



软件科学基础

# Maps: Total and Partial Maps

熊英飞  
北京大学



# 复习：Lists章节的Map定义

```
Inductive id : Type :=  
  | Id (n : nat).  
  
Definition eqb_id (x1 x2 : id) :=  
  match x1, x2 with  
  | Id n1, Id n2 => n1 =? n2  
  end.
```



# 复习：Lists章节的Map定义

```
Inductive partial_map : Type :=
  | empty
  | record (i : id) (v : nat) (m : partial_map).

Definition update (d : partial_map)
  (x : id) (value : nat)
  : partial_map :=
  record x value d.

Fixpoint find (x : id) (d : partial_map) : natoption :=
  match d with
  | empty          => None
  | record y v d' => if eqb_id x y
                     then Some v
                     else find x d'
  end.
```



# 重新用函数定义Map

可以应用函数相关的定理，使得一些证明更简单

```
Definition eqb_string (x y : string) : bool :=  
  if string_dec x y then true else false.
```

```
Definition total_map (A : Type) := string -> A.
```

```
Definition t_empty {A : Type} (v : A) : total_map A :=  
  (fun _ => v).
```

```
Definition t_update {A : Type} (m : total_map A)  
  (x : string) (v : A) :=  
  fun x' => if eqb_string x x' then v else m x'.
```



# 重新用函数定义Map

```
Notation "'__' '!->' v" := (t_empty v)
  (at level 100, right associativity).
```

```
Notation "x '!->' v ';' m" := (t_update m x v)
  (at level 100, v at next level, right associativity).
```

```
Definition examplemap' :=
  ( "bar" !-> true;
    "foo" !-> true;
    _     !-> false
  ).
```



# 重新用函数定义Map

```
Definition partial_map (A : Type) := total_map (option A
).
```

```
Definition empty {A : Type} : partial_map A :=
  t_empty None.
```

```
Definition update {A : Type} (m : partial_map A)
  (x : string) (v : A) :=
  (x !-> Some v ; m).
```

```
Notation "x '|->' v ';' m" := (update m x v)
  (at level 100, v at next level, right associativity).
```

```
Notation "x '|->' v" := (update empty x v)
  (at level 100).
```

```
Example examplemap :=
  ("Church" |-> true ; "Turing" |-> false).
```



# 作业

- 完成Maps中2道standard非optional的习题
  - 请使用最新英文版教材