# Numerical semigroups in Sage

Christopher O'Neill

University of California Davis

*coneill@math.ucdavis.edu*

Oct 26, 2016

# 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 $|\mathbb{N} \setminus S| < \infty$.

## 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 $|\mathbb{N} \setminus S| < \infty$.

## 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 $|\mathbb{N} \setminus S| < \infty$.

## Example

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

Factorizations:

$$60 \quad =$$

# Numerical monoids

## Definition

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

## Example

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

$$60 = 7(6) + 2(9)$$

# Numerical monoids

## Definition

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

## Example

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

$$\begin{aligned} 60 &= 7(6) + 2(9) \\ &= \qquad\qquad 3(20) \end{aligned}$$

# Numerical monoids

## Definition

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

## Example

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

Factorizations:

$$\begin{array}{rcllcl} 60 & = & 7(6) + 2(9) & & \rightsquigarrow & (7, 2, 0) \\ & = & & 3(20) & \rightsquigarrow & (0, 0, 3) \end{array}$$

# Numerical monoids

## Definition

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

## Example

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

Factorizations:

$$\begin{aligned} 60 &= 7(6) + 2(9) & \rightsquigarrow & \quad (7, 2, 0) \\ &= 3(20) & \rightsquigarrow & \quad (0, 0, 3) \end{aligned}$$

Unique minimal generating set:

# Numerical monoids

## Definition

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

## Example

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

Factorizations:

$$\begin{array}{rclcl} 60 & = & 7(6) + 2(9) & \rightsquigarrow & (7,2,0) \\ & = & 3(20) & \rightsquigarrow & (0,0,3) \end{array}$$

Unique minimal generating set:

$$\langle 15, 17, 22, 32, 40, 42, 56, 58 \rangle$$

# Numerical monoids

**Definition**

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

**Example**

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

$$\begin{aligned} 60 &= 7(6) + 2(9) & \rightsquigarrow & \quad (7, 2, 0) \\ &= 3(20) & \rightsquigarrow & \quad (0, 0, 3) \end{aligned}$$

Unique minimal generating set:

$$\langle 15, 17, 22, 32, 40, 42, 56, 58 \rangle \rightsquigarrow \langle 15, 17, 22, \quad 40, 42, \quad 58 \rangle$$

# Numerical monoids

**Definition**

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

**Example**

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

Factorizations:

$$\begin{array}{rclcc} 60 & = & 7(6) + 2(9) & \rightsquigarrow & (7, 2, 0) \\ & = & 3(20) & \rightsquigarrow & (0, 0, 3) \end{array}$$

Unique minimal generating set:

$$\langle 15, 17, 22, 32, 40, 42, 56, 58 \rangle \rightsquigarrow \langle 15, 17, 22, \quad 40, 42, \quad 58 \rangle$$

$$32 = 15 + 17 \qquad\qquad 56 = 2 \cdot 17 + 22$$

# Factorization invariants

Fix a numerical monoid $S = \langle n_1, \ldots, n_k \rangle$. For $n \in S$,

# Factorization invariants

Fix a numerical monoid $S = \langle n_1, \ldots, n_k \rangle$. For $n \in S$,

- Max and min factorization length:

$$\mathsf{M}(n), \mathsf{m}(n) \in \mathbb{Z}_{\geq 0}$$

# Factorization invariants

Fix a numerical monoid $S = \langle n_1, \ldots, n_k \rangle$. For $n \in S$,

- Max and min factorization length:

$$\mathsf{M}(n), \mathsf{m}(n) \in \mathbb{Z}_{\ge 0}$$

- Delta set:

$$\Delta(n) \subset \mathbb{Z}_{\ge 1} \qquad \Delta(S) = \bigcup_{m \in S} \Delta(m)$$

# Factorization invariants

Fix a numerical monoid $S = \langle n_1, \ldots, n_k \rangle$. For $n \in S$,

- Max and min factorization length:

$$\mathsf{M}(n), \mathsf{m}(n) \in \mathbb{Z}_{\geq 0}$$

- Delta set:

$$\Delta(n) \subset \mathbb{Z}_{\geq 1} \qquad \Delta(S) = \bigcup_{m \in S} \Delta(m)$$

- Catenary degree:

$$\mathsf{c}(n) \in \mathbb{Z}_{\geq 1} \qquad \mathsf{c}(S) = \max\{\mathsf{c}(m) : m \in S\}$$

# Factorization invariants

Fix a numerical monoid $S = \langle n_1, \ldots, n_k \rangle$. For $n \in S$,

- Max and min factorization length:

$$\mathsf{M}(n), \mathsf{m}(n) \in \mathbb{Z}_{\ge 0}$$

- Delta set:

$$\Delta(n) \subset \mathbb{Z}_{\ge 1} \qquad \Delta(S) = \bigcup_{m \in S} \Delta(m)$$

