Chapter 1

Library c06_logic

Capítulo 6 -

From LF Require Export $c05\_tactics$.

Theorem $plus\_2\_2\_is\_4$:
  $2 + 2 = 4$.

Definition $plus\_fact$: Prop := $2 + 2 = 4$.

Theorem $plus\_fact\_is\_true$:
  $plus\_fact$.

Definition $injective$ $\{A\ B\}$ $(f : A \to B)$:=
  $\forall\ x\ y : A, f\ x = f\ y \to x = y$.

Lemma $succ\_inj$: $injective\ S$.

Example $and\_example$: $3 + 4 = 7 \land 2 \times 2 = 4$.

Lemma and_intro : $\forall\ A\ B$: Prop,
  $A \to$
  $B \to$
  $A \land B$.

Example $and\_example'$: $3 + 4 = 7 \land 2 \times 2 = 4$.

Exercise: 2 stars (and_exercise) Example $and\_exercise$:
  $\forall\ n\ m : nat, n + m = 0 \to n = 0 \land m = 0$.

Lemma $and\_example2$:
  $\forall\ n\ m : nat, n = 0 \land m = 0 \to n + m = 0$.

Lemma $and\_example2'$:
  $\forall\ n\ m : nat, n = 0 \land m = 0 \to n + m = 0$.

Lemma $and\_example2''$:
  $\forall\ n\ m : nat, n = 0 \to m = 0 \to n + m = 0$.

Lemma $and\_example3$:

$\forall$ n m : nat, $n + m = 0 \rightarrow n \times m = 0$.

`Lemma` *proj1* : $\forall$ P Q : `Prop`,
$P \wedge Q \rightarrow P$.

Exercise: 1 star, optional (proj2) `Lemma` *proj2* : $\forall$ P Q : `Prop`,
$P \wedge Q \rightarrow Q$.

`Theorem` *and_commut* : $\forall$ P Q : `Prop`,
$P \wedge Q \rightarrow Q \wedge P$.

Exercise: 2 stars (and_assoc) `Theorem` *and_assoc* : $\forall$ P Q R : `Prop`,
$P \wedge (Q \wedge R) \rightarrow (P \wedge Q) \wedge R$.

`Lemma` *or_example* :
$\forall$ n m : nat, $n = 0 \vee m = 0 \rightarrow n \times m = 0$.

`Lemma` *or_intro* : $\forall$ A B : `Prop`, $A \rightarrow A \vee B$.

`Lemma` *or_intro'* : $\forall$ A B : `Prop`, $B \rightarrow A \vee B$.

`Lemma` *zero_or_succ* :
$\forall$ n : nat, $n = 0 \vee n = S$ $(pred\ n)$.

Exercise: 1 star (mult_eq_0) `Lemma` *mult_eq_0* :
$\forall$ n m, $n \times m = 0 \rightarrow n = 0 \vee m = 0$.

Exercise: 1 star (or_commut) `Theorem` *or_commut* : $\forall$ P Q : `Prop`,
$P \vee Q \rightarrow Q \vee P$.

`Theorem` *ex_falso_quodlibet* : $\forall$ (P:`Prop`),
False $\rightarrow P$.

Exercise: 2 stars, optional (not_implies_our_not) `Fact` *not_implies_our_not* : $\forall$ (P:`Prop`),
$\neg$ $P \rightarrow$ ($\forall$ (Q:`Prop`), $P \rightarrow Q$).

`Notation` "x <> y" := ($\tilde{}(x = y)$).

`Theorem` *zero_not_one* : $\tilde{}(0 = 1)$.

`Theorem` *zero_not_one'* : $0 \neq 1$.

`Theorem` *not_False* :
$\neg$ *False*.

`Theorem` *contradiction_implies_anything* : $\forall$ P Q : `Prop`,
$(P \wedge \neg P) \rightarrow Q$.

