Basic notations

<table>
<thead>
<tr>
<th>Logic</th>
<th>Prop</th>
<th>bool</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊤, ⊥</td>
<td>True, False</td>
<td>true, false</td>
</tr>
<tr>
<td>¬p</td>
<td>¬p</td>
<td>¬p</td>
</tr>
<tr>
<td>p ∨ q</td>
<td>p ∨ q</td>
<td>p</td>
</tr>
<tr>
<td>a = b</td>
<td>a = b</td>
<td>a = b</td>
</tr>
<tr>
<td>a ≠ b</td>
<td>a &lt;&gt; b</td>
<td>a != b</td>
</tr>
<tr>
<td>∀x ∈ A . q(x, y)</td>
<td>forall (x:A) y, q x y</td>
<td></td>
</tr>
<tr>
<td>∃x ∈ A . p(x)</td>
<td>exists x:A, p x</td>
<td></td>
</tr>
</tbody>
</table>

Paper

| Lemma 1. For all natural numbers n and m if m ≤ n then the following equation holds: |
|--------|---------------------------------------------------------------|
| □   | □                                                             |
| □   | □                                                             |
| □   | □                                                             |
| □   | □                                                             |

Coq

| Lemma good_name : forall n m, m <= n -> n - m + m = n. |
|--------|---------------------------------------------------------------|
| □   | □                                                             |
| □   | □                                                             |
| □   | □                                                             |
| □   | □                                                             |

Search _ (_ + _) "mul" modn.

Search lemmas whose name contains the infix plus operation and whose statement mentions the constant modn. [caveat] always put an underscore after Search

About lem.

Print the statement of lemma lem and some informations associated with it, like the arguments that are marked as implicit

Print def.

Prints the body of the definition

Basic proof commands

Require Import ssreflect ssrbool.

Load libraries ssreflect and ssrbool

Section name.

Open a section

Variable name : type.

Declares a section variable

Hypothesis name : statement.

Declares a section assumption

End name.

Close a section

Lemma name : statement.

State a lemma

Proof.

Start the proof of a lemma

Qed.

Terminate the proof of a lemma

Cheat Sheet — lesson 2

Basic proof commands

done.

Prove the goal by trivial means, or fail

exact: H.

Apply H to the current goal and then assert all remaining goals, if any, are trivial. Equivalent to by apply: H.

apply: H.

Apply H to the current goal

split.

Prove a conjunction

left.

Prove a disjunction choosing the left part.

right chooses the right part

rewrite Eab.

Rewrite with Eab left to right
rewrite -Eab.
  Rewrite with Eab right to left

  \[ Eab : a = b \quad Eab : a = b \]
  \[ \equiv \quad \equiv \]
  \[ P b \quad P a \]

have pa : P a.
  Open a new goal for P a. Once resolved introduce a
  new entry in the context for it named pa

  \[ a : T \quad a : T \quad a : T \]
  \[ \equiv \quad \equiv \quad \equiv \]
  \[ G \quad P a \quad P a \]

suffices pa : P a.
  Open a new goal with an extra item in the context for
  P a named pa. When resolved, it asks to prove P a

  \[ a : T \quad a : T \quad a : T \]
  \[ \equiv \quad \equiv \quad \equiv \]
  \[ G \quad G \quad P a \]

Proof General
  Ctrl-C Ctrl-Enter
  Goto the cursor
  Ctrl-X Ctrl-F
  Open a file
  Ctrl-X Ctrl-S
  Save
  Ctrl-X Ctrl-C
  Quit

Simple notations

"[ /\ P1 , P2 & P3 ]" := (and3 P1 P2 P3)
"[ \// P1 , P2 | P3 ]" := (or3 P1 P2 P3)
"[ \&\& b1 , b2 , .. , bn & c ]" :=
  (b1 \&\& (b2 \&\& .. (bn \&\& c) .. ))
"[ || b1 , b2 , .. , bn | c ]" :=
  (b1 || (b2 || .. (bn || c) .. ))
"b1 (+) b2" := (addb b1 b2)