Formalizing the solution to the cap set problem

Sander R. Dahmen, Johannes Hölzl,
Robert Y. Lewis

Vrije Universiteit Amsterdam

ITP 2019
September 12, 2019

Motivation

Lean Forward

A new project at the VU: formalize modern results in number theory, in Lean.

- Develop comprehensive libraries that will help with many results.
- Target “research areas”/collections of moderate difficulty results, instead of single challenge theorems.
- Work on the system and automation alongside the formalizing.
- PI: Jasmin Blanchette

Can we formalize current results yet?

Sander Dahmen's first proposal: formalize Ellenberg and Gijswijt's solution to the cap set problem.

- Recent: *Annals of Mathematics*, 2017
- The theorem can be stated in elementary terms.
- The proof does not depend on any high-powered results, but…
- it uses a lot of elementary linear algebra: a good stress test.
- The "second half" of the proof can be made even more elementary.

Can we formalize current results yet? Yes! *

We have completed a proof of Ellenberg and Gijswijt's theorem in Lean.

- The first half of our proof is faithful to their argument.
- The second half takes a much more elementary approach.
- A lot of linear algebra, combinatorics, etc. was added to Lean's `mathlib`.
- We followed a detailed informal blueprint by Sander.

Paper and blueprint: `https://lean-forward.github.io/e-g/`

Can we formalize current results yet? Yes! *

We have completed a proof of Ellenberg and Gijswijt's theorem in Lean.

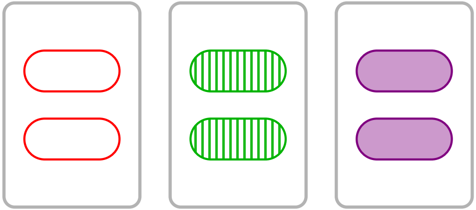
- The first half of our proof is faithful to their argument.
- The second half takes a much more elementary approach.
- A lot of linear algebra, combinatorics, etc. was added to Lean's `mathlib`.
- We followed a detailed informal blueprint by Sander.

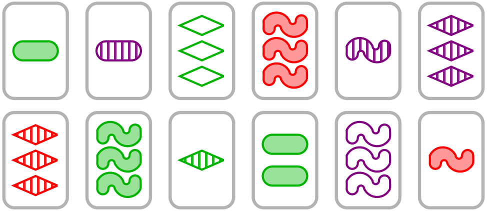
Paper and blueprint: `https://lean-forward.github.io/e-g/`

(*) This was a very special case.

The cap set problem

The cap set problem





The cap set problem

Specific statement

Let $r_3(G)$ denote the cardinality of a largest subset of an abelian group G containing no three-term arithmetic progression. Is there a constant $c < 3$ such that $r_3((\mathbb{Z}/3\mathbb{Z})^n)$ grows in n no faster than c^n?

The cap set problem

Specific statement

Let $r_3(G)$ denote the cardinality of a largest subset of an abelian group G containing no three-term arithmetic progression. Is there a constant $c < 3$ such that $r_3((\mathbb{Z}/3\mathbb{Z})^n)$ grows in n no faster than c^n?

General statement

Let $\alpha, \beta, \gamma \in \mathbb{F}_q$ such that $\alpha + \beta + \gamma = 0$ and $\gamma \neq 0$. Let A be a largest subset of $\mathbb{F}_q^n$ such that the equation $\alpha a_1 + \beta a_2 + \gamma a_3 = 0$ has no solutions with $a_1, a_2, a_3 \in A$ apart from those with $a_1 = a_2 = a_3$. Is there a constant $c < q$ such that $|A|$ grows in n no faster than c^n?

The cap set problem

Specific statement

Let $r_3(G)$ denote the cardinality of a largest subset of an abelian group G containing no three-term arithmetic progression. Is there a constant $c < 3$ such that $r_3((\mathbb{Z}/3\mathbb{Z})^n)$ grows in n no faster than c^n?

General statement

Let $\alpha, \beta, \gamma \in \mathbb{F}_q$ such that $\alpha + \beta + \gamma = 0$ and $\gamma \neq 0$. Let A be a largest subset of $\mathbb{F}_q^n$ such that the equation $\alpha a_1 + \beta a_2 + \gamma a_3 = 0$ has no solutions with $a_1, a_2, a_3 \in A$ apart from those with $a_1 = a_2 = a_3$. Is there a constant $c < q$ such that $|A|$ grows in n no faster than c^n?

