Applications of Model Theory to Families of Integer Sequences

Arseniy (Senia) Sheydvasser

May 29, 2019

Acknowledgements:

- Joint work with Joshua Hinman, Borys Kuca, and Alexander Schlesinger. (*The Unreasonable Rigidity of Ulam Sequences* and *Rigidity of Ulam Sets and Sequences*.)

Acknowledgements:

- Joint work with Joshua Hinman, Borys Kuca, and Alexander Schlesinger. (*The Unreasonable Rigidity of Ulam Sequences* and *Rigidity of Ulam Sets and Sequences*.)
- Special thanks to the organizers of SUMRY 2017, to Stefan Steinerberger for introducing me to the problem, and to Nathan Fox and Kevin O'Bryant for valuable insight and examples.

General Setting:

- Let $S_1, S_2, S_3, \ldots$ be a family of integer sequences.

General Setting:

- Let $S_1, S_2, S_3, \ldots$ be a family of integer sequences.
- Suppose there is an algorithm $\mathcal{A}$ that takes two inputs n and k, such that
  1. $\mathcal{A}(n, k) = \text{T}$ if $k \in S_n$.
  2. $\mathcal{A}(n, k) = \text{F}$ if $k \notin S_n$.

General Setting:

- Let $S_1, S_2, S_3, \ldots$ be a family of integer sequences.
- Suppose there is an algorithm $\mathcal{A}$ that takes two inputs n and k, such that
  1. $\mathcal{A}(n, k) = \mathrm{T}$ if $k \in S_n$.
  2. $\mathcal{A}(n, k) = \mathrm{F}$ if $k \notin S_n$.

Question

What if instead we have an algorithm $\mathcal{A}$ so that it can accept as inputs non-standard integers n and k; what information does this give us about the family S_n?

Specifics:

- Obviously, if we have an algorithm for standard inputs, we can always use an ultra-filter to get a semi-algorithm $\mathcal{A}$ that runs over the non-standard integers.

Specifics:

- Obviously, if we have an algorithm for standard inputs, we can always use an ultra-filter to get a semi-algorithm $\mathcal{A}$ that runs over the non-standard integers.
- However, we want to avoid infinite loops.

```
def A(n,k):
    i = 0

    while(i < n):
        if i == k:
            print True

    print False
```

Bad

```
def A(n,k):
    print (k >= 0) and (k < n)
```

Good

Specifics:

- Obviously, if we have an algorithm for standard inputs, we can always use an ultra-filter to get a semi-algorithm $\mathcal{A}$ that runs over the non-standard integers.
- However, we want to avoid infinite loops.

```
def A(n,k):
    i = 0

    while(i < n):
        if i == k:
            print True

    print False
```

Bad

```
def A(n,k):
    print (k >= 0) and (k < n)
```

Good

- Thus, we shall insist that the algorithm halt in finite time.

First Example:

Definition (Hofstader, "Gödel, Escher, Bach")

The *Hofstader Q-sequence* is defined by
$Q(n) = Q(n - Q(n-1)) + Q(n - Q(n-2))$ and initial conditions
$Q(1) = 1$ and $Q(2) = 1$.

- The first few terms are
  $1, 1, 2, 3, 3, 4, 5, 5, 6, 6, 6, 8, 8, 8, 10, 9, 10, 11, 11 \ldots$
- Open question whether this sequence is infinite or not.

First Example:

Definition (Hofstader, "Gödel, Escher, Bach")

The *Hofstader Q-sequence* is defined by $Q(n) = Q(n - Q(n-1)) + Q(n - Q(n-2))$ and initial conditions $Q(1) = 1$ and $Q(2) = 1$.

- The first few terms are $1, 1, 2, 3, 3, 4, 5, 5, 6, 6, 6, 8, 8, 8, 10, 9, 10, 11, 11 \dots$
- Open question whether this sequence is infinite or not.

Definition (Fox 2018)

Define the sequence Q_r by the recurrence relation $Q_r(n) = Q_r(n - Q_r(n-1)) + Q_r(n - Q_r(n-2))$ and initial conditions $Q_r(1) = 1, Q_r(2) = 2, \dots Q_r(r) = r$.

Non-standard Q-Hofstader Sequences:

- If N is non-standard, what does Q_N look like?

