Cycle Decompositions of de Bruijn Graphs for Robot Identification and Tracking

Tony Grubman

Joint Supervisors: Y. Ahmet Şekercioğlu David R. Wood

Department of Electrical and Computer Systems Engineering
School of Mathematical Sciences

September 23, 2013

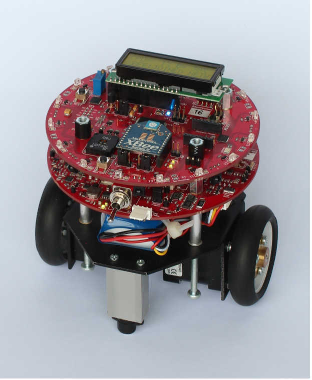
Outline

Outline

Outline

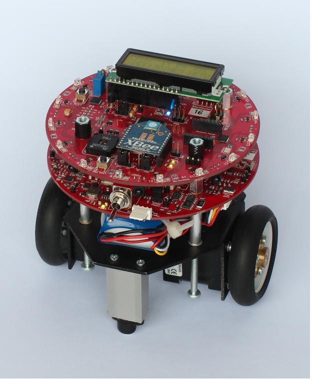
eBugs — colourful robots

- Wireless robot network research platform



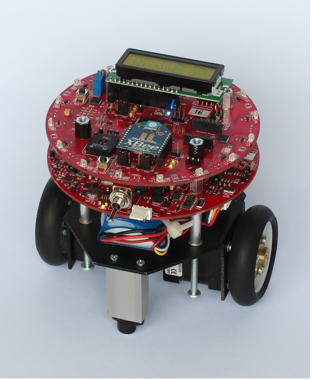
eBugs — colourful robots

- Wireless robot network research platform
  - ‘Swarm’ of up to 20 robots



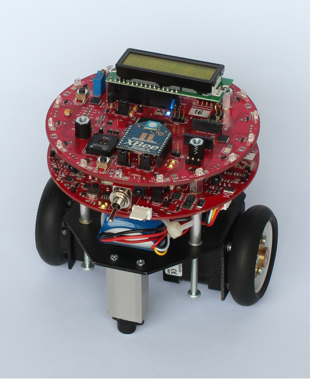
eBugs — colourful robots

- Wireless robot network research platform
  - ‘Swarm’ of up to 20 robots
- Mobile



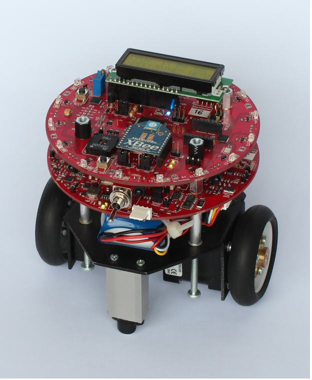
eBugs — colourful robots

- Wireless robot network research platform
  - 'Swarm' of up to 20 robots
- Mobile
  - Precision controlled stepper motors



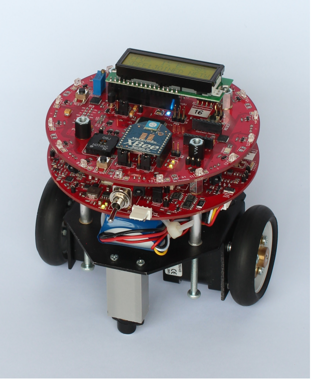
eBugs — colourful robots

- Wireless robot network research platform
  - 'Swarm' of up to 20 robots
- Mobile
  - Precision controlled stepper motors
- 16 multicolour LEDs (red, green and blue)



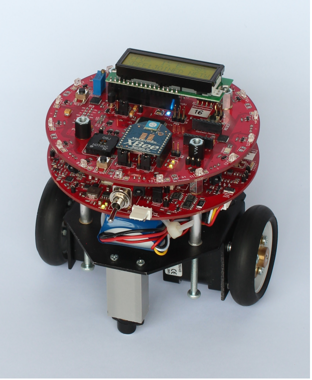
eBugs — colourful robots

- Wireless robot network research platform
  - 'Swarm' of up to 20 robots
- Mobile
  - Precision controlled stepper motors
- 16 multicolour LEDs (red, green and blue)
  - Can display a sequence of colours around its perimeter



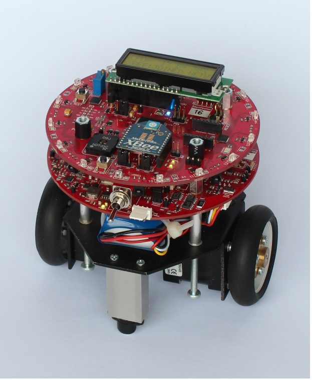
eBugs — colourful robots

- Wireless robot network research platform
  - ‘Swarm’ of up to 20 robots
- Mobile
  - Precision controlled stepper motors
- 16 multicolour LEDs (red, green and blue)
  - Can display a sequence of colours around its perimeter
- Expandable



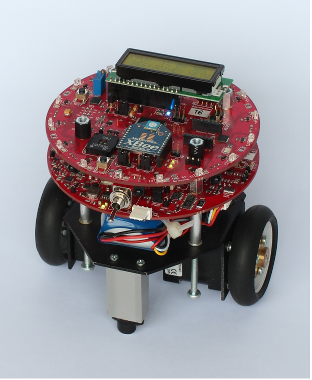
eBugs — colourful robots

- Wireless robot network research platform
  - 'Swarm' of up to 20 robots
- Mobile
  - Precision controlled stepper motors
- 16 multicolour LEDs (red, green and blue)
  - Can display a sequence of colours around its perimeter
- Expandable
  - Vision capabilities can be provided with a camera



eBugs — colourful robots

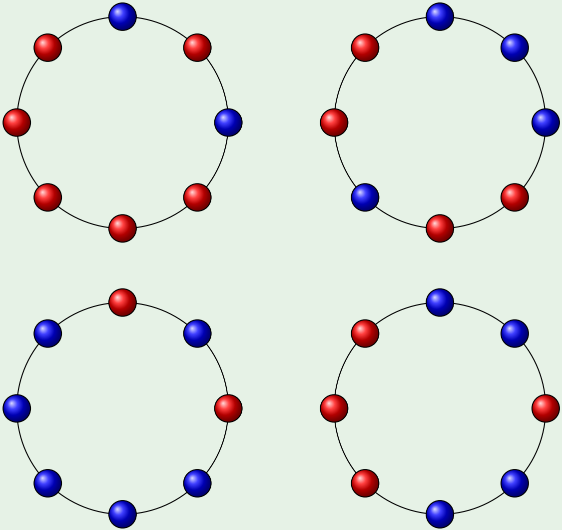
- Wireless robot network research platform
  - ‘Swarm’ of up to 20 robots
- Mobile
  - Precision controlled stepper motors
- 16 multicolour LEDs (red, green and blue)
  - Can display a sequence of colours around its perimeter
- Expandable
  - Vision capabilities can be provided with a camera



Problem

Can a sequence of colours be assigned to the LEDs of each eBug such that any observer (camera) can identify the eBug and its orientation?

Example (4 eBugs, 8 LEDs, 2 colours)



Preliminary bounds

Definition (eBug number)

Suppose every eBug has k LEDs, each of which can be illuminated in one of q colours, and that a camera can reliably detect ℓ adjacent LEDs. An assignment of colours to the LEDs of all eBugs is valid if the camera can distinguish each eBug in each of the k orientations.

Preliminary bounds

Definition (eBug number)

Suppose every eBug has k LEDs, each of which can be illuminated in one of q colours, and that a camera can reliably detect ℓ adjacent LEDs. An assignment of colours to the LEDs of all eBugs is valid if the camera can distinguish each eBug in each of the k orientations.
The eBug number $\mathcal{E}(q, k, \ell)$ is the maximum number of eBugs for which there exists a valid assignment of colours.

Preliminary bounds

Definition (eBug number)

Suppose every eBug has k LEDs, each of which can be illuminated in one of q colours, and that a camera can reliably detect ℓ adjacent LEDs. An assignment of colours to the LEDs of all eBugs is valid if the camera can distinguish each eBug in each of the k orientations.

The eBug number $\mathcal{E}(q,k,\ell)$ is the maximum number of eBugs for which there exists a valid assignment of colours.

- Upper bound: $\mathcal{E}(q,k,\ell) \leq \left\lfloor \frac{q^\ell}{k} \right\rfloor$

Preliminary bounds

Definition (eBug number)

Suppose every eBug has k LEDs, each of which can be illuminated in one of q colours, and that a camera can reliably detect ℓ adjacent LEDs. An assignment of colours to the LEDs of all eBugs is valid if the camera can distinguish each eBug in each of the k orientations.

The eBug number $\mathcal{E}(q,k,\ell)$ is the maximum number of eBugs for which there exists a valid assignment of colours.

- Upper bound: $\mathcal{E}(q,k,\ell) \leq \left\lfloor \frac{q^\ell}{k} \right\rfloor$
  - Each of the q^ℓ possible sequences cannot appear more than once

Preliminary bounds

Definition (eBug number)

Suppose every eBug has k LEDs, each of which can be illuminated in one of q colours, and that a camera can reliably detect ℓ adjacent LEDs. An assignment of colours to the LEDs of all eBugs is valid if the camera can distinguish each eBug in each of the k orientations.

The eBug number $\mathcal{E}(q,k,\ell)$ is the maximum number of eBugs for which there exists a valid assignment of colours.

- Upper bound: $\mathcal{E}(q,k,\ell) \leq \left\lfloor \frac{q^\ell}{k} \right\rfloor$
  - Each of the q^ℓ possible sequences cannot appear more than once
  - Each eBug will account for k of the sequences

Preliminary bounds

Definition (eBug number)

Suppose every eBug has k LEDs, each of which can be illuminated in one of q colours, and that a camera can reliably detect ℓ adjacent LEDs. An assignment of colours to the LEDs of all eBugs is valid if the camera can distinguish each eBug in each of the k orientations.
The eBug number $\mathcal{E}(q, k, \ell)$ is the maximum number of eBugs for which there exists a valid assignment of colours.

- Upper bound: $\mathcal{E}(q, k, \ell) \leq \left\lfloor \frac{q^\ell}{k} \right\rfloor$
  - Each of the q^ℓ possible sequences cannot appear more than once
  - Each eBug will account for k of the sequences
- Initial lower bound: Lovász local lemma gives $\mathcal{E} \geq \frac{q^\ell}{8\ell k}$

Preliminary bounds

Definition (eBug number)

Suppose every eBug has k LEDs, each of which can be illuminated in one of q colours, and that a camera can reliably detect ℓ adjacent LEDs. An assignment of colours to the LEDs of all eBugs is valid if the camera can distinguish each eBug in each of the k orientations.
The eBug number $\mathcal{E}(q, k, \ell)$ is the maximum number of eBugs for which there exists a valid assignment of colours.

- Upper bound: $\mathcal{E}(q, k, \ell) \leq \left\lfloor \frac{q^\ell}{k} \right\rfloor$
  - Each of the q^ℓ possible sequences cannot appear more than once
  - Each eBug will account for k of the sequences
- Initial lower bound: Lovász local lemma gives $\mathcal{E} \geq \dfrac{q^\ell}{8\ell k}$
- Computation shows that upper bound is achieved in small cases

Preliminary bounds

Definition (eBug number)

Suppose every eBug has k LEDs, each of which can be illuminated in one of q colours, and that a camera can reliably detect ℓ adjacent LEDs. An assignment of colours to the LEDs of all eBugs is valid if the camera can distinguish each eBug in each of the k orientations.
The eBug number $\mathcal{E}(q, k, \ell)$ is the maximum number of eBugs for which there exists a valid assignment of colours.

- Upper bound: $\mathcal{E}(q, k, \ell) \leq \left\lfloor \frac{q^\ell}{k} \right\rfloor$
  - Each of the q^ℓ possible sequences cannot appear more than once
  - Each eBug will account for k of the sequences
- Initial lower bound: Lovász local lemma gives $\mathcal{E} \geq \frac{q^\ell}{8\ell k}$
- Computation shows that upper bound is achieved in small cases
- Main problem — when is the upper bound achievable?

Demonstration

What is a de Bruijn graph?

Definition

The ℓ-th order q-ary de Bruijn graph $\mathrm{dB}(q,\ell)$ is the digraph (V,E), where $V=\mathbb{Z}_q^\ell$ and $E=\{(a_0a_1\ldots a_{\ell-1},a_1a_2\ldots a_\ell)\mid a_i\in\mathbb{Z}_q\}$.

What is a de Bruijn graph?

Definition

The ℓ-th order q-ary de Bruijn graph $\mathrm{dB}(q,\ell)$ is the digraph (V,E), where $V=\mathbb{Z}_q^\ell$ and $E=\{(a_0a_1\ldots a_{\ell-1},a_1a_2\ldots a_\ell)\mid a_i\in\mathbb{Z}_q\}$.

- Vertices are words of length ℓ over an alphabet of size q

What is a de Bruijn graph?

Definition

The ℓ-th order q-ary de Bruijn graph $\mathrm{dB}(q,\ell)$ is the digraph (V,E), where $V=\mathbb{Z}_q^\ell$ and $E=\{(a_0a_1\ldots a_{\ell-1},a_1a_2\ldots a_\ell)\mid a_i\in\mathbb{Z}_q\}$.

- Vertices are words of length ℓ over an alphabet of size q
- Edge from u to v if shifting u left and appending any letter gives v

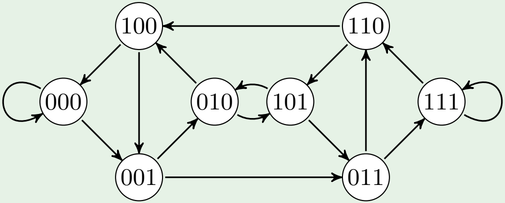
What is a de Bruijn graph?

Definition

The ℓ-th order q-ary de Bruijn graph $\mathrm{dB}(q, \ell)$ is the digraph (V, E), where $V = \mathbb{Z}_q^\ell$ and $E = \{(a_0 a_1 \dots a_{\ell-1}, a_1 a_2 \dots a_\ell) \mid a_i \in \mathbb{Z}_q\}$.

