Successively Solvable Shift-Add Systems — a Graphical Characterization

Xiaopeng Cheng, Ximing Fu, Yuanxin Guo,
Kenneth W. Shum, and Shenghao Yang

The Chinese University of Hong Kong, Shenzhen
ISIT 2021

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

ISIT 2021

Overview

- Motivation: Coding theory — Shift-XOR coding

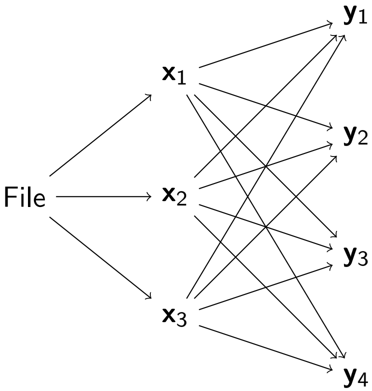
Overview

- Motivation: Coding theory — Shift-XOR coding
- Low complexity: only (non-cyclic) shift and XOR operations involved

Overview

- Motivation: Coding theory — Shift-XOR coding
- Low complexity: only (non-cyclic) shift and XOR operations involved
- Applications:
  - Storage codes: Sung & Gong '13, Fu et al. '14, Dai et al. '17
  - Regenerating codes: Hou et al. '13, Fu et al. '15
  - Fountain codes: Nozaki '14, Jun et al. '17
  - Network codes: Sung & Gong '14

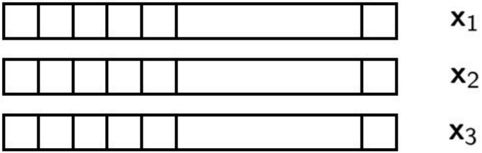
Toy example: storage coding



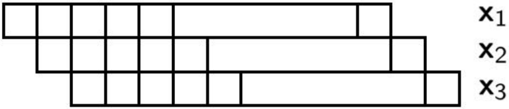
A storage system with 4 storage nodes that can tolerate one node failure.

Toy example: shift-XOR coding

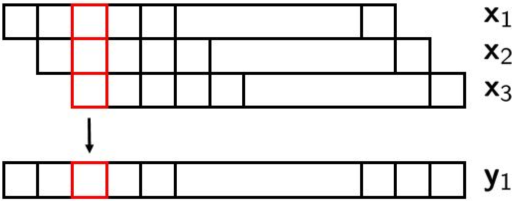
Toy example: shift-XOR coding



Toy example: shift-XOR coding

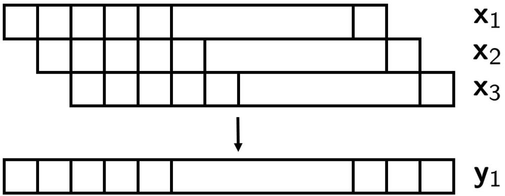


Toy example: shift-XOR coding



$$\mathbf{y}_1[2] = \mathbf{x}_1[2] + \mathbf{x}_2[1] + \mathbf{x}_3[0]$$

Toy example: shift-XOR coding



$$\mathbf{y}_1 = \mathbf{x}_1 + z\mathbf{x}_2 + z^2\mathbf{x}_3 = \begin{bmatrix} 1 & z & z^2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 \end{bmatrix}^\top$$

where z^t is the shift operator, defined as

$$(z^t\mathbf{s})[\ell] = \begin{cases} \mathbf{s}[\ell - t], & \ell \geq \max\{t, 0\}, \\ 0, & \text{otherwise.} \end{cases}$$

- $t < 0$: discard/truncate the first $(-t)$ symbols
- $t = \infty$: $\mathbf{x}_i$ does not involve in the forming of $\mathbf{y}_1$

Shift-add system

- Model data symbols as elements of a finite abelian group $(\mathcal{A}, +)$. A sequence **s** is a mapping $\mathbb{Z} \to \mathcal{A}$ satisfying $\mathbf{s}[i] = 0$ for $i < 0$.

Shift-add system

- Model data symbols as elements of a finite abelian group $(\mathcal{A}, +)$.
  A sequence $\mathbf{s}$ is a mapping $\mathbb{Z} \to \mathcal{A}$ satisfying $\mathbf{s}[i] = 0$ for $i < 0$.
- Consider a *shift-add system* with n sequences $\{\mathbf{x}_i\}_{i=1}^{n}$ as input and m sequences $\{\mathbf{y}_j\}_{j=1}^{m}$ as output.

