Exhaustive Generation of Linear Orthogonal CA

Enrico Formenti, Luca Mariot

l.mariot@utwente.nl

AUTOMATA 2023 – August 30, 2023

Coprime Polynomials

Object: pairs of binary polynomials of degree $n \in \mathbb{N}$:

$$f(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} + x^n \ ,$$
$$g(x) = b_0 + b_1 x + \cdots + b_{n-1} x^{n-1} + x^n \ ,$$

where $a_i, b_i \in GF(2) = \mathbb{F}_2 = \{0,1\}$

$f, g \in \mathbb{F}_2[x]$ are **coprime** $\Leftrightarrow \gcd(f,g) = 1$

Applications of **enumeration/counting** of coprime pairs:

- *Discrete logarithms* in finite fields [C84]
- Decoding *alternant codes* [F95]
- *Invertible Toeplitz matrices* [GR11]

Euclid's Algorithm

Check if $\gcd(f,g) = 1 \Rightarrow$ **Euclid's algorithm**

Example: $n = 4, \quad f(x) = x^4 + x^2, \quad g(x) = x^4 + x^3 + 1$

$$f(x) = q(x) \cdot g(x) + r(x)$$

$x^4 + x^2 = 1 \cdot (x^4 + x^3 + 1) + (x^3 + x^2 + 1)$
$x^4 + x^3 + 1 = x \cdot (x^3 + x^2 + 1) + (x + 1)$
$x^3 + x^2 + 1 = x^2 \cdot (x + 1) + 1$
$x + 1 = 1 \cdot (x + 1) + 0$

Compact notation:

$(x^4 + x^2, x^4 + x^3 + 1) \xrightarrow{1} (x^4 + x^3 + 1, x^3 + x^2 + 1) \xrightarrow{x}$
$(x^3 + x^2 + 1, x + 1) \xrightarrow{x^2} (x + 1, 1) \xrightarrow{x+1} (1, 0)$

Euclid's Algorithm

Check if $\gcd(f,g) = 1 \Rightarrow$ **Euclid's algorithm**

Example: $n = 4, \quad f(x) = x^4 + x^2, \quad g(x) = x^4 + x^3 + 1$

$$f(x) = q(x) \cdot g(x) + r(x)$$

$x^4 + x^2 = 1 \cdot (x^4 + x^3 + 1) + (x^3 + x^2 + 1)$
$x^4 + x^3 + 1 = x \cdot (x^3 + x^2 + 1) + (x + 1)$
$x^3 + x^2 + 1 = x^2 \cdot (x + 1) + 1$
$x + 1 = 1 \cdot (x + 1) + 0$

Compact notation:

$(x^4 + x^2, x^4 + x^3 + 1) \xrightarrow{1} (x^4 + x^3 + 1, x^3 + x^2 + 1) \xrightarrow{x}$
$(x^3 + x^2 + 1, x + 1) \xrightarrow{x^2} (x + 1, 1) \xrightarrow{x+1} (1, 0)$

Euclid's Algorithm

Check if $\gcd(f,g) = 1 \Rightarrow$ **Euclid's algorithm**

Example: $n = 4, \quad f(x) = x^4 + x^2, \quad g(x) = x^4 + x^3 + 1$

$$f(x) = q(x) \cdot g(x) + r(x)$$

$x^4 + x^2 = 1 \cdot (x^4 + x^3 + 1) + (x^3 + x^2 + 1)$
$x^4 + x^3 + 1 = x \cdot (x^3 + x^2 + 1) + (x + 1)$
$x^3 + x^2 + 1 = x^2 \cdot (x + 1) + 1$
$x + 1 = 1 \cdot (x + 1) + 0$

Compact notation:

$(x^4 + x^2, x^4 + x^3 + 1) \xrightarrow{1} (x^4 + x^3 + 1, x^3 + x^2 + 1) \xrightarrow{x}$
$(x^3 + x^2 + 1, x + 1) \xrightarrow{x^2} (x + 1, 1) \xrightarrow{x+1} (1, 0)$

Euclid's Algorithm

Check if $\gcd(f,g) = 1 \Rightarrow$ **Euclid's algorithm**

Example: $n = 4, \;\; f(x) = x^4 + x^2, \;\; g(x) = x^4 + x^3 + 1$

$$f(x) = q(x) \cdot g(x) + r(x)$$

$x^4 + x^2 = 1 \cdot (x^4 + x^3 + 1) + (x^3 + x^2 + 1)$
$x^4 + x^3 + 1 = x \cdot (x^3 + x^2 + 1) + (x + 1)$
$x^3 + x^2 + 1 = x^2 \cdot (x + 1) + 1$
$x + 1 = 1 \cdot (x + 1) + 0$

