Embedding Reversible Logic Elements in Cellular Automata

Kenichi Morita, Katsunobu Imai,
Shiho Hyodo, Atsushi Kanno, Hiroko Kato,
Tsuyoshi Ogiro, and Keiji Tanaka
Hiroshima University

Outline of the Talk

- Computation-universality and logical universality are interesting problems in the theory of CAs.
- When studying logical universality of reversible CAs, reversible logic elements are useful.
- There are two types of reversible logic elements: with and without memory.
- We investigate which reversible logic elements have logical universality.
- We show how such universal reversible logic elements can be embedded in simple RCAs.

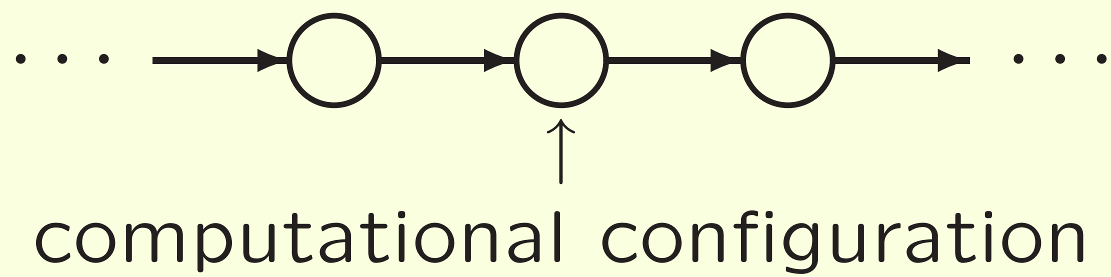
Contents

1. Reversible Computing and
   Reversible Cellular Automata (RCAs)
2. Reversible Logic Elements (RLEs)
3. How to Embed RLEs in RCAs

1. Reversible Computing and Reversible Cellular Automata (RCAs)

Reversible Computing

- A "backward deterministic" computing;
  i.e., every computational configuration has at most one predecessor.



- It reflects "physical reversibility".
- Related to energy consumption in computing.

Several Models of Reversible Computing

- Reversible Turing Machines (RTM)

  Computation-universality of an RTM [Bennett, 1973]
- Reversible Counter Machines (RCM)
- Reversible Logic Elements and Circuits
- Reversible Cellular Automata (RCA)
- etc.

Reversible Cellular Automata (RCAs)

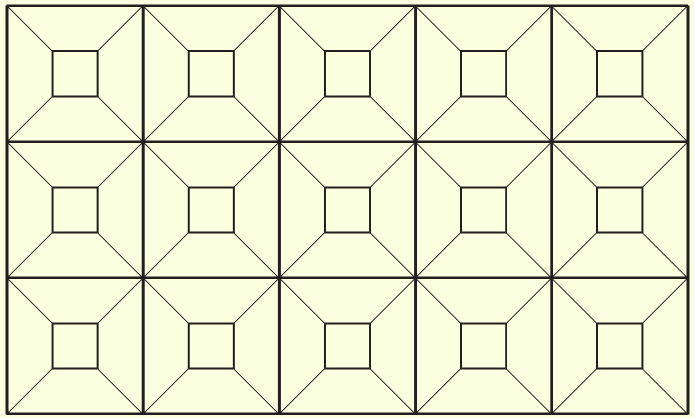
In spite of the strong restriction of reversibility, they have rich ability of computing.

- Computation-universality:
  Any recursive function is computed in an RCA (by finite means).
- Logical universality:
  Any sequential machine is embedded in an RCA.
- Self-reproduction
- etc.

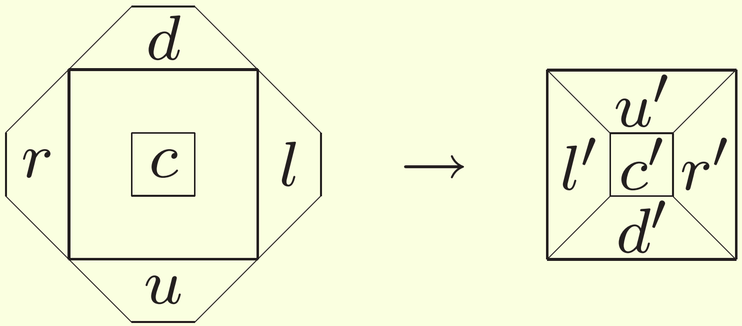
How Can We Design an RCA?

- Difficulty of designing an RCA.
  - Reversibility for 2-D CA is undecidable. [Kari, 1994]
  - Reversibility for 1-D CA is decidable but not simple. [Amoroso and Patt, 1972]
- A partitioned CA (PCA). [Morita and Harao, 1989]
  - A subclass of (usual) CAs.
  - *Useful for designing an RCA*.
  - Injectivity of a local function is equivalent to that of the global function.

A 2-D 5-Neighbor Partitioned CA (PCA)

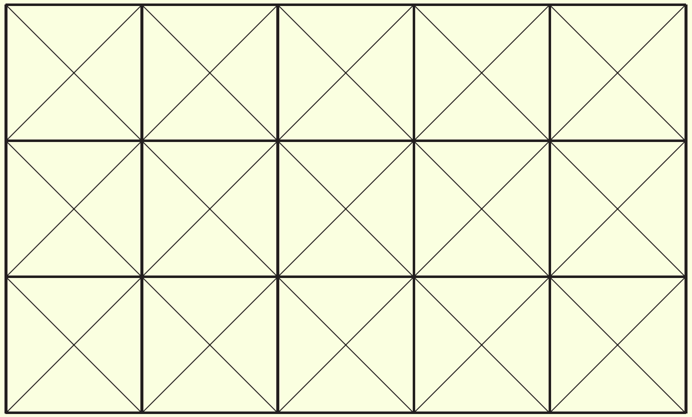


A cellular space of a 2-D 5-neighbor PCA.

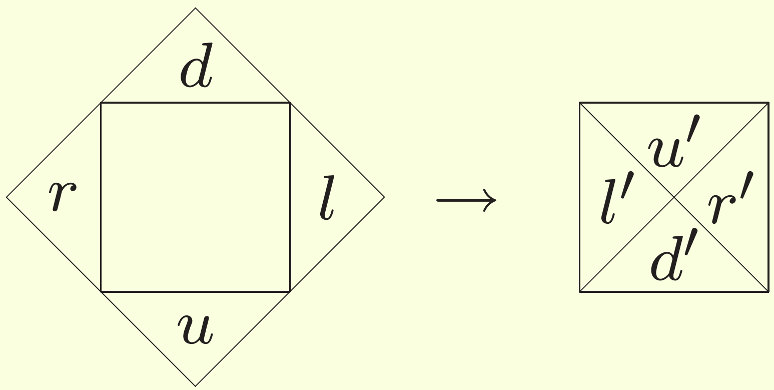


A local transition rule of a 5-neighbor PCA.

A 2-D 4-Neighbor Partitioned CA (PCA)



A cellular space of a 2-D 4-neighbor PCA.



A local transition rule of a 4-neighbor PCA.

Definition of a 2-D PCA

$$P = (\mathbf{Z}^2, (C, U, R, D, L), g, (\#, \#, \#, \#, \#))$$

- $\mathbf{Z}$: the set of all integers.
- C, U, R, D, and L: non-empty finite sets of states of center, up, right, down and left parts of each cell.
- g: $C \times D \times L \times U \times R \to C \times U \times R \times D \times L$: a local function.
- $(\#, \#, \#, \#, \#)$: a quiescent state satisfying $g(\#, \#, \#, \#, \#) = (\#, \#, \#, \#, \#)$.

Some Definitions on a 2-D PCA

$$P = (\mathbf{Z}^2, (C, U, R, D, L), g, (\#, \#, \#, \#, \#))$$

- The global function G:

$$\begin{aligned}&\forall (x, y) \in \mathbf{Z}^2 : \\ &\quad G(\alpha)(x, y) \; = \\ &\qquad g(\mathtt{CENTER}(\alpha(x, y)), \mathtt{DOWN}(\alpha(x, y + 1)), \\ &\qquad\quad \mathtt{LEFT}(\alpha(x + 1, y)), \mathtt{UP}(\alpha(x, y - 1)), \\ &\qquad\quad \mathtt{RIGHT}(\alpha(x - 1, y)))\end{aligned}$$

- P is globally reversible iff G is one-to-one.
- P is locally reversible iff g is one-to-one.

Properties of 2-D PCA

Theorem 1 [Morita and Harao, 1989]
Let P be a PCA. P is globally reversible iff it is locally reversible.

Theorem 2 [Morita and Harao, 1989]
For any PCA P, there is a CA A whose global function is identical with that of P.

(I.e., PCA is a subclass of CA.)

- To construct an RCA, it is sufficient to give a PCA whose local function g is one-to-one.

Methods to Show Computation-/Logical Universality of RCAs

- Computation-universality:
  - Simulating an irreversible CA by an RCA.
  - Simulating a reversible Turing machine (TM) by an RCA.
- Logical universality:
  - Embedding a universal reversible logic element in an RCA.

Computation-Universality of an RCA

Theorem 3 [Toffoli, 1977]
For any k-D (irreversible) CA, there exists a $(k+1)$-D RCA which simulates the former in real time.

Theorem 4 [Morita, 1995]
For any 1-D irreversible CA with finite configurations, there is a 1-D RPCA which simulates the former (but not in real time).

Theorem 5 [Morita and Harao, 1989]
For any reversible TM, there is a 1-D RPCA which simulates the former.

Another Proof of Toffoli's Theorem (1)

Theorem 3 [Toffoli, 1977]
For any k-D (irreversible) CA, there exists a $(k+1)$-D RCA which simulates the former in real time.

Proof outline: $A = (\mathbf{Z}, Q, f, \#)$: a 1-D CA.

$t = 0$

q_1^0	q_2^0	q_3^0	q_4^0

$t = 1$

q_1^1	q_2^1	q_3^1	q_4^1

$t = 2$

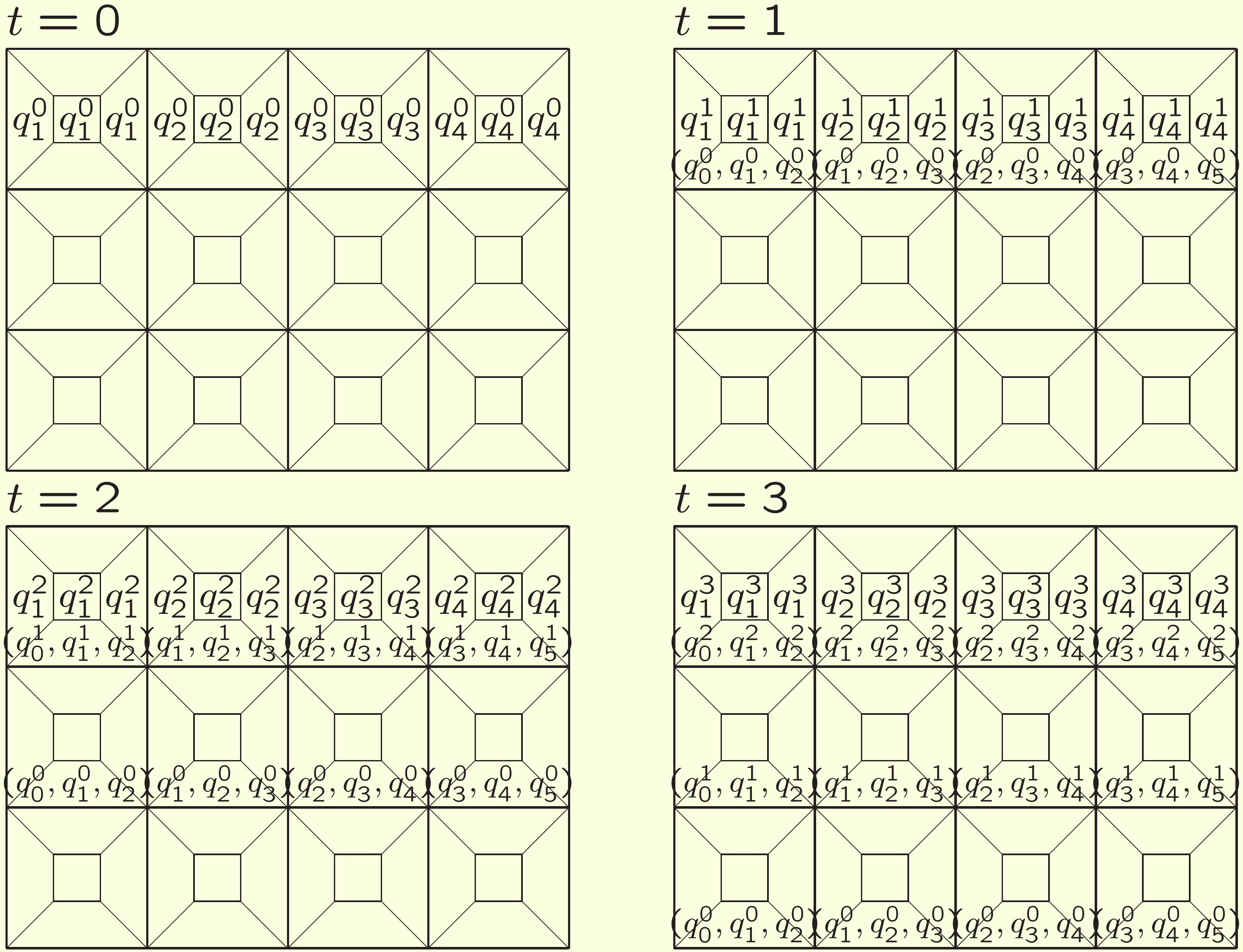
q_1^2	q_2^2	q_3^2	q_4^2

$t = 3$

q_1^3	q_2^3	q_3^3	q_4^3

Another Proof of Toffoli's Theorem (2)

A 2-D RPCA P that simulates A

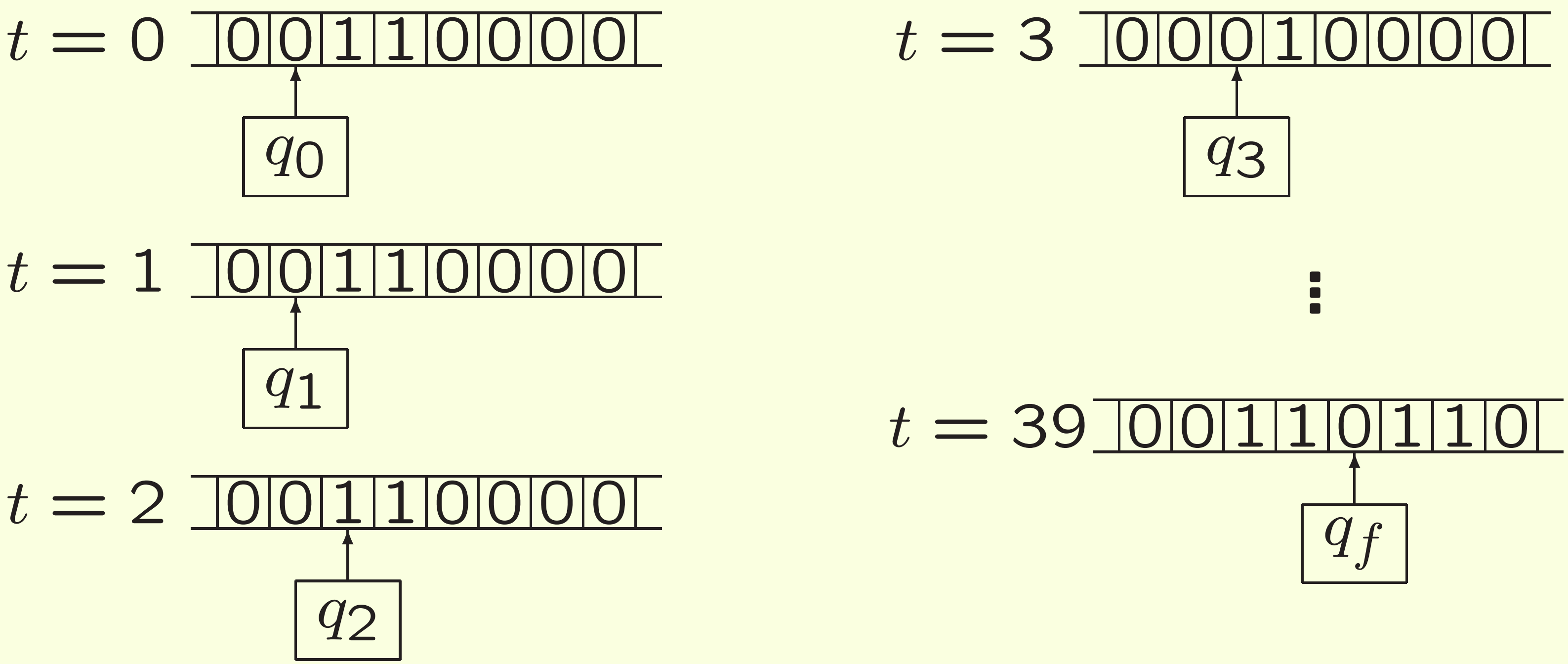


Simulating an RTM by an RCA (1)

Theorem 5 [Morita and Harao, 1989]

For any reversible TM, there is a 1-D RPCA which simulates the former.

