

# The Plünnecke-Ruzsa Inequality

Angeliki Koutsoukou-Argyraki and Lawrence C. Paulson

February 6, 2026

## **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

<b>1</b>	<b>The Plünnecke-Ruzsa Inequality</b>	<b>3</b>
1.1	Key definitions (sumset, difference set) and basic lemmas . . .	3
1.1.1	Sumsets . . . . .	3
1.1.2	Iterated sumsets . . . . .	7
1.1.3	Difference sets . . . . .	8
1.2	The Ruzsa triangle inequality . . . . .	9
1.3	Petridis's proof of the Plünnecke-Ruzsa inequality . . . . .	11
1.4	Supplementary material on sumsets for sets of integers: basic inequalities . . . . .	18

**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-Argraki 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 Pluenecke-Ruzsa-Inequality
  imports
    Jacobson-Basic-Algebra.Ring-Theory
    Complex-Main
```

```
begin
```

```
notation plus (infixl  $\langle + \rangle$  65)
notation minus (infixl  $\langle - \rangle$  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  $G$  ( $\oplus$ )  $\mathbf{0}$ 
  for  $G$  and addition (infixl  $\langle \oplus \rangle$  65) and zero ( $\langle \mathbf{0} \rangle$ )
```

```
begin
```

```
abbreviation G-minus:: 'a  $\Rightarrow$  'a  $\Rightarrow$  'a (infixl  $\langle \ominus \rangle$  70)
  where  $x \ominus y \equiv x \oplus \text{inverse } y$ 
```

```
lemma inverse-closed:  $x \in G \implies \text{inverse } x \in G$ 
  by blast
```

### 1.1.1 Sumsets

```
inductive-set sumset :: 'a set  $\Rightarrow$  'a set  $\Rightarrow$  'a set for  $A B$ 
  where
    sumsetI[intro]:  $\llbracket a \in A; a \in G; b \in B; b \in G \rrbracket \implies a \oplus b \in \text{sumset } A B$ 
```

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

```
lemma sumset:  $\text{sumset } A B = (\bigcup a \in A \cap G. \bigcup b \in B \cap G. \{a \oplus b\})$ 
  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*:  $[[A' \subseteq A; B' \subseteq B]] \implies sumset A' B' \subseteq sumset A B$   
**by** (*auto simp: sumset-eq*)

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

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

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

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

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

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

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

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

by (auto simp: sumset-eq)

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

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

**lemma** *finite-sumset*:  
**assumes** *finite*  $A$  *finite*  $B$  **shows** *finite*  $(\text{sumset } A B)$   
**using** *assms* **by** (auto simp: sumset-eq)

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

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

**lemma** *card-sumset-singleton-eq*:  
**assumes** *finite*  $A$  **shows**  $\text{card } (\text{sumset } A \{a\}) = (\text{if } a \in G \text{ then } \text{card } (A \cap G) \text{ else } 0)$   
**proof** (cases  $a \in G$ )  
  **case** *True*  
  **then have**  $\text{sumset } A \{a\} = (\lambda x. x \oplus a) ` (A \cap G)$   
  **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**  $\text{card } (\text{sumset } A \{a\}) \leq \text{card } A$   
**by** (simp add: *assms* card-mono card-sumset-singleton-eq)

**lemma** *infinite-sumset-aux*:  
**assumes** *infinite*  $(A \cap G)$   
**shows** *infinite*  $(\text{sumset } A B) \longleftrightarrow B \cap G \neq \{\}$   
**proof** (cases  $B \cap G = \{\}$ )  
  **case** *False*  
  **then obtain**  $b$  **where**  $b \in B$   $b \in G$  **by** blast  
  **with** *assms* *commutative* **have**  $((\oplus)b) ` (A \cap G) \subseteq \text{sumset } A B$   
  **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**  $\text{infinite } (\text{sumset } A \ B) \longleftrightarrow \text{infinite } (A \cap G) \wedge B \cap G \neq \{\} \vee A \cap G \neq \{\} \wedge \text{infinite } (B \cap G)$   
**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**  $\text{card } B \leq \text{card } (\text{sumset } A \ B)$   
**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'*:  $\text{card } (\text{sumset } A \ B) = 0 \longleftrightarrow \text{card } (A \cap G) = 0 \vee \text{card } (B \cap G) = 0$   
**proof** (*cases infinite (A ∩ G) ∨ infinite (B ∩ 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**  $\text{card } (\text{sumset } A \ B) = 0 \longleftrightarrow \text{card } A = 0 \vee \text{card } B = 0$   
**by** (*metis assms le-iff-inf card-sumset-0-iff'*)

