ソフトウェア構成特論 第7回

大学院理工学研究科 電気電子情報工学専攻　篠埜 功

2014年5月29日

1　はじめに

今回は、以前紹介した算術式に型システムを導入する。

2　算術式の型

まず、以前、算術式は以下のように定義した。

```
t ::= true
    | false
    | if t then t else t
    | 0
    | succ t
    | pred t
    | iszero t
```

値は以下のように定義した。

```
v ::= true
    | false
    | nv
nv ::= 0
     | succ nv
```

算術式を評価すると、値まで評価されるか、あるいは`pred false`のような算術式になって適用できる評価規則がなくなり、評価が *stuck* するかのいずれかである。今回目標とするのは、与えられた算術式を評価せずに、評価結果がstuckしない、つまり、正規形（適用できる規則がない算術式）が値であることを保証したいということである。算術式においては値はブール値（`true`と`false`）か、数値(メタ変数`nv`で表している）のいずれかであるので、評価しようとしている算術式が、評価結果がブール値になる算術式なのか、評価結果が数値になる算術式なのかが評価前に判断できればよい。そこで、`Nat`と`Bool`という2つの**型** (*type*) を導入し、算術式に対して**型システム** (*type system*) を導入する。今後、「算術式`t`は型`T`を持つ」という言い方をする。今回提示する型システムにより、算術式`t`が型`T`を持つとき、算術式`t`の評価結果はその型`T`の値になるということを示す。例

えば、算術式 `if true then false else true` は型 `Bool` を持ち、それにより、評価結果の型が `Bool` であることが言える。実際、この算術式の評価結果は `false` であり、`false` は型 `Bool` を持つ。また、算術式 `pred (succ (pred (succ 0)))` は型 `Nat` を持ち、それにより、評価結果の型が `Nat` であることが言える。この算術式の評価結果は `0` であり、`0` は型 `Nat` を持つ。

3 算術式の型システム

まず、型は

$$\mathtt{T} ::= \mathtt{Bool} \mid \mathtt{Nat}$$

と定義する。算術式 `t` が型 `T` を持つ場合、`t:T` と書き、これを**型判定** (*typing judgement* あるいは *typing assertion*) という。この : は型と算術式の間の二項関係であり、以下の 7 個の規則（**型付け規則** (*typing rule*) という）により定義する。また、この二項関係 : を**型付け関係** (*typing relation*) という。

$$\frac{}{\mathtt{true} : \mathtt{Bool}}\ (\text{T-TRUE}) \quad \frac{}{\mathtt{false} : \mathtt{Bool}}\ (\text{T-FALSE}) \quad \frac{\mathtt{t_1} : \mathtt{Bool} \quad \mathtt{t_2} : \mathtt{T} \quad \mathtt{t_3} : \mathtt{T}}{\mathtt{if}\ \mathtt{t_1}\ \mathtt{then}\ \mathtt{t_2}\ \mathtt{else}\ \mathtt{t_3} : \mathtt{T}}\ (\text{T-IF})$$

$$\frac{}{\mathtt{0} : \mathtt{Nat}}\ (\text{T-ZERO}) \quad \frac{\mathtt{t_1} : \mathtt{Nat}}{\mathtt{succ}\ \mathtt{t_1} : \mathtt{Nat}}\ (\text{T-SUCC}) \quad \frac{\mathtt{t_1} : \mathtt{Nat}}{\mathtt{pred}\ \mathtt{t_1} : \mathtt{Nat}}\ (\text{T-PRED})$$

$$\frac{\mathtt{t_1} : \mathtt{Nat}}{\mathtt{iszero}\ \mathtt{t_1} : \mathtt{Bool}}\ (\text{T-ISZERO})$$

以前、ブール式の small step semantics で 1 ステップ評価関係 $\rightarrow$ を定義した時と同様、算術式と型の間の二項関係 : は形式的には以下のように定義される。

