The Plünnecke-Ruzsa Inequality

Angeliki Koutsoukou-Argyraki and Lawrence C. Paulson

March 17, 2025

Abstract

We formalise Plünnecke's inequality and the Plünnecke-Ruzsa inequality, following the notes by Timothy Gowers: "Introduction to Additive Combinatorics" (2022) for the University of Cambridge. To this end, we first introduce basic definitions and prove elementary facts on sumsets and difference sets. Then, we show two versions of the Ruzsa triangle inequality. We follow with a proof due to Petridis [1].

Contents

Acknowledgements The authors were supported by the ERC Advanced Grant ALEXANDRIA (Project 742178) funded by the European Research Council.

1 The Plünnecke-Ruzsa Inequality

Authors: Angeliki Koutsoukou-Argyraki and Lawrence C. Paulson, University of Cambridge.

We formalise Plünnecke's inequality and the Plünnecke-Ruzsa inequality, following the notes by Timothy Gowers: "Introduction to Additive Combinatorics" (2022) for the University of Cambridge. To this end, we first introduce basic definitions and prove elementary facts on sumsets and difference sets. Then, we show (two versions of) the Ruzsa triangle inequality. We follow with a proof due to Petridis.

theory *Pluennecke-Ruzsa-Inequality*
  imports
    Jacobson-Basic-Algebra.Ring-Theory
    Complex-Main

begin

notation *plus* (**infixl** ‹+› *65*)
notation *minus* (**infixl** ‹−› *65*)
unbundle *uminus-syntax*

1.1 Key definitions (sumset, difference set) and basic lemmas

Working in an arbitrary Abelian group, with additive syntax

locale *additive-abelian-group* = *abelian-group G* (⊕) **0**
  for *G* **and** *addition* (**infixl** ‹⊕› *65*) **and** *zero* (‹**0**›)

begin

abbreviation *G-minus*:: $'a \Rightarrow 'a \Rightarrow 'a$ (**infixl** ‹⊖› *70*)
  where $x \ominus y \equiv x \oplus \mathit{inverse}\ y$

lemma *inverse-closed*: $x \in G \Longrightarrow \mathit{inverse}\ x \in G$
  by *blast*

1.1.1 Sumsets

inductive-set *sumset* :: $'a\ set \Rightarrow 'a\ set \Rightarrow 'a\ set$ **for** *A B*
  where
    sumsetI[*intro*]: $[\![a \in A;\ a \in G;\ b \in B;\ b \in G]\!] \Longrightarrow a \oplus b \in \mathit{sumset}\ A\ B$

lemma *sumset-eq*: $\mathit{sumset}\ A\ B = \{c.\ \exists\, a \in A \cap G.\ \exists\, b \in B \cap G.\ c = a \oplus b\}$
  by (*auto simp*: *sumset.simps elim*!: *sumset.cases*)

lemma *sumset*: $\mathit{sumset}\ A\ B = (\bigcup a \in A \cap G.\ \bigcup b \in B \cap G.\ \{a \oplus b\})$
  by (*auto simp*: *sumset-eq*)

lemma *sumset-subset-carrier*: *sumset* A $B \subseteq G$
  by (*auto simp*: *sumset-eq*)

lemma *sumset-Int-carrier* [*simp*]: *sumset* A $B \cap G$ = *sumset* A B
  by (*simp add*: *Int-absorb2 sumset-subset-carrier*)

lemma *sumset-mono*: $[\![A' \subseteq A; B' \subseteq B]\!] \Longrightarrow$ *sumset* A' $B' \subseteq$ *sumset* A B
  by (*auto simp*: *sumset-eq*)

lemma *sumset-insert1*: *NO-MATCH* $\{\}$ $A \Longrightarrow$ *sumset* (*insert* x A) B = *sumset* $\{x\}$ $B \cup$ *sumset* A B
  by (*auto simp*: *sumset-eq*)

lemma *sumset-insert2*: *NO-MATCH* $\{\}$ $B \Longrightarrow$ *sumset* A (*insert* x B) = *sumset* A $\{x\} \cup$ *sumset* A B
  by (*auto simp*: *sumset-eq*)

lemma *sumset-subset-Un1*: *sumset* $(A \cup A')$ B = *sumset* A $B \cup$ *sumset* A' B
  by (*auto simp*: *sumset-eq*)

lemma *sumset-subset-Un2*: *sumset* A $(B \cup B')$ = *sumset* A $B \cup$ *sumset* A B'
  by (*auto simp*: *sumset-eq*)

lemma *sumset-subset-insert*: *sumset* A $B \subseteq$ *sumset* A (*insert* x B) *sumset* A $B \subseteq$ *sumset* (*insert* x A) B
  by (*auto simp*: *sumset-eq*)

lemma *sumset-subset-Un*: *sumset* A $B \subseteq$ *sumset* A $(B \cup C)$ *sumset* A $B \subseteq$ *sumset* $(A \cup C)$ B
  by (*auto simp*: *sumset-eq*)

lemma *sumset-empty* [*simp*]: *sumset* A $\{\} = \{\}$ *sumset* $\{\}$ $A = \{\}$
  by (*auto simp*: *sumset-eq*)

lemma *sumset-empty'*:
  assumes $A \cap G = \{\}$
  shows *sumset* B $A = \{\}$ *sumset* A $B = \{\}$
  using *assms* **by** (*auto simp*: *sumset-eq*)

lemma *sumset-is-empty-iff* [*simp*]: *sumset* A $B = \{\} \longleftrightarrow A \cap G = \{\} \lor B \cap G = \{\}$
  by (*auto simp*: *sumset-eq*)

lemma *sumset-D* [*simp*]: *sumset* A $\{\mathbf{0}\} = A \cap G$ *sumset* $\{\mathbf{0}\}$ $A = A \cap G$
  by (*auto simp*: *sumset-eq*)

lemma *sumset-Int-carrier-eq* [*simp*]: *sumset* A $(B \cap G)$ = *sumset* A B *sumset* $(A \cap G)$ B = *sumset* A B

by (*auto simp*: *sumset-eq*)

lemma *sumset-assoc*:
shows *sumset* (*sumset A B*) *C* = *sumset A* (*sumset B C*)
by (*fastforce simp add*: *sumset-eq associative Bex-def*)

