Cheat Sheet — lesson 2

Basic notations

\mathbf{Logic}	Prop	bool
\top, \bot	True, False	true, false
$\neg p$	~ p	~~ p
$p \wedge q$	p /\ q	р && q
$p \lor q$	p \/ q	p II q
a = b	a = b	a == b
$a \neq b$	a <> b	a != b
$p \Rightarrow q$	p -> q	p ==> q
$\forall x \in A . \forall y . q(x,y)$	forall (x:A) y, q x y	
$\exists x \in A . p(x)$	exists x:A, p x	

Paper	\mathbf{Coq}
Lemma 1. For all natural numbers n and m if $m \le n$ then the following equation holds:	<pre>Lemma good_name : forall n m, m <= n -> n - m + m = n.</pre>
n-m+m=n	Proof. (* your proof *)
<i>Proof.</i> Trivial \Box	Qed.

Basic 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.

D 1

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

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

Search lemmas whose name contains the string muland whose statement mentions the infix plus operation and contains 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

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.

move=> x px.

Introduce x and P x naming them x and px

forall x,
$$P \times -> Q \times -> G$$

$$x : T$$

$$px : P \times$$

apply: H.

Apply H to the current goal

case: ab.

Eliminate the conjunction or disjunction

ab : A
$$\land$$
 B =======
G A -> B -> G

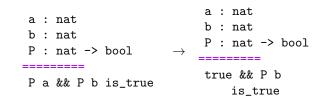
$$\begin{array}{c} \text{ab} : \text{A} \setminus / \text{ B} \\ ======= \\ \text{G} \end{array} \rightarrow \begin{array}{c} ====== \\ \text{A} \rightarrow \text{G} \end{array} \qquad \begin{array}{c} ======= \\ \text{B} \rightarrow \text{G} \end{array}$$

case: exg3.

Eliminate the existential quantification

case: (P a).

Reason by cases on a boolean predicate



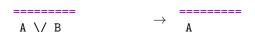
split.

Prove a conjunction



left.

Prove a disjunction choosing the left part. right chooses the right part



rewrite Eab.

Rewrite with Eab left to right

Eab:
$$a = b$$
 Eab: $a = b$ P a P b

rewrite -Eab.

Rewrite with Eab right to left

have pa : P a.

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

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

Proof General

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)
```