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
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 $A \cap B = \emptyset$
using assms by (auto simp: card-Suc-eq)

**Theorem** card-singleton-set:

assumes 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 cardOne show $\bigwedge a. a \in A \implies \text{finite } a$ by (auto intro: card-ge-0-finite)
next
  show pairwise disjoint A
unfolding pairwise-def disjoint-def
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 $\text{x } \neq y$ show $x \cap y = \emptyset$ by (metis singleton-intersection)
  qed
  qed
also from cardOne have ... $= \text{card } A$ by simp
finally show $\text{thesis}$.
  qed

Intersecting Cosets are equal:

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

assumes 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 $\text{grouphom} : \text{group-hom } G$ (BijGroup $M$) $\varphi$

context group-action
begin

lemma is-group-action : group-action $G$ $\varphi$ $M$ ..
The action of 1 has no effect:

lemma one-is-id:
assumes $m : m \in M$
shows $(\varphi 1) \ m = m$

proof –
from $\text{grouphom}$ have $(\varphi 1) \ m = 1$(BijGroup $M$) $m$ by (metis group-hom.hom-one)
also have ... = $(\lambda x \in M. \ x) \ m$ unfolding BijGroup-def by (metis monoid.select-convs(2))
also from assms have ... = $m$ by simp
finally show ?thesis.
qed

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

lemma img-in-bij:
assumes $g : g \in \text{carrier } G$
shows $(\varphi g) \in \text{Bij } M$
using assms $\text{grouphom}$ 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 phi:($\varphi g$) $n = m$
shows $(\varphi \ (\text{inv } g)) \ m = n$

proof –
from $g$ have 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 $\varphi (\text{inv } g) \ m = (\text{restrict } (\text{inv-into } M \ (\varphi \ g)) \ M) \ m$ by simp
also from $m \ n$ have $\ldots = (\text{inv-into } M \ (\varphi \ g)) \ m$ by (metis restrict-def)
also from $g \ \varphi$ have $\ldots = (\text{inv-into } M \ (\varphi \ g)) \ ((\varphi \ g) \ n) \ m$ by simp
also from $\varphi \ g \in \text{Bij } M$ Bij-def have bij-betw (\varphi \ g) \ M \ M \ M \ M \ M \ by \ auto
hence inj-on (\varphi \ g) \ M \ M \ by \ (metis \ bij-betw-imp-inj-on)
with $g \ m$ have $(\text{inv-into } M \ (\varphi \ g)) \ M \ M \ m \ M$ by simp
finally show $\varphi (\text{inv } g) \ m = n$.
qed

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

lemma action-mult:
  assumes $g \ g \in \text{carrier } G$
  assumes $h \ h \in \text{carrier } G$
  assumes $m \ M \ M$
  shows $(\varphi \ g) \ ((\varphi \ h) \ m) = ((\varphi \ g) \ (\varphi \ h)) \ m$
proof --
  from $g$ have $\varphi \ g (\varphi \ g) \ M \ M \ by \ (rule \ img-in-bij)$
  from $h$ have $\varphi \ h (\varphi \ h) \ M \ M \ by \ (rule \ img-in-bij)$
  from $h$ have bij-betw (\varphi \ h) \ M \ M \ by \ (rule \ images-are-bij)$
  hence $((\varphi \ h) \ M = M \ M \ by \ (metis \ bij-betw-def)$
  with $m$ have $\text{hm:} (\varphi \ h) \ M \ M \ by \ (metis \ imageI)$
  from grouphom $g \ h$ have $(\varphi \ (\varphi \ h)) = ((\varphi \ g) \ (\varphi \ h)) \ M \ M \ by \ (rule \ group-hom.hom-mult)$
  hence $\varphi (\varphi \ h) \ M = ((\varphi \ g) \ (\varphi \ h)) \ M \ M \ by \ simp$
  also from $\varphi \ g \ \varphi h$ have $\ldots = (\text{compose } M \ (\varphi \ g) \ (\varphi \ h)) \ M \ M \ \text{unfolding } \text{BijGroup-def}$
  by simp
  also from $\varphi \ g \ \varphi h \ \text{hm}$ have $\ldots = (\varphi \ g) \ M \ M \ by \ (metis \ compose-eq \ M)$
  finally show $(\varphi \ g) \ (\varphi \ h) \ m = ((\varphi \ g) \ (\varphi \ h)) \ m$.
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 = $\{ p \in M \times M. \exists g \in \text{carrier } G. (\varphi \ g) \ (\text{snd } p) = (\text{fst } p)\}$

Use the library about equivalence relations to define the set of orbits and the map assigning to each element of $M$ its orbit
definition orbits
  where 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))
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 ⊆ 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 \( g : g \in \text{carrier } G \varphi g n = m \) unfolding same-orbit-rel-def by auto

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

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

hence \( mn^2 : m \in M n \in M \) by auto

from \( g \) \( mn^2 \) have \( \varphi (\text{invg }) m = n \) by (metis group-inv-rel)

with \( \text{invg } mn^2 \) 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 \( \ldots = x \) 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 ⇒ -

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 $ginv(\varphi (inv g)) = m$ 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 $h \in stabilizer m$
assume $h : h \in stabilizer m$
hence $gh : g \otimes h \in carrier G$ by (rule m-closed)
from $g h m$ have $\varphi (g \otimes h) m = (\varphi g) ((\varphi h) m)$ by (metis action-mult)
also from $g h$ have $... = m$ 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
assumption
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 $m \in orbit m$ 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 \( m \) 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 \( \text{rep : } \bigwedge U. \ U \in (\text{carrier } (G \text{ Mod (stabilizer } m))) \implies \text{rep } U \in U \)
shows bij-betw \((\lambda H. (\varphi (\text{inv (rep } H))) m)) (\text{carrier } (G \text{ Mod (stabilizer } m))) (\text{orbit } m)\)
proof (auto simp add: bij-betw-def)
show inj-on \((\lambda H. (\varphi (\text{inv (rep } H))) m) (\text{carrier } (G \text{ Mod stabilizer } m))\)
proof (rule inj-onI)
fix \( U \ V \)
assume \( U : U \in \text{carrier } (G \text{ Mod (stabilizer } m)) \)
assume \( V : V \in \text{carrier } (G \text{ Mod (stabilizer } m)) \)
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: \( \text{subgroup } (\text{stabilizer } m) G \) 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 hinvg: \( \text{inv } h \in \text{carrier } G \) 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 ginvG: \( \text{inv } g \in \text{carrier } G \) by (metis inv-closed)
from \( gG \) hinvG have ginvhG: \( g \otimes \text{inv } h \in \text{carrier } G \) by (metis m-closed)
assume repsG: \( \varphi (\text{inv (rep } U)) m = \varphi (\text{inv (rep } V)) m \)
hence g: \( \varphi (\text{inv } g) m = \varphi (\text{inv } h) m \) 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 \( \ldots = \varphi (g \otimes \text{inv } g) m \) by (metis action-mult)
also from \( m \ gG \) have \( \ldots = m \) by (auto simp: one-is-id)
finally have \( \varphi (g \otimes \text{inv } h) m = m \).
with \( \text{ginv} h \in G \) have \( (g \otimes \text{inv} h) \in \text{stabilizer} m \)

