ソフトウェア基礎科学：Coq 演習課題

2012年1月13日

Exercise 2-1

Enter the following commands in Coq. What does the output for the command "`Check mynat_ind.`" mean?

```
Inductive mynat : Set:=
  |  Z: mynat
  |  S: mynat -> mynat.

Check mynat_ind.

Definition One := S Z.
Definition Two := S One.

Fixpoint plus (m n : mynat) {struct m} : mynat :=
  match m with
     Z => n
   | (S m') => S(plus m' n)
  end.

Eval compute in (plus One Two).
```

Exercise 2-2

Prove the following theorems. (Replace the part "..." with proof commands.)

```
Theorem plus_unit_l:
  forall n:mynat, (plus Z n)  = n.
```

```
Proof.
 ...
Qed.

Theorem plus_unit_r:
  forall n:mynat, (plus n Z) = n.
Proof.
 ...
Qed.

Theorem plus_is_associative:
  forall k m n: mynat,
     (plus k (plus m n)) = (plus (plus k m) n).
Proof.
 ...
Qed.

Lemma plus_m_Sn:
   forall m n:mynat, (plus m (S n)) = S (plus m n).
Proof.
 ...
Qed.

Theorem plus_is_commutative:
  forall m n:mynat, (plus m n) = (plus n m).
Proof.
 ...
Qed.
```

Exercise 2-3

Define a function `mult` for multiplication.

Exercise 2-4

Prove the following theorems.

```
Theorem mult_zero_r:
  forall n: mynat, (mult n Z) = Z.
```

```
Proof.
...
Qed.

Theorem mult_unit_l:
  forall n:mynat, (mult One n) = n.
Proof.
...
Qed.

Theorem mult_unit_r:
  forall n:mynat, (mult n One) = n.
Proof.
...
Qed.

Theorem mult_distr_l:
   forall k m n: mynat,
      (mult (plus k m) n) = (plus (mult k n) (mult m n)).
Proof.
...
Qed.

Theorem mult_is_associative:
  forall k m n : mynat,
    (mult k (mult m n)) = (mult (mult k m) n).
Proof.
...
Qed.

Lemma mult_m_Sn:
  forall m n: mynat,
    (mult m (S n)) = (plus (mult m n) m).
Proof.
...
Qed.

Theorem mult_is_commutative:
  forall m n : mynat,
```

```
   (mult m n) = (mult n m).
Proof.
...
Qed.
```

Exercise 2-5 (Optional)

Define a function `exp` for exponentiation (`exp` $m\ n = m^n$), and prove properties of `exp`.