Non-standard Q-Hofstader Sequences:

- If N is non-standard, what does Q_N look like?

$$Q_N = 1, 2, 3, \ldots N - 1, N, ???$$

Non-standard Q-Hofstader Sequences:

- If N is non-standard, what does Q_N look like?
- Recall that $Q_N(n) = Q_N(n - Q_N(n-1)) + Q_N(n - Q_N(n-2))$.

$$Q_N = 1, 2, 3, \ldots N-1, N, ???$$

Non-standard Q-Hofstader Sequences:

- If N is non-standard, what does Q_N look like?
- Recall that $Q_N(n) = Q_N(n - Q_N(n-1)) + Q_N(n - Q_N(n-2))$.

$$Q_N = 1, 2, 3, \ldots N-1, N, 3, ???$$

Non-standard Q-Hofstader Sequences:

- If N is non-standard, what does Q_N look like?
- Recall that $Q_N(n) = Q_N(n - Q_N(n-1)) + Q_N(n - Q_N(n-2))$.
- We can keep computing in this way until we hit the $(N+29)$-nd term.

$$\begin{aligned} Q_N = & 1, 2, 3, \ldots N-1, N, 3, N+1, N+2, 5, N+3, 6, 7, N+4, \\ & N+6, 10, 8, N+6, N+10, 12, N+7, 14, N+12, 11, \\ & N+11, N+15, 16, 13, 17, 15, N+14, 20, 20, 2N+8. \end{aligned}$$

Non-standard Q-Hofstader Sequences:

- If N is non-standard, what does Q_N look like?
- Recall that $Q_N(n) = Q_N(n - Q_N(n-1)) + Q_N(n - Q_N(n-2))$.
- We can keep computing in this way until we hit the $(N+29)$-nd term.

$$\begin{aligned} Q_N = & 1, 2, 3, \ldots N-1, N, 3, N+1, N+2, 5, N+3, 6, 7, N+4, \\ & N+6, 10, 8, N+6, N+10, 12, N+7, 14, N+12, 11, \\ & N+11, N+15, 16, 13, 17, 15, N+14, 20, 20, 2N+8. \end{aligned}$$

$$\begin{aligned} Q_N(N+29) &= Q_N(N+29-Q_N(N+28)) + Q_N(N+29-Q_N(N+27)) \\ &= Q_N(N+29-2N-8) + Q_N(N+29-20) \\ &= Q_N(21-N) + Q_N(N+9) \text{ ☹} \end{aligned}$$

Non-standard Algorithm:

- We can now write down what $\mathcal{A}(N, k)$ does if N is non-standard:
  1. If $k < 1$, return F.
  2. If $k \leq N$, return T.
  3. Otherwise, compute the 28 terms on the previous slide.
  4. If k is one of the terms, return T. Otherwise, return F.

Logical Consequences:

- What we have therefore proved over the hyper-naturals is that for all N sufficiently large, the sequence Q_N has $N + 28$ terms.

Logical Consequences:

- What we have therefore proved over the hyper-naturals is that for all N sufficiently large, the sequence Q_N has $N + 28$ terms.
- This can be phrased in a first-order way, and so we conclude that for all *naturals* N sufficiently large, the sequence Q_N has $N + 28$ terms!

Logical Consequences:

- What we have therefore proved over the hyper-naturals is that for all N sufficiently large, the sequence Q_N has $N+28$ terms.
- This can be phrased in a first-order way, and so we conclude that for all *naturals* N sufficiently large, the sequence Q_N has $N+28$ terms!
- The bad news is that this isn't exciting: there is a completely elementary proof of an even stronger result in *A New Approach to the Hofstadter Q-Recurrence*, Fox 2018.

Second Example:

Definition

A *Sidon set* is a set $S \subset \mathbb{N}$ such that $\forall w, x, y, z \in S$, $w + x = y + z$ if and only if $\{w, x\} = \{y, z\}$.

An (A, B)-*form Sidon set* is a set $S \subset \mathbb{N}$ such $\forall w, x, y, z \in S$, $Aw + Bx = Ay + Bz$ if and only if $\{w, x\} = \{y, z\}$.

