Basic tactics

- intro, intros – introduction rule for Π (several times).
- apply – elimination rule for Π.
- assumption – match conclusion with an hypothesis.
- exact – gives directly the exact proof term of the goal.
- contradiction – attempts to find in the current context a contradiction.

Tactics for first-order reasoning

Proposition (P)	Introduction	Elimination (H of type P)
$\bot$		elim H, contradiction
$\neg A$	intro	apply H
$A \wedge B$	split	elim H, destruct H as [H1 H2]
$A \Rightarrow B$	intro	apply H
$A \vee B$	left, right	elim H, destruct H as [H1|H2]
$\forall x\!:\!A.\,Q$	intro	apply H
$\exists x\!:\!A.\,Q$	exists *witness*	elim H, destruct H as [x H1]

Tactics for equational reasoning

- rewrite – rewrites a goal using an equality.
- rewrite <- – rewrites a goal using an equality in the reverse direction.
- reflexivity – reflexivity property for equality.
- symmetry – symmetry property for equality.
- transitivity – transitivity property for equality.
- replace a with b – replaces a by b while generating the subgoal a=b.
- f_equal – appliable to goals of the form $f\,a_1\,\ldots\,a_n = f'\,a'_1\,\ldots\,a'_n$.
- ...

Convertibility tactics

- simpl, red, cbv, lazy, compute – performs evaluation.
- unfold – applies the δ rule for a transparent constant.
- pattern – performs a beta-expansion on the goal.
- change – replaces the goal by a convertible one.
- ...

Tactics for inductive reasoning

- `elim` – to apply the corresponding induction principle.
- `induction` – performs induction on an identifier.
- `case`, `destruct` – performs case analysis.
- `constructor` – applies to a goal such that the head of its conclusion is an inductive constant.
- `discriminate` – discriminates objects built from different constructors.
- `injection` – applies the fact that constructors of inductive types are injections.
- `inversion` – given an inductive type instance, find all the necessary condition that must hold on the arguments of its constructors.
- ...

Other useful tactics and commands

- `clear` – removes an hypothesis from the environment.
- `generalize` – reintroduces an hypothesis into the goal.
- `cut`, `assert` – proves the goal through an intermediate result.
- `absurd` – applies `False` elimination.
- `contradict` – allows to manipulate negated hypothesis and goals.
- `refine` – allows to give an exact proof but still with some holes (“_”).
- ...

- `Admitted` – aborts the current proof and replaces the statement by an axiom that can be used in later proofs.
- `Abort` – aborts the current proof without saving anything.

Combining tactics

The basic tactics can be combined into more powerful tactics using tactics combinators, also called *tacticals*.

- `t1 ; t2` – applies tactic `t1` to the current goal and then `t2` to each generated subgoal.
- `t1 || t2` – applies tactic `t1`; if it fails then applies `t2`.
- `t ; [ t1 | ...| tn]` – applies `t` and then `ti` to the i-th generated subgoals; there must be exactly n subgoals generated by `t`.
- `idtac` – does nothing.
- `try t` – applies `t` if it does not fail; otherwise does nothing.
- `repeat t` – repeats `t` as long as it does not fail.
- `solve t` – applies `t` only if it solves the current goal.
- ...

Automatic tactics

- `trivial` – tries those tactics that can solve the goal in one step.
- `auto` – tries a combination of tactics `intro`, `apply` and `assumption` using the theorems stored in a database as hints for this tactic.
- `eauto` – like `auto` but more powerful but also more time-consuming.
- `tauto` – useful to prove facts that are tautologies in intuitionistic PL.
- `intuition` – useful to prove facts that are tautologies in intuitionistic PL.
- `firstorder` – useful to prove facts that are tautologies in intuitionistic FOL.
- `lia` – a tactic for linear integer arithmetic
- `nia` – a tactic for non-linear integer arithmetic
- `lra` – tactic for linear (real or rational) arithmetic.
- `ring` – does proves of equality for expressions containing addition and multiplication.
- `field` – like `ring` but for a field structure (it also considers division).
- `subst` – replaces all the occurrences of a variable defined in the hypotheses.