An example of an RTM T_{copy}:



Simulating an RTM by an RCA (2)

Simulation process of T_{copy} by a 1-D RPCA P_{copy}:

$t=0$	#	0	#	#	$\overset{q_0}{0}$	#	#	1	#	#	1	#	#	0	#	#	0	#	#	0	#	#	0	#
$t=1$	#	0	#	#	$\overset{q_1}{0}$	#	#	1	#	#	1	#	#	0	#	#	0	#	#	0	#	#	0	#
$t=2$	#	0	#	#	0	q_2	#	1	#	#	1	#	#	0	#	#	0	#	#	0	#	#	0	#
$t=3$	#	0	#	#	0	#	#	$\overset{q_2}{1}$	#	#	1	#	#	0	#	#	0	#	#	0	#	#	0	#
$t=4$	#	0	#	#	0	#	#	$\overset{q_3}{0}$	#	#	1	#	#	0	#	#	0	#	#	0	#	#	0	#
⋮																								
$t=58$	#	0	#	#	0	#	#	1	#	#	1	#	#	$\overset{q_f}{0}$	#	#	1	#	#	1	#	#	0	#
$t=59$	#	0	#	#	0	#	#	1	#	#	1	#	$*$	$\overset{*}{0}$	#	#	1	#	#	1	#	#	0	#

2. Reversible Logic Elements (RLEs)

Universal Logic Elements

Definition A set of logic elements is called *logically universal* if any sequential machine can be built as a circuit composed only of the elements in the set.

- It is very well known that the set

  $$\{\ \text{AND},\ \text{NOT},\ \text{delay}\ \}$$

  is logically universal.

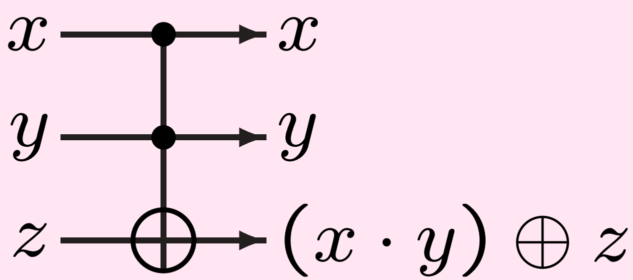
Two Types of Reversible Logic Elements (RLE)

- An RLE without memory (reversible logic gates):
  A gate whose logical function is one-to-one.
- An RLE with memory:
  A Mealy-type sequential machine whose move function is one-to-one.

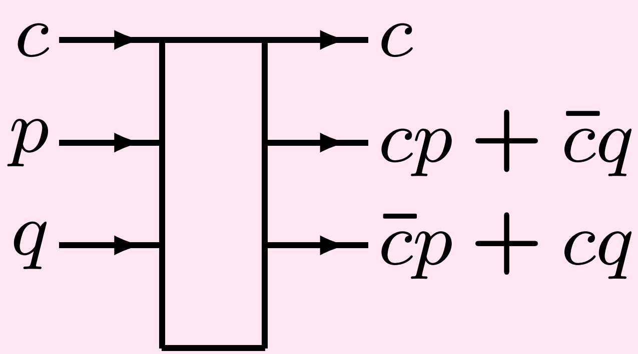
Note: In reversible computing, RLEs with memory are often very useful.

Typical Universal Reversible Logic Gates

Toffoli gate



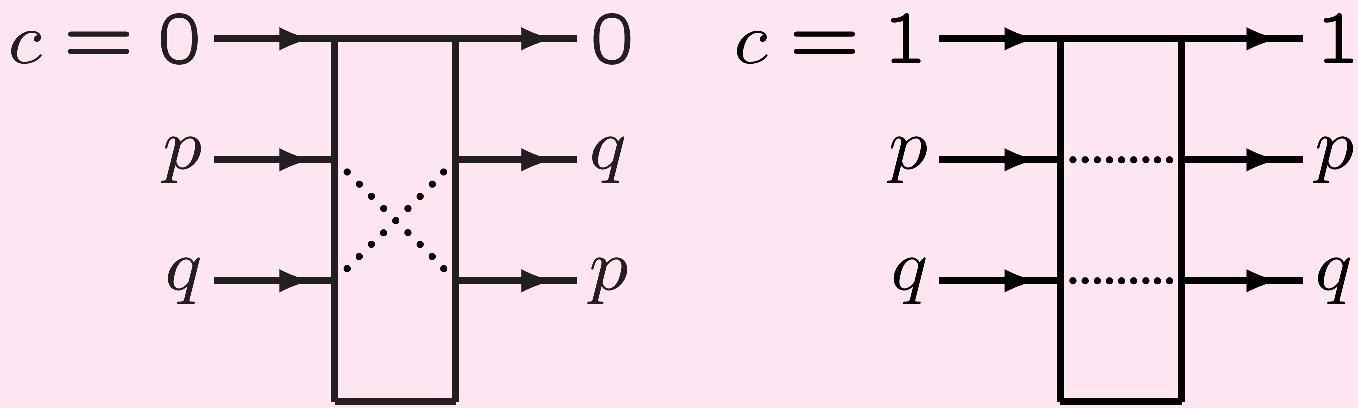
Fredkin gate



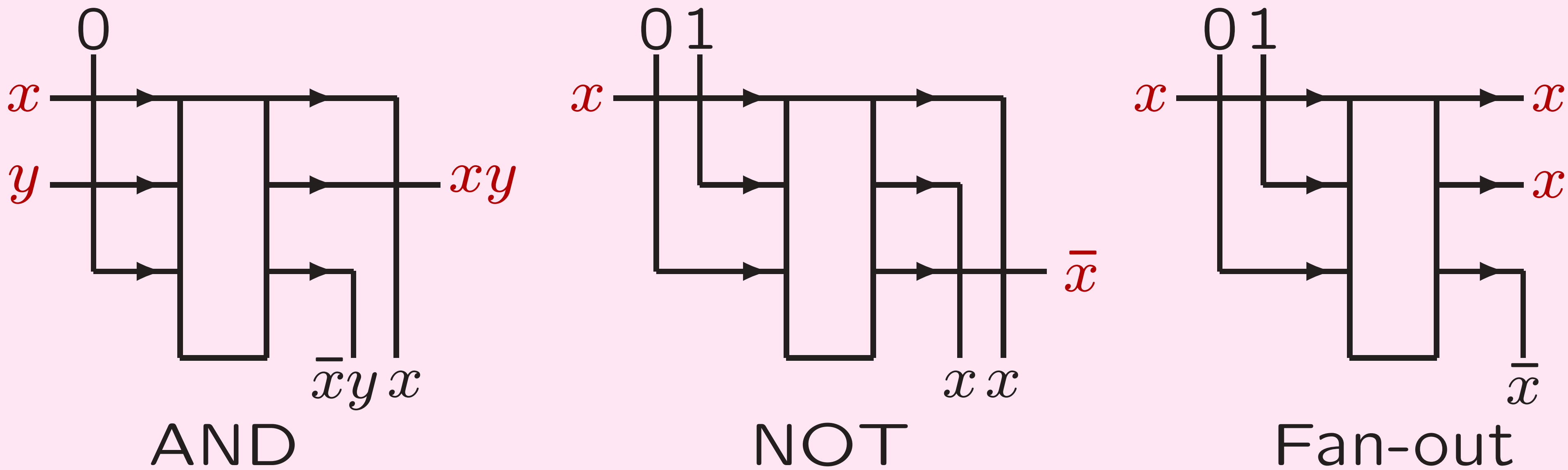
Input			Toffoli gate			Fredkin gate		
0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	1	0
0	1	0	0	1	0	0	0	1
0	1	1	0	1	1	0	1	1
1	0	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0	1
1	1	0	1	1	1	1	1	0
1	1	1	1	1	0	1	1	1

Fredkin gate (F-gate)

[Fredkin and Toffoli, 1982]



Logical Universality of {F-gate, delay}



A “Garbage-Less” Reversible Logic Circuit

- Any combinatorial logic circuit can be realized as a garbage-less circuits composed of F-gates. [Fredkin and Toffoli, 1982]
- Any *reversible* sequential machine can be realized as a garbage-less circuits composed of F-gates and delays. [Morita, 1990]

Reversible Logic Elements with Memory (RLEM)

Definition A reversible logic elements with memory (RLEM) is defined by

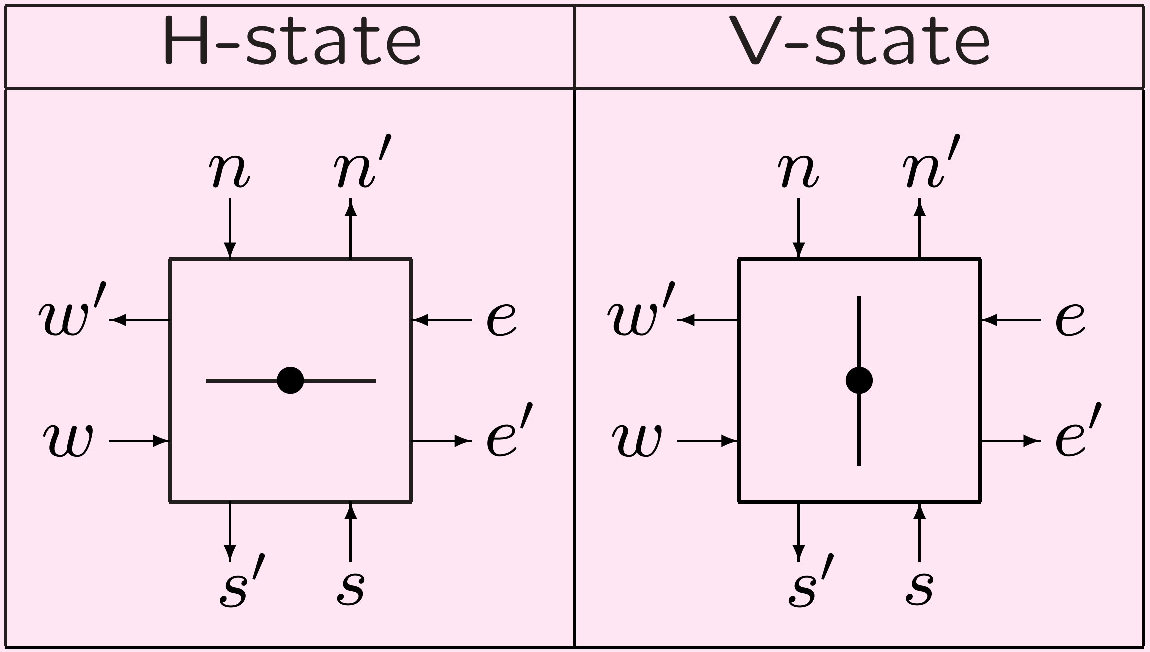
$$E = (Q, \Sigma, \Gamma, \delta).$$

Q: a finite nonempty set of states.
Σ: a finite nonempty set of input symbols.
Γ: a finite nonempty set of output symbols.
δ: $Q \times \Sigma \to Q \times \Gamma$,
a one-to-one mapping called a move function.

- It is in fact a *reversible sequential machine*.

Rotary Element (RE)
— A Typical 2-State 4-Symbol RLEM —

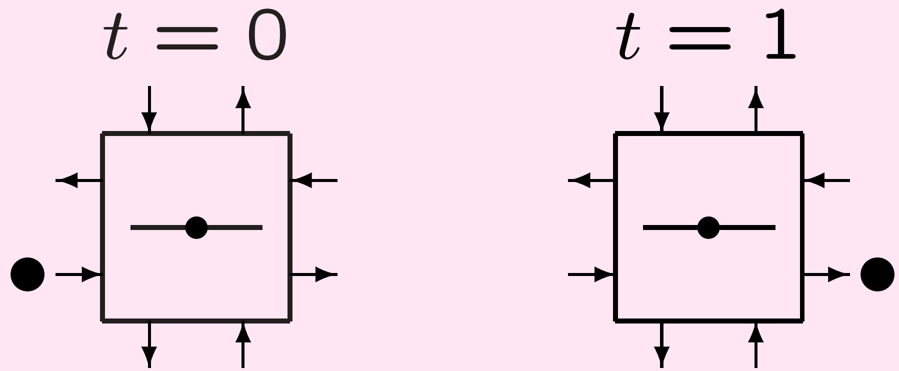
[Morita, Tojima, and Imai, 2001]



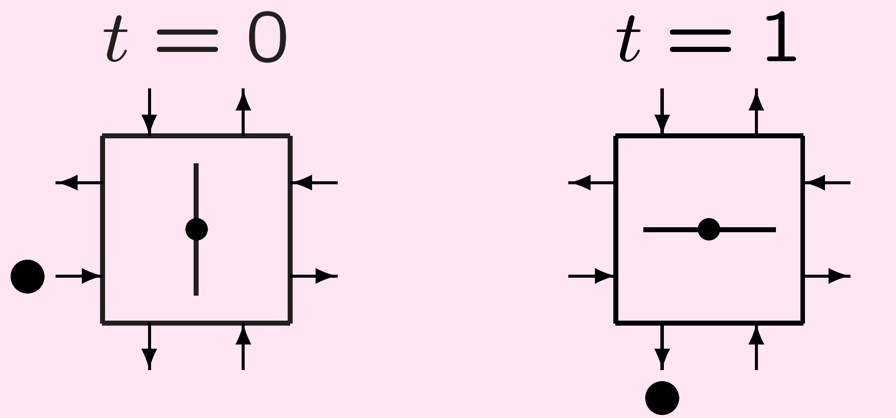
(Remark) Signal “1” should be given at most one input line.

Operations of an RE

- Parallel case:

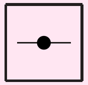
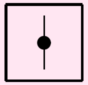


- Orthogonal case:

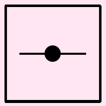
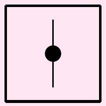
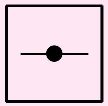
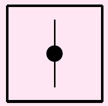
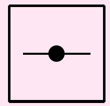
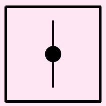
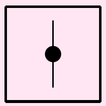
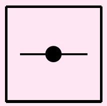
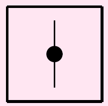
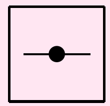


(Remark) Reversibility is easily verified.

