Waterproof Tactics Sheet

Help.

Tries to give you a hint on what to do next.

```
Lemma example_help :
  0 = 0.
Proof.
Help.
We conclude that (0 = 0).
Qed.
```

Take (*name*) : ((*type*)).

Take an arbitrary element from (*type*) and call it (*name*).

```
Lemma example_take :
  for all x : ℝ,
    x = x.
Proof.
Take x : (ℝ).
We conclude that (x = x).
Qed.
```

Take (*name*) ∈ ((*set*)).

Take an arbitrary element from (*set*) and call it (*name*).

```
Lemma example_take :
  ∀ x ∈ ℝ,
    x = x.
Proof.
Take x ∈ (ℝ).
We conclude that (x = x).
Qed.
```

Take (*name*) > ((*number*)).

Take an arbitrary element larger than (*number*) and call it (*name*).

```
Lemma example_take :
  ∀ x > 3,
    x = x.
Proof.
Take x > (3).
```

```
We conclude that (x = x).
Qed.
```

Take (*name*) ≥ ((*number*)).

Take an arbitrary element larger than or equal to (*number*) and call it (*name*).

```
Lemma example_take :
  ∀ x ≥ 5,
    x = x.
Proof.
Take x ≥ (5).
We conclude that (x = x).
Qed.
```

We need to show that ((*(alternative) formulation of current goal*)).

Generally makes a proof more readable. Has the additional functionality that you can write a slightly different but equivalent formulation of the goal: you can for instance change names of certain variables.

```
Lemma example_we_need_to_show_that :
  0 = 0.
Proof.
We need to show that (0 = 0).
We conclude that (0 = 0).
Qed.
```

We conclude that ((*current goal*)).

Tries to automatically prove the current goal.

```
Lemma example_we_conclude_that :
  0 = 0.
Proof.
We conclude that (0 = 0).
Qed.
```

We conclude that ((*(alternative) formulation of current goal*)).

Tries to automatically prove the current goal.

```
Lemma example_we_conclude_that :
  0 = 0.
Proof.
We conclude that (0 = 0).
Qed.
```

Choose (*name_var*) := ((*expression*)).

You can use this tactic when you need to show that there exists an x such that a certain property holds. You do this by proposing (*expression*) as a choice for x, giving it the name (*name_var*).

```
Lemma example_choose :
  ∃ y ∈ ℝ,
    y < 3.
Proof.
Choose y := (2).
* Indeed, (y ∈ ℝ).
* We conclude that (y < 3).
Qed.
```

Assume that ((*statement*)).

If you need to prove (*statement*) ⇒ B, this allows you to assume that (*statement*) holds.

```
Lemma example_assume :
  ∀ a ∈ ℝ, a < 0 ⇒ - a > 0.
Proof.
Take a ∈ (ℝ).
Assume that (a < 0).
We conclude that (- a > 0).
Qed.
```

Assume that ((*statement*)) ((*label*)).

If you need to prove (*statement*) ⇒ B, this allows you to assume that (*statement*) holds, giving it the label (*label*). You can leave out ((*label*)) if you don't wish to name your assumption.

```
Lemma example_assume :
  ∀ a ∈ ℝ, a < 0 ⇒ - a > 0.
Proof.
Take a ∈ (ℝ).
Assume that (a < 0) (a_less_than_0).
```

```
We conclude that (- a > 0).
Qed.
```

(& 3 < 5 = 2 + 3 ≤ 7) (chain of (in)equalities, with opening parenthesis)

Example of a chain of (in)equalities in which every inequality should.

```
Lemma example_inequalities :
  ∀ ε > 0, Rmin(ε,1) < 2.
Proof.
Take ε > 0.
We conclude that (& Rmin(ε,1) ≤ 1 < 2).
Qed.
```

& 3 < 5 = 2 + 3 ≤ 7 (chain of (in)equalities)

Example of a chain of (in)equalities in which every inequality should.

```
Lemma example_inequalities :
  ∀ ε > 0, Rmin(ε,1) < 2.
Proof.
Take ε : (ℝ).
Assume that (ε > 0).
We conclude that (& Rmin(ε,1) ≤ 1 < 2).
Qed.
```

Obtain such a (*name_var*)

In case a hypothesis that you just proved starts with ‘there exists’ s.t. some property holds, then you can use this tactic to select such a variable. The variable will be named (*name_var*).

```
Lemma example_obtain :
  ∀ x ∈ ℝ,
    (∃ y ∈ ℝ, 10 < y ∧ y < x) ⇒
      10 < x.
Proof.
Take x ∈ (ℝ).
Assume that (∃ y ∈ ℝ, 10 < y ∧ y < x) (i).
Obtain such a y.
Qed.
```

Obtain (*name_var*) according to ((*name_hyp*)).

