On the seeds and the great-grandchildren of a numerical semigroup

Maria Bras-Amorós

INdAM workshop
International meeting on numerical semigroups

Roma, June 13, 2022

Counting semigroups by genus

Let n_g denote the number of numerical semigroups of genus g.

Counting semigroups by genus

Let n_g denote the number of numerical semigroups of genus g.

- $n_0 = 1$, since the unique numerical semigroup of genus 0 is $\mathbb{N}_0$

Counting semigroups by genus

Let n_g denote the number of numerical semigroups of genus g.

- $n_0 = 1$, since the unique numerical semigroup of genus 0 is $\mathbb{N}_0$
- $n_1 = 1$, since the unique numerical semigroup of genus 1 is $\mathbb{N}_0 \setminus \{1\}$

Counting semigroups by genus

Let n_g denote the number of numerical semigroups of genus g.

- $n_0 = 1$, since the unique numerical semigroup of genus 0 is $\mathbb{N}_0$
- $n_1 = 1$, since the unique numerical semigroup of genus 1 is $\mathbb{N}_0 \setminus \{1\}$
- $n_2 = 2$. Indeed the unique numerical semigroups of genus 2 are

$$\{0, 3, 4, 5, \dots\},$$

$$\{0, 2, 4, 5, \dots\}.$$

Counting semigroups by genus

Let n_g denote the number of numerical semigroups of genus g.

- $n_0 = 1$, since the unique numerical semigroup of genus 0 is $\mathbb{N}_0$
- $n_1 = 1$, since the unique numerical semigroup of genus 1 is $\mathbb{N}_0 \setminus \{1\}$
- $n_2 = 2$. Indeed the unique numerical semigroups of genus 2 are

$$\{0, 3, 4, 5, \dots\},$$

$$\{0, 2, 4, 5, \dots\}.$$

- $n_3 = 4$
- $n_4 = 7$
- $n_5 = 12$
- $n_6 = 23$
- $n_7 = 39$
- $n_8 = 67$

  $\vdots$

Conjecture

[IMNS(!) 2008, Segovia 2007, Semigroup Forum 2008]

$$n_g \geqslant n_{g-1} + n_{g-2}$$
$$n_g \geqslant n_{g-1}$$

Conjecture

[IMNS(!) 2008, Segovia 2007, Semigroup Forum 2008]

$$n_g \geqslant n_{g-1} + n_{g-2}$$
$$n_g \geqslant n_{g-1}$$

Many efforts to compute n_g for large g:

- N. Medeiros, B. (2008), M. Delgado: Brute approach
- B. (2009): Need to check only one new generator
- J. Fromentin - F. Hivert (2016): Decomposition numbers, parallelization, depth first search (in GitHub)
- B. - J. Fernández-González (2018): Seeds
- B. - J. Fernández-González (2020): Right-generators descendant (in GitHub)
- (2022) (Seeds algorithm revisited in GitHub)

Counting semigroups by genus

Conjecture

[IMNS(!) 2008, Segovia 2007, Semigroup Forum 2008]

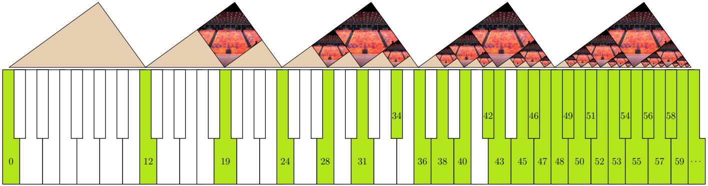
$$n_g \geqslant n_{g-1} + n_{g-2}$$
$$n_g \geqslant n_{g-1}$$

Many efforts to compute n_g for large g:

- N. Medeiros, B. (2008), M. Delgado: Brute approach
- B. (2009): Need to check only one new generator
- J. Fromentin - F. Hivert (2016): Decomposition numbers, parallelization, depth first search (in GitHub)
- B. - J. Fernández-González (2018): Seeds
- B. - J. Fernández-González (2020): Right-generators descendant (in GitHub)
- (2022) (Seeds algorithm revisited in GitHub)

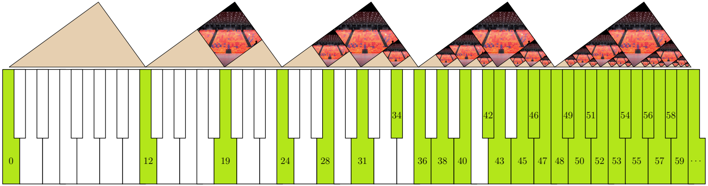
On-line encyclopedia of integer sequences: A007323

Example: The Well-tempered semigroup



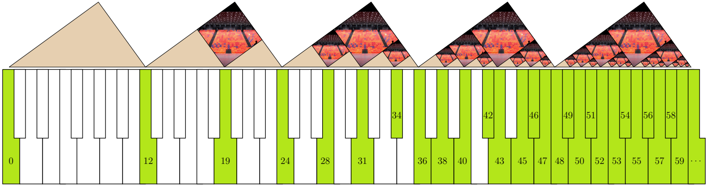
$$H = \{0, 12, 19, 24, 28, 31, 34, 36, 38, 40, 42, 43, 45, 46, 47, 48, \dots\}$$

Example: The Well-tempered semigroup



$$H = \{\underbrace{0, 12, 19, 24, 28, 31, 34, 36, 38, 40, 42, 43}_{\text{L:=left elemets}}, 45, 46, 47, 48, \ldots\}$$

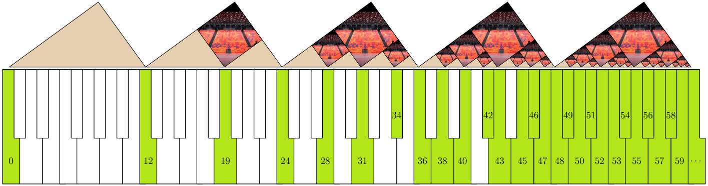
Example: The Well-tempered semigroup



$$H = \{\underbrace{0, 12, 19, 24, 28, 31, 34, 36, 38, 40, 42, 43}_{\text{L:=left elemets}}, 45, 46, 47, 48, \ldots\}$$

- $m = 12$ (multiplicity)

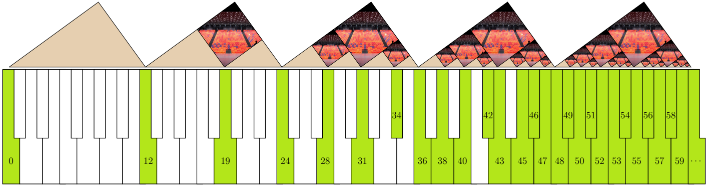
Example: The Well-tempered semigroup



$$H = \{\underbrace{0, 12, 19, 24, 28, 31, 34, 36, 38, 40, 42, 43}_{\text{L:=left elemets}}, 45, 46, 47, 48, \dots\}$$

- $m = 12$ (multiplicity)
- $g = 33$ (genus)

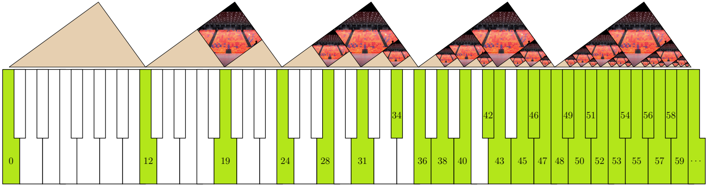
Example: The Well-tempered semigroup



$$H = \{\underbrace{0, 12, 19, 24, 28, 31, 34, 36, 38, 40, 42, 43}_{\text{L:=left elemets}}, 45, 46, 47, 48, \dots\}$$

- $m = 12$ (multiplicity)
- $g = 33$ (genus)
- $c = 45$ (conductor)

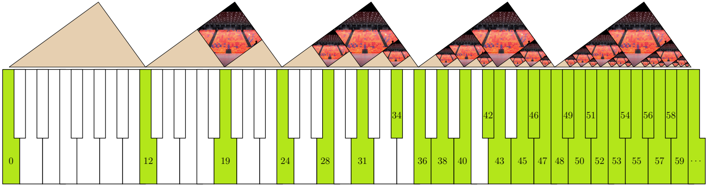
Example: The Well-tempered semigroup



$$H = \{\underbrace{0, 12, 19, 24, 28, 31, 34, 36, 38, 40, 42, 43}_{\text{L:=left elemets}}, 45, 46, 47, 48, \dots\}$$

- $m = 12$ (multiplicity)
- $g = 33$ (genus)
- $c = 45$ (conductor)
- $k := \#L = 12$ (rank),

Example: The Well-tempered semigroup



$$H = \{\underbrace{0, 12, 19, 24, 28, 31, 34, 36, 38, 40, 42, 43}_{\text{L:=left elemets}}, 45, 46, 47, 48, \ldots\}$$

- $m = 12$ (multiplicity)
- $g = 33$ (genus)
- $c = 45$ (conductor)
- $k := \#L = 12$ (rank),
  $k = c - g$.

Some notation

Enumeration of
$H = \{0, 12, 19, 24, 28, 31, 34, 36, 38, 40, 42, 43, 45, 46, 47, 48, \dots\}$:

$$
\begin{aligned}
\lambda_0 &= 0 \\
\lambda_1 &= 12 \\
\lambda_2 &= 19 \\
\lambda_3 &= 24 \\
\lambda_4 &= 28 \\
\lambda_5 &= 31 \\
\lambda_6 &= 34 \\
\lambda_7 &= 36 \\
\lambda_8 &= 38 \\
\lambda_9 &= 40 \\
\lambda_{10} &= 42 \\
\lambda_{11} &= 43 \\
\lambda_{12} &= 45 \\
\lambda_{13} &= 46 \\
&\vdots
\end{aligned}
$$

Some notation

Enumeration of
$H = \{0, 12, 19, 24, 28, 31, 34, 36, 38, 40, 42, 43, 45, 46, 47, 48, \ldots\}$:

$$
\begin{aligned}
\lambda_0 &= 0\\
\lambda_1 &= 12\\
\lambda_2 &= 19\\
\lambda_3 &= 24\\
\lambda_4 &= 28\\
\lambda_5 &= 31\\
\lambda_6 &= 34\\
\lambda_7 &= 36\\
\lambda_8 &= 38\\
\lambda_9 &= 40\\
\lambda_{10} &= 42\\
\lambda_{11} &= 43\\
\lambda_{12} &= 45\\
\lambda_{13} &= 46\\
&\vdots
\end{aligned}
$$

Notice that $\lambda_k = c$.

Primitive elements

A primitive element of Λ is a non-gap $\lambda_t \neq 0$ such that, equivalently,

Primitive elements

A primitive element of Λ is a non-gap $\lambda_t \neq 0$ such that, equivalently,

- $\Lambda \setminus \{\lambda_t\}$ is a numerical semigroup,

Primitive elements

A primitive element of Λ is a non-gap $\lambda_t \neq 0$ such that, equivalently,

- $\Lambda \setminus \{\lambda_t\}$ is a numerical semigroup,
- $\lambda_t \neq \lambda_j + \lambda_{j'}$ for all $0 < j, j' < t$.

