How do you measure primality?

Christopher O'Neill

Texas A&M University

coneill@math.tamu.edu

Joint with Thomas Barron and Roberto Pelayo

September 26, 2014

ω-primality

Definition (ω-primality)

Fix a cancellative, commutative, atomic monoid M. For $x \in M$, $\omega(x)$ is the smallest positive integer m such that whenever $x \mid \prod_{i=1}^{r} u_i$ for $r > m$, there exists a subset $T \subset \{1, \ldots, r\}$ with $|T| \le m$ such that $x \mid \prod_{i \in T} u_i$.

ω-primality

Definition (ω-primality)

Fix a cancellative, commutative, atomic monoid M. For $x \in M$, $\omega(x)$ is the smallest positive integer m such that whenever $x \mid \prod_{i=1}^{r} u_i$ for $r > m$, there exists a subset $T \subset \{1, \ldots, r\}$ with $|T| \leq m$ such that $x \mid \prod_{i \in T} u_i$.

Fact

$\omega(x) = 1$ if and only if x is prime (i.e. $x \mid ab$ implies $x \mid a$ or $x \mid b$).

ω-primality

Definition (ω-primality)

Fix a cancellative, commutative, atomic monoid M. For $x \in M$, $\omega(x)$ is the smallest positive integer m such that whenever $x \mid \prod_{i=1}^{r} u_i$ for $r > m$, there exists a subset $T \subset \{1, \ldots, r\}$ with $|T| \leq m$ such that $x \mid \prod_{i \in T} u_i$.

Fact

$\omega(x) = 1$ if and only if x is prime (i.e. $x \mid ab$ implies $x \mid a$ or $x \mid b$).

Fact

M is factorial if and only if every irreducible element $u \in M$ is prime. Moreover, $\omega(p_1 \cdots p_r) = r$ for any primes $p_1, \ldots, p_r \in M$.

ω-primality

Definition (ω-primality)

Fix a cancellative, commutative, atomic monoid M. For $x \in M$, $\omega(x)$ is the smallest positive integer m such that whenever $x \mid \prod_{i=1}^{r} u_i$ for $r > m$, there exists a subset $T \subset \{1, \ldots, r\}$ with $|T| \leq m$ such that $x \mid \prod_{i \in T} u_i$.

ω-primality

Definition (ω-primality)

Fix a cancellative, commutative, atomic monoid M. For $x \in M$, $\omega(x)$ is the smallest positive integer m such that whenever $x \mid \prod_{i=1}^{r} u_i$ for $r > m$, there exists a subset $T \subset \{1, \ldots, r\}$ with $|T| \leq m$ such that $x \mid \prod_{i \in T} u_i$.

Definition

A *bullet* for $x \in M$ is a product $u_1 \cdots u_r$ of irreducible elements such that (i) x divides $u_1 \cdots u_r$, and (ii) x does not divide $u_1 \cdots u_r / u_i$ for each $i \leq r$. The set of bullets of x is denoted $\mathrm{bul}(x)$.

ω-primality

Definition (ω-primality)

Fix a cancellative, commutative, atomic monoid M. For $x \in M$, $\omega(x)$ is the smallest positive integer m such that whenever $x \mid \prod_{i=1}^{r} u_i$ for $r > m$, there exists a subset $T \subset \{1, \ldots, r\}$ with $|T| \leq m$ such that $x \mid \prod_{i \in T} u_i$.

Definition

A *bullet* for $x \in M$ is a product $u_1 \cdots u_r$ of irreducible elements such that (i) x divides $u_1 \cdots u_r$, and (ii) x does not divide $u_1 \cdots u_r / u_i$ for each $i \leq r$. The set of bullets of x is denoted $\text{bul}(x)$.

Proposition

$\omega_M(x) = \max\{r : u_1 \cdots u_r \in \text{bul}(x)\}$.

Numerical monoids

Definition

A *numerical monoid* S is an additive submonoid of $\mathbb{N}$ with $|\mathbb{N} \setminus S| < \infty$.

Numerical monoids

Definition

A *numerical monoid* S is an additive submonoid of $\mathbb{N}$ with $|\mathbb{N} \setminus S| < \infty$.

Fact

Any numerical monoid S has a unique minimal generating set $g_1, \ldots, g_k$.

Numerical monoids

Definition

A *numerical monoid* S is an additive submonoid of $\mathbb{N}$ with $|\mathbb{N} \setminus S| < \infty$.

Fact

Any numerical monoid S has a unique minimal generating set $g_1, \ldots, g_k$.

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

Numerical monoids

Definition

A *numerical monoid* S is an additive submonoid of $\mathbb{N}$ with $|\mathbb{N} \setminus S| < \infty$.

