Complexity Analysis in Cyclic Tag System Emulated by Rule 110

AUTOMATA2013

Sep. 17 – 19, 2013

Giessen, Germany

Shigeru Ninagawa [1,3] and Genaro J. Martínez [2,3]

[1] Kanazawa Institute of Technology, Japan

[2] Instituto Politécnico Nacional, Mexico

[3] University of the West of England, United Kingdom

Table of contents

1. Background
2. Lempel-Ziv complexity
3. Cyclic tag system emulated by rule 110
4. Results
5. Conclusion

Basic idea

- CAs have no memory except for cell → all the information necessary to perform computation is in its configuration
- Complexity of conf. is reflective of the complexity of information necessary for computation

Related works

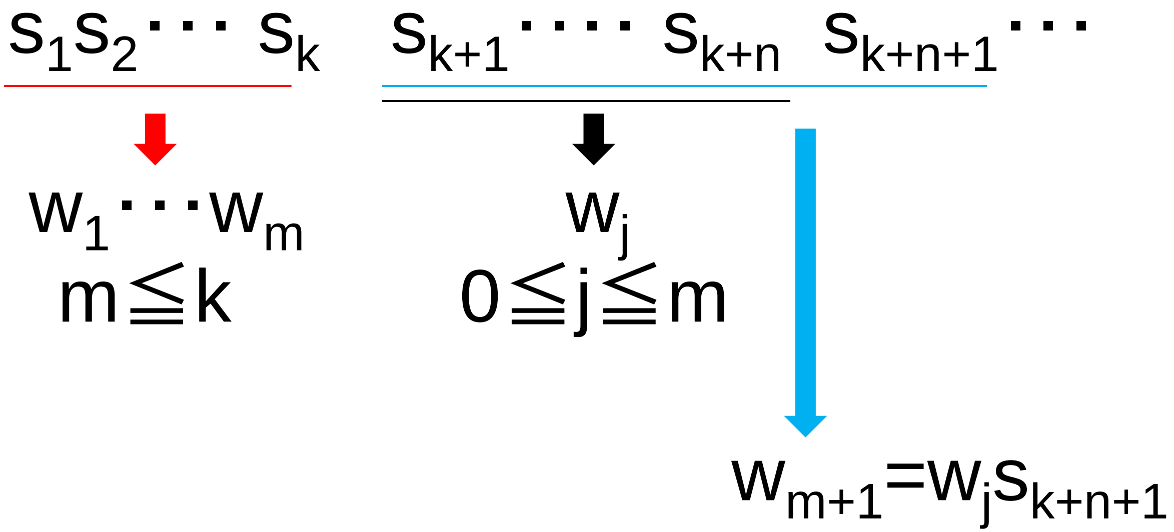
- Compression-based classification of ECA by DEFLATE (LZ77 + Huffman coding) (Zenil, 2010)
- Parity problem solving process by ECA rule 60 in array size of 2^n (Ninagawa, 2012) ← stepwise decrease by period halving (Lempel-Ziv complexity, LZ78)

Motivation

- ECA rule 110 is supporting universal computation by emulating cyclic tag system
- How does complexity vary during the emulation process by rule 110?
- We employ Lempel-Ziv complexity as a measure of complexity

Lempel-Ziv complexity (Ziv, Lempel, 1978)(1/2)

$s_1 s_2 \cdots s_k s_{k+1} \cdots$: given string, $s_i \in$ alphabet
$s_1 \cdots s_k$ has already been divided into phrases $w_1 \cdots w_m (= s_1 s_2 \cdots s_k)$, $m \leqq k$, $w_0 = \varepsilon$ (empty string)
search the longest substring $s_{k+1} \cdots s_{k+n} = w_j$ $(0 \leqq j \leqq m)$
and set $w_{m+1} = w_j s_{k+n+1}$



$s_1 s_2 \cdots s_k$ $s_{k+1} \cdots s_{k+n}$ $s_{k+n+1} \cdots$

$w_1 \cdots w_m$
$m \leqq k$

w_j
$0 \leqq j \leqq m$

$w_{m+1} = w_j s_{k+n+1}$

Lempel-Ziv complexity (2/2)

For example: $010010101\cdots$ is given

$w_0 = \varepsilon$, $w_1 = 0 = w_0 0$,
$w_2 = 1 = w_0 1$, $w_3 = 00 = w_1 0$,
$w_4 = 10 = w_2 0$, $w_5 = 101 = w_4 1$, $\cdots$

$010010101\cdots$ ⇨ $0 \mid 1 \mid 00 \mid 10 \mid 101 \mid \cdots$
$w_1 \quad w_2 \quad w_3 \quad w_4 \quad w_5$

The number of divided phrases: Lempel-Ziv complexity

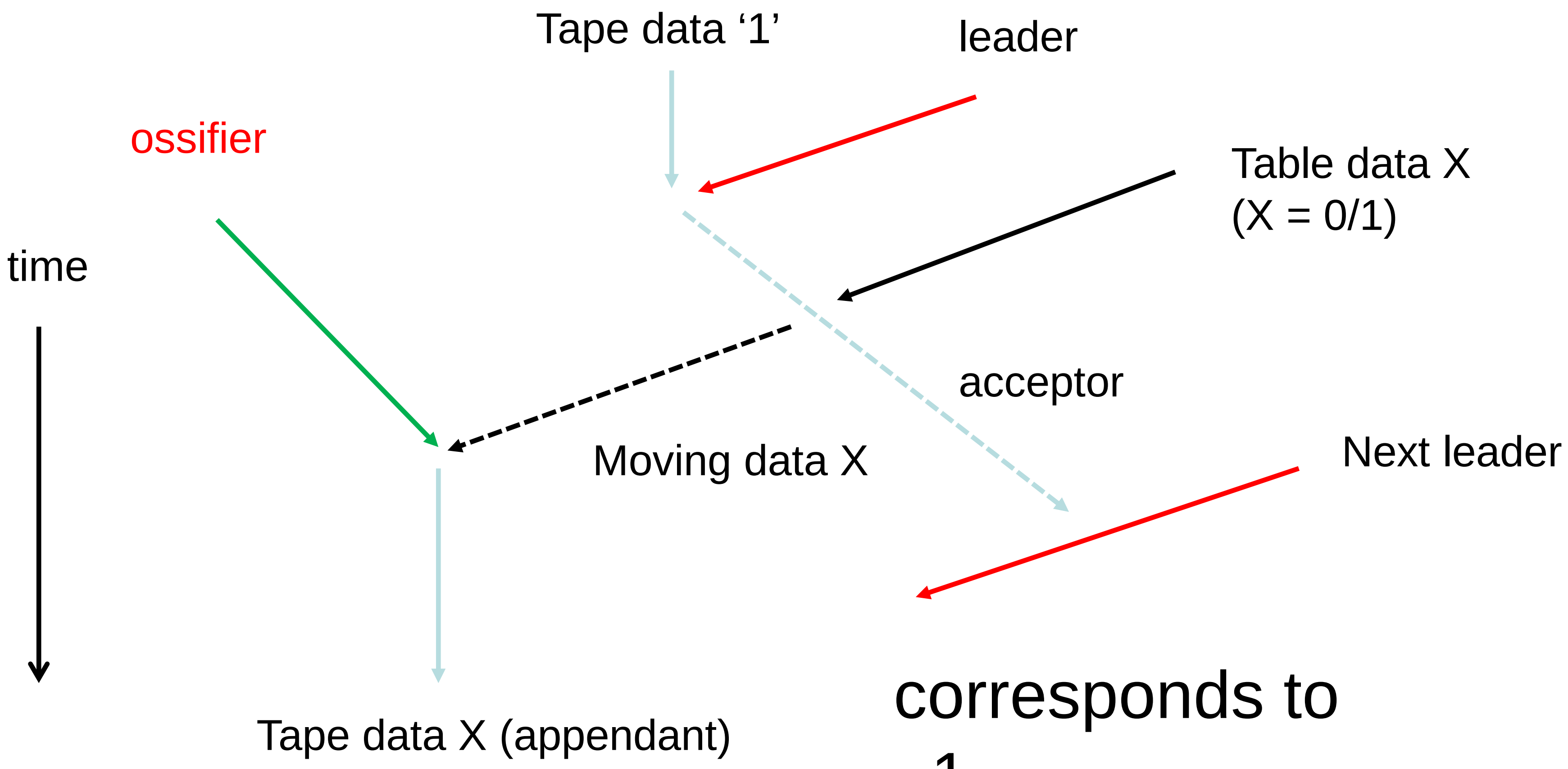
Cyclic tag system(Cook, 2004)

- Σ={0,1}, tape is read from the front and appended according to appendant table
- Example appendant table (1, 101)

1	appendant
11	1
1101	101
1011	1
011	101 skipped in reading ‘0’
111	1

- CTS can emulate tag system and rule 110 can emulate CTS

Basic mechanism (tape data ‘1’)



Tape data ‘1’

leader

ossifier

Table data X
(X = 0/1)

time

acceptor

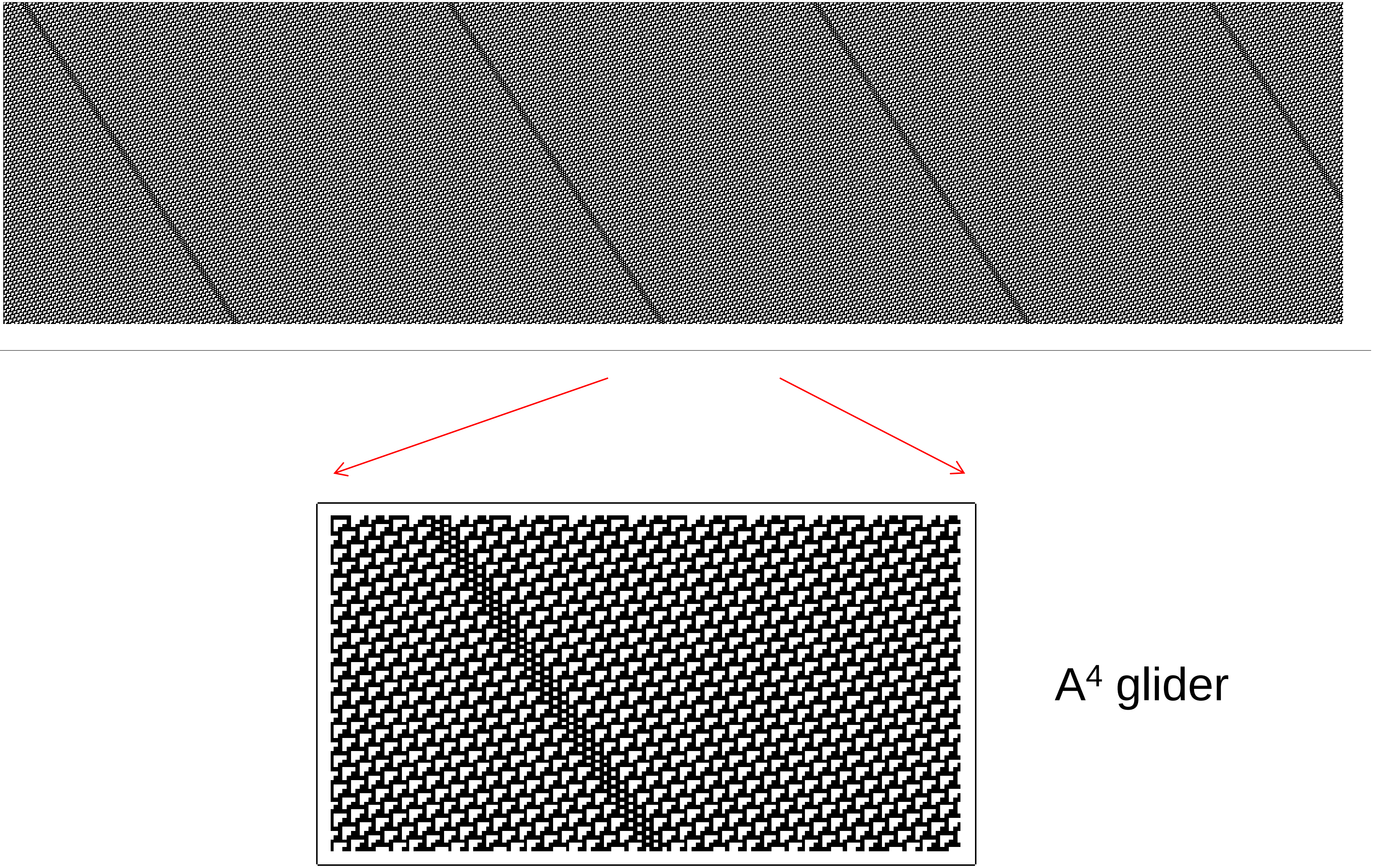
Moving data X

Next leader

Tape data X (appendant)

