The Erdős–Ginzburg–Ziv Theorem

Arthur F. Ramos David Barros Hulak Ruy J. G. B. de Queiroz

May 29, 2026

Abstract

We formalize the Erdős–Ginzburg–Ziv theorem for integer multisets: every multiset of at least $2n-1$ integers contains a submultiset of cardinality n whose sum is divisible by n. The proof is split into a prime-modulus argument over residue multisets and a strong-induction argument for the general case, following the classical theorem of Erdős, Ginzburg, and Ziv [1]. AI assistance was used for proof engineering. The final definitions, statements, and proofs are checked by Isabelle.

1 Introduction

The Erdős–Ginzburg–Ziv theorem is a classical result in additive number theory. In the form formalized here, every multiset of at least $2n-1$ integers contains an n-element submultiset whose sum is congruent to 0 modulo n.

This Isabelle/HOL development separates the finite combinatorial arguments from the induction on the modulus. The prime case is handled on residue multisets modulo p by sorting the nontrivial case and reducing the final choice to a subset-sum problem over $\mathbb{Z}/p\mathbb{Z}$. The general case factors a composite modulus as $n = mp$, extracts many prime-sized zero-sum blocks, and applies strong induction to the multiset of block sums divided by p.

2 Organization

The session is split into three main theories:

- `EGZ_Basics` develops modular translations and subset-sum coverage.
- `EGZ_Prime` proves the prime-modulus theorem.
- `Erdos_Ginzburg_Ziv` derives the full theorem for arbitrary positive moduli.

Contents

theory *EGZ_Basics*
imports
Main
HOL−Library.Multiset
HOL−Number_Theory.Cong
HOL−Computational_Algebra.Primes
begin

3 Residue and subset-sum infrastructure

This theory isolates the finite combinatorial infrastructure used in the prime case. We work with a recursive set of subset sums modulo p, together with modular translations on the residue set *residues p*. The key output is that a list of $p - 1$ nonzero residues modulo a prime already generates every residue class by subset summation.

3.1 Residues and modular translations

definition *residues* :: *nat* ⇒ *int set* **where**
residues p = {*0*..<*int p*}

definition *mod_translate* :: *nat* ⇒ *int* ⇒ *int set* ⇒ *int set* **where**
mod_translate p d A = ((λ*x*. (*x* + *d*) *mod int p*) ‘ *A*)

definition *list_index_sum* :: *int list* ⇒ *nat set* ⇒ *int* **where**
list_index_sum xs I = (∑ *i*∈*I*. *xs* ! *i*)

fun *mod_subset_sums* :: *nat* ⇒ *int list* ⇒ *int set* **where**
mod_subset_sums p [] = {*0*}
| *mod_subset_sums p* (*d* # *ds*) = *mod_subset_sums p ds* ∪ *mod_translate p d* (*mod_subset_sums p ds*)

lemma *residues_finite* [*simp*]:
finite (*residues p*)
⟨*proof*⟩

lemma *card_residues* [*simp*]:
assumes $0 < p$
shows *card* (*residues* p) $= p$
⟨*proof*⟩

lemma *mod_translate_iff*:
$x \in$ *mod_translate* $p\ d\ A \longleftrightarrow (\exists\, a{\in}A.\ x = (a + d)$ *mod int* $p)$
⟨*proof*⟩

lemma *mod_translate_subset_residues*:
assumes $0 < p$
assumes $A \subseteq$ *residues* p
shows *mod_translate* $p\ d\ A \subseteq$ *residues* p
⟨*proof*⟩

lemma *mod_translate_inj_on_residues*:
assumes $0 < p$
shows *inj_on* $(\lambda x.\ (x + d)$ *mod int* $p)$ (*residues* p)
⟨*proof*⟩

lemma *card_mod_translate_eq*:
assumes $0 < p$
assumes $A \subseteq$ *residues* p
shows *card* (*mod_translate* $p\ d\ A$) $=$ *card* A
⟨*proof*⟩

lemma *mod_translate_compose*:
assumes $0 < p$
shows *mod_translate* $p\ c$ (*mod_translate* $p\ d\ A$) $=$ *mod_translate* $p\ (c + d)\ A$
⟨*proof*⟩

