# Formalising Sylow's theorems in Coq

Laurent Théry, Laurence Rideau

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *Formalising Sylow's theorems in Coq*

Laurence Rideau — Laurent Théry

**N° 0327**

Novembre 2006

Thème SYM

Rapport technique

ISSN 0249-0803 ISRN INRIA/RT-0327--FR+ENG

# Formalising Sylow's theorems in Coq

Laurence Rideau , Laurent Théry

Thème SYM — Systèmes symboliques
Projet Marelle

Rapport technique n° 0327 — Novembre 2006 — 23 pages

**Abstract:** This report presents a formalisation of Sylow's theorems done in Coq. The formalisation has been done in a couple of weeks on top of Georges Gonthier's SSREFLECT [2]. There were two ideas behind formalising Sylow's theorems. The first one was to get familiar with Georges way of doing proofs. The second one was to contribute to the collective effort to formalise a large subset of group theory in Coq with some non-trivial proofs.

**Key-words:** Group theory, Sylow's theorems, Formalisation of mathematics

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

# Formalisation des théorèmes de Sylow dans Coq

**Résumé :** Ce rapport présente une formalisation des théorèmes de Sylow faite dans le système Coq. La formalisation s'est faite en deux semaines au dessus de la librairie SSREFLECT de Georges Gonthier. Il y avait deux principales motivations pour formaliser les théorèmes de Sylow. La première était de se familiariser avec la façon qu'a Georges de faire des preuves. La seconde était de contribuer à l'effort collectif de formaliser un large ensemble de la théorie des groupes en Coq.

**Mots-clés :** Théorie des groupes, Théorème de Sylow, Formalisation des mathématiques

# 1 Introduction

Sylow's theorems are central in group theory. Any course has a section or a chapter on them. Taking them as a first step in an effort to formalise group theory seemed a good idea. One of these theorems is number 72 in the list of the 100 theorems [4] maintained by Freek Wiedijk. Surprisingly, only one formalisation is known. It has been done in Isabelle by Florian Kammüller [3]. The proof that has been formalised in Isabelle is due to Wielandt [5]. It is a very concise and elegant proof. A central step in the proof is a non-trivial combinatorial argument that is used to show the existence of a group with a particular property. This is not the proof we have chosen to formalise. As we are interested in formalising Sylow's theorems not only as a mere exercise but as a base for further development, conciseness is nice but reusability is much more important. We have chosen to follow the proof given by Gregory Constantine [1] in his group theory course. It has the nice property of using one main tool, namely group actions, to prove most of the key results. The combinatorial argument that was present in the proof of Wielandt is then reduced to a minimum. Most of our formalising time has then been spent proving theorems about groups not about numbers.

The presentation of this work is organised as follows. In a first section, we describe what we started from. The main points we want to address are how SSREFLECT is organised and how using this dedicated version of COQ differs from using the standard one. In a second section, we outline the main steps of our proofs. Then, in a last section we conclude.

# 2 From types with decidable equality to finite types

## 2.1 Types with decidable equalitiy

One of the key decision of SSREFLECT is to base the development on objects not in `Type` but in `eqType`, i.e objects for which equality is decidable.

```
Structure eqType : Type := EqType {
  sort :> Set;
    eq :  sort -> sort -> bool;
   eqP :  forall x y, reflect (x = y) (eq x y)
}.
```

`eq` is the function that decides equality and `eqP` the theorem that insures that `(eq x y)`, written in the following as `x == y`, is true iff `x = y`. We call this the adequacy of equality.

Adding decidability on objects has the nice consequence to equate the type `bool`, the booleans, with the type `Prop`, the propositions. Of course, these two types are not identified since we are completely compatible with the standard way of doing proofs in COQ. Still, an inductive relation `reflect` of type `Prop -> bool -> Type` holds all the information to coerce one into the other.

In practice, booleans are always privileged with respect to propositions. For this, the coercion `is_true` from booleans to propositions is used.

```
Coercion is_true b := b = true.
```

As an example, let us consider equality and conjunction. Instead of stating a conjunction of two equalities as `x = y /\ z = t`, we prefer writing it using booleans as `x == y && z == t`. This simple modification gives a classical flavour to the usually intuitionistic prover Coq. Moreover, proof scripts become more similar to the ones of other systems like Hol. In particular, as booleans accommodate the substitutivity property, rewriting becomes the tactic number one. This reflection between `bool` and `Prop` is supported by the tactic language with the so-called views. As an example, consider the reflection over conjunction which is represented by the theorem `andP`

```
Theorem andP: forall b1 b2 : bool, reflect (b1 /\ b2) (b1 && b2).
```

Suppose now that we have to prove the following goal `x == y && z == t`. In order to split this goal into two subgoals, we use a combination of two tactics: `(apply/andP; split)`. The first tactic converts the `&&` into a `/\` , the second tactic can then perform the splitting. Similarly for an hypothesis, if the goal is `x == y && z == t -> A` for an arbitrary `A`, the tactic `(move/andP; case)` performs the convertion and the destructuring. Note that we can do even shorter combining view and case: `case/andP`.

Some standard operations are defined on `eqType`. For example, it is possible to build the set of pairs of objects. The construction is the following:

```
Structure eq_pair (d1 d2:eqType):  Type := EqPair {
  eq_pi1:  d1;
  eq_pi2:  d2
}.
```

```
Definition pair_eq (d1 d2:eqType) (u v:eq_pair d1 d2):bool:=
  let EqPair x1 x2:= u in
  let EqPair y1 y2:= v in (x1 == y1) && (x2 == y2).
```

Once the adequacy of the equality is proved, we can build the expected type with decidable equality. This is represented by the function `prod_eqType` with the following type `prod_eqType:  eqType -> eqType -> eqType`.

## 2.2 Sets

Sets are represented by their indicator function:

```
Definition set (d: eqType) := d -> bool.
```

For example, the constructor of a singleton is defined as

```
Definition set1 x := fun y => (y == x).
```

A key construction is the one that allows to build a type $\mathtt{d}_1$ with decidable equality from a set `A` whose carrier is a type `d` with decidable equality. This is done using the constructor `sub_eqType`:

```
sub_eqType:  forall d:  eqType, set d -> eqType.
```

$\mathtt{d}_1$ is then `(sub_eqType d A)` and elements of $\mathtt{d}_1$ are composed of elements of `d` and a proof that they belong to `A`.

```
Structure eq_sig (d: eqType) (A: set d): Set :=  EqSig {
   val: d;
  valP: A val
}.
```

Equality then only checks the first elements of the two records. As sets are represented as indicators, this equality is adequate (there is only one proof of `x` = `true`). Over sets, there is also the usual extensional equality, i.e. $\mathtt{A}_1 =_1 \mathtt{A}_2$ iff $\mathtt{A}_1$ `x ==` $\mathtt{A}_2$ `x` for all `x`.

## 2.3 Sequence

Sequences are represented in a standard way

```
Inductive seq (d: eqType): Type := Seq0 | Adds (x : d) (s : seq d).
```

Sequences are equipped with all the basic operations. In the following, we are going to use two of these operations: `size`, `count`. `size` gives the number of elements of a sequence. `count` returns the number of elements of a set inside a sequence.

## 2.4 Finite type

The last construction before defining groups is the one for creating finite types. A finite type is composed of a type `sort` with decidable equality, its sequence of elements and a proof that the sequence contains each element of `sort` once and only once.

```
Structure finType: Type := FinSet {
   sort :> eqType;
   enum :  seq sort;
  enumP :  forall x, count (set1 x) enum = 1
}.
```

Note that this encoding of finite sets gives for free an order on the elements of the finite set, i.e. the index of its occurrence in the sequence. The cardinality of a set `A` over a finite type `S` is defined as `(count A (enum S))`. It is written in the following as `(card A)`.