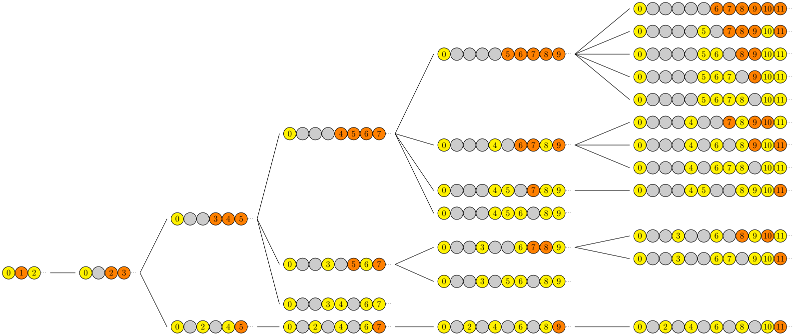
Primitive elements

A primitive element of Λ is a non-gap $\lambda_t \neq 0$ such that, equivalently,

- $\Lambda \setminus \{\lambda_t\}$ is a numerical semigroup,
- $\lambda_t \neq \lambda_j + \lambda_{j'}$ for all $0 < j, j' < t$.

Primitive elements larger than or equal to c are denoted right generators.

The tree of numerical semigroups



Strong generators

Suppose $k(\Lambda) > 1$ and suppose λ_t is a right generator of Λ.

The right generators of $\Lambda \setminus \{\lambda_t\}$ are

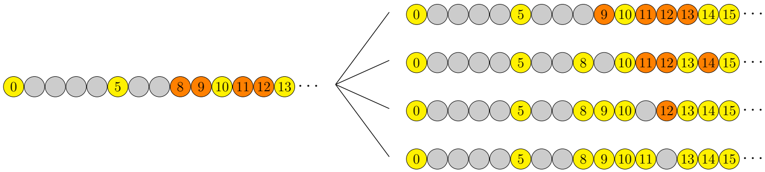
- the right generators of Λ larger than λ_t
- and, maybe, $m + \lambda_t$.

Strong generators

Suppose $k(\Lambda) > 1$ and suppose λ_t is a right generator of Λ.

The right generators of $\Lambda \setminus \{\lambda_t\}$ are

- the right generators of Λ larger than λ_t
- and, maybe, $m + \lambda_t$.

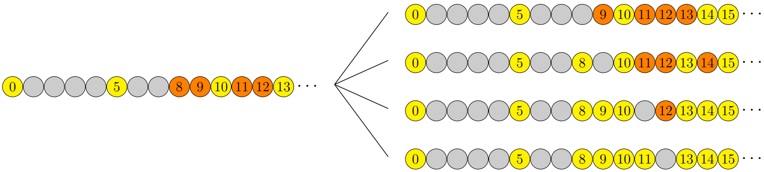


Strong generators

Suppose $k(\Lambda) > 1$ and suppose λ_t is a right generator of Λ.

The right generators of $\Lambda \setminus \{\lambda_t\}$ are

- the right generators of Λ larger than λ_t
- and, maybe, $m + \lambda_t$.



A strong generator of Λ is a right generator λ_t such that $m + \lambda_t$ is a new generator of $\Lambda \setminus \{\lambda_t\}$.

Seeds of a numerical semigroup

Definition. A non-gap λ_t with $\lambda_t \geqslant c$ is an order-i seed of Λ if

$$\lambda_t + \lambda_i \neq \lambda_j + \lambda_{j'}$$

for all $i < j, j' < t$.

Seeds of a numerical semigroup

Definition. A non-gap λ_t with $\lambda_t \geqslant c$ is an order-i seed of Λ if

$$\lambda_t + \lambda_i \neq \lambda_j + \lambda_{j'}$$

for all $i < j, j' < t$.

What makes $\lambda_t \geqslant c(\Lambda)$ an order-zero seed?

Seeds of a numerical semigroup

Definition. A non-gap λ_t with $\lambda_t \geqslant c$ is an order-i seed of Λ if

$$\lambda_t + \lambda_i \neq \lambda_j + \lambda_{j'}$$

for all $i < j, j' < t$.

What makes $\lambda_t \geqslant c(\Lambda)$ an order-zero seed?

$\lambda_t + 0 \neq \lambda_j + \lambda_{j'}$ for all positive j, j'.

Seeds of a numerical semigroup

Definition. A non-gap λ_t with $\lambda_t \geqslant c$ is an order-i seed of Λ if

$$\lambda_t + \lambda_i \neq \lambda_j + \lambda_{j'}$$

for all $i < j, j' < t$.

What makes $\lambda_t \geqslant c(\Lambda)$ an order-zero seed?

$\lambda_t + 0 \neq \lambda_j + \lambda_{j'}$ for all positive j, j'.

That is, the order-zero seeds of Λ are its right generators.

Seeds of a numerical semigroup

Definition. A non-gap λ_t with $\lambda_t \geqslant c$ is an order-i seed of Λ if

$$\lambda_t + \lambda_i \neq \lambda_j + \lambda_{j'}$$

for all $i < j, j' < t$.

What makes $\lambda_t \geqslant c(\Lambda)$ an order-zero seed?

$\lambda_t + 0 \neq \lambda_j + \lambda_{j'}$ for all positive j, j'.

That is, the order-zero seeds of Λ are its right generators.

What makes $\lambda_t \geqslant c(\Lambda)$ an order-one seed?

Seeds of a numerical semigroup

Definition. A non-gap λ_t with $\lambda_t \geqslant c$ is an order-i seed of Λ if

$$\lambda_t + \lambda_i \neq \lambda_j + \lambda_{j'}$$

for all $i < j, j' < t$.

What makes $\lambda_t \geqslant c(\Lambda)$ an order-zero seed?

$\lambda_t + 0 \neq \lambda_j + \lambda_{j'}$ for all positive j, j'.

That is, the order-zero seeds of Λ are its right generators.

What makes $\lambda_t \geqslant c(\Lambda)$ an order-one seed?

$\lambda_t + m \neq \lambda_j + \lambda_{j'}$ for all $1 < j, j' < t$.

Seeds of a numerical semigroup

Definition. A non-gap λ_t with $\lambda_t \geqslant c$ is an order-i seed of Λ if

$$\lambda_t + \lambda_i \neq \lambda_j + \lambda_{j'}$$

for all $i < j, j' < t$.

What makes $\lambda_t \geqslant c(\Lambda)$ an order-zero seed?

$\lambda_t + 0 \neq \lambda_j + \lambda_{j'}$ for all positive j, j'.

That is, the order-zero seeds of Λ are its right generators.

What makes $\lambda_t \geqslant c(\Lambda)$ an order-one seed?

$\lambda_t + m \neq \lambda_j + \lambda_{j'}$ for all $1 < j, j' < t$.

This is equivalent to λ_t being a strong generator.

Seeds of a numerical semigroup

Definition. A non-gap λ_t with $\lambda_t \geqslant c$ is an order-i seed of Λ if

$$\lambda_t + \lambda_i \neq \lambda_j + \lambda_{j'}$$

for all $i < j, j' < t$.

What makes $\lambda_t \geqslant c(\Lambda)$ an order-zero seed?

$\lambda_t + 0 \neq \lambda_j + \lambda_{j'}$ for all positive j, j'.

That is, the order-zero seeds of Λ are its right generators.

What makes $\lambda_t \geqslant c(\Lambda)$ an order-one seed?

$\lambda_t + m \neq \lambda_j + \lambda_{j'}$ for all $1 < j, j' < t$.

This is equivalent to λ_t being a strong generator.

That is, the order-one seeds of Λ are its strong generators.

Seeds of a numerical semigroup

Definition. A non-gap λ_t with $\lambda_t \geqslant c$ is an order-i seed of Λ if

$$\lambda_t + \lambda_i \neq \lambda_j + \lambda_{j'}$$

for all $i < j, j' < t$.

What makes $\lambda_t \geqslant c(\Lambda)$ an order-zero seed?

$\lambda_t + 0 \neq \lambda_j + \lambda_{j'}$ for all positive j, j'.

That is, the order-zero seeds of Λ are its right generators.

What makes $\lambda_t \geqslant c(\Lambda)$ an order-one seed?

$\lambda_t + m \neq \lambda_j + \lambda_{j'}$ for all $1 < j, j' < t$.

This is equivalent to λ_t being a strong generator.

That is, the order-one seeds of Λ are its strong generators.

order-0 seeds	$\longleftrightarrow$	right generators
order-1 seeds	$\longleftrightarrow$	strong generators

Example

$$\Lambda = \{ \overbrace{0}^{\lambda_0}, \overbrace{8}^{\lambda_1}, \overbrace{16}^{\lambda_2}, \overbrace{18}^{\lambda_3}, \overbrace{19}^{\lambda_4}, \overbrace{24}^{\lambda_5}, \overbrace{26}^{\lambda_6}, \overbrace{27}^{\lambda_7}, \underbrace{\overbrace{30}^{\lambda_8}}_{c}, \overbrace{31}^{\lambda_9}, \overbrace{32}^{\lambda_{10}}, \dots \}$$

Seeds of a numerical semigroup

Example

$$\Lambda = \{ \overbrace{0}^{\lambda_0}, \overbrace{8}^{\lambda_1}, \overbrace{16}^{\lambda_2}, \overbrace{18}^{\lambda_3}, \overbrace{19}^{\lambda_4}, \overbrace{24}^{\lambda_5}, \overbrace{26}^{\lambda_6}, \overbrace{27}^{\lambda_7}, \underbrace{\overbrace{30}^{\lambda_8}}_{c}, \overbrace{31}^{\lambda_9}, \overbrace{32}^{\lambda_{10}}, \dots \}$$

- $\lambda_9 = 31$ is an order-one seed because

  $\lambda_9 + \lambda_1 = 39 \notin \{\lambda_2, \dots, \lambda_8\} + \{\lambda_2, \dots, \lambda_8\} = \{32,34,35,36,37,38,40,\dots\}$

Seeds of a numerical semigroup

Example

$$\Lambda = \{ \overbrace{0}^{\lambda_0}, \overbrace{8}^{\lambda_1}, \overbrace{16}^{\lambda_2}, \overbrace{18}^{\lambda_3}, \overbrace{19}^{\lambda_4}, \overbrace{24}^{\lambda_5}, \overbrace{26}^{\lambda_6}, \overbrace{27}^{\lambda_7}, \underbrace{\overbrace{30}^{\lambda_8}}_{c}, \overbrace{31}^{\lambda_9}, \overbrace{32}^{\lambda_{10}}, \dots \}$$

- $\lambda_9 = 31$ is an order-one seed because

  $\lambda_9 + \lambda_1 = 39 \notin \{\lambda_2, \dots, \lambda_8\} + \{\lambda_2, \dots, \lambda_8\} = \{32,34,35,36,37,38,40,\dots\}$

- $\lambda_{10} = 32$ is not an order-one seed because

  $\lambda_{10} + \lambda_1 = 40 = 16 + 24 = \lambda_2 + \lambda_5$

The table of seeds of a semigroup

Lemma. There are no seeds of order $\geqslant k$.

