

Coqによるプログラムの検証入門

集中講義 千葉大学大学院

アフエルト レナルド

産業技術総合研究所

2017 年 01 月 25–26 日

目 次

- 1 依存型を用いた自然数の引き算 1
- 2 ホーア論理を用いて、階乗のプログラムを検証する 2

1 依存型を用いた自然数の引き算

授業の `pred` 関数のように、引き算を実装する。具体的に、 m と n は自然数とする。その場合、 $m - n$ は $m \geq n$ が成り立つときにしか定義されていない。

自然数の引き算を行う完全な関数の例:

```
Fixpoint tminus (n m : nat) : nat :=  
  match n with  
  | 0 => 0  
  | S n' => match m with  
            | 0 => n  
            | S m' => tminus n' m'  
          end  
end.  
end.
```

Compute `tminus 5 3`.

Compute `tminus 5 6`.

次の型を持つ引き算関数を実装する:

```
Require Import Arith.
```

```
Fixpoint pminus (n m : nat) : m ≤ n → nat.
```

```
Abort.
```

提案した関数をテストする.

```
Lemma O_le_5 : 3 ≤ 5.
```

```
Proof.
```

auto.

Qed.

Compute *pminus* 5 3 *O_le_5*.

次の型を持つ引き算の関数を実装する。ただし、対話的な方法を使う。(時間があれば、直接に記述してみる.)

Fixpoint *iminus* (*n m* : nat) : $m \leq n \rightarrow \{ k : nat \mid k + m = n \}$.

Abort.

2 ホーア論理を用いて、階乗のプログラムを検証する

授業で紹介したホーア論理を用いて、階乗のプログラムを検証する。仕様として、Coq の *Factorial* 標準ライブラリの *fact* 関数を利用する。

授業で見たホーア論理を用いて、次の証明を完成させる。

Require Import *Omega*.

Require Import *Factorial*.

Lemma *facto_fact* *x X ret* : $x \neq ret \rightarrow$

hoare

($\text{fun } s \Rightarrow \text{eval } (\text{exp_var } x) s = X \wedge$

$\text{eval } (\text{exp_var } ret) s = 1$)

(*facto* *x ret*)

($\text{fun } s \Rightarrow \text{eval } (\text{exp_var } ret) s = \text{fact } X$).

Proof.

intros *xret*.

set ($P' := \text{fun } s : \text{state} \Rightarrow \text{eval } (\text{exp_var } ret) s \times$
fact ($\text{eval } (\text{exp_var } x) s = \text{fact } X$).

set ($Q' := \text{fun } s : \text{state} \Rightarrow \text{eval } (\text{exp_var } ret) s \times$
fact ($\text{eval } (\text{exp_var } x) s = \text{fact } X \wedge$
 $\neg (\text{beval } (\text{neg } (\text{equa } (\text{exp_var } x) (\text{cst } O))) s$)).

apply (*hoare_conseq* $P' Q'$).