# 3 From finite groups to Sylow's theorems

## 3.1 Finite group, coset and subgroup

A finite group contains a finite set, an unit element, an inverse function and a multiplication with the usual properties.

```
Structure finGroup : Type := Finite {
  element:> finType;
     unit:  element;
      inv:  element -> element;
```

```
      mul:  element -> element -> element;
    unitP:  forall x, mul unit x = x;
     invP:  forall x, mul (inv x) x = unit;
     mulP:  forall x1 x2 x3, mul x1 (mul x2 x3) = mul (mul x1 x2) x3
}.
```

Given a multiplicative finite group `G` and `x`, `y` two elements of `G`, `1` is encoded as `(unit G)`, $\mathtt{x}^{-1}$ as `(inv G x)`, and `xy` as `(mul G x y)`. Given a finite group `G`, a set `H` of `G` and an element `a` of `G`, the left coset `aH` (the right coset `Ha`) is the set of the elements `ax` (respectively the set of elements `xa`) for all `x` in `H`. As we have `x` in `aH` iff $\mathtt{a}^{-1}\mathtt{x}$ is in `H` (respectively `x` in `Ha` iff $\mathtt{xa}^{-1}$ is in `H`), we have the following definitions:

```
Definition lcoset H a:  set G := fun x => H (a⁻¹x).
Definition rcoset H a:  set G := fun x => H (xa⁻¹).
```

The function $\mathtt{x} \mapsto \mathtt{ax}$ is a bijection between `H` and `aH`, so both sets have same cardinality. Furthermore, every coset `aH` can be represented by a canonical element $\overline{\mathtt{a}}$ such that $\mathtt{aH} =_1 \mathtt{bH}$ iff $\overline{\mathtt{a}}$ `==` $\overline{\mathtt{b}}$. Technically, $\overline{\mathtt{a}}$ is encoded as `(root (lcoset H) a)`, which is the first element in the sequence of the finite set that belongs to `aH`.

Subgroups are not defined as structures but as sets. Their definition is a bit intricate. The idea is to say that a set `H` is a subgroup if it is not empty, and if `x` and `y` are in `H` so is $\mathtt{xy}^{-1}$. This is sufficient. Since if `H` is non empty, it contains at least an element `z`, so we have $\mathtt{zz}^{-1}$ `== 1` belongs to `H`. Also, for all `x` in `H`, $\mathtt{1x}^{-1}$ `==` $\mathtt{x}^{-1}$ also belongs to `H`. Finally, if `x` and `y` belongs to `H`, we have $\mathtt{y}^{-1}$ belongs to `H`, so is $\mathtt{x}(\mathtt{y}^{-1})^{-1}$ `== xy`. In our definition, `1` is used as a witness of non-emptiness. For the second condition, we rewrite it as "if `x` is in `H` then `H` is included in `Hx`".

```
Definition subgrp H :=
  H 1 && subset H (fun x => subset H (rcoset H x)).
```

where (`subset` $\mathtt{H}_1$ $\mathtt{H}_2$) is true iff for all `x` in $\mathtt{H}_1$, `x` is also in $\mathtt{H}_2$. In this definition, `G` is given implicitly since the type of `H` is `(set G)`. This definition is of little use for proving that a set is a subgroup. As we are in a finite setting, a much more practical characterisation of a subgroup is that it is a non-empty set that is stable by multiplication. This is represented in our development by the theorem `finstbl_sbgrp`:

```
Lemma finstbl_sbgrp: forall G (H : set G) (a : G),
       H a -> (forall x y, H x -> H y -> H (xy)) -> subgrp H.
```

If `H` is a subgroup, its left cosets partition `G`: if `z` is in the intersection `aH` and `bH`, there exist $\mathtt{h}_1$ and $\mathtt{h}_2$ such that $\mathtt{ah}_1$ `== z == b` $\mathtt{h}_2$, we get `a ==` $\mathtt{b}(\mathtt{h}_2\mathtt{h}_1{}^{-1})$ and `b ==` $\mathtt{a}(\mathtt{h}_1\mathtt{h}_2{}^{-1})$, so $\mathtt{aH} =_1 \mathtt{bH}$. We denote `(lindex H)` the number of canonical elements. We then get that `card G = lindex H * card H`. As in our development groups and subgroups differ in nature, groups hold the carrier while subgroups are only indicators, it is preferable to state Lagrange's theorem at the level of subgroups:

```
Theorem lLaGrange:
 forall G (H K: set G),
 subgrp H -> subgrp K -> subset H K => card H * lindex H K = card K.
```

Now, `lindex H K` denotes the number of coset of `H` with respect to `K`. Note that we can always get back to the usual statement, using the fact that `G` is a subgroup of itself.

## 3.2 Conjugate, normaliser and normal subgroub

Normal subgroups are needed for the proof of Sylow's theorem. In order to define them, we first define the conjugate operation.

Definition $y^x$ := $x^{-1}yx$.

Then, given an arbitrary element `x` and an arbitrary set `H` the conjugate set $xHx^{-1}$ is defined as follows:

Definition conjsg H x := fun y => H $y^x$.

`y` is in $xHx^{-1}$ iff $x^{-1}yx$ is in `H`. We are now ready to define the notion of normal subgroup. `H` is normal in `K` iff for all element `x` in `K`, $xHx^{-1} =_1 H$. It is in fact sufficient to require that `H` is included in $xHx^{-1}$ as both sets have same cardinality. This gives the following definition:

```
Definition normal H K := subset K (fun x => subset H (conjsg H x)).
```

Later in the proof of the first Sylow's theorem we use the property that the quotient of a group by a normal subgroup is a group. This is a direct consequence of normality that imposes that the operation of the group behaves well with respect to cosets. The quotient group is represented in our development by the group `RG` composed with the roots of `G` with respect to the left coset relation.

Given a subgroup `H`, it is possible to build its normaliser, the set of all `x` in `K` such that $xHx^{-1}$ = `H` as:

```
Definition normaliser H K x :=
  (subset K (fun z => (conjsg H x z == H z))) && K x.
```

By definition, we have that `H` is normal in `(normaliser H K)`. This is the theorem `normaliser_normal`:

```
Lemma normaliser_normal:
  forall G (H K : set G), subset H K -> normal H (normaliser H K).
```

## 3.3 Group actions

Group actions are the key construction for our final theorems. To define an action, we need a group `G`, a subgroup `H` and a finite set `S`. This is written in our development as:

```
Variable G : finGroup.
Variable H : set G.
Hypothesis sgrp_H: subgrp H.
Variable S : finType.
```

An action `to` is a homomorphism from `H` to the permutations of `S` (the bijections from `S` to `S`). This is defined as:

```
Variable to: G -> (S -> S).
Hypothesis to_bij: forall x, H x -> bijective (to x).
Hypothesis to_morph: forall (x y: G) z,
  H x -> H y -> to (xy) z = to x (to y z).
```

where the predicate `bijective` indicates that the function is a bijection. Note that we have arbitrary chosen to define our action `to` on `G` and only require the properties of homomorphism and permutation to hold for elements of `H`.

