# A Generalization of the Cauchy–Davenport Theorem

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

December 27, 2024

**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

**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*: $wf\ ((inv\text{-}image\ r\ f) \cap t)$ **and**
*h2*: $\bigwedge a.\ a \in range\ f \Longrightarrow wf\ ((\{x.\ f\ x = a\} \times \{x.\ f\ x = a\} \cap (inv\text{-}image\ s\ g)) \cap t)$ **and**
*h3*: $trans\ t$
**shows** $wf\ ((inv\text{-}image\ (r <*lex*> s)\ (\lambda\ c.\ (f\ c,\ g\ c))) \cap t)$
⟨*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*: $wf\ (inv\text{-}image\ r\ f)$ **and**
*h2*: $\bigwedge a.\ a \in range\ f \Longrightarrow wf\ (\{x.\ f\ x = a\} \times \{x.\ f\ x = a\} \cap (inv\text{-}image\ s\ g))$
**shows** $wf\ (inv\text{-}image\ (r <*lex*> s)\ (\lambda\ c.\ (f\ c,\ g\ c)))$
⟨*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**
$smulI[intro]: [\![a \in A;\ a \in M;\ b \in B;\ b \in M]\!] \Longrightarrow a \cdot b \in smul\ A\ B$

**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*: $[\![A' \subseteq A;\ B' \subseteq B]\!] \Longrightarrow$ *smul A' B'* $\subseteq$ *smul A B*
⟨*proof*⟩

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

**lemma** *smul-insert2*: *NO-MATCH* {} *B* $\Longrightarrow$ *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*) *smul A B* $\subseteq$ *smul* (*insert x A*) *B*
⟨*proof*⟩

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

**lemma** *smul-empty* [*simp*]: *smul A* {} = {} *smul* {} *A* = {}
⟨*proof*⟩

**lemma** *smul-empty'*:
**assumes** $A \cap M$ = {}
**shows** *smul B A* = {} *smul A B* = {}
⟨*proof*⟩

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

**lemma** *smul-D* [*simp*]: *smul A* {**1**} = *A* ∩ *M smul* {**1**} *A* = *A* ∩ *M*
⟨*proof*⟩

**lemma** *smul-Int-carrier-eq* [*simp*]: *smul A* (*B* ∩ *M*) = *smul A B smul* (*A* ∩ *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* ∩ *M*) *finite* (*B* ∩ *M*)
**shows** *finite* (*smul A B*)
⟨*proof*⟩

## 1.3 Exponentiation in a monoid: definitions and lemmas

**primrec** *power* :: ′*a* ⇒ *nat* ⇒ ′*a* (**infix** ^ *100*)
**where**
*power0*: *power g 0* = **1**
| *power-suc*: *power g* (*Suc n*) = *power g n* · *g*

**lemma** *power-one*:
**assumes** *g* ∈ *M*
**shows** *power g 1* = *g* ⟨*proof*⟩

**lemma** *power-mem-carrier*:
**fixes** *n*
**assumes** *g* ∈ *M*
**shows** *g* ^ *n* ∈ *M*
⟨*proof*⟩

**lemma** *power-mult*:
**assumes** *g* ∈ *M*
**shows** *g* ^ *n* · *g* ^ *m* = *g* ^ (*n* + *m*)
⟨*proof*⟩

**lemma** *mult-inverse-power*:
**assumes** *g* ∈ *M* **and** *invertible g*
**shows** *g* ^ *n* · ((*inverse g*) ^ *n*) = **1**
⟨*proof*⟩

**lemma** *inverse-mult-power*:
**assumes** *g* ∈ *M* **and** *invertible g*

**shows** $((inverse\ g)\ \widehat{}\ n) \cdot g\ \widehat{}\ n = \mathbf{1}$ ⟨*proof*⟩

**lemma** *inverse-mult-power-eq*:
**assumes** $g \in M$ **and** *invertible g*
**shows** $inverse\ (g\ \widehat{}\ n) = (inverse\ g)\ \widehat{}\ n$
⟨*proof*⟩

**definition** *power-int* :: $'a \Rightarrow int \Rightarrow 'a$ (**infixr** *powi 80*) **where**
$power\text{-}int\ g\ n = (if\ n \geq 0\ then\ g\ \widehat{}(nat\ n)\ else\ (inverse\ g)\ \widehat{}(nat\ (-n)))$