**lemma** *card-sumset-leq*:  
**assumes**  $A \subseteq G$   
**shows**  $\text{card } (\text{sumset } A \ A) \leq \text{Suc}(\text{card } A)$  *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**  $a \in A$   $A' = A - \{a\}$   $a \in G$   
**by** (*metis Zero-neq-Suc card-eq-0-iff subset-empty subset-eq*)  
**then have**  $n$ :  $\text{card } A' = n$   
**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*:  $\text{sumset-iterated } A \ k \subseteq G$   
**by** (*cases k*; *simp add: sumset-subset-carrier*)

**lemma** *finite-sumset-iterated*:  $\text{finite } A \implies \text{finite } (\text{sumset-iterated } A \ r)$   
**by** (*induction r*) (*auto simp: finite-sumset*)

**lemma** *sumset-iterated-empty*:  $r > 0 \implies \text{sumset-iterated } \{\} \ r = \{\}$   
**by** (*induction r*) *auto*

### 1.1.3 Difference sets

**inductive-set** *minusset* :: 'a set  $\Rightarrow$  'a set **for**  $A$   
**where**

*minussetI*[*intro*]:  $\llbracket a \in A; a \in G \rrbracket \implies \text{inverse } a \in \text{minusset } A$

**lemma** *minusset-eq*:  $\text{minusset } A = \text{inverse } (A \cap G)$   
**by** (*auto simp: minusset.simps*)

**abbreviation** *differenceset*  $A \ B \equiv \text{sumset } A \ (\text{minusset } B)$

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

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

**lemma** *minusset-subset-carrier*:  $\text{minusset } A \subseteq G$   
**by** (*auto simp: minusset-eq*)

**lemma** *minus-minusset* [*simp*]:  $\text{minusset } (\text{minusset } A) = A \cap G$   
**apply** (*auto simp: minusset-eq*)  
**by** (*metis inverse-equality invertible invertibleE minusset.minussetI minusset-eq*)

**lemma** *card-minusset* [*simp*]:  $\text{card } (\text{minusset } A) = \text{card } (A \cap G)$

**proof** (*rule bij-betw-same-card*)

**show** *bij-betw* (*inverse*) (*minusset A*) (*A  $\cap$  G*)

**unfolding** *minusset-eq* **by** (*force intro: bij-betwI*)

**qed**

**lemma** *card-minusset'*:  $A \subseteq G \implies \text{card } (\text{minusset } A) = \text{card } A$   
**by** (*simp add: Int-absorb2*)

**lemma** *diff-minus-set*:

*differenceset* (*minusset A*)  $B = \text{minusset } (\text{sumset } A \ B)$  (**is** *?lhs = ?rhs*)

**proof** (*rule Set.set-eqI*)

**fix**  $u$

**have**  $u \in ?lhs \longleftrightarrow$

$(\exists x \in A \cap G. \exists y \in B \cap G. u = \text{inverse } x \ominus y)$

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

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

**lemma** *card-differenceset-commute*:  $\text{card } (\text{differenceset } B A) = \text{card } (\text{differenceset } A B)$   
 by (*metis card-minusset' differenceset-commute sumset-subset-carrier*)

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

**lemma** *minusset-iterated-minusset*:  $\text{sumset-iterated } (\text{minusset } A) k = \text{minusset } (\text{sumset-iterated } A k)$   
 by (*induction k*) (*auto simp: diff-minus-set*)

**lemma** *card-sumset-iterated-minusset*:  
 $\text{card } (\text{sumset-iterated } (\text{minusset } A) k) = \text{card } (\text{sumset-iterated } A k)$   
 by (*metis card-minusset' minusset-iterated-minusset sumset-iterated-subset-carrier*)

**lemma** *finite-minusset*:  $\text{finite } A \implies \text{finite } (\text{minusset } A)$   
 by (*simp add: minusset-eq*)

**lemma** *finite-differenceset*:  $\text{finite } A \implies \text{finite } B \implies \text{finite } (\text{differenceset } A B)$   
 by (*simp add: finite-minusset finite-sumset*)

