Abstract

These theories extend the existent proof of the first sylow theorem (written by Florian Kammueller 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
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 $\text{noteq } A \neq B$
shows $A \cap B = \{\}$
using assms by (auto simp: card-Suc-eq)

**Theorem card-singleton-set:**

assumes $\text{cardOne } \forall x \in A. (\text{card } x = 1)$
shows $\text{card } (\bigcup A) = \text{card } A$
proof -
  have $\text{card } (\bigcup A) = (\sum x \in A. \text{card } x)$
proof (rule card-Union-disjoint)
  from $\text{cardOne } \text{show } \bigwedge a. a \in A \implies \text{finite } a$ by (auto intro: card-ge-0-finite)
next
  show $\text{pairwise disjnt } A$
  unfolding pairwise-def disjnt-def
proof (clarify)
  fix $x y$
  assume $x : x \in A \text{ and } y : y \in A \text{ and } x \neq y$
  with $\text{cardOne } \text{have } \text{card } x = 1 \text{ card } y = 1$ by auto
  with $x \neq y$ show $x \cap y = \{\}$ by (metis singleton-intersection)
qed
qed
also from $\text{cardOne } \text{have } ... = \text{card } A$ by simp
finally show "$\text{thesis}$."
qed

Intersecting Cosets are equal:

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

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

locale group-action = group +
  fixes ϕ M
  assumes grouphom:group-hom G (BijGroup M) ϕ

context group-action
begin

lemma is-group-action:group-action G ϕ M ..

The action of 1 has no effect:

lemma one-is-id:
  assumes m ∈ M
  shows (ϕ 1) m = m

proof –
  from grouphom have (ϕ 1) m = 1 (BijGroup M) m by (metis group-hom.hom-one)
  also have ... = (λx∈M. x) m unfolding BijGroup-def by (metis monoid.select-convs(2))
  also from assms have ... = m by simp
  finally show ?thesis.
  qed

lemma action-closed:
  assumes m:m ∈ M
  assumes g:g ∈ carrier G
  shows ϕ g m ∈ M
  using assms grouphom group-hom.hom-closed unfolding BijGroup-def Bij-def bij-betw-def
  by fastforce

lemma img-in-bij:
  assumes g ∈ carrier G
  shows (ϕ g) ∈ 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 ∈ carrier G
  assumes mn:m ∈ M n ∈ M
  assumes phi:(ϕ g) n = m
  shows (ϕ (inv g)) m = n

proof –
  from g have bij:(ϕ g) ∈ Bij M unfolding BijGroup-def by (metis img-in-bij)
  with g grouphom have ϕ (inv g) = restrict (inv-into M (ϕ g)) M by (metis inv-BijGroup group-hom.hom-inv)
hence \( \varphi (\text{inv g}) \ m = (\text{restrict} \ (\text{inv-into M} \ (\varphi g)) \ M) \) \( \mbox{by simp} \)
also from \( mn \) have \( \ldots = (\text{inv-into M} \ (\varphi g)) \ m \) \( \mbox{by (metis restrict-def)} \)
also from \( g \ phi \) have \( \ldots = (\text{inv-into M} \ (\varphi g)) \ ((\varphi g) \ n) \) \( \mbox{by simp} \)
also from \( \varphi g \in \text{Bij M} \) \text{Bij-def} have \( \text{bij-betw} \ (\varphi g) \ M \) \( \mbox{by auto} \)
\hence \( \text{inj-on} \ (\varphi g) \) \( M \) \( \mbox{by (metis bij-betw-imp-inj-on)} \)
with \( g \ mn \) have \( (\text{inv-into M} \ (\varphi g)) \ ((\varphi g) \ n) = n \) \( \mbox{by (metis inv-into-f-f)} \)
finally show \( \varphi (\text{inv g}) \ m = n. \)
\( \mbox{qed} \)

\textbf{lemma images-are-bij:}
assumes \( g : g \in \text{carrier G} \)
shows \( \text{bij-betw} \ (\varphi g) \ M \ M \)
proof –
from \( g \) have \( \text{bij} : (\varphi g) \in \text{Bij M} \)
unfolding \( \text{BijGroup-def} \)
by (\text{metis img-in-bij})
with \( \text{Bij-def} \) show \( \text{bij-betw} \ (\varphi g) \ M \ M \)
by auto
\( \mbox{qed} \)

\textbf{lemma action-mult:}
assumes \( g : g \in \text{carrier G} \)
assumes \( h : h \in \text{carrier G} \)
assumes \( m : m \in M \)
sows \( (\varphi g) ((\varphi h) m) = (\varphi (g \otimes h)) \)
proof –
from \( g \) have \( \varphi g : (\varphi g) \in \text{Bij M} \)
unfolding \( \text{BijGroup-def} \)
by (\text{rule img-in-bij})
from \( h \) have \( \varphi h : (\varphi h) \in \text{Bij M} \)
unfolding \( \text{BijGroup-def} \)
by (\text{rule img-in-bij})
from \( h \) have \( \text{bij-betw} \ (\varphi h) M \) \( \mbox{by (rule images-are-bij)} \)
\hence \( (\varphi h) \ M = M \) \( \mbox{by (metis bij-betw-def)} \)
with \( m \) have \( \text{hm} : (\varphi h) \) \( m \in M \)
by (\text{metis imageI})
from \( \text{grouphom g h have} \ (\varphi (g \otimes h)) = ((\varphi g) \otimes (\text{BijGroup M}) \ (\varphi h)) \)
by (\text{rule group-hom.hom-mult})
\hence \( \varphi (g \otimes h) m = ((\varphi g) \otimes (\text{BijGroup M}) \ (\varphi h)) \)
by simp
also from \( \varphi g \) \( \varphi h \) have \( \ldots = (\text{compose} M \ (\varphi g) \ (\varphi h)) \)
unfolding \( \text{BijGroup-def} \)
by simp
finally show \( (\varphi g) ((\varphi h) m) = (\varphi (g \otimes h)) \)
.. 
\( \mbox{qed} \)

\textbf{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

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

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

\textbf{definition orbits}
where \( \text{orbits} = M // \text{same-orbit-rel} \)
definition orbit :: 'c ⇒ '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 ∈ M
  shows orbit m = {n. ∃ g. g ∈ carrier G ∧ (ϕ g) m = n}
using assms unfolding orbit-def Image-def same-orbit-rel-def
proof(auto)
  fix x g
  assume g:g ∈ carrier G and φ g x ∈ M x ∈ M
  hence φ (inv g) (φ g x) = x by (metis group-inv-rel)
  moreover from g have inv g ∈ carrier G by (rule inv-closed)
  ultimately show ∃ h. h ∈ carrier G ∧ φ h (φ g x) = x by auto
next
  fix g
  assume g:g ∈ carrier G
  with m show φ g m ∈ 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 ∈ carrier G by (rule inv-closed)
  ultimately show ∃ h∈carrier G. φ h (φ g m) = m by auto
qed

lemma same-orbit-char:
  assumes m ∈ M n ∈ M
  shows (m, n) ∈ same-orbit-rel = (∃ g ∈ carrier G. ((φ g) n = m))
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 ⊆ M × M unfolding same-orbit-rel-def by auto
next
  fix m
  assume m ∈ M
  hence (φ 1) m = m by(rule one-is-id)
  with (m ∈ M) show (m, m) ∈ 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) ∈ same-orbit-rel
then obtain \( g \) where \( g \in \text{carrier } G \) \( g \cdot n = m \) unfolding same-orbit-rel-def by auto

hence \( \text{invg} \cdot \text{invg} \in \text{carrier } G \) by (metis inv-closed)

from \( mm \) have \( (m, n) \in M \times M \) unfolding same-orbit-rel-def by simp

hence \( \text{invg} \cdot \text{invg} \in \text{carrier } G \) by auto

with \( \text{invg} \cdot \text{invg} \) 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 \( \text{xy} \cdot \text{xy} \in \text{same-orbit-rel} \)

then obtain \( g \) where \( g \in \text{carrier } G \) and \( \text{grel} \cdot (\varphi \cdot g) \cdot y = x \) unfolding same-orbit-rel-def by auto

assume \( \text{yz} \cdot \text{yz} \in \text{same-orbit-rel} \)

then obtain \( h \) where \( h \in \text{carrier } G \) and \( \text{hrel} \cdot (\varphi \cdot h) \cdot z = y \) unfolding same-orbit-rel-def by auto

from \( g \cdot h \) have \( \text{gh} \cdot \text{gh} \in \text{carrier } G \) by simp

from \( \text{xy} \cdot \text{yz} \) have \( x \in M \) \( z \in M \) unfolding same-orbit-rel-def by auto

with \( \text{gh} \cdot \text{gh} \) 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 \( m \) = \{ \( g \in \text{carrier } G \) . \( \varphi \cdot g \cdot m = m \) \}