For an element `a` of `S`, we define its orbit as all the elements of `S` that can be reached from `a` by the function `to`. In other words, it is the image of `H` by the function that given an `x` in `G` associates `(to x a)`.

```
Definition orbit a := image (fun x => to x a) H.
```

We can partition `S` using the orbits. A key property of group action comes with the notion of stabiliser. Given an element `a` of `S`, we call its stabiliser the set of all the elements `x` of `H` that leave `a` unchanged by the function `to x`. Formally, this gives

```
Definition stabiliser a := fun x => (((to x a) == a)) && (H x)).
```

The stabiliser is clearly a subgroup of `H` but the key property is that the cardinal of the orbit of `a` and the index of the stabiliser of `a` are equal.

```
Lemma card_orbit: forall a, card (orbit a) = lindex (stabiliser a) H.
```

to see this we just have to notice that we have `(to x a)` $=_d$ `(to y a)` iff $\mathtt{x}^{-1}\mathtt{y}$ is in `(stabiliser a)`. For this, we write `(to y a)` as `(to x (to (`$\mathtt{x}^{-1}\mathtt{y}$`) a)))` and use the fact that `to` is injective.

In the particular case where `H` has cardinality $p^\alpha$ with $p$ prime, as orbits partition `S` and their cardinality is an index, Lagrange's theorem gives us that these orbits are of cardinality $p^\beta$ with $\beta \leq \alpha$. Now, if we collect in the set $\mathtt{S}_0$ all the elements of `S` whose orbit has cardinality $1 = p^0$, i.e elements that are in the stabiliser of every element of `H`:

```
Definition S0 a := subset H (stabiliser a).
```

we get our central lemma

```
Lemma mpl:  (card S) % p = (card S0) % p.
```

where `%` is the usual modulo operation. All the orbits of cardinality $p^\beta$ with $0 < \beta \leq \alpha$ cancel out in the modulo.

## 3.4 Cauchy's theorem

The proof of the first Sylow theorem is an inductive proof. Cauchy's theorem solves the base case. This theorem states that if a prime $p$ divides the cardinality of a group, then there exists a subgroup of cardinality $p$. More precisely, there exists an element `a`, such that its cyclic group, i.e. the set of all the $\mathtt{a}^i$, is of cardinality $p$. As we did for Lagrange's, we state this theorem at the level of subgroups. We take `H` a subgroup of `G` and a prime $p$ that divides the cardinality of `H`. We first consider $\mathtt{H}^{p-1}$ the cartesian product $\underbrace{\mathtt{H} \times \ldots \times \mathtt{H}}_{p-1}$. An element `x` of $\mathtt{H}^{p-1}$ is written as $(\mathtt{h}_0,\ \ldots,\ \mathtt{h}_{p-2})$. We have $(\mathtt{card}\ \mathtt{H}^{p-1})\ =\ (\mathtt{card}\ \mathtt{H})^{p-1}$. We define $H^*$ a subset of $\mathtt{H}^p$ as the image of $\mathtt{H}^{p-1}$ by the function

$(\mathtt{h}_0,\ \ldots,\ \mathtt{h}_{p-2}) \mapsto ((\prod_{i=0}^{p-2} \mathtt{h}_i)^{-1},\ \mathtt{h}_0,\ \ldots,\ \mathtt{h}_{p-2})$.

Clearly, we have $(\mathtt{card}\ \mathtt{H}^*)\ =\ (\mathtt{card}\ \mathtt{H})^{p-1}$ and every element $(\mathtt{h}_0,\ \ldots,\ \mathtt{h}_{p-1})$ of $\mathtt{H}^p$ such that $\prod_{i=0}^{p-1} \mathrm{h}_i = 1$ is in $\mathtt{H}^*$. Now we consider the additive group $\mathbb{Z}_p$ and the action `to` from $\mathbb{Z}_p$ to $\mathtt{H}^*$ defined as

$\mathtt{n} \mapsto \{\ (\mathtt{h}_0,\ \mathtt{h}_1 \ldots,\ \mathtt{h}_{p-1}) \mapsto (\mathtt{h}_{(0+n)\%p},\ \mathtt{h}_{(1+n)\%p}, \ldots,\ \mathtt{h}_{(p-1+n)\%p})\}$

Now, if we look at the set $\mathtt{S}_0$ of the elements of orbit with cardinality 1. We can easily prove that $\mathtt{S}_0$ is composed of the elements `(h, ..., h)` such that $\mathtt{h}^p\ =\ 1$. In one direction, such elements clearly belong to $\mathtt{S}_0$ since they are left unchanged by any permutation of indexes. Conversely, if an element `x` belongs to $\mathtt{S}_0$, in particular `(to 1 x)` is equal to `x`. So, if we write `x` as $(\mathtt{h}_0,\ \ldots,\ \mathtt{h}_{p-1})$, this means $(\mathtt{h}_0,\ \ldots,\ \mathtt{h}_{p-1})$ is equal to $(\mathtt{h}_1,\ \ldots,\ \mathtt{h}_0)$ which in turn implies that $\mathtt{h}_0$ is equal to $\mathtt{h}_1$, $\mathtt{h}_1$ is equal to $\mathtt{h}_2$ and so on. Now, the `mpl` lemma tells us that $(\mathtt{card}\ \mathtt{H}^*)\ \%\ \mathtt{p}\ =\ (\mathtt{card}\ \mathtt{S}_0)\ \%\ \mathtt{p}$, but the cardinality of $\mathtt{H}^*$ is divisible by $p$ so we can conclude that the cardinality of $\mathtt{S}_0$ is also divisible by $p$. As, $p \geq 2$, this means that there exists at least one element `a` different from `1` in $\mathtt{S}_0$. For this element, we have $\mathtt{a}^p = 1$. We have that the cardinality of the cyclic group of `a` divides $p$ but as $p$ is prime and `a` is different of 1, the cardinality of its cyclic group is then exactly $p$. The exact statement of Cauchy's theorem in our development is

```
Theorem cauchy:  forall G, (H : set G) p,
    subgrp H -> prime p -> p | (card h) ->
    exists a, H a && (card (cyclic a) == p).
```

where `|` denotes the divisibility and `cyclic` builds the cyclic group of an element.

### 3.5 Sylow's theorems

The first Sylow theorem tells us that if `G` is a group and `K` is a subgroup of `G` of cardinality $p^n s$ with $p$ prime and $p$, $s$ relatively prime, then there exists a subgroup of `K` of cardinality $p^n$. Such a subgroup of maximal cardinality in $p$ is called a Sylow $p$ subgroup. It is defined in our development as

```
Definition sylow K p H:=
  subgrpb H && subset H K && card H == expn p (dlogn p (card K)).
```

where `expn` is the exponential function and `dlogn` is the divisor logarithm, i.e `(dlogn p u)` is the maximal power of `p` that divides `u`.

The proof of the first Sylow theorem is done by induction. We are going to prove that for all $i$, $0 < i \leq n$, there exists a subgroup of cardinality $p^i$. For $i = 1$, the existence is given by Cauchy's theorem. Now, suppose that there exists a subgroup `H` of cardinality $p^i$, we are going to prove that there exists a subgroup `L` of cardinality $p^{i+1}$. We are acting by left translation with `H` on the left cosets of `H` with respect to `K` as follows:

`x` $\mapsto$ `{ yH` $\mapsto$ `(xy)H }`

The `mpl` lemma gives us $(\mathtt{card}\ \mathtt{S}_0)\ \%\ \mathtt{p}\ =\ (\mathtt{lindex}\ \mathtt{H}\ \mathtt{K})\ \%\ \mathtt{p}$. But by Lagrange's theorem we know that `(lindex H K)` is equal to $p^{n-i}s$. As $i < n$, we can conclude that the cardinal of $\mathtt{S}_0$ is divisible by $p$. Now, if we look at the cosets that are in $\mathtt{S}_0$. They are the `yH` such that `(xy)H = yH` for all `x` in `H`. This corresponds to $\mathtt{y}^{-1}$`Hy = H` so `y` is in `(normaliser H K)`. So, we can deduce that `(card` $\mathtt{S}_0$`) = (lindex H (normaliser H K))`. This means that if we take the quotient of the normaliser `(normaliser H K)` by `H`, this is a group (`H` is normal in its normaliser) and its cardinality which is `(lindex H (normaliser H K))` is divisible by $p$. We can then apply Cauchy's theorem and get the existence of a subgroup $\mathtt{L}_1$ of cardinality $p$ in the quotient. Taking the inverse image of $\mathtt{L}_1$ by the quotient operation, we get a subgroup `L` of `G` whose cardinality is `card` $\mathtt{L}_1$ `* card H = p` $\mathtt{p}^i$ `=` $\mathtt{p}^{i+1}$. This ends the proof of the first Sylow theorem. The exact formal statement of this theorem is the following:

```
Theorem sylow1_cor: forall G (K: set G) p,
       subgrp K -> prime p -> 0 < dlogn p (card K) ->
       exists H : set G, sylow K p H.
```

The second Sylow theorem says that two Sylow $p$ subgroups $\mathtt{L}_1$ and $\mathtt{L}_2$ of `K` are conjugate. For the proof, we act by left translation with $\mathtt{L}_2$ on the left coset of $\mathtt{L}_1$. By the `mpl` lemma, we know the `(card` $\mathtt{S}_0$`) % p = (lindex` $\mathtt{L}_1$ `K) % p`. As $\mathtt{L}_1$ is a Sylow $p$ group, we have by Lagrange's theorem that `(lindex` $\mathtt{L}_1$ `K)` is equal to $s$, so is not divisible by $p$. This means that `(card` $\mathtt{S}_0$`)` is not divisible by $p$, so there exists an `x` in `K` such that $\mathtt{xL}_1$ is in $\mathtt{S}_0$. But for this `x`, we know that for all `y` in $\mathtt{L}_2$, `(yx)`$\mathtt{L}_1$ `=` $\mathtt{xL}_1$, this means that $\mathtt{L}_2$ is included in $\mathtt{xL}_1\mathtt{x}^{-1}$. As both sets have same cardinality, we have $\mathtt{L}_2$ $=_1$ $\mathtt{xL}_1\mathtt{x}^{-1}$. The exact formal statement of this theorem is the following:

```
Theorem sylow2_cor:  forall G (K: set G) p L₁ L₂,
    subgrp K -> prime p -> 0 < dlogn p (card K) ->
    sylow K p L₁ -> sylow K p L₂ ->
      exists x :  G, K x /\ L₂ =₁ conjsg L₁ x.
```

The third Sylow theorem gives an indication on the number of Sylow $p$ groups. It says that this number divides the cardinality of `K` and is equal to 1 modulo $p$. In order to count the number of Sylow $p$ subgroup, we have to define the sylow subset of the power set of `G` as:

```
Definition syset K p := fun (H: powerSet G) => sylow K p (subdE H).
```

Now, the first part of the third theorem that regards divisibility is proved acting with `K` on `(syset K p)` as follows:

$\mathtt{x} \mapsto \{ \mathtt{L} \mapsto \mathtt{xLx}^{-1} \}$

The second theorem tells us that all the elements of `(syset K p)` are conjugate. So, from one Sylow $p$ subgroup `L` we can reach any other by conjugation. This means that `(syset K p)` contains one single orbit. So, `(card (syset K p)) = (card (orbit L))`. The theorem `card_orbit` tells us the `card (orbit L)` is equal to `(lindex (stabiliser L) K)`. Using Lagrange's theorem, we get that it divides `(card K)`. The formal statement of the first part of the third Sylow theorem is the following:

```
Theorem sylow3_div: forall G (K: set G) p,
       subgrp K -> prime p -> 0 < dlogn p (card k) ->
       (card (syset K p)) | (card K).
```

For the second part, we consider `H` a Sylow $p$ group for `K`. We act with `H` on `(syset K p)` by conjugation as before:

`x` $\mapsto$ `{ L` $\mapsto$ `xLx`$^{-1}$ `}`
An element L is in $S_0$ if `xLx`$^{-1}$ $=_1$ `L` for all `x` in `H`. This means that `H` is included in `(normaliser L K)`. As we have `(sylow K p H)`, we have also `(sylow (normaliser L K) p H)`. This holds also for `L`, so we have `(sylow (normaliser L K) p L)`. The second theorem tells us that `H` and `L` are then conjugate in `(normaliser L K)`. But as `L` is normal in its normaliser, this implies that `H` $=_1$ `L`. So `(card` $S_0$`)` is equal to 1. If we apply the `mpl` lemma we get the expected result. The formal statement of the second part of the third Sylow theorem is the following:

```
Theorem sylow3_mod: forall G (K: set G) p,
       subgrp K -> prime p -> 0 < dlogn p (card k) ->
       (card (syset K p)) % p = 1.
```

# 4 Conclusion

Formalising Sylow's theorems has been surprisingly smooth. One reason has to do with the fact that we have built our development on top of SSREFLECT. This base was used by Georges Gonthier for his proof of the four colour theorem. It has already been tested on a large development, so it is quite complete. The only basic construction we had to add is the power set. Another reason that made our life simpler is that we were working in a decidable fragment of the COQ logic. No philosophical issue about constructiveness slowed down our formalisation. Finally, Gregory Constantine's proof was perfect for our formalisation work. The only part of the formalisation that was ad-hoc was the construction of the set `H`$^*$. It represents only 360 lines of the 3550 lines of the formalisation. The fact that this experiment was positive is clearly a good sign for further formalisations in group theory.

# References


[1] Gregory M. Constantine. Group Theory. Available at `http://www.pitt.edu/~gmc/algsyl.html`.

[2] Georges Gonthier. Notation of the Four Colour Theorem proof. Available at `http://research.microsoft.com/~gonthier/4colnotations.pdf`.

[3] Florian Kammüller and Lawrence C. Paulson. A Formal Proof of Sylow's Theorem. *Journal of Automating Reasoning*, 23(3-4):235–264, 1999.

[4] Freek Wiedijk. Formalizing 100 theorems. Available at `http://www.cs.ru.nl/~freek/100/`.

[5] Helmut Wielandt. Ein beweis für die existenz der sylowgruppen. *Archiv der Mathematik*, 10:401–402, 1959.

## Module groups

*Structure finGroup*: *Type*:= *Finite* {
  *element*:> *finType*;
  *unit* : *element*;
  *inv* : *element* → *element*;
  *mul* : *element* → *element* → *element*;
  *unitP* : $\forall x$, *mul unit* $x = x$;
  *invP* : $\forall x$, *mul* (*inv* $x$) $x =$ *unit*;
  *mulP* : $\forall x_1\ x_2\ x_3$, *mul* $x_1$ (*mul* $x_2\ x_3$) = *mul* (*mul* $x_1\ x_2$) $x_3$
}.

Section *GroupIdentities*.

Variable $G$: *finGroup*.

Lemma *mulgA*: $\forall x_1\ x_2\ x_3$: $G$, $x_1 \times (x_2 \times x_3) = x_1 \times x_2 \times x_3$.

Lemma *mul1g*: $\forall x$: $G$, $1 \times x = x$.

Lemma *mulVg*: $\forall x$: $G$, $x^{-1} \times x = 1$.

Lemma *mulg_invl*: $\forall x$: $G$, *cancel* (*mulg* $x$) (*mulg* $x^{-1}$).

Lemma *mulg_injl*: $\forall x$: $G$, *injective* (*mulg* $x$).

Lemma *mulg1*: $\forall x$: $G$, $x \times 1 = x$.

Lemma *invg1*: $1^{-1} = 1$.