## 1.2 The Ruzsa triangle inequality

**lemma** *Ruzsa-triangle-ineq1*:  
 assumes  $U: \text{finite } U \ U \subseteq G$   
 and  $V: \text{finite } V \ V \subseteq G$   
 and  $W: \text{finite } W \ W \subseteq G$   
 shows  $(\text{card } U) * \text{card}(\text{differenceset } V W) \leq \text{card } (\text{differenceset } U V) * \text{card } (\text{differenceset } U W)$

**proof** –

have *fn*:  $\text{finite } (\text{differenceset } U V) \ \text{finite } (\text{differenceset } U W)$   
 using  $U \ V \ W$  *finite-minusset finite-sumset* by *auto*  
 have  $\exists v \ w. v \in V \wedge w \in W \wedge x = v \ominus w$  if  $x \in \text{differenceset } V W$  for  $x$   
 using *that* by (*auto simp: sumset-eq minusset-eq*)  
 then obtain  $v \ w$  where *vinV*:  $v \in V$  and *winW*:  $w \in W$  and *vw-eq*:  $v \ominus (w \ominus x) = x$

```

    if  $x \in \text{differenceset } V \ W$  for  $x$  by metis
  have  $\text{vin}G: v \ x \in G$  and  $\text{win}G: w \ x \in G$  if  $x \in \text{differenceset } V \ W$  for  $x$ 
    using  $V \ W$  that  $\text{vin}V \ \text{win}W$  by auto
  define  $\varphi$  where  $\varphi \equiv \lambda(u,x). (u \ominus (v \ x), u \ominus (w \ x))$ 
  have inj-on  $\varphi \ (U \times \text{differenceset } V \ W)$ 
  proof (clarsimp simp add:  $\varphi$ -def inj-on-def)
    fix  $u1 :: 'a$  and  $x1 :: 'a$  and  $u2 :: 'a$  and  $x2 :: 'a$ 
    assume  $u1 \in U \ u2 \in U$ 
      and  $x1: x1 \in \text{differenceset } V \ W$ 
      and  $x2: x2 \in \text{differenceset } V \ W$ 
      and  $v: u1 \ominus v \ x1 = u2 \ominus v \ x2$ 
      and  $w: u1 \ominus w \ x1 = u2 \ominus w \ x2$ 
    then obtain  $u1 \in G \ u2 \in G \ x1 \in G \ x2 \in G$ 
      by (meson  $\langle U \subseteq G \rangle \text{subset-iff sumset-subset-carrier}$ )
    show  $u1 = u2 \wedge x1 = x2$ 
  proof
    have  $v \ x1 \ominus w \ x1 = (u1 \ominus w \ x1) \ominus (u1 \ominus v \ x1)$ 
    by (smt (verit, del-insts)  $\langle u1 \in G \rangle \text{associative commutative composition-closed}$ 
inverse-closed
      invertible invertible-right-inverse2 vinG winG x1)
    also have  $\dots = (u2 \ominus w \ x2) \ominus (u2 \ominus v \ x2)$ 
      using  $v \ w$  by presburger
    also have  $\dots = v \ x2 \ominus w \ x2$ 
    by (smt (verit, del-insts)  $\langle u2 \in G \rangle \text{associative commutative composition-closed}$ 
inverse-equality
      invertible invertible-def invertible-right-inverse2 vinG winG x2)
    finally have  $v \ x1 \ominus w \ x1 = v \ x2 \ominus w \ x2$  .
    then show  $x1=x2$ 
      by (simp add:  $x1 \ x2 \ vw$ -eq)
    then show  $u1=u2$ 
      using  $\langle u1 \in G \rangle \langle u2 \in G \rangle \ w \ \text{win}G \ x1$  by force
  qed
  qed
  moreover have  $\varphi \in (U \times \text{differenceset } V \ W) \rightarrow (\text{differenceset } U \ V) \times (\text{differenceset } U \ W)$ 
    using  $\langle U \subseteq G \rangle \langle V \subseteq G \rangle \langle W \subseteq G \rangle$ 
    by (fastforce simp:  $\varphi$ -def intro: vinV winW)
  ultimately have  $\text{card } (U \times \text{differenceset } V \ W) \leq \text{card } (\text{differenceset } U \ V \times \text{differenceset } U \ W)$ 
    using card-inj fin by blast
  then show ?thesis
    by (simp flip: card-cartesian-product)
  qed

