Inversion of Mutually Orthogonal Cellular Automata

Luca Mariot, Alberto Leporati

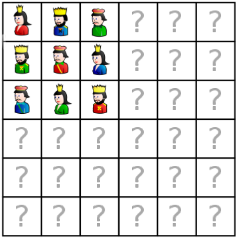
Bicocca Security Lab (BiSLab)
Dipartimento di Informatica, Sistemistica e Comunicazione (DISCo)
Università degli Studi Milano - Bicocca

ACRI 2018 – Como, September 17-21, 2018

Euler's 36 Officers Problem

« *A very curious question [...] revolves around arranging 36 officers to be drawn from 6 different ranks and also from 6 different regiments so that they are ranged in a square so that in each line (both horizontal and vertical) there are 6 officers of different ranks and different regiments.* »

L. Euler, *Sur une nouvelle espèce de quarrés magiques*, 1782



Latin Squares

Definition

A *Latin square* of order N is a $N \times N$ matrix L such that every row and every column are permutations of $[N] = \{1, \cdots, N\}$

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

Orthogonal Latin Squares (OLS)

Definition

Two Latin squares L_1 and L_2 of order N 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

(a) L_1

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

(b) L_2

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	2,4	4,1

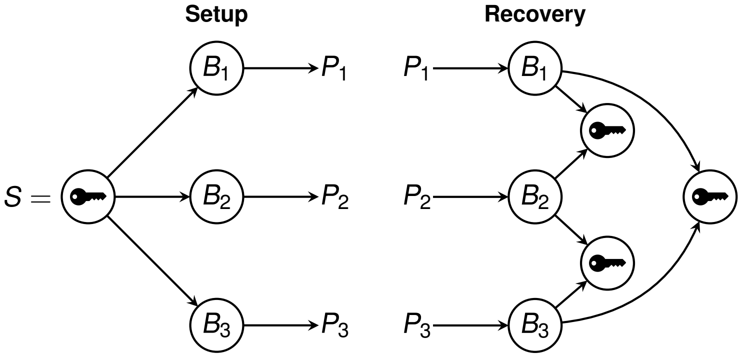
(c) (L_1, L_2)

A set of n pairwise orthogonal Latin squares is denoted as n-MOLS

Secret Sharing Schemes (SSS)

(k,n) Threshold Secret Sharing Scheme: a procedure enabling a dealer to share a secret S among n players so that at least k players out of n can recover S [Shamir79].

Example: $(2,3)$–scheme



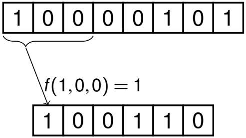
Remark: $(2,n)$–scheme $\Leftrightarrow$ set of n-MOLS

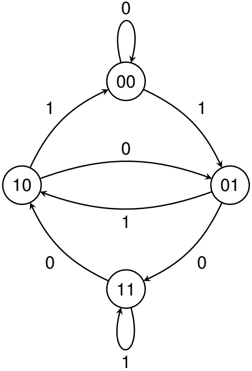
One-Dimensional Cellular Automata (CA)

Definition

One-dimensional CA: triple $\langle m, n, f \rangle$ where $n \in \mathbb{N}$ is the number of cells on a one-dimensional array, $n \in \mathbb{N}$ is the neighborhood and $f : \{0,1\}^n \to \{0,1\}$ is the local rule.

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





Latin Squares through Bipermutive CA (1/2)

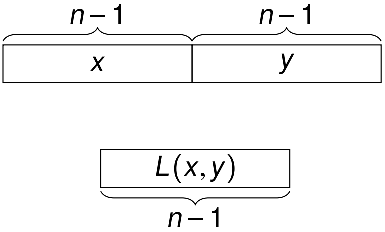
- Idea: determine which CA induce orthogonal Latin squares
- Bipermutive CA: local rule f is defined as

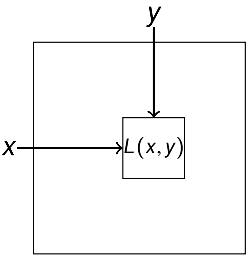
$$f(x_1, \cdots, x_n) = x_1 \oplus \varphi(x_2, \cdots, x_{n-1}) \oplus x_n$$