Lemma *mulgV*: $\forall x$: $G$, $x \times x^{-1} = 1$.

Lemma *mulg_invr*: $\forall x$: $G$, *monic* (*mulgr* $x$) (*mulgr* $x^{-1}$).

Lemma *mulg_injr*: $\forall x$: $G$, *injective* (*mulgr* $x$).

Lemma *invg_inv*: *monic invg invg*.

Lemma *invg_inj*: *injective invg*.

Lemma *invg_mul*: $\forall x_1\ x_2$: $G$, $(x_2 \times x_1)^{-1} = {x_1}^{-1} \times {x_2}^{-1}$.

Lemma *mulVg_invl*: $\forall x$: $G$, *monic* (*mulg* $x^{-1}$) (*mulg* $x$).

Lemma *mulVg_invr*: $\forall x$, *monic* (*mulgr* $x^{-1}$) (*mulgr* $x$).

Theorem *mulg_s*$_1$: $\forall a\ b{:}G$, $(b \times a^{-1}) \times a = b$.

Theorem *mulg_s*$_2$: $\forall a\ b{:}G$, $(b \times a) \times a^{-1} = b$.

End *GroupIdentities*.

Definition *conjg* ($G$: *finGroup*) ($x\ y$: $G$):= $x^{-1} \times y \times x$.

Section $Conjugation$.

Variable $G$: $finGroup$.

Lemma $conjgE$: $\forall x\ y$: $G$, $x^y = y^{-1} \times x \times y$.

Lemma $conjg1$: $conjg\ 1 =_1 id$.

Lemma $conj1g$: $\forall x$: $G$, $1^x = 1$.

Lemma $conjg\_mul$: $\forall x_1\ x_2\ y$: $G$, $(x_1 \times x_2)^y = x_1{}^y \times x_2{}^y$.

Lemma $conjg\_invg$: $\forall x\ y$: $G$, $(x^{-1})^y = (x^{\ y})^{-1}$.

Lemma $conjg\_conj$: $\forall x\ y_1\ y_2$: $G$, $(x^{y_1})^{y_2} = x^{y_1 \times y_2}$.

Lemma $conjg\_inv$: $\forall y$: $G$, $monic\ (conjg\ y)\ (conjg\ y^{-1})$.

Lemma $conjg\_invV$: $\forall y$: $G$, $monic\ (conjg\ y^{-1})\ (conjg\ y)$.

Lemma $conjg\_inj$: $\forall y$: $G$, $injective\ (conjg\ y)$.

Definition $conjg\_fp\ (y\ x$: $G)$:= $x^y =_d x$.

Definition $commg\ (x\ y$: $G)$:= $x \times y = y \times x$.

Lemma $conjg\_fpP$: $\forall x\ y$: $G$, $reflect\ (commg\ x\ y)\ (conjg\_fp\ y\ x)$.

Lemma $conjg\_fp\_sym$: $\forall x\ y$: $G$, $conjg\_fp\ x\ y = conjg\_fp\ y\ x$.

End $Conjugation$.

Section $SubGroup$.

Variables ($G$: $finGroup$) ($H$: $set\ G$).

Definition $lcoset\ x$: $set\ G$:= $fun\ y \Rightarrow H\ (x^{-1} \times y)$.
Definition $rcoset\ x$: $set\ G$:= $fun\ y \Rightarrow H\ (y \times x^{-1})$.

Definition $subgrpb$:= $H\ 1\ \&\&\ subset\ H\ (fun\ x \Rightarrow subset\ H\ (rcoset\ x))$.

Definition $subgrp$: $Prop$:= $subgrpb$.

Lemma $subgrpP$: $reflect\ (H\ 1 \land \forall x\ y, H\ x \to H\ y \to rcoset\ x\ y)\ subgrpb$.

Hypothesis $Hh$: $subgrp$.

Lemma $subgrp1$: $H\ 1$.

Lemma $subgrpV$: $\forall x$, $H\ x \to H\ x^{-1}$.

Lemma $subgrpM$: $\forall x\ y$, $H\ x \to H\ y \to H\ (x \times y)$.

Lemma $subgrpMl$: $\forall x\ y$, $H\ x \to H\ (x \times y) = H\ y$.

Lemma $subgrpMr$: $\forall x\ y$, $H\ x \to H\ (y \times x) = H\ y$.

Lemma $subgrpVl$: $\forall x,\ H\ x^{-1} \rightarrow H\ x.$

Definition $subFinGroup$: $finGroup.$

End $SubGroup.$

Lemma $subgrp\_of\_group$: $\forall G$: $finGroup,\ subgrp\ G.$

*Coercion* $subgrp\_of\_group$: $finGroup$ >-> $subgrp.$

Section $LaGrange.$

Variables $(G$: $finGroup)$ $(H$: $set\ G).$
Hypothesis $(Hh$: $subgrp\ H).$

Lemma $rcoset\_refl$: $\forall x,\ rcoset\ H\ x\ x.$

Lemma $rcoset\_sym$: $\forall x\ y,\ rcoset\ H\ x\ y = rcoset\ H\ y\ x.$

Lemma $rcoset\_trans$: $\forall x\ y,\ connect\ (rcoset\ H)\ x\ y = rcoset\ H\ x\ y.$

Lemma $rcoset\_csym$: $connect\_sym\ (rcoset\ H).$

Lemma $rcoset1$: $rcoset\ H\ 1 =_1 H.$

Lemma $card\_rcoset$: $\forall x,\ card\ (rcoset\ H\ x) = card\ H.$

Definition $rindex := n\_comp\ (rcoset\ H).$

Theorem $rLaGrange$: $\forall K$: $set\ G,$
$subgrp\ K \rightarrow subset\ H\ K \rightarrow card\ H \times rindex\ K = card\ K.$

Theorem $sugrp\_divn$: $\forall K$: $set\ G,$
$subgrp\ K \rightarrow subset\ H\ K \rightarrow card\ H \mid card\ K.$

Lemma $lcoset\_refl$: $\forall x,\ lcoset\ H\ x\ x.$

Lemma $lcoset\_sym$: $\forall x\ y,\ lcoset\ H\ x\ y = lcoset\ H\ y\ x.$

Lemma $lcoset\_trans$: $\forall x\ y,\ connect\ (lcoset\ H)\ x\ y = lcoset\ H\ x\ y.$

Lemma $lcoset\_csym$: $connect\_sym\ (lcoset\ H).$

Lemma $lcoset1$: $lcoset\ H\ 1 =_1 H.$

Lemma $card\_lcoset$: $\forall x,\ card\ (lcoset\ H\ x) = card\ H.$

Definition $lindex := n\_comp\ (lcoset\ H).$

Theorem $lLaGrange$: $\forall K$: $set\ G,$
$subgrp\ K \rightarrow subset\ H\ K \rightarrow card\ H \times lindex\ K = card\ K.$

End $LaGrange.$

Section $FinPart.$

Variables ($G$: *finGroup*) ($H$: *set* $G$) ($a$: $G$).
Hypothesis *Ha*: $H$ $a$.
Hypothesis *Hstable*: $\forall x\ y$, $H\ x \rightarrow H\ y \rightarrow H\ (x \times y)$.

Lemma *heqah*: (*lcoset* $H$ $a$) $=_1$ $H$.

Lemma *heqxh*: $\forall x$, $H\ x \rightarrow$ (*lcoset* $H$ $x$) $=_1$ $H$.

Lemma *heqhx*: $\forall x$, $H\ x \rightarrow$ (*rcoset* $H$ $x$) $=_1$ $H$.

Lemma *finstbl_sbgrp1*: $H$ 1.

