

Secondary Sylow Theorems

Jakob von Raumer

August 16, 2018

Abstract

These theories extend the existent proof of the first sylow theorem (written by Florian Kammüller and L. C. Paulson) by what is often called the second, third and fourth sylow theorem. These theorems state propositions about the number of Sylow p -subgroups of a group and the fact that they are conjugate to each other. The proofs make use of an implementation of group actions and their properties.

Contents

1	Group Actions	2
1.1	Preliminaries and Definition	2
1.2	The orbit relation	4
1.3	Stabilizer and fixed points	6
1.4	The Orbit-Stabilizer Theorem	8
1.5	Some Examples for Group Actions	13
2	Conjugation of Subgroups and Cosets	16
2.1	Definitions and Preliminaries	16
2.2	Conjugation is a group action	18
2.3	Properties of the Conjugation Action	23
3	The Secondary Sylow Theorems	25
3.1	Preliminaries	25
3.2	Extending the Sylow Locale	26
3.3	Every p -group is Contained in a conjugate of a p -Sylow-Group	26
3.4	Every p -Group is Contained in a p -Sylow-Group	28
3.5	p -Sylow-Groups are conjugates of each other	28
3.6	Counting Sylow-Groups	29

```
theory GroupAction
imports
  HOL-Algebra.Bij
```

HOL-Algebra.Sylow
begin

1 Group Actions

This is an implementation of group actions based on the group implementation of HOL-Algebra. An action a group G on a set M is represented by a group homomorphism between G and the group of bijections on M

1.1 Preliminaries and Definition

First, we need two theorems about singletons and sets of singletons which unfortunately are not included in the library.

theorem *singleton-intersection:*

assumes $A: \text{card } A = 1$
assumes $B: \text{card } B = 1$
assumes *noteq*: $A \neq B$
shows $A \cap B = \{\}$

using *assms* **by**(*auto simp: card-Suc-eq*)

theorem *card-singleton-set:*

assumes *finA*: *finite* A
assumes *cardOne*: $\forall x \in A. (\text{card } x = 1)$
shows $\text{card } (\bigcup A) = \text{card } A$

proof –

from *finA* **have** $\text{card } (\bigcup A) = (\sum x \in A. \text{card } x)$

proof(*rule card-Union-disjoint*)

from *cardOne* **show** $\forall A \in A. \text{finite } A$ **by** (*auto intro: card-ge-0-finite*)

next

show $\forall x \in A. \forall y \in A. x \neq y \longrightarrow x \cap y = \{\}$

proof(*clarify*)

fix $x y$

assume $x: x \in A$ **and** $y: y \in A$ **and** $x \neq y$

with *cardOne* **have** $\text{card } x = 1$ $\text{card } y = 1$ **by** *auto*

with $(x \neq y)$ **show** $x \cap y = \{\}$ **by** (*metis singleton-intersection*)

qed

qed

also from *cardOne* **have** $\dots = \text{card } A$ **by** *simp*

finally show *?thesis*.

qed

Intersecting Cosets are equal:

lemma (*in subgroup*) *repr-independence2:*

assumes *group*: *group* G

assumes $U: U \in \text{rcosets}_G H$

assumes $g: g \in U$

shows $U = H \#> g$

proof –
from U **obtain** h **where** $h:h \in \text{carrier } G \ U = H \ \#> h$ **unfolding** *RCOSETS-def*
by *auto*
with g **have** $g \in H \ \#> h$ **by** *simp*
with *group* h **show** $U = H \ \#> g$ **by** (*metis group.repr-independence is-subgroup*)
qed

locale *group-action* = *group* +
fixes $\varphi \ M$
assumes *grouphom:group-hom* $G \ (\text{BijGroup } M) \ \varphi$

context *group-action*
begin

lemma *is-group-action:group-action* $G \ \varphi \ M..$

The action of $\mathbf{1}$ has no effect:

lemma *one-is-id*:
assumes $m \in M$
shows $(\varphi \ \mathbf{1}) \ m = m$
proof –
from *grouphom* **have** $(\varphi \ \mathbf{1}) \ m = \mathbf{1}_{(\text{BijGroup } M)} \ m$ **by** (*metis group-hom.hom-one*)
also **have** $\dots = (\lambda x \in M. x) \ m$ **unfolding** *BijGroup-def* **by** (*metis monoid.select-conv(2)*)
also **from** *assms* **have** $\dots = m$ **by** *simp*
finally **show** *?thesis*.
qed

lemma *action-closed*:
assumes $m:m \in M$
assumes $g:g \in \text{carrier } G$
shows $\varphi \ g \ m \in M$
using *assms grouphom group-hom.hom-closed* **unfolding** *BijGroup-def Bij-def bij-betw-def*
by *fastforce*

lemma *img-in-bij*:
assumes $g \in \text{carrier } G$
shows $(\varphi \ g) \in \text{Bij } M$
using *assms grouphom* **unfolding** *BijGroup-def* **by** (*auto dest: group-hom.hom-closed*)

The action of *inv* g reverts the action of g

lemma *group-inv-rel*:
assumes $g:g \in \text{carrier } G$
assumes $mn:m \in M \ n \in M$
assumes $\text{phi}:(\varphi \ g) \ n = m$
shows $(\varphi \ (\text{inv } g)) \ m = n$
proof –
from g **have** $\text{bij}:(\varphi \ g) \in \text{Bij } M$ **unfolding** *BijGroup-def* **by** (*metis img-in-bij*)
with g *grouphom* **have** $\varphi \ (\text{inv } g) = \text{restrict } (\text{inv-into } M \ (\varphi \ g)) \ M$ **by**(*metis inv-BijGroup group-hom.hom-inv*)

hence $\varphi (inv\ g)\ m = (restrict\ (inv\ into\ M\ (\varphi\ g))\ M)\ m$ **by** *simp*
 also from *mn* **have** $\dots = (inv\ into\ M\ (\varphi\ g))\ m$ **by** (*metis restrict-def*)
 also from *g phi* **have** $\dots = (inv\ into\ M\ (\varphi\ g))\ ((\varphi\ g)\ n)$ **by** *simp*
 also from $\langle \varphi\ g \in Bij\ M \rangle$ *Bij-def* **have** *bij-betw* $(\varphi\ g)\ M\ M$ **by** *auto*
 hence *inj-on* $(\varphi\ g)\ M$ **by** (*metis bij-betw-imp-inj-on*)
 with *g mn* **have** $(inv\ into\ M\ (\varphi\ g))\ ((\varphi\ g)\ n) = n$ **by** (*metis inv-into-f-f*)
 finally **show** $\varphi (inv\ g)\ m = n$.
qed

lemma *images-are-bij*:

assumes $g:g \in carrier\ G$
 shows *bij-betw* $(\varphi\ g)\ M\ M$

proof –

from *g* **have** $bij:(\varphi\ g) \in Bij\ M$ **unfolding** *BijGroup-def* **by** (*metis img-in-bij*)
 with *Bij-def* **show** *bij-betw* $(\varphi\ g)\ M\ M$ **by** *auto*

qed

lemma *action-mult*:

assumes $g:g \in carrier\ G$
 assumes $h:h \in carrier\ G$
 assumes $m:m \in M$
 shows $(\varphi\ g)\ ((\varphi\ h)\ m) = (\varphi\ (g \otimes h))\ m$

proof –

from *g* **have** $\varphi g:(\varphi\ g) \in Bij\ M$ **unfolding** *BijGroup-def* **by** (*rule img-in-bij*)
 from *h* **have** $\varphi h:(\varphi\ h) \in Bij\ M$ **unfolding** *BijGroup-def* **by** (*rule img-in-bij*)
 from *h* **have** *bij-betw* $(\varphi\ h)\ M\ M$ **by** (*rule images-are-bij*)
 hence $(\varphi\ h)\ 'M = M$ **by** (*metis bij-betw-def*)
 with *m* **have** $hm:(\varphi\ h)\ m \in M$ **by** (*metis imageI*)
 from *group-hom* *g h* **have** $(\varphi\ (g \otimes h)) = ((\varphi\ g) \otimes_{(BijGroup\ M)} (\varphi\ h))$ **by** (*rule group-hom.hom-mult*)
 hence $\varphi (g \otimes h)\ m = ((\varphi\ g) \otimes_{(BijGroup\ M)} (\varphi\ h))\ m$ **by** *simp*
 also from $\varphi g\ \varphi h$ **have** $\dots = (compose\ M\ (\varphi\ g)\ (\varphi\ h))\ m$ **unfolding** *BijGroup-def*
by *simp*
 also from $\varphi g\ \varphi h\ hm$ **have** $\dots = (\varphi\ g)\ ((\varphi\ h)\ m)$ **by** (*metis compose-eq m*)
 finally **show** $(\varphi\ g)\ ((\varphi\ h)\ m) = (\varphi\ (g \otimes h))\ m..$

qed

1.2 The orbit relation

The following describes the relation containing the information whether two elements of M lie in the same orbit of the action

definition *same-orbit-rel*

where $same\ orbit\ rel = \{p \in M \times M. \exists g \in carrier\ G. (\varphi\ g)\ (snd\ p) = (fst\ p)\}$

Use the library about equivalence relations to define the set of orbits and the map assigning to each element of M its orbit

definition *orbits*

where $orbits = M // same\ orbit\ rel$

definition *orbit* :: 'c \Rightarrow 'c set
where *orbit* m = *same-orbit-rel* “ {m}

Next, we define a more easy-to-use characterization of an orbit.

lemma *orbit-char*:
assumes m:m \in M
shows *orbit* m = {n. \exists g. g \in carrier G \wedge (φ g) m = n}
using *assms unfolding orbit-def Image-def same-orbit-rel-def*
proof(*auto*)
fix x g
assume g:g \in carrier G **and** φ g x \in M x \in M
hence φ (inv g) (φ g x) = x **by** (*metis group-inv-rel*)
moreover from g **have** inv g \in carrier G **by** (*rule inv-closed*)
ultimately show \exists h. h \in carrier G \wedge φ h (φ g x) = x **by** *auto*
next
fix g
assume g:g \in carrier G
with m **show** φ g m \in M **by** (*metis action-closed*)
with m g **have** φ (inv g) (φ g m) = m **by** (*metis group-inv-rel*)
moreover from g **have** inv g \in carrier G **by** (*rule inv-closed*)
ultimately show \exists h \in carrier G. φ h (φ g m) = m **by** *auto*
qed

lemma *same-orbit-char*:
assumes m \in M n \in M
shows (m, n) \in *same-orbit-rel* = (\exists g \in carrier G. ((φ g) n = m))
unfolding *same-orbit-rel-def* **using** *assms* **by** *auto*

Now we show that the relation we’ve defined is, indeed, an equivalence relation:

lemma *same-orbit-is-equiv*:
shows *equiv* M *same-orbit-rel*
proof(*rule equivI*)
show *refl-on* M *same-orbit-rel*
proof(*rule refl-onI*)
show *same-orbit-rel* \subseteq M \times M **unfolding** *same-orbit-rel-def* **by** *auto*
next
fix m
assume m \in M
hence (φ 1) m = m **by**(*rule one-is-id*)
with (m \in M) **show** (m, m) \in *same-orbit-rel* **unfolding** *same-orbit-rel-def*
by (*auto simp:same-orbit-char*)
qed
next
show *sym* *same-orbit-rel*
proof(*rule symI*)
fix m n
assume mn:(m, n) \in *same-orbit-rel*

then obtain g where $g:g \in \text{carrier } G \ \varphi \ g \ n = m$ unfolding same-orbit-rel-def
by auto
hence $\text{inv}g:\text{inv } g \in \text{carrier } G$ by (metis inv-closed)
from mn have $(m, n) \in M \times M$ unfolding same-orbit-rel-def by simp
hence $mn2:m \in M \ n \in M$ by auto
from $g \ mn2$ have $\varphi (\text{inv } g) \ m = n$ by (metis group-inv-rel)
with $\text{inv}g \ mn2$ show $(n, m) \in \text{same-orbit-rel}$ unfolding same-orbit-rel-def by
auto
qed
next
show trans same-orbit-rel
proof(rule transI)
fix $x \ y \ z$
assume $xy:(x, y) \in \text{same-orbit-rel}$
then obtain g where $g:g \in \text{carrier } G$ and $\text{grel}:(\varphi \ g) \ y = x$ unfolding
same-orbit-rel-def by auto
assume $yz:(y, z) \in \text{same-orbit-rel}$
then obtain h where $h:h \in \text{carrier } G$ and $\text{hrel}:(\varphi \ h) \ z = y$ unfolding
same-orbit-rel-def by auto
from $g \ h$ have $gh:g \otimes h \in \text{carrier } G$ by simp
from $xy \ yz$ have $x \in M \ z \in M$ unfolding same-orbit-rel-def by auto
with $g \ h$ have $\varphi (g \otimes h) \ z = (\varphi \ g) ((\varphi \ h) \ z)$ by (metis action-mult)
also from $\text{hrel} \ \text{grel}$ have $\dots = x$ by simp
finally have $\varphi (g \otimes h) \ z = x$.
with $gh \ \langle x \in M \rangle \ \langle z \in M \rangle$ show $(x, z) \in \text{same-orbit-rel}$ unfolding same-orbit-rel-def
by auto
qed
qed