In case the hypothesis with name (*name_hyp*) starts with 'there exists' s.t. some property holds, then you can use this tactic to select such a variable. The variable will be named (*name_var*).

```
Lemma example_obtain :
  ∀ x ∈ ℝ,
    (∃ y ∈ ℝ, 10 < y ∧ y < x) ⇒
      10 < x.
Proof.
Take x ∈ (ℝ).
Assume that (∃ y ∈ ℝ, 10 < y ∧ y < x) (i).
Obtain y according to (i).
Qed.
```

It suffices to show that ((*statement*)).

Waterproof tries to verify automatically whether it is indeed enough to show (*statement*) to prove the current goal. If so, (*statement*) becomes the new goal.

```
Lemma example_it_suffices_to_show_that :
  ∀ ε > 0,
      3 + Rmax(ε,2) ≥ 3.
Proof.
Take ε > 0.
It suffices to show that (Rmax(ε,2) ≥ 0).
We conclude that (& Rmax(ε,2) ≥ 2 ≥ 0).
Qed.
```

By ((*lemma or assumption*)) it suffices to show that ((*statement*)).

Waterproof tries to verify automatically whether it is indeed enough to show (*statement*) to prove the current goal, using (*lemma or assumption*). If so, (*statement*) becomes the new goal.

```
Lemma example_it_suffices_to_show_that :
  ∀ ε ∈ ℝ,
    ε > 0 ⇒
      3 + Rmax(ε,2) ≥ 3.
Proof.
Take ε ∈ (ℝ).
Assume that (ε > 0) (i).
By (i) it suffices to show that (Rmax(ε,2) ≥ 0).
```

```
We conclude that (& Rmax(ε,2) ≥ 2 ≥ 0).
Qed.
```

It holds that ((*statement*)) ((*label*)).

Tries to automatically prove (*statement*). If that works, (*statement*) is added as a hypothesis with name (*optional_label*).

```
Lemma example_it_holds_that :
  ∀ ε > 0,
    4 - Rmax(ε,1) ≤ 3.

Proof.
Take ε > 0.
It holds that (Rmax(ε,1) ≥ 1) (i).
We conclude that (4 - Rmax(ε,1) ≤ 3).
Qed.
```

It holds that ((*statement*)).

Tries to automatically prove (*statement*). If that works, (*statement*) is added as a hypothesis.

```
Lemma example_it_holds_that :
  ∀ ε > 0,
    4 - Rmax(ε,1) ≤ 3.

Proof.
Take ε > 0.
It holds that (Rmax(ε,1) ≥ 1).
We conclude that (4 - Rmax(ε,1) ≤ 3).
Qed.
```

By ((*lemma or assumption*)) it holds that ((*statement*)) ((*label*)).

Tries to prove (*statement*) using (*lemma*) or (*assumption*). If that works, (*statement*) is added as a hypothesis with name (*optional_label*). You can leave out ((*optional_label*)) if you don't wish to name the statement.

```
Lemma example_forwards :
  7 < f(-1) ⇒ 2 < f(6).
Proof.
Assume that (7 < f(-1)).
By (f_increasing) it holds that (f(-1) ≤ f(6)) (i).
```

```
We conclude that (2 < f(6)).
Qed.
```

By ((*lemma or assumption*)) it holds that ((*statement*)).

Tries to prove (*statement*) using (*lemma*) or (*assumption*). If that works, (*statement*) is added as a hypothesis with name (*optional_label*). You can leave out ((*optional_label*)) if you don't wish to name the statement.

```
Lemma example_forwards :
  7 < f(-1) ⇒ 2 < f(6).
Proof.
Assume that (7 < f(-1)).
By (f_increasing) it holds that (f(-1) ≤ f(6)) (i).
We conclude that (2 < f(6)).
Qed.
```

We claim that ((*statement*)).

Lets you first show (*statement*) before continuing with the rest of the proof. After you showed (*statement*), it will be available as a hypothesis with name (*optional_name*).

```
We claim that (2 = 2) (two_is_two).
```

We claim that ((*statement*)) ((*label*)).

Lets you first show (*statement*) before continuing with the rest of the proof. After you showed (*statement*), it will be available as a hypothesis with name (*label*).

```
We claim that (2 = 2) (two_is_two).
```

We argue by contradiction.

