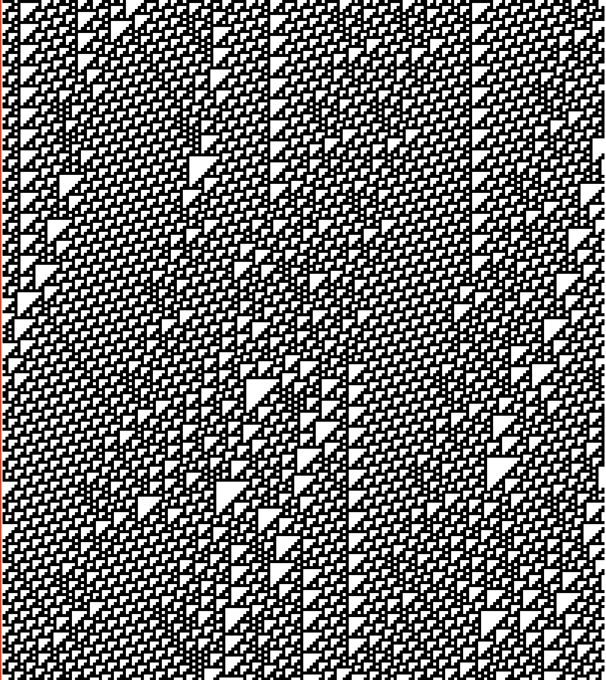
Quantifying Information Modification in Cellular Automata using Pointwise Partial Information Decomposition

CCS 2019

Conor Finn
Joseph Lizier

October 2019

THE UNIVERSITY OF SYDNEY



How can we quantify intrinsic, emergent computation?

How can we quantify intrinsic, emergent computation?

- Turing described computation in terms of
  - information storage
  - information transfer
  - information modification
- Langton (1990) informally discusses emergent computation using this terminology
- Can we formalise these quantities as information-theoretic quantites?

  $\rightarrow$ **Information dynamics**

Information theory

- Mutual information

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$
$$= \mathbb{E}_{X,Y}\left[i(x,y)\right]$$

- Pointwise mutual information

$$i(x;y) = \log \frac{p(x,y)}{p(x)p(y)}$$

- Joint mutual information

$$I(X;YZ) = \sum_{x,y,z} p(x,y,z) \log \frac{p(x,y,z)}{p(x)p(y,z)}$$

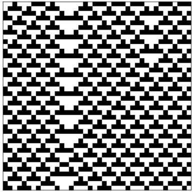
Information dynamics

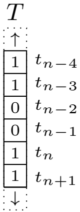
- Use pointwise information theory to quantify
  - storage
  - transfer
  - modification
- Local in time and space

Information dynamics

- Use pointwise information theory to quantify
  - storage
  - transfer
  - modification
- Local in time and space

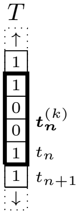
Rule 54



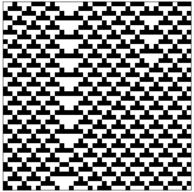


Information dynamics

- Use pointwise information theory to quantify
  - storage
  - transfer
  - modification
- Local in time and space

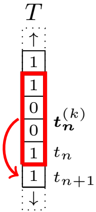


Rule 54

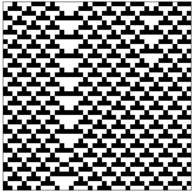


Information dynamics

- Use pointwise information theory to quantify
  - storage ← Lizier et al. (2012) $a_T(n) = i\left(t_{n+1}; \boldsymbol{t_n^{(k)}}\right)$
  - transfer
  - modification
- Local in time and space

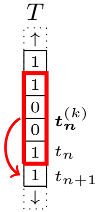


Rule 54



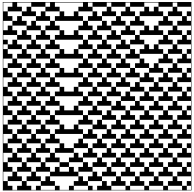
Information dynamics

- Use pointwise information theory to quantify
  - storage ← Lizier et al. (2012) $a_T(n) = i\left(t_{n+1}; \boldsymbol{t}_{\boldsymbol{n}}^{(k)}\right)$
  - transfer
  - modification

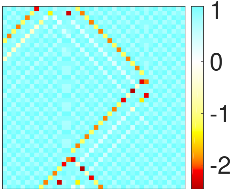


- Local in time and space

Rule 54

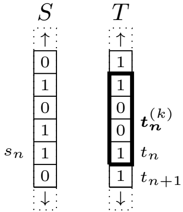


Storage



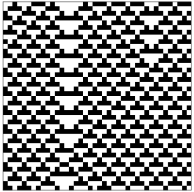
Information dynamics

- Use pointwise information theory to quantify
  - storage $\leftarrow$ Lizier et al. (2012) $a_T(n) = i\left(t_{n+1}; \boldsymbol{t_n^{(k)}}\right)$
  - transfer
  - modification

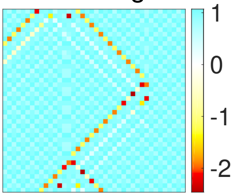


- Local in time and space

Rule 54

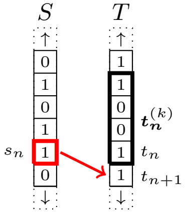


Storage



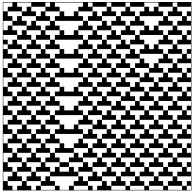
Information dynamics

- Use pointwise information theory to quantify
  - storage $\leftarrow$ Lizier et al. (2012) $a_T(n) = i\big(t_{n+1}; \boldsymbol{t_n^{(k)}}\big)$
  - transfer $\leftarrow$ Lizier et al. (2008) $t_{S\to T}(n) = i\big(t_{n+1}; s_n \big| \boldsymbol{t_n^{(k)}}\big)$
  - modification

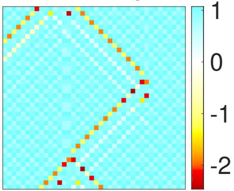


- Local in time and space

Rule 54

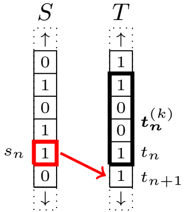


Storage



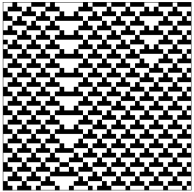
Information dynamics

- Use pointwise information theory to quantify
  - storage ← Lizier et al. (2012) $a_T(n) = i\big(t_{n+1}; \boldsymbol{t_n^{(k)}}\big)$
  - transfer ← Lizier et al. (2008) $t_{S\to T}(n) = i\big(t_{n+1}; s_n \big| \boldsymbol{t_n^{(k)}}\big)$
  - modification