1.3 Stabilizer and fixed points

The following definition models the stabilizer of a group action:

definition stabilizer :: 'c \Rightarrow -
where stabilizer $m = \{g \in \text{carrier } G. (\varphi \ g) \ m = m\}$

This shows that the stabilizer of m is a subgroup of G .

lemma stabilizer-is-subgroup:

assumes $m:m \in M$
shows subgroup (stabilizer m) G
proof(rule subgroupI)
show stabilizer $m \subseteq \text{carrier } G$ unfolding stabilizer-def by auto
next
from m have $(\varphi \ 1) \ m = m$ by (rule one-is-id)
hence $1 \in \text{stabilizer } m$ unfolding stabilizer-def by simp
thus stabilizer $m \neq \{\}$ by auto
next
fix g
assume $g:g \in \text{stabilizer } m$
hence $g \in \text{carrier } G \ (\varphi \ g) \ m = m$ unfolding stabilizer-def by simp+

```

with  $m$  have  $ginv: (\varphi (inv\ g))\ m = m$  by (metis group-inv-rel)
from  $\langle g \in carrier\ G \rangle$  have  $inv\ g \in carrier\ G$  by (metis inv-closed)
with  $ginv$  show  $(inv\ g) \in stabilizer\ m$  unfolding stabilizer-def by simp
next
  fix  $g\ h$ 
  assume  $g: g \in stabilizer\ m$ 
  hence  $g^2: g \in carrier\ G$  unfolding stabilizer-def by simp
  assume  $h: h \in stabilizer\ m$ 
  hence  $h^2: h \in carrier\ G$  unfolding stabilizer-def by simp
  with  $g^2$  have  $gh: g \otimes h \in carrier\ G$  by (rule m-closed)
  from  $g^2\ h^2\ m$  have  $\varphi (g \otimes h)\ m = (\varphi\ g)\ ((\varphi\ h)\ m)$  by (metis action-mult)
  also from  $g\ h$  have  $\dots = m$  unfolding stabilizer-def by simp
  finally have  $\varphi (g \otimes h)\ m = m$ .
  with  $gh$  show  $g \otimes h \in stabilizer\ m$  unfolding stabilizer-def by simp
qed

```

Next, we define and characterize the fixed points of a group action.

```

definition fixed-points :: 'c set
  where fixed-points =  $\{m \in M. carrier\ G \subseteq stabilizer\ m\}$ 

```

```

lemma fixed-point-char:
  assumes  $m \in M$ 
  shows  $(m \in fixed-points) = (\forall g \in carrier\ G. \varphi\ g\ m = m)$ 
using assms unfolding fixed-points-def stabilizer-def by force

```

```

lemma orbit-contains-rep:
  assumes  $m: m \in M$ 
  shows  $m \in orbit\ m$ 
unfolding orbit-def using assms by (metis equiv-class-self same-orbit-is-equiv)

```

```

lemma singleton-orbit-eq-fixed-point:
  assumes  $m: m \in M$ 
  shows  $(card (orbit\ m) = 1) = (m \in fixed-points)$ 
proof
  assume  $card: card (orbit\ m) = 1$ 
  from  $m$  have  $m \in orbit\ m$  by (rule orbit-contains-rep)
  from  $m$  show  $m \in fixed-points$  unfolding fixed-points-def
  proof(auto)
    fix  $g$ 
    assume  $gG: g \in carrier\ G$ 
    with  $m$  have  $\varphi\ g\ m \in orbit\ m$  by (auto dest: orbit-char)
    with  $\langle m \in orbit\ m \rangle\ card$  have  $\varphi\ g\ m = m$  by (auto simp add: card-Suc-eq)
    with  $gG$  show  $g \in stabilizer\ m$  unfolding stabilizer-def by simp
  qed
next
  assume  $m \in fixed-points$ 
  hence  $fixed: carrier\ G \subseteq stabilizer\ m$  unfolding fixed-points-def by simp
  from  $m$  have  $orbit\ m = \{m\}$ 
  proof(auto simp add: orbit-contains-rep)

```

```

fix  $n$ 
assume  $n \in \text{orbit } m$ 
with  $m$  obtain  $g$  where  $g: g \in \text{carrier } G \ \varphi \ g \ m = n$  by (auto dest: orbit-char)
moreover with fixed have  $\varphi \ g \ m = m$  unfolding stabilizer-def by auto
ultimately show  $n = m$  by simp
qed
thus  $\text{card } (\text{orbit } m) = 1$  by simp
qed

```

1.4 The Orbit-Stabilizer Theorem

This section contains some theorems about orbits and their quotient groups. The first one is the well-known orbit-stabilizer theorem which establishes a bijection between the the quotient group of the an element's stabilizer and its orbit.

```

theorem orbit-thm:
assumes  $m: m \in M$ 
assumes  $\text{rep}: \bigwedge U. U \in (\text{carrier } (G \text{ Mod } (\text{stabilizer } m))) \implies \text{rep } U \in U$ 
shows bij-betw  $(\lambda H. (\varphi \ (\text{inv } (\text{rep } H)) \ m)) \ (\text{carrier } (G \text{ Mod } (\text{stabilizer } m))) \ (\text{orbit } m)$ 
proof (auto simp add: bij-betw-def)
show inj-on  $(\lambda H. \varphi \ (\text{inv } (\text{rep } H)) \ m) \ (\text{carrier } (G \text{ Mod } \text{stabilizer } m))$ 
proof (rule inj-onI)
fix  $U \ V$ 
assume  $U: U \in \text{carrier } (G \text{ Mod } (\text{stabilizer } m))$ 
assume  $V: V \in \text{carrier } (G \text{ Mod } (\text{stabilizer } m))$ 
define  $h$  where  $h = \text{rep } V$ 
define  $g$  where  $g = \text{rep } U$ 
have stabSubset:  $(\text{stabilizer } m) \subseteq \text{carrier } G$  unfolding stabilizer-def by auto
from  $m$  have stabSubgroup: subgroup  $(\text{stabilizer } m) \ G$  by (metis stabilizer-is-subgroup)
from  $V$  rep have  $hV: h \in V$  unfolding h-def by simp
from  $V$  stabSubset have  $V \subseteq \text{carrier } G$  unfolding FactGroup-def RCOSETS-def
r-coset-def by auto
with  $hV$  have  $hG: h \in \text{carrier } G$  by auto
hence  $h \text{inv } G: \text{inv } h \in \text{carrier } G$  by (metis inv-closed)
from  $U$  rep have  $gU: g \in U$  unfolding g-def by simp
from  $U$  stabSubset have  $U \subseteq \text{carrier } G$  unfolding FactGroup-def RCOSETS-def
r-coset-def by auto
with  $gU$  have  $gG: g \in \text{carrier } G$  by auto
hence  $g \text{inv } G: \text{inv } g \in \text{carrier } G$  by (metis inv-closed)
from  $gG$   $h \text{inv } G$  have  $g \text{inv } hG: g \otimes \text{inv } h \in \text{carrier } G$  by (metis m-closed)
assume  $\text{reps}: \varphi \ (\text{inv } \text{rep } U) \ m = \varphi \ (\text{inv } \text{rep } V) \ m$ 
hence  $gh: \varphi \ (\text{inv } g) \ m = \varphi \ (\text{inv } h) \ m$  unfolding g-def h-def.
from  $gG$   $h \text{inv } G$   $m$  have  $\varphi \ (g \otimes (\text{inv } h)) \ m = \varphi \ g \ (\varphi \ (\text{inv } h) \ m)$  by (metis
action-mult)
also from  $gh$   $g \text{inv } G$   $gG$   $m$  have  $\dots = \varphi \ (g \otimes \text{inv } g) \ m$  by (metis action-mult)
also from  $m$   $gG$  have  $\dots = m$  by (auto simp: one-is-id)
finally have  $\varphi \ (g \otimes \text{inv } h) \ m = m$ .

```


with $g \otimes \text{inv } h \in \text{stabilizer } m$
unfolding *stabilizer-def* **by** *simp*
hence $(\text{stabilizer } m) \#> (g \otimes \text{inv } h) = (\text{stabilizer } m) \#> \mathbf{1}$
by (*metis coset-join2 coset-mult-one m stabSubset stabilizer-is-subgroup subgroup.mem-carrier*)
with $\text{hinvg } hG \text{ } gG \text{ } \text{stabSubset}$ **have** $\text{stabgstabh}:(\text{stabilizer } m) \#> g = (\text{stabilizer } m) \#> h$
by (*metis coset-mult-inv1 group.coset-mult-one is-group*)
from *stabSubgroup is-group U gU* **have** $U = (\text{stabilizer } m) \#> g$
unfolding *FactGroup-def* **by** (*simp add:subgroup.repr-independence2*)
also from *stabgstabh is-group stabSubgroup V hV subgroup.repr-independence2*
have $\dots = V$
unfolding *FactGroup-def* **by** *force*
finally show $U = V$.
qed
next
have $\text{stabSubset}:\text{stabilizer } m \subseteq \text{carrier } G$ **unfolding** *stabilizer-def* **by** *auto*
fix H
assume $H:H \in \text{carrier } (G \text{ Mod } \text{stabilizer } m)$
with *rep* **have** $\text{rep } H \in H$ **by** *simp*
moreover with $H \text{ } \text{stabSubset}$ **have** $H \subseteq \text{carrier } G$ **unfolding** *FactGroup-def*
RCOSETS-def r-coset-def **by** *auto*
ultimately have $\text{rep } H \in \text{carrier } G$.
hence $\text{inv } \text{rep } H \in \text{carrier } G$ **by** (*rule inv-closed*)
with m **show** $\varphi (\text{inv } \text{rep } H) m \in \text{orbit } m$ **by** (*auto dest:orbit-char*)
next
fix n
assume $n \in \text{orbit } m$
with m **obtain** g **where** $g:g \in \text{carrier } G \text{ } \varphi g m = n$ **by** (*auto dest:orbit-char*)
hence $\text{invg}:\text{inv } g \in \text{carrier } G$ **by** *simp*
hence $\text{stabinvg}:(\text{stabilizer } m) \#> (\text{inv } g) \in \text{carrier } (G \text{ Mod } \text{stabilizer } m)$ **un-**
folding *FactGroup-def RCOSETS-def* **by** *auto*
hence $\text{rep } ((\text{stabilizer } m) \#> (\text{inv } g)) \in (\text{stabilizer } m) \#> (\text{inv } g)$ **by** (*metis*
rep)
then obtain h **where** $h:h \in \text{stabilizer } m \text{ } \text{rep } ((\text{stabilizer } m) \#> (\text{inv } g)) = h \otimes$
 $(\text{inv } g)$ **unfolding** *r-coset-def* **by** *auto*
with g **have** $\varphi (\text{inv } \text{rep } ((\text{stabilizer } m) \#> (\text{inv } g))) m = \varphi (\text{inv } (h \otimes (\text{inv } g)))$
 m **by** *simp*
also from h **have** $hG:h \in \text{carrier } G$ **unfolding** *stabilizer-def* **by** *simp*
with g **have** $\varphi (\text{inv } (h \otimes (\text{inv } g))) m = \varphi (g \otimes (\text{inv } h)) m$ **by** (*metis inv-closed*
inv-inv inv-mult-group)
also from $g \text{ } hG \text{ } m$ **have** $\dots = \varphi g (\varphi (\text{inv } h) m)$ **by** (*metis action-mult inv-closed*)
also from $h \text{ } m$ **have** $\text{inv } h \in \text{stabilizer } m$ **by** (*metis stabilizer-is-subgroup sub-*
group.m-inv-closed)
hence $\varphi g (\varphi (\text{inv } h) m) = \varphi g m$ **unfolding** *stabilizer-def* **by** *simp*
also from g **have** $\dots = n$ **by** *simp*
finally have $n = \varphi (\text{inv } \text{rep } ((\text{stabilizer } m) \#> (\text{inv } g))) m$.
with *stabinvg* **show** $n \in (\lambda H. \varphi (\text{inv } \text{rep } H) m) \text{ `carrier } (G \text{ Mod } \text{stabilizer } m)$
by *simp*

qed

In the case of G being finite, the last theorem can be reduced to a statement about the cardinality of orbit and stabilizer:

corollary *orbit-size:*

assumes $fin:finite$ ($carrier\ G$)

assumes $m:m \in M$

shows $order\ G = card\ (orbit\ m) * card\ (stabilizer\ m)$

proof –

define rep **where** $rep = (\lambda U \in (carrier\ (G\ Mod\ (stabilizer\ m))).\ SOME\ x.\ x \in U)$