An RE Formalized as a Reversible Sequential Machine

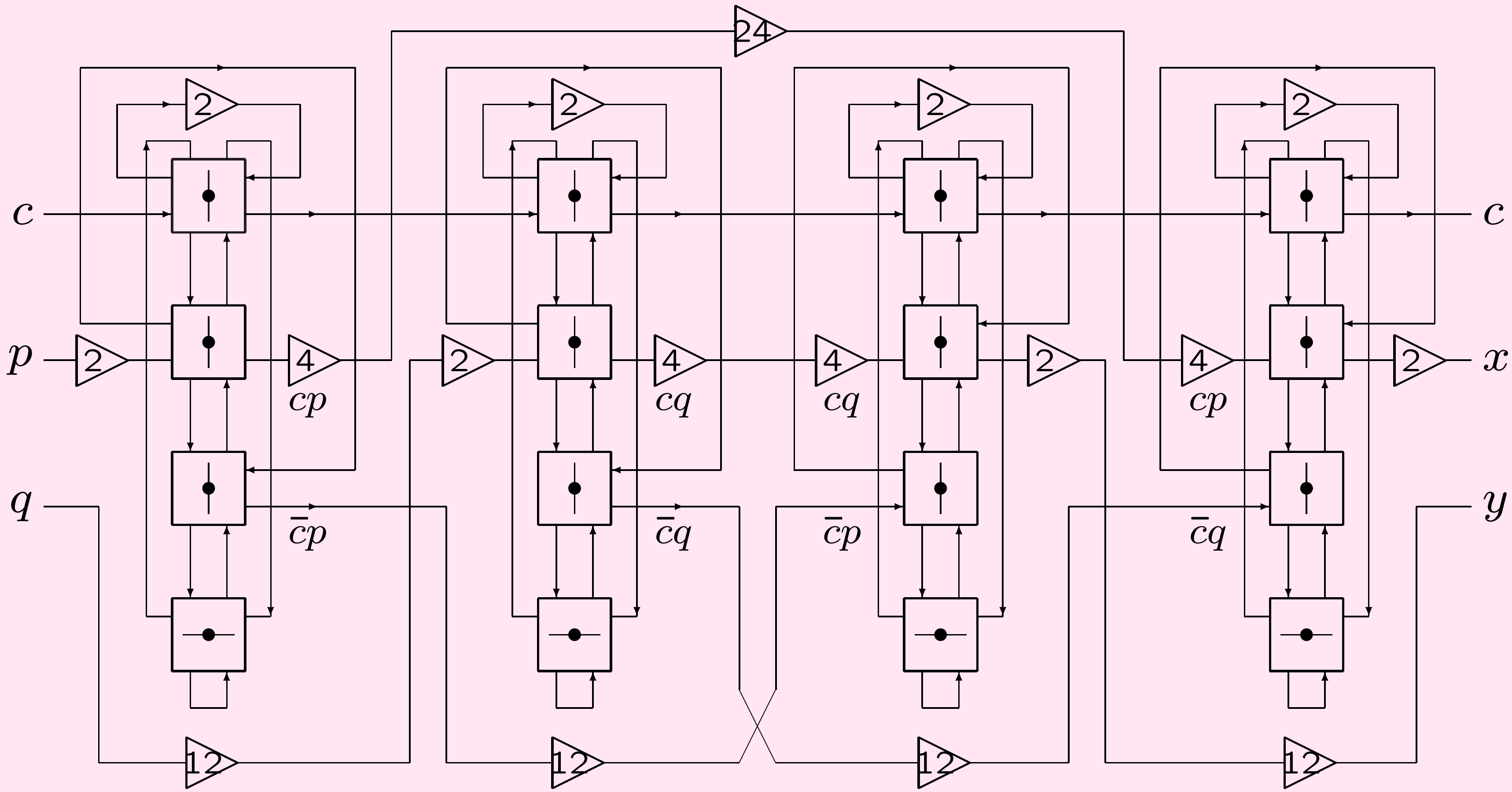
$M_{\mathsf{RE}} = (\{$  $,$  $\}, \{n, e, s, w\}, \{n', e', s', w'\}, \delta_{\mathsf{RE}}),$

The move function δ_{RE}:

Present state	Input			
	n	e	s	w
H-state: 	 w'	 w'	 e'	 e'
V-state: 	 s'	 n'	 n'	 s'

An F-gate Is Realized by REs

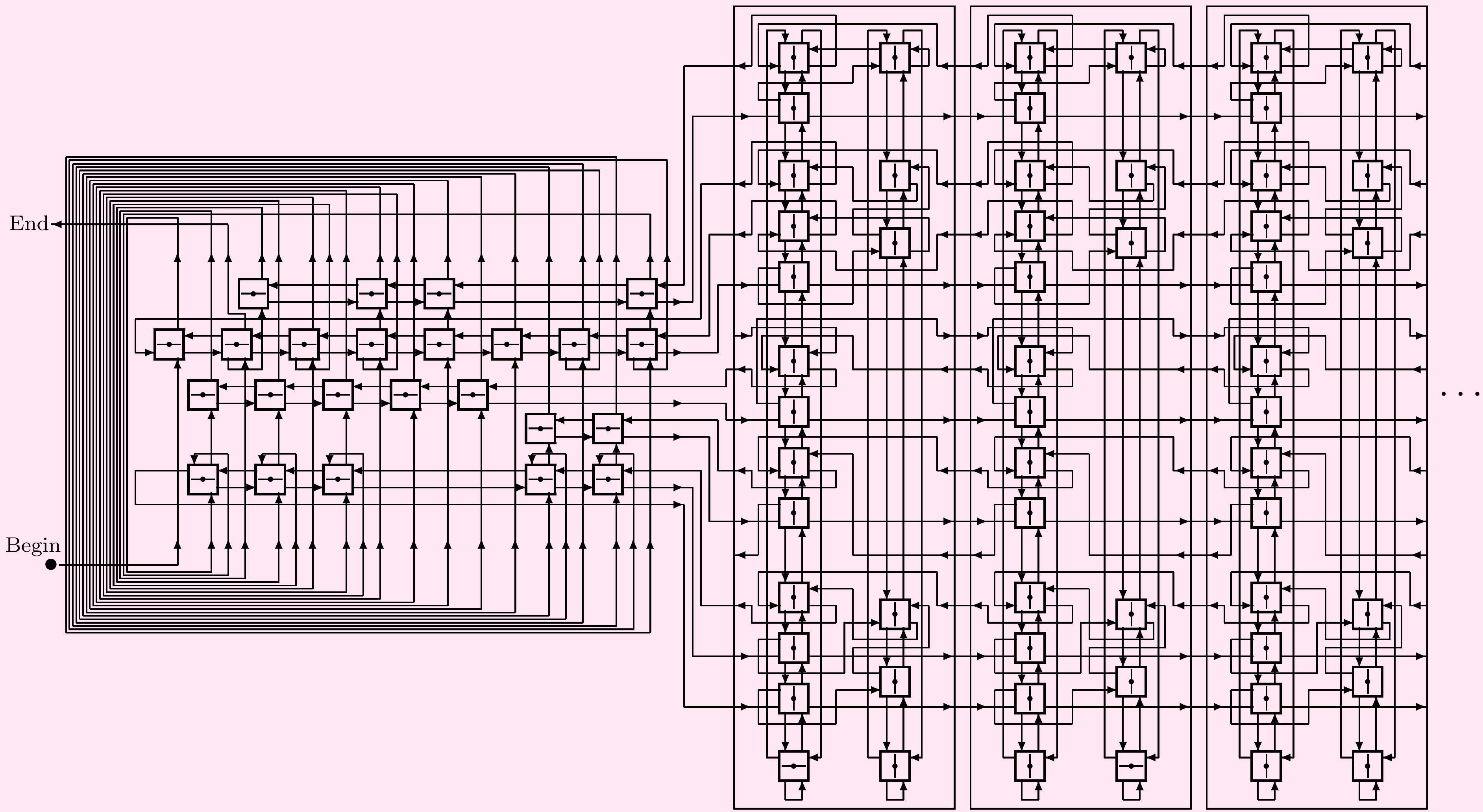
[Morita, 2001]



(Remark) A triangle shows a delay element.

A Reversible TM Is Directly Realized by REs

[Morita, 2001]



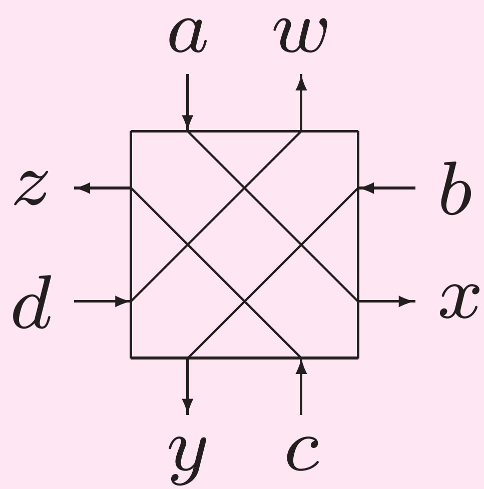
RLEMs Having Logical Universality

1. 2-state 4-symbol elements:
   - Rotary element (RE)
   - Left-/right-rotate element
   - etc.
2. 2-state 3-symbol elements:
   - 14 nondegenerate elements

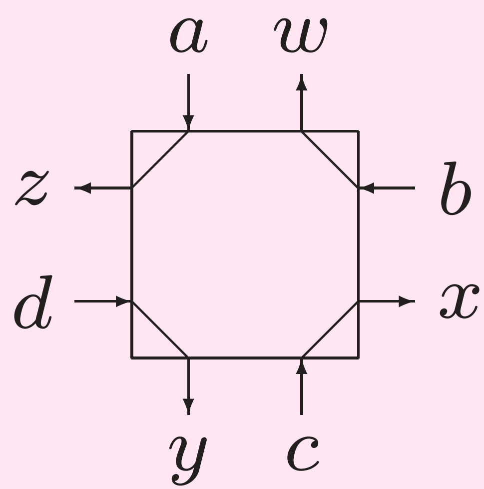
A New 2-State 4-Symbol Universal RLEM

[Morita, Ogiro, Tanaka, and Kato, to appear]

Left-/right-rotate element (LRRE)



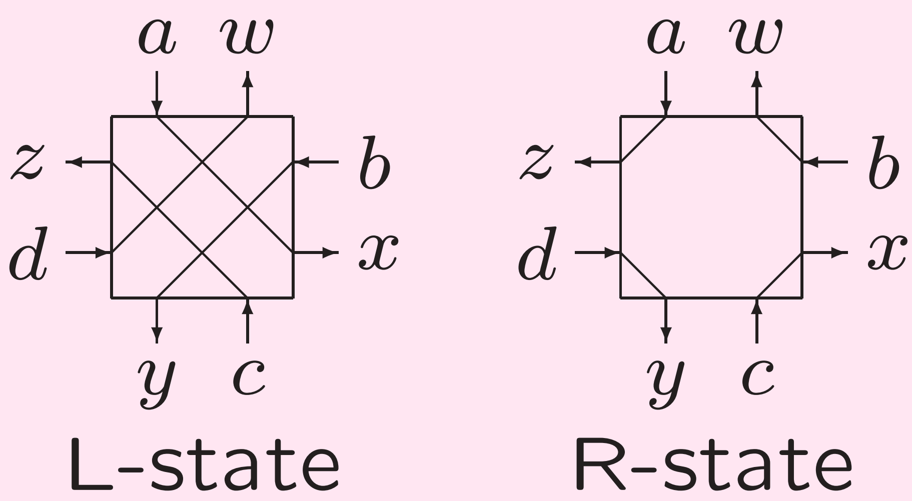
L-state



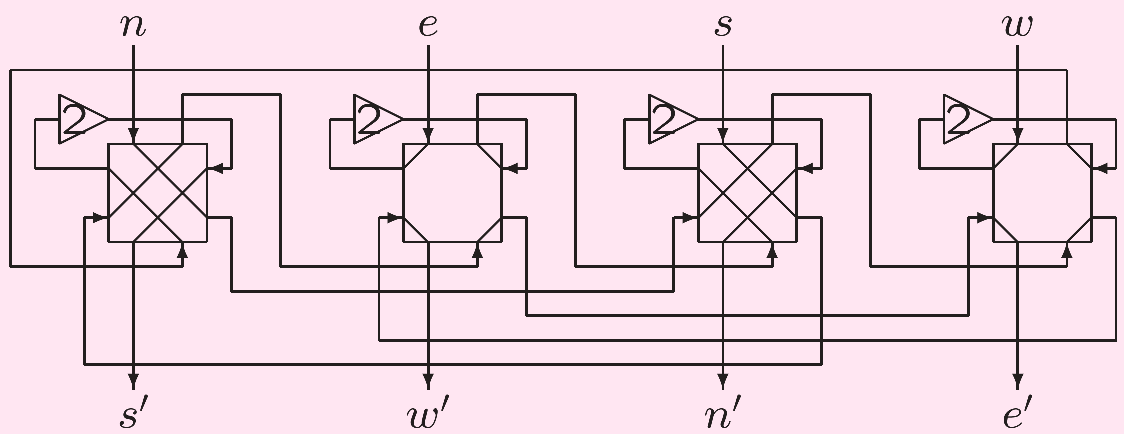
R-state

Move Function of a Left-/Right-Rotate Element

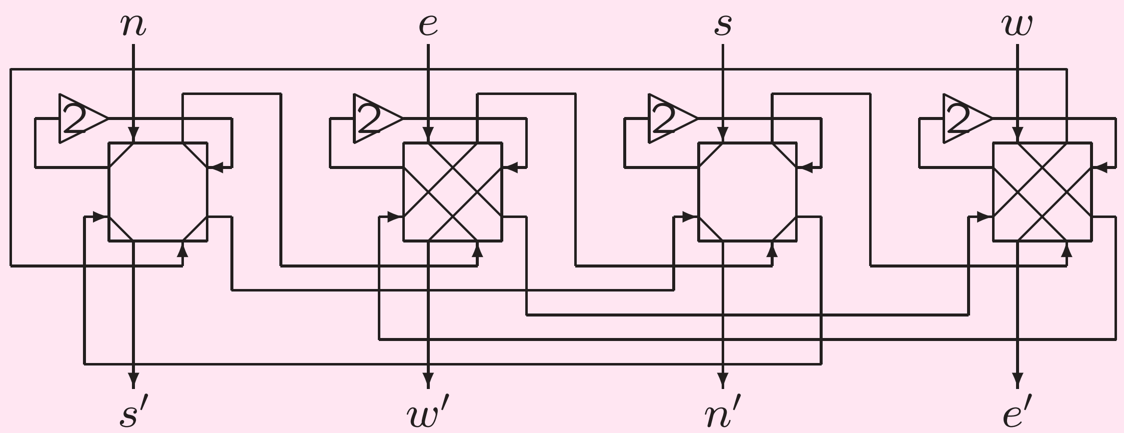
Present state	Input a	Input b	Input c	Input d
L-state	R x	R y	R z	R w
R-state	L z	L w	L x	L y



Constructing an RE out of Left-/Right-Rotate Elements



(a) An RE of H-state.



(b) An RE of V-state.

3. How to Embed RLEs in RCAs

3.1. Embedding a Fredkin gate
3.2. Embedding a rotary element (RE)
3.3. Embedding a left-/right-rotate element
3.4. Further problems

3.1. Embedding a Fredkin Gate

- A 2-state RCA with Margolus neighborhood

  [Margolus, 1984]

- A 16-state RPCA (model S_1) on square grid

  [Morita and Ueno, 1992]

- A 16-state RPCA (model S_2) on square grid

  [Morita and Ueno, 1992]

- An 8-state triangular RPCA (model T_1)

  [Imai and Morita, 1998]

- A 64-state hexagonal RPCA (model H_1)

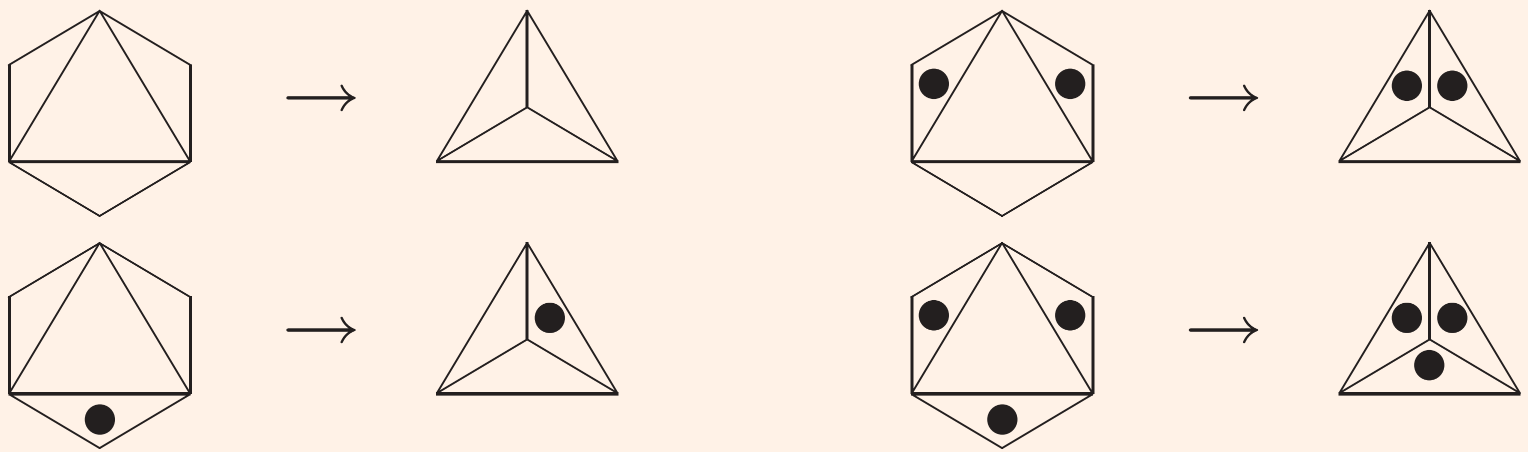
  [Morita, Margenstern and Imai, 1998]

An 8-State Triangular RPCA T_1

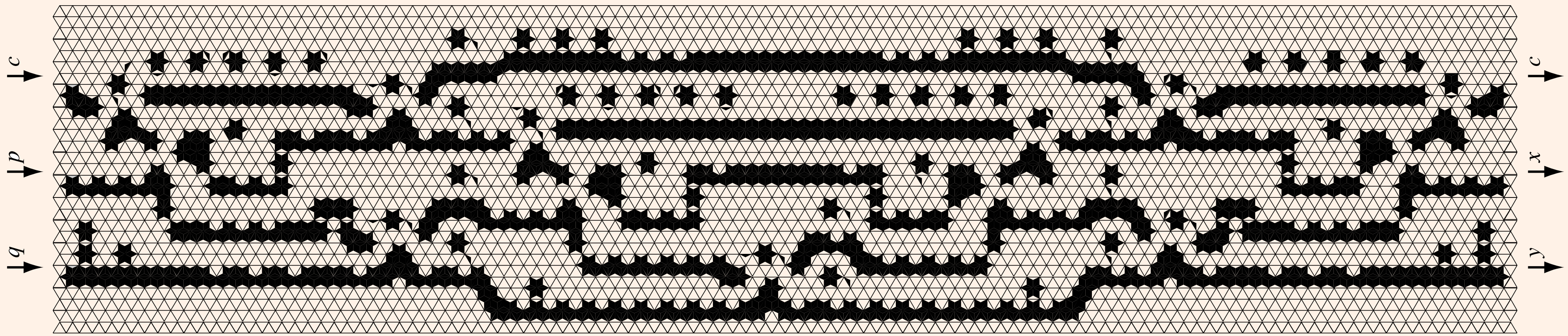
[Imai and Morita, 1998]

$$P_{\mathrm{tri}} = (\mathsf{Z}^2, \{0,1\}^3, g_{\mathrm{tri}}, (0,0,0))$$

- It has extremely simple local function:



An F-Gate in a Triangular 8-State RPCA T_1



Note: Signal routing, crossing, and delay (as well as F-gate itself) can be realized in this cellular space.

