

A Generalization of the Cauchy–Davenport Theorem

Mantas Bakšys
University of Cambridge
mb2412@cam.ac.uk

March 10, 2025

Abstract

The Cauchy–Davenport theorem is a fundamental result in additive combinatorics. It was originally independently discovered by Cauchy [2] and Davenport [3] and has been formalized in the AFP entry [1] as a corollary of Kneser’s theorem. More recently, many generalizations of this theorem have been found. In this entry, we formalise a generalization due to DeVos [4], which proves the theorem in a non-abelian setting.

Contents

1 Preliminaries on well-orderings, groups, and sumsets	3
1.1 Well-ordering lemmas	3
1.2 Pointwise set multiplication in a monoid: definition and key lemmas	3
1.3 Exponentiation in a monoid: definitions and lemmas	5
1.4 Definition of the order of an element in a monoid	8
1.5 Sumset scalar multiplication cardinality lemmas	8
1.6 Pointwise set multiplication in a group: auxiliary lemmas	9
1.7 <i>ecard</i> – extended definition of cardinality of a set	10
2 Generalized Cauchy–Davenport theorem: main proof	10
2.1 The counterexample pair relation in [4]	10
2.2 $p(G)$ – the order of the smallest nontrivial finite subgroup of a group: definition and lemmas	11
2.3 Proof of the generalized Cauchy–Davenport theorem for (non-abelian) groups	11

1 Preliminaries on well-orderings, groups, and sum-sets

theory *Generalized-Cauchy-Davenport-preliminaries*
imports
 Complex-Main
 Jacobson-Basic-Algebra.Group-Theory
 HOL-Library.Extended-Nat

begin

1.1 Well-ordering lemmas

lemma *wf-prod-lex-fibers-inter*:

fixes $r :: ('a \times 'a)$ set **and** $s :: ('b \times 'b)$ set **and** $f :: 'c \Rightarrow 'a$ **and** $g :: 'c \Rightarrow 'b$
and
 $t :: ('c \times 'c)$ set
assumes $h1: wf ((inv-image\ r\ f) \cap t)$ **and**
 $h2: \bigwedge a. a \in range\ f \implies wf (\{x. f\ x = a\} \times \{x. f\ x = a\} \cap (inv-image\ s\ g))$
 $\cap t)$ **and**
 $h3: trans\ t$
shows $wf ((inv-image\ (r\ <*\lex*\>\ s)\ (\lambda\ c. (f\ c, g\ c))) \cap t)$
<proof>

lemma *wf-prod-lex-fibers*:

fixes $r :: ('a \times 'a)$ set **and** $s :: ('b \times 'b)$ set **and** $f :: 'c \Rightarrow 'a$ **and** $g :: 'c \Rightarrow 'b$
assumes $h1: wf (inv-image\ r\ f)$ **and**
 $h2: \bigwedge a. a \in range\ f \implies wf (\{x. f\ x = a\} \times \{x. f\ x = a\} \cap (inv-image\ s\ g))$
shows $wf (inv-image\ (r\ <*\lex*\>\ s)\ (\lambda\ c. (f\ c, g\ c)))$
<proof>

context *monoid*

begin

1.2 Pointwise set multiplication in a monoid: definition and key lemmas

inductive-set $smul :: 'a\ set \Rightarrow 'a\ set \Rightarrow 'a\ set$ **for** $A\ B$

where

$smul[intro]: \llbracket a \in A; a \in M; b \in B; b \in M \rrbracket \implies a \cdot b \in smul\ A\ B$

syntax $smul :: 'a\ set \Rightarrow 'a\ set \Rightarrow 'a\ set$ $((-\ \cdots -))$

lemma $smul\ eq: smul\ A\ B = \{c. \exists a \in A \cap M. \exists b \in B \cap M. c = a \cdot b\}$
<proof>

lemma $smul: smul\ A\ B = (\bigcup a \in A \cap M. \bigcup b \in B \cap M. \{a \cdot b\})$
<proof>

lemma *smul-subset-carrier*: $smul\ A\ B \subseteq M$

<proof>

lemma *smul-Int-carrier* [*simp*]: $smul\ A\ B \cap M = smul\ A\ B$

<proof>

lemma *smul-mono*: $\llbracket A' \subseteq A; B' \subseteq B \rrbracket \implies smul\ A'\ B' \subseteq smul\ A\ B$