The *greedy* (A, B)-*form Sidon sequence* $S_{A,B}$ is the sequence starting with 0, such that each subsequent term is the next smallest term such that the sequence is an (A, B)-form Sidon set.

Second Example:

Definition

A *Sidon set* is a set $S \subset \mathbb{N}$ such that $\forall w, x, y, z \in S$, $w + x = y + z$ if and only if $\{w, x\} = \{y, z\}$.

An (A, B)-*form Sidon set* is a set $S \subset \mathbb{N}$ such $\forall w, x, y, z \in S$, $Aw + Bx = Ay + Bz$ if and only if $\{w, x\} = \{y, z\}$.

The *greedy* (A, B)-*form Sidon sequence* $S_{A,B}$ is the sequence starting with 0, such that each subsequent term is the next smallest term such that the sequence is an (A, B)-form Sidon set.

$$S_{1,1} = 0, 1, 2, 4, 8, 13, 21, 31, 45, 66, 81, 97 \ldots$$
$$S_{1,2} = 0, 1, 4, 5, 16, 17, 20, 21, 64, 65, 68, 69 \ldots$$
$$S_{1,3} = 0, 1, 2, 9, 10, 11, 18, 19, 20, 81, 82, 83 \ldots$$

Non-standard Extension:

- We shall consider $S_{1,N}$, where N is again a non-standard natural. I claim we can again extend to non-standard inputs.

Non-standard Extension:

- We shall consider $S_{1,N}$, where N is again a non-standard natural. I claim we can again extend to non-standard inputs.
- Because the extension to hyper-naturals preserves first-order statements, each term t in $S_{1,N}$ is the smallest such that for all $w, x, y, z \in S_{1,N} \cap [1, t]$, $w + xN = y + zN$ if and only if $\{w, x\} = \{y, z\}$.

Non-standard Extension:

- We shall consider $S_{1,N}$, where N is again a non-standard natural. I claim we can again extend to non-standard inputs.
- Because the extension to hyper-naturals preserves first-order statements, each term t in $S_{1,N}$ is the smallest such that for all $w, x, y, z \in S_{1,N} \cap [1, t]$, $w + xN = y + zN$ if and only if $\{w, x\} = \{y, z\}$.
- Thus, at each step, we need to check if $t = x + (y - z)N$ or $t = x + (y - z)/N$ for $x, y, z \in S_{1,N} \cap [1, t - 1]$. This can be done recursively.

Non-standard Extension:

- We shall consider $S_{1,N}$, where N is again a non-standard natural. I claim we can again extend to non-standard inputs.
- Because the extension to hyper-naturals preserves first-order statements, each term t in $S_{1,N}$ is the smallest such that for all $w, x, y, z \in S_{1,N} \cap [1, t]$, $w + xN = y + zN$ if and only if $\{w, x\} = \{y, z\}$.
- Thus, at each step, we need to check if $t = x + (y - z)N$ or $t = x + (y - z)/N$ for $x, y, z \in S_{1,N} \cap [1, t - 1]$. This can be done recursively.

$$S_{1,N} \cap [1, N] = 0, 1, 2, \ldots N - 1$$

Non-standard Extension:

- We shall consider $S_{1,N}$, where N is again a non-standard natural. I claim we can again extend to non-standard inputs.
- Because the extension to hyper-naturals preserves first-order statements, each term t in $S_{1,N}$ is the smallest such that for all $w, x, y, z \in S_{1,N} \cap [1, t]$, $w + xN = y + zN$ if and only if $\{w, x\} = \{y, z\}$.
- Thus, at each step, we need to check if $t = x + (y - z)N$ or $t = x + (y - z)/N$ for $x, y, z \in S_{1,N} \cap [1, t - 1]$. This can be done recursively.

$$S_{1,N} \cap [1, N^2 - 1] = 0, 1, 2, \ldots N - 1$$

Non-standard Extension:

- We shall consider $S_{1,N}$, where N is again a non-standard natural. I claim we can again extend to non-standard inputs.
- Because the extension to hyper-naturals preserves first-order statements, each term t in $S_{1,N}$ is the smallest such that for all $w, x, y, z \in S_{1,N} \cap [1, t]$, $w + xN = y + zN$ if and only if $\{w, x\} = \{y, z\}$.
- Thus, at each step, we need to check if $t = x + (y - z)N$ or $t = x + (y - z)/N$ for $x, y, z \in S_{1,N} \cap [1, t - 1]$. This can be done recursively.

