

Coq ゼミ 第7回 (演習)

浅井研究室 (担当: 廣田知子)

2008年7月24日

今回は演習です。Coq上で簡単な型システムを実際に実装し、その簡単な定理を証明してみましょう。型システムにはラムダ計算等は導入せず、booleanと自然数のみを用います。

(実際には「Types and Programming Languages」の8章のLanguageと同じものを定義していきます。)

1 構文

まずはじめに構文を以下に示します:

```
(* term *)
t :=
  tm_true
  tm_false
  tm_if t t t
  tm_zero
  tm_succ t
  tm_pred t
  tm_iszero t
```

```
(* value *)
v :=
  tm_true
  tm_false
  nv
```

```
nv :=
  tm_zero
  tm_succ nv
```

2 簡約規則

「 t は t' へと簡約出来る」という規則は、ここでは $t \rightarrow t'$ と表記する。

```
(* evaluation rule *)
e_iftrue :
```

```
(tm_if tm_true t2 t3) --> t2
e_iffalse :
  (tm_if tm_false t2 t3) --> t3
e_if :
  t1 --> t1'
-----
(tm_if t1 t2 t3) --> (tm_if t1' t2 t3)
e_succ :
  t --> t'
-----
tm_succ t --> tm_succ t'
e_predzero :
  tm_pred tm_zero --> tm_zero
e_predsucc :
  tm_pred (tm_succ nv) --> nv
e_pred :
  t --> t'
-----
tm_pred t --> tm_pred t'
e_iszerozero :
  tm_iszero tm_zero --> tm_true
e_iszerosucc :
  tm_iszero (tm_succ nv) --> tm_false
e_iszero :
  t --> t'
-----
tm_iszero t --> tm_iszero t'
```

3 型

```
(* type *)
T :=
  Bool
  Nat
```

4 型付け規則

「 t は T という型を持つ」という型規則は、ここでは $t \sim : T$ と表記する。

```

(* typing rule *)
typing_true : tm_true ~: Bool
typing_false : tm_false ~: Bool
typing_if :
  t1 ~: Bool   t2 ~: T   t3 ~: T
  -----
  tm_if t1 t2 t3 ~: T
typing_zero : tm_zero ~: Nat
typing_succ :
  t ~: Nat
  -----
  tm_succ t ~: Nat
typing_pred :
  t ~: Nat
  -----
  tm_pred t ~: Nat
typing_iszero :
  t ~: Nat
  -----
  tm_iszero t ~: Bool

```

演習問題

問 1

Coq 上で `term`, `value` を定義せよ。

ヒント: `term` は `t` という型を持つ要素として (つまり帰納的なデータ型として) 定義する。`value` は帰納的な命題として定義する。これを使って、例えば `Lemma le : value tm_true. 「tm_true は value である」` のような補題等を作ることが出来る。

問 2

簡約規則を定義せよ。その際の名前は `eval` とせよ。

ヒント: これも帰納的な命題となる。例えば、

```
Lemma le : eval (tm_pred tm_zero) tm_zero.
```

とすれば、これは「`(tm_pred tm_zero)` は `tm_zero` へと簡約出来る」という意味の補題となる。

問 3

型 `Bool` と `Nat` を定義し、そのあと型付け規則 `typing` を定義せよ。

問 4

「`t` が `nv` ならば、`(tm_succ (tm_pred (tm_succ t)))` は `(tm_succ t)` へと簡約できる」という補題 `eval_succ_pred_succ` を証明せよ。

問 5

「`t` が `nv` ならば、`t` は `Nat` 型を持つ」という補題 `type_nvalue` を証明せよ。

問 6

「`(tm_if t1 t2 t3)` が `T` 型を持つなら、`t1` は `Bool` 型を持ち、`t2` は `T` 型を持ち、`t3` は `T` 型を持つ」という補題 `type_if` を証明せよ。

ヒント: `inversion H`. 仮定 `H` に対して使うと、`H` が成り立つためには前提条件として何が成立していなくてはならないかを勝手に推論して仮定に追加してくれる。