The table of seeds of a semigroup

Lemma. There are no seeds of order $\geqslant k$.
Any order-i seed of Λ is at most $c + \lambda_{i+1} - \lambda_i - 1$.

The table of seeds of a semigroup

Lemma. There are no seeds of order $\geqslant k$.
Any order-i seed of Λ is at most $c + \underbrace{\lambda_{i+1} - \lambda_i}_{i\text{th jump}=:u_i} - 1$.

Lemma. There are no seeds of order $\geqslant k$.
Any order-i seed of Λ is at most $c + \lambda_{i+1} - \lambda_i - 1 = c + u_i - 1$.

Example. Table of seeds of

$$\Lambda = \{ \overbrace{0}^{\lambda_0}, \overbrace{8}^{\lambda_1}, \overbrace{16}^{\lambda_2}, \overbrace{18}^{\lambda_3}, \overbrace{19}^{\lambda_4}, \overbrace{24}^{\lambda_5}, \overbrace{26}^{\lambda_6}, \overbrace{27}^{\lambda_7}, \underbrace{\overbrace{30}^{\lambda_8}}_{c}, \overbrace{31}^{\lambda_9}, \overbrace{32}^{\lambda_{10}}, \dots \}$$

1	1	0	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1						
0							
1	0	0	0	0			
0	1						
1							
1	1	1					

The table of seeds of a semigroup

Lemma. There are no seeds of order $\geqslant k$.
Any order-i seed of Λ is at most $c + \lambda_{i+1} - \lambda_i - 1 = c + u_i - 1$.

Example. Table of seeds of

$$\Lambda = \{ \overbrace{0}^{\lambda_0}, \overbrace{8}^{\lambda_1}, \overbrace{16}^{\lambda_2}, \overbrace{18}^{\lambda_3}, \overbrace{19}^{\lambda_4}, \overbrace{24}^{\lambda_5}, \overbrace{26}^{\lambda_6}, \overbrace{27}^{\lambda_7}, \underbrace{\overbrace{30}^{\lambda_8}}_{c}, \overbrace{31}^{\lambda_9}, \overbrace{32}^{\lambda_{10}}, \dots \}$$

k (rows):

1	1	0	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1						
0							
1	0	0	0	0			
0	1						
1							
1	1	1					

Its rows are indexed by the possible seed orders, $0 \leqslant i \leqslant k-1$.

The table of seeds of a semigroup

Lemma. There are no seeds of order $\geqslant k$.
Any order-i seed of Λ is at most $c + \lambda_{i+1} - \lambda_i - 1 = c + u_i - 1$.

Example. Table of seeds of

$$\Lambda = \{ \overbrace{0}^{\lambda_0}, \overbrace{8}^{\lambda_1}, \overbrace{16}^{\lambda_2}, \overbrace{18}^{\lambda_3}, \overbrace{19}^{\lambda_4}, \overbrace{24}^{\lambda_5}, \overbrace{26}^{\lambda_6}, \overbrace{27}^{\lambda_7}, \underbrace{\overbrace{30}^{\lambda_8}}_{c}, \overbrace{31}^{\lambda_9}, \overbrace{32}^{\lambda_{10}}, \dots \}$$

$$\leftarrow u_i = \lambda_{i+1} - \lambda_i \rightarrow$$

1	1	0	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1						
0							
1	0	0	0	0			
0	1						
1							
1	1	1					

Its rows are indexed by the possible seed orders, $0 \leqslant i \leqslant k-1$.

The i-th row has u_i entries.

Lemma. There are no seeds of order $\geqslant k$.
Any order-i seed of Λ is at most $c+\lambda_{i+1}-\lambda_i-1 = c+u_i-1$.

Example. Table of seeds of

$$\Lambda = \{ \overbrace{0}^{\lambda_0}, \overbrace{8}^{\lambda_1}, \overbrace{16}^{\lambda_2}, \overbrace{18}^{\lambda_3}, \overbrace{19}^{\lambda_4}, \overbrace{24}^{\lambda_5}, \overbrace{26}^{\lambda_6}, \overbrace{27}^{\lambda_7}, \underbrace{\overbrace{30}^{\lambda_8}}_{c}, \overbrace{31}^{\lambda_9}, \overbrace{32}^{\lambda_{10}}, \dots \}$$

1	1	0	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1						
0							
1	0	0	0	0			
0	1						
1							
1	1	1					

Its rows are indexed by the possible seed orders, $0 \leqslant i \leqslant k-1$.

The i-th row has u_i entries.

The j-th entry in the i-th row is $\begin{cases} 1 & \text{if } c+j \text{ is an order-}i \text{ seed,} \\ 0 & \text{otherwise.} \end{cases}$

The table of seeds of a semigroup

Lemma. There are no seeds of order $\geqslant k$.
Any order-i seed of Λ is at most $c+\lambda_{i+1}-\lambda_i-1 = c+u_i-1$.

Example. Table of seeds of

$$\Lambda=\{\overbrace{0}^{\lambda_0},\overbrace{8}^{\lambda_1},\overbrace{16}^{\lambda_2},\overbrace{18}^{\lambda_3},\overbrace{19}^{\lambda_4},\overbrace{24}^{\lambda_5},\overbrace{26}^{\lambda_6},\overbrace{27}^{\lambda_7},\underbrace{\overbrace{30}^{\lambda_8}}_{c},\overbrace{31}^{\lambda_9},\overbrace{32}^{\lambda_{10}},\dots\}$$

1	1	0	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1						
0							
1	0	0	0	0			
0	1						
1							
1	1	1					

Its rows are indexed by the possible seed orders, $0 \leqslant i \leqslant k-1$.

The i-th row has u_i entries.

The j-th entry in the i-th row is $\begin{cases} 1 & \text{if } c+j \text{ is an order-}i\text{ seed,} \\ 0 & \text{otherwise.} \end{cases}$

The total number of entries in the table is c.

The table of seeds of a semigroup

Example. $\Lambda = \{ \overbrace{0}^{\lambda_0}, \overbrace{8}^{\lambda_1}, \overbrace{16}^{\lambda_2}, \overbrace{18}^{\lambda_3}, \overbrace{19}^{\lambda_4}, \overbrace{24}^{\lambda_5}, \overbrace{26}^{\lambda_6}, \overbrace{27}^{\lambda_7}, \underbrace{\overbrace{30}^{\lambda_8}}_{c}, \overbrace{31}^{\lambda_9}, \overbrace{32}^{\lambda_{10}}, \dots \}$

	c+0	c+1	c+2	c+3	...				
order 0	1	1	0	1	0	0	0	0	$u_0 = \lambda_1 - \lambda_0 = 8$
order 1	0	1	0	1	0	0	0	0	$u_1 = \lambda_2 - \lambda_1 = 8$
order 2	0	1							$u_2 = \lambda_3 - \lambda_2 = 2$
order 3	0								$u_3 = \lambda_4 - \lambda_3 = 1$
order 4	1	0	0	0	0				$u_4 = \lambda_5 - \lambda_4 = 5$
order 5	0	1							$u_5 = \lambda_6 - \lambda_5 = 2$
order 6	1								$u_6 = \lambda_7 - \lambda_6 = 1$
order 7	1	1	1						$u_7 = \lambda_8 - \lambda_7 = 3$

The table of seeds of a semigroup

Example. $\Lambda = \{ \overbrace{0}^{\lambda_0}, \overbrace{8}^{\lambda_1}, \overbrace{16}^{\lambda_2}, \overbrace{18}^{\lambda_3}, \overbrace{19}^{\lambda_4}, \overbrace{24}^{\lambda_5}, \overbrace{26}^{\lambda_6}, \overbrace{27}^{\lambda_7}, \underbrace{\overbrace{30}^{\lambda_8}}_{c}, \overbrace{31}^{\lambda_9}, \overbrace{32}^{\lambda_{10}}, \dots \}$

	c+0	c+1	c+2	c+3	...			
order 0	1	1	0	1	0	0	0	0
order 1	0	1	0	1	0	0	0	0
order 2	0	1						
order 3	0							
order 4	1	0	0	0	0			
order 5	0	1						
order 6	1							
order 7	1	1	1					

$u_0 = \lambda_1 - \lambda_0 = 8$

$u_1 = \lambda_2 - \lambda_1 = 8$

$u_2 = \lambda_3 - \lambda_2 = 2$

$u_3 = \lambda_4 - \lambda_3 = 1$

$u_4 = \lambda_5 - \lambda_4 = 5$

$u_5 = \lambda_6 - \lambda_5 = 2$

$u_6 = \lambda_7 - \lambda_6 = 1$

$u_7 = \lambda_8 - \lambda_7 = 3$

$\lambda_9 = c + 1$ **is** an order-one seed

The table of seeds of a semigroup

Example. $\Lambda = \{ \overbrace{0}^{\lambda_0}, \overbrace{8}^{\lambda_1}, \overbrace{16}^{\lambda_2}, \overbrace{18}^{\lambda_3}, \overbrace{19}^{\lambda_4}, \overbrace{24}^{\lambda_5}, \overbrace{26}^{\lambda_6}, \overbrace{27}^{\lambda_7}, \underbrace{\overbrace{30}^{\lambda_8}}_{c}, \overbrace{31}^{\lambda_9}, \overbrace{32}^{\lambda_{10}}, \dots \}$

	c+0	c+1	c+2	c+3	...			
order 0	1	1	0	1	0	0	0	0
order 1	0	1	0	1	0	0	0	0
order 2	0	1						
order 3	0							
order 4	1	0	0	0	0			
order 5	0	1						
order 6	1							
order 7	1	1	1					

$u_0 = \lambda_1 - \lambda_0 = 8$

$u_1 = \lambda_2 - \lambda_1 = 8$

$u_2 = \lambda_3 - \lambda_2 = 2$

$u_3 = \lambda_4 - \lambda_3 = 1$

$u_4 = \lambda_5 - \lambda_4 = 5$

$u_5 = \lambda_6 - \lambda_5 = 2$

$u_6 = \lambda_7 - \lambda_6 = 1$

$u_7 = \lambda_8 - \lambda_7 = 3$

$\lambda_9 = c + 1$ **is** an order-one seed

$\lambda_{10} = c + 2$ **is not** an order-one seed

Behavior of seeds along the semigroup tree

Suppose λ_s is an order-zero seed of Λ ($s \geqslant k$).

Behavior of seeds along the semigroup tree

Suppose λ_s is an order-zero seed of Λ ($s \geqslant k$).

$\tilde{\Lambda} := \Lambda \setminus \{\lambda_s\}$ is a semigroup of genus $g + 1$.

Behavior of seeds along the semigroup tree

Suppose λ_s is an order-zero seed of Λ ($s \geqslant k$).

$\tilde{\Lambda} := \Lambda \setminus \{\lambda_s\}$ is a semigroup of genus $g+1$.

