Spatiotemporal Computational Mechanics

Paul Riechers

Outline

- How do Information-theoretic and CompMech-theoretic quantities generalize in spatially extended systems?
  - An overview of the strictly-spatial case
- How spatial Information evolves in time
  - Towards a Computational Mechanics for information processing
- The big picture: How space and time are intricately connected, spacetime probability densities, and playing outside of your light-cone.

Rethinking the Basics for Spatially Extended Systems:

Block Entropies

$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L)\log_2 P(s^L)$$

- What does this mean?
  - What is the appropriate L for a system?
  - What is s^L
  - What is the alphabet?
  - How are the probabilities calculated?

Rethinking the Basics for Spatially Extended Systems:

Block Entropies

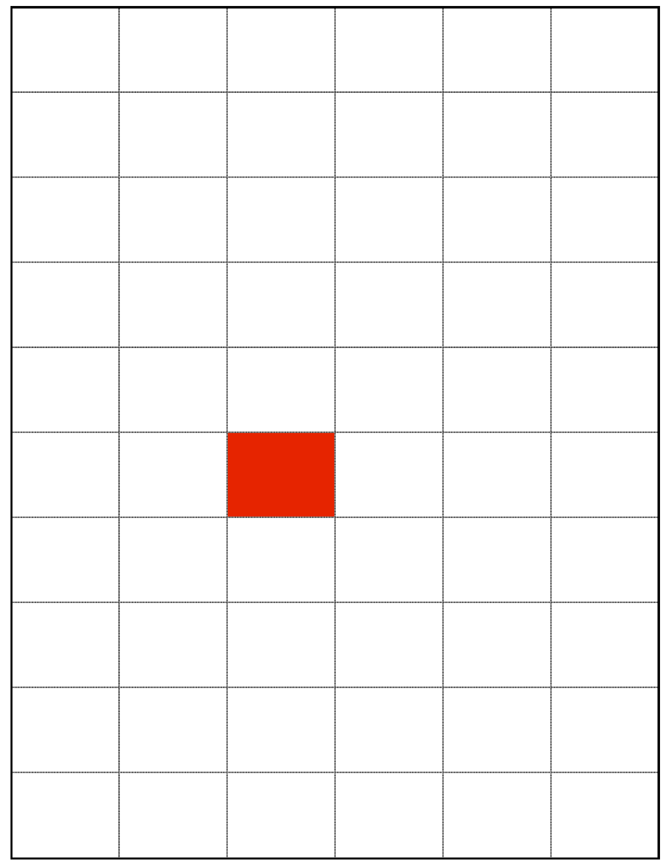
$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L)\log_2 P(s^L)$$

- In 1D… options
  - Consider the case of general dimensions (or at least 2D…)
    - What we should do for the 1D case will follow out as a special case

Rethinking the Basics for Spatially Extended Systems:

Block Entropies: H(L) for L = 1

- L = 1 considers the value lone cells

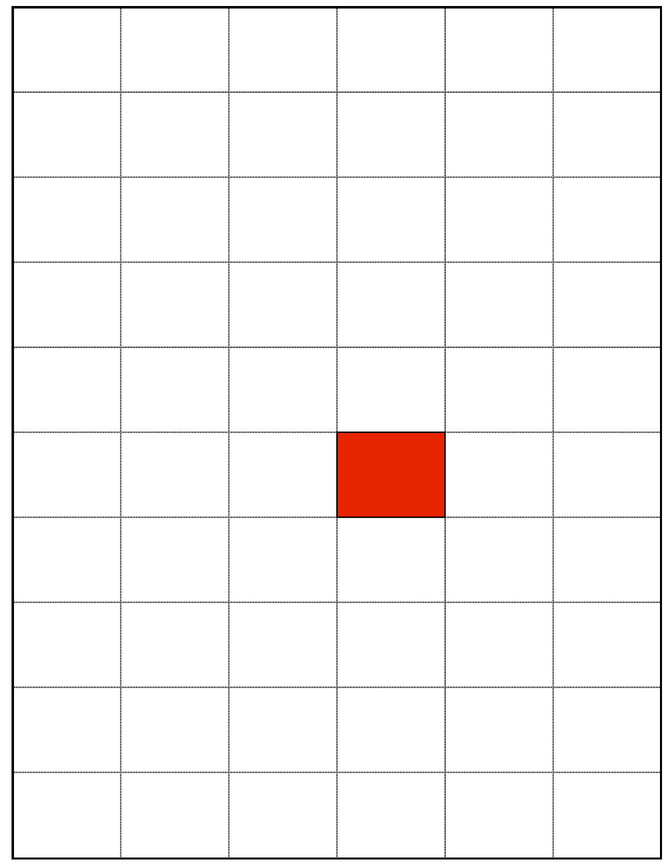


$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L) \log_2 P(s^L)$$

Rethinking the Basics for Spatially Extended Systems:

Block Entropies: H(L) for L = 1

- L = 1 considers the value lone cells
- But P(s) is taken from the statistics of the whole array
    - So, scan this template across the whole array to find P(s)!

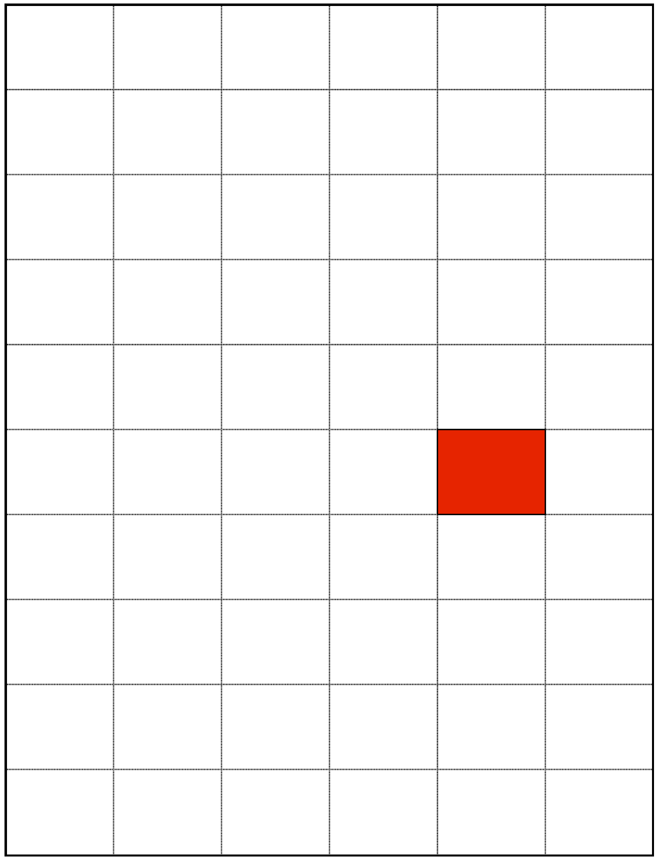


$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L)\log_2 P(s^L)$$

Rethinking the Basics for Spatially Extended Systems:

Block Entropies: H(L) for L = 1

- L = 1 considers the value lone cells
- But P(s) is taken from the statistics of the whole array
    - So, scan this template across the whole array to find P(s)!

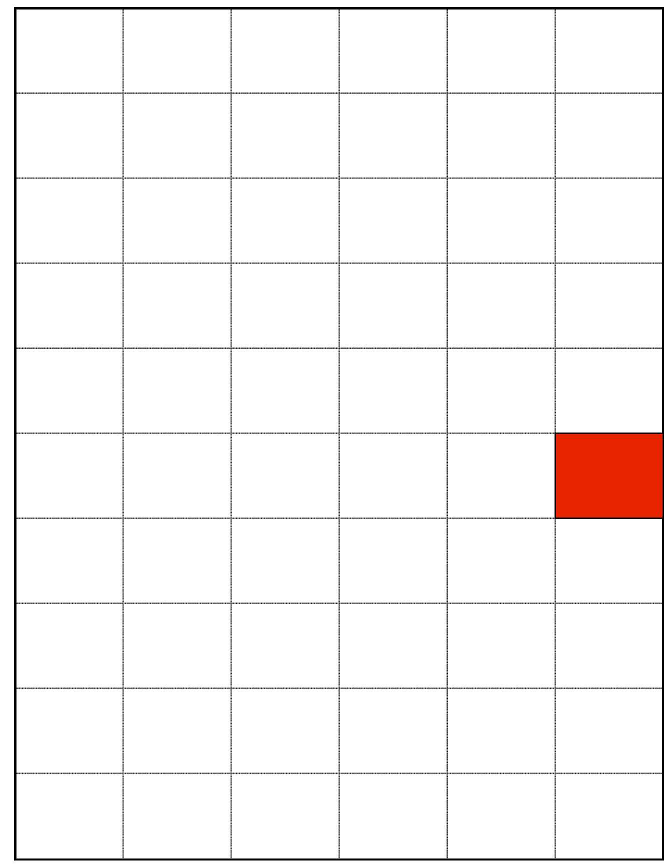


$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L)\log_2 P(s^L)$$

Rethinking the Basics for Spatially Extended Systems:

Block Entropies: H(L) for L = 1

- L = 1 considers the value lone cells
- But P(s) is taken from the statistics of the whole array
    - So, scan this template across the whole array to find P(s)!