Shift-add system

- Model data symbols as elements of a finite abelian group $(\mathcal{A}, +)$. A sequence $\mathbf{s}$ is a mapping $\mathbb{Z} \to \mathcal{A}$ satisfying $\mathbf{s}[i] = 0$ for $i < 0$.
- Consider a *shift-add system* with n sequences $\{\mathbf{x}_i\}_{i=1}^n$ as input and m sequences $\{\mathbf{y}_j\}_{j=1}^m$ as output.
- The output sequences are obtained by

$$\mathbf{y}_i = \sum_{j=1}^{n} z^{a_{ij}} \mathbf{x}_j$$

where the addition is performed in the additive group $\mathcal{A}$, and the shift operator z^t is defined as in the last slide:

$$(z^t \mathbf{s})[\ell] = \begin{cases} \mathbf{s}[\ell - t], & \ell \geq \max\{t, 0\}, \\ 0, & \text{otherwise.} \end{cases}$$

Shift-add system

- Model data symbols as elements of a finite abelian group $(\mathcal{A}, +)$. A sequence $\mathbf{s}$ is a mapping $\mathbb{Z} \to \mathcal{A}$ satisfying $\mathbf{s}[i] = 0$ for $i < 0$.
- Consider a *shift-add system* with n sequences $\{\mathbf{x}_i\}_{i=1}^n$ as input and m sequences $\{\mathbf{y}_j\}_{j=1}^m$ as output.
- The output sequences are obtained by

$$\mathbf{y}_i = \sum_{j=1}^{n} z^{a_{ij}} \mathbf{x}_j$$

  where the addition is performed in the additive group $\mathcal{A}$, and the shift operator z^t is defined as in the last slide:

$$(z^t \mathbf{s})[\ell] = \begin{cases} \mathbf{s}[\ell - t], & \ell \geq \max\{t, 0\}, \\ 0, & \text{otherwise.} \end{cases}$$

- The matrices $\Phi = (z^{a_{ij}})$ and $A = (a_{ij})$ are both called *shift matrices*.

Infinite linear system

- Motivation: extend the definition in order to develop a general notion of solvability.

Infinite linear system

- Motivation: extend the definition in order to develop a general notion of solvability.
- An *infinite linear system* takes sequence $\mathbf{v}$ as input and outputs sequence $\mathbf{u}$, with

$$\mathbf{u}[j] = \sum_{i \in E_j} \mathbf{v}[i], \; \forall j \in \mathbb{N}_0$$

  where $\mathbb{N}_0 = \{n \in \mathbb{Z}, n \geq 0\}$, and $E_j \subseteq N_0$ is some finite index set for each $j \in \mathbb{N}_0$.

Infinite linear system: example

Example 1 (Shift-add system is an infinite linear system).

- Recall a shift-add system takes the form

$$\mathbf{y}_i = \sum_{j=1}^{n} z^{a_{ij}} \mathbf{x}_j \quad \text{or} \quad \mathbf{y}_i[\ell] = \sum_{j=1}^{n} \mathbf{x}_j[\ell - a_{ij}] = \sum_{j \in [n],\, \ell \geq a_{ij}} \mathbf{x}_j[\ell - a_{ij}]$$

Infinite linear system: example

Example 1 (Shift-add system is an infinite linear system).

- Recall a shift-add system takes the form

$$\mathbf{y}_i = \sum_{j=1}^{n} z^{a_{ij}} \mathbf{x}_j \quad \text{or} \quad \mathbf{y}_i[\ell] = \sum_{j=1}^{n} \mathbf{x}_j[\ell - a_{ij}] = \sum_{j \in [n],\, \ell \geq a_{ij}} \mathbf{x}_j[\ell - a_{ij}]$$

- Rearrange the entries in the input sequences $\mathbf{x}_j$ (resp. output sequences $\mathbf{y}_i$) into a single sequence $\mathbf{v}$ (resp. $\mathbf{u}$)., given by

$$\mathbf{v}[n\ell + j - 1] = \mathbf{x}_j[\ell], \qquad \mathbf{u}[m\ell + i - 1] = \mathbf{y}_i[\ell]$$

Infinite linear system: example

Example 1 (Shift-add system is an infinite linear system).