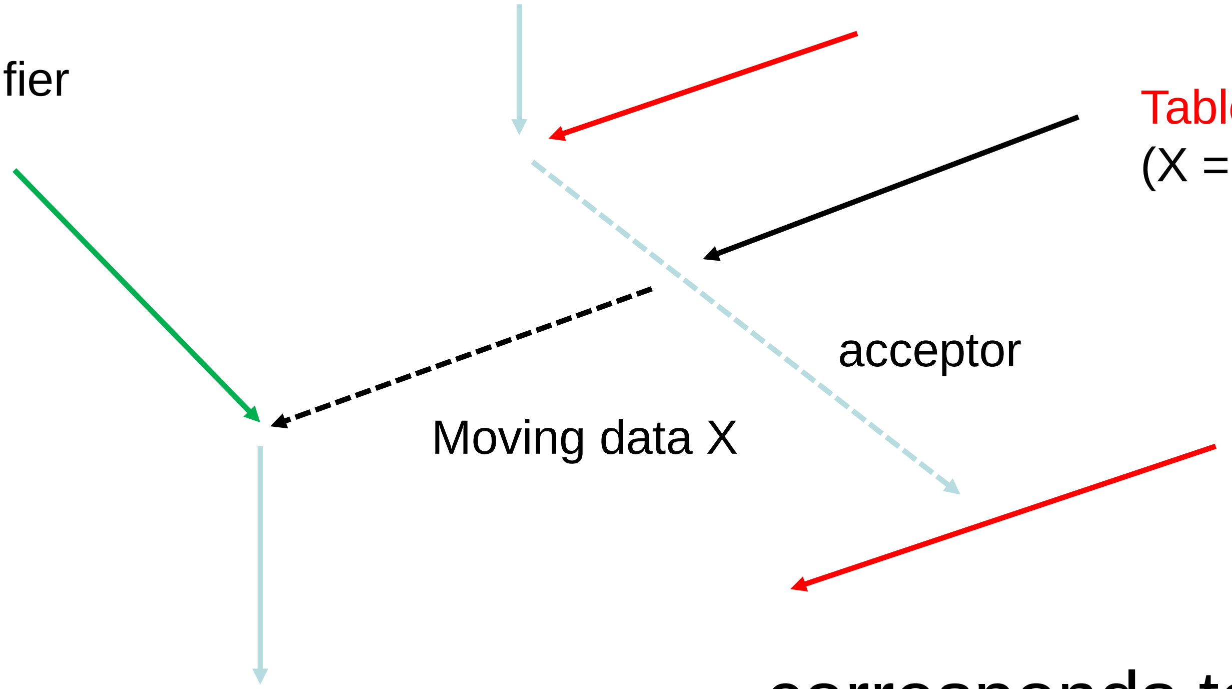
corresponds to
1···
1···X

Space-time pattern of ossifier



A^4 glider

Basic mechanism (tape data ‘1’)



Tape data ‘1’

leader

ossifier

Table data X
(X = 0/1)

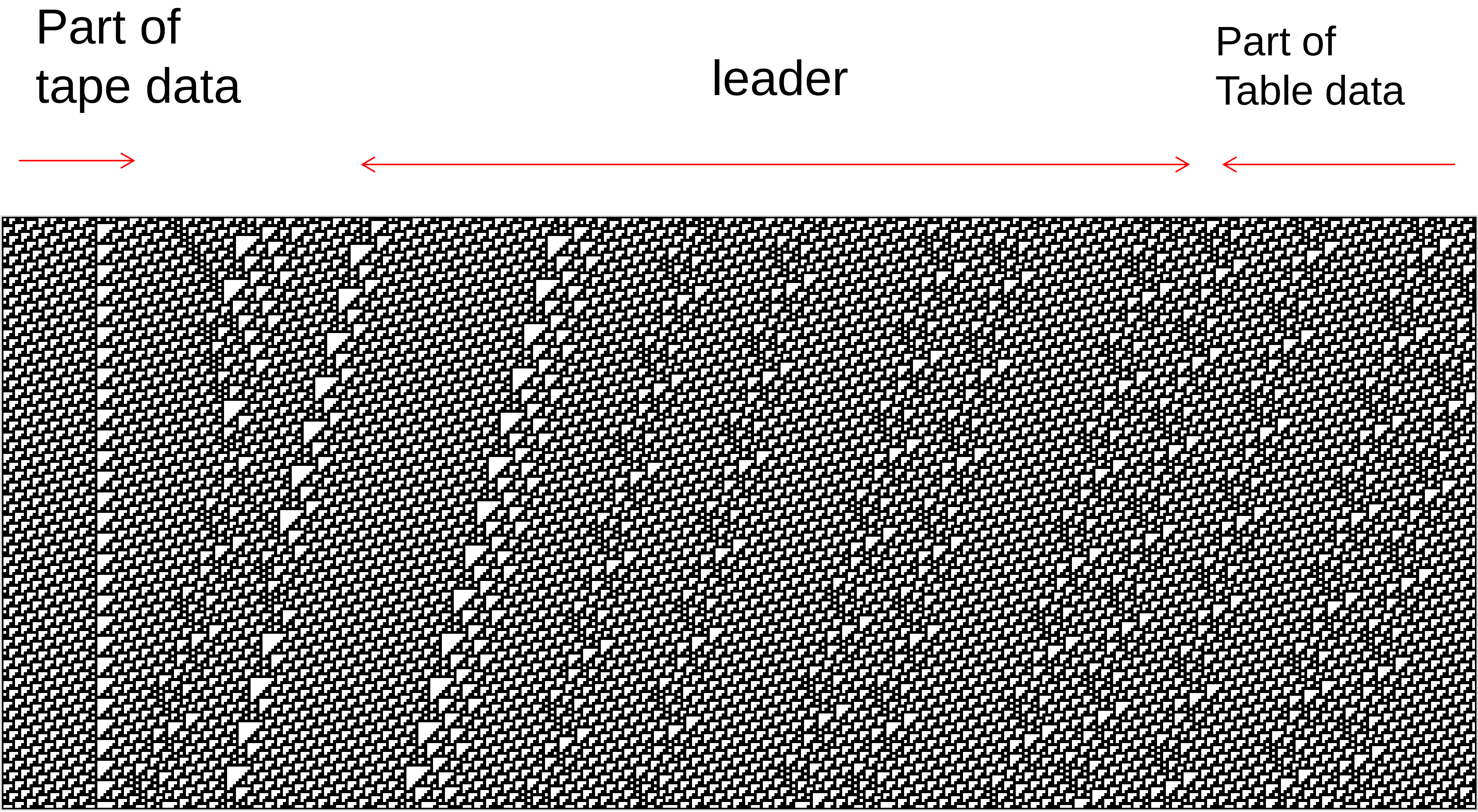
acceptor

Moving data X

Next leader

Tape data X (appendant)

corresponds to
1···
1···X

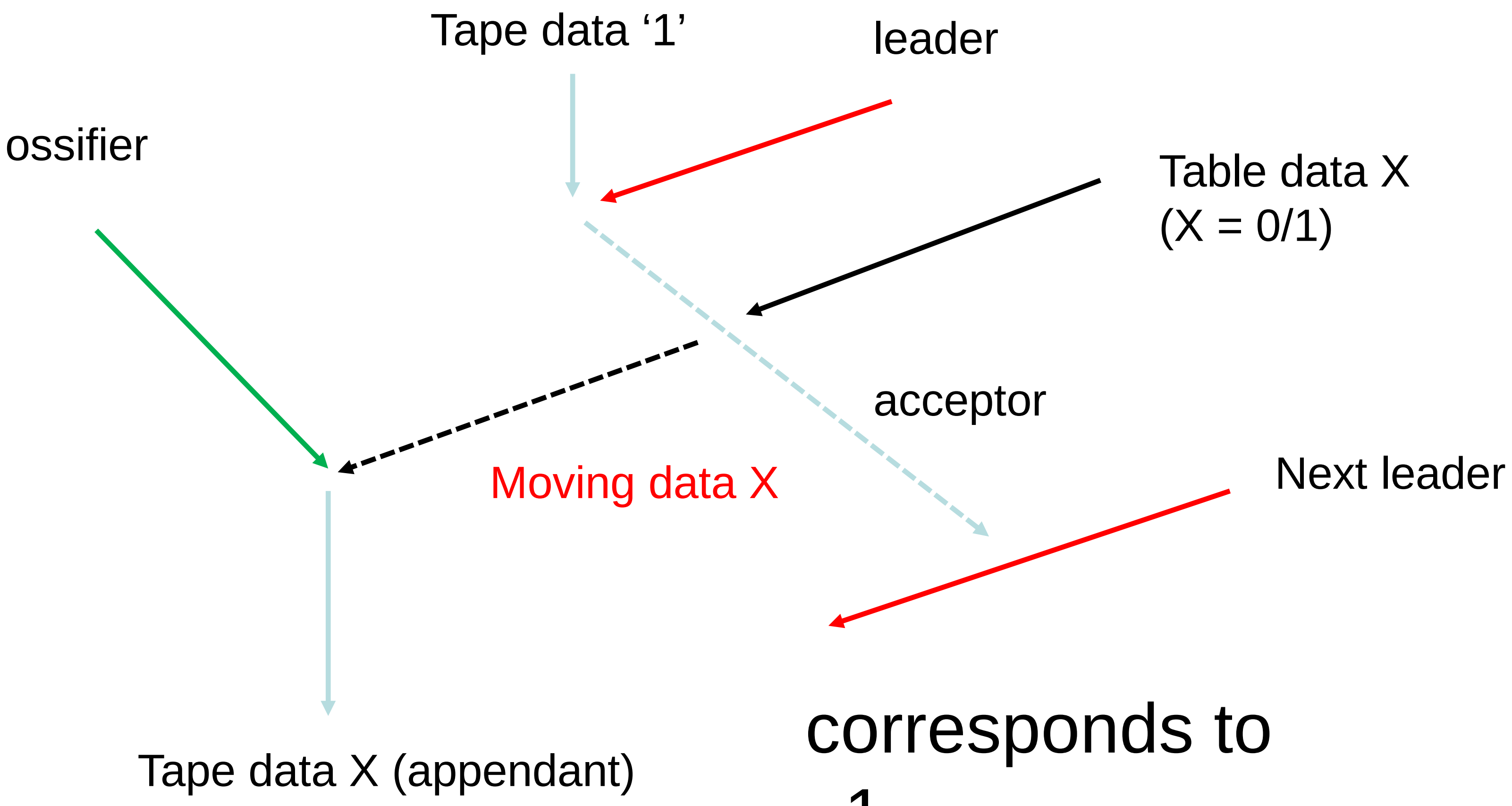


Part of tape data

leader

Part of Table data

Basic mechanism (tape data ‘1’)



Tape data ‘1’

leader

ossifier

Table data X
(X = 0/1)