Compact notation:

$(x^4 + x^2, x^4 + x^3 + 1) \xrightarrow{1} (x^4 + x^3 + 1, x^3 + x^2 + 1) \xrightarrow{x}$
$(x^3 + x^2 + 1, x + 1) \xrightarrow{x^2} (x + 1, 1) \xrightarrow{x+1} (1, 0)$

Euclid's Algorithm

Check if $\gcd(f, g) = 1 \Rightarrow$ **Euclid's algorithm**

Example: $n = 4, \quad f(x) = x^4 + x^2, \quad g(x) = x^4 + x^3 + 1$

$$f(x) = q(x) \cdot g(x) + r(x)$$

$x^4 + x^2 = 1 \cdot (x^4 + x^3 + 1) + (x^3 + x^2 + 1)$
$x^4 + x^3 + 1 = x \cdot (x^3 + x^2 + 1) + (x + 1)$
$x^3 + x^2 + 1 = x^2 \cdot (x + 1) + 1$
$x + 1 = 1 \cdot (x + 1) + 0$

Compact notation:

$(x^4 + x^2, x^4 + x^3 + 1) \xrightarrow{1} (x^4 + x^3 + 1, x^3 + x^2 + 1) \xrightarrow{x}$
$(x^3 + x^2 + 1, x + 1) \xrightarrow{x^2} (x + 1, 1) \xrightarrow{x+1} (1, 0)$

Euclid's Algorithm

Check if $\gcd(f,g) = 1 \Rightarrow$ **Euclid's algorithm**

Example: $n = 4, \quad f(x) = x^4 + x^2, \quad g(x) = x^4 + x^3 + 1$

$$f(x) = q(x) \cdot g(x) + r(x)$$

$x^4 + x^2 = 1 \cdot (x^4 + x^3 + 1) + (x^3 + x^2 + 1)$
$x^4 + x^3 + 1 = x \cdot (x^3 + x^2 + 1) + (x + 1)$
$x^3 + x^2 + 1 = x^2 \cdot (x + 1) + 1$
$x + 1 = 1 \cdot (x + 1) + 0$

Compact notation:

$(x^4 + x^2, x^4 + x^3 + 1) \xrightarrow{1} (x^4 + x^3 + 1, x^3 + x^2 + 1) \xrightarrow{x}$
$(x^3 + x^2 + 1, x + 1) \xrightarrow{x^2} (x + 1, 1) \xrightarrow{x+1} (1, 0)$

Euclid's Algorithm

Check if $\gcd(f,g) = 1 \Rightarrow$ **Euclid's algorithm**

Example: $n = 4, \quad f(x) = x^4 + x^2, \quad g(x) = x^4 + x^3 + 1$

$$f(x) = q(x) \cdot g(x) + r(x)$$

$$x^4 + x^2 = 1 \cdot (x^4 + x^3 + 1) + (x^3 + x^2 + 1)$$
$$x^4 + x^3 + 1 = x \cdot (x^3 + x^2 + 1) + (x + 1)$$
$$x^3 + x^2 + 1 = x^2 \cdot (x + 1) + 1$$
$$x + 1 = 1 \cdot (x + 1) + 0$$

Compact notation:

$$(x^4 + x^2, x^4 + x^3 + 1) \xrightarrow{1} (x^4 + x^3 + 1, x^3 + x^2 + 1) \xrightarrow{x}$$
$$(x^3 + x^2 + 1, x + 1) \xrightarrow{x^2} (x + 1, 1) \xrightarrow{x+1} (1, 0)$$

DilcuE's Algorithm

- **Remark:** (f, g) can be recovered from $(1, 0)$ by applying the same sequence of quotients $(1, x, x^2, x+1)$ *backward*
- This is called **DilcuE's algorithm** in [BB07]

$$(0,1) \xrightarrow{x+1} (1, x+1) \xrightarrow{x^2} (x+1, x^3+x^2+1) \xrightarrow{x} (x^3+x^2+1, x^4+x^3+1) \xrightarrow{1} (x^4+x^3+1, x^4+x^2) = (f,g)$$

