Cyclic Codes and Cellular Automata

Journées Calculabilités 2016

Luca Mariot[1,2]

[1] Dipartimento di Informatica, Sistemistica e Comunicazione (DISCo)
Università degli Studi Milano - Bicocca

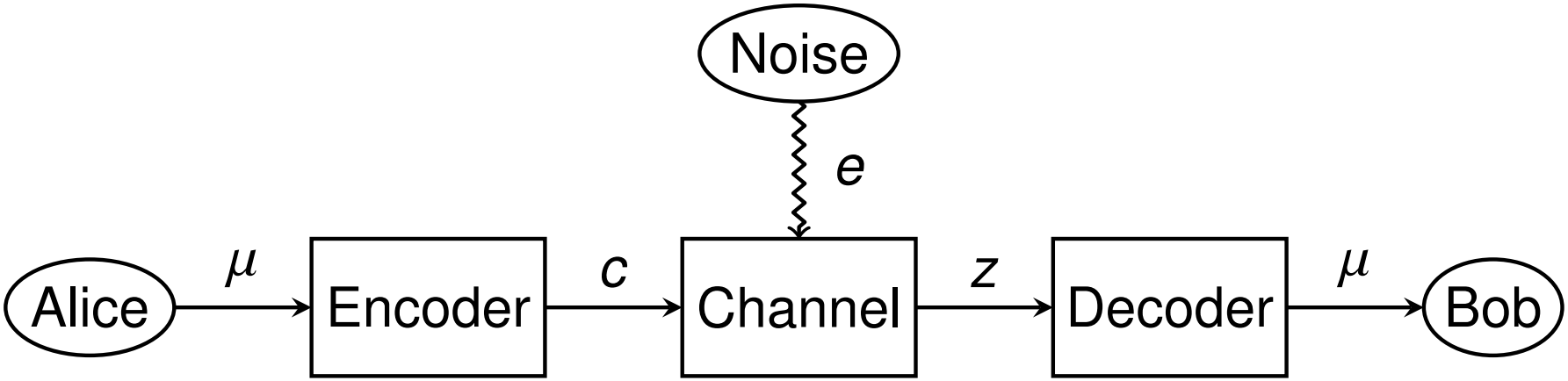
[2] Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis (I3S)
Université Nice Sophia Antipolis

luca.mariot@disco.unimib.it

April 12, 2016

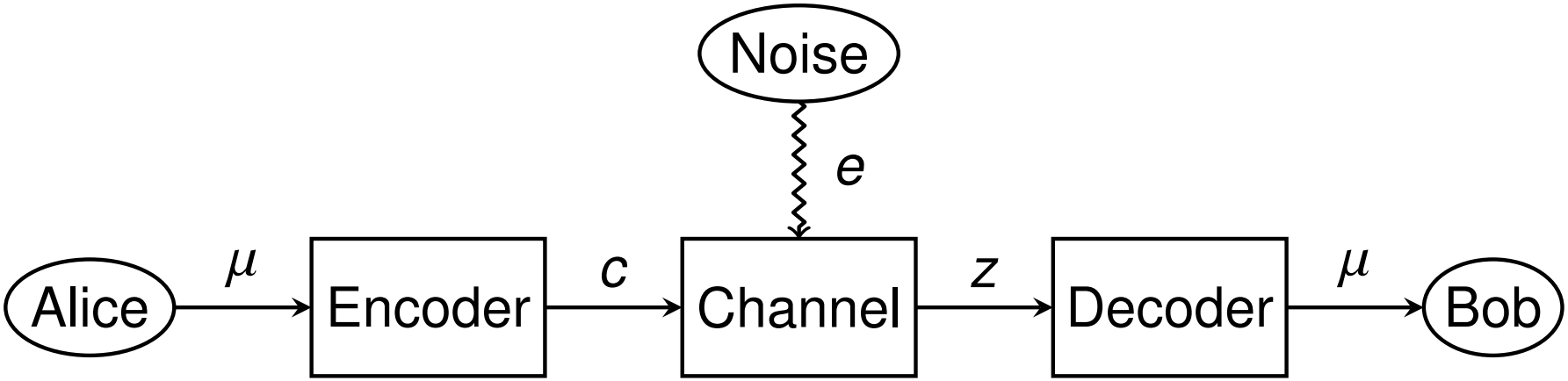
Part 1: Error-Correcting Codes Basics

Communication Model



- $\mu \in \{0,1\}^m$: message
- $c \in \{0,1\}^n$: codeword ($n > m$)
- $e \in \{0,1\}^n$: error pattern
- $z = c \oplus e$ (received word)

Communication Model



- $\mu \in \{0,1\}^m$: message
- $c \in \{0,1\}^n$: codeword ($n > m$)
- $e \in \{0,1\}^n$: error pattern
- $z = c \oplus e$ (received word)

ASSUMPTION: at most $t < \left\lfloor \frac{n}{2} \right\rfloor$ errors $\Rightarrow w_H(e) \leq t$

Linear Codes

Definition

A (n, m, d) binary linear code C of minimum distance d is an m-dimensional subspace of $\mathbb{F}_2^n$, such that for all distinct $c_1, c_2 \in C$

$$d_H(c_1, c_2) \geq d$$

Linear Codes

Definition

A (n, m, d) binary linear code C of minimum distance d is an m-dimensional subspace of $\mathbb{F}_2^n$, such that for all distinct $c_1, c_2 \in C$

$$d_H(c_1, c_2) \geq d$$

$$g_1, \cdots, g_m \in \mathbb{F}_2^n \text{ basis of } C \Leftrightarrow G = \begin{pmatrix} g_1 \\ \vdots \\ g_m \end{pmatrix} m \times n \text{ generator matrix of } C$$

Linear Codes

Definition

A (n, m, d) binary linear code C of minimum distance d is an m-dimensional subspace of $\mathbb{F}_2^n$, such that for all distinct $c_1, c_2 \in C$