定義 1. 算術式と型の間の二項関係 : は、上記 7 個の規則を満たす最小の関係と定義する。

上記型システムの定義より、以下の補題が成り立つ。

補題 1 (Inversion lemma)**.**

1. `true:R` ならば `R=Bool`
2. `false:R` ならば `R=Bool`
3. `if` $\mathtt{t_1}$ `then` $\mathtt{t_2}$ `else` $\mathtt{t_3} : \mathtt{R}$ ならば $\mathtt{t_1} : \mathtt{Bool}$ かつ $\mathtt{t_2} : \mathtt{R}$ かつ $\mathtt{t_3} : \mathtt{R}$
4. `0:R` ならば `R=Nat`
5. `succ` $\mathtt{t_1} : \mathtt{R}$ ならば `R=Nat` かつ $\mathtt{t_1} : \mathtt{Nat}$
6. `pred` $\mathtt{t_1} : \mathtt{R}$ ならば `R=Nat` かつ $\mathtt{t_1} : \mathtt{Nat}$
7. `iszero` $\mathtt{t_1} : \mathtt{R}$ ならば `R=Bool` かつ $\mathtt{t_1} : \mathtt{Nat}$

証明. 型システムの定義から明らか。 □

算術式の評価関係の導出木は算術式の評価規則のインスタンスが繋がってできた木であった。これと同様、**型判定の導出木** (*typing derivation*) は型付け規則のインスタンスが繋がってできた木である。例えば、型判定 `if iszero 0 then 0 else pred 0 : Nat` の導出木は以下のように構築できる。

$$\dfrac{\dfrac{\dfrac{}{\texttt{0 : Nat}}\,\text{(T-ZERO)}}{\texttt{iszero 0 : Bool}}\,\text{(T-ISZERO)} \quad \dfrac{}{\texttt{0 : Nat}}\,\text{(T-ZERO)} \quad \dfrac{\dfrac{}{\texttt{0 : Nat}}\,\text{(T-ZERO)}}{\texttt{pred 0 : Nat}}\,\text{(T-PRED)}}{\texttt{if iszero 0 then 0 else pred 0 : Nat}}\,\text{(T-IF)}$$

練習問題 1. 型判定 `if true then pred 0 else succ 0 : Nat` に対する導出木を構築せよ。

定理 1. 各算術式 `t` は 2 つ以上の型は持たない。つまり、算術式は、型を持たないか、あるいは、型を持つなら 1 つの型に定まる。さらに、算術式 `t` が型を持つならその型判定に対する導出木は唯一に定まる。

証明. 算術式 `t` の構造に関する帰納法で証明できるが、この講義では証明は省略する。 □

補足 1. 部分型（subtyping）のある型システムにおいては、1 つの term が 2 つ以上の型を持つ場合があり、1 つの型判定の導出木が複数ある場合がある。

補足 2. 一般に、型システムは評価が停止することを保証する訳でない。つまり、ある term `t` が型 `T` を持つことが言えた場合は、その term `t` を評価して正規形になった場合に、それが値であり、その型が `T` であることが保証されるということである。算術式は評価が必ず停止するが、一般の計算体系では評価が停止しない term もあり、そういう term も型を持ちうる。

補足 3. 型を持たない term でも評価結果が値になる場合がある。例えば、算術式 `if (iszero 0) then 0 else false` は型を持たないが、

$$\dfrac{}{\texttt{iszero 0} \to \texttt{true}}\,\text{(E-ISZEROZERO)}$$
$$\dfrac{\qquad\qquad}{\texttt{if (iszero 0) then 0 else false} \to \texttt{if true then 0 else false}}\,\text{(E-IF)}$$

と

$$\dfrac{}{\texttt{if true then 0 else false} \to \texttt{0}}\,\text{(E-IFTRUE)}$$

より、評価結果は `0` という値になる。

4 型システムの健全性

型の付いた算術式は評価時に stuck しない（値でない正規形にはならない）ということをこれから示す。これは型システムの最も基本的な性質であり、型システムの**健全性** (*soundness*) という。これを以下の 2 つを示すことにより示す。