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 \( \varphi \cdot 1 \cdot m = m \) 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 \( g \in \text{carrier } G \) \( \varphi \cdot g \cdot m = m \) unfolding stabilizer-def by simp+
with \( m \) \textbf{have} \( \text{ginv}(\varphi(\text{inv } g)) \) \( m = m \) by (metis group-inv-rel)

\textbf{from} \( \langle g \in \text{carrier } G \rangle \) \textbf{have} \( \text{inv } g \in \text{carrier } G \) by (metis inv-closed)

\textbf{with} \( \text{ginv} \) \textbf{show} \( \text{inv } g \in \text{stabilizer } m \) \textbf{unfolding} \( \text{stabilizer-def} \) by simp

\textbf{next}

fix \( g \ h \)

\textbf{assume} \( g : g \in \text{stabilizer } m \)

\textbf{hence} \( g^2 : g \in \text{carrier } G \) \textbf{unfolding} \( \text{stabilizer-def} \) by simp

\textbf{assume} \( h : h \in \text{stabilizer } m \)

\textbf{hence} \( h^2 : h \in \text{carrier } G \) \textbf{unfolding} \( \text{stabilizer-def} \) by simp

\textbf{with} \( g^2 \) \textbf{have} \( gh : g \otimes h \in \text{carrier } G \) by (rule m-closed)

\textbf{from} \( g^2 \ h^2 \ m \) \textbf{have} \( \varphi(g \otimes h) \ m = (\varphi g) ((\varphi h) \ m) \) by (metis action-mult)

\textbf{also from} \( g \ h \) \textbf{have} \( \ldots = m \) \textbf{unfolding} \( \text{stabilizer-def} \) by simp

\textbf{finally have} \( \varphi(g \otimes h) \ m = m \).

\textbf{with} \( gh \) \textbf{show} \( g \otimes h \in \text{stabilizer } m \) \textbf{unfolding} \( \text{stabilizer-def} \) by simp

\textbf{qed}

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

\textbf{definition} \( \text{fixed-points} :: \ 'c \ \text{set} \)

\textbf{where} \( \text{fixed-points} = \{ m \in M. \ \text{carrier } G \subseteq \text{stabilizer } m \} \)

\textbf{lemma} \( \text{fixed-point-char} : \)

\textbf{assumes} \( m \in M \)

\textbf{shows} \( (m \in \text{fixed-points}) = (\forall g \in \text{carrier } G. \ \varphi g \ m = m) \)

\textbf{using} \( \text{assms} \) \textbf{unfolding} \( \text{fixed-points-def} \) \textbf{stabilizer-def} by force

\textbf{lemma} \( \text{orbit-contains-rep} : \)

\textbf{assumes} \( m : m \in M \)

\textbf{shows} \( m \in \text{orbit } m \)

\textbf{unfolding} \( \text{orbit-def} \) \textbf{using} \( \text{assms} \) by (metis equiv-class-self same-orbit-is-equiv)

\textbf{lemma} \( \text{singleton-orbit-eq-fixed-point} : \)

\textbf{assumes} \( m : m \in M \)

\textbf{shows} \( (\text{card } (\text{orbit } m) = 1) = (m \in \text{fixed-points}) \)

\textbf{proof}

\textbf{assume} \( \text{card} : \text{card } (\text{orbit } m) = 1 \)

\textbf{from} \( m \) \textbf{have} \( m \in \text{orbit } m \) by (rule orbit-contains-rep)

\textbf{from} \( m \) \textbf{show} \( m \in \text{fixed-points} \) \textbf{unfolding} \( \text{fixed-points-def} \)

\textbf{proof(auto)}

\textbf{fix} \( g \)

\textbf{assume} \( g G : g \in \text{carrier } G \)

\textbf{with} \( m \) \textbf{have} \( \varphi g m \in \text{orbit } m \) by (auto dest:orbit-char)

\textbf{with} \( m \in \text{orbit } m \) \textbf{card} \textbf{have} \( \varphi g m = m \) by (auto simp add: card-Suc-eq)

\textbf{with} \( g G \) \textbf{show} \( g \in \text{stabilizer } m \) \textbf{unfolding} \( \text{stabilizer-def} \) by simp

\textbf{qed}

\textbf{next}

\textbf{assume} \( m \in \text{fixed-points} \)

\textbf{hence} \( \text{fixed} : \text{carrier } G \subseteq \text{stabilizer } m \) \textbf{unfolding} \( \text{fixed-points-def} \) by simp

\textbf{from} \( m \) \textbf{have} \( \text{orbit } m = \{m\} \)

\textbf{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 \( \text{(auto dest: orbit-char)} \)

moreover with fixed have \( \varphi g m = m \) unfolding stabilizer-def by auto

ultimately show \( n = m \) by simp

qed

thus \( \text{card (orbit } m) = 1 \) 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**:

\[
\begin{align*}
\text{assumes } & m : m \in M \\
\text{assumes } & \text{rep } \bigwedge U : U \in (\text{carrier } (G \text{ Mod } (\text{stabilizer } m))) \implies \text{rep } U \in U \\
\text{shows } & \text{bij-betw } (\lambda H. (\varphi (\text{inv } (\text{rep } H))) m) (\text{carrier } (G \text{ Mod } (\text{stabilizer } m))) (\text{orbit } m)
\end{align*}
\]

**Proof** (auto simp add bij-betw-def)

show inj-on \((\lambda H. \varphi (\text{inv } (\text{rep } H))) m) (\text{carrier } (G \text{ Mod 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 stabSub: \((\text{stabilizer } m) \subseteq \text{carrier } G \) unfolding stabilizer-def by auto

from \( m \) have stabSubgroup: subgroup \((\text{stabilizer } m) G \) by \( \text{(metis stabilizer-is-subgroup)} \)

from \( V \) rep have \( hV : h \in V \) unfolding h-def by simp

from \( V \) stabSub 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 hinvg:inv h \in \text{carrier } G by (metis inv-closed)

from \( U \) rep have \( gU : g \in U \) unfolding g-def by simp

from \( U \) stabSub 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 ginvg:inv g \in \text{carrier } G by (metis inv-closed)

from \( gG \) hinvG have ginvhG: \( g \otimes \text{inv } h \in \text{carrier } G \) by (metis m-closed)

assume 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 \( gG \) hinvG m have \( \varphi (g \otimes (\text{inv } h)) m = \varphi g (\varphi (\text{inv } h) m) \) by (metis action-mult)

also from \( gh \) ginvg \( gG \) m have \( ... = \varphi (g \otimes \text{inv } g) m \) by (metis action-mult)

also from \( m \) gG have \( ... = m \) by (auto simp: one-is-id)