$$S_{1,N} \cap [1, N^2 + N] = 0, 1, 2, \ldots N - 1, \\ N^2, N^2 + 1, \ldots N^2 + N - 1$$

Non-standard Extension:

- We shall consider $S_{1,N}$, where N is again a non-standard natural. I claim we can again extend to non-standard inputs.
- Because the extension to hyper-naturals preserves first-order statements, each term t in $S_{1,N}$ is the smallest such that for all $w, x, y, z \in S_{1,N} \cap [1, t]$, $w + xN = y + zN$ if and only if $\{w, x\} = \{y, z\}$.
- Thus, at each step, we need to check if $t = x + (y - z)N$ or $t = x + (y - z)/N$ for $x, y, z \in S_{1,N} \cap [1, t - 1]$. This can be done recursively.

$$
\begin{aligned}
S_{1,N} = [1, 2N^2 + N] = & 0, 1, 2, \ldots N - 1, \\
& N^2, N^2 + 1, \ldots N^2 + N - 1, \\
& 2N^2, 2N^2 + 1, 2N^2 + 2, \ldots 2N^2 + N - 1
\end{aligned}
$$

Non-standard Algorithm:

- In this recursive fashion, we can prove that

$$x \in S_{1,N} \Leftrightarrow \exists T \in {}^*\mathbb{N} \text{ s.t. } x = \sum_{l=0}^{T} a_l N^{2l},\ 0 \leq a_l < N.$$

Non-standard Algorithm:

- In this recursive fashion, we can prove that

$$x \in S_{1,N} \Leftrightarrow \exists T \in {}^*\mathbb{N} \text{ s.t. } x = \sum_{l=0}^{T} a_l N^{2l},\ 0 \leq a_l < N.$$

- Thus, we again can form an algorithm expressing $S_{1,N}$ even if N is non-standard, and using the transfer principle, we can conclude that for all sufficiently large integers N,

$$x \in S_{1,N} \Leftrightarrow \exists T \in \mathbb{N} \text{ s.t. } x = \sum_{l=0}^{T} a_l N^{2l},\ 0 \leq a_l < N.$$

Non-standard Algorithm:

- In this recursive fashion, we can prove that

$$x \in S_{1,N} \Leftrightarrow \exists T \in {}^*\mathbb{N} \text{ s.t. } x = \sum_{l=0}^{T} a_l N^{2l},\ 0 \leq a_l < N.$$

- Thus, we again can form an algorithm expressing $S_{1,N}$ even if N is non-standard, and using the transfer principle, we can conclude that for all sufficiently large integers N,

$$x \in S_{1,N} \Leftrightarrow \exists T \in \mathbb{N} \text{ s.t. } x = \sum_{l=0}^{T} a_l N^{2l},\ 0 \leq a_l < N.$$

- Unfortunately, it is a theorem in the folklore (due to Kevin O'Bryant) that this is true for all $N \geq 2$, and this is again proved by elementary means.

Third Example:

Definition

An *Ulam sequence* is an increasing sequence $U(a, b)$ of integers defined by

- $u_0 = a$, $u_1 = b$, and
- u_k (for $k > 1$) is the smallest integer that can be written as the sum of two distinct smaller terms u_m, u_n in exactly one way.

Third Example:

Definition

An *Ulam sequence* is an increasing sequence $U(a, b)$ of integers defined by

- $u_0 = a$, $u_1 = b$, and
- u_k (for $k > 1$) is the smallest integer that can be written as the sum of two distinct smaller terms u_m, u_n in exactly one way.

Examples:

- $U(1, 2) : 1, 2, 3, 4, 6, 8, 11, 13, 16, 18 \ldots$
- $U(1, 3) : 1, 3, 4, 5, 6, 8, 10, 12, 17, 21 \ldots$
- $U(2, 3) : 2, 3, 5, 7, 8, 9, 13, 14, 18, 19 \ldots$

Third Example:

Definition

An *Ulam sequence* is an increasing sequence $U(a, b)$ of integers defined by

- $u_0 = a$, $u_1 = b$, and
- u_k (for $k > 1$) is the smallest integer that can be written as the sum of two distinct smaller terms u_m, u_n in exactly one way.

