Computing the entropy of two-dimensional shifts of finite type

Brian Marcus

University of British Columbia
www.math.ubc.ca/∼marcus

November 11, 2009, Texas A&M University

Collaborators

Erez Louidor (PhD student):

Collaborators

Erez Louidor (PhD student):

Collaborators

Erez Louidor (PhD student):

Ronnie Pavlov (Postdoctoral Fellow):

Collaborators

Erez Louidor (PhD student):

Ronnie Pavlov (Postdoctoral Fellow):

1-dimensional Shifts of finite type

- A **1-dimensional shift of finite type** (SFT) is defined by:

1-dimensional Shifts of finite type

- A **1-dimensional shift of finite type** (SFT) is defined by:
  - A finite **alphabet** A.

1-dimensional Shifts of finite type

- A **1-dimensional shift of finite type** (SFT) is defined by:
  - A finite **alphabet** A.
  - A finite set $\mathcal{F}$ of finite words,

1-dimensional Shifts of finite type

- A **1-dimensional shift of finite type** (SFT) is defined by:
  - A finite **alphabet** A.
  - A finite set $\mathcal{F}$ of finite words,
- The SFT X is the set of all elements of $A^{\mathbb{Z}}$ (bi-infinite sequences) which do not contain any of the words from $\mathcal{F}$.

1-dimensional Shifts of finite type

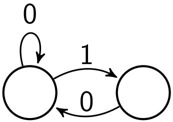
- A **1-dimensional shift of finite type** (SFT) is defined by:
  - A finite **alphabet** A.
  - A finite set $\mathcal{F}$ of finite words,
- The SFT X is the set of all elements of $A^{\mathbb{Z}}$ (bi-infinite sequences) which do not contain any of the words from $\mathcal{F}$.
- An SFT is a "constraint" on the set of allowable words.

Examples

- Example 1: the **golden mean shift,** $(G^{(1)})$, $A = \{0, 1\}$:

$$\mathcal{F} = \{11\}.$$

Typical allowed sequence: $\ldots 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0 \ldots$

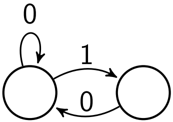


Examples

- Example 1: the **golden mean shift,** $(G^{(1)})$, $A = \{0, 1\}$:

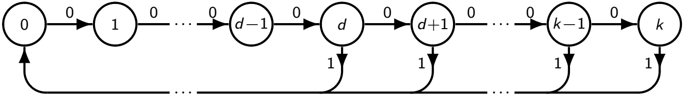
$$\mathcal{F} = \{11\}.$$

Typical allowed sequence: $\ldots 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0 \ldots$



- Example 2: the **run-length-limited shift** (RLL(d, k)), $A = \{0, 1\}$

$$\mathcal{F} = \{11, 101, 1001, \ldots, 10^{d-1}1, 0^{k+1}\}$$

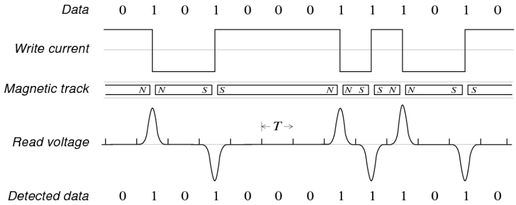


Motivation for 1-dimensional SFT's: Constraints on data sequences recorded in storage devices

- Magnetic recording:

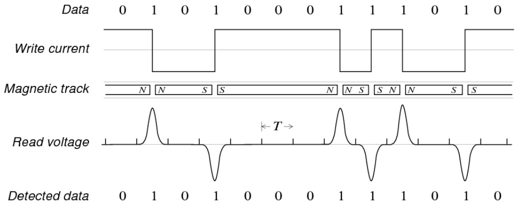
Motivation for 1-dimensional SFT's: Constraints on data sequences recorded in storage devices

- Magnetic recording:



Motivation for 1-dimensional SFT's: Constraints on data sequences recorded in storage devices

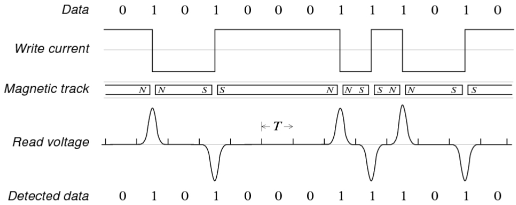
- Magnetic recording:



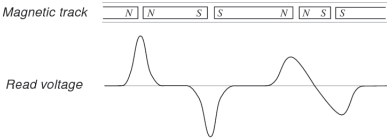
- Intersymbol interference:

Motivation for 1-dimensional SFT's: Constraints on data sequences recorded in storage devices

- Magnetic recording:

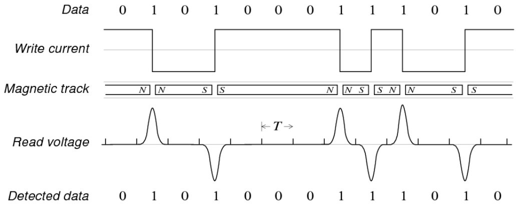


- Intersymbol interference:

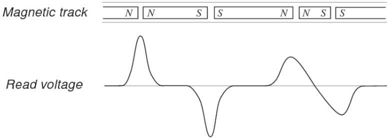


Motivation for 1-dimensional SFT's: Constraints on data sequences recorded in storage devices

- Magnetic recording:



- Intersymbol interference:



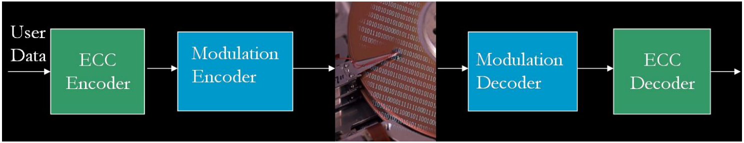
- Hence an RLL constraint on allowed stored sequences.

Encoding

- Modulation encoder: encodes arbitrary data sequences into X.

Encoding

- Modulation encoder: encodes arbitrary data sequences into X.



Topological entropy of 1-D SFT's (a.k.a. entropy, noiseless capacity)

- A word w is **admissible** if it contains no sub-word from $\mathcal{F}$.

Topological entropy of 1-D SFT's (a.k.a. entropy, noiseless capacity)

- A word w is **admissible** if it contains no sub-word from $\mathcal{F}$.
- Let $B_n(X)$ be the set of admissible words of length n.

Topological entropy of 1-D SFT's (a.k.a. entropy, noiseless capacity)

- A word w is **admissible** if it contains no sub-word from $\mathcal{F}$.
- Let $B_n(X)$ be the set of admissible words of length n.
- Define the **entropy**: $h(X) = \lim_{n\to\infty} \frac{\log |B_n(X)|}{n}$

Topological entropy of 1-D SFT's (a.k.a. entropy, noiseless capacity)

- A word w is **admissible** if it contains no sub-word from $\mathcal{F}$.
- Let $B_n(X)$ be the set of admissible words of length n.
- Define the **entropy**: $h(X) = \lim_{n\to\infty} \frac{\log |B_n(X)|}{n}$
- The entropy is the maximal rate of encoder from the set of all arbitrary data sequences into X.

Computation of entropy

- At the expense of enlarging the alphabet, we can assume that $\mathcal{F}$ consists of words of length 2, and so the SFT is defined by *nearest neighbours*.

Computation of entropy

- At the expense of enlarging the alphabet, we can assume that $\mathcal{F}$ consists of words of length 2, and so the SFT is defined by *nearest neighbours*.
- In this case, one constructs a 0-1 **transition matrix** M which determines the allowed neighbours and $h(X) = \log \lambda(M)$, where $\lambda(M)$ is the largest eigenvalue of M.

Computation of entropy

- At the expense of enlarging the alphabet, we can assume that $\mathcal{F}$ consists of words of length 2, and so the SFT is defined by *nearest neighbours*.
- In this case, one constructs a 0-1 **transition matrix** M which determines the allowed neighbours and $h(X) = \log \lambda(M)$, where $\lambda(M)$ is the largest eigenvalue of M.
- Example:

Computation of entropy

- At the expense of enlarging the alphabet, we can assume that $\mathcal{F}$ consists of words of length 2, and so the SFT is defined by *nearest neighbours*.
- In this case, one constructs a 0-1 **transition matrix** M which determines the allowed neighbours and $h(X) = \log \lambda(M)$, where $\lambda(M)$ is the largest eigenvalue of M.
- Example:
    - X: the golden mean shift,

Computation of entropy

- At the expense of enlarging the alphabet, we can assume that $\mathcal{F}$ consists of words of length 2, and so the SFT is defined by *nearest neighbours*.
- In this case, one constructs a 0-1 **transition matrix** M which determines the allowed neighbours and $h(X) = \log \lambda(M)$, where $\lambda(M)$ is the largest eigenvalue of M.
- Example:
  - X: the golden mean shift,
  - $M = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, $\lambda = \frac{1+\sqrt{5}}{2}$, and $h(X) = \log \frac{1+\sqrt{5}}{2} \approx .69$.

Computation of entropy

- At the expense of enlarging the alphabet, we can assume that $\mathcal{F}$ consists of words of length 2, and so the SFT is defined by *nearest neighbours*.
- In this case, one constructs a 0-1 **transition matrix** M which determines the allowed neighbours and $h(X) = \log \lambda(M)$, where $\lambda(M)$ is the largest eigenvalue of M.
- Example:
  - X: the golden mean shift,
  - $M = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, $\lambda = \frac{1+\sqrt{5}}{2}$, and $h(X) = \log \frac{1+\sqrt{5}}{2} \approx .69$.
- So, we can compute entropies of 1-dimensional SFT's.

Computation of entropy

- At the expense of enlarging the alphabet, we can assume that $\mathcal{F}$ consists of words of length 2, and so the SFT is defined by *nearest neighbours*.
- In this case, one constructs a 0-1 **transition matrix** M which determines the allowed neighbours and $h(X) = \log \lambda(M)$, where $\lambda(M)$ is the largest eigenvalue of M.
- Example:
  - X: the golden mean shift,
  - $M = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, $\lambda = \frac{1+\sqrt{5}}{2}$, and $h(X) = \log \frac{1+\sqrt{5}}{2} \approx .69$.
- So, we can compute entropies of 1-dimensional SFT's.
- And we can characterize the set of numbers that occur as entropies of 1-dimensional SFT's:

Computation of entropy

- At the expense of enlarging the alphabet, we can assume that $\mathcal{F}$ consists of words of length 2, and so the SFT is defined by *nearest neighbours*.
- In this case, one constructs a 0-1 **transition matrix** M which determines the allowed neighbours and $h(X) = \log \lambda(M)$, where $\lambda(M)$ is the largest eigenvalue of M.
- Example:
  - X: the golden mean shift,
  - $M = \left[\begin{smallmatrix}1&1\\1&0\end{smallmatrix}\right]$, $\lambda = \frac{1+\sqrt{5}}{2}$, and $h(X) = \log \frac{1+\sqrt{5}}{2} \approx .69$.
- So, we can compute entropies of 1-dimensional SFT's.
- And we can characterize the set of numbers that occur as entropies of 1-dimensional SFT's:
  - **Theorem** (Lind, 1983)): A number h is the entropy of a one-dimensional SFT if and only if h is the log of a root of a Perron number (special kind of algebraic integer).