lemma *mod_subset_sums_contains_zero* [*simp*]:
$0 \in$ *mod_subset_sums* $p\ ds$
⟨*proof*⟩

lemma *mod_subset_sums_nonempty* [*simp*]:
mod_subset_sums $p\ ds \neq \{\}$
⟨*proof*⟩

lemma *mod_subset_sums_subset_residues*:
assumes $0 < p$
shows *mod_subset_sums* $p\ ds \subseteq$ *residues* p
⟨*proof*⟩

lemma *mod_mult_inj_on_residues*:
assumes *prime_p*: *prime* p
assumes *d_nz*: d *mod int* $p \neq 0$
shows *inj_on* $(\lambda x.\ (x * d)$ *mod int* $p)$ (*residues* p)
⟨*proof*⟩

lemma *image_mult_residues*:
assumes *prime_p*: *prime* p

assumes *d_nz*: *d mod int p* $\neq$ *0*
shows $((\lambda x.\ (x * d)\ mod\ int\ p)$ ‘ *residues p*) = *residues p*
⟨*proof*⟩

lemma *mod_translate_eq_self_imp_full*:
assumes *prime_p*: *prime p*
assumes *A_sub*: *A* $\subseteq$ *residues p*
assumes *A_nonempty*: *A* $\neq$ {}
assumes *A_fix*: *mod_translate p d A* = *A*
assumes *d_nz*: *d mod int p* $\neq$ *0*
shows *A* = *residues p*
⟨*proof*⟩

lemma *mod_translate_proper_union_grows*:
assumes *prime_p*: *prime p*
assumes *A_sub*: *A* $\subseteq$ *residues p*
assumes *A_nonempty*: *A* $\neq$ {}
assumes *A_proper*: *A* $\neq$ *residues p*
assumes *d_nz*: *d mod int p* $\neq$ *0*
shows *card* (*A* $\cup$ *mod_translate p d A*) > *card A*
⟨*proof*⟩

3.2 Growth of subset sums

lemma *card_mod_subset_sums_lower_bound*:
assumes *prime_p*: *prime p*
assumes *nonzero*: $\forall$ *d*$\in$*set ds. d mod int p* $\neq$ *0*
shows *card* (*mod_subset_sums p ds*) $\geq$ *min p* (*Suc* (*length ds*))
⟨*proof*⟩

3.3 Realizing subset sums by index sets

lemma *mod_subset_sums_iff_nths*:
x $\in$ *mod_subset_sums p ds* $\longleftrightarrow$ ($\exists$ *I*. *I* $\subseteq$ {..<*length ds*} $\wedge$ *x* = *sum_list* (*nths ds I*) *mod int p*)
⟨*proof*⟩

lemma *mod_subset_sums_eq_residues*:
assumes *prime_p*: *prime p*
assumes *len*: *length ds* = *p* − *1*
assumes *nonzero*: $\forall$ *d*$\in$*set ds. d mod int p* $\neq$ *0*
shows *mod_subset_sums p ds* = *residues p*

The lower-bound lemma shows that each nonzero increment strictly enlarges the current set of subset sums until all p residues have been reached. Since all subset sums stay inside *residues p*, cardinality forces equality with the full residue system.

⟨*proof*⟩

lemma *sum_list_nths_eq_list_index_sum*:
assumes *I_sub*: *I* $\subseteq$ {..<*length xs*}
shows *sum_list* (*nths xs I*) = *list_index_sum xs I*
⟨*proof*⟩

lemma *mod_subset_sums_iff_list_index_sum*:

x ∈ mod_subset_sums p ds ⟷ (∃ I. I ⊆ {..<length ds} ∧ x = list_index_sum ds I mod int p)
⟨*proof*⟩

lemma *mset_nths_subseteq*:
mset (nths xs I) ⊆# mset xs
⟨*proof*⟩

lemma *list_index_sum_partition*:
assumes *I_sub*: *I ⊆ {..<length xs}*
shows *list_index_sum xs ({..<length xs} − I) + list_index_sum xs I = sum_list xs*
⟨*proof*⟩

lemma *list_index_sum_map2_diff*:
assumes *len*: *length xs = length ys*
assumes *I_sub*: *I ⊆ {..<length xs}*
shows *list_index_sum (map2 (λx y. y − x) xs ys) I = list_index_sum ys I − list_index_sum xs I*
⟨*proof*⟩

end
theory *EGZ_Prime*
imports
EGZ_Basics
begin

4 The prime case

The prime-modulus argument works entirely on residues modulo a prime p. After reducing an integer multiset modulo p, there are two possibilities: either some residue occurs at least p times, yielding an immediate zero-sum block, or every residue occurs fewer than p times. In the latter case we sort the remaining residues, pair the lower and upper halves, and use the subset-sum coverage theorem from the basics theory on the resulting list of nonzero gaps.