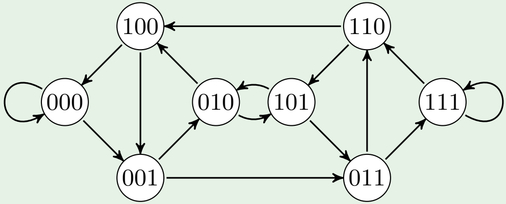
- Vertices are words of length ℓ over an alphabet of size q
- Edge from u to v if shifting u left and appending any letter gives v

Example ($\mathrm{dB}(2, 3)$)



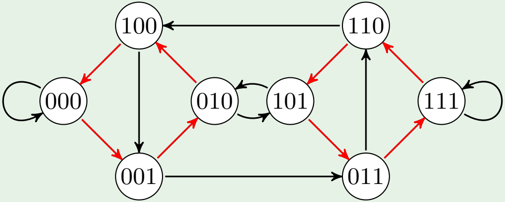
de Bruijn graphs and eBug numbers

Example $(\mathrm{dB}(2,3))$



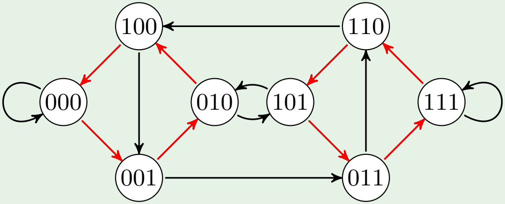
de Bruijn graphs and eBug numbers

Example ($\mathrm{dB}(2,3)$)



de Bruijn graphs and eBug numbers

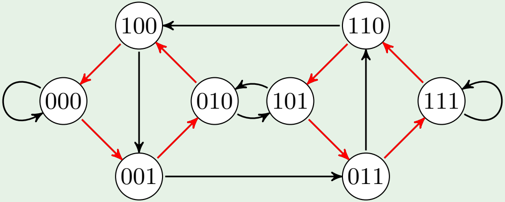
Example $(\mathrm{dB}(2,3))$



- Every vertex is a sequence of ℓ colours

de Bruijn graphs and eBug numbers

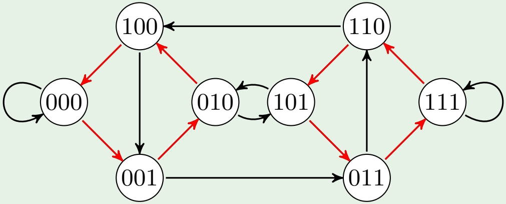
Example ($\mathrm{dB}(2,3)$)



- Every vertex is a sequence of ℓ colours
  - This represents the camera's view

de Bruijn graphs and eBug numbers

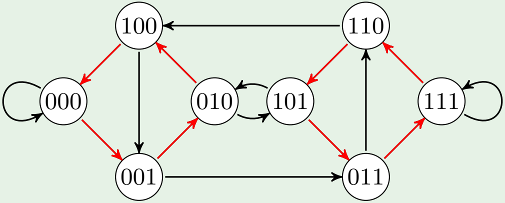
Example (dB(2, 3))



- Every vertex is a sequence of ℓ colours
  - This represents the camera's view
- Rotating the eBug corresponds to following an edge

de Bruijn graphs and eBug numbers

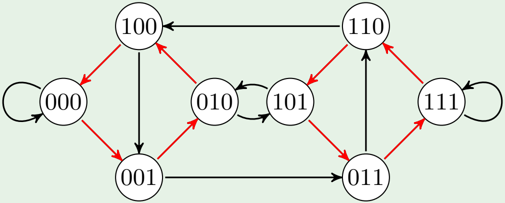
Example $(\mathrm{dB}(2,3))$



- Every vertex is a sequence of ℓ colours
  - This represents the camera's view
- Rotating the eBug corresponds to following an edge
- A cycle of length k represents the whole eBug

de Bruijn graphs and eBug numbers

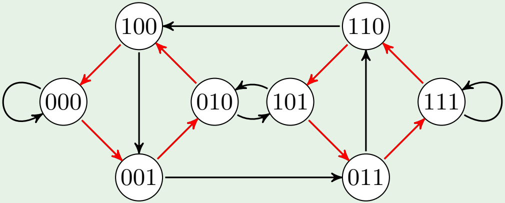
Example ($\mathrm{dB}(2,3)$)



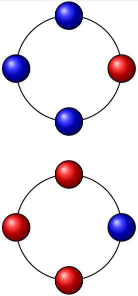
- Every vertex is a sequence of ℓ colours
  - This represents the camera's view
- Rotating the eBug corresponds to following an edge
- A cycle of length k represents the whole eBug
- $\mathcal{E}(q,k,\ell)$ is the maximum number of disjoint k-cycles in $\mathrm{dB}(q,\ell)$

de Bruijn graphs and eBug numbers

Example $(\mathrm{dB}(2,3))$



- Every vertex is a sequence of ℓ colours
  - This represents the camera's view
- Rotating the eBug corresponds to following an edge
- A cycle of length k represents the whole eBug
- $\mathcal{E}(q,k,\ell)$ is the maximum number of disjoint k-cycles in $\mathrm{dB}(q,\ell)$



Construction via line digraphs

Alternate construction

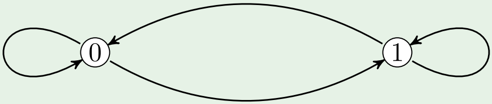
$\mathrm{dB}(q,1) = \overrightarrow{K_q}$

Construction via line digraphs

Alternate construction

$\mathrm{dB}(q,1) = \overrightarrow{K_q}$

Example ($q = 2$)

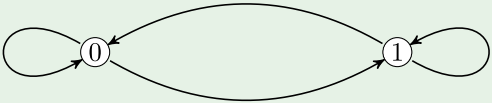


Construction via line digraphs

Alternate construction

$\mathrm{dB}(q,1) = \overrightarrow{K_q}; \qquad \mathrm{dB}(q,\ell+1) = L(\mathrm{dB}(q,\ell))$

Example ($q = 2$)

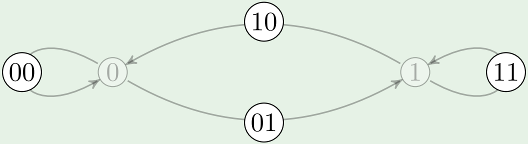


Construction via line digraphs

Alternate construction

$\mathrm{dB}(q,1) = \overrightarrow{K_q}; \qquad \mathrm{dB}(q,\ell+1) = L(\mathrm{dB}(q,\ell))$

Example ($q = 2$)

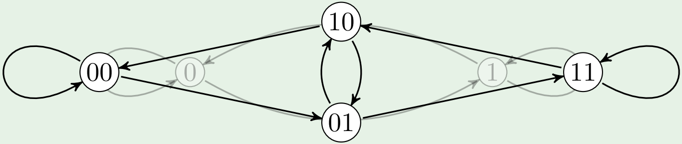


Construction via line digraphs

Alternate construction

$\mathrm{dB}(q,1) = \overrightarrow{K_q}; \qquad \mathrm{dB}(q,\ell+1) = L(\mathrm{dB}(q,\ell))$

Example ($q = 2$)

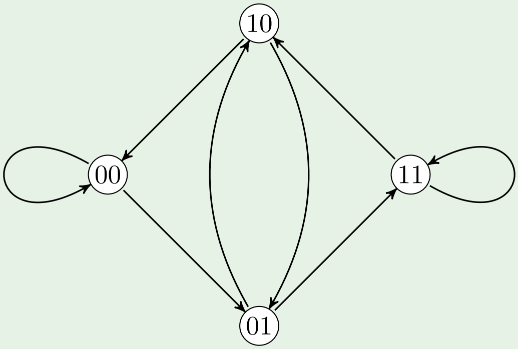


Construction via line digraphs

Alternate construction

$$\mathrm{dB}(q,1) = \overrightarrow{K_q}; \qquad \mathrm{dB}(q,\ell+1) = L(\mathrm{dB}(q,\ell))$$

Example ($q = 2$)

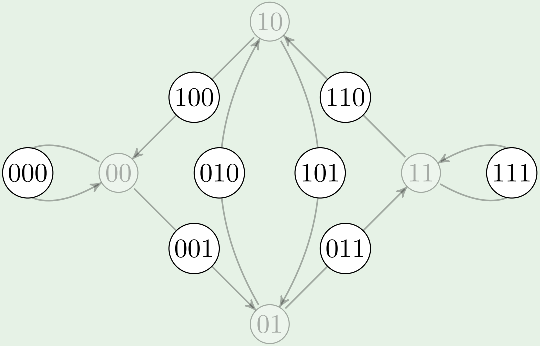


Construction via line digraphs

Alternate construction

$\mathrm{dB}(q,1) = \overrightarrow{K_q}; \qquad \mathrm{dB}(q,\ell+1) = L(\mathrm{dB}(q,\ell))$

Example ($q = 2$)

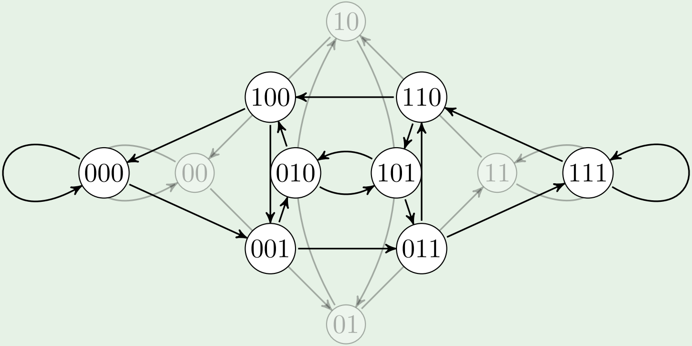


Construction via line digraphs

Alternate construction

$\mathrm{dB}(q,1) = \overrightarrow{K_q}; \qquad \mathrm{dB}(q,\ell+1) = L(\mathrm{dB}(q,\ell))$

Example ($q = 2$)

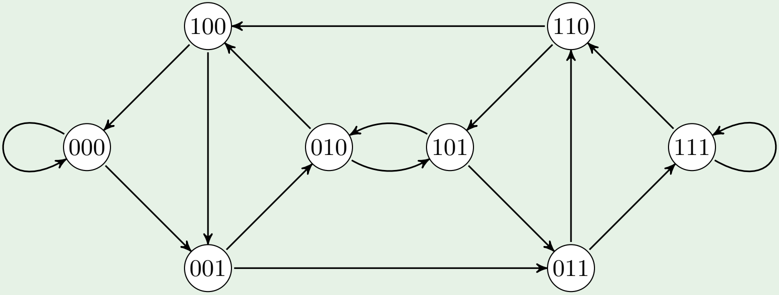


Construction via line digraphs

Alternate construction

$$\mathrm{dB}(q,1)=\overrightarrow{K_q}; \qquad \mathrm{dB}(q,\ell+1)=L(\mathrm{dB}(q,\ell))$$

Example ($q=2$)

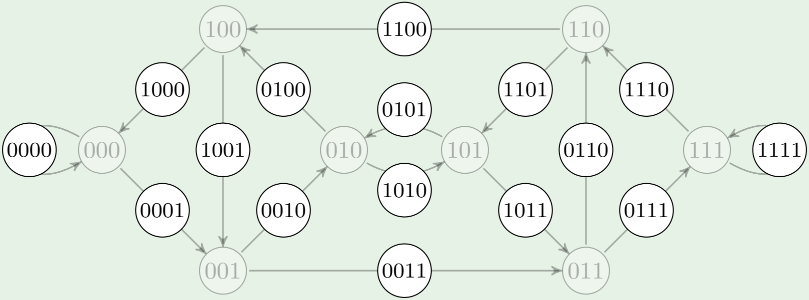


Construction via line digraphs

Alternate construction

$\mathrm{dB}(q,1) = \overrightarrow{K_q}$; $\quad \mathrm{dB}(q,\ell+1) = L(\mathrm{dB}(q,\ell))$

Example ($q = 2$)

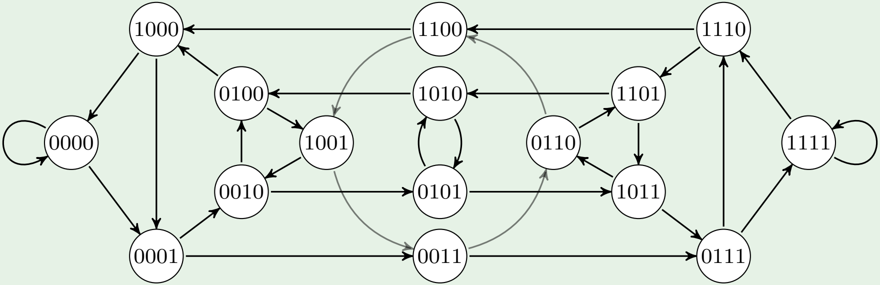


Construction via line digraphs

Alternate construction

$\mathrm{dB}(q,1) = \overrightarrow{K_q}; \qquad \mathrm{dB}(q,\ell+1) = L(\mathrm{dB}(q,\ell))$

Example ($q = 2$)

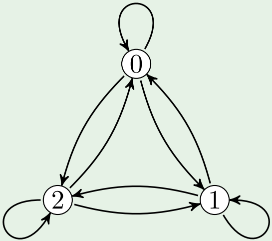


Construction via line digraphs

Alternate construction

$$\mathrm{dB}(q,1) = \overrightarrow{K_q}; \qquad \mathrm{dB}(q,\ell+1) = L(\mathrm{dB}(q,\ell))$$

Example ($q = 3$)

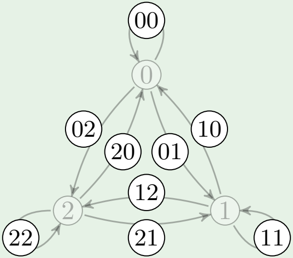


Construction via line digraphs

Alternate construction

$\mathrm{dB}(q,1) = \overrightarrow{K_q}; \qquad \mathrm{dB}(q,\ell+1) = L(\mathrm{dB}(q,\ell))$

Example ($q = 3$)

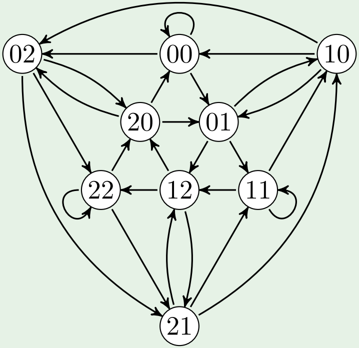


Construction via line digraphs

Alternate construction

$$\mathrm{dB}(q,1) = \overrightarrow{K_q}; \qquad \mathrm{dB}(q,\ell+1) = L(\mathrm{dB}(q,\ell))$$

Example ($q = 3$)



de Bruijn graphs are Hamiltonian

Theorem

If a digraph G is Eulerian, then the line digraph $L(G)$ is Hamiltonian.

de Bruijn graphs are Hamiltonian

Theorem

If a digraph G is Eulerian, then the line digraph $L(G)$ is Hamiltonian.