- $\varphi : \{0,1\}^{n-2} \rightarrow \{0,1\}$: generating function of f

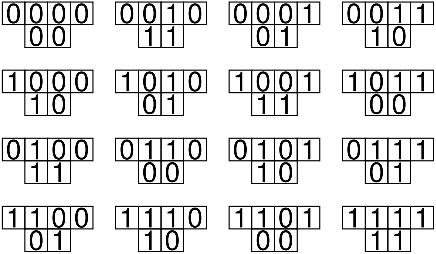
Lemma ([Eloranta93, Mariot16])

Let $\langle 2(n-1), n, f \rangle$ be a CA with bipermutive rule. Then, the global rule F generates a Latin square of order $N = 2^{n-1}$





- Example: CA $\langle 4, 1, f \rangle$, $f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$ (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 n bipermutive CA generating n-MOLS

MOCA by Linear CA

- Bipermutive Linear rule: $f(x) = x_1 \oplus a_2 x_2 \oplus \cdots \oplus a_{n-1} x_{n-1} \oplus x_n$
- Associated polynomial: $f \mapsto P_f(X) = a_1 + a_2 X + \cdots + a_n X^{n-1}$

Theorem ([Mariot16])

A set of bipermutive linear CA are MOCA if and only if their associated polynomials are pairwise 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)

Inversion Problem in OCA

- **Input**: A pair $w, z \in \{0,1\}^{n-1}$ of final configurations
- **Output**: The unique preimage x generating w, z under the action of two OCA

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

(a) rule 90-150

?	?	?	?
	0	1	
	1	1	

(b) Input

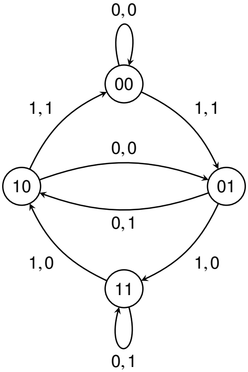
1	1	0	0
	0	1	
	1	1	

(c) Output

Coupled De Bruijn Graph

Idea: Walk on the De Bruijn graph labelled with both rules until a matching path is found.

(x_1, x_2, x_3)	f_{90}	f_{150}
000	0	0
100	1	1
010	0	1
110	1	0
001	1	1
101	0	0
011	1	0
111	0	1

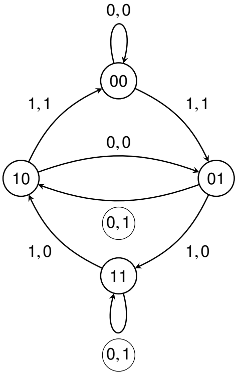


?	?	?	?
	0	1	
	1	1	

Coupled De Bruijn Graph

Idea: Walk on the De Bruijn graph labelled with both rules until a matching path is found.

(x_1, x_2, x_3)	f_{90}	f_{150}
000	0	0
100	1	1
010	0	1
110	1	0
001	1	1
101	0	0
011	1	0
111	0	1

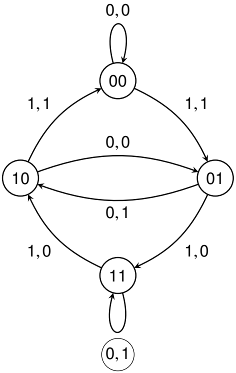


?	?	?	?
	0	1	
	1	1	

Coupled De Bruijn Graph

Idea: Walk on the De Bruijn graph labelled with both rules until a matching path is found.

(x_1, x_2, x_3)	f_{90}	f_{150}
000	0	0
100	1	1
010	0	1
110	1	0
001	1	1
101	0	0
011	1	0
111	0	1

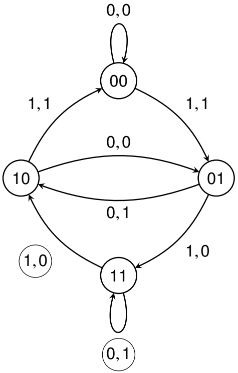


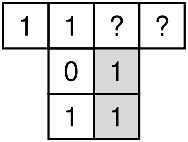
1	1	?	?
	0	1	
	1	1	

Coupled De Bruijn Graph

Idea: Walk on the De Bruijn graph labelled with both rules until a matching path is found.