have $\bigwedge U.\ U \in (carrier\ (G\ Mod\ (stabilizer\ m))) \implies rep\ U \in U$

proof –

fix U

assume $U:U \in carrier\ (G\ Mod\ stabilizer\ m)$

then obtain g **where** $g \in carrier\ G\ U = (stabilizer\ m) \#> g$ **unfolding** *FactGroup-def RCOSETS-def* **by** *auto*

with m **have** $(SOME\ x.\ x \in U) \in U$ **by** *(metis rcos-self stabilizer-is-subgroup someI-ex)*

with U **show** $rep\ U \in U$ **unfolding** *rep-def* **by** *simp*

qed

with m **have** $bij:card\ (carrier\ (G\ Mod\ (stabilizer\ m))) = card\ (orbit\ m)$ **by** *(metis bij-betw-same-card orbit-thm)*

from $fin\ m$ **have** $card\ (carrier\ (G\ Mod\ (stabilizer\ m))) * card\ (stabilizer\ m) = order\ G$ **unfolding** *FactGroup-def* **by** *(simp add: stabilizer-is-subgroup lagrange)*

with bij **show** *?thesis* **by** *simp*

qed

lemma *orbit-not-empty:*

assumes $fin:finite\ M$

assumes $A:A \in orbits$

shows $card\ A > 0$

proof –

from A **obtain** m **where** $m \in M\ A = orbit\ m$ **unfolding** *orbits-def quotient-def orbit-def* **by** *auto*

hence $m \in A$ **by** *(metis orbit-contains-rep)*

hence $A \neq \{\}$ **unfolding** *orbits-def* **by** *auto*

moreover from $fin\ A$ **have** $finite\ A$ **unfolding** *orbits-def quotient-def Image-def same-orbit-rel-def* **by** *auto*

ultimately show *?thesis* **by** *auto*

qed

lemma *fin-set-imp-fin-orbits:*

assumes $finM:finite\ M$

shows $finite\ orbits$

using *assms* **unfolding** *orbits-def quotient-def* **by** *simp*

lemma *singleton-orbits:*

```

shows  $\bigcup \{N \in \text{orbits}. \text{card } N = 1\} = \text{fixed-points}$ 
proof
show  $\bigcup \{N \in \text{orbits}. \text{card } N = 1\} \subseteq \text{fixed-points}$ 
proof
  fix  $x$ 
  assume  $a: x \in \bigcup \{N \in \text{orbits}. \text{card } N = 1\}$ 
  hence  $x \in M$  unfolding orbits-def quotient-def Image-def same-orbit-rel-def
by auto
  from  $a$  obtain  $N$  where  $N: N \in \text{orbits} \text{ card } N = 1 \ x \in N$  by auto
  then obtain  $y$  where  $\text{Norbit}: N = \text{orbit } y \ y \in M$  unfolding orbits-def
quotient-def orbit-def by auto
  hence  $y \in N$  by (metis orbit-contains-rep)
  with  $N$  have  $\text{Nsing}: N = \{x\} \ N = \{y\}$  by (auto simp: card-Suc-eq)
  hence  $x = y$  by simp
  with  $\text{Norbit}$  have  $\text{Norbit2}: N = \text{orbit } x$  by simp
  have  $\{g \in \text{carrier } G. \varphi \ g \ x = x\} = \text{carrier } G$ 
  proof(auto)
    fix  $g$ 
    assume  $g \in \text{carrier } G$ 
    with  $\langle x \in M \rangle$  have  $\varphi \ g \ x \in \text{orbit } x$  by (auto dest: orbit-char)
    with  $\text{Nsing}$  show  $\varphi \ g \ x = x$  by (metis Norbit2 singleton-iff)
  qed
  with  $\langle x \in M \rangle$  show  $x \in \text{fixed-points}$  unfolding fixed-points-def stabilizer-def
by simp
  qed
next
  show  $\text{fixed-points} \subseteq \bigcup \{N \in \text{orbits}. \text{card } N = 1\}$ 
  proof
    fix  $m$ 
    assume  $m: m \in \text{fixed-points}$ 
    hence  $mM: m \in M$  unfolding fixed-points-def by simp
    hence  $\text{orbit}: \text{orbit } m \in \text{orbits}$  unfolding orbits-def quotient-def orbit-def by
auto
    from  $mM \ m$  have  $\text{card } (\text{orbit } m) = 1$  by (metis singleton-orbit-eq-fixed-point)
    with  $\text{orbit}$  have  $\text{orbit } m \in \{N \in \text{orbits}. \text{card } N = 1\}$  by simp
    with  $mM$  show  $m \in \bigcup \{N \in \text{orbits}. \text{card } N = 1\}$  by (auto dest: orbit-contains-rep)
  qed
qed

```

If G is a p -group acting on a finite set, a given orbit is either a singleton or p divides its cardinality.

```

lemma p-dvd-orbit-size:
  assumes orderG: order G = p ^ a
  assumes prime: prime p
  assumes finM: finite M
  assumes Norbit: N ∈ orbits
  assumes card N > 1
  shows p dvd card N
proof –

```

```

from Norbit obtain m where m:m ∈ M N = orbit m unfolding orbits-def
quotient-def orbit-def by auto
from prime have 0 < p ^ a by (simp add: prime-gt-0-nat)
with orderG have finite (carrier G) unfolding order-def by (metis card-infinite
less-nat-zero-code)
with m have order G = card (orbit m) * card (stabilizer m) by (metis orbit-size)
with orderG m have p ^ a = card N * card (stabilizer m) by simp
with ⟨card N > 1⟩ show ?thesis
by (metis dvd-mult2 dvd-mult-cancel1 nat-dvd-not-less nat-mult-1 prime
prime-dvd-power-nat prime-factor-nat prime-nat-iff zero-less-one)
qed

```

As a result of the last lemma the only orbits that count modulo p are the fixed points

lemma *fixed-point-congruence*:

```

assumes order G = p ^ a
assumes prime p
assumes finM:finite M
shows card M mod p = card fixed-points mod p
proof –
define big-orbits where big-orbits = {N∈orbits. card N > 1}
from finM have orbit-part:orbits = big-orbits ∪ {N∈orbits. card N = 1} un-
folding big-orbits-def by (auto dest:orbit-not-empty)
have orbit-disj:big-orbits ∩ {N∈orbits. card N = 1} = {} unfolding big-orbits-def
by auto
from finM have orbits-fin:finite orbits by (rule fin-set-imp-fin-orbits)
hence fin-parts:finite big-orbits finite {N∈orbits. card N = 1} unfolding big-orbits-def
by simp+
from assms have  $\bigwedge N. N \in \text{big-orbits} \implies p \text{ dvd } \text{card } N$  unfolding big-orbits-def
by (auto simp: p-dvd-orbit-size)
hence orbit-div: $\bigwedge N. N \in \text{big-orbits} \implies \text{card } N = (\text{card } N \text{ div } p) * p$  by (metis
dvd-mult-div-cancel mult.commute)
have card M = card ( $\bigcup$  orbits) unfolding orbits-def by (metis Union-quotient
same-orbit-is-equiv)
also from orbits-fin have card ( $\bigcup$  orbits) = ( $\sum N \in \text{orbits. card } N$ ) unfolding
orbits-def
apply(rule card-Union-disjoint)
defer 1
apply(metis same-orbit-is-equiv quotient-disj)
using finM same-orbit-rel-def apply(auto dest:finite-equiv-class)
done
also from orbit-part orbit-disj fin-parts have ... = ( $\sum N \in \text{big-orbits. card } N$ ) +
( $\sum N \in \{N' \in \text{orbits. card } N' = 1\}. \text{card } N$ ) by (metis (lifting) sum.union-disjoint)
also from assms orbit-div fin-parts have ... = ( $\sum N \in \text{big-orbits. } (\text{card } N \text{ div } p) * p$ ) +
card ( $\bigcup \{N' \in \text{orbits. card } N' = 1\}$ ) by (auto simp: card-singleton-set)
also have ... = ( $\sum N \in \text{big-orbits. card } N \text{ div } p$ ) * p + card fixed-points using
singleton-orbits by (auto simp:sum-distrib-right)
finally have card M = ( $\sum N \in \text{big-orbits. card } N \text{ div } p$ ) * p + card fixed-points.
hence card M mod p = (( $\sum N \in \text{big-orbits. card } N \text{ div } p$ ) * p + card fixed-points)

```

```

mod p by simp
  also have ... = (card fixed-points) mod p by (metis mod-mult-self3)
  finally show ?thesis.
qed

```

We can restrict any group action to the action of a subgroup:

```

lemma subgroup-action:
  assumes H:subgroup H G
  shows group-action (G(\carrier := H))  $\varphi$  M
unfolding group-action-def group-action-axioms-def group-hom-def group-hom-axioms-def
hom-def
using assms
proof (auto simp add: is-group subgroup.subgroup-is-group group-BijGroup)
  fix x
  assume x  $\in$  H
  with H have x  $\in$  carrier G by (metis subgroup.mem-carrier)
  with group-hom show  $\varphi$  x  $\in$  carrier (BijGroup M) by (metis group-hom.hom-closed)
next
  fix x y
  assume x:x  $\in$  H and y:y  $\in$  H
  with H have x  $\in$  carrier G y  $\in$  carrier G by (metis subgroup.mem-carrier)+
  with group-hom show  $\varphi$  (x  $\otimes$  y) =  $\varphi$  x  $\otimes$ _{BijGroup M}  $\varphi$  y by (simp add:
group-hom.hom-mult)
qed

end

```

1.5 Some Examples for Group Actions

```

lemma (in group) right-mult-is-bij:
  assumes h:h  $\in$  carrier G
  shows ( $\lambda$ g  $\in$  carrier G. h  $\otimes$  g)  $\in$  Bij (carrier G)
proof (auto simp add: Bij-def bij-betw-def inj-on-def)
  fix x y
  assume x:x  $\in$  carrier G and y:y  $\in$  carrier G and h  $\otimes$  x = h  $\otimes$  y
  with h show x = y
  by simp
next
  fix x
  assume x:x  $\in$  carrier G
  with h show h  $\otimes$  x  $\in$  carrier G by (metis m-closed)
  from x h have inv h  $\otimes$  x  $\in$  carrier G by (metis m-closed inv-closed)
  moreover from x h have h  $\otimes$  (inv h  $\otimes$  x) = x by (metis inv-closed r-inv
m-assoc l-one)
  ultimately show x  $\in$  ( $\otimes$ ) h ' carrier G by force
qed

```

```

lemma (in group) right-mult-group-action:
  shows group-action G ( $\lambda$ h.  $\lambda$ g  $\in$  carrier G. h  $\otimes$  g) (carrier G)

```

unfolding *group-action-def group-action-axioms-def group-hom-def group-hom-axioms-def hom-def*
proof(*auto simp add:is-group group-BijGroup*)
fix *h*
assume $h \in \text{carrier } G$
thus $(\lambda g \in \text{carrier } G. h \otimes g) \in \text{carrier } (\text{BijGroup } (\text{carrier } G))$ **unfolding**
BijGroup-def **by** (*auto simp:right-mult-is-bij*)
next
fix *x y*
assume $x : x \in \text{carrier } G$ **and** $y : y \in \text{carrier } G$
define *multx multy*
where $\text{multx} = (\lambda g \in \text{carrier } G. x \otimes g)$
and $\text{multy} = (\lambda g \in \text{carrier } G. y \otimes g)$
with *x y* **have** $\text{multx} \in (\text{Bij } (\text{carrier } G))$ $\text{multy} \in (\text{Bij } (\text{carrier } G))$ **by** (*metis right-mult-is-bij*)
hence $\text{multx} \otimes_{\text{BijGroup } (\text{carrier } G)} \text{multy} = (\lambda g \in \text{carrier } G. \text{multx } (\text{multy } g))$
unfolding *BijGroup-def* **by** (*auto simp: compose-def*)
also have $\dots = (\lambda g \in \text{carrier } G. (x \otimes y) \otimes g)$ **unfolding** *multx-def multy-def*
proof(*rule restrict-ext*)
fix *g*
assume $g : g \in \text{carrier } G$
with *x y* **have** $x \otimes y \in \text{carrier } G$ $y \otimes g \in \text{carrier } G$ **by** *simp+*
with *x y g* **show** $(\lambda g \in \text{carrier } G. x \otimes g) ((\lambda g \in \text{carrier } G. y \otimes g) g) = x \otimes y$
 $\otimes g$ **by** (*auto simp:m-assoc*)
qed
finally show $(\lambda g \in \text{carrier } G. (x \otimes y) \otimes g) = (\lambda g \in \text{carrier } G. x \otimes g) \otimes_{\text{BijGroup } (\text{carrier } G)}$
 $(\lambda g \in \text{carrier } G. y \otimes g)$ **unfolding** *multx-def multy-def* **by** *simp*
qed