finally have \( \varphi (g \otimes \text{inv } h) m = m. \)
with \( ginv G \) have \((g \otimes inv h) \in stabilizer m\)
unfolding stabilizer-def by simp
hence \((stabilizer m) \#> (g \otimes inv h) = (stabilizer m) \#> 1\)
by (metis coset-join2 coset-mult-one m stabSubset stabilizer-is-subgroup subgroup.mem-carrier)
with \( inv g h G \) stabSubsets have stabgstabh:((stabilizer m) \#> g = (stabilizer m) \#> h)
by (metis coset-mult-inv1 group.coset-mult-one is-group)
from stabSubset is-group U gU have \( U = (stabilizer m) \#> g\)
unfolding FactGroup-def by (simp add:subgroup.repr-independence2)
also from stabgstabh is-group stabSubgroup V hV subgroup have \( ... = V\)
unfolding FactGroup-def by force
finally show \( U = V\).
qed
next
have stabSubset:stabilizer m \( \subseteq \) carrier G unfolding stabilizer-def by auto
fix \( H\)
assume \( H: H \in carrier (G \ Mod \ stabilizer m)\)
with rep have \( rep H \in H\) by simp
moreover with \( H\) stabSubset have \( H \subseteq carrier G\) unfolding FactGroup-def
RCOSETS-def r-coset-def by auto
ultimately have \( rep H \in carrier G..\)
hence \( inv rep H \in carrier G\) by (rule inv-closed)
with \( m\) show \( \varphi \ (inv rep H) \in orbit m\) by (auto dest:orbit-char)
next
fix \( n\)
assume \( n \in orbit m\)
with \( m\) obtain \( g\) where \( g: g \in carrier G\) \( \varphi \ g \in m = n\) by (auto dest:orbit-char)
hence \( meg: \ (inv g) \in carrier (G \ Mod \ stabilizer m)\)
unfolding FactGroup-def RCOSETS-def by auto
hence \( inv rep ((stabilizer m) \#> (inv g)) \in (stabilizer m) \#> (inv g)\)
by (metis rep)
then obtain \( h\) where \( h: h \in stabilizer m\) rep \((stabilizer m) \#> (inv g)) = h \otimes (inv g)\)
unfolding r-coset-def by auto
with \( g\) have \( \varphi \ (inv rep ((stabilizer m) \#> (inv g))) = \varphi \ (inv \ h \otimes (inv g))\)
by simp
also from \( h\) have \( hG: h \in carrier G\)
unfolding stabilizer-def by simp
with \( g\) have \( \varphi \ (inv \ (h \otimes (inv g))) = \varphi \ g \ (inv \ h)\)
by (metis inv-closed)
also from \( g\) hG m have \( \varphi \ (inv rep ((stabilizer m) \#> (inv g))) \in stabilizer m\)
by (metis stabilizer-is-subgroup subgroup.m-inv-closed)
unfolding stabilizer-def by simp
also from \( g\) have \( \varphi \ (inv \ h) \in stabilizer m\)
by (metis stabilizer-is-subgroup subgroup.m-inv-closed)
unfolding stabilizer-def by simp
finally have \( n = \varphi \ (inv rep ((stabilizer m) \#> (inv g))) \in m..\)
with stabinv show \( n \in (\lambda H. \varphi \ (inv rep H)) m \) carrier \((G \ Mod \ stabilizer m)\)
by simp
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 \( \text{fin:finite (carrier } G \) \)
assumes \( m : m \in M \)
shows \( \text{order } G = \text{card } (\text{orbit } m) \ast \text{card } (\text{stabilizer } m) \)
proof –
define rep where \( \text{rep } = (\lambda U \in (\text{carrier } (G \text{ Mod (stabilizer } m)))). \text{SOME } x. x \in U \)
have \( \bigwedge U. U \in (\text{carrier } (G \text{ Mod (stabilizer } m))) \implies \text{rep } U \in U \)
proof –
fix \( U \)
assume \( U : U \in \text{carrier } (G \text{ Mod stabilizer } m) \)
then obtain \( g \) where \( g \in \text{carrier } G \quad U = (\text{stabilizer } m) \not\triangleright g \quad \text{unfolding FactGroup-def RCOSETS-def by auto} \)
with \( m \) have \( (\text{SOME } x. x \in U) \in U \) by \( (\text{metis rcos-self stabilizer-is-subgroup somel-ex}) \)
with \( U \) show \( \text{rep } U \in U \) unfolding rep-def by simp
qed
with \( m \) have \( \text{bij:card } (\text{carrier } (G \text{ Mod (stabilizer } m))) = \text{card } (\text{orbit } m) \) by \( (\text{metis bij-betw-same-card orbit-thm}) \)
from \( \text{fin } m \) have \( \text{card } (\text{carrier } (G \text{ Mod (stabilizer } m))) \ast \text{card } (\text{stabilizer } m) = \text{order } G \) unfolding FactGroup-def by \( (\text{simp add: stabilizer-is-subgroup lagrange}) \)
with \( \text{bij} \) show \( ?\text{thesis} \) by simp
qed

lemma orbit-not-empty:
assumes \( \text{fin:finite } M \)
assumes \( A : A \in \text{orbits} \)
shows \( \text{card } A > 0 \)
proof –
from \( A \) obtain \( m \) where \( m \in M \ A = \text{orbit } m \) unfolding orbits-def quotient-def orbit-def by auto
hence \( m \in A \) by \( (\text{metis orbit-contains-rep}) \)

hence \( A \not= \{ \} \) unfolding orbits-def by auto
moreover from \( \text{fin } A \) have \( \text{finite } A \) unfolding orbits-def quotient-def Image-def same-orbit-rel-def by auto
ultimately show \( ?\text{thesis} \) by auto
qed

lemma fin-set-imp-fin-orbits:
assumes \( \text{finM:finite } M \)
shows \( \text{finite } \text{orbits} \)
using \( \text{assms unfolding orbits-def quotient-def by simp} \)

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 $N : N \in \text{orbits} \land \text{card } N = 1$
x by auto
then obtain $y$ where $N = \{ x \} \land y \in M$
unfolding orbits-def quotient-def orbit-def
by auto
hence $y \in N$x by (metis orbit-contains-rep)
with $N$ have $N\text{sing}: N = \{ x \}$ by (auto simp: card-Suc-eq)
hence $x = y$x by simp
with Norbit have Norbit2: $N = \{ x \}$ 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 $x \in M$ have $\varphi g x \in \text{orbit } x$x by (auto dest: orbit-char)
with $N\text{sing}$ show $\varphi g x = x$x by (metis Norbit2 singleton-iff)
qed
with $N \in \text{orbits}$ 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 $\text{orbit}\{ m \} : m \in \text{orbits}$ unfolding orbits-def quotient-def orbit-def by auto
from $mM$ have $\text{card } \{ m \} = 1$x by (metis singleton-orbit-eq-fixed-point)
with $\text{orbit}$ have $m \in \bigcup \{ N \in \text{orbits. } \text{card } N = 1 \}$ 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 $p\text{-dvd-orbit-size}$:
assumes orderG: $\text{order } G = p^a$
assumes prime: prime $p$
assumes finM: finite $M$
assumes Norbit: $N \in \text{orbits}$
assumes card $N > 1$
shows $p \text{ dvd } \text{card } N$
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-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\} unfolding 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 \(\bigwedge N. N ∈ \text{big-orbits} \implies p \mid \text{card N}\) unfolding big-orbits-def by (auto simp: p-dvd-orbit-size)
hence orbit-div:\(\bigwedge N. N ∈ \text{big-orbits} \implies \text{card N} = (\text{card N} \div p) ∗ p\) by (metis dvd-mult-div-cancel mult.commute)
have card M = card (⨆ orbits) unfolding orbits-def by (metis Union-quotient same-orbit-is-equiv)
also have \(\text{card (⨆ orbits) = (∑ N ∈ \text{orbits. card N = 1})}\) unfolding orbits-def
proof (rule card-Union-disjoint)
  show pairwise disjoint (M // same-orbit-rel)
  unfolding pairwise-def disjoint-def by (metis same-orbit-is-equiv quotient-disj)
  show \(\bigwedge A. A ∈ M // \text{same-orbit-rel} \implies \text{finite \text{A}}\)
  using finM same-orbit-rel-def by (auto dest:finite-equiv-class)
qed
also from orbit-part orbit-disj fin-parts have ... = (∑ N ∈ big-orbits. card N) + (∑ N ∈ \{N’ ∈ \text{orbits. card N’ = 1}\}. card N) by (metis (lifting) sum.union-disjoint)
also from assms orbit-div fin-parts have ... = (∑ N ∈ big-orbits. (card N div p) ∗ p) + card (⨆ \{N’ ∈ \text{orbits. card N’ = 1}\}) by (auto simp: card-singleton-set)
also have ... = (∑ N ∈ big-orbits. card N div p) ∗ p + card fixed-points using singleton-orbits by (auto simp:sum-distrib-right)
finally have card M = (∑ N ∈ big-orbits. card N div p) ∗ p + card fixed-points.
hence card M mod p = ((∑ N ∈ big-orbits. card N 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]) ϕ 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 ∈ H
  with H have x ∈ carrier G by (metis subgroup.mem-carrier)
  with grouphom show ϕ x ∈ carrier (BijGroup M) by (metis group-hom.hom-closed)
next
  fix x y
  assume x:x ∈ H and y:y ∈ H
  with H have x ∈ carrier G y ∈ carrier G by (metis subgroup.mem-carrier)
  with grouphom show ϕ (x ⊗ y) = ϕ x ⊗ BijGroup M ϕ 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 ∈ carrier G
  shows (λg ∈ carrier G. h ⊗ g) ∈ Bij (carrier G)
proof (auto simp add: Bij-def bij-betw-def inj-on-def)
  fix x y
  assume x:x ∈ carrier G and y:y ∈ carrier G and h ⊗ x = h ⊗ y
  with h show x = y
  by simp
next
  fix x
  assume x:x ∈ carrier G
  with h show h ⊗ x ∈ carrier G by (metis m-closed)
  from x h have inv h ⊗ x ∈ carrier G by (metis m-closed inv-closed)
  moreover from x h have h ⊗ (inv h ⊗ x) = x by (metis inv-closed r-inv m-assoc l-one)
  ultimately show x ∈ (⊗) h ' carrier G by force
qed