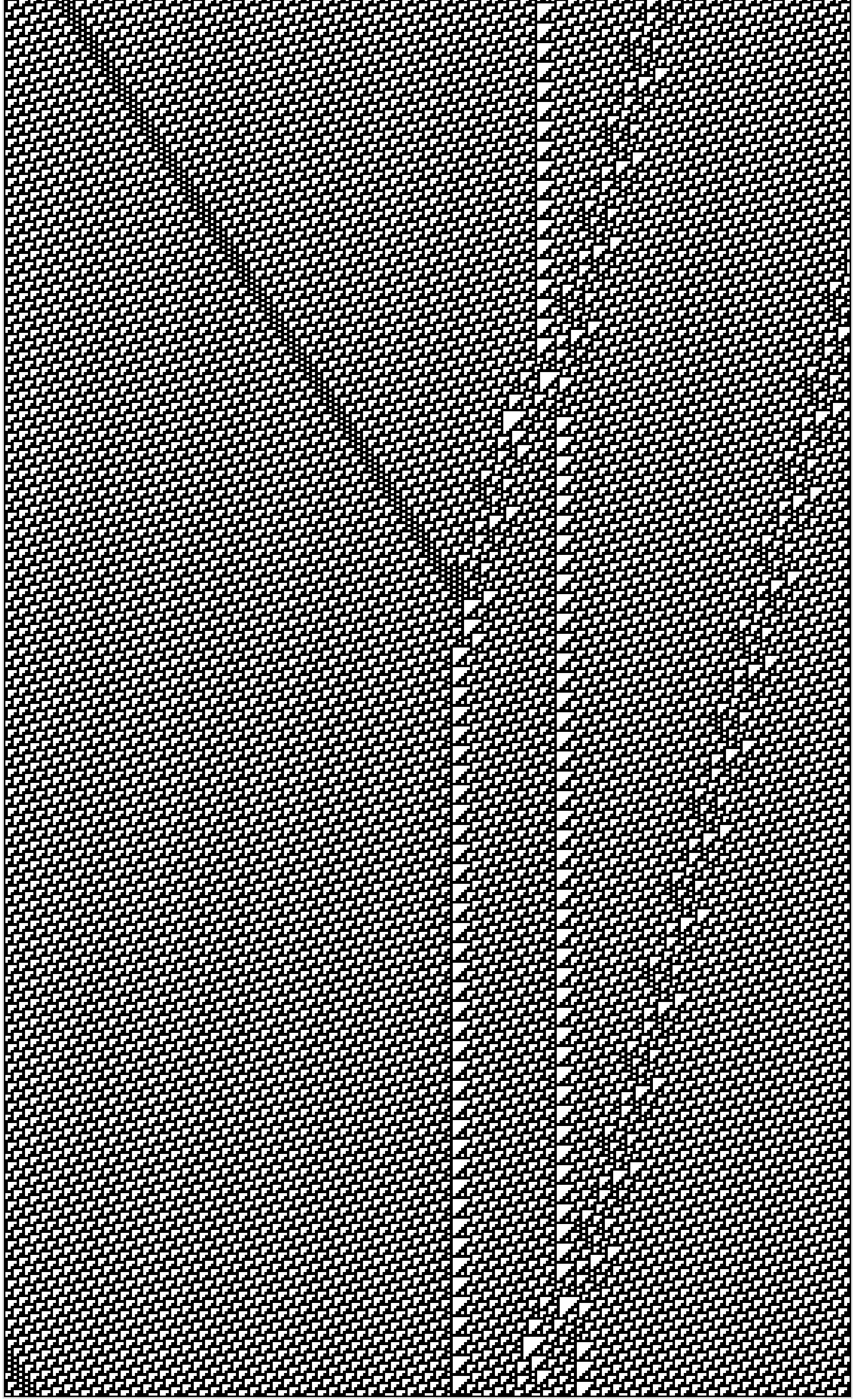
acceptor

Moving data X

Next leader

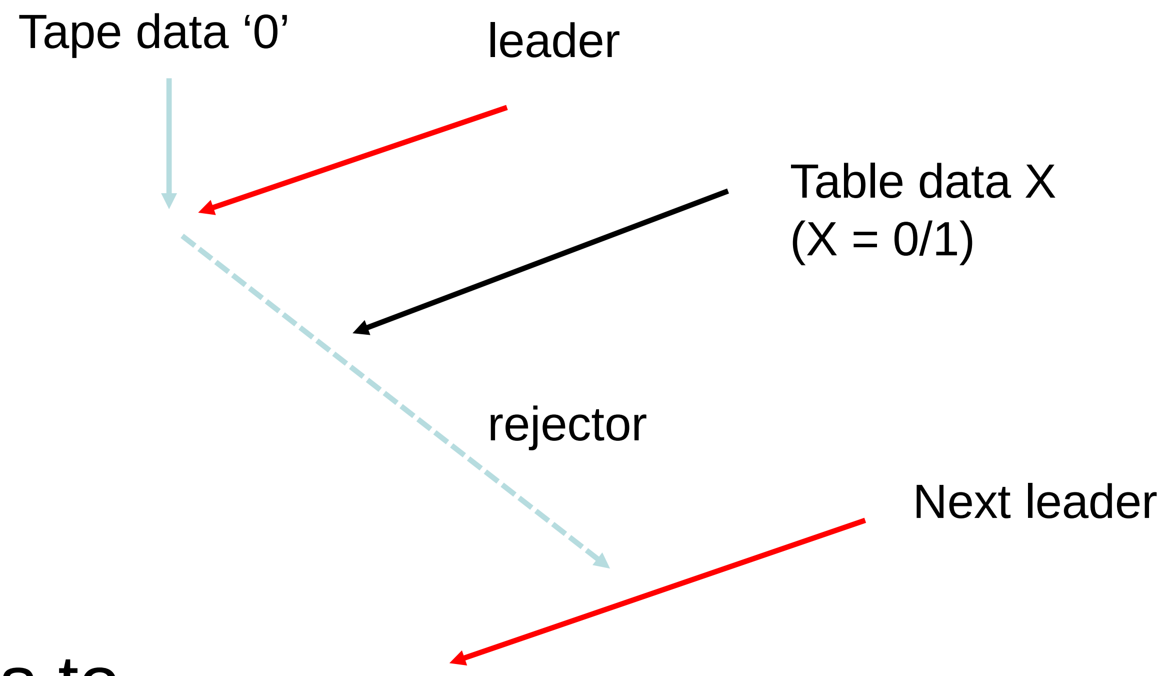
Tape data X (appendant)

corresponds to
1･･･
1･･･X



Collision between
ossifier and moving
data creates tape data

Basic mechanism (tape data ‘0’)



Tape data ‘0’

leader

Table data X
(X = 0/1)

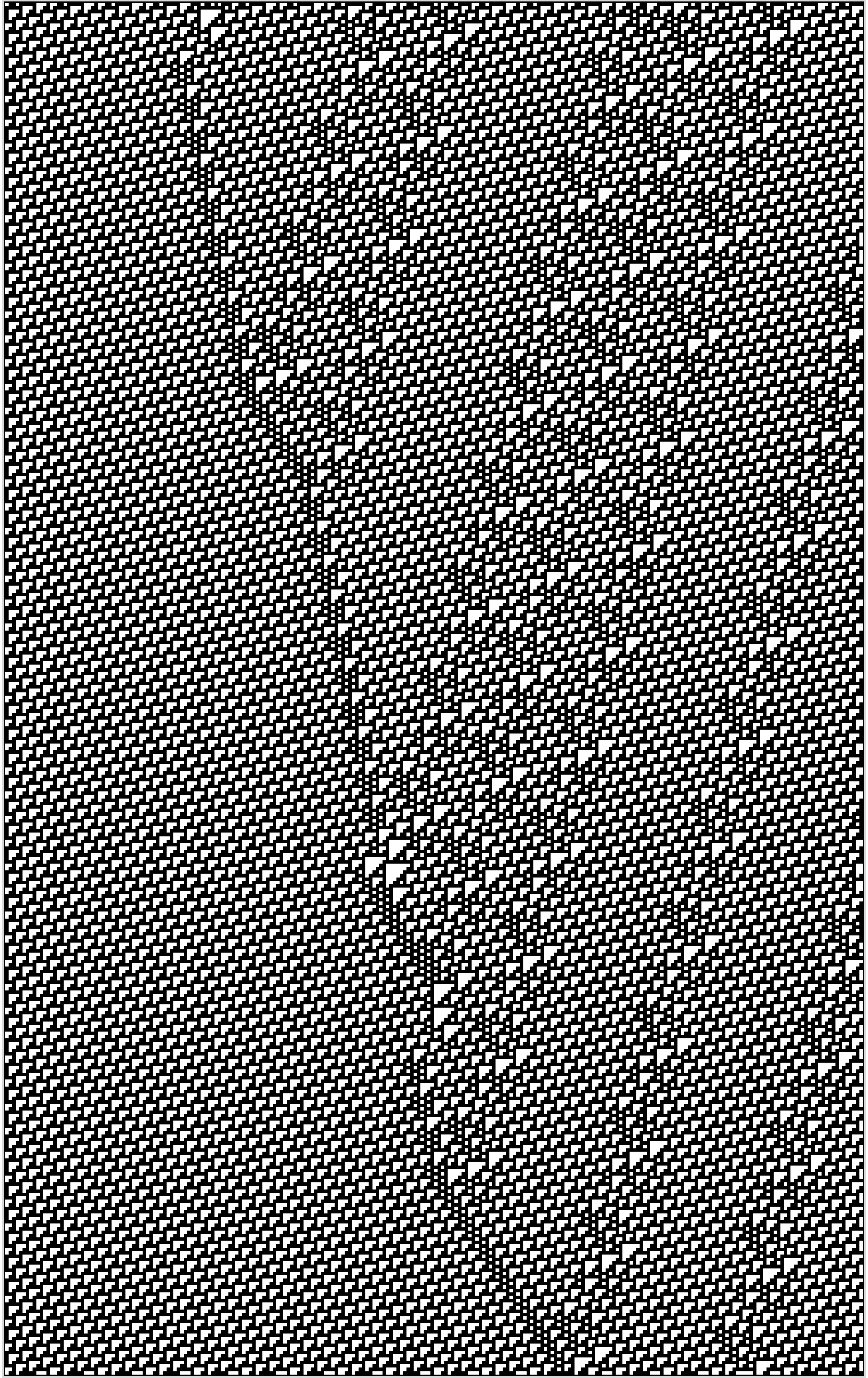
rejector

Next leader

Corresponds to

0･･･

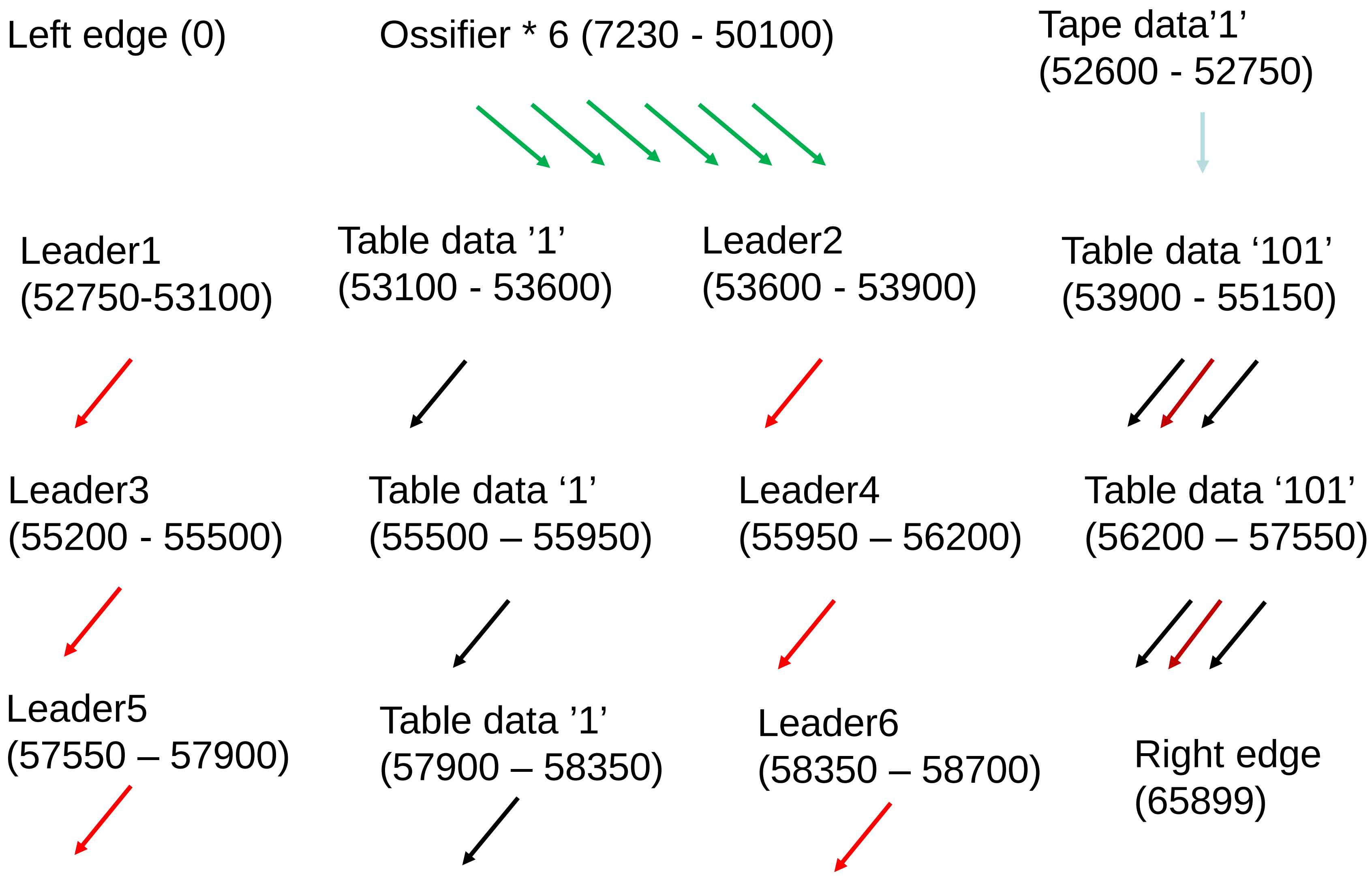
0･･･ appendant X skipped



Rejector is erasing table data

IC emulating CTS (N=65,900)

http:ucomp.uwe.ac.uk/genaro/rule110/ctsRule110.html



Left edge (0)