Fact

Any numerical monoid S has a unique minimal generating set $g_1, \ldots, g_k$.

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$. "McNugget Monoid"

Numerical monoids

Definition

A *numerical monoid* S is an additive submonoid of $\mathbb{N}$ with $|\mathbb{N} \setminus S| < \infty$.

Fact

Any numerical monoid S has a unique minimal generating set $g_1, \ldots, g_k$.

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$. "McNugget Monoid"

$n \in McN$	$\omega(n)$	bullet
6	3	$3 \cdot 20$
9	3	$3 \cdot 20$
12	3	$3 \cdot 20$
15	4	$4 \cdot 6$
18	3	$3 \cdot 6$

$n \in McN$	$\omega(n)$	bullet
20	10	$10 \cdot 6$
21	5	$5 \cdot 6$
24	4	$4 \cdot 6$
26	11	$11 \cdot 6$
27	6	$6 \cdot 6$

Algorithms to compute ω-primality

ω-primality in a numerical monoid S:

Algorithms to compute ω-primality

ω-primality in a numerical monoid S:

- Bullets in S: $b_1 g_1 + \cdots + b_k g_k \longleftrightarrow \vec{b} = (b_1, \ldots, b_k) \in \mathbb{N}^k$.

Algorithms to compute ω-primality

ω-primality in a numerical monoid S:

- Bullets in S: $b_1g_1 + \cdots + b_kg_k \longleftrightarrow \vec{b} = (b_1, \ldots, b_k) \in \mathbb{N}^k$.
- For each $i \leq k$, we have $c_i\vec{e}_i \in \text{bul}(n)$ for some $c_i > 0$.

Algorithms to compute ω-primality

ω-primality in a numerical monoid S:

- Bullets in S: $b_1g_1 + \cdots + b_kg_k \longleftrightarrow \vec{b} = (b_1, \ldots, b_k) \in \mathbb{N}^k$.
- For each $i \le k$, we have $c_i\vec{e}_i \in \text{bul}(n)$ for some $c_i > 0$.
- $\text{bul}(n) \subset \prod_{i=1}^{k}[0, c_i]$.

Algorithms to compute ω-primality

ω-primality in a numerical monoid S:

- Bullets in S: $b_1 g_1 + \cdots + b_k g_k \longleftrightarrow \vec{b} = (b_1, \ldots, b_k) \in \mathbb{N}^k$.
- For each $i \le k$, we have $c_i \vec{e}_i \in \mathrm{bul}(n)$ for some $c_i > 0$.
- $\mathrm{bul}(n) \subset \prod_{i=1}^k [0, c_i]$.

Algorithm

Search $\prod_{i=1}^k [0, c_i]$ for bullets, compute $\omega(n) = \max\{|\vec{b}| : \vec{b} \in \mathrm{bul}(n)\}$.

Algorithms to compute ω-primality

ω-primality in a numerical monoid S:

- Bullets in S: $b_1 g_1 + \cdots + b_k g_k \longleftrightarrow \vec{b} = (b_1, \ldots, b_k) \in \mathbb{N}^k$.
- For each $i \le k$, we have $c_i \vec{e}_i \in \text{bul}(n)$ for some $c_i > 0$.
- $\text{bul}(n) \subset \prod_{i=1}^k [0, c_i]$.

Algorithm

Search $\prod_{i=1}^k [0, c_i]$ for bullets, compute $\omega(n) = \max\{|\vec{b}| : \vec{b} \in \text{bul}(n)\}$.

Remark

Several improvements on this algorithm exist.

Quasilinearity for numerical monoids

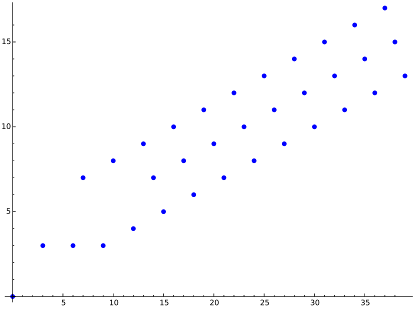
Theorem ((O.–Pelayo, 2013), (García-García et.al., 2013))

$\omega_S(n) = \frac{1}{g_1}n + a_0(n)$ *for* $n \gg 0$, *where* $a_0(n)$ *periodic with period* g_1.

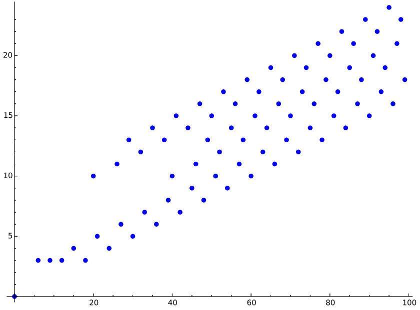
Quasilinearity for numerical monoids

Theorem ((O.–Pelayo, 2013), (García-García et.al., 2013))

$\omega_S(n) = \frac{1}{g_1} n + a_0(n)$ for $n \gg 0$, where $a_0(n)$ periodic with period g_1.