lemma *sumset-commute*:
shows *sumset A B* = *sumset B A*
by (*auto simp*: *sumset-eq*; *meson Int-iff commutative*)

lemma *finite-sumset*:
assumes *finite A finite B* **shows** *finite* (*sumset A B*)
using *assms* **by** (*auto simp*: *sumset-eq*)

lemma *finite-sumset'*:
assumes *finite* $(A \cap G)$ *finite* $(B \cap G)$
shows *finite* (*sumset A B*)
using *assms* **by** (*auto simp*: *sumset-eq*)

lemma *sumsetdiff-sing*: *sumset* $(A - B)$ $\{x\}$ = *sumset A* $\{x\}$ − *sumset B* $\{x\}$
by (*auto simp*: *sumset-eq*)

lemma *card-sumset-singleton-eq*:
assumes *finite A* **shows** *card* (*sumset A* $\{a\}$) = (*if* $a \in G$ *then card* $(A \cap G)$ *else 0*)
proof (*cases* $a \in G$)
case *True*
then have *sumset A* $\{a\} = (\lambda x.\ x \oplus a)$ ‘ $(A \cap G)$
by (*auto simp*: *sumset-eq*)
moreover have *inj-on* $(\lambda x.\ x \oplus a)$ $(A \cap G)$
by (*auto simp*: *inj-on-def True*)
ultimately show *?thesis*
by (*metis True card-image*)
qed (*auto simp*: *sumset-eq*)

lemma *card-sumset-le*:
assumes *finite A* **shows** *card* (*sumset A* $\{a\}$) $\le$ *card A*
by (*simp add*: *assms card-mono card-sumset-singleton-eq*)

lemma *infinite-sumset-aux*:
assumes *infinite* $(A \cap G)$
shows *infinite* (*sumset A B*) $\longleftrightarrow B \cap G \neq \{\}$
proof (*cases* $B \cap G = \{\}$)
case *False*
then obtain *b* **where** *b*: $b \in B$ $b \in G$ **by** *blast*
with *assms commutative* **have** $((\oplus)b)$ ‘ $(A \cap G) \subseteq$ *sumset A B*
by (*auto simp*: *sumset*)
moreover have *inj-on* $((\oplus)b)$ $(A \cap G)$
by (*meson IntD2 b inj-onI invertible invertible-left-cancel*)

ultimately show *?thesis*
by (*metis False assms inj-on-finite*)
qed (*auto simp*: *sumset-eq*)

lemma *infinite-sumset-iff*:
shows *infinite* (*sumset A B*) $\longleftrightarrow$ *infinite* $(A \cap G) \wedge B \cap G \neq \{\} \vee A \cap G \neq \{\} \wedge$ *infinite* $(B \cap G)$
by (*metis* (*no-types*, *lifting*) *finite-sumset′ infinite-sumset-aux sumset-commute*)

lemma *card-le-sumset*:
assumes *A*: *finite A* $a \in A$ $a \in G$
and *B*: *finite B* $B \subseteq G$
shows *card B* $\leq$ *card* (*sumset A B*)
proof −
have $B \subseteq (\oplus)$ (*inverse a*) ‘ *sumset A B*
using *A B*
apply (*clarsimp simp*: *sumset image-iff*)
by (*metis Int-absorb2 Int-iff invertible invertible-left-inverse2*)
with *A B* **show** *?thesis*
by (*meson finite-sumset surj-card-le*)
qed

lemma *card-sumset-0-iff′*: *card* (*sumset A B*) = *0* $\longleftrightarrow$ *card* $(A \cap G)$ = *0* $\vee$ *card* $(B \cap G)$ = *0*
proof (*cases infinite* $(A \cap G) \vee$ *infinite* $(B \cap G)$)
case *True*
then show *?thesis*
by (*metis card-eq-0-iff infinite-sumset-iff sumset-empty′*)
qed (*auto simp*: *sumset-eq*)

lemma *card-sumset-0-iff*:
assumes $A \subseteq G$ $B \subseteq G$
shows *card* (*sumset A B*) = *0* $\longleftrightarrow$ *card A* = *0* $\vee$ *card B* = *0*
by (*metis assms le-iff-inf card-sumset-0-iff′*)

lemma *card-sumset-leq*:
assumes $A \subseteq G$
shows *card*(*sumset A A*) $\leq$ *Suc*(*card A*) *choose 2*
using *assms*
proof (*induction card A arbitrary*: *A*)
case *0*
then show *?case*
by (*metis card-sumset-0-iff zero-le*)
next
case (*Suc n A*)
then obtain *a A′* **where** *a*: $a \in A$ $A' = A - \{a\}$ $a \in G$
by (*metis Zero-neq-Suc card-eq-0-iff subset-empty subset-eq*)
then have *n*: *card A′* = *n*
by (*metis Suc*(*2*) *card-Diff-singleton diff-Suc-Suc minus-nat.diff-0 One-nat-def*)

have *finite A*
by (*metis Suc*(*2*) *Zero-neq-Suc card.infinite*)
have *card* (*sumset A A*) ≤ *card* (*sumset A′ A′*) + *card A*
proof −
have *A*: *A* = *A′* ∪ {*a*}
using *a* **by** *auto*
then have *sumset A A* = (*sumset A′ A′*) ∪ (*sumset A* {*a*})
by (*auto simp*: *sumset-eq commutative*)
with *a* ‹*finite A*› *card-sumset-le* **show** *?thesis*
by (*simp add*: *order-trans*[*OF card-Un-le*])
qed
also have ... ≤ (*card A choose 2*) + *card A*
using *Suc a* **by** (*metis add-le-mono1 insert-Diff-single insert-absorb insert-subset n*)
also have ... ≤ *Suc* (*card A*) *choose 2*
by (*simp add*: *numeral-2-eq-2*)
finally show *?case* **.**
qed