Examples:

- $U(1, 2) : 1, 2, 3, 4, 6, 8, 11, 13, 16, 18 \ldots$
- $U(1, 3) : 1, 3, 4, 5, 6, 8, 10, 12, 17, 21 \ldots$
- $U(2, 3) : 2, 3, 5, 7, 8, 9, 13, 14, 18, 19 \ldots$

- Introduced in 1964 by Ulam, who wanted to understand their growth properties.

Third Example:

Definition

An *Ulam sequence* is an increasing sequence $U(a, b)$ of integers defined by

- $u_0 = a$, $u_1 = b$, and
- u_k (for $k > 1$) is the smallest integer that can be written as the sum of two distinct smaller terms u_m, u_n in exactly one way.

Examples:

- $U(1, 2) : 1, 2, 3, 4, 6, 8, 11, 13, 16, 18 \ldots$
- $U(1, 3) : 1, 3, 4, 5, 6, 8, 10, 12, 17, 21 \ldots$
- $U(2, 3) : 2, 3, 5, 7, 8, 9, 13, 14, 18, 19 \ldots$

- Introduced in 1964 by Ulam, who wanted to understand their growth properties.
- Despite their apparent simplicity, almost nothing is known about Ulam sequences.

Rigidity of the $U(1, n)$ Sequences:

Rigidity

The family of sequences $U(1, n)$ seems to have a rather surprising pattern.

Rigidity of the $U(1, n)$ Sequences:

Rigidity

The family of sequences $U(1, n)$ seems to have a rather surprising pattern.

$U(1,2)$:	1,	2,	3,	4,	6,	8,	11,	13,	16,	18,	26,	28 ...
$U(1,3)$:	1,	3,	4,	5,	6,	8,	10,	12,	17,	21,	23,	28 ...
$U(1,4)$:	1,	4,	5,	6,	7,	8,	10,	16,	18,	19,	21,	31 ...
$U(1,5)$:	1,	5,	6,	7,	8,	9,	10,	12,	20,	22,	23,	24 ...
$U(1,6)$:	1,	6,	7,	8,	9,	10,	11,	12,	14,	24,	26,	27 ...

Rigidity of the $U(1, n)$ Sequences:

Rigidity

The family of sequences $U(1, n)$ seems to have a rather surprising pattern.

$U(1,2)$:	$\boxed{1}$,	$\boxed{2,3,4}$,	$\boxed{6}$,	8,	11,		13 ...
$U(1,3)$:	$\boxed{1}$,	$\boxed{3,4,5,6}$,	$\boxed{8}$,	10,	12,		17 ...
$U(1,4)$:	$\boxed{1}$,	$\boxed{4,5,6,7,8}$,	$\boxed{10}$,	$\boxed{16}$,	$\boxed{18,19}$,		$\boxed{21}$ ...
$U(1,5)$:	$\boxed{1}$,	$\boxed{5,6,7,8,9,10}$,	$\boxed{12}$,	$\boxed{20}$,	$\boxed{22,23,24}$,		$\boxed{26}$ ...
$U(1,6)$:	$\boxed{1}$,	$\boxed{6,7,8,9,10,11,12}$,	$\boxed{14}$,	$\boxed{24}$,	$\boxed{26,27,28,29}$,		$\boxed{31}$ ...

Rigidity of the $U(1,n)$ Sequences:

Rigidity

The family of sequences $U(1,n)$ seems to have a rather surprising pattern.

$$
\begin{array}{l|llllll}
U(1,2): & \boxed{1}, & \boxed{2,\dots 4}, & \boxed{6}, & 8, & 11, & 13\dots \\
U(1,3): & \boxed{1}, & \boxed{3,\dots 6}, & \boxed{8}, & 10, & 12, & 17\dots \\
U(1,4): & \boxed{1}, & \boxed{4,\dots 8}, & \boxed{10}, & \boxed{16}, & \boxed{18,19}, & \boxed{21}\dots \\
U(1,5): & \boxed{1}, & \boxed{5,\dots 10}, & \boxed{12}, & \boxed{20}, & \boxed{22,\dots 24}, & \boxed{26}\dots \\
U(1,6): & \boxed{1}, & \boxed{6,\dots 12}, & \boxed{14}, & \boxed{24}, & \boxed{26,\dots 29}, & \boxed{31}\dots \\
U(1,n): & \boxed{1}, & \boxed{n,\dots 2n}, & \boxed{2n+2}, & \boxed{4n}, & \boxed{4n+2,\dots 5n-1}, & \boxed{5n+1}\dots
\end{array}
$$