$$d_H(c_1, c_2) \geq d$$

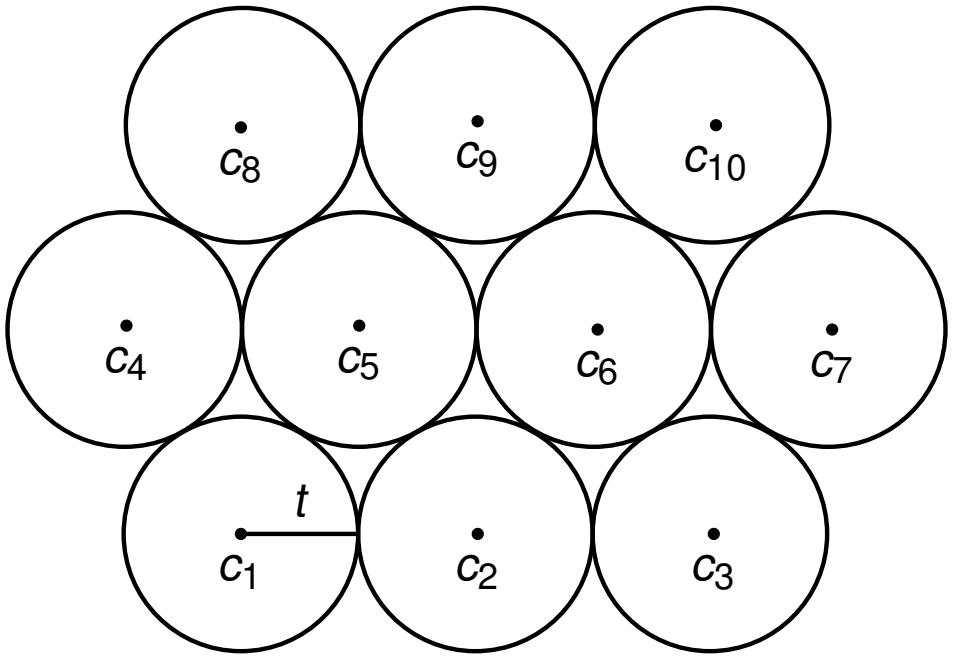
$$g_1, \cdots, g_m \in \mathbb{F}_2^n \text{ basis of } C \Leftrightarrow G = \begin{pmatrix} g_1 \\ \vdots \\ g_m \end{pmatrix} m \times n \text{ generator matrix of } C$$

Encoding: vector-matrix multiplication

$$\mu \mapsto c = \mu G$$

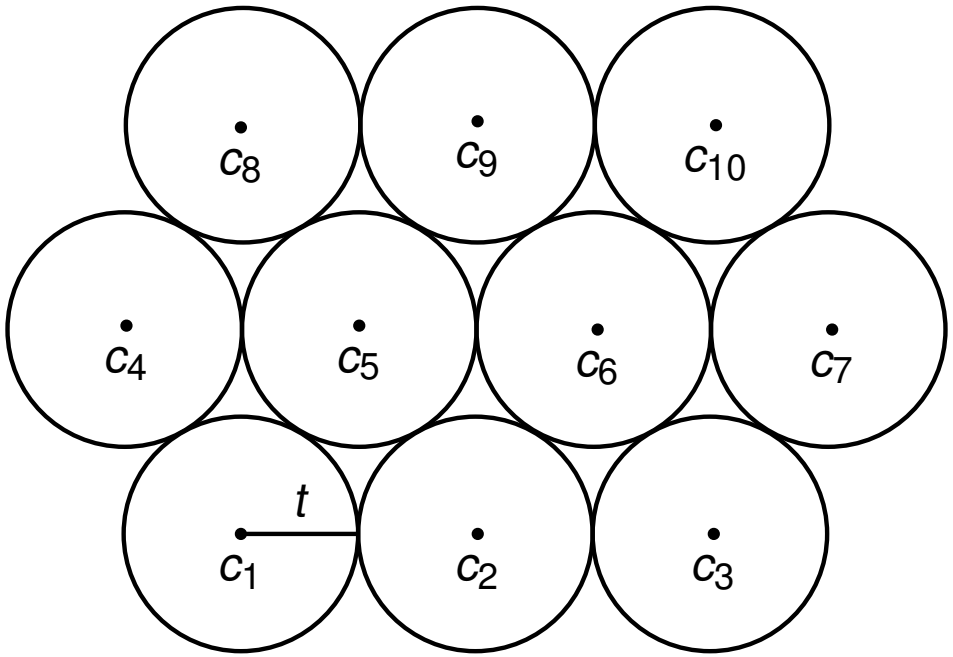
Error Correction (Sphere Shrinking – SS)

- $t = \left\lfloor \frac{d-1}{2} \right\rfloor \Leftrightarrow$ Error-correction capability of C
- Sphere of $c \in C \Leftrightarrow S_c = \{z \in \mathbb{F}_2^n : \ d_H(z, c) \leq t\}$



Error Correction (Sphere Shrinking – SS)

- $t = \left\lfloor \frac{d-1}{2} \right\rfloor \Leftrightarrow$ Error-correction capability of C
- Sphere of $c \in C \Leftrightarrow S_c = \{z \in \mathbb{F}_2^n : \ d_H(z, c) \leq t\}$



SS-Decoding: return the nearest codeword $c \in C$ to $z \in \mathbb{F}_2^n$

Error Correction – Syndrome Decoding

- Parity Check Matrix: a $(n-m)\times n$ matrix H such that

$$s = Hz^{\top} = 0 \Leftrightarrow z \in C$$

s: Syndrome of z

Error Correction – Syndrome Decoding

- Parity Check Matrix: a $(n-m)\times n$ matrix H such that

$$s = Hz^\top = 0 \Leftrightarrow z \in C$$

  s: Syndrome of z

- Suppose $z = c \oplus e$, $c \in C$ and $e \in \mathbb{F}_2^n$. Then

$$Hz^\top = H(c \oplus e)^\top = \cancel{Hc^\top} \oplus He^\top = He^\top$$

Error Correction – Syndrome Decoding

- Parity Check Matrix: a $(n-m)\times n$ matrix H such that

$$s = Hz^\top = 0 \Leftrightarrow z \in C$$

  s: Syndrome of z

- Suppose $z = c \oplus e$, $c \in C$ and $e \in \mathbb{F}_2^n$. Then

$$Hz^\top = H(c \oplus e)^\top = \cancel{Hc^\top} \oplus He^\top = He^\top$$

Syndrome Decoding: find $e \in \mathbb{F}_2^n$ and return $c = z \oplus e$

Cyclic Codes

Definition

A (n, m, d) linear code is cyclic if it is closed under cyclic shifts, i.e. for all $c = (c_0, c_1, \cdots, c_{n-1}) \in C$

$$\sigma(c) = (c_1, \cdots, c_{n-1}, c_0) \in C$$

- Polynomial representation:

$$\mu = (\mu_0, \cdots, \mu_{m-1}) \mapsto \mu(X) = \mu_0 + \mu_1 X + \cdots + \mu_{m-1} X^{m-1}$$

