Solving Problems in Additive Number Theory with Automata Theory

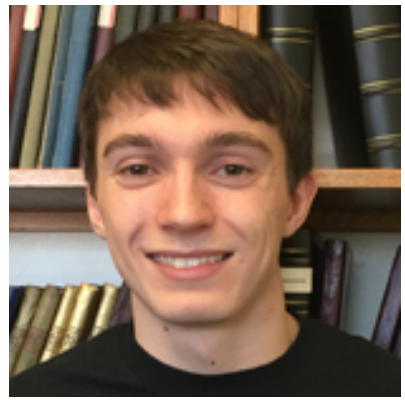
Jeffrey Shallit
School of Computer Science
University of Waterloo
Waterloo, ON N2L 3G1
Canada
shallit@uwaterloo.ca

CiE 2018 — Kiel, Germany — August 1 2018

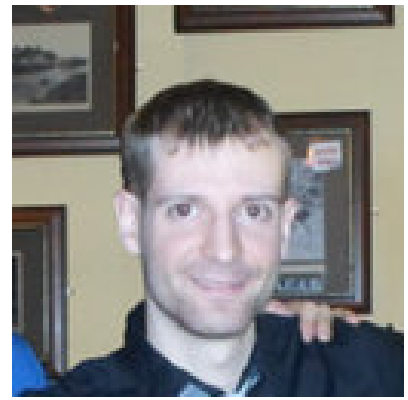
Joint Work With

Jason Bell

Kathryn Hare



Thomas F. Lidbetter



Carlo Sanna

Daniel Kane

P. Madhusudan

Dirk Nowotka

Aayush Rajasekaran

Tim Smith

Additive number theory

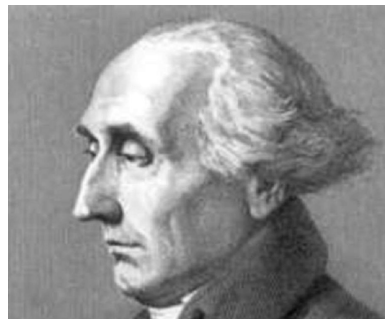
Let S, T be subsets of the natural numbers $\mathbb{N} = \{0, 1, 2, \ldots\}$.

The **principal problem** of additive number theory is to determine whether every element of T (or every sufficiently large element of T) can be written as the sum of some **constant** number of elements of S, not necessarily distinct.

Often (but not always) $T = \mathbb{N}$ and S is a relatively sparse subset of $\mathbb{N}$.

Example: Lagrange's theorem

Probably the most famous example is
Lagrange's theorem (1770):



(a) every natural number is the sum of four squares; and

(b) three squares do not suffice for numbers of the form $4^a(8k+7)$.

(Conjectured by Bachet in 1621.)

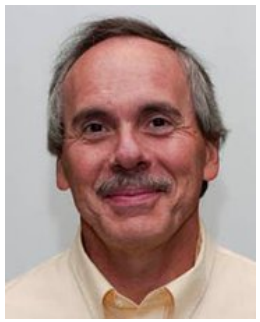
Goldbach's conjecture

Let $\mathbb{P} = \{2, 3, 5, \ldots, \}$ be the prime numbers.

Goldbach's conjecture (1742): every even number ≥ 4 is the sum of two primes.

So here $T = 2\mathbb{N}$.

Zwillinger's conjecture (1979): every even number > 4208 is the sum of 2 numbers, each of which is part of a twin-prime pair.



Additive bases

Let $S \subseteq \mathbb{N}$.

We say that a subset S is an **basis of order** h if every natural number can be written as the sum of h elements of S, not necessarily distinct.

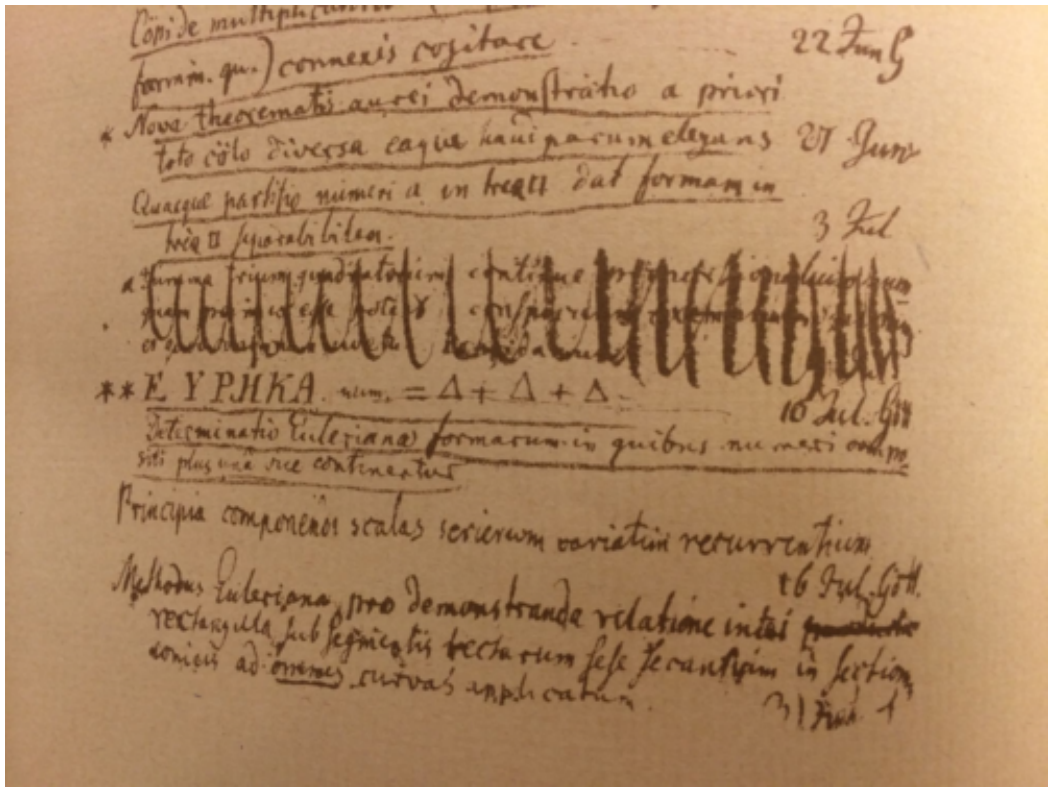
We say that a subset S is an **asymptotic basis of order** h if every *sufficiently large* natural number can be written as the sum of h elements of S, not necessarily distinct.