lemma (in group) right-mult-group-action:
  shows group-action G (λh. λg ∈ carrier G. h ⊗ g) (carrier G)
unfolding group-action-def group-action-axioms-def group-hom-def group-hom-axioms-def

proof(auto simp add:is-group group-BijGroup)
  fix h
  assume h ∈ carrier G
  thus (λg ∈ carrier G. h ⊗ g) ∈ carrier (BijGroup (carrier G)) unfolding BijGroup-def by (auto simp:right-mult-is-bij)
next
  fix x y
  assume x: x ∈ carrier G and y: y ∈ carrier G
  define multx multy
  where multx = (λg ∈ carrier G. x ⊗ g)
  and multy = (λg ∈ carrier G. y ⊗ g)
  with x y have multx ∈ (Bij (carrier G)) multy ∈ (Bij (carrier G)) by (metis right-mult-is-bij)+
  hence multx ⊗ BijGroup (carrier G) multy = (λg ∈ carrier G. multx (multy g)) unfolding BijGroup-def by (auto simp:compose-def)
also have ... = (λg ∈ carrier G. (x ⊗ y) ⊗ g) unfolding multx-def multy-def
proof(rule restrict-ext)
  fix g
  assume g: g ∈ carrier G
  with x y have x ⊗ y ∈ carrier G y ⊗ g ∈ carrier G by simp+
  with x y g show (λg ∈ carrier G. x ⊗ g) ((λg ∈ carrier G. y ⊗ g) g) = x ⊗ y ⊗ g by (auto simp:m-assoc)
qed

lemma (in group) rcosets-closed:
  assumes HG: subgroup H G
  assumes g: g ∈ carrier G
  assumes M: M ∈ rcosets H
  shows M #> g ∈ rcosets H
proof −
  from M obtain h where h: h ∈ carrier G M = H #> h unfolding RCOSETS-def
  by auto
  with HG g have M #> g = H #> (h ⊗ g) by (metis coset-mult-assoc subgroup.subset)
  with HG g h show M #> g ∈ 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 ∈ carrier G
  shows (λU ∈ rcosets H. U #> inv g) ∈ Bij (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 $M \#> inv g \in \text{rcosets } H$ 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 $M \#> g \in \text{rcosets } H$ by (metis coset-mult-assoc rcosets-closed)
ultimately show $M \in (\lambda U. U \#> inv g) \cdot (\text{rcosets } H)$ by auto
next
  fix $M N x$
  assume $M : M \in \text{rcosets } H$ and $N : N \in \text{rcosets } H$ and $M \#> inv g \equiv N \#> inv g$
  hence $(M \#> inv g) \#> g = (N \#> inv g) \#> g$ by simp
  with $HG M N g$ have $(M \#> (inv g \otimes g)) = (N \#> (inv g \otimes g))$ by (metis coset-mult-assoc rcosets-closed subgroup subgroup-self)
  with $HG M N g$ have $a1 : M = N$ by (metis l-inv coset-mult-one subgroup subgroup-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 $G (\lambda g. \lambda U \in \text{rcosets } H. U \#> inv g) \cdot (\text{rcosets } H)$
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 $(\lambda U \in \text{rcosets } H. U \#> inv h) \in \text{carrier } (\text{BijGroup (rcosets } H))$
  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 $\text{cosx cosy}$
    where $\text{cosx} = (\lambda U \in \text{rcosets } H. U \#> inv x)$
    and $\text{cosy} = (\lambda U \in \text{rcosets } H. U \#> inv y)$
  with $x y HG$ have $\text{cosx} \equiv (\text{Bij } (\text{rcosets } H)) \cdot \text{cosy} \equiv (\text{Bij } (\text{rcosets } H))$
  by (metis inv-mult-on-rcosets-is-bij)
  hence $\text{cosx} \otimes \text{BijGroup } (\text{rcosets } H) \cdot \text{cosy} = (\lambda U \in \text{rcosets } H. \text{cosx} \cdot (\text{cosy } U))$
unfolding BijGroup-def by (auto simp: compose-def)
also have \ldots = (\lambda U : rcosets H. U \#> \text{inv} \ (x \otimes y)) unfolding cosx-def cosy-def
proof (rule restrict-ext)
  fix U
  assume U : U \in rcosets H
  with HG y have U \#> \text{inv} y \in rcosets H by (metis inv-closed rcosets-closed)
  with x y HG U have (\lambda U : rcosets H. U \#> \text{inv} x) \ ((\lambda U : rcosets H. U \#> \text{inv} y) \ U) = U \#> \text{inv} y \#> \text{inv} x
    by auto
  also from x y HG have \ldots = U \#> \text{inv} \ (x \otimes y)
    by (metis inv-mult-group coset-mult-assoc inv-closed is-group subgroup.rcosets-carrier)
  finally show (\lambda U : rcosets H. U \#> \text{inv} x) \ ((\lambda U : rcosets H. U \#> \text{inv} y) \ U)
    = U \#> \text{inv} \ (x \otimes y).
  qed
  finally show (\lambda U : rcosets H. U \#> \text{inv} \ (x \otimes y)) = (\lambda U : rcosets H. U \#> \text{inv} x) \otimes \text{BijGroup} \ (rcosets H) \ (\lambda U : rcosets H. U \#> \text{inv} y)
    unfolding cosx-def cosy-def by simp
  qed
end

theory SubgroupConjugation
import 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 \subseteq 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 \Rightarrow -
where subgroups-of-size $p = \{H. \text{subgroup } H G \land \text{card } H = p\}$

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

lemma lcoset-join2:
  assumes $H$ : subgroup $H G$
  assumes $g$ : $g \in H$
  shows $g <\# H = H$
proof
  auto
next
fix $x$
assume $x : x \in \langle g <\# H \rangle$
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)
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 $M$ : $M \subseteq \text{carrier } G$
  assumes $g$ : $g \in \text{carrier } G$
  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 invg : invg $g \in \text{carrier } G$ 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
    qed
  qed
  qed

17
with $g \inv g xG yG$ have $(\inv g \otimes g) \otimes x = (\inv g \otimes g) \otimes y$ by (metis m-assoc)

with $g \inv g xG yG$ show $x = y$ by simp

qed

text

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 $gG : g \in \text{carrier } G$
- shows subgroup $(g <\# (H >\# \inv g)) G$

**Proof**

- from $gG$ have $\inv g \in \text{carrier } G$ by (rule inv-closed)
  - with $HG$ have $(H >\# \inv g) \subseteq \text{carrier } G$ by (metis r-coset-subset-G subgroup.subset)
  - with $gG$ show $g <\# (H >\# \inv g) \subseteq \text{carrier } G$ by (metis l-coset-subset-G)

next

- from $gG$ have $\inv gG: g \in \text{carrier } G$ by (metis inv-closed)
  - with $HG$ have lcosSubset:$(H >\# \inv g) \subseteq \text{carrier } G$ by (metis r-coset-subset-G subgroup.subset)

fix $x y$

- assume $x : x \in g <\# (H >\# \inv g)$ and $y : y \in g <\# (H >\# \inv g)$
  - then obtain $x' y'$ where $x' : x' \in H >\# \inv g x = g \otimes x'$ and $y' : y' \in H >\# \inv g y = g \otimes y'$ unfolding l-coset-def by auto

then obtain $hx hy$ where $hx : hx \in H x' = hx \otimes \inv g$ and $hy : hy \in H y' = hy \otimes \inv g$ unfolding r-coset-def by auto