<proof>

lemma *smul-insert1*: *NO-MATCH* $\{\} A \implies smul\ (insert\ x\ A)\ B = smul\ \{x\}\ B \cup smul\ A\ B$

<proof>

lemma *smul-insert2*: *NO-MATCH* $\{\} B \implies smul\ A\ (insert\ x\ B) = smul\ A\ \{x\} \cup smul\ A\ B$

<proof>

lemma *smul-subset-Un1*: $smul\ (A \cup A')\ B = smul\ A\ B \cup smul\ A'\ B$

<proof>

lemma *smul-subset-Un2*: $smul\ A\ (B \cup B') = smul\ A\ B \cup smul\ A\ B'$

<proof>

lemma *smul-subset-Union1*: $smul\ (\bigcup A)\ B = (\bigcup a \in A. smul\ a\ B)$

<proof>

lemma *smul-subset-Union2*: $smul\ A\ (\bigcup B) = (\bigcup b \in B. smul\ A\ b)$

<proof>

lemma *smul-subset-insert*: $smul\ A\ B \subseteq smul\ A\ (insert\ x\ B) \wedge smul\ A\ B \subseteq smul\ (insert\ x\ A)\ B$

<proof>

lemma *smul-subset-Un*: $smul\ A\ B \subseteq smul\ A\ (B \cup C) \wedge smul\ A\ B \subseteq smul\ (A \cup C)\ B$

<proof>

lemma *smul-empty* [*simp*]: $smul\ A\ \{\} = \{\} \wedge smul\ \{\} A = \{\}$

<proof>

lemma *smul-empty'*:

assumes $A \cap M = \{\}$

shows $smul\ B\ A = \{\} \wedge smul\ A\ B = \{\}$

<proof>

lemma *smul-is-empty-iff* [*simp*]: $smul\ A\ B = \{\} \iff A \cap M = \{\} \vee B \cap M = \{\}$

<proof>

lemma *smul-D* [*simp*]: $smul\ A\ \{\mathbf{1}\} = A \cap M\ smul\ \{\mathbf{1}\}\ A = A \cap M$
<proof>

lemma *smul-Int-carrier-eq* [*simp*]: $smul\ A\ (B \cap M) = smul\ A\ B\ smul\ (A \cap M)\ B$
 $= smul\ A\ B$
<proof>

lemma *smul-assoc*:
shows $smul\ (smul\ A\ B)\ C = smul\ A\ (smul\ B\ C)$
<proof>

lemma *finite-smul*:
assumes *finite* A *finite* B **shows** *finite* $(smul\ A\ B)$
<proof>

lemma *finite-smul'*:
assumes *finite* $(A \cap M)$ *finite* $(B \cap M)$
shows *finite* $(smul\ A\ B)$
<proof>

1.3 Exponentiation in a monoid: definitions and lemmas

primrec *power* :: 'a \Rightarrow nat \Rightarrow 'a (*infix* $\hat{\ } 100$)
where
power0: $power\ g\ 0 = \mathbf{1}$
| *power-suc*: $power\ g\ (Suc\ n) = power\ g\ n \cdot g$

lemma *power-one*:
assumes $g \in M$
shows $power\ g\ 1 = g$ *<proof>*

lemma *power-mem-carrier*:
fixes n
assumes $g \in M$
shows $g \hat{\ } n \in M$
<proof>

lemma *power-mult*:
assumes $g \in M$
shows $g \hat{\ } n \cdot g \hat{\ } m = g \hat{\ } (n + m)$
<proof>

lemma *mult-inverse-power*:
assumes $g \in M$ **and** *invertible* g
shows $g \hat{\ } n \cdot ((inverse\ g) \hat{\ } n) = \mathbf{1}$
<proof>

lemma *inverse-mult-power*:
assumes $g \in M$ **and** *invertible* g

shows $((\text{inverse } g) \hat{\ } n) \cdot g \hat{\ } n = \mathbf{1}$ $\langle \text{proof} \rangle$

lemma *inverse-mult-power-eq*:

assumes $g \in M$ **and** *invertible* g

shows $\text{inverse } (g \hat{\ } n) = (\text{inverse } g) \hat{\ } n$

$\langle \text{proof} \rangle$

definition *power-int* :: $'a \Rightarrow \text{int} \Rightarrow 'a$ (**infixr** *powi* 80) **where**