- An Eulerian circuit in G is equivalent to a Hamiltonian cycle in $L(G)$

de Bruijn graphs are Hamiltonian

Theorem

If a digraph G is Eulerian, then the line digraph $L(G)$ is Hamiltonian.

- An Eulerian circuit in G is equivalent to a Hamiltonian cycle in $L(G)$

Corollary

Every de Bruijn graph has a Hamiltonian cycle.

de Bruijn graphs are Hamiltonian

Theorem

If a digraph G is Eulerian, then the line digraph $L(G)$ is Hamiltonian.

- An Eulerian circuit in G is equivalent to a Hamiltonian cycle in $L(G)$

Corollary

Every de Bruijn graph has a Hamiltonian cycle.

Proof

de Bruijn graphs are Hamiltonian

Theorem

If a digraph G is Eulerian, then the line digraph $L(G)$ is Hamiltonian.

- An Eulerian circuit in G is equivalent to a Hamiltonian cycle in $L(G)$

Corollary

Every de Bruijn graph has a Hamiltonian cycle.

Proof

- Can assume $\ell \geq 2$ as $\mathrm{dB}(q, 1) = \overrightarrow{K_q}$ is Hamiltonian

de Bruijn graphs are Hamiltonian

Theorem

If a digraph G is Eulerian, then the line digraph $L(G)$ is Hamiltonian.

- An Eulerian circuit in G is equivalent to a Hamiltonian cycle in $L(G)$

Corollary

Every de Bruijn graph has a Hamiltonian cycle.

Proof

- Can assume $\ell \geq 2$ as $\mathrm{dB}(q, 1) = \overrightarrow{K_q}$ is Hamiltonian
- Every vertex in $\mathrm{dB}(q, \ell - 1)$ has in-degree q and out-degree q

de Bruijn graphs are Hamiltonian

Theorem

If a digraph G is Eulerian, then the line digraph $L(G)$ is Hamiltonian.

- An Eulerian circuit in G is equivalent to a Hamiltonian cycle in $L(G)$

Corollary

Every de Bruijn graph has a Hamiltonian cycle.

Proof

- Can assume $\ell \geq 2$ as $\mathrm{dB}(q, 1) = \overrightarrow{K_q}$ is Hamiltonian
- Every vertex in $\mathrm{dB}(q, \ell - 1)$ has in-degree q and out-degree q
  - $\mathrm{dB}(q, \ell - 1)$ is Eulerian

de Bruijn graphs are Hamiltonian

Theorem

If a digraph G is Eulerian, then the line digraph $L(G)$ is Hamiltonian.

- An Eulerian circuit in G is equivalent to a Hamiltonian cycle in $L(G)$

Corollary

Every de Bruijn graph has a Hamiltonian cycle.

Proof

- Can assume $\ell \geq 2$ as $\mathrm{dB}(q, 1) = \overrightarrow{K_q}$ is Hamiltonian
- Every vertex in $\mathrm{dB}(q, \ell - 1)$ has in-degree q and out-degree q
  - $\mathrm{dB}(q, \ell - 1)$ is Eulerian
  - $\mathrm{dB}(q, \ell)$ is Hamiltonian

□

Definition

A q-ary de Bruijn sequence of order ℓ is a Hamiltonian cycle in $\mathrm{dB}(q, \ell)$.

Definition

A q-ary de Bruijn sequence of order ℓ is a Hamiltonian cycle in $\mathrm{dB}(q, \ell)$.

Theorem

The number of Eulerian circuits in an Eulerian digraph G is

$$\tau(G) \prod_{v \in V(G)} (d^+(v) - 1)!$$

Definition

A q-ary de Bruijn sequence of order ℓ is a Hamiltonian cycle in $\mathrm{dB}(q, \ell)$.

Theorem

The number of Eulerian circuits in an Eulerian digraph G is

$$\tau(G) \prod_{v \in V(G)} (d^+(v) - 1)!$$

- $d^+(v)$ is the out-degree of vertex v

Definition

A q-ary de Bruijn sequence of order ℓ is a Hamiltonian cycle in $\mathrm{dB}(q, \ell)$.

Theorem

The number of Eulerian circuits in an Eulerian digraph G is

$$\tau(G) \prod_{v \in V(G)} (d^+(v) - 1)!$$

- $d^+(v)$ is the out-degree of vertex v
- $\tau(G)$ is the number of spanning arborescences rooted at some vertex

Definition

A q-ary de Bruijn sequence of order ℓ is a Hamiltonian cycle in $\mathrm{dB}(q, \ell)$.

Theorem

The number of Eulerian circuits in an Eulerian digraph G is

$$\tau(G) \prod_{v \in V(G)} (d^+(v) - 1)!$$

- $d^+(v)$ is the out-degree of vertex v
- $\tau(G)$ is the number of spanning arborescences rooted at some vertex
  - Does not depend on choice of root vertex

Definition

A q-ary de Bruijn sequence of order ℓ is a Hamiltonian cycle in $\mathrm{dB}(q, \ell)$.

Theorem

The number of Eulerian circuits in an Eulerian digraph G is

$$\tau(G) \prod_{v \in V(G)} (d^+(v) - 1)!$$

- $d^+(v)$ is the out-degree of vertex v
- $\tau(G)$ is the number of spanning arborescences rooted at some vertex
  - Does not depend on choice of root vertex

Corollary

There are exactly $\dfrac{(q!)^{q^{\ell-1}}}{q^\ell}$ *q-ary de Bruijn sequences of order ℓ.*

Galois LFSRs

- Let q be a prime power, and consider the Galois field $\mathrm{GF}(q)$

Galois LFSRs

- Let q be a prime power, and consider the Galois field $\mathrm{GF}(q)$
- Choose a degree ℓ primitive polynomial $p(x)$ over $\mathrm{GF}(q)$

Galois LFSRs

- Let q be a prime power, and consider the Galois field $\mathrm{GF}(q)$
- Choose a degree ℓ primitive polynomial $p(x)$ over $\mathrm{GF}(q)$
  - The quotient $F = \mathrm{GF}(q)[x]/\langle p(x)\rangle$ is generated by x

Galois LFSRs

- Let q be a prime power, and consider the Galois field $\mathrm{GF}(q)$
- Choose a degree ℓ primitive polynomial $p(x)$ over $\mathrm{GF}(q)$
  - The quotient $F = \mathrm{GF}(q)[x]/\langle p(x) \rangle$ is generated by x
- Repeatedly multiplying by x gives every non-zero element of F

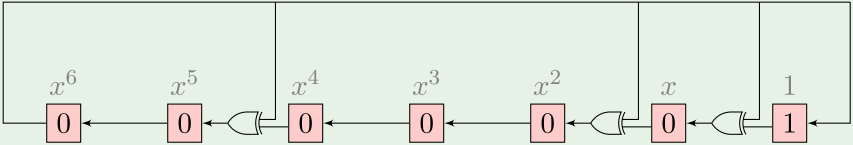
Galois LFSRs

- Let q be a prime power, and consider the Galois field $\mathrm{GF}(q)$
- Choose a degree ℓ primitive polynomial $p(x)$ over $\mathrm{GF}(q)$
  - The quotient $F = \mathrm{GF}(q)[x]/\langle p(x)\rangle$ is generated by x
- Repeatedly multiplying by x gives every non-zero element of F
- Easily implemented as a digital logic circuit

Galois LFSRs

- Let q be a prime power, and consider the Galois field $\mathrm{GF}(q)$
- Choose a degree ℓ primitive polynomial $p(x)$ over $\mathrm{GF}(q)$
  - The quotient $F = \mathrm{GF}(q)[x]/\langle p(x)\rangle$ is generated by x
- Repeatedly multiplying by x gives every non-zero element of F
- Easily implemented as a digital logic circuit

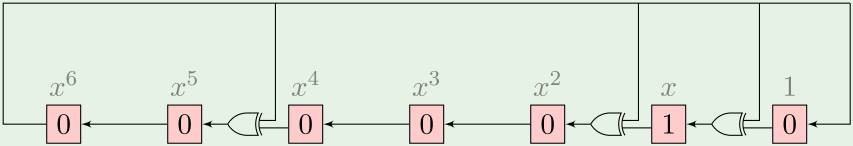
Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$)



Galois LFSRs

- Let q be a prime power, and consider the Galois field $\mathrm{GF}(q)$
- Choose a degree ℓ primitive polynomial $p(x)$ over $\mathrm{GF}(q)$
  - The quotient $F = \mathrm{GF}(q)[x]/\langle p(x) \rangle$ is generated by x
- Repeatedly multiplying by x gives every non-zero element of F
- Easily implemented as a digital logic circuit

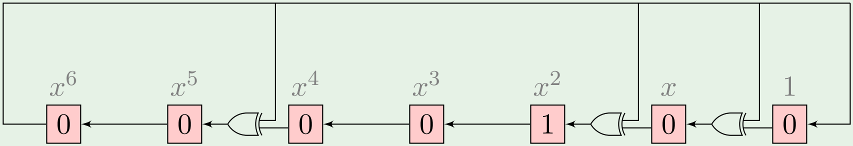
Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$)



Galois LFSRs

- Let q be a prime power, and consider the Galois field $\mathrm{GF}(q)$
- Choose a degree ℓ primitive polynomial $p(x)$ over $\mathrm{GF}(q)$
  - The quotient $F = \mathrm{GF}(q)[x]/\langle p(x)\rangle$ is generated by x
- Repeatedly multiplying by x gives every non-zero element of F
- Easily implemented as a digital logic circuit

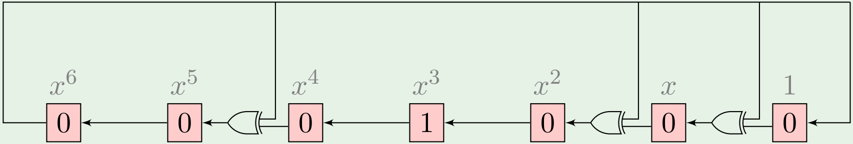
Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$)



Galois LFSRs

- Let q be a prime power, and consider the Galois field $\mathrm{GF}(q)$
- Choose a degree ℓ primitive polynomial $p(x)$ over $\mathrm{GF}(q)$
  - The quotient $F = \mathrm{GF}(q)[x]/\langle p(x) \rangle$ is generated by x
- Repeatedly multiplying by x gives every non-zero element of F
- Easily implemented as a digital logic circuit

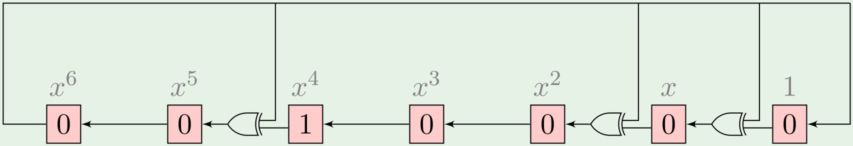
Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$)



Galois LFSRs

- Let q be a prime power, and consider the Galois field $\mathrm{GF}(q)$
- Choose a degree ℓ primitive polynomial $p(x)$ over $\mathrm{GF}(q)$
  - The quotient $F = \mathrm{GF}(q)[x]/\langle p(x) \rangle$ is generated by x
- Repeatedly multiplying by x gives every non-zero element of F
- Easily implemented as a digital logic circuit

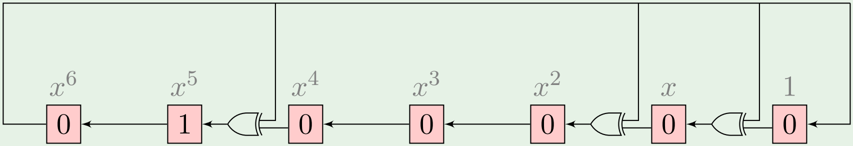
Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$)



Galois LFSRs

- Let q be a prime power, and consider the Galois field $\mathrm{GF}(q)$
- Choose a degree ℓ primitive polynomial $p(x)$ over $\mathrm{GF}(q)$
  - The quotient $F = \mathrm{GF}(q)[x]/\langle p(x) \rangle$ is generated by x
- Repeatedly multiplying by x gives every non-zero element of F
- Easily implemented as a digital logic circuit

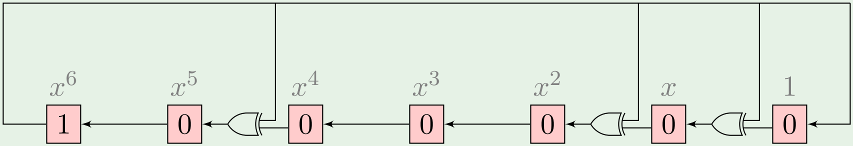
Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$)



Galois LFSRs

- Let q be a prime power, and consider the Galois field $\mathrm{GF}(q)$
- Choose a degree ℓ primitive polynomial $p(x)$ over $\mathrm{GF}(q)$
  - The quotient $F = \mathrm{GF}(q)[x]/\langle p(x)\rangle$ is generated by x
- Repeatedly multiplying by x gives every non-zero element of F
- Easily implemented as a digital logic circuit

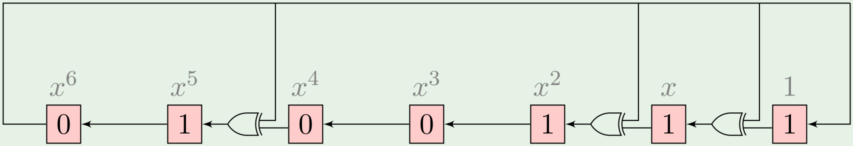
Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$)



Galois LFSRs

- Let q be a prime power, and consider the Galois field $\mathrm{GF}(q)$
- Choose a degree ℓ primitive polynomial $p(x)$ over $\mathrm{GF}(q)$
  - The quotient $F = \mathrm{GF}(q)[x]/\langle p(x)\rangle$ is generated by x
- Repeatedly multiplying by x gives every non-zero element of F
- Easily implemented as a digital logic circuit