Example. Descendants of

$$\Lambda = \{0, 8, 16, 18, 19, 24, 26, 27, \underbrace{\overbrace{\mathbf{30}}^{\lambda_8}}_{c}, \overbrace{\mathbf{31}}^{\lambda_9}, 32, \overbrace{\mathbf{33}}^{\lambda_{11}}, 34 \dots\}$$

Behavior of seeds along the semigroup tree

Suppose λ_s is an order-zero seed of Λ ($s \geqslant k$).

$\tilde{\Lambda} := \Lambda \setminus \{\lambda_s\}$ is a semigroup of genus $g + 1$.

Example. Descendants of

$$\Lambda = \{0, 8, 16, 18, 19, 24, 26, 27, \underbrace{\overbrace{\mathbf{30}}^{\lambda_8}}_{c}, \overbrace{\mathbf{31}}^{\lambda_9}, 32, \overbrace{\mathbf{33}}^{\lambda_{11}}, 34 \dots\}$$

$s = 8$

$\tilde{\Lambda} = \{0, 8, 16, 18, 19, 24, 26, 27, 31, 32, 33, 34 \dots\}$

$s = 9$

$\tilde{\Lambda} = \{0, 8, 16, 18, 19, 24, 26, 27, 30, 32, 33, 34 \dots\}$

$s = 11$

$\tilde{\Lambda} = \{0, 8, 16, 18, 19, 24, 26, 27, 30, 31, 32, 34 \dots\}$

Behavior of seeds along the semigroup tree

Suppose λ_s is an order-zero seed of Λ ($s \geqslant k$).

$\tilde{\Lambda} := \Lambda \setminus \{\lambda_s\}$ is a semigroup of genus $g + 1$.

Example. Descendants of

$$\Lambda = \{0, 8, 16, 18, 19, 24, 26, 27, \underbrace{\overbrace{\mathbf{30}}^{\lambda_8}}_{c}, \overbrace{\mathbf{31}}^{\lambda_9}, 32, \overbrace{\mathbf{33}}^{\lambda_{11}}, 34 \ldots\}$$

$s = 8$

$\tilde{\Lambda} = \{0, 8, 16, 18, 19, 24, 26, 27, 31, 32, 33, 34 \ldots\}$

$s = 9$

$\tilde{\Lambda} = \{0, 8, 16, 18, 19, 24, 26, 27, 30, 32, 33, 34 \ldots\}$

$s = 11$

$\tilde{\Lambda} = \{0, 8, 16, 18, 19, 24, 26, 27, 30, 31, 32, 34 \ldots\}$

Goal. Obtain the seeds of $\tilde{\Lambda}$ from those of Λ.

Old-order recycled seeds

Suppose $i < k$.

i is an old-order of $\tilde{\Lambda}$ since it was already an order of Λ.

Old-order recycled seeds

Suppose $i < k$.

i is an old-order of $\tilde{\Lambda}$ since it was already an order of Λ.

Lemma. Any order-i seed λ_t of Λ with $t > s$ is also an order-i seed of $\tilde{\Lambda}$.

Old-order recycled seeds

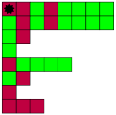
Suppose $i < k$.

i is an old-order of $\tilde{\Lambda}$ since it was already an order of Λ.

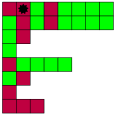
Lemma. Any order-i seed λ_t of Λ with $t > s$ is also an order-i seed of $\tilde{\Lambda}$.

Example. $\Lambda = \{0, 8, 16, 18, 19, 24, 26, 27, \underbrace{\overbrace{\mathbf{30}}^{\lambda_8}}_{c}, \overbrace{\mathbf{31}}^{\lambda_9}, 32, \overbrace{\mathbf{33}}^{\lambda_{11}}, 34 \ldots\}$

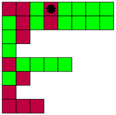
$s = 8$



$s = 9$



$s = 11$



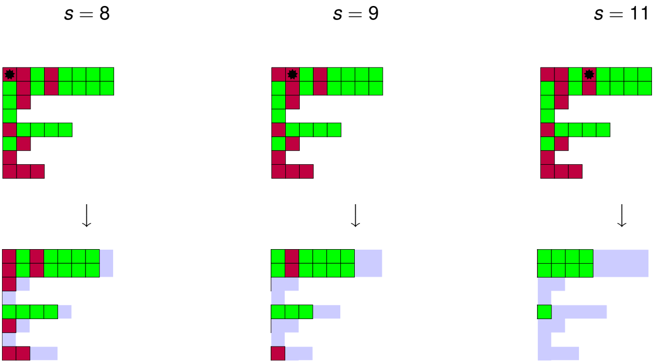
Old-order recycled seeds

Suppose $i < k$.

i is an old-order of $\tilde{\Lambda}$ since it was already an order of Λ.

Lemma. Any order-i seed λ_t of Λ with $t > s$ is also an order-i seed of $\tilde{\Lambda}$.

Example. $\Lambda = \{0, 8, 16, 18, 19, 24, 26, 27, \underbrace{\overbrace{\mathbf{30}}^{\lambda_8}}_{c}, \overbrace{\mathbf{31}}^{\lambda_9}, 32, \overbrace{\mathbf{33}}^{\lambda_{11}}, 34 \ldots\}$



Old-order recycled and new seeds

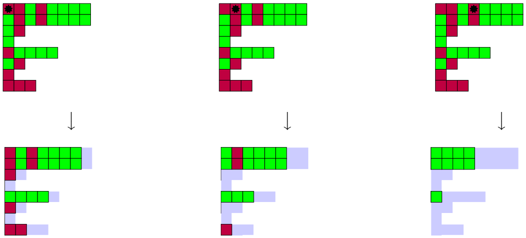
Suppose $i < k$.

Old-order recycled and new seeds

Suppose $i < k$.
Theorem 1. $\lambda_t > \lambda_s$ is an order-i seed of $\tilde{\Lambda}$ if and only if either

Suppose $i < k$.
Theorem 1. $\lambda_t > \lambda_s$ is an order-i seed of $\tilde{\Lambda}$ if and only if either

(1) λ_t is an order-i seed of Λ



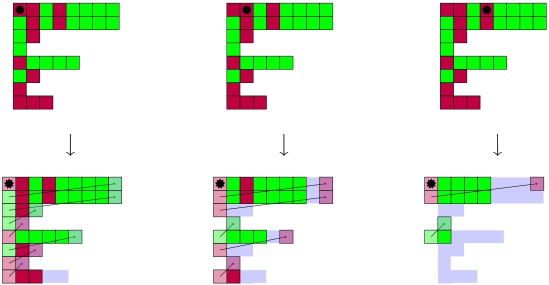
Old-order recycled and new seeds

Suppose $i < k$.

Theorem 1. $\lambda_t > \lambda_s$ is an order-i seed of $\tilde{\Lambda}$ if and only if either

(1) λ_t is an order-i seed of Λ

(2) $\lambda_t = \lambda_s + u_i$ if $i < k - 1$ and λ_s is an order-$(i+1)$ seed of Λ



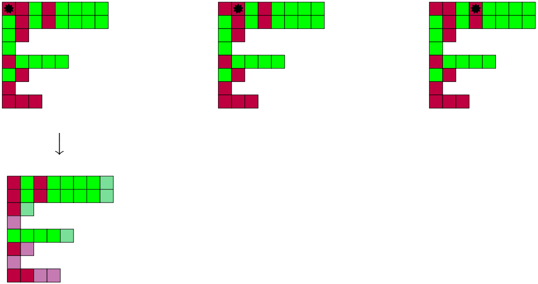
Suppose $i < k$.

Theorem 1. $\lambda_t > \lambda_s$ is an order-i seed of $\tilde{\Lambda}$ if and only if either

(1) λ_t is an order-i seed of Λ

(2) $\lambda_t = \lambda_s + u_i$ if $i < k - 1$ and λ_s is an order-$(i+1)$ seed of Λ

(3) $\begin{cases} \lambda_t = \lambda_s + u_i \\ \lambda_t = \lambda_s + u_i + 1 \end{cases}$ if $i = k - 1$ and $\lambda_s = c$



Suppose $i < k$.

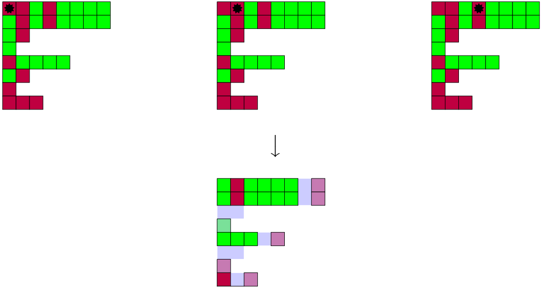
Theorem 1. $\lambda_t > \lambda_s$ is an order-i seed of $\tilde{\Lambda}$ if and only if either

(1) λ_t is an order-i seed of Λ

(2) $\lambda_t = \lambda_s + u_i$ if $i < k-1$ and λ_s is an order-$(i+1)$ seed of Λ

(3) $\begin{cases} \lambda_t = \lambda_s + u_i \\ \lambda_t = \lambda_s + u_i + 1 \end{cases}$ if $i = k-1$ and $\lambda_s = c$

(4) $\lambda_t = \lambda_s + u_i$ if $i = k-1$ and $\lambda_s = c+1$



New-order seeds

Suppose $i \geqslant k$.

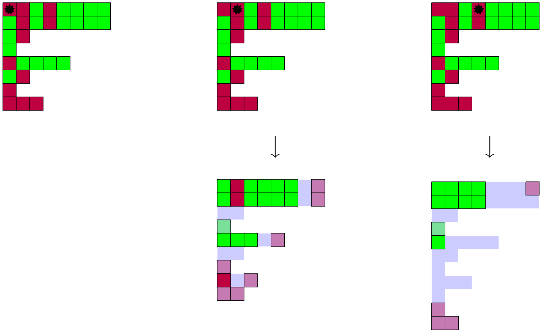
i is a new-order of $\tilde{\Lambda}$ since it was not an order of Λ.

Suppose $i \geqslant k$.

i is a new-order of $\tilde{\Lambda}$ since it was not an order of Λ.

Theorem 2.

- If $k \leqslant i < s - 2$, then $\tilde{\Lambda}$ has no order-i seeds.
- If $k \leqslant i = s - 2$, then the only order-i seed of $\tilde{\Lambda}$ is $\lambda_s + 1$.
- If $k \leqslant i = s - 1$, then the only order-i seeds of $\tilde{\Lambda}$ are $\lambda_s + 1$ and $\lambda_s + 2$.

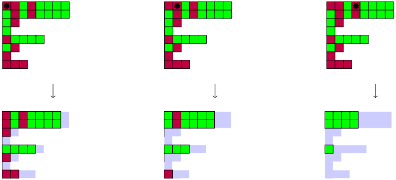


All seeds of the descendants

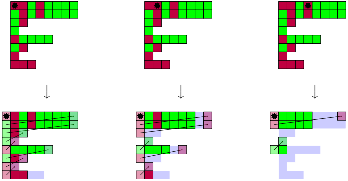
In any case, the last three positions in the table of seeds are 1.

