Aula 6 - Duplas de Naturais

Rafael Castro - rafaelcgs10.github.io/coq

14/05/2018

Produto

- Produto, par ordenado, dupla etc. . . .
- Uma sequência finita de elementos, cuja ordem importa.
- Os elementos de uma dupla podem ser de tipos diferentes: (1, 'a'). Mas não por hora.

Tipo Produdo de Naturais em Coq

```
Inductive natprod : Type :=
| pair : nat -> nat -> natprod.

Check (pair 3 5).
```

Projeções do Produto

- Definimos duas funções que projetam o primeiro e o segundo elemento do par.

```
Definition fst (p : natprod) : nat :=
  match p with
  | pair x y => x
  end.

Definition snd (p : natprod) : nat :=
  match p with
  | pair x y => y
  end.

Compute (fst (pair 3 5)).
(* ===> 3 *)
```

Uma notação para facilitar a vida

- Definimos uma notação para duplas.

```
Notation "( x , y )" := (pair x y).
```

Função *swap*

- Definimos uma função que troca os elementos de uma dupla.

```
Definition swap_pair (p : natprod) : natprod :=
  match p with
  | (x,y) => (y,x)
  end.
```

Alguns fatos sobre duplas

- Vamos demonstrar alguns fatos simples sobre o tipo dupla de naturais.

```
Theorem surjective_pairing' : forall (n m : nat),
  (n,m) = (fst (n,m), snd (n,m)).
Proof.
  reflexivity.  Qed.
```

Um caso que a simplificação não funciona

- Se tentarmos utilizar a simplificação aqui não vai ser possível.
- Ainda que *p* seja um *natprod* para fazer o casamento de padrão é necessário expor a sua estrutura para fazer o casamento de padrão.

```
Theorem surjective_pairing_stuck : forall (p : natprod),
  p = (fst p, snd p).
Proof.
  simpl. (* Doesn't reduce anything! *)
Abort.

Theorem surjective_pairing : forall (p : natprod),
  p = (fst p, snd p).
Proof.
  intros p.  destruct p as [n m].  simpl.  reflexivity.  Qed
```