1.1.2 Iterated sumsets

definition *sumset-iterated* :: ′*a set* ⇒ *nat* ⇒ ′*a set*
where *sumset-iterated A r* ≡ *Finite-Set.fold* (*sumset* ∘ (λ-. *A*)) {**0**} {..<*r*}

lemma *sumset-iterated-0* [*simp*]: *sumset-iterated A 0* = {**0**}
by (*simp add*: *sumset-iterated-def*)

lemma *sumset-iterated-Suc* [*simp*]: *sumset-iterated A* (*Suc k*) = *sumset A* (*sumset-iterated A k*)
(**is** *?lhs* = *?rhs*)
proof −
interpret *comp-fun-commute-on* {..*k*} *sumset* ∘ (λ-. *A*)
using *sumset-assoc sumset-commute* **by** (*auto simp*: *comp-fun-commute-on-def*)
have *?lhs* = (*sumset* ∘ (λ-. *A*)) *k* (*Finite-Set.fold* (*sumset* ∘ (λ-. *A*)) {**0**} {..<*k*})
unfolding *sumset-iterated-def lessThan-Suc*
by (*subst fold-insert*, *auto*)
also have ... = *?rhs*
by (*simp add*: *sumset-iterated-def*)
finally show *?thesis* **.**
qed

lemma *sumset-iterated-2*:
shows *sumset-iterated A 2* = *sumset A A*
by (*simp add*: *eval-nat-numeral*)

lemma *sumset-iterated-r*: *r* > *0* ⟹ *sumset-iterated A r* = *sumset A* (*sumset-iterated A* (*r*−*1*))
using *gr0-conv-Suc* **by** *force*

lemma *sumset-iterated-subset-carrier*: *sumset-iterated A k* ⊆ *G*
  by (*cases k*; *simp add*: *sumset-subset-carrier*)

lemma *finite-sumset-iterated*: *finite A* ⟹ *finite* (*sumset-iterated A r*)
  by(*induction r*) (*auto simp*: *finite-sumset*)

lemma *sumset-iterated-empty*: *r>0* ⟹ *sumset-iterated* {} *r* = {}
  by (*induction r*) *auto*

1.1.3 Difference sets

inductive-set *minusset* :: *'a set* ⇒ *'a set* **for** *A*
  where
    minussetI[*intro*]: ⟦*a* ∈ *A*; *a* ∈ *G*⟧ ⟹ *inverse a* ∈ *minusset A*

lemma *minusset-eq*: *minusset A* = *inverse* ‘ (*A* ∩ *G*)
  by (*auto simp*: *minusset.simps*)

abbreviation *differenceset A B* ≡ *sumset A* (*minusset B*)

lemma *minusset-is-empty-iff* [*simp*]: *minusset A* = {} ⟷ *A* ∩ *G* = {}
  by (*auto simp*: *minusset-eq*)

lemma *minusset-triv* [*simp*]: *minusset* {**0**} = {**0**}
  by (*auto simp*: *minusset-eq*)

lemma *minusset-subset-carrier*: *minusset A* ⊆ *G*
  by (*auto simp*: *minusset-eq*)

lemma *minus-minusset* [*simp*]: *minusset* (*minusset A*) = *A* ∩ *G*
  apply (*auto simp*: *minusset-eq*)
  by (*metis inverse-equality invertible invertibleE minusset.minussetI minusset-eq*)

lemma *card-minusset* [*simp*]: *card* (*minusset A*) = *card* (*A* ∩ *G*)
proof (*rule bij-betw-same-card*)
  show *bij-betw* (*inverse*) (*minusset A*) (*A* ∩ *G*)
    unfolding *minusset-eq* **by** (*force intro*: *bij-betwI*)
qed

lemma *card-minusset'*: *A* ⊆ *G* ⟹ *card* (*minusset A*) = *card A*
  by (*simp add*: *Int-absorb2*)

lemma *diff-minus-set*:
  differenceset (*minusset A*) *B* = *minusset* (*sumset A B*) (**is** *?lhs* = *?rhs*)
proof (*rule Set.set-eqI*)
  fix *u*
  have *u* ∈ *?lhs* ⟷
        (∃ *x* ∈ *A* ∩ *G*. ∃ *y* ∈ *B* ∩ *G*. *u* = *inverse x* ⊖ *y*)

by (*auto simp*: *sumset minusset-eq*)
also have ... $\longleftrightarrow$ ($\exists$ $x \in A \cap G$. $\exists$ $y \in B \cap G$. u = *inverse* ($y \oplus x$))
using *inverse-composition-commute* **by** *auto*
also have ... $\longleftrightarrow$ $u \in$ *?rhs*
by (*auto simp*: *sumset minusset-eq commutative*)
finally show $u \in$ *?lhs* $\longleftrightarrow$ $u \in$ *?rhs* **.**
qed

lemma *differenceset-commute* [*simp*]:
shows *minusset* (*differenceset B A*) = *differenceset A B*
by (*metis diff-minus-set minus-minusset sumset-Int-carrier-eq*(*1*) *sumset-commute*)

lemma *card-differenceset-commute*: *card* (*differenceset B A*) = *card* (*differenceset A B*)
by (*metis card-minusset′ differenceset-commute sumset-subset-carrier*)

lemma *minusset-distrib-sum*:
shows *minusset* (*sumset A B*) = *sumset* (*minusset A*) (*minusset B*)
by (*simp add*: *diff-minus-set*)

lemma *minusset-iterated-minusset*: *sumset-iterated* (*minusset A*) *k* = *minusset* (*sumset-iterated A k*)
by (*induction k*) (*auto simp*: *diff-minus-set*)

lemma *card-sumset-iterated-minusset*:
card (*sumset-iterated* (*minusset A*) *k*) = *card* (*sumset-iterated A k*)
by (*metis card-minusset′ minusset-iterated-minusset sumset-iterated-subset-carrier*)

lemma *finite-minusset*: *finite A* $\Longrightarrow$ *finite* (*minusset A*)
by (*simp add*: *minusset-eq*)

lemma *finite-differenceset*: *finite A* $\Longrightarrow$ *finite B* $\Longrightarrow$ *finite* (*differenceset A B*)
by (*simp add*: *finite-minusset finite-sumset*)