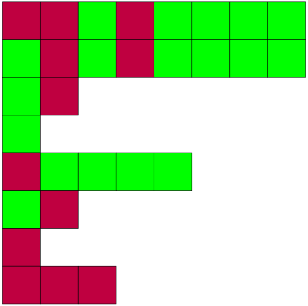
All the other 1's in the table correspond either to old-order recycled seeds or to old-order new seeds. That is, to the cases

(1) λ_t was already an order-i seed of Λ

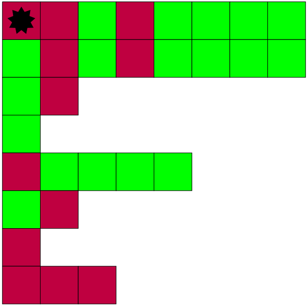


(2) $\lambda_t = \lambda_s + u_i$ if $i < k - 1$ and λ_s is an order-$(i+1)$ seed of Λ

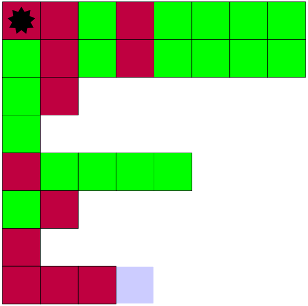




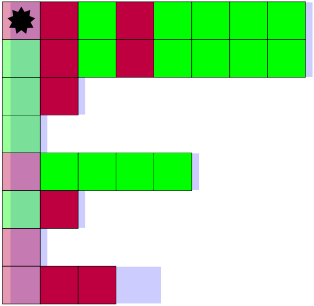
Consider the numerical semigroup $\{0, 8, 16, 18, 19, 24, 26, 27, 30, \ldots\}$



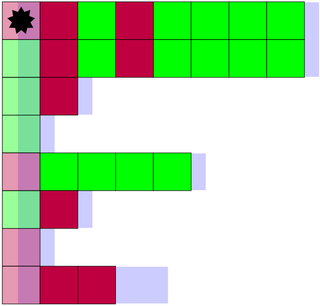
Suppose we want to take away the generator c+0=30+0=30.



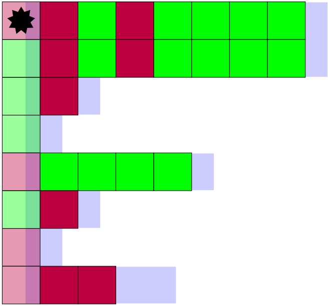
Draw the contour of the new table of seeds.



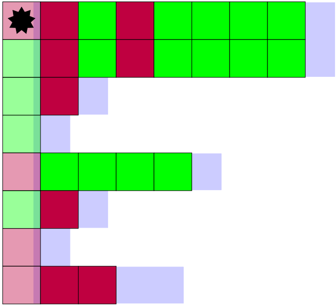
Move the old values to the left of the table
and fix the old-order recycled seeds.



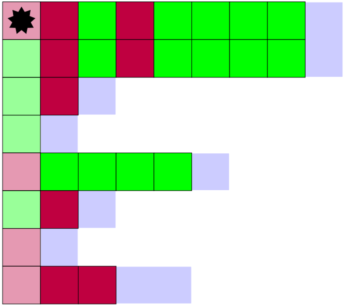
Move the old values to the left of the table
and fix the old-order recycled seeds.



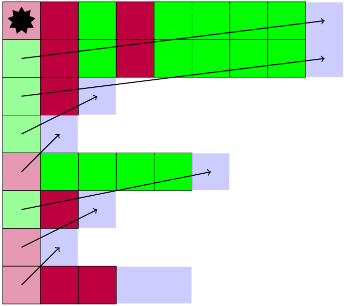
Move the old values to the left of the table
and fix the old-order recycled seeds.



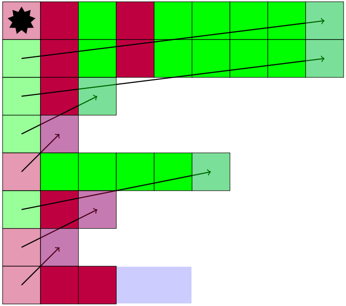
Move the old values to the left of the table
and fix the old-order recycled seeds.



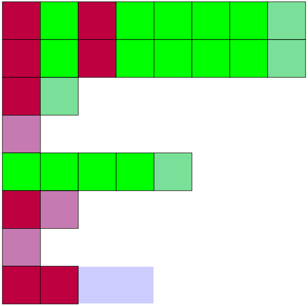
Move the old values to the left of the table
and fix the old-order recycled seeds.



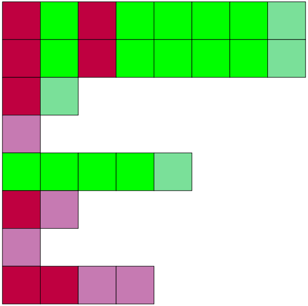
Obtain the old-order new seeds.



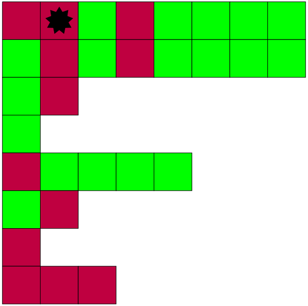
Obtain the old-order new seeds.



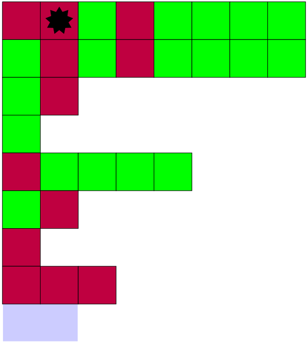
Obtain the old-order new seeds.



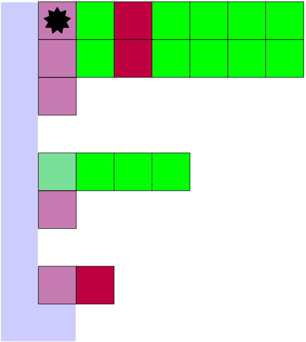
Set the last two elments in the table as old-order new seeds.



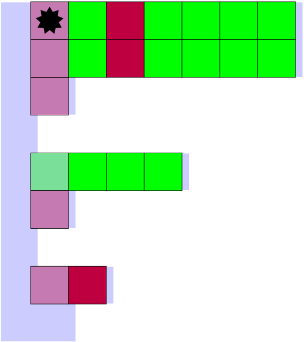
Suppose that now we want to take away the generator c+1=30+1=31.



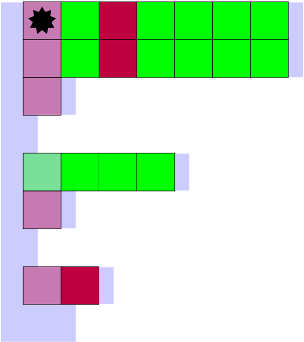
Draw the contour of the new table of seeds.



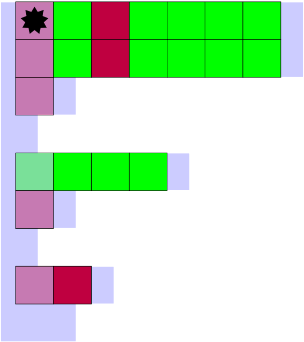
Discard the values corresponding to elements that are smaller than the new Frobenius number, keep shadowed the values corresponding to the new Frobenius number.



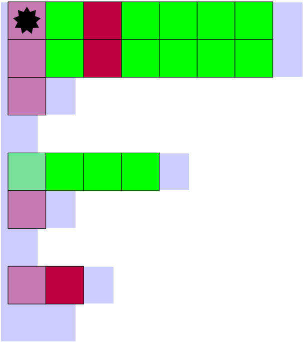
Move the old values to the left of the table
and fix the old-order recycled seeds.



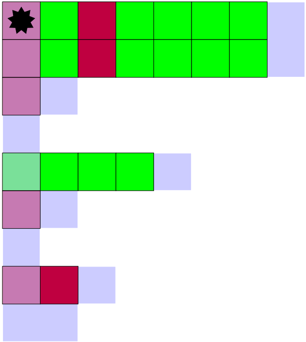
Move the old values to the left of the table
and fix the old-order recycled seeds.



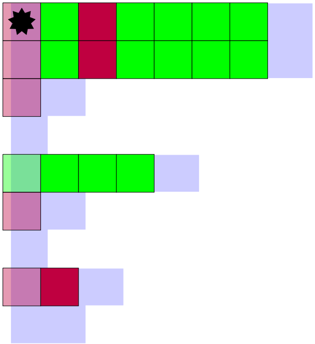
Move the old values to the left of the table
and fix the old-order recycled seeds.



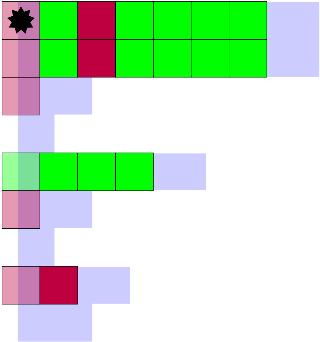
Move the old values to the left of the table
and fix the old-order recycled seeds.



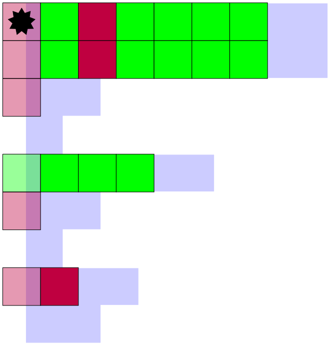
Move the old values to the left of the table
and fix the old-order recycled seeds.



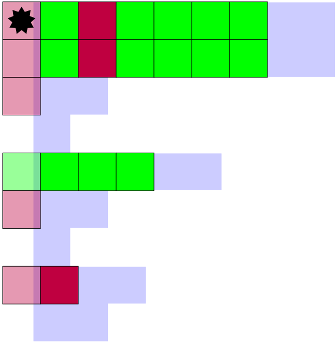
Move the old values to the left of the table
and fix the old-order recycled seeds.



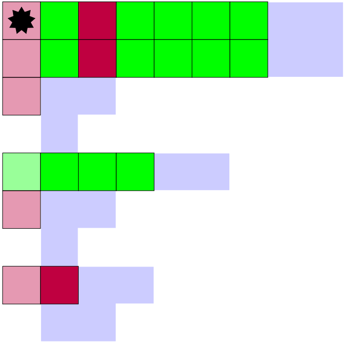
Move the old values to the left of the table
and fix the old-order recycled seeds.



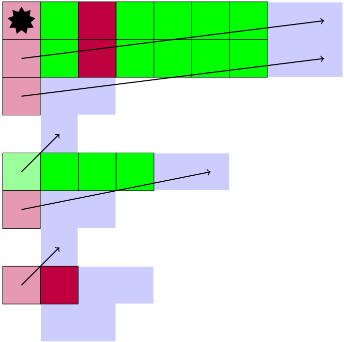
Move the old values to the left of the table
and fix the old-order recycled seeds.