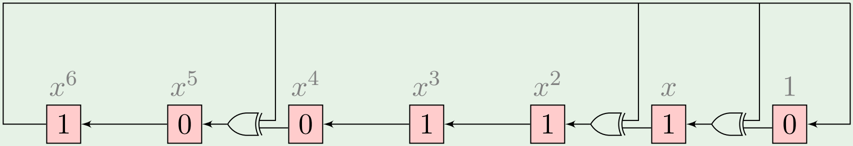
Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$)



Galois LFSRs

- Let q be a prime power, and consider the Galois field $\mathrm{GF}(q)$
- Choose a degree ℓ primitive polynomial $p(x)$ over $\mathrm{GF}(q)$
  - The quotient $F = \mathrm{GF}(q)[x]/\langle p(x)\rangle$ is generated by x
- Repeatedly multiplying by x gives every non-zero element of F
- Easily implemented as a digital logic circuit

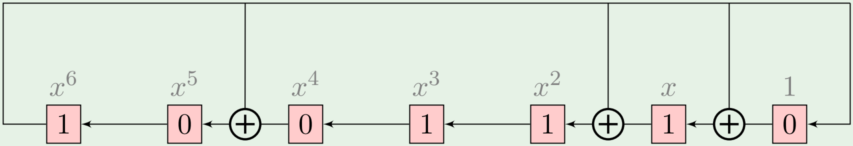
Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$)



Galois LFSRs

- Let q be a prime power, and consider the Galois field $\mathrm{GF}(q)$
- Choose a degree ℓ primitive polynomial $p(x)$ over $\mathrm{GF}(q)$
  - The quotient $F = \mathrm{GF}(q)[x]/\langle p(x)\rangle$ is generated by x
- Repeatedly multiplying by x gives every non-zero element of F
- Easily implemented as a digital logic circuit

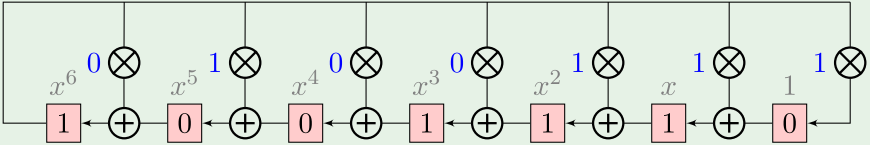
Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$)



Galois LFSRs

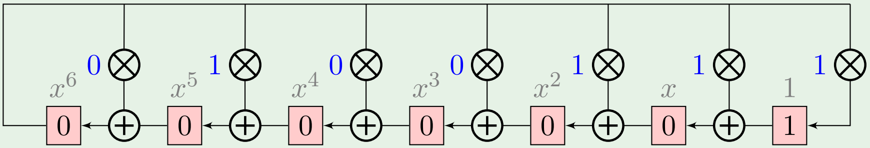
- Let q be a prime power, and consider the Galois field $\mathrm{GF}(q)$
- Choose a degree ℓ primitive polynomial $p(x)$ over $\mathrm{GF}(q)$
  - The quotient $F = \mathrm{GF}(q)[x]/\langle p(x)\rangle$ is generated by x
- Repeatedly multiplying by x gives every non-zero element of F
- Easily implemented as a digital logic circuit

Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$)



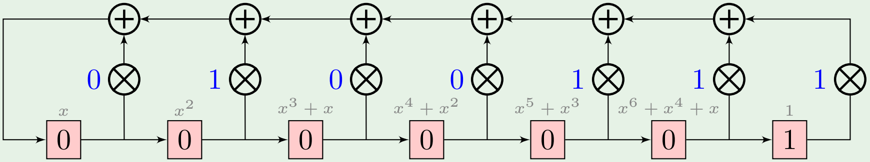
Fibonacci LFSRs

Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$) — Galois



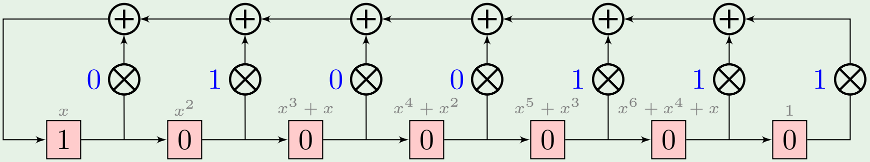
Fibonacci LFSRs

Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$) — Fibonacci



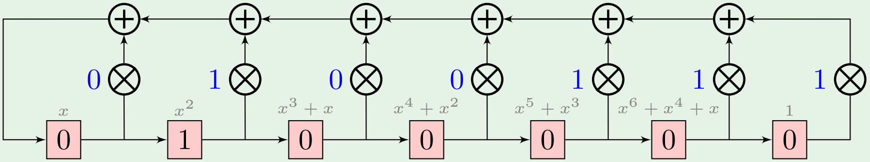
Fibonacci LFSRs

Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$) — Fibonacci



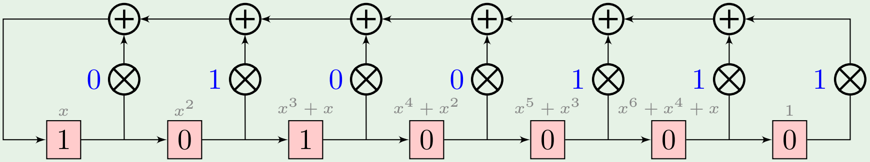
Fibonacci LFSRs

Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$) — Fibonacci



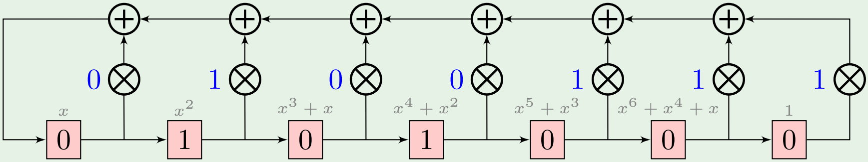
Fibonacci LFSRs

Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$) — Fibonacci



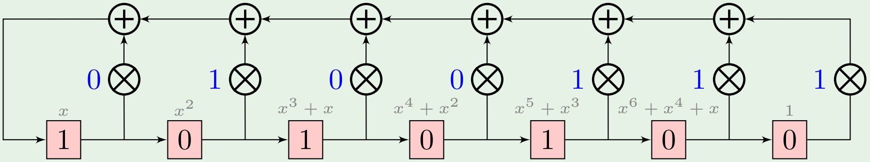
Fibonacci LFSRs

Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$) — Fibonacci



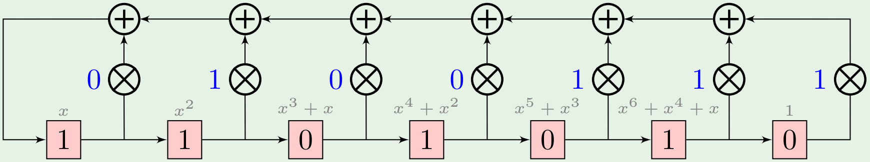
Fibonacci LFSRs

Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$) — Fibonacci



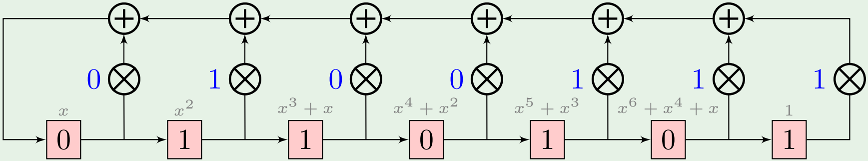
Fibonacci LFSRs

Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$) — Fibonacci



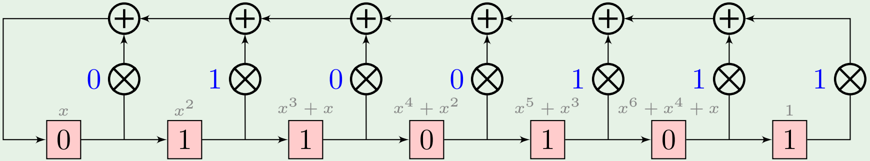
Fibonacci LFSRs

Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$) — Fibonacci



Fibonacci LFSRs

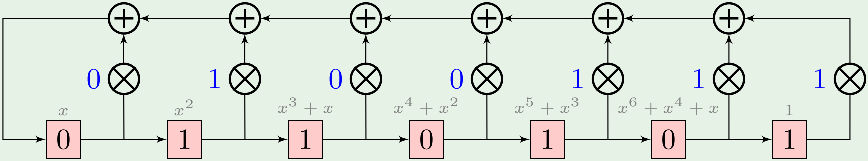
Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$) — Fibonacci



- Different logic configuration

Fibonacci LFSRs

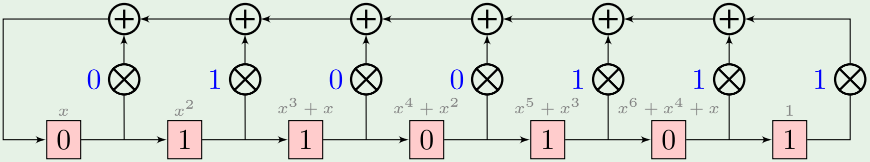
Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$) — Fibonacci



- Different logic configuration
  - Additive feedback is many-to-one, instead of one-to-many

Fibonacci LFSRs

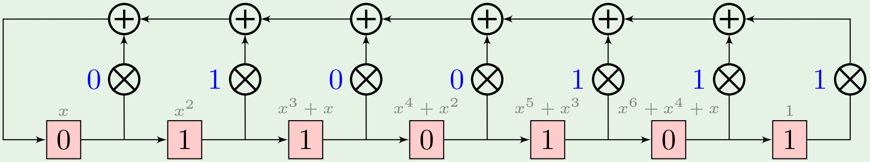
Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$) — Fibonacci



- Different logic configuration
  - Additive feedback is many-to-one, instead of one-to-many
  - The shift direction is reversed

Fibonacci LFSRs

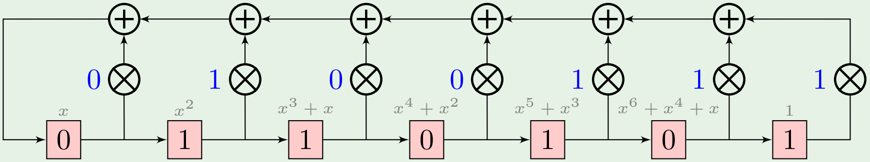
Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$) — Fibonacci



- Different logic configuration
  - Additive feedback is many-to-one, instead of one-to-many
  - The shift direction is reversed
- Same polynomial may be used

Fibonacci LFSRs

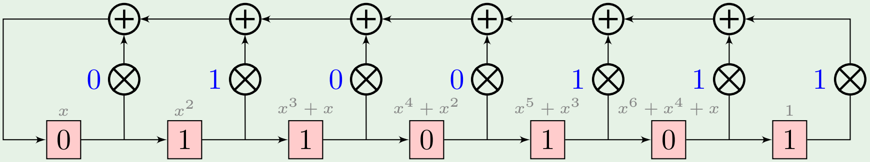
Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$) — Fibonacci



- Different logic configuration
  - Additive feedback is many-to-one, instead of one-to-many
  - The shift direction is reversed
- Same polynomial may be used
- Also represents consecutive powers of x, but in a different basis

Fibonacci LFSRs

Example ($q = 2$, $p(x) = x^7 + x^5 + x^2 + x + 1$) — Fibonacci



- Different logic configuration
  - Additive feedback is many-to-one, instead of one-to-many
  - The shift direction is reversed
- Same polynomial may be used
- Also represents consecutive powers of x, but in a different basis
- Produces an identical sequence in the last digit

de Bruijn sequences from LFSRs

- In the Fibonacci configuration, state transitions correspond to edges in $\mathrm{dB}(q, \ell)$

de Bruijn sequences from LFSRs

- In the Fibonacci configuration, state transitions correspond to edges in $\mathrm{dB}(q, \ell)$
- All non-zero states are traversed in a single cycle

de Bruijn sequences from LFSRs

- In the Fibonacci configuration, state transitions correspond to edges in $\mathrm{dB}(q, \ell)$
- All non-zero states are traversed in a single cycle
  - Gives a Hamiltonian cycle in $\mathrm{dB}(q, \ell) \setminus \{00 \ldots 0\}$

de Bruijn sequences from LFSRs

- In the Fibonacci configuration, state transitions correspond to edges in $\mathrm{dB}(q,\ell)$
- All non-zero states are traversed in a single cycle
  - Gives a Hamiltonian cycle in $\mathrm{dB}(q,\ell) \setminus \{00\ldots0\}$
- Can be extended to all of $\mathrm{dB}(q,\ell)$

de Bruijn sequences from LFSRs

- In the Fibonacci configuration, state transitions correspond to edges in $\mathrm{dB}(q, \ell)$
- All non-zero states are traversed in a single cycle
  - Gives a Hamiltonian cycle in $\mathrm{dB}(q, \ell) \setminus \{00\ldots0\}$
- Can be extended to all of $\mathrm{dB}(q, \ell)$
  - Insert $00\ldots0$ before $00\ldots1$

de Bruijn sequences from LFSRs

- In the Fibonacci configuration, state transitions correspond to edges in $\mathrm{dB}(q, \ell)$
- All non-zero states are traversed in a single cycle
  - Gives a Hamiltonian cycle in $\mathrm{dB}(q, \ell) \setminus \{00 \dots 0\}$
- Can be extended to all of $\mathrm{dB}(q, \ell)$
  - Insert $00 \dots 0$ before $00 \dots 1$
  - Previous edge $c0 \dots 0 \to 00 \dots 1$ becomes two edges $c0 \dots 0 \to 00 \dots 0 \to 00 \dots 1$

de Bruijn sequences from LFSRs

- In the Fibonacci configuration, state transitions correspond to edges in $\mathrm{dB}(q, \ell)$
- All non-zero states are traversed in a single cycle
  - Gives a Hamiltonian cycle in $\mathrm{dB}(q, \ell) \setminus \{00\ldots0\}$
- Can be extended to all of $\mathrm{dB}(q, \ell)$
  - Insert $00\ldots0$ before $00\ldots1$
  - Previous edge $c0\ldots0 \to 00\ldots1$ becomes two edges $c0\ldots0 \to 00\ldots0 \to 00\ldots1$
- Every primitive polynomial gives a different de Bruijn sequence

de Bruijn sequences from LFSRs

- In the Fibonacci configuration, state transitions correspond to edges in $\mathrm{dB}(q, \ell)$
- All non-zero states are traversed in a single cycle
  - Gives a Hamiltonian cycle in $\mathrm{dB}(q, \ell) \setminus \{00\ldots0\}$
- Can be extended to all of $\mathrm{dB}(q, \ell)$
  - Insert $00\ldots0$ before $00\ldots1$
  - Previous edge $c0\ldots0 \to 00\ldots1$ becomes two edges $c0\ldots0 \to 00\ldots0 \to 00\ldots1$
- Every primitive polynomial gives a different de Bruijn sequence
  - There are $\dfrac{\varphi(q^\ell - 1)}{\ell}$ q-ary de Bruijn sequences of order ℓ arising from linear feedback shift registers

de Bruijn sequences from LFSRs

- In the Fibonacci configuration, state transitions correspond to edges in $\mathrm{dB}(q, \ell)$
- All non-zero states are traversed in a single cycle
  - Gives a Hamiltonian cycle in $\mathrm{dB}(q, \ell) \setminus \{00\ldots0\}$
- Can be extended to all of $\mathrm{dB}(q, \ell)$
  - Insert $00\ldots0$ before $00\ldots1$
  - Previous edge $c0\ldots0 \to 00\ldots1$ becomes two edges $c0\ldots0 \to 00\ldots0 \to 00\ldots1$
- Every primitive polynomial gives a different de Bruijn sequence
  - There are $\frac{\varphi(q^\ell - 1)}{\ell}$ q-ary de Bruijn sequences of order ℓ arising from linear feedback shift registers
  - Most are "non-linear" feedback shift registers