3.2. Embedding a Rotary Element

- A 4^4-state RPCA (model P_4)

  [Morita, Tojima and Imai, 2001]

- A 3^4-state RPCA (model P_3)

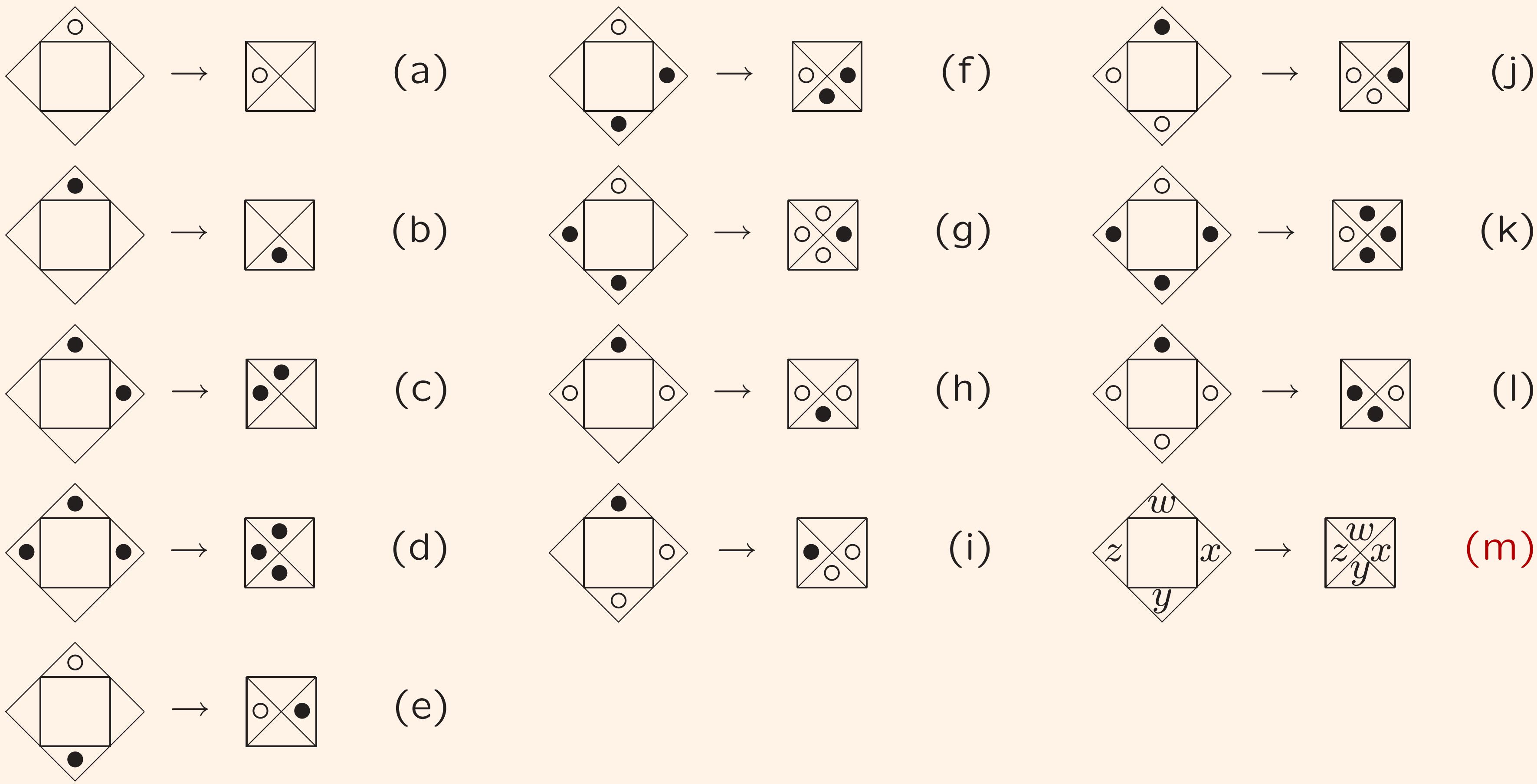
  [Ogiro and Morita, 2002]

- A 3^4-state RPCA (model P'_3)

  [Morita and Ogiro, 2004]

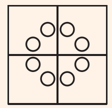
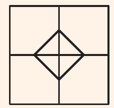
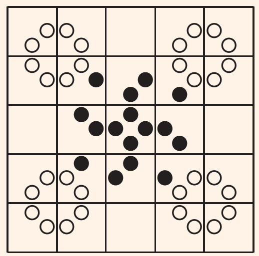
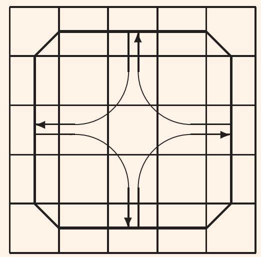
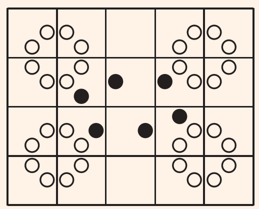
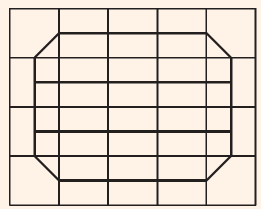
3.2 (1) 3^4-State Model P_3

$$P_3 = (\mathrm{Z}^2, \{0,1,2\}^4, g_3, (0,0,0,0))$$



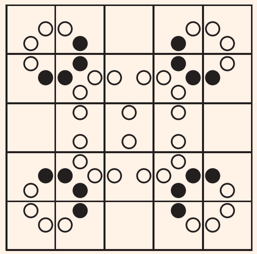
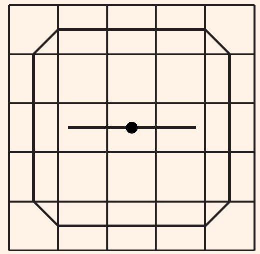
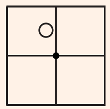
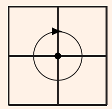
The rule scheme (m) represents 33 rules not specified by (a)–(l)
$(w, x, y, z \in \{$ blank, ○, ● $\} = \{0, 1, 2\})$.

Five Signal Processing Elements in P_3

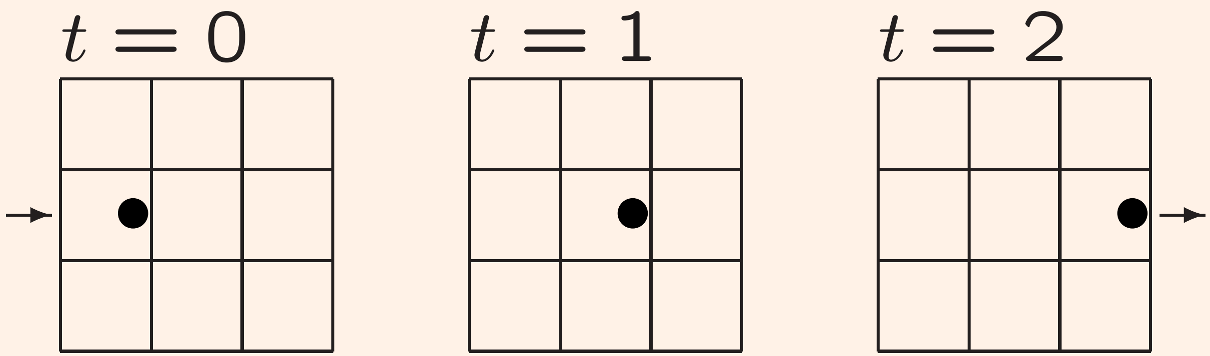
Element Name	Pattern	Symbolic Notation
LR-turn element		
R-turn element		
Reflector		

(Continues to the next page.)

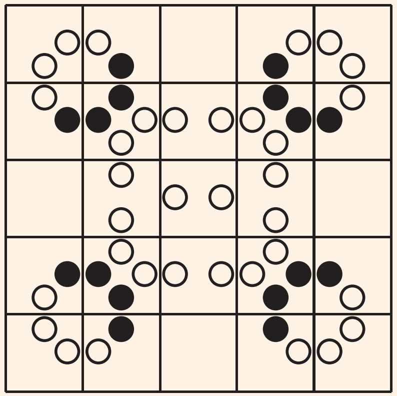
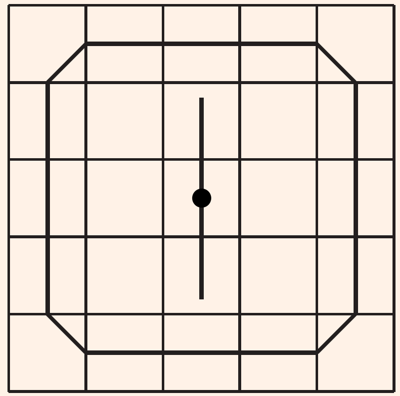
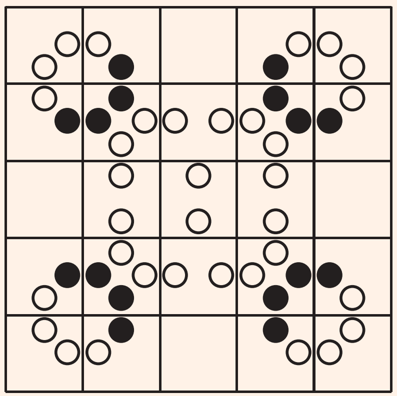
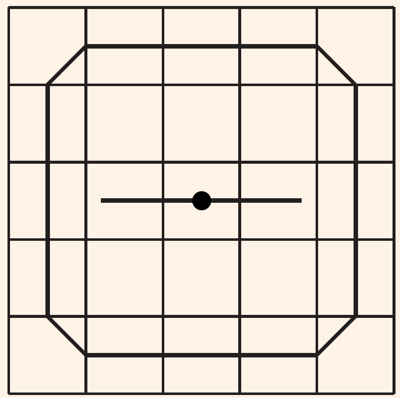
Five Signal Processing Elements in P_3 (Continued)