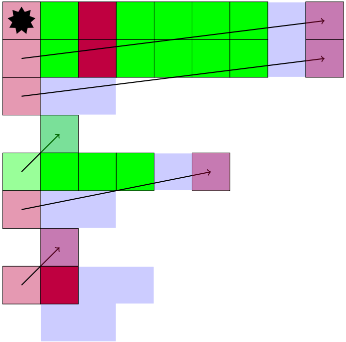
Move the old values to the left of the table
and fix the old-order recycled seeds.



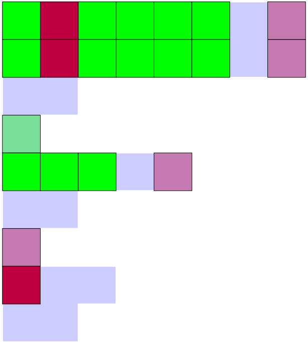
Move the old values to the left of the table
and fix the old-order recycled seeds.



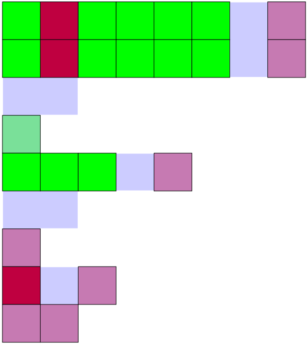
Obtain the old-order new seeds.



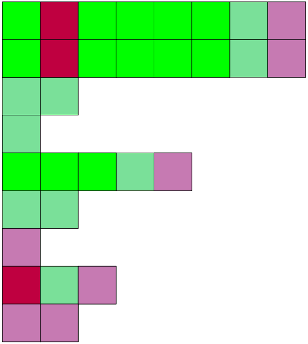
Obtain the old-order new seeds.



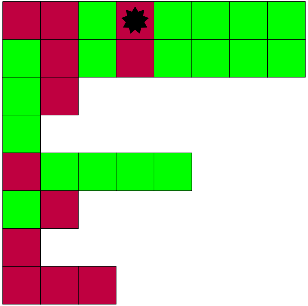
Obtain the old-order new seeds.



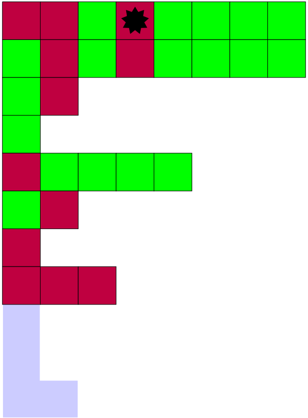
Set the last three elments in the table as
one old-order new seed and two new-order seeds.



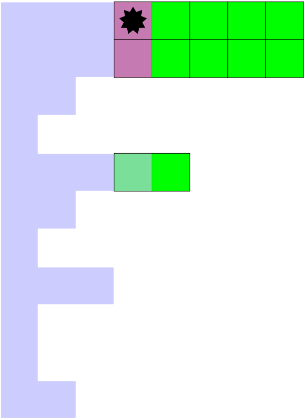
The remaining empty boxes are non-seeds.



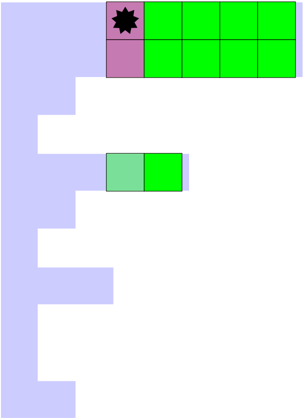
Suppose that now we want to take away the generator c+3=30+3=33.



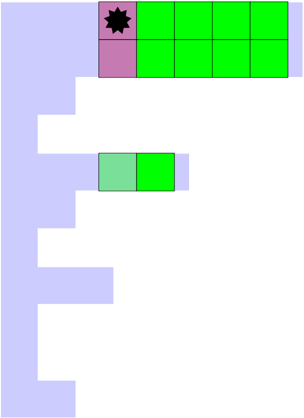
Draw the contour of the new table of seeds.



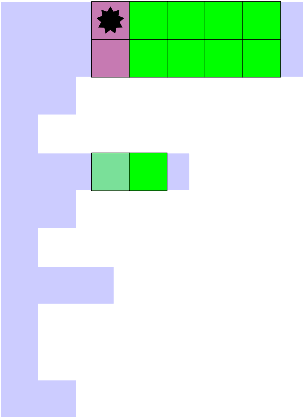
Discard the values corresponding to elements that are smaller than the new Frobenius number, keep shadowed the values corresponding to the new Frobenius num-



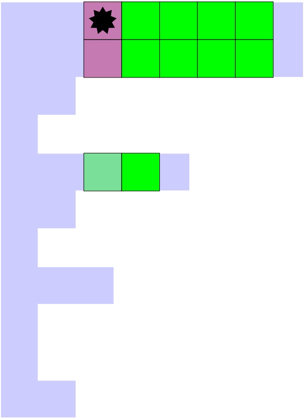
Move the old values to the left of the table
and fix the old-order recycled seeds.



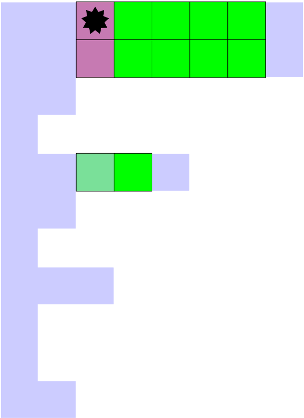
Move the old values to the left of the table
and fix the old-order recycled seeds.



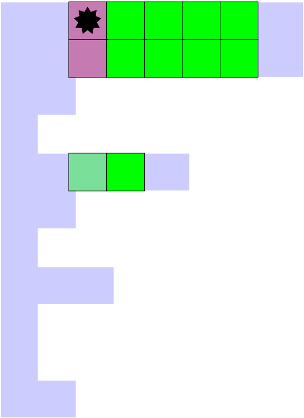
Move the old values to the left of the table
and fix the old-order recycled seeds.



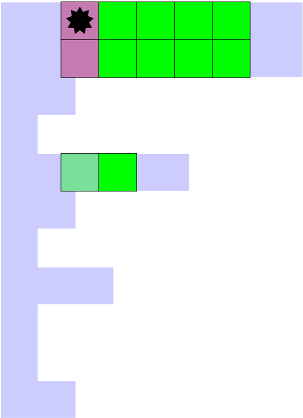
Move the old values to the left of the table
and fix the old-order recycled seeds.



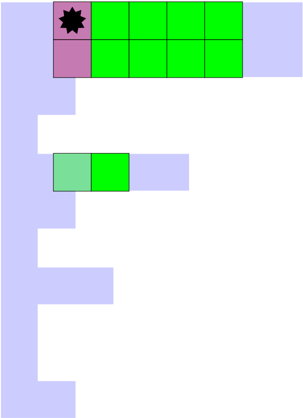
Move the old values to the left of the table
and fix the old-order recycled seeds.



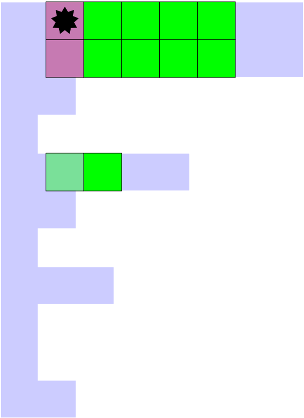
Move the old values to the left of the table
and fix the old-order recycled seeds.



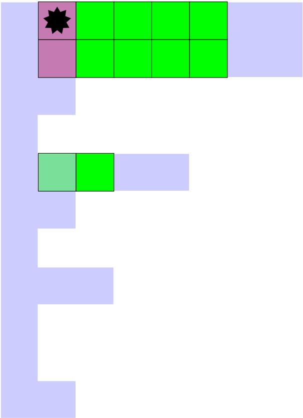
Move the old values to the left of the table
and fix the old-order recycled seeds.



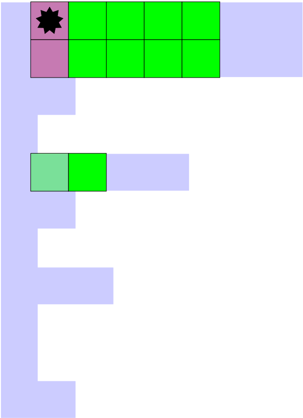
Move the old values to the left of the table
and fix the old-order recycled seeds.



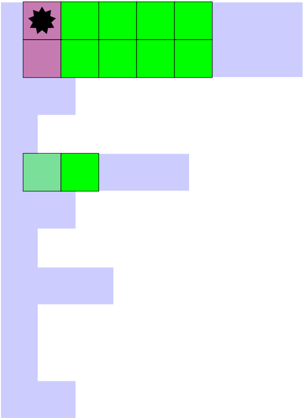
Move the old values to the left of the table
and fix the old-order recycled seeds.



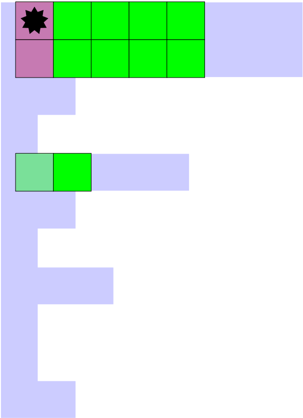
Move the old values to the left of the table
and fix the old-order recycled seeds.



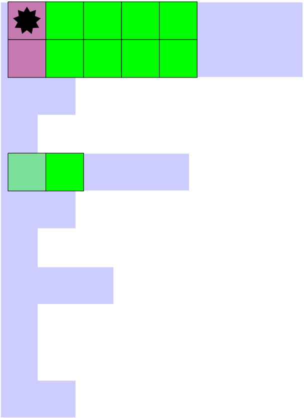
Move the old values to the left of the table
and fix the old-order recycled seeds.



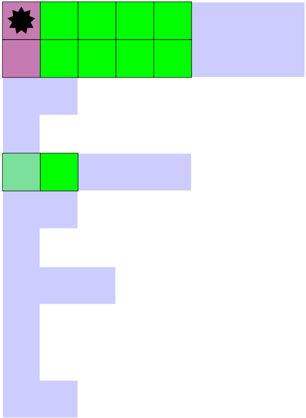
Move the old values to the left of the table
and fix the old-order recycled seeds.



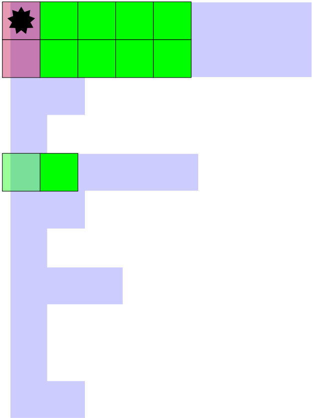
Move the old values to the left of the table
and fix the old-order recycled seeds.