- Suppose we change the last remainder to 0:

$$(0, x+1) \xrightarrow{x^2} (x+1, x^3+x^2) \xrightarrow{x} (x^3+x^2, x^4+x^3+x+1) \xrightarrow{1} (x^4+x^3+x+1, x^4+x^2+x+1) = (f', g')$$

- By construction, (f', g') are *non-coprime* with $\gcd(f', g') = x+1$

DilcuE's Algorithm

- **Remark:** (f, g) can be recovered from $(1, 0)$ by applying the same sequence of quotients $(1, x, x^2, x+1)$ *backward*
- This is called **DilcuE's algorithm** in [BB07]

$$(0,1) \xrightarrow{x+1} (1, x+1) \xrightarrow{x^2} (x+1, x^3+x^2+1) \xrightarrow{x} (x^3+x^2+1, x^4+x^3+1) \xrightarrow{1} (x^4+x^3+1, x^4+x^2) = (f, g)$$

- Suppose we change the last remainder to 0:

$$(0, x+1) \xrightarrow{x^2} (x+1, x^3+x^2) \xrightarrow{x} (x^3+x^2, x^4+x^3+x+1) \xrightarrow{1} (x^4+x^3+x+1, x^4+x^2+x+1) = (f', g')$$

- By construction, (f', g') are *non-coprime* with $\gcd(f', g') = x+1$

DilcuE's Algorithm

- **Remark:** (f, g) can be recovered from $(1, 0)$ by applying the same sequence of quotients $(1, x, x^2, x+1)$ *backward*
- This is called **DilcuE's algorithm** in [BB07]

$$(0,1) \xrightarrow{x+1} (1, x+1) \xrightarrow{x^2} (x+1, x^3+x^2+1) \xrightarrow{x} (x^3+x^2+1, x^4+x^3+1) \xrightarrow{1} (x^4+x^3+1, x^4+x^2) = (f, g)$$

- Suppose we change the last remainder to 0:

$$(0, x+1) \xrightarrow{x^2} (x+1, x^3+x^2) \xrightarrow{x} (x^3+x^2, x^4+x^3+x+1) \xrightarrow{1} (x^4+x^3+x+1, x^4+x^2+x+1) = (f', g')$$

- By construction, (f', g') are *non-coprime* with $\gcd(f', g') = x+1$

DilcuE's Algorithm

- **Remark:** (f, g) can be recovered from $(1, 0)$ by applying the same sequence of quotients $(1, x, x^2, x+1)$ *backward*
- This is called **DilcuE's algorithm** in [BB07]

$$(0,1) \xrightarrow{x+1} (1, x+1) \xrightarrow{x^2} (x+1, x^3+x^2+1) \xrightarrow{x} (x^3+x^2+1, x^4+x^3+1) \xrightarrow{1} (x^4+x^3+1, x^4+x^2) = (f,g)$$

- Suppose we change the last remainder to 0:

$$(0, x+1) \xrightarrow{x^2} (x+1, x^3+x^2) \xrightarrow{x} (x^3+x^2, x^4+x^3+x+1) \xrightarrow{1} (x^4+x^3+x+1, x^4+x^2+x+1) = (f', g')$$

- By construction, (f', g') are *non-coprime* with $\gcd(f', g') = x+1$

DilcuE's Algorithm

- **Remark:** (f, g) can be recovered from $(1, 0)$ by applying the same sequence of quotients $(1, x, x^2, x+1)$ *backward*
- This is called **DilcuE's algorithm** in [BB07]

$$(0,1) \xrightarrow{x+1} (1, x+1) \xrightarrow{x^2} (x+1, x^3+x^2+1) \xrightarrow{x} (x^3+x^2+1, x^4+x^3+1) \xrightarrow{1} (x^4+x^3+1, x^4+x^2) = (f, g)$$

- Suppose we change the last remainder to 0:

$$(0, x+1) \xrightarrow{x^2} (x+1, x^3+x^2) \xrightarrow{x} (x^3+x^2, x^4+x^3+x+1) \xrightarrow{1} (x^4+x^3+x+1, x^4+x^2+x+1) = (f', g')$$

- By construction, (f', g') are *non-coprime* with $\gcd(f', g') = x+1$

DilcuE's Algorithm

- **Remark:** (f, g) can be recovered from $(1, 0)$ by applying the same sequence of quotients $(1, x, x^2, x+1)$ *backward*
- This is called **DilcuE's algorithm** in [BB07]

$$(0,1) \xrightarrow{x+1} (1, x+1) \xrightarrow{x^2} (x+1, x^3+x^2+1) \xrightarrow{x} (x^3+x^2+1, x^4+x^3+1) \xrightarrow{1} (x^4+x^3+1, x^4+x^2) = (f, g)$$