4.1 Compatibility of reduction modulo p

lemma *sum_mset_mod_image*:
sum_mset (image_mset (λx::int. x mod m) M) mod m = sum_mset M mod m
⟨*proof*⟩

4.2 Sorted pairings and nonzero gaps

lemma *sorted_nth_gap*:
assumes *prime_p*: *prime p*
assumes *sorted_ys*: *sorted ys*
assumes *len*: *length ys = 2 ∗ (p − 1)*
assumes *count_bound*: *∀ r. count (mset ys) r < p*
assumes *i_lt*: *i < p − 1*
shows *ys ! i < ys ! (i + (p − 1))*
⟨*proof*⟩

lemma *pair_differences_nonzero*:
assumes *prime_p*: *prime p*

assumes *sorted_ys*: *sorted ys*
assumes *len*: *length ys* = *2* ∗ (*p* − *1*)
assumes *ys_res*: *set ys* ⊆ *residues p*
assumes *count_bound*: ∀ *r*. *count* (*mset ys*) *r* < *p*
shows ∀ *d*∈*set* (*map2* (λ*a b*. *b* − *a*) (*take* (*p* − *1*) *ys*) (*drop* (*p* − *1*) *ys*)). *d mod int p* ≠ *0*
⟨*proof*⟩

4.3 Choosing one element from each pair

lemma *paired_choice_length*:
assumes *len_ys*: *length ys* = *2* ∗ *n*
assumes *I_sub*: *I* ⊆ {..<*n*}
shows *length* (*nths* (*take n ys*) ({..<*n*} − *I*) @ *nths* (*drop n ys*) *I*) = *n*
⟨*proof*⟩

lemma *paired_choice_subset*:
mset (*nths* (*take n ys*) ({..<*n*} − *I*) @ *nths* (*drop n ys*) *I*) ⊆# *mset ys*
⟨*proof*⟩

lemma *paired_choice_sum*:
assumes *len_ys*: *length ys* = *2* ∗ *n*
assumes *I_sub*: *I* ⊆ {..<*n*}
shows *sum_list* (*nths* (*take n ys*) ({..<*n*} − *I*) @ *nths* (*drop n ys*) *I*) =
sum_list (*take n ys*) + *list_index_sum* (*map2* (λ*a b*. *b* − *a*) (*take n ys*) (*drop n ys*)) *I*
⟨*proof*⟩

4.4 The residue-valued prime theorem

In the nontrivial branch we remove one distinguished residue z, sort the remaining residues, split them into lower and upper halves, and consider the pairwise differences. Those differences are nonzero modulo p, so the subset-sum coverage theorem from the basics theory can realize the correction term needed to turn the lower half into a p-element zero-sum submultiset.

lemma *prime_residue_zero_sum_submultiset*:
assumes *prime_p*: *prime p*
assumes *size_R*: *size R* = *2* ∗ *p* − *1*
assumes *R_sub*: *set_mset R* ⊆ *residues p*
shows ∃ *N*. *N* ⊆# *R* ∧ *size N* = *p* ∧ *sum_mset N mod int p* = *0*
⟨*proof*⟩

4.5 Lifting the prime theorem back to integers

theorem *prime_egz*:
assumes *prime_p*: *prime p*
assumes *size_M*: *size M* = *2* ∗ *p* − *1*
shows ∃ *N*. *N* ⊆# *M* ∧ *size N* = *p* ∧ *sum_mset N mod int p* = *0*
⟨*proof*⟩

end
theory *Erdos_Ginzburg_Ziv*
imports
EGZ_Prime
HOL−Computational_Algebra.Primes

begin

5 The full Erdős-Ginzburg-Ziv theorem

The composite-modulus case is obtained from the prime case by strong induction. Writing n as $m * p$ with p prime, we repeatedly extract p-element zero-sum blocks, divide their sums by p, and apply the induction hypothesis to the resulting multiset of quotients.

