プログラム言語論

亀山幸義

筑波大学 情報科学類

No. 6: 型システムその 2 (型推論の基本)

いろいろな型システム

プログラム言語ごとに、「型」がどう扱われるかは異なる。

- 型の概念がない言語: 機械語
- 型の概念はあるが型の整合性の検査は実行時に行う言語 (動的型付け言語): Ruby,Python,Perl,JavaScript,Lisp,...
- 型の概念はあり、型の整合性の検査を実行前に行う言語 (静的型付け言語; C,Java,OCaml,Haskell,...

静的型付け言語がさらに 2 つに分かれる。

- 変数の型を明示的に記述する必要があり、型の整合性の検査のみが必要な言語 (型検査)
- 変数の型を明示的に記述する必要がなく (記述してもよい)、型を推論しつつ整合性を検査する必要がある言語 (型推論)

型推論の例

(授業で配付した資料は、関数 goo の定義が間違っていたので修正しました。)

```
let rec foo x =
   if x = 0 then 1
   else x * (foo (x - 1))
==>
 foo : int → int

let rec goo x =
   if x = 0 then 1
   else x +. (goo (x - 1))
==>
type error
(* The operator +. is for floating-point numbers.*)
```

型推論アルゴリズム

- 入力: プログラム
- 出力: 「型がつくかどうか」と、つく場合はその型

型推論; 先週の例

```
let rec tarai x y z =
  if x <= y then y
  else tarai (tarai (x-1) y z)
             (tarai (y-1) z x)
             (tarai (z-1) x y)
```

- `x-1` 等があるので x,y,z は int 型
- 関数本体が `if..then y else...` なので、関数の返り値の型は y と同じ型、よって int 型
- `tarai: int -> int -> int -> int` とすると、全体の型が整合する (どうやってチェックするか？)

型推論できるだろうか

```
let rec misc1 x y =
    let f x y = misc1 y x in
    let g x y = (f y x) + 5 in
    let h x y = g y x in
      h x (x+2)

let rec misc2 x y =
    let f x y = misc2 y x in
    let g x y = f y x in
    let h x y = g y x in
      h (x+1) ("Hello" ^ x ^ "World")
```

注. ^ は文字列を結合する関数 (中置演算子)

型推論をどうやってやればいいか？

```
let rec misc1 x y =
    let f x y = misc1 y x in
    let g x y = f y x in
    let h x y = g y x in
      (h x (x+2)) + 5
```

- `(h x (x+2))+5` から、`misc1 : int -> ? -> int`、`h : int -> int -> int` がわかる。
- `let h x y = g y x` より `g : int -> int -> int` がわかる。
- 同様に `f : int -> int -> int` がわかる。
- `let f x y = misc1 y x` より `misc1 : int -> int -> int` がわかる。

型推論アルゴリズムの準備

準備としての考察：

- 「まだわからない型」を表す型変数が必要。いくらでも多くの異なる型変数 (必要な時にいくらでも「新しい」ものを作ることができればよい)。
- 型推論の最中に持つべき情報は、対象としている項がどういう型か、また、自由変数がどういう型か
- 大きな項 (プログラム) の場合、その一部でも「型が (整合的に) つかない」なら、項全体も型がつかない。

型推論アルゴリズム (1)

型推論の対象となる項の構文 (i は整数定数):

$$e ::= i \mid (e + e) \mid x \mid (e\ e) \mid (\lambda x.e)$$

ジャッジメント $\Gamma \vdash e : T$

- 項 e の型が T であることを意味する。
- ただし、項 e に含まれる自由変数たちの型は Γ に書く。

例 1: $[x : \mathtt{Int},\ y : \mathtt{bool}] \vdash \mathtt{if}\ y\ \mathtt{then}\ x\ \mathtt{else}\ x + 1 : \mathtt{Int}$
例 2: $[] \vdash \lambda x.\lambda y.\ \mathtt{if}\ y\ \mathtt{then}\ x\ \mathtt{else}\ x + 1 : \mathtt{Int} \to (\mathtt{bool} \to \mathtt{Int})$
Γ の [..] はときどき省略。

型推論アルゴリズム (2)