2-dimensional Shifts of finite type

- A **2-dimensional shift of finite type** (SFT) is defined by:

2-dimensional Shifts of finite type

- A **2-dimensional shift of finite type** (SFT) is defined by:
  - A finite alphabet A.

2-dimensional Shifts of finite type

- A **2-dimensional shift of finite type** (SFT) is defined by:
  - A finite alphabet A.
  - A finite set $\mathcal{F}$ of finite patterns on rectangles.

2-dimensional Shifts of finite type

- A **2-dimensional shift of finite type** (SFT) is defined by:
  - A finite alphabet A.
  - A finite set $\mathcal{F}$ of finite patterns on rectangles.
- The SFT X is defined to be all elements of $A^{\mathbb{Z}^2}$ (i.e., configurations on the entire Z^2 lattice) which do not contain any of the words from $\mathcal{F}$.

2-dimensional Shifts of finite type

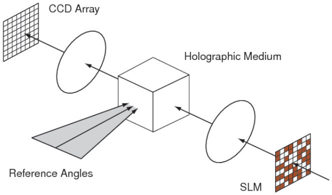
- A **2-dimensional shift of finite type** (SFT) is defined by:
  - A finite alphabet A.
  - A finite set $\mathcal{F}$ of finite patterns on rectangles.
- The SFT X is defined to be all elements of $A^{\mathbb{Z}^2}$ (i.e., configurations on the entire Z^2 lattice) which do not contain any of the words from $\mathcal{F}$.
- Example 1: the two-dimensional golden mean shift $G^{(2)}$: $A = \{0, 1\}$, $\mathcal{F} = \{\text{any pair of adjacent } 1's\} = \{\, 11\,, \begin{smallmatrix}1\\1\end{smallmatrix}\}$.

2-dimensional Shifts of finite type

- A **2-dimensional shift of finite type** (SFT) is defined by:
  - A finite alphabet A.
  - A finite set $\mathcal{F}$ of finite patterns on rectangles.
- The SFT X is defined to be all elements of $A^{\mathbb{Z}^2}$ (i.e., configurations on the entire Z^2 lattice) which do not contain any of the words from $\mathcal{F}$.
- Example 1: the two-dimensional golden mean shift $G^{(2)}$: $A = \{0, 1\}$, $\mathcal{F} = \{\text{any pair of adjacent } 1's\} = \{\, 11\,,\, \begin{smallmatrix}1\\1\end{smallmatrix} \}$.
- Typical allowed configuration:

$$
\begin{array}{cccccccccccccccc}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \cdot \\
\cdot & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & \cdot \\
\cdot & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & \cdot \\
\cdot & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{array}
$$

Motivation for 2-dimensional SFT's: Holographic storage



More examples of 2-dimensional SFT's

- NAK (Non-attacking kings): $\mathcal{F} = \left\{ 11,\ \begin{smallmatrix} 1 \\ 1 \end{smallmatrix},\ \begin{smallmatrix} 1 & \\ & 1 \end{smallmatrix},\ \begin{smallmatrix} & 1 \\ 1 & \end{smallmatrix} \right\}$.

More examples of 2-dimensional SFT's

- NAK (Non-attacking kings): $\mathcal{F} = \{ 11, \begin{smallmatrix} 1 \\ 1 \end{smallmatrix}, \begin{smallmatrix} 1 & \\ & 1 \end{smallmatrix}, \begin{smallmatrix} & 1 \\ 1 & \end{smallmatrix} \}$.
-

$$\begin{array}{cccccccccccccccc}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \cdot \\
\cdot & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot \\
\cdot & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & \cdot \\
\cdot & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{array}$$

More examples of 2-dimensional SFT's

- RWIM (Read/Write Isolated Memory): $\mathcal{F} = \{ 11, \begin{smallmatrix} 1 & \\ & 1 \end{smallmatrix}, \begin{smallmatrix} & 1 \\ 1 & \end{smallmatrix} \}$.

More examples of 2-dimensional SFT's

- RWIM (Read/Write Isolated Memory): $\mathcal{F} = \{ 11, \begin{smallmatrix} 1 & \\ & 1 \end{smallmatrix}, \begin{smallmatrix} & 1 \\ 1 & \end{smallmatrix} \}$.
-

$$
\begin{array}{cccccccccccccccc}
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & . \\
. & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & . \\
. & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & . \\
. & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & .
\end{array}
$$

Entropy of 2-dimensional SFT's

- A pattern w on a rectangle of any size is **admissible** if it contains no sub-pattern from $\mathcal{F}$.

Entropy of 2-dimensional SFT's

- A pattern w on a rectangle of any size is **admissible** if it contains no sub-pattern from $\mathcal{F}$.
- Let $B_{n\times n}(X)$ be the set of admissible patterns of size $n \times n$.

Entropy of 2-dimensional SFT's

- A pattern w on a rectangle of any size is **admissible** if it contains no sub-pattern from $\mathcal{F}$.
- Let $B_{n \times n}(X)$ be the set of admissible patterns of size $n \times n$.
- Define the **entropy** $h(X) = \lim_{n \to \infty} \frac{\log |B_{n \times n}(X)|}{n^2}$

Entropy of 2-dimensional SFT's

- A pattern w on a rectangle of any size is **admissible** if it contains no sub-pattern from $\mathcal{F}$.
- Let $B_{n \times n}(X)$ be the set of admissible patterns of size $n \times n$.
- Define the **entropy** $h(X) = \lim_{n\to\infty} \frac{\log |B_{n\times n}(X)|}{n^2}$
- At the expense of enlarging the alphabet, we can assume that $\mathcal{F}$ consists of patterns on 1×2 and 2×1 rectangles, i.e. *nearest neighbours*.

Entropy of 2-dimensional SFT's

- A pattern w on a rectangle of any size is **admissible** if it contains no sub-pattern from $\mathcal{F}$.
- Let $B_{n\times n}(X)$ be the set of admissible patterns of size $n \times n$.
- Define the **entropy** $h(X) = \lim_{n\to\infty} \frac{\log|B_{n\times n}(X)|}{n^2}$
- At the expense of enlarging the alphabet, we can assume that $\mathcal{F}$ consists of patterns on 1×2 and 2×1 rectangles, i.e. *nearest neighbours*.
- This yields horizontal and vertical transition matrices.

Entropy of 2-dimensional SFT's

- A pattern w on a rectangle of any size is **admissible** if it contains no sub-pattern from $\mathcal{F}$.
- Let $B_{n\times n}(X)$ be the set of admissible patterns of size $n \times n$.
- Define the **entropy** $h(X) = \lim_{n\to\infty} \frac{\log |B_{n\times n}(X)|}{n^2}$
- At the expense of enlarging the alphabet, we can assume that $\mathcal{F}$ consists of patterns on 1×2 and 2×1 rectangles, i.e. *nearest neighbours*.
- This yields horizontal and vertical transition matrices.
- However, there is no known way to compute entropy from these matrices.

Entropy of 2-dimensional SFT's

- A pattern w on a rectangle of any size is **admissible** if it contains no sub-pattern from $\mathcal{F}$.
- Let $B_{n\times n}(X)$ be the set of admissible patterns of size $n \times n$.
- Define the **entropy** $h(X) = \lim_{n\to\infty} \frac{\log |B_{n\times n}(X)|}{n^2}$
- At the expense of enlarging the alphabet, we can assume that $\mathcal{F}$ consists of patterns on 1×2 and 2×1 rectangles, i.e. *nearest neighbours*.
- This yields horizontal and vertical transition matrices.
- However, there is no known way to compute entropy from these matrices.
- exact value of entropy is known for only a handful of 2-D SFT's (unknown even for $G^{(2)}$).

Entropy of 2-dimensional SFT's

- A pattern w on a rectangle of any size is **admissible** if it contains no sub-pattern from $\mathcal{F}$.
- Let $B_{n\times n}(X)$ be the set of admissible patterns of size $n \times n$.
- Define the **entropy** $h(X) = \lim_{n\to\infty} \frac{\log |B_{n\times n}(X)|}{n^2}$
- At the expense of enlarging the alphabet, we can assume that $\mathcal{F}$ consists of patterns on 1×2 and 2×1 rectangles, i.e. *nearest neighbours*.
- This yields horizontal and vertical transition matrices.
- However, there is no known way to compute entropy from these matrices.
- exact value of entropy is known for only a handful of 2-D SFT's (unknown even for $G^{(2)}$).
- Even worse: given $\mathcal{F}$, it is algorithmically undecidable whether or not $X = \varnothing$!

Computing entropy

- Holy Grail: an exact formula for the entropy of a 2-dimensional SFT, in particular $G^{(2)}$.

Computing entropy

- Holy Grail: an exact formula for the entropy of a 2-dimensional SFT, in particular $G^{(2)}$.
- If not an exact formula, try to efficiently estimate $h(G^{(2)})$.

Computing entropy

- Holy Grail: an exact formula for the entropy of a 2-dimensional SFT, in particular $G^{(2)}$.
- If not an exact formula, try to efficiently estimate $h(G^{(2)})$.
- Current best estimates (Friedland, 2007): $0.58789116177534 \leq h(G^{(2)}) \leq 0.58789116177535$.

Strip systems

- Define H_n to be the set of configurations on an n-high strip which do not include any of the forbidden neighbours in $\mathcal{F}$.

.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$\uparrow$	...	0	1	0	0	0	0	0	0	1	0	0	0	1	0	...
n	...	0	0	1	0	0	1	0	1	0	0	1	0	0	0	...
$\vert$	...	0	0	0	1	0	0	0	0	0	1	0	1	0	0	...
$\downarrow$	...	0	1	0	0	0	1	0	1	0	0	0	0	1	0	...
..	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	

Strip systems

- Define H_n to be the set of configurations on an n-high strip which do not include any of the forbidden neighbours in $\mathcal{F}$.