1.2 The Ruzsa triangle inequality

lemma *Ruzsa-triangle-ineq1*:
assumes *U*: *finite U* $U \subseteq G$
and *V*: *finite V* $V \subseteq G$
and *W*: *finite W* $W \subseteq G$
shows (*card U*) $*$ *card*(*differenceset V W*) $\le$ *card* (*differenceset U V*) $*$ *card* (*differenceset U W*)
proof −
have *fin*: *finite* (*differenceset U V*) *finite* (*differenceset U W*)
using *U V W finite-minusset finite-sumset* **by** *auto*
have $\exists$ *v w*. $v \in V \land w \in W \land x = v \ominus w$ **if** $x \in$ *differenceset V W* **for** *x*
using *that* **by** (*auto simp*: *sumset-eq minusset-eq*)
then obtain *v w* **where** *vinV*: $v\ x \in V$ **and** *winW*: $w\ x \in W$ **and** *vw-eq*: $v\ x \ominus (w\ x) = x$

```
    if x ∈ differenceset V W for x   by metis
  have vinG: v x ∈ G and winG: w x ∈ G if x ∈ differenceset V W for x
    using V W that vinV winW by auto
  define φ where φ ≡ λ(u,x). (u ⊖ (v x), u ⊖ (w x))
  have inj-on φ (U × differenceset V W)
  proof (clarsimp simp add: φ-def inj-on-def)
    fix u1 :: 'a and x1 :: 'a and u2 :: 'a and x2 :: 'a
    assume u1 ∈ U u2 ∈ U
      and x1: x1 ∈ differenceset V W
      and x2: x2 ∈ differenceset V W
      and v: u1 ⊖ v x1 = u2 ⊖ v x2
      and w: u1 ⊖ w x1 = u2 ⊖ w x2
    then obtain u1 ∈ G u2 ∈ G x1 ∈ G x2 ∈ G
      by (meson ‹U ⊆ G› subset-iff sumset-subset-carrier)
    show u1 = u2 ∧ x1 = x2
    proof
      have v x1 ⊖ w x1 = (u1 ⊖ w x1) ⊖ (u1 ⊖ v x1)
      by (smt (verit, del-insts) ‹u1 ∈ G› associative commutative composition-closed
inverse-closed
            invertible invertible-right-inverse2 vinG winG x1)
      also have ... = (u2 ⊖ w x2) ⊖ (u2 ⊖ v x2)
        using v w by presburger
      also have ... = v x2 ⊖ w x2
      by (smt (verit, del-insts) ‹u2 ∈ G› associative commutative composition-closed
inverse-equality
            invertible invertible-def invertible-right-inverse2 vinG winG x2)
      finally have v x1 ⊖ w x1 = v x2 ⊖ w x2 .
      then show x1=x2
        by (simp add: x1 x2 vw-eq)
      then show u1=u2
        using ‹u1 ∈ G› ‹u2 ∈ G› w winG x1 by force
    qed
  qed
 moreover have φ ∈ (U × differenceset V W) → (differenceset U V) × (differenceset
U W)
    using ‹U ⊆ G› ‹V ⊆ G› ‹W ⊆ G›
    by (fastforce simp: φ-def intro: vinV winW)
  ultimately have card (U × differenceset V W) ≤ card (differenceset U V ×
differenceset U W)
    using card-inj fin by blast
  then show ?thesis
    by (simp flip: card-cartesian-product)
qed


definition Ruzsa-distance:: 'a set ⇒ 'a set ⇒ real
   where Ruzsa-distance A B ≡ card(differenceset A B)/(sqrt(card A) ∗ sqrt(card
B))
```

lemma *Ruzsa-triangle-ineq2*:
assumes *U*: *finite U U* $\subseteq$ *G U* $\neq$ {}
and *V*: *finite V V* $\subseteq$ *G*
and *W*: *finite W W* $\subseteq$ *G*
shows *Ruzsa-distance V W* $\leq$ (*Ruzsa-distance V U*) $*$ (*Ruzsa-distance U W*)
proof −
have *card U* $*$ *card* (*differenceset V W*) $\leq$ *card* (*differenceset U V*) $*$ *card* (*differenceset U W*)
using *assms Ruzsa-triangle-ineq1* **by** *metis*
— now divide both sides with the same quantity
then have *card U* $*$ *card* (*differenceset V W*) / (*card U* $*$ *sqrt* (*card V*) $*$ *sqrt* (*card W*))
$\leq$ *card* (*differenceset U V*) $*$ *card* (*differenceset U W*) / (*card U* $*$ *sqrt* (*card V*) $*$ *sqrt* (*card W*))
using *assms*
by (*metis divide-right-mono mult-eq-0-iff mult-left-mono of-nat-0-le-iff of-nat-mono real-sqrt-ge-0-iff*)
then have $*$: *card*(*differenceset V W*) / (*sqrt*(*card V*) $*$ *sqrt*(*card W*)) $\leq$
card (*differenceset U V*) $*$ *card* (*differenceset U W*)
/ (*card U* $*$ *sqrt*(*card V*) $*$ *sqrt*(*card W*))
using *assms* **by** *simp*
have *card* (*differenceset U V*) $*$ *card* (*differenceset U W*)/(*card U* $*$ *sqrt*(*card V*) $*$ *sqrt*(*card W*))
= *card*(*differenceset V U*) / (*sqrt*(*card U*) $*$ *sqrt*(*card V*))$*$
card(*differenceset U W*) / (*sqrt*(*card U*) $*$ *sqrt*(*card W*))
using *assms*
by (*simp add*: *divide-simps*) (*metis card-minusset differenceset-commute minus-minusset*)
then have
card(*differenceset V W*) / (*sqrt*(*card V*) $*$ *sqrt*(*card W*)) $\leq$
card(*differenceset V U*) / (*sqrt*(*card U*) $*$ *sqrt*(*card V*)) $*$
card(*differenceset U W*) / (*sqrt*(*card U*) $*$ *sqrt*(*card W*))
using $*$ *assms* **by** *auto*
then show *?thesis* **unfolding** *Ruzsa-distance-def*
by (*metis divide-divide-eq-left divide-divide-eq-left′ times-divide-eq-right*)
qed

