一化し，型代入 $\theta 4$ を得る
  - $\theta 4$ と $\tau 2$ を出力として返す
- let id = e1 in e2
  - Γ,e1 を入力として型推論を行い，$\theta 1$，$\tau 1$ を得る
  - Γ を id, $\tau 1$ で拡張し Γ2 を得る
  - Γ2,e2 を入力として型推論を行い，$\theta 2$，$\tau 2$ を得る
  - 新しく型変数 a1 を確保する
  - 型代入 $\theta 1$,$\theta 2$ を $\alpha = \tau$ という形の方程式の集まりとみなして，
    $\theta 1 \cup \{(a1, \tau 1)\} \cup \theta 2$ を単一化し，型代入 $\theta 3$ を得る
  - $\theta 3$ と $\tau 2$ を出力として返す
- fun id → e1
  - 新しく型変数 domty を確保する
  - Γ を id, domty で拡張し Γ2 を得る
  - Γ2, e1 を入力として型推論を行い，$\theta 1$, ranty を得る
  - $\theta 1$ と domty から 型 a2 を得る
  - $\theta 1$ と TyFun(a2, ranty) を出力して返す
- e1 e2
  - Γ,e1 を入力として型推論を行い，$\theta 1$，$\tau 1$ を得る
  - Γ,e2 を入力として型推論を行い，$\theta 2$，$\tau 2$ を得る
  - 新しく型変数 domty を確保する
  - 型代入 $\theta 1$,$\theta 2$ を $\alpha = \tau$ という形の方程式の集まりとみなして，
    $\{(\tau 1, TyFun(\tau 2, domty))\} \cup \theta 1 \cup \theta 2$ を単一化し，型代入 $\theta 3$ を得る．
  - $\theta 3$ と domty を出力として返す

- ty_Decl の実装
  (型代入, 型) のペアを、期待されている (型環境、型) のペアに書き直した。(型代入, 型) のペアから型のみを取り出すのには、let 式を用いた。

実装において工夫した点

型推論はインタープリタ実験を参考にしながら実装した。

感想

難しかった。