Length k subsequences in de Bruijn sequences

- Consider a de Bruijn sequence in $\mathrm{dB}(q, \ell)$ constructed from a LFSR

Length k subsequences in de Bruijn sequences

- Consider a de Bruijn sequence in $\mathrm{dB}(q, \ell)$ constructed from a LFSR
  - This corresponds to an Eulerian circuit in $\mathrm{dB}(q, \ell - 1)$

Length k subsequences in de Bruijn sequences

- Consider a de Bruijn sequence in $\mathrm{dB}(q, \ell)$ constructed from a LFSR
  - This corresponds to an Eulerian circuit in $\mathrm{dB}(q, \ell - 1)$
  - State polynomial of LFSR represents an edge

Length k subsequences in de Bruijn sequences

- Consider a de Bruijn sequence in $\mathrm{dB}(q, \ell)$ constructed from a LFSR
  - This corresponds to an Eulerian circuit in $\mathrm{dB}(q, \ell - 1)$
  - State polynomial of LFSR represents an edge
- Suppose we want to find a k-subcircuit in this Eulerian circuit

Length k subsequences in de Bruijn sequences

- Consider a de Bruijn sequence in $\mathrm{dB}(q, \ell)$ constructed from a LFSR
  - This corresponds to an Eulerian circuit in $\mathrm{dB}(q, \ell - 1)$
  - State polynomial of LFSR represents an edge
- Suppose we want to find a k-subcircuit in this Eulerian circuit
  - After k iterations of the LFSR, we should be at the same vertex

Length k subsequences in de Bruijn sequences

- Consider a de Bruijn sequence in $\mathrm{dB}(q, \ell)$ constructed from a LFSR
  - This corresponds to an Eulerian circuit in $\mathrm{dB}(q, \ell - 1)$
  - State polynomial of LFSR represents an edge
- Suppose we want to find a k-subcircuit in this Eulerian circuit
  - After k iterations of the LFSR, we should be at the same vertex
  - Equivalent to changing the constant term in state polynomial

Length k subsequences in de Bruijn sequences

- Consider a de Bruijn sequence in $\mathrm{dB}(q, \ell)$ constructed from a LFSR
  - This corresponds to an Eulerian circuit in $\mathrm{dB}(q, \ell - 1)$
  - State polynomial of LFSR represents an edge
- Suppose we want to find a k-subcircuit in this Eulerian circuit
  - After k iterations of the LFSR, we should be at the same vertex
  - Equivalent to changing the constant term in state polynomial
- This can be expressed as an equation in the quotient field:

Length k subsequences in de Bruijn sequences

- Consider a de Bruijn sequence in $\mathrm{dB}(q, \ell)$ constructed from a LFSR
  - This corresponds to an Eulerian circuit in $\mathrm{dB}(q, \ell - 1)$
  - State polynomial of LFSR represents an edge
- Suppose we want to find a k-subcircuit in this Eulerian circuit
  - After k iterations of the LFSR, we should be at the same vertex
  - Equivalent to changing the constant term in state polynomial
- This can be expressed as an equation in the quotient field:
  - $x^k f(x) = f(x) + c$ for some $c \in \mathrm{GF}(q) \setminus \{0\}$

Length k subsequences in de Bruijn sequences

- Consider a de Bruijn sequence in $\mathrm{dB}(q, \ell)$ constructed from a LFSR
  - This corresponds to an Eulerian circuit in $\mathrm{dB}(q, \ell - 1)$
  - State polynomial of LFSR represents an edge
- Suppose we want to find a k-subcircuit in this Eulerian circuit
  - After k iterations of the LFSR, we should be at the same vertex
  - Equivalent to changing the constant term in state polynomial
- This can be expressed as an equation in the quotient field:
  - $x^k f(x) = f(x) + c$ for some $c \in \mathrm{GF}(q) \setminus \{0\}$
  - Unique solution for each c: $f(x) = \frac{c}{x^k - 1}$

Length k subsequences in de Bruijn sequences

- Consider a de Bruijn sequence in $\mathrm{dB}(q,\ell)$ constructed from a LFSR
  - This corresponds to an Eulerian circuit in $\mathrm{dB}(q,\ell-1)$
  - State polynomial of LFSR represents an edge
- Suppose we want to find a k-subcircuit in this Eulerian circuit
  - After k iterations of the LFSR, we should be at the same vertex
  - Equivalent to changing the constant term in state polynomial
- This can be expressed as an equation in the quotient field:
  - $x^k f(x) = f(x) + c$ for some $c \in \mathrm{GF}(q) \setminus \{0\}$
  - Unique solution for each c: $f(x) = \frac{c}{x^k - 1}$
- We have found $q-1$ k-cycles in $\mathrm{dB}(q,\ell)$

Length k subsequences in de Bruijn sequences

- Consider a de Bruijn sequence in $\mathrm{dB}(q, \ell)$ constructed from a LFSR
  - This corresponds to an Eulerian circuit in $\mathrm{dB}(q, \ell - 1)$
  - State polynomial of LFSR represents an edge
- Suppose we want to find a k-subcircuit in this Eulerian circuit
  - After k iterations of the LFSR, we should be at the same vertex
  - Equivalent to changing the constant term in state polynomial
- This can be expressed as an equation in the quotient field:
  - $x^k f(x) = f(x) + c$ for some $c \in \mathrm{GF}(q) \setminus \{0\}$
  - Unique solution for each c: $f(x) = \frac{c}{x^k - 1}$
- We have found $q - 1$ k-cycles in $\mathrm{dB}(q, \ell)$
  - Requires $(q - 1)k \leq q^\ell - 1$ for cycles to be disjoint

Length k subsequences in de Bruijn sequences

- Consider a de Bruijn sequence in $\mathrm{dB}(q,\ell)$ constructed from a LFSR
  - This corresponds to an Eulerian circuit in $\mathrm{dB}(q,\ell-1)$
  - State polynomial of LFSR represents an edge
- Suppose we want to find a k-subcircuit in this Eulerian circuit
  - After k iterations of the LFSR, we should be at the same vertex
  - Equivalent to changing the constant term in state polynomial
- This can be expressed as an equation in the quotient field:
  - $x^k f(x) = f(x) + c$ for some $c \in \mathrm{GF}(q) \setminus \{0\}$
  - Unique solution for each c: $f(x) = \frac{c}{x^k - 1}$
- We have found $q-1$ k-cycles in $\mathrm{dB}(q,\ell)$
  - Requires $(q-1)k \leq q^\ell - 1$ for cycles to be disjoint
  - This is sufficient because cycles are evenly distributed

Length k subsequences in de Bruijn sequences

- Consider a de Bruijn sequence in $\mathrm{dB}(q, \ell)$ constructed from a LFSR
  - This corresponds to an Eulerian circuit in $\mathrm{dB}(q, \ell - 1)$
  - State polynomial of LFSR represents an edge
- Suppose we want to find a k-subcircuit in this Eulerian circuit
  - After k iterations of the LFSR, we should be at the same vertex
  - Equivalent to changing the constant term in state polynomial
- This can be expressed as an equation in the quotient field:
  - $x^k f(x) = f(x) + c$ for some $c \in \mathrm{GF}(q) \setminus \{0\}$
  - Unique solution for each c: $f(x) = \frac{c}{x^k - 1}$
- We have found $q - 1$ k-cycles in $\mathrm{dB}(q, \ell)$
  - Requires $(q-1)k \leq q^\ell - 1$ for cycles to be disjoint
  - This is sufficient because cycles are evenly distributed
- $\mathcal{E}(q, k, \ell) \geq q - 1$ for $k \leq \dfrac{q^\ell - 1}{q - 1}$

Optimality for $k = q^{\ell-1}$

- Consider the case when $k = q^{\ell-1}$

Optimality for $k = q^{\ell-1}$

- Consider the case when $k = q^{\ell-1}$
  - $q^\ell - k$ states of the LFSR are covered by the $q-1$ k-cycles

Optimality for $k = q^{\ell-1}$

- Consider the case when $k = q^{\ell-1}$
  - $q^\ell - k$ states of the LFSR are covered by the $q-1$ k-cycles
- Take these cycles out of the LFSR sequence

Optimality for $k = q^{\ell-1}$

- Consider the case when $k = q^{\ell-1}$
  - $q^\ell - k$ states of the LFSR are covered by the $q-1$ k-cycles
- Take these cycles out of the LFSR sequence
  - The remaining $k-1$ states form a cycle C

Optimality for $k = q^{\ell-1}$

- Consider the case when $k = q^{\ell-1}$
  - $q^\ell - k$ states of the LFSR are covered by the $q-1$ k-cycles
- Take these cycles out of the LFSR sequence
  - The remaining $k-1$ states form a cycle C
- Suppose one of the states in C is a constant polynomial

Optimality for $k = q^{\ell-1}$

- Consider the case when $k = q^{\ell-1}$
  - $q^\ell - k$ states of the LFSR are covered by the $q-1$ k-cycles
- Take these cycles out of the LFSR sequence
  - The remaining $k-1$ states form a cycle C
- Suppose one of the states in C is a constant polynomial
  - The zero polynomial can be inserted into C

Optimality for $k = q^{\ell-1}$

- Consider the case when $k = q^{\ell-1}$
  - $q^\ell - k$ states of the LFSR are covered by the $q-1$ k-cycles
- Take these cycles out of the LFSR sequence
  - The remaining $k-1$ states form a cycle C
- Suppose one of the states in C is a constant polynomial
  - The zero polynomial can be inserted into C
  - $\mathrm{dB}(q, \ell)$ contains q disjoint k-cycles

Optimality for $k = q^{\ell-1}$

- Consider the case when $k = q^{\ell-1}$
  - $q^\ell - k$ states of the LFSR are covered by the $q-1$ k-cycles
- Take these cycles out of the LFSR sequence
  - The remaining $k-1$ states form a cycle C
- Suppose one of the states in C is a constant polynomial
  - The zero polynomial can be inserted into C
  - $\mathrm{dB}(q, \ell)$ contains q disjoint k-cycles
  - $\mathcal{E}(q, q^{\ell-1}, \ell) = q$

Optimality for $k = q^{\ell-1}$

- Consider the case when $k = q^{\ell-1}$
  - $q^\ell - k$ states of the LFSR are covered by the $q-1$ k-cycles
- Take these cycles out of the LFSR sequence
  - The remaining $k-1$ states form a cycle C
- Suppose one of the states in C is a constant polynomial
  - The zero polynomial can be inserted into C
  - $\mathrm{dB}(q,\ell)$ contains q disjoint k-cycles
  - $\mathcal{E}(q, q^{\ell-1}, \ell) = q$
- C contains a constant iff $-\log_x(x^k - 1) \mod \frac{q^\ell-1}{q-1} \leq \frac{k-1}{q-1}$

Optimality for $k = q^{\ell-1}$

- Consider the case when $k = q^{\ell-1}$
  - $q^\ell - k$ states of the LFSR are covered by the $q-1$ k-cycles
- Take these cycles out of the LFSR sequence
  - The remaining $k-1$ states form a cycle C
- Suppose one of the states in C is a constant polynomial
  - The zero polynomial can be inserted into C
  - $\mathrm{dB}(q,\ell)$ contains q disjoint k-cycles
  - $\mathcal{E}(q, q^{\ell-1}, \ell) = q$
- C contains a constant iff $-\log_x(x^k - 1) \mod \frac{q^\ell-1}{q-1} \leq \frac{k-1}{q-1}$
  - This may be true for some primitive polynomials and false for others

Optimality for $k = q^{\ell-1}$

- Consider the case when $k = q^{\ell-1}$
  - $q^\ell - k$ states of the LFSR are covered by the $q-1$ k-cycles
- Take these cycles out of the LFSR sequence
  - The remaining $k-1$ states form a cycle C
- Suppose one of the states in C is a constant polynomial
  - The zero polynomial can be inserted into C
  - $\mathrm{dB}(q,\ell)$ contains q disjoint k-cycles
  - $\mathcal{E}(q, q^{\ell-1}, \ell) = q$
- C contains a constant iff $-\log_x(x^k - 1) \mod \frac{q^\ell-1}{q-1} \leq \frac{k-1}{q-1}$
  - This may be true for some primitive polynomials and false for others
  - For a given q and ℓ, we only need one polynomial