$\uparrow$	...	0	1	0	0	0	0	0	0	1	0	0	0	1	0	...
n	...	0	0	1	0	0	1	0	1	0	0	1	0	0	0	...
$\vert$	...	0	0	0	1	0	0	0	0	0	1	0	1	0	0	...
$\downarrow$	...	0	1	0	0	0	1	0	1	0	0	0	0	1	0	...

- Then H_n itself can be thought of as a 1-dimensional SFT:

Strip systems

- Define H_n to be the set of configurations on an n-high strip which do not include any of the forbidden neighbours in $\mathcal{F}$.

.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$\uparrow$	...	0	1	0	0	0	0	0	0	1	0	0	0	1	0	...
n	...	0	0	1	0	0	1	0	1	0	0	1	0	0	0	...
$\vert$	...	0	0	0	1	0	0	0	0	0	1	0	1	0	0	...
$\downarrow$	...	0	1	0	0	0	1	0	1	0	0	0	0	1	0	...
..	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	

- Then H_n itself can be thought of as a 1-dimensional SFT:
  - Alphabet A_n: set of n-letter columns $\begin{matrix} a_n \\ \vdots \\ a_2 \\ a_1 \end{matrix}$ such that each $\begin{matrix} a_i \\ a_{i-1} \end{matrix}$ is admissible.

Strip systems

- Define H_n to be the set of configurations on an n-high strip which do not include any of the forbidden neighbours in $\mathcal{F}$.

$\uparrow$	...	0	1	0	0	0	0	0	0	1	0	0	0	1	0	...
n	...	0	0	1	0	0	1	0	1	0	0	1	0	0	0	...
$\mid$	...	0	0	0	1	0	0	0	0	0	1	0	1	0	0	...
$\downarrow$	...	0	1	0	0	0	1	0	1	0	0	0	0	1	0	...

- Then H_n itself can be thought of as a 1-dimensional SFT:
  - Alphabet A_n: set of n-letter columns $\begin{matrix} a_n \\ \vdots \\ a_2 \\ a_1 \end{matrix}$ such that each $\begin{matrix} a_i \\ a_{i-1} \end{matrix}$ is admissible.
  - The pair $\begin{matrix} a_n \\ \vdots \\ a_2 \\ a_1 \end{matrix} \quad \begin{matrix} b_n \\ \vdots \\ b_2 \\ b_1 \end{matrix}$ may appear if and only if each $a_i b_i$ is admissible.

Lower Bounds on Entropy, via strip systems

- For any n, define $h_n = h(H_n)$.

Lower Bounds on Entropy, via strip systems

- For any n, define $h_n = h(H_n)$.
- Fact: $h(X) = \lim_{n\to\infty} \frac{h_n}{n}$.

Lower Bounds on Entropy, via strip systems

- For any n, define $h_n = h(H_n)$.
- Fact: $h(X) = \lim_{n\to\infty} \frac{h_n}{n}$.
- Assume horizontal constraint is *symmetric*: ab is allowed if and only if ba is allowed.

Lower Bounds on Entropy, via strip systems

- For any n, define $h_n = h(H_n)$.
- Fact: $h(X) = \lim_{n\to\infty} \frac{h_n}{n}$.
- Assume horizontal constraint is *symmetric*: ab is allowed if and only if ba is allowed.
- Transition matrix M_n, for H_n, is symmetric.

+

Lower Bounds on Entropy, via strip systems

- For any n, define $h_n = h(H_n)$.
- Fact: $h(X) = \lim_{n\to\infty} \frac{h_n}{n}$.
- Assume horizontal constraint is *symmetric*: ab is allowed if and only if ba is allowed.
- Transition matrix M_n, for H_n, is symmetric.
- $h_n = \log(\lambda(M_n))$

Lower Bounds on Entropy, via strip systems

- For any n, define $h_n = h(H_n)$.
- Fact: $h(X) = \lim_{n\to\infty} \frac{h_n}{n}$.
- Assume horizontal constraint is *symmetric*: ab is allowed if and only if ba is allowed.
- Transition matrix M_n, for H_n, is symmetric.
- $h_n = \log(\lambda(M_n))$
- $\lambda(M_n)$ is lower bounded by *Rayleigh quotient*:
  Let $\mathbf{1}_n$ denote the vector of all 1's. For any p

$$\lambda((M_n)^p) \geq \frac{\mathbf{1}_n (M_n)^p \mathbf{1}_n^{\mathrm{t}}}{\mathbf{1}_n \cdot \mathbf{1}_n^{\mathrm{t}}},$$

  where numerator is a count of admissible $n \times p$ patterns.

- (Markley and Paul, 1981)

$$h(X) = \lim_{n\to\infty} \frac{h_n}{n} = \lim_{n\to\infty} \frac{\log(\lambda(M_n))}{n} \geq \lim_{n\to\infty} \frac{1}{pn} \log \frac{\mathbf{1}_n (M_n)^p \mathbf{1}_n^{\mathtt{t}}}{\mathbf{1}_n \cdot \mathbf{1}_n^{\mathtt{t}}}$$

- (Markley and Paul, 1981)

$$h(X) = \lim_{n\to\infty} \frac{h_n}{n} = \lim_{n\to\infty} \frac{\log(\lambda(M_n))}{n} \geq \lim_{n\to\infty} \frac{1}{pn} \log \frac{\mathbf{1}_n (M_n)^p \mathbf{1}_n^{\mathrm{t}}}{\mathbf{1}_n \cdot \mathbf{1}_n^{\mathrm{t}}}$$

-

$\uparrow$	...	0	1	0	0	0	0	0	0	1	0	0	0	1	0	...
n	...	0	0	1	0	0	1	0	1	0	0	1	0	0	0	...
$\vert$	...	0	0	0	1	0	0	0	0	0	1	0	1	0	0	...
$\downarrow$	...	0	1	0	0	0	1	0	1	0	0	0	0	1	0	...

- (Markley and Paul, 1981)

$$h(X) = \lim_{n\to\infty} \frac{h_n}{n} = \lim_{n\to\infty} \frac{\log(\lambda(M_n))}{n} \geq \lim_{n\to\infty} \frac{1}{pn} \log \frac{\mathbf{1}_n (M_n)^p \mathbf{1}_n^t}{\mathbf{1}_n \cdot \mathbf{1}_n^t}$$

-

	$\longleftarrow$	—	—	—	—	p	—	—	—	—	$\longrightarrow$
$\uparrow$	1	0	0	0	0	0	0	1	0	0	0
n	0	1	0	0	1	0	1	0	0	1	0
$\vert$	0	0	1	0	0	0	0	0	1	0	1
$\downarrow$	1	0	0	0	1	0	1	0	0	0	0

- (Markley and Paul, 1981)

$$h(X) = \lim_{n\to\infty} \frac{h_n}{n} = \lim_{n\to\infty} \frac{\log(\lambda(M_n))}{n} \geq \lim_{n\to\infty} \frac{1}{pn} \log \frac{\mathbf{1}_n (M_n)^p \mathbf{1}_n^{\mathrm{t}}}{\mathbf{1}_n \cdot \mathbf{1}_n^{\mathrm{t}}}$$

-

	$\longleftarrow$	$-$	$-$	$-$	$-$	p	$-$	$-$	$-$	$-$	$\longrightarrow$
$\uparrow$	1	0	0	0	0	0	0	1	0	0	0
n	0	1	0	0	1	0	1	0	0	1	0
$\vert$	0	0	1	0	0	0	0	0	1	0	1
$\downarrow$	1	0	0	0	1	0	1	0	0	0	0

- Letting V_p denote a vertical transition matrix of width p,

$$\mathbf{1}_n (M_n)^p \mathbf{1}_n^{\mathrm{t}} = \mathbf{1}_p (V_p)^n \mathbf{1}_p^{\mathrm{t}}$$

(can count patterns generated from left to right or patterns generated from bottom to top)

- (Markley and Paul, 1981)

$$h(X) = \lim_{n\to\infty} \frac{h_n}{n} = \lim_{n\to\infty} \frac{\log(\lambda(M_n))}{n} \geq \lim_{n\to\infty} \frac{1}{pn} \log \frac{\mathbf{1}_n (M_n)^p \mathbf{1}_n^{\mathrm{t}}}{\mathbf{1}_n \cdot \mathbf{1}_n^{\mathrm{t}}}$$

-

	$\longleftarrow$	—	—	—	—	p	—	—	—	—	$\longrightarrow$
$\uparrow$	1	0	0	0	0	0	0	1	0	0	0
n	0	1	0	0	1	0	1	0	0	1	0
$\vert$	0	0	1	0	0	0	0	0	1	0	1
$\downarrow$	1	0	0	0	1	0	1	0	0	0	0

- Letting V_p denote a vertical transition matrix of width p,

$$\mathbf{1}_n (M_n)^p \mathbf{1}_n^{\mathrm{t}} = \mathbf{1}_p (V_p)^n \mathbf{1}_p^{\mathrm{t}}$$

  (can count patterns generated from left to right or patterns generated from bottom to top)
- Thus,

$$h(X) \geq (1/p)(\log(\lambda(V_p)) - \log(\lambda(V_0)))$$

- (Calkin and Wilf, 1999)

$$h(X) \geq \lim_{m \to \infty} \frac{1}{pn} \log \frac{\mathbf{1}_n (M_n)^{p+2q} \mathbf{1}_n^{\mathtt{t}}}{\mathbf{1}_n (M_n)^{2q} \mathbf{1}_n^{\mathtt{t}}}$$

- (Calkin and Wilf, 1999)

$$h(X) \geq \lim_{m \to \infty} \frac{1}{pn} \log \frac{\mathbf{1}_n (M_n)^{p+2q} \mathbf{1}_n^{\mathtt{t}}}{\mathbf{1}_n (M_n)^{2q} \mathbf{1}_n^{\mathtt{t}}}$$

- Thus,

$$h(X) \geq (1/p)(\log(\lambda(V_{p+2q})) - \log(\lambda(V_{2q})))$$

- (Calkin and Wilf, 1999)

$$h(X) \geq \lim_{m\to\infty} \frac{1}{pn} \log \frac{\mathbf{1}_n (M_n)^{p+2q} \mathbf{1}_n^{\mathtt{t}}}{\mathbf{1}_n (M_n)^{2q} \mathbf{1}_n^{\mathtt{t}}}$$

- Thus,

$$h(X) \geq (1/p)(\log(\lambda(V_{p+2q})) - \log(\lambda(V_{2q})))$$

- Led to Friedland's (2007) lower bound for $h(G^{(2)})$.