Theorem (Ellenberg and Gijswijt, *Annals of Mathematics*, 2017)

Yes.

The cap set problem

Ellenberg and Gijswijt follow a breakthrough due to Croot, Lev, and Pach.

Idea: translate the problem to one about systems or spaces of polynomials. (the *polynomial method*)

1. Bound the size of the cap set by the dimension of a subspace of polynomials with coefficients in $\mathbb{F}_q$.
2. Control the asymptotic behavior of this bound.

The cap set problem in Lean

```
theorem general_cap_set {α : Type} [discrete_field α] [fintype α] :
∃ C B : ℝ, B > 0 ∧ C > 0 ∧ C < fintype.card α ∧
  ∀ {a b c : α} {n : ℕ} {A : finset (fin n → α)},
    c ≠ 0 → a + b + c = 0 →
    (∀ x y z : fin n → α, x ∈ A → y ∈ A → z ∈ A →
              a • x + b • y + c • z = 0 → x = y ∧ x = z) →
    ↑A.card ≤ B * C^n
```

Constructing the bound

Goal:

```
theorem thm_12_1 {α : Type} [discrete_field α] [fintype α]
  (n : ℕ) {a b c : α} (hc : c ≠ 0) (habc : a + b + c = 0)
  (hn : n > 0) {A : finset (fin n → α)}
  (ha : ∀ x y z ∈ A, a · x + b · y + c · z = 0 → x = y ∧ x = z) :
  A.card ≤ 3 * m α n (1 / 3 * ((card α - 1) * n))
```

We fix a parameter `α : Type` instantiating the type classes `[discrete_field α]` and `[fintype α]`, and `n : ℕ`. We use `q : ℕ` to abbreviate `card α`.

For `d : ℚ`, we make the following definitions:

- `M` is the set of monomials in `n` variables where the exponent of each variable is less than `q`.
- `M'` is the subset of `M` whose elements have total degree at most `d`.
- `S'` is the span of `M'`. This is a subspace of `mv_polynomial (fin n)` α.
- `m` is the dimension of `S'`.

Since `M'` is linearly independent, it follows that the cardinality of `M'` is equal to `m`.

```
def M : finset (mv_polynomial (fin n) α) :=
(finset.univ.image
  (λ f : fin n →₀ fin q, f.map_range fin.val rfl)).image
    (λ v : fin n →₀ ℕ, monomial v (1:α))

def M' (d : ℚ) : finset (mv_polynomial (fin n) α) :=
M.filter (λ m, d ≥ mv_polynomial.total_degree m)

def S' (d : ℚ) : subspace α (mv_polynomial (fin n) α) :=
submodule.span α ((M' d) : set (mv_polynomial (fin n) α))

def m (d : ℚ) : ℕ := (vector_space.dim α (S' d)).to_nat

lemma M'_card (d : ℚ) : (M' d).card = m d
```

Preliminaries

Our goal was:

```
theorem thm_12_1 {α : Type} [discrete_field α] [fintype α]
  (n : ℕ) {a b c : α} (hc : c ≠ 0) (habc : a + b + c = 0)
  (hn : n > 0) {A : finset (fin n → α)}
  (ha : ∀ x y z ∈ A, a · x + b · y + c · z = 0 → x = y ∧ x = z) :
  A.card ≤ 3 * m α n (1 / 3 * ((card α - 1) * n))
```

Fix the hypotheses, and define:

```
def neg_cA : finset (fin n → α) := A.image (λ z, (-c) · z)
```

```
def V : subspace α (S' d) :=
zero_set_subspace (S' d) (finset.univ \ neg_cA)
```

```
def V_dim : ℕ := (vector_space.dim α V).to_nat
```

We prove a sequence of lemmas controlling `V_dim`.