- Suppose we change the last remainder to 0:

$$(0, x+1) \xrightarrow{x^2} (x+1, x^3+x^2) \xrightarrow{x} (x^3+x^2, x^4+x^3+x+1) \xrightarrow{1} (x^4+x^3+x+1, x^4+x^2+x+1) = (f', g')$$

- By construction, (f', g') are *non-coprime* with $\gcd(f', g') = x+1$

DilcuE's Algorithm

- **Remark:** (f, g) can be recovered from $(1, 0)$ by applying the same sequence of quotients $(1, x, x^2, x+1)$ *backward*
- This is called **DilcuE's algorithm** in [BB07]

$$(0,1) \xrightarrow{x+1} (1, x+1) \xrightarrow{x^2} (x+1, x^3+x^2+1) \xrightarrow{x} (x^3+x^2+1, x^4+x^3+1) \xrightarrow{1} (x^4+x^3+1, x^4+x^2) = (f, g)$$

- Suppose we change the last remainder to 0:

$$(0, x+1) \xrightarrow{x^2} (x+1, x^3+x^2) \xrightarrow{x} (x^3+x^2, x^4+x^3+x+1) \xrightarrow{1} (x^4+x^3+x+1, x^4+x^2+x+1) = (f', g')$$

- By construction, (f', g') are *non-coprime* with $\gcd(f', g') = x+1$

DilcuE's Algorithm

- **Remark:** (f, g) can be recovered from $(1, 0)$ by applying the same sequence of quotients $(1, x, x^2, x+1)$ *backward*
- This is called **DilcuE's algorithm** in [BB07]

$$(0,1) \xrightarrow{x+1} (1, x+1) \xrightarrow{x^2} (x+1, x^3+x^2+1) \xrightarrow{x} (x^3+x^2+1, x^4+x^3+1) \xrightarrow{1} (x^4+x^3+1, x^4+x^2) = (f, g)$$

- Suppose we change the last remainder to 0:

$$(0, x+1) \xrightarrow{x^2} (x+1, x^3+x^2) \xrightarrow{x} (x^3+x^2, x^4+x^3+x+1) \xrightarrow{1} (x^4+x^3+x+1, x^4+x^2+x+1) = (f', g')$$

- By construction, (f', g') are *non-coprime* with $\gcd(f', g') = x+1$

DilcuE's Algorithm

- **Remark:** (f, g) can be recovered from $(1, 0)$ by applying the same sequence of quotients $(1, x, x^2, x+1)$ *backward*
- This is called **DilcuE's algorithm** in [BB07]

$$(0,1) \xrightarrow{x+1} (1, x+1) \xrightarrow{x^2} (x+1, x^3+x^2+1) \xrightarrow{x} (x^3+x^2+1, x^4+x^3+1) \xrightarrow{1} (x^4+x^3+1, x^4+x^2) = (f, g)$$

- Suppose we change the last remainder to 0:

$$(0, x+1) \xrightarrow{x^2} (x+1, x^3+x^2) \xrightarrow{x} (x^3+x^2, x^4+x^3+x+1) \xrightarrow{1} (x^4+x^3+x+1, x^4+x^2+x+1) = (f', g')$$

- By construction, (f', g') are *non-coprime* with $\gcd(f', g') = x+1$

Counting by Bijection

In essence: we can construct a bijection between coprime and non-coprime pairs over $\mathbb{F}_2$ as follows

1. Apply Euclid to (f, g)
2. If the last remainder is 0, change it to 1. Otherwise, set it to the second-last remainder
3. Apply DilcuE's algorithm to the reversed quotients

Theorem ([BB07, CSWZ98, R00])

Let $f, g \in \mathbb{F}_2[x]$ of degree n be randomly chosen. Then, the probability that $\gcd(f, g) = 1$ is $\frac{1}{2}$. Equivalently, the number of coprime pairs is 2^{2n-1}.

This result is generalized to $\mathbb{F}_q$ by a 1-to-q correspondence

Counting/enumeration with nonzero constant terms

We require now that both f and g have a *nonzero* constant term:

$$f(x) = 1 + a_1 x + \cdots + a_{n-1} x^{n-1} + x^n \ ,$$
$$g(x) = 1 + b_1 x + \cdots + b_{n-1} x^{n-1} + x^n \ .$$

Problems:

1. *Count* all such pairs
2. *Enumeration algorithm*

Remark: the trick above does not work! Changing the last remainder gives no control over the final constant terms

... Why do we want to do that?