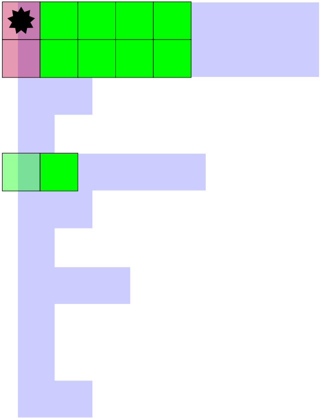
Move the old values to the left of the table
and fix the old-order recycled seeds.



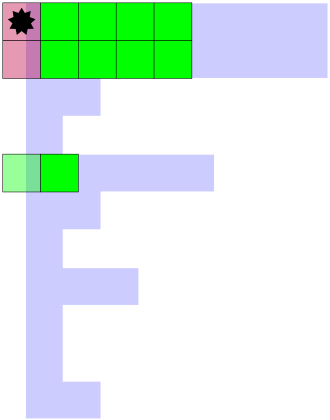
Move the old values to the left of the table
and fix the old-order recycled seeds.



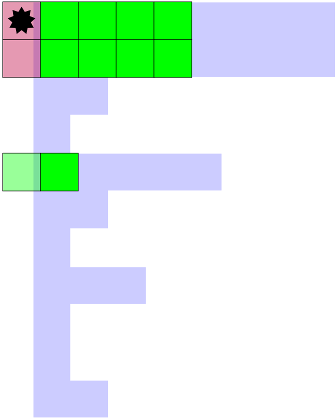
Move the old values to the left of the table
and fix the old-order recycled seeds.



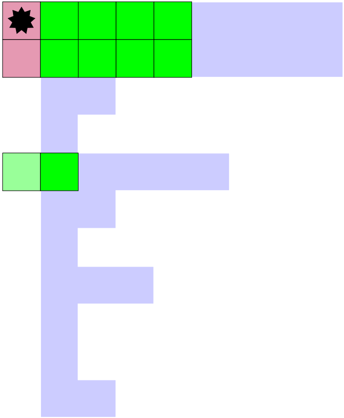
Move the old values to the left of the table
and fix the old-order recycled seeds.



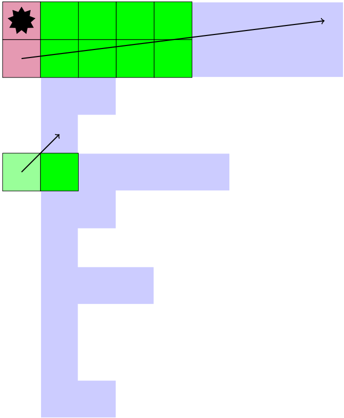
Move the old values to the left of the table
and fix the old-order recycled seeds.



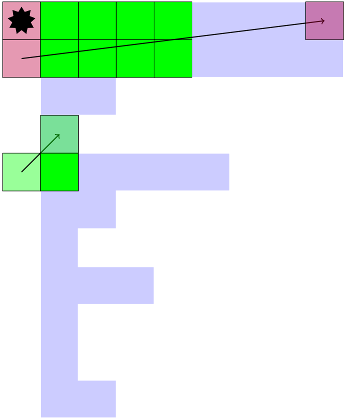
Move the old values to the left of the table
and fix the old-order recycled seeds.



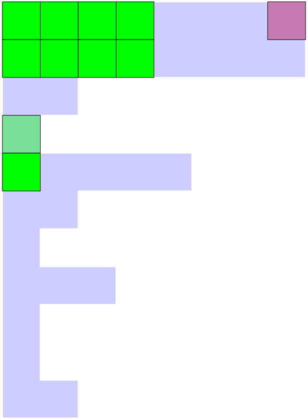
Move the old values to the left of the table
and fix the old-order recycled seeds.



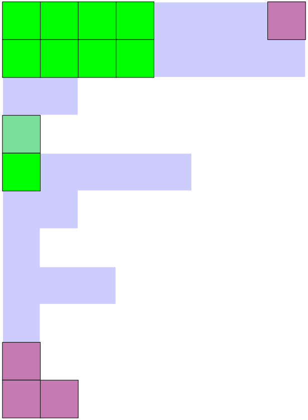
Obtain the old-order new seeds.



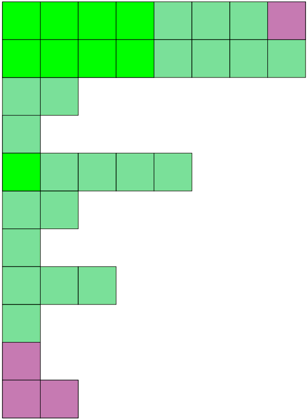
Obtain the old-order new seeds.



Obtain the old-order new seeds.



Set the last three elments in the table as new-order seeds.



The remaining empty boxes are non-seeds.

The string $G = G(\Lambda)$ stores the gaps of Λ.

Strings G, S

The string $G = G(\Lambda)$ stores the gaps of Λ.

The string $S = S(\Lambda)$ stores all the rows of the table of seeds of Λ merged in a single string.

The string $G = G(\Lambda)$ stores the gaps of Λ.

The string $S = S(\Lambda)$ stores all the rows of the table of seeds of Λ merged in a single string.

Example. For $\Lambda = \{0, 3, 6, \underbrace{8}_{c=\lambda_3}, 9, 10, \dots\}$,

Strings G, S

The string $G = G(\Lambda)$ stores the gaps of Λ.

The string $S = S(\Lambda)$ stores all the rows of the table of seeds of Λ merged in a single string.

Example. For $\Lambda = \{0, 3, 6, \underbrace{8}_{c=\lambda_3}, 9, 10, \dots\}$,

with table of seeds

1	0	1
1	0	1
1	1	

The string $G = G(\Lambda)$ stores the gaps of Λ.

The string $S = S(\Lambda)$ stores all the rows of the table of seeds of Λ merged in a single string.

Example. For $\Lambda = \{0, 3, 6, \underbrace{8}_{c=\lambda_3}, 9, 10, \dots\}$,

with table of seeds

1	0	1
1	0	1
1	1	

the strings G, S are

$$\begin{aligned} G(\Lambda) &= 11011010 \\ S(\Lambda) &= 10110111 \end{aligned}$$

Strings G, S

Lemma. Let $\Sigma(\Lambda)$ be the bitstream $\Sigma_0\Sigma_1\ldots\Sigma_{2c-1}$, with

$$\Sigma_i = \begin{cases} 0 & \text{if } i \in L(\Lambda) + L(\Lambda) \\ 1 & \text{otherwise} \end{cases}$$

then

$$\begin{aligned} G_0 G_1 \ldots G_{c-2} &= \Sigma_1 \ldots \Sigma_{c-1} \\ S_0 S_1 \ldots S_{c-1} &= \Sigma_c \Sigma_{c+1} \ldots \Sigma_{2c-1} \end{aligned}$$

Example. For $\Lambda = \{0, 3, 6, 8, 9, 10, \ldots\}$,

$$\begin{aligned} L(\Lambda) &= \{0, 3, 6\} \\ L(\Lambda) + L(\Lambda) &= \{0, 3, 6, 9, 12\} \\ \Sigma(\Lambda) &= 0110110110110111 \end{aligned}$$

$$\overbrace{0\,\underbrace{1\,1\,0\,1\,1\,0\,1}_{G(\Lambda)_{0,\ldots,(c-2)}}\,\underbrace{1\,0\,1\,1\,0\,1\,1\,1}_{S(\Lambda)}}^{\Sigma(\Lambda)}$$

Descending algorithm for G, S

It is useful to manipulate $G(\Lambda)$ and $S(\Lambda)$ as integers in binary form.

Descending algorithm for G, S

It is useful to manipulate $G(\Lambda)$ and $S(\Lambda)$ as integers in binary form.

Bitwise operations on binary strings:

- $\&$ *and*,
- $|$ inclusive *or*,
- $\gg$ *right shift* by a non-negative integer x (i.e., multiplying by 2^x),
- $\ll$ *left shift* by a non-negative integer x.

Descending algorithm for G, S

It is useful to manipulate $G(\Lambda)$ and $S(\Lambda)$ as integers in binary form.

Bitwise operations on binary strings:

- $\&$ *and*,
- $|$ inclusive *or*,
- $\gg$ *right shift* by a non-negative integer x (i.e., multiplying by 2^x),
- $\ll$ *left shift* by a non-negative integer x.

Given a bitstream B we use the bounded weight:

$$w_i^j(B)$$

defined as the number of 1's in B between positions i and j.

Let $\tilde{\Lambda} = \Lambda \backslash \{\lambda_s\}$.

Let $\tilde{\Lambda} = \Lambda \backslash \{\lambda_s\}$. Set $\Delta = s - k(\Lambda) = \lambda_s - c(\Lambda)$.

Descending algorithm for G, S

Let $\tilde{\Lambda} = \Lambda \backslash \{\lambda_s\}$. Set $\Delta = s - k(\Lambda) = \lambda_s - c(\Lambda)$.

Input: $c := c(\Lambda)$, $G := G(\Lambda)$, $S := S(\Lambda)$, Δ

Output: $c(\tilde{\Lambda})$, $G(\tilde{\Lambda})$, $S(\tilde{\Lambda})$

1. $\tilde{S} := S$
2. $\texttt{rake} := G$
3. **from** 1 **to** Δ **do**
4. $\quad \texttt{rake} := \texttt{rake} \gg 1$
5. $\quad \tilde{S} := \tilde{S} \,\&\, \texttt{rake}$

return $\tilde{c} := c + \Delta + 1$, $G \mid (1 \gg \tilde{c} - 2)$, $(\tilde{S} \ll \Delta + 1) \mid (1\,1\,1 \gg \tilde{c} - 3)$

The number of children, grandchildren, and great-grandchildren

Theorem. Suppose $k(\Lambda) \geqslant 3$.
Let $S = S(\Lambda)$, $m = m(\Lambda)$, $u = u_0(\Lambda)$, $v = u_1(\Lambda)$.