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

**lemma** *nat-powers-eq-Union*: $nat\text{-}powers\ g = (\bigcup\ n.\ \{g\ \widehat{}\ n\})$ ⟨*proof*⟩

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

**lemma** *nat-powers-subset*:
$nat\text{-}powers\ g \subseteq powers\ g$
⟨*proof*⟩

**lemma** *inverse-nat-powers-subset*:
$nat\text{-}powers\ (inverse\ g) \subseteq powers\ g$
⟨*proof*⟩

**lemma** *powers-eq-union-nat-powers*:
$powers\ g = nat\text{-}powers\ g \cup nat\text{-}powers\ (inverse\ g)$
⟨*proof*⟩

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

**lemma** *nat-powers-subset-carrier*:
**assumes** $g \in M$
**shows** $nat\text{-}powers\ g \subseteq M$
⟨*proof*⟩

**lemma** *nat-powers-mult-closed*:
**assumes** $g \in M$
**shows** $\bigwedge\ x\ y.\ x \in nat\text{-}powers\ g \Longrightarrow y \in nat\text{-}powers\ g \Longrightarrow x \cdot y \in nat\text{-}powers\ g$
⟨*proof*⟩

**lemma** *nat-powers-inv-mult*:
**assumes** $g \in M$ **and** *invertible g*
**shows** $\bigwedge\ x\ y.\ x \in nat\text{-}powers\ g \Longrightarrow y \in nat\text{-}powers\ (inverse\ g) \Longrightarrow x \cdot y \in powers\ g$
⟨*proof*⟩

**lemma** *inv-nat-powers-mult*:
**assumes** $g \in M$ **and** *invertible g*

**shows** $\bigwedge x\ y.\ x \in$ *nat-powers* (*inverse* $g$) $\Longrightarrow y \in$ *nat-powers* $g \Longrightarrow x \cdot y \in$ *powers* $g$
⟨*proof*⟩

**lemma** *powers-mult-closed*:
**assumes** $g \in M$ **and** *invertible* $g$
**shows** $\bigwedge x\ y.\ x \in$ *powers* $g \Longrightarrow y \in$ *powers* $g \Longrightarrow x \cdot y \in$ *powers* $g$
⟨*proof*⟩

**lemma** *nat-powers-submonoid*:
**assumes** $g \in M$
**shows** *submonoid* (*nat-powers* $g$) $M$ $(\cdot)$ **1**
⟨*proof*⟩

**lemma** *nat-powers-monoid*:
**assumes** $g \in M$
**shows** *Group-Theory.monoid* (*nat-powers* $g$) $(\cdot)$ **1**
⟨*proof*⟩

**lemma** *powers-submonoid*:
**assumes** $g \in M$ **and** *invertible* $g$
**shows** *submonoid* (*powers* $g$) $M$ $(\cdot)$ **1**
⟨*proof*⟩

**lemma** *powers-monoid*:
**assumes** $g \in M$ **and** *invertible* $g$
**shows** *Group-Theory.monoid* (*powers* $g$) $(\cdot)$ **1**
⟨*proof*⟩

**lemma** *mem-nat-powers-invertible*:
**assumes** $g \in M$ **and** *invertible* $g$ **and** $u \in$ *nat-powers* $g$
**shows** *monoid.invertible* (*powers* $g$) $(\cdot)$ **1** $u$
⟨*proof*⟩

**lemma** *mem-nat-inv-powers-invertible*:
**assumes** $g \in M$ **and** *invertible* $g$ **and** $u \in$ *nat-powers* (*inverse* $g$)
**shows** *monoid.invertible* (*powers* $g$) $(\cdot)$ **1** $u$
⟨*proof*⟩

**lemma** *powers-group*:
**assumes** $g \in M$ **and** *invertible* $g$
**shows** *Group-Theory.group* (*powers* $g$) $(\cdot)$ **1**
⟨*proof*⟩

**lemma** *nat-powers-ne-one*:
**assumes** $g \in M$ **and** $g \neq \mathbf{1}$
**shows** *nat-powers* $g \neq \{\mathbf{1}\}$
⟨*proof*⟩

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

**end**

**context** *group*

**begin**

**lemma** *powers-subgroup*:
**assumes** $g \in G$
**shows** *subgroup* (*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** *order* $g = ($*if* $(\exists\ n.\ n > 0 \wedge g \,\hat{}\, n = \mathbf{1})$ *then* *Min* $\{n.\ g \,\hat{}\, n = \mathbf{1} \wedge n > 0\}$ *else* $\infty)$

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

**end**

## 1.5 Sumset scalar multiplication cardinality lemmas

**context** *group*

**begin**

**lemma** *card-smul-singleton-right-eq*:
**assumes** *finite* $A$ **shows** *card* (*smul* $A$ $\{a\}$) = (*if* $a \in G$ *then* *card* $(A \cap G)$ *else* $0$)
⟨*proof*⟩