Orthogonal Latin Squares by Linear Cellular Automata

- Bipermutive Linear rule: $f(x) = x_1 \oplus a_2 x_2 \oplus \cdots \oplus a_{d-1} x_{d-1} \oplus x_d$
- Polynomial rule: $P_f(X) = 1 + a_2 X + \cdots + a_{d-1} X^{d-2} + X^{d-1}$

Theorem ([MFL16, MGFL20])

Two bipermutive linear CA generates orthogonal Latin squares if and only if their associated polynomials are coprime

1	4	3	2
2	3	4	1
4	1	2	3
3	2	1	4

(a) Rule 150

1	2	3	4
2	1	4	3
3	4	1	2
4	3	2	1

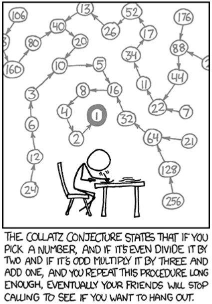
(b) Rule 90

1, 1	4, 2	3, 3	2, 4
2, 2	3, 1	4, 4	1, 3
4, 3	1, 4	2, 1	3, 2
3, 4	2, 3	1, 2	4, 1

(c) Superposition

Figure: $P_{150}(X) = 1 + X + X^2$, $P_{90}(X) = 1 + X^2$ (coprime)

Counting by Recurrence



S `https://xkcd.com/710/`

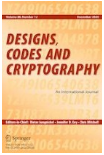
- Number of coprime polynomial pairs of degree n and nonzero constant term:

$$a(n) = 4^{n-1} + a(n-1) = \frac{4^{n-1} - 1}{3}$$
$$= 0, 1, 5, 21, 85, \ldots$$

- Corresponds to OEIS A002450

- Generalized for any finite field $\mathbb{F}_q$ in [MGFL20] (but enumeration not addressed)

L. Mariot, M. Gadouleau, E. Formenti, and A. Leporati. Mutually orthogonal latin squares based on cellular automata. Des. Codes Cryptogr. 88(2):391–411 (2020)



Problem Structure

Strategy: characterize the *sequences* of quotients that gives only $(1,1)$ coprime pairs when starting from the remainders $(1,0)$

Three parts of the problem:

$$\begin{array}{llllll} & \overbrace{\phantom{x^{d_1}}}^{\text{degrees}} & & \overbrace{\phantom{q_{1,d_1-1}x^{d_1-1}+\cdots+q_{1,1}x}}^{\text{middle terms}} & & \overbrace{}^{\text{constant terms}} \\ q_1 \rightarrow & x^{d_1} & + & q_{1,d_1-1}x^{d_1-1}+\cdots+q_{1,1}x & + & s_1 \\ q_2 \rightarrow & x^{d_2} & + & q_{2,d_2-1}x^{d_2-1}+\cdots+q_{2,1}x & + & s_2 \\ \vdots \rightarrow & \vdots & + & \vdots \quad +\cdots+ \quad \vdots & + & \vdots \\ q_k \rightarrow & x^{d_k} & + & q_{k,d_k-1}x^{d_k-1}+\cdots+q_{k,1}x & + & s_k \end{array}$$

Notation: $r_i, r_{i+1} \rightarrow$ consecutive remainders produced by Euclid's algorithm at step i. Step $i+1$:

$$r_i(x) = q_{i+1}(x)r_{i+1}(x) + r_{i+2}(x)$$

Problem Structure

Strategy: characterize the *sequences* of quotients that gives only $(1,1)$ coprime pairs when starting from the remainders $(1,0)$

Three parts of the problem:

$$
\begin{array}{lllll}
q_1 \rightarrow & \overbrace{x^{d_1}}^{\textit{degrees}} & + \overbrace{q_{1,d_1-1}x^{d_1-1} + \cdots + q_{1,1}x}^{\text{middle terms}} & + & \overbrace{s_1}^{\text{constant terms}} \\
q_2 \rightarrow & x^{d_2} & + q_{2,d_2-1}x^{d_2-1} + \cdots + q_{2,1}x & + & s_2 \\
\vdots \rightarrow & \vdots & + \quad \vdots \quad + \cdots + \quad \vdots & + & \vdots \\
q_k \rightarrow & x^{d_k} & + q_{k,d_k-1}x^{d_k-1} + \cdots + q_{k,1}x & + & s_k
\end{array}
$$