Assumes the opposite of what you need to show. Afterwards, you need to make substeps that show that both A and ¬ A (i.e. not A) for some statement A. Finally, you can finish your proof with 'Contradiction.'

```
Lemma example_contradicition :
  ∀ x ∈ ℝ,
    (∀ ε > 0, x > 1 - ε) ⇒
      x ≥ 1.
Proof.
Take x ∈ (ℝ).
```

```
Assume that (∀ ε > 0, x > 1 - ε) (i).
We need to show that (x ≥ 1).
We argue by contradiction.
Assume that (¬ (x ≥ 1)).
It holds that ((1 - x) > 0).
By (i) it holds that (x > 1 - (1 - x)).
Contradiction.
Qed.
```

Contradiction

If you have shown both A and not A for some statement A, you can write “Contradiction” to finish the proof of the current goal.

```
Contradiction.
```

Because ((*name_combined_hyp*)) both ((*statement_1*)) and ((*statement_2*)).

If you currently have a hypothesis with name (*name_combined_hyp*) which is in fact of the form H1 ∧ H2, then this tactic splits it up in two separate hypotheses.

```
Lemma and_example : for all A B : Prop, A ∧ B ⇒ A.
Take A : Prop. Take B : Prop.
Assume that (A ∧ B) (i). Because (i) both (A) (ii) and (B) (iii).
```

Because ((*name_combined_hyp*)) both ((*statement_1*)) ((*label_1*)) and ((*statement_2*)) ((*label_2*)).

If you currently have a hypothesis with name (*name_combined_hyp*) which is in fact of the form H1 ∧ H2, then this tactic splits it up in two separate hypotheses.

```
Lemma and_example : for all A B : Prop, A ∧ B ⇒ A.
Take A : Prop. Take B : Prop.
Assume that (A ∧ B) (i). Because (i) both (A) (ii) and (B) (iii).
```

Either ((*case_1*)) or ((*case_2*)).

Split in two cases (*case_1*) and (*case_2*).

```
Lemma example_cases :
  ∀ x ∈ ℝ, ∀ y ∈ ℝ,
    Rmax(x,y) = x ∨ Rmax(x,y) = y.
Proof.
```

```
Take x ∈ ℝ. Take y ∈ ℝ.
Either (x < y) or (x ≥ y).
- Case (x < y).
  It suffices to show that (Rmax(x,y) = y).
  We conclude that (Rmax(x,y) = y).
- Case (x ≥ y).
  It suffices to show that (Rmax(x,y) = x).
  We conclude that (Rmax(x,y) = x).
Qed.
```

Expand the definition of (*name_kw*).

Expands the definition of the keyword (*name_kw*) in relevant statements in the proof, and gives suggestions on how to use them.

```
Expand the definition of upper bound.
```

Expand the definition of (*name_kw*) in ((*expression*)).

Expands the definition of the keyword (*name_kw*) in the statement (*expression*).

```
Expand the definition of upper bound in (4 is an upper bound for [0, 3)).
```

We show both statements.

Splits the goal in two separate goals, if it is of the form A ∧ B

```
Lemma example_both_statements:
  ∀ x ∈ ℝ, (x^2 ≥ 0) ∧ (| x | ≥ 0).
Proof.
Take x ∈ (ℝ).
We show both statements.
- We conclude that (x^2 ≥ 0).
- We conclude that (| x | ≥ 0).
Qed.
```

We show both directions.

Splits a goal of the form A ⇔ B, into the goals (A ⇒ B) and (B ⇒ A)

```
Lemma example_both_directions:
  ∀ x ∈ ℝ, ∀ y ∈ ℝ,
    x < y ⇔ y > x.
Proof.
Take x ∈ (ℝ). Take y ∈ (ℝ).
```

```
We show both directions.
- We need to show that (x < y ⇒ y > x).
  Assume that (x < y).
  We conclude that (y > x).
- We need to show that (y > x ⇒ x < y).
  Assume that (y > x).
  We conclude that (x < y).
Qed.
```

We use induction on (*name_var*).

Prove a statement by induction on the variable with (*name_var*).

```
Lemma example_induction :
  ∀ n : ℕ → ℕ, (∀ k ∈ ℕ, n(k) < n(k + 1))%nat ⇒
    ∀ k ∈ ℕ, (k ≤ n(k))%nat.
Proof.
Take n : (ℕ → ℕ).
Assume that (∀ k ∈ ℕ, n(k) < n(k + 1))%nat (i).
We use induction on k.
- We first show the base case, namely (0 ≤ n(0))%nat.
  We conclude that (0 ≤ n(0))%nat.
- We now show the induction step.
  Take k ∈ ℕ.
  Assume that (k ≤ n(k))%nat.
  By (i) it holds that (n(k) < n(k + 1))%nat.
  We conclude that (& k + 1 ≤ n(k) + 1 ≤ n(k + 1))%nat.
Qed.
```

By ((*lemma or assumption*)) we conclude that ((*(alternative) formulation of current goal*)).

Tries to directly prove the goal using (*lemma or assumption*) when the goal corresponds to (*statement*).

Define (*name*) := ((*expression*)).