Lemma *finstbl_mulV*: $\forall x$, $H\ x \rightarrow H\ x^{-1}$.

Lemma *finstbl_sbgrp*: *subgrp* $H$.

End *FinPart*.

Section *Eq*.

Variable $G$: *finGroup*.

Theorem *eq_subgroup*: $\forall a\ b$: *set* $G$, $a =_1 b \rightarrow$ *subgrpb* $a$ = *subgrpb* $b$.

End *Eq*.

Section *SubProd*.
Variable $G$: *finGroup*.

Section *SubProd_subgrp*.

Variables ($H$ $K$: *set* $G$).
Hypothesis *h_subgroup*: *subgrp* $H$.
Hypothesis *k_subgroup*: *subgrp* $K$.

Lemma *subprod_sbgrp*: *prod* $H$ $K$ $=_1$ *prod* $K$ $H$ $\rightarrow$ *subgrp* (*prod* $H$ $K$).

Lemma *sbgrp_subprod*: *subgrp* (*prod* $H$ $K$) $\rightarrow$ *prod* $H$ $K$ $=_1$ *prod* $K$ $H$.

End *SubProd_subgrp*.

Variables ($H$ $K$: *set* $G$).
Hypothesis *h_subgroup*: *subgrp* $H$.
Hypothesis *k_subgroup*: *subgrp* $K$.

Lemma *sbgrphk_sbgrpkh*: *subgrpb* (*prod* $H$ $K$) = *subgrpb* (*prod* $K$ $H$).

End *SubProd*.

# Module action

Section *Action*.

Variable $(G\colon \mathit{finGroup})$ $(H\colon \mathit{set}\ G)$.

Hypothesis $\mathit{sgrp\_h}$: $\mathit{subgrp}\ H$.
Variable $s$: $\mathit{finType}$.

Variable $\mathit{to}$: $G \to (s \to s)$.
Hypothesis $\mathit{to\_bij}$: $\forall x,\ H\ x \to \mathit{bijective}\ (\mathit{to}\ x)$.
Hypothesis $\mathit{to\_morph}$: $\forall (x\ y\colon G)\ z,$
$H\ x \to H\ y \to \mathit{to}\ (x \times y)\ z = \mathit{to}\ x\ (\mathit{to}\ y\ z)$.

Theorem $\mathit{to\_1}$: $\forall x,\ \mathit{to}\ 1\ x = x$.

Definition $\mathit{stabiliser}\ a{:=}\ \mathit{setI}\ (\mathit{fun}\ x \Rightarrow ((\mathit{to}\ x\ a) =_d a))\ H$.
Definition $\mathit{orbit}\ a{:=}\ \mathit{image}\ (\mathit{fun}\ z \Rightarrow \mathit{to}\ z\ a)\ H$.

Theorem $\mathit{orbit\_to}$: $\forall a\ x,\ H\ x \to \mathit{orbit}\ a\ (\mathit{to}\ x\ a)$.

Lemma $\mathit{orbit\_refl}$: $\forall x,\ \mathit{orbit}\ x\ x$.

Lemma $\mathit{orbit\_sym}$: $\forall x\ y,\ \mathit{orbit}\ x\ y = \mathit{orbit}\ y\ x$.

Lemma $\mathit{orbit\_trans}$: $\forall x\ y,\ \mathit{connect}\ \mathit{orbit}\ x\ y = \mathit{orbit}\ x\ y$.

Lemma $\mathit{orbit\_csym}$: $\mathit{connect\_sym}\ \mathit{orbit}$.

Definition $S_0\ a{:=}\ \mathit{subset}\ H\ (\mathit{stabiliser}\ a)$.

Theorem $\mathit{S0P}$: $\forall a,\ \mathit{reflect}\ (\mathit{orbit}\ a =_1 \mathit{set1}\ a)\ (S_0\ a)$.

Theorem $\mathit{stab\_1}$: $\forall a,\ \mathit{stabiliser}\ a\ 1$.

Theorem $\mathit{subgr\_stab}$: $\forall a,\ \mathit{subgrp}\ (\mathit{stabiliser}\ a)$.

Theorem $\mathit{subset\_stab}$: $\forall a,\ \mathit{subset}\ (\mathit{stabiliser}\ a)\ H$.

Theorem $\mathit{orbit\_from}$: $\forall a\ x\ (\mathit{Hx}\colon \mathit{orbit}\ a\ x),$
$(\mathit{setI}\ (\mathit{roots}\ (\mathit{lcoset}\ (\mathit{stabiliser}\ a)))\ H)\ (\mathit{root}\ (\mathit{lcoset}\ (\mathit{iinv1}\ \mathit{Hx}))$.

Theorem $\mathit{card\_orbit}$: $\forall a,\ \mathit{card}\ (\mathit{orbit}\ a) = \mathit{lindex}\ (\mathit{stabiliser}\ a)\ H$.

Theorem $\mathit{card\_orbit\_div}$: $\forall a,\ \mathit{card}\ (\mathit{orbit}\ a) \mid \mathit{card}\ H$.

Variable $n\ p$: $\mathit{nat}$.
Hypothesis $\mathit{prime\_p}$: $\mathit{prime}\ p$.
Hypothesis $\mathit{card\_h}$: $\mathit{card}\ H = p^n$.

Theorem $\mathit{mpl}$: $(\mathit{card}\ s)\ \%\ p = (\mathit{card}\ S_0)\ \%\ p$.

End $\mathit{Action}$.

# Module cyclic

Section *Phi*.
Definition *phi* $n$:= *if* $n$ *is* $n_1 + 1$ *then card* (*fun* $x \Rightarrow$ *coprime* $n$ (*val* $x$)) *else* 0.

Theorem *phi_mult*: $\forall m\ n$, *coprime* $m\ n \rightarrow$ *phi* $(m \times n) =$ *phi* $m \times$ *phi* $n$.

Theorem *phi_prime_k*: $\forall p\ k$, *prime* $p \rightarrow$ *phi* $p^{k+1} = p^{k+1}$ - $p^k$.

End *Phi*.

Section *Cyclic*.

Variable $G$: *finGroup*.

Fixpoint *gexpn* ($a$:$G$) ($n$: *nat*) {*struct* $n$}: $G$:=
  *if* $n$ *is* $n_1 + 1$ *then* $a \times$ (*gexpn* $a\ n_1$) *else* 1.

Theorem *gexpn0*: $\forall a$, *gexpn* $a$ $0 = 1$.

Theorem *gexpn1*: $\forall a$, *gexpn* $a$ $1 = a$.

Theorem *gexp1n*: $\forall n$, *gexpn* $1\ n = 1$.

Theorem *gexpnS*: $\forall a\ n$, *gexpn* $a$ $(n + 1)) = a \times$ *gexpn* $a\ n$.

Theorem *gexpn_h*: $\forall n\ a\ H$, *subgrp* $H \rightarrow H\ a \rightarrow H$ (*gexpn* $a\ n$).

Theorem *gexpn_add*: $\forall a\ n\ m$, *gexpn* $a\ n \times$ *gexpn* $a\ m =$ *gexpn* $a$ $(n + m)$.

Theorem *gexpn_mul*: $\forall a\ n\ m$, *gexpn* (*gexpn* $a\ n$) $m =$ *gexpn* $a$ $(n \times m)$.

Fixpoint *seq_fn* ($f$: $G \rightarrow G$) ($n$: *nat*) ($a$: $G$) ($L$: *seq* $G$) {*struct* $n$}: *seq* $G$:=
  *if* $n$ *is* $n_1 + 1$ *then*
    *if negb* ($L\ a$) *then seq_fn* $f\ n_1$ ($f\ a$) (*Adds* $a\ L$) *else* $L$ *else* $L$.