Notation: $r_i, r_{i+1} \rightarrow$ consecutive remainders produced by Euclid's algorithm at step i. Step $i+1$:

$$r_i(x) = q_{i+1}(x)r_{i+1}(x) + r_{i+2}(x)$$

Problem Structure

Strategy: characterize the *sequences* of quotients that gives only $(1,1)$ coprime pairs when starting from the remainders $(1,0)$

Three parts of the problem:

$$\begin{array}{rcl} q_1 \rightarrow & \overbrace{x^{d_1}}^{degrees} & + \overbrace{q_{1,d_1-1}x^{d_1-1} + \cdots + q_{1,1}x}^{\text{middle terms}} + \overbrace{s_1}^{\text{constant terms}} \\ q_2 \rightarrow & x^{d_2} & + q_{2,d_2-1}x^{d_2-1} + \cdots + q_{2,1}x + s_2 \\ \vdots \rightarrow & \vdots & + \vdots + \cdots + \vdots + \vdots \\ q_k \rightarrow & x^{d_k} & + q_{k,d_k-1}x^{d_k-1} + \cdots + q_{k,1}x + s_k \end{array}$$

Notation: $r_i, r_{i+1} \rightarrow$ consecutive remainders produced by Euclid's algorithm at step i. Step $i+1$:

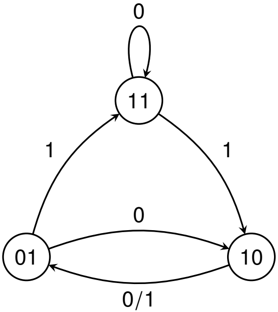
$$r_i(x) = q_{i+1}(x)r_{i+1}(x) + r_{i+2}(x)$$

Finite State Automaton of Remainders

- $(c_i, c_{i+1}) \rightarrow$ constant terms of r_i and r_{i+1}
- $X_{i+1} \rightarrow$ constant term of q_{i+1}
- $\delta((c_i, c_{i+1}), X_{i+1}) \rightarrow$ *next* pair (c_{i+1}, c_{i+2})

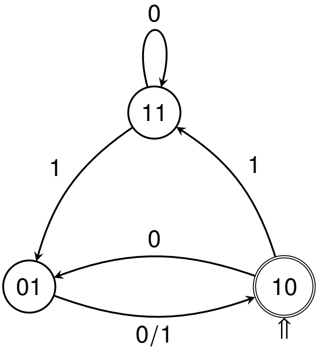
(c_i, c_{i+1})	X_{i+1}	$\delta((c_i, c_{i+1}), X_{i+1})$
$(1,1)$	0	$(1,1)$
$(1,1)$	1	$(1,0)$
$(1,0)$	0	$(0,1)$
$(1,0)$	1	$(0,1)$
$(0,1)$	0	$(1,0)$
$(0,1)$	1	$(1,1)$

Remark: the pair $(0,0)$ *never* occurs



$\Rightarrow$ the sequences of constant terms form a **regular language**

The Language of Constant Terms Sequences



Inverse FSA

- The FSA is the *de Bruijn* graph over the set $\{11, 10, 01\}$
- The FSA is *permutative*: for DilcuE's, simply reverse the arrows
- **Initial state**: 10
- **Final state**: 11 (but we can use 10)

Regular Expression of the Language:

$$L = (0(0+1) + (10^*1(0+1)))^*$$

Enumeration/counting of Constant Terms Sequences

- **Enumeration**: visit the FSA graph with DFS up to depth n
- **Counting**: exploit *algebraic language theory*

Transform $L = (0(0+1) + (10^*1(0+1)))^*$ as follows:

- $0, 1 \Rightarrow X$
- $+, \cdot \Rightarrow +, \cdot$
- $^* \Rightarrow \frac{1}{1-X}$

Generating Function:

$$\sum_{n=0}^{\infty} a_n \cdot X^n = \frac{1-X}{1-X-2X^2} \ ,$$

Closed Form:

$$a_n = \frac{2^n + 2\cdot(-1)^n}{3}$$

Sequences of quotients' degrees

Second part: Characterize the *degrees* of the quotients

Example: $n = 4$, $\{1, x, x^2, x, 1\}$

$$(0,1) \xrightarrow{1} (1,1) \xrightarrow{x} (1,x+1) \xrightarrow{x^2} (x+1, x^3+x^2+1) \xrightarrow{x} (x^3+x^2+1, x^4+x^3+1) \xrightarrow{1} (x^4+x^3+1, x^4+x^2+1)$$

Sum of degrees: $1 + 2 + 1 = 4$

Question: what are the combinations of *ordered sums* of n?