```

**definition** *Ruzsa-distance*::  $'a \ \text{set} \Rightarrow 'a \ \text{set} \Rightarrow \text{real}$   
 where *Ruzsa-distance*  $A \ B \equiv \text{card}(\text{differenceset } A \ B) / (\text{sqrt}(\text{card } A) * \text{sqrt}(\text{card } B))$

**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**  $\text{Ruzsa-distance } V W \leq (\text{Ruzsa-distance } V U) * (\text{Ruzsa-distance } U W)$   
**proof** –  
**have**  $\text{card } U * \text{card } (\text{differenceset } V W) \leq \text{card } (\text{differenceset } U V) * \text{card } (\text{differenceset } U W)$   
**using** *assms Ruzsa-triangle-ineq1 by metis*  
– now divide both sides with the same quantity  
**then have**  $\text{card } U * \text{card } (\text{differenceset } V W) / (\text{card } U * \text{sqrt } (\text{card } V) * \text{sqrt } (\text{card } W))$   
 $\leq \text{card } (\text{differenceset } U V) * \text{card } (\text{differenceset } U W) / (\text{card } U * \text{sqrt } (\text{card } V) * \text{sqrt } (\text{card } W))$   
**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**  $*$ :  $\text{card}(\text{differenceset } V W) / (\text{sqrt}(\text{card } V) * \text{sqrt}(\text{card } W)) \leq$   
 $\text{card } (\text{differenceset } U V) * \text{card } (\text{differenceset } U W)$   
 $/ (\text{card } U * \text{sqrt}(\text{card } V) * \text{sqrt}(\text{card } W))$   
**using** *assms by simp*  
**have**  $\text{card } (\text{differenceset } U V) * \text{card } (\text{differenceset } U W) / (\text{card } U * \text{sqrt}(\text{card } V) * \text{sqrt}(\text{card } W))$   
 $= \text{card}(\text{differenceset } V U) / (\text{sqrt}(\text{card } U) * \text{sqrt}(\text{card } V)) * \text{card}(\text{differenceset } U W) / (\text{sqrt}(\text{card } U) * \text{sqrt}(\text{card } W))$   
**using** *assms*  
**by** (*simp add: divide-simps*) (*metis card-minusset differenceset-commute minus-minusset*)  
**then have**  
 $\text{card}(\text{differenceset } V W) / (\text{sqrt}(\text{card } V) * \text{sqrt}(\text{card } W)) \leq$   
 $\text{card}(\text{differenceset } V U) / (\text{sqrt}(\text{card } U) * \text{sqrt}(\text{card } V)) * \text{card}(\text{differenceset } U W) / (\text{sqrt}(\text{card } U) * \text{sqrt}(\text{card } W))$   
**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**  $K0$ :  $\text{card } (\text{sumset } A0 B) \leq K0 * \text{real } (\text{card } A0)$   
**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**  $\bigwedge C. C \subseteq G \implies \text{finite } C \implies \text{card } (\text{sumset } A (\text{sumset } B C)) \leq K * \text{real } (\text{card}(\text{sumset } A C))$   
**proof**

```

define  $KS$  where  $KS \equiv (\lambda A. \text{card} (\text{sumset } A \ B) / \text{real} (\text{card } A)) \text{ ‘ } (\text{Pow } A0 - \{\{\}\})$ 
define  $K$  where  $K \equiv \text{Min } KS$ 
define  $A$  where  $A \equiv @A. A \in \text{Pow } A0 - \{\{\}\} \wedge K = \text{card} (\text{sumset } A \ B) / \text{real} (\text{card } A)$ 
obtain  $KS$ : finite  $KS$   $KS \neq \{\}$ 
  using  $KS\text{-def}$   $A0$  by blast
then have  $K \in KS$ 
  using  $K\text{-def}$   $\text{Min-in}$  by blast
then have  $\exists A. A \in \text{Pow } A0 - \{\{\}\} \wedge K = \text{card} (\text{sumset } A \ B) / \text{real} (\text{card } A)$ 
  using  $KS\text{-def}$  by blast
then obtain  $A \in \text{Pow } A0 - \{\{\}\}$  and  $\text{Keq}$ :  $K = \text{card} (\text{sumset } A \ B) / \text{real} (\text{card } A)$ 
by (metis (mono-tags, lifting)  $A\text{-def}$  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  $gt0$ :  $0 < \text{real} (\text{card} (\text{sumset } A \ B)) / \text{real} (\text{card } A)$  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  $A$   $\text{Keq}$  by presburger
have  $K\text{-card}A$ :  $K * (\text{card } A) = \text{card} (\text{sumset } A \ B)$ 
  unfolding  $\text{Keq}$  using  $\text{Keq} \langle 0 < K \rangle$  by force
have  $K\text{-le}$ :  $\text{real} (\text{card} (\text{sumset } A' \ B)) / \text{card } A' \geq K$  if  $A' \subseteq A$   $A' \neq \{\}$  for  $A'$ 
  using  $KS$   $K\text{-def}$   $KS\text{-def}$   $\langle A \subseteq A0 \rangle$  that by force
with  $A0$  have  $\text{card} (\text{sumset } A0 \ B) / \text{real} (\text{card } A0) \in KS$ 
  by (auto simp:  $KS\text{-def}$ )
with  $A0$  show  $K \leq K0$ 
  by (metis  $KS$   $K\text{-def}$   $\text{Min-le-iff}$  card-gt-0-iff mult-imp-div-pos-le of-nat-0-less-iff  $K0$ )
show  $\text{card} (\text{sumset } A \ (\text{sumset } B \ C)) \leq K * \text{real} (\text{card}(\text{sumset } A \ C))$ 
  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  $\text{real} (\text{card} (\text{sumset } A \ B)) = K * \text{real} (\text{card } A)$  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  $A' \equiv A \cap \{a. (a \oplus x) \in \text{sumset } A \ C\}$ 
with  $\langle \text{finite } A \rangle$  have finite  $A'$   $A' \subseteq A$  by auto
then have [simp]:  $\text{real} (\text{card } A - \text{card } A') = \text{real} (\text{card } A) - \text{real} (\text{card } A')$ 