$\text{power-int } g \ n = (\text{if } n \geq 0 \text{ then } g \hat{\ } (\text{nat } n) \text{ else } (\text{inverse } g) \hat{\ } (\text{nat } (-n)))$

definition *nat-powers* :: $'a \Rightarrow 'a$ **set** **where** $\text{nat-powers } g = ((\lambda n. g \hat{\ } n) \text{ ` UNIV})$

lemma *nat-powers-eq-Union*: $\text{nat-powers } g = (\bigcup n. \{g \hat{\ } n\})$ $\langle \text{proof} \rangle$

definition *powers* :: $'a \Rightarrow 'a$ **set** **where** $\text{powers } g = ((\lambda n. g \text{ powi } n) \text{ ` UNIV})$

lemma *nat-powers-subset*:

$\text{nat-powers } g \subseteq \text{powers } g$

$\langle \text{proof} \rangle$

lemma *inverse-nat-powers-subset*:

$\text{nat-powers } (\text{inverse } g) \subseteq \text{powers } g$

$\langle \text{proof} \rangle$

lemma *powers-eq-union-nat-powers*:

$\text{powers } g = \text{nat-powers } g \cup \text{nat-powers } (\text{inverse } g)$

$\langle \text{proof} \rangle$

lemma *one-mem-nat-powers*: $\mathbf{1} \in \text{nat-powers } g$

$\langle \text{proof} \rangle$

lemma *nat-powers-subset-carrier*:

assumes $g \in M$

shows $\text{nat-powers } g \subseteq M$

$\langle \text{proof} \rangle$

lemma *nat-powers-mult-closed*:

assumes $g \in M$

shows $\bigwedge x \ y. x \in \text{nat-powers } g \Longrightarrow y \in \text{nat-powers } g \Longrightarrow x \cdot y \in \text{nat-powers } g$

$\langle \text{proof} \rangle$

lemma *nat-powers-inv-mult*:

assumes $g \in M$ **and** *invertible* g

shows $\bigwedge x \ y. x \in \text{nat-powers } g \Longrightarrow y \in \text{nat-powers } (\text{inverse } g) \Longrightarrow x \cdot y \in \text{powers } g$

$\langle \text{proof} \rangle$

lemma *inv-nat-powers-mult*:

assumes $g \in M$ **and** *invertible* g

shows $\bigwedge x y. x \in \text{nat-powers } (\text{inverse } g) \implies y \in \text{nat-powers } g \implies x \cdot y \in \text{powers } g$
<proof>

lemma *powers-mult-closed*:

assumes $g \in M$ **and** *invertible* g

shows $\bigwedge x y. x \in \text{powers } g \implies y \in \text{powers } g \implies x \cdot y \in \text{powers } g$

<proof>

lemma *nat-powers-submonoid*:

assumes $g \in M$

shows *submonoid* $(\text{nat-powers } g) M (\cdot) \mathbf{1}$

<proof>

lemma *nat-powers-monoid*:

assumes $g \in M$

shows *Group-Theory.monoid* $(\text{nat-powers } g) (\cdot) \mathbf{1}$

<proof>

lemma *powers-submonoid*:

assumes $g \in M$ **and** *invertible* g

shows *submonoid* $(\text{powers } g) M (\cdot) \mathbf{1}$

<proof>

lemma *powers-monoid*:

assumes $g \in M$ **and** *invertible* g

shows *Group-Theory.monoid* $(\text{powers } g) (\cdot) \mathbf{1}$

<proof>

lemma *mem-nat-powers-invertible*:

assumes $g \in M$ **and** *invertible* g **and** $u \in \text{nat-powers } g$

shows *monoid.invertible* $(\text{powers } g) (\cdot) \mathbf{1} u$

<proof>

lemma *mem-nat-inv-powers-invertible*:

assumes $g \in M$ **and** *invertible* g **and** $u \in \text{nat-powers } (\text{inverse } g)$

shows *monoid.invertible* $(\text{powers } g) (\cdot) \mathbf{1} u$

<proof>

lemma *powers-group*:

assumes $g \in M$ **and** *invertible* g

shows *Group-Theory.group* $(\text{powers } g) (\cdot) \mathbf{1}$

<proof>

lemma *nat-powers-ne-one*:

assumes $g \in M$ **and** $g \neq \mathbf{1}$

shows $\text{nat-powers } g \neq \{\mathbf{1}\}$

<proof>

lemma *powers-ne-one*:
assumes $g \in M$ **and** $g \neq \mathbf{1}$
shows $\text{powers } g \neq \{\mathbf{1}\}$ *<proof>*