Element Name	Pattern	Symbolic Notation
Rotary element		
Position marker		

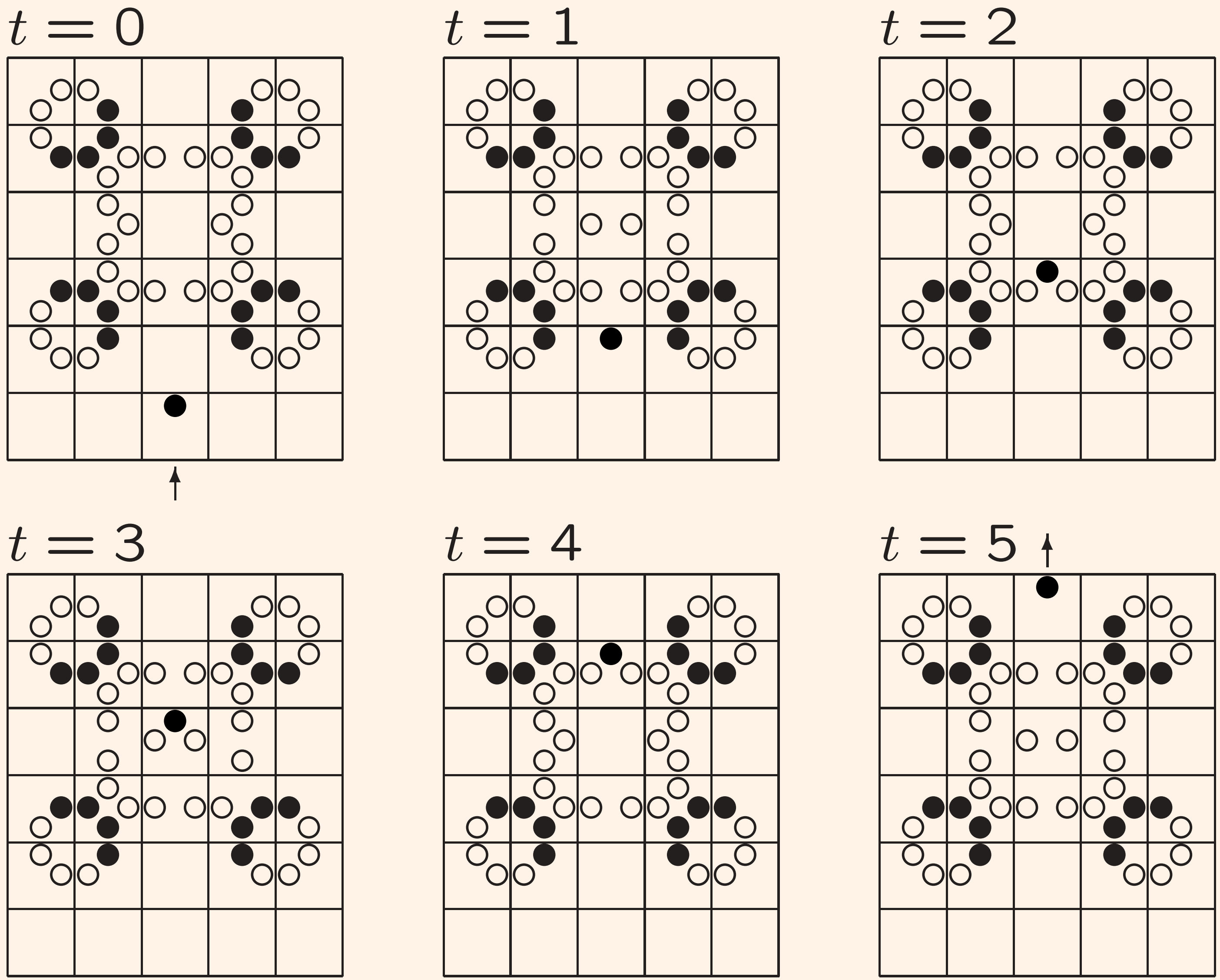
A Signal in P_3



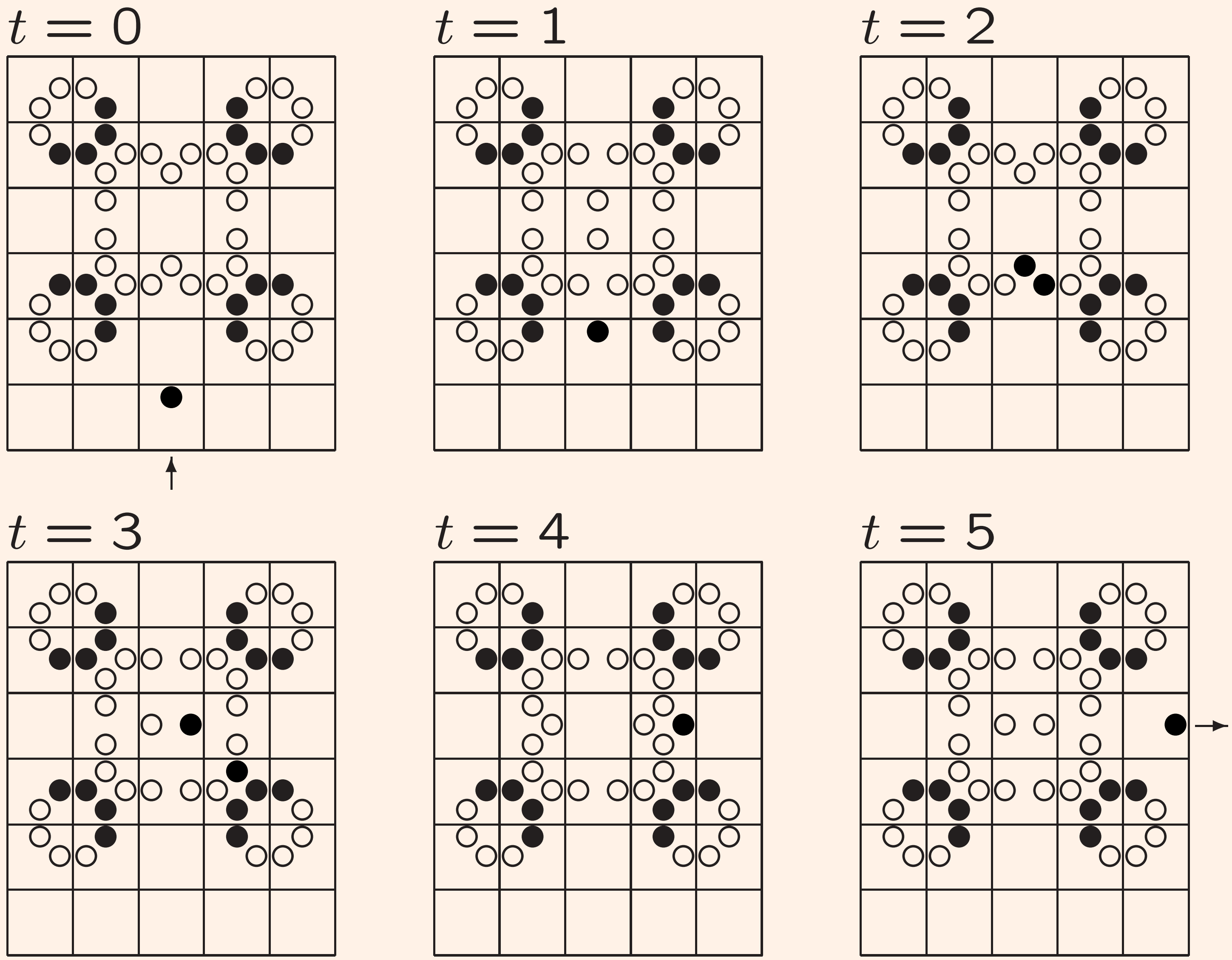
Two States of a Rotary Element in P_3

V-state		
H-state		

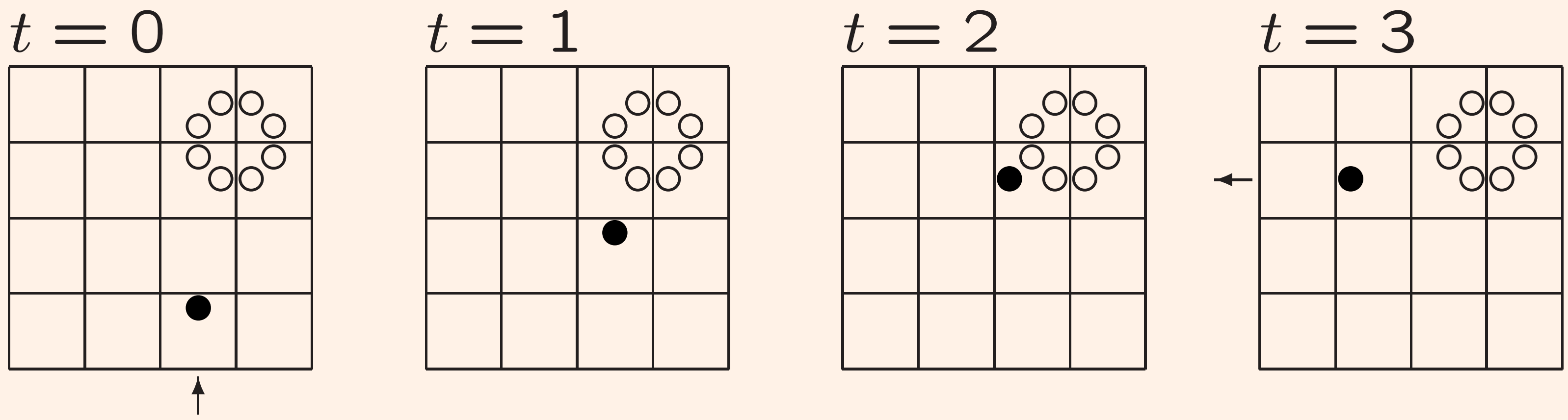
A Rotary Element in P_3 (Parallel Case)



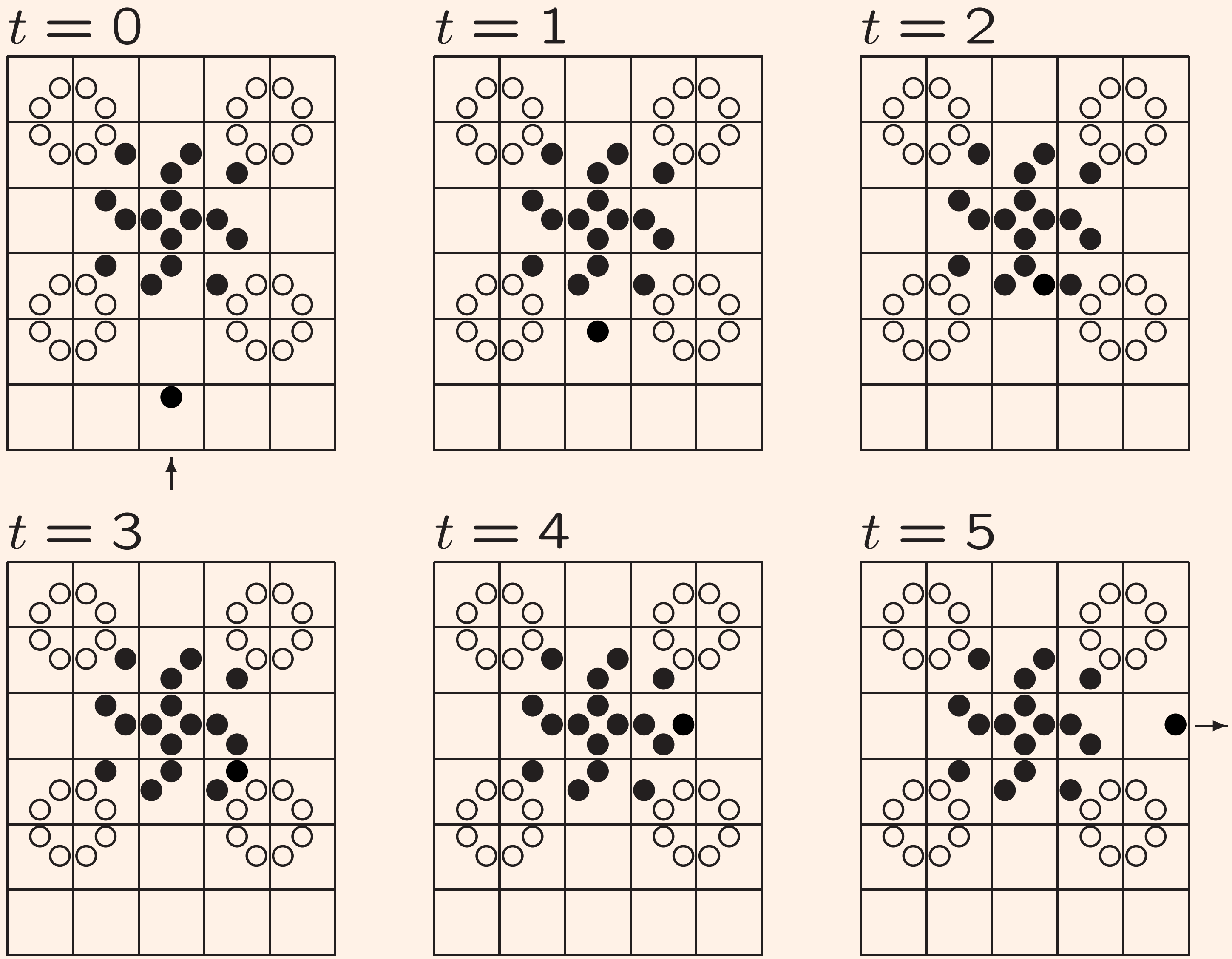
A Rotary Element in P_3 (Orthogonal Case)