```

by (meson ⟨finite A⟩ card-mono of-nat-diff)  
 have 0:  $\text{sumset } A \ C \cap \text{sumset } (A - A') \ \{x\} = \{\}$   
 by (clarsimp simp add: A'-def sumset-eq disjoint-iff) (metis IntI)  
 have 1:  $\text{sumset } A \ (\text{insert } x \ C) = \text{sumset } A \ C \cup \text{sumset } (A - A') \ \{x\}$   
 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 \*:  $\text{card } (\text{sumset } A \ (\text{insert } x \ C)) = \text{card } (\text{sumset } A \ C) + (\text{card } A - \text{card } A')$ .  
 have  $\text{sumset } A' \ (\text{sumset } B \ \{x\}) \subseteq \text{sumset } A \ (\text{sumset } B \ C)$   
 by (clarsimp simp add: A'-def sumset-eq Bex-def) (metis associative commutative composition-closed)  
 then have  $\text{sumset } A \ (\text{sumset } B \ (\text{insert } x \ C)) \subseteq \text{sumset } A \ (\text{sumset } B \ C) \cup (\text{sumset } A \ (\text{sumset } B \ \{x\}) - \text{sumset } A' \ (\text{sumset } B \ \{x\}))$   
 by (auto simp: sumset-insert2 sumset-subset-Un2)  
 then have  $\text{card } (\text{sumset } A \ (\text{sumset } B \ (\text{insert } x \ C))) \leq \text{card } (\text{sumset } A \ (\text{sumset } B \ C)) + \text{card } ((\text{sumset } A \ (\text{sumset } B \ \{x\}) - \text{sumset } A' \ (\text{sumset } B \ \{x\}))$   
 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 ... =  $\text{card } (\text{sumset } A \ (\text{sumset } B \ C)) + (\text{card } (\text{sumset } A \ (\text{sumset } B \ \{x\})) - \text{card } (\text{sumset } A' \ (\text{sumset } B \ \{x\})))$   
 by (simp add: ⟨A' ⊆ A⟩ ⟨finite A'⟩ ⟨finite B⟩ card-Diff-subset finite-sumset sumset-mono)  
 also have ... ≤  $\text{card } (\text{sumset } A \ (\text{sumset } B \ C)) + (\text{card } (\text{sumset } A \ B) - \text{card } (\text{sumset } A' \ B))$   
 using ⟨finite A⟩ ⟨finite A'⟩ ⟨finite B⟩ by (simp add: card-sumset-singleton-eq finite-sumset flip: sumset-assoc)  
 also have ... ≤  $K * \text{card } (\text{sumset } A \ C) + (K * \text{card } A - K * \text{card } A')$   
 proof (cases A' = {x})  
 case True  
 with local.insert ⟨C ⊆ G⟩ K-cardA show ?thesis by auto  
 next  
 case False  
 then have  $K * \text{card } A' \leq \text{real } (\text{card } (\text{sumset } A' \ B))$   
 using K-le[OF ⟨A' ⊆ A⟩] by (simp add: divide-simps split: if-split-asm)  
 then have  $\text{real } (\text{card } (\text{sumset } A \ B) - \text{card } (\text{sumset } A' \ B)) \leq K * \text{real } (\text{card } A) - K * \text{real } (\text{card } A')$   
 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 ... ≤ K * real (card (sumset A (insert x C)))
  using * ⟨A' ⊆ 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) ≤ K * real (card A)
    and A: finite A A ⊆ G A ≠ {}
    and B: finite B B ⊆ G
  obtains A' where A' ⊆ A A' ≠ {}
    ∧ r. card (sumset A' (sumset-iterated B r)) ≤ K ^ r * real (card A')
proof (cases B = {})
case True
  have K ≥ 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)))
    ≤ K ^ r * real (card A) for r
    by (simp add: * Int-commute Int-absorb2 ⟨A ⊆ G⟩)
  with ⟨A ≠ {}⟩ that show ?thesis by blast