Generator and Parity Check Polynomials

- Generator polynomial: $g(X) = g_0 + g_1 X + \cdots + g_{n-m} X^{n-m}$

Encoding: $\mu(X) \mapsto c(X) = \mu(X) g(X)$

Generator and Parity Check Polynomials

- Generator polynomial: $g(X) = g_0 + g_1 X + \cdots + g_{n-m} X^{n-m}$

Encoding: $\mu(X) \mapsto c(X) = \mu(X) g(X)$

- Parity-check polynomial: $h(X) = (X^n - 1)/g(X)$

Syndrome: $s(X) = z(X) h(X) = 0 \Leftrightarrow z \in C$

Generator and Parity Check Polynomials

- Generator polynomial: $g(X) = g_0 + g_1 X + \cdots + g_{n-m} X^{n-m}$

Encoding: $\mu(X) \mapsto c(X) = \mu(X) g(X)$

- Parity-check polynomial: $h(X) = (X^n - 1)/g(X)$

Syndrome: $s(X) = z(X) h(X) = 0 \Leftrightarrow z \in C$

Cyclic codes of length n $\Leftrightarrow$ Divisors of $X^n - 1$

Generator and Parity Check Matrices of Cyclic Codes

- Generator polynomial: $g(X) = g_0 + g_1 X + \cdots + g_{n-m} X^{n-m}$

$$G = \begin{pmatrix} g_0 & \cdots & g_{n-m} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & g_0 & \cdots & g_{n-m} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & g_0 & \cdots & g_{n-m} \end{pmatrix}$$

Generator and Parity Check Matrices of Cyclic Codes

- Generator polynomial: $g(X) = g_0 + g_1 X + \cdots + g_{n-m} X^{n-m}$

$$G = \begin{pmatrix} g_0 & \cdots & g_{n-m} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & g_0 & \cdots & g_{n-m} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & g_0 & \cdots & g_{n-m} \end{pmatrix}$$

- Parity-check polynomial: $h(X) = h_0 + h_1 X + \cdots + h_m X^m$

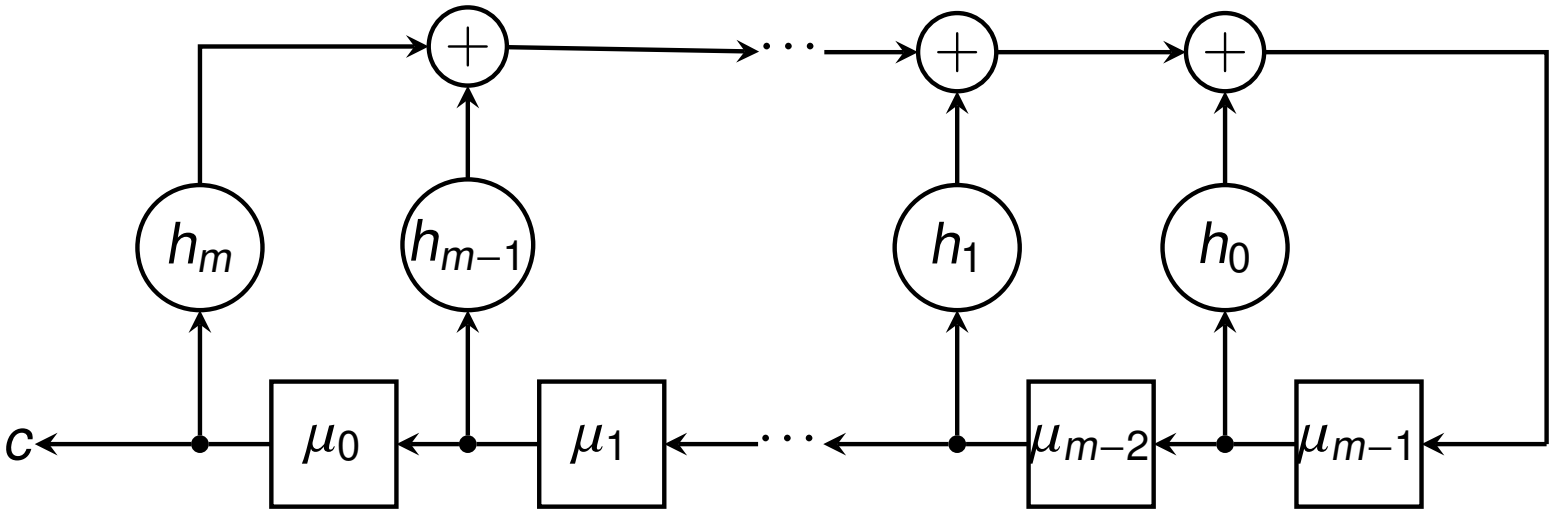
$$H = \begin{pmatrix} h_m & \cdots & h_0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & h_m & \cdots & h_0 & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & h_m & \cdots & h_0 \end{pmatrix}$$

Systematic Encoding through LFSR [McEl85]

- Parity-check polynomial: $h(X) = h_0 + h_1 X + \cdots + h_m X^m$
- Reciprocal: $h^*(X) = X^m h(1/X) = h_m + h_{m-1} X + \cdots + h_0 X^m$

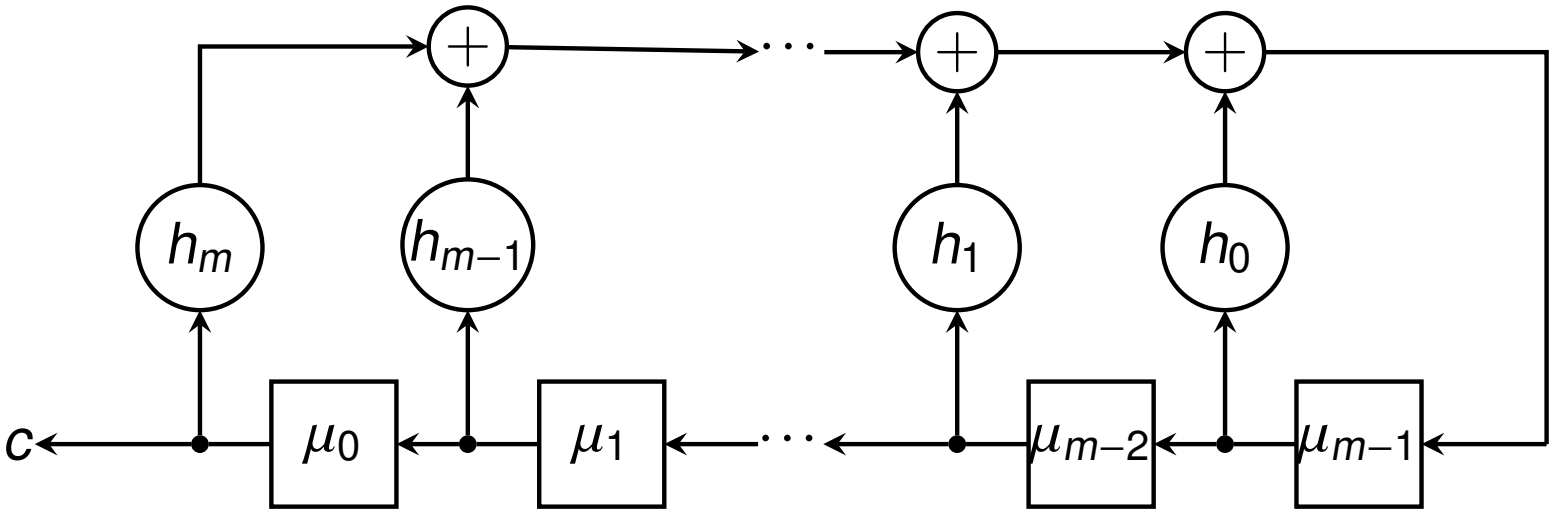
Systematic Encoding through LFSR [McEl85]