Ossifier * 6 (7230 - 50100)

Tape data’1’
(52600 - 52750)

Leader1
(52750-53100)

Table data ’1’
(53100 - 53600)

Leader2
(53600 - 53900)

Table data ‘101’
(53900 - 55150)

Leader3
(55200 - 55500)

Table data ‘1’
(55500 – 55950)

Leader4
(55950 – 56200)

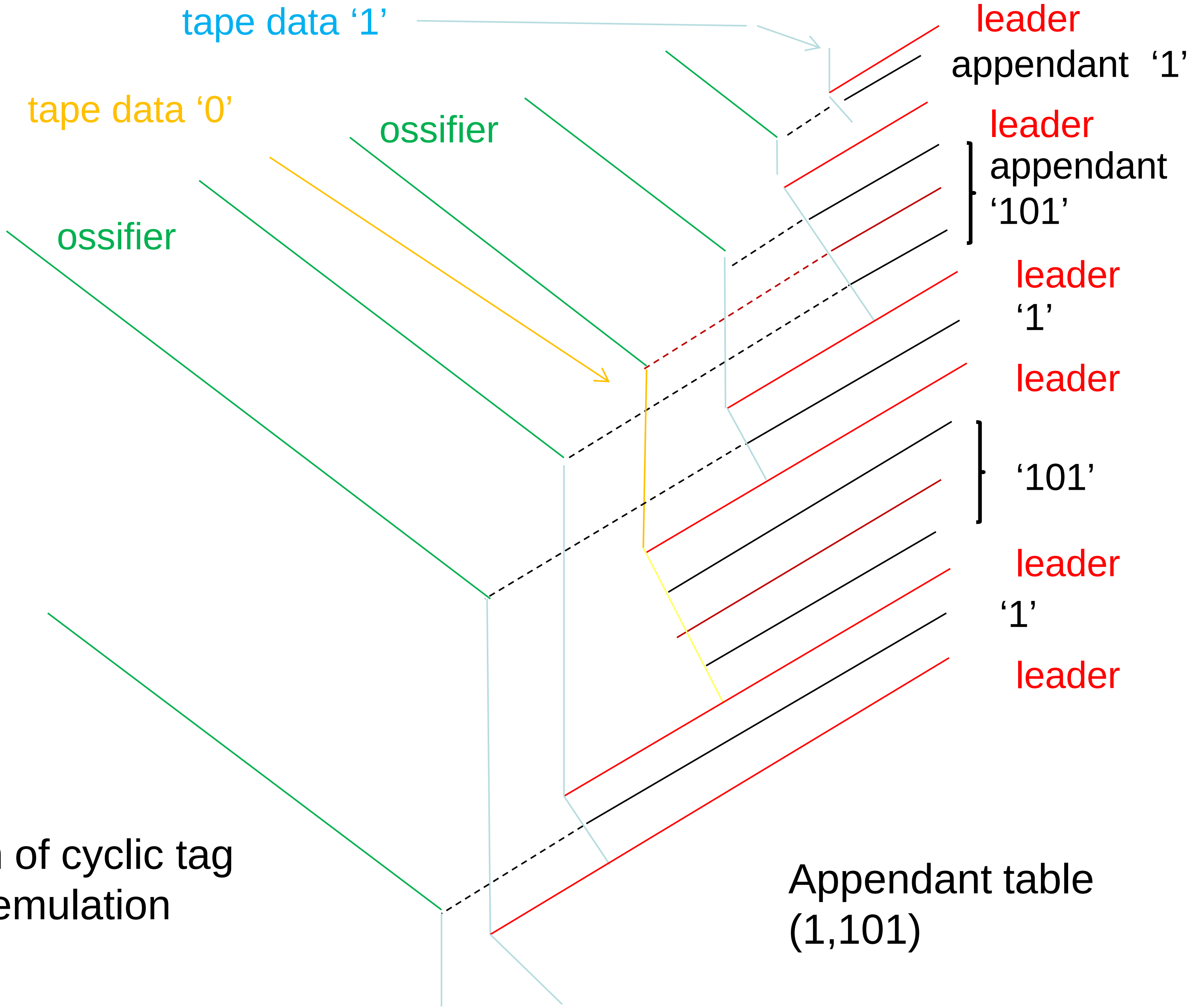
Table data ‘101’
(56200 – 57550)

Leader5
(57550 – 57900)

Table data ’1’
(57900 – 58350)

Leader6
(58350 – 58700)

Right edge
(65899)



tape data ‘1’

leader

appendant ‘1’

tape data ‘0’

ossifier

leader

appendant ‘101’

ossifier

leader

‘1’

leader

‘101’

leader

‘1’

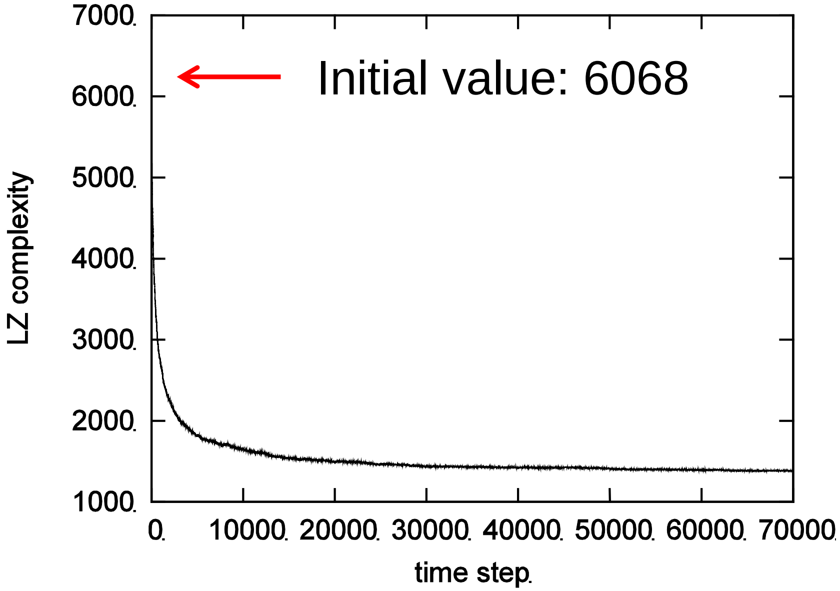
leader

Diagram of cyclic tag system emulation

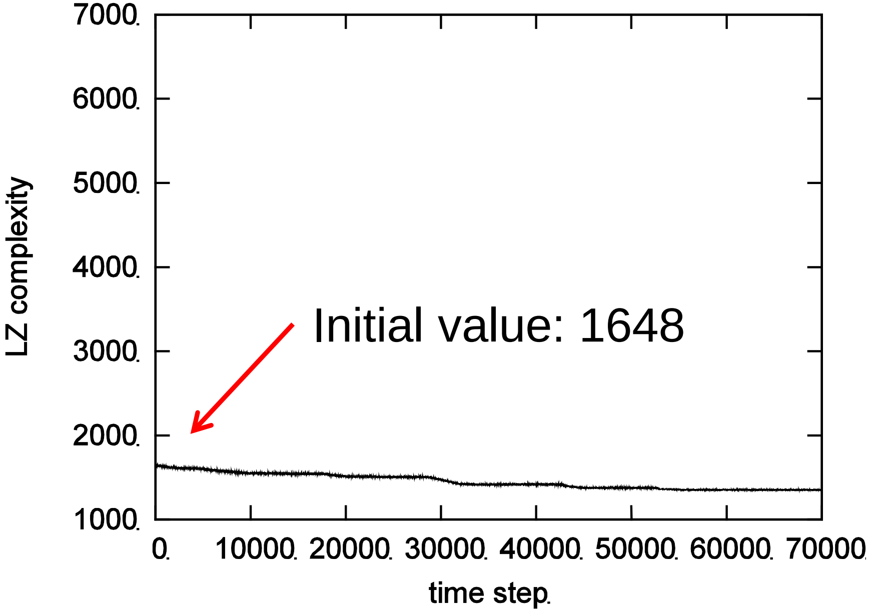
Appendant table (1,101)