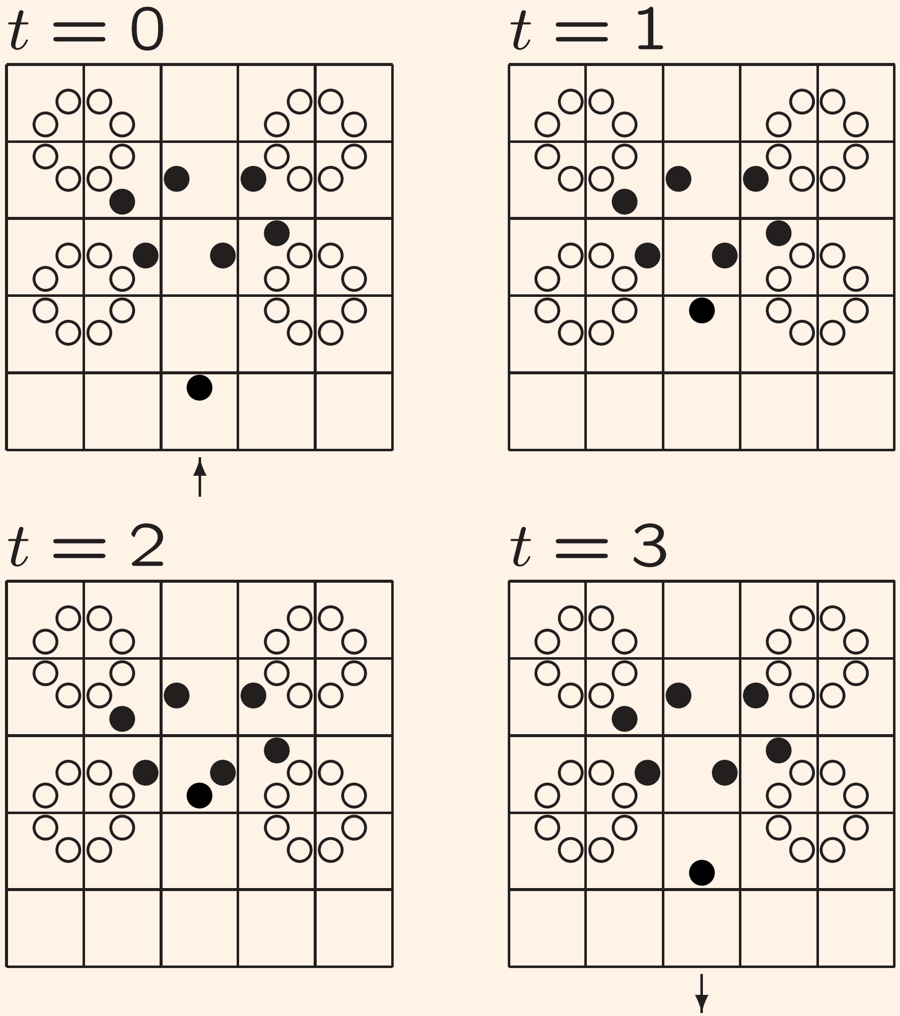
An LR-Turn Element in P_3



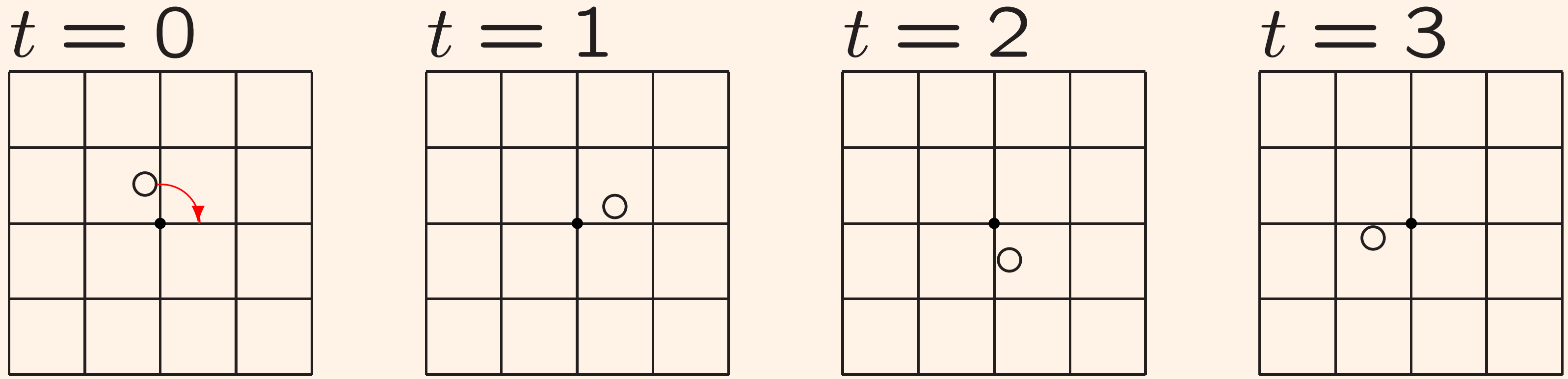
An R-Turn Element in P_3



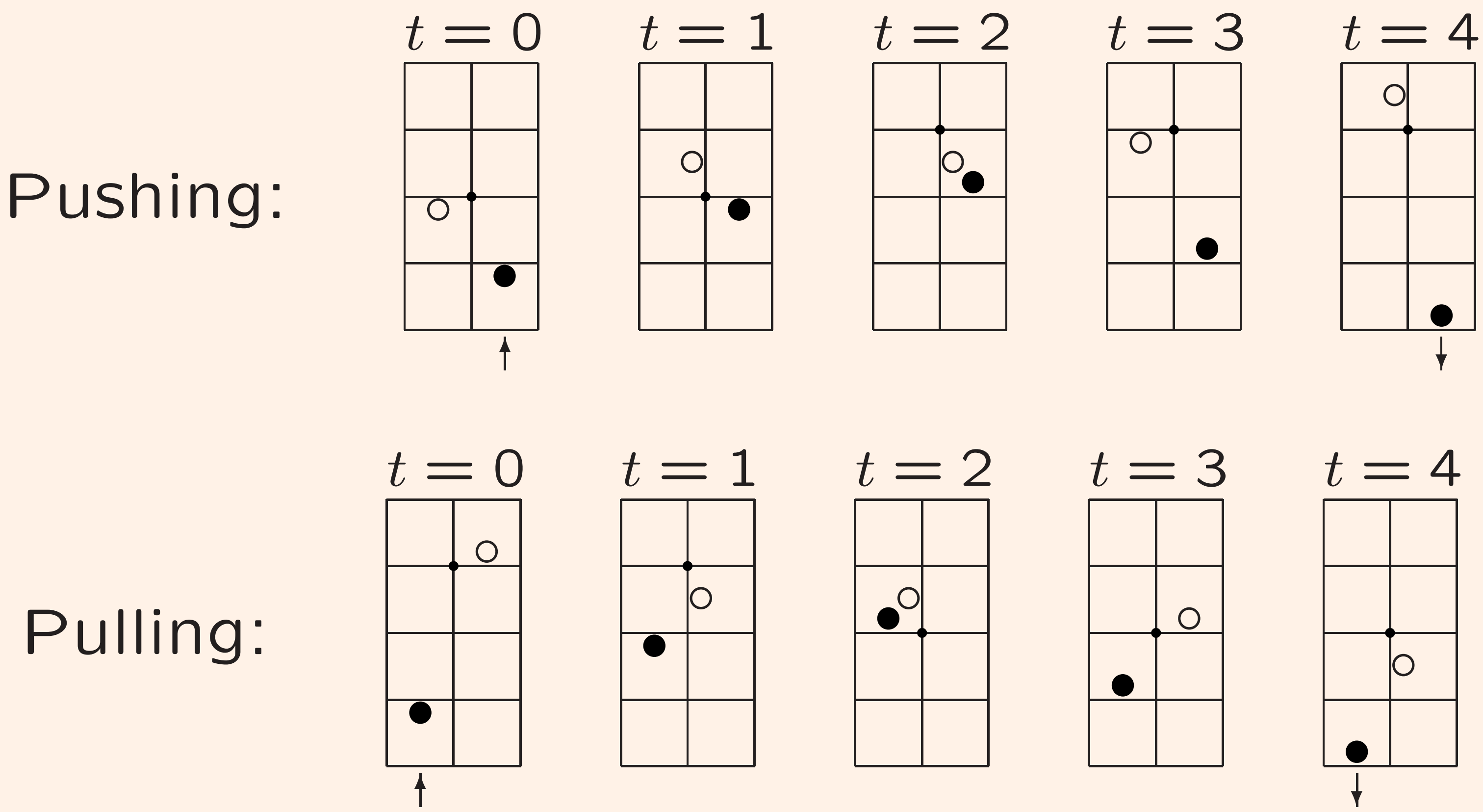
A Reflector in P_3



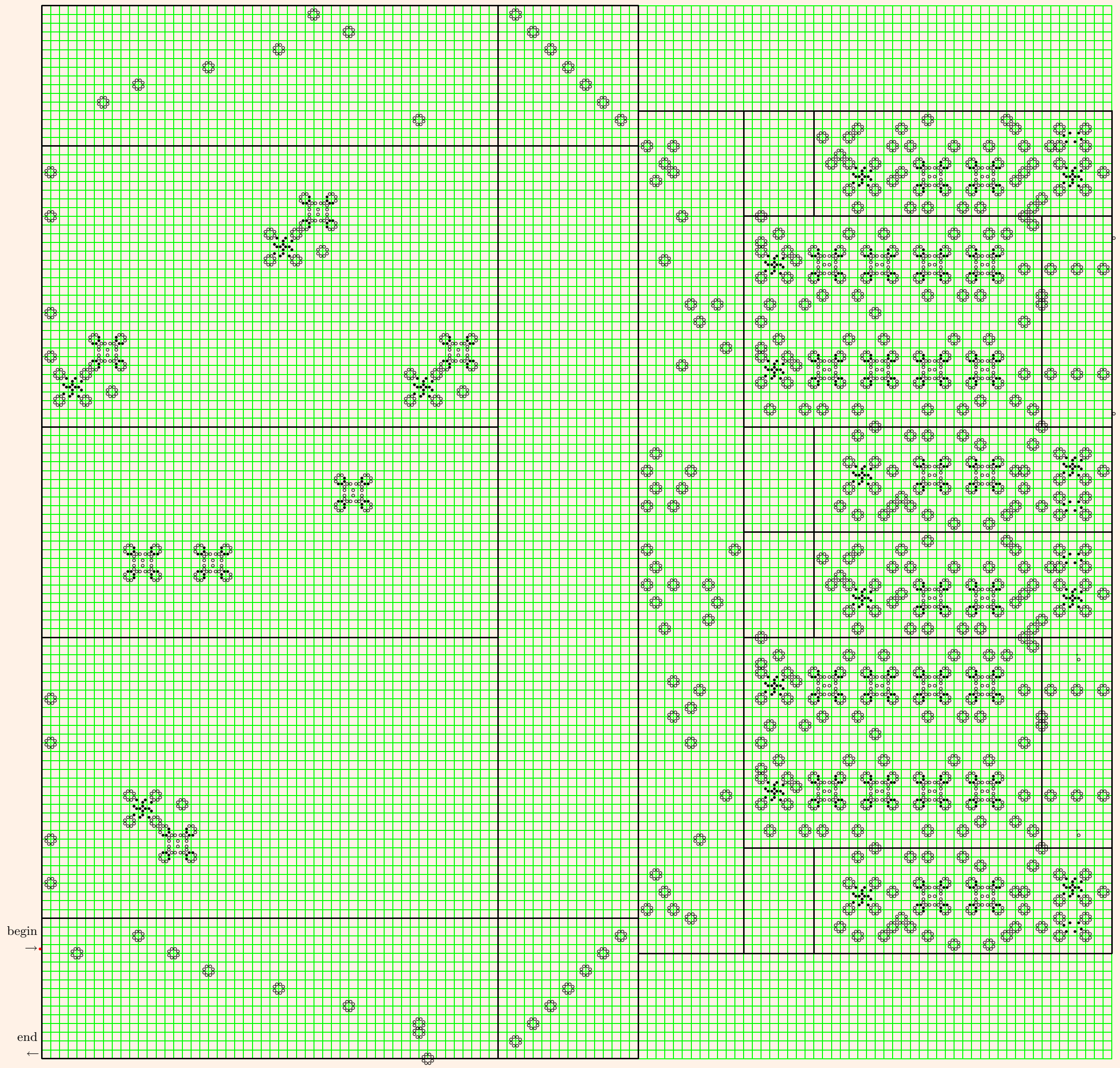
A Position Marker in P_3



Pushing and Pulling a Position Marker in P_3

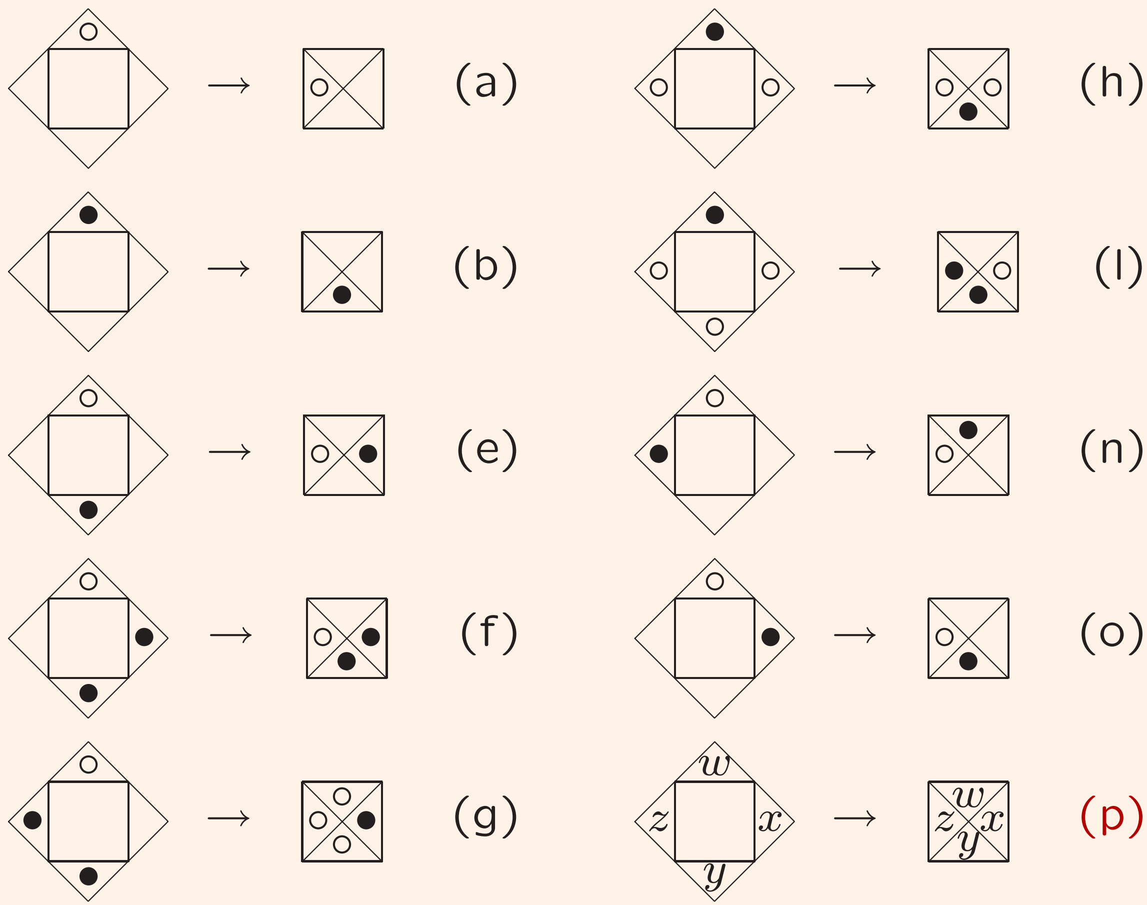


Any Reversible Counter Machine Is Embeddable in P_3



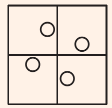
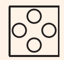
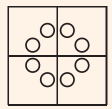
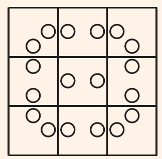
3.2 (2) 3^4-State Model P'_3

$$P'_3 = (\mathrm{Z}^2, \{0,1,2\}^4, g'_3, (0,0,0,0))$$

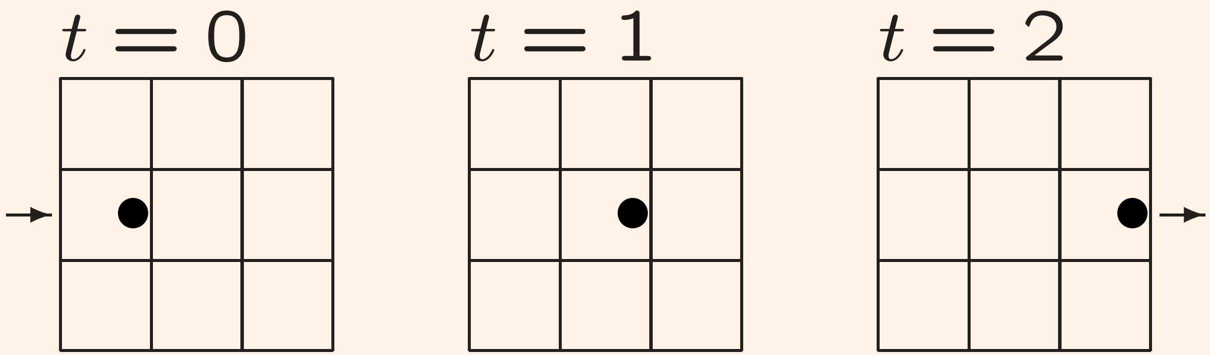


The rule scheme (p) 45 rules not specified by the schemes (a)–(o).
($w, x, y, z \in$ { blank, ○, ● }).

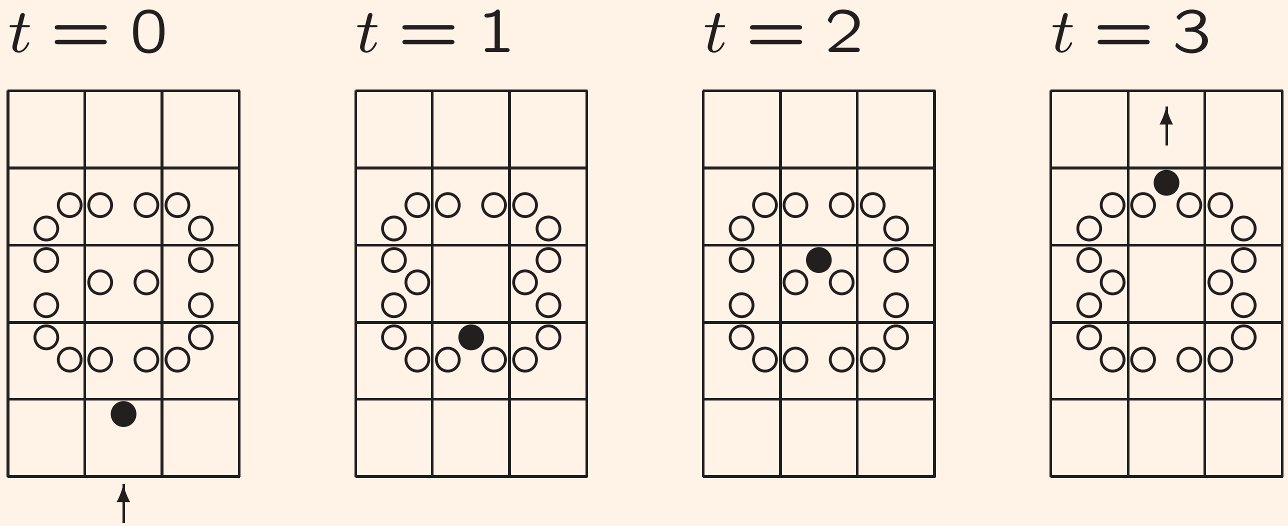
Four Signal Processing Elements in P'_3

Element Name	Pattern
LR-turn element	
R-turn element	
Reflector	
Rotary element	

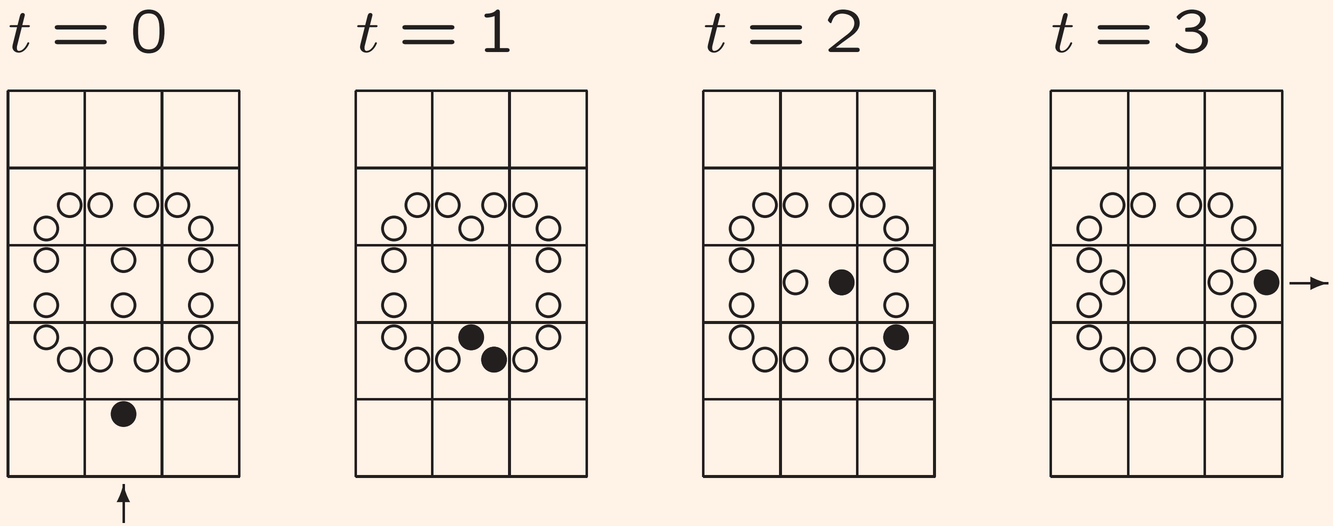
A Signal in P'_3



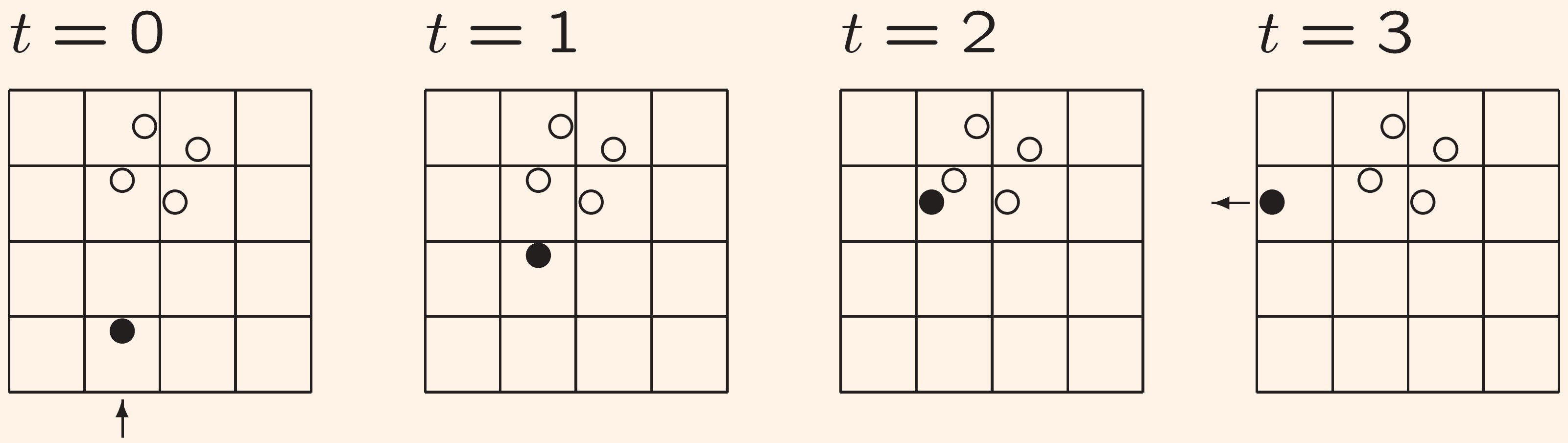
A Rotary Element in P_3' (Parallel Case)



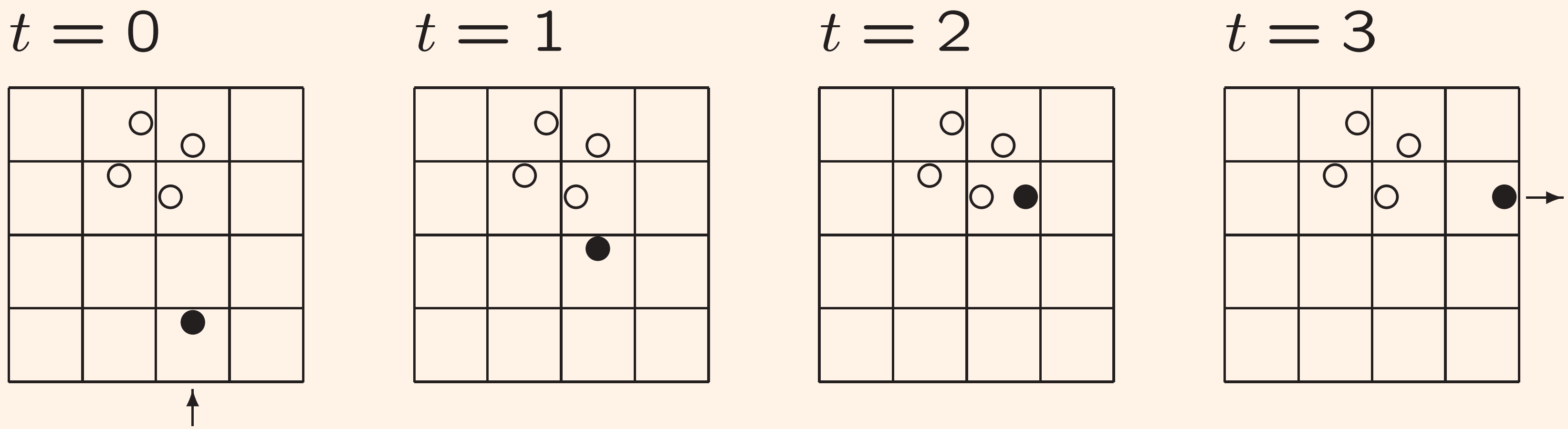
A Rotary Element in P'_3 (Orthogonal Case)