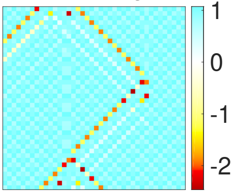


- Local in time and space

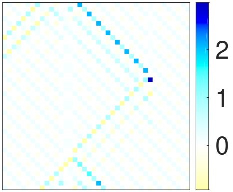
Rule 54



Storage

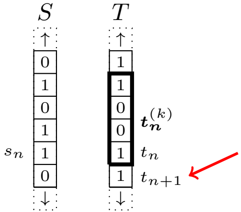


Transfer from left

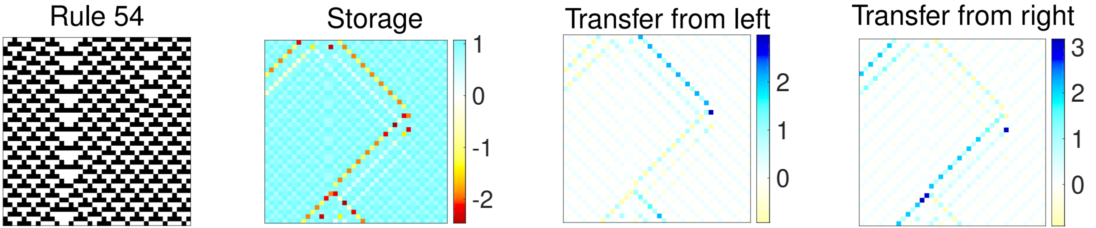


Information dynamics

- Use pointwise information theory to quantify
  - storage $\leftarrow$ Lizier et al. (2012) $a_T(n) = i\big(t_{n+1}; \boldsymbol{t_n^{(k)}}\big)$
  - transfer $\leftarrow$ Lizier et al. (2008) $t_{S\to T}(n) = i\big(t_{n+1}; s_n \big| \boldsymbol{t_n^{(k)}}\big)$
  - modification

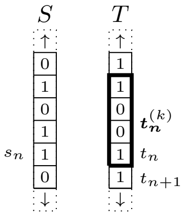


- Local in time and space



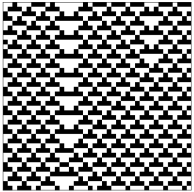
Information dynamics

- Use pointwise information theory to quantify
  - storage ← Lizier et al. (2012) $a_T(n) = i\big(t_{n+1}; \boldsymbol{t_n^{(k)}}\big)$
  - transfer ← Lizier et al. (2008) $t_{S\to T}(n) = i\big(t_{n+1}; s_n \big| \boldsymbol{t_n^{(k)}}\big)$
  - modification ← this talk

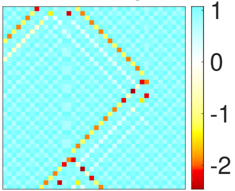


- Local in time and space

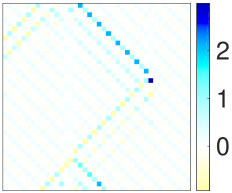
Rule 54



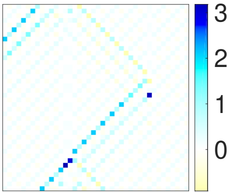
Storage

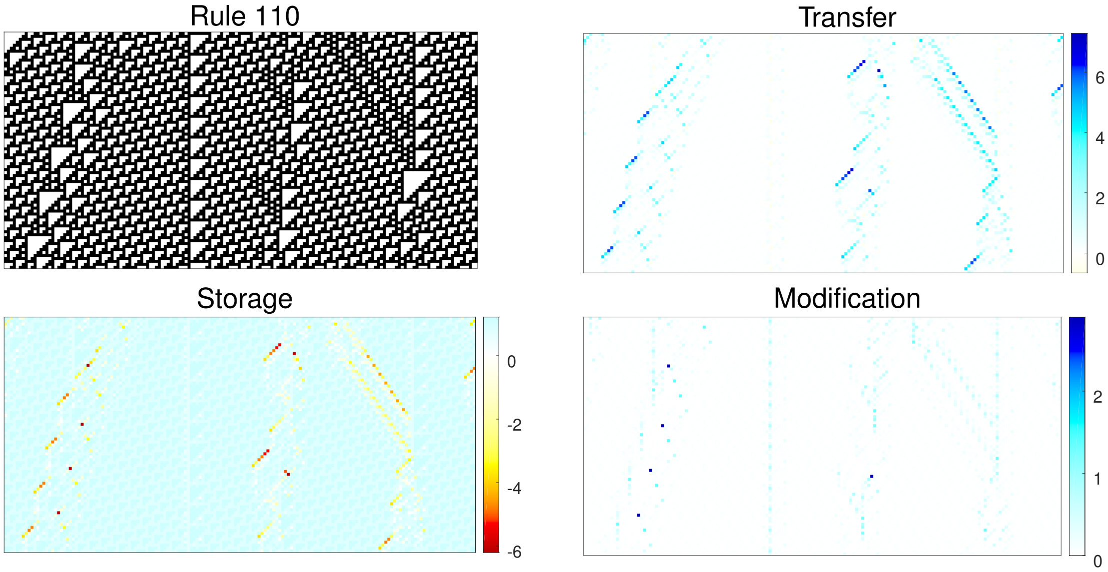


Transfer from left



Transfer from right





Rule 110
Transfer
6
4
2
0
Storage
0
-2
-4
-6
Modification
2
1
0

How can we quantify information modification?

