Chapter 1

Library c01_basics

Capítulo 1 - Functional Programming in Coq (Basics)

```
Require Import Notations.
```

```
Require Import Nat.
```

```
Local Open Scope nat_scope.
```

Exercise: 1 star, standard (nandb)

```
Definition nandb (b1 b2: bool) : bool :=
  match (b1, b2) with
    | (true, true) ⇒ false
    | _ ⇒ true
  end.
```

```
Example test_nandb1: (nandb true false) = true.
```

```
Example test_nandb2: (nandb false false) = true.
```

```
Example test_nandb3: (nandb false true) = true.
```

```
Example test_nandb4: (nandb true true) = false.
```

Exercise: 1 star, standard (andb3)

```
Definition andb3 (b1 b2 b3: bool) : bool :=
  match (b1, b2, b3) with
    | (true, true, true) ⇒ true
    | _ ⇒ false
  end.
```

```
Example test_andb31: (andb3 true true true) = true.
```

```
Example test_andb32: (andb3 false true true) = false.
```

```
Example test_andb33: (andb3 true false true) = false.
```

```
Example test_andb34: (andb3 true true false) = false.
```

Exercise: 1 star, standard (factorial)

```
Fixpoint factorial (n: nat) : nat :=
  match n with
    | O ⇒ 1
    | S n' ⇒ n × (factorial n')
  end.

Example test_factorial1: (factorial 3) = 6.

Example test_factorial2: (factorial 5) = (mult 10 12).
```

Exercise: 1 star, standard (ltb)

```
Definition ltb (n m : nat) : bool :=
  andb (leb n m) (negb (eqb n m)).

Example test_ltb1: (ltb 2 2) = false.

Example test_ltb2: (ltb 2 4) = true.

Example test_ltb3: (ltb 4 2) = false.
```

Exercise: 1 star, standard (plus_id_exercise)

```
Theorem plus_id_exercise : ∀ n m o : nat,
  n = m → m = o →
  n + m = m + o.
```

Exercise: 2 stars, standard (mult_S_1)

```
Theorem mult_S_1 : ∀ n m : nat,
  m = S n →
  m × (1 + n) = m × m.
```

Exercise: 2 stars, standard (andb_true_elim2)

```
Theorem andb_true_elim2 : ∀ b c : bool,
  andb b c = true →
  c = true.
```

Exercise: 1 star (zero_nbeq_plus_1)

```
Theorem zero_nbeq_plus_1 : ∀ n : nat,
  0 =? (n + 1) = false.
```

Exercise: 1 star, standard (indentity_fn_applied_twice)

```
Theorem identity_fn_applied_twice :
  ∀ (f : bool → bool), (∀ (x : bool), f x = x) →
  ∀ (b : bool), f (f b) = b.
```

Exercise: 1 star, standard (negation_fn_applied_twice)

```
Theorem negb_involutive : ∀ b : bool,
  negb (negb b) = b.

Theorem negation_fn_applied_twice :
```

$\forall\ (f : bool \to bool),\ (\forall\ (x : bool),\ f\ x = negb\ x) \to$
$\forall\ (b : bool),\ f\ (f\ b) = b.$

Exercise: 2 stars (andb_eq_orb)

```
Theorem andb_eq_orb : ∀ (b c : bool),
  andb b c = orb b c →
  b = c.
```

Exercise: 3 stars, standard (binary)

```
Inductive bin : Type :=
  | Z : bin
  | A : bin → bin
  | B : bin → bin.

Fixpoint incr (m : bin) : bin :=
  match m with
  | Z ⇒ B Z
  | A m' ⇒ B m'
  | B m' ⇒ A (incr m')
  end.

Fixpoint bin_to_nat (m : bin) : nat :=
  match m with
  | Z ⇒ O
  | A m' ⇒ mult 2 (bin_to_nat m')
  | B m' ⇒ S (mult 2 (bin_to_nat m'))
  end.

Example inc_three_four: (bin_to_nat (incr (B (B Z)))) = 4.

Example inc_nine_ten: (bin_to_nat (incr (B (A (A (B Z)))))) = 10.

Example zero_is_zero: (bin_to_nat Z) = 0.

Example five_is_five: (bin_to_nat (B (A (B Z)))) = 5.

Fixpoint incN (n : nat) (m : bin) :=
  match n with
  | 0 ⇒ m
  | S n' ⇒ incN n' (incr m)
  end.

Example SanityCheck: bin_to_nat (incN 15 Z) = 15.
```