1.3 Petridis's proof of the Plünnecke-Ruzsa inequality

lemma *Plu-2-2*:
assumes *K0*: *card* (*sumset A0 B*) $\leq$ *K0* $*$ *real* (*card A0*)
and *A0*: *finite A0 A0* $\subseteq$ *G A0* $\neq$ {}
and *B*: *finite B B* $\subseteq$ *G B* $\neq$ {}
obtains *A K*
where *A* $\subseteq$ *A0 A* $\neq$ {} *0* $<$ *K K* $\leq$ *K0*
and $\bigwedge$*C*. *C* $\subseteq$ *G* $\Longrightarrow$ *finite C* $\Longrightarrow$ *card* (*sumset A* (*sumset B C*)) $\leq$ *K* $*$ *real* (*card*(*sumset A C*))
proof

define *KS* **where** $KS \equiv (\lambda A.\ card\ (sumset\ A\ B)\ /\ real\ (card\ A))$ ‘ $(Pow\ A0 - \{\{\}\})$
define *K* **where** $K \equiv Min\ KS$
define *A* **where** $A \equiv @A.\ A \in Pow\ A0 - \{\{\}\} \wedge K = card\ (sumset\ A\ B)\ /\ real\ (card\ A)$
obtain *KS*: *finite KS KS* $\neq \{\}$
  using *KS-def A0* **by** *blast*
then have $K \in KS$
  using *K-def Min-in* **by** *blast*
then have $\exists A.\ A \in Pow\ A0 - \{\{\}\} \wedge K = card\ (sumset\ A\ B)\ /\ real\ (card\ A)$
  using *KS-def* **by** *blast*
then obtain $A \in Pow\ A0 - \{\{\}\}$ **and** *Keq*: $K = card\ (sumset\ A\ B)\ /\ real\ (card\ A)$
  by (*metis* (*mono-tags*, *lifting*) *A-def someI-ex*)
then show *A*: $A \subseteq A0\ A \neq \{\}$
  by *auto*
with *A0 finite-subset* **have** $A \subseteq G$ *finite A*
  by *blast+*
have *gt0*: $0 < real\ (card\ (sumset\ A\ B))\ /\ real\ (card\ A)$ **if** $A \neq \{\}$ **and** $A \subseteq A0$ **for** *A*
  using *that assms*
    by (*smt* (*verit*, *best*) *order-trans card-0-eq card-sumset-0-iff divide-pos-pos of-nat-le-0-iff finite-subset*)
then show $K > 0$
  using *A Keq* **by** *presburger*
have *K-cardA*: $K * (card\ A) = card\ (sumset\ A\ B)$
  unfolding *Keq* **using** *Keq* ‹$0 < K$› **by** *force*
have *K-le*: $real\ (card\ (sumset\ A'\ B))\ /\ card\ A' \geq K$ **if** $A' \subseteq A\ A' \neq \{\}$ **for** A'
  using *KS K-def KS-def* ‹$A \subseteq A0$› *that* **by** *force*
with *A0* **have** $card\ (sumset\ A0\ B)\ /\ real\ (card\ A0) \in KS$
  by (*auto simp*: *KS-def*)
with *A0* **show** $K \leq K0$
  by (*metis KS K-def Min-le-iff card-gt-0-iff mult-imp-div-pos-le of-nat-0-less-iff K0*)
show $card\ (sumset\ A\ (sumset\ B\ C)) \leq K * real\ (card(sumset\ A\ C))$
  if *finite C C* $\subseteq G$ **for** *C*
  using *that*
proof (*induction C*)
  case *empty*
  then show *?case* **by** *simp*
    — This is actually trivial: it does not follow from $real\ (card\ (sumset\ A\ B)) = K * real\ (card\ A)$ as claimed in the notes.
next
  case (*insert x C*)
  then have $x \in G\ C \subseteq G$ *finite C*
    by *auto*
  define A' **where** $A' \equiv A \cap \{a.\ (a \oplus x) \in sumset\ A\ C\}$
  with ‹*finite A*› **have** *finite* $A'\ A' \subseteq A$ **by** *auto*
  then have [*simp*]: $real\ (card\ A - card\ A') = real\ (card\ A) - real\ (card\ A')$

by (*meson* ‹*finite A*› *card-mono of-nat-diff*)
have *0*: *sumset A C* ∩ *sumset* (*A* − *A'*) {*x*} = {}
by (*clarsimp simp add*: *A'-def sumset-eq disjoint-iff*) (*metis IntI*)
have *1*: *sumset A* (*insert x C*) = *sumset A C* ∪ *sumset* (*A* − *A'*) {*x*}
by (*auto simp*: *A'-def sumset-eq*)
have *card* (*sumset A* (*insert x C*)) = *card* (*sumset A C*) + *card* (*sumset* (*A* − *A'*) {*x*})
by (*simp add*: *0 1* ‹*finite A*› *card-Un-disjoint finite-sumset local.insert*)
also have ... = *card* (*sumset A C*) + *card* ((*A* − *A'*) ∩ *G*)
using ‹*finite A*› ‹*x* ∈ *G*› **by** (*simp add*: *card-sumset-singleton-eq*)
also have ... = *card* (*sumset A C*) + *card* (*A* − *A'*)
by (*metis* ‹*A* ⊆ *G*› *Int-absorb2 Int-Diff Int-commute*)
also have ... = *card* (*sumset A C*) + (*card A* − *card A'*)
by (*simp add*: *A'-def* ‹*finite A*› *card-Diff-subset*)
finally have ∗: *card* (*sumset A* (*insert x C*)) = *card* (*sumset A C*) + (*card A* − *card A'*) **.**
have *sumset A'* (*sumset B* {*x*}) ⊆ *sumset A* (*sumset B C*)
by (*clarsimp simp add*: *A'-def sumset-eq Bex-def*) (*metis associative commutative composition-closed*)
then have *sumset A* (*sumset B* (*insert x C*))
⊆ *sumset A* (*sumset B C*) ∪ (*sumset A* (*sumset B* {*x*}) − *sumset A'* (*sumset B* {*x*}))
by (*auto simp*: *sumset-insert2 sumset-subset-Un2*)
then have *card* (*sumset A* (*sumset B* (*insert x C*))) ≤ *card* (*sumset A* (*sumset B C*))
+ *card* ((*sumset A* (*sumset B* {*x*}) − *sumset A'* (*sumset B* {*x*})))
by (*smt* (*verit*, *best*) *B*(*1*) ‹*finite A*› ‹*finite C*› *order-trans card-Un-le card-mono finite.emptyI*
finite.insertI finite-Diff finite-Un finite-sumset)
also have ... = *card* (*sumset A* (*sumset B C*)) + (*card* (*sumset A* (*sumset B* {*x*})) − *card* (*sumset A'* (*sumset B* {*x*})))
by (*simp add*: ‹*A'* ⊆ *A*› ‹*finite A'*› ‹*finite B*› *card-Diff-subset finite-sumset sumset-mono*)
also have ... ≤ *card* (*sumset A* (*sumset B C*)) + (*card* (*sumset A B*) − *card* (*sumset A' B*))
using ‹*finite A*› ‹*finite A'*› ‹*finite B*› **by** (*simp add*: *card-sumset-singleton-eq finite-sumset flip*: *sumset-assoc*)
also have ... ≤ *K* ∗ *card* (*sumset A C*) + (*K* ∗ *card A* − *K* ∗ *card A'*)
proof (*cases A'* = {})
case *True*
with *local.insert* ‹*C* ⊆ *G*› *K-cardA* **show** *?thesis* **by** *auto*
next
case *False*
then have *K* ∗ *card A'* ≤ *real* (*card* (*sumset A' B*))
using *K-le*[*OF* ‹*A'* ⊆ *A*›] **by** (*simp add*: *divide-simps split*: *if-split-asm*)
then have *real* (*card* (*sumset A B*) − *card* (*sumset A' B*)) ≤ *K* ∗ *real* (*card A*) − *K* ∗ *real* (*card A'*)
by (*simp add*: *B*(*1*) *K-cardA* ‹*A'* ⊆ *A*› ‹*finite A*› *card-mono finite-sumset*