- (Calkin and Wilf, 1999)

$$h(X) \geq \lim_{m\to\infty} \frac{1}{pn} \log \frac{\mathbf{1}_n (M_n)^{p+2q} \mathbf{1}_n^{\mathrm{t}}}{\mathbf{1}_n (M_n)^{2q} \mathbf{1}_n^{\mathrm{t}}}$$

- Thus,

$$h(X) \geq (1/p)(\log(\lambda(V_{p+2q})) - \log(\lambda(V_{2q})))$$

- Led to Friedland's (2007) lower bound for $h(G^{(2)})$.
- All above used $\mathbf{1}_n$ so that the limit above may be computed as the log of largest eigenvalue of a vertical transition matrix.

Improved Lower bounds

- (Louidor and Marcus, 2009) *Improved Rayleigh Method:* Replace $\mathbf{1}_n$ with sequence of vectors $\mathbf{y}_n$ such that $\mathbf{y}_n(M_n)^p\mathbf{y}_n^{\mathsf{t}}$ represents weighted counts of patterns; incorporate $\mathbf{y}_n$ into a vertical transition matrix $\tilde{V}_p$ and find $\mathbf{x}_p$ such that

$$\mathbf{y}_n(M_n)^p\mathbf{y}_n^{\mathsf{t}} = \mathbf{x}_p(\tilde{V}_p)^n\mathbf{x}_p^{\mathsf{t}}$$

Improved Lower bounds

- (Louidor and Marcus, 2009) *Improved Rayleigh Method:* Replace $\mathbf{1}_n$ with sequence of vectors $\mathbf{y}_n$ such that $\mathbf{y}_n(M_n)^p\mathbf{y}_n^t$ represents weighted counts of patterns; incorporate $\mathbf{y}_n$ into a vertical transition matrix $\tilde{V}_p$ and find $\mathbf{x}_p$ such that

$$\mathbf{y}_n(M_n)^p\mathbf{y}_n^t = \mathbf{x}_p(\tilde{V}_p)^n\mathbf{x}_p^t$$

Constraint	Old lower bound	New lower bound	Upper bound
NAK	0.4250636891	0.4250767745	0.4250767997
RWIM	0.5350150	0.5350151497	0.5350428519

Convergence of entropy approximations

- For $G^{(2)}$, h_n/n convergence appears to have error $\Theta(\frac{1}{n})$.

Convergence of entropy approximations

- For $G^{(2)}$, h_n/n convergence appears to have error $\Theta(\frac{1}{n})$.
- Computation of h_n takes exponential time.

Convergence of entropy approximations

- For $G^{(2)}$, h_n/n convergence appears to have error $\Theta(\frac{1}{n})$.
- Computation of h_n takes exponential time.
- In the 80's and 90's, data suggested that $\lim_{n\to\infty} h_{n+1} - h_n = h(G^{(2)})$, and that the error is exponentially small.

Convergence of entropy approximations

- For $G^{(2)}$, h_n/n convergence appears to have error $\Theta(\frac{1}{n})$.
- Computation of h_n takes exponential time.
- In the 80's and 90's, data suggested that $\lim_{n\to\infty} h_{n+1} - h_n = h(G^{(2)})$, and that the error is exponentially small.
- However, a proof of convergence of $h_{n+1} - h_n$ for any nondegenerate $\mathbb{Z}^2$ SFT has been an open problem.

An excellent approximation (but not quite the Holy Grail)

- **Theorem** (Pavlov, 2009): There exist positive constants A and B so that $|h_{n+1} - h_n - h(G^{(2)})| < Ae^{-Bn}$ for any n.

An excellent approximation (but not quite the Holy Grail)

- **Theorem** (Pavlov, 2009): There exist positive constants A and B so that $|h_{n+1} - h_n - h(G^{(2)})| < Ae^{-Bn}$ for any n.
- Corollary (Pavlov, 2009): $\exists$ a polynomial $p(n)$ so that $h(G^{(2)})$ can be approximated to within $\frac{1}{n}$ in $p(n)$ steps.

An excellent approximation (but not quite the Holy Grail)

- **Theorem** (Pavlov, 2009): There exist positive constants A and B so that $|h_{n+1} - h_n - h(G^{(2)})| < Ae^{-Bn}$ for any n.
- Corollary (Pavlov, 2009): $\exists$ a polynomial $p(n)$ so that $h(G^{(2)})$ can be approximated to within $\frac{1}{n}$ in $p(n)$ steps.
- 2-dimensional characterization of set of entropies:

An excellent approximation (but not quite the Holy Grail)

- **Theorem** (Pavlov, 2009): There exist positive constants A and B so that $|h_{n+1} - h_n - h(G^{(2)})| < Ae^{-Bn}$ for any n.
- Corollary (Pavlov, 2009): $\exists$ a polynomial $p(n)$ so that $h(G^{(2)})$ can be approximated to within $\frac{1}{n}$ in $p(n)$ steps.
- 2-dimensional characterization of set of entropies:
  - **Theorem** (Hochman and Meyerovitch, 2007): A number h is the entropy of a 2-dimensional SFT if and only if there is a Turing machine that can generate a list of rationals $\frac{p_n}{q_n}$ which approach h from above.

An excellent approximation (but not quite the Holy Grail)

- **Theorem** (Pavlov, 2009): There exist positive constants A and B so that $|h_{n+1} - h_n - h(G^{(2)})| < Ae^{-Bn}$ for any n.
- Corollary (Pavlov, 2009): $\exists$ a polynomial $p(n)$ so that $h(G^{(2)})$ can be approximated to within $\frac{1}{n}$ in $p(n)$ steps.
- 2-dimensional characterization of set of entropies:
  - **Theorem** (Hochman and Meyerovitch, 2007): A number h is the entropy of a 2-dimensional SFT if and only if there is a Turing machine that can generate a list of rationals $\frac{p_n}{q_n}$ which approach h from above.
  - Strikingly different from Lind's 1-dimensional characterization.

An excellent approximation (but not quite the Holy Grail)

- **Theorem** (Pavlov, 2009): There exist positive constants A and B so that $|h_{n+1} - h_n - h(G^{(2)})| < Ae^{-Bn}$ for any n.
- Corollary (Pavlov, 2009): $\exists$ a polynomial $p(n)$ so that $h(G^{(2)})$ can be approximated to within $\frac{1}{n}$ in $p(n)$ steps.
- 2-dimensional characterization of set of entropies:
  - **Theorem** (Hochman and Meyerovitch, 2007): A number h is the entropy of a 2-dimensional SFT if and only if there is a Turing machine that can generate a list of rationals $\frac{p_n}{q_n}$ which approach h from above.
  - Strikingly different from Lind's 1-dimensional characterization.
  - For a typical such entropy, $p_n/q_n \to h$ very slowly and there is no indication of error size, $(p_n/q_n - h)$.

An excellent approximation (but not quite the Holy Grail)

- **Theorem** (Pavlov, 2009): There exist positive constants A and B so that $|h_{n+1} - h_n - h(G^{(2)})| < Ae^{-Bn}$ for any n.
- Corollary (Pavlov, 2009): $\exists$ a polynomial $p(n)$ so that $h(G^{(2)})$ can be approximated to within $\frac{1}{n}$ in $p(n)$ steps.
- 2-dimensional characterization of set of entropies:
  - **Theorem** (Hochman and Meyerovitch, 2007): A number h is the entropy of a 2-dimensional SFT if and only if there is a Turing machine that can generate a list of rationals $\frac{p_n}{q_n}$ which approach h from above.
  - Strikingly different from Lind's 1-dimensional characterization.
  - For a typical such entropy, $p_n/q_n \to h$ very slowly and there is no indication of error size, $(p_n/q_n - h)$.
  - Thus, $h(G^{(2)})$ is much "nicer" than the typical entropy.

Outline of Pavlov's Proof

- Introduce a stationary process μ_n on each H_n of maximal measure-theoretic (Shannon) entropy: $h_{\mu_n} = h(H_n)$.

Outline of Pavlov's Proof

- Introduce a stationary process μ_n on each H_n of maximal measure-theoretic (Shannon) entropy: $h_{\mu_n} = h(H_n)$.
- Decompose h_{μ_n} into a sum of n conditional measure-theoretic entropies, row by row.

Outline of Pavlov's Proof

- Introduce a stationary process μ_n on each H_n of maximal measure-theoretic (Shannon) entropy: $h_{\mu_n} = h(H_n)$.
- Decompose h_{μ_n} into a sum of n conditional measure-theoretic entropies, row by row.
- Pair off:

Outline of Pavlov's Proof

- Introduce a stationary process μ_n on each H_n of maximal measure-theoretic (Shannon) entropy: $h_{\mu_n} = h(H_n)$.
- Decompose h_{μ_n} into a sum of n conditional measure-theoretic entropies, row by row.
- Pair off:
  - top $n/2$ rows of $h_{\mu_{n+1}}$ and h_{μ_n}

Outline of Pavlov's Proof

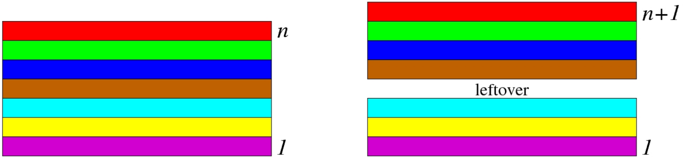
- Introduce a stationary process μ_n on each H_n of maximal measure-theoretic (Shannon) entropy: $h_{\mu_n} = h(H_n)$.
- Decompose h_{μ_n} into a sum of n conditional measure-theoretic entropies, row by row.
- Pair off:
  - top $n/2$ rows of $h_{\mu_{n+1}}$ and h_{μ_n}
  - bottom $n/2$ rows of $h_{\mu_{n+1}}$ and h_{μ_n}

Outline of Pavlov's Proof

- Introduce a stationary process μ_n on each H_n of maximal measure-theoretic (Shannon) entropy: $h_{\mu_n} = h(H_n)$.
- Decompose h_{μ_n} into a sum of n conditional measure-theoretic entropies, row by row.
- Pair off:
  - top $n/2$ rows of $h_{\mu_{n+1}}$ and h_{μ_n}
  - bottom $n/2$ rows of $h_{\mu_{n+1}}$ and h_{μ_n}
  - the middle row of $h_{\mu_{n+1}}$ remains.

Outline of Pavlov's Proof

- Introduce a stationary process μ_n on each H_n of maximal measure-theoretic (Shannon) entropy: $h_{\mu_n} = h(H_n)$.
- Decompose h_{μ_n} into a sum of n conditional measure-theoretic entropies, row by row.
- Pair off:
  - top $n/2$ rows of $h_{\mu_{n+1}}$ and h_{μ_n}
  - bottom $n/2$ rows of $h_{\mu_{n+1}}$ and h_{μ_n}
  - the middle row of $h_{\mu_{n+1}}$ remains.