Optimality for $k = q^{\ell-1}$

- Consider the case when $k = q^{\ell-1}$
  - $q^\ell - k$ states of the LFSR are covered by the $q-1$ k-cycles
- Take these cycles out of the LFSR sequence
  - The remaining $k-1$ states form a cycle C
- Suppose one of the states in C is a constant polynomial
  - The zero polynomial can be inserted into C
  - $\mathrm{dB}(q,\ell)$ contains q disjoint k-cycles
  - $\mathcal{E}(q, q^{\ell-1}, \ell) = q$
- C contains a constant iff $-\log_x(x^k - 1) \mod \frac{q^\ell - 1}{q-1} \leq \frac{k-1}{q-1}$
  - This may be true for some primitive polynomials and false for others
  - For a given q and ℓ, we only need one polynomial
  - Depends on the distribution of the discrete logarithm

Multiplying cycles

Theorem

Fix a value of ℓ *and set* $\mathcal{E}_1 = \mathcal{E}(q_1, k_1, \ell)$ *and* $\mathcal{E}_2 = \mathcal{E}(q_2, k_2, \ell)$*. Then*

$$\mathcal{E}(q_1 q_2, \operatorname{lcm}(k_1, k_2), \ell) \geq \gcd(k_1, k_2)\, \mathcal{E}_1\, \mathcal{E}_2.$$

Multiplying cycles

Theorem

Fix a value of ℓ *and set* $\mathcal{E}_1 = \mathcal{E}(q_1, k_1, \ell)$ *and* $\mathcal{E}_2 = \mathcal{E}(q_2, k_2, \ell)$*. Then*

$$\mathcal{E}(q_1 q_2, \operatorname{lcm}(k_1, k_2), \ell) \geq \gcd(k_1, k_2)\, \mathcal{E}_1\, \mathcal{E}_2.$$

- Preserves optimality w.r.t. upper bound $\mathcal{E} \leq \dfrac{q^\ell}{k}$

Multiplying cycles

Theorem

Fix a value of ℓ *and set* $\mathcal{E}_1 = \mathcal{E}(q_1, k_1, \ell)$ *and* $\mathcal{E}_2 = \mathcal{E}(q_2, k_2, \ell)$*. Then*

$$\mathcal{E}(q_1 q_2, \operatorname{lcm}(k_1, k_2), \ell) \geq \gcd(k_1, k_2)\, \mathcal{E}_1\, \mathcal{E}_2.$$

- Preserves optimality w.r.t. upper bound $\mathcal{E} \leq \frac{q^\ell}{k}$
  - If $\mathcal{E}_1$ and $\mathcal{E}_2$ are optimal, then so is $\mathcal{E}(q_1 q_2, \operatorname{lcm}(k_1, k_2), \ell)$

Multiplying cycles

Theorem

Fix a value of ℓ *and set* $\mathcal{E}_1 = \mathcal{E}(q_1, k_1, \ell)$ *and* $\mathcal{E}_2 = \mathcal{E}(q_2, k_2, \ell)$*. Then*

$$\mathcal{E}(q_1 q_2, \operatorname{lcm}(k_1, k_2), \ell) \geq \gcd(k_1, k_2)\, \mathcal{E}_1\, \mathcal{E}_2.$$

- Preserves optimality w.r.t. upper bound $\mathcal{E} \leq \dfrac{q^\ell}{k}$
  - If $\mathcal{E}_1$ and $\mathcal{E}_2$ are optimal, then so is $\mathcal{E}(q_1 q_2, \operatorname{lcm}(k_1, k_2), \ell)$
- When k_1 and k_2 are coprime, not much is gained

Multiplying cycles

Theorem

Fix a value of ℓ *and set* $\mathcal{E}_1 = \mathcal{E}(q_1, k_1, \ell)$ *and* $\mathcal{E}_2 = \mathcal{E}(q_2, k_2, \ell)$. *Then*

$$\mathcal{E}(q_1 q_2, \operatorname{lcm}(k_1, k_2), \ell) \geq \gcd(k_1, k_2)\, \mathcal{E}_1\, \mathcal{E}_2.$$

- Preserves optimality w.r.t. upper bound $\mathcal{E} \leq \dfrac{q^\ell}{k}$
  - If $\mathcal{E}_1$ and $\mathcal{E}_2$ are optimal, then so is $\mathcal{E}(q_1 q_2, \operatorname{lcm}(k_1, k_2), \ell)$
- When k_1 and k_2 are coprime, not much is gained
- Largest increase in $\mathcal{E}$ when $k_1 = k_2$

Multiplying cycles

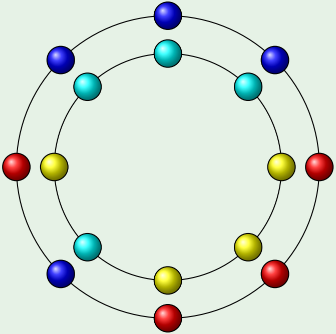
Theorem

Fix a value of ℓ *and set* $\mathcal{E}_1 = \mathcal{E}(q_1, k_1, \ell)$ *and* $\mathcal{E}_2 = \mathcal{E}(q_2, k_2, \ell)$*. Then*

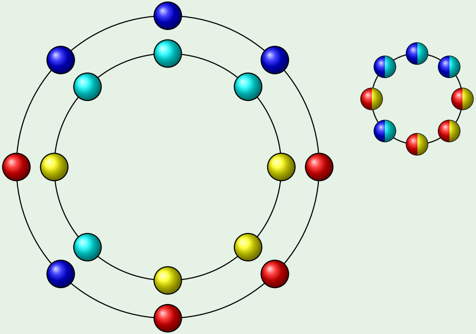
$$\mathcal{E}(q_1 q_2, \operatorname{lcm}(k_1, k_2), \ell) \geq \gcd(k_1, k_2)\, \mathcal{E}_1\, \mathcal{E}_2.$$

- Preserves optimality w.r.t. upper bound $\mathcal{E} \leq \dfrac{q^\ell}{k}$
  - If $\mathcal{E}_1$ and $\mathcal{E}_2$ are optimal, then so is $\mathcal{E}(q_1 q_2, \operatorname{lcm}(k_1, k_2), \ell)$
- When k_1 and k_2 are coprime, not much is gained
- Largest increase in $\mathcal{E}$ when $k_1 = k_2$
  - Easy to get large $\mathcal{E}$ when q is divisible by a high power

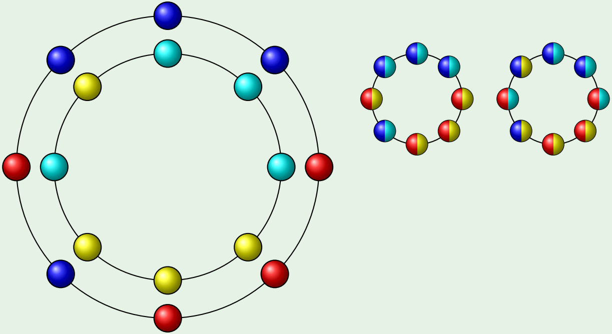
Construction for $k_1 = k_2$



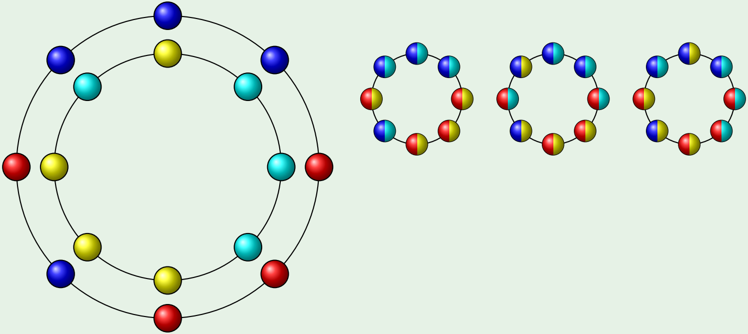
Construction for $k_1 = k_2$



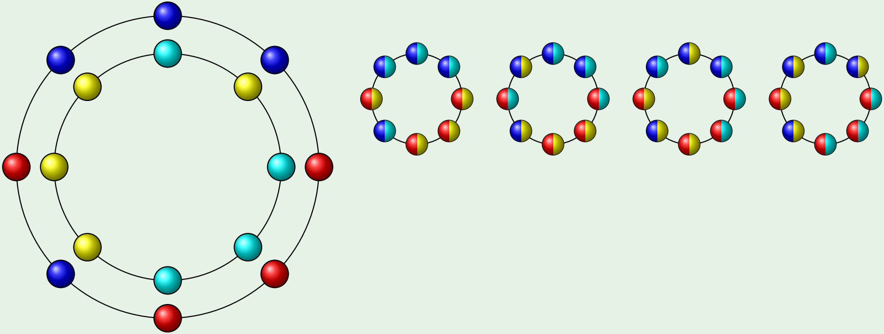
Construction for $k_1 = k_2$



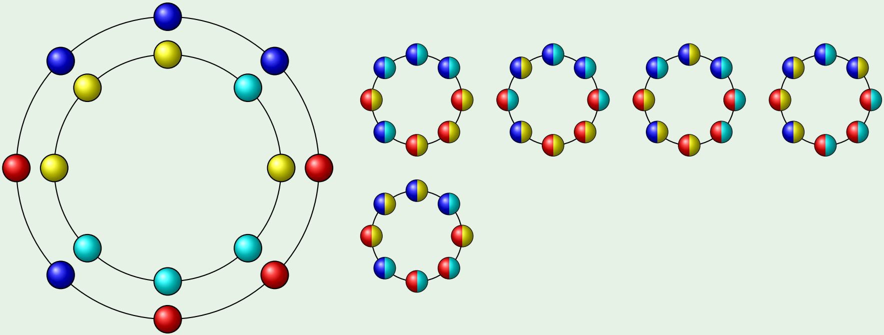
Construction for $k_1 = k_2$



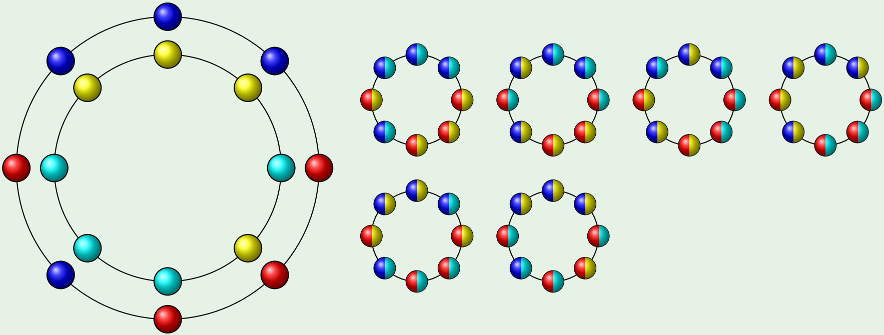
Construction for $k_1 = k_2$



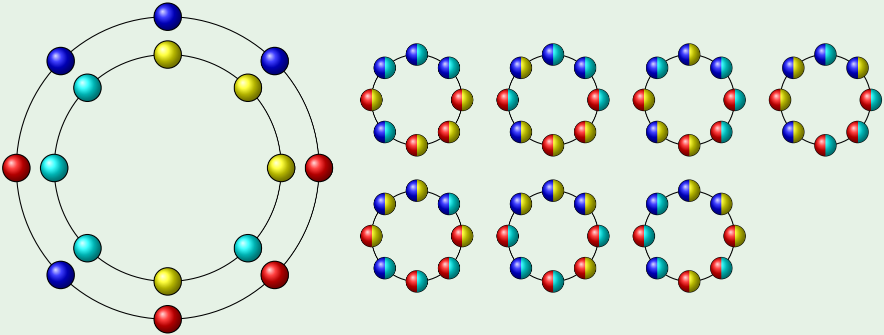
Construction for $k_1 = k_2$



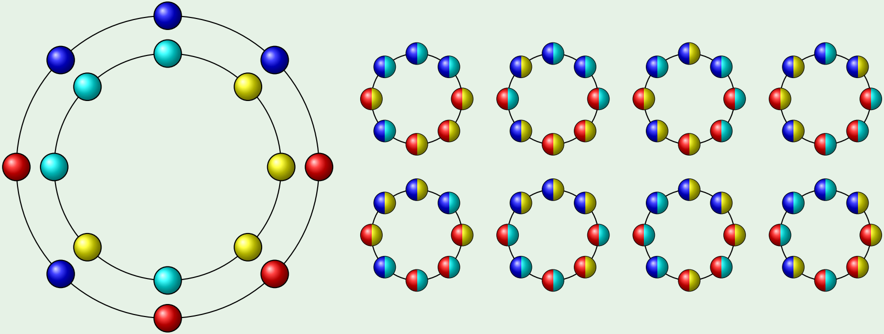
Construction for $k_1 = k_2$



Construction for $k_1 = k_2$



Construction for $k_1 = k_2$



Proof of correctness (sketch)

- Consider any sequence of ℓ colour pairs in resulting eBugs

Proof of correctness (sketch)

- Consider any sequence of ℓ colour pairs in resulting eBugs
  - This gives a red/blue sequence and a yellow/cyan sequence

Proof of correctness (sketch)

- Consider any sequence of ℓ colour pairs in resulting eBugs
  - This gives a red/blue sequence and a yellow/cyan sequence
  - These had unique positions in original colourings

Proof of correctness (sketch)

- Consider any sequence of ℓ colour pairs in resulting eBugs
  - This gives a red/blue sequence and a yellow/cyan sequence
  - These had unique positions in original colourings
- Sequence corresponds to a particular pair of eBugs in a particular orientation

Proof of correctness (sketch)

- Consider any sequence of ℓ colour pairs in resulting eBugs
  - This gives a red/blue sequence and a yellow/cyan sequence
  - These had unique positions in original colourings
- Sequence corresponds to a particular pair of eBugs in a particular orientation
  - Resulting eBugs can still be uniquely identified and oriented

Proof of correctness (sketch)

- Consider any sequence of ℓ colour pairs in resulting eBugs
  - This gives a red/blue sequence and a yellow/cyan sequence
  - These had unique positions in original colourings
- Sequence corresponds to a particular pair of eBugs in a particular orientation
  - Resulting eBugs can still be uniquely identified and oriented
- Theorem easily extends to $k_1 \neq k_2$ case

Necklaces

Definition

A necklace is an equivalence class of words under cyclic rotation. The length of a necklace is the length of any word in the class, while the size of a necklace is the number of words in the class.

Necklaces

Definition

A necklace is an equivalence class of words under cyclic rotation. The length of a necklace is the length of any word in the class, while the size of a necklace is the number of words in the class.