Definition *seq_f* $f\ a$:= *seq_fn* $f$ (*card* $G$) $a$ (*Seq0* _).

Definition *cyclic* $a$:= *seq_f* (*fun* $x \Rightarrow a \times x$) 1.

Theorem *cyclic1*: $\forall a$, *cyclic* $a$ 1.

Theorem *cyclicP*: $\forall a\ b$, *reflect* ($\exists\ n$, *gexpn* $a\ n =_d b$) (*cyclic* $a\ b$).

Theorem *cyclic_h*: $\forall a\ H$, *subgrp* $H \rightarrow H\ a \rightarrow$ *subset* (*cyclic* $a$) $H$.

Theorem *cyclic_min*: $\forall a\ b$,
  *cyclic* $a\ b \rightarrow \exists\ m$, ($m <$ *card* (*cyclic* $a$)) && (*gexpn* $a\ m =_d b$).

Theorem *cyclic_in*: $\forall a\ m$, *cyclic* $a$ (*gexpn* $a\ m$).

Theorem *subgr_cyclic*: $\forall a$, *subgrp* (*cyclic* $a$).

Theorem *cyclic_expn_card*: $\forall a$, *gexpn* $a$ (*card* (*cyclic* $a$)) $=_d 1$.

Theorem *cyclic_div_card*: $\forall a\ n$, *card* (*cyclic* $a$) $\mid n$) = (*gexpn* $a\ n =_d 1$).

Theorem *cyclic_div_g*: $\forall a$, *card* (*cyclic* $a$) $\mid$ *card* $G$.

# Module normal

Section *Normal*.

Variables $(G$: *finGroup*$)$ $(H\ K$: *set* $G)$.

Hypothesis *sgrp_h*: *subgrp* $H$.
Hypothesis *sgrp_k*: *subgrp* $K$.
Hypothesis *subset_hk*: *subset* $H\ K$.

Definition *conjsg* $x\ y$:= $H(y^x)$.

Theorem *conjsg1*: $\forall x$, *conjsg* $x$ 1.

Theorem *conjs*$_1$*g*: $\forall x$, *conjsg* $1\ x = H\ x$.

Theorem *conjsg_inv*: $\forall x\ y$, *conjsg* $x\ y \rightarrow$ *conjsg* $x\ y^{-1}$.

Theorem *conjsg_conj*: $\forall x\ y\ z$, *conjsg* $(x \times y)\ z =$ *conjsg* $y\ (z^x)$.

Theorem *conjsg_subgrp*: $\forall x$, *subgrp* (*conjsg* $x$).

Theorem *conjsg_image*: $\forall y$,
*conjsg* $y =_1$ *image* (*conjg* $y^{-1}$) $H$.

Theorem *conjsg_inv1*: $\forall x$,
(*conjsg* $x$) $=_1 H \rightarrow$ (*conjsg* $x^{-1}$) $=_1 H$.

Theorem *conjsg_card*: $\forall x$,
*card* (*conjsg* $x$) = *card* $H$.

Theorem *conjsg_subset*: $\forall x$,
*subset* $H$ (*conjsg* $x$) $\rightarrow$ (*conjsg* $x$) $=_1 H$.

Theorem *lcoset_root*: $\forall x$, *lcoset* $H\ x$ (*root* (*lcoset* $H$) $x$).

Definition *normalb*:= *subset* $K$ (*fun* $x \Rightarrow$ *subset* $H$ (*conjsg* $H\ x$)).

Definition *normal*: *Prop*:= *normalb*.

Hypothesis *normal_k*: *normal*.

Theorem *conjsg_normal*: $\forall x$, $K\ x \rightarrow$ *conjsg* $x =_1 H$.

Definition *rootSet*:= *subFin* (*setI* (*roots* (*lcoset* $H$)) $K$).

Theorem *card_rootSet*: *card rootSet* = *lindex* $H\ K$.

Theorem *unit_root_sub*:
*setI* (*roots* (*lcoset* $H$)) $K$ (*root* (*lcoset* $H$) 1).

Definition *unit_root*: *rootSet*.

Definition *mult_root*: *rootSet* $\rightarrow$ *rootSet* $\rightarrow$ *rootSet*.

Definition $inv\_root$: $rootSet \rightarrow rootSet$.

Theorem $unitP\_root$: $\forall x$, $mult\_root\ unit\_root\ x = x$.

Theorem $invP\_root$: $\forall x$, $mult\_root\ (inv\_root\ x)\ x = unit\_root$.

Theorem $mulP\_root$: $\forall x_1\ x_2\ x_3$,
$mult\_root\ x_1\ (mult\_root\ x_2\ x_3) = mult\_root\ (mult\_root\ x_1\ x_2)\ x_3$.

Definition $root\_group$:= $(Group.Finite\ unitP\_root\ invP\_root\ mulP\_root)$.

Theorem $card\_root\_group$: $card\ root\_group = lindex\ H\ K$.

End $Normal$.

Section $NormalProp$.

Variables $(G$: $finGroup)$ $(H\ K$: $set\ G)$.
Hypothesis $sgrp\_h$: $subgrp\ H$.
Hypothesis $sgrp\_k$: $subgrp\ K$.
Hypothesis $subset\_hk$: $subset\ H\ K$.
Hypothesis $normal\_hk$: $normal\ H\ K$.

Theorem $normal\_subset$: $\forall L$,
$subgrp\ L \rightarrow subset\ H\ L \rightarrow subset\ L\ K \rightarrow normal\ H\ L$.

Definition $RG$:= $(root\_group\ sgrp\_h\ sgrp\_k\ subset\_hk\ normal\_hk)$.

Theorem $th\_quotient$: $\forall x$, $K\ x \rightarrow$
$(setI\ (roots\ (lcoset\ H))\ K\ (root\ (lcoset\ H)\ x))$.

Definition $quotient$: $G \rightarrow RG$.

Theorem $quotient\_lcoset$: $\forall x$, $K\ x \rightarrow lcoset\ H\ x\ (val\ (quotient\ x))$.

Theorem $quotient1$: $\forall x$, $H\ x \rightarrow quotient\ x = 1$.

Theorem $quotient\_morph$: $\forall x\ y$,
$K\ x \rightarrow K\ y \rightarrow quotient(x \times y) = quotient(x) \times quotient(y)$.

Theorem $quotient\_image\_subgrp$: $\forall L$,
$subset\ H\ L \rightarrow subset\ L\ K \rightarrow subgrp\ L \rightarrow subgrp\ (image\ quotient\ L)$.

Theorem $quotient\_preimage\_subgrp$: $\forall L$,
$subgrp\ L \rightarrow subgrp\ (setI\ (preimage\ quotient\ L)\ K)$.

Theorem $quotient\_preimage\_subset\_h$: $\forall L$,
$subgrp\ L \rightarrow subset\ H\ (setI\ (preimage\ quotient\ L)\ K)$.

Theorem $quotient\_preimage\_subset\_k$: $\forall L$, $subset\ (setI\ (preimage\ quotient\ L)\ K)\ K$.

Theorem $quotient\_index$: $\forall L$, $subset\ H\ L \rightarrow subset\ L\ K \rightarrow subgrp\ L \rightarrow$
$lindex\ H\ L = card\ (image\ quotient\ L)$.

Theorem $\mathit{quotient\_image\_preimage}$: $\forall L$,
$\mathit{image}\ \mathit{quotient}\ (\mathit{setI}\ (\mathit{preimage}\ \mathit{quotient}\ L)\ K) =_1 L$.

End $\mathit{NormalProp}$.

Section $\mathit{Normalizer}$.