- Parity-check polynomial: $h(X) = h_0 + h_1 X + \cdots + h_m X^m$
- Reciprocal: $h^*(X) = X^m h(1/X) = h_m + h_{m-1} X + \cdots + h_0 X^m$



Systematic Encoding through LFSR [McEl85]

- Parity-check polynomial: $h(X) = h_0 + h_1 X + \cdots + h_m X^m$
- Reciprocal: $h^*(X) = X^m h(1/X) = h_m + h_{m-1} X + \cdots + h_0 X^m$



$$c = (\underbrace{\mu_0, \cdots, \mu_{m-1}}_{\text{original message}}, \underbrace{p_0, \cdots, p_{n-m-1}}_{\text{parity check bits}})$$

Part 2: Cellular Automata

One-Dimensional Cellular Automata (CA)

Definition

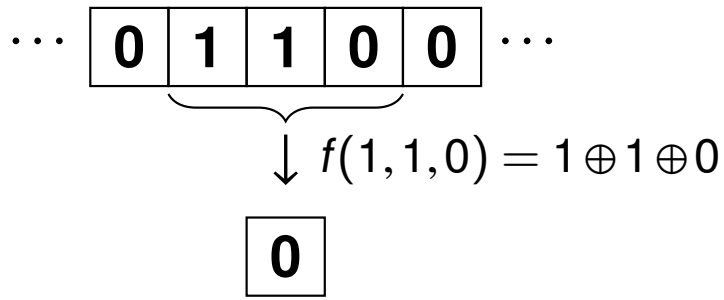
One-dimensional cellular automaton: triple $\langle n, \delta, f \rangle$ where $n \in \mathbb{N}$ is the number of cells arranged on a one-dimensional array, $r \in \mathbb{N}$ is the diameter and $f : \{0,1\}^{\delta} \to \{0,1\}$ is the local rule.

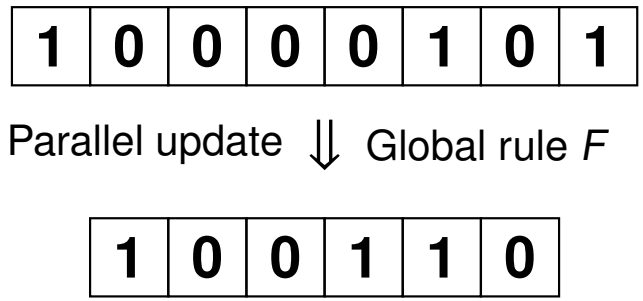
One-Dimensional Cellular Automata (CA)

Definition

One-dimensional cellular automaton: triple $\langle n, \delta, f \rangle$ where $n \in \mathbb{N}$ is the number of cells arranged on a one-dimensional array, $r \in \mathbb{N}$ is the diameter and $f : \{0,1\}^{\delta} \to \{0,1\}$ is the local rule.

Example: $n = 8$, $\delta = 3$, $f(s_{i-1}, s_i, s_{i+1}) = s_{i-1} \oplus s_i \oplus s_{i+1}$



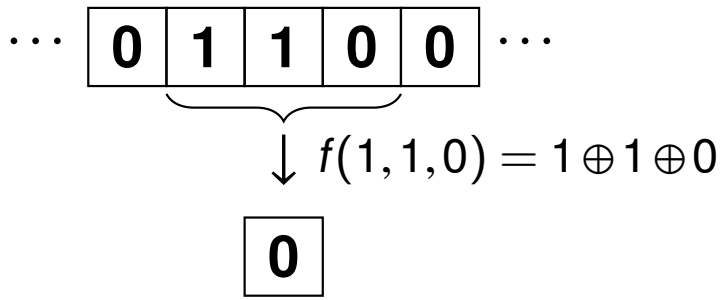


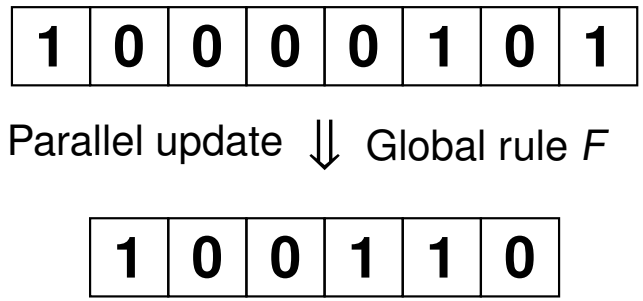
One-Dimensional Cellular Automata (CA)

Definition

One-dimensional cellular automaton: triple $\langle n, \delta, f \rangle$ where $n \in \mathbb{N}$ is the number of cells arranged on a one-dimensional array, $r \in \mathbb{N}$ is the diameter and $f : \{0,1\}^{\delta} \to \{0,1\}$ is the local rule.

Example: $n = 8$, $\delta = 3$, $f(s_{i-1}, s_i, s_{i+1}) = s_{i-1} \oplus s_i \oplus s_{i+1}$





Remark: No boundary conditions ⇒ The array "shrinks"

Linear CA

- Local rule: linear combination of the neighborhood cells

$$f(x_0, \cdots, x_{\delta-1}) = a_0 x_0 \oplus \cdots \oplus a_{\delta-1} x_{\delta-1} \quad , \; a_i \in \mathbb{F}_2$$

Linear CA

- Local rule: linear combination of the neighborhood cells

$$f(x_0, \cdots, x_{\delta-1}) = a_0 x_0 \oplus \cdots \oplus a_{\delta-1} x_{\delta-1} \quad , \; a_i \in \mathbb{F}_2$$

- Associated polynomial:

$$f \mapsto \varphi(X) = a_0 + a_1 X + \cdots + a_{\delta-1} X^{\delta-1}$$

Linear CA

- Local rule: linear combination of the neighborhood cells

$$f(x_0, \cdots, x_{\delta-1}) = a_0 x_0 \oplus \cdots \oplus a_{\delta-1} x_{\delta-1} \ , \ a_i \in \mathbb{F}_2$$