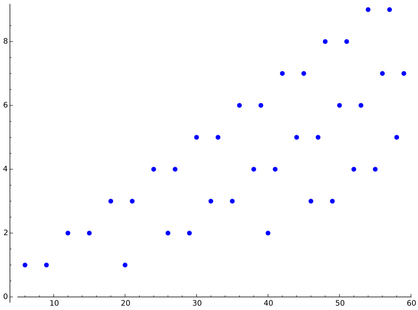
- Catenary degree:

$$\mathsf{c}(n) \in \mathbb{Z}_{\ge 1} \qquad \mathsf{c}(S) = \max\{\mathsf{c}(m) : m \in S\}$$

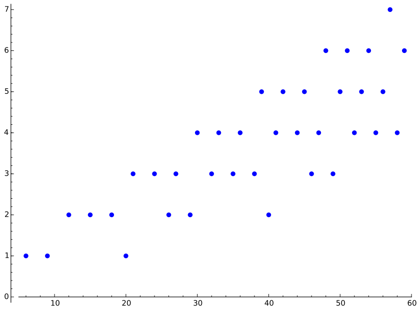
- $\omega$-primality:

$$\omega(n) \in \mathbb{Z}_{\ge 1}$$

Invariant behavior for large numerical monoid elements:



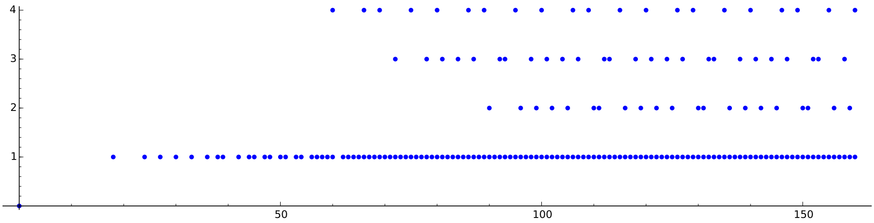

$\mathsf{M} : \langle 6, 9, 20 \rangle \to \mathbb{N}$




$\mathsf{m} : \langle 6, 9, 20 \rangle \to \mathbb{N}$

# Factorization invariants

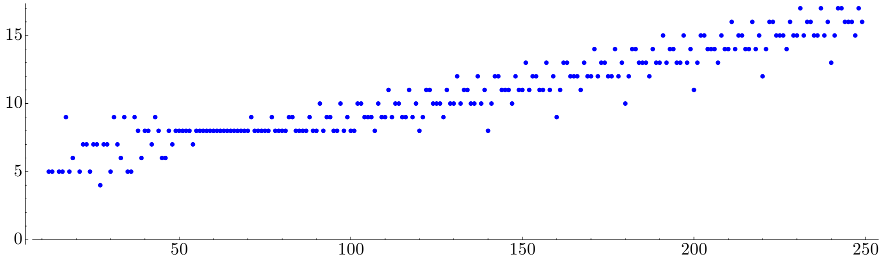
Invariant behavior for large numerical monoid elements:




$$\Delta : \langle 6, 9, 20 \rangle \to 2^{\mathbb{N}}$$

# Factorization invariants

Invariant behavior for large numerical monoid elements:




$$n \longmapsto \mathsf{c}(\langle n, n+6, n+9, n+20 \rangle)$$

# Software!

# Software!

Input:

- Numerical monoid $S = \langle n_1, \ldots, n_k \rangle$
- Monoid element $n \in S$

# Software!

Input:

- Numerical monoid $S = \langle n_1, \ldots, n_k \rangle$
- Monoid element $n \in S$

Output:

- Invariant value
  Delta set, catenary degree, list of factorizations, ...

# Software!

Input:

- Numerical monoid $S = \langle n_1, \ldots, n_k \rangle$
- Monoid element $n \in S$

Output:

- Invariant value
  Delta set, catenary degree, list of factorizations, ...

GAP Numerical Semigroups Package, available at

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

# Software!

Input:

- Numerical monoid $S = \langle n_1, \ldots, n_k \rangle$
- Monoid element $n \in S$

Output:

- Invariant value
  Delta set, catenary degree, list of factorizations, ...

GAP Numerical Semigroups Package, available at

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

Let's see it in action!