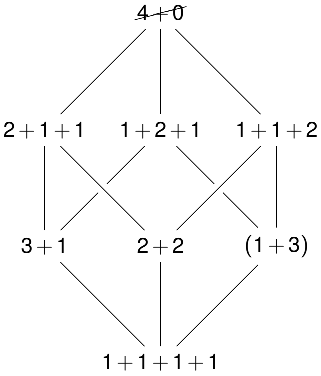
$\Rightarrow$ **compositions** of $n \in \mathbb{N}$

Quotients' degrees as compositions of n

- **Representation:** $n-1$ *boxes* that can be either "+" or ","

$$1 \overbrace{\square 1 \square \ldots \square 1 \square}^{n-1} 1$$

- **Example:** $1,1+1,1 \rightarrow 1+2+1 \quad (n=4)$



- We remove the top of the poset
- **Enumeration**: generate all binary strings of length $1 < k < n$
- **Counting**: $\binom{n-1}{k-1}$
- Remaining coefficients of the quotients are **free**

Enumeration Algorithm

Remark: once we fix the length of the sequence, the three elements (constant terms, degrees, middle terms) are *independent*

So for **enumeration**, given $n \in \mathbb{N}$:

For each composition *comp* of n (except $n+0$) do:

- Generate all quotients' sequences of *comp* (2^{n-k})
- For each quotients' sequence *seq* do:
  - For each constant term sequence of length $|seq|$ do:
    - Add the constant terms to the quotients
    - Apply DilcuE's from $(1,0)$ by applying *seq*

And for **counting**, we reobtain the formula $\frac{4^{n-1}-1}{3}$ from:

$$\sum_{k=2}^{n} 2^{n-k} \cdot \binom{n-1}{k-1} \cdot \frac{2^k + 2 \cdot (-1)^k}{3}$$

Conclusions and Future Work

Summing up:

- Enumeration of binary coprime polynomials is more complicated when both constant terms are nonzero
- We divided the problem in two enumeration tasks:
  - sequences of constant terms ($\Rightarrow$ regular language)
  - sequences of degrees ($\Rightarrow$ compositions)

Future directions:

- Generalize to polynomials over any finite field $\mathbb{F}_q$
- Generalize to *m*-tuples of pairwise coprime polynomials
- Applications to cryptography and coding theory [GMP20, GM20, M21]

Thank you!

Appendix: Orthogonal Latin Squares (OLS)

Definition

A *Latin square* is a $n \times n$ matrix where all rows and columns are permutations of $[n] = \{1, \cdots, n\}$. Two Latin squares are *orthogonal* if their superposition yields all the pairs $(x, y) \in [n] \times [n]$.

1	3	4	2
4	2	1	3
2	4	3	1
3	1	2	4

1	4	2	3
3	2	4	1
4	1	3	2
2	3	1	4

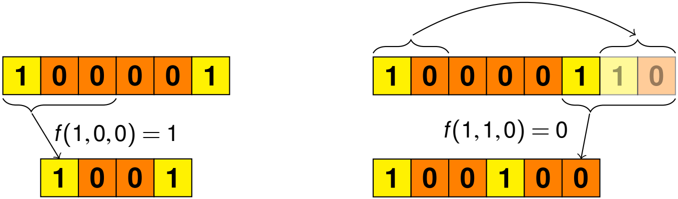
1, 1	3, 4	4, 2	2, 3
4, 3	2, 2	1, 4	3, 1
2, 4	4, 1	3, 3	1, 2
3, 2	1, 3	2, 1	4, 4

- k pairwise OLS are denoted as k-MOLS (**Mutually Orthogonal Latin Squares**)
- k-MOLS are **equivalent** $OA(n^2, k, n, 2)$

Appendix: Cellular Automata

- One-dimensional Cellular Automaton (CA): a discrete parallel computation model composed of a finite array of n cells

Example: $n = 6$, $d = 3$, $\omega = 0$, $f(s_i, s_{i+1}, s_{i+2}) = s_i \oplus s_{i+1} \oplus s_{i+2}$ (rule 150)



No Boundary CA – NBCA

Periodic Boundary CA – PBCA

- Each cell updates its state $s \in \{0,1\}$ by applying a local rule $f : \{0,1\}^d \rightarrow \{0,1\}$ to itself, the ω cells on its left and the $d - 1 - \omega$ cells on its right