of-nat-diff sumset-mono)
      with *local.insert* **show** *?thesis* **by** *simp*
    qed
    also have ... $\leq K * real$ (*card* (*sumset A* (*insert x C*)))
      using $*$ ‹$A' \subseteq A$› **by** (*simp add*: *algebra-simps*)
    finally show *?case*
      using *of-nat-mono* **by** *blast*
  qed
qed

lemma *Cor-Plu-2-3*:
  assumes *K*: *card* (*sumset A B*) $\leq K * real$ (*card A*)
    and *A*: *finite A* $A \subseteq G$ $A \neq \{\}$
    and *B*: *finite B* $B \subseteq G$
  obtains A' **where** $A' \subseteq A$ $A' \neq \{\}$
    $\bigwedge r$. *card* (*sumset* A' (*sumset-iterated B r*)) $\leq K\hat{}r * real$ (*card* A')
proof (*cases* $B = \{\}$)
  case *True*
  have $K \geq 0$
    using *assms* **by** (*simp add*: *True zero-le-mult-iff*)
  moreover have $*$: *sumset-iterated B r* = (*if r=0 then* {**0**} *else* {}) **for** *r*
    by (*metis True sumset-iterated-0 sumset-iterated-empty zero-less-iff-neq-zero*)
  ultimately have *real* (*card* (*sumset A* (*sumset-iterated B r*)))
        $\leq K \hat{}\ r * real$ (*card A*) **for** *r*
    by (*simp add*: $*$ *Int-commute Int-absorb2* ‹$A \subseteq G$›)
  with ‹$A \neq \{\}$› *that* **show** *?thesis* **by** *blast*
next
  case *False*
  obtain A' K'
    where A': $A' \subseteq A$ $A' \neq \{\}$ $0 < K'$ $K' \leq K$
      and A'-*card*: $\bigwedge C$. $C \subseteq G \Longrightarrow$ *finite C* $\Longrightarrow$ *card* (*sumset* A' (*sumset B C*))
$\leq K' * real$ (*card*(*sumset* A' *C*))
    by (*metis A B Plu-2-2 K False*)
  with *A* **have** $A' \subseteq G$ **by** *blast*
  have $*$: *card* (*sumset* A' (*sumset-iterated B* (*Suc r*))) $\leq K' *$ *card* (*sumset* A'
(*sumset-iterated B r*))
        (**is** *?lhs* $\leq$ *?rhs*)
    for *r*
  proof −
    have *?lhs* = *card* (*sumset* A' (*sumset B* (*sumset-iterated B r*)))
      using *that* **by** (*simp add*: *sumset-iterated-r*)
    also have ... $\leq$ *?rhs*
      using A'-*card B finite-sumset-iterated sumset-iterated-subset-carrier* **by** *meson*
    finally show *?thesis* **.**
  qed
  have $**$: *card* (*sumset* A' (*sumset-iterated B r*)) $\leq K'\hat{}r * real$ (*card* A') **for** *r*
  proof (*induction r*)
    case *0*
    with ‹$A' \subseteq G$› **show** *?case*

by (*simp add*: *Int-absorb2*)
next
case (*Suc r*)
then show *?case*
by (*smt* (*verit*) ∗ ‹*0* < *K'*› *mult.commute mult.left-commute mult-le-cancel-left-pos power-Suc*)
qed
show *thesis*
proof
show *real* (*card* (*sumset A'* (*sumset-iterated B r*))) ≤ *K* ^ *r* ∗ *real* (*card A'*) for *r*
by (*meson* ∗∗ *A' order-trans less-eq-real-def mult-right-mono of-nat-0-le-iff power-mono*)
qed (*use A'* in *auto*)
qed

The following Corollary of the above is an important special case, also referred to as the original version of Plünnecke's inequality first shown by Plünnecke.

lemma *Cor-Plu-2-3-Pluennecke-ineq*:
assumes *K*: *card* (*sumset A B*) ≤ *K* ∗ *real* (*card A*)
and *A*: *finite A A* ⊆ *G A* ≠ {}
and *B*: *finite B B* ⊆ *G*
shows *real* (*card* (*sumset-iterated B r*)) ≤ *K* ^ *r* ∗ *real* (*card A*)
proof−
obtain *A'* **where** ∗:*A'* ⊆ *A A'* ≠ {}
card (*sumset A'* (*sumset-iterated B r*)) ≤ *K*^*r* ∗ *real* (*card A'*)
using *assms Cor-Plu-2-3* **by** *metis*
with *assms* **have** ∗∗: *card* (*sumset-iterated B r*) ≤ *card* (*sumset A'* (*sumset-iterated B r*))
by (*meson card-le-sumset finite-subset finite-sumset-iterated subset-empty subset-iff sumset-iterated-subset-carrier*)
with ∗ **show** *?thesis*
by (*smt* (*verit*, *best*) *A*(*1*) *K card-mono mult-left-mono of-nat-0-le-iff of-nat-le-iff zero-le-mult-iff zero-le-power*)
qed