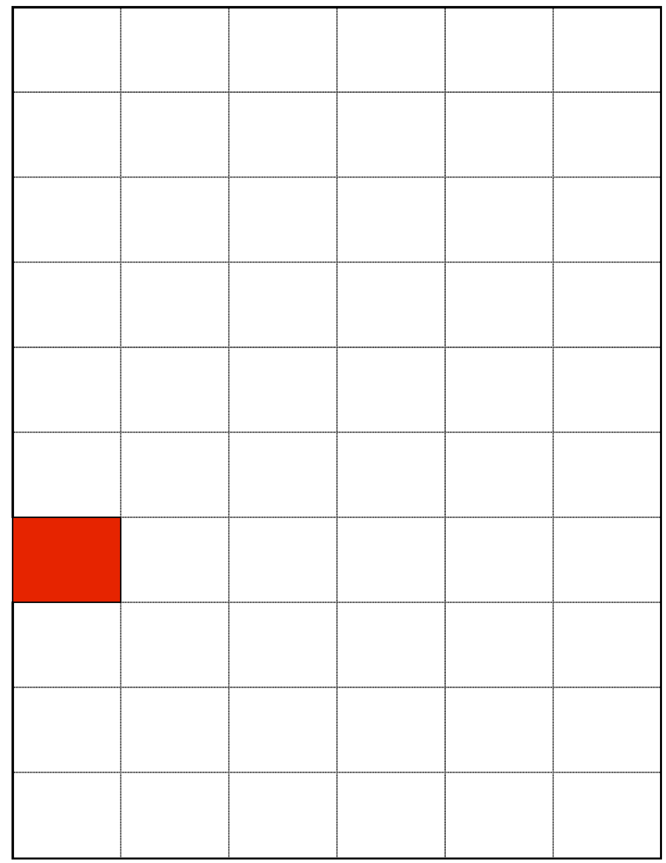


$$H(L) \quad = \quad H(P(s^L)) \quad = \quad -\sum_{s^L \in A} P(s^L) \log_2 P(s^L)$$

Rethinking the Basics for Spatially Extended Systems:

Block Entropies: H(L) for L = 1

- L = 1 considers the value lone cells
- But P(s) is taken from the statistics of the whole array
    - So, scan this template across the whole array to find P(s)!

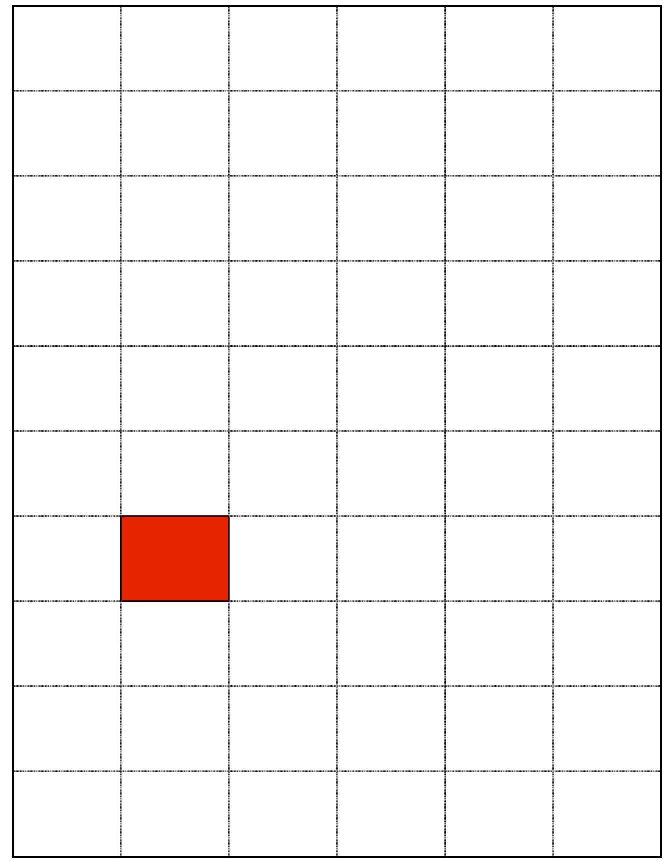


$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L)\log_2 P(s^L)$$

Rethinking the Basics for Spatially Extended Systems:

Block Entropies: H(L) for L = 1

- L = 1 considers the value lone cells
- But P(s) is taken from the statistics of the whole array
    - So, scan this template across the whole array to find P(s)!

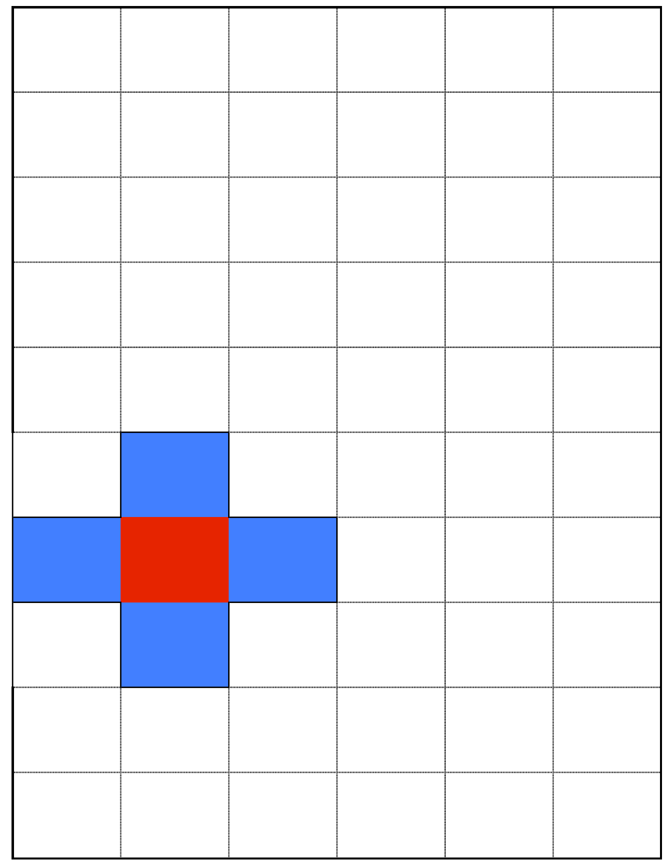


$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L)\log_2 P(s^L)$$

Rethinking the Basics for Spatially Extended Systems:

Block Entropies: H(L) for L = 2

- L = 2 template depends on geometry of coupling (assumed or explicit)
- The case shown is for four nearest neighbors

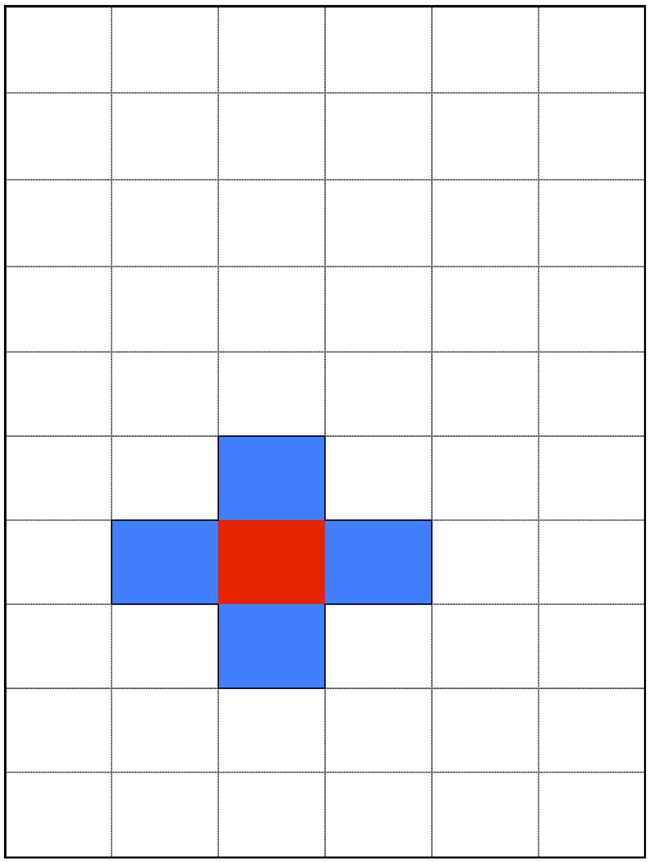


$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L) \log_2 P(s^L)$$

Rethinking the Basics for Spatially Extended Systems:

Block Entropies: H(L) for L = 2

- L = 2 template depends on geometry of coupling (assumed or explicit)
- But again, $P(s_1 s_2)$ is taken from the statistics of the whole array, so we scan this template across the whole array to find $P(s_1, s_2)$.
- Notice that s_i is no longer binary!



