Tactics

Expression	Meaning	Using	Proving
`forall`	universal quant.	`apply`, `specialize`	`intros`
`A -> B`	implication	`apply`, `specialize`	`intros`
`A /\ B`	conjunction	`destruct`	`split`
`A \/ B`	disjunction	`destruct`	`left`, `right`
`exists`	existential quant.	`destruct`	`exists`, `eexists`
`~ P`	negation	`apply`,`exfalso`	`intros`
`x = y`	equality	`rewrite`, `subst`, `inversion`	`reflexivity`, `f_equal`, `symmetry`, `transitivity`...
`x + y`	defined function		`simpl`
`n : nat`	inductive value	`destruct`	`constructor`
`m <= n`	inductive predicate	`inversion`, `induction`	`apply`, `constructor`

General tactics:

Command	Usage
`simpl`	fire computation
`assumption`	find the goal in the hypotheses
`easy`	very basic automation; fails if does not prove the goal
`now tac`	`tac` followed by easy; fails if does not prove the goal
`auto`	extensible proof search

Commands

Command	Usage
`Search (_ + _)`	Find theorems involving certain patterns
`Locate "+"`	Find out what a notation stands for
`About add`	Gives various information about a defined constant (type, implicit arguments...)
`Print add`	Gives the body of a definition
`Check (0 + 0)`	Type-checks a term
`Compute (0 + 0)`	Evaluates a term
`Print Assumptions NNPP`	Check what axioms a proof relies on