lemma (*in group*) *rcosets-closed*:
assumes *HG:subgroup H G*
assumes $g : g \in \text{carrier } G$
assumes *M:M ∈ rcosets H*
shows $M \#> g \in \text{rcosets } H$
proof –
from *M* **obtain** *h* **where** $h : h \in \text{carrier } G$ $M = H \#> h$ **unfolding** *RCOSETS-def*
by *auto*
with *g HG* **have** $M \#> g = H \#> (h \otimes g)$ **by** (*metis coset-mult-assoc subgroup.subset*)
with *HG g h* **show** $M \#> g \in \text{rcosets } H$ **by** (*metis rcosetsI subgroup.m-closed subgroup.subset subgroup-self*)
qed

lemma (*in group*) *inv-mult-on-rcosets-is-bij*:
assumes *HG:subgroup H G*
assumes $g : g \in \text{carrier } G$
shows $(\lambda U \in \text{rcosets } H. U \#> \text{inv } g) \in \text{Bij } (\text{rcosets } H)$
proof(*auto simp add:Bij-def bij-betw-def inj-on-def*)
fix *M*

```

assume  $M \in \text{rcosets } H$ 
with  $HG\ g$  show  $M \#> \text{inv } g \in \text{rcosets } H$  by (metis inv-closed rcosets-closed)
next
fix  $M$ 
assume  $M:M \in \text{rcosets } H$ 
with  $HG\ g$  have  $M \#> g \in \text{rcosets } H$  by (rule rcosets-closed)
moreover from  $M\ HG\ g$  have  $M \#> g \#> \text{inv } g = M$  by (metis coset-mult-assoc
coset-mult-inv2 inv-closed is-group subgroup.rcosets-carrier)
ultimately show  $M \in (\lambda U. U \#> \text{inv } g) \text{ ' } (\text{rcosets } H)$  by auto
next
fix  $M\ N\ x$ 
assume  $M:M \in \text{rcosets } H$  and  $N:N \in \text{rcosets } H$  and  $M \#> \text{inv } g = N \#>$ 
inv g
hence  $(M \#> \text{inv } g) \#> g = (N \#> \text{inv } g) \#> g$  by simp
with  $HG\ M\ N\ g$  have  $M \#> (\text{inv } g \otimes g) = N \#> (\text{inv } g \otimes g)$  by (metis
coset-mult-assoc is-group subgroup.m-inv-closed subgroup.rcosets-carrier subgroup-self)
with  $HG\ M\ N\ g$  have  $a1:M = N$  by (metis l-inv coset-mult-one is-group sub-
group.rcosets-carrier)
{
  assume  $x \in M$ 
  with  $a1$  show  $x \in N$  by simp
}
{
  assume  $x \in N$ 
  with  $a1$  show  $x \in M$  by simp
}
qed

```

lemma (*in group*) *inv-mult-on-rcosets-action*:

```

assumes  $HG:\text{subgroup } H\ G$ 
shows group-action  $G (\lambda g. \lambda U \in \text{rcosets } H. U \#> \text{inv } g) (\text{rcosets } H)$ 
unfolding group-action-def group-action-axioms-def group-hom-def group-hom-axioms-def
hom-def
proof(auto simp add:is-group group-BijGroup)
fix  $h$ 
assume  $h \in \text{carrier } G$ 
with  $HG$  show  $(\lambda U \in \text{rcosets } H. U \#> \text{inv } h) \in \text{carrier } (\text{BijGroup } (\text{rcosets } H))$ 
unfolding BijGroup-def by (auto simp:inv-mult-on-rcosets-is-bij)
next
fix  $x\ y$ 
assume  $x:x \in \text{carrier } G$  and  $y:y \in \text{carrier } G$ 
define  $\text{cos } x\ \text{cos } y$ 
where  $\text{cos } x = (\lambda U \in \text{rcosets } H. U \#> \text{inv } x)$ 
and  $\text{cos } y = (\lambda U \in \text{rcosets } H. U \#> \text{inv } y)$ 
with  $x\ y\ HG$  have  $\text{cos } x \in (\text{Bij } (\text{rcosets } H))$   $\text{cos } y \in (\text{Bij } (\text{rcosets } H))$ 
by (metis inv-mult-on-rcosets-is-bij)+
hence  $\text{cos } x \otimes \text{BijGroup } (\text{rcosets } H)\ \text{cos } y = (\lambda U \in \text{rcosets } H. \text{cos } x (\text{cos } y\ U))$ 

```

```

    unfolding BijGroup-def by (auto simp: compose-def)
  also have ... = (λU∈rcosets H. U #> inv (x ⊗ y)) unfolding cosx-def cosy-def
  proof (rule restrict-ext)
    fix U
    assume U:U ∈ rcosets H
    with HG y have U #> inv y ∈ rcosets H by (metis inv-closed rcosets-closed)
    with x y HG U have (λU∈rcosets H. U #> inv x) ((λU∈rcosets H. U #>
  inv y) U) = U #> inv y #> inv x
      by auto
    also from x y U HG have ... = U #> inv (x ⊗ y)
      by (metis inv-mult-group coset-mult-assoc inv-closed is-group subgroup.rcosets-carrier)
    finally show (λU∈rcosets H. U #> inv x) ((λU∈rcosets H. U #> inv y) U)
  = U #> inv (x ⊗ y).
  qed
  finally show (λU∈rcosets H. U #> inv (x ⊗ y)) = (λU∈rcosets H. U #> inv
x) ⊗BijGroup (rcosets H) (λU∈rcosets H. U #> inv y)
    unfolding cosx-def cosy-def by simp
  qed
end

```

```

theory SubgroupConjugation
imports GroupAction
begin

```

2 Conjugation of Subgroups and Cosets

This theory examines properties of the conjugation of subgroups of a fixed group as a group action

2.1 Definitions and Preliminaries

We define the set of all subgroups of G which have a certain cardinality. G will act on those sets. Afterwards some theorems which are already available for right cosets are dualized into statements about left cosets.

```

lemma (in subgroup) subgroup-of-subset:
  assumes G:group G
  assumes PH:H ⊆ K
  assumes KG:subgroup K G
  shows subgroup H (G⟨carrier := K⟩)
using assms subgroup-def group.m-inv-consistent m-inv-closed by fastforce

context group
begin

definition subgroups-of-size ::nat ⇒ -

```


where *subgroups-of-size* $p = \{H. \text{ subgroup } H \ G \wedge \text{ card } H = p\}$

lemma *lcosI*: $[[h \in H; H \subseteq \text{carrier } G; x \in \text{carrier } G]] \implies x \otimes h \in x <\# H$
by (*auto simp add: l-coset-def*)

lemma *lcoset-join2*:
assumes $H:\text{subgroup } H \ G$
assumes $g:g \in H$
shows $g <\# H = H$
proof *auto*
fix x
assume $x:x \in g <\# H$
then obtain h **where** $h:h \in H \ x = g \otimes h$ **unfolding** *l-coset-def* **by** *auto*
with $g \ H$ **show** $x \in H$ **by** (*metis subgroup.m-closed*)
next
fix x
assume $x:x \in H$
with $g \ H$ **have** $\text{inv } g \otimes x \in H$ **by** (*metis subgroup.m-closed subgroup.m-inv-closed*)
with $x \ g \ H$ **show** $x \in g <\# H$ **by** (*metis is-group subgroup.lcos-module-rev subgroup.mem-carrier*)
qed

lemma *cardeq-rcoset*:
assumes *finite* (*carrier* G)
assumes $M \subseteq \text{carrier } G$
assumes $g \in \text{carrier } G$
shows $\text{card } (M \ \#\!> \ g) = \text{card } M$
proof –
have $M \ \#\!> \ g \in \text{rcosets } M$ **by** (*metis assms(2) assms(3) rcosetsI*)
thus $\text{card } (M \ \#\!> \ g) = \text{card } M$
using *assms(2) card-rcosets-equal* **by** *auto*
qed

lemma *cardeq-lcoset*:
assumes *finite* (*carrier* G)
assumes $M:M \subseteq \text{carrier } G$
assumes $g:g \in \text{carrier } G$
shows $\text{card } (g <\# M) = \text{card } M$
proof –
have *bij-betw* $(\lambda m. g \otimes m) \ M \ (g <\# M)$
proof(*auto simp add: bij-betw-def*)
show *inj-on* $((\otimes) \ g) \ M$
proof(*rule inj-onI*)
from g **have** *inv* $g:\text{inv } g \in \text{carrier } G$ **by** (*rule inv-closed*)
fix $x \ y$
assume $x:x \in M$ **and** $y:y \in M$
with M **have** $xG:x \in \text{carrier } G$ **and** $yG:y \in \text{carrier } G$ **by** *auto*
assume $g \otimes x = g \otimes y$
hence $(\text{inv } g) \otimes (g \otimes x) = (\text{inv } g) \otimes (g \otimes y)$ **by** *simp*

```

      with g invg xG yG have (inv g ⊗ g) ⊗ x = (inv g ⊗ g) ⊗ y by (metis
m-assoc)
      with g invg xG yG show x = y by simp
    qed
  next
  fix x
  assume x ∈ M
  thus g ⊗ x ∈ g <# M unfolding l-coset-def by auto
next
fix x
assume x: x ∈ g <# M
then obtain m where x = g ⊗ m m ∈ M unfolding l-coset-def by auto
thus x ∈ (⊗) g ' M by simp
qed
thus card (g <# M) = card M by (metis bij-betw-same-card)
qed

```

2.2 Conjugation is a group action

We will now prove that conjugation acts on the subgroups of a certain group. A large part of this proof consists of showing that the conjugation of a subgroup with a group element is, again, a subgroup.

lemma *conjugation-subgroup*:

```

  assumes HG: subgroup H G
  assumes gG: g ∈ carrier G
  shows subgroup (g <# (H #> inv g)) G

```

proof

```

  from gG have inv g ∈ carrier G by (rule inv-closed)
  with HG have (H #> inv g) ⊆ carrier G by (metis r-coset-subset-G subgroup.subset)
  with gG show g <# (H #> inv g) ⊆ carrier G by (metis l-coset-subset-G)
next
  from gG have invgG: inv g ∈ carrier G by (metis inv-closed)
  with HG have lcosSubset: (H #> inv g) ⊆ carrier G by (metis r-coset-subset-G subgroup.subset)
  fix x y
  assume x: x ∈ g <# (H #> inv g) and y: y ∈ g <# (H #> inv g)
  then obtain x' y' where x': x' ∈ H #> inv g x = g ⊗ x' and y': y' ∈ H #>
inv g y = g ⊗ y' unfolding l-coset-def by auto
  then obtain hx hy where hx: hx ∈ H x' = hx ⊗ inv g and hy: hy ∈ H y' = hy
⊗ inv g unfolding r-coset-def by auto
  with x' y' have x2: x = g ⊗ (hx ⊗ inv g) and y2: y = g ⊗ (hy ⊗ inv g) by auto
  hence x ⊗ y = (g ⊗ (hx ⊗ inv g)) ⊗ (g ⊗ (hy ⊗ inv g)) by simp
  also from hx hy HG have hxG: hx ∈ carrier G and hyG: hy ∈ carrier G by
(metis subgroup.mem-carrier)+
  with gG hy x2 invgG have (g ⊗ (hx ⊗ inv g)) ⊗ (g ⊗ (hy ⊗ inv g)) = g ⊗ hx
⊗ (inv g ⊗ g) ⊗ hy ⊗ inv g by (metis m-assoc m-closed)
  also from invgG gG have ... = g ⊗ hx ⊗ 1 ⊗ hy ⊗ inv g by simp
  also from gG hxG have ... = g ⊗ hx ⊗ hy ⊗ inv g by (metis m-closed r-one)