$s_1 \in \{0,1\}$ but $s_2 \in \{0,1,2,\ldots,2^4 - 1\}$

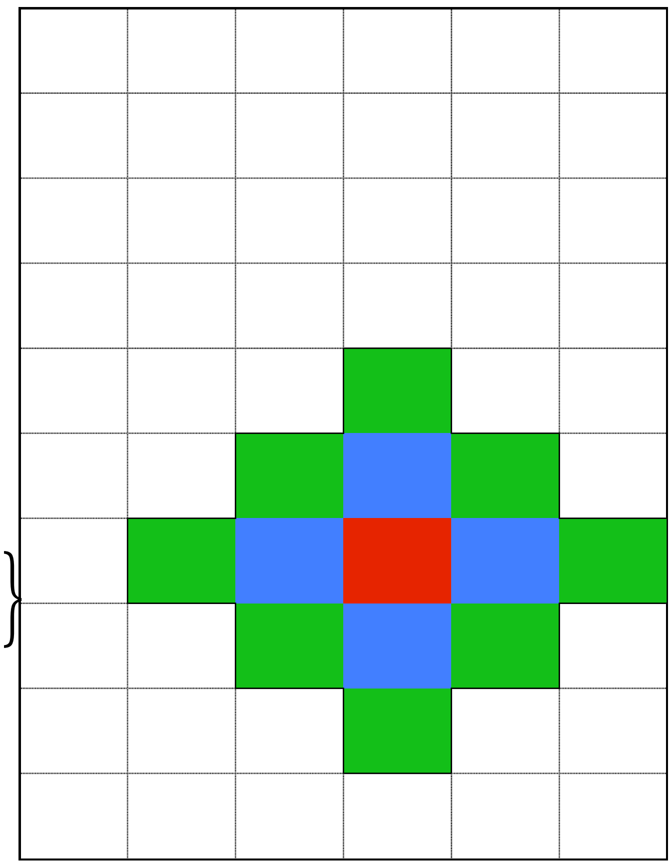
$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L) \log_2 P(s^L)$$

Rethinking the Basics for Spatially Extended Systems:

Block Entropies: H(L) for L = 3

- L = 3 template expands naturally from previous geometry
- We scan this template across the whole array to find $P(s_1,s_2,s_3)$.
- We have a different alphabet for different L. In general:

$$length\{A(L_{L^1-coupling})\} \quad = \quad \prod_{k=0}^{L-1}\left\{n^{(\delta_{k,0}+2^D k^{(D-1)})}\right\}$$

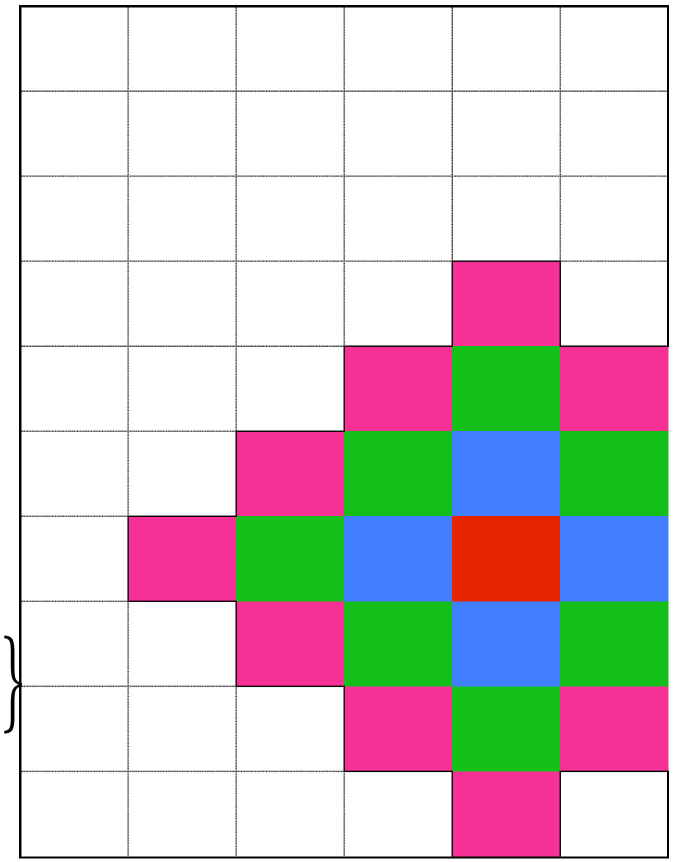


for number of intrinsic states per cell, n, and coupling number (number of nearest neighbors), $N_{C.}$

$$H(L) \quad = \quad H(P(s^L)) \quad = \quad -\sum_{s^L \in A} P(s^L)\log_2 P(s^L)$$

Rethinking the Basics for Spatially Extended Systems:

Block Entropies: H(L) for L = 4



- L = 4 template expands naturally from previous geometry
- We scan this template across the whole array to find $P(s_1,s_2,s_3, s_4)$.
- We have a different alphabet for different L:

$$length\{A(L_{L^1-coupling})\} = \prod_{k=0}^{L-1}\left\{n^{(\delta_{k,0}+2^D k^{(D-1)})}\right\}$$

for number of intrinsic states per cell, n, and coupling number (number of nearest neighbors), $N_{C.}$

$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L)\log_2 P(s^L)$$

Rethinking the Basics for Spatially Extended Systems:

Block Entropies: H(L) for any L

• We scan every L-template across the whole array to find $P(s^L)$.

•Special considerations for boundary conditions?

$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L) \log_2 P(s^L)$$

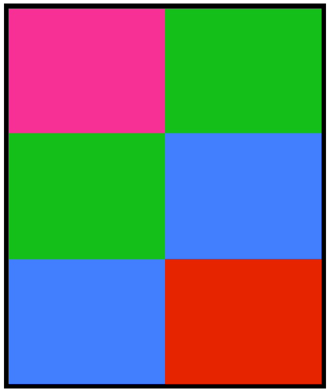
Rethinking the Basics for Spatially Extended Systems:

Block Entropies: H(L) for any L

- We scan every L-template across the whole array to find $P(s^L)$.

$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L)\log_2 P(s^L)$$

- Special considerations for boundary conditions?
  - H(L) saturates at L = n+m-1 for n × m array



e.g.
3 × 2 array:

All cells are seen by every other cell at L = 4.

$$h_\mu = \lim_{L\to\infty}\{h_\mu(L)\} = \lim_{L\to\infty}\{H(L)-H(L-1)\} = \lim_{L\to\infty}\{H_{saturation} - H_{saturation}\} = 0 \quad ?$$

$$H(L_{saturation}) = E \quad ?$$

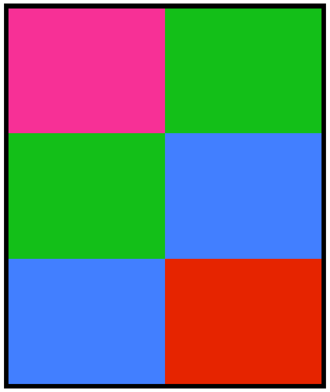
Rethinking the Basics for Spatially Extended Systems:

Block Entropies: H(L) for any L

- We scan every L-template across the whole array to find $P(s^L)$.