Evolution of LZ complexity

Random initial configuration

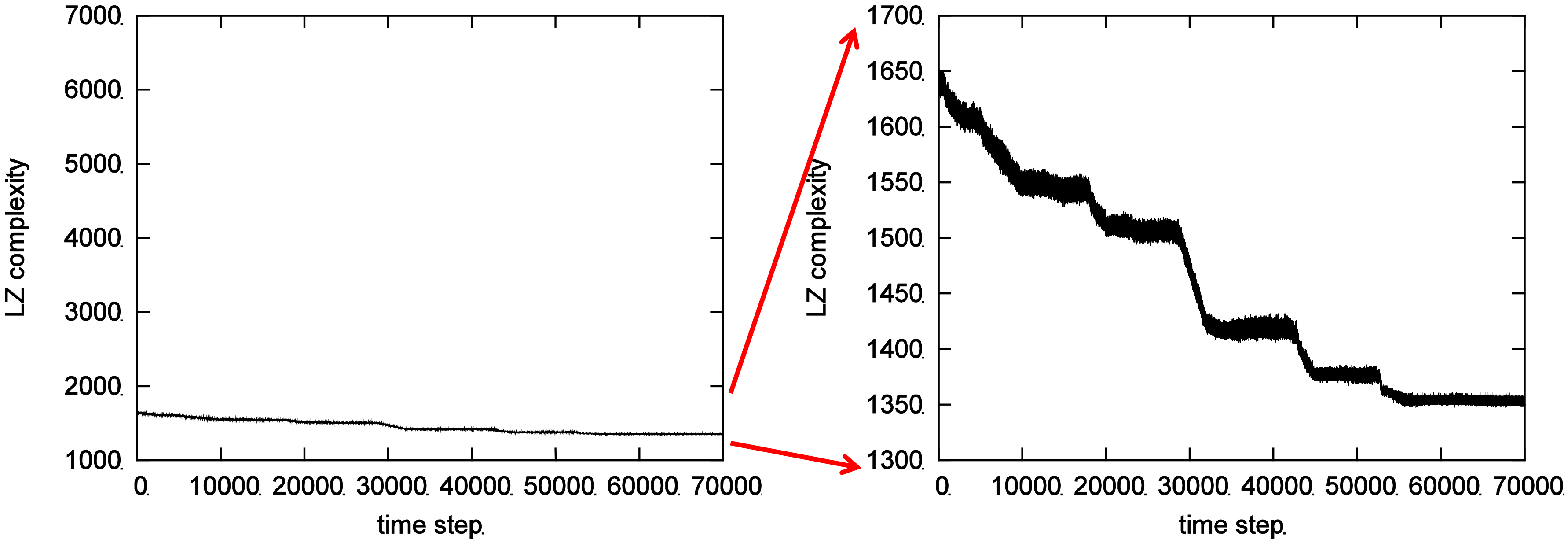


Cyclic tag system emulation

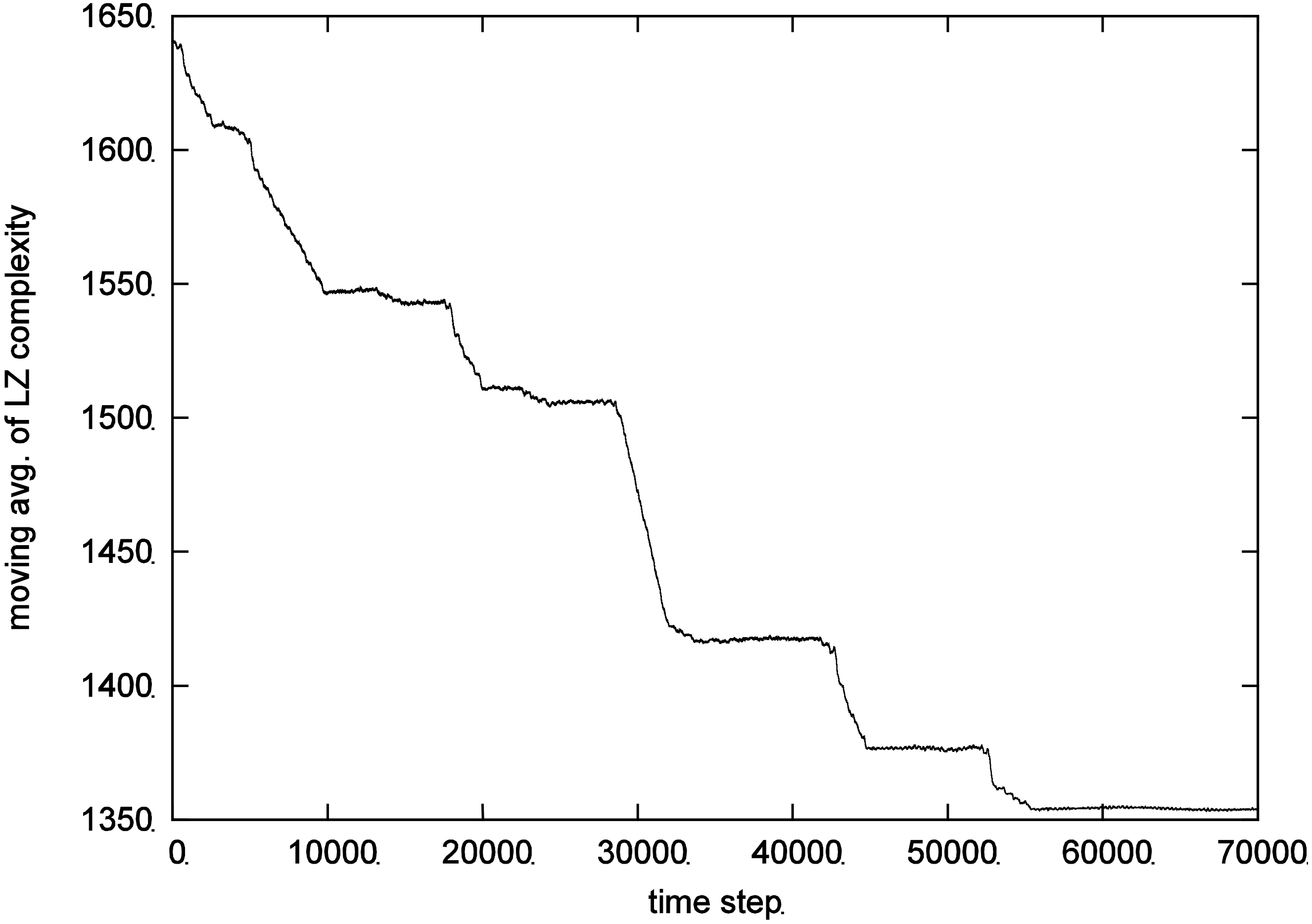


Array size: 65,900

Enlarged view of LZ complexity in cts emulation

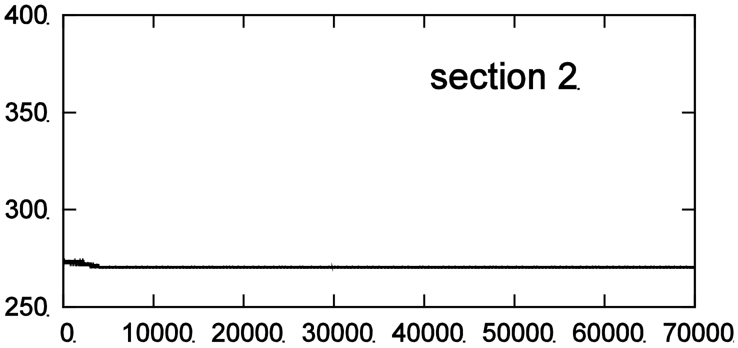


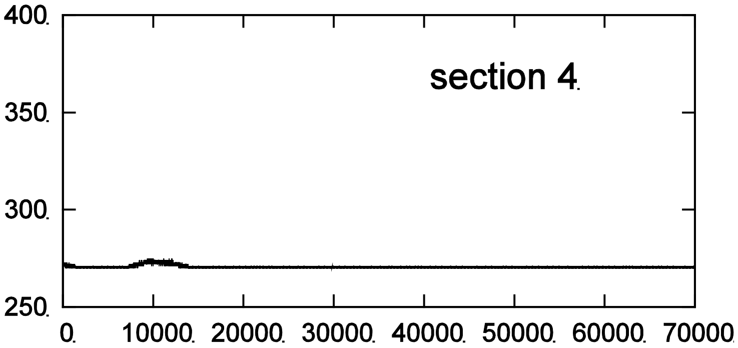
Moving average of LZ complexity in CTS emulation (period: 100)

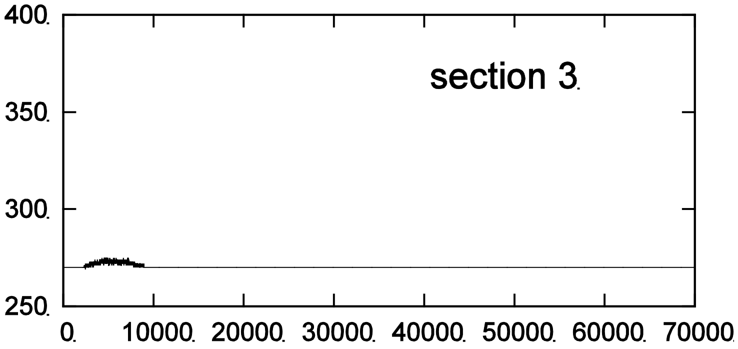


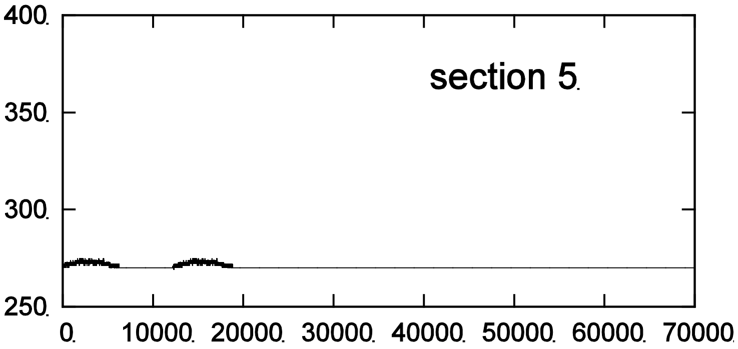
Evolution of LZ complexity in each section

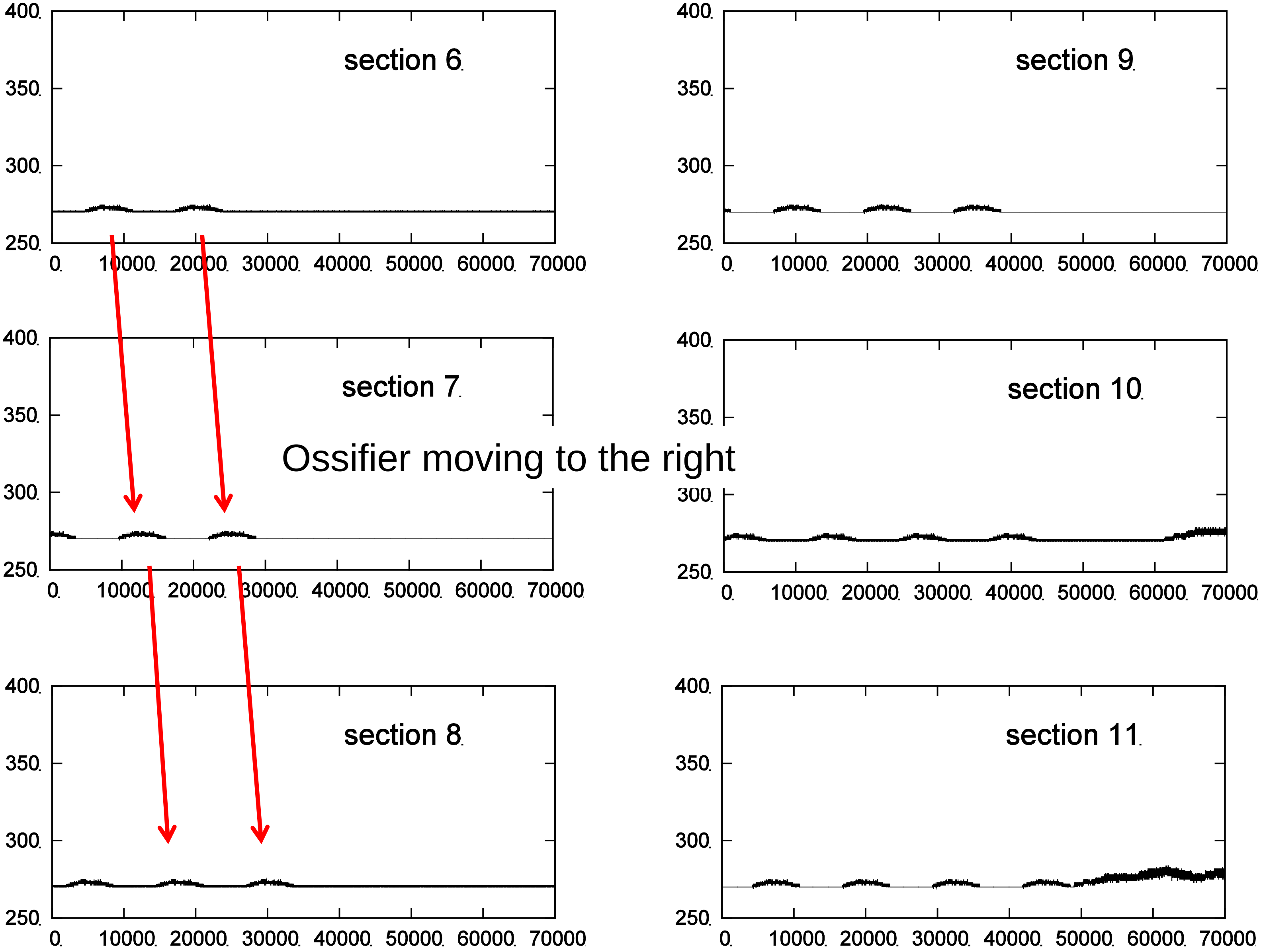
whole array is divided into 20 sections (3,295 cells each)
Sec. 0: leftmost, sec.19: rightmost










