# Coq Tactic Quick Reference

## Context Manipulation
- **intro/revert**: shift goal premises to/from context
- **rename**: rename a hypothesis in the context
- **clear**: drop a hypothesis from the context
- **assert**: add a hypothesis to the context (proving it first)

## Theorems and Assumptions
- **assumption**: goal is identical to a hypothesis
- **apply**: use theorem \( A \rightarrow B \) to reduce goal \( B \) to subgoal \( A \), or convert hypothesis \( A \) to hypothesis \( B \)

## Simplification
- **simpl**: evaluate expressions until no more progress is possible
- **unfold**: expand an identifier into its definition
- **fold**: contract a definition back to its identifier

## Equalities
- **reflexivity**: prove equality of two identical expressions
- **symmetry**: change \( e_1 = e_2 \) to \( e_2 = e_1 \)
- **transitivity**: reduce goal \( e_1 = e_2 \) to two subgoals \( e_1 = e \) and \( e = e_2 \)
- **rewrite**: use hypothesis \( e_1 = e_2 \) to replace \( e_1 \) with \( e_2 \) or vice versa
- **subst**: use and clear hypothesis \( v = e \) by replacing all \( v \)'s with \( e \)'s
- **inversion**: from equality of structures, infer equality of substructures
- **remember**: introduce a new variable that names a subexpression

## Logical Operators
- **split**: prove \( A \land B \) by proving \( A \) and \( B \)
- **left/right**: prove \( A \lor B \) by proving \( A \) (left) or \( B \) (right)
- **exists**: prove an existential by supplying a witness
- **destruct**: decompose an and/or/exists hypothesis or pair variable
- **specialize**: instantiate a forall hypothesis

## Case Distinction and Induction
- **destruct**: introduce separate cases for each possible constructor
- **induction**: same as **destruct**, but generate an inductive hypothesis
- **inversion**: perform case distinction on an inductive proposition

## Negation and Contradiction
- **discriminate**: drop a goal by identifying a contradictory hypothesis
- **exfalso**: drop a goal by proving False