- Associated polynomial:

$$f \mapsto \varphi(X) = a_0 + a_1 X + \cdots + a_{\delta-1} X^{\delta-1}$$

- Global rule: $m \times (m + \delta - 1)$ δ-diagonal transition matrix

$$M_F = \begin{pmatrix} a_0 & \cdots & a_{\delta-1} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & a_0 & \cdots & a_{\delta-1} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & a_0 & \cdots & a_{\delta-1} \end{pmatrix}$$

$$x = (x_0, \cdots, x_{n-1}) \mapsto M_F x^\top$$

Linear CA are Cyclic Codes

$$M_F = \begin{pmatrix} a_0 & \cdots & a_{\delta-1} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & a_0 & \cdots & a_{\delta-1} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & a_0 & \cdots & a_{\delta-1} \end{pmatrix}$$

$$G = \begin{pmatrix} g_0 & \cdots & g_{n-m} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & g_0 & \cdots & g_{n-m} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & g_0 & \cdots & g_{n-m} \end{pmatrix}$$

Linear CA are Cyclic Codes

$$M_F = \begin{pmatrix} a_0 & \cdots & a_{\delta-1} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & a_0 & \cdots & a_{\delta-1} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & a_0 & \cdots & a_{\delta-1} \end{pmatrix}$$

$$G = \begin{pmatrix} g_0 & \cdots & g_{n-m} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & g_0 & \cdots & g_{n-m} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & g_0 & \cdots & g_{n-m} \end{pmatrix}$$

Linear CA $\Leftrightarrow$ Cyclic codes

Linear CA are Cyclic Codes

$$M_F = \begin{pmatrix} a_0 & \cdots & a_{\delta-1} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & a_0 & \cdots & a_{\delta-1} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & a_0 & \cdots & a_{\delta-1} \end{pmatrix}$$

$$G = \begin{pmatrix} g_0 & \cdots & g_{n-m} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & g_0 & \cdots & g_{n-m} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & g_0 & \cdots & g_{n-m} \end{pmatrix}$$

Linear CA $\Leftrightarrow$ Cyclic codes

Question: How is encoding/decoding performed?

Preimage Computation in Linear CA

Remark: if a_0, $a_{\delta-1} \neq 0$ then

$$y_i = a_0 x_0 \oplus \cdots \oplus a_{\delta-1} x_{\delta-1} \;\Rightarrow\; x_{\delta-1} = a_0 x_0 \oplus \cdots \oplus y_i$$

Preimage Computation in Linear CA

Remark: if a_0, $a_{\delta-1} \neq 0$ then

$$y_i = a_0 x_0 \oplus \cdots \oplus a_{\delta-1} x_{\delta-1} \;\Rightarrow\; x_{\delta-1} = a_0 x_0 \oplus \cdots \oplus y_i$$

1. Initialize the leftmost $\delta - 1$ cells $(x_0, \cdots, x_{\delta-1})$

$x =$	**0**	**1**	**?**	**?**	**?**	**?**	**?**	**?**
$y =$		**1**	**0**	**0**	**1**	**1**	**0**	

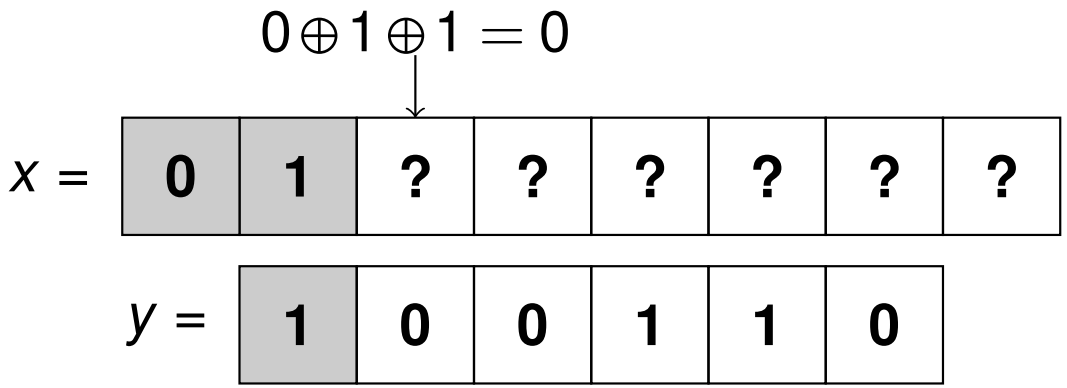
Example: rule 150, $f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$

Preimage Computation in Linear CA

Remark: if a_0, $a_{\delta-1} \neq 0$ then

$$y_i = a_0 x_0 \oplus \cdots \oplus a_{\delta-1} x_{\delta-1} \Rightarrow x_{\delta-1} = a_0 x_0 \oplus \cdots \oplus y_i$$

2. Compute $x_{\delta-1} = x_{\delta-1} = a_0 x_0 \oplus \cdots \oplus y_0$



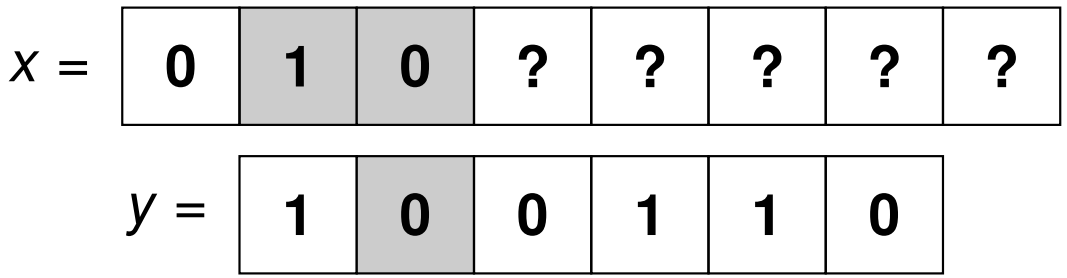
Example: rule 150, $f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$

Preimage Computation in Linear CA

Remark: if a_0, $a_{\delta-1} \neq 0$ then

$$y_i = a_0 x_0 \oplus \cdots \oplus a_{\delta-1} x_{\delta-1} \;\Rightarrow\; x_{\delta-1} = a_0 x_0 \oplus \cdots \oplus y_i$$

3. Shift the $(\delta - 1)$-cell window one place to the right



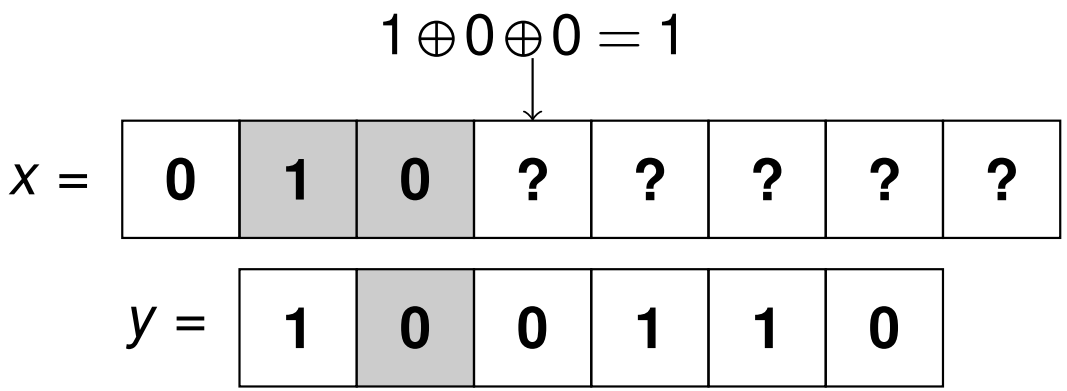
Example: rule 150, $f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$