$S = \langle 3, 7 \rangle$



$McN = \langle 6, 9, 20 \rangle$

Quasilinearity for numerical monoids

Dissonance point: minimum N_0 such that $\omega(n)$ is quasilinear for $n > N_0$.

Quasilinearity for numerical monoids

Dissonance point: minimum N_0 such that $\omega(n)$ is quasilinear for $n > N_0$.

Question (O.-Pelayo, 2013)

The upper bound for dissonance point is large. Can we do better?

Quasilinearity for numerical monoids

Dissonance point: minimum N_0 such that $\omega(n)$ is quasilinear for $n > N_0$.

Question (O.-Pelayo, 2013)

The upper bound for dissonance point is large. Can we do better?

Roadblock

Existing algorithms are slow for large n.

Quasilinearity for numerical monoids

Dissonance point: minimum N_0 such that $\omega(n)$ is quasilinear for $n > N_0$.

Question (O.-Pelayo, 2013)

The upper bound for dissonance point is large. Can we do better?

Roadblock

Existing algorithms are slow for large n.

Question (O.–Pelayo, 2014)

Can we dynamically (inductively) compute several ω-values at once?

Quasilinearity for numerical monoids

Dissonance point: minimum N_0 such that $\omega(n)$ is quasilinear for $n > N_0$.

Question (O.-Pelayo, 2013)

The upper bound for dissonance point is large. Can we do better?

Roadblock

Existing algorithms are slow for large n.

Question (O.–Pelayo, 2014)

Can we dynamically (inductively) compute several ω-values at once?

Answer (Barron-O.-Pelayo, 2014)

Yes!

Toward a dynamic algorithm. . . the inductive step

For $n \in S$, let $\mathsf{Z}(n) = \{\vec{a} \in \mathbb{N}^k : \sum_{i=1}^k a_i g_i = n\}$.

Toward a dynamic algorithm. . . the inductive step

For $n \in S$, let $\mathsf{Z}(n) = \{\vec{a} \in \mathbb{N}^k : \sum_{i=1}^k a_i g_i = n\}$.
For each $i \leq k$,

Toward a dynamic algorithm…the inductive step

For $n \in S$, let $\mathsf{Z}(n) = \{\vec{a} \in \mathbb{N}^k : \sum_{i=1}^k a_i g_i = n\}$.
For each $i \leq k$,

$$\begin{array}{rcl} \phi_i : \mathsf{Z}(n - g_i) & \longrightarrow & \mathsf{Z}(n) \\ \vec{a} & \longmapsto & \vec{a} + \vec{e}_i. \end{array}$$

Toward a dynamic algorithm... the inductive step

For $n \in S$, let $\mathsf{Z}(n) = \{\vec{a} \in \mathbb{N}^k : \sum_{i=1}^k a_i g_i = n\}$.
For each $i \leq k$,

$$\begin{array}{rcl} \phi_i : \mathsf{Z}(n - g_i) & \longrightarrow & \mathsf{Z}(n) \\ \vec{a} & \longmapsto & \vec{a} + \vec{e}_i. \end{array}$$

In particular,

$$\mathsf{Z}(n) = \bigcup_{i \leq k} \phi_i(\mathsf{Z}(n - g_i))$$

Toward a dynamic algorithm... the inductive step

For $n \in S$, let $\mathsf{Z}(n) = \{\vec{a} \in \mathbb{N}^k : \sum_{i=1}^k a_i g_i = n\}$.
For each $i \leq k$,

$$\begin{array}{rcl} \phi_i : \mathsf{Z}(n - g_i) & \longrightarrow & \mathsf{Z}(n) \\ \vec{a} & \longmapsto & \vec{a} + \vec{e}_i. \end{array}$$

In particular,

$$\mathsf{Z}(n) = \bigcup_{i \leq k} \phi_i(\mathsf{Z}(n - g_i))$$

Definition/Proposition (Cover morphisms)

Fix $n \in S$ and $i \leq k$. The *i-th cover morphism for n* is the map

$$\psi_i : \mathsf{bul}(n - g_i) \longrightarrow \mathsf{bul}(n)$$

Toward a dynamic algorithm...the inductive step

For $n \in S$, let $\mathsf{Z}(n) = \{\vec{a} \in \mathbb{N}^k : \sum_{i=1}^k a_i g_i = n\}$.
For each $i \leq k$,

$$\begin{array}{rcl} \phi_i : \mathsf{Z}(n - g_i) & \longrightarrow & \mathsf{Z}(n) \\ \vec{a} & \longmapsto & \vec{a} + \vec{e}_i. \end{array}$$