Percolation Model

- differences between corresponding rows decay exponentially

Percolation Model

- differences between corresponding rows decay exponentially
- middle row converges exponentially

Percolation Model

- differences between corresponding rows decay exponentially
- middle row converges exponentially
- All exponential decay/convergence statements come from comparison with an associated *percolation process* (vandenBerg-Maes (1994)):

Percolation Model

- differences between corresponding rows decay exponentially
- middle row converges exponentially
- All exponential decay/convergence statements come from comparison with an associated *percolation process* (vandenBerg-Maes (1994)):
  - On the Z^2 lattice, a site is “open” with probability p and closed with probability $1 - p$, independent from site to site.

Percolation Model

- differences between corresponding rows decay exponentially
- middle row converges exponentially
- All exponential decay/convergence statements come from comparison with an associated *percolation process* (vandenBerg-Maes (1994)):
  - On the Z^2 lattice, a site is "open" with probability p and closed with probability $1 - p$, independent from site to site.
  - For $p < p_c$, the critical probability, the probability of an "open" path from the origin to the boundary of an $n \times n$ square decays exponentially fast in n.

Generalizations

Theorem (Marcus and Pavlov, 2009):

- Exponential approximations (differences of strip entropies) to entropy for a class of 2-dimensional SFT's (generalizing Pavlov's result for $G^{(2)}$).

Generalizations

Theorem (Marcus and Pavlov, 2009):

- Exponential approximations (differences of strip entropies) to entropy for a class of 2-dimensional SFT's (generalizing Pavlov's result for $G^{(2)}$).
- Exponential approximations (differences of strip entropies) to measure-theoretic entropy for a class of Markov Random Fields (2-dimensional analogue of 1-dimensional Markov chain and probabilistic analogue of 2-dimensional SFT)

1-dimensional sofic shifts

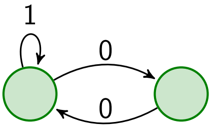
- A **1 dimensional sofic shift** is the set of all bi-infinite sequences obtained from a labelled finite directed graph.

1-dimensional sofic shifts

- A **1 dimensional sofic shift** is the set of all bi-infinite sequences obtained from a labelled finite directed graph.
- Examples: All 1-dimensional SFT's.

1-dimensional sofic shifts

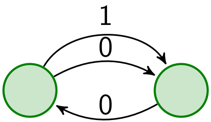
- A **1 dimensional sofic shift** is the set of all bi-infinite sequences obtained from a labelled finite directed graph.
- Examples: All 1-dimensional SFT's.
- Example: (a sofic, non-SFT shift) The EVEN Shift
  $A = \{0, 1\}$:



More examples of 1-dimensional sofic, non-SFT, shifts

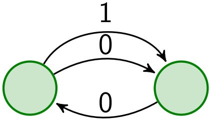
- The ODD Shift

  $A = \{0, 1\}$:

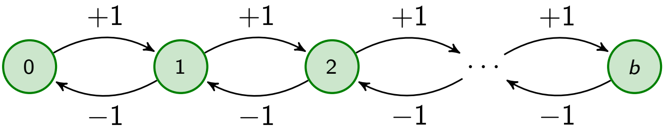


More examples of 1-dimensional sofic, non-SFT, shifts

- The ODD Shift
  $A = \{0, 1\}$:



- The CHG(b) shift
  $A = \{+1, -1\}$:



$$w_1 \dots w_m \in B_m(X) \iff \text{for all } 1 \leq s \leq t \leq m, \left| \sum_{i=s}^{t} w_i \right| \leq b$$

2-dimensional sofic shifts

- A **2-dimensional sofic shift** is the set of all configurations on the entire $\mathbb{Z}^2$ lattice obtained from two (one horizontal and one vertical) finite directed labelled graphs with the same set of edges.

2-dimensional sofic shifts

- A **2-dimensional sofic shift** is the set of all configurations on the entire $\mathbb{Z}^2$ lattice obtained from two (one horizontal and one vertical) finite directed labelled graphs with the same set of edges.
- Examples:

2-dimensional sofic shifts

- A **2-dimensional sofic shift** is the set of all configurations on the entire $\mathbb{Z}^2$ lattice obtained from two (one horizontal and one vertical) finite directed labelled graphs with the same set of edges.
- Examples:
  - All 2-dimensional SFT's.

2-dimensional sofic shifts

- A **2-dimensional sofic shift** is the set of all configurations on the entire $\mathbb{Z}^2$ lattice obtained from two (one horizontal and one vertical) finite directed labelled graphs with the same set of edges.
- Examples:
  - All 2-dimensional SFT's.
  - $\mathrm{EVEN}^{\otimes^2}$: all rows and columns satisfy the 1-dimensional EVEN shift.

2-dimensional sofic shifts

- A **2-dimensional sofic shift** is the set of all configurations on the entire $\mathbb{Z}^2$ lattice obtained from two (one horizontal and one vertical) finite directed labelled graphs with the same set of edges.
- Examples:
  - All 2-dimensional SFT's.
  - $\mathrm{EVEN}^{\otimes^2}$: all rows and columns satisfy the 1-dimensional EVEN shift.
  - $\mathrm{CHG}(b)^{\otimes^2}$: all rows and columns satisfy the 1-dimensional $\mathrm{CHG}(b)$ shift.

Computing entropy of 2-dimensional sofic shifts

- (Louidor and Marcus, 2009): applied improved Rayleigh method to estimate entropies of sofic shifts $\mathrm{EVEN}^{\otimes^2}$ and $\mathrm{CHG}(3)^{\otimes^2}$:

Constraint	Old lower bound	New lower bound	Upper bound
$\mathrm{EVEN}^{\otimes 2}$	0.4385027973	0.4402086447	0.4452873312
$\mathrm{CHG}(3)^{\otimes 2}$	0.4210209862	0.4222689819	0.5328488954

Computing entropy of 2-dimensional sofic shifts

- (Louidor and Marcus, 2009): applied improved Rayleigh method to estimate entropies of sofic shifts $\mathrm{EVEN}^{\otimes^2}$ and $\mathrm{CHG}(3)^{\otimes^2}$:

Constraint	Old lower bound	New lower bound	Upper bound
$\mathrm{EVEN}^{\otimes 2}$	0.4385027973	0.4402086447	0.4452873312
$\mathrm{CHG}(3)^{\otimes 2}$	0.4210209862	0.4222689819	0.5328488954

- **Theorem** (Louidor and Marcus, 2009): For all dimensions D,

Computing entropy of 2-dimensional sofic shifts

- (Louidor and Marcus, 2009): applied improved Rayleigh method to estimate entropies of sofic shifts $\mathrm{EVEN}^{\otimes^2}$ and $\mathrm{CHG}(3)^{\otimes^2}$:

Constraint	Old lower bound	New lower bound	Upper bound
$\mathrm{EVEN}^{\otimes 2}$	0.4385027973	0.4402086447	0.4452873312
$\mathrm{CHG}(3)^{\otimes 2}$	0.4210209862	0.4222689819	0.5328488954

- **Theorem** (Louidor and Marcus, 2009): For all dimensions D,
  - $h(\mathrm{ODD}^{\otimes^D}) = 1/2$.

Computing entropy of 2-dimensional sofic shifts

- (Louidor and Marcus, 2009): applied improved Rayleigh method to estimate entropies of sofic shifts $\mathrm{EVEN}^{\otimes^2}$ and $\mathrm{CHG}(3)^{\otimes^2}$:

Constraint	Old lower bound	New lower bound	Upper bound
$\mathrm{EVEN}^{\otimes 2}$	0.4385027973	0.4402086447	0.4452873312
$\mathrm{CHG}(3)^{\otimes 2}$	0.4210209862	0.4222689819	0.5328488954

- **Theorem** (Louidor and Marcus, 2009): For all dimensions D,
  - $h(\mathrm{ODD}^{\otimes^D}) = 1/2$.
  - $h(\mathrm{CHG}(2)^{\otimes^D}) = 1/2^d$.

Why is $h(\mathrm{CHG}(2)^{\otimes D}) \geq 1/2^D$?

- $X = \mathrm{CHG}(2)^{\otimes D}$.
- $D=2$. Consider the two "checkerboard" 2×2 arrays, $\Gamma^{(0)}$, $\Gamma^{(1)}$

$$\Gamma^{(0)} = \begin{pmatrix} + & - \\ - & + \end{pmatrix} \qquad \Gamma^{(1)} = \begin{pmatrix} - & + \\ + & - \end{pmatrix}$$

- Any tiling consisting of $n \times n$ copies of $\Gamma^{(0)}$ or $\Gamma^{(1)}$ is a $2n \times 2n$ array that satisfies X.

$$\begin{pmatrix} \Gamma^{(i_{1,1})} & \Gamma^{(i_{1,2})} & \dots & \Gamma^{(i_{1,n})} \\ \Gamma^{(i_{2,1})} & \Gamma^{(i_{2,2})} & \dots & \Gamma^{(i_{2,n})} \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma^{(i_{n,1})} & \Gamma^{(i_{n,2})} & \dots & \Gamma^{(i_{n,n})} \end{pmatrix}, \quad i_{s,t} \in \{0, 1\}$$

Why is $h(\mathrm{CHG}(2)^{\otimes D}) \geq 1/2^D$? (cont.)

- Generally, for arbitrary D, consider the two $2 \times 2 \times \ldots \times 2$ checkerboard arrays:

$$\Gamma^{(0)}_{i_1,\ldots,i_D} = (-1)^{\sum i_j} \qquad \Gamma^{(1)}_{i_1,\ldots,i_D} = (-1)^{1+\sum i_j}$$

- Any tiling of $n \times n \times \ldots \times n$ copies of $\Gamma^{(0)}$ or $\Gamma^{(1)}$ is a $2n \times 2n \times \ldots \times 2n$ array that satisfies X.

$$\implies |B_{2n \times 2n \times \ldots \times 2n}(X)| \geq 2^{n^D}$$

$$\implies \frac{\log |B_{2n \times 2n \times \ldots \times 2n}(X)|}{(2n)^D} \geq \frac{n^D}{(2n)^D}$$

$$\implies h(X) \geq \frac{1}{2^D}.$$

Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$?