Latin Squares through Bipermutive CA (1/2)

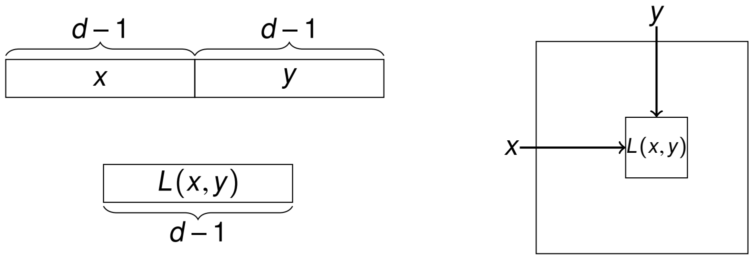
- Bipermutive CA: denoting $\mathbb{F}_2 = \{0,1\}$, local rule f is defined as

$$f(x_1,\cdots,x_d) = x_1 \oplus \varphi(x_2,\cdots,x_{d-1}) \oplus x_d$$

- $\varphi : \mathbb{F}_2^{d-2} \to \mathbb{F}_2$: generating function of f

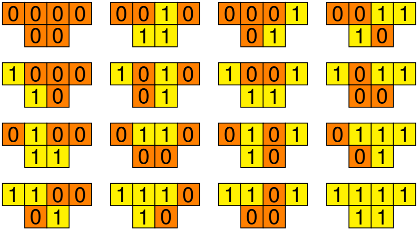
Lemma ([MGFL20])

A CA $F : \mathbb{F}_2^{2(d-1)} \to \mathbb{F}_2^d$ with bipermutive rule $f : \mathbb{F}_2^d \to \mathbb{F}_2$ generates a Latin square of order $N = 2^{d-1}$



Latin Squares through Bipermutive CA (2/2)

- Example: CA $F : \mathbb{F}_2^4 \rightarrow \mathbb{F}_2^2$, $f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$ (Rule 150)
- Encoding: $00 \mapsto 1, 10 \mapsto 2, 01 \mapsto 3, 11 \mapsto 4$



(a) Rule 150 on 4 bits

1	4	3	2
2	3	4	1
4	1	2	3
3	2	1	4

(b) Latin square L_{150}

Mutually Orthogonal Cellular Automata (MOCA): set of k bipermutive CA generating k-MOLS

References

[BB07] Benjamin, A.T., Bennett, C.D.: The probability of relatively prime polynomials. Mathematics Magazine 80(3): 196-202 (2007).

[C84] Coppersmith, D.: Fast evaluation of logarithms in fields of characteristic two. IEEE Trans. Inf. Theory 30(4): 587-593 (1984)

[CSWZ98] Corteel, S. Savage, C.D., Wilf, H.S., Zeilberger,D.: A pentagonal number sieve. Journal of Combinatorial Theory, Series A 82: 186-192 (1998)

[F95] Fitzpatrick, P.: On the key equation. IEEE Trans. Inf. Theory 41(5): 1290-1302 (1995)

[GM20] Gadouleau, M., Mariot, L.: Latin Hypercubes and Cellular Automata. Proceedings of Automata 2020, pp. 139-151 (2020)

[GMP20] Gadouleau, M., Mariot, L., Picek, S.: Bent Functions from Cellular Automata. IACR Cryptol. ePrint Arch. 2020: 1272 (2020)9)

[GR11] Ghorpade, S. R., Ram, S.: Block companion Singer cycles, primitive recursive vector sequences, and coprime polynomial pairs over finite fields. Finite Fields Their Appl. 17(5): 461-472 (2011)

[M21] Mariot, L.: Hip to Be (Latin) Square: Maximal Period Sequences from Orthogonal Cellular Automata. In: Proceedings of CANDAR 2021, pp. 29-37 (2021)

[MFL16] Mariot, L., Formenti, E., Leporati, A.: Constructing Orthogonal Latin Squares from Linear Cellular Automata. In: Exploratory papers of AUTOMATA 2016. CoRR abs/1610.00139 (2016)

[MGFL20] Mariot, L., Gadouleau, M. Formenti, E., Leporati A.: Mutually orthogonal latin squares based on cellular automata. Des. Codes Cryptogr. 88(2):391–411 (2020)

[R00] Reifegerste, A.: On an involution concerning pairs of polynomials in $\mathbb{F}_2$. J. Combin. Theory Ser. A 90, 216-220 (2000)