section 6

section 7

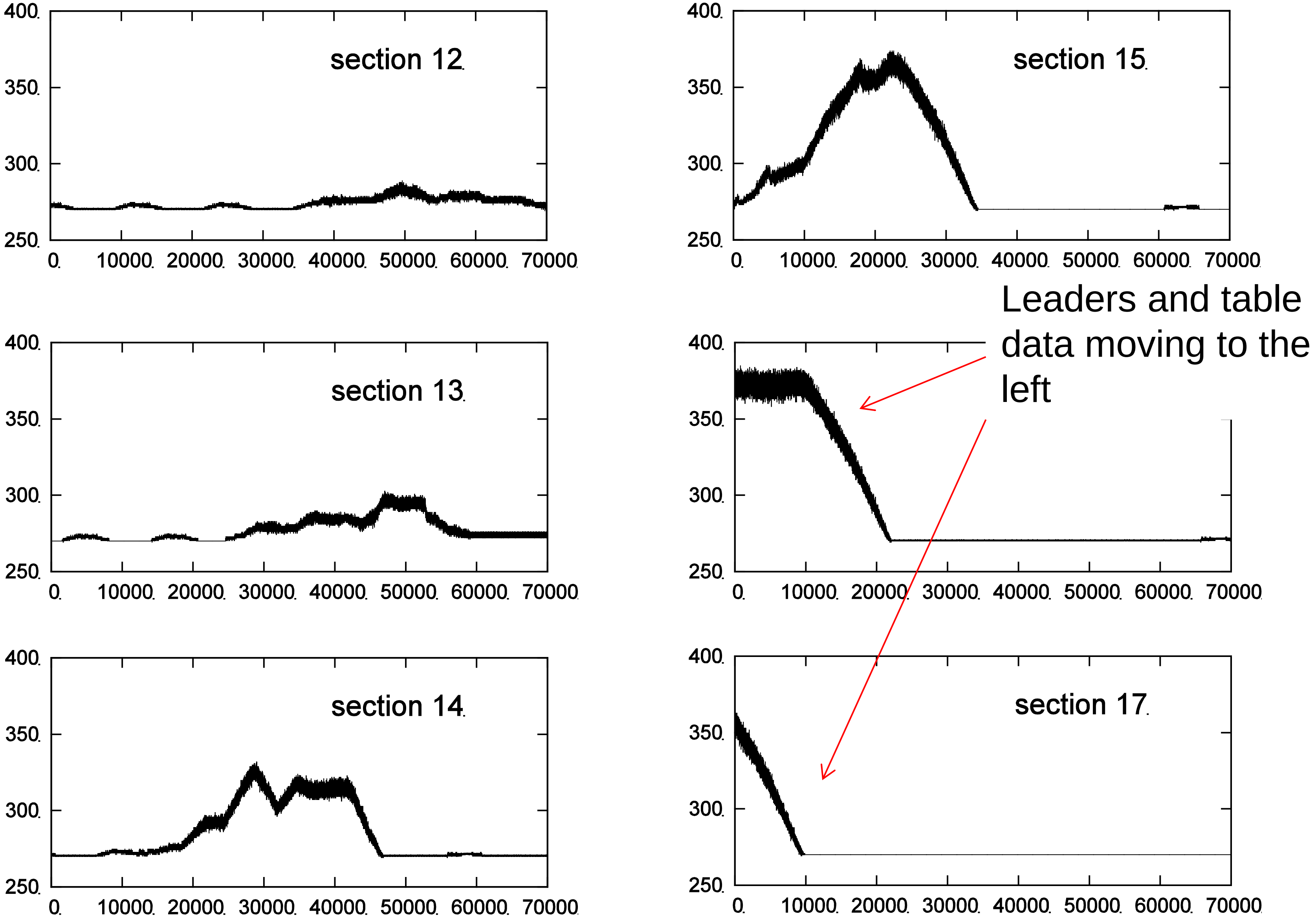
section 8

section 9

section 10

section 11

Ossifier moving to the right

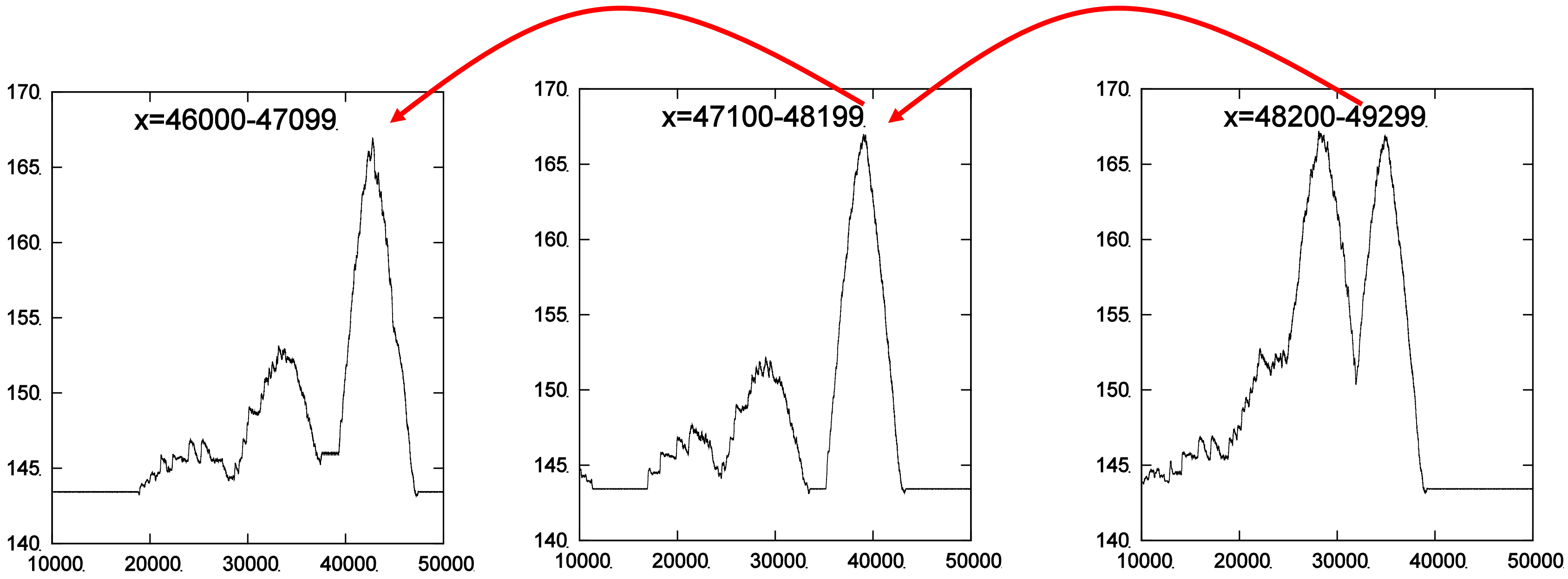


Leaders and table data moving to the left

↑ Look at this in detail

Moving average of LZ complexity in the three parts (array size: 1,100) of sec.14

Leaders and table data moving to the left



x=46,000 – 47,099

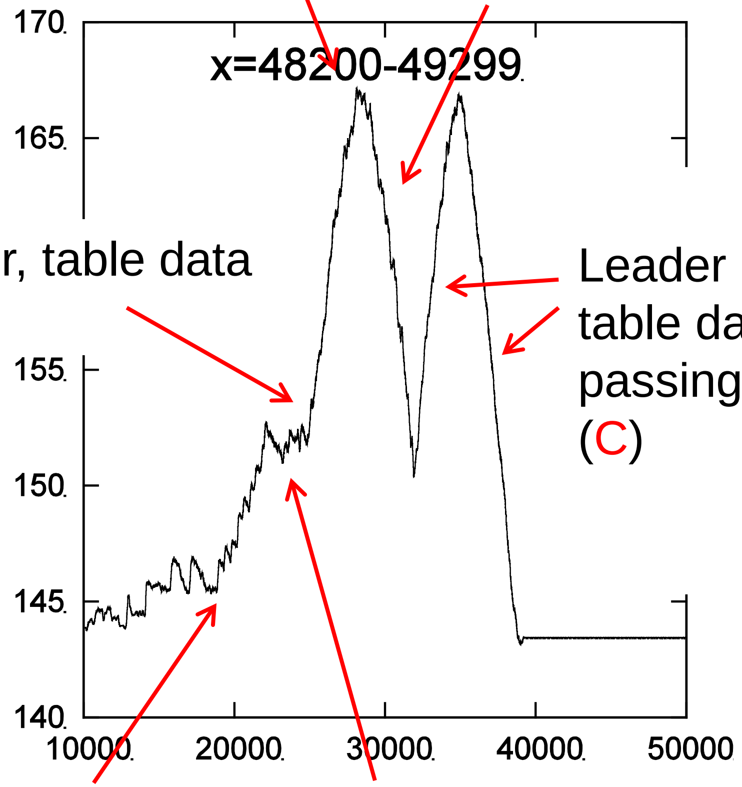
X=47,100 – 48,199

X=48,200 – 49,299

↑
Look at this in detail

Collision between tape data ‘0’ and leader (B)

Rejector erasing table data

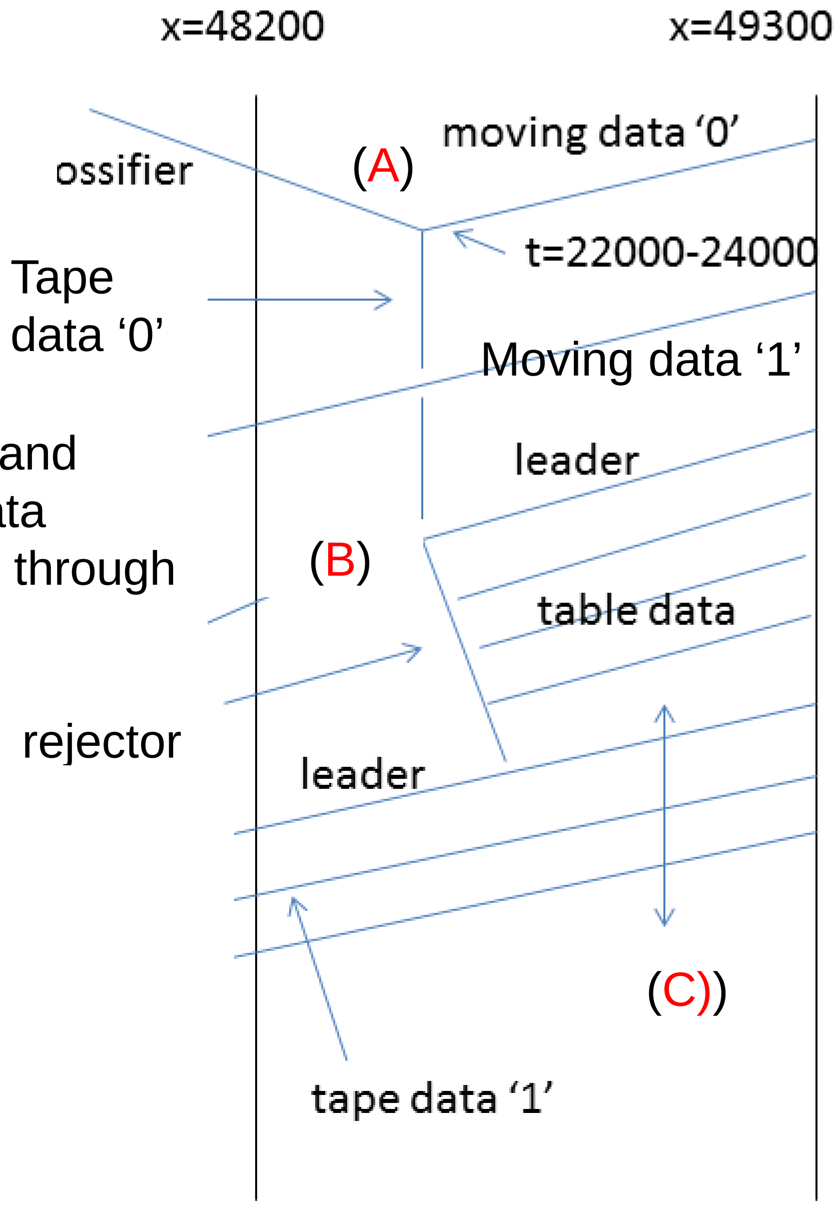


Leader, table data come

Leader and table data passing through (C)