Bounding from below

A general theorem (following from rank-nullity):

```
theorem lemma_9_2 (T : subspace α (mv_polynomial (fin n) α))
  (A : finset (fin n → α)) :
  (vector_space.dim α zero_set_subspace).to_nat + A.card ≥
    (vector_space.dim α T).to_nat
```

From this, we derive:

```
lemma diff_card : (univ \ neg_cA).card + A.card = q^n
```

```
theorem lemma_12_2 : q^n + V_dim ≥ m d + A.card
```

Bounding from above

There is a polynomial in `V` with maximal support:

```
lemma exi_max_sup :
  ∃ P ∈ V, ∀ P' ∈ V, sup P ⊆ sup P' → sup P = sup P'
```

Define `P` to be a witness to this.

```
theorem lemma_12_3 : (sup P).card ≥ V_dim
```

Bounding from above

```
theorem lemma_12_4 : (sup P).card ≤ 2 * m (d/2)
```

This follows from a more general result:

```
theorem prop_11_1 {p : mv_polynomial (fin n) α} (A : finset (fin n → α)) :
  p ∈ S' n d → (∀ x ∈ A, ∀ y ∈ A, x ≠ y → p.eval (a · x + b · y) = 0) →
  (A.filter (λ x, p.eval (-c · x) ≠ 0)).card ≤ 2 * m (d / 2)
```

Proposition (Ellenberg and Gijswijt)

Let $A \subseteq \mathbb{F}_q^n$ and $\alpha, \beta, \gamma \in \mathbb{F}_q$ with $\alpha + \beta + \gamma = 0$. Let $P \in S_n^d$ such that for all $a, b \in A$ with $a \neq b$ we have $P(\alpha a + \beta b) = 0$. Then

$$|\{a \in A \mid P(-\gamma a) \neq 0\}| \leq 2m_{d/2}.$$

Proposition 11.1

- This was the most intricate proof in our development.
  - (In line with E-G. This lemma makes up most of their paper.)
- Stated in terms of the linear transformation `p.eval`, but more naturally proved with matrices.
- Needed to extend libraries to unify these two concepts.

Proposition 11.1 proof sketch

Given `a b : α`, `x y : fin n → α`, `p : mv_polynomial (fin n) α` with `p ∈ S' d`:

- `p.eval (a · x + b · y)` can be written as a linear combination of evaluated monomials in `M' d`.
- Define an `A × A` matrix `B` such that `B x y = p.eval (a · x + b · y)`.
- Prove that `B` factors:

```
lemma B_eq_sum_matrix : B =
  split_left.sum (λ _ _, matrix.vec_mul_vec _ _) +
  split_right.sum (λ  _ _, matrix.vec_mul_vec _ _)
```

- Cardinalities of the finite sets `split_left` and `split_right` are at most `m (d/2)`.
- Rank of `B` is at most `2 * m (d/2)`, since `matrix.vec_mul_vec` has rank at most 1.
- But `B` is diagonal, so its rank is equal to what we want to bound.

A combinatorial calculation

The last lemma relates values of `m` at different inputs.

```
theorem lemma_12_5 : q^n ≤ m ((q-1)*n - d) + m d
```

- Largely independent of the previous lemmas.
- Go by carving up the space `fin n → fin q` into subsets.
- The encoding matters!

Putting things together

```
theorem lemma_12_6 : A.card ≤ 2 * m (d/2) + m ((q-1)*n - d) :=
by linarith [lemma_12_2, lemma_12_3, lemma_12_4, lemma_12_5]
```

Abstracting the parameter `d` and instantiating it with `2/3*(q-1)*n`:

```
theorem theorem_12_1 : A.card ≤ 3*(m (1/3*((q-1)*n)))
```

Asymptotics

Controlling the growth of our bound

We want to know how our bound grows in `n`.

```
theorem theorem_12_1 : A.card ≤ 3*(m (1/3*((q-1)*n)))
```

Recall:

- `q` is the cardinality of the underlying field α.
- `m d` is the number of monomials with total degree at most `d`.

Controlling the growth of our bound

We want to know how our bound grows in `n`.

```
theorem theorem_12_1 : A.card ≤ 3*(m n (1/3*((q-1)*n)))
```

Recall:

- `q` is the cardinality of the underlying field α.
- `m n d` is the number of monomials in `n` variables with total degree at most `d`.

Controlling the growth of our bound

```
theorem general_cap_set {α : Type} [discrete_field α] [fintype α] :
∃ B C : ℝ, B > 0 ∧ C > 0 ∧ C < card α ∧
  ∀ {a b c : α} {n : ℕ} {A : finset (fin n → α)},
   c ≠ 0 → a + b + c = 0 →
  (∀ x y z ∈ A, a · x + b · y + c · z = 0 → x = y ∧ x = z) →
      A.card ≤ B * C^n
```

It suffices:

```
theorem general_cap_set' {α : Type} [discrete_field α] [fintype α] :
  ∃ B C : ℝ, B > 0 ∧ C > 0 ∧ C < card α ∧
    3*(m n (1/3*((q-1)*n))) ≤ B * C^n
```

Changing the original argument

E-G 2017, 10 lines

It is not hard to check that $m_{(q-1)n/3}/q^n$ is exponentially small as n grows with q fixed. We can be more precise. . . . By Cramér's theorem . . . $m_{(q-1)n/3}/q^n = \mathcal{O}(c^n)$ for some $c < q$.