> *progress*: 型の付いた算術式は値であるか、あるいは、1 ステップ評価できる。
> *preservation*: 型の付いた算術式が 1 ステップ評価されたならば、その結果の算術式には型が付く。

この 2 つが示せたら、健全性が示せたことになる。
progress を示すために Bool 型と Nat 型の値がある特定の形をしていることを示す。

補題 2 (算術式の**標準形** (*canonical form*))**.**

1. もし v が Bool 型の値ならば、v は true か false である。
2. もし v が Nat 型の値ならば、v は数値である。数値は以下の文法で定義されるものである。

$$\mathtt{nv} ::= \mathtt{0} \mid \mathtt{succ\ nv}$$

証明. まず、1 を示す。v が Bool 型であると仮定する。値 v は以下の 4 通りの形を取りうるのでこの 4 通りで場合分けする。

$$\mathtt{true}, \mathtt{false}, \mathtt{0}, \mathtt{succ\ nv}$$

inversion lemma の 4 と 5 から、0 と succ nv の形の算術式は Nat 型しか持てず除外される。また、true と false は Bool 型を持つ。よって、v は true か false のいずれかである。
次に 2 を示す。v が Nat 型であると仮定する。値 v は以下の 3 通りの形を取りうるのでこの 3 通りで場合分けする。

$$\mathtt{true}, \mathtt{false}, \mathtt{nv}$$

inversion lemma の 1 と 2 から、true と false は Bool 型しか持てず除外される。また、任意の数値 nv は Nat 型を持つ (*)。よって v は数値である。 □

(*) 任意の数値 nv が Nat 型を持つことは以下のようにして証明できる。

証明. nv の構造に関する帰納法で証明する。まず、(T-ZERO) 規則より 0 は Nat 型を持つ。次に、nv が Nat 型を持つ (nv : Nat が成立する) ことを仮定する。(T-SUCC) 規則を適用すると succ nv : Nat が得られる。以上より任意の数値 nv は Nat 型を持つ。 □

定理 2 (Progress)**.** t が型の付く算術式とする。つまり、何らかの T に対して t : T が成り立つとする。このとき、t は値であるか、あるいは何らかの t′ に対して t → t′ が成り立つ。

証明. 型判定 t : T の導出に関する帰納法で証明する。型判定 t : T の導出木において最後に使われた規則（一番下の規則）で場合分けをする。T-TRUE, T-FALSE, T-ZERO の場合は、t が値である。T-IF の場合を以下に記述し、残りの場合を練習問題とする。

T-IF の場合: 型判定の導出木の一番下の部分は

$$\frac{\mathtt{t_1} : \mathtt{Bool} \quad \mathtt{t_2} : \mathtt{T} \quad \mathtt{t_3} : \mathtt{T}}{\mathtt{if\ t_1\ then\ t_2\ else\ t_3} : \mathtt{T}}\ \text{(T-IF)}$$

という形をしている。帰納法の仮定より、$\mathtt{t_1}$ は値であるか、何らかの算術式 $\mathtt{t_1}'$ について $\mathtt{t_1} \to \mathtt{t_1}'$ が成り立つ。もし $\mathtt{t_1}$ が値なら、標準形の補題（補題 2）より、$\mathtt{t_1}$ は true か false であり、t に対して E-IFTRUE 規則か E-IFFALSE 規則が適用できる。もし何らかの算術式 $\mathtt{t_1}'$ について $\mathtt{t_1} \to \mathtt{t_1}'$ が成り立つなら、E-IF 規則により、$\mathtt{t} \to \mathtt{if\ t_1'\ then\ t_2\ else\ t_3}$ が成立する。 □

練習問題 2. 上記の定理の証明の残りの場合（T-SUCC, T-PRED, T-ISZERO の場合）を示せ。