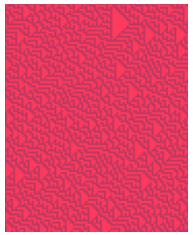
Topological Dynamical Properties in Turing Machines

Rodrigo Torres-Avilés[1]

1. Dpto. de Sistemas de Información, Universidad del Bío-Bío, Chile

AUTOMATA, August 10-12, 2020

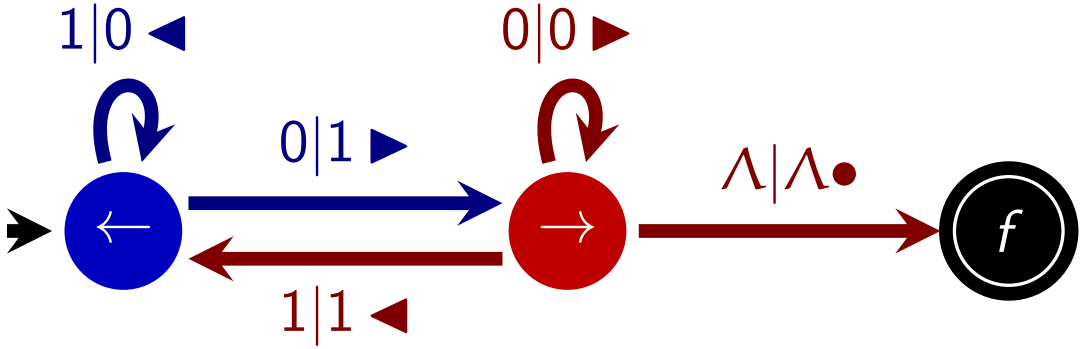


Content

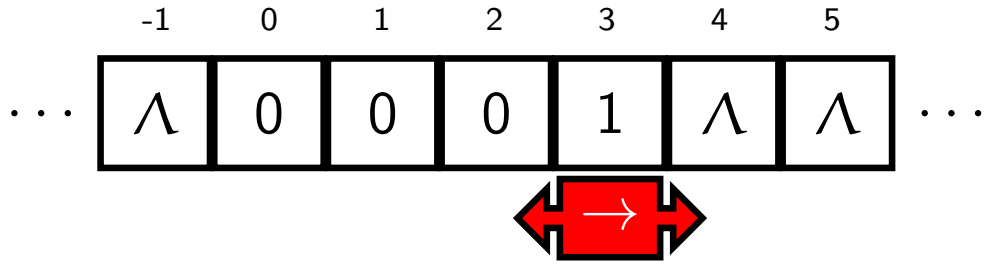
Traditional Turing Machine

Turing machine is a six-tuple: $T = (Q, \Sigma, q_0, F, \Lambda, \delta)$

Partial Transition function: $\delta : Q \times \Sigma \cup \{\Lambda\} \to Q \times \Sigma \cup \{\Lambda\} \times \{\blacktriangleleft, \bullet, \blacktriangleright\}$



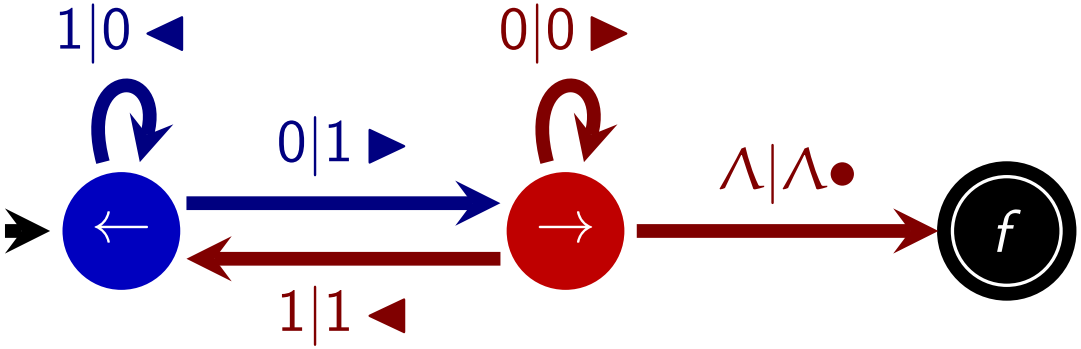
Configuration: $(Q, \mathbb{Z}, (\Sigma \cup \{\Lambda\})^{\mathbb{Z}})$