(授業で配付した資料は、加算の推論規則が落ちていたので修正しました。)
ジャッジメントを「推論」するための規則:
(整数定数に対しての推論)

$$\frac{}{\Gamma \vdash i : \mathtt{Int}}$$

(変数に対しての推論)

$$\frac{(x : T \in \Gamma)}{\Gamma \vdash x : T}$$

(加算に対しての推論)

$$\frac{\Gamma \vdash e_1 : \mathtt{int} \quad \Gamma \vdash e_2 : \mathtt{int}}{\Gamma \vdash (e_1 + e_2) : \mathtt{int}}$$

型推論アルゴリズム (3)

(ラムダ抽象に対しての推論)

$$\frac{\Gamma \cup (x : T_1) \vdash e : T_2}{\Gamma \vdash \lambda x.e : T_1 \to T_2}$$

($\cup$ は 2 つの列をつなげる操作)
(関数適用に対しての推論)

$$\frac{\Gamma \vdash e_1 : T_2 \to T_1 \quad \Gamma \vdash e_2 : T_2}{\Gamma \vdash (e_1\ e_2) : T_1}$$

型推論をしてみよう

対象となる項: $\lambda x.\ x + 3$

$$\frac{\dfrac{(x : T \in [x : T])}{[x : T] \vdash x : T} \quad \dfrac{}{[x : T] \vdash 3 : \mathtt{int}}}{[x : T] \vdash x + 3 : ???}$$

ここで $T = \mathtt{int}$ とわかる。

$$\frac{\dfrac{\dfrac{(x : \mathtt{int} \in [x : \mathtt{int}])}{[x : \mathtt{int}] \vdash x : \mathtt{int}} \quad \dfrac{}{[x : \mathtt{int}] \vdash 3 : \mathtt{int}}}{[x : \mathtt{int}] \vdash x + 3 : \mathtt{int}}}{[] \vdash \lambda x.\ x + 3 : \mathtt{int} \to \mathtt{int}}$$

型推論をしてみよう

対象となる項: $\lambda x.\lambda y.\ x + y$

$$\frac{\dfrac{(x : T \in [x : T])}{[x : T, y : S] \vdash x : T} \quad \dfrac{(y : S \in [y : S])}{[x : T, y : S] \vdash y : S}}{[x : T, y : S] \vdash x + y : ???}$$

$T = S = \mathtt{int}$ となる。

$$\frac{\dfrac{\dfrac{\dfrac{(x : \mathtt{int} \in [x : \mathtt{int}])}{[x : \mathtt{int}, y : \mathtt{int}] \vdash x : \mathtt{int}} \quad \dfrac{(y : \mathtt{int} \in [y : \mathtt{int}])}{[x : \mathtt{int}, y : \mathtt{int}] \vdash y : \mathtt{int}}}{[x : \mathtt{int}, y : \mathtt{int}] \vdash x + y : \mathtt{int}}}{[x : \mathtt{int}] \vdash \lambda y.\ x + y : \mathtt{int} \to \mathtt{int}}}{[] \vdash \lambda x.\lambda y.\ x + y : \mathtt{int} \to (\mathtt{int} \to \mathtt{int})}$$

型推論をしてみよう

対象となる項: $\lambda f.\lambda x.\ (f\ (f\ (x+3)))+5$
$\Gamma_1 = [f : S \to T,\ x : \mathtt{int}]$ とする。

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma_1 \vdash f : S \to T\end{array} \quad \begin{array}{c}\vdots\\ \Gamma_1 \vdash x+3 : \mathtt{int}\end{array}}{\Gamma_1 \vdash f\ (x+3) : T}$$

$S = \mathtt{int}$ とわかる。$\Gamma_2 = [f : \mathtt{int} \to T,\ x : \mathtt{int}]$ とする。

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma_2 \vdash f : \mathtt{int} \to T\end{array} \quad \dfrac{\begin{array}{c}\vdots\\ \Gamma_2 \vdash f : \mathtt{int} \to T\end{array} \quad \begin{array}{c}\vdots\\ \Gamma_2 \vdash x+3 : \mathtt{int}\end{array}}{\Gamma_2 \vdash f\ (x+3) : T}}{\Gamma_2 \vdash f\ (f\ (x+3)) : T}$$

$T = \mathtt{int}$ とわかる。

型推論をしてみよう

(授業で配付した資料は、以下の型推論が間違っていたので修正しました。)
対象となる項: $\lambda f.\lambda x.\ (f\ (f\ (x+3)))+5$
$\Gamma_3 = [f : \mathtt{int} \to \mathtt{int},\ x : \mathtt{int}]$ とする。