- with $x' y'$ have $x2 : x = g \otimes (hx \otimes \inv g)$ and $y2 : y = g \otimes (hy \otimes \inv g)$ by auto
  - hence $x \otimes y = (g \otimes (hx \otimes \inv g)) \otimes (g \otimes (hy \otimes \inv g))$ by simp
  - also from $hx hy HG$ have $hxG : hx \in \text{carrier } G$ and $hyG : hy \in \text{carrier } G$ by (metis subgroup.mem-carrier)+

with $gH hy xG$ have $(g \otimes (hx \otimes \inv g)) \otimes (g \otimes (hy \otimes \inv g)) = g \otimes hx \otimes (\inv g \otimes g) \otimes hy \otimes \inv g$ by (metis m-assoc m-closed)

also from $\inv gG$ have ... = $g \otimes hx \otimes 1 \otimes hy \otimes \inv g$ by simp

also from $gG hxG$ have ... = $g \otimes hx \otimes hy \otimes \inv g$ by (metis m-closed r-one)
also from \( gG \) have \( hxG \) \( \text{inv} g \) \( \ldots = g \otimes ((hx \otimes hy) \otimes \text{inv} g) \) by (metis \( gG \) \( hxG \) \( \text{inv} G \) \( m\)-assoc \( m\)-closed)

finally have \( xy:x \otimes y = g \otimes (hx \otimes hy \otimes \text{inv} g) \).
from \( hx \otimes hy \otimes H \) have \( hx \otimes hy \in H \) by (metis subgroup.m-closed)
with \( \text{inv} G \) \( HG \) have \( (hx \otimes hy) \otimes \text{inv} g \in H \) \#> \( \text{inv} g \) by (metis rcosI subgroup.subgroup)
with \( gG \) \( HG \) have \( g \otimes (hx \otimes hy \otimes \text{inv} g) \in g <\# (H \) \#> \( \text{inv} g) \) by (metis lcosI)
with \( xy \) show \( x \otimes y \in g <\# (H \) \#> \( \text{inv} g) \) by simp

next
from \( gG \) have \( \text{inv} G \) \( g \in \) carrier \( G \) by (metis inv-closed)
with \( HG \) have \( \text{lcosSubset}: (H \) \#> \( \text{inv} g) \subseteq \) carrier \( G \) by (metis r-coset-subset-G subgroup.subgroup)
from \( HG \) have \( 1 \in H \) by (rule subgroup.one-closed)
with \( \text{inv} G \) \( HG \) have \( 1 \otimes \text{inv} g \in H \) \#> \( \text{inv} g \) by (metis rcosI subgroup)
with \( gG \) \( \text{lcosSubset} \) have \( g \otimes (1 \otimes \text{inv} g) \in g <\# (H \) \#> \( \text{inv} g) \) by (metis lcosI)
with \( gG \) \( \text{inv} G \) show \( 1 \in g <\# (H \) \#> \( \text{inv} g) \) by simp

next
from \( gG \) have \( \text{inv} G \) \( g \in \) carrier \( G \) by (metis inv-closed)
with \( HG \) have \( \text{lcosSubset}: (H \) \#> \( \text{inv} g) \subseteq \) carrier \( G \) by (metis r-coset-subset-G subgroup.subgroup)

fix \( x \)
assume \( x \in g <\# (H \) \#> \( \text{inv} g) \)
then obtain \( x' \) where \( x':x' \in H \) \#> \( \text{inv} g \) \( x = g \otimes x' \) unfolding \( l\)-coset-def
by auto
then obtain \( hx \) where \( hx:hx \in H \) \( x' = hx \otimes \text{inv} g \) unfolding \( r\)-coset-def
by auto
with \( x' \) \( hx \) have \( \text{inv} hx: \text{inv} hx \in H \) by (metis subgroup.m-inv-closed)
from \( x' \) \( hx \) have \( \text{inv} x = \text{inv} (g \otimes (hx \otimes \text{inv} g)) \) by simp
also from \( x' \) \( hx \) \( HG \) \( gG \) \( \text{inv} G \) have \( \ldots = \text{inv} (\text{inv} g) \otimes \text{inv} hx \otimes \text{inv} g \) by (metis calculation in-mono inv-mult-group lcosSubset subgroup.mem-carrier)
also from \( gG \) have \( \ldots = g \otimes \text{inv} hx \otimes \text{inv} g \) by simp
also from \( gG \) \( \text{inv} hx \) \( HG \) have \( \ldots = g \otimes (\text{inv} hx \otimes \text{inv} g) \) by (metis m-assoc subgroup.mem-carrier)
finally have \( \text{inv}: \text{inv} x = g \otimes (\text{inv} hx \otimes \text{inv} g) \).
with \( \text{inv} hx \) \( \text{inv} G \) \( HG \) have \( (\text{inv} hx) \otimes \text{inv} g \in H \) \#> \( \text{inv} g \) by (metis rcosI subgroup)
with \( gG \) \( \text{lcosSubset} \) have \( g \otimes (\text{inv} hx \otimes \text{inv} g) \in g <\# (H \) \#> \( \text{inv} g) \) by (metis lcosI)
with \( \text{inrix} \) show \( \text{inrix} x \in g <\# (H \) \#> \( \text{inv} g) \) by simp

qed

definition \( \text{conjugation-action}\::\text{nat} \Rightarrow - \)
where \( \text{conjugation-action} p = (\lambda g:G. \lambda P:G\text{-subgroup-of-size} p. g <\# (P \) \#> \( \text{inv} g)) \)

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

lemma $\text{conjugation-is-Bij :}$
assumes $\text{fin : finite } (\text{carrier } G)$
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 $\text{invg : inv } g \in \text{carrier } G$ by (rule $\text{inv-closed}$)
from $g$ have $\text{conjugation-action } p \ g \in \text{extensional } (\text{subgroups-of-size } p)$ unfolding $\text{conjugation-action-def}$ 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 : $\text{bij-betw-def}$)
  show inj-on $(\text{conjugation-action } p \ g) \ (\text{subgroups-of-size } p)$
    proof (rule inj-onto1)
    fix $U \ V$
    assume $U : U \in \text{subgroups-of-size } p$ and $V : V \in \text{subgroups-of-size } p$
    hence $\text{subsetG : U } \subseteq \text{carrier } G \ V \subseteq \text{carrier } G$ unfolding $\text{subgroups-of-size-def}$
    by (metis (lifting) $\text{mem-Collect-eq subgroup.subset}$)+
    hence $\text{subsetL : U } \# > \text{inv } g \subseteq \text{carrier } G \ V \# > \text{inv } g \subseteq \text{carrier } G$ by (metis $\text{invg r-coset-subset-G}$)+
    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 $\text{conjugation-action-def}$ 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 $\text{g invg lcos-m-assoc r-coset-subset-G subsetG}$)
    hence $1 <# (U \# > \text{inv } g) = 1 <# (V \# > \text{inv } g)$ by (metis $\text{g l-inv}$)
    hence $U \# > \text{inv } g = V \# > \text{inv } g$ by (metis $\text{subsetL lcos-mult-one}$)
    hence $(U \# > \text{inv } g) \# > g = (V \# > \text{inv } g) \# > g$ by simp

20
hence $U \#> (\text{inv } g \otimes g) = V \#> (\text{inv } g \otimes g)$ by (metis \text{coset-mult-assoc} g \text{ inv-closed subsetG})

hence $U \#> 1 = V \#> 1$ by (metis g \text{l-inv})

thus $U = V$ by (metis \text{coset-mult-one subsetG})

qed

next

fix $P$

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

thus $\text{conjugation-action } p \text{ g } P \in \text{subgroups-of-size } p$ by (metis \text{fin g conjugation-is-size-invariant})

next

fix $P$

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

with \text{invg} have $\text{conjugation-action } p \text{ (inv g) } P \in \text{subgroups-of-size } p$ by (metis \text{fin invg conjugation-is-size-invariant})

with \text{invg P} have $(\text{inv g}) <# (P \#> g) \in \text{subgroups-of-size } p$ unfolding \text{r-coset-def l-coset-def} by (simp add: m-assoc)

also from $P$ have $PG : P \subseteq \text{carrier G}$ unfolding \text{subgroups-of-size-def} by (auto simp add: subgroup)

have $\forall p \in P. \ g \otimes (\text{inv g} \otimes (p \otimes g) \otimes \text{inv g}) = p$

proof (auto)

fix $p$

assume $p \in P$

with $PG$ have $p : p \in \text{carrier G}$

with $g$ \text{invg} have $g \otimes (\text{inv g} \otimes (p \otimes g) \otimes \text{inv g}) = (g \otimes \text{inv g}) \otimes p \otimes (g \otimes \text{inv g})$ by (metis \text{m-assoc m-closed})

also with $g$ \text{invg g p} have $\ldots = p$ by (metis \text{l-one r-inv r-one})

finally show $g \otimes (\text{inv g} \otimes (p \otimes g) \otimes \text{inv g}) = p$.