Example

$$001021 \equiv 010210 \equiv 102100 \equiv 021001 \equiv 210010 \equiv 100102$$

Necklaces

Definition

A necklace is an equivalence class of words under cyclic rotation. The length of a necklace is the length of any word in the class, while the size of a necklace is the number of words in the class.

Example

$001021 \equiv 010210 \equiv 102100 \equiv 021001 \equiv 210010 \equiv 100102$

- Every necklace gives a cycle in a de Bruijn graph

Necklaces

Definition

A necklace is an equivalence class of words under cyclic rotation. The length of a necklace is the length of any word in the class, while the size of a necklace is the number of words in the class.

Example

$001021 \equiv 010210 \equiv 102100 \equiv 021001 \equiv 210010 \equiv 100102$

- Every necklace gives a cycle in a de Bruijn graph
  - The length of the cycle is the size of the necklace

Necklaces

Definition

A necklace is an equivalence class of words under cyclic rotation. The length of a necklace is the length of any word in the class, while the size of a necklace is the number of words in the class.

Example

$001021 \equiv 010210 \equiv 102100 \equiv 021001 \equiv 210010 \equiv 100102$

- Every necklace gives a cycle in a de Bruijn graph
  - The length of the cycle is the size of the necklace
- Cycles can be concatenated in the previous de Bruijn graph

Necklaces

Definition

A necklace is an equivalence class of words under cyclic rotation. The length of a necklace is the length of any word in the class, while the size of a necklace is the number of words in the class.

Example

$001021 \equiv 010210 \equiv 102100 \equiv 021001 \equiv 210010 \equiv 100102$

- Every necklace gives a cycle in a de Bruijn graph
  - The length of the cycle is the size of the necklace
- Cycles can be concatenated in the previous de Bruijn graph
  - Two necklaces of length ℓ are mergable if they share a subword of length $\ell - 1$

Necklaces

Definition

A necklace is an equivalence class of words under cyclic rotation. The length of a necklace is the length of any word in the class, while the size of a necklace is the number of words in the class.

Example

$001021 \equiv 010210 \equiv 102100 \equiv 021001 \equiv 210010 \equiv 100102$

- Every necklace gives a cycle in a de Bruijn graph
  - The length of the cycle is the size of the necklace
- Cycles can be concatenated in the previous de Bruijn graph
  - Two necklaces of length ℓ are mergable if they share a subword of length $\ell - 1$
- Construct a graph $N(q, \ell)$ of necklaces of length ℓ over q letters

Necklaces

Definition

A necklace is an equivalence class of words under cyclic rotation. The length of a necklace is the length of any word in the class, while the size of a necklace is the number of words in the class.

Example

$001021 \equiv 010210 \equiv 102100 \equiv 021001 \equiv 210010 \equiv 100102$

- Every necklace gives a cycle in a de Bruijn graph
  - The length of the cycle is the size of the necklace
- Cycles can be concatenated in the previous de Bruijn graph
  - Two necklaces of length ℓ are mergable if they share a subword of length $\ell - 1$
- Construct a graph $N(q, \ell)$ of necklaces of length ℓ over q letters
  - Edge between two necklaces if they are mergable

Merging necklaces

- Consider a connected subgraph of $N(q, \ell)$

Merging necklaces

- Consider a connected subgraph of $N(q, \ell)$
  - Edges can be contracted by merging the appropriate cycles

Merging necklaces

- Consider a connected subgraph of $N(q, \ell)$
  - Edges can be contracted by merging the appropriate cycles
  - The whole subgraph will produce one long cycle

Merging necklaces

- Consider a connected subgraph of $N(q, \ell)$
  - Edges can be contracted by merging the appropriate cycles
  - The whole subgraph will produce one long cycle
  - The length of the cycle is the sum of the necklace sizes

Merging necklaces

- Consider a connected subgraph of $N(q, \ell)$
  - Edges can be contracted by merging the appropriate cycles
  - The whole subgraph will produce one long cycle
  - The length of the cycle is the sum of the necklace sizes
- Consider a partition of $N(q, \ell)$ into connected pieces of total size k

Merging necklaces

- Consider a connected subgraph of $N(q, \ell)$
  - Edges can be contracted by merging the appropriate cycles
  - The whole subgraph will produce one long cycle
  - The length of the cycle is the sum of the necklace sizes
- Consider a partition of $N(q, \ell)$ into connected pieces of total size k
  - Gives a partition of $\mathrm{dB}(q, \ell)$ into k-cycles

Merging necklaces

- Consider a connected subgraph of $N(q, \ell)$
  - Edges can be contracted by merging the appropriate cycles
  - The whole subgraph will produce one long cycle
  - The length of the cycle is the sum of the necklace sizes
- Consider a partition of $N(q, \ell)$ into connected pieces of total size k
  - Gives a partition of $\mathrm{dB}(q, \ell)$ into k-cycles
  - Difficult to find because $N(q, \ell)$ has necklaces of different sizes

Merging necklaces

- Consider a connected subgraph of $N(q, \ell)$
  - Edges can be contracted by merging the appropriate cycles
  - The whole subgraph will produce one long cycle
  - The length of the cycle is the sum of the necklace sizes
- Consider a partition of $N(q, \ell)$ into connected pieces of total size k
  - Gives a partition of $\mathrm{dB}(q, \ell)$ into k-cycles
  - Difficult to find because $N(q, \ell)$ has necklaces of different sizes
- $N(q, \ell)$ contains necklaces of each size m that divides ℓ

Merging necklaces

- Consider a connected subgraph of $N(q, \ell)$
  - Edges can be contracted by merging the appropriate cycles
  - The whole subgraph will produce one long cycle
  - The length of the cycle is the sum of the necklace sizes
- Consider a partition of $N(q, \ell)$ into connected pieces of total size k
  - Gives a partition of $\mathrm{dB}(q, \ell)$ into k-cycles
  - Difficult to find because $N(q, \ell)$ has necklaces of different sizes
- $N(q, \ell)$ contains necklaces of each size m that divides ℓ
  - Moreau's necklace counting function: $M(q, m) = \frac{1}{m} \sum_{d|m} \mu(d) q^{m/d}$

Merging necklaces

- Consider a connected subgraph of $N(q, \ell)$
  - Edges can be contracted by merging the appropriate cycles
  - The whole subgraph will produce one long cycle
  - The length of the cycle is the sum of the necklace sizes
- Consider a partition of $N(q, \ell)$ into connected pieces of total size k
  - Gives a partition of $\mathrm{dB}(q, \ell)$ into k-cycles
  - Difficult to find because $N(q, \ell)$ has necklaces of different sizes
- $N(q, \ell)$ contains necklaces of each size m that divides ℓ
  - Moreau's necklace counting function: $M(q, m) = \frac{1}{m} \sum_{d|m} \mu(d) q^{m/d}$
- If q and ℓ are coprime, $M(q, m)$ is divisible by q for each m

Merging necklaces

- Consider a connected subgraph of $N(q, \ell)$
  - Edges can be contracted by merging the appropriate cycles
  - The whole subgraph will produce one long cycle
  - The length of the cycle is the sum of the necklace sizes
- Consider a partition of $N(q, \ell)$ into connected pieces of total size k
  - Gives a partition of $\mathrm{dB}(q, \ell)$ into k-cycles
  - Difficult to find because $N(q, \ell)$ has necklaces of different sizes
- $N(q, \ell)$ contains necklaces of each size m that divides ℓ
  - Moreau's necklace counting function: $M(q, m) = \frac{1}{m} \sum_{d|m} \mu(d) q^{m/d}$
- If q and ℓ are coprime, $M(q, m)$ is divisible by q for each m
  - Potential to partition $N(q, \ell)$ into q connected pieces

Merging necklaces

- Consider a connected subgraph of $N(q, \ell)$
  - Edges can be contracted by merging the appropriate cycles
  - The whole subgraph will produce one long cycle
  - The length of the cycle is the sum of the necklace sizes
- Consider a partition of $N(q, \ell)$ into connected pieces of total size k
  - Gives a partition of $\mathrm{dB}(q, \ell)$ into k-cycles
  - Difficult to find because $N(q, \ell)$ has necklaces of different sizes
- $N(q, \ell)$ contains necklaces of each size m that divides ℓ
  - Moreau's necklace counting function: $M(q, m) = \frac{1}{m} \sum_{d|m} \mu(d) q^{m/d}$
- If q and ℓ are coprime, $M(q, m)$ is divisible by q for each m
  - Potential to partition $N(q, \ell)$ into q connected pieces
  - For each size m, every piece has same number of size m necklaces

Merging necklaces

- Consider a connected subgraph of $N(q, \ell)$
  - Edges can be contracted by merging the appropriate cycles
  - The whole subgraph will produce one long cycle
  - The length of the cycle is the sum of the necklace sizes
- Consider a partition of $N(q, \ell)$ into connected pieces of total size k
  - Gives a partition of $\mathrm{dB}(q, \ell)$ into k-cycles
  - Difficult to find because $N(q, \ell)$ has necklaces of different sizes
- $N(q, \ell)$ contains necklaces of each size m that divides ℓ
  - Moreau's necklace counting function: $M(q, m) = \frac{1}{m} \sum_{d|m} \mu(d) q^{m/d}$
- If q and ℓ are coprime, $M(q, m)$ is divisible by q for each m
  - Potential to partition $N(q, \ell)$ into q connected pieces
  - For each size m, every piece has same number of size m necklaces
  - Resulting cycles all have the same length

Merging necklaces

- Consider a connected subgraph of $N(q,\ell)$
  - Edges can be contracted by merging the appropriate cycles
  - The whole subgraph will produce one long cycle
  - The length of the cycle is the sum of the necklace sizes
- Consider a partition of $N(q,\ell)$ into connected pieces of total size k
  - Gives a partition of $\mathrm{dB}(q,\ell)$ into k-cycles
  - Difficult to find because $N(q,\ell)$ has necklaces of different sizes
- $N(q,\ell)$ contains necklaces of each size m that divides ℓ
  - Moreau's necklace counting function: $M(q,m) = \frac{1}{m}\sum_{d|m}\mu(d)q^{m/d}$
- If q and ℓ are coprime, $M(q,m)$ is divisible by q for each m
  - Potential to partition $N(q,\ell)$ into q connected pieces
  - For each size m, every piece has same number of size m necklaces
  - Resulting cycles all have the same length
  - Will give $\mathcal{E}(q,q^{\ell-1},\ell) = q$

Partitioning $N(q, \ell)$

Example ($q = 2$)

As ℓ must be odd, each necklace contains a majority of some letter (0 or 1). Let N_0 (N_1) be the set of necklaces with more 0s (1s).

Partitioning $N(q, \ell)$

Example ($q = 2$)

As ℓ must be odd, each necklace contains a majority of some letter (0 or 1). Let N_0 (N_1) be the set of necklaces with more 0s (1s).

- Let $n_0 \in N_0$ be the all zeroes necklace

Partitioning $N(q, \ell)$

Example ($q = 2$)

As ℓ must be odd, each necklace contains a majority of some letter (0 or 1). Let N_0 (N_1) be the set of necklaces with more 0s (1s).

- Let $n_0 \in N_0$ be the all zeroes necklace
- Pick a necklace n in $N_0 \setminus \{n_0\}$

Partitioning $N(q, \ell)$

Example ($q = 2$)

As ℓ must be odd, each necklace contains a majority of some letter (0 or 1). Let N_0 (N_1) be the set of necklaces with more 0s (1s).

- Let $n_0 \in N_0$ be the all zeroes necklace
- Pick a necklace n in $N_0 \setminus \{n_0\}$
  - Change the first 1 in n to a 0, and call this necklace $\bar{n}$

Partitioning $N(q, \ell)$

Example ($q = 2$)

As ℓ must be odd, each necklace contains a majority of some letter (0 or 1). Let N_0 (N_1) be the set of necklaces with more 0s (1s).

- Let $n_0 \in N_0$ be the all zeroes necklace
- Pick a necklace n in $N_0 \setminus \{n_0\}$
  - Change the first 1 in n to a 0, and call this necklace $\bar{n}$
  - There is an edge in $N(q, \ell)$ between n and $\bar{n}$

Partitioning $N(q, \ell)$

Example ($q = 2$)

As ℓ must be odd, each necklace contains a majority of some letter (0 or 1). Let N_0 (N_1) be the set of necklaces with more 0s (1s).

- Let $n_0 \in N_0$ be the all zeroes necklace
- Pick a necklace n in $N_0 \setminus \{n_0\}$
  - Change the first 1 in n to a 0, and call this necklace $\bar{n}$
  - There is an edge in $N(q, \ell)$ between n and $\bar{n}$
  - Note that $\bar{n} \in N_0$

Partitioning $N(q, \ell)$

Example ($q = 2$)

As ℓ must be odd, each necklace contains a majority of some letter (0 or 1). Let N_0 (N_1) be the set of necklaces with more 0s (1s).

- Let $n_0 \in N_0$ be the all zeroes necklace
- Pick a necklace n in $N_0 \setminus \{n_0\}$
  - Change the first 1 in n to a 0, and call this necklace $\bar{n}$
  - There is an edge in $N(q, \ell)$ between n and $\bar{n}$
  - Note that $\bar{n} \in N_0$
- By induction, there is a path in N_0 from n to n_0

Partitioning $N(q, \ell)$

Example ($q = 2$)

As ℓ must be odd, each necklace contains a majority of some letter (0 or 1). Let N_0 (N_1) be the set of necklaces with more 0s (1s).

- Let $n_0 \in N_0$ be the all zeroes necklace
- Pick a necklace n in $N_0 \setminus \{n_0\}$
  - Change the first 1 in n to a 0, and call this necklace $\bar{n}$
  - There is an edge in $N(q, \ell)$ between n and $\bar{n}$
  - Note that $\bar{n} \in N_0$
- By induction, there is a path in N_0 from n to n_0
  - N_0 is connected

Partitioning $N(q, \ell)$

Example ($q = 2$)

As ℓ must be odd, each necklace contains a majority of some letter (0 or 1). Let N_0 (N_1) be the set of necklaces with more 0s (1s).