In particular,

$$\mathsf{Z}(n) = \bigcup_{i \leq k} \phi_i(\mathsf{Z}(n - g_i))$$

Definition/Proposition (Cover morphisms)

Fix $n \in S$ and $i \leq k$. The *i-th cover morphism for n* is the map

$$\psi_i : \mathsf{bul}(n - g_i) \longrightarrow \mathsf{bul}(n)$$

given by

$$\vec{b} \longmapsto \begin{cases} \vec{b} + e_i & \sum_{j=1}^k b_j g_j - n - g_i \notin S \\ \vec{b} & \sum_{j=1}^k b_j g_j - n - g_i \in S \end{cases}$$

Toward a dynamic algorithm... the inductive step

For $n \in S$, let $\mathsf{Z}(n) = \{\vec{a} \in \mathbb{N}^k : \sum_{i=1}^k a_i g_i = n\}$.
For each $i \leq k$,

$$\begin{array}{rcl} \phi_i : \mathsf{Z}(n - g_i) & \longrightarrow & \mathsf{Z}(n) \\ \vec{a} & \longmapsto & \vec{a} + \vec{e}_i. \end{array}$$

In particular,

$$\mathsf{Z}(n) = \bigcup_{i \leq k} \phi_i(\mathsf{Z}(n - g_i))$$

Definition/Proposition (Cover morphisms)

Fix $n \in S$ and $i \leq k$. The *i-th cover morphism for n* is the map

$$\psi_i : \mathrm{bul}(n - g_i) \longrightarrow \mathrm{bul}(n)$$

given by

$$\vec{b} \longmapsto \begin{cases} \vec{b} + e_i & \sum_{j=1}^k b_j g_j - n - g_i \notin S \\ \vec{b} & \sum_{j=1}^k b_j g_j - n - g_i \in S \end{cases}$$

Moreover, $\mathrm{bul}(n) = \bigcup_{i \leq k} \psi_i(\mathrm{bul}(n - g_i))$.**

Toward a dynamic algorithm. . . the base case

Definition (ω-primality in numerical monoids)

Fix a numerical monoid S and $n \in S$.

Toward a dynamic algorithm. . . the base case

Definition (ω-primality in numerical monoids)

Fix a numerical monoid S and $n \in S$.
$\omega_S(n)$ is the minimal m such that whenever $(\sum_{i=1}^{r} g_{j_i}) - n \in S$ for $r > m$, there exists $T \subset \{1, \ldots, r\}$ with $|T| \le m$ and $(\sum_{i \in T} g_{j_i}) - n \in S$.

Toward a dynamic algorithm…the base case

Definition (ω-primality in numerical monoids)

Fix a numerical monoid S and $n \in S$.
$\omega_S(n)$ is the minimal m such that whenever $(\sum_{i=1}^{r} g_{j_i}) - n \in S$ for $r > m$, there exists $T \subset \{1, \ldots, r\}$ with $|T| \leq m$ and $(\sum_{i \in T} g_{j_i}) - n \in S$.

Toward a dynamic algorithm... the base case

Definition (ω-primality in numerical monoids)

Fix a numerical monoid S and $n \in \mathbb{Z} = \mathsf{q}(S)$.
$\omega_S(n)$ is the minimal m such that whenever $(\sum_{i=1}^{r} g_{j_i}) - n \in S$ for $r > m$, there exists $T \subset \{1, \ldots, r\}$ with $|T| \le m$ and $(\sum_{i \in T} g_{j_i}) - n \in S$.

Toward a dynamic algorithm. . . the base case

Definition (ω-primality in numerical monoids)

Fix a numerical monoid S and $n \in \mathbb{Z} = \mathrm{q}(S)$.
$\omega_S(n)$ is the minimal m such that whenever $(\sum_{i=1}^{r} g_{j_i}) - n \in S$ for $r > m$, there exists $T \subset \{1, \ldots, r\}$ with $|T| \leq m$ and $(\sum_{i \in T} g_{j_i}) - n \in S$.

Remark

All properties of ω extend from S to $\mathbb{Z}$.

Toward a dynamic algorithm...the base case

Definition (ω-primality in numerical monoids)

Fix a numerical monoid S and $n \in \mathbb{Z} = \mathsf{q}(S)$.
$\omega_S(n)$ is the minimal m such that whenever $(\sum_{i=1}^{r} g_{j_i}) - n \in S$ for $r > m$, there exists $T \subset \{1, \ldots, r\}$ with $|T| \leq m$ and $(\sum_{i \in T} g_{j_i}) - n \in S$.