$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L)\log_2 P(s^L)$$

- Special considerations for boundary conditions?
  - H(L) saturates at L = n+m-1 for n × m array



e.g.
3 × 2 array:

All cells are seen by every other cell at L = 4.

$$h_\mu = \lim_{L\to\infty}\{h_\mu(L)\} = \lim_{L\to\infty}\{H(L)-H(L-1)\} = \lim_{L\to\infty}\{H_{saturation} - H_{saturation}\} = 0 \quad ? \qquad H(L_{saturation}) = E \quad ?$$

Instead, define new quantities:

$$h_\mu^{sat} \equiv h_\mu(L_{sat}) = H(L_{sat}) - H(L_{sat}-1) \quad ? \qquad E^{sat} \equiv ?$$

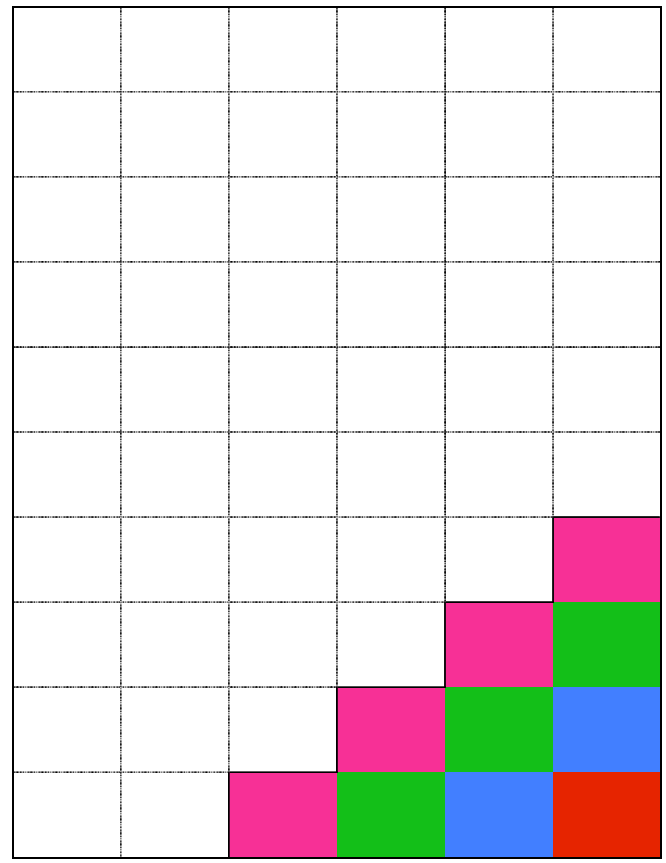
Rethinking the Basics for Spatially Extended Systems:

Block Entropies: H(L) for any L

- We scan every L-template across the whole array to find $P(s^L)$.

- Special considerations for boundary conditions?
  - H(L) saturates at L = n+m-1 for n x m array

  - Just average over possibilities?

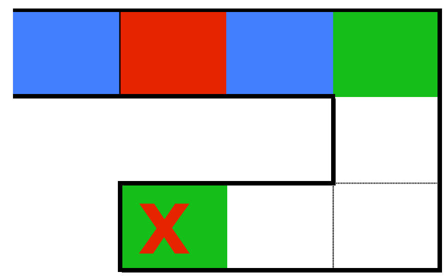


$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L) \log_2 P(s^L)$$

Rethinking the Basics for Spatially Extended Systems:

Block Entropies: H(L)

- Special considerations for boundary conditions?
  - Just average over possibilities? Maybe. *Although care must be given to pathological topologies…*

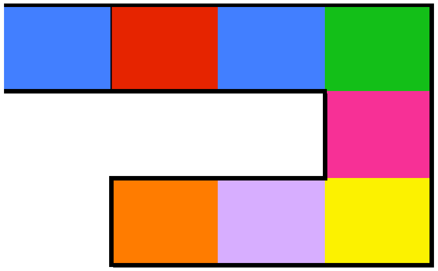


$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L)\log_2 P(s^L)$$

Rethinking the Basics for Spatially Extended Systems:

Block Entropies: H(L)

- Special considerations for boundary conditions?
  - Just average over possibilities? Maybe. Although care must be given to pathological topologies…

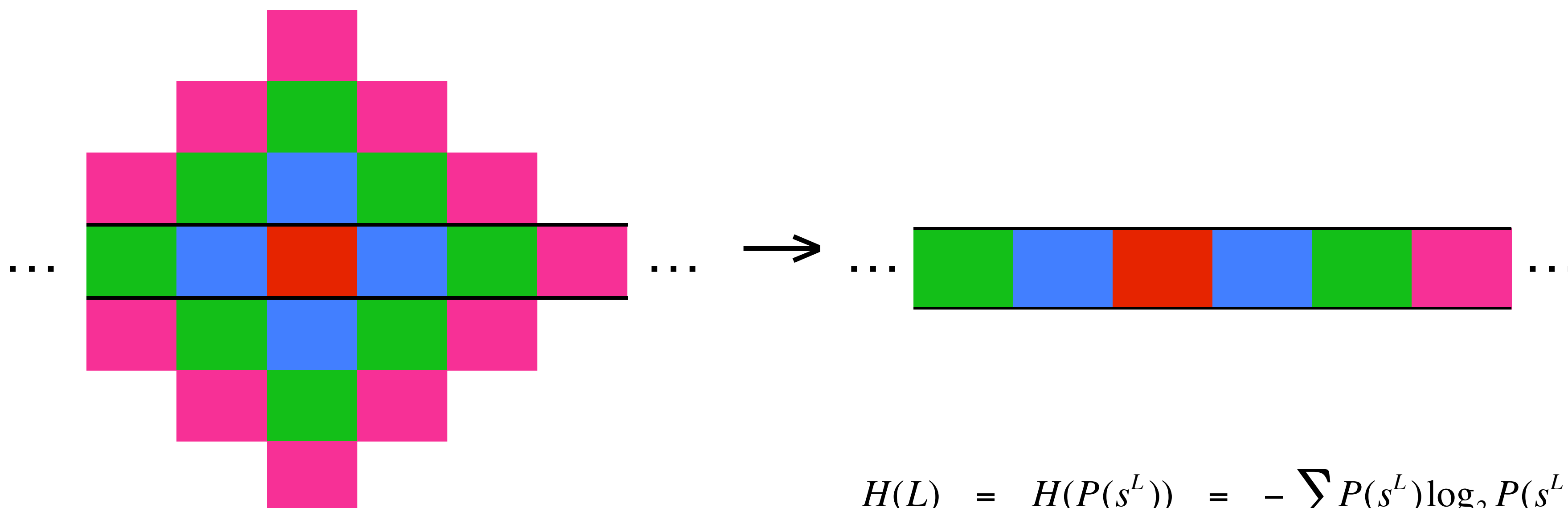


$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L)\log_2 P(s^L)$$

Rethinking the Basics for Spatially Extended Systems:

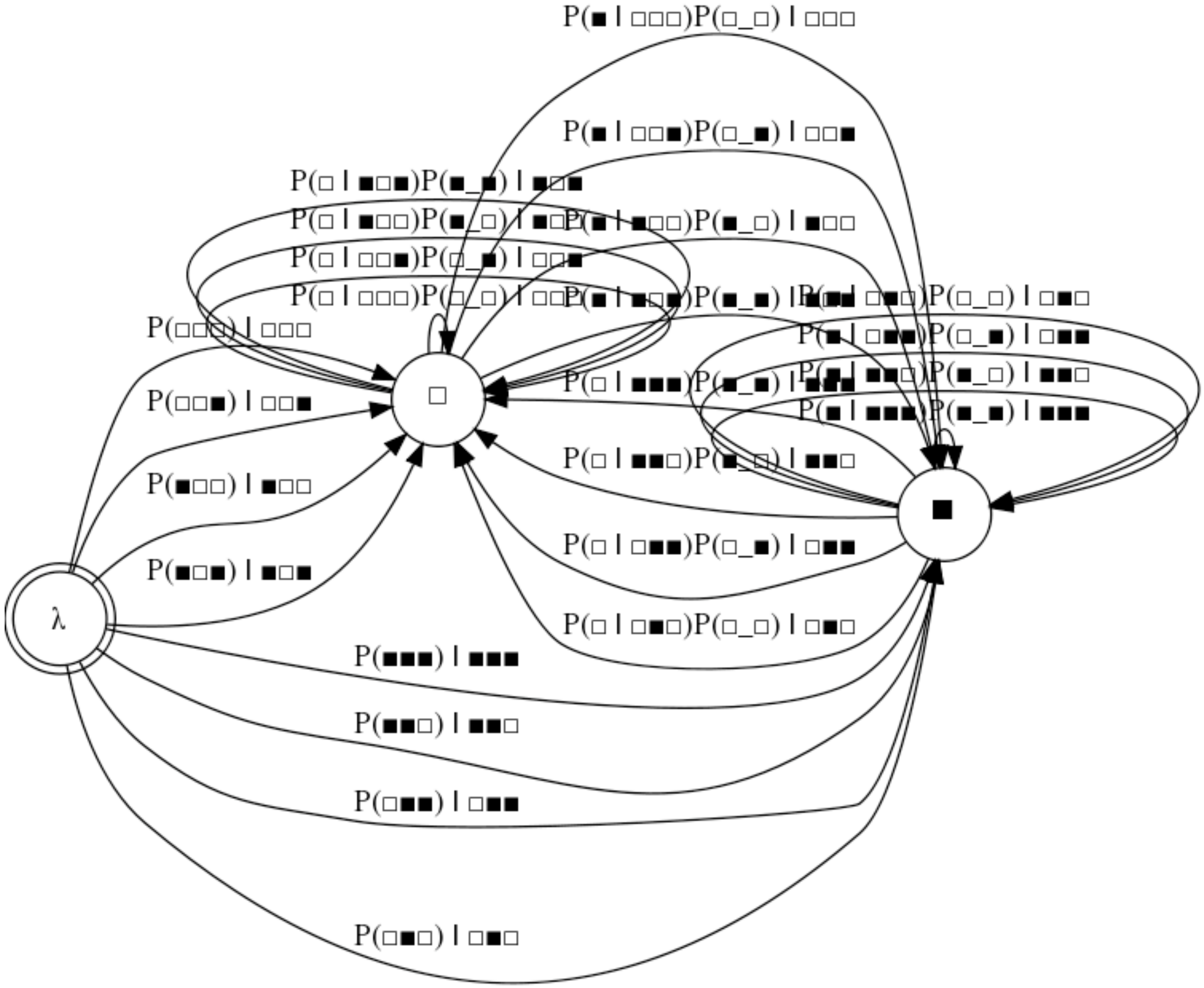
Block Entropies: H(L) for 1-D

- 1-Dimension as boundary condition
- We scan the natural L-templates across the whole linear array to find $P(s^L)$.

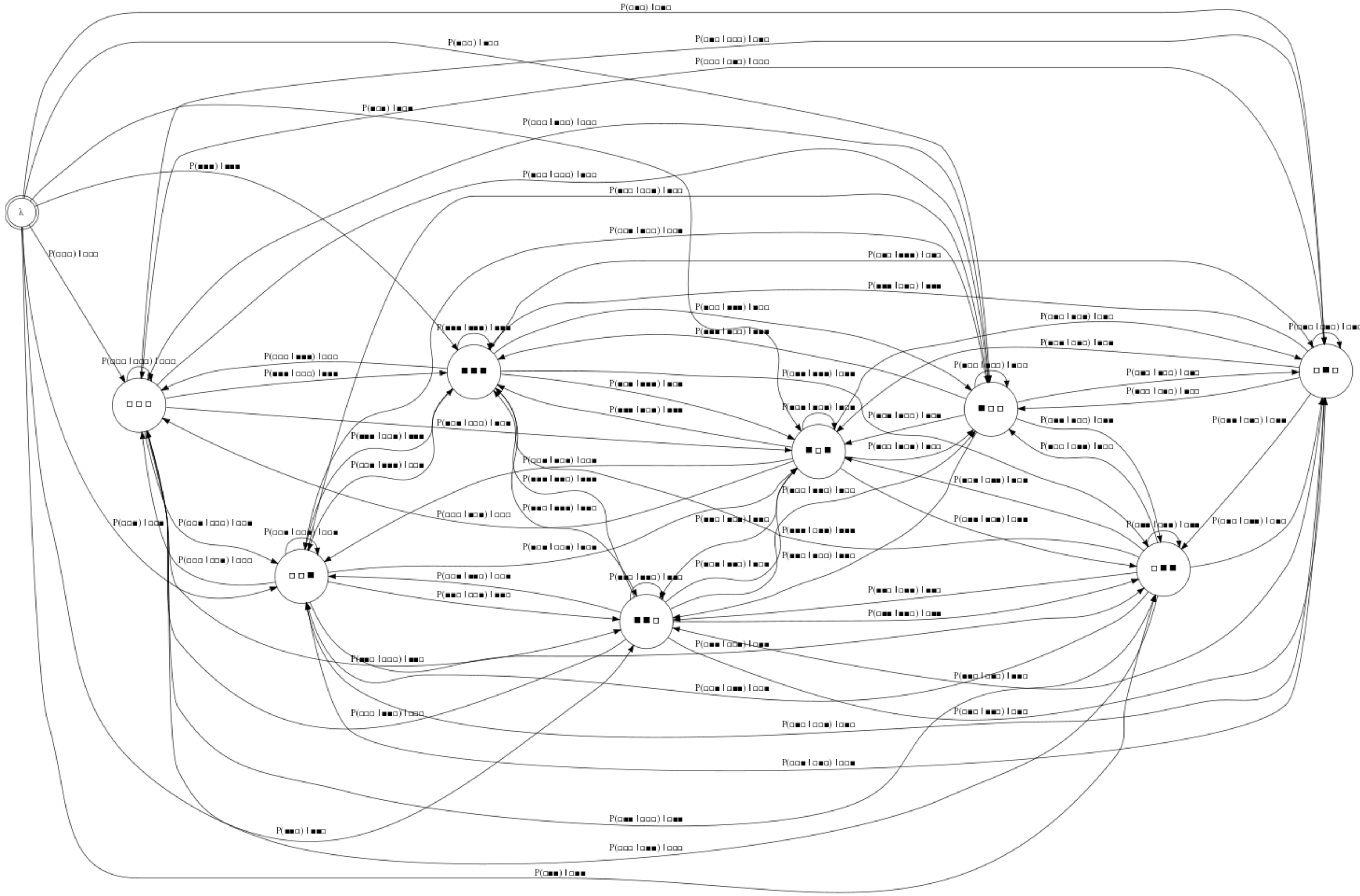


$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L)\log_2 P(s^L)$$

Rethinking Temporal Measurements for Spatially Extended States: 1-D nearest neighbor CA



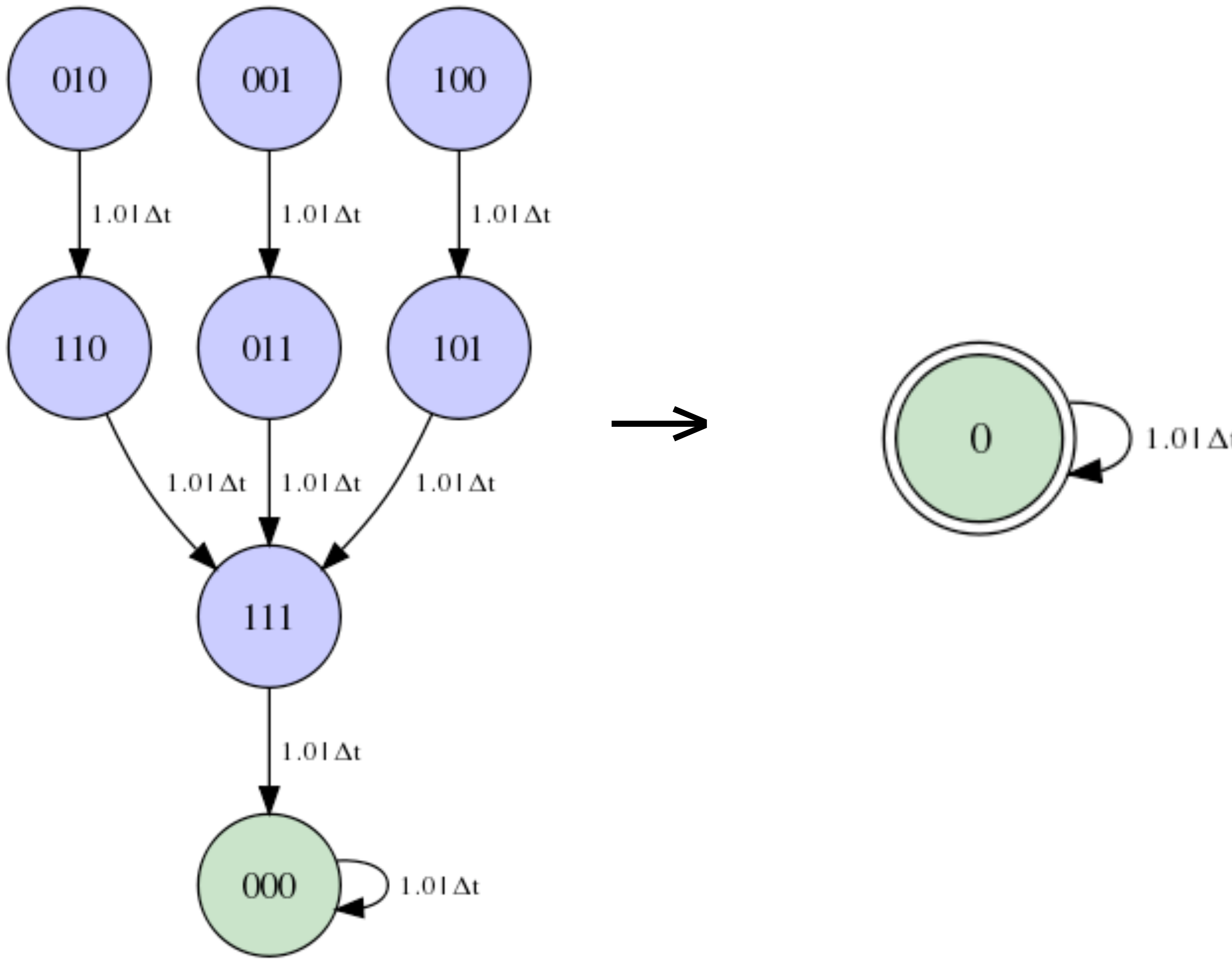
Rethinking Temporal Measurements for Spatially Extended States:
1-D nearest neighbor CA



