第21章： 等式理论

胡振江，张伟

计算机学院

2022年12月6日

等式理论

```
infix 4 _≡_
data _≡_ {A : Set} (x : A) : A → Set where
    refl : x ≡ x
```

```
sym : ∀ {A : Set} {x y : A}
    → x ≡ y
      -----
    → y ≡ x
sym refl = refl
```

```
trans : ∀ {A : Set} {x y z : A}
    → x ≡ y
    → y ≡ z
     ------
    → x ≡ z
```

```
cong : ∀ {A B : Set} (f : A → B) {x y : A}
    → x ≡ y
      ----------
    → f x ≡ f y
```

```
cong-app : ∀ {A B : Set} {f g : A → B}
    → f ≡ g
      ------------------------
    → ∀ (x : A) → f x ≡ g x
```

```
subst : ∀ {A : Set} {x y : A} (P : A → Set)
    → x ≡ y
      ----------
    → P x → P y
```

等式推理

```
infix      1 begin_
Infixr     2 _≡⟨⟩_ _≡⟨_⟩_
infix      3 _∎
```

```
begin_ : ∀ {x y : A}
    → x ≡ y
      -----
    → x ≡ y
begin x≡y = x≡y
```

推理开始

```
_≡⟨⟩_ : ∀ (x : A) {y : A}
    → x ≡ y
      -----
    → x ≡ y
x ≡⟨⟩ x≡y = x≡y
```

从x新开始
经过等式变换

```
_≡⟨_⟩_ : ∀ (x : A) {y z : A}
    → x ≡ y
    → y ≡ z
      -----
    → x ≡ z
x ≡⟨ x≡y ⟩ y≡z = trans x≡y y≡z
```

从x开始经过
等式传递变换

```
_∎ : ∀ (x : A)
      -----
    → x ≡ x
x ∎ = refl
```

证明结束

```
trans′ : ∀ {A : Set} {x y z : A}
    → x ≡ y
    → y ≡ z
      -----
    → x ≡ z
trans′ {A} {x} {y} {z} x≡y y≡z =
  begin
    x
  ≡⟨ x≡y ⟩
    y
  ≡⟨ y≡z ⟩
    z
  ∎
```

```
trans′ : ∀ {A : Set} {x y z : A}
    → x ≡ y
    → y ≡ z
      -----
    → x ≡ z
trans′ {A} {x} {y} {z} x≡y y≡z =
  begin
     x
  ≡⟨ x≡y ⟩
     y
  ≡⟨ y≡z ⟩
     z
  ∎
```

自然数上的证明例

```
data ℕ : Set where
    zero : ℕ
    suc : ℕ → ℕ

_+_ : ℕ → ℕ → ℕ
zero + n = n
(suc m) + n = suc (m + n)

postulate
    +-identity : ∀ (m : ℕ) → m + zero ≡ m
    +-suc : ∀ (m n : ℕ) → m + suc n ≡ suc (m + n)
```

如何通过等式变换证明加法的交换性?

```
+-comm : ∀ (m n : ℕ) → m + n ≡ n + m
```

```
+-comm m zero =
 begin
   m + zero
 ≡⟨ +-identity m ⟩
   m
 ≡⟨⟩
   zero + m
 ∎
+-comm m (suc n) =
 begin
   m + suc n
 ≡⟨ +-suc m n ⟩
   suc (m + n)
 ≡⟨ cong suc (+-comm m n) ⟩
   suc (n + m)
 ≡⟨⟩
  suc n + m
 ∎
```

作业

21-1. 利用等式推理证明下面两个性质：

```
+-identity : ∀ (m : ℕ) → m + zero ≡ m

+-suc : ∀ (m n : ℕ) → m + suc n ≡ suc (m + n)
```