The Rigidity Conjecture:

Conjecture

There exists a positive integer N and integer coefficients m_i, p_i, k_i, r_i such that for all $n \geq N$,

$$U(1, n) = \bigsqcup_{i \in \mathbb{N}} [m_i n + p_i, k_i n + r_i].$$

In fact, we conjecture that we can take $N = 4$.

The Rigidity Conjecture:

Conjecture

There exists a positive integer N and integer coefficients m_i, p_i, k_i, r_i such that for all $n \geq N$,

$$U(1, n) = \bigsqcup_{i \in \mathbb{N}} [m_i n + p_i, k_i n + r_i].$$

In fact, we conjecture that we can take $N = 4$.

- This is very well supported numerically (more on that later).

The Rigidity Conjecture:

Conjecture

There exists a positive integer N and integer coefficients m_i, p_i, k_i, r_i such that for all $n \geq N$,

$$U(1, n) = \bigsqcup_{i \in \mathbb{N}} [m_i n + p_i, k_i n + r_i].$$

In fact, we conjecture that we can take $N = 4$.

- This is very well supported numerically (more on that later).
- Note that the coefficients don't depend on n, and can be calculated using any two consecutive Ulam sequences.

The Rigidity Conjecture:

Conjecture

There exists a positive integer N and integer coefficients m_i, p_i, k_i, r_i such that for all $n \geq N$,

$$U(1, n) = \bigsqcup_{i \in \mathbb{N}} [m_i n + p_i, k_i n + r_i].$$

In fact, we conjecture that we can take $N = 4$.

- This is very well supported numerically (more on that later).
- Note that the coefficients don't depend on n, and can be calculated using any two consecutive Ulam sequences.
- Effectively, the conjecture says that once you have seen two (sufficiently large) Ulam sequences $U(1, n)$, you have seen them all.

Next Best Result:

Theorem (Weak Rigidity Theorem)

There exist integer coefficients m_i, p_i, k_i, r_i *such that for every* $C > 0$, *there exists a positive integer* N *such that for all* $n \geq N$,

$$U(1, n) \cap [1, Cn] = \bigsqcup_{i \in \mathbb{N}} [m_i n + p_i, k_i n + r_i] \cap [1, Cn].$$

Next Best Result:

Theorem (Weak Rigidity Theorem)

There exist integer coefficients m_i, p_i, k_i, r_i such that for every $C > 0$, there exists a positive integer N such that for all $n \geq N$,

$$U(1, n) \cap [1, Cn] = \bigsqcup_{i \in \mathbb{N}} [m_i n + p_i, k_i n + r_i] \cap [1, Cn].$$

- We shall prove this by making use of the machinery we have developed.

Passing to the Hyper-Naturals:

- Fix a standard natural $C > 0$.
- Consider the set $U(1, N) \cap [1, CN]$, where N is non-standard.

Passing to the Hyper-Naturals:

- Fix a standard natural $C > 0$.
- Consider the set $U(1, N) \cap [1, CN]$, where N is non-standard.
- It must go

$$1, N, N+1, N+2, \dots 2N-1, 2N, \cancel{2N+1}, 2N+2, 4N, \dots$$

Passing to the Hyper-Naturals:

- Fix a standard natural $C > 0$.
- Consider the set $U(1, N) \cap [1, CN]$, where N is non-standard.
- It must go

$$1, N, N+1, N+2, \ldots 2N-1, 2N, \cancel{2N+1}, 2N+2, 4N, \ldots$$

- To make this formal, argue by induction on C and i.
- We thus construct m_i, p_i, k_i, r_i such that

$$U(1, N) \cap [1, CN] = \bigsqcup_{i \in \mathbb{N}} [m_i N + p_i, k_i N + r_i] \cap [1, CN].$$

Passing to the Hyper-Naturals:

- Fix a standard natural $C > 0$.
- Consider the set $U(1,N) \cap [1, CN]$, where N is non-standard.
- It must go

$$1, N, N+1, N+2, \dots 2N-1, 2N, \cancel{2N+1}, 2N+2, 4N, \dots$$

- To make this formal, argue by induction on C and i.
- We thus construct m_i, p_i, k_i, r_i such that

$$U(1,N) \cap [1, CN] = \bigsqcup_{i \in \mathbb{N}} [m_i N + p_i, k_i N + r_i] \cap [1, CN].$$

- In fact, we produce an algorithm capable of constructing these coefficients up to C!