```

also from gG hxG $invG$ **have** $\dots = g \otimes ((hx \otimes hy) \otimes inv\ g)$ **by** (*metis* gG hxG hyG $invG$ *m-assoc* *m-closed*)
finally have $xy:x \otimes y = g \otimes (hx \otimes hy \otimes inv\ g)$.
from hx hy HG **have** $hx \otimes hy \in H$ **by** (*metis* *subgroup.m-closed*)
with $invG$ HG **have** $(hx \otimes hy) \otimes inv\ g \in H \#> inv\ g$ **by** (*metis* *rcosI* *subgroup.subset*)
with gG $lcosSubset$ **have** $g \otimes (hx \otimes hy \otimes inv\ g) \in g <\# (H \#> inv\ g)$ **by** (*metis* *lcosI*)
with xy **show** $x \otimes y \in g <\# (H \#> inv\ g)$ **by** *simp*
next
from gG **have** $invG:inv\ g \in carrier\ G$ **by** (*metis* *inv-closed*)
with HG **have** $lcosSubset:(H \#> inv\ g) \subseteq carrier\ G$ **by** (*metis* *r-coset-subset-G* *subgroup.subset*)
from HG **have** $\mathbf{1} \in H$ **by** (*rule* *subgroup.one-closed*)
with $invG$ HG **have** $\mathbf{1} \otimes inv\ g \in H \#> inv\ g$ **by** (*metis* *rcosI* *subgroup.subset*)
with gG $lcosSubset$ **have** $g \otimes (\mathbf{1} \otimes inv\ g) \in g <\# (H \#> inv\ g)$ **by** (*metis* *lcosI*)
with gG $invG$ **show** $\mathbf{1} \in g <\# (H \#> inv\ g)$ **by** *simp*
next
from gG **have** $invG:inv\ g \in carrier\ G$ **by** (*metis* *inv-closed*)
with HG **have** $lcosSubset:(H \#> inv\ g) \subseteq carrier\ G$ **by** (*metis* *r-coset-subset-G* *subgroup.subset*)
fix x
assume $x \in g <\# (H \#> inv\ g)$
then obtain x' **where** $x':x' \in H \#> inv\ g$ $x = g \otimes x'$ **unfolding** *l-coset-def*
by *auto*
then obtain hx **where** $hx:hx \in H$ $x' = hx \otimes inv\ g$ **unfolding** *r-coset-def* **by** *auto*
with HG **have** $invhx:inv\ hx \in H$ **by** (*metis* *subgroup.m-inv-closed*)
from x' hx **have** $inv\ x = inv\ (g \otimes (hx \otimes inv\ g))$ **by** *simp*
also from x' hx HG gG $invG$ **have** $\dots = inv\ (inv\ g) \otimes inv\ hx \otimes inv\ g$ **by** (*metis* *calculation* *in-mono* *inv-mult-group* *lcosSubset* *subgroup.mem-carrier*)
also from gG **have** $\dots = g \otimes inv\ hx \otimes inv\ g$ **by** *simp*
also from gG $invG$ $invhx$ HG **have** $\dots = g \otimes (inv\ hx \otimes inv\ g)$ **by** (*metis* *m-assoc* *subgroup.mem-carrier*)
finally have $invx:inv\ x = g \otimes (inv\ hx \otimes inv\ g)$.
with $invhx$ $invG$ HG **have** $(inv\ hx) \otimes inv\ g \in H \#> inv\ g$ **by** (*metis* *rcosI* *subgroup.subset*)
with gG $lcosSubset$ **have** $g \otimes (inv\ hx \otimes inv\ g) \in g <\# (H \#> inv\ g)$ **by** (*metis* *lcosI*)
with $invx$ **show** $inv\ x \in g <\# (H \#> inv\ g)$ **by** *simp*
qed

definition *conjugation-action::nat* \Rightarrow -

where *conjugation-action* $p = (\lambda g \in carrier\ G. \lambda P \in subgroups-of-size\ p. g <\# (P \#> inv\ g))$

lemma *conjugation-is-size-invariant*:

assumes *fin:finite* (*carrier* G)

```

    assumes  $P:P \in \text{subgroups-of-size } p$ 
    assumes  $g:g \in \text{carrier } G$ 
    shows  $\text{conjugation-action } p \ g \ P \in \text{subgroups-of-size } p$ 
  proof -
    from  $g$  have  $\text{inv}g:\text{inv } g \in \text{carrier } G$  by (metis inv-closed)
    from  $P$  have  $PG:\text{subgroup } P \ G$  and  $\text{card}:\text{card } P = p$  unfolding subgroups-of-size-def
  by simp+
    hence  $P\text{sub}G:P \subseteq \text{carrier } G$  by (metis subgroup.subset)
    hence  $P\text{inv}g\text{sub}G:P \ \#\> \ \text{inv } g \subseteq \text{carrier } G$  by (metis inv}g \ r\text{-coset-subset-}G)
    have  $g <\# (P \ \#\> \ \text{inv } g) \in \text{subgroups-of-size } p$ 
    proof(auto simp add:subgroups-of-size-def)
      show  $\text{subgroup } (g <\# (P \ \#\> \ \text{inv } g)) \ G$  by (metis  $g \ PG$  conjugation-subgroup)
    next
      from  $\text{card } P\text{sub}G \ \text{fin } \text{inv}g$  have  $\text{card } (P \ \#\> \ \text{inv } g) = p$  by (metis cardeq-r coset)
      with  $g \ P\text{inv}g\text{sub}G \ \text{fin}$  show  $\text{card } (g <\# (P \ \#\> \ \text{inv } g)) = p$  by (metis cardeq-l coset)
    qed
    with  $P \ g$  show ?thesis unfolding conjugation-action-def by simp
  qed

```

```

lemma conjugation-is-Bij:
  assumes  $\text{fin}:\text{finite } (\text{carrier } G)$ 
  assumes  $g:g \in \text{carrier } G$ 
  shows  $\text{conjugation-action } p \ g \in \text{Bij } (\text{subgroups-of-size } p)$ 
  proof -
    from  $g$  have  $\text{inv}g:\text{inv } g \in \text{carrier } G$  by (rule inv-closed)
    from  $g$  have  $\text{conjugation-action } p \ g \in \text{extensional } (\text{subgroups-of-size } p)$  unfolding
    conjugation-action-def by simp
    moreover have  $\text{bij-betw } (\text{conjugation-action } p \ g) (\text{subgroups-of-size } p) (\text{subgroups-of-size } p)$ 
    proof(auto simp add:bij-betw-def)
      show  $\text{inj-on } (\text{conjugation-action } p \ g) (\text{subgroups-of-size } p)$ 
      proof(rule inj-onI)
        fix  $U \ V$ 
        assume  $U:U \in \text{subgroups-of-size } p$  and  $V:V \in \text{subgroups-of-size } p$ 
        hence  $\text{subset}G:U \subseteq \text{carrier } G \ V \subseteq \text{carrier } G$  unfolding subgroups-of-size-def
      by (metis (lifting) mem-Collect-eq subgroup.subset)+
        hence  $\text{subset}L:U \ \#\> \ \text{inv } g \subseteq \text{carrier } G \ V \ \#\> \ \text{inv } g \subseteq \text{carrier } G$  by (metis inv}g \ r\text{-coset-subset-}G)+
        assume  $\text{conjugation-action } p \ g \ U = \text{conjugation-action } p \ g \ V$ 
        with  $g \ U \ V$  have  $g <\# (U \ \#\> \ \text{inv } g) = g <\# (V \ \#\> \ \text{inv } g)$  unfolding
        conjugation-action-def by simp
        hence  $(\text{inv } g) <\# (g <\# (U \ \#\> \ \text{inv } g)) = (\text{inv } g) <\# (g <\# (V \ \#\> \ \text{inv } g))$  by simp
        hence  $(\text{inv } g \otimes g) <\# (U \ \#\> \ \text{inv } g) = (\text{inv } g \otimes g) <\# (V \ \#\> \ \text{inv } g)$  by
        (metis  $g \ \text{inv}g \ \text{lcos-m-assoc } r\text{-coset-subset-}G \ \text{subset}G$ )
        hence  $\mathbf{1} <\# (U \ \#\> \ \text{inv } g) = \mathbf{1} <\# (V \ \#\> \ \text{inv } g)$  by (metis  $g \ l\text{-inv}$ )
        hence  $U \ \#\> \ \text{inv } g = V \ \#\> \ \text{inv } g$  by (metis subsetL lcos-mult-one)
        hence  $(U \ \#\> \ \text{inv } g) \ \#\> \ g = (V \ \#\> \ \text{inv } g) \ \#\> \ g$  by simp
      qed
    qed
  qed

```

hence $U \#> (inv\ g \otimes g) = V \#> (inv\ g \otimes g)$ **by** (*metis coset-mult-assoc g inv-closed subsetG*)
hence $U \#> \mathbf{1} = V \#> \mathbf{1}$ **by** (*metis g l-inv*)
thus $U = V$ **by** (*metis coset-mult-one subsetG*)
qed
next
fix P
assume $P \in \text{subgroups-of-size } p$
thus *conjugation-action* $p\ g\ P \in \text{subgroups-of-size } p$ **by** (*metis fin g conjugation-is-size-invariant*)
next
fix P
assume $P:P \in \text{subgroups-of-size } p$
with *invg* **have** *conjugation-action* $p\ (inv\ g)\ P \in \text{subgroups-of-size } p$ **by** (*metis fin invg conjugation-is-size-invariant*)
with *invg* P **have** $(inv\ g) <\# (P \#> (inv\ (inv\ g))) \in \text{subgroups-of-size } p$
unfolding *conjugation-action-def* **by** *simp*
hence $1:(inv\ g) <\# (P \#> g) \in \text{subgroups-of-size } p$ **by** (*metis g inv-inv*)
have $g <\# (((inv\ g) <\# (P \#> g)) \#> inv\ g) = (\bigcup p \in P. \{g \otimes (inv\ g \otimes (p \otimes g) \otimes inv\ g)\})$ **unfolding** *r-coset-def l-coset-def* **by** (*simp add:m-assoc*)
also from P **have** $PG:P \subseteq \text{carrier } G$ **unfolding** *subgroups-of-size-def* **by** (*auto simp add:subgroup.subset*)
have $\forall p \in P. g \otimes (inv\ g \otimes (p \otimes g) \otimes inv\ g) = p$
proof(*auto*)
fix p
assume $p \in P$
with PG **have** $p:p \in \text{carrier } G..$
with $g\ invg$ **have** $g \otimes (inv\ g \otimes (p \otimes g) \otimes inv\ g) = (g \otimes inv\ g) \otimes p \otimes (g \otimes inv\ g)$ **by** (*metis m-assoc m-closed*)
also with $g\ invg\ g\ p$ **have** $\dots = p$ **by** (*metis l-one r-inv r-one*)
finally show $g \otimes (inv\ g \otimes (p \otimes g) \otimes inv\ g) = p.$
qed
hence $(\bigcup p \in P. \{g \otimes (inv\ g \otimes (p \otimes g) \otimes inv\ g)\}) = P$ **by** *simp*
finally have $g <\# (((inv\ g) <\# (P \#> g)) \#> inv\ g) = P.$
with 1 **have** $P \in (\lambda P. g <\# (P \#> inv\ g))$ ‘*subgroups-of-size p*’ **by** *auto*
with $P\ g$ **show** $P \in \text{conjugation-action } p\ g$ ‘*subgroups-of-size p*’ **unfolding** *conjugation-action-def* **by** *simp*
qed
ultimately show *?thesis* **unfolding** *BijGroup-def Bij-def* **by** *simp*
qed

lemma *lr-coset-assoc*:

assumes $g:g \in \text{carrier } G$
assumes $h:h \in \text{carrier } G$
assumes $P:P \subseteq \text{carrier } G$
shows $g <\# (P \#> h) = (g <\# P) \#> h$
proof(*auto*)
fix x
assume $x \in g <\# (P \#> h)$
then obtain p **where** $p \in P$ **and** $p:x = g \otimes (p \otimes h)$ **unfolding** *l-coset-def*

```

r-coset-def by auto
  with  $P$  have  $p \in \text{carrier } G$  by auto
  with  $g \ h \ p$  have  $x = (g \otimes p) \otimes h$  by (metis m-assoc)
  with  $\langle p \in P \rangle$  show  $x \in (g \langle \# P \rangle \#) h$  unfolding l-coset-def r-coset-def by
auto
next
  fix  $x$ 
  assume  $x \in (g \langle \# P \rangle \#) h$ 
  then obtain  $p$  where  $p \in P$  and  $p : x = (g \otimes p) \otimes h$  unfolding l-coset-def
r-coset-def by auto
  with  $P$  have  $p \in \text{carrier } G$  by auto
  with  $g \ h \ p$  have  $x = g \otimes (p \otimes h)$  by (metis m-assoc)
  with  $\langle p \in P \rangle$  show  $x \in g \langle \# (P \#) h \rangle$  unfolding l-coset-def r-coset-def by
auto
qed

theorem acts-on-subsets:
  assumes fin:finite (carrier  $G$ )
  shows group-action  $G$  (conjugation-action  $p$ ) (subgroups-of-size  $p$ )
unfolding group-action-def group-action-axioms-def group-hom-def group-hom-axioms-def
hom-def
apply(auto simp add:is-group group-BijGroup)
proof –
  fix  $g$ 
  assume  $g : g \in \text{carrier } G$ 
  with fin show conjugation-action  $p \ g \in \text{carrier } (BijGroup \ (\text{subgroups-of-size } p))$ 
  unfolding BijGroup-def by (metis conjugation-is-Bij partial-object.select-convs(1))
next
  fix  $x \ y$ 
  assume  $x : x \in \text{carrier } G$  and  $y : y \in \text{carrier } G$ 
  hence invx:inv  $x \in \text{carrier } G$  and invy:inv  $y \in \text{carrier } G$  by (metis inv-closed)+
  from  $x \ y$  have xyG: $x \otimes y \in \text{carrier } G$  by (metis m-closed)
  define conjx where conjx = conjugation-action  $p \ x$ 
  define conjy where conjy = conjugation-action  $p \ y$ 
  from fin  $x$  have xBij:conjx  $\in Bij$  (subgroups-of-size  $p$ ) unfolding conjx-def by
(metis conjugation-is-Bij)
  from fin  $y$  have yBij:conjy  $\in Bij$  (subgroups-of-size  $p$ ) unfolding conjy-def by
(metis conjugation-is-Bij)
  have conjx  $\otimes_{BijGroup \ (\text{subgroups-of-size } p)}$  conjy
  =  $(\lambda g \in Bij \ (\text{subgroups-of-size } p). \text{restrict } (\text{compose } (\text{subgroups-of-size } p) \ g) \ (Bij$ 
(subgroups-of-size  $p$ ))) conjx conjy unfolding BijGroup-def by simp
  also from xBij yBij have ... = compose (subgroups-of-size  $p$ ) conjx conjy by
simp
  also have ... =  $(\lambda P \in \text{subgroups-of-size } p. \text{conjx } (\text{conjy } P))$  by (metis compose-def)
  also have ... =  $(\lambda P \in \text{subgroups-of-size } p. \ x \otimes y \langle \# (P \#) \text{inv } (x \otimes y) \rangle)$ 
proof(rule restrict-ext)
  fix  $P$ 
  assume  $P : P \in \text{subgroups-of-size } p$ 
  hence  $PG : P \subseteq \text{carrier } G$  unfolding subgroups-of-size-def by (auto simp:subgroup.subset)