Temporarily give the name (*name*) to the expression (*expression*)

Since ((*extra_statement*)) it holds that ((*statement*)).

Tries to first verify (*extra_statement*) after it uses that to verify (*statement*). The statement gets added as a hypothesis.

```
Since (x = y) it holds that (x = z).
```

Since ((*extra_statement*)) it holds that ((*statement*)) ((*label*)).

Tries to first verify (*extra_statement*) after it uses that to verify (*statement*). The statement gets added as a hypothesiwe need to show{s, optionally with the name (*optional_label*).

```
Since (x = y) it holds that (x = z).
```

Since ((*extra_statement*)) we conclude that ((*(alternative) formulation of current goal*)).

Tries to automatically prove the current goal, after first trying to prove (*extra_statement*).

```
Since (x = y) we conclude that (x = z).
```

Since ((*extra_statement*)) it suffices to show that ((*statement*)).

Waterproof tries to verify automatically whether it is indeed enough to show (*statement*) to prove the current goal, after first trying to prove (*extra_statement*). If so, (*statement*) becomes the new goal.

```
Lemma example_backwards :
  3 < f(0) ⇒ 2 < f(5).
Proof.
Assume that (3 < f(0)).
It suffices to show that (f(0) ≤ f(5)).
By (f_increasing) we conclude that (f(0) ≤ f(5)).
Qed.
```

Use (*name*) := ((*expression*)) in ((*label*)).

Use a forall statement, i.e. apply it to a particular expression.

```
Lemma example_use_for_all :
  ∀ x ∈ ℝ,
    (∀ ε > 0, x < ε) ⇒
       x + 1/2 < 1.
Proof.
Take x ∈ ℝ.
Assume that (∀ ε > 0, x < ε) (i).
Use ε := (1/2) in (i).
* Indeed, (1 / 2 > 0).
* It holds that  (x < 1 / 2).
```

```
  We conclude that (x + 1/2 < 1).
Qed.
```

Indeed, ((*statement*)).

A synonym for “We conclude that ((*statement*))”.

```
Lemma example_choose :
  ∃ y ∈ ℝ,
    y < 3.
Proof.
Choose y := (2).
* Indeed, (y ∈ ℝ).
* We conclude that (y < 3).
Qed.
```

We need to verify that ((*statement*)).

Used to indicate what to check after using the “Choose” or “Use” tactic.

```
Lemma example_choose :
  ∃ y ∈ ℝ,
    y < 3.
Proof.
Choose y := (2).
* We need to verify that (y ∈ ℝ).
We conclude that (y ∈ ℝ).
* We conclude that (y < 3).
Qed.
```

By magic it holds that ((*statement*)) ((*label*)).

Postpones the proof of (*statement*), and (*statement*) is added as a hypothesis with name (*optional_label*). You can leave out ((*optional_label*)) if you don’t wish to name the statement.

```
Lemma example_forwards :
  7 < f(-1) ⇒ 2 < f(6).
Proof.
Assume that (7 < f(-1)).
By magic it holds that (f(-1) ≤ f(6)) (i).
We conclude that (2 < f(6)).
Qed.
```

By magic it holds that ((*statement*)).

Postpones the proof of (*statement*), and (*statement*) is added as a hypothesis.

```
Lemma example_forwards :
  7 < f(-1) ⇒ 2 < f(6).
Proof.
Assume that (7 < f(-1)).
By magic it holds that (f(-1) ≤ f(6)) (i).
We conclude that (2 < f(6)).
Qed.
```

By magic we conclude that ((*(alternative) formulation of current goal*)).

Postpones for now the proof of (a possible alternative formulation of) the current goal.

By magic it suffices to show that ((*statement*)).

Postpones for now the proof that (*statement*) is enough to prove the current goal. Now, (*statement*) becomes the new goal.

```
Lemma example_backwards :
  3 < f(0) ⇒ 2 < f(5).
Proof.
Assume that (3 < f(0)).
By magic it suffices to show that (f(0) ≤ f(5)).
By (f_increasing) we conclude that (f(0) ≤ f(5)).
Qed.
```

Case ((*statement*)).

Used to indicate the case after an “Either” sentence.

```
Lemma example_cases :
  ∀ x ∈ ℝ, ∀ y ∈ ℝ,
    Rmax(x,y) = x ∨ Rmax(x,y) = y.
Proof.
Take x ∈ ℝ. Take y ∈ ℝ.
Either (x < y) or (x ≥ y).
- Case (x < y).
  It suffices to show that (Rmax(x,y) = y).
  We conclude that (Rmax(x,y) = y).
- Case (x ≥ y).
```

```
  It suffices to show that (Rmax(x,y) = x).
  We conclude that (Rmax(x,y) = x).
Qed.
```