Remark

All properties of ω extend from S to $\mathbb{Z}$.

Proposition

For $n \in \mathbb{Z}$, the following are equivalent:

(i) $\omega(n) = 0$,

(ii) $\mathsf{bul}(n) = \{\vec{0}\}$,

(iii) $-n \in S$.

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	bul(n)

$n \in \mathbb{Z}$	$\omega(n)$	bul(n)

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	$\mathrm{bul}(n)$
≤ -44	0	$\{\vec{0}\}$

$n \in \mathbb{Z}$	$\omega(n)$	$\mathrm{bul}(n)$

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$
≤ -44	0	$\{\vec{0}\}$
-43	1	$\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	$\mathrm{bul}(n)$
≤ -44	0	$\{\vec{0}\}$
-43	1	$\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-42	0	$\{\vec{0}\}$

$n \in \mathbb{Z}$	$\omega(n)$	$\mathrm{bul}(n)$

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$
≤ -44	0	$\{\vec{0}\}$
-43	1	$\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-42	0	$\{\vec{0}\}$
$\vdots$	$\vdots$	$\vdots$
-38	0	$\{\vec{0}\}$

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$
≤ -44	0	$\{\vec{0}\}$
-43	1	$\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-42	0	$\{\vec{0}\}$
$\vdots$	$\vdots$	$\vdots$
-38	0	$\{\vec{0}\}$
-37	2	$\{2\vec{e}_1, \vec{e}_2, \vec{e}_3\}$

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	$\mathrm{bul}(n)$
≤ -44	0	$\{\vec{0}\}$
-43	1	$\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-42	0	$\{\vec{0}\}$
$\vdots$	$\vdots$	$\vdots$
-38	0	$\{\vec{0}\}$
-37	2	$\{2\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-36	0	$\{\vec{0}\}$
-35	0	$\{\vec{0}\}$

$n \in \mathbb{Z}$	$\omega(n)$	$\mathrm{bul}(n)$

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	$\mathrm{bul}(n)$
≤ -44	0	$\{\vec{0}\}$
-43	1	$\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-42	0	$\{\vec{0}\}$
$\vdots$	$\vdots$	$\vdots$
-38	0	$\{\vec{0}\}$
-37	2	$\{2\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-36	0	$\{\vec{0}\}$
-35	0	$\{\vec{0}\}$
-34	2	$\{\vec{e}_1, 2\vec{e}_2, \vec{e}_3\}$

$n \in \mathbb{Z}$	$\omega(n)$	$\mathrm{bul}(n)$

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	$\mathrm{bul}(n)$
≤ -44	0	$\{\vec{0}\}$
-43	1	$\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-42	0	$\{\vec{0}\}$
$\vdots$	$\vdots$	$\vdots$
-38	0	$\{\vec{0}\}$
-37	2	$\{2\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-36	0	$\{\vec{0}\}$
-35	0	$\{\vec{0}\}$
-34	2	$\{\vec{e}_1, 2\vec{e}_2, \vec{e}_3\}$
-33	0	$\{\vec{0}\}$
-32	0	$\{\vec{0}\}$

$n \in \mathbb{Z}$	$\omega(n)$	$\mathrm{bul}(n)$

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	$\mathsf{bul}(n)$
≤ -44	0	$\{\vec{0}\}$
-43	1	$\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-42	0	$\{\vec{0}\}$
$\vdots$	$\vdots$	$\vdots$
-38	0	$\{\vec{0}\}$
-37	2	$\{2\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-36	0	$\{\vec{0}\}$
-35	0	$\{\vec{0}\}$
-34	2	$\{\vec{e}_1, 2\vec{e}_2, \vec{e}_3\}$
-33	0	$\{\vec{0}\}$
-32	0	$\{\vec{0}\}$
-31	3	$\{3\vec{e}_1, \vec{e}_2, \vec{e}_3\}$

$n \in \mathbb{Z}$	$\omega(n)$	$\mathsf{bul}(n)$

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$
≤ -44	0	$\{\vec{0}\}$
-43	1	$\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-42	0	$\{\vec{0}\}$
$\vdots$	$\vdots$	$\vdots$
-38	0	$\{\vec{0}\}$
-37	2	$\{2\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-36	0	$\{\vec{0}\}$
-35	0	$\{\vec{0}\}$
-34	2	$\{\vec{e}_1, 2\vec{e}_2, \vec{e}_3\}$
-33	0	$\{\vec{0}\}$
-32	0	$\{\vec{0}\}$
-31	3	$\{3\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
$\vdots$	$\vdots$	$\vdots$