unfolding \text{stabilizer-def} by \text{simp}

hence \((\text{stabilizer} m) \ni (g \otimes \text{inv} h) = (\text{stabilizer} m) \ni 1\)
by (metis \text{coset-join2} \text{coset-mult-one} \text{mstabSubst} \text{stabilizer-is-subgroup sub-}

with \( \text{hinv} hG \in G \text{stabSubst} \) have \( \text{stabstabh} : (\text{stabilizer} m) \ni g = (\text{stabilizer}

m) \ni h \)
by (metis \text{coset-mult-inv1} \text{group.coset-mult-one is-group})

from \( \text{stabSubg} \text{is-group} U gU \) have \( U = (\text{stabilizer} m) \ni g \)

unfolding \text{FactGroup-def} by (simp add: \text{subgroup.repr-independence2})

also from \( \text{stabgstabh} \text{is-group} \text{stabSubg} V hV \text{subgroup.repr-independence2} \)

have \( \ldots = V \)

unfolding \text{FactGroup-def} by \text{force}

finally show \( U = V \).

qed next

have \( \text{stabSubst} : \text{stabilizer} m \subseteq \text{carrier} G \) unfolding \text{stabilizer-def} by \text{auto}

fix \( H \)

assume \( H : H \in \text{carrier} (G \text{ Mod stabilizer} m) \)

with \( \text{rep} \) have \( \text{rep} H \in H \) by \text{simp}

moreover with \( H \) \( \text{stabSubst} \) have \( H \subseteq \text{carrier} G \) unfolding \text{FactGroup-def}

\text{RCOSETS-def r-coset-def} by \text{auto}

ultimately have \( \text{rep} H \in \text{carrier} G \).

hence \( \text{inv} \text{rep} H \in \text{carrier} G \) by (rule \text{inv-closed})

with \( m \) show \( \varphi (\text{inv} \text{rep} H) m \in \text{orbit} m \) by (auto dest: \text{orbit-char})

next

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: \text{orbit-char})

hence \( \text{inv} g \in \text{carrier} G \) by \text{simp}

hence \( \text{stabilize} : ((\text{stabilizer} m) \ni (\text{inv} g)) \in \text{carrier} (G \text{ Mod stabilizer} m) \) unfolding

\text{FactGroup-def} \text{RCOSETS-def} \text{r-coset-def} by \text{auto}

hence \( \text{rep} ((\text{stabilizer} m) \ni (\text{inv} g))) \in (\text{stabilizer} m) \ni (\text{inv} g) \) by (metis \text{rep})

then obtain \( h \) where \( h : h \in \text{stabilizer} m \) \text{rep} \((\text{stabilizer} m) \ni (\text{inv} g)) = h \otimes

(\text{inv} g) \) unfolding \text{r-coset-def} by \text{auto}

with \( g \) have \( \varphi (\text{inv} \text{rep} ((\text{stabilizer} m) \ni (\text{inv} g))) m = \varphi (\text{inv} (h \otimes (\text{inv} g))) m \) by \text{simp}

also from \( hG \) \( h \in \text{carrier} G \) unfolding \text{stabilizer-def} by \text{simp}

with \( g \) have \( \varphi (\text{inv} (h \otimes (\text{inv} g))) m = \varphi (g \otimes (\text{inv} h)) m \) by (metis \text{inv-closed}

\text{inv-inv-mul} \text{group})

also from \( g hG \) \( h \) \( \in \text{carrier} G \)

have \( \ldots = \varphi g (\varphi (\text{inv} h) m) \) by (metis \text{action-mul} \text{inv-closed})

also from \( h m \) have \( \text{inv} h \in \text{stabilizer} m \) by (metis \text{stabilizer-is-subgroup sub-}

\text{group.m-inv-closed})

hence \( \varphi g (\varphi (\text{inv} h) m) = \varphi g m \) unfolding \text{stabilizer-def} by \text{simp}

also from \( g \) have \( \ldots = n \) by \text{simp}

finally have \( n = \varphi (\text{inv} \text{rep} ((\text{stabilizer} m) \ni (\text{inv} g))) m \)
with \( \text{stabinv} \) show \( n \in (\lambda H. \varphi (\text{inv} \text{rep} H) m) \cdot \text{carrier} (G \text{ Mod stabilizer} m) \)
by \text{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 $\text{fin} : \text{finite} (\text{carrier } G)$
- assumes $m : m \in M$
- shows $\text{order } G = \text{card } (\text{orbit } m) \ast \text{card } (\text{stabilizer } m)$

**proof**
- define $\text{rep where } \text{rep} = (\lambda U \in (\text{carrier } (G \text{ Mod } (\text{stabilizer } m))). \text{SOME } x. x \in U)$
- have $\bigwedge U. U \in (\text{carrier } (G \text{ Mod } (\text{stabilizer } m))) \implies \text{rep } U \in U$
  - proof
    - fix $U$
    - assume $U : U \in \text{carrier } (G \text{ Mod } \text{stabilizer } m)$
    - then obtain $g$ where $g \in \text{carrier } G \quad U = (\text{stabilizer } m) \#> g$ unfolding $\text{FactGroup-def } \text{RCSETS-def } \text{by } \text{auto}$
    - with $m$ have $(\text{SOME } x. x \in U) \in U$ by $(\text{metis } \text{rcos-self } \text{stabilizer-is-subgroup } \text{somel-ex})$
    - with $U$ show $\text{rep } U \in U$ unfolding $\text{rep-def } \text{by } \text{simp}$
- qed
  - with $m$ have $\text{bij:card } (\text{carrier } (G \text{ Mod } (\text{stabilizer } m))) = \text{card } (\text{orbit } m)$ by $(\text{metis } \text{bij-betw-same-card } \text{orbit-thm})$
  - from $\text{fin } m$ have $\text{card } (\text{carrier } (G \text{ Mod } (\text{stabilizer } m))) \ast \text{card } (\text{stabilizer } m) = \text{order } G$ unfolding $\text{FactGroup-def } \text{by } (\text{simp add: stabilizer-is-subgroup } \text{lagrange})$
    - with $\text{bij}$ show $?\text{thesis } \text{by } \text{simp}$
- qed

**lemma orbit-not-empty:**
- assumes $\text{fin} : \text{finite } M$
- assumes $A : A \in \text{orbits}$
- shows $\text{card } A > 0$