- Recall a shift-add system takes the form

$$\mathbf{y}_i = \sum_{j=1}^{n} z^{a_{ij}} \mathbf{x}_j \quad \text{or} \quad \mathbf{y}_i[\ell] = \sum_{j=1}^{n} \mathbf{x}_j[\ell - a_{ij}] = \sum_{j \in [n],\, \ell \geq a_{ij}} \mathbf{x}_j[\ell - a_{ij}]$$

- Rearrange the entries in the input sequences $\mathbf{x}_j$ (resp. output sequences $\mathbf{y}_i$) into a single sequence $\mathbf{v}$ (resp. $\mathbf{u}$)., given by

$$\mathbf{v}[n\ell + j - 1] = \mathbf{x}_j[\ell], \qquad \mathbf{u}[m\ell + i - 1] = \mathbf{y}_i[\ell]$$

- Then the shift-add system can be rewritten in the notation of an infinite linear system as

$$\mathbf{u}[m\ell + i - 1] = \sum_{k \in E_{m\ell+i-1}} \mathbf{v}[k], \quad E_{m\ell+i-1} = \mathbb{N}_0 \cap \bigcup_{i=1}^{n} \{n(\ell - a_{ij}) + j - 1\}$$

Solvability of an infinite linear system

- Analytic viewpoint: an infinite linear system is *solvable* if the mapping $\mathbf{v} \mapsto \mathbf{u}$ is injective.

Solvability of an infinite linear system

- Analytic viewpoint: an infinite linear system is *solvable* if the mapping $\mathbf{v} \mapsto \mathbf{u}$ is injective.
- Algorithmic viewpoint: a notion of solvability that ensures an efficient solving algorithm.

Solvability of an infinite linear system

- Analytic viewpoint: an infinite linear system is *solvable* if the mapping $\mathbf{v} \mapsto \mathbf{u}$ is injective.
- Algorithmic viewpoint: a notion of solvability that ensures an efficient solving algorithm.

Definition 1 (Zigzag solvable).

An infinite linear system is *zigzag solvable* if there exists two functions $f, g : \mathbb{N}_0 \to \mathbb{N}_0$ such that

1. f is bijective;
2. $\{f(i)\} \subseteq E_{g(i)} \subseteq \bigcup_{\ell=0}^{i} \{f(i)\}$, for all $i \in \mathbb{N}_0$.

Zigzag solvability: intuition

- Zigzag solvability ensures a solving algorithm called *zigzag decoding*.
- In each of the i-th stage ($i \in \mathbb{N}_0$), we use a symbol in sequence $\mathbf{u}$ to solve for a symbol in sequence $\mathbf{v}$.
- $f(i)$ indicates the index of the symbol in $\mathbf{v}$ solved in the i-th stage.
- $g(i)$ indicates the index of the symbol in $\mathbf{u}$ used in the i-th stage.

Zigzag solvability: intuition

- Zigzag solvability ensures a solving algorithm called *zigzag decoding*.
- In each of the i-th stage ($i \in \mathbb{N}_0$), we use a symbol in sequence $\mathbf{u}$ to solve for a symbol in sequence $\mathbf{v}$.
- $f(i)$ indicates the index of the symbol in $\mathbf{v}$ solved in the i-th stage.
- $g(i)$ indicates the index of the symbol in $\mathbf{u}$ used in the i-th stage.

- Condition 1 guarantees that every symbol in $\mathbf{v}$ is solved exactly once.
- Condition 2 guarantees that the substitution algorithm can proceed, i.e., $\mathbf{u}[g(i)]$ can be written as the sum of $\mathbf{v}[f(i)]$ and input symbols that are solved prior to the i-th stage.

Successive solvability

- Motivation: determining f and g is generally hard.
- Aim: develop a stronger notion of solvability that puts some constraints on f and g.

Successive solvability

- Motivation: determining f and g is generally hard.
- Aim: develop a stronger notion of solvability that puts some constraints on f and g.
- Focus on shift-add systems.
- Abusing notation: use (j, ℓ) in place of $(n\ell + j - 1)$ for the index of $\mathbf{x}_j[\ell]$ in $\mathbf{v}$, and (i, ℓ) in place of $(m\ell + i - 1)$ for the index of $\mathbf{y}_i[\ell]$ in $\mathbf{u}$.