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$
≤ -44	0	$\{\vec{0}\}$
-43	1	$\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-42	0	$\{\vec{0}\}$
$\vdots$	$\vdots$	$\vdots$
-38	0	$\{\vec{0}\}$
-37	2	$\{2\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-36	0	$\{\vec{0}\}$
-35	0	$\{\vec{0}\}$
-34	2	$\{\vec{e}_1, 2\vec{e}_2, \vec{e}_3\}$
-33	0	$\{\vec{0}\}$
-32	0	$\{\vec{0}\}$
-31	3	$\{3\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
$\vdots$	$\vdots$	$\vdots$

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$
1	5	$\{5\vec{e}_1, (2, 1, 0), \ldots\}$

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$
≤ -44	0	$\{\vec{0}\}$
-43	1	$\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-42	0	$\{\vec{0}\}$
$\vdots$	$\vdots$	$\vdots$
-38	0	$\{\vec{0}\}$
-37	2	$\{2\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-36	0	$\{\vec{0}\}$
-35	0	$\{\vec{0}\}$
-34	2	$\{\vec{e}_1, 2\vec{e}_2, \vec{e}_3\}$
-33	0	$\{\vec{0}\}$
-32	0	$\{\vec{0}\}$
-31	3	$\{3\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
$\vdots$	$\vdots$	$\vdots$

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$
1	5	$\{5\vec{e}_1, (2, 1, 0), \ldots\}$
2	7	$\{7\vec{e}_1, 6\vec{e}_2, \ldots\}$

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$
≤ -44	0	$\{\vec{0}\}$
-43	1	$\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-42	0	$\{\vec{0}\}$
$\vdots$	$\vdots$	$\vdots$
-38	0	$\{\vec{0}\}$
-37	2	$\{2\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-36	0	$\{\vec{0}\}$
-35	0	$\{\vec{0}\}$
-34	2	$\{\vec{e}_1, 2\vec{e}_2, \vec{e}_3\}$
-33	0	$\{\vec{0}\}$
-32	0	$\{\vec{0}\}$
-31	3	$\{3\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
$\vdots$	$\vdots$	$\vdots$

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$
1	5	$\{5\vec{e}_1, (2, 1, 0), \ldots\}$
2	7	$\{7\vec{e}_1, 6\vec{e}_2, \ldots\}$
3	3	$\{3\vec{e}_3, 2\vec{e}_2, \ldots\}$
4	4	$\{4\vec{e}_1, 4\vec{e}_2, \ldots\}$
5	9	$\{9\vec{e}_1, (6, 1, 0), \ldots\}$
6	3	$\{3\vec{e}_3, 2\vec{e}_2, \ldots\}$
7	6	$\{6\vec{e}_1, (3, 1, 0), \ldots\}$
8	8	$\{8\vec{e}_1, (5, 2, 0), \ldots, \}$
9	3	$\{3\vec{e}_1, 3\vec{e}_3, \ldots\}$
$\vdots$	$\vdots$	$\vdots$

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$
≤ -44	0	$\{\vec{0}\}$
-43	1	$\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-42	0	$\{\vec{0}\}$
$\vdots$	$\vdots$	$\vdots$
-38	0	$\{\vec{0}\}$
-37	2	$\{2\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-36	0	$\{\vec{0}\}$
-35	0	$\{\vec{0}\}$
-34	2	$\{\vec{e}_1, 2\vec{e}_2, \vec{e}_3\}$
-33	0	$\{\vec{0}\}$
-32	0	$\{\vec{0}\}$
-31	3	$\{3\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
$\vdots$	$\vdots$	$\vdots$

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$
1	5	$\{5\vec{e}_1, (2, 1, 0), \ldots\}$
2	7	$\{7\vec{e}_1, 6\vec{e}_2, \ldots\}$
3	3	$\{3\vec{e}_3, 2\vec{e}_2, \ldots\}$
4	4	$\{4\vec{e}_1, 4\vec{e}_2, \ldots\}$
5	9	$\{9\vec{e}_1, (6, 1, 0), \ldots\}$
6	3	$\{3\vec{e}_3, 2\vec{e}_2, \ldots\}$
7	6	$\{6\vec{e}_1, (3, 1, 0), \ldots\}$
8	8	$\{8\vec{e}_1, (5, 2, 0), \ldots,\}$
9	3	$\{3\vec{e}_1, 3\vec{e}_3, \ldots\}$
$\vdots$	$\vdots$	$\vdots$
148	28	$\{28\vec{e}_1, \ldots\}$

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$
≤ -44	0	$\{\vec{0}\}$
-43	1	$\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-42	0	$\{\vec{0}\}$
$\vdots$	$\vdots$	$\vdots$
-38	0	$\{\vec{0}\}$
-37	2	$\{2\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-36	0	$\{\vec{0}\}$
-35	0	$\{\vec{0}\}$
-34	2	$\{\vec{e}_1, 2\vec{e}_2, \vec{e}_3\}$
-33	0	$\{\vec{0}\}$
-32	0	$\{\vec{0}\}$
-31	3	$\{3\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
$\vdots$	$\vdots$	$\vdots$