Lizier et al. (2010)
– Separable information heuristic $\rightarrow$ conflates redundant and synergistic information

Langton (1990)
– An interaction between transmitted and stored information which changes either

Lizier et al. (2013)
– Define non-modified information to antonymically define modified information
– Non-modified information is information identifiable in any **single** source
– Modified information is a non-trivial synthesis of **two or more** sources

Information decomposition

Consider trying to predict T from S_1 and S_2

- Several types of information

Information decomposition

Consider trying to predict T from S_1 and S_2

- Several types of information
  - **Unique information** $U(S_1 \setminus S_2 \to T)$

Unq

p	s_1	s_2	t
1/4	0	0	0
1/4	0	1	0
1/4	1	0	1
1/4	1	1	1

Information decomposition

Consider trying to predict T from S_1 and S_2

- Several types of information
  - **Unique information** $U(S_1 \backslash S_2 \to T)$
  - **Redundant information** $R(S_1, S_2 \to T)$

Unq

p	s_1	s_2	t
1/4	0	0	0
1/4	0	1	0
1/4	1	0	1
1/4	1	1	1

Rdn

p	s_1	s_2	t
1/2	0	0	0
1/2	1	1	1

Information decomposition

Consider trying to predict T from S_1 and S_2

- Several types of information
  - **Unique information** $U(S_1 \setminus S_2 \to T)$
  - **Redundant information** $R(S_1, S_2 \to T)$
  - **Synergistic information** $C(S_1, S_2 \to T)$

UNQ

p	s_1	s_2	t
1/4	0	0	0
1/4	0	1	0
1/4	1	0	1
1/4	1	1	1

RDN

p	s_1	s_2	t
1/2	0	0	0
1/2	1	1	1

XOR

p	s_1	s_2	t
1/4	0	0	0
1/4	0	1	1
1/4	1	0	1
1/4	1	1	0

Information decomposition

Consider trying to predict T from S_1 and S_2

- Several types of information
  - **Unique information** $U(S_1 \setminus S_2 \to T)$
  - **Redundant information** $R(S_1, S_2 \to T)$
  - **Synergistic information** $C(S_1, S_2 \to T)$

- Mutual information captures

$$I(T; S_1) = R(T : S_1, S_2) + U(T : S_1 \setminus S_2)$$
$$I(T; S_2) = R(T : S_1, S_2) + U(T : S_2 \setminus S_1)$$

UNQ

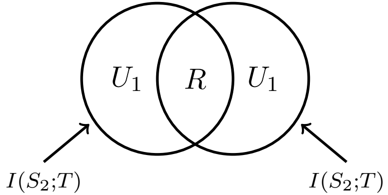
p	s_1	s_2	t
1/4	0	0	0
1/4	0	1	0
1/4	1	0	1
1/4	1	1	1

RDN

p	s_1	s_2	t
1/2	0	0	0
1/2	1	1	1

XOR

p	s_1	s_2	t
1/4	0	0	0
1/4	0	1	1
1/4	1	0	1
1/4	1	1	0



Information decomposition

Consider trying to predict T from S_1 and S_2

- Several types of information
  - **Unique information** $U(S_1 \backslash S_2 \to T)$
  - **Redundant information** $R(S_1, S_2 \to T)$
  - **Synergistic information** $C(S_1, S_2 \to T)$

UNQ

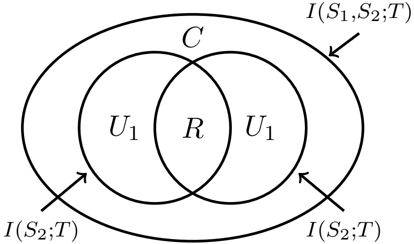
p	s_1	s_2	t
1/4	0	0	0
1/4	0	1	0
1/4	1	0	1
1/4	1	1	1

RDN

p	s_1	s_2	t
1/2	0	0	0
1/2	1	1	1

XOR

p	s_1	s_2	t
1/4	0	0	0
1/4	0	1	1
1/4	1	0	1
1/4	1	1	0



- Mutual information captures

$$I(T;S_1) = R(T:S_1,S_2) + U(T:S_1 \backslash S_2)$$
$$I(T;S_2) = R(T:S_1,S_2) + U(T:S_2 \backslash S_1)$$

- Joint mutual information captures

$$I(T;S_1 S_2) = R(S_1,S_2 \to T) + U(S_1\ S_2 \to T) + U(S_2\ S_1 \to T) + C(S_1,S_2 \to T)$$

Partial Information Decomposition (Williams and Beer, 2010)

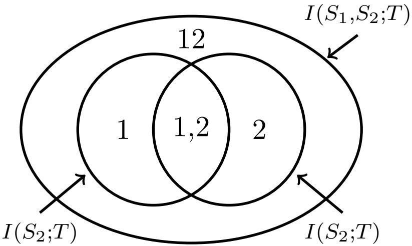
- Axioms for redundant information
  1. Commutativity
  2. Monotonically decreasing
  3. Self-redundancy (idempotency)

- Yields a redundancy lattice

Partial Information Decomposition (Williams and Beer, 2010)

- Axioms for redundant information
  1. Commutativity
  2. Monotonically decreasing
  3. Self-redundancy (idempotency)

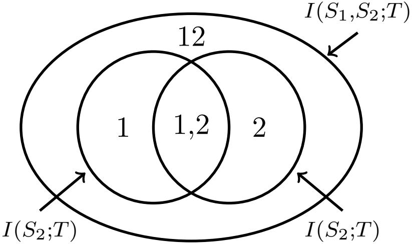
- Yields a redundancy lattice

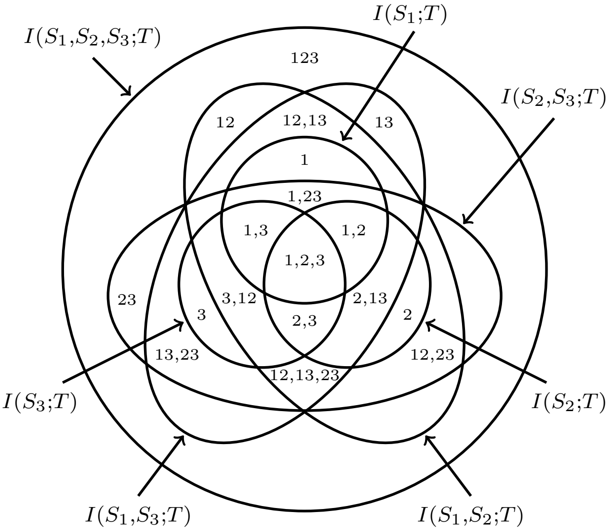


Partial Information Decomposition (Williams and Beer, 2010)

- Axioms for redundant information
  1. Commutativity
  2. Monotonically decreasing
  3. Self-redundancy (idempotency)

- Yields a redundancy lattice





PID is elegant, however...