**proof**
- from $A$ obtain $m$ where $m \in M \quad A = \text{orbit } m$ unfolding $\text{orbits-def } \text{quotient-def } \text{orbit-def } \text{by } \text{auto}$
  - hence $m \in A$ by $(\text{metis orbit-contains-rep})$
  - hence $A \neq \{\}$ unfolding $\text{orbits-def } \text{by } \text{auto}$
  - moreover from $\text{fin } A$ have $\text{finite } A$ unfolding $\text{orbits-def } \text{quotient-def } \text{Image-def } \text{same-orbit-rel-def } \text{by } \text{auto}$
    - ultimately show $?\text{thesis } \text{by } \text{auto}$
- qed

**lemma fin-set-imp-fin-orbits:**
- assumes $\text{finM} : \text{finite } M$
- shows $\text{finite } \text{orbits}$

**using asssms** unfolding $\text{orbits-def } \text{quotient-def } \text{by } \text{simp}$

**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 $N : N \in \text{orbits card } N = 1 \ x \in N$ by auto

then obtain $y$ where $N : N = \text{orbit } y \ y \in M$ unfolding orbits-def quotient-def orbit-def

by auto

hence $y \in N$ by (metis orbit-contains-rep)

with $N$ have $N_{\text{sing}} : N = \{ x \}$ by (auto simp: card-Suc-eq)

hence $x = y$ by simp

with $\text{Norbit}$ have $\text{Norbit}2 : N = \text{orbit } x$ 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 $\{ x \in M : \}$ have $\{ \varphi \ g \ x = x \}$ by (auto dest: orbit-char)

with $N_{\text{sing}}$ show $\varphi \ g \ x = x$ by (metis $\text{Norbit}2$ singleton-iff)

qed

with $\{ x \in M : \}$ 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 $\text{orbit:orbit } m \in \text{orbits}$ unfolding orbits-def quotient-def orbit-def by auto

from $mM$ have $\text{card (orbit } m) = 1$ by (metis singleton-orbit-eq-fixed-point)

with $\text{orbit}$ have $\text{orbit } m \in \{ N \in \text{orbits. card } N = 1 \}$ by simp

with $mM$ show $m \in \bigcup \{ N \in \text{orbits. card } N = 1 \}$ 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 $\text{orderG:order } G = p^a$

assumes $\text{prime:prime } p$

assumes $\text{finM:finite } M$

assumes $\text{Norbit:} N \in \text{orbits}$

assumes $\text{card } N > 1$

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
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 N. N ∈ big-orbits ⇒ p dvd card N unfolding big-orbits-def
  by (auto simp: p-dvd-orbit-size)
  hence orbit-div: N. N ∈ big-orbits ⇒ card N = (card N div p) * p by (metis
dvd-mult-div-cancel1 nat-dvd-not-less nat-mult-1 prime
dvd-mult-power-nat prime-factor-nat prime-nat-iff zero-less-one)
  have card M = card (∪ orbits) unfolding orbits-def by (metis Union-quotient
  same-orbit-is-equiv)
  also have card (∪ orbits) = (∑ N∈orbits. card N) 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 A. A ∈ M // same-orbit-rel ⇒ finite A
    using finM same-orbit-rel-def by (auto dest:finite-equiv-class)
  qed
  also from orbit-part orbit-disj fin-parts have ...
  also from assms orbit-div fin-parts have ...
  also have ...
  finally have card M = (∑ N∈big-orbits. card N div p) * p + card fixed-points using
  singleton-orbits by (auto simp:sum-distrib-right)
hence card M mod p = ((∑ N∈big-orbits. card N div p) * p + card fixed-points)
mod $p$ by simp
also have \ldots = (\text{card fixed-points}) \mod p \text{ by (metis mod-mult-self3)}
finally show \text{thesis}.
qed

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

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

1.5 Some Examples for Group Actions

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

\textbf{lemma} \textbf{(in group)} right-mult-group-action:
\begin{itemize}
\item shows $\text{group-action} \ G \ (\lambda h. \ \lambda g \in \text{carrier} \ G. \ h \otimes g) \ (\text{carrier} \ G)$
\end{itemize}
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 ∈ 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
finally show (λg ∈ carrier G. (x ⊗ y) ⊗ g) = (λg ∈ carrier G. x ⊗ g) ⊗ BijGroup (carrier G)
(λg ∈ carrier G. y ⊗ g) unfolding multx-def multy-def by simp
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 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-between 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 \( M \#> \inv g \in \text{rcosets} \ H \) by (metis coset-mult-assoc
rcoset-mult-inv2 inv-closed is-group subgroup rcosets-carrier)
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 \( M \#> \inv g \)
\( = \) \( N \#> \inv g \) by simp
with \( HG \ M \) \( N \) \( g \) have \( M \#> (\inv g \otimes g) = N \#> (\inv g \otimes g) \) by (metis
coset-mult-assoc is-group subgroup m-inv-closed subgroup-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 \( \text{group-action} \ G (\lambda g. \lambda U \in \text{rcosets} \ H. \ U \#> \inv g) \) \( (\text{rcosets} \ H) \)
unfolding \( \text{group-action-def} \ \text{group-action-axioms-def} \ \text{group-hom-def} \ \text{group-hom-axioms-def} \ \text{hom-def} \)
proof (auto simp add:is-group group-BijGroup)
fix \( h \)
assume \( h \in \text{carrier} \ G \)
with \( HG \) show \( (\lambda U \in \text{rcosets} \ H. \ U \#> \inv h) \in \text{carrier} \ (BijGroup \ (\text{rcosets} \ H)) \)
unfolding \( \text{BijGroup-def} \) 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 \( \cos x \) \( \cos y \)
where \( \cos x = (\lambda U \in \text{rcosets} \ H. \ U \#> \inv x) \)
and \( \cos y = (\lambda U \in \text{rcosets} \ H. \ U \#> \inv y) \)
with \( x \) \( y \) \( HG \) have \( \cos x \in (\text{Bij} \ (\text{rcosets} \ H)) \) \( \cos y \in (\text{Bij} \ (\text{rcosets} \ H)) \)
by (metis inv-mult-on-rcosets-is-bij)+
hence \( \cos x \otimes \text{BijGroup} \ (\text{rcosets} \ H) \ \cos y = (\lambda U \in \text{rcosets} \ H. \ \cos x \ (\cos y U)) \)
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 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 $p = \{ H. \text{subgroup } H \subseteq G \wedge \text{card } H = p \}$

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 $H$: subgroup $H \subseteq G$
  assumes $g$: $g \in H$
  shows $g <\# H = H$
proof
  fix $x$
  assume $x$: $x \in g <\# H$
  then obtain $h$ where $h$: $h \in H \otimes x = g \otimes h$
    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 $M$: $M \subseteq \text{carrier } G$
  assumes $g$: $g \in \text{carrier } G$
  shows card ($M #> g$) = card $M$
proof
  have $M #> g \in \text{rcosets } M$
    by (metis assms(2) assms(3) rcosetsI)
  thus card ($M #> g$) = 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 card ($g <\# M$) = 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: $\text{inv } g \in \text{carrier } G$
    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
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’t 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’t 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 invG 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 invG G 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 \( gG hxG \) have \( \ldots = g \otimes ((hx \otimes hy) \otimes inv g) \) by (metis \( gG \otimes m-assoc \) \( m-closed \))