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$
1	5	$\{5\vec{e}_1, (2, 1, 0), \ldots\}$
2	7	$\{7\vec{e}_1, 6\vec{e}_2, \ldots\}$
3	3	$\{3\vec{e}_3, 2\vec{e}_2, \ldots\}$
4	4	$\{4\vec{e}_1, 4\vec{e}_2, \ldots\}$
5	9	$\{9\vec{e}_1, (6, 1, 0), \ldots\}$
6	3	$\{3\vec{e}_3, 2\vec{e}_2, \ldots\}$
7	6	$\{6\vec{e}_1, (3, 1, 0), \ldots\}$
8	8	$\{8\vec{e}_1, (5, 2, 0), \ldots, \}$
9	3	$\{3\vec{e}_1, 3\vec{e}_3, \ldots\}$
$\vdots$	$\vdots$	$\vdots$
148	28	$\{28\vec{e}_1, \ldots\}$
149	33	$\{33\vec{e}_1, \ldots\}$

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$
≤ -44	0	$\{\vec{0}\}$
-43	1	$\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-42	0	$\{\vec{0}\}$
$\vdots$	$\vdots$	$\vdots$
-38	0	$\{\vec{0}\}$
-37	2	$\{2\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
-36	0	$\{\vec{0}\}$
-35	0	$\{\vec{0}\}$
-34	2	$\{\vec{e}_1, 2\vec{e}_2, \vec{e}_3\}$
-33	0	$\{\vec{0}\}$
-32	0	$\{\vec{0}\}$
-31	3	$\{3\vec{e}_1, \vec{e}_2, \vec{e}_3\}$
$\vdots$	$\vdots$	$\vdots$

$n \in \mathbb{Z}$	$\omega(n)$	$\text{bul}(n)$
1	5	$\{5\vec{e}_1, (2, 1, 0), \ldots\}$
2	7	$\{7\vec{e}_1, 6\vec{e}_2, \ldots\}$
3	3	$\{3\vec{e}_3, 2\vec{e}_2, \ldots\}$
4	4	$\{4\vec{e}_1, 4\vec{e}_2, \ldots\}$
5	9	$\{9\vec{e}_1, (6, 1, 0), \ldots\}$
6	3	$\{3\vec{e}_3, 2\vec{e}_2, \ldots\}$
7	6	$\{6\vec{e}_1, (3, 1, 0), \ldots\}$
8	8	$\{8\vec{e}_1, (5, 2, 0), \ldots, \}$
9	3	$\{3\vec{e}_1, 3\vec{e}_3, \ldots\}$
$\vdots$	$\vdots$	$\vdots$
148	28	$\{28\vec{e}_1, \ldots\}$
149	33	$\{33\vec{e}_1, \ldots\}$
150	25	$\{25\vec{e}_1, \ldots\}$

A dynamic algorithm!

Example

$McN = \langle 6, 9, 20 \rangle = \{0, 6, 9, 12, 15, 18, 20, 21, \ldots\}$.

$n \in \mathbb{Z}$	$\omega(n)$	bul(n)

$n \in \mathbb{Z}$	$\omega(n)$	bul(n)
6	3	$\{3\vec{e}_3, 2\vec{e}_2, \ldots\}$
9	3	$\{3\vec{e}_1, 3\vec{e}_3, \ldots\}$
$\vdots$	$\vdots$	$\vdots$
148	28	$\{28\vec{e}_1, \ldots\}$
149	33	$\{33\vec{e}_1, \ldots\}$
150	25	$\{25\vec{e}_1, \ldots\}$

Runtime comparison

Runtime comparison

S	$n \in S$	$\omega_S(n)$	Existing	Dynamic
$\langle 6, 9, 20 \rangle$	1000	170	0m 67s	0.4s
$\langle 6, 9, 20 \rangle$	2000	340	17m 20s	3.1s
$\langle 31, 39, 45, 52 \rangle$	1000	40	0m 39s	0.3s
$\langle 31, 39, 45, 52 \rangle$	2000	71	24m 31s	2.0s
$\langle 54, 67, 69, 73, 75 \rangle$	1000	23	2m 02s	0.7s
$\langle 54, 67, 69, 73, 75 \rangle$	1500	33	22m 50s	2.3s
$\langle 54, 67, 69, 73, 75 \rangle$	3000	61	———	44.3s
$\langle 54, 67, 69, 73, 75 \rangle$	5000	98	———	8m 59.0s