- Unique evaluation requires a definition of redundant information
  - providing this definition has been a contentious area of research
- Most appoaches do not work for two or more sources (not very useful)
- Information dynamics requires a pointwise information decomposition

entropy

Article

Pointwise Partial Information Decomposition Using the Specificity and Ambiguity Lattices

Conor Finn [1,2,*] and Joseph T. Lizier [1]

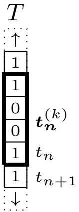
[1] Complex Systems Research Group and Centre for Complex Systems, Faculty of Engineering & IT, The University of Sydney, NSW 2006, Australia; joseph.lizier@sydney.edu.au
[2] CSIRO Data61, Marsfield NSW 2122, Australia
* Correspondence: conor.finn@sydney.edu.au

Received: 10 July 2017; Accepted: 10 April 2018; Published: 18 April 2018

PID and Information Dynamics

Order 1 information

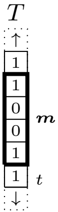
- identifiable in single sources
- non-modified information



PID and Information Dynamics

Order 1 information

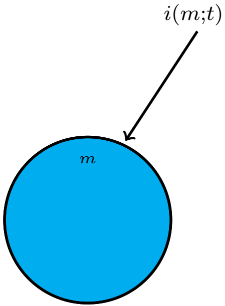
- identifiable in single sources
- non-modified information