(x_1, x_2, x_3)	f_{90}	f_{150}
000	0	0
100	1	1
010	0	1
110	1	0
001	1	1
101	0	0
011	1	0
111	0	1

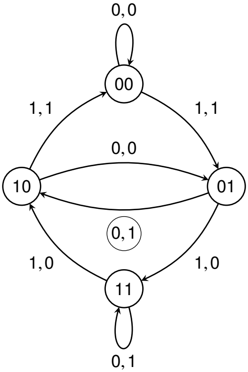


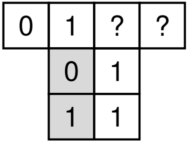


Coupled De Bruijn Graph

Idea: Walk on the De Bruijn graph labelled with both rules until a matching path is found.

(x_1, x_2, x_3)	f_{90}	f_{150}
000	0	0
100	1	1
010	0	1
110	1	0
001	1	1
101	0	0
011	1	0
111	0	1

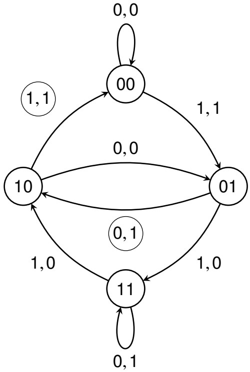




Coupled De Bruijn Graph

Idea: Walk on the De Bruijn graph labelled with both rules until a matching path is found.

(x_1, x_2, x_3)	f_{90}	f_{150}
000	0	0
100	1	1
010	0	1
110	1	0
001	1	1
101	0	0
011	1	0
111	0	1



0	1	0	0
	0	1	
	1	1	

Inversion Algorithm

$\textsc{Invert-OCA}(G_{DB}(f,g), w, z)$

$V := \textsc{Vertex}(G_{DB}(f,g))$
$E := \textsc{Edges}(G_{DB}(f,g))$
$l := \textsc{Labels}(G_{DB}(f,g))$
$c := NIL$
while $e \in \{(v_1, v_2) \in E : \ l(v_1, v_2) = (w_1, z_1)\}$ AND $c = NIL$ **do**
  $c := \text{DFS-}\textsc{Mod}(V, E, l, v_1, w, z)$
end while
return c

Theorem

Given two OCA rules $f, g : \{0,1\}^n \to \{0,1\}$ and two final configurations $w, z \in \{0,1\}^{n-1}$, algorithm $\textsc{Invert-OCA}$ *returns the preimage $x \in \{0,1\}^{2(n-1)}$ of w, z in $\mathcal{O}(n \cdot 2^n)$ steps*

Conclusions and Future Directions

Summing up:

- We considered the problem of inverting a pair of final configurations under the action of two OCA
- We devised an algorithm which solves the problem in exponential time wrt the CA diameter (but can be brought down to linear with parallelization!)

Future directions:

- Design a cheater-immune SSS based on INV-OCA
- Apply Genetic Programming (GP) to evolve MOCA with compact representation

References

- [delRey05] del Rey, Á.M., Mateus, J.P., Sánchez, G.R.: A secret sharing scheme based on cellular automata. Appl. Math. Comput. 170(2), 1356–1364 (2005)
- [Eloranta93] Eloranta, K.: Partially Permutive Cellular Automata. Nonlinearity 6(6), 1009–1023 (1993)
- [Mariot17] Mariot, L., Picek, S., Jakobovic, D., Leporati, A.: Evolutionary Algorithms for the Design of Orthogonal Latin Squares based on Cellular Automata. In: Proceedings of GECCO'17 (2017)
- [Mariot16] Mariot, L., Formenti, E., Leporati, A.: Construting Orthogonal Latin Squares from Linear Cellular Automata. In: Exploratory papers of AUTOMATA 2016 (2016)
- [Mariot14] Mariot, L., Leporati, A.: Sharing Secrets by Computing Preimages of Bipermutive Cellular Automata. In: Proceedings of ACRI 2014. LNCS vol. 8751, pp. 417–426. Springer (2014)
- [Shamir79] Shamir, A.: How to share a secret. Commun. ACM 22(11):612–613 (1979)
- [Tompa88] Tompa, M., Woll, H.: How to share a secret with cheaters. J. Cryptology 1(2), 133–138 (1988)