end

context *group*

begin

lemma *powers-subgroup*:
assumes $g \in G$
shows $\text{subgroup } (\text{powers } g) \ G \ (\cdot) \ \mathbf{1}$
<proof>

end

context *monoid*

begin

1.4 Definition of the order of an element in a monoid

definition *order*

where $\text{order } g = (\text{if } (\exists n. n > 0 \wedge g \wedge n = \mathbf{1}) \text{ then } \text{Min } \{n. g \wedge n = \mathbf{1} \wedge n > 0\} \text{ else } \infty)$

definition *min-order* **where** $\text{min-order} = \text{Min } ((\text{order } 'M) - \{0\})$

end

1.5 Sumset scalar multiplication cardinality lemmas

context *group*

begin

lemma *card-smul-singleton-right-eq*:

assumes *finite* A **shows** $\text{card } (\text{smul } A \ \{a\}) = (\text{if } a \in G \text{ then } \text{card } (A \cap G) \text{ else } 0)$
<proof>

lemma *card-smul-singleton-left-eq*:

assumes *finite* A **shows** $\text{card } (\text{smul } \{a\} \ A) = (\text{if } a \in G \text{ then } \text{card } (A \cap G) \text{ else } 0)$
<proof>

lemma *card-smul-sing-right-le*:

assumes *finite* A **shows** $\text{card } (\text{smul } A \ \{a\}) \leq \text{card } A$
<proof>

lemma *card-smul-sing-left-le*:
assumes *finite A* **shows** $\text{card } (\text{smul } \{a\} A) \leq \text{card } A$
 $\langle \text{proof} \rangle$

lemma *card-le-smul-right*:
assumes *A: finite A a ∈ A a ∈ G*
and *B: finite B B ⊆ G*
shows $\text{card } B \leq \text{card } (\text{smul } A B)$
 $\langle \text{proof} \rangle$

lemma *card-le-smul-left*:
assumes *A: finite A b ∈ B b ∈ G*
and *B: finite B A ⊆ G*
shows $\text{card } A \leq \text{card } (\text{smul } A B)$
 $\langle \text{proof} \rangle$

lemma *infinite-smul-right*:
assumes $A \cap G \neq \{\}$ **and** *infinite (B ∩ G)*
shows *infinite (A ··· B)*
 $\langle \text{proof} \rangle$

lemma *infinite-smul-left*:
assumes $B \cap G \neq \{\}$ **and** *infinite (A ∩ G)*
shows *infinite (A ··· B)*
 $\langle \text{proof} \rangle$

1.6 Pointwise set multiplication in a group: auxiliary lemmas

lemma *set-inverse-composition-commute*:
assumes $X \subseteq G$ **and** $Y \subseteq G$
shows $\text{inverse } '(X \cdots Y) = (\text{inverse } ' Y) \cdots (\text{inverse } ' X)$
 $\langle \text{proof} \rangle$

lemma *smul-singleton-eq-contains-nat-powers*:
fixes $n :: \text{nat}$
assumes $X \subseteq G$ **and** $g \in G$ **and** $X \cdots \{g\} = X$
shows $X \cdots \{g \wedge n\} = X$
 $\langle \text{proof} \rangle$

lemma *smul-singleton-eq-contains-inverse-nat-powers*:
fixes $n :: \text{nat}$
assumes $X \subseteq G$ **and** $g \in G$ **and** $X \cdots \{g\} = X$
shows $X \cdots \{(\text{inverse } g) \wedge n\} = X$
 $\langle \text{proof} \rangle$

lemma *smul-singleton-eq-contains-powers*:
fixes $n :: \text{nat}$

assumes $X \subseteq G$ **and** $g \in G$ **and** $X \cdots \{g\} = X$
shows $X \cdots (\text{powers } g) = X$ $\langle \text{proof} \rangle$

end

1.7 *ecard* – extended definition of cardinality of a set

ecard – definition of a cardinality of a set taking values in *enat* – extended natural numbers, defined to be ∞ for infinite sets

definition *ecard* **where** $ecard\ A = (\text{if } \text{finite } A \text{ then } \text{card } A \text{ else } \infty)$

lemma *ecard-eq-card-finite*:

assumes *finite* A
shows $ecard\ A = \text{card } A$
 $\langle \text{proof} \rangle$