5.1 Auxiliary multiset decompositions

lemma *exists_submultiset_of_size*:
fixes $M :: 'a::linorder\ multiset$
assumes *n_le*: $n \leq size\ M$
shows $\exists N.\ N \subseteq\# M \wedge size\ N = n$
⟨*proof*⟩

lemma *union_blocks_size*:
fixes $Bs :: 'a\ multiset\ multiset$
assumes *blocks_size*: $\forall B{\in}\#Bs.\ size\ B = n$
shows $size\ (sum\_mset\ Bs) = size\ Bs * n$
⟨*proof*⟩

lemma *union_blocks_div_sum*:
assumes *blocks_div*: $\forall B{\in}\#Bs.\ sum\_mset\ B\ mod\ int\ p = 0$
shows $sum\_mset\ (sum\_mset\ Bs) = int\ p * sum\_mset\ (image\_mset\ (\lambda B.\ sum\_mset\ B\ div\ int\ p)\ Bs)$
⟨*proof*⟩

lemma *extract_prime_blocks*:
assumes *prime_p*: *prime p*
assumes *size_M*: $size\ M = k * p + (p - 1)$
shows $\exists Bs\ R.\ M = sum\_mset\ Bs + R \wedge size\ Bs = k \wedge size\ R = p - 1 \wedge$
$(\forall B{\in}\#Bs.\ size\ B = p \wedge sum\_mset\ B\ mod\ int\ p = 0)$
⟨*proof*⟩

5.2 Strong induction on the modulus

The prime branch is handled directly by *prime_egz*. For a composite modulus n, the previous lemma extracts $2 * m - 1$ prime-sized blocks with sums divisible by p. Applying the induction hypothesis to the block sums divided by p selects enough blocks whose union has size n and total sum divisible by n.

theorem *erdos_ginzburg_ziv_exact*:
assumes *n_pos*: $0 < n$
assumes *size_M*: $size\ M = 2 * n - 1$
shows $\exists N.\ N \subseteq\# M \wedge size\ N = n \wedge sum\_mset\ N\ mod\ int\ n = 0$
⟨*proof*⟩

5.3 The monotone cardinality form

theorem *erdos_ginzburg_ziv*:
assumes *n_pos*: $0 < n$

assumes *size_M*: *size M* $\geq$ *2* $*$ *n* $-$ *1*
shows $\exists N.\ N \subseteq\# M \wedge$ *size N* $=$ *n* $\wedge$ *sum_mset N mod int n* $=$ *0*
⟨*proof*⟩

end
theory *Erdos_Ginzburg_Ziv_Outline*
imports
Erdos_Ginzburg_Ziv
begin

6 Overview

This entry formalizes the Erdős-Ginzburg-Ziv theorem for integer multisets. The development is layered as follows: the basics theory develops modular subset-sum infrastructure, the prime theory proves the prime-modulus case, and the final theory derives the full theorem for arbitrary positive moduli by strong induction.

7 Main results

The main exported facts are *prime_egz*, for prime moduli, and *erdos_ginzburg_ziv*, for the arbitrary-modulus monotone form. The latter states that every integer multiset of size at least *2* $*$ *n* $-$ *1* has an n-element submultiset whose sum is divisible by n.

8 Proof architecture

In the prime case, either some residue already occurs p times or the residues can be sorted and paired so that the lower-to-upper gaps are all nonzero modulo p. Subset sums of those gaps then cover the entire residue system and yield the required correction term.

For composite n, write $n = m * p$ with p prime. Repeated applications of the prime theorem produce *2* $*$ *m* $-$ *1* disjoint p-blocks whose sums are divisible by p; applying the induction hypothesis to the corresponding quotients selects blocks whose union gives the desired n-element zero-sum submultiset.

end

References

[1] P. Erdős, A. Ginzburg, and A. Ziv. Theorem in the additive number theory. *Bulletin of the Research Council of Israel, Section F: Mathematics and Physics*, 10F(1):41–43, 1961. Reprint: `https://renyi.hu/~p_erdos/1961-25.pdf`.