qed

hence $(\bigcup p \in P. \ {g \otimes (\text{inv g} \otimes (p \otimes g) \otimes \text{inv g})}) = P$ by simp

finally have $g <# ((\text{inv g}) <# (P \#> g)) \#> \text{inv g}) = P$

with $I$ have $P \in (\lambda P. \ g <# (P \#> g)) \ ' \text{subgroups-of-size } p$ by auto

with $P$ \text{g show} $P \in \text{conjugation-action } p \ g \ ' \text{subgroups-of-size } p$ unfolding \text{conjugation-action-def} by simp

qed

ultimately show $\text{thesis}$ unfolding \text{BijGroup-def Bij-def} by simp

qed

lemma $l$-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 $x \in g \#> h$

then obtain $p$ where $p \in P$ and $p: x = g \otimes (p \otimes h)$ unfolding $l$-coset-def
theorem acts-on-subsets:
  assumes fin:finite (carrier G)
  shows group-action G (conjugation-action p) (subgroups-of-size p)
proof
  with fin show conjugation-action p g ∈ carrier (BijGroup (subgroups-of-size p)) unfolding BijGroup-def by (metis conjugation-is-Bij partial-object.select-convs(1))
next
fix x y
assume x:x ∈ carrier G and y:y ∈ carrier G
hence invx:inv x ∈ carrier G and invy:inv y ∈ carrier G by (metis inv-closed)+
from x y have xyG:x ⊗ y ∈ 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 ∈ Bij (subgroups-of-size p) unfolding conjx-def by (metis conjugation-is-Bij)
from fin y have yBij:conjy ∈ Bij (subgroups-of-size p) unfolding conjy-def by (metis conjugation-is-Bij)
  have conjx ⊗ BijGroup (subgroups-of-size p) conjy
    = (λg∈Bij (subgroups-of-size p). restrict (compose (subgroups-of-size p) g) (Bij (subgroups-of-size p))) conjy conjy unfolding BijGroup-def by simp
    also from xBij yBij have ... = compose (subgroups-of-size p) conjy conjy by simp
  also have ... = (λP∈subgroups-of-size p. conjx (conjy P)) by (metis compose-def)
  also have ... = (λP∈subgroups-of-size p. x ⊗ y ▸ P) by (metis inv-closed)+
proof(rule restrict-ext)
  fix P
  assume P:P ∈ subgroups-of-size p
  hence PG:P ⊆ carrier G unfolding subgroups-of-size-def by (auto simp:subgroup.subset)
with y have yPG: y <$> P ⊆ carrier G by (metis l-coset-subset-G)
  from x y have invxyG: inv (x ⊗ y) ∈ carrier G and xyG: x ⊗ y ∈ carrier G
  using inv-closed m-closed by auto
  from yBij have conjy: subgroups-of-size p = subgroups-of-size p unfolding Bij-def bij-betw-def by simp
    with P have conjyP: conjy P ∈ subgroups-of-size p unfolding Bij-def bij-betw-def
      by (metis (full-type) imageI)
      with x y have conjx (conjy P) = x <$> ((y <$> (P #> inv y)) #> inv x)
      unfolding conjy-def conjx-def conjugation-action-def by simp
      also from y invy PG have ... = x <$> (((y <$> P) #> inv y) #> inv x) by (metis lr-coset-assoc)
      also from PG inx invy y have ... = x <$> ((y <$> P) #> (inv y ⊗ inv x)) by (metis inv-mul-group)
      also from inxvyG x yPG have ... = (x <$> (y <$> P)) #> inv (x ⊗ y) by (metis lr-coset-assoc)
      also from x y PG have ... = ((x ⊗ y) <$> P) #> inv (x ⊗ y) by (metis lcos-m-assoc)
      also from xG inxvyG PG have ... = (x ⊗ y) <$> (P #> inv (x ⊗ y)) by (metis lr-coset-assoc)
      finally show conjx (conjy P) = x ⊗ y <$> (P #> inv (x ⊗ y)).
    qed
  finally have conjx ⊗ BijGroup (subgroups-of-size p) conjy = (λP∈subgroups-of-size p. x ⊗ y <$> (P #> inv (x ⊗ y))).
  with xG show conjugation-action p (x ⊗ y)
    = conjugation-action p x ⊗ BijGroup (subgroups-of-size p) 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 fin:finite (carrier G)
  assumes P:P ∈ subgroups-of-size p
  shows P ⊆ group-action.stabilizer G (conjugation-action p) P
proof
  from P have PG:subgroup P G unfolding subgroups-of-size-def by simp
  from fin interpret conj: group-action G (conjugation-action p) (subgroups-of-size p) by (rule acts-on-subsets)
  fix x
  assume x:x ∈ P
  with PG have inv x ∈ P by (metis subgroup.m-inv-closed)
  from x P have xG: x ∈ carrier G unfolding subgroups-of-size-def subgroup-def by auto
    with P have conjugation-action p x P = x <$> (P #> inv x) unfolding conjugation-action-def by simp
    also from ⟨inv x ∈ P⟩ PG have ... = x <$> P by (metis coset-join2 subgroup.mem-carrier)
also from \( PG \) \( x \) have \( \ldots = P \) by (rule \( lcoset\)-join2)

finally have conjugation-action \( p \) \( x P = P \).

with \( xG \) show \( x \in \) group-action \( (\text{conjugation-action } p) P \) unfolding conj.stabilizer-def by simp

qed

corollary stabilizer-supergyp-P:
assumes fin:finite (carrier \( G \))
assumes \( P: P \in \) subgroups-of-size \( p \)
shows subgroup \( P \) \( (G\langle\text{carrier} := \text{group-action.stabilizer } G \ (\text{conjugation-action } p) P\rangle) \)

proof –
from assms have \( P \subseteq \text{group-action.stabilizer } G \ (\text{conjugation-action } p) P \) 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 \( (\text{group-action.stabilizer } G \ (\text{conjugation-action } p) P) \) \( G \) by (metis acts-on-subsets group-action.stabilizer-is-subgroup)
ultimately show \( ?\)thesis by (metis is-group 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 \( (G\langle\text{carrier} := P\rangle) \) (conjugation-action \( p \))

proof –
from fin interpret \( \text{conjG: group-action } G \) 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 \( \text{conjP: group-action } G\langle\text{carrier} := P\rangle \) (conjugation-action \( p \)) (subgroups-of-size \( p \)) by (metis acts-on-subsets group-action.subgroup-action)
from fin \( P \) have \( P \subseteq \text{conjG.stabilizer } P \) by (rule stabilizer-contains-P)
hence \( P \subseteq \text{conjP.stabilizer } P \) using \( \text{conjG.stabilizer-def conjP.stabilizer-def} \) by auto
with \( P \) show \( P \in \text{conjP.fixed-points} \) unfolding \( \text{conjP.fixed-points-def} \) by auto

qed

lemma conj-wo-inv:
assumes \( QG: \text{subgroup } Q \) \( G \)
assumes \( PG: \text{subgroup } P \) \( G \)
assumes \( g: g \in \text{carrier } G \)
assumes \( \text{conj: inv } g <\# (Q \#> g) = P \)
shows \( Q \#> g = g <\# P \)

proof –
from \( g \) have invg:inv \( g \in \text{carrier } G \) by (metis inv-closed)
from \( \text{conj} \) have \( g <\# (\text{inv } g <\# (Q \#> g)) = g <\# P \) by simp
with \( QG \) \( \text{g invg have} \ (g \otimes \text{inv } g) <\# (Q \#> g) = g <\# P \) by (metis lcos-m-associ r-coset-subset-G subgroup.subset)
with \( g \) \( \text{invg have} \ 1 <\# (Q \#> g) = g <\# P \) by (metis r-inv)
with \( Q G g \) 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 \( \forall x. \ x \in A \Longrightarrow x = y \)
assumes \( y \in A \)
shows \( A = \{y\} \)
using assms by fastforce