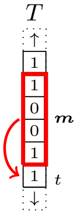


PID and Information Dynamics

Order 1 information

- identifiable in single sources
- non-modified information

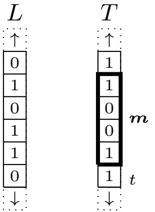


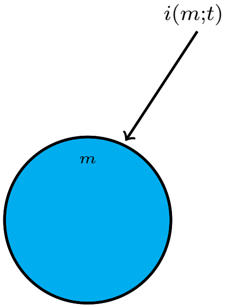


PID and Information Dynamics

Order 1 information

- identifiable in single sources
- non-modified information

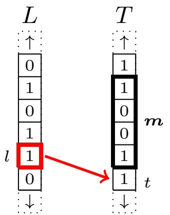


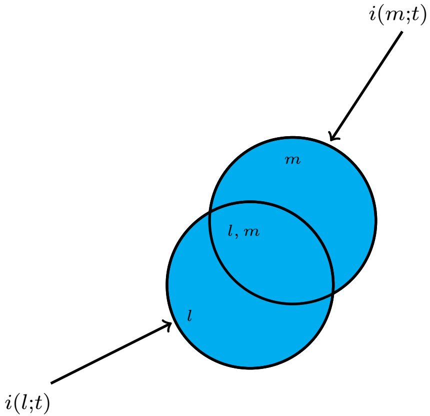


PID and Information Dynamics

Order 1 information

- identifiable in single sources
- non-modified information

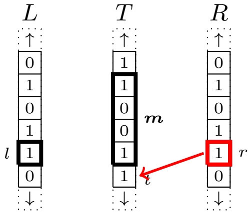


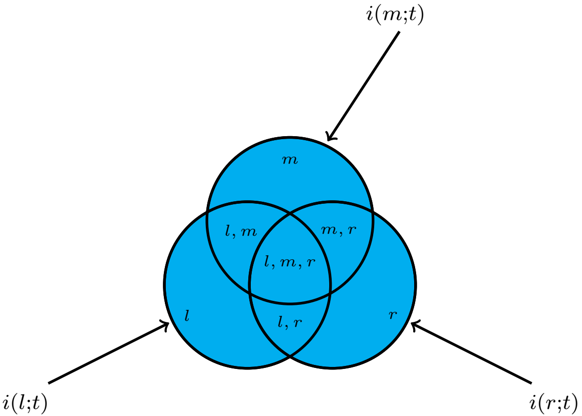


PID and Information Dynamics

Order 1 information

- identifiable in single sources
- non-modified information





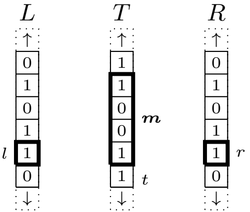
PID and Information Dynamics

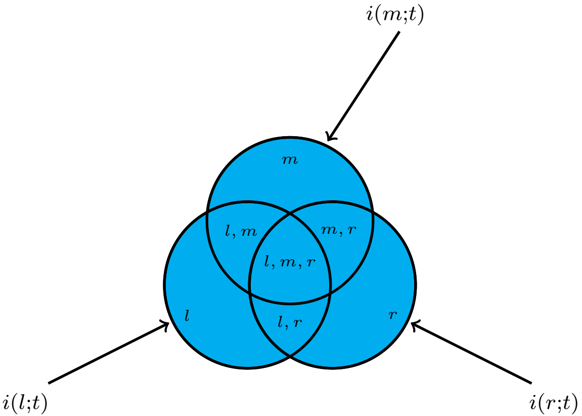
Order 1 information

- identifiable in single sources
- non-modified information

Order 2 information

- identifiable in pairs of source





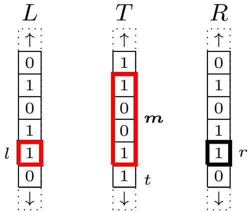
PID and Information Dynamics

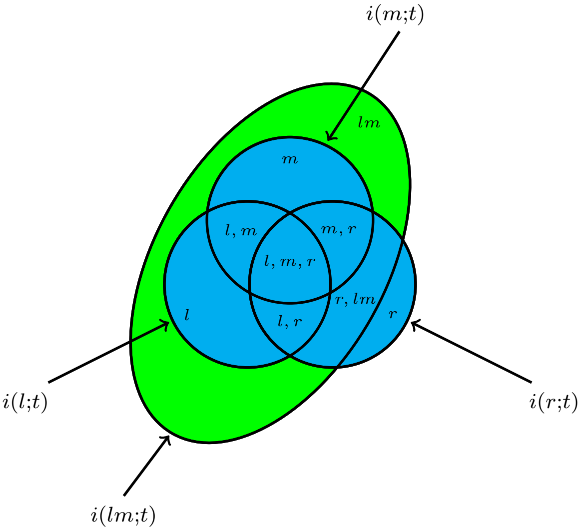
Order 1 information

- identifiable in single sources
- non-modified information

Order 2 information

- identifiable in pairs of source





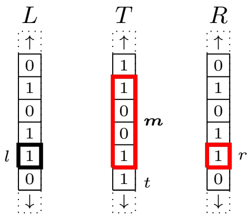
PID and Information Dynamics

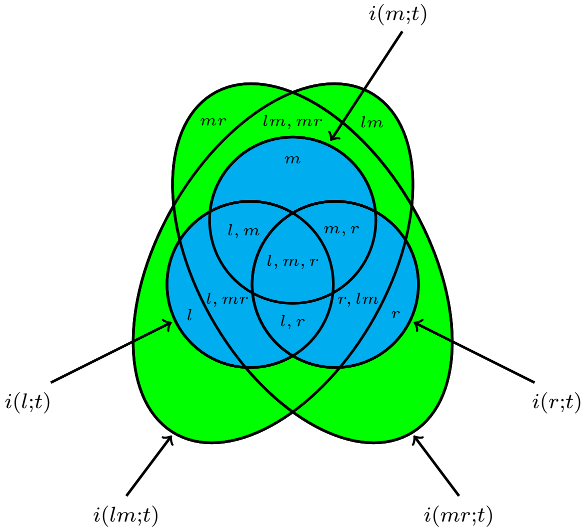
Order 1 information

- identifiable in single sources
- non-modified information

Order 2 information

- identifiable in pairs of source





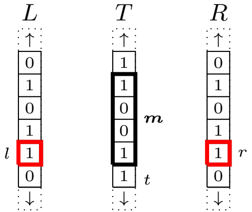
PID and Information Dynamics

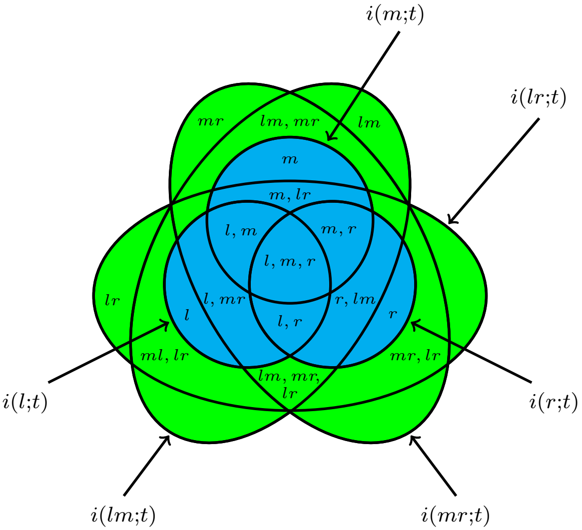
Order 1 information

- identifiable in single sources
- non-modified information

Order 2 information

- identifiable in pairs of source



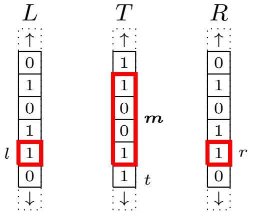


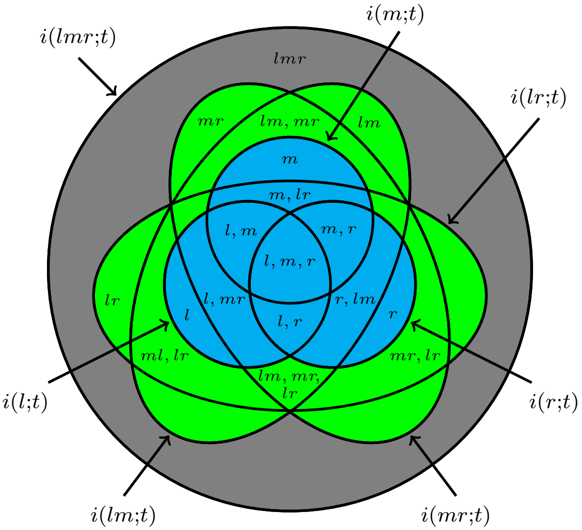
PID and Information Dynamics

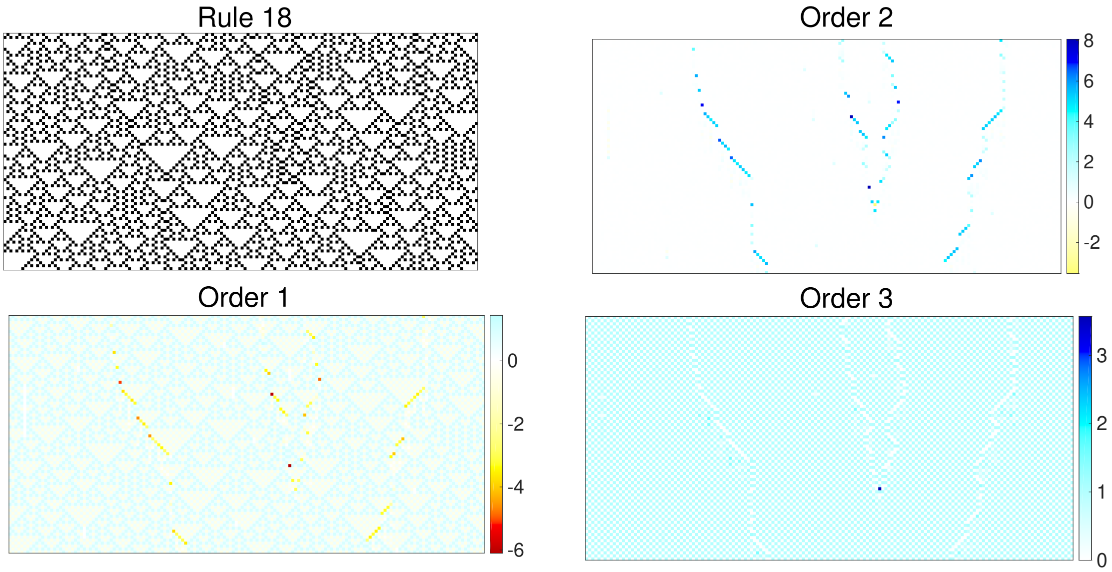
Order 1 information
- identifiable in single sources
- non-modified information