Successive solvability

- Motivation: determining f and g is generally hard.
- Aim: develop a stronger notion of solvability that puts some constraints on f and g.
- Focus on shift-add systems.
- Abusing notation: use (j, ℓ) in place of $(n\ell + j - 1)$ for the index of $\mathbf{x}_j[\ell]$ in $\mathbf{v}$, and (i, ℓ) in place of $(m\ell + i - 1)$ for the index of $\mathbf{y}_i[\ell]$ in $\mathbf{u}$.

Definition 2 (Successively solvable).

A shift-add system is *successively solvable* if it is zigzag solvable with functions f, g, and satisfies an additional condition:

3. $f^{-1}(j, \ell) < f^{-1}(j, \ell')$ whenever $0 \leq \ell < \ell'$, for all $1 \leq j \leq n$.

Successive solvability: intuition

- Successive solvability guarantees that the symbols of each variable sequence are solved **successively** from left to right.
- This adds regularity to the solving algorithm, as in each stage, the function f takes value in the index of the first unsolved symbol of each variable sequence.

Successive solvability: intuition

- Successive solvability guarantees that the symbols of each variable sequence are solved **successively** from left to right.
- This adds regularity to the solving algorithm, as in each stage, the function f takes value in the index of the first unsolved symbol of each variable sequence.
- For shift-add systems, zigzag solvable does NOT implies successively solvable.

Zigzag solvability and successive solvability

Example 2 (Zigzag solvable $\not\Rightarrow$ successively solvable).

- Consider the shift-add system with shift matrix

$$A = \begin{pmatrix} 0 & 0 & \infty \\ \infty & -1 & \infty \\ -1 & 0 & 1 \end{pmatrix}$$

- The system is zigzag solvable but not successively solvable.
- In the 0-th stage, can only decode $\mathbf{x}_2[\ell]$ for some $\ell \geq 1$.
- $f^{-1}(2,0) > f^{-1}(2,\ell)$, violating successive solvability.

Zigzag solvability and successive solvability

Example 2 (Zigzag solvable $\not\Rightarrow$ successively solvable).

- Consider the shift-add system with shift matrix

$$A = \begin{pmatrix} 0 & 0 & \infty \\ \infty & -1 & \infty \\ -1 & 0 & 1 \end{pmatrix}$$

- The system is zigzag solvable but not successively solvable.
- In the 0-th stage, can only decode $\mathbf{x}_2[\ell]$ for some $\ell \geq 1$.
- $f^{-1}(2,0) > f^{-1}(2,\ell)$, violating successive solvability.

Proposition 1.

For a shift-add system corresponding to a **non-negative** *shift matrix, it is zigzag solvable if and only if it is successively solvable.*

Reduction of shift-add systems

- Focus on shift-add systems with square shift matrices.

Reduction of shift-add systems

- Focus on shift-add systems with square shift matrices.
- Consider the 0-th stage of a zigzag decoding algorithm.
- Suppose there exists i, j such that $a_{ij} = 0$ and $a_{ik} > 0$ for all $k \neq j$. We have $\mathbf{y}_i[0] = \mathbf{x}_j[0]$. Let $f(0) = (j, 0)$, $g(0) = (i, 0)$.

Reduction of shift-add systems

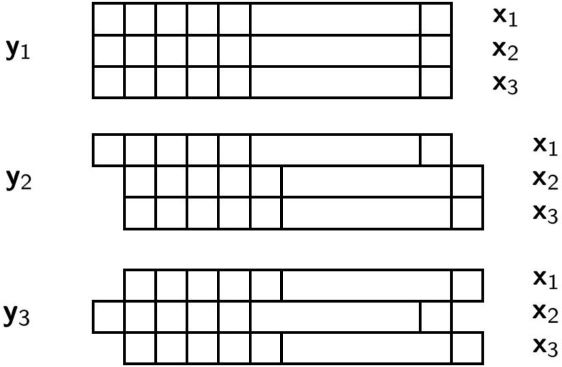
- Focus on shift-add systems with square shift matrices.

- Consider the 0-th stage of a zigzag decoding algorithm.
- Suppose there exists i, j such that $a_{ij} = 0$ and $a_{ik} > 0$ for all $k \neq j$. We have $\mathbf{y}_i[0] = \mathbf{x}_j[0]$. Let $f(0) = (j, 0)$, $g(0) = (i, 0)$.