context group
begin

lemma set-mult-inclusion:
assumes \( H:subgroup H G \)
assumes \( Q:P \subseteq carrier G \)
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 \( card (rcosets H) \ast card H = order G \) by (metis lagrange)
thus ?thesis by (metis dvd-triv-left mult.commute)
next
case False
hence 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:¬ (p dvd m)

context snd-sylow
begin

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

lemma sylow-greater-zero:
  shows card (subgroups-of-size \( (p^a) \)) > 0
  proof
    obtain P where PG:subgroup P G and cardP:card P = p^a by (metis sylow-thm)
    hence P ∈ subgroups-of-size \( (p^a) \) unfolding subgroups-of-size-def by auto
    moreover from finite-G have finite (subgroups-of-size \( (p^a) \)) unfolding subgroups-of-size-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 ∈ subgroups-of-size \( (p^b) \)
  assumes Psize:P ∈ subgroups-of-size \( (p^a) \)
obtains $g$ where $g \in carrier\ G\ H \subseteq g \ < \#\ (P \ #\ >\ inv\ g)$

proof:

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

27
with $g \text{ HG } PG$ invg $P \ll (<g \ll H \rt) \text{ inv } g \subseteq P$ by (metis lr-coset-assoc r-coset-subset-G rcos-assoc-lcos setmult-rcos-assoc subgroup.subset)

with $PG$ HG invg $g \ll H \rt$ invg $g \subseteq P$ by (metis l-coset-subset-G r-coset-subset-G set-mul-inclusion)

with $g$ have $(g \ll H \rt \text{ inv } g) \rt$ invg $(g \ll H \rt \text{ inv } g) \subseteq \text{ inv } g \rt$ $g \subseteq \text{ inv } g \rt$ unfolding r-coset-def by auto

with HG $g$ invg $H \subseteq P$ by (metis coset-mult-assoc coseq-mult-inv2 l-coset-subset-G)

with $g$ have $(\text{ inv } g) \ll (g \ll H \rt) \subseteq \text{ inv } g \ll (P \rt) \text{ inv } g)$ unfolding l-coset-def by auto

with HG $g$ invg $H \subseteq g \ll (P \rt) \text{ inv } g)$ by (metis inv-inv leos-m-assoc leos-mult-one r-inv)

with invg $g$ have thesis by (auto dest:that)

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

theorem sylow-contained-in-sylow-group:

assumes $H:G \subseteq \text{ subgroups-of-size (p } \cdot a)$$$

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

proof

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

obtain $P$ where PG $P$ G and cardP:card $P = p \cdot a$ by (metis sylow-thm)

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

with $H$ obtain $g$ where $g:G \in G$ $H \subseteq g \ll (P \rt) \text{ inv } g)$ by (metis ex-conj-sylow-group)

moreover note $P$size $g$

moreover with finite-$G$ have conjugation-action (p $\cdot a) \ g \ P \subseteq \text{ subgroups-of-size (p } \cdot a)$ 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 $P:G \subseteq \text{ subgroups-of-size (p } \cdot a)$

assumes $Q:G \subseteq \text{ subgroups-of-size (p } \cdot a)$

obtains $g$ where $g:G \in G$ $Q = g \ll (P \rt) \text{ inv } g)$

proof

from $P$ have card $P = p \cdot a$ unfolding subgroups-of-size-def by simp

from $Q$ have card $Q = p \cdot a$ unfolding subgroups-of-size-def by simp

from $Q \ P$ obtain $g$ where $g:G \in G$ $Q \subseteq g \ll (P \rt) \text{ inv } g)$ by (rule ex-conj-sylow-group)

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

moreover from $g \ P$ have conjugation-action (p $\cdot a) \ g \ P = g \ll (P \rt) \text{ inv } g)$ unfolding conjugation-action-def by simp

28
ultimately have conjSize: g <| (P #>) inv g) ∈ subgroups-of-size (p ^ a)
unfolding conjugation-action-def by simp
  with Q card 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 ∈ subgroups-of-size (p ^ a)
  assumes Q:Q ∈ subgroups-of-size (p ^ a)
  shows (P,Q) ∈ group-action.same-orbit-rel G (conjugation-action (p ^ a)) (subgroups-of-size (p ^ a))
unfolding group-action.same-orbit-rel-def
proof –
  from Q P obtain g where g:g ∈ carrier G P = g <| (Q #>) inv g) by (rule sylow-conjugate)
  with Q P have g' : conjugation-action (p ^ a) g Q = P unfolding conjugation-action-def by simp
  from finite-G interpret conj: group-action G (conjugation-action (p ^ a)) (subgroups-of-size (p ^ a)) by (rule acts-on-subsets)
  have conj.same-orbit-rel = \{X ∈ (subgroups-of-size (p ^ a)) × subgroups-of-size (p ^ a)). ∃g ∈ carrier G. ((conjugation-action (p ^ 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:

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

29
unfolding conjugation-action-def by auto

}\}
assume Q ∈ group-action.orbit G (conj. (p ^ a)) (subgroups-of-size (p ^ a)) P
with P conj. orbit-char obtain g where g: g ∈ carrier G Q = conj. (p ^ a) g P by auto
with finite-G P show Q ∈ subgroups-of-size (p ^ a) by (metis conj. is-size-invariant)
qed

theorem num-sylow-normalizer:
assumes Ps: P ∈ subgroups-of-size (p ^ a)
shows card (rcosets G|carrier := group-action.stabilizer G (conj. (p ^ a)) P) P) * p ^ a = card (group-action.stabilizer G (conj. (p ^ a)) P)
proof –
  from finite-G interpret conj: group-action G (conj. (p ^ a)) (subgroups-of-size (p ^ a)) by (rule acts-on-subsets)
  from Ps have PG:subgroup P G and cardP:card P = p ^ a unfolding subgroups-of-size-def by auto
  with finite-G have order G = card (conj.orbit P) * card (conj.stabilizer P) by (metis Ps order-action-orbit-size)
  with order-G Ps have p ^ a * m = card (subgroups-of-size (p ^ a)) * card (conj.stabilizer P) by (metis num-eq-card-orbit)
  moreover from Ps interpret slG: group G|carrier := conj.stabilizer P P | by (metis conj.stabilizer-is-subgroup subgroup-imp-group)
  from finite-G Ps have PG:subgroup P G and card P := card G|carrier := conj.stabilizer P P | by (rule stabilizer-supergp-P)
  from finite-G Ps have finite (conj.stabilizer P) by (metis card-infinite conj.stabilizer-is-subgroup less-nat-zero-code subgroup-finite-imp-card-positive)
  with finite-G Ps have PPG: subgroup P G and cardP:card P = p ^ a by (metis sylow-thm)
  hence Ps: P ∈ subgroups-of-size (p ^ a) unfolding subgroups-of-size-def by simp
  with finite-G have order G = card (conj.orbit P) * card (conj.stabilizer P) by (metis Ps order-action-orbit-size)
  with order-G Ps have orderEq:p ^ a * m = card (subgroups-of-size (p ^ a)) * card (conj.stabilizer P) by (metis num-eq-card-orbit)
  define k where k = card (rcosets G|carrier := conj.stabilizer P P)
  with Ps have k * p ^ a = card (conj.stabilizer P) by (metis num-sylow-normalizer)
qed

theorem (in sylow) num-sylow-ded-remainder:
shows card (subgroups-of-size (p ^ a)) dvd m
proof –
  from finite-G interpret conj: group-action G (conj. (p ^ a)) (subgroups-of-size (p ^ a)) by (rule acts-on-subsets)
  obtain P where PG:subgroup P G and cardP:card P = p ^ a by (metis sylow-thm)
  hence Ps: P ∈ subgroups-of-size (p ^ a) unfolding subgroups-of-size-def by simp
  with finite-G have order G = card (conj.orbit P) * card (conj.stabilizer P) by (metis Ps order-action-orbit-size)
  with order-G Ps have orderEq:p ^ a * m = card (subgroups-of-size (p ^ a)) * card (conj.stabilizer P) by (metis num-eq-card-orbit)
  define k where k = card (rcosets G|carrier := conj.stabilizer P P)
  with Ps have k * p ^ a = card (conj.stabilizer P) by (metis num-sylow-normalizer)
qed