Variables $(G\colon \mathit{finGroup})\ (H\ K\colon \mathit{set}\ G)$.

Hypothesis $\mathit{sgrp\_h}$: $\mathit{subgrp}\ H$.
Hypothesis $\mathit{sgrp\_k}$: $\mathit{subgrp}\ K$.
Hypothesis $\mathit{subset\_hk}$: $\mathit{subset}\ H\ K$.

Definition $\mathit{normaliser}\ x$:=
$(\mathit{subset}\ K\ (\mathit{fun}\ z \Rightarrow (\mathit{conjsg}\ x\ z =_d H\ z)))\ \&\&\ K\ x$.

Theorem $\mathit{normaliser\_grp}$: $\mathit{subgrp}\ \mathit{normaliser}$.

Theorem $\mathit{normaliser\_subset}$: $\mathit{subset}\ \mathit{normaliser}\ K$.

Theorem $\mathit{subset\_normaliser}$: $\mathit{subset}\ H\ \mathit{normaliser}$.

Theorem $\mathit{normaliser\_normal}$: $\mathit{normal}\ H\ \mathit{normaliser}$.

Theorem $\mathit{card\_normaliser}$:
$\mathit{card}\ (\mathit{root\_group}\ \mathit{sgrp\_h}\ \mathit{normaliser\_grp}\ \mathit{subset\_normaliser}$
$\mathit{normaliser\_normal}) = \mathit{lindex}\ H\ \mathit{normaliser}$.

End $\mathit{Normalizer}$.

Section $\mathit{Eq}$.

Variables $G$: $\mathit{finGroup}$.

Theorem $\mathit{eq\_conjsg}$: $\forall a\ b\ x,\ a =_1 b \rightarrow \mathit{conjsg}\ a\ x =_1 \mathit{conjsg}\ b\ x$.

End $\mathit{Eq}$.

Section $\mathit{Root}$.

Variable $(G\colon \mathit{finGroup})\ (H\colon \mathit{set}\ G)$.

Hypothesis $\mathit{sgrp\_h}$: $\mathit{subgrp}\ H$.

Theorem $\mathit{root\_lcoset1}$: $H\ (\mathit{root}\ (\mathit{lcoset}\ H)\ 1)$.

Theorem $\mathit{root\_lcosetd}$: $\forall a,\ H\ (a^{-1} \times \mathit{root}\ (\mathit{lcoset}\ H)\ a)$.

End $\mathit{Root}$.

# Module leftTranslation

Section *LeftTrans*.

Variable ($G$: *finGroup*) ($H$ $K$ $L$: *set* $G$).

Hypothesis *sgrp_k*: *subgrp* $K$.
Hypothesis *sgrp_l*: *subgrp* $L$.
Hypothesis *sgrp_h*: *subgrp* $H$.
Hypothesis *subset_hk*: *subset* $H$ $K$.
Hypothesis *subset_lk*: *subset* $L$ $K$.

Definition *ltrans*: $G \rightarrow$ *rootSet* $L$ $K \rightarrow$ *rootSet* $L$ $K$.

Theorem *ltrans_bij*: $\forall x$, $H$ $x \rightarrow$ *bijective* (*ltrans* $x$).

Theorem *ltrans_morph*: $\forall x$ $y$ $z$,
$H$ $x \rightarrow H$ $y \rightarrow$ *ltrans* $(x \times y)$ $z =$ *ltrans* $x$ (*ltrans* $y$ $z$).

End *LeftTrans*.

# Module sylow

Section *Cauchy*.

Variable ($G$: *finGroup*) ($H$: *set* $G$).
Hypothesis *sgrp_h*: *subgrp* $H$.

Variable $p$: *nat*.
Hypothesis *prime_p*: *prime* $p$.
Hypothesis *p_divides_h*: $p$ | *card* $H$.

Theorem *cauchy*: $\exists$ $a$,$H$ $a$ && *card* (*cyclic* $a$) $=_d$ $p$.

End *Cauchy*.

Section *Sylow*.

Variable ($G$: *finGroup*) ($K$: *set* $G$).
Hypothesis *sgrp_k*: *subgrp* $K$.

Variable $p$: *nat*.
Hypothesis *prime_p*: *prime* $p$.

Let $n$:= *dlogn* $p$ (*card* $K$).

Hypothesis *n_pos*: $0 < n$.

Definition *sylow* $L$:= (*subgrpb* $L$) && (*subset* $L$ $K$) && (*card* $L =_d p^n$).

Theorem $eq\_sylow$: $\forall a\ b,\ a =_1 b \rightarrow sylow\ a = sylow\ b$.

Theorem $sylow\_conjsg$: $\forall L_1\ x,\ K\ x \rightarrow sylow\ L_1 \rightarrow sylow\ (conjsg\ L_1\ x)$.

Theorem $sylow1\_rec$: $\forall i\ Hi,\ 0 < i \rightarrow i < n \rightarrow$
$subgrp\ Hi \rightarrow subset\ Hi\ K \rightarrow card\ Hi = p^i \rightarrow$
$\exists\ H\colon set\ G,$
$subgrp\ H \wedge subset\ Hi\ H \wedge subset\ H\ K \wedge normal\ Hi\ H \wedge card\ H = p^{i+1}$.

Theorem $sylow1$: $\forall i,\ 0 < i \rightarrow i \leq n \rightarrow$
$\exists\ H\colon set\ G,\ subgrp\ H \wedge subset\ H\ K \wedge card\ H = p^i$.

Theorem $sylow1\_cor$: $\exists\ H\colon set\ G,\ sylow\ H$.

Theorem $sylow2$: $\forall H\ L\ i, 0 < i \rightarrow i \leq n \rightarrow$
$subgrp\ H \rightarrow subset\ H\ K \rightarrow card\ H = p^i \rightarrow sylow\ L \rightarrow$
$\exists\ x,\ (K\ x)\ \&\&\ subset\ H\ (conjsg\ L\ x)$.

Theorem $sylow2\_cor$: $\forall L_1\ L_2,\ sylow\ L_1 \rightarrow sylow\ L_2 \rightarrow$
$\exists\ x,\ (K\ x) \wedge (L_2 =_1 conjsg\ L_1\ x)$.

Definition $syset\ p := sylow\ (val\ p)$.

Theorem $sylow3\_div$: $card\ syset \mid card\ K$.

End *Sylow*.

Section *SylowAux*.

Variable $(G\colon finGroup)\ (H\ K\ L\colon set\ G)$.
Hypothesis $sgrp\_k$: $subgrp\ K$.
Hypothesis $sgrp\_l$: $subgrp\ L$.
Hypothesis $sgrp\_h$: $subgrp\ H$.
Hypothesis $subset\_hl$: $subset\ H\ L$.
Hypothesis $subset\_lk$: $subset\ L\ K$.

Variable $p$: $nat$.
Hypothesis $prime\_p$: $prime\ p$.
Let $n := dlogn\ p\ (card\ K)$.
Hypothesis $n\_pos$: $0 < n$.

Theorem $sylow\_subset$: $sylow\ K\ p\ H \rightarrow sylow\ L\ p\ H$.

End *SylowAux*.

Section *Sylow3*.

Variable $(G\colon finGroup)\ (K\colon set\ G)$.
Hypothesis $sgrp\_k$: $subgrp\ K$.

Variable $p$: $nat$.

Hypothesis $prime\_p$: $prime\ p$.

Let $n$:= $dlogn\ p\ (card\ K)$.

Hypothesis $n\_pos$: $0 < n$.

Theorem $sylow3\_mod$: $card\ (syset\ K\ p)\ \%\ p = 1$.

End $Sylow3$.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-0803