- For $D=1$ every legal word of X is essentially such a tiling of checkerboard arrays:

Lemma

$x_0 \ldots x_{n-1}$ satisfies $\mathrm{CHG}(2)$, iff
$x_i = -x_{i+1}$ for all <u>even</u> $i \in \{0, \ldots, n-2\}$ or
$x_i = -x_{i+1}$ for all <u>odd</u> $i \in \{0, \ldots, n-2\}$.

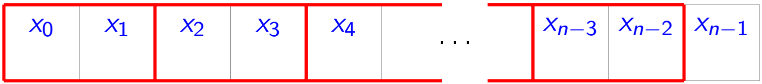
x_0	x_1	x_2	x_3	x_4	$\ldots$	x_{n-3}	x_{n-2}	x_{n-1}

Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$?

- For $D=1$ every legal word of X is essentially such a tiling of checkerboard arrays:

Lemma

$x_0 \ldots x_{n-1}$ satisfies $\mathrm{CHG}(2)$, iff
$x_i = -x_{i+1}$ for all <u>even</u> $i \in \{0, \ldots, n-2\}$ or
$x_i = -x_{i+1}$ for all <u>odd</u> $i \in \{0, \ldots, n-2\}$.

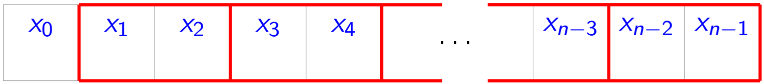


Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$?

- For $D=1$ every legal word of X is essentially such a tiling of checkerboard arrays:

Lemma

$x_0 \ldots x_{n-1}$ satisfies $\mathrm{CHG}(2)$, iff
$x_i = -x_{i+1}$ for all <u>even</u> $i \in \{0, \ldots, n-2\}$ or
$x_i = -x_{i+1}$ for all <u>odd</u> $i \in \{0, \ldots, n-2\}$.

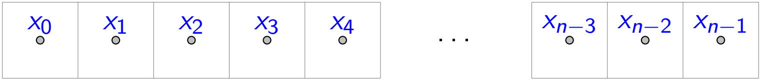


Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$?

- For $D{=}1$ every legal word of X is essentially such a tiling of checkerboard arrays:

Lemma

$x_0 \ldots x_{n-1}$ satisfies $\mathrm{CHG}(2)$, iff
$x_i = -x_{i+1}$ for all <u>even</u> $i \in \{0, \ldots, n-2\}$ or
$x_i = -x_{i+1}$ for all <u>odd</u> $i \in \{0, \ldots, n-2\}$.

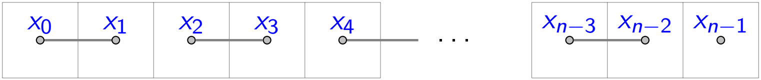


Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$?

- For $D=1$ every legal word of X is essentially such a tiling of checkerboard arrays:

Lemma

$x_0 \ldots x_{n-1}$ satisfies $\mathrm{CHG}(2)$, iff
$x_i = -x_{i+1}$ for all <u>even</u> $i \in \{0, \ldots, n-2\}$ or
$x_i = -x_{i+1}$ for all <u>odd</u> $i \in \{0, \ldots, n-2\}$.



<u>Phase-0</u> sequence

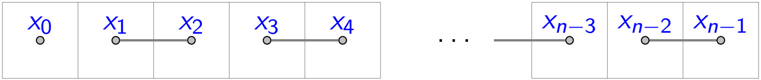
$$T_0(i) = \begin{cases} i+1 & \text{if } i \text{ is even} \\ i-1 & \text{if } i \text{ is odd} \end{cases}$$

Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$?

- For $D = 1$ every legal word of X is essentially such a tiling of checkerboard arrays:

Lemma

$x_0 \ldots x_{n-1}$ satisfies $\mathrm{CHG}(2)$, iff
$x_i = -x_{i+1}$ for all <u>even</u> $i \in \{0, \ldots, n-2\}$ or
$x_i = -x_{i+1}$ for all <u>odd</u> $i \in \{0, \ldots, n-2\}$.

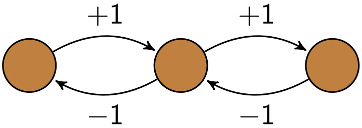


<u>Phase-1</u> sequence

$$T_1(i) = \begin{cases} i-1 & \text{if } i \text{ is even} \\ i+1 & \text{if } i \text{ is odd} \end{cases}$$

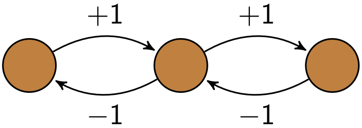
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$?

- Proof:



Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$?

- Proof:



- Unfortunately, the previous Lemma does not generalize to larger dimension:

+	−	−	+
+	+	−	−
−	+	+	−
−	−	+	+

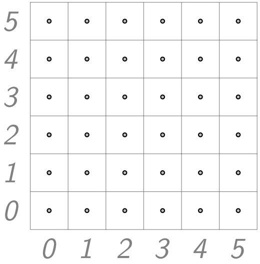
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- $\Gamma \in B_{n\times n\times \ldots \times n}(X)$ iff every row of Γ is either a phase-0 or a phase-1 sequence.
- $\mathbf{r} = (r_i)$: binary vector with an entry for each row of $\{0, \ldots, n-1\}^D$.
- $A(\mathbf{r}) = \{\Gamma \in B_{n\times n\times \ldots \times n}(X) \;:\; \text{row } i \text{ of } \Gamma \text{ has phase } r_i\}$

$$\stackrel{\text{Lemma 1}}{\Longrightarrow} B_{n\times n\times \ldots \times n}(X) = \bigcup_{\mathbf{r}} A(\mathbf{r}).$$

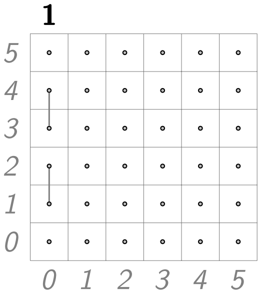
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.* $D = 2$:



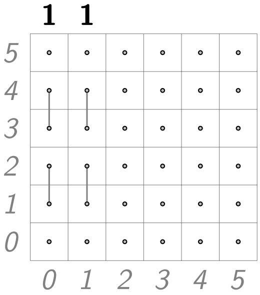
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.* $D = 2$:



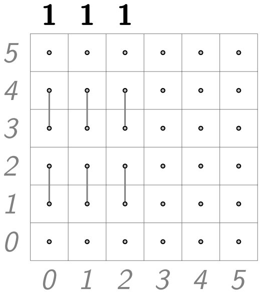
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.* $D = 2$:



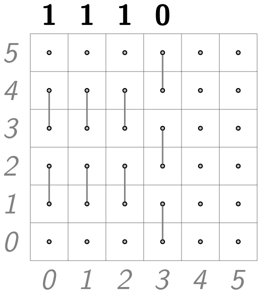
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \le 1/2^D$? (cont.)

- *Example.* $D = 2$:



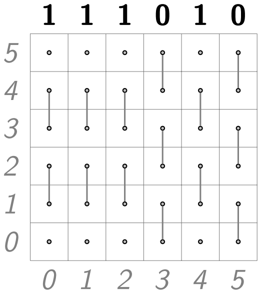
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \le 1/2^D$? (cont.)

- *Example.* $D = 2$:



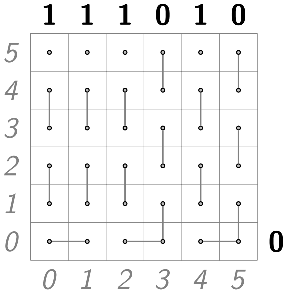
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.* $D = 2$:



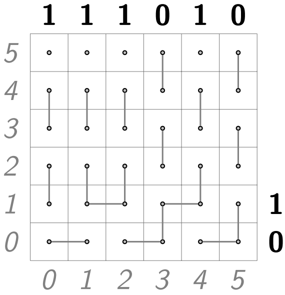
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.* $D = 2$:



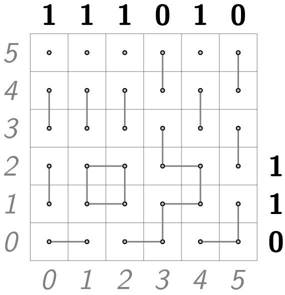
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.* $D = 2$:



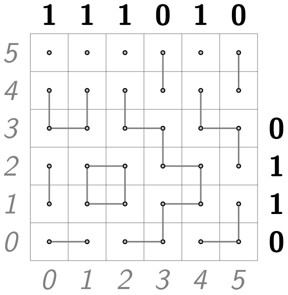
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.* $D = 2$:



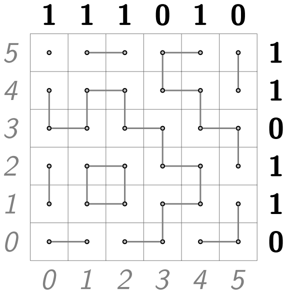
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.*$D = 2$:



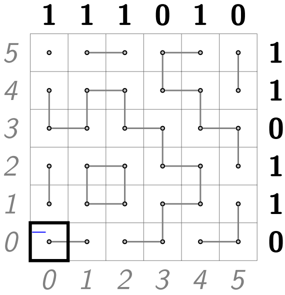
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \le 1/2^D$? (cont.)

- *Example.* $D = 2$:



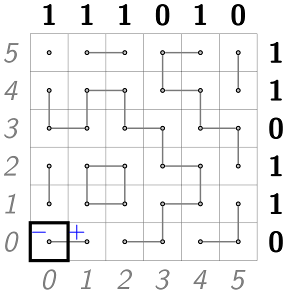
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.* $D = 2$:



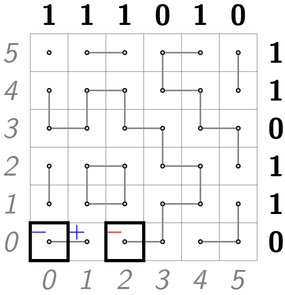
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.* $D = 2$:



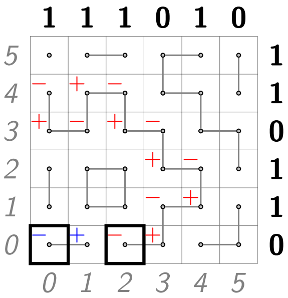
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \le 1/2^D$? (cont.)