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 \( invgG \) \( HG \) 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 \( invgG \otimes inv g \in \) 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 \( 1 \in H \) by (rule subgroup.one-closed)

with \( invgG \) \( HG \) have \( 1 \otimes inv g \in H \) \( \# > inv g \) by (metis rcosI subgroup.subset)

with \( gG \) \( lcosSubset \) have \( g \otimes (1 \otimes inv g) \in g \# < (H \# > inv g) \) by (metis lcosI)

with \( gG \) \( invgG \) show \( 1 \in g \# < (H \# > inv g) \) by simp

next

from \( gG \) have \( invgG \otimes inv g \in \) 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 \( x' : hx \) 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 \( x' : hx \) \( HG \) \( gG \) \( invgG \) have \( \ldots = inv (inv g) \otimes inv hx \otimes inv g \) by (metis calculation in-mono in-mult-group lcosSubset subgroup.mem-carrier)

also from \( gG \) have \( \ldots = g \otimes inv hx \otimes inv g \) by simp

also from \( gG \) \( invhx \) \( HG \) have \( \ldots = g \otimes (inv hx \otimes inv g) \) by (metis m-assoc subgroup.mem-carrier)

finally have \( invx : inv x = g \otimes (inv hx \otimes inv g) \).

with \( invhx \) \( invgG \) \( HG \) 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 :: \( \text{nat} \Rightarrow - \)

where conjugation-action \( p = (\lambda g \in \text{carrier} \ G. \lambda \mathcal{P} \in \text{subgroups-of-size} \ p. \ g \# < (P \# > inv g)) \)

lemma conjugation-is-size-invariant:

assumes fin: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 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 \( PsubG : P \subset \text{carrier } G \) by (metis subgroup.subset)
  hence \( \text{PinvgsubG: } P \# > \text{inv } g \subset \text{carrier } G \) by (metis invg r-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 \( \text{card } PsubG \) fin have \( \text{have } (P \# > \text{inv } g) = p \) by (metis card PsubG fin)
with \( g \) have \( \text{PinvgsubG } \) fin show \( \text{card } (g \# \ (P \# > \text{inv } g)) = p \) by (metis card eq-rcoset)
qed

with \( P \ g \) show \( ?\text{thesis unfolding conjugation-action-def} \) 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 \( \text{invg: } inv \ g \in \text{carrier } G \) 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-on1)
    fix \( U \ V \)
    assume \( \text{U: } U \in \text{subgroups-of-size } p \) and \( \text{V: } V \in \text{subgroups-of-size } p \)
    hence \( \text{subsetG: } U \subset \text{carrier } G \ V \subset \text{carrier } G \) unfolding subgroups-of-size-def
    by (metis (lifting) mem-Collect-eq subgroup.subset)+
    hence \( \text{subselL: } U \# > \text{inv } g \subset \text{carrier } G \ V \# > \text{inv } g \subset \text{carrier } G \) by (metis 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 conjugation-action-def by simp
    hence \( \text{invg: } inv \ g \ (g \# \ (U \# > \text{inv } g)) = (\text{inv } g) \# \ (g \# \ (V \# > \text{inv } g)) \) by simp
    hence \( \text{invg: } inv \ g \# \ (U \# > \text{inv } g) = (\text{inv } g) \# \ (V \# > \text{inv } g) \) by (metis g invg lcos-m-assoc r-coset-subset-G subsetG)
    hence \( 1 \# \ (U \# > \text{inv } g) = 1 \# \ (V \# > \text{inv } g) \) by (metis g l-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

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

hence $U \triangleright> 1 = V \triangleright> 1$ by (metis $g$ l-inv)
thus $U = V$ by (metis coset-mult-one subsetG)

qed

next

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

next

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

with $\text{invg} \ P$
have $(\text{inv } g) \ <\ # \ (P \ #\ > ) \in$ subgroups-of-size $p$

unfolding conjugation-action-def by simp

hence $I:(\text{inv } g) \ <\ # \ (P \ #\ > ) \in$ subgroups-of-size $p$ by (metis $g$ inv-inv)

have $g \ <\ # \ ((\text{inv } g) \ <\ # \ (P \ #\ > ) \ #\ > \text{ inv } g) = (\bigcup p \in P. \ (g \otimes (\text{inv } g \otimes (p \otimes g) \otimes \text{inv } g)))$

unfolding r-coset-def l-coset-def by (simp add: m-assoc)

also from $P$ have $PG \subseteq \text{carrier } G$

unfolding subgroups-of-size-def by (auto simp add: subgroup subset)

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 $g$ 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 m-assoc m-closed)

also with $g$ invg $p$
have $\ldots = p$ by (metis 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 $g \ #\ ((\text{inv } g) \ #\ ((P \ #\ > ) \ #\ > \text{ inv } g)) \ P$.

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

with $P$ $g$
show $P \in \text{conjugation-action } p \ g \ (\text{subgroups-of-size } p)$

unfolding conjugation-action-def by simp

qed

ultimately show $\text{thesis}$

unfolding 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 $x \in g \ #\ ((P \ #\ > ) \ #\ > h)$

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)
  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 (clarsimp)
  fix g 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))) conjx conjy unfolding BijGroup-def by simp
  also from xBij yBij have ... = compose (subgroups-of-size p) conjx 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 #> inv (x ⊗ y)))
proof (rule restrict-ext)
  fix P
  assume P: P <∈ 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 \( inxyG; \text{inv} (x \otimes y) \in \text{carrier} G \) and \( xyG; x \otimes y \in \text{carrier} G \)
using inv-closed m-closed by auto
from \( yBij \) have \( \text{conjy} \) ' subgroups-of-size \( p \) = subgroups-of-size \( p \) unfolding Bij-def bij-betw-def
with \( P \) have \( \text{conjyP; conjy} P \in \text{subgroups-of-size} p \) unfolding Bij-def bij-betw-def
by (metis (full-type) imageI)
with \( x y \) have \( \text{conjx} \) (conjy \( P \)) = \( x < \# ((y < \# (P > \text{inv} y)) > \text{inv} x) \)
unfolding conjy-def conjx-def conj-action-def by simp
also from \( y \text{ inv} y PG \) have ... = \( x < \# (((y < \# P) > \text{inv} y) > \text{inv} x) \) by (metis l-coset-assoc)
also from \( PG \) have ... = \( x < \# ((y < \# P) > (\text{inv} y \otimes \text{inv} x)) \) by (metis inv-mul-group)
also from \( x y \) have ... = \( x < \# ((y < \# P) > \text{inv} (x \otimes y)) \) by (metis l-coset-assoc)
also from \( x y PG \) have ... = \( ((x \otimes y) < \# P) > \text{inv} (x \otimes y) \) by (metis lcos-m-assoc)
also from \( xyG \) have ... = \( (x \otimes y) < \# (P > \text{inv} (x \otimes y)) \) by (metis l-coset-assoc)
finally show \( \text{conjx} \) (conjy \( P \)) = \( x \otimes y < \# (P > \text{inv} (x \otimes y)) \).
qed
finally have \( \text{conjx} \otimes \text{BijGroup} \) (subgroups-of-size \( p \)) conjy = \( \lambda P \in \text{subgroups-of-size} p. x \otimes y < \# (P > \text{inv} (x \otimes y)) \).
with \( xyG \) show conj-action \( p \) \( x \otimes \text{BijGroup} \) (subgroups-of-size \( p \)) conj-action \( p y \)
unfolding conjy-def conjx-def conj-action-def by simp
qed