Usual convention: $0 \in S$.

Gauss's theorem for triangular numbers

A *triangular number* is a number of the form $n(n+1)/2$.

Gauss wrote the following in his diary on July 10 1796:



i.e., **The triangular numbers form an additive basis of order** 3

Waring's problem for powers

Edward Waring (1770) asserted,
without proof, that
every natural number is
– the sum of 4 squares
– the sum of 9 cubes
– the sum of 19 fourth powers
– "and so forth".

9. Omnis integer numerus vel eſt cubus; vel e duobus, tribus, 4, 5, 6, 7, 8, vel novem cubis compoſitus: eſt etiam quadrato-quadratus; vel e duobus, tribus, &c. uſque ad novemdecim compoſitus, & ſic deinceps: conſimilia etiam affirmari poſſunt (exceptis excipiendis) de eodem numero quantitatum earundem dimenſionum.

Waring's problem

Let $g(k)$ be the least natural number m such that every natural number is the sum of m k'th powers.

Let $G(k)$ be the least natural number m such that every sufficiently large natural number is the sum of m k'th powers.

Proving that $g(k)$ and $G(k)$ exist, and determining their values, is **Waring's problem**.

By Lagrange we know $g(2) = G(2) = 4$.

Hilbert proved in 1909 that $g(k)$ and $G(k)$ exist for all k.

By Wieferich and Kempner we know $g(3) = 9$.

We know that $4 \leq G(3) \leq 7$, but the true value is still unknown.

How have problems in additive number theory been proved, traditionally?

- Waring's problem: solved by Hilbert in 1909 using polynomial identities and geometry of numbers.
- Hardy & Littlewood: in 1920, introduced their *circle method* from complex analysis.
  - use powers of generating functions like $\sum_{n \geq 0} X^{n^2}$ and complex analysis
  - compute the residues around 0
  - break unit circle up into "major arcs" and "minor arcs" (the latter containing the main singularities)
- I. M. Vinogradov: in 1926 modified the Hardy-Littlewood method, replacing exponential sums with trigonometric sums, to attack Goldbach's conjecture.
- Linnik: in 1943, used Schnirelmann's method instead (a much more elementary approach).

What could formal languages and automata theory possibly offer the number theorist?

- New kinds of sets to consider (i.e., sets of numbers accepted by automata of various kinds)
- New approaches for proving results
- Replacing long case-based arguments with a single machine computation

Other additive bases?

What other kinds of sets can form additive bases for $\mathbb{N}$?

Not the powers of 2 – too sparse.

Need a set whose natural density is at least $N^{1/k}$ for some k.

But this is not sufficient: consider the set

$$S = \{2^{2^n} + i \ : \ n \geq 1 \text{ and } 0 \leq i < 2^n\}.$$

Its density is $\Omega(N^{1/2})$.

But S does not form an additive basis of any finite order, because adding k elements of S in decreasing order can only result in at most $2k + 1$ "one" bits in the highest-order positions.

A simple example: the OOPS numbers

A number is said to be an *OOPS number* if it has ones in all odd-numbered positions of its binary representation (where the most significant digit is position 1).

The first few OOPS numbers are

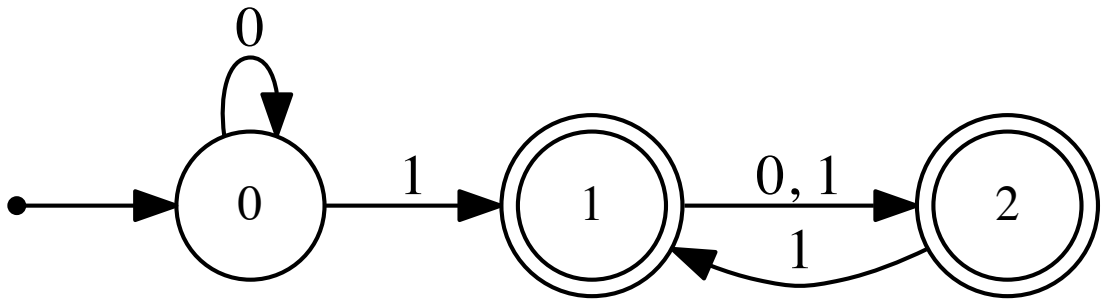
$$1, 2, 3, 5, 7, 10, 11, 14, 15, 21, 23, 29, 31, 42, 43, 46, 47, 58, 59, \ldots$$

The density of the OOPS numbers is $\Theta(N^{1/2})$, so they are a good candidate for forming an additive basis.