Rethinking Temporal Measurements for Spatially Extended States:

1-D nearest neighbor CA

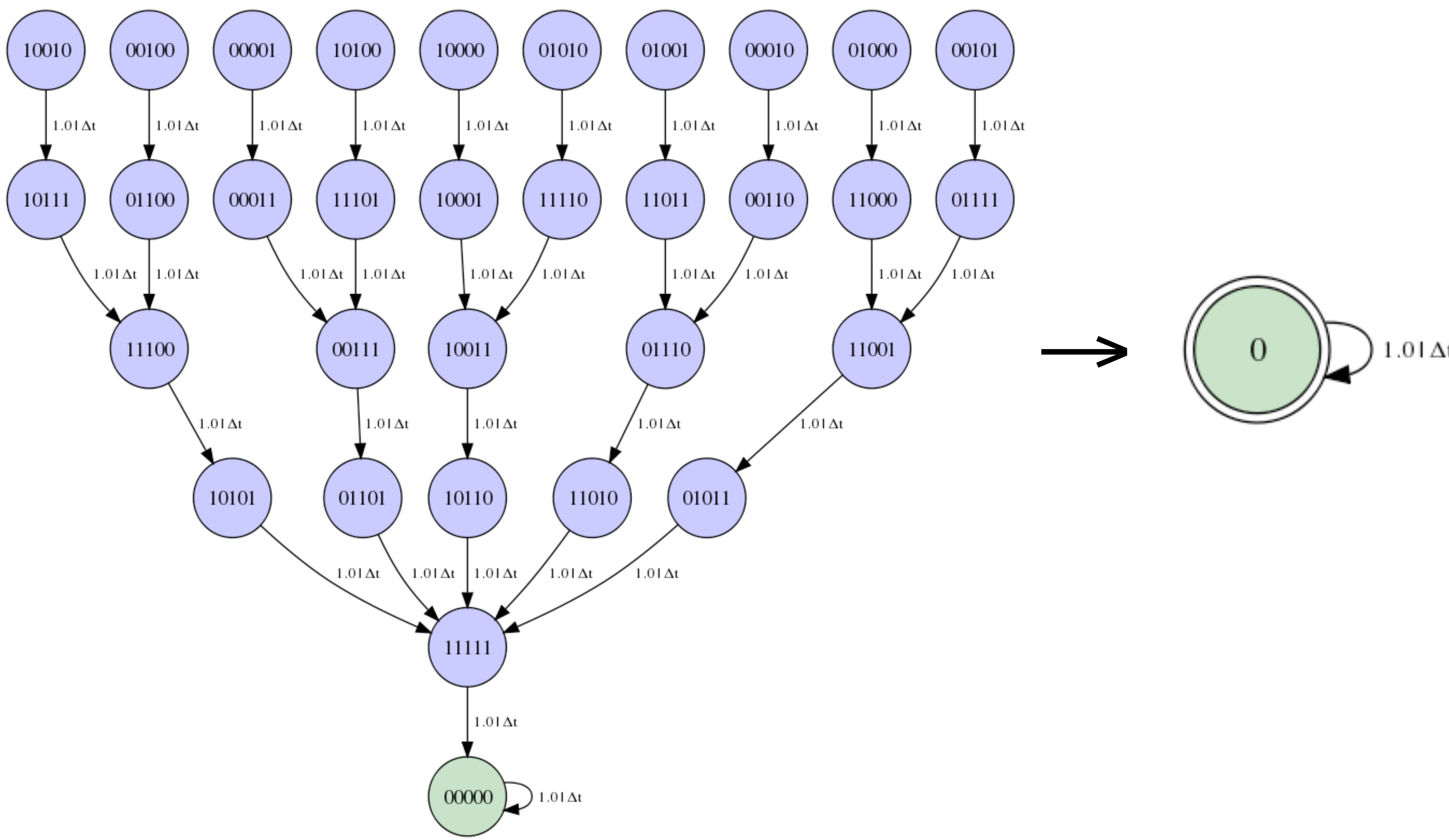
Starting Small: Rule 110 with only three cells in the array



Rethinking Temporal Measurements for Spatially Extended States:

1-D nearest neighbor CA

Starting Small: Rule 110 with only five cells in the array



Rethinking Temporal Measurements for Spatially Extended States:

1-D nearest neighbor CA

Starting Small: Rule 110 with only ten cells in the array

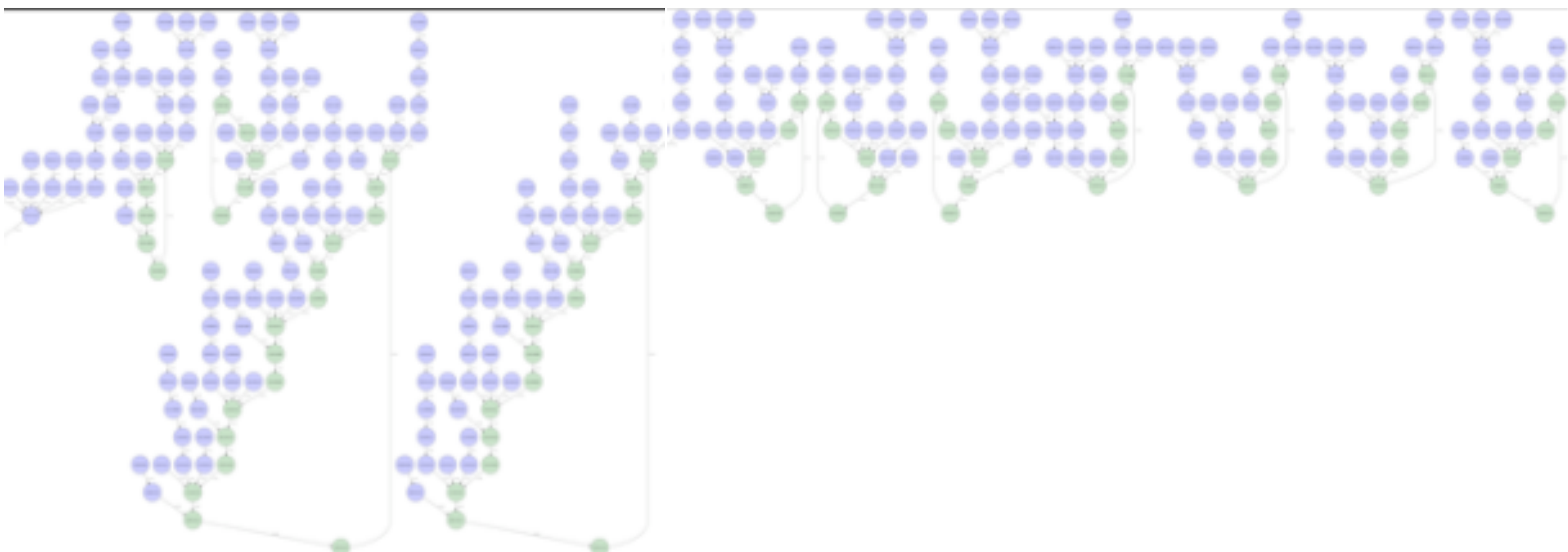
…

Rethinking Temporal Measurements for Spatially Extended States:

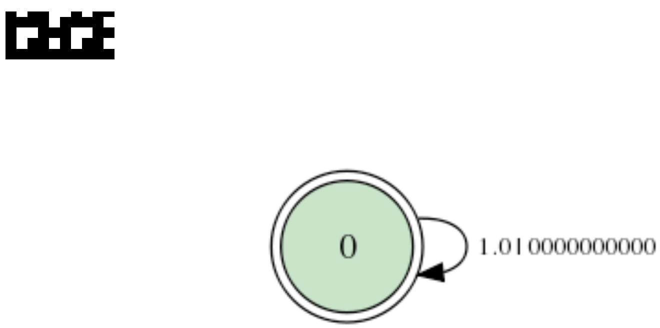
1-D nearest neighbor CA

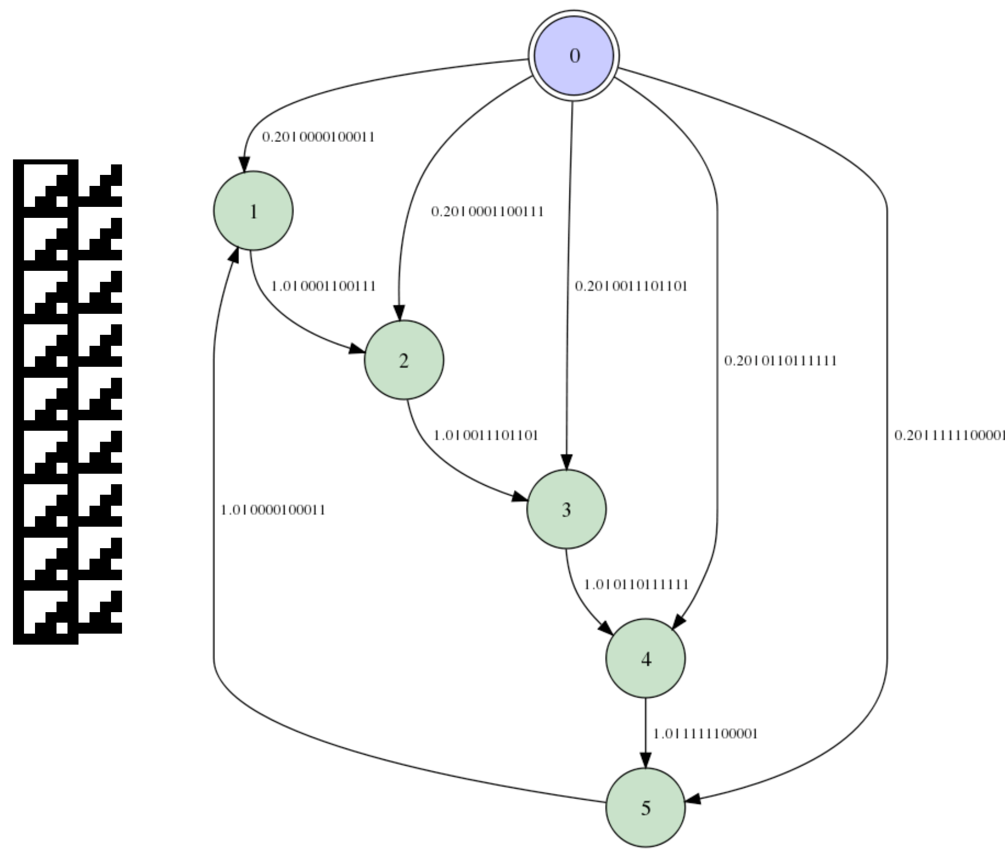
Starting Small: Rule 110 with only ten cells in the array

. . .



Rethinking Temporal Measurements for Spatially Extended States:
1-D nearest neighbor CA
Starting Small: Rule 110 with only ten cells in the array