2.3 Properties of the Conjugation Action

lemma stabilizer-contains-P:
assumes fin:finite (carrier G)
assumes \( P; P \in \text{subgroups-of-size} p \)
shows \( P \subseteq \text{group-action.stabilizer} G \) (conj-action \( p \)) \( P \)
proof
from \( P \) have \( PG; \text{subgroup} P G \) unfolding subgroups-of-size-def by simp
from fin interpret conj:group-action G (conj-action \( p \)) (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 conj-action \( p \) \( x P = x < \# (P > \text{inv} x) \) unfolding conj-action-def by simp
also from \( \text{inv} x \in P \) PG have ... = \( x < \# P \) by (metis coset-join2 subgroup.mem-carrier)
also from \( PG \times \) have \( \ldots = P \) by (rule lcoset-join2)
finally have conjugation-action \( p \times P = P \).
with \( xG \) show \( x \in \text{group-action.stabilizer} \) \( G \) (conjugation-action \( p \)) \( P \) unfolding
conjug.stabilizer-def by simp
qed

\textbf{corollary} stabilizer-superyp-P:
\begin{itemize}
  \item assumes fin:finite (carrier \( G \))
  \item assumes \( P:P \in \text{subgroups-of-size} \) \( p \)
  \item shows subgroup \( P \) (\( G[\text{carrier := group-action.stabilizer} \ G (\text{conjugation-action} \ p) \ P] \))
\end{itemize}
proof
\begin{itemize}
  \item from assms have \( P \subseteq \text{group-action.stabilizer} \ G (\text{conjugation-action} \ p) \ P \) by (rule stabilizer-contains-P)
  \item moreover from \( P \) have subgroup \( P \subseteq G \) unfolding subgroups-of-size-def by simp
  \item 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)
  \item ultimately show \( \)?thesis by (metis is-group subgroup-of-subset)
\end{itemize}
qed

\textbf{lemma} (in group) P-fixed-point-of-P-conj:
\begin{itemize}
  \item assumes fin:finite (carrier \( G \))
  \item assumes \( P:P \in \text{subgroups-of-size} \) \( p \)
  \item shows \( P \in \text{group-action.fixed-points} \) \( G[\text{carrier := P}] \) (conjugation-action \( p \) (subgroups-of-size \( p \)))
\end{itemize}
proof
\begin{itemize}
  \item from fin interpret conjG: group-action \( G \) conjugation-action \( p \) subgroups-of-size \( p \) by (rule acts-on-subsets)
  \item from \( P \) have subgroup \( P \subseteq G \) unfolding subgroups-of-size-def by simp
  \item with fin interpret conjG: group-action \( G[\text{carrier := P}] \) (conjugation-action \( p \) (subgroups-of-size \( p \))) by (metis acts-on-subsets group-action.subgroup-action)
  \item from fin \( P \) have \( P \subseteq \text{conjG.stabilizer} \ P \) by (rule stabilizer-contains-P)
  \item hence \( P \subseteq \text{conjP.stabilizer} \ P \) using conjG.stabilizer-def conjP.stabilizer-def by auto
  \item with \( P \) show \( P \in \text{conjP.fixed-points} \) unfolding conjP.fixed-points-def by auto
\end{itemize}
qed

\textbf{lemma} conj-wo-inv:
\begin{itemize}
  \item assumes \( QG: \text{subgroup} \ Q \ G \)
  \item assumes \( PG: \text{subgroup} \ P \ G \)
  \item assumes \( g: g \in \text{carrier} \ G \)
  \item assumes conj:inv g \# (\( Q \ \# \) \( g \)) = \( P \)
  \item shows \( Q \ \# \) \( g \) = \( g \ \# \) \( P \)
\end{itemize}
proof
\begin{itemize}
  \item from \( g \) have invg:inv g \in \text{carrier} \ G by (metis inv-closed)
  \item from conj have \( g \ \# \) (\( \text{invg} \ \# \ (Q \ \# \ g) \)) = \( g \ \# \) \( P \) by simp
  \item with \( QG \) invg have \( g \ \odot \ \text{invg} \ \# (Q \ \# \ g) \) = \( g \ \# \) \( P \) by (metis lcos-m-assoc r-coset-subset-G subgroup.subset)
  \item with \( g \ \text{invg} \) have \( 1 \ \# (Q \ \# \ g) \) = \( g \ \# \) \( P \) by (metis r-inv)
\end{itemize}

24
theory SndSylow
imports SubgroupConjugation
begin

no-notation Multisetsubset-mset (infix "# 50")

3 The Secondary Sylow Theorems

3.1 Preliminaries

lemma singletonI:
assumes \( \forall x. x \in A \implies x = y \)
assumes \( y \in A \)
shows \( A = \{ y \} \)
using assms by fastforce

context group
begin

lemma set-mult-inclusion:
assumes H: 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)

end
end
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: \( \neg (p \text{ dvd } m) \)
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 \in subgroups-of-size \( p^a \) unfolding subgroups-of-size-def by auto
    hence subgroups-of-size \( p^a \) \( \neq \{\} \) by auto
    moreover from finite-G have finite (subgroups-of-size \( p^a \)) unfolding
    subgroups-of-size-def subgroups-by auto
    ultimately show \( \neg \)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 subgroups-of-size \( p^b \)
  assumes Psize: P \in subgroups-of-size \( p^a \)
obtains \( g \) where \( g \in \text{carrier } G H \subseteq g <\# (P \#> \text{inv } g) \)