next
case False
  obtain A' K'
    where A': A' ⊆ A A' ≠ {} 0 < K' K' ≤ K
      and A'-card: ∧ C. C ⊆ G ⇒ finite C ⇒ card (sumset A' (sumset B C))
        ≤ K' * real (card (sumset A' C))
    by (metis A B Plu-2-2 K False)
  with A have A' ⊆ G by blast
  have *: card (sumset A' (sumset-iterated B (Suc r))) ≤ K' * card (sumset A'
    (sumset-iterated B r))
    (is ?lhs ≤ ?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 ... ≤ ?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)) ≤ K' ^ r * real (card A') for r
  proof (induction r)
    case 0
    with ⟨A' ⊆ 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 sub-
set-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)
  qed

```

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

Special case where  $B = - A$

**lemma** *Cor-Plu-2-3-2*:

**assumes**  $K$ :  $\text{card}(\text{differenceset } A \ A) \leq K * \text{real}(\text{card } A)$   
**and**  $A$ : *finite*  $A$   $A \subseteq G$   $A \neq \{\}$   
**shows**  $\text{card}(\text{sumset-iterated } A \ r) \leq K^{\wedge}r * \text{real}(\text{card } A)$   
**proof** –  
**have**  $\text{card } A > 0$   
**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*:  $\bigwedge r. \text{card}(\text{sumset } A' (\text{sumset-iterated}(\text{minusset } A) \ r)) \leq K^{\wedge}r * \text{real}(\text{card } A')$   
**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**  $\text{card}(\text{sumset-iterated } A \ r) \leq \text{card}(\text{sumset } A' (\text{sumset-iterated}(\text{minusset } A) \ r))$   
**by** (*metis A(1) card-le-sumset card-sumset-iterated-minusset finite-minusset finite-sumset-iterated sumset-iterated-subset-carrier*)  
**also have**  $\dots \leq K^{\wedge}r * \text{real}(\text{card } A')$   
**using**  $A'$ -*card* **by** *meson*  
**also have**  $\dots \leq K^{\wedge}r * \text{real}(\text{card } A)$   
**by** (*simp add:  $\langle A' \subseteq A \rangle \langle \text{finite } A \rangle \langle 0 \leq K \rangle$  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** *Pluenncke-Ruzsa-ineq*:

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

**shows**  $\text{card}(\text{differenceset}(\text{sumset-iterated } B \ r) (\text{sumset-iterated } B \ s)) \leq K^{\wedge(r+s)}$   
 $* \text{real}(\text{card } A)$   
**proof** –  
**have**  $\text{card } A > 0$   
**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**  $A'\text{-le}$ :  $\bigwedge r. \text{card}(\text{sumset } A' (\text{sumset-iterated } B \ r)) \leq K^{\wedge r} * \text{real}(\text{card } A')$   
**using** *Cor-Plu-2-3 assms by metis*  
**define**  $C$  **where**  $C \equiv \text{minusset } A'$   
**have**  $\text{minusset } C = A'$  **and**  $C \neq \{\}$  **and**  $\text{card}A$ :  $\text{card } A' \leq \text{card } A$  **and**  $\text{card}C$ :  
 $\text{card } C = \text{card } A'$   