```

with y **have** $yPG:y <\# P \subseteq \text{carrier } G$ **by** (*metis l-coset-subset-G*)
from $x y$ **have** $\text{inv}xyG:\text{inv } (x \otimes y) \in \text{carrier } G$ **and** $xyG:x \otimes y \in \text{carrier } G$
using *inv-closed m-closed by auto*
from $yBij$ **have** $\text{conj}y$ ' *subgroups-of-size p = subgroups-of-size p* **unfolding**
Bij-def bij-betw-def **by** *simp*
with P **have** $\text{conj}yP:\text{conj}y P \in \text{subgroups-of-size } p$ **unfolding** *Bij-def bij-betw-def*
by (*metis (full-types) imageI*)
with $x y P$ **have** $\text{conj}x (\text{conj}y P) = x <\# ((y <\# (P \#> \text{inv } y)) \#> \text{inv } x)$
unfolding *conjy-def conjx-def conjugation-action-def* **by** *simp*
also from $y \text{inv}y PG$ **have** $\dots = x <\# (((y <\# P) \#> \text{inv } y) \#> \text{inv } x)$ **by**
(*metis lr-coset-assoc*)
also from $PG \text{inv}x \text{inv}y y$ **have** $\dots = x <\# ((y <\# P) \#> (\text{inv } y \otimes \text{inv } x))$
by (*metis coset-mult-assoc yPG*)
also from $x y$ **have** $\dots = x <\# ((y <\# P) \#> \text{inv } (x \otimes y))$ **by** (*metis*
inv-mult-group)
also from $\text{inv}xyG x yPG$ **have** $\dots = (x <\# (y <\# P)) \#> \text{inv } (x \otimes y)$ **by**
(*metis lr-coset-assoc*)
also from $x y PG$ **have** $\dots = ((x \otimes y) <\# P) \#> \text{inv } (x \otimes y)$ **by** (*metis*
lcos-m-assoc)
also from $xyG \text{inv}xyG PG$ **have** $\dots = (x \otimes y) <\# (P \#> \text{inv } (x \otimes y))$ **by**
(*metis lr-coset-assoc*)
finally show $\text{conj}x (\text{conj}y P) = x \otimes y <\# (P \#> \text{inv } (x \otimes y))$.
qed
finally have $\text{conj}x \otimes_{\text{BijGroup}} (\text{subgroups-of-size } p) \text{conj}y = (\lambda P \in \text{subgroups-of-size}$
 $p. x \otimes y <\# (P \#> \text{inv } (x \otimes y)))$.
with xyG **show** *conjugation-action p* $(x \otimes y)$
 $= \text{conjugation-action } p x \otimes_{\text{BijGroup}} (\text{subgroups-of-size } p) \text{conjugation-action } p y$
unfolding *conjx-def conjy-def conjugation-action-def* **by** *simp*
qed

2.3 Properties of the Conjugation Action

lemma *stabilizer-contains-P:*

assumes $\text{fin}:\text{finite } (\text{carrier } G)$

assumes $P:P \in \text{subgroups-of-size } p$

shows $P \subseteq \text{group-action.stabilizer } G (\text{conjugation-action } p) P$

proof

from P **have** $PG:\text{subgroup } P G$ **unfolding** *subgroups-of-size-def* **by** *simp*

from fin **interpret** $\text{conj}:\text{group-action } G (\text{conjugation-action } p) (\text{subgroups-of-size } p)$ **by** (*rule acts-on-subsets*)

fix x

assume $x:x \in P$

with PG **have** $\text{inv } x \in P$ **by** (*metis subgroup.m-inv-closed*)

from $x P$ **have** $xG:x \in \text{carrier } G$ **unfolding** *subgroups-of-size-def subgroup-def*
by *auto*

with P **have** $\text{conjugation-action } p x P = x <\# (P \#> \text{inv } x)$ **unfolding**
conjugation-action-def **by** *simp*

also from $(\text{inv } x \in P) PG$ **have** $\dots = x <\# P$ **by** (*metis coset-join2 subgroup.mem-carrier*)

also from PG x have $\dots = P$ by (rule *lcoset-join2*)
 finally have *conjugation-action* p x $P = P$.
 with xG show $x \in$ *group-action.stabilizer* G (*conjugation-action* p) P **unfolding**
conj.stabilizer-def by *simp*
 qed

corollary *stabilizer-supergroup-P*:
 assumes *fin:finite* (*carrier* G)
 assumes $P:P \in$ *subgroups-of-size* p
 shows *subgroup* P ($G(\backslash$ *carrier* := *group-action.stabilizer* G (*conjugation-action* p) P))
proof –
 from *assms* have $P \subseteq$ *group-action.stabilizer* G (*conjugation-action* p) P by
 (rule *stabilizer-contains-P*)
 moreover from P have *subgroup* P G **unfolding** *subgroups-of-size-def* by *simp*
 moreover from P *fin* have *subgroup* (*group-action.stabilizer* G (*conjugation-action* p) P) G by
 (*metis acts-on-subsets group-action.stabilizer-is-subgroup*)
 ultimately show *?thesis* by (*metis is-group subgroup.subgroup-of-subset*)
 qed

lemma (*in group*) *P-fixed-point-of-P-conj*:
 assumes *fin:finite* (*carrier* G)
 assumes $P:P \in$ *subgroups-of-size* p
 shows $P \in$ *group-action.fixed-points* ($G(\backslash$ *carrier* := P)) (*conjugation-action* p)
 (*subgroups-of-size* p)
proof –
 from *fin* **interpret** *conjG*: *group-action* G *conjugation-action* p *subgroups-of-size* p
 by (rule *acts-on-subsets*)
 from P have *subgroup* P G **unfolding** *subgroups-of-size-def* by *simp*
 with *fin* **interpret** *conjP*: *group-action* $G(\backslash$ *carrier* := P) (*conjugation-action* p)
 (*subgroups-of-size* p) by (*metis acts-on-subsets group-action.subgroup-action*)
 from *fin* P have $P \subseteq$ *conjG.stabilizer* P by (rule *stabilizer-contains-P*)
 hence $P \subseteq$ *conjP.stabilizer* P **using** *conjG.stabilizer-def* *conjP.stabilizer-def* by
auto
 with P show $P \in$ *conjP.fixed-points* **unfolding** *conjP.fixed-points-def* by *auto*
 qed

lemma *conj-wo-inv*:
 assumes QG :*subgroup* Q G
 assumes PG :*subgroup* P G
 assumes $g:g \in$ *carrier* G
 assumes *conj:inv* g $<\#$ (Q $\#>$ g) = P
 shows Q $\#>$ g = g $<\#$ P
proof –
 from g have *invg:inv* $g \in$ *carrier* G by (*metis inv-closed*)
 from *conj* have g $<\#$ (*inv* g $<\#$ (Q $\#>$ g)) = g $<\#$ P by *simp*
 with QG g *invg* have $(g \otimes$ *inv* $g)$ $<\#$ (Q $\#>$ g) = g $<\#$ P by (*metis*
lcos-m-assoc r-coset-subset-G subgroup.subset)
 with g *invg* have 1 $<\#$ (Q $\#>$ g) = g $<\#$ P by (*metis r-inv*)

with QG g **show** $Q \#> g = g <\# P$ **by** (*metis lcos-mult-one r-coset-subset-G subgroup.subset*)

qed

end

end

theory *SndSylow*
imports *SubgroupConjugation*
begin

no-notation *Multiset.subset-mset* (**infix** $<\#$ 50)

3 The Secondary Sylow Theorems

3.1 Preliminaries

lemma *singletonI*:

assumes $\bigwedge x. x \in A \implies x = y$

assumes $y \in A$

shows $A = \{y\}$

using *assms* **by** *fastforce*

context *group*

begin

lemma *set-mult-inclusion*:

assumes $H: \text{subgroup } H \ G$

assumes $Q: P \subseteq \text{carrier } G$

assumes $PQ: H <\#> P \subseteq H$

shows $P \subseteq H$

proof

fix x

from H **have** $1 \in H$ **by** (*rule subgroup.one-closed*)

moreover **assume** $x: x \in P$

ultimately **have** $1 \otimes x \in H <\#> P$ **unfolding** *set-mult-def* **by** *auto*

with PQ **have** $1 \otimes x \in H$ **by** *auto*

with $H \ Q \ x$ **show** $x \in H$ **by** (*metis in-mono l-one*)

qed

lemma *card-subgrp-dvd*:

assumes *subgroup* $H \ G$

shows *card* H *dvd* *order* G

proof(*cases finite (carrier G)*)

case *True*

with *assms* **have** *card* (*rcosets* H) * *card* H = *order* G **by** (*metis lagrange*)

thus *?thesis* **by** (*metis dvd-triv-left mult commute*)

```

next
  case False
  hence  $\text{order } G = 0$  unfolding order-def by (metis card-infinite)
  thus ?thesis by (metis dvd-0-right)
qed

```

```

lemma subgroup-finite:
  assumes subgroup:subgroup  $H G$ 
  assumes finite:finite (carrier  $G$ )
  shows finite  $H$ 
by (metis finite finite-subset subgroup subgroup.subset)

```

end

3.2 Extending the Sylow Locale

This locale extends the originale *syLOW* locale by adding the constraint that the p must not divide the remainder m , i.e. p^a is the maximal size of a p -subgroup of G .

```

locale snd-syLOW = syLOW +
  assumes pNotDvdM:  $\neg (p \text{ dvd } m)$ 

```

```

context snd-syLOW
begin

```

```

lemma pa-not-zero:  $p \wedge a \neq 0$ 
  by (simp add: prime-gt-0-nat prime-p)

```

```

lemma syLOW-greater-zero:
  shows card (subgroups-of-size ( $p \wedge a$ ))  $> 0$ 
proof –
  obtain  $P$  where PG:subgroup  $P G$  and cardP:card  $P = p \wedge a$  by (metis syLOW-thm)
  hence  $P \in \text{subgroups-of-size } (p \wedge a)$  unfolding subgroups-of-size-def by auto
  hence subgroups-of-size ( $p \wedge a$ )  $\neq \{\}$  by auto
  moreover from finite-G have finite (subgroups-of-size ( $p \wedge a$ )) unfolding
subgroups-of-size-def subgroup-def by auto
  ultimately show ?thesis by auto
qed

```

```

lemma is-snd-syLOW: snd-syLOW  $G p a m$  by (rule snd-syLOW-axioms)

```

3.3 Every p -group is Contained in a conjugate of a p -Sylow-Group

```

lemma ex-conj-syLOW-group:
  assumes  $H:H \in \text{subgroups-of-size } (p \wedge b)$ 
  assumes  $Psize:P \in \text{subgroups-of-size } (p \wedge a)$ 