Major simplifications suggested by Tao and Zeilberger.

We work out a different approach inspired by Zeilberger and improved by Gijswijt:

- explicit values of c for specific q
- no mathematics beyond high-school calculus

m as a sum of coefficients

We will rewrite `m` as a sum of coefficients of a certain polynomial:

$$(1 + x + \ldots + x^{q-1})^n$$

```
def one_coeff_poly (m : ℕ) : polynomial ℕ :=
(finset.range m).sum (λ k, polynomial.X ^ k)
```

m as a sum of coefficients

Informally, we define:

$$c_j^{(n)} := \left| \left\{ (a_1, \ldots, a_n) \;\middle|\; a_i \in \{0, 1, \ldots, q-1\} \text{ and } \sum_{i=1}^{n} a_i = j \right\} \right|.$$

How to encode these tuples in Lean?

m as a sum of coefficients

```
def sf (n j : ℕ) : finset (vector (fin q) n) :=
finset.univ.filter (λ f, (f.nat_sum = j))

def cf (n j : ℕ) : ℕ := (sf n j).card

theorem lemma_13_8 (n : ℕ) {d : ℚ} (hd : d ≥ 0) :
  m n d = (finset.range (⌊d⌋.nat_abs + 1)).sum (cf n)

lemma cf_mul (n j : ℕ) : cf (n+2) j =
  (finset.range (j + 1)).sum (λ i, (cf 1 (j - i)) * cf (n + 1) i)

theorem lemma_13_9 (hq : q > 0) (n j : ℕ) :
  ((one_coeff_poly q) ^ n).coeff j = cf n j
```

m as a sum of coefficients

Define:

```
def crq (r : ℝ) (q : ℕ) : ℝ :=
((one_coeff_poly q).eval₂ coe r) / r ^ ((q-1)/3)
```

For every r between 0 and 1, crq bounds m:

```
theorem theorem_14_1 {r : ℝ} (hr : 0 < r) (hr2 : r < 1) :
  m n ((q - 1)*n / 3) ≤ (crq r q) ^ n
```

(Derived from theorem_13_8 and theorem_13_9.)

Concrete bounds on m

Since `crq 1 q = q` and the derivative of `crq` with respect to `r` is positive at `r = 1`, we have from elementary calculus:

```
theorem lemma_13_15 : ∃ r : ℝ, 0 < r ∧ r < 1 ∧ crq r q < q
```

Instantiating `theorem_14_1` with such an `r`:

- `m n 1/3*(q-1)*n) ≤ (crq r q)^n`

From `theorem_12_1`:

- `A.card ≤ 3*(m n (1/3*(q-1)*n))`

Even more concrete bounds

For the motivating case when `q = 3`, we compute the optimal value `r := (real.sqrt 33 - 1) / 8`.

We show `0 < r < 1` and `crq r 3 = ((3 / 8)^3 * (207 + 33*sqrt 33))^(1/3)` (which is approximately 2.76).

```
theorem cap_set {n : ℕ} {A : finset (fin n → ℤ/3ℤ)} :
    (∀ x y z ∈ A, x + y + z = 0 → x = y ∧ x = z) →
     A.card ≤ 3 * (((3/8)^3 * (207 + 33*sqrt 33))^(1/3))^n
```

Morals

Statistics

- Ellenberg–Gijswijt proof: about 2 pages of content. (construction of bound: 1.5 pages)
- Our informal writeup: 9 pages of non-background content (construction of bound: 5 pages)
- Our formalization: 2000 lines (construction of bound: 900 lines)

Morals

- This is formalized contemporary math—rare!
- It was “smooth” (for a formalization).
- As is often the case: library development may have been the biggest gain. (`https://github.com/leanprover-community/lean-sensitivity`)
- Collaboration was essential.

- January 6-10, 2020
- Pittsburgh, PA, USA
- `http://www.andrew.cmu.edu/user/avigad/meetings/fomm2020`