`Theorem` *double_neg* : $\forall$ P : `Prop`,
$P \rightarrow \tilde{}\tilde{}P$.

Exercise: 2 stars, recommended (contrapositive) `Theorem` *contrapositive* : $\forall$ (P Q : `Prop`),
$(P \rightarrow Q) \rightarrow (\tilde{}Q \rightarrow \neg P)$.

`Theorem` *contrapositive'* : $\forall$ (P Q : `Prop`),

$(P \to Q) \to (\tilde{}\, Q \to \neg P)$.

Exercise: 1 star (not_both_true_and_false) `Theorem` *not_both_true_and_false* : $\forall$ P : `Prop`,
$\neg$ $(P \wedge \neg P)$.

Exercise: 1 star (not_both_true_and_false) `Theorem` *not_both_true_and_false'* : $\forall$ P : `Prop`,
$\neg$ $(P \wedge \neg P)$.

`Theorem` *not_true_is_false* : $\forall$ b : *bool*,
$b \neq$ *true* $\to b =$ *false*.

`Theorem` *not_true_is_false'* : $\forall$ b : *bool*,
$b \neq$ *true* $\to b =$ *false*.

`Lemma` *True_is_true* : *True*.

`Notation` "P <-> Q" := (*iff* P Q) (`at level` 95, `no associativity`) : *type_scope*.

`Theorem` *iff_sym* : $\forall$ P Q : `Prop`,
$(P \leftrightarrow Q) \to (Q \leftrightarrow P)$.

`Lemma` *not_true_iff_false* : $\forall$ b,
$b \neq$ *true* $\leftrightarrow b =$ *false*.

Exercise: 1 star, optional (iff_properties)

`Theorem` *iff_refl* : $\forall$ P : `Prop`,
$P \leftrightarrow P$.

`Theorem` *iff_trans* : $\forall$ P Q R : `Prop`,
$(P \leftrightarrow Q) \to (Q \leftrightarrow R) \to (P \leftrightarrow R)$.

Exercise: 3 stars (or_distributes_over_and) `Theorem` *or_distributes_over_and* : $\forall$ P Q R : `Prop`,
$P \vee (Q \wedge R) \leftrightarrow (P \vee Q) \wedge (P \vee R)$.

`Require Import` *Coq.Setoids.Setoid*.

`Lemma` *mult_0* : $\forall$ n m, $n \times m = 0 \leftrightarrow n = 0 \vee m = 0$.

`Lemma` *or_assoc* :
$\forall$ P Q R : `Prop`, $P \vee (Q \vee R) \leftrightarrow (P \vee Q) \vee R$.

`Lemma` *mult_0_3* :
$\forall$ n m p, $n \times m \times p = 0 \leftrightarrow n = 0 \vee m = 0 \vee p = 0$.

`Lemma` *apply_iff_example* :
$\forall$ n m : *nat*, $n \times m = 0 \to n = 0 \vee m = 0$.

Olhar esse erro Lemma four_is_even : exists n : nat, 4 = n + n. Proof. exists 2. reflexivity. Qed.

Theorem exists_example_2 : forall n, (exists m, n = 4 + m) -> (exists o, n = 2 + o). Proof. intros n *m Hm*. exists (2 + m). apply Hm. Qed.

Theorem dist_not_exists : forall (X:Type) (P : X -> Prop), (forall x, P x) -> ˜ (exists x, ˜ P x). Proof. intros X P H. unfold not. intros x E. destruct E. apply H. Qed.

Theorem dist_exists_or : forall (X:Type) (P Q : X -> Prop), (exists x, P x \/ Q x) <-> (exists x, P x) \/ (exists x, Q x). Proof. intros X P Q. split.

- intros x $[PE \mid QE]$. + left. exists x. apply PE. + right. exists x. apply QE.
- intros $[x\ HPx] \mid [x\ HQx]$. + exists x. left. apply HPx. + exists x. right. apply HQx.

Qed.