Order 2 information
- identifiable in pairs of source

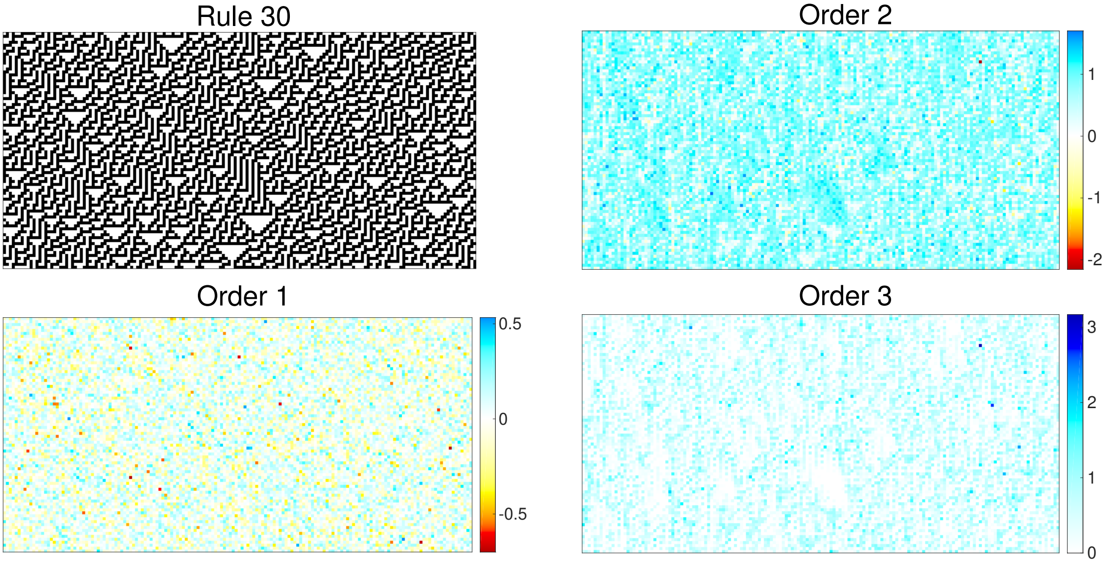
Order 3 information
- identifiable in the triplet







Rule 18
Order 2
8
6
4
2
0
-2
Order 1
0
-2
-4
-6
Order 3
3
2
1
0


Rule 30
Order 2
1
0
-1
-2
Order 1
0.5
0
-0.5
Order 3
3
2
1
0

Questions?

- Full paper on information modification will appear soon
- An extended abstract is available:
  – `https://finnconor.github.io/publications/`
- Measures will soon be released in JIDT:
  – `https://github.com/jlizier/jidt`
- Interested in information decomposition:
  – Information Processing in Complex Systems satellite
  – 12:00 in LHN-TR+04: “Generalised Measures of Multivariate Information Content”

References

Conor Finn and Joseph T. Lizier. Pointwise partial information decomposition using the specificity and ambiguity lattices. *Entropy*, 20(4), 2018.

Chris G Langton. Computation at the edge of chaos: phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1-3):12–37, 1990.

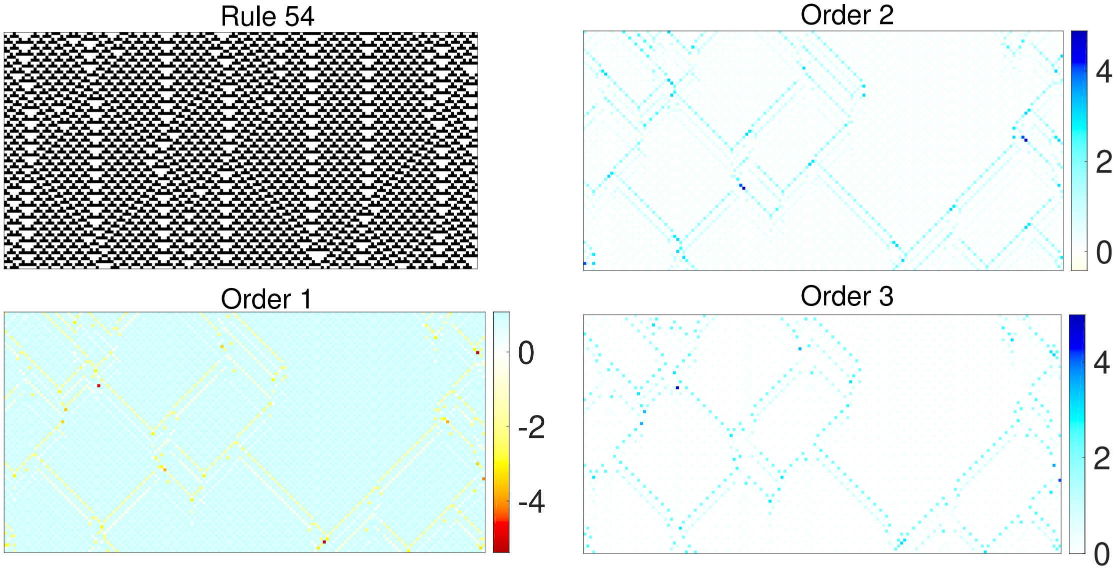
Joseph T Lizier, Mikhail Prokopenko, and Albert Y Zomaya. Local information transfer as a spatiotemporal filter for complex systems. *Physical Review E*, 77(2):026110, 2008.

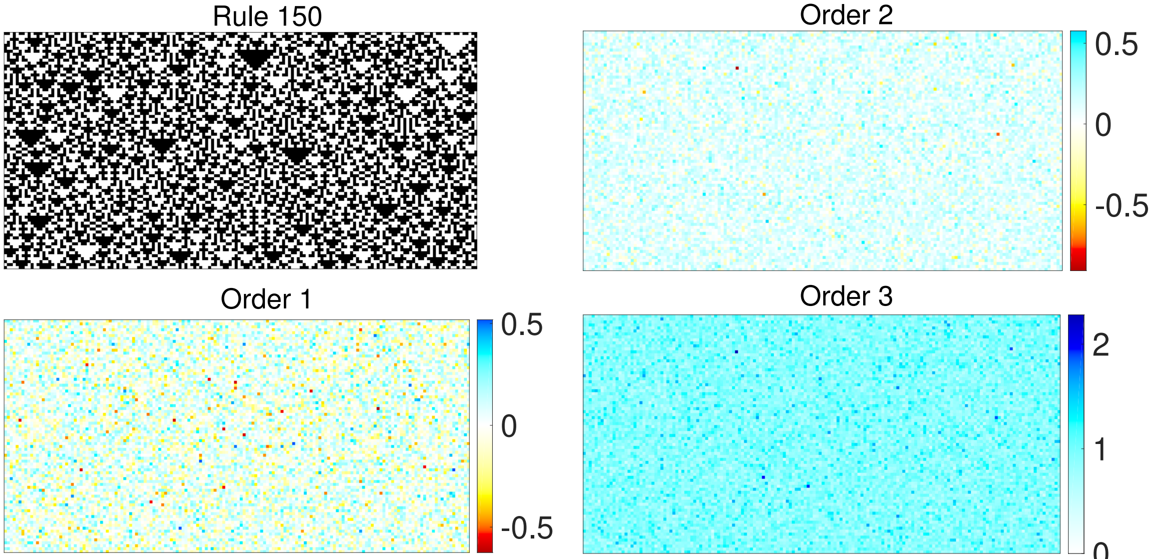
Joseph T. Lizier, Mikhail Prokopenko, and Albert Y. Zomaya. Information modification and particle collisions in distributed computation. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(3):037109, 2010. doi: 10.1063/1.3486801. URL `https://doi.org/10.1063/1.3486801`.