Moving data ‘0’ come from the right

Collision between moving data and ossifier (A)



Moving average of LZ complexity

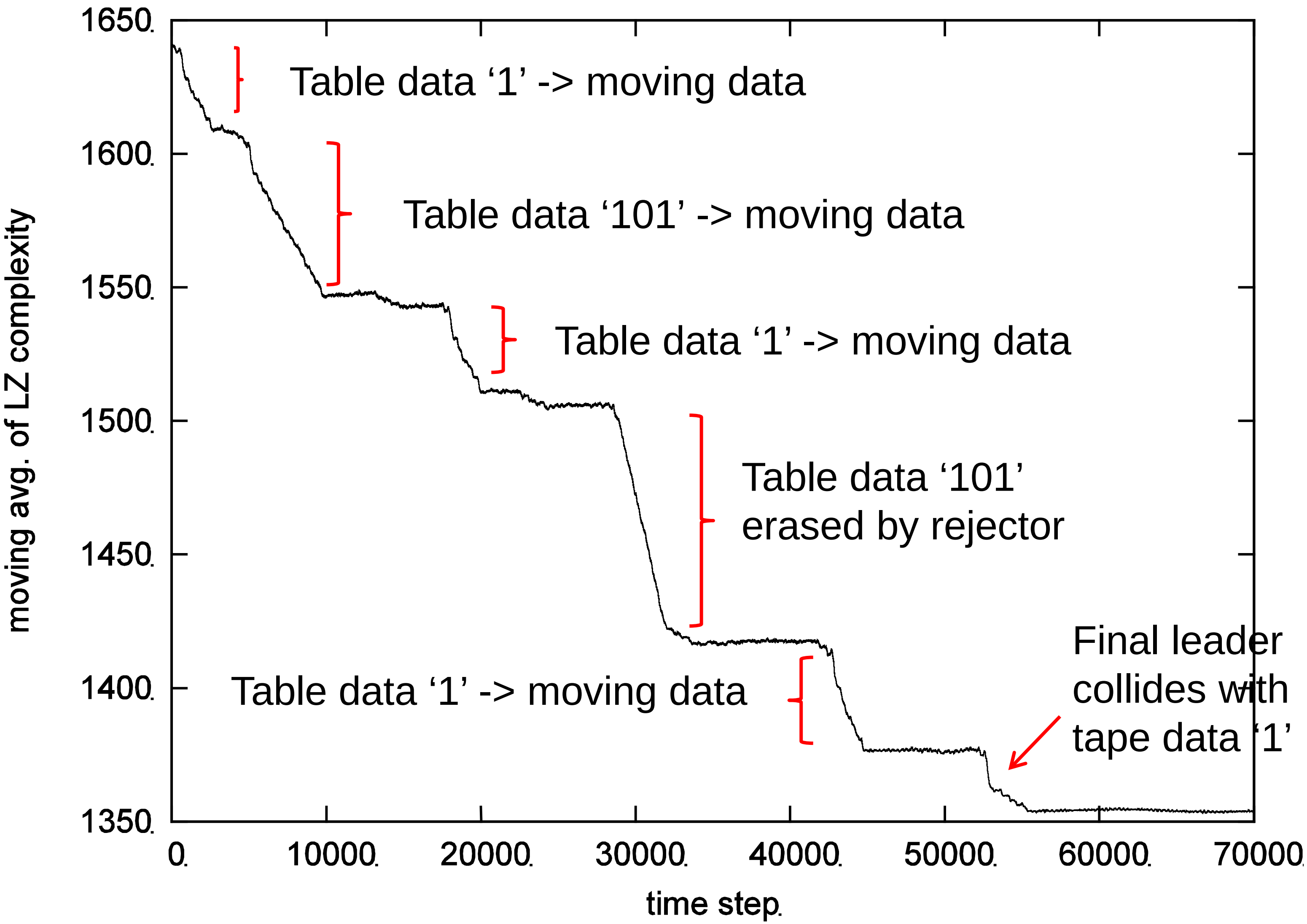


Table data ‘1’ -> moving data

Table data ‘101’ -> moving data

Table data ‘1’ -> moving data

Table data ‘101’
erased by rejector

Table data ‘1’ -> moving data

Final leader
collides with
tape data ‘1’

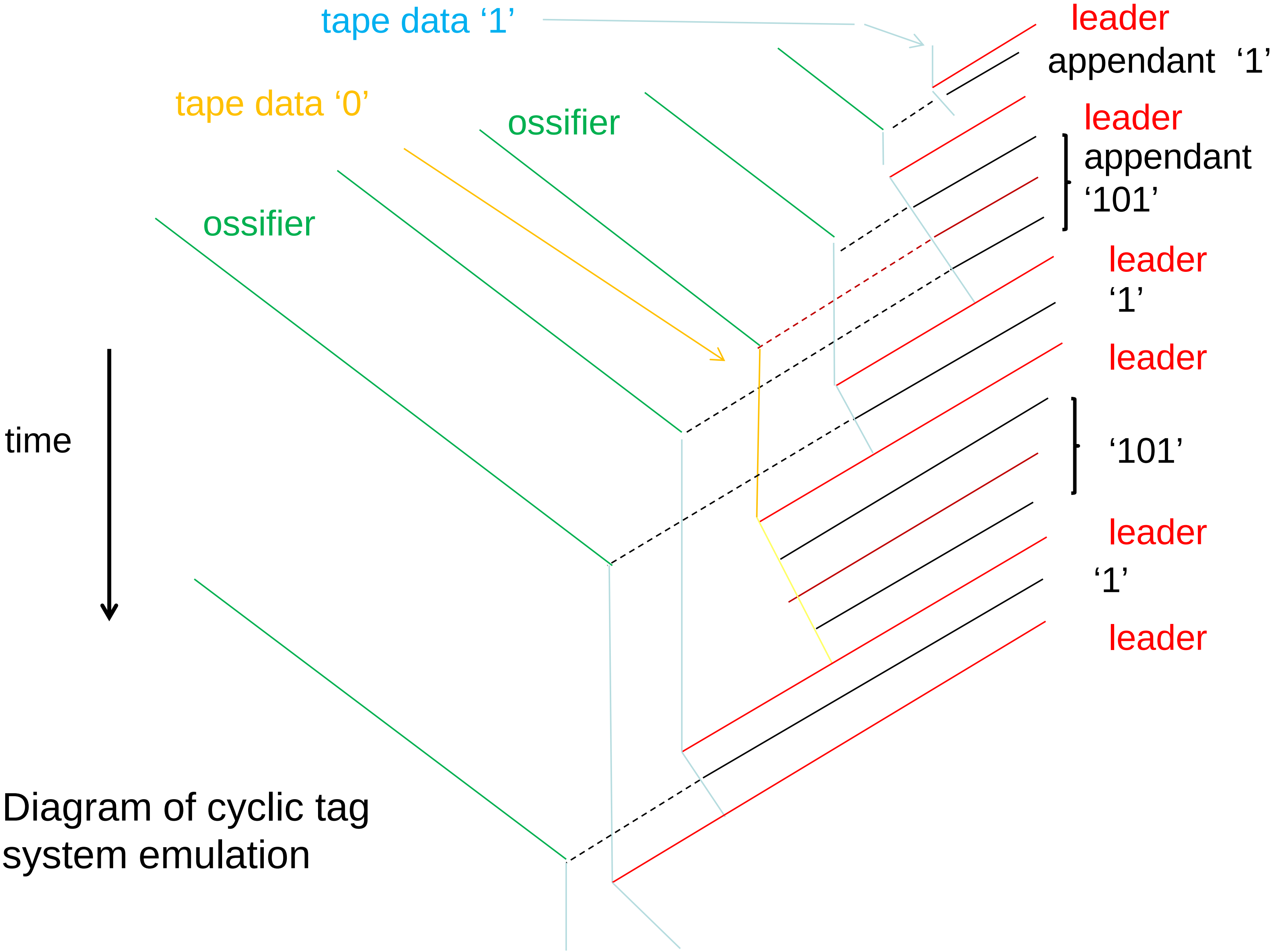
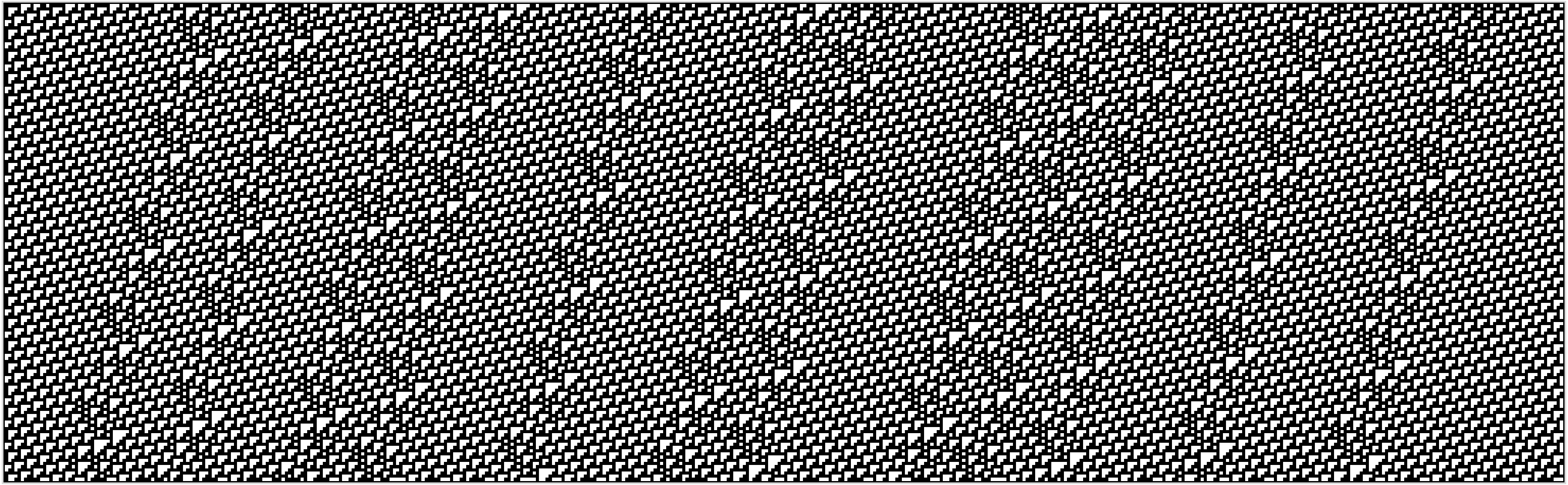
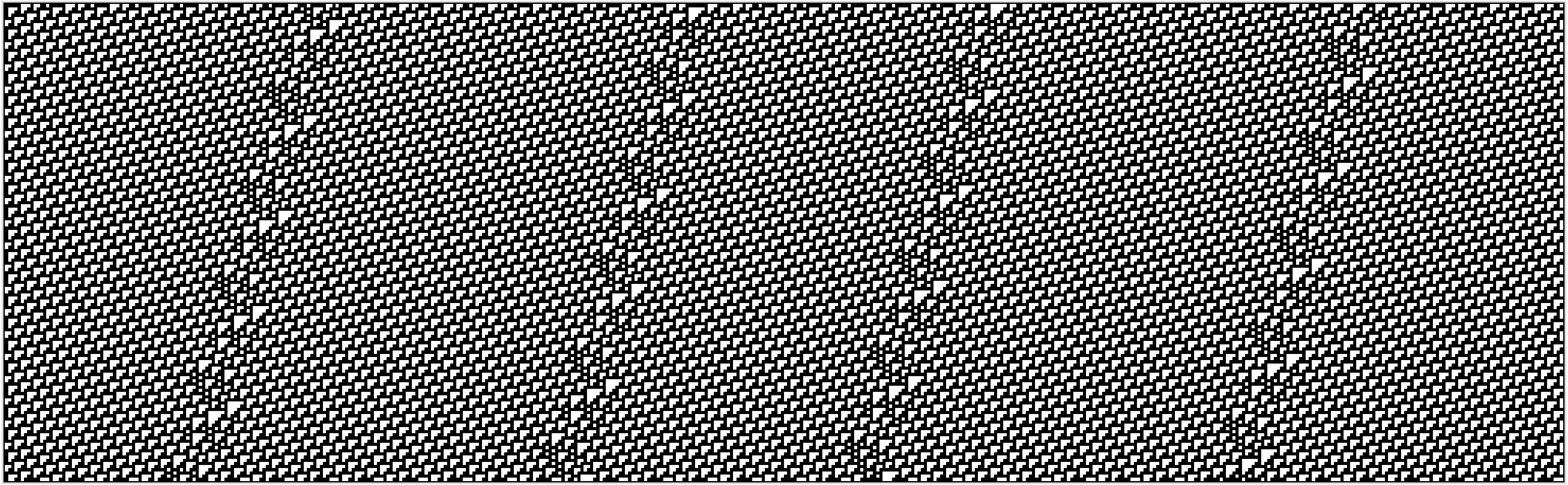