- Let $n_0 \in N_0$ be the all zeroes necklace
- Pick a necklace n in $N_0 \setminus \{n_0\}$
  - Change the first 1 in n to a 0, and call this necklace $\bar{n}$
  - There is an edge in $N(q, \ell)$ between n and $\bar{n}$
  - Note that $\bar{n} \in N_0$
- By induction, there is a path in N_0 from n to n_0
  - N_0 is connected
  - By symmetry, N_1 is connected

Partitioning $N(q, \ell)$

Example ($q = 2$)

As ℓ must be odd, each necklace contains a majority of some letter (0 or 1). Let N_0 (N_1) be the set of necklaces with more 0s (1s).

- Let $n_0 \in N_0$ be the all zeroes necklace
- Pick a necklace n in $N_0 \setminus \{n_0\}$
  - Change the first 1 in n to a 0, and call this necklace $\bar{n}$
  - There is an edge in $N(q, \ell)$ between n and $\bar{n}$
  - Note that $\bar{n} \in N_0$
- By induction, there is a path in N_0 from n to n_0
  - N_0 is connected
  - By symmetry, N_1 is connected
- Each component gives a cycle of length $2^{\ell-1}$ in $\mathrm{dB}(q, \ell)$

Partitioning $N(q, \ell)$

Example ($q = 2$)

As ℓ must be odd, each necklace contains a majority of some letter (0 or 1). Let N_0 (N_1) be the set of necklaces with more 0s (1s).

- Let $n_0 \in N_0$ be the all zeroes necklace
- Pick a necklace n in $N_0 \setminus \{n_0\}$
  - Change the first 1 in n to a 0, and call this necklace $\bar{n}$
  - There is an edge in $N(q, \ell)$ between n and $\bar{n}$
  - Note that $\bar{n} \in N_0$
- By induction, there is a path in N_0 from n to n_0
  - N_0 is connected
  - By symmetry, N_1 is connected
- Each component gives a cycle of length $2^{\ell-1}$ in $\mathrm{dB}(q, \ell)$
- Similar methods give a partition for general q whenever $\gcd(q, \ell) = 1$

Partitioning $N(q, \ell)$

Example ($q = 2$)

As ℓ must be odd, each necklace contains a majority of some letter (0 or 1). Let N_0 (N_1) be the set of necklaces with more 0s (1s).

- Let $n_0 \in N_0$ be the all zeroes necklace
- Pick a necklace n in $N_0 \setminus \{n_0\}$
  - Change the first 1 in n to a 0, and call this necklace $\bar{n}$
  - There is an edge in $N(q, \ell)$ between n and $\bar{n}$
  - Note that $\bar{n} \in N_0$
- By induction, there is a path in N_0 from n to n_0
  - N_0 is connected
  - By symmetry, N_1 is connected
- Each component gives a cycle of length $2^{\ell-1}$ in $\mathrm{dB}(q, \ell)$
- Similar methods give a partition for general q whenever $\gcd(q, \ell) = 1$
  - $\mathcal{E}(q, q^{\ell-1}, \ell) = q$

Summary

- Overall goal: investigate behaviour of eBug number $\mathcal{E}(q, k, \ell)$

Summary

- Overall goal: investigate behaviour of eBug number $\mathcal{E}(q, k, \ell)$
    - Each eBug has k LEDs with q available colours

Summary

- Overall goal: investigate behaviour of eBug number $\mathcal{E}(q, k, \ell)$
  - Each eBug has k LEDs with q available colours
  - Camera can see ℓ consecutive LEDs

Summary

- Overall goal: investigate behaviour of eBug number $\mathcal{E}(q, k, \ell)$
  - Each eBug has k LEDs with q available colours
  - Camera can see ℓ consecutive LEDs
- Equivalent to finding k-cycles in the de Bruijn graph $\mathrm{dB}(q, \ell)$

Summary

- Overall goal: investigate behaviour of eBug number $\mathcal{E}(q, k, \ell)$
  - Each eBug has k LEDs with q available colours
  - Camera can see ℓ consecutive LEDs
- Equivalent to finding k-cycles in the de Bruijn graph $\mathrm{dB}(q, \ell)$
  - $\mathcal{E}(q, k, \ell)$ maximised when $\mathrm{dB}(q, \ell)$ is partitioned into k-cycles

Summary

- Overall goal: investigate behaviour of eBug number $\mathcal{E}(q, k, \ell)$
  - Each eBug has k LEDs with q available colours
  - Camera can see ℓ consecutive LEDs
- Equivalent to finding k-cycles in the de Bruijn graph $\mathrm{dB}(q, \ell)$
  - $\mathcal{E}(q, k, \ell)$ maximised when $\mathrm{dB}(q, \ell)$ is partitioned into k-cycles
- de Bruijn graphs are Hamiltonian, so $\mathcal{E}(q, q^\ell, \ell) = 1$

Summary

- Overall goal: investigate behaviour of eBug number $\mathcal{E}(q, k, \ell)$
  - Each eBug has k LEDs with q available colours
  - Camera can see ℓ consecutive LEDs
- Equivalent to finding k-cycles in the de Bruijn graph $\mathrm{dB}(q, \ell)$
  - $\mathcal{E}(q, k, \ell)$ maximised when $\mathrm{dB}(q, \ell)$ is partitioned into k-cycles
- de Bruijn graphs are Hamiltonian, so $\mathcal{E}(q, q^\ell, \ell) = 1$
  - Called de Bruijn sequences

Summary

- Overall goal: investigate behaviour of eBug number $\mathcal{E}(q,k,\ell)$
  - Each eBug has k LEDs with q available colours
  - Camera can see ℓ consecutive LEDs
- Equivalent to finding k-cycles in the de Bruijn graph $\mathrm{dB}(q,\ell)$
  - $\mathcal{E}(q,k,\ell)$ maximised when $\mathrm{dB}(q,\ell)$ is partitioned into k-cycles
- de Bruijn graphs are Hamiltonian, so $\mathcal{E}(q,q^\ell,\ell)=1$
  - Called de Bruijn sequences
  - One big eBug is not very useful

Summary

- Overall goal: investigate behaviour of eBug number $\mathcal{E}(q, k, \ell)$
  - Each eBug has k LEDs with q available colours
  - Camera can see ℓ consecutive LEDs
- Equivalent to finding k-cycles in the de Bruijn graph $\mathrm{dB}(q, \ell)$
  - $\mathcal{E}(q, k, \ell)$ maximised when $\mathrm{dB}(q, \ell)$ is partitioned into k-cycles
- de Bruijn graphs are Hamiltonian, so $\mathcal{E}(q, q^\ell, \ell) = 1$
  - Called de Bruijn sequences
  - One big eBug is not very useful
- In some cases, partition into q $q^{\ell-1}$-cycles exists

Summary

- Overall goal: investigate behaviour of eBug number $\mathcal{E}(q,k,\ell)$
  - Each eBug has k LEDs with q available colours
  - Camera can see ℓ consecutive LEDs
- Equivalent to finding k-cycles in the de Bruijn graph $\mathrm{dB}(q,\ell)$
  - $\mathcal{E}(q,k,\ell)$ maximised when $\mathrm{dB}(q,\ell)$ is partitioned into k-cycles
- de Bruijn graphs are Hamiltonian, so $\mathcal{E}(q,q^\ell,\ell) = 1$
  - Called de Bruijn sequences
  - One big eBug is not very useful
- In some cases, partition into q $q^{\ell-1}$-cycles exists
  - Guaranteed when q and ℓ are coprime

Summary

- Overall goal: investigate behaviour of eBug number $\mathcal{E}(q,k,\ell)$
  - Each eBug has k LEDs with q available colours
  - Camera can see ℓ consecutive LEDs
- Equivalent to finding k-cycles in the de Bruijn graph $\mathrm{dB}(q,\ell)$
  - $\mathcal{E}(q,k,\ell)$ maximised when $\mathrm{dB}(q,\ell)$ is partitioned into k-cycles
- de Bruijn graphs are Hamiltonian, so $\mathcal{E}(q,q^{\ell},\ell)=1$
  - Called de Bruijn sequences
  - One big eBug is not very useful
- In some cases, partition into q $q^{\ell-1}$-cycles exists
  - Guaranteed when q and ℓ are coprime
  - Likely when q is a prime power

Summary

- Overall goal: investigate behaviour of eBug number $\mathcal{E}(q, k, \ell)$
  - Each eBug has k LEDs with q available colours
  - Camera can see ℓ consecutive LEDs
- Equivalent to finding k-cycles in the de Bruijn graph $\mathrm{dB}(q, \ell)$
  - $\mathcal{E}(q, k, \ell)$ maximised when $\mathrm{dB}(q, \ell)$ is partitioned into k-cycles
- de Bruijn graphs are Hamiltonian, so $\mathcal{E}(q, q^\ell, \ell) = 1$
  - Called de Bruijn sequences
  - One big eBug is not very useful
- In some cases, partition into q $q^{\ell-1}$-cycles exists
  - Guaranteed when q and ℓ are coprime
  - Likely when q is a prime power
  - Requires primitive polynomial with certain properties

Summary

- Overall goal: investigate behaviour of eBug number $\mathcal{E}(q, k, \ell)$
  - Each eBug has k LEDs with q available colours
  - Camera can see ℓ consecutive LEDs
- Equivalent to finding k-cycles in the de Bruijn graph $\mathrm{dB}(q, \ell)$
  - $\mathcal{E}(q, k, \ell)$ maximised when $\mathrm{dB}(q, \ell)$ is partitioned into k-cycles
- de Bruijn graphs are Hamiltonian, so $\mathcal{E}(q, q^\ell, \ell) = 1$
  - Called de Bruijn sequences
  - One big eBug is not very useful
- In some cases, partition into q $q^{\ell-1}$-cycles exists
  - Guaranteed when q and ℓ are coprime
  - Likely when q is a prime power
  - Requires primitive polynomial with certain properties
- Two eBug colourings can be multiplied to give many eBugs

Summary

- Overall goal: investigate behaviour of eBug number $\mathcal{E}(q,k,\ell)$
  - Each eBug has k LEDs with q available colours
  - Camera can see ℓ consecutive LEDs
- Equivalent to finding k-cycles in the de Bruijn graph $\mathrm{dB}(q,\ell)$
  - $\mathcal{E}(q,k,\ell)$ maximised when $\mathrm{dB}(q,\ell)$ is partitioned into k-cycles
- de Bruijn graphs are Hamiltonian, so $\mathcal{E}(q,q^{\ell},\ell)=1$
  - Called de Bruijn sequences
  - One big eBug is not very useful
- In some cases, partition into q $q^{\ell-1}$-cycles exists
  - Guaranteed when q and ℓ are coprime
  - Likely when q is a prime power
  - Requires primitive polynomial with certain properties
- Two eBug colourings can be multiplied to give many eBugs
  - Must have the same ℓ value

Summary

- Overall goal: investigate behaviour of eBug number $\mathcal{E}(q, k, \ell)$
  - Each eBug has k LEDs with q available colours
  - Camera can see ℓ consecutive LEDs
- Equivalent to finding k-cycles in the de Bruijn graph $\mathrm{dB}(q, \ell)$
  - $\mathcal{E}(q, k, \ell)$ maximised when $\mathrm{dB}(q, \ell)$ is partitioned into k-cycles
- de Bruijn graphs are Hamiltonian, so $\mathcal{E}(q, q^{\ell}, \ell) = 1$
  - Called de Bruijn sequences
  - One big eBug is not very useful
- In some cases, partition into q $q^{\ell-1}$-cycles exists
  - Guaranteed when q and ℓ are coprime
  - Likely when q is a prime power
  - Requires primitive polynomial with certain properties
- Two eBug colourings can be multiplied to give many eBugs
  - Must have the same ℓ value
  - Resulting eBugs have $q_1 q_2$ possible colours

Summary

- Overall goal: investigate behaviour of eBug number $\mathcal{E}(q,k,\ell)$
  - Each eBug has k LEDs with q available colours
  - Camera can see ℓ consecutive LEDs
- Equivalent to finding k-cycles in the de Bruijn graph $\mathrm{dB}(q,\ell)$
  - $\mathcal{E}(q,k,\ell)$ maximised when $\mathrm{dB}(q,\ell)$ is partitioned into k-cycles
- de Bruijn graphs are Hamiltonian, so $\mathcal{E}(q,q^{\ell},\ell)=1$
  - Called de Bruijn sequences
  - One big eBug is not very useful
- In some cases, partition into q $q^{\ell-1}$-cycles exists
  - Guaranteed when q and ℓ are coprime
  - Likely when q is a prime power
  - Requires primitive polynomial with certain properties
- Two eBug colourings can be multiplied to give many eBugs
  - Must have the same ℓ value
  - Resulting eBugs have $q_1 q_2$ possible colours
  - Resulting cycle length is $\mathrm{lcm}(k_1,k_2)$

Summary

- Overall goal: investigate behaviour of eBug number $\mathcal{E}(q,k,\ell)$
  - Each eBug has k LEDs with q available colours
  - Camera can see ℓ consecutive LEDs
- Equivalent to finding k-cycles in the de Bruijn graph $\mathrm{dB}(q,\ell)$
  - $\mathcal{E}(q,k,\ell)$ maximised when $\mathrm{dB}(q,\ell)$ is partitioned into k-cycles
- de Bruijn graphs are Hamiltonian, so $\mathcal{E}(q,q^\ell,\ell)=1$
  - Called de Bruijn sequences
  - One big eBug is not very useful
- In some cases, partition into q $q^{\ell-1}$-cycles exists
  - Guaranteed when q and ℓ are coprime
  - Likely when q is a prime power
  - Requires primitive polynomial with certain properties
- Two eBug colourings can be multiplied to give many eBugs
  - Must have the same ℓ value
  - Resulting eBugs have q_1q_2 possible colours
  - Resulting cycle length is $\mathrm{lcm}(k_1,k_2)$
  - Each pair of eBugs gives $\gcd(k_1,k_2)$ new eBugs

Summary

Minimum k for which $\mathcal{E}(q,k,\ell) = \frac{q^\ell}{k}$ guaranteed

$\boldsymbol{\ell}$	$\boldsymbol{q=2}$	$\boldsymbol{q=3}$	$\boldsymbol{q=4}$	$\boldsymbol{q=6}$	$\boldsymbol{q=12}$
1	1	1	1	1	1
2	4	3	4	12	12
3	4	27	4	108	108
4	16	27	16	432	432
5	16	81	16	1296	1296
6	64	729	64	46656	46656
7	64	729	64	46656	46656