proof –
from \( H \) have \( H \text{subgroup } G \) unfolding subgroups-of-size-def by auto
hence \( HG.H \subseteq \text{carrier } G \) unfolding subgroups-of-size-def by (simp add:subgroup.subset)
from \( P \text{size} \) have \( PG.P G \) and \( \text{card } P = p \cdot \text{a} \) unfolding subgroups-of-size-def by auto
  define \( H' \) where \( H' = G\langle \text{carrier } := H \rangle \)
from \( H \text{subgroup } G \) interpret \( H \text{group} ; \text{group } H' \) unfolding \( H' \)-def by (metis subgroup-imp-group)
from \( H \) have \( \text{order } H' : \text{order } H' = \text{p} \cdot \text{b} \) unfolding \( H' \)-def subgroups-of-size-def
order-def by simp
define \( \varphi \) where \( \varphi = (\lambda g. \lambda U \in \text{rossets } P. U \#> \text{inv } g) \)
  with \( PG \) interpret \( G \text{act} ; \text{group-action } G \varphi \) \( \text{rossets } P \) unfolding \( \varphi \)-def by
(metis inv-mult-on-rcosets-action)
from \( H \) interpret \( H' \text{act} ; \text{group-action } H' \varphi \) \( \text{rossets } P \) unfolding \( H' \)-def subgroups-of-size-def
by (metis (mono-tags) \( G \text{act} ; \text{group-action mem-Collect-eq} \))
from \( P \text{finite-G } PG \) have \( \text{finite } (\text{rossets } P) \) unfolding \( \text{RCOSETS-def } r \)-coset-def
by (metis (lifting) \( P \text{finite-emptyI } \text{finite-UN-I } \text{finite-insert} \))
  with \( \text{order } H' \) \( \text{sylow-axioms } \text{card } P \) have \( \text{card } H' \text{act.fixed-points mod } p = \text{card} \)
(\( \text{rossets } P \)) \( \text{mod } p \) unfolding \( \text{sylow-def } \text{sylow-axioms-def } \text{by} \) (metis \( H' \text{act.fixed-point-congruence} \))
moreover from \( P \text{finite-G } PG \) \( \text{order } G \) \( \text{card } P \) have \( \text{card} \) (\( \text{rossets } P \)) \( \text{p} \cdot \text{a} = \text{p} \cdot \text{a} \cdot \text{m} \) by (metis lagrange)
  with \( \text{prime-p } \) have \( \text{card} \) (\( \text{rossets } P \)) = \( \text{m} \) by (metis \( \text{less-nat-zero-code } \text{mult-cancel2} \)
\( \text{mult-is-0 } \text{mult.commute } \text{order } G \) \( \text{zero-less-o-G} \))
  hence \( \text{card} \) (\( \text{rossets } P \)) \( \text{mod } p = \text{m} \) \( \text{mod } p \) by simp
moreover from \( \text{pNotDvd} \) \( \text{prime-p } \) have \( ... \neq \text{0} \) by (metis \( \text{dvd-eq-mod-eq-0} \))
ultimately have \( \text{card } H' \text{act.fixed-points } \neq \text{0} \) by (metis \( \text{mod-0} \))
then obtain \( N \) where \( N:N \in H' \text{act.fixed-points } \) by fastforce
hence \( N \textset:N \in \text{rossets } P \) unfolding \( H' \text{act.fixed-points-def } \) by simp
then obtain \( g \) where \( g: g \in \text{carrier } G N = P \#> g \) unfolding \( \text{RCOSETS-def} \)
by auto
  hence \( \text{meg} : \text{inv } g \in \text{carrier } G \) by (metis \( \text{inv-closed} \))
  hence \( \text{meaning} : \text{inv } (\text{inv } g) \in \text{carrier } G \) by (metis \( \text{inv-closed} \))
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 \( H' \text{act.stabilizer-def } \text{using } H' \)-def by auto
with \( HG \) \( \text{N} \text{set} \) have \( a1: \forall h \in H. N \#> \text{inv } h \subseteq \text{N unfolding } \varphi \)-def by simp
have \( N \#> H \subseteq N \) unfolding \( \text{set-mult-def } r \)-coset-def
proof(auto)
  fix \( h \)
  assume \( n:n \in N \) and \( h:h \in H \)
  with \( H \) have \( \text{inv } h \in H \) by (metis \( \text{(mono-tags) mem-Collect-eq subgroup.m-inv-closed} \)
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 (metis \( \text{in-mono } \text{inv-inv} \))
qed
with \( g \) have \((P \#> g) \#> H) \#> \text{inv } g \subseteq (P \#> g) \#> \text{inv } g \) unfolding \( r \)-coset-def by auto
with \( PG \) \( g \text{inv} \) have \((P \#> g) \#> H) \#> \text{inv } g \subseteq P \) by (metis \( \text{coset-mult-assoc} \)
coset-mult-one \( r \)-inv subgroup.subset)
with $g\ H\ G$ invg have $P <\#> (g <\# H > inv g) \subseteq P$ by (metis l-coset-assoc r-coset-subset-G rcos-assoc-leos setmult-rcos-assoc subgroup.subset)
with $PG\ H\ G$ invg have $g <\# H > inv g \subseteq P$ by (metis l-coset-subset-G r-coset-subset-G set-mult-inclusion)
with $g$ have $(g <\# H > inv g) \# inv (inv g) \subseteq P \# inv (inv g)$ unfolding r-coset-def by auto
with $HG\ g$ invg invg have $g <\# H \subseteq P \# inv (inv g)$ by (metis coset-mult-assoc l-coset-mult-inv2 l-coset-subset-G)
with $g$ have $(inv g) <\# (g <\# H) \subseteq inv g <\# (P \# inv (inv g))$ unfolding l-coset-def by auto
with $HG\ g$ invg invg have $H \subseteq inv g <\# (P \# inv (inv g))$ by (metis inv-inv leos-m-assoc leos-mult-one r-inv)
with invg 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 $H: H \in$ subgroups-of-size $(p ^ a)$
obtains $S$ where $H \subseteq S$ and $S \in$ subgroups-of-size $(p ^ a)$
proof –
from $H$ have $HG:H \subseteq$ carrier $G$ unfolding subgroups-of-size-def by (simp add: subgroup.subset)
obtain $P$ where $PG$ subgroup $P G$ and cardP:card $P = p ^ a$ by (metis sylow-thm)
hence $P$ size $P \in$ subgroups-of-size $(p ^ a)$ unfolding subgroups-of-size-def by simp
with $H$ obtain $g$ where $g:g \in$ carrier $G H \subseteq g <\# (P \# inv g)$ by (metis ex-conj-sylow-group)
moreover note $P$ size $g$
moreover with finite-$G$ have conjugation-action $(p ^ a) g P \in$ subgroups-of-size $(p ^ a)$ 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 $P:P \in$ subgroups-of-size $(p ^ a)$
assumes $Q:Q \in$ subgroups-of-size $(p ^ a)$
obtains $g$ where $g \in$ carrier $G Q = g <\# (P \# inv g)$
proof –
from $P$ have card $P = p ^ a$ unfolding subgroups-of-size-def by simp
from $Q$ have Qcard:card $Q = p ^ a$ unfolding subgroups-of-size-def by simp
from $Q$ obtain $g$ where $g:g \in$ carrier $G Q \subseteq g <\# (P \# inv g)$ by (rule ex-conj-sylow-group)
moreover with $P$ finite-$G$ have conjugation-action $(p ^ a) g P \in$ subgroups-of-size $(p ^ a)$ by (metis conjugation-is-size-invariant)
moreover from $g P$ have conjugation-action $(p ^ a) g P = g <\# (P \# inv g)$ unfolding conjugation-action-def by simp

