Coq Tactics

Manipulating the Context

Tactic	Arguments	Effect
intros	n_1 ... n_n	Introduce terms into the context, giving them names n_1 through n_n
revert	n_1 ... n_n	The opposite of intros, removes terms from context and quantifies over them in the goal.
clear	n_1 ... n_n	Deletes the terms n_1 through n_n from the context.
unfold	*name*	Replaces the function *name* with its underlying definition.

Making Progress

simpl		Simplifies the goal
destruct	*x*	Considers all possible constructors of the term *x*, generating subgoals that need to be solved separately.
	x eqn:*eq*	Destructs x, remembers which constructor is being considered in the equality hypothesis *eq*.
induction	*x* as [| n_1 IH_1 | n_2 IH_2 | ...]	Same as destruct but adds an inductive hypothesis to inductively defined cases. Names the variables and inductive hypotheses generated by induction with the given names.
rewrite	*H*	Where *H* is of type $e_1 = e_2$, replaces e_1 in the goal with e_2.
	<- *H*	Replaces e_2 with e_1
apply	*H*	Uses lemma or hypothesis *H* to solve the goal. If *H* has hypotheses, adds them as new goals.
	PQ in *HP*	Allows us to conclude *Q* from hypotheses *P* -> *Q* and *P*. (In logic, *modus ponens*.)
inversion	*H*	If hypothesis *H* states that *e1* = *e2*, where *e1* and *e2* are expressions that start with different constructors, then inversion *H* completes the current subgoal. If they start with the same constructor, it generates hypotheses relating the subterms.
assert	*H* : *P*	Adds a new hypothesis *H* that *P* is true to the goal. Adds the new subgoal *P* as the current goal.

Solving the Goal

Tactic	Effect
reflexivity	Solves a goal of the form $x = x$.
assumption	If we have a hypothesis that is equal to the goal, solves the goal.
discriminate	If we have a contradictory hypothesis involving an equality, solves the goal. (A special case of the logical *ex falso quodlibet*.)
contradiction	If we have a contradictory hypothesis not involving an equality, solves the goal. (A special case of the logical *ex falso quodlibet*.)
trivial	Checks if the goal is trivially true or equivalent to a hypothesis, solves if so. Otherwise, does nothing (does not fail).
auto	Tries a collection of basic tactics to solve the goal. Otherwise, does nothing (does not fail).
congruence	A powerful automation technique that subsumes reflexivity, assumption, discriminate and contradiction.
omega	Solves arithmetic equations over natural numbers.
lia	A more powerful sibling of omega.

Note that these are often special cases and simplifications. Click through to the Coq documentation for a description of the tactics' general behavior and underlying theory.

Also see Adam Chlipala's Coq Tactics Quick Reference for additional tactics and automation.