The number n_c of children, the number n_{gc} of grandchildren, and the number n_{ggc} of great-grandchildren of Λ is, respectively,

$$\begin{aligned}
n_c(\Lambda) &= w_0^{m-1}(S) \\
n_{gc}(\Lambda) &= \binom{n_c(\Lambda)}{2} + w_0^{u-1}(S \wedge (S \ll m)) \\
n_{ggc}(\Lambda) &= \binom{n_c(\Lambda)}{3} + w_0^{u-1}(S \wedge (S \ll m))(n_c(\Lambda) - 1) + w_0^{v-1}(S \wedge (S \ll u) \wedge (S \ll (u + m)))
\end{aligned}$$

The number of children, grandchildren, and great-grandchildren

Example. Consider the semigroup
$\Lambda = \{0, 8, 16, 18, 19, 24, 26, 27, 30, 31, 32, 33, 34 \dots\}$ with table of seeds

1	1	0	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1						
0							
1	0	0	0	0			
0	1						
1							
1	1	1					

In this case,

$$\begin{aligned} n_c &= w_0^{m-1}(S) = 3 \\ n_{gc} &= \binom{n_c(\Lambda)}{2} + w_0^{u-1}(S \wedge (S \ll m)) = \binom{3}{2} + 2 = 5 \end{aligned}$$

The number of children, grandchildren, and great-grandchildren

Example. Consider the semigroup
$\Lambda = \{0, 8, 16, 18, 19, 24, 26, 27, 30, 31, 32, 33, 34 \ldots\}$ with table of seeds

1	1	0	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1						
0							
1	0	0	0	0			
0	1						
1							
1	1	1					

In this case,

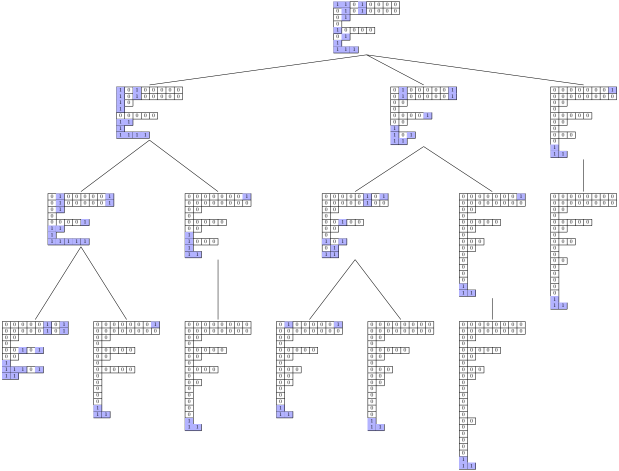
$$n_{ggc} = \binom{n_c(\Lambda)}{3} + w_0^{u-1}(S \wedge (S \ll m))(n_c(\Lambda) - 1) + w_0^{v-1}(S \wedge (S \ll u) \wedge (S \ll (u+m))),$$

where

$$\begin{aligned} \binom{n_c(\Lambda)}{3} &= \binom{3}{3} = 1 \\ w_0^{u-1}(S \wedge (S \ll m))(n_c(\Lambda) - 1) &= 2(3-1) = 4 \\ w_0^{v-1}(S \wedge (S \ll u) \wedge (S \ll (u+m))) &= 1 \end{aligned}$$

Hence, $n_{ggc} = 6$.

The number of children, grandchildren, and great-grandchildren



The number of children, grandchildren, and great-grandchildren

Theorem. Let $S = S(\Lambda)$, $m = m(\Lambda)$, $u = u_0(\Lambda)$, $v = u_1(\Lambda)$.

The number n_c of children, the number n_{gc} of grandchildren, and the number n_{ggc} of great-grandchildren of Λ is, respectively,

$$n_c(\Lambda) = \begin{cases} 1 & \text{if } k(\Lambda) = 0, \\ m & \text{if } k(\Lambda) = 1, \\ m-1 & \text{if } k(\Lambda) = 2, \\ w_0^{m-1}(S) & \text{if } k(\Lambda) \geqslant 3, \end{cases}$$

$$n_{gc}(\Lambda) = \begin{cases} 2 & \text{if } k(\Lambda) = 0, \\ \binom{m}{2} + 3 & \text{if } k(\Lambda) = 1, \\ \binom{n_c(\Lambda)}{2} + w_0^{u-1}(S \wedge (S \ll m)) & \text{if } k(\Lambda) > 1, \end{cases}$$

The number of children, grandchildren, and great-grandchildren

$$
n_{ggc}(\Lambda) = \begin{cases} 4 & \text{if } k(\Lambda) = 0, \\ \binom{m}{3} + 3m + 1 & \text{if } k(\Lambda) = 1, m = 2 \\ \binom{m}{3} + 3m + 3 & \text{if } k(\Lambda) = 1, m \geqslant 4 \\ \binom{m-1}{3} + (u - \delta_a)(m-2) + \delta_b + 2\delta_c + \delta_d & \text{if } k(\Lambda) = 2, \\ \binom{n_c(\Lambda)}{3} + w_0^{u-1}(S \wedge (S \ll m))(n_c(\Lambda) - 1) & \\ \qquad + w_0^{v-1}(S \wedge (S \ll u) \wedge (S \ll (u+m))) & \text{if } k(\Lambda) \geqslant 3, \end{cases}
$$

where

$$
\delta_a = \begin{cases} 1 & \text{if } m < 2u \\ 0 & \text{otherwise} \end{cases}
$$

$$
\delta_b = \begin{cases} 1 & \text{if } u = m \text{ or } 2u \neq m \\ 0 & \text{otherwise} \end{cases}
$$

$$
\delta_c = \begin{cases} 1 & \text{if } u < m - 1 \text{ and } 2u + 1 \neq m \\ 0 & \text{otherwise} \end{cases}
$$

$$
\delta_d = \begin{cases} 1 & \text{if } u = m - 1 \\ 0 & \text{otherwise} \end{cases}
$$

The seeds algorithm revisited

We implemented a faster version of the seeds algorithm using

- Bitwise operations on long integers representing G and S
- Explicit formulae for G and S for semigroups with $k(\Lambda) \leqslant 3$
- Formulae for n_{ggc}
- Parallelization and DFS as Fromentin-Hivert

The seeds algorithm revisited

We implemented a faster version of the seeds algorithm using

- Bitwise operations on long integers representing G and S
- Explicit formulae for G and S for semigroups with $k(\Lambda) \leqslant 3$
- Formulae for n_{ggc}
- Parallelization and DFS as Fromentin-Hivert

We can compare the number of seconds spent by FH, RGD, and our new implementation of the seeds algorithm when using parallelization with 12 workers.

genus	40	42	44	46	48	50	52	54	56	58	60
FH	1	2	7	19	53	145	372	978	2760	7398	21880
RGD	1	3	6	18	45	121	291	799	2101	5292	13785
Seeds	1	2	4	11	27	73	195	503	1329	3556	9459

The seeds algorithm revisited

We implemented a faster version of the seeds algorithm using

- Bitwise operations on long integers representing G and S
- Explicit formulae for G and S for semigroups with $k(\Lambda) \leqslant 3$
- Formulae for n_{ggc}
- Parallelization and DFS as Fromentin-Hivert

We can compare the number of seconds spent by FH, RGD, and our new implementation of the seeds algorithm when using parallelization with 12 workers.

genus	40	42	44	46	48	50	52	54	56	58	60
FH	1	2	7	19	53	145	372	978	2760	7398	21880
RGD	1	3	6	18	45	121	291	799	2101	5292	13785
Seeds	1	2	4	11	27	73	195	503	1329	3556	9459

Limitation! The main limitation of this implementation is that we can only compute for genera up to half the maximum size of native integers plus four (nowadays with 128 bits per integer we can compute up to n_{68}).

See: `https://github.com/mbrasamoros/seeds-algorithm`

Computation of Eliahou exceptions

Fix Λ and let
$k = \text{rank}(\Lambda)$,
$p =$ number of primitive elements of Λ,
$r =$ number of right generators of Λ.

Wilf's conjecture. $c(\Lambda) \leqslant kp$

Let $q = \left\lceil \frac{c(\Lambda)}{m(\Lambda)} \right\rceil$ and $\rho = qm(\Lambda) - c(\Lambda)$.

Eliahou number: $E(\Lambda) = k(p - r) - q(m - r) + \rho$

Eliahou exception: Λ such that $E(\Lambda) \geqslant 0$

If the Wilf conjecture fails for Λ then $E(\Lambda) < 0$ (S. Eliahou, 2018).

Eliahou exceptions are very rare.

Computation of Eliahou exceptions

Let $\langle a, b, c \rangle \mid_\kappa$ be the minimum semigroup containing a, b, c and all integers larger than or equal to κ.

The unique Eliahou exceptions of genus $g \leqslant 60$ are exactly (J. Fromentin)

$$\begin{aligned}
\varepsilon_1 &= \langle 14, 22, 23 \rangle \mid_{56}, \\
\varepsilon_2 &= \langle 16, 25, 26 \rangle \mid_{64}, \\
\varepsilon_3 &= \langle 17, 26, 28 \rangle \mid_{68}, \\
\varepsilon_4 &= \langle 17, 27, 28 \rangle \mid_{68}, \\
\varepsilon_5 &= \langle 18, 28, 29 \rangle \mid_{72},
\end{aligned}$$

The unique Eliahou exceptions with genus between 61 and 65 are exactly (B. - C. Marín-Rodríguez, 2021)

$$\begin{aligned}
\varepsilon_6 &= \langle 19, 29, 31 \rangle \mid_{76}, \\
\varepsilon_7 &= \langle 19, 30, 31 \rangle \mid_{76},
\end{aligned}$$

Now with the last implementation we could check the following:

- The unique Eliahou exceptions with $g \leqslant 66$ are $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5, \varepsilon_6, \varepsilon_7$.
- Hence, the Wilf conjecture holds for all semigroups of genus up to 66.

Computation of Eliahou exceptions

We found three further Eliahou exceptions of genus 67.

$$\begin{aligned} \varepsilon_8 &= \langle 20, 31, 32\rangle \,|_{80} \\ \varepsilon_9 &= \langle 20, 32, 33\rangle \,|_{80} \\ \varepsilon_{10} &= \langle 19, 26, 27\rangle \,|_{90} \end{aligned}$$

Computation of Eliahou exceptions

We found three further Eliahou exceptions of genus 67.

$$\begin{aligned} \varepsilon_8 &= \langle 20, 31, 32 \rangle \,|_{80} \\ \varepsilon_9 &= \langle 20, 32, 33 \rangle \,|_{80} \\ \varepsilon_{10} &= \langle 19, 26, 27 \rangle \,|_{90} \end{aligned}$$

We can not assert that those are the unique Eliahou exceptions of genus 67.

Computation of Eliahou exceptions

All semigroups ε_1, ε_2, ε_3, ε_4, ε_5, ε_6, ε_7, ε_8, ε_9 belong either to a family of Eliahou exceptions found by M. Delgado (2018) or to a family of Eliahou exceptions found by S. Eliahou and J. Fromentin (2019). Indeed,

$$\begin{aligned}
\varepsilon_1 &= EF(14,22,23) = D^{(0,0)}(4,0) \\
\varepsilon_2 &= EF(16,25,26) = D^{(0,1)}(4,0) \\
\varepsilon_3 &= EF(17,26,28) \\
\varepsilon_4 &= EF(17,27,28) \\
\varepsilon_5 &= EF(18,28,29) = D^{(0,2)}(4,0) \\
\varepsilon_6 &= EF(19,29,31) \\
\varepsilon_7 &= EF(19,30,31) \\
\varepsilon_8 &= EF(20,31,32) = D^{(0,3)}(4,0) \\
\varepsilon_9 &= EF(20,32,33)
\end{aligned}$$

Computation of Eliahou exceptions

The semigroup ε_{10} does not belong to the previous families.

This led Shalom Eliahou to describe a new family of Eliahou exceptions as follows:

$$BEF_t = \langle 2t+1, 3t-1, 3t \rangle \,|_{10t},$$

for $t \geqslant 9$.

Now,

$$\varepsilon_{10} = BEF_9.$$

Grazie tante!

Grazie tante!

E grazie Manuel e Shalom!

Grazie tante!

E grazie Manuel e Shalom!

E grazie Julio, César e Paul!