**lemma** *card-smul-singleton-left-eq*:
**assumes** *finite* $A$ **shows** *card* (*smul* $\{a\}$ $A$) = (*if* $a \in G$ *then* *card* $(A \cap G)$ *else* $0$)
⟨*proof*⟩

**lemma** *card-smul-sing-right-le*:
**assumes** *finite* $A$ **shows** *card* (*smul* $A$ $\{a\}$) $\leq$ *card* $A$
⟨*proof*⟩

**lemma** *card-smul-sing-left-le*:
**assumes** *finite A* **shows** *card* (*smul* {*a*} *A*) ≤ *card A*
⟨*proof*⟩

**lemma** *card-le-smul-right*:
**assumes** *A*: *finite A a* ∈ *A a* ∈ *G*
**and** *B*: *finite B B* ⊆ *G*
**shows** *card B* ≤ *card* (*smul A B*)
⟨*proof*⟩

**lemma** *card-le-smul-left*:
**assumes** *A*: *finite A b* ∈ *B b* ∈ *G*
**and** *B*: *finite B A* ⊆ *G*
**shows** *card A* ≤ *card* (*smul A B*)
⟨*proof*⟩

**lemma** *infinite-smul-right*:
**assumes** *A* ∩ *G* ≠ {} **and** *infinite* (*B* ∩ *G*)
**shows** *infinite* (*A* ··· *B*)
⟨*proof*⟩

**lemma** *infinite-smul-left*:
**assumes** *B* ∩ *G* ≠ {} **and** *infinite* (*A* ∩ *G*)
**shows** *infinite* (*A* ··· *B*)
⟨*proof*⟩

## 1.6 Pointwise set multiplication in a group: auxiliary lemmas

**lemma** *set-inverse-composition-commute*:
**assumes** *X* ⊆ *G* **and** *Y* ⊆ *G*
**shows** *inverse* ‘ (*X* ··· *Y*) = (*inverse* ‘ *Y*) ··· (*inverse* ‘ *X*)
⟨*proof*⟩

**lemma** *smul-singleton-eq-contains-nat-powers*:
**fixes** *n* :: *nat*
**assumes** *X* ⊆ *G* **and** *g* ∈ *G* **and** *X* ··· {*g*} = *X*
**shows** *X* ··· {*g* ⌢ *n*} = *X*
⟨*proof*⟩

**lemma** *smul-singleton-eq-contains-inverse-nat-powers*:
**fixes** *n* :: *nat*
**assumes** *X* ⊆ *G* **and** *g* ∈ *G* **and** *X* ··· {*g*} = *X*
**shows** *X* ··· {(*inverse g*) ⌢ *n*} = *X*
⟨*proof*⟩

**lemma** *smul-singleton-eq-contains-powers*:
**fixes** *n* :: *nat*

**assumes** $X \subseteq G$ **and** $g \in G$ **and** $X \cdots \{g\} = X$
**shows** $X \cdots (powers\ g) = X$ ⟨*proof*⟩

**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 $\infty$ for infinite sets

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

**lemma** *ecard-eq-card-finite*:
**assumes** *finite A*
**shows** *ecard A* = *card A*
⟨*proof*⟩

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

**lemma** *devos-rel-iff*:
((*A*, *B*), (*C*, *D*)) $\in$ *devos-rel* $\longleftrightarrow$ *card*(*A* $\cdots$ *B*) < *card*(*C* $\cdots$ *D*) $\vee$

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

**lemma** *devos-rel-le-smul*:
$((A, B), (C, D)) \in devos\text{-}rel \Longrightarrow card(A \cdots B) \leq card(C \cdots D)$
⟨*proof*⟩

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

**lemma** *devos-rel-wf* : *wf* (*Restr devos-rel*
$\{(A, B).\ finite\ A \wedge A \neq \{\} \wedge A \subseteq G \wedge finite\ B \wedge B \neq \{\} \wedge B \subseteq G\}$) (**is** *wf* (*Restr devos-rel ?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 $\infty$ if none exist

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

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

**lemma** *subgroup-infinite-or-card-ge*:
**assumes** *subgroup H G* $(\cdot)$ **1 and** $H \neq \{\mathbf{1}\}$
**shows** $infinite\ H \vee card\ H \geq p$ ⟨*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*: *finite A* **and** *hBfin*: *finite B*
**shows** $card\ (A \cdots B) \geq min\ p\ (card\ A + card\ B - 1)$
⟨*proof*⟩

**end**

## References

[1] M. Bakšys and A. Koutsoukou-Argyraki. 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.