- A decoding stage can be seen as a reduction of a shift-add system.
- Let $\mathbf{x}'_j = z^{-1}\mathbf{x}_j$, i.e., $\mathbf{x}'_j[t] = \mathbf{x}_j[t+1]$ for $t \geq 0$.
- Let $\mathbf{y}'_i = z^{-1}\mathbf{y}_i$, i.e., $\mathbf{y}'_i[t] = \mathbf{y}_i[t+1]$ for $t \geq 0$.
- For $1 \leq k \leq n$, $k \neq i$, let $\mathbf{y}'_k$ be such that

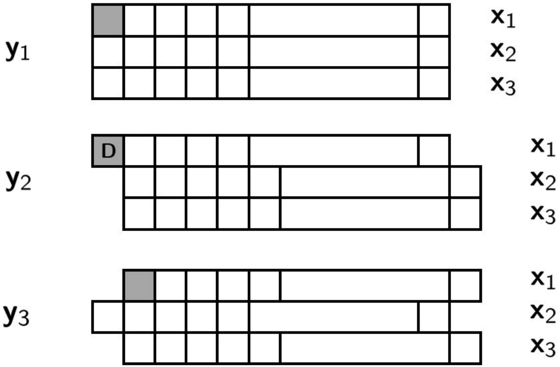
$$\mathbf{y}'_k[t] = \begin{cases} \mathbf{y}_k[t] - \mathbf{x}_j[0], & \text{if } t = a_{kj}, \\ \mathbf{y}_k[t], & \text{otherwise,} \end{cases} \qquad \text{for } t \geq 0.$$

Reduction: example



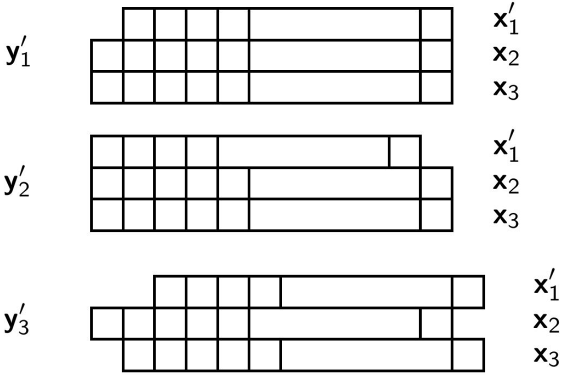
Shift matrix: $A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$

Reduction: example



Shift matrix: $A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$

Reduction: example



Shift matrix: $A' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 2 & 0 & 1 \end{pmatrix}$

Definitions on shift matrix

Definition 3 (Pivot, reductive).

Given a shift matrix A, the (i, j)-th entry is called a *pivot* if (i) $a_{ij} \geq 0$, and (ii) $a_{ij} < a_{ik}$ for all $k \neq j$. We say that a shift matrix A is *reductive* if A contains a pivot.

Definitions on shift matrix

Definition 3 (Pivot, reductive).

Given a shift matrix A, the (i,j)-th entry is called a *pivot* if (i) $a_{ij} \geq 0$, and (ii) $a_{ij} < a_{ik}$ for all $k \neq j$. We say that a shift matrix A is *reductive* if A contains a pivot.

Definition 4 (Equivalent).

Denote the i-th row of matrix A by $\mathbf{a}_i$ and the all-one row vector by $\mathbf{1}$. Two shift matrices A and B are *equivalent*, or $A \sim B$, if $\mathbf{a}_i - \mathbf{b}_i = c_i \cdot \mathbf{1}$, where $c_i = 0$ whenever $\exists\, j$ s.t. $a_{ij} < 0$ or $b_{ij} < 0$.

Definitions on shift matrix

Definition 3 (Pivot, reductive).

Given a shift matrix A, the (i,j)-th entry is called a *pivot* if (i) $a_{ij} \geq 0$, and (ii) $a_{ij} < a_{ik}$ for all $k \neq j$. We say that a shift matrix A is *reductive* if A contains a pivot.

Definition 4 (Equivalent).