Preimage Computation in Linear CA

Remark: if a_0, $a_{\delta-1} \neq 0$ then

$$y_i = a_0 x_0 \oplus \cdots \oplus a_{\delta-1} x_{\delta-1} \;\Rightarrow\; x_{\delta-1} = a_0 x_0 \oplus \cdots \oplus y_i$$

4. Compute $x_\delta = a_0 x_1 \oplus \cdots \oplus y_1$



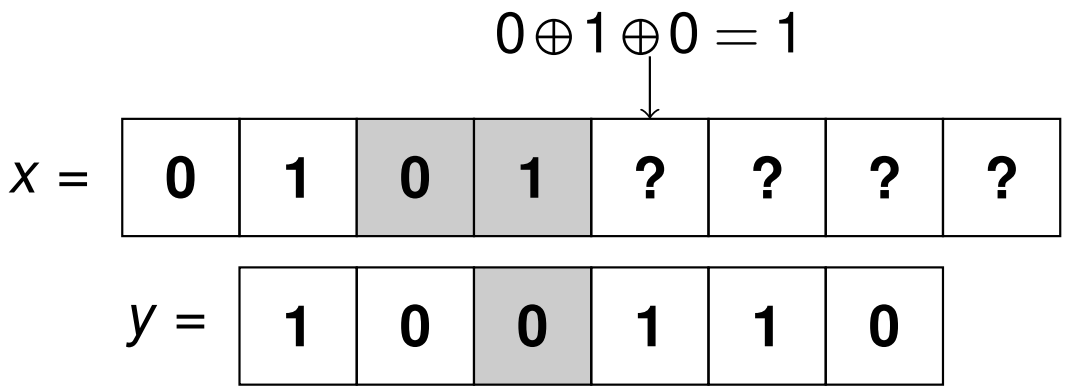
Example: rule 150, $f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$

Preimage Computation in Linear CA

Remark: if a_0, $a_{\delta-1} \neq 0$ then

$$y_i = a_0 x_0 \oplus \cdots \oplus a_{\delta-1} x_{\delta-1} \;\Rightarrow\; x_{\delta-1} = a_0 x_0 \oplus \cdots \oplus y_i$$

5. Repeat until preimage is complete



Example: rule 150, $f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$

Preimage Computation in Linear CA

Remark: if a_0, $a_{\delta-1} \neq 0$ then

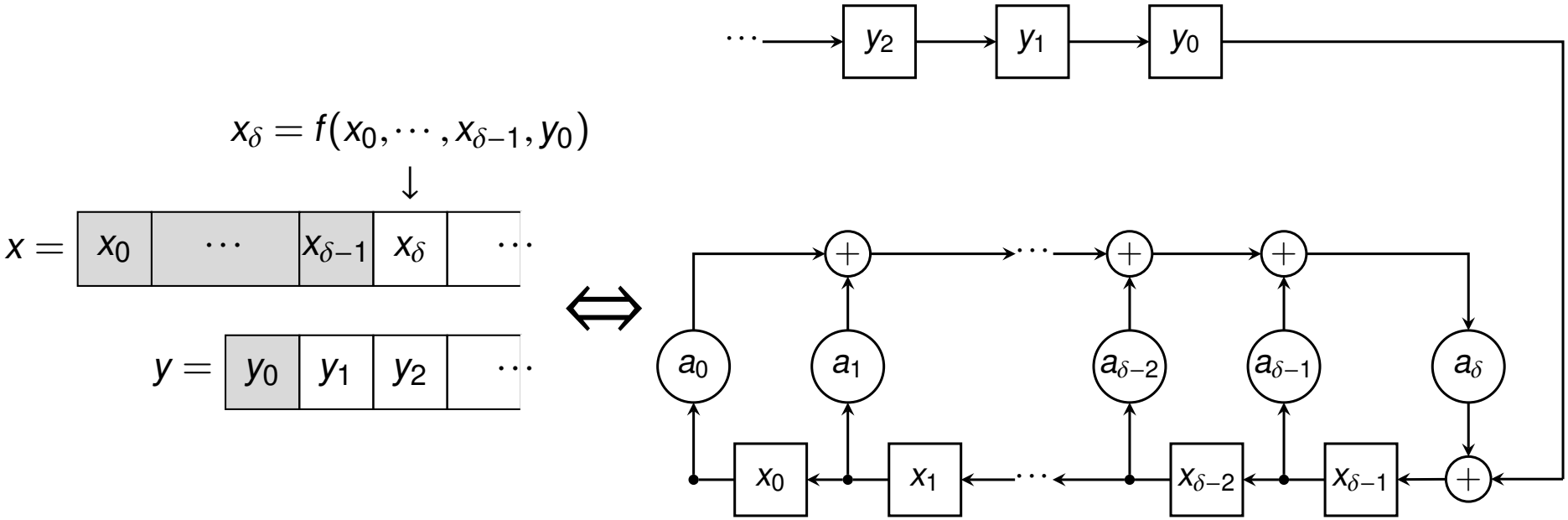
$$y_i = a_0 x_0 \oplus \cdots \oplus a_{\delta-1} x_{\delta-1} \;\Rightarrow\; x_{\delta-1} = a_0 x_0 \oplus \cdots \oplus y_i$$

5. Repeat until preimage is complete

$x =$	**0**	**1**	**0**	**1**	**1**	**1**	**1**	**0**
$y =$		**1**	**0**	**0**	**1**	**1**	**0**	