Diagram of cyclic tag system emulation

Table data ‘1’



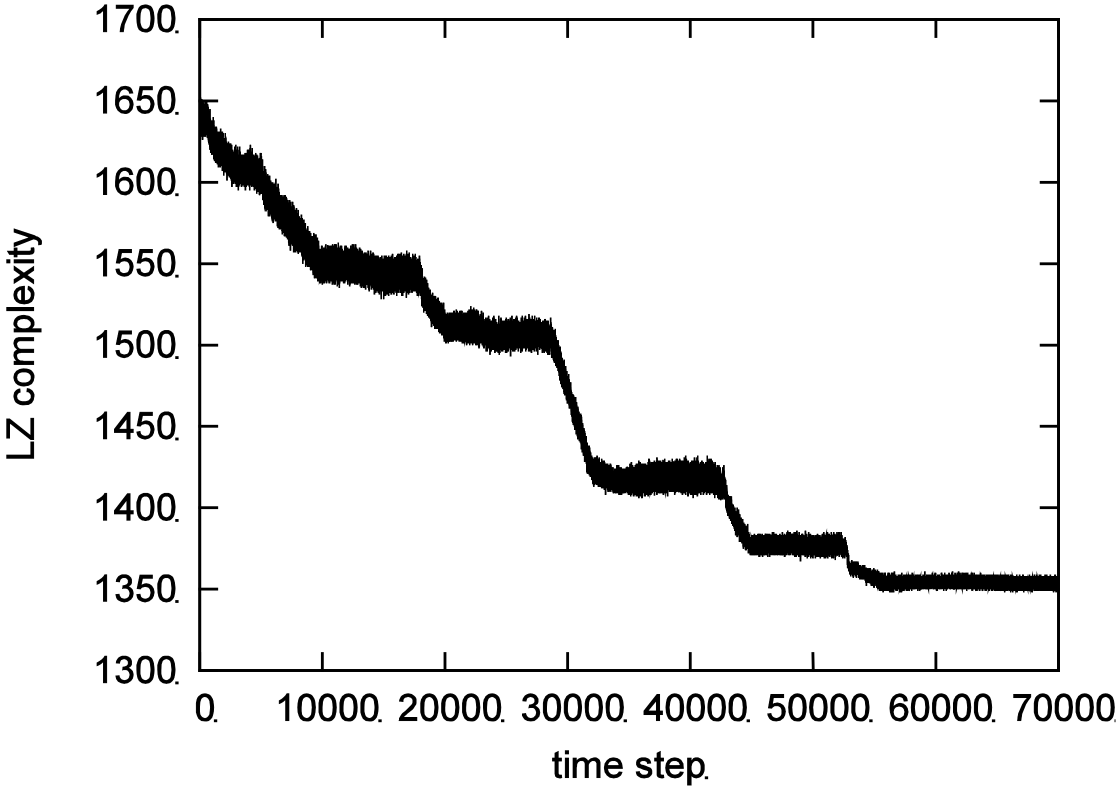
Moving data ‘1’



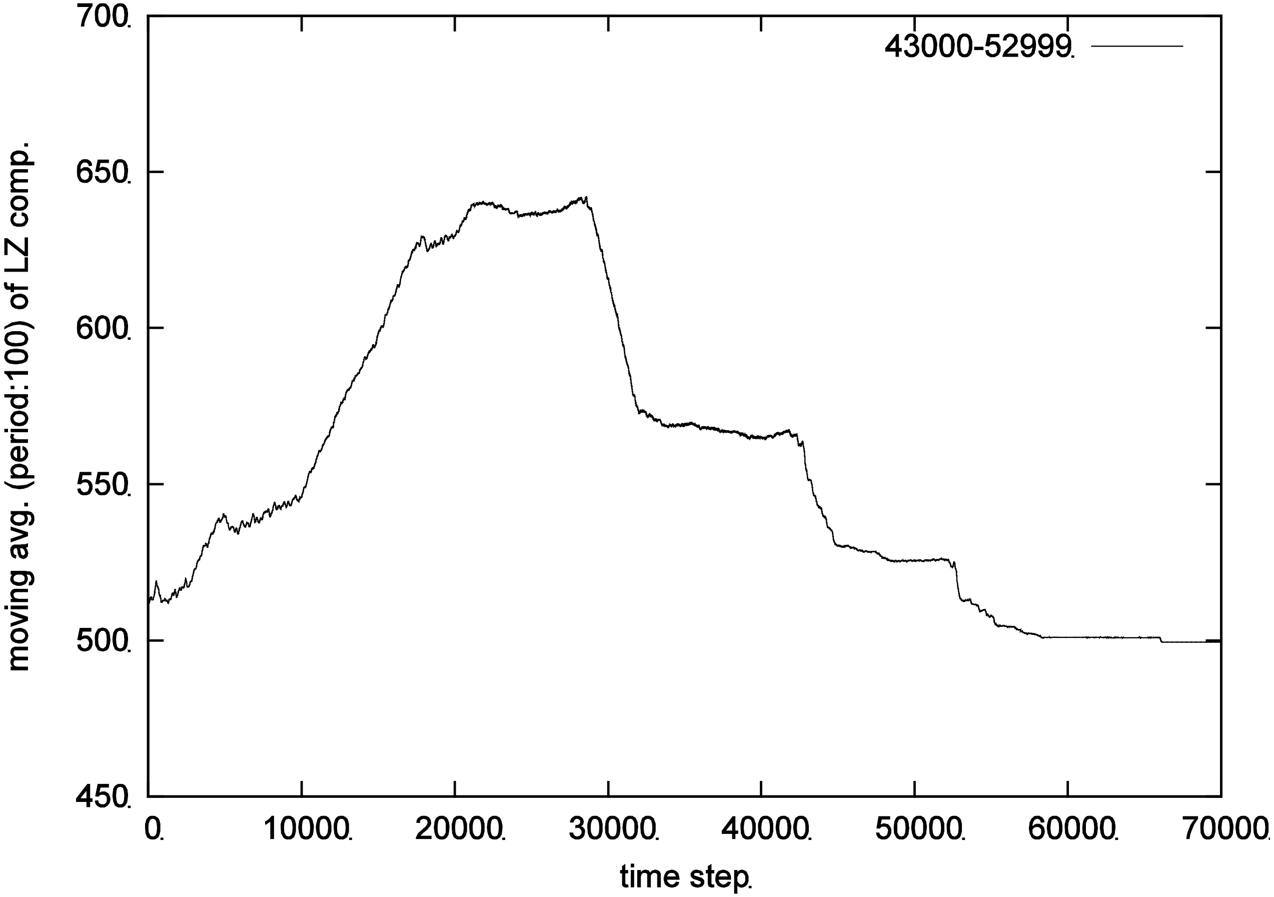
Conclusion

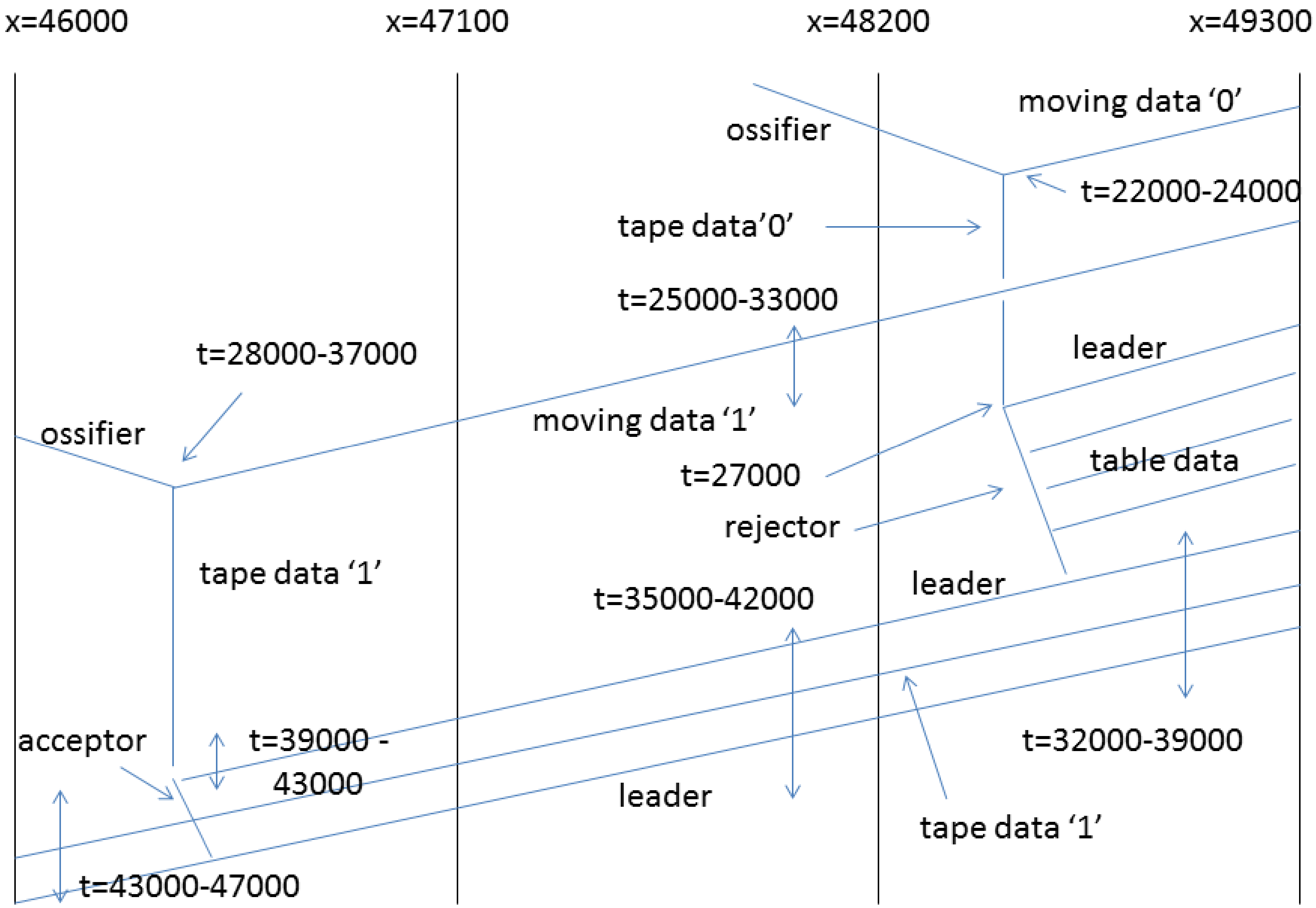
- In the emulation process of CTS by rule 110, stepwise decrease of LZ complexity is observed
- When table data are transformed into moving data by acceptor or erased by rejector, LZc decreases quickly
- These results might generally hold for decision problem solving process.

Thank you for listening!


1700.
1650.
1600.
1550.
1500.
1450.
1400.
1350.
1300.
LZ complexity
0.
10000.
20000.
30000.
40000.
50000.
60000.
70000.
time step.

X=43000..52999




x=46000
x=47100
x=48200
x=49300
moving data ‘0’
ossifier
t=22000-24000
tape data’0’
t=25000-33000
leader
t=28000-37000
moving data ‘1’
ossifier
t=27000
table data
rejector
tape data ‘1’
leader
t=35000-42000
acceptor
t=39000 -
43000
t=32000-39000
leader
tape data ‘1’
t=43000-47000