Special case where $B = A$

lemma *Cor-Plu-2-3-1*:
assumes *K*: *card* (*sumset A A*) ≤ *K* ∗ *real* (*card A*)
and *A*: *finite A A* ⊆ *G A* ≠ {}
shows *card* (*sumset-iterated A r*) ≤ *K*^*r* ∗ *real* (*card A*)
proof −
have *K* > *0*
by (*meson A K Plu-2-2 less-le-trans*)
obtain *A'* **where** *A'*: *A'* ⊆ *A A'* ≠ {}
and *A'-card*: ⋀*r*. *card* (*sumset A'* (*sumset-iterated A r*)) ≤ *K*^*r* ∗ *real* (*card A'*)
by (*meson A Cor-Plu-2-3 K*)

with *A* **obtain** *a* **where** $a \in A'$ $a \in G$ *finite* A'
by (*metis ex-in-conv finite-subset subset-iff*)
then have *card* (*sumset-iterated A r*) $\leq$ *card* (*sumset* A' (*sumset-iterated A r*))
using *A card-le-sumset finite-sumset-iterated sumset-iterated-subset-carrier* **by** *meson*
also have ... $\leq K\hat{}r *$ *real* (*card* A')
using A'-*card* **by** *meson*
also have ... $\leq K\hat{}r *$ *real* (*card A*)
by (*simp add*: ‹$A' \subseteq A$› ‹*finite A*› ‹$0 < K$› *card-mono*)
finally show *?thesis*
by *linarith*
qed

Special case where $B = - A$

lemma *Cor-Plu-2-3-2*:
assumes *K*: *card* (*differenceset A A*) $\leq K *$ *real* (*card A*)
and *A*: *finite A* $A \subseteq G$ $A \neq \{\}$
shows *card* (*sumset-iterated A r*) $\leq K\hat{}r *$ *real* (*card A*)
proof −
have *card* $A > 0$
by (*simp add*: *A card-gt-0-iff*)
with *K* **have** $K \geq 0$
by (*smt* (*verit*, *del-insts*) *of-nat-0-less-iff of-nat-less-0-iff zero-le-mult-iff*)
obtain A' **where** A': $A' \subseteq A$ $A' \neq \{\}$
and A'-*card*: $\bigwedge r$. *card* (*sumset* A' (*sumset-iterated* (*minusset A*) *r*)) $\leq K\hat{}r *$ *real* (*card* A')
by (*metis A Cor-Plu-2-3 assms*(*1*) *card-eq-0-iff card-minusset*$'$ *minusset-subset-carrier*)
with *A* **obtain** *a* **where** $a \in A'$ $a \in G$ *finite* A'
by (*metis ex-in-conv finite-subset subset-iff*)
then have *card* (*sumset-iterated A r*) $\leq$ *card* (*sumset* A' (*sumset-iterated* (*minusset A*) *r*))
by (*metis A*(*1*) *card-le-sumset card-sumset-iterated-minusset finite-minusset finite-sumset-iterated sumset-iterated-subset-carrier*)
also have ... $\leq K\hat{}r *$ *real* (*card* A')
using A'-*card* **by** *meson*
also have ... $\leq K\hat{}r *$ *real* (*card A*)
by (*simp add*: ‹$A' \subseteq A$› ‹*finite A*› ‹$0 \leq K$› *card-mono mult-left-mono*)
finally show *?thesis*
by *linarith*
qed

The following result is known as the Plünnecke-Ruzsa inequality (Theorem 2.5 in Gowers's notes). The proof will make use of the Ruzsa triangle inequality.

theorem *Pluennecke-Ruzsa-ineq*:
assumes *K*: *card* (*sumset A B*) $\leq K *$ *real* (*card A*)
and *A*: *finite A* $A \subseteq G$ $A \neq \{\}$
and *B*: *finite B* $B \subseteq G$
and $0 < r$ $0 < s$

shows *card* (*differenceset* (*sumset-iterated* B r) (*sumset-iterated* B s)) $\leq K^\frown(r+s)$ $*$ *real*(*card* A)
proof $-$
have *card* $A > 0$
by (*simp add*: A *card-gt-0-iff*)
with K **have** $K \geq 0$
by (*smt* (*verit*, *del-insts*) *of-nat-0-less-iff of-nat-less-0-iff zero-le-mult-iff*)
obtain A' **where** A': $A' \subseteq A$ $A' \neq \{\}$
and A'*-le*: $\bigwedge r$. *card* (*sumset* A' (*sumset-iterated* B r)) $\leq K^\frown r * $ *real* (*card* A')
using *Cor-Plu-2-3 assms* **by** *metis*
define C **where** $C \equiv$ *minusset* A'
have *minusset* $C = A'$ **and** $C \neq \{\}$ **and** *cardA*: *card* $A' \leq$ *card* A **and** *cardC*: *card* C = *card* A'
using A' A *card-mono* **by** (*auto simp*: *C-def card-minusset' Int-absorb2*)
then have *cardCA*: *card* $C \leq$ *card* A **by** *linarith*
have $\bigwedge r$. *card* (*differenceset* C (*sumset-iterated* B r)) $\leq K^\frown r *$ *real* (*card* A')
using A'*-le C-def card-minusset' diff-minus-set sumset-subset-carrier* **by** *presburger*
then have r: *card* (*differenceset* C (*sumset-iterated* B r)) $\leq K^\frown r *$ *real* (*card* C)
and s: *card* (*differenceset* C (*sumset-iterated* B s)) $\leq K^\frown s *$ *real* (*card* C)
using *cardC* **by** *presburger*+
have *card* $C > 0$
by (*metis* A' ‹*finite* A› *cardC card-gt-0-iff finite-subset*)
moreover have $C \subseteq G$
by (*simp add*: *C-def minusset-subset-carrier*)
ultimately have *card* $C *$ *card* (*differenceset* (*sumset-iterated* B r) (*sumset-iterated* B s))
$\leq$ *card* (*differenceset* C (*sumset-iterated* B r)) $*$
card (*differenceset* C (*sumset-iterated* B s))
by (*meson Ruzsa-triangle-ineq1 B card-gt-0-iff finite-sumset-iterated sumset-iterated-subset-carrier*)
also have $\ldots \leq K^\frown(r+s) *$ *card* $C *$ *card* C
using *mult-mono* [*OF* r s] ‹$0 \leq K$› **by** (*simp add*: *power-add field-simps*)
finally have *card* (*differenceset* (*sumset-iterated* B r) (*sumset-iterated* B s)) $\leq$ $K^\frown(r+s) *$ *card* C
using ‹*card* $C > 0$› **by** (*simp add*: *field-simps*)
then show *?thesis*
by (*smt* (*verit*, *ccfv-SIG*) ‹$0 \leq K$› *cardA cardC mult-left-mono of-nat-mono zero-le-power*)
qed