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \Gamma_3 \vdash f\ (f\ (x+3)) : \mathtt{int}\end{array}}{[f : \mathtt{int} \to \mathtt{int}] \vdash \lambda x.\ f\ (f\ (x+3)) : \mathtt{int} \to \mathtt{int}}}{[\,] \vdash \lambda f.\lambda x.\ f\ (f\ (x+3)) : (\mathtt{int} \to \mathtt{int}) \to (\mathtt{int} \to \mathtt{int})}$$

完成!

問題

以下の項に対する型推論をせよ。
項 1: $\lambda f.\lambda x.\lambda y.\ (f\ x)+(f\ y)+x$
項 2: $\lambda f.\lambda x.\ f\ (f\ x)$
項 3: $\lambda f.\ f\ (f\ f)$

問題の答え (1).

項 1: $\lambda f.\lambda x.\lambda y.\ (f\ x)+(f\ y)+x$
$\Gamma_1 = [f : S, x : T, y : U]$ とする。

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_1 \vdash f : S} \quad \dfrac{}{\Gamma_1 \vdash x : T}}{\Gamma_1 \vdash f\ x : \mathtt{int}} \quad \dfrac{\dfrac{}{\Gamma_1 \vdash f : S} \quad \dfrac{}{\Gamma_1 \vdash y : U}}{\Gamma_1 \vdash f\ y : \mathtt{int}}}{\Gamma_1 \vdash (f\ x)+(f\ y) : \mathtt{int}} \quad \dfrac{}{\Gamma_1 \vdash x : \mathtt{int}}}{\Gamma_1 \vdash (f\ x)+(f\ y)+x : \mathtt{int}}$$

ここまでで、$S = T \to \mathtt{int}$, $S = U \to \mathtt{int}, T = \mathtt{int}$ となり、結局、$T = U = \mathtt{int}$ となる。よって、

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \Gamma_1 \vdash (f\ x)+(f\ y)+x : \mathtt{int}\end{array}}{f : \mathtt{int} \to \mathtt{int} \vdash \lambda x.\lambda y.(f\ x)+(f\ y)+x : \mathtt{int} \to (\mathtt{int} \to \mathtt{int})}}{\vdash \lambda f.\lambda x.\lambda y.\ (f\ x)+(f\ y)+x : (\mathtt{int} \to \mathtt{int}) \to (\mathtt{int} \to (\mathtt{int} \to \mathtt{int}))}$$

問題の答え (2).

項 2: $\lambda f.\lambda x.\ f\ (f\ x)$
$\Gamma_2 = [f : S, x : T]$ とする。

$$\dfrac{\dfrac{}{\Gamma_1 \vdash f : V \to U} \qquad \dfrac{\dfrac{}{\Gamma_1 \vdash f : W \to V} \quad \dfrac{}{\Gamma_1 \vdash x : W}}{\Gamma_1 \vdash f\ x : V}}{\Gamma_1 \vdash f\ (f\ x) : U}$$

これより、$S = V \to U = W \to V$ かつ $T = W$ であり、
$T = W = V = U$ となる。よって、

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \Gamma_1 \vdash f\ (f\ x) : T\end{array}}{f : V \to U \vdash \lambda x.\ f\ (f\ x) : T \to T}}{\vdash \lambda f.\lambda x.\ f\ (f\ x) : (T \to T) \to (T \to T)}$$

これで型推論は完成ある。つまり、上記の項はどんな型 T に対しても $(T \to T) \to (T \to T)$ という型を持つ。

問題の答え (3).

項 3: $\lambda f.\ f\ (f\ f)$
$\Gamma_3 = [f : S]$ とする。

$$\dfrac{\dfrac{}{\Gamma_1 \vdash f : S} \quad \dfrac{}{\Gamma_1 \vdash f : S}}{\Gamma_1 \vdash f\ f : V}$$

これより、$S = S \to V$ である。しかし、これは S が有限長の型であるかぎり満たすことができない等式である。よって、項 $f\ f$ は型がつかない。したがって、$f\ f$ を一部として含む $\lambda f.\ f\ (f\ f)$ も型がつかない。