context *monoid*

begin

orderOf – abbreviation for the order of a monoid

abbreviation *orderOf* **where** $orderOf == ecard$

end

end

2 Generalized Cauchy–Davenport theorem: main proof

theory *Generalized-Cauchy-Davenport-main-proof*

imports *Generalized-Cauchy-Davenport-preliminaries*

begin

context *group*

begin

2.1 The counterexample pair relation in [4]

definition *devos-rel* **where**

$devos-rel = (\lambda\ (A, B). \text{card}(A \cdots B)) <*\text{mlex}*> (\text{inv-image } (\{(n, m). n > m\}$
 $<*\text{llex}*>$
 $\text{measure } (\lambda\ (A, B). \text{card } A))) (\lambda\ (A, B). (\text{card } A + \text{card } B, (A, B)))$

lemma *devos-rel-iff*:

$((A, B), (C, D)) \in devos-rel \iff \text{card}(A \cdots B) < \text{card}(C \cdots D) \vee$

$(\text{card}(A \cdots B) = \text{card}(C \cdots D) \wedge \text{card } A + \text{card } B > \text{card } C + \text{card } D) \vee$
 $(\text{card}(A \cdots B) = \text{card}(C \cdots D) \wedge \text{card } A + \text{card } B = \text{card } C + \text{card } D \wedge \text{card}$
 $A < \text{card } C)$
 ⟨proof⟩

lemma *devos-rel-le-smul*:

$((A, B), (C, D)) \in \text{devos-rel} \implies \text{card}(A \cdots B) \leq \text{card}(C \cdots D)$
 ⟨proof⟩

Lemma stating that the above relation due to DeVos is well-founded

lemma *devos-rel-wf* : *wf (Restr devos-rel*

$\{(A, B). \text{finite } A \wedge A \neq \{\} \wedge A \subseteq G \wedge \text{finite } B \wedge B \neq \{\} \wedge B \subseteq G\}$ (is *wf*
 $(\text{Restr devos-rel } ?\text{fin})$)
 ⟨proof⟩

2.2 $p(G)$ – the order of the smallest nontrivial finite subgroup of a group: definition and lemmas

$p(G)$ – the size of the smallest nontrivial finite subgroup of G , set to ∞ if none exist

definition $p :: \text{enat}$ **where** $p = \text{Inf } (\text{orderOf } \{H. \text{subgroup } H \ G \ (\cdot) \ \mathbf{1} \wedge H \neq \{\mathbf{1}\}\})$

lemma *subgroup-finite-ge*:

assumes *subgroup* $H \ G \ (\cdot) \ \mathbf{1}$ **and** $H \neq \{\mathbf{1}\}$ **and** *finite* H
shows $\text{card } H \geq p$
 ⟨proof⟩

lemma *subgroup-infinite-or-card-ge*:

assumes *subgroup* $H \ G \ (\cdot) \ \mathbf{1}$ **and** $H \neq \{\mathbf{1}\}$
shows $\text{infinite } H \vee \text{card } H \geq p$ ⟨proof⟩

end

2.3 Proof of the generalized Cauchy–Davenport theorem for (non-abelian) groups

Generalized Cauchy–Davenport theorem for (non-abelian) groups due to Matt DeVos [4]

theorem (in *group*) *Generalized-Cauchy-Davenport*:

assumes $hAne: A \neq \{\}$ **and** $hBne: B \neq \{\}$ **and** $hAG: A \subseteq G$ **and** $hBG: B \subseteq G$ **and**
 $hAfin: \text{finite } A$ **and** $hBfin: \text{finite } B$
shows $\text{card } (A \cdots B) \geq \min p (\text{card } A + \text{card } B - 1)$
 ⟨proof⟩

end

References

- [1] M. Bakšys and A. Koutsoukou-Argyaki. Kneser’s Theorem and the Cauchy–Davenport Theorem. *Archive of Formal Proofs*, November 2022. https://isa-afp.org/entries/Kneser_Cauchy_Davenport.html, Formal proof development.
- [2] A. L. B. Cauchy. Recherches sur les nombres. *J. École Polytech.*, 9:99–116, 1813.
- [3] H. Davenport. On the Addition of Residue Classes. *Journal of the London Mathematical Society*, s1-10(1):30–32, 01 1935.
- [4] M. DeVos. On a Generalization of the Cauchy–Davenport Theorem. *Integers*, 16:A7, 2016.