Joseph T Lizier, Mikhail Prokopenko, and Albert Y Zomaya. Local measures of information storage in complex distributed computation. *Information Sciences*, 208:39–54, 2012.

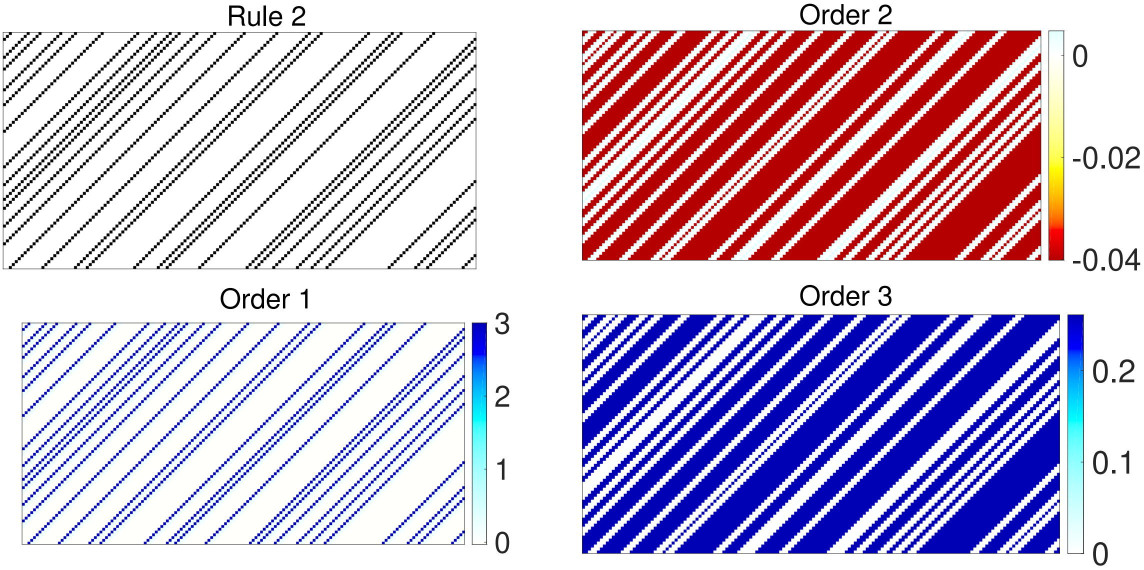
Joseph T Lizier, Benjamin Flecker, and Paul L Williams. Towards a synergy-based approach to measuring information modification. In *Artificial Life (ALIFE), 2013 IEEE Symposium on*, pages 43–51. IEEE, 2013.

Paul L Williams and Randall D Beer. Nonnegative decomposition of multivariate information. *arXiv preprint arXiv:1004.2515*, 2010.