The following is an alternative version of the Plünnecke-Ruzsa inequality (Theorem 2.1 in Gowers's notes).

theorem *Pluennecke-Ruzsa-ineq-alt*:
assumes *finite* A $A \subseteq G$
and *card* (*sumset* A A) $\leq K *$ *real* (*card* A) $r > 0$ $s > 0$
shows *card* (*differenceset* (*sumset-iterated* A r) (*sumset-iterated* A s)) $\leq K^\frown(r+s)$ $*$ *real*(*card* A)
proof (*cases* $A = \{\}$)
case *True*

then have *sumset-iterated A r* = {} **if** *r>0* **for** *r*
using *sumset-iterated-empty that* **by** *force*
with *assms* **show** *?thesis*
by (*auto simp*: *True*)
next
case *False*
with *assms Pluennecke-Ruzsa-ineq* **show** *?thesis* **by** *presburger*
qed

theorem *Pluennecke-Ruzsa-ineq-alt-2*:
assumes *finite A A* ⊆ *G*
and *card* (*differenceset A A*) ≤ *K* ∗ *real* (*card A*) *r* > *0 s* > *0*
shows *card* (*differenceset* (*sumset-iterated A r*) (*sumset-iterated A s*)) ≤ *K*^(*r*+*s*) ∗ *real*(*card A*)
proof (*cases A* = {})
case *True*
then have *sumset-iterated A r* = {} **if** *r>0* **for** *r*
using *sumset-iterated-empty that* **by** *force*
with *assms* **show** *?thesis*
by (*auto simp*: *True*)
next
case *False*
with *assms Pluennecke-Ruzsa-ineq* **show** *?thesis*
by (*smt* (*verit*, *ccfv-threshold*) *card-minusset′ differenceset-commute finite-minusset*

minusset-distrib-sum minusset-iterated-minusset minusset-subset-carrier)
qed

end

1.4 Supplementary material on sumsets for sets of integers: basic inequalities

lemma *moninv-int*: *monoid.invertible UNIV* (+) *0 u* **for** *u*::*int*
using *monoid.invertibleI* [**where** *v* = −*u*] **by** (*simp add*: *Group-Theory.monoid-def*)

interpretation *int*: *additive-abelian-group UNIV* (+) *0*::*int*
by *unfold-locales* (*use moninv-int* **in** *auto*)

lemma *card-sumset-geq1*:
assumes *A*: *A* ≠ {} *finite A* **and** *B*: *B* ≠ {} *finite B*
shows *card*(*int.sumset A B*) ≥ (*card A*) + (*card B*) − *1*
using *A*
proof (*induction card A arbitrary*: *A*)
case (*Suc n*)
define *a* **where** *a* = *Max A*
define *A′* **where** *A′* ≡ *A* − {*a*}
then obtain *a*: *a* ∈ *A A′* = *A* − {*a*} *finite A′ a* ∉ *A′* **and** *A*: *A* = *insert a A′*
using *Max-in Suc a-def* **by** *blast*

with *Suc* **have** *n*: *card* $A' = n$
by (*metis card-Diff-singleton diff-Suc-Suc minus-nat.diff-0 One-nat-def*)
show *?case*
proof (*cases* $A' = \{\}$)
case *True*
then show *?thesis*
by (*simp add*: *A B*(*2*) *int.card-sumset-singleton-eq int.sumset-commute*)
next
case *False*
have $a + Max\ B \notin int.sumset\ A'\ B$
using ‹*finite A*› ‹*finite B*›
by (*smt* (*verit*, *best*) *DiffE Max-ge a a-def int.sumset.cases singleton-iff*)
then have ∗: $\neg\ int.sumset\ A'\ B \cup (+)\ a$ ‘ $B \subseteq int.sumset\ A'\ B$
using *B Max-in* **by** *blast*
have $card\ A + card\ B - 1 \leq Suc\ (card\ (int.sumset\ A'\ B))$
using *Suc False A a* **using** *le-diff-conv* **by** *force*
also have $\ldots \leq card\ (int.sumset\ A'\ B \cup (+)\ a$ ‘ $B)$
using *a B*
by (*metis* ∗ *card-seteq finite-Un finite-imageI int.finite-sumset not-less-eq-eq sup-ge1*)
also have $\ldots \leq card\ (int.sumset\ A\ B)$
proof (*rule card-mono*)
show *finite* (*int.sumset A B*)
using *B Suc.prems int.finite-sumset* **by** *blast*
show $int.sumset\ A'\ B \cup (+)\ a$ ‘ $B \subseteq int.sumset\ A\ B$
using *A* **by** (*force simp*: *int.sumset*)
qed
finally show *?thesis* **.**
qed
qed *auto*

lemma *card-sumset-geq2*:
shows $card(int.sumset\ A\ A) \geq 2 * (card\ A) - 1$
using *card-sumset-geq1* [*of A*]
by (*metis mult.commute Nat.add-0-right card-eq-0-iff diff-0-eq-0 le0 mult-2-right*)

end

References

[1] G. Petridis. The Plünnecke–Ruzsa inequality: An overview. In M. B. Nathanson, editor, *Combinatorial and Additive Number Theory*, pages 229–241. Springer, 2014.