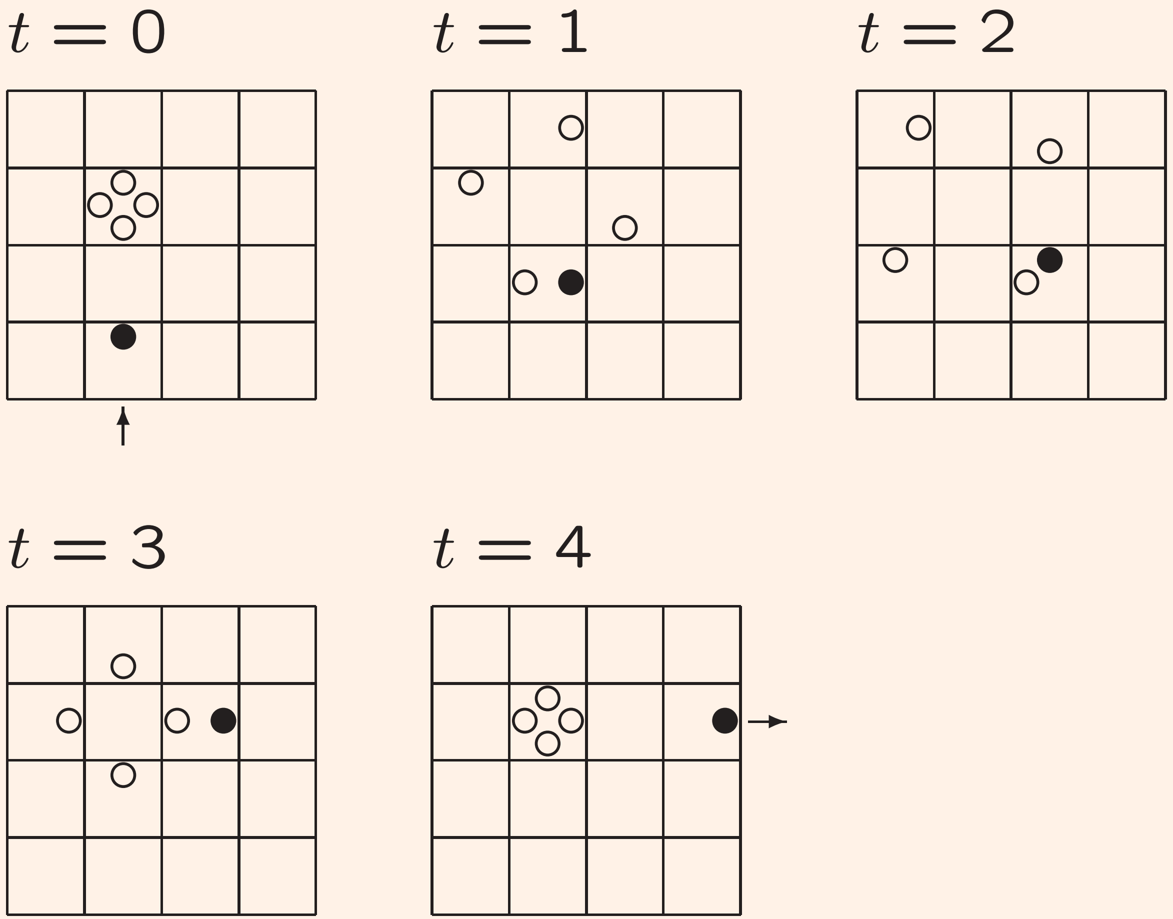
An LR-Turn Element in P_3' (Left Turn)



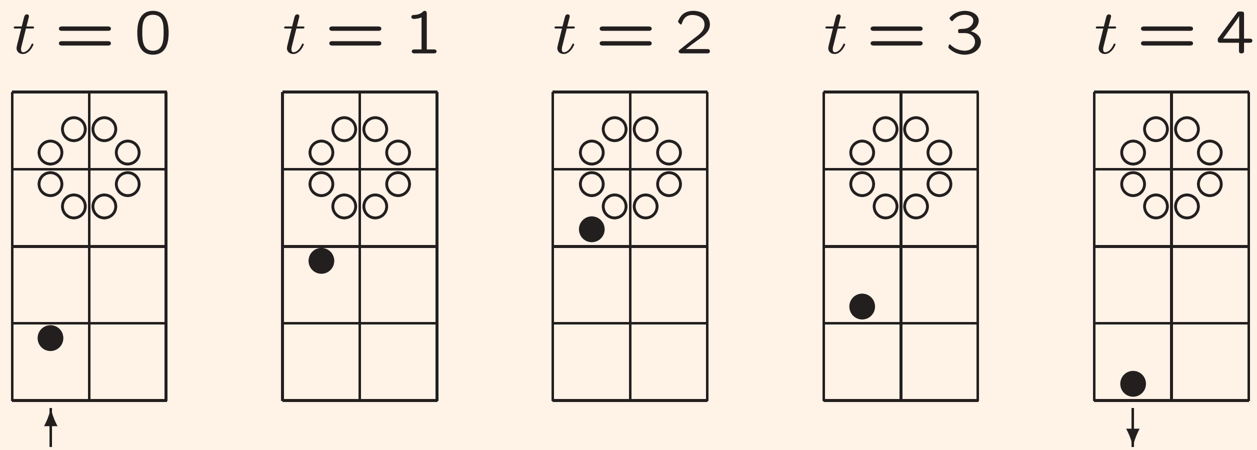
An LR-Turn Element in P_3' (Right Turn)



An R-Turn Element in P_3'

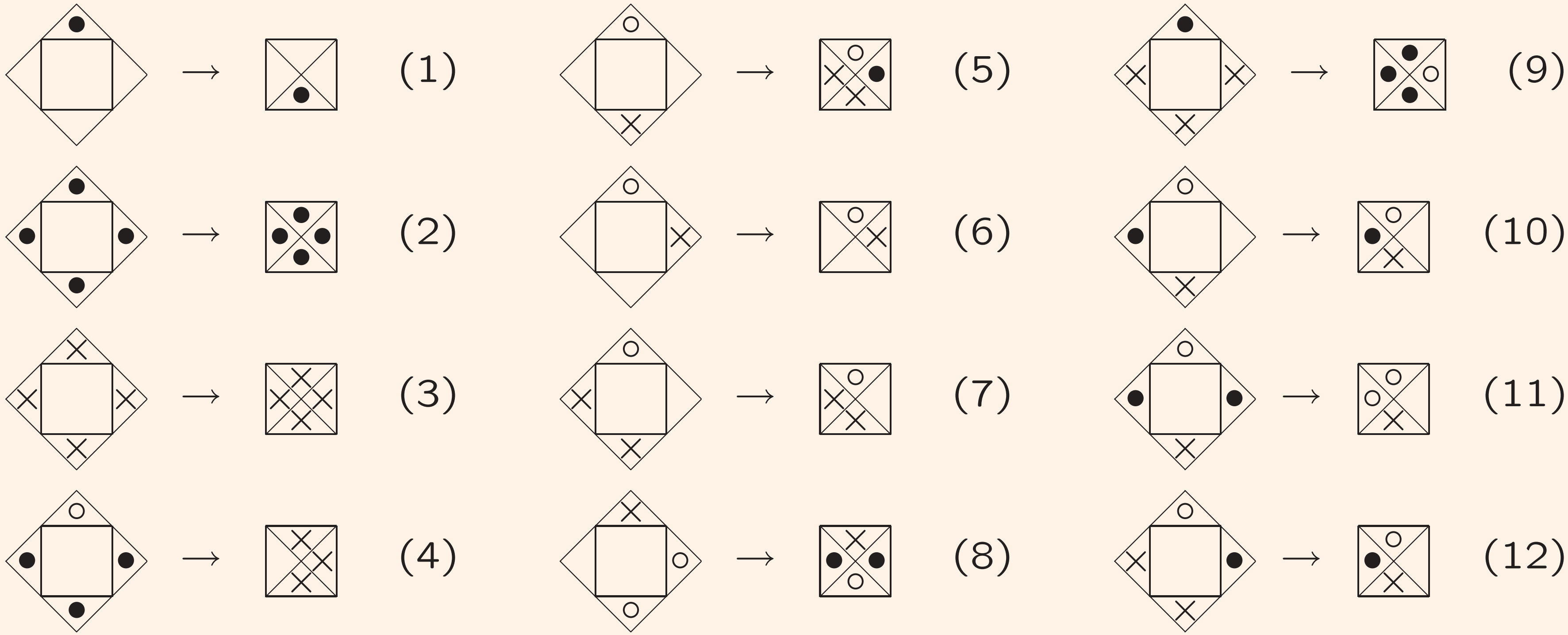


A Reflector in P_3'



3.3. Embedding a Left-/Right-Rotate Element

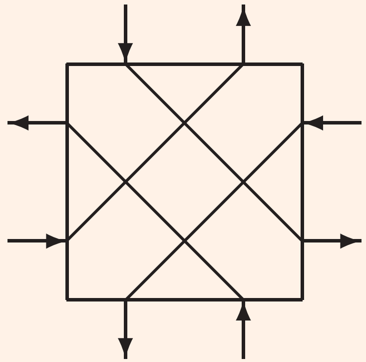
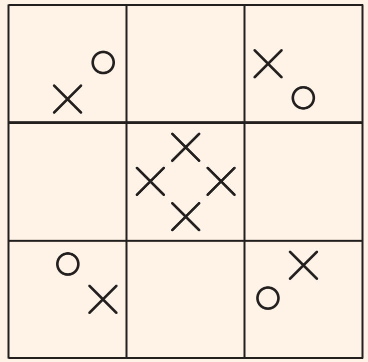
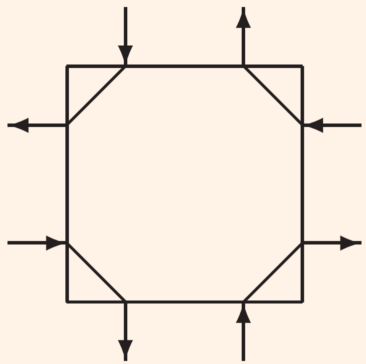
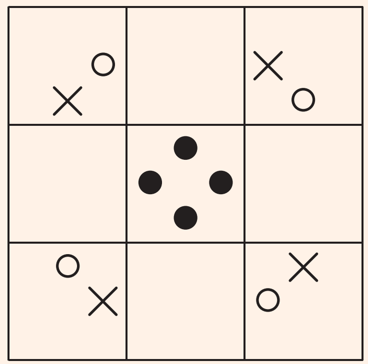
$$P_{\mathsf{LRRE}} = (\mathsf{Z}^2, \{0, 1, 2, 3\}^4, g_{\mathsf{LRRE}}, (0, 0, 0, 0))$$



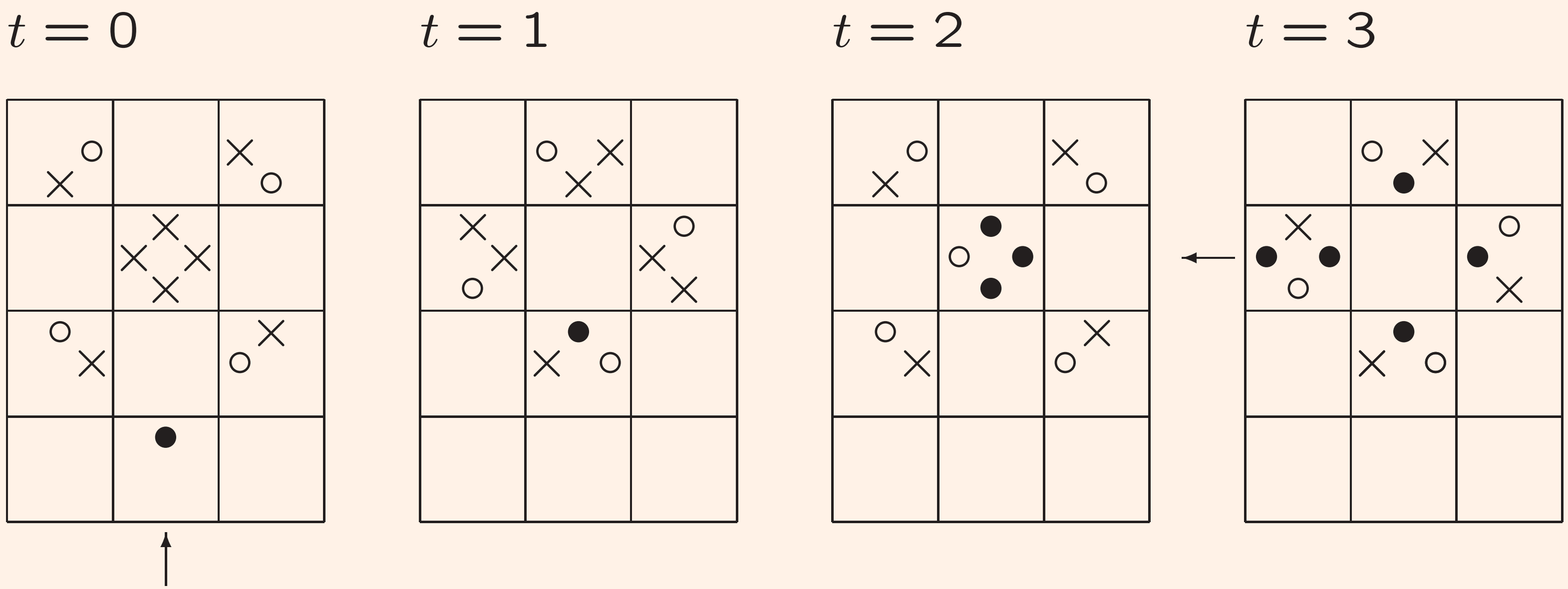
Only the rules to simulate an LRRE are shown.

(0: blank, 1: ∘, 2: •, 3: ×.)

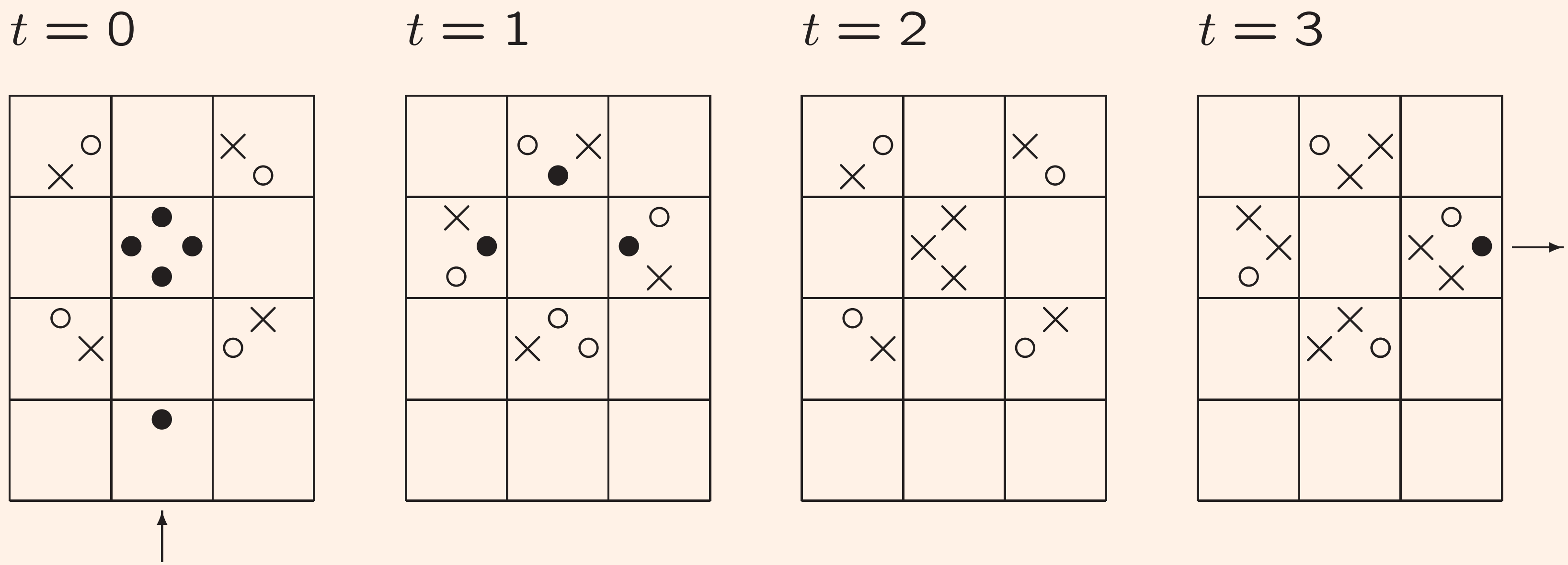
Two States of an LRRE in P_{LRRE}

L-state		
R-state		

An LRRE in P_{LRRE} (L-State)



An LRRE in P_{LRRE} (R-State)



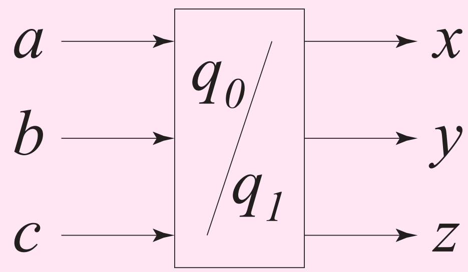
3.4. Further Problems

- How can 2-state 2- or 3-symbol RLEMs be embedded in RCAs?
- So far, no satisfactory result has been obtained.
- Below, we show some results on 2- or 3-symbol RLEMs themselves.