Theorem. The OOPS numbers form an asymptotic additive basis of order 3, but not of order 2.

Proving the OOPS result

Let's observe that the OOPS numbers are recognized by the following simple DFA, which takes the base-2 representation of n as input:



Now we can use a great theorem, due to Büchi and Bruyère:

Theorem. Every first-order statement about a sequence defined by a finite automaton in base k, using operations such as addition, comparison, logical operations, and the $\exists$ and $\forall$ quantifiers, is decidable. Furthermore, there is an automaton recognizing those inputs corresponding to values of the free variables that make the statement true.

Proving the OOPS result

The first-order statements for additive number theory are generally rather easy. For example, here's the statement defining those n that are the sum of two OOPS numbers:

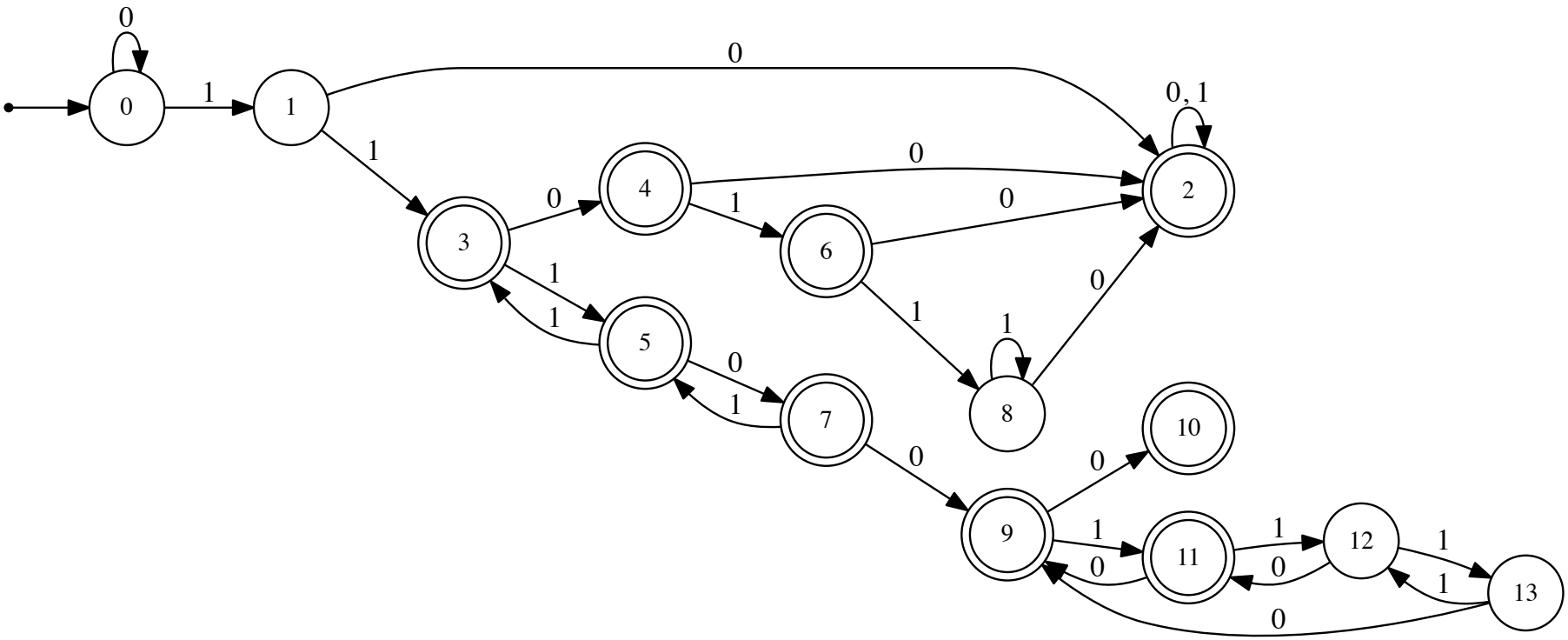
$$\exists x, y\, (n = x + y) \ \wedge \ \text{oops}(x) = 1 \ \wedge \ \text{oops}(y) = 1.$$

We can use the `Walnut` software package, written by Hamoon Mousavi, to consruct an automaton recognizing those n for which this statement is true:

```
eval oop2 "E x,y (n=x+y) & OOP[x]=@1 & OOP[y]=@1":
```

Sums of two OOPS numbers

This gives us the following automaton:



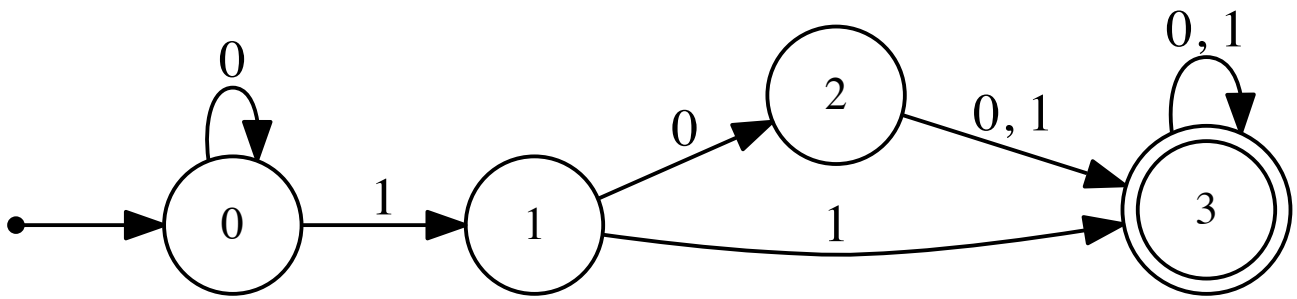
State 8 (a non-accepting state) is recurrent, so the OOPS numbers don't form an asymptotic additive basis of order 2.

Sums of three OOPS numbers

By contrast, the Walnut statement for the sum of three OOPS numbers

```
eval oop3 "E x,y,z (n=x+y+z) & OOP[x]=@1 &
           OOP[y]=@1 & OOP[z] = @1":
```

returns the following automaton



thus proving that every number ≥ 3 is the sum of three OOPS numbers.

A recent theorem of Bell, Hare, and JOS

Theorem. If a set S of natural numbers is recognized (in base k) by a finite automaton of n states, then

(a) it is decidable if S forms an additive basis (resp., an asymptotic additive basis) of finite order;

(b) if S is a basis, there is a computable bound (as a function of k and n) on the order of the basis;

(c) the minimal order is computable.

But what about other sets?

We could try to do additive number theory with more complicated sets, such as

$$\mathtt{BAL} = \{0, 2, 10, 12, 42, 44, 50, 52, 56, \ldots\},$$

the numbers whose base-2 representation forms a string of balanced parentheses (where 1 represents a left paren and 0 a right paren).

Note that $\gcd(\mathtt{BAL}) = 2$, so it cannot be an additive basis for $\mathbb{N}$, but it could be an additive basis for $2\mathbb{N}$, the even numbers.

But BAL is not regular, so the previous method cannot work.

Nevertheless, we can prove results about some sets like this, using the *method of regular approximation*.

Method of regular approximation

Recent work of Bell, Lidbetter, and JOS.

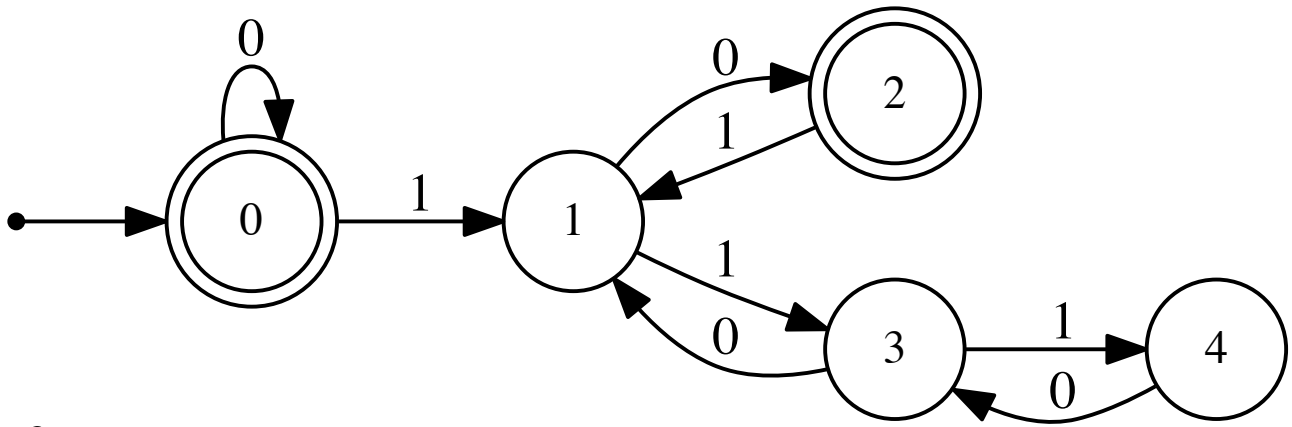
Idea: find a suitable regular language that is an *underapproximation* (subset) of `BAL`, and another regular language that is an *overapproximation* (superset) of `BAL`.

The underapproximation gives an *upper* bound on the order of the basis and the overapproximation gives a *lower* bound.

Approximating BAL

We can underapproximate BAL by considering those numbers that are balanced and have a "nesting level" of at most three.

We use the following DFA:

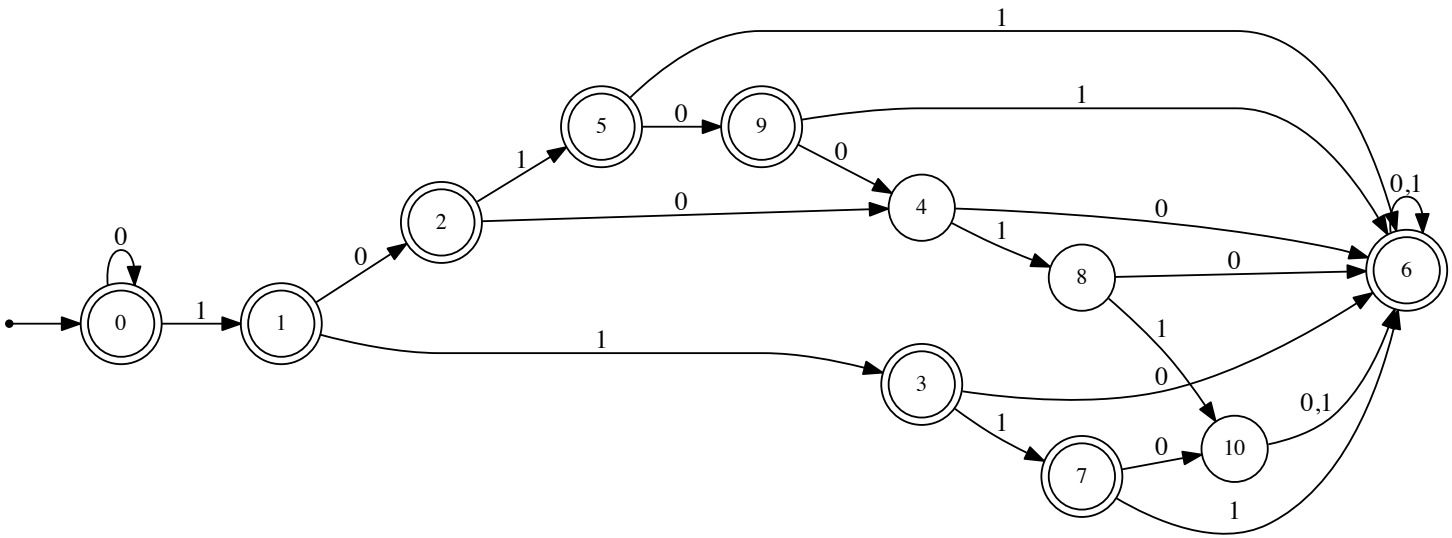


and prove

Theorem. Every even natural number except $8, 18, 28, 38, 40, 82, 166$ is the sum of 3 balanced numbers.

Balanced numbers

Proof. We use `Walnut` and get the following automaton accepting n such that $2n = x + y + z$, with x, y, z balanced.



Palindromes

- How about numbers with palindromic base-b expansions?
- A *palindrome* is any string that is equal to its reversal
- Examples are `radar` (English), `reliefpfeiler` (German), and `10001`.
- We call a natural number a *base-b palindrome* if its base-b representation (without leading zeroes) is a palindrome
- Examples are $16 = [121]_3$ and $297 = [100101001]_2$.
- Binary palindromes ($b = 2$) form sequence A006995 in the *On-Line Encyclopedia of Integer Sequences* (OEIS):
  $$0, 1, 3, 5, 7, 9, 15, 17, 21, 27, 31, 33, 45, 51, 63, \ldots$$
- They have density $\Theta(N^{1/2})$.

The problem

Do the base-b palindromes form an additive basis, and if so, of what order?

William Banks (2015) showed that every natural number is the sum of at most 49 base-10 palindromes. (*INTEGERS* **16** (2016), #A3)

Javier Cilleruelo, Florian Luca, and Lewis Baxter (2017) showed that for all bases $b \geq 5$, every natural number is the sum of three base-b palindromes. (*Math. Comp.* (2017), to appear)

What we proved

However, the case of bases $b = 2, 3, 4$ was left unsolved. We proved

Theorem. (Rajasekaran, JOS, Smith) Every natural number N is the sum of 4 binary palindromes. The number 4 is optimal.

For example,

$$\begin{aligned} 10011938 &= 5127737 + 4851753 + 32447 + 1 \\ &= [100111000111111000111001]_2 \\ &+ [100101000001000000101001]_2 \\ &+ [1111111010111111]_2 \\ &+ [1]_2. \end{aligned}$$

4 is optimal: 10011938 is not the sum of 2 binary palindromes.

Previous proofs were complicated (1)

Excerpt from Banks (2015):

2.4. **Inductive passage from $\mathbb{N}_{\ell,k}(5^+;c_1)$ to $\mathbb{N}_{\ell-1,k+1}(5^+;c_2)$.**

LEMMA 2.4. *Let $\ell, k \in \mathbb{N}$, $\ell \geqslant k+6$, and $c_\ell \in \mathcal{D}$ be given. Given $n \in \mathbb{N}_{\ell,k}(5^+;c_1)$, one can find digits $a_1,\dots,a_{18},b_1,\dots,b_{18} \in \mathcal{D}\backslash\{0\}$ and $c_2 \in \mathcal{D}$ such that the number*

$$n - \sum_{j=1}^{18} q_{\ell-1,k}(a_j,b_j)$$

lies in the set $\mathbb{N}_{\ell-1,k+1}(5^+;c_2)$.