- *Example.* $D = 2$:



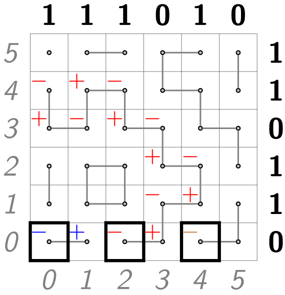
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.* $D = 2$:



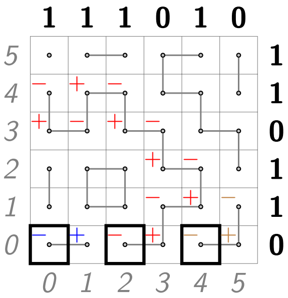
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example*. $D = 2$:



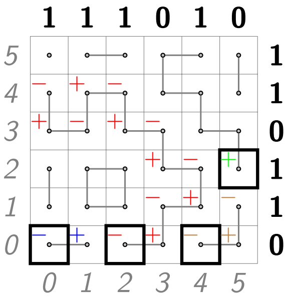
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.* $D = 2$:



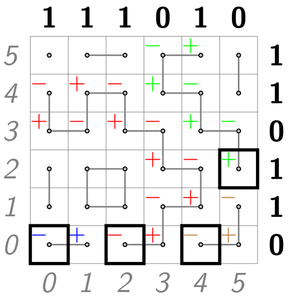
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.* $D = 2$:



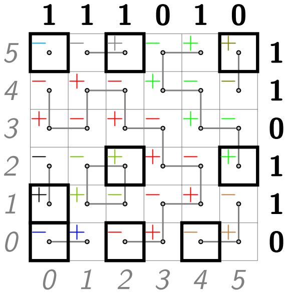
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \le 1/2^D$? (cont.)

- *Example.* $D = 2$:



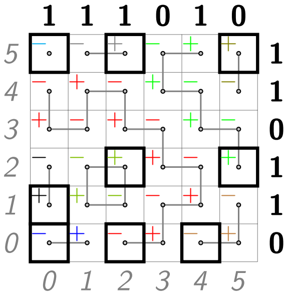
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.* $D = 2$:



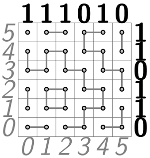
Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.* $D = 2$:



$$|A(\mathbf{r})| = 2^{(\#\text{ of connected components})}.$$

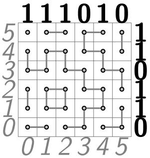
- *Example.* $D = 2$:



- For a site $\mathbf{x} \in \{0, 1, \ldots, n-1\}^D$:
- $\phi(\mathbf{r}, i, \mathbf{x})$ = "phase of row passing through $\mathbf{x}$ in direction i."

Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.* $D = 2$:

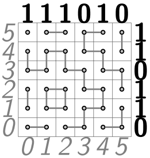


- For a site $\mathbf{x} \in \{0, 1, \ldots, n-1\}^D$:
- $\phi(\mathbf{r}, i, \mathbf{x})$ = "phase of row passing through $\mathbf{x}$ in direction i."
- D "match" functions $M_{\mathbf{r},1}, \ldots, M_{\mathbf{r},D}$.
  $M_{\mathbf{r},i} : \{0, 1, \ldots, n-1\}^D \to \mathbb{Z}^D$.

$$M_{\mathbf{r},i}(\mathbf{x}) = (x_1, \ldots, x_{i-1}, T_{\phi(\mathbf{r},i,\mathbf{x})}(x_i), x_{i+1}, \ldots, x_D),$$
$$\mathbf{x} = (x_1, \ldots, x_D).$$

Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.* $D = 2$:



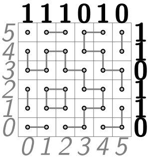
- For a site $\mathbf{x} \in \{0, 1, \ldots, n-1\}^D$:
- $\phi(\mathbf{r}, i, \mathbf{x})$ = "phase of row passing through $\mathbf{x}$ in direction i."
- D "match" functions $M_{\mathbf{r},1}, \ldots, M_{\mathbf{r},D}$.
  $M_{\mathbf{r},i} : \{0, 1, \ldots, n-1\}^D \to \mathbb{Z}^D$.

$$M_{\mathbf{r},i}(\mathbf{x}) = (x_1, \ldots, x_{i-1}, T_{\phi(\mathbf{r},i,\mathbf{x})}(x_i), x_{i+1}, \ldots, x_D),$$
$$\mathbf{x} = (x_1, \ldots, x_D).$$

- $G_{\mathbf{r}} = (V = \{0, 1, \ldots, n-1\}^D, E)$.
  $\mathbf{u} — \mathbf{v} \in E$ iff $\mathbf{v} = M_{\mathbf{r},i}(\mathbf{u})$.

Why is $h(\mathrm{CHG}(2)^{\otimes D}) \leq 1/2^D$? (cont.)

- *Example.* $D = 2$:



- For a site $\mathbf{x} \in \{0, 1, \ldots, n-1\}^D$:
- $\phi(\mathbf{r}, i, \mathbf{x})$ = "phase of row passing through $\mathbf{x}$ in direction i."
- D "match" functions $M_{\mathbf{r},1}, \ldots, M_{\mathbf{r},D}$. $M_{\mathbf{r},i} : \{0, 1, \ldots, n-1\}^D \to \mathbb{Z}^D$.

$$M_{\mathbf{r},i}(\mathbf{x}) = (x_1, \ldots, x_{i-1}, T_{\phi(\mathbf{r},i,\mathbf{x})}(x_i), x_{i+1}, \ldots, x_D),$$
$$\mathbf{x} = (x_1, \ldots, x_D).$$

- $G_{\mathbf{r}} = (V = \{0, 1, \ldots, n-1\}^D, E)$. $\mathbf{u} — \mathbf{v} \in E$ iff $\mathbf{v} = M_{\mathbf{r},i}(\mathbf{u})$.
- $|A(\mathbf{r})| = 2^{(\#\text{ of connected components of } G_{\mathbf{r}})}$.

Max # of Connected Components in $G_{\mathbf{r}}$.

- D=2.

Max # of Connected Components in $G_{\mathbf{r}}$.

- $D=2$.
  - For $(x, y) \in \{1, 2, \ldots, n-2\}^2$ (not on the "border"):

$$(x, y),\ M_{\mathbf{r},1}(x, y),\ M_{\mathbf{r},2}(x, y),\ M_{\mathbf{r},1}(M_{\mathbf{r},2}(x, y))$$

Max # of Connected Components in $G_{\mathbf{r}}$.

- $D = 2$.
  - For $(x, y) \in \{1, 2, \ldots, n-2\}^2$ (not on the "border"):

$$(x, y),\ M_{\mathbf{r},1}(x, y),\ M_{\mathbf{r},2}(x, y),\ M_{\mathbf{r},1}(M_{\mathbf{r},2}(x, y))$$

  - Are all in the connected component of (x, y).

Max # of Connected Components in $G_{\mathbf{r}}$.

- $D=2$.
  - For $(x,y)\in\{1,2,\ldots,n-2\}^2$ (not on the "border"):

$$(x,y),\ M_{\mathbf{r},1}(x,y),\ M_{\mathbf{r},2}(x,y),\ M_{\mathbf{r},1}(M_{\mathbf{r},2}(x,y))$$

  - Are all in the connected component of (x,y).
  - Are all distinct.

Max # of Connected Components in $G_{\mathbf{r}}$.

- $D=2$.
  - For $(x,y)\in\{1,2,\ldots,n-2\}^2$ (not on the "border"):

$$(x,y),\ M_{\mathbf{r},1}(x,y),\ M_{\mathbf{r},2}(x,y),\ M_{\mathbf{r},1}(M_{\mathbf{r},2}(x,y))$$

  - Are all in the connected component of (x,y).
  - Are all distinct.
- For general D:
  - For $\mathbf{x}\in\{1,2,\ldots,n-2\}^D$ (not on the "border"), the 2^D entries:

$$M_{\mathbf{r},i_1}(M_{\mathbf{r},i_2}(\ldots(M_{\mathbf{r},i_s}(\mathbf{x}))\ldots)),$$
$$\text{For each } \{i_1,\ldots,i_s\}\subseteq\{1,2,\ldots,D\},\ \ 1\le i_1<i_2<\ldots<i_s\le D$$

Max # of Connected Components in $G_{\mathbf{r}}$.

- $D = 2$.
  - For $(x, y) \in \{1, 2, \ldots, n-2\}^2$ (not on the "border"):

$$(x, y),\ M_{\mathbf{r},1}(x, y),\ M_{\mathbf{r},2}(x, y),\ M_{\mathbf{r},1}(M_{\mathbf{r},2}(x, y))$$

  - Are all in the connected component of (x, y).
  - Are all distinct.
- For general D:
  - For $\mathbf{x} \in \{1, 2, \ldots, n-2\}^D$ (not on the "border"), the 2^D entries:

$$M_{\mathbf{r},i_1}(M_{\mathbf{r},i_2}(\ldots(M_{\mathbf{r},i_s}(\mathbf{x}))\ldots)),$$
$$\text{For each } \{i_1, \ldots, i_s\} \subseteq \{1, 2, \ldots, D\},\ \ 1 \le i_1 < i_2 < \ldots < i_s \le D$$

  - Are all in the connected component of $\mathbf{x}$.

Max # of Connected Components in $G_{\mathbf{r}}$.

- $D{=}2$.
  - For $(x,y){\in}\{1,2,\ldots,n{-}2\}^2$ (not on the "border"):

$$(x,y),\ M_{\mathbf{r},1}(x,y),\ M_{\mathbf{r},2}(x,y),\ M_{\mathbf{r},1}(M_{\mathbf{r},2}(x,y))$$

  - Are all in the connected component of (x,y).
  - Are all distinct.
- For general D:
  - For $\mathbf{x}{\in}\{1,2,\ldots,n{-}2\}^D$ (not on the "border"), the 2^D entries:

$$M_{\mathbf{r},i_1}(M_{\mathbf{r},i_2}(\ldots(M_{\mathbf{r},i_s}(\mathbf{x}))\ldots)),$$
$$\text{For each } \{i_1,\ldots,i_s\}{\subseteq}\{1,2,\ldots,D\},\ \ 1{\le}i_1{<}i_2{<}\ldots{<}i_s{\le}D$$

  - Are all in the connected component of $\mathbf{x}$.
  - Are all distinct.

Max # of Connected Components in $G_{\mathbf{r}}$.