**using**  $A' \ A \ \text{card-mono}$  **by** (*auto simp: C-def card-minusset' Int-absorb2*)  
**then have**  $\text{card}CA$ :  $\text{card } C \leq \text{card } A$  **by** *linarith*  
**have**  $\bigwedge r. \text{card}(\text{differenceset } C (\text{sumset-iterated } B \ r)) \leq K^{\wedge r} * \text{real}(\text{card } A')$   
**using**  $A'\text{-le } C\text{-def } \text{card-minusset}' \ \text{diff-minus-set } \text{sumset-subset-carrier}$  **by** *presburger*  
**then have**  $r$ :  $\text{card}(\text{differenceset } C (\text{sumset-iterated } B \ r)) \leq K^{\wedge r} * \text{real}(\text{card } C)$   
**and**  $s$ :  $\text{card}(\text{differenceset } C (\text{sumset-iterated } B \ s)) \leq K^{\wedge s} * \text{real}(\text{card } C)$   
**using**  $\text{card}C$  **by** *presburger+*  
**have**  $\text{card } C > 0$   
**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**  $\text{card } C * \text{card}(\text{differenceset}(\text{sumset-iterated } B \ r) (\text{sumset-iterated } B \ s))$   
 $\leq \text{card}(\text{differenceset } C (\text{sumset-iterated } B \ r)) * \text{card}(\text{differenceset } C (\text{sumset-iterated } B \ s))$   
**by** (*meson Ruzsa-triangle-ineq1 B card-gt-0-iff finite-sumset-iterated sumset-iterated-subset-carrier*)  
**also have**  $\dots \leq K^{\wedge(r+s)} * \text{card } C * \text{card } C$   
**using** *mult-mono [OF r s] <0 ≤ K>* **by** (*simp add: power-add field-simps*)  
**finally have**  $\text{card}(\text{differenceset}(\text{sumset-iterated } B \ r) (\text{sumset-iterated } B \ s)) \leq K^{\wedge(r+s)} * \text{card } C$   
**using**  $\langle \text{card } C > 0 \rangle$  **by** (*simp add: field-simps*)  
**then show** *?thesis*  
**by** (*smt (verit, ccfv-SIG) <0 ≤ 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** *Pluenecke-Ruzsa-ineq-alt*:  
**assumes** *finite A*  $A \subseteq G$   
**and**  $\text{card}(\text{sumset } A \ A) \leq K * \text{real}(\text{card } A)$   $r > 0$   $s > 0$   
**shows**  $\text{card}(\text{differenceset}(\text{sumset-iterated } A \ r) (\text{sumset-iterated } A \ s)) \leq K^{\wedge(r+s)}$   
 $* \text{real}(\text{card } A)$   
**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  $A$   $A \subseteq G$ 
  and card (differenceset  $A$   $A$ )  $\leq K * \text{real}(\text{card } A)$   $r > 0$   $s > 0$ 
  shows card (differenceset (sumset-iterated  $A$   $r$ ) (sumset-iterated  $A$   $s$ ))  $\leq K^{\wedge(r+s)}$ 
  * 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  $u::\text{int}$ 
  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  $A: A \neq \{\}$  finite  $A$  and  $B: B \neq \{\}$  finite  $B$ 
  shows card(int.sumset  $A$   $B$ )  $\geq (\text{card } A) + (\text{card } B) - 1$ 
  using  $A$ 
proof (induction card  $A$  arbitrary: A)
  case (Suc  $n$ )
  define  $a$  where  $a = \text{Max } A$ 
  define  $A'$  where  $A' \equiv A - \{a\}$ 
  then obtain  $a: a \in A$   $A' = A - \{a\}$  finite  $A'$   $a \notin A'$  and  $A: A = \text{insert } a$   $A'$ 
    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 ...  $\leq$  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 ...  $\leq$  card (int.sumset A B)
proof (rule card-mono)
  show finite (int.sumset A B)
    using B Suc.premis 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.