# GAP from within Sage

Using GAP within Sage via `gap_console()`

# GAP from within Sage

Using GAP within Sage via `gap_console()`

- Conflated message

# GAP from within Sage

Using GAP within Sage via `gap_console()`

- Conflated message
- Less intuitive language

# GAP from within Sage

Using GAP within Sage via `gap_console()`

- Conflated message
- Less intuitive language
- Port data to Sage for plots:

# GAP from within Sage

Using GAP within Sage via `gap_console()`

- Conflated message
- Less intuitive language
- Port data to Sage for plots:

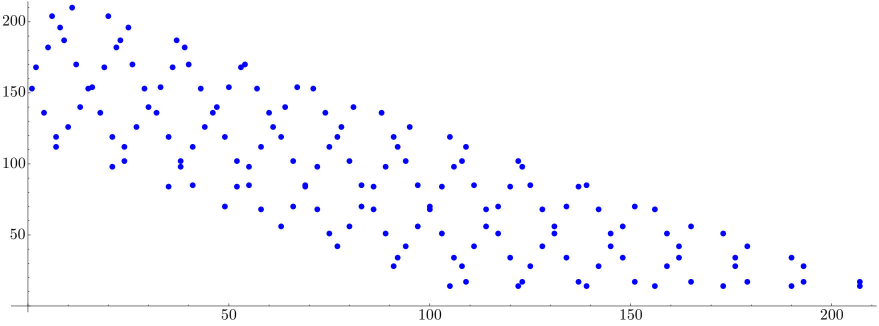
Conjecture: For all $S$, $s \in \mathbb{N} \setminus S$, there exists a height 2 Leamer $s$-atom.

# GAP from within Sage

Using GAP within Sage via `gap_console()`

- Conflated message
- Less intuitive language
- Port data to Sage for plots:

Conjecture: For all $S$, $s \in \mathbb{N} \setminus S$, there exists a height 2 Leamer $s$-atom.




$S = \langle 14, 17 \rangle$

# Sage wrapper

Solution: NumericalSemigroup Sage class!

# Sage wrapper

Solution: NumericalSemigroup Sage class!

NumericalSemigroup instance $\rightsquigarrow$ GAP object

# Sage wrapper

Solution: NumericalSemigroup Sage class!

$$\text{NumericalSemigroup instance} \rightsquigarrow \text{GAP object}$$

Benefits:

# Sage wrapper

Solution: NumericalSemigroup Sage class!

$$\text{NumericalSemigroup instance} \rightsquigarrow \text{GAP object}$$

Benefits:

- Returns "Sage types"

# Sage wrapper

Solution: NumericalSemigroup Sage class!

$$\text{NumericalSemigroup instance} \rightsquigarrow \text{GAP object}$$

Benefits:

- Returns "Sage types"
- Shorter function names

# Sage wrapper

Solution: NumericalSemigroup Sage class!

NumericalSemigroup instance $\rightsquigarrow$ GAP object

Benefits:

- Returns "Sage types"
- Shorter function names
- Some computed values stored internally

# Sage wrapper

Solution: NumericalSemigroup Sage class!

NumericalSemigroup instance $\rightsquigarrow$ GAP object

Benefits:

- Returns "Sage types"
- Shorter function names
- Some computed values stored internally

Calling `S.CatenaryDegree(500)` from Sage:

# Sage wrapper

Solution: NumericalSemigroup Sage class!

NumericalSemigroup instance $\rightsquigarrow$ GAP object

Benefits:

- Returns "Sage types"
- Shorter function names
- Some computed values stored internally

Calling `S.CatenaryDegree(500)` from Sage:

- Call `CatenaryDegreeOfElementInNumericalSemigroup(500,-)`

# Sage wrapper

Solution: NumericalSemigroup Sage class!

NumericalSemigroup instance $\rightsquigarrow$ GAP object

Benefits:

- Returns "Sage types"
- Shorter function names
- Some computed values stored internally

Calling `S.CatenaryDegree(500)` from Sage:

- Call `CatenaryDegreeOfElementInNumericalSemigroup(500,-)`
- Convert result to "Sage int"

# Sage wrapper

Solution: NumericalSemigroup Sage class!

$$\text{NumericalSemigroup instance} \rightsquigarrow \text{GAP object}$$

Benefits:

- Returns "Sage types"
- Shorter function names
- Some computed values stored internally

Calling `S.CatenaryDegree(500)` from Sage:

- Call `CatenaryDegreeOfElementInNumericalSemigroup(500,-)`
- Convert result to "Sage int"
- Archive value and return

# Sage wrapper

Solution: NumericalSemigroup Sage class!

NumericalSemigroup instance $\rightsquigarrow$ GAP object

Benefits:

- Returns "Sage types"
- Shorter function names
- Some computed values stored internally

Calling `S.CatenaryDegree(500)` from Sage:

- Call `CatenaryDegreeOfElementInNumericalSemigroup(500,-)`
- Convert result to "Sage int"
- Archive value and return

Let's see it in action!