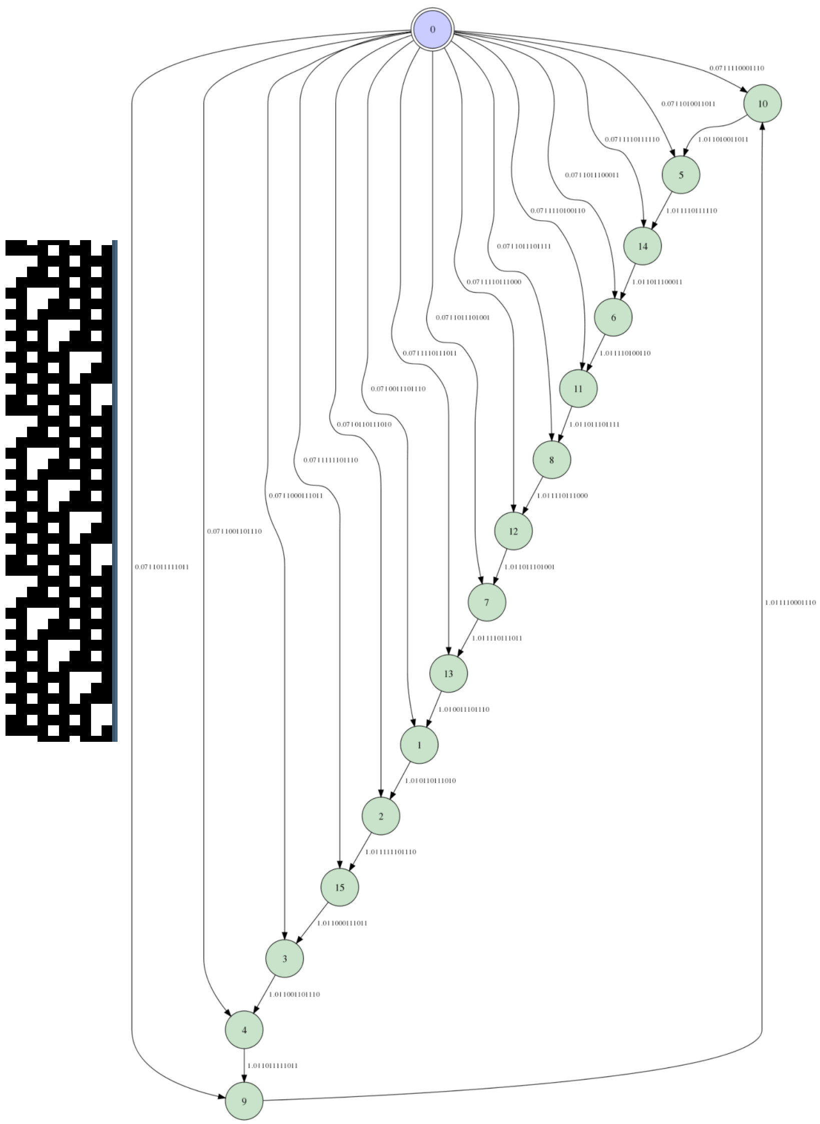


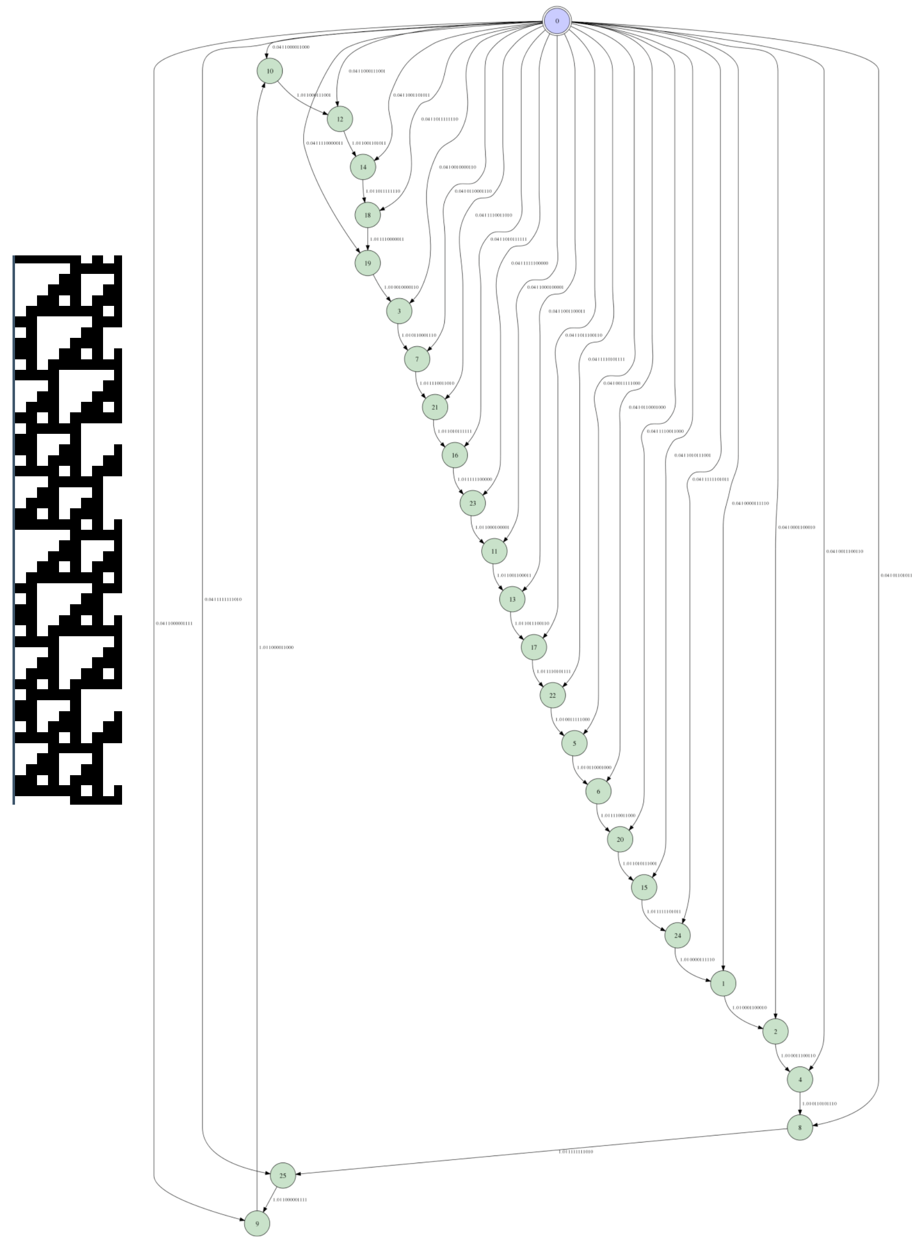


Rethinking Temporal Measurements for Spatially Extended States:

1-D nearest neighbor CA

Starting Small: Rule 110 with only ten cells in the array



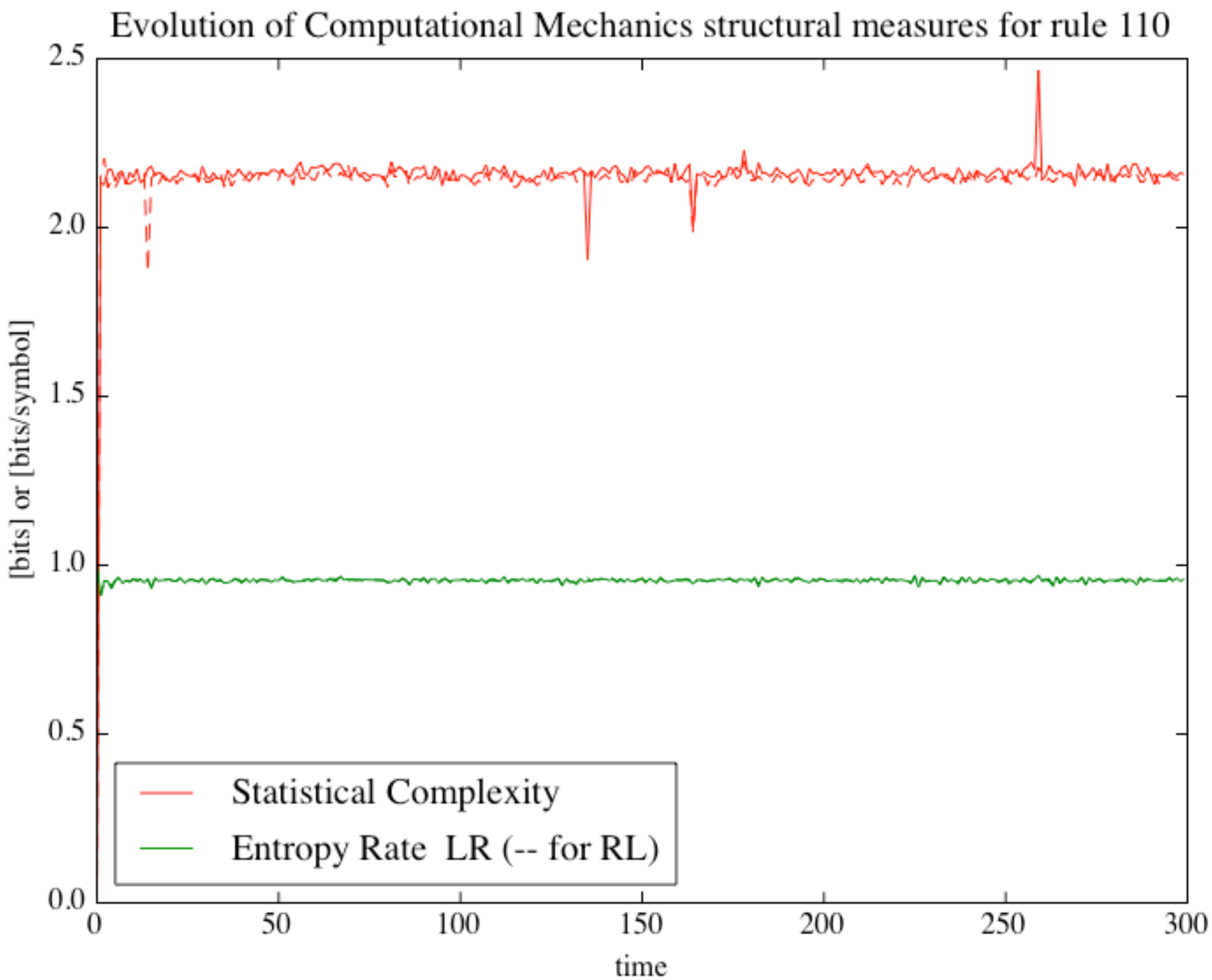


Rethinking Temporal Measurements for Spatially Extended States:

1-D nearest neighbor CA

Looking Ahead: Rule 110

Impulse excitation for 3000 sites: CompMech quantities from a timeseries

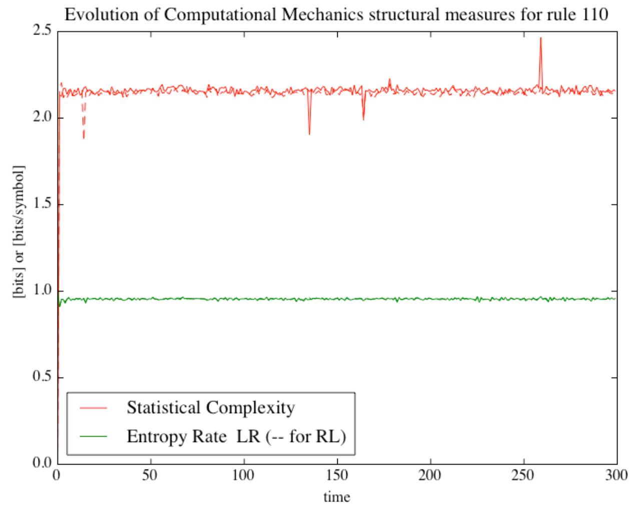


Rethinking Temporal Measurements for Spatially Extended States:
1-D nearest neighbor CA
Looking Ahead: Rule 110
Impulse excitation for 3000 sites: CompMech quantities from a timeseries

- The current methods/ algorithms are limited… working on that

- There are many new questions to address

- Space and time probabilities are intricately connected



Rethinking Temporal Measurements for Spatially Extended States:
1-D nearest neighbor CA

- Some inspired new measures:
  - Effective coupling length
    - Helps to digitize rules of dynamic systems
  - Exploring chiral rules