```

obtains g where $g \in \text{carrier } G$ $H \subseteq g <\#> (P \#> \text{inv } g)$
proof –
from H have $H\text{sub}G:\text{subgroup } H \ G$ unfolding $\text{subgroups-of-size-def}$ by auto
hence $HG:H \subseteq \text{carrier } G$ unfolding $\text{subgroups-of-size-def}$ by $(\text{simp add:subgroup.subset})$
from $P\text{size}$ have $PG:\text{subgroup } P \ G$ and $\text{card}P:\text{card } P = p \wedge a$ unfolding
 $\text{subgroups-of-size-def}$ **by auto**
define H' where $H' = G(\text{carrier} := H)$
from $H\text{sub}G$ interpret $H\text{group}:\text{group } H'$ unfolding $H'\text{-def}$ by $(\text{metis subgroup-imp-group})$
from H have $\text{order}H':\text{order } H' = p \wedge b$ unfolding $H'\text{-def}$ $\text{subgroups-of-size-def}$
 order-def **by simp**
define φ where $\varphi = (\lambda g. \lambda U \in \text{rcosets } P. U \#> \text{inv } g)$
with PG interpret $G\text{act}:\text{group-action } G \ \varphi \ \text{rcosets } P$ unfolding $\varphi\text{-def}$ by
 $(\text{metis inv-mult-on-rcosets-action})$
from H interpret $H'\text{act}:\text{group-action } H' \ \varphi \ \text{rcosets } P$ unfolding $H'\text{-def}$ $\text{subgroups-of-size-def}$
by $(\text{metis (mono-tags) Gact.subgroup-action mem-Collect-eq})$
from $\text{finite-}G \ PG$ have $\text{finite } (\text{rcosets } P)$ unfolding RCOSETS-def r-coset-def
by $(\text{metis (lifting) finite.emptyI finite-UN-I finite-insert})$
**with $\text{order}H'$ sylow-axioms $\text{card}P$ have $\text{card } H'\text{act.fixed-points mod } p = \text{card}$
 $(\text{rcosets } P) \text{ mod } p$ unfolding sylow-def sylow-axioms-def by $(\text{metis } H'\text{act.fixed-point-congruence})$**
**moreover from $\text{finite-}G \ PG$ $\text{order-}G \ \text{card}P$ have $\text{card } (\text{rcosets } P) * p \wedge a =$
 $p \wedge a * m$ by (metis lagrange)**
**with $\text{prime-}p$ have $\text{card } (\text{rcosets } P) = m$ by $(\text{metis less-nat-zero-code mult-cancel2}$
 $\text{mult-is-0 mult.commute order-}G \ \text{zero-less-o-}G)$**
hence $\text{card } (\text{rcosets } P) \text{ mod } p = m \text{ mod } p$ by simp
moreover from $p \text{NotDvd}m$ $\text{prime-}p$ have $\dots \neq 0$ by $(\text{metis dvd-eq-mod-eq-0})$
ultimately have $\text{card } H'\text{act.fixed-points} \neq 0$ by (metis mod-0)
then obtain N where $N:N \in H'\text{act.fixed-points}$ by fastforce
hence $N\text{coset}:N \in \text{rcosets } P$ unfolding $H'\text{act.fixed-points-def}$ by simp
then obtain g where $g:g \in \text{carrier } G$ $N = P \#> g$ unfolding RCOSETS-def
by auto
hence $\text{inv}g:\text{inv } g \in \text{carrier } G$ by $(\text{metis inv-closed})$
hence $\text{invinv}g:\text{inv } (\text{inv } g) \in \text{carrier } G$ by $(\text{metis inv-closed})$
from N have $\text{carrier } H' \subseteq H'\text{act.stabilizer } N$ unfolding $H'\text{act.fixed-points-def}$
by simp
hence $\forall h \in H. \varphi h N = N$ unfolding $H'\text{act.stabilizer-def}$ using $H'\text{-def}$ by auto
with $HG \ N\text{coset}$ have $a1:\forall h \in H. N \#> \text{inv } h \subseteq N$ unfolding $\varphi\text{-def}$ by simp
have $N <\#> H \subseteq N$ unfolding set-mult-def r-coset-def
proof(auto)
fix $n \ h$
assume $n:n \in N$ and $h:h \in H$
**with H have $\text{inv } h \in H$ by $(\text{metis (mono-tags) mem-Collect-eq subgroup.m-inv-closed}$
 $\text{subgroups-of-size-def})$**
with $n \ HG \ PG \ a1$ have $n \otimes \text{inv } (\text{inv } h) \in N$ unfolding r-coset-def by auto
with $HG \ h$ show $n \otimes h \in N$ by $(\text{metis in-mono inv-inv})$
qed
with g have $((P \#> g) <\#> H) \#> \text{inv } g \subseteq (P \#> g) \#> \text{inv } g$ unfolding
 r-coset-def **by auto**
**with $PG \ g \ \text{inv}g$ have $((P \#> g) <\#> H) \#> \text{inv } g \subseteq P$ by $(\text{metis coset-mult-assoc}$
 $\text{coset-mult-one r-inv subgroup.subset})$**

with $g \in HG$ $PG \text{ inv } g$ **have** $P \langle \# \rangle (g \langle \# H \# \rangle \text{ inv } g) \subseteq P$ **by** (*metis lr-coset-assoc r-coset-subset-G rcos-assoc-lcos setmult-rcos-assoc subgroup.subset*)
with $PG \in HG$ $g \text{ inv } g$ **have** $g \langle \# H \# \rangle \text{ inv } g \subseteq P$ **by** (*metis l-coset-subset-G r-coset-subset-G set-mult-inclusion*)
with g **have** $(g \langle \# H \# \rangle \text{ inv } g) \# \rangle \text{ inv } (\text{inv } g) \subseteq P \# \rangle \text{ inv } (\text{inv } g)$ **unfolding** *r-coset-def* **by** *auto*
with $HG \in g \text{ inv } g$ $\text{inv } g$ **have** $g \langle \# H \subseteq P \# \rangle \text{ inv } (\text{inv } g)$ **by** (*metis coset-mult-assoc coset-mult-inv2 l-coset-subset-G*)
with g **have** $(\text{inv } g) \langle \# (g \langle \# H \rangle \subseteq \text{inv } g \langle \# (P \# \rangle \text{ inv } (\text{inv } g))$ **unfolding** *l-coset-def* **by** *auto*
with $HG \in g \text{ inv } g$ $\text{inv } g$ **have** $H \subseteq \text{inv } g \langle \# (P \# \rangle \text{ inv } (\text{inv } g))$ **by** (*metis inv-inv lcos-m-assoc lcos-mult-one r-inv*)
with $\text{inv } g$ **show** *thesis* **by** (*auto dest:that*)
qed

3.4 Every p -Group is Contained in a p -Sylow-Group

theorem *syLOW-contained-in-syLOW-group*:

assumes $H: H \in \text{subgroups-of-size } (p \wedge b)$

obtains S **where** $H \subseteq S$ **and** $S \in \text{subgroups-of-size } (p \wedge a)$

proof –

from H **have** $HG: H \subseteq \text{carrier } G$ **unfolding** *subgroups-of-size-def* **by** (*simp add:subgroup.subset*)

obtain P **where** $PG: \text{subgroup } P \subseteq G$ **and** $\text{card } P: \text{card } P = p \wedge a$ **by** (*metis syLOW-thm*)

hence $Psize: P \in \text{subgroups-of-size } (p \wedge a)$ **unfolding** *subgroups-of-size-def* **by** *simp*

with H **obtain** g **where** $g: g \in \text{carrier } G$ $H \subseteq g \langle \# (P \# \rangle \text{ inv } g)$ **by** (*metis ex-conj-syLOW-group*)

moreover **note** $Psize \ g$

moreover **with** *finite-G* **have** *conjugation-action* $(p \wedge a) \ g \ P \in \text{subgroups-of-size } (p \wedge a)$ **by** (*metis conjugation-is-size-invariant*)

ultimately **show** *thesis* **unfolding** *conjugation-action-def* **by** (*auto dest:that*)

qed

3.5 p -Sylow-Groups are conjugates of each other

theorem *syLOW-conjugate*:

assumes $P: P \in \text{subgroups-of-size } (p \wedge a)$

assumes $Q: Q \in \text{subgroups-of-size } (p \wedge a)$

obtains g **where** $g \in \text{carrier } G$ $Q = g \langle \# (P \# \rangle \text{ inv } g)$

proof –

from P **have** $\text{card } P = p \wedge a$ **unfolding** *subgroups-of-size-def* **by** *simp*

from Q **have** $Q\text{card}: \text{card } Q = p \wedge a$ **unfolding** *subgroups-of-size-def* **by** *simp*

from $Q \subseteq P$ **obtain** g **where** $g: g \in \text{carrier } G$ $Q \subseteq g \langle \# (P \# \rangle \text{ inv } g)$ **by** (*rule ex-conj-syLOW-group*)

moreover **with** *finite-G* **have** *conjugation-action* $(p \wedge a) \ g \ P \in \text{subgroups-of-size } (p \wedge a)$ **by** (*metis conjugation-is-size-invariant*)

moreover **from** $g \ P$ **have** *conjugation-action* $(p \wedge a) \ g \ P = g \langle \# (P \# \rangle \text{ inv } g)$ **unfolding** *conjugation-action-def* **by** *simp*

ultimately have $conjSize:g <\# (P \#> inv\ g) \in subgroups-of-size (p \wedge a)$
unfolding *conjugation-action-def* **by** *simp*
with $Qcard$ **have** $card:card (g <\# (P \#> inv\ g)) = card\ Q$ **unfolding**
subgroups-of-size-def **by** *simp*
from $conjSize$ *finite-G* **have** $finite (g <\# (P \#> inv\ g))$ **by** (*metis (mono-tags)*
finite-subset mem-Collect-eq subgroup.subset subgroups-of-size-def)
with $g\ card$ **have** $Q = g <\# (P \#> inv\ g)$ **by** (*metis card-subset-eq*)
with g **show** *thesis* **by** (*metis that*)
qed

corollary *syLOW-conj-orbit-rel*:

assumes $P:P \in subgroups-of-size (p \wedge a)$
assumes $Q:Q \in subgroups-of-size (p \wedge a)$
shows $(P,Q) \in group-action.same-orbit-rel\ G (conjugation-action (p \wedge a)) (subgroups-of-size (p \wedge a))$
unfolding *group-action.same-orbit-rel-def*
proof –
from $Q\ P$ **obtain** g **where** $g:g \in carrier\ G\ P = g <\# (Q \#> inv\ g)$ **by** (*rule syLOW-conjugate*)
with $Q\ P$ **have** $g':conjugation-action (p \wedge a)\ g\ Q = P$ **unfolding** *conjugation-action-def*
by *simp*
from *finite-G* **interpret** $conj: group-action\ G (conjugation-action (p \wedge a))$
(subgroups-of-size (p \wedge a)) **by** (*rule acts-on-subsets*)
have $conj.same-orbit-rel = \{X \in (subgroups-of-size (p \wedge a) \times subgroups-of-size (p \wedge a)). \exists g \in carrier\ G. ((conjugation-action (p \wedge a))\ g)\ (snd\ X) = (fst\ X)\}$ **by**
(rule conj.same-orbit-rel-def)
with $g\ g'\ P\ Q$ **show** *?thesis* **by** *auto*
qed

3.6 Counting SyLOW-Groups

The number of syLOW groups is the orbit size of one of them:

theorem *num-eq-card-orbit*:

assumes $P:P \in subgroups-of-size (p \wedge a)$
shows $subgroups-of-size (p \wedge a) = group-action.orbit\ G (conjugation-action (p \wedge a)) (subgroups-of-size (p \wedge a))\ P$
proof(*auto*)
from *finite-G* **interpret** $conj: group-action\ G (conjugation-action (p \wedge a))$
(subgroups-of-size (p \wedge a)) **by** (*rule acts-on-subsets*)
have $group-action.orbit\ G (conjugation-action (p \wedge a)) (subgroups-of-size (p \wedge a))\ P = group-action.same-orbit-rel\ G (conjugation-action (p \wedge a)) (subgroups-of-size (p \wedge a))\ \{P\}$ **by** (*rule conj.orbit-def*)
fix Q
{
assume $Q:Q \in subgroups-of-size (p \wedge a)$
from $P\ Q$ **obtain** g **where** $g:g \in carrier\ G\ Q = g <\# (P \#> inv\ g)$ **by**
(rule syLOW-conjugate)
with $P\ conj.orbit-char$ **show** $Q \in group-action.orbit\ G (conjugation-action (p \wedge a)) (subgroups-of-size (p \wedge a))\ P$

unfolding *conjugation-action-def* **by** *auto*
} **{**
assume $Q \in \text{group-action.orbit } G \text{ (conjugation-action } (p \wedge a)) \text{ (subgroups-of-size } (p \wedge a)) P$
with $P \text{ conj.orbit-char}$ **obtain** g **where** $g: g \in \text{carrier } G \ Q = \text{conjugation-action } (p \wedge a) \ g \ P$ **by** *auto*
with *finite-G* P **show** $Q \in \text{subgroups-of-size } (p \wedge a)$ **by** (*metis conjugation-is-size-invariant*)
}
qed

theorem *num-sylow-normalizer*:

assumes $Psize: P \in \text{subgroups-of-size } (p \wedge a)$
shows $\text{card } (rcosets \ G(\text{carrier} := \text{group-action.stabilizer } G \text{ (conjugation-action } (p \wedge a)) \ P)) \ P) * p \wedge a = \text{card } (\text{group-action.stabilizer } G \text{ (conjugation-action } (p \wedge a)) \ P)$

proof –

from *finite-G* **interpret** *conj*: $\text{group-action } G \text{ (conjugation-action } (p \wedge a)) \text{ (subgroups-of-size } (p \wedge a))$ **by** (*rule acts-on-subsets*)

from $Psize$ **have** $PG: \text{subgroup } P \ G$ **and** $\text{card}P: \text{card } P = p \wedge a$ **unfolding** *subgroups-of-size-def* **by** *auto*

with *finite-G* **have** $\text{order } G = \text{card } (\text{conj.orbit } P) * \text{card } (\text{conj.stabilizer } P)$ **by** (*metis Psize acts-on-subsets group-action.orbit-size*)

with *order-G* $Psize$ **have** $p \wedge a * m = \text{card } (\text{subgroups-of-size } (p \wedge a)) * \text{card } (\text{conj.stabilizer } P)$ **by** (*metis num-eq-card-orbit*)

moreover from $Psize$ **interpret** *stabGroup*: $\text{group } G(\text{carrier} := \text{conj.stabilizer } P)$ **by** (*metis conj.stabilizer-is-subgroup subgroup-imp-group*)

from *finite-G* $Psize$ **have** $PStab: \text{subgroup } P \ (G(\text{carrier} := \text{conj.stabilizer } P))$ **by** (*rule stabilizer-supergp-P*)

from *finite-G* $Psize$ **have** *finite* ($\text{conj.stabilizer } P$) **by** (*metis card-infinite conj.stabilizer-is-subgroup less-nat-zero-code subgroup.finite-imp-card-positive*)

with *finite-G* $PStab$ *stabGroup.lagrange* **have** $\text{card } (rcosets \ G(\text{carrier} := \text{conj.stabilizer } P)) \ P) * \text{card } P = \text{order } (G(\text{carrier} := \text{conj.stabilizer } P))$ **by** *force*

with $\text{card}P$ **show** *?thesis* **unfolding** *order-def* **by** *auto*

qed

theorem (*in snd-sylow*) *num-sylow-dvd-remainder*:

shows $\text{card } (\text{subgroups-of-size } (p \wedge a)) \ \text{dvd } m$

proof –

from *finite-G* **interpret** *conj*: $\text{group-action } G \text{ (conjugation-action } (p \wedge a)) \text{ (subgroups-of-size } (p \wedge a))$ **by** (*rule acts-on-subsets*)

obtain P **where** $PG: \text{subgroup } P \ G$ **and** $\text{card}P: \text{card } P = p \wedge a$ **by** (*metis sylow-thm*)

hence $Psize: P \in \text{subgroups-of-size } (p \wedge a)$ **unfolding** *subgroups-of-size-def* **by** *simp*

with *finite-G* **have** $\text{order } G = \text{card } (\text{conj.orbit } P) * \text{card } (\text{conj.stabilizer } P)$ **by** (*metis Psize acts-on-subsets group-action.orbit-size*)

with *order-G* $Psize$ **have** $\text{orderEq}: p \wedge a * m = \text{card } (\text{subgroups-of-size } (p \wedge a)) * \text{card } (\text{conj.stabilizer } P)$ **by** (*metis num-eq-card-orbit*)

define k **where** $k = \text{card } (rcosets \ G(\text{carrier} := \text{conj.stabilizer } P)) \ P)$

with $Psize$ **have** $k * p \wedge a = \text{card } (\text{conj.stabilizer } P)$ **by** (*metis num-sylow-normalizer*)

with *orderEq* **have** $p^a * m = \text{card}(\text{subgroups-of-size}(p^a)) * p^a * k$ **by**
(auto simp:mult.assoc mult.commute)
hence $p^a * m = p^a * \text{card}(\text{subgroups-of-size}(p^a)) * k$ **by** *auto*
then have $m = \text{card}(\text{subgroups-of-size}(p^a)) * k$
using *pa-not-zero* **by** *auto*
then show *?thesis ..*
qed

We can restrict this locale to refer to a subgroup of order at least p^a :

lemma (in *snd-sylow*) *restrict-locale*:

assumes *subgrp:subgroup* $P G$
assumes *card:p^a dvd card P*
shows *snd-sylow* ($G(\text{carrier} := P)$) $p a ((\text{card } P) \text{ div } (p^a))$
proof –
from *subgrp* **interpret** *groupP: group* $G(\text{carrier} := P)$ **by** (*metis subgroup-imp-group*)
define k **where** $k = (\text{card } P) \text{ div } (p^a)$
with *card* **have** *cardP:card P = p^a * k* **by** *auto*
hence *orderP:order (G(carrier := P)) = p^a * k* **unfolding** *order-def* **by**
simp
from *cardP subgrp order-G* **have** $p^a * k \text{ dvd } p^a * m$ **by** (*metis card-subgrp-dvd*)
hence $k \text{ dvd } m$
by (*metis nat-mult-dvd-cancel-disj pa-not-zero*)
with *pNotDvd m* **have** *ndvd:¬ p dvd k*
by (*blast intro: dvd-trans*)
define *PcalM* **where** $PcalM = \{s. s \subseteq \text{carrier } (G(\text{carrier} := P)) \wedge \text{card } s = p^a\}$
define *PRelM* **where** $PRelM = \{(N1, N2). N1 \in PcalM \wedge N2 \in PcalM \wedge (\exists g \in \text{carrier } (G(\text{carrier} := P)). N1 = N2 \#>_{G(\text{carrier} := P)} g)\}$
from *subgrp finite-G* **have** *finite-groupP:finite (carrier (G(carrier := P)))* **by**
(auto simp:subgroup-finite)
interpret *Nsylow: snd-sylow* $G(\text{carrier} := P)$ $p a k PcalM PRelM$
unfolding *snd-sylow-def snd-sylow-axioms-def sylow-def sylow-axioms-def k-def*
using *groupP.is-group prime-p orderP finite-groupP ndvd PcalM-def PRelM-def*
k-def **by** *fastforce+*
show *?thesis* **using** *k-def* **by** (*metis Nsylow.is-snd-sylow*)
qed

theorem (in *snd-sylow*) *p-sylow-mod-p*:

shows $\text{card}(\text{subgroups-of-size}(p^a)) \text{ mod } p = 1$

proof –

obtain P **where** *PG:subgroup* $P G$ **and** *cardP:card P = p^a* **by** (*metis sylow-thm*)
hence *orderP:order (G(carrier := P)) = p^a* **unfolding** *order-def* **by** *auto*
from *PG* **have** *PsubG:P ⊆ carrier G* **by** (*metis subgroup.subset*)
from *PG cardP* **have** *PSize:P ∈ subgroups-of-size (p^a)* **unfolding** *subgroups-of-size-def*
by *auto*
from *PG* **interpret** *groupP:group* $(G(\text{carrier} := P))$ **by** (*rule subgroup-imp-group*)
from *cardP* **have** *PSize2:P ∈ groupP.subgroups-of-size (p^a)* **using** *groupP.subgroups-of-size-def*
groupP.subgroup-self **by** *auto*

from *finite-G* **interpret** *conjG*: *group-action* G *conjugation-action* $(p \wedge a)$ *subgroups-of-size* $(p \wedge a)$ **by** (*rule acts-on-subsets*)
from *PG* **interpret** *conjP*: *group-action* $G(\text{carrier} := P)$ *conjugation-action* $(p \wedge a)$ *subgroups-of-size* $(p \wedge a)$ **by** (*rule conjG.subgroup-action*)
from *finite-G* **have** *finite* (*subgroups-of-size* $(p \wedge a)$) **unfolding** *subgroups-of-size-def* *subgroup-def* **by** *auto*
with *orderP* *prime-p* **have** *card* (*subgroups-of-size* $(p \wedge a)$) *mod* $p = \text{card } \textit{conjP.fixed-points} *mod* p **by** (*rule conjP.fixed-point-congruence*)
also **have** $\dots = 1$
proof –
have $\bigwedge Q. Q \in \textit{conjP.fixed-points} \implies Q = P$
proof –
fix Q
assume $Q\textit{fixed}: Q \in \textit{conjP.fixed-points}$
hence $Q\textit{size}: Q \in \textit{subgroups-of-size} (p \wedge a)$ **unfolding** *conjP.fixed-points-def*
by *simp*
hence *cardQ*: *card* $Q = p \wedge a$ **unfolding** *subgroups-of-size-def* **by** *simp*
– The normalizer of Q in G
– Let’s first show some basic properties of N
define N **where** $N = \textit{conjG.stabilizer } Q$
define k **where** $k = (\textit{card } N) \textit{ div } (p \wedge a)$
from *N-def* $Q\textit{size}$ **have** NG : *subgroup* N G **by** (*metis conjG.stabilizer-is-subgroup*)
then **interpret** *groupN*: *group* $G(\text{carrier} := N)$ **by** (*metis subgroup-imp-group*)
from $Q\textit{size}$ *N-def* **have** QN : *subgroup* Q $(G(\text{carrier} := N))$ **using** *stabilizer-supergrp-P*
by *auto*
– The following proposition is used to show that $P = Q$ later
from $Q\textit{size}$ **have** $N\textit{fixes}Q: \forall g \in N. \textit{conjugation-action} (p \wedge a) g Q = Q$
unfolding *N-def conjG.stabilizer-def* **by** *auto*
from $Q\textit{fixed}$ **have** $P\textit{fixes}Q: \forall g \in P. \textit{conjugation-action} (p \wedge a) g Q = Q$
unfolding *conjP.fixed-points-def conjP.stabilizer-def* **by** *auto*
with $P\textit{sub}G$ **have** $P \subseteq N$ **unfolding** *N-def conjG.stabilizer-def* **by** *auto*
with *PG* *N-def* $Q\textit{size}$ **have** PN : *subgroup* P $(G(\text{carrier} := N))$ **by** (*metis conjG.stabilizer-is-subgroup is-group subgroup.subgroup-of-subset*)
with *cardP* **have** $p \wedge a \textit{ dvd order} (G(\text{carrier} := N))$ **using** *groupN.card-subgrp-dvd*
by *force*
hence $p \wedge a \textit{ dvd card } N$ **unfolding** *order-def* **by** *simp*
with NG **have** $\textit{smaller-sylow}: \textit{snd-sylow} (G(\text{carrier} := N)) p a k$ **unfolding** *k-def* **by** (*rule restrict-locale*)
– Instantiate the *snd-sylow* Locale a second time for the normalizer of Q
define $N\textit{cal}M$ **where** $N\textit{cal}M = \{s. s \subseteq \textit{carrier} (G(\text{carrier} := N)) \wedge \textit{card } s = p \wedge a\}$
define $N\textit{rel}M$ **where** $N\textit{rel}M = \{(N1, N2). N1 \in N\textit{cal}M \wedge N2 \in N\textit{cal}M \wedge (\exists g \in \textit{carrier} (G(\text{carrier} := N)). N1 = N2 \#>_{G(\text{carrier} := N)} g)\}$
interpret $N\textit{sy}low$: *snd-sylow* $G(\text{carrier} := N)$ $p a k$ $N\textit{cal}M$ $N\textit{rel}M$
unfolding *NcalM-def NrelM-def* **using** *smaller-sylow* .
– P and Q are conjugate in N :
from *cardP* PN **have** $P\textit{size}N: P \in \textit{groupN.subgroups-of-size} (p \wedge a)$ **unfolding** *groupN.subgroups-of-size-def* **by** *auto*
from *cardQ* QN **have** $Q\textit{size}N: Q \in \textit{groupN.subgroups-of-size} (p \wedge a)$ **unfold-**$


```

ing groupN.subgroups-of-size-def by auto
  from QsizeN PsizeN obtain g where g:g ∈ carrier (G(|carrier := N)) P
= g <# G(|carrier := N) (Q #> G(|carrier := N) inv G(|carrier := N) g) by (rule
NsyLOW.sylow-conjugate)
  with NG have P = g <# (Q #> inv g) unfolding r-coset-def l-coset-def
by (auto simp:m-inv-consistent)
  with NG g Qsize have conjugation-action (p ^ a) g Q = P unfolding
conjugation-action-def using subgroup.subset by force
  with g NfixesQ show Q = P by auto
  qed
moreover from finite-G PSize have P ∈ conjP.fixed-points using P-fixed-point-of-P-conj
by auto
  ultimately have conjP.fixed-points = {P} by fastforce
  hence one:card conjP.fixed-points = 1 by (auto simp: card-Suc-eq)
  with prime-p have card conjP.fixed-points < p unfolding prime-nat-iff by
auto
  with one show ?thesis using mod-pos-pos-trivial by auto
  qed
finally show ?thesis.
qed

end

end

```