Consequences:

- We have therefore proved over the hyper-naturals that for all sufficiently large N,

$$U(1, N) \cap [1, CN] = \bigsqcup_{i \in \mathbb{N}} [m_i N + p_i, k_i N + r_i] \cap [1, CN].$$

Consequences:

- We have therefore proved over the hyper-naturals that for all sufficiently large N,

$$U(1, N) \cap [1, CN] = \bigsqcup_{i \in \mathbb{N}} [m_i N + p_i, k_i N + r_i] \cap [1, CN].$$

- It follows that the same is true over the naturals, proving the theorem.

Consequences:

- We have therefore proved over the hyper-naturals that for all sufficiently large N,

$$U(1, N) \cap [1, CN] = \bigsqcup_{i \in \mathbb{N}} [m_i N + p_i, k_i N + r_i] \cap [1, CN].$$

- It follows that the same is true over the naturals, proving the theorem.
- This is the first example of an algorithm where we needed to restrict the domain.
- Also the first example where the theorem is not known independently.

Consequences:

- We have therefore proved over the hyper-naturals that for all sufficiently large N,

$$U(1, N) \cap [1, CN] = \bigsqcup_{i \in \mathbb{N}} [m_i N + p_i, k_i N + r_i] \cap [1, CN].$$

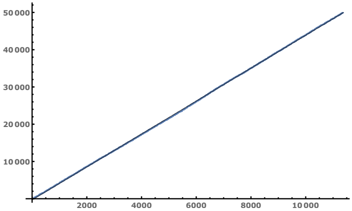
- It follows that the same is true over the naturals, proving the theorem.
- This is the first example of an algorithm where we needed to restrict the domain.
- Also the first example where the theorem is not known independently.
- The proof is vaguely non-constructive, but we can make the result completely constructive.

Growth Rate of Coefficients:

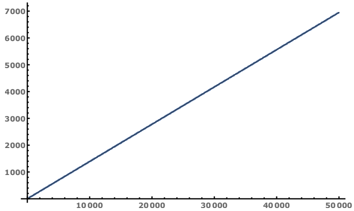
- What is the growth rate of the coefficients m_i, p_i, k_i, r_i?

Growth Rate of Coefficients:

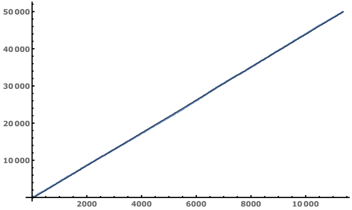
- What is the growth rate of the coefficients m_i, p_i, k_i, r_i?



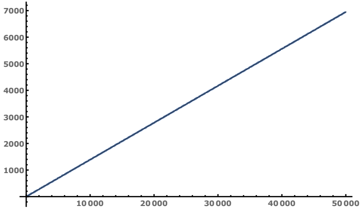
i vs. m_i



m_i vs. p_i



i vs. k_i



k_i vs. r_i

Growth Rate of Coefficients:

- In particular, for all i such that $k_i \leq 50000$, we find that

$$|p_i - 0.139m_i|, |r_i - 0.139k_i| < 2.5$$

Growth Rate of Coefficients:

- In particular, for all i such that $k_i \leq 50000$, we find that

$$|p_i - 0.139m_i|, |r_i - 0.139k_i| < 2.5$$

- This is useful, because we can use this statement about the growth rate to make the weak rigidity theorem effective.