Proof. Fix $n \in \mathbb{N}_{\ell,k}(5^+;c_1)$, and let $\{\delta_j\}_{j=0}^{\ell-1}$ be defined as in (1.1) (with $L := \ell$). Let m be the three-digit integer formed by the first three digits of n; that is,

$$m := 100\delta_{\ell-1} + 10\delta_{\ell-2} + \delta_{\ell-3}.$$

Clearly, m is an integer in the range $500 \leqslant m \leqslant 999$, and we have

$$n = \sum_{j=k}^{\ell-1} 10^j \delta_j = 10^{\ell-3} m + \sum_{j=k}^{\ell-4} 10^j \delta_j. \tag{2.4}$$

Let us denote

$$\mathcal{S} := \{19, 29, 39, 49, 59\}.$$

In view of the fact that

$$9\mathcal{S} := \underbrace{\mathcal{S} + \cdots + \mathcal{S}}_{\text{nine copies}} = \{171, 181, 191, \dots, 531\},$$

it is possible to find an element $h \in 9\mathcal{S}$ for which $m - 80 < 2h \leqslant m - 60$. With h fixed, let $s_1,\dots,s_9$ be elements of $\mathcal{S}$ such that

$$s_1 + \cdots + s_9 = h.$$

Finally, let $\varepsilon_1,\dots,\varepsilon_9$ be natural numbers, each equal to zero or two: $\varepsilon_j \in \{0,2\}$ for $j = 1,\dots,9$. A specific choice of these numbers is given below.

Previous proofs were complicated (2)

Excerpt from Cilleruelo et al. (2017)

II.2 $\boldsymbol{c_m = 0}$. We distinguish the following cases:

II.2.i) $\boldsymbol{y_m \neq 0}$.

δ_m	δ_{m-1}
0	0
$*$	y_m
$*$	$*$

$\longrightarrow$

δ_m	δ_{m-1}
1	1
$*$	$y_m - 1$
$*$	$*$

II.2.ii) $\boldsymbol{y_m = 0}$.

II.2.ii.a) $\boldsymbol{y_{m-1} \neq 0}$.

δ_m	δ_{m-1}	δ_{m-2}
0	0	$*$
y_{m-1}	0	y_{m-1}
$*$	z_{m-1}	z_{m-1}

$\longrightarrow$

δ_m	δ_{m-1}	δ_{m-2}
1	1	$*$
$y_{m-1} - 1$	$g - 2$	$y_{m-1} - 1$
$*$	$z_{m-1} + 1$	$z_{m-1} + 1$

The above step is justified for $z_{m-1} \neq g - 1$. But if $z_{m-1} = g - 1$, then $c_{m-1} \geq (y_{m-1} + z_{m-1})/g \geq 1$, so $c_m = (z_{m-1} + c_{m-1})/g = (g - 1 + 1)/g = 1$, a contradiction.

II.2.ii.b) $\boldsymbol{y_{m-1} = 0,\ z_{m-1} \neq 0}$.

δ_m	δ_{m-1}	δ_{m-2}
0	0	$*$
0	0	0
$*$	z_{m-1}	z_{m-1}

$\longrightarrow$

δ_m	δ_{m-1}	δ_{m-2}
0	0	$*$
1	1	1
$*$	$z_{m-1} - 1$	$z_{m-1} - 1$

II.2.ii.c) $\boldsymbol{y_{m-1} = 0,\ z_{m-1} = 0}$.

If also $c_{m-1} = 0$, then $\delta_{m-1} = 0$, which is not allowed. Thus, $c_{m-1} = 1$. This means that $x_{m-1} \in \{g - 1, g - 2\}$. Since $x_i \in \{0, 1, 2\}$ for $i \geq 3$, it follows that $m = 3$ and we are in one of the cases A.5) or A.6). Further, $\delta_2 = 1$. In this case we change the above configuration to:

Previous proofs were complicated (3)

- Proofs of Banks and Cilleruelo et al. were long and case-based
- Difficult to establish
- Difficult to understand
- Difficult to check, too: the original Cilleruelo et al. proof had some minor flaws that were only noticed when the proof was implemented as a `Python` program
- Idea: could we automate such proofs?

Regular approximation can't work for the palindromes

A theorem of Horváth, Karhumäki, and Kleijn (1987) shows that any regular underapproximation of the palindromes is *slender*: the number of words of length n is at most a constant.

So we will never have a dense enough underapproximation to get results this way.

Is there some other method that could work?

The main idea of our proof

- Construct a finite-state machine that takes natural numbers as input, expressed in the desired base
- Allow the machine to nondeterministically "guess" a representation of the input as a sum of palindromes
- The machine accepts an input if it verifies its guess
- Then use a decision procedure to establish properties about the language of representations accepted by this machine (e.g., universality)

Picking a machine model for palindromes

What machine model?

- it should be possible to check if the guessed summands are palindromes
  - can be done with a pushdown automaton (PDA)
- it should be possible to add the summands and compare to the input
  - can be done with a finite automaton (DFA or NFA)

However

- Can't add summands with these machine models unless they are guessed in parallel
- Can't check if summands are palindromes if they are wildly different in length & presented in parallel
- Universality is not decidable for PDA's

What to do?