- $D{=}2$.
  - For $(x, y) \in \{1, 2, \ldots, n{-}2\}^2$ (not on the "border"):

$$(x, y),\ M_{\mathbf{r},1}(x, y),\ M_{\mathbf{r},2}(x, y),\ M_{\mathbf{r},1}(M_{\mathbf{r},2}(x, y))$$

  - Are all in the connected component of (x, y).
  - Are all distinct.
- For general D:
  - For $\mathbf{x} \in \{1, 2, \ldots, n{-}2\}^D$ (not on the "border"), the 2^D entries:

$$M_{\mathbf{r},i_1}(M_{\mathbf{r},i_2}(\ldots(M_{\mathbf{r},i_s}(\mathbf{x}))\ldots)),$$
$$\text{For each } \{i_1, \ldots, i_s\} \subseteq \{1, 2, \ldots, D\},\ \ 1 \le i_1 < i_2 < \ldots < i_s \le D$$

  - Are all in the connected component of $\mathbf{x}$.
  - Are all distinct.
- $\implies$ Any component having a vertex in the interior has at least 2^D vertices.

Max # of Connected Components in $G_{\mathbf{r}}$.

- $D=2$.
  - For $(x,y) \in \{1,2,\ldots,n-2\}^2$ (not on the "border"):

$$(x,y),\ M_{\mathbf{r},1}(x,y),\ M_{\mathbf{r},2}(x,y),\ M_{\mathbf{r},1}(M_{\mathbf{r},2}(x,y))$$

  - Are all in the connected component of (x,y).
  - Are all distinct.
- For general D:
  - For $\mathbf{x} \in \{1,2,\ldots,n-2\}^D$ (not on the "border"), the 2^D entries:

$$M_{\mathbf{r},i_1}(M_{\mathbf{r},i_2}(\ldots(M_{\mathbf{r},i_s}(\mathbf{x}))\ldots)),$$
$$\text{For each } \{i_1,\ldots,i_s\} \subseteq \{1,2,\ldots,D\},\ \ 1 \le i_1 < i_2 < \ldots < i_s \le D$$

  - Are all in the connected component of $\mathbf{x}$.
  - Are all distinct.
- $\implies$ Any component having a vertex in the interior has at least 2^D vertices.
- $\implies$ There are at most $n^D/2^D$ such components.

Max # of Connected Components in $G_{\mathbf{r}}$ (cont.).

$$\implies \begin{pmatrix} \text{\# of components having a vertex in } \{1, 2, \ldots, n-2\}^D \end{pmatrix} \le n^D / 2^D$$

Max # of Connected Components in $G_{\mathbf{r}}$ (cont.).

$$\Longrightarrow \begin{pmatrix}\text{\# of components having a vertex in } \{1,2,\ldots,n-2\}^D\end{pmatrix} \le n^D/2^D$$

$$\begin{pmatrix}\text{\# of components not having a vertex in } \{1,2,\ldots,n-2\}^D\end{pmatrix} \le \begin{pmatrix}\text{\# of vertices not in } \{1,2,\ldots,n-2\}^D\end{pmatrix} = n^D - (n-2)^D$$

$$\implies \begin{pmatrix}\text{\# of components} \\ \text{having a vertex in} \\ \{1,2,\ldots,n-2\}^D\end{pmatrix} \le n^D/2^D$$

$$\begin{pmatrix}\text{\# of components} \\ \text{not having a vertex} \\ \text{in } \{1,2,\ldots,n-2\}^D\end{pmatrix} \le \begin{pmatrix}\text{\# of vertices not in} \\ \{1,2,\ldots,n-2\}^D\end{pmatrix} = n^D - (n-2)^D$$

$$\implies (\text{Total \# of components}) \le n^D/2^D + n^D - (n-2)^D.$$

Max # of Connected Components in $G_{\mathbf{r}}$ (cont.).

$$\implies \begin{pmatrix} \text{\# of components having a vertex in } \{1,2,\ldots,n-2\}^D \end{pmatrix} \le n^D/2^D$$

$$\begin{pmatrix} \text{\# of components not having a vertex in } \{1,2,\ldots,n-2\}^D \end{pmatrix} \le \begin{pmatrix} \text{\# of vertices not in } \{1,2,\ldots,n-2\}^D \end{pmatrix} = n^D-(n-2)^D$$

$$\implies (\text{Total \# of components}) \le n^D/2^D + n^D - (n-2)^D.$$

$$\implies |A(\mathbf{r})| = 2^{(\text{Total \# of components})} \le 2^{n^D/2^D+n^D-(n-2)^D}$$

Max # of Connected Components in $G_{\mathbf{r}}$ (cont.).

$$\implies \begin{pmatrix} \text{\# of components having a vertex in} \\ \{1,2,\ldots,n-2\}^D \end{pmatrix} \le n^D/2^D$$

$$\begin{pmatrix} \text{\# of components not having a vertex} \\ \text{in } \{1,2,\ldots,n-2\}^D \end{pmatrix} \le \begin{pmatrix} \text{\# of vertices not in} \\ \{1,2,\ldots,n-2\}^D \end{pmatrix} = n^D - (n-2)^D$$

$$\implies (\text{Total \# of components}) \le n^D/2^D + n^D - (n-2)^D.$$

$$\implies |A(\mathbf{r})| = 2^{(\text{Total \# of components})} \le 2^{n^D/2^D + n^D - (n-2)^D}$$

$$\implies |B_{n\times n\times\ldots\times n}(X)| \le \sum_{\mathbf{r}} |A(\mathbf{r})| \le 2^{Dn^{D-1}} 2^{n^D/2^D + n^D - (n-2)^D}$$

Max # of Connected Components in $G_{\mathbf{r}}$ (cont.).

$$\Longrightarrow \begin{pmatrix} \text{\# of components having a vertex in } \{1,2,\ldots,n-2\}^D \end{pmatrix} \le n^D/2^D$$

$$\begin{pmatrix} \text{\# of components not having a vertex in } \{1,2,\ldots,n-2\}^D \end{pmatrix} \le \begin{pmatrix} \text{\# of vertices not in } \{1,2,\ldots,n-2\}^D \end{pmatrix} = n^D - (n-2)^D$$

$$\Longrightarrow (\text{Total \# of components}) \le n^D/2^D + n^D - (n-2)^D.$$

$$\Longrightarrow |A(\mathbf{r})| = 2^{(\text{Total \# of components})} \le 2^{n^D/2^D + n^D - (n-2)^D}$$

$$\Longrightarrow |B_{n\times n\times\ldots\times n}(X)| \le \sum_{\mathbf{r}} |A(\mathbf{r})| \le 2^{Dn^{D-1}} 2^{n^D/2^D + n^D - (n-2)^D}$$

$$\Longrightarrow |B_{n\times n\times\ldots\times n}(X)| \le 2^{n^D/2^D + O(n^{D-1})}$$

Max # of Connected Components in $G_{\mathbf{r}}$ (cont.).

$$\Longrightarrow \begin{pmatrix} \text{\# of components} \\ \text{having a vertex in} \\ \{1,2,\ldots,n-2\}^D \end{pmatrix} \le n^D/2^D$$

$$\begin{pmatrix} \text{\# of components} \\ \text{not having a vertex} \\ \text{in } \{1,2,\ldots,n-2\}^D \end{pmatrix} \le \begin{pmatrix} \text{\# of vertices not in} \\ \{1,2,\ldots,n-2\}^D \end{pmatrix} = n^D-(n-2)^D$$

$$\Longrightarrow (\text{Total \# of components}) \le n^D/2^D + n^D - (n-2)^D.$$

$$\Longrightarrow |A(\mathbf{r})| = 2^{(\text{Total \# of components})} \le 2^{n^D/2^D+n^D-(n-2)^D}$$

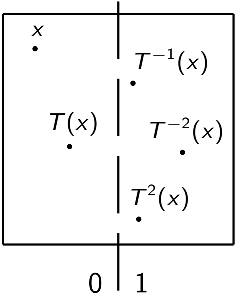
$$\Longrightarrow |B_{n\times n\times\ldots\times n}(X)| \le \sum_{\mathbf{r}} |A(\mathbf{r})| \le 2^{Dn^{D-1}} 2^{n^D/2^D+n^D-(n-2)^D}$$

$$\Longrightarrow |B_{n\times n\times\ldots\times n}(X)| \le 2^{n^D/2^D+O(n^{D-1})}$$

$$\Longrightarrow h(X) \le 1/2^D \quad \square$$

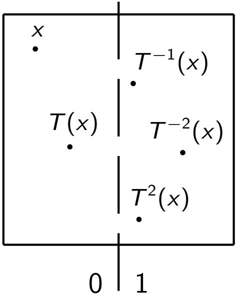
Motivation for 1-dimensional SFT's: Modelling dynamical systems

$T : \Omega \to \Omega$:



Motivation for 1-dimensional SFT's: Modelling dynamical systems

$T : \Omega \to \Omega$:

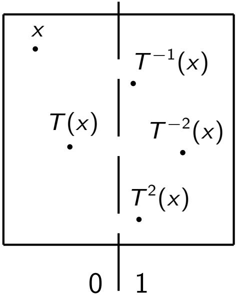


- Represent $x \in \Omega$ by binary itinerary sequence:

$$\begin{array}{ccc} x & \longleftrightarrow & \dots 11.001 \dots \\ T(x) & \longleftrightarrow & \dots 110.01 \dots \end{array}$$

Motivation for 1-dimensional SFT's: Modelling dynamical systems

$T : \Omega \to \Omega$:

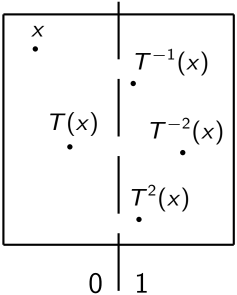


- Represent $x \in \Omega$ by binary itinerary sequence:

$$\begin{array}{ccc} x & \longleftrightarrow & \ldots 11.001 \ldots \\ T(x) & \longleftrightarrow & \ldots 110.01 \ldots \end{array}$$

Motivation for 1-dimensional SFT's: Modelling dynamical systems

$T : \Omega \to \Omega$:



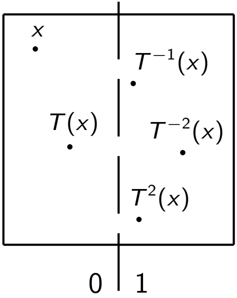
- Represent $x \in \Omega$ by binary itinerary sequence:

$$\begin{array}{ccc} x & \longleftrightarrow & \ldots 11.001 \ldots \\ T(x) & \longleftrightarrow & \ldots 110.01 \ldots \end{array}$$

Ω replaced by an SFT X

Motivation for 1-dimensional SFT's: Modelling dynamical systems

$T : \Omega \to \Omega$:



- Represent $x \in \Omega$ by binary itinerary sequence:

$$\begin{array}{ccc} x & \longleftrightarrow & \ldots 11.001 \ldots \\ T(x) & \longleftrightarrow & \ldots 110.01 \ldots \end{array}$$

Ω replaced by an SFT X
T replaced by the shift mapping.