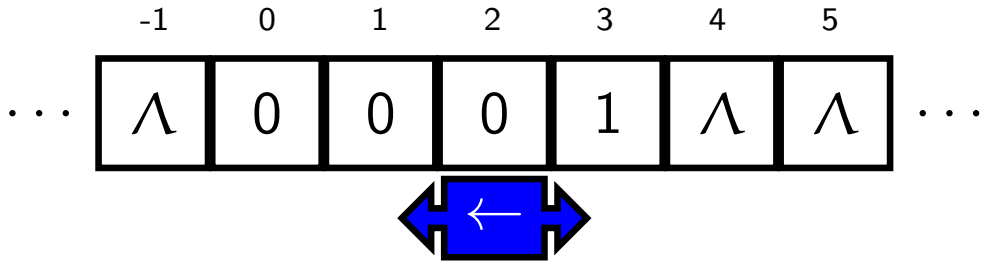
Traditional Turing Machine

Turing machine is a six-tuple: $T = (Q, \Sigma, q_0, F, \Lambda, \delta)$

Partial Transition function: $\delta : Q \times \Sigma \cup \{\Lambda\} \to Q \times \Sigma \cup \{\Lambda\} \times \{\blacktriangleleft, \bullet, \blacktriangleright\}$



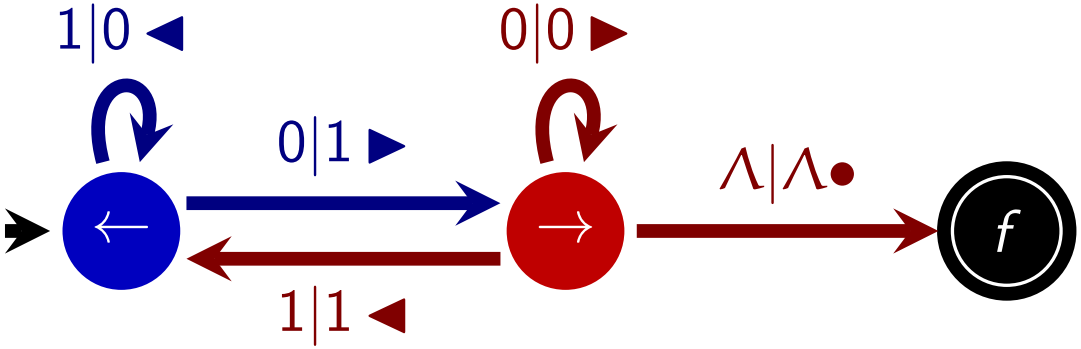
Configuration: $(Q, \mathbb{Z}, (\Sigma \cup \{\Lambda\})^{\mathbb{Z}})$



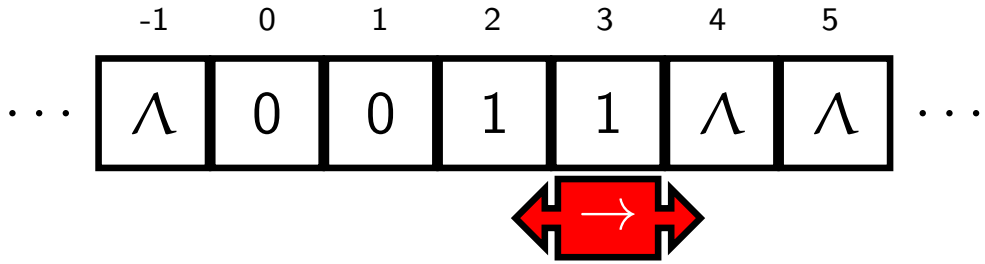
Traditional Turing Machine

Turing machine is a six-tuple: $T = (Q, \Sigma, q_0, F, \Lambda, \delta)$

Partial Transition function: $\delta : Q \times \Sigma \cup \{\Lambda\} \to Q \times \Sigma \cup \{\Lambda\} \times \{\blacktriangleleft, \bullet, \blacktriangleright\}$