Effective Estimates:

Theorem

Suppose that for some positive integer M,

$$U(1,N_0)\cap[1,k_M N_0+r_M+1]=\bigsqcup_{i=1}^{M}[m_i N_0+p_i,k_i N_0+r_i]$$

where for some $B,\epsilon>0$, $|p_i-m_iB|,|r_i-k_iB|<\epsilon$, and $N_0>4(1+\epsilon)-B$. Then for all $N>N_0$,

$$U(1,N)\cap[1,k_M N+r_M+1]=\bigsqcup_{i=1}^{M}[m_i N+p_i,k_i N+r_i].$$

Effective Estimates:

Theorem

Suppose that for some positive integer M,

$$U(1, N_0) \cap [1, k_M N_0 + r_M + 1] = \bigsqcup_{i=1}^{M} [m_i N_0 + p_i, k_i N_0 + r_i]$$

where for some $B, \epsilon > 0$, $|p_i - m_i B|, |r_i - k_i B| < \epsilon$, and $N_0 > 4(1+\epsilon) - B$. Then for all $N > N_0$,

$$U(1, N) \cap [1, k_M N + r_M + 1] = \bigsqcup_{i=1}^{M} [m_i N + p_i, k_i N + r_i] .$$

- The proof proceeds by induction over M and N.

Effective Estimates:

- This yields an algorithm for computing a bound N_0 that works in the statement of the weak rigidity theorem.

Effective Estimates:

- This yields an algorithm for computing a bound N_0 that works in the statement of the weak rigidity theorem.
  1. Calculate coefficients m_i, p_i, k_i, r_i.
  2. Do a linear regression to fit the best value of B to the computed coefficients. Calculate the corresponding maximum error ϵ.
  3. Compute $N_0' = \lceil 4(1+\epsilon) - B \rceil$.
  4. Use the coefficients m_i, p_i, k_i, r_i to predict the first CN terms of $U(1, N_0'), U(1, N_0' \pm 1) \ldots$.
  5. Halt once you find the smallest N_0 such that $U(1, n)$ matches the prediction for all $N_0 \leq n \leq \max\{N_0, N_0'\}$.

Effective Estimates:

- This yields an algorithm for computing a bound N_0 that works in the statement of the weak rigidity theorem.
  1. Calculate coefficients m_i, p_i, k_i, r_i.
  2. Do a linear regression to fit the best value of B to the computed coefficients. Calculate the corresponding maximum error ϵ.
  3. Compute $N_0' = \lceil 4(1+\epsilon) - B \rceil$.
  4. Use the coefficients m_i, p_i, k_i, r_i to predict the first CN terms of $U(1, N_0'), U(1, N_0' \pm 1) \dots$.
  5. Halt once you find the smallest N_0 such that $U(1, n)$ matches the prediction for all $N_0 \leq n \leq \max\{N_0, N_0'\}$.
- Using this, we prove that for all $n \geq 4$,

$$U(1, n) \cap [1, 50000n] = \bigsqcup_{i \in \mathbb{N}} [m_i n + p_i, k_i n + r_i] \cap [1, 50000n].$$

Next Steps:

- What other families of integer sequences can we find that fit this general paradigm? (Known examples are variations on the listed ones.)

Next Steps:

- What other families of integer sequences can we find that fit this general paradigm? (Known examples are variations on the listed ones.)
- Are there any general theorems that we can prove about integer sequences coming from an algorithm extendable to non-standard inputs?

Next Steps:

- What other families of integer sequences can we find that fit this general paradigm? (Known examples are variations on the listed ones.)
- Are there any general theorems that we can prove about integer sequences coming from an algorithm extendable to non-standard inputs?
- If we can prove some restrictions on the growth rate of the sequences, does this tells us something, like it does for the Ulam sequence?

Next Steps:

- What other families of integer sequences can we find that fit this general paradigm? (Known examples are variations on the listed ones.)
- Are there any general theorems that we can prove about integer sequences coming from an algorithm extendable to non-standard inputs?
- If we can prove some restrictions on the growth rate of the sequences, does this tells us something, like it does for the Ulam sequence?
- Does there exist any $\epsilon > 0$ such that there are integer coefficients m_i, p_i, k_i, r_i so that for any $C > 0$, there is an $N > 0$ such that for all $n \geq N$,

$$U(1, n) \cap [1, Cn^{1+\epsilon}] = \bigsqcup_{i \in \mathbb{N}} [m_i n + p_i, k_i n + r_i] \cap [1, Cn^{1+\epsilon}]?$$