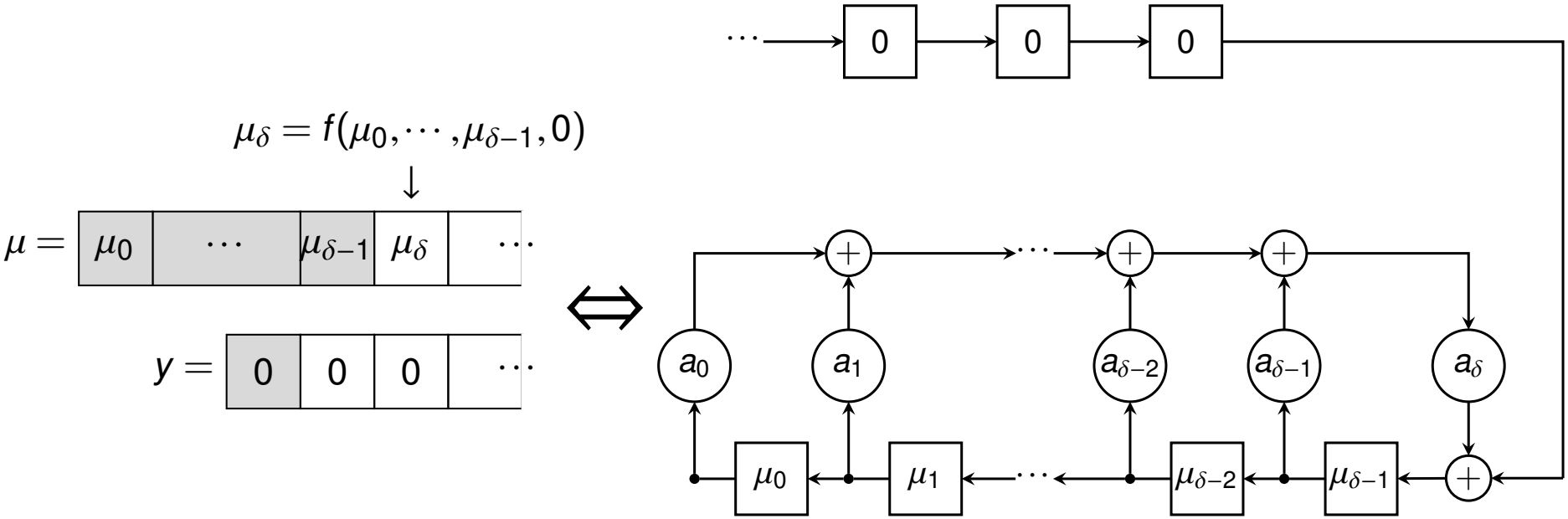
Example: rule 150, $f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$

Preimage Computation = Systematic Encoding



Preimage of linear CA $\Leftrightarrow$ LFSR "disturbed" by configuration y

Preimage Computation = Systematic Encoding



Systematic Encoding $\Leftrightarrow$ 0-preimage of CA initialized with message

Syndrome Computation = CA Iteration

- Polynomial of the CA rule $\Leftrightarrow$ Parity check polynomial

Syndrome Computation = CA Iteration

- Polynomial of the CA rule $\Leftrightarrow$ Parity check polynomial
- Syndrome computation is performed by CA global rule

Syndrome Computation = CA Iteration

- Polynomial of the CA rule $\Leftrightarrow$ Parity check polynomial
- Syndrome computation is performed by CA global rule

$z =$ | **1** | **0** | **1** | **1** | **0** | **1** | **1** | **0** |

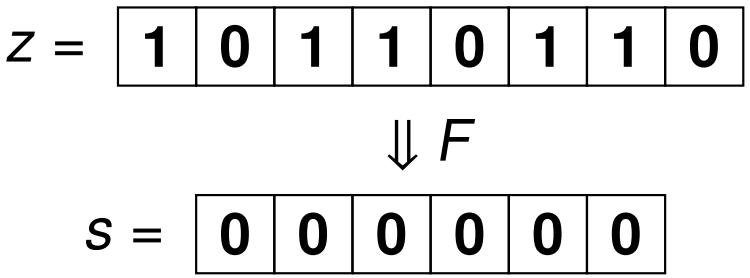
$\Downarrow F$

$s =$ | **0** | **0** | **0** | **0** | **0** | **0** |

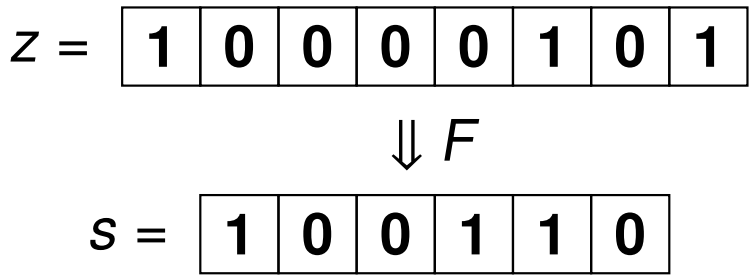
(a) $s = \underline{0} \Rightarrow$ No errors

Syndrome Computation = CA Iteration

- Polynomial of the CA rule $\Leftrightarrow$ Parity check polynomial
- Syndrome computation is performed by CA global rule



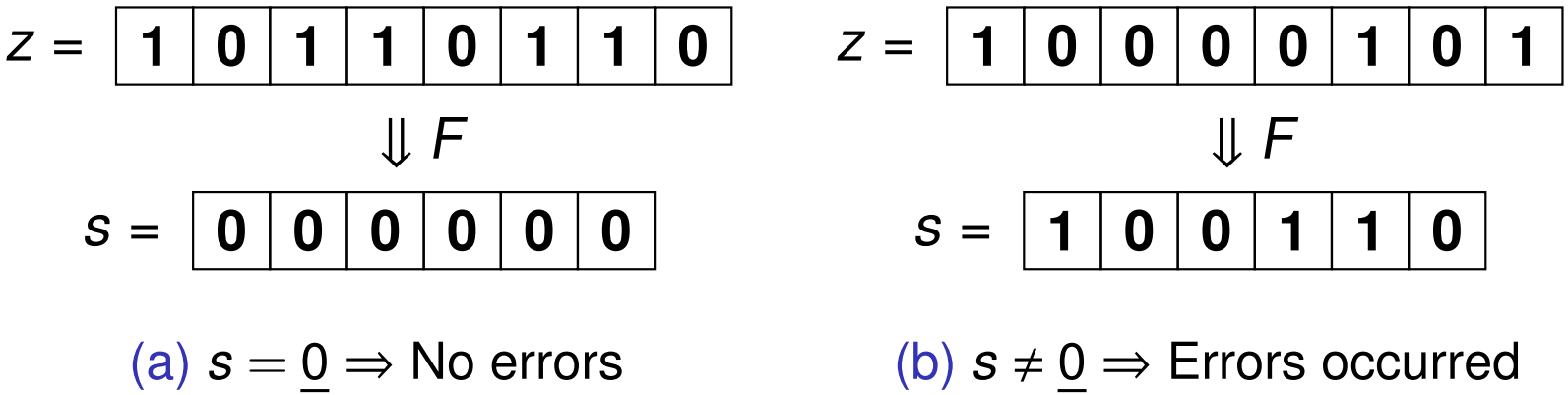
(a) $s = \underline{0} \Rightarrow$ No errors



(b) $s \neq \underline{0} \Rightarrow$ Errors occurred

Syndrome Computation = CA Iteration

- Polynomial of the CA rule $\Leftrightarrow$ Parity check polynomial
- Syndrome computation is performed by CA global rule



(a) $s = \underline{0} \Rightarrow$ No errors

(b) $s \neq \underline{0} \Rightarrow$ Errors occurred

Last Missing Piece: minimum distance d

Resilient Boolean Functions [Sieg84]

Definition

$F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is *t-resilient* if, by fixing any t input variables $x_{i_1}, \cdots, x_{i_t}$, the resulting restriction $\tilde{F} : \mathbb{F}_2^{n-t} \to \mathbb{F}_2^m$ is *balanced*.

