Induction (Induction.v)

Keywords

- `assert (H: xxx). { ... }` subproof + local proof (using local variables)

Tactics

- `induction n as [| n' IHn']` Prove by induction on `n`
- `replace (t) with (u).` Replace all copies of expession `t` in the goal by expression `u`, and generate `t = u` as an additional subgoal.

List of Types, Definitions, Notations, (Useful) Examples

- `Theorem plus_n_O : forall n : nat, n = n + 0.`
- `Theorem minus_diag : forall n : nat, minus n n = 0.`
- `Theorem mult_0_r : forall n : nat, n * 0 = 0.`
- `Theorem plus_n_Sm : forall n m : nat, S(n + m) = n + (S m).`
- `Theorem plus_comm : forall n m : nat, n + m = m + n.`
- `Theorem plus_assoc : forall n m p : nat, n + (m + p) = (n + m) + p.`
-
- `Fixpoint double (n : nat)`
- `Lemma double_plus : forall n : nat, double n = n + n.`
-
- `Theorem evenb_S : forall n : nat, evenb (S n') = negb (evenb n).`
- `Theorem plus_rearrange : forall n m p q : nat, (n + m) + (p + q) = (m + n) + (p + q).`
- `Theorem plus_swap : forall n m p : nat, n + (m + p) = m + (n + p).`
- `Theorem mult_n_Sm : forall n m : nat, n * (S m) = n + n * m.`
- `Theorem mult_comm: forall m n : nat, m * n = n * m.`
-

- `Theorem leb_refl : forall n : nat, true = (n <=? n).`
- `Theorem zero_nbeq_S : forall n : nat, 0 =? (S n) = false.`
- `Theorem addb_false_r : forall b : bool, andb b false = false.`
- `Theorem plus_ble_compat_l : forall n m p : nat, n <=? m = true -> (p + n) <=? (p + m) = true.`
- `Theorem S_nbeq_0 : forall n : nat, (S n) =? 0 = false.`
-
- `Theorem mult_1_r : forall n : nat, n * 1 = n.`
- `Theorem mult_1_l: forall n : nat, 1 * n = n.`
-
- `Theorem all3_spec : forall b c : bool,`
-
- `Theorem mult_plus_distr_r : forall n m p : nat, (n + m) * p = (n * p) + (m * p).`
- `Theorem mult_assoc : forall n m p : nat, n * (m * p) = (n * m) * p.`
-
- `Theorem eqb_refl : forall n : nat, true = (n =? n).`
-
- `Theorem bin_to_nat_pres_incr : forall b : bin, bin_to_nat (incr b) = S (bin_to_nat b).`
- `Fixpoint nat_to_bin (n : nat) : bin`
- `Theorem nat_bin_nat : forall n : nat, bin_to_nat (nat_to_bin n) = n.`

Problems

- `Theorem mult_plus_distr_r : forall n m p : nat, (n + m) * p = (n * p) + (m * p).` Easy Proofs?
- How to use `QuickChick` ?
- Why does `Theorem bin_nat_bin` fail?
- `normalize`