2-State 3-Symbol RLEMs

The form of the move function:

Present state	Input		
	a	b	c
State q_0	?	?	?
State q_1	?	?	?



There are $6! = 720$ kinds of move functions.

Numbers 0–719 are given in a systematic way.

Equivalent RLEMs

No.10:

Present state	Input		
	a	b	c
State q_0	$q_0\ x$	$q_0\ y$	$q_1\ x$
State q_1	$q_1\ z$	$q_0\ z$	$q_1\ y$

No.363:

Present state	Input		
	a	b	c
State q_0	$q_1\ x$	$q_0\ x$	$q_0\ y$
State q_1	$q_1\ y$	$q_1\ z$	$q_0\ z$

No. 363 is obtaind from No. 10 by the permutation

$$(a, b, c) \to (b, c, a).$$

There are 24 equivalence classes.

24 Representative 2-state 3-symbol RLEMs

No.0	a	b	c
q0	q0x	q0y	q0z
q1	q1x	q1y	q1z

No.1	a	b	c
q0	q0x	q0y	q0z
q1	q1x	q1z	q1y

No.3	a	b	c
q0	q0x	q0y	q0z
q1	q1y	q1z	q1x

No.6	a	b	c
q0	q0x	q0y	q1x
q1	q0z	q1y	q1z

No.7	a	b	c
q0	q0x	q0y	q1x
q1	q0z	q1z	q1y

No.9	a	b	c
q0	q0x	q0y	q1x
q1	q1y	q1z	q0z

No.10	a	b	c
q0	q0x	q0y	q1x
q1	q1z	q0z	q1y

No.11	a	b	c
q0	q0x	q0y	q1x
q1	q1z	q1y	q0z

No.18	a	b	c
q0	q0x	q0y	q1z
q1	q0z	q1x	q1y

No.19	a	b	c
q0	q0x	q0y	q1z
q1	q0z	q1y	q1x

No.21	a	b	c
q0	q0x	q0y	q1z
q1	q1x	q1y	q0z

No.23	a	b	c
q0	q0x	q0y	q1z
q1	q1y	q1x	q0z

No.60	a	b	c
q0	q0x	q1x	q1y
q1	q0y	q0z	q1z

No.61	a	b	c
q0	q0x	q1x	q1y
q1	q0y	q1z	q0z

No.63	a	b	c
q0	q0x	q1x	q1y
q1	q0z	q1z	q0y

No.64	a	b	c
q0	q0x	q1x	q1y
q1	q1z	q0y	q0z

No.65	a	b	c
q0	q0x	q1x	q1y
q1	q1z	q0z	q0y

No.90	a	b	c
q0	q0x	q1y	q1z
q1	q0y	q0z	q1x

No.91	a	b	c
q0	q0x	q1y	q1z
q1	q0y	q1x	q0z

No.94	a	b	c
q0	q0x	q1y	q1z
q1	q1x	q0y	q0z

No.95	a	b	c
q0	q0x	q1y	q1z
q1	q1x	q0z	q0y

No.450	a	b	c
q0	q1x	q1y	q1z
q1	q0x	q0y	q0z

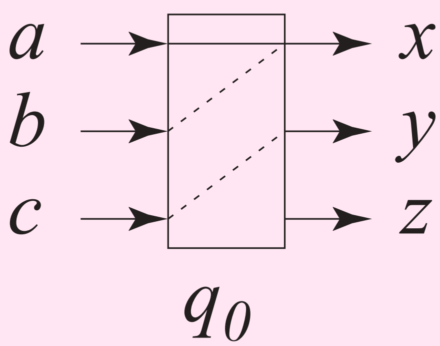
No.451	a	b	c
q0	q1x	q1y	q1z
q1	q0x	q0z	q0y

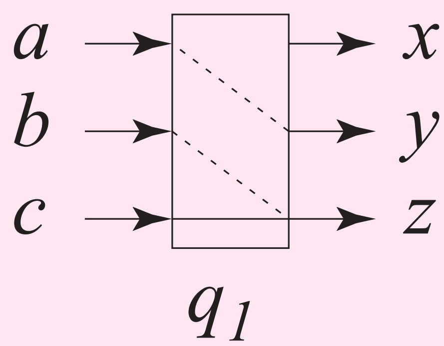
No.453	a	b	c
q0	q1x	q1y	q1z
q1	q0y	q0z	q0x

Symbolic Representation: An Example

No.363 (equivalent to No.10):

Present state	Input		
	a	b	c
State q_0	q_1 x	q_0 x	q_0 y
State q_1	q_1 y	q_1 z	q_0 z

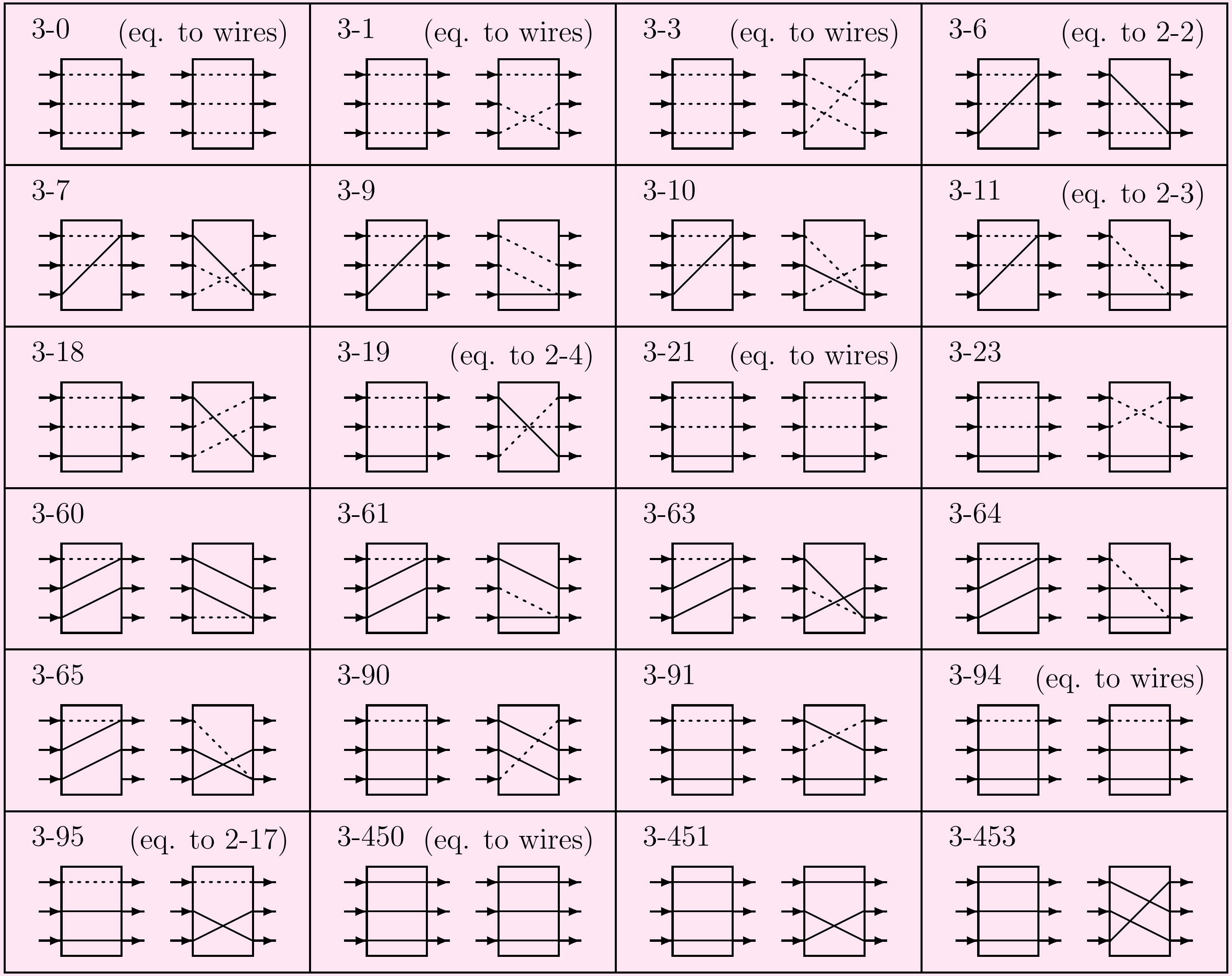




Solid line: makes the state change.

Dotted line: leaves the state unchanged.

Symbolic Representation of 24 RLEMs

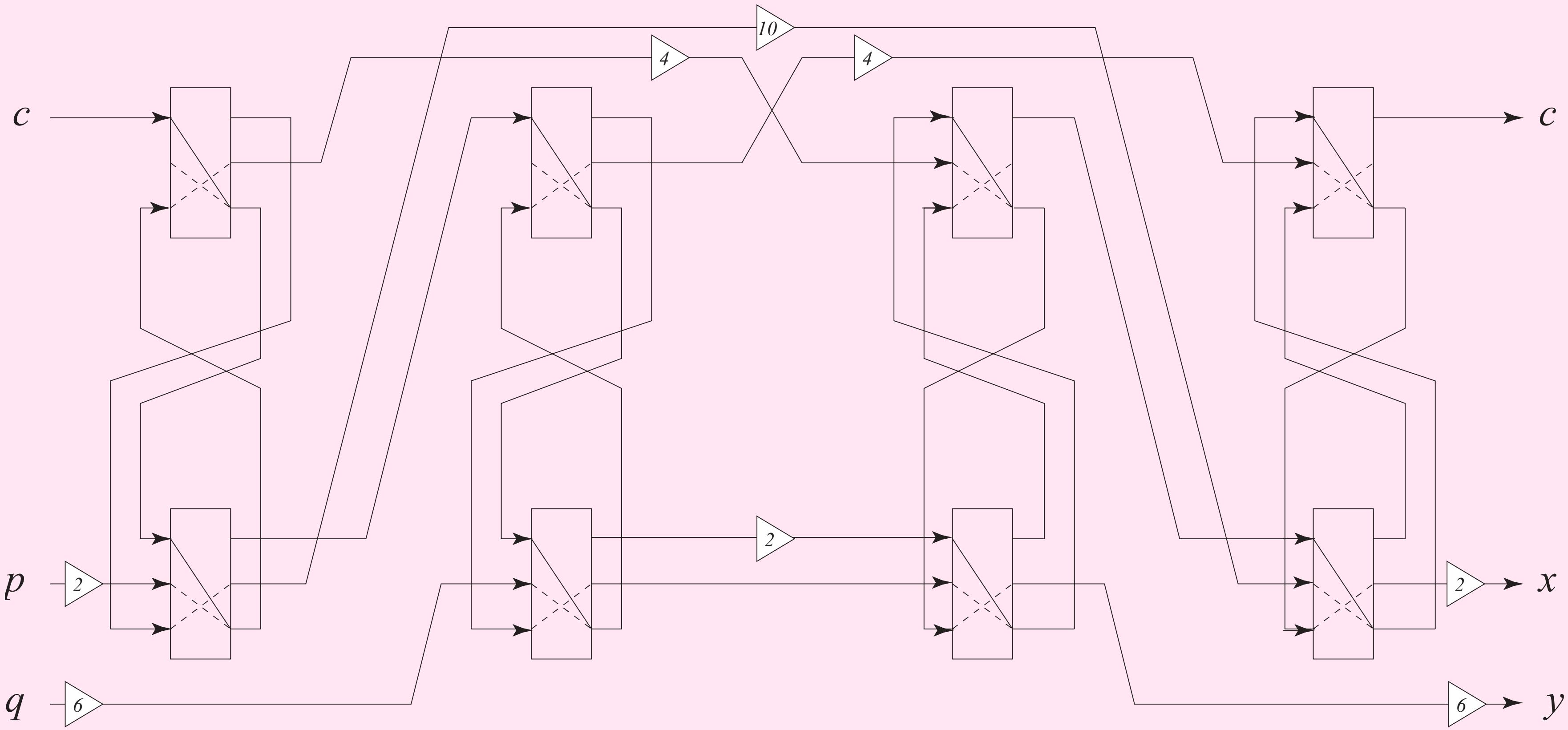


Classification of 24 RLEMs

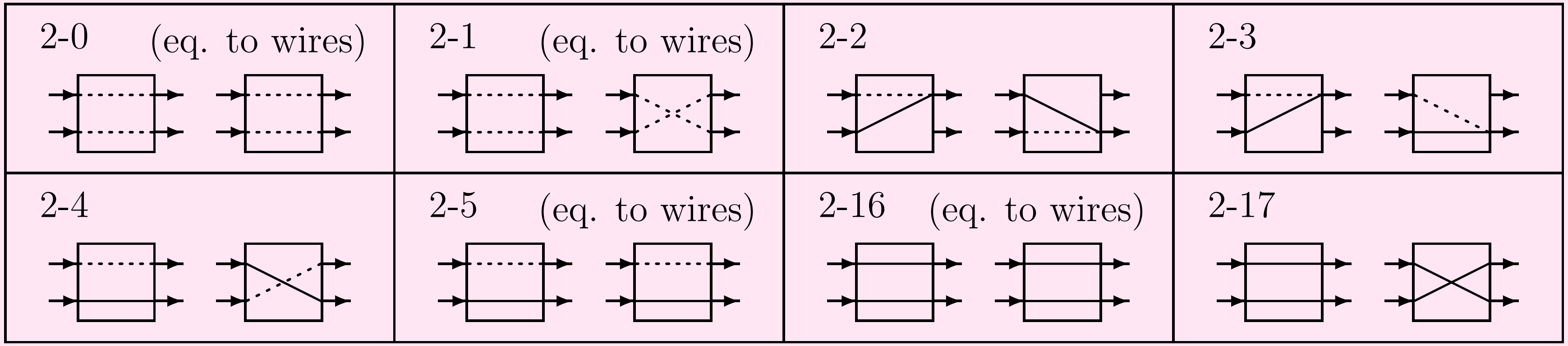
1. Equivalent to Wires (6):
   Nos. 0, 1, 3, 21, 94, 450
   Non-universal.
2. Equivalent to 2-State 2-Symbol Elements (4):
   Nos. 6, 11, 19, 95
   Conjectured to be non-universal.
3. Nondegenerate 2-State 3-Symbol Elements (14):
   Nos. 7, 9, 10, 18, 23, 60, 61, 63, 64, 65, 90, 91, 451, 453
   All are Universal! (An F-gate can be constructed.)
   [Ogiro, Kanno, Tanaka, Kato, and Morita, submitted]

Constructing an F-gate out of No.7 RLEM

[Ogiro, Kanno, Tanaka, Kato, and Morita, submitted]



2-State 2-Symbol RLEMs



- $4! = 24$ kinds of move functions.
- 8 equivalence classes.
- 4 equivalence classes are nondegenerate ones.
- Conjecture: No single element is universal.
- { No.2-3, No.2-4 } is logically universal.

[Lee, Peper, Adachi, and Mashiko, submitted]

4. Concluding Remarks

- RLEMs are often very useful in reversible computing.
- Rotary element (RE) can be embedded very simply in a reversible cellular automata.
- Future problem:
  - Are there other RLEMs that are useful, and can be embedded more simply than an RE?