Denote the i-th row of matrix A by $\mathbf{a}_i$ and the all-one row vector by $\mathbf{1}$. Two shift matrices A and B are *equivalent*, or $A \sim B$, if $\mathbf{a}_i - \mathbf{b}_i = c_i \cdot \mathbf{1}$, where $c_i = 0$ whenever $\exists\, j$ s.t. $a_{ij} < 0$ or $b_{ij} < 0$.

Definition 5 (Reducing operator).

The (i,j)-th *reducing operator* R_{ij} is defined by $(R_{ij})_{k\ell} = \delta_{j,\ell} - \delta_{i,k}$, where $\delta_{\cdot,\cdot}$ is the Kronecker's delta function.

Graphical construction: Intuition

- Successive zigzag decoding algorithm $\Leftrightarrow$ Infinite sequence of reducing operators

Graphical construction: Intuition

- Successive zigzag decoding algorithm $\Leftrightarrow$ Infinite sequence of reducing operators
- Hypothesis: recurrent pattern in the infinite sequence

Graphical construction: Intuition

- Successive zigzag decoding algorithm $\Leftrightarrow$ Infinite sequence of reducing operators
- Hypothesis: recurrent pattern in the infinite sequence
- Graphical characterization: intuition
  - Shift matrix $\leftrightarrow$ vertex
  - Reducing operator $\leftrightarrow$ edge
  - Sequence of operators $\leftrightarrow$ infinite walk
  - Recurrent pattern $\leftrightarrow$ walk on cycle

Graphical construction: Intuition

- Successive zigzag decoding algorithm $\Leftrightarrow$ Infinite sequence of reducing operators
- Hypothesis: recurrent pattern in the infinite sequence
- Graphical characterization: intuition
  - Shift matrix $\leftrightarrow$ vertex
  - Reducing operator $\leftrightarrow$ edge
  - Sequence of operators $\leftrightarrow$ infinite walk
  - Recurrent pattern $\leftrightarrow$ walk on cycle

Lemma 2.

Suppose the (i,j)-th entry of an $n \times n$ shift matrix A is a pivot. If $A' \sim A$, then the (i,j)-th entry is a pivot of A', and $(R_{ij} + A) \sim (R_{ij} + A')$.

Definitions on $\mathcal{G}_n$

Definition 6 ($\mathcal{G}_n$).

For a fixed positive integer n, we define a directed (multi)graph $\mathcal{G}_n$ whose vertices are equivalence classes of shift matrices with finite entries. Given two shift matrices A and B with finite entries, there is a directed edge from $\{A\}$ to $\{B\}$ (written $\{A\} \xrightarrow{R_{ij}} \{B\}$) if A has a pivot (i,j) and $R_{ij} + A \sim B$.

Definitions on $\mathcal{G}_n$

Definition 6 ($\mathcal{G}_n$).

For a fixed positive integer n, we define a directed (multi)graph $\mathcal{G}_n$ whose vertices are equivalence classes of shift matrices with finite entries. Given two shift matrices A and B with finite entries, there is a directed edge from $\{A\}$ to $\{B\}$ (written $\{A\} \xrightarrow{R_{ij}} \{B\}$) if A has a pivot (i, j) and $R_{ij} + A \sim B$.

Definition 7 (Path, cycle).

For positive integer L, we define a *path* of length L in $\mathcal{G}_n$ as a sequence of edges $\{A_k\} \xrightarrow{R_{i_k j_k}} \{B_k\}$, $k = 1, 2, \cdots, L$, where $B_k \sim A_{k+1}$ for $k = 1, 2, \cdots, L-1$. When $B_L \sim A_1$, the path is called a *cycle*.

Main result

Theorem 8.

An $n \times n$ shift-add system defined by shift matrix A with only finite entries is successively solvable if and only if there exists $\{B\}$ in a cycle of $\mathcal{G}_n$ and there exists a path from $\{A\}$ to $\{B\}$.

Main result

Theorem 8.

An $n \times n$ shift-add system defined by shift matrix A with only finite entries is successively solvable if and only if there exists $\{B\}$ in a cycle of $\mathcal{G}_n$ and there exists a path from $\{A\}$ to $\{B\}$.

Intuition: if a shift-add system is solvable, the infinite sequence of reducing operators ends up repeating a recurrent pattern.

Future work

- Infinity entries?
- Rectangular matrices?
- Algorithm for successive solvability?

Future work

Thank you!