Resilient Boolean Functions [Sieg84]

Definition

$F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is *t-resilient* if, by fixing any t input variables $x_{i_1}, \cdots, x_{i_t}$, the resulting restriction $\tilde{F} : \mathbb{F}_2^{n-t} \to \mathbb{F}_2^m$ is *balanced*.

Theorem ([Stin04])

A linear function $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is $(d-1)$–resilient iff M_F is the generator matrix of a (n, m, d) linear code.

Part 3: Cyclic Hamming Codes by CA

Hamming Codes

- Minimum distance: $d = 3$ ($\Rightarrow$ can correct up to 1 error)
- Syndrome value $\Leftrightarrow$ column of the parity check matrix H where the error occurred
- A $(n, m, 3)$ cyclic code is a Hamming code iff the generator polynomial is primitive

The $(7,4,3)$ Hamming Code through CA (1/4)

- Let $\langle 7,4,f \rangle$ be a CA induced by local rule $f(x) = x_1 \oplus x_2 \oplus x_4$.

The $(7,4,3)$ Hamming Code through CA (1/4)

- Let $\langle 7,4,f \rangle$ be a CA induced by local rule $f(x) = x_1 \oplus x_2 \oplus x_4$.
- Transition matrix of $F : \mathbb{F}_2^7 \to \mathbb{F}_2^4$:

$$M_F = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

The $(7,4,3)$ Hamming Code through CA (1/4)

- Let $\langle 7,4,f\rangle$ be a CA induced by local rule $f(x) = x_1 \oplus x_2 \oplus x_4$.
- Transition matrix of $F : \mathbb{F}_2^7 \to \mathbb{F}_2^4$:

$$M_F = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

- Associated polynomial: $f \mapsto g(X) = 1 + X + X^3$

The $(7,4,3)$ Hamming Code through CA (1/4)

- Let $\langle 7,4,f \rangle$ be a CA induced by local rule $f(x) = x_1 \oplus x_2 \oplus x_4$.
- Transition matrix of $F : \mathbb{F}_2^7 \to \mathbb{F}_2^4$:

$$M_F = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

- Associated polynomial: $f \mapsto g(X) = 1 + X + X^3$
- $g(X)$ is primitive and divides $X^7 - 1$, F is 2–resilient

The $(7,4,3)$ Hamming Code through CA (2/4)

Parity check polynomial:

- $h(X) = (X^7 - 1)/g(X) = 1 + X + X^2 + X^4$

The $(7,4,3)$ Hamming Code through CA (2/4)

Parity check polynomial:

- $h(X) = (X^7 - 1)/g(X) = 1 + X + X^2 + X^4$
- Local rule $\rightarrow f^*(x) = x_1 \oplus x_3 \oplus x_4 \oplus x_5$

The $(7,4,3)$ Hamming Code through CA (2/4)

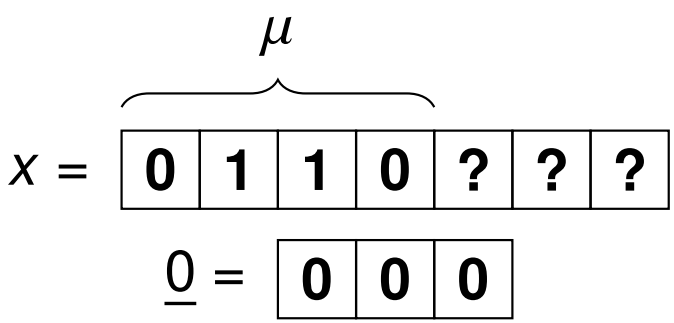
Parity check polynomial:

- $h(X) = (X^7 - 1)/g(X) = 1 + X + X^2 + X^4$
- Local rule $\rightarrow f^*(x) = x_1 \oplus x_3 \oplus x_4 \oplus x_5$
- Transition matrix of $F^* : \mathbb{F}_2^7 \rightarrow \mathbb{F}_2^3$:

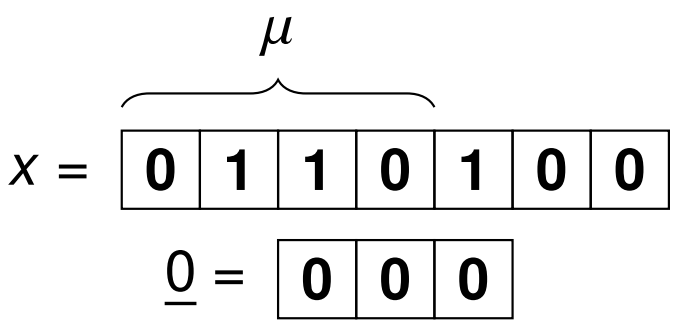
$$M_{F^*} = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

The (7,4,3) Hamming Code through CA (3/4)

Example of systematic encoding: Let $\mu = (0,1,1,0)$



(a) Initialization



(b) Complete codeword

The $(7,4,3)$ Hamming Code through CA (4/4)

Example of error correction: suppose the 4th bit has been flipped

$$x = \begin{array}{|c|c|c|c|c|c|c|} \hline \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \hline \end{array}$$

$$\begin{array}{c} \\ s = \end{array} \begin{array}{|c|c|c|} \hline \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \hline \end{array}$$

(a) Syndrome computation

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

(b) Error correction

Conclusions

- Linear CA are equivalent to linear cyclic codes
- Systematic encoding $\Leftrightarrow$ CA preimage computation
- Syndrome computation $\Leftrightarrow$ CA forward evolution
- Minimum distance $\Leftrightarrow$ Resiliency of CA global rule

Future directions

Cyclic codes form a broad category of linear codes:

- Reed-Solomon Codes
- BCH Codes
- Reed-Muller Codes
- ...

Applications to cryptography:

- MDS matrices for diffusion layer in block ciphers
- Secret sharing schemes

References

- [McEl85] McEliece, R.J.: The Theory of Information and Coding. Cambridge University Press, New York (1985)
- [Sieg84]Siegenthaler, T.: Decrypting a Class of Stream Ciphers Using Ciphertext Only. IEEE Trans. Comput. C-34(1), 81–85 (1985)
- [Stins04] Stinson, D.R.: Combinatorial Designs: Constructions and Analysis. Springer, Heidelberg (2004)