...
ultimately have \( \text{conjSize}: g \triangleleft (P \triangleright g) \) \( \in \text{subgroups-of-size } (p \cdot a) \)

unfolding \text{conjugation-action-def} by simp

with \( Q \) \( \text{card} \) have \( \text{card:card} (g \triangleleft (P \triangleright g)) = \text{card } Q \) unfolding \text{subgroups-of-size-def} by simp

from \text{conjSize finite-G} have \( \text{finite } (g \triangleleft (P \triangleright g)) \) by (metis (mono-tags) \text{finite-subset mem-Collect-eq subgrou}

subset \text{subgroups-of-size-def})

with \( g \) \( \text{card} \) have \( Q = g \triangleleft (P \triangleright g) \) by (metis \text{card:subset-eq})

with \( g \) show \( \text{thesis} \) by (metis that)

qed

corollary \text{sylow-conj-orbit-rel}:

assumes \( P;P \in \text{subgroups-of-size } (p \cdot a) \)

assumes \( Q;Q \in \text{subgroups-of-size } (p \cdot a) \)

shows \( (P,Q) \in \text{group-action.same-orbit-rel } G (\text{conjugation-action } (p \cdot a)) (\text{subgroups-of-size} (p \cdot a)) \)

unfolding \text{group-action.same-orbit-rel-def}

proof –

from \( Q \) \( P \) obtain \( g \) where \( g\in \text{carrier } G \) \( P = g \triangleleft (Q \triangleright inv g) \) by (rule \text{sylow-conjugate})

with \( Q \) \( P \) have \( g\text{':conjugation-action } (p \cdot a) g Q = P \) unfolding \text{conjugation-action-def}

by simp

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

have \text{conj.same-orbit-rel } \{X \in (\text{subgroups-of-size } (p \cdot a) \times \text{subgroups-of-size} (p \cdot a)). \exists g \in \text{carrier } G. ((\text{conjugation-action } (p \cdot a)) g) (\text{snd } X) = (\text{fst } X)\} by (rule \text{conj.same-orbit-rel-def})

with \( g\) \( g' \) \( P \) \( Q \) show \( ?\text{thesis} \) by auto

qed

3.6 Counting Sylow-Groups

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

theorem \text{num-eq-card-orbit}:

assumes \( P;P \in \text{subgroups-of-size } (p \cdot a) \)

shows \( \text{subgroups-of-size } (p \cdot a) = \text{group-action.orbit } G (\text{conjugation-action } (p \cdot a)) (\text{subgroups-of-size } (p \cdot a)) P \)

proof (auto)

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

have \text{group-action.orbit } G (\text{conjugation-action } (p \cdot a)) (\text{subgroups-of-size } (p \cdot a)) P = \text{group-action.same-orbit-rel } G (\text{conjugation-action } (p \cdot a)) (\text{subgroups-of-size} (p \cdot a)) \{ P \} by (rule \text{conj.orbit-def})

fix \( Q \)

\{ 

assume \( Q;Q \in \text{subgroups-of-size } (p \cdot a) \)

from \( Q \) \( P \) obtain \( g \) where \( g\in \text{carrier } G \) \( Q = g \triangleleft (P \triangleright inv g) \) by (rule sylow-conjugate)

with \( P \) \( \text{conj.orbit-char} \) show \( Q \in \text{group-action.orbit } G (\text{conjugation-action } (p \cdot a)) (\text{subgroups-of-size } (p \cdot a)) P \)

29
unfolding conjugation-action-def by auto

qed

theorem num-sylow-normalizer:
  assumes Ps: P ∈ subgroups-of-size (p ^ a)
  shows card (rcosets G[carrier := group-action.stabilizer G (conjugation-action (p ^ a)) P]) * p ^ a = card (group-action.stabilizer G (conjugation-action (p ^ a)) P)
proof --
  from finite-G interpret conj: group-action G (conjugation-action (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.size acts-on-subsets group-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 stabGroup: group G[carrier := conj.stabilizer P] by (metis conj.stabilizer-is-subgroup subgroup-imp-group)
  from finite-G Ps have PStab: subgroup P (G[carrier := conj.stabilizer P]) by (rule stabilizer-supergroup-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 PStab stabGroup.lagrange have card (rcosets G[carrier := conj.stabilizer P]) P * card P = order (G[carrier := conj.stabilizer P]) by force
  with cardP show thesis unfolding order-def by auto
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 (conjugation-action (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.size acts-on-subsets group-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)
with orderEq have \( p \cdot a \cdot m = \text{card} \ (\text{subgroups-of-size} \ (p \cdot a)) \cdot p \cdot a \cdot k \) \(\text{by}\) (auto \ simp:mult.assoc \ mult.commute)
  hence \( p \cdot a \cdot m = p \cdot a \cdot \text{card} \ (\text{subgroups-of-size} \ (p \cdot a)) \cdot k \) \(\text{by}\) auto
then have \( m = \text{card} \ (\text{subgroups-of-size} \ (p \cdot a)) \cdot k \)
  using \( p\cdot a\cdot k \) \(\text{by}\) auto
then show \( \text{thesis} \ ..\)
qed