Visibly pushdown automata (VPA)

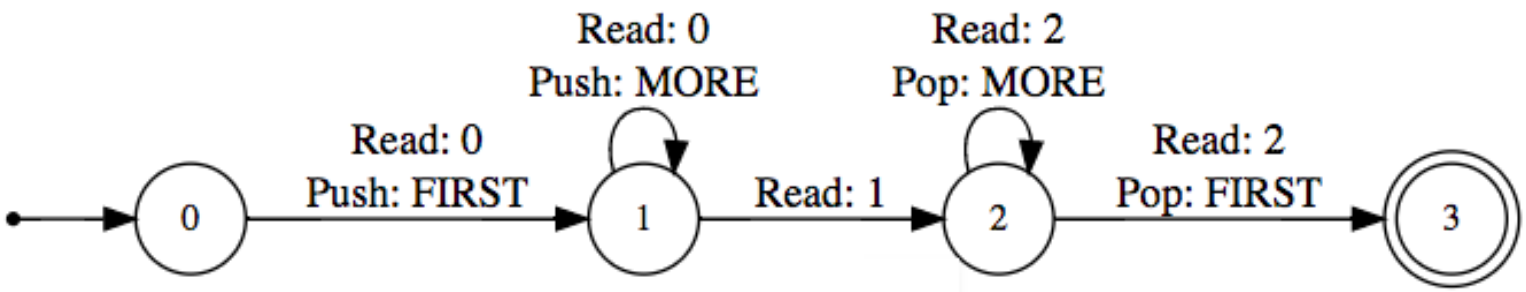
- Use *visibly-pushdown automata*!
- Popularized by Alur and Madhusudan in 2004, though similar ideas have been around for longer
- VPA's receive an input string, and read the string one letter at a time
- They have a (finite) set of states and a stack
- Upon reading a letter of the input string, the VPA can transition to a new state, and might modify the stack
- The states of the VPA are either *accepting* or *non-accepting*
- If the VPA can end up in an accepting state after it is done reading the input, then the VPA "accepts" the input, else it "rejects" it

Using the VPA's stack

- The VPA can only take very specific stack actions
- The input alphabet, Σ, is partitioned into three disjoint sets
  - Σ_c, the push alphabet
  - Σ_l, the local alphabet
  - Σ_r, the pop alphabet
- If the letter of the input string we read is from the push alphabet, the VPA pushes *exactly* one symbol onto its stack
- If the letter of the input string we read is from the pop alphabet, the VPA pops *exactly* one symbol off its stack
- If the letter of the input string we read is from the local alphabet, the VPA does not consult its stack at all

Example VPA

A VPA for the language $\{0^n 1 2^n \ : \ n \geq 1\}$:



The push alphabet is $\{0\}$, the local alphabet is $\{1\}$, and the pop alphabet is $\{2\}$.

Determinization and Decidability

- A nondeterministic VPA can have several matching transition rules for a single input letter
- Nondeterministic VPA's are as powerful as deterministic VPA's
- VPL's are closed under union, intersection and complement. There are algorithms for all these operations.
- Testing emptiness, universality and language inclusion are decidable problems for VPA's
- But a nondeterministic VPA with n states can have as many as $2^{\Theta(n^2)}$ states when determinized!

Proof strategy

- We build a VPA that "guesses" inputs of roughly the same size, in parallel
- It checks to see that they are palindromes
- And it adds them together and verifies that the sum equals the input.
- There are some complications due to the VPA restrictions.

More details of the proof strategy

- To prove our result, we built 2 VPA's A and B:
  - A accepts all n-bit odd integers, $n \geq 8$, that are the sum of three binary palindromes of length either
    - n, $n-2$, $n-3$, or
    - $n-1$, $n-2$, $n-3$.
  - B accepts all valid representations of odd integers of length $n \geq 8$
- We then prove that all inputs accepted by B are accepted by A
- We used the ULTIMATE Automata Library
- Once A and B are built, we simply have to issue the command

$$\texttt{assert}(\texttt{IsIncluded}(\texttt{B}, \texttt{A}))$$

in ULTIMATE.

Bases 3 and 4

- Unfortunately, the VPA's for bases 3 and 4 are too large to handle in this way.
- So we need a different approach.
- Instead, we use ordinary nondeterministic finite automata (NFA).
- But they cannot recognize palindromes...
- Instead, we change the input representation so that numbers are represented in a "folded" way, where each digit at the beginning of its representation is paired with its corresponding digit at the end.
- With this we can prove...

Other results

Theorem. (Rajasekaran, JOS, Smith) Every natural number is the sum of at most three base-3 palindromes.

Theorem. (Rajasekaran, JOS, Smith) Every natural number is the sum of at most three base-4 palindromes.

This completes the classification for base-b palindromes for all $b \geq 2$.

An analogue of Lagrange's theorem

Using NFA's we can also establish an analogue of Lagrange's four-square theorem.

- A *square* is any string that is some shorter string repeated twice
- Examples are `hotshots` (English), `nennen` (German), and 100100.
- We call an integer a *base-b square* if its base-b representation is a square
- Examples are $36 = [100100]_2$ and $3 = [11]_2$.
- The binary squares form sequence A020330 in the OEIS

$$3, 10, 15, 36, 45, 54, 63, 136, 153, 170, 187, 204, 221, \ldots$$

Lagrange's theorem strategy

Lemma.

(a) Every length-n integer, n odd, $n \geq 13$, is the sum of binary squares as follows: either

- one of length $n-1$ and one of length $n-3$, or
- two of length $n-1$ and one of length $n-3$, or
- one of length $n-1$ and two of length $n-3$, or
- one each of lengths $n-1$, $n-3$, and $n-5$, or
- two of length $n-1$ and two of length $n-3$, or
- two of length $n-1$, one of length $n-3$, and one of length $n-5$.

(b) Every length-n integer, n even, $n \geq 18$, is the sum of binary squares as follows: either

- two of length $n-2$ and two of length $n-4$, or
- three of length $n-2$ and one of length $n-4$, or
- one each of lengths n, $n-4$, and $n-6$, or
- two of length $n-2$, one of length $n-4$, and one of length $n-6$.