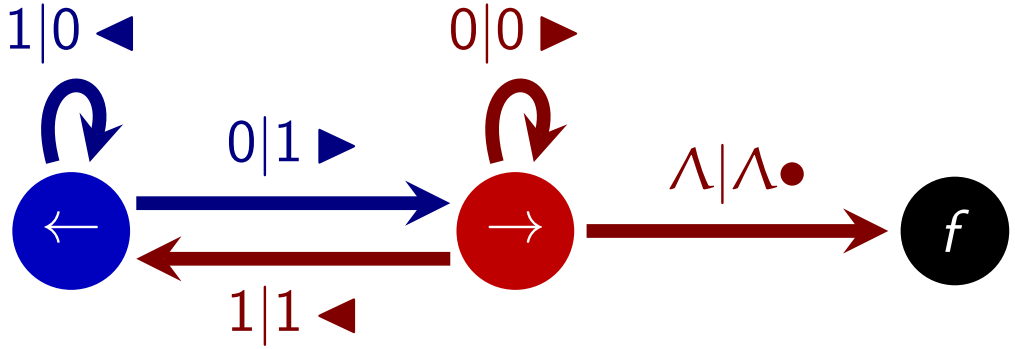
Configuration: $(Q, \mathbb{Z}, (\Sigma \cup \{\Lambda\})^{\mathbb{Z}})$



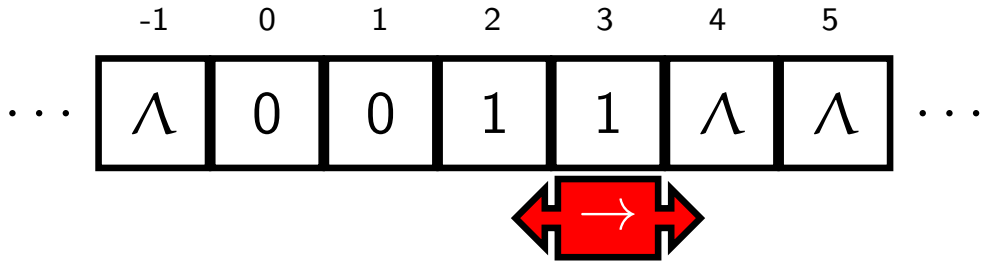
Converting to Turing Machine Dynamical Systems

Turing machine is a **four**-tuple: $T = (Q, \Sigma, \Lambda, \delta)$

Partial Transition function: $\delta : Q \times \Sigma \cup \{\Lambda\} \to Q \times \Sigma \cup \{\Lambda\} \times \{\blacktriangleleft, \bullet, \blacktriangleright\}$



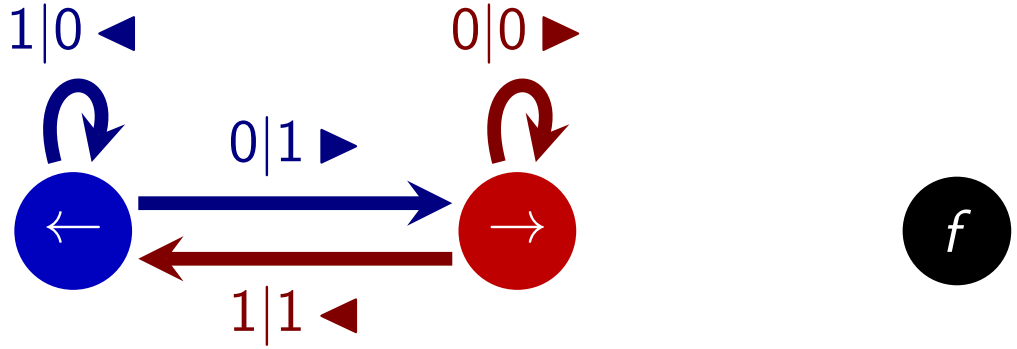
Configuration: $(Q, \mathbb{Z}, (\Sigma \cup \{\Lambda\})^{\mathbb{Z}})$