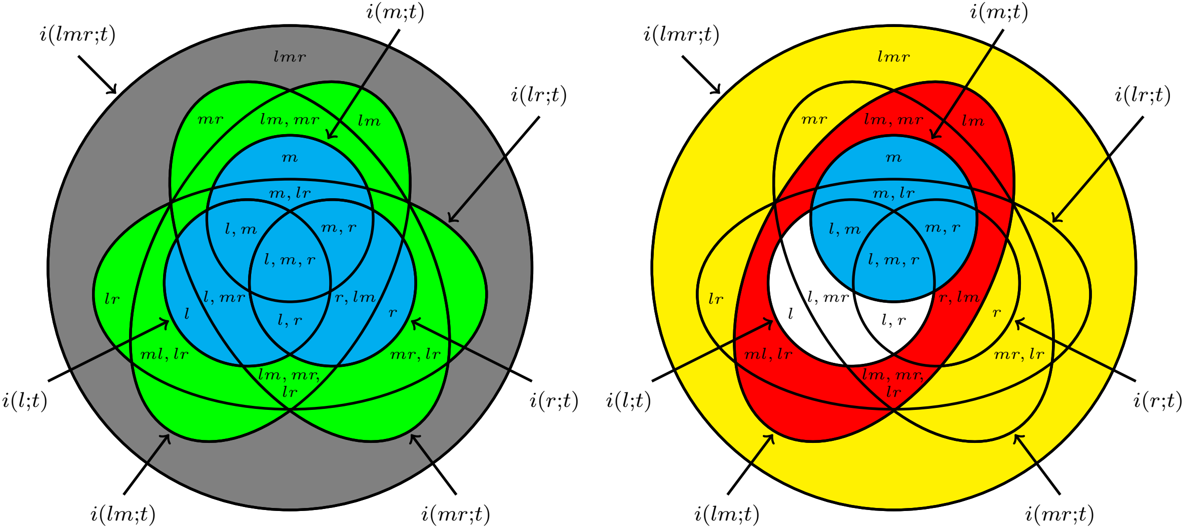

Rule 54
Order 2
4
2
0
Order 1
0
-2
-4
Order 3
4
2
0

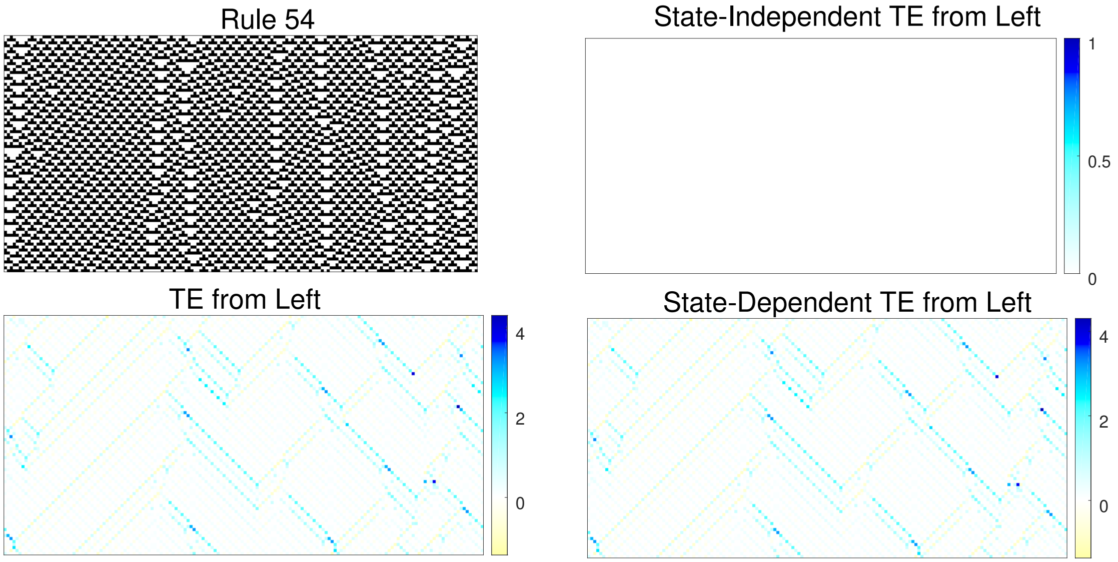


Rule 150
Order 2
0.5
0
-0.5
Order 1
0.5
0
-0.5
Order 3
2
1
0

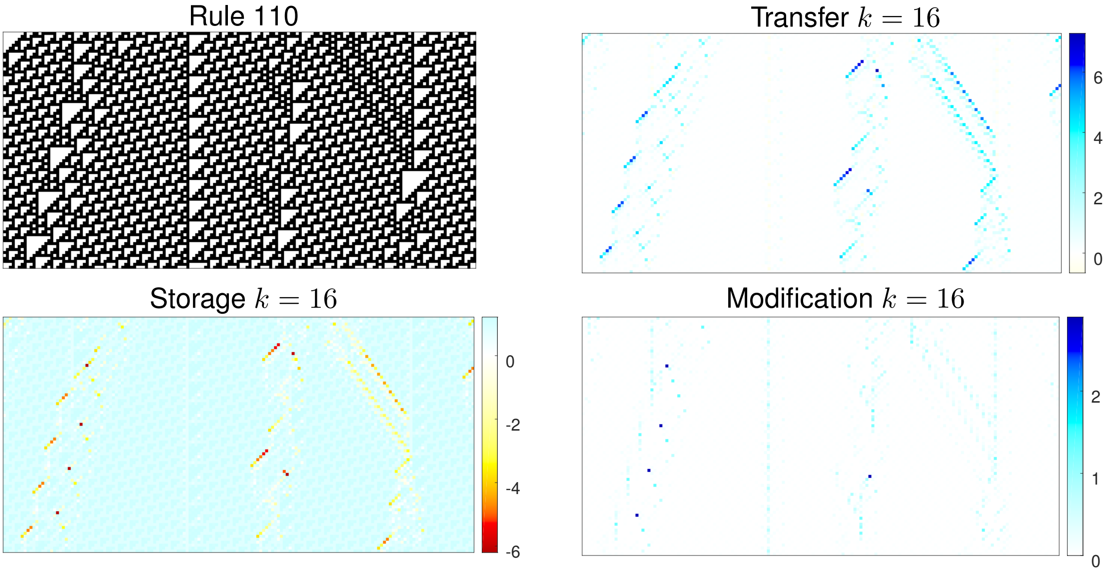


Rule 2
Order 2
0
-0.02
-0.04
Order 1
3
2
1
0
Order 3
0.2
0.1
0

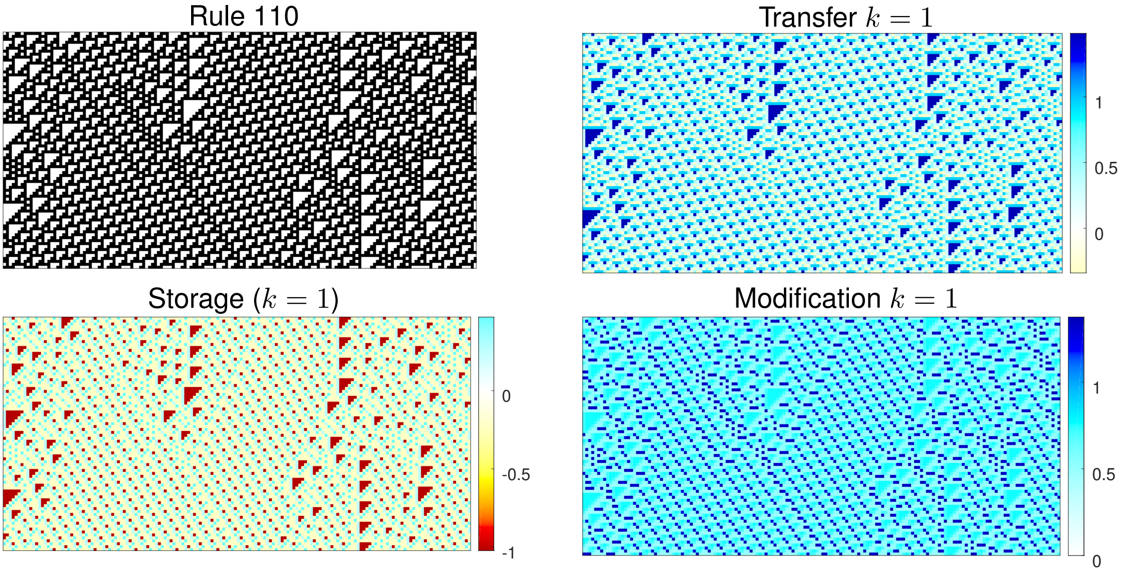

$i(lmr;t)$
$i(m;t)$
$i(lr;t)$
lmr
mr
lm, mr
lm
m
m, lr
l, m
m, r
l, m, r
lr
l, mr
r, lm
l
r
l, r
ml, lr
mr, lr
$lm, mr,$
lr
$i(l;t)$
$i(r;t)$
$i(lm;t)$
$i(mr;t)$
$i(lmr;t)$
$i(m;t)$
$i(lr;t)$
lmr
mr
lm, mr
lm
m
m, lr
l, m
m, r
l, m, r
lr
l, mr
r, lm
l
r
l, r
ml, lr
mr, lr
$lm, mr,$
lr
$i(l;t)$
$i(r;t)$
$i(lm;t)$
$i(mr;t)$



Rule 54
State-Independent TE from Left
1
0.5
0
TE from Left
4
2
0
State-Dependent TE from Left
4
2
0



Rule 110
Transfer $k = 16$
6
4
2
0
Storage $k = 16$
0
-2
-4
-6
Modification $k = 16$
2
1
0


Rule 110
Transfer $k = 1$
1
0.5
0
Storage ($k = 1$)
0
-0.5
-1
Modification $k = 1$
1
0.5
0