30
with orderEq have \( p \cdot a \cdot m = \text{card} (\text{subgroups-of-size} (p \cdot a)) \cdot p \cdot a \cdot k\) by (auto simp:mult.assoc mult.commute)

hence \( p \cdot a \cdot m = p \cdot a \cdot \text{card} (\text{subgroups-of-size} (p \cdot a)) \cdot k\) by auto

then have \( m = \text{card} (\text{subgroups-of-size} (p \cdot a)) \cdot k\)

using pa-not-zero by auto

then show \( \text{thesis} \).

qed

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

lemma (in \text{snd-sylow}) restrict-locale:

assumes \( \text{subgrp} G \)

assumes \( \text{card} P \cdot a \cdot \text{dvd} \text{card} P \)

shows \( \text{snd-sylow} (G\{\text{carrier} := P\[]\}) p a ((\text{card} P) \text{ div} (p \cdot a)) \)

proof –

from \text{subgrp} interpret groupP: group \( G\{\text{carrier} := P\[]\} \) by (metis \text{subgroup-imp-group})

define \( k \) where \( k = (\text{card} P) \text{ div} (p \cdot a) \)

with \( \text{card have} \text{cardP:card} P = p \cdot a \cdot \text{card} P \cdot k \) by auto

hence \( \text{orderP:order} (G\{\text{carrier} := P\[]\}) = p \cdot a \cdot \text{card} P \cdot k \)

unfolding order-def by simp

from \( \text{cardP subgrp order-G have} \ p \cdot a \cdot \text{dvd} \ p \cdot a \cdot m \)

hence \( k \text{ dvd} m \)

by (metis nat-mult-dvd-cancel-disj pa-not-zero)

with pNotDum have nded:-\( p \text{ dvd} k \)

by (blast intro: dvd-trans)

define \( \text{PcalM} \) where \( \text{PcalM} = \{s. \ s \subseteq \text{carrier} (G\{\text{carrier} := P\[]\}) \land \text{card} s = p \cdot a\} \)

define \( \text{PRelM} \) where \( \text{PRelM} = \{N1, N2. \ N1 \in \text{PcalM} \land N2 \in \text{PcalM} \land (\exists g \in \text{carrier} (G\{\text{carrier} := P\[]\}). N1 = N2 \#> G\{\text{carrier} := P\[]\} \} \)

from \text{subgrp finite-G have} finite-groupP:finite (\text{carrier} (G\{\text{carrier} := P\[]\})) by (auto simp:subgroup-finite)

interpret \( \text{Nsylow} \) s:

unfolding snd-sylow-def snd-sylow-axioms-def sylow-def sylow-axioms-def k-def

using groupP.is-group prime-p orderP finite-groupP nded PcalM-def PRelM-def k-def by fastforce;

show ?thesis using k-def by (metis \( \text{Nsylow. is-snd-sylow} \))

qed

theorem (in \text{snd-sylow}) \( p\text{-sylow-mod-p} \):

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

proof –

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

hence \( \text{orderP:order} (G\{\text{carrier} := P\[]\}) = p \cdot a \)

unfolding order-def by auto

from PG have PsubG:P \subseteq \text{carrier} G by (metis subgroup-subset)

from PG \text{cardP have} PSize:P \in \text{subgroups-of-size} (p \cdot a)

unfolding subgroups-of-size-def by auto

from PG interpret groupP:group \( (G\{\text{carrier} := P\[]\}) \) by (rule subgroup-imp-group)

from \( \text{cardP have} PSize2:P \in \text{groupP.subgroups-of-size} (p \cdot a) \)

using groupP.subgroups-of-size-def groupP.subgroup-self by auto

31
from finite-G interpret conjG: group-action $G$ conjugation-action $(p \cdot a)$ subgroups-of-size $(p \cdot a)$ by (rule acts-on-subsets)

from $PG$ interpret conjP: group-action $G\langle\text{carrier} := P\rangle$ conjugation-action $(p \cdot a)$ subgroups-of-size $(p \cdot a)$ by (rule conjG.subgroup-action)

from finite-G have finite (subgroups-of-size $(p \cdot a)$) unfolding subgroups-of-size-def subgroup-def by auto

with orderP prime-p have card (subgroups-of-size $(p \cdot a)$) mod $p = \text{card conjP.fixed-points}$ mod $p$ by (rule conjP.fixed-point-congruence)

also have $\ldots = 1$

proof –

have $\bigwedge Q. Q \in \text{conjP.fixed-points} \Rightarrow Q = P$

proof –

fix $Q$

assume $Q\text{fixed}: Q \in \text{conjP.fixed-points}$

hence $Q\text{size}: Q \in \text{subgroups-of-size} (p \cdot a)$ unfolding conjP.fixed-points-def by simp

hence $\text{card} Q : \text{card} Q = p \cdot a$ 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 $N = \text{conjG.stabilizer} Q$

define $k$ where $k = (\text{card} N) \div (p \cdot a)$

from $N\text{-def} Q\text{size}$ have $\text{NG}\langle\text{carrier} := N\rangle G\langle\text{carrier} := P\rangle$ by (metis conjG.stabilizer-is-subgroup)

then interpret groupN: group $G\langle\text{carrier} := N\rangle$ by (metis subgroup-imp-group)

from $Q\text{size} N\text{-def} NQ\langle\text{carrier} := N\rangle$ unfolding conjP.fixed-points-def by auto

by auto

— The following proposition is used to show that $P = Q$ later

from $Q\text{size} N\text{-def} NQ\langle\text{carrier} := N\rangle$ have $\text{NfixesQ}: \forall g \in N. \text{conjugation-action} (p \cdot a) g Q = Q$

unfolding $N\text{-def} \text{conjG.stabilizer-def}$ by auto

from $Q\text{fixed} N\text{-def} NQ\langle\text{carrier} := N\rangle$ unfolding congP.fixed-points-def by auto

with $P\text{subG}$ have $P \subseteq N$ unfolding $N\text{-def} \text{conjG.stabilizer-def}$ by auto

with $PG$ N-def Qsize have $\text{PN}\langle\text{carrier} := P\rangle (G\langle\text{carrier} := N\rangle)$ by (metis conjG.stabilizer-is-subgroup is-group subgroup subgroup-of-subset)

with cardP have $p \cdot a$ dvd order $(G\langle\text{carrier} := N\rangle)$ using groupN.card-subgrp-dvd by force

hence $p \cdot a$ dvd card $N$ unfolding order-def by simp

with $NG$ have smaller-sylow:snd-sylow $(G\langle\text{carrier} := N\rangle)$ $p \cdot a$ unfolding k-def by (rule restrict-locale)

— Instantiate the snd-sylow Locale a second time for the normalizer of $Q$

define NcalM where $NcalM = \{ s, s \subseteq \text{carrier} (G\langle\text{carrier} := N\rangle) \wedge \text{card} s = p \cdot a \}$

define $NRelM$ where $NRelM = \{ (N1, N2). N1 \in NcalM \wedge N2 \in NcalM \wedge (\exists g \in \text{carrier} (G\langle\text{carrier} := N\rangle). N1 = N2 \# > G\langle\text{carrier} := N\rangle.g) \}$

interpret Nsylow: snd-sylow $G\langle\text{carrier} := N\rangle$ $p \cdot a$ $k$ $NcalM NRelM$

unfolding NcalM-def NRelM-def using smaller-sylow .

— $P$ and $Q$ are conjugate in $N$:

from cardP $PN$ have $P\text{size}N: P \in \text{groupN.subgroups-of-size} (p \cdot a)$ unfolding groupN.subgroups-of-size-def by auto

from cardQ $QN$ have $Q\text{size}N: Q \in \text{groupN.subgroups-of-size} (p \cdot a)$ unfold-
ing groupN.subgroups-of-size-def by auto
  from QsizeN PsizeN obtain g where g\in carrier (G\{carrier := N\}) P
  = g <# G\{carrier := N\} (Q \#> G\{carrier := N\} inv G\{carrier := N\} g) by (rule Nsglow.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 \( (p \cdot a) \) \( g \) \( Q \) = \( P \) 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 \in 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