Runtime comparison

S	$n \in S$	$\omega_S(n)$	Existing	Dynamic
$\langle 6, 9, 20 \rangle$	1000	170	0m 67s	0.4s
$\langle 6, 9, 20 \rangle$	2000	340	17m 20s	3.1s
$\langle 31, 39, 45, 52 \rangle$	1000	40	0m 39s	0.3s
$\langle 31, 39, 45, 52 \rangle$	2000	71	24m 31s	2.0s
$\langle 54, 67, 69, 73, 75 \rangle$	1000	23	2m 02s	0.7s
$\langle 54, 67, 69, 73, 75 \rangle$	1500	33	22m 50s	2.3s
$\langle 54, 67, 69, 73, 75 \rangle$	3000	61	———	44.3s
$\langle 54, 67, 69, 73, 75 \rangle$	5000	98	———	8m 59.0s

Sage: Open Source Mathematics Software, available at

`www.sagemath.org`.

GAP Numerical Semigroups Package, available at

`http://www.gap-system.org/Packages/numericalsgps.html`.

Future directions

What about more general (finitely generated) monoids M?

Future directions

What about more general (finitely generated) monoids M?

- Characterization of ω_M in terms of maximal length bullets?

Future directions

What about more general (finitely generated) monoids M?

- Characterization of ω_M in terms of maximal length bullets? ✓

Future directions

What about more general (finitely generated) monoids M?

- Characterization of ω_M in terms of maximal length bullets? ✓
- Extension of ω_M to $q(M)$?

Future directions

What about more general (finitely generated) monoids M?

- Characterization of ω_M in terms of maximal length bullets? ✓
- Extension of ω_M to $q(M)$? ✓

Future directions

What about more general (finitely generated) monoids M?

- Characterization of ω_M in terms of maximal length bullets? ✓
- Extension of ω_M to $q(M)$? ✓
- Iterative construction of bullets from cover maps?

Future directions

What about more general (finitely generated) monoids M?

- Characterization of ω_M in terms of maximal length bullets? ✓
- Extension of ω_M to $q(M)$? ✓
- Iterative construction of bullets from cover maps? ✓

Future directions

What about more general (finitely generated) monoids M?

- Characterization of ω_M in terms of maximal length bullets? ✓
- Extension of ω_M to $\mathsf{q}(M)$? ✓
- Iterative construction of bullets from cover maps? ✓

Problem:

Future directions

What about more general (finitely generated) monoids M?

- Characterization of ω_M in terms of maximal length bullets? ✓
- Extension of ω_M to $\mathrm{q}(M)$? ✓
- Iterative construction of bullets from cover maps? ✓

Problem: the base case!

Future directions

What about more general (finitely generated) monoids M?

- Characterization of ω_M in terms of maximal length bullets? $\checkmark$
- Extension of ω_M to $\mathsf{q}(M)$? $\checkmark$
- Iterative construction of bullets from cover maps? $\checkmark$

Problem: the base case!

Problem

Find a dynamic algorithm to compute ω-primality in M.

Future directions

What about more general (finitely generated) monoids M?

- Characterization of ω_M in terms of maximal length bullets? ✓
- Extension of ω_M to $\mathsf{q}(M)$? ✓
- Iterative construction of bullets from cover maps? ✓

Problem: the base case!

Problem

Find a dynamic algorithm to compute ω-primality in M.

Question

Are there dynamic algorithms for computing other factorization invariants?

References

Alfred Geroldinger (1997)
Chains of factorizations in weakly Krull domains.
Colloq. Math. 72 (1997) 53 – 81.

David Anderson, Scott Chapman, Nathan Kaplan, and Desmond Torkornoo (2011)
An algorithm to compute ω-primality in a numerical monoid.
Semigroup Forum 82 (2011), no. 1, 96 – 108.

Christopher O'Neill, Roberto Pelayo (2014)
How do you measure primality?
American Mathematical Monthly, forthcoming.

Thomas Barron, Christopher O'Neill, Roberto Pelayo (2014)
On the computation of delta sets and ω-primality in numerical monoids.
preprint.

References

Alfred Geroldinger (1997)
Chains of factorizations in weakly Krull domains.
Colloq. Math. 72 (1997) 53 – 81.

David Anderson, Scott Chapman, Nathan Kaplan, and Desmond Torkornoo (2011)
An algorithm to compute ω-primality in a numerical monoid.
Semigroup Forum 82 (2011), no. 1, 96 – 108.

Christopher O'Neill, Roberto Pelayo (2014)
How do you measure primality?
American Mathematical Monthly, forthcoming.

Thomas Barron, Christopher O'Neill, Roberto Pelayo (2014)
On the computation of delta sets and ω-primality in numerical monoids.
preprint.

Thanks!