

▼ CMSC631: Program Analysis and Understanding

- 1 Syllabus
- 2 Resources
- 3 Schedule
- 4 Piazza
- 5 Project
- 6 Coq Cheatsheet

6 Coq Cheatsheet

Tactic	Explanation
intros <i>m n</i>	Introduces variables into the context
generalize dependent <i>n</i>	Quantifies over <i>n</i> and anything that depends on it
simpl [in <i>H</i>]	Simplifies the goal, where possible "in <i>H</i> " simplifies the hypothesis <i>H</i> "in *" simplifies the goal and all hypotheses
apply <i>H</i>	Matches the hypothesis/lemma <i>H</i> with the goal
rewrite -> <i>H</i> [in <i>H'</i>]	Given a hypothesis <i>H</i> of the form "x=y", replaces x with y. "<-" reverses the order
destruct <i>x</i> as [<i>m</i> <i>n</i>] eqn: <i>E</i>	Case analysis on <i>x</i> . "as.. " names the variables that appear "Eqn: <i>E</i> " remembers the given case as <i>E</i>
induction <i>x</i> as [<i>m</i> <i>n</i> <i>IH</i>] eqn: <i>E</i>	Like destruct but adds an inductive hypothesis for the inductive cases.
inversion <i>H</i> <i>x</i> as [<i>m</i> <i>n</i> <i>IH</i>]	Like destruct but doesn't throw out information, solves cases that fail to match.
injection (<i>H</i>) [as (<i>H'</i>)] remember <i>x</i> as <i>y</i> eqn: <i>E</i>	Removes constructors <i>K</i> from the hypothesis " <i>H</i> : <i>K</i> <i>m</i> = <i>K</i> <i>n</i> " Adds a new variable <i>y</i> into the context, remembering (<i>E</i> : <i>x</i> = <i>y</i>).
assert (<i>H</i> : <i>P</i>)	Adds <i>P</i> as a new subgoal. Once <i>P</i> is proven, adds it as the hypothesis <i>H</i> .
symmetry [in (<i>H</i>)]	Replaces " <i>x</i> = <i>y</i> " with " <i>y</i> = <i>x</i> "
discriminate (<i>H</i>)	If <i>H</i> has the form " <i>J</i> <i>m</i> = <i>K</i> <i>n</i> ", where <i>J</i> and <i>K</i> are different constructors, solves the goal
contradict (<i>H</i>)	Replaces the goal with the negation of <i>H</i> (most useful when (<i>H</i> : ~ <i>P</i>))

Tactic	shorthand for	apply to
reflexivity	apply eq_refl	<i>x</i> = <i>y</i>
split	apply conj	<i>P</i> ∧ <i>Q</i>
left/right	apply or_introl/orintror	<i>P</i> ∨ <i>Q</i>
exists <i>x</i>	apply (ex_intro _ <i>x</i>)	exists <i>y</i> , <i>P</i>