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

lemma (in \( \text{snd-sylow} \)) \( \text{restrict-locale} \):
assumes \( \text{subgrp:subgroup \ G} \)
assumes \( \text{card:P \cdot a \ d\ v \ d \ P} \)
shows \( \text{snd-sylow} \ (G\{\text{carrier} := P\}]) \ p \ a \ (\text{(card} \ P) \ \text{div} \ (p \cdot a)) \)
proof –
  from \( \text{subgrp:subgroup \ G} \) have \( p \cdot a \cdot k \) \(\text{by}\) (metis \( \text{subgroup-imp-group} \))
  define \( k \) \where \( k = (\text{card} \ P) \ \text{div} \ (p \cdot a) \)
  with \( \text{card:P\cdot card:P} = p \cdot a \cdot k \) \(\text{by}\) auto
  hence \( \text{order:P:order} \ (G\{\text{carrier} := P\}]) = p \cdot a \cdot k \) \(\text{unfolding}\) \( \text{order-def} \) \(\text{by}\) simp
  from \( \text{card:P\cdot subgrp\ order-G} \) have \( p \cdot a \cdot k \) \(\text{dvd} \ p \cdot a \cdot m \) \(\text{by}\) (metis \( \text{card-subgrp-dvd} \))
  hence \( k \) \(\text{dvd} \ m \)
  by (metis \( \text{nat-mult-dvd-cancel-disj} \ \text{pa-not-zero} \))
with \( \text{pNotDvdM} \) have \( \text{ndvd:} p \ \text{dvd} \ k \)
  by (blast intro: \( \text{dvd-trans} \))
define \( P\text{calM} \) \(\text{where}\) \( P\text{calM} = \{s. s \subseteq \text{carrier} \ (G\{\text{carrier} := P\}]) \ \text{\wedge} \ \text{card} \ s = p \cdot a \} \)
define \( P\text{RelM} \) \(\text{where}\) \( P\text{RelM} = \{\{N1, N2\}. N1 \in P\text{calM} \ \wedge \ N2 \in P\text{calM} \ \wedge \ (\exists g \in \text{carrier} \ (G\{\text{carrier} := P\}]) \ \wedge \ N1 = N2 \ \#> G\{\text{carrier} := P\}]) 9\} \)
from \( \text{subgrp\ finite-G} \) have \( \text{finite-group:P:finite} \ (\text{carrier} \ (G\{\text{carrier} := P\}])) \) \(\text{by}\) (auto \ simp:subgroup-finite)
interpret \( \text{Nsylow: snd-sylow} \) \(G\{\text{carrier} := P\}]) \ p \ a \ k \ P\text{calM} \ P\text{RelM} \)
  unfolding \( \text{snd-sylow-def snd-sylow-axioms-def sylow-def sylow-axioms-def k-def} \)
  using \( \text{group:P\cdot is-group prime-p order:P\cdot finite-group P\cdot ndvd P\text{calM-def PRelM-def} k-def} \) \(\text{by}\) fastforce+
  show \( \text{thesis} \) using \( k\cdot d\ v\ d\) \(\text{by}\) (metis \( \text{Nsylow.is-snd-sylow} \))
qed

theorem (in \( \text{snd-sylow} \)) \( p\cdot sylow-mod-p: \)
shows \( \text{card} \ (\text{subgroups-of-size} \ (p \cdot a)) \mod p = 1 \)
proof –
  obtain \( P \) \(\text{where}\) \( \text{PG:subgroup \ P \ G \ \text{and} \ \text{card:P\cdot card:P} = p \cdot a} \) \(\text{by}\) (metis \( \text{sylow-thm} \))
  hence \( \text{order:P:order} \ (G\{\text{carrier} := P\}]) = p \cdot a \) \(\text{unfolding}\) \( \text{order-def} \) \(\text{by}\) auto
from \( PG \) have \( P\text{subG:P} \subseteq \text{carrier} \ G \) \(\text{by}\) (metis \( \text{subgroup-subset} \))
from \( PG \) \(\text{card:P} \) have \( P\text{Size:P} \in \text{subgroups-of-size} \ (p \cdot a) \) \(\text{unfolding}\) \( \text{subgroups-of-size-def} \) \(\text{by}\) auto
from \( PG \) interpret \( \text{group:P\cdot group} \ (G\{\text{carrier} := P\}]) \) \(\text{by}\) (rule \( \text{subgroup-imp-group} \))
from \( \text{card:P\cdot card:P} \) have \( P\text{Size2:P} \in \text{group:P\cdot subgroups-of-size} \ (p \cdot a) \) \(\text{using}\) \( \text{group:P\cdot subgroups-of-size-def} \)
\( \text{group:P} \cdot \text{subgroup-self} \) \(\text{by}\) 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 $G(\text{carrier} := P)$ 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 $\{ Q. \ Q \in \text{conjP.fixed-points} \implies Q = P \}$
proof
  fix $Q$
assume Qfixed: $Q \in \text{conjP.fixed-points}$
  hence $\forall Q. \ Q \in \text{subgroups-of-size} (p \cdot a)$ unfolding conjP.fixed-points-def
by simp
  hence cardQ: card $Q = p \cdot a$ unfolding subgroups-of-size-def by simp
— The normalization of $Q$ in $G$
— Let’s first show some basic properties of $N$
define N where $N = \text{conjG.stabilizer} \ Q$
define k where $k = (\text{card} \ N) \ \text{div} \ (p \cdot a)$
from N-def Qsize 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 Qsize N-def have QN subgroup $Q \ (G(\text{carrier} := N))$ using stabilizer-supergp-P
by auto
— The following proposition is used to show that $P = Q$ later
  from Qsize have NfixedQ: $\forall g \in N. \ \text{conjugation-action} \ (p \cdot a) \ g \ Q = Q$
unfolding N-def conjG.stabilizer-def by auto
from Qfixed have PfixedQ: $\forall g \in P. \ \text{conjugation-action} \ (p \cdot a) \ g \ Q = Q$
unfolding conjP.fixed-points-def conjP.stabilizer-def by auto
with PsizeG have $P \subseteq N$ unfolding N-def conjG.stabilizer-def by auto
with PG N-def Qsize have PN subgroup $P \ (G(\text{carrier} := N))$ by (metis conjG.stabilizer-is-subgroup is-group subgroup subgroup subgroup-of-subset)
with cardP have $p \cdot a \ \text{dvd order} \ (G(\text{carrier} := N))$ using groupN.card-subgrp-dvd
by force
  hence $p \cdot a \ \text{dvd card} \ N$ unfolding order-def by simp
with NG have smaller-sylow: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 NcalM where $\text{NcalM} = \{ s. \ s \subseteq \text{carrier} \ (G(\text{carrier} := N)) \ \land \ \text{card} \ s = p \cdot a \}$
define NRrelM where $\text{NRrelM} = \{ (N1, N2). \ N1 \in \text{NcalM} \ \land \ N2 \in \text{NcalM} \ \land \ (\exists g \in \text{carrier} \ (G(\text{carrier} := N)). \ N1 = N2 \ #> G(\text{carrier} := N) \ g) \}$
interpret Nsylow: snd-sylow $G(\text{carrier} := N)$ $p \ a \ k \ \text{NcalM \ NRrelM}$ unfolding NcalM-def NRrelM-def using smaller-sylow.
— $P$ and $Q$ are conjugate in $N$:
from cardP PN have PsizeN: $P \in \text{groupN.subgroups-of-size} (p \cdot a)$ unfolding groupN.subgroups-of-size-def by auto
from cardQ QN have QsizeN: $Q \in \text{groupN.subgroups-of-size} (p \cdot a)$ unfold-
ing groupN.subgroups-of-size-def by auto
from Q.sizeN P.sizeN 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 Q.size have conjugation-action (p ^ a) g Q = P unfolding
conjugation-action-def using subgroup.subset by force
  with g N.fixesQ show Q = P by auto
  qed
moreover from finite-G P.size 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