Lagrange's theorem

Note that

- Using automata we cannot state the theorem we *want* to prove
- This is due to the fact that we can't add squares of wildly differing lengths using the representation we chose
- But we *can* state the *stronger* result of the lemma on the previous slide
- So we are combining a decision procedure together with a heuristic search for an appropriate lemma to prove.

Results

Theorem. (Madhusudan, Nowotka, Rajasekaran, JOS) Every natural number $N > 686$ is the sum of at most 4 binary squares.

For example:

$$\begin{aligned} 10011938 &= 9291996 + 673425 + 46517 \\ &= [100011011100100011011100]_2 + [10100100011010010001]_2 \\ &\quad + [1011010110110101]_2 \end{aligned}$$

Here the 686 is optimal.

The list of all exceptions is

$$1, 2, 4, 5, 7, 8, 11, 14, 17, 22, 27, 29, 32, 34, 37, 41, 44, 47,$$
$$53, 62, 95, 104, 107, 113, 116, 122, 125, 131, 134, 140, 143,$$
$$148, 155, 158, 160, 167, 407, 424, 441, 458, 475, 492, 509,$$
$$526, 552, 560, 569, 587, 599, 608, 613, 620, 638, 653, 671, 686.$$

Another result

Theorem. Every natural number is the sum of at most two binary squares and at most two powers of 2.

Generalizing: Waring's theorem for binary k'th powers

Recall Waring's theorem: *for every $k \geq 1$ there exists a constant $g(k)$ such that every natural number is the sum of $g(k)$ k'th powers of natural numbers.*

Could the same result hold for the binary k'th powers?

Two issues:

- 1 is not a binary k'th power, so it has to be "every sufficiently large natural number" and not "every natural number".
- The gcd g of the binary k'th powers need not be 1, so it actually has to be "every sufficiently large multiple of g".

gcd of the binary k'th powers

Theorem. The gcd of the binary k'th powers is $\gcd(k, 2^k - 1)$.

Example:
The binary 6'th powers are

$$63, 2730, 4095, 149796, 187245, 224694, 262143, 8947848, 10066329, \ldots$$

with gcd equal to $\gcd(6, 63) = 3$.

Waring's theorem for binary k'th powers

Theorem. Every sufficiently large multiple of $\gcd(k, 2^k - 1)$ is the sum of a constant number (depending on k) of binary k'th powers.

Obtained with Daniel Kane and Carlo Sanna.

Outline of the proof

Given a number N we wish to represent as a sum of binary k'th powers:

- choose a suitable power of 2, say 2^n, and express N as a polynomial in $x = 2^n$. We want $x^k \approx N$.
- use linear algebra to change the basis and instead express N as a linear combination of $c_k(n), c_k(n+1), \ldots, c_k(n+k-1)$ where

$$c_k(n) = \frac{2^{kn}-1}{2^n-1} = 1 + 2^n + 2^{2n} + \cdots + 2^{(k-1)n}.$$

- Such a linear combination would seem to provide an expression for N in terms of binary k'th powers, but there are three problems to overcome:
  (a) the coefficients of $c_k(i)$, $n \le i < n+k$, could be much too large;
  (b) the coefficients could be too small or negative;
  (c) the coefficients might not be integers.

All of these problems can be handled with some work...

Could additive number theorists be replaced by a decision procedure?

- Unfortunately, probably not: for example, every context-free subset of the prime numbers is finite...
- ... so we will never be able to prove Goldbach's conjecture using these naive methods.
- Similarly, every regular subset of the powers expressed in base k is slender ...
- ... so Waring's theorem is also probably out of reach.

Open Problems

- Say something about the *number* of representations as the sum of two, three, or four palindromes.
- Are there arbitrarily large even numbers that cannot be written as the *difference* of two binary palindromes? The sequence of unrepresentable numbers starts

  $$1844, 1892, 2512, 3700, 4702, 5476, 5534, 7364, \ldots$$

Moral of the story

- Formal languages and automata are a source of new problems for additive number theory
- And they offer new techniques as well...
- They offer the prospect of proving nontrivial theorems of interest using *decision procedures* and *brute-force computation*
- They can replace long case-based proofs that are prone to error.

For further reading

- A. Rajasekaran, J. Shallit, and T. Smith, Sums of Palindromes: an Approach via Automata, in 36th Symposium on Theoretical Aspects of Computer Science (STACS 2018), Article No. 54, pp. 54:154:12, 2018.
- D. M. Kane, C. Sanna, and J. Shallit, Waring's theorem for binary powers, arxiv preprint, January 13 2018. Available at https://arxiv.org/abs/1801.04483. Submitted.
- P. Madhusudan, D. Nowotka, A. Rajasekaran and J. Shallit. Lagrange's Theorem for Binary Squares, to appear, 44rd MFCS, 2018. https://arxiv.org/abs/1710.04247.
- J. Bell, T. F. Lidbetter, and J. Shallit, Additive Number Theory via Approximation by Regular Languages, to appear, DLT 2018, Tokyo. https://arxiv.org/abs/1804.11175.
- J. P. Bell, K. E. Hare, and J. Shallit, When is an automatic set an additive basis?", to appear, *Proc. AMS*, 2018. Available at https://arxiv.org/abs/1710.08353.