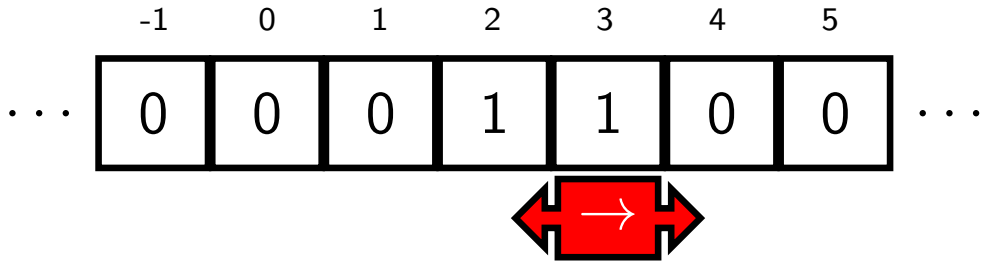
Converting to Turing Machine Dynamical Systems

Turing machine is a **three**-tuple: $T = (Q, \Sigma, \delta)$

Partial Transition function: $\delta : Q \times \Sigma \to Q \times \Sigma \times \{\blacktriangleleft, \bullet, \blacktriangleright\}$



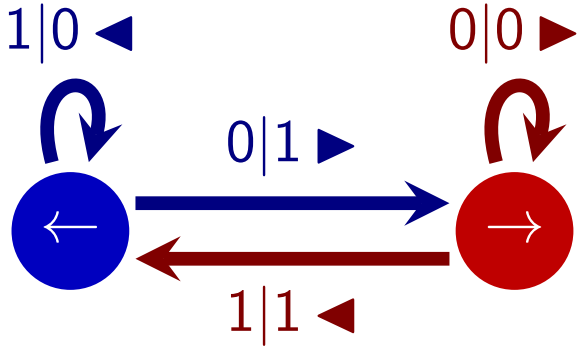
Configuration: $(Q, \mathbb{Z}, \mathbf{\Sigma}^{\mathbb{Z}})$



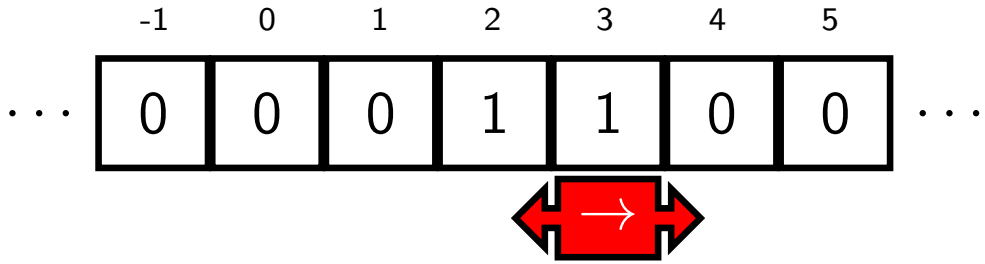
Converting to Turing Machine Dynamical Systems

Complete Turing machine: $T = (Q, \Sigma, \delta)$

Total Transition function: $\delta : Q \times \Sigma \to Q \times \Sigma \times \{\blacktriangleleft, \bullet, \blacktriangleright\}$



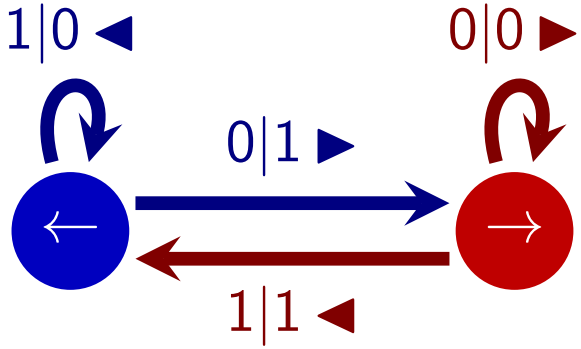
Configuration: $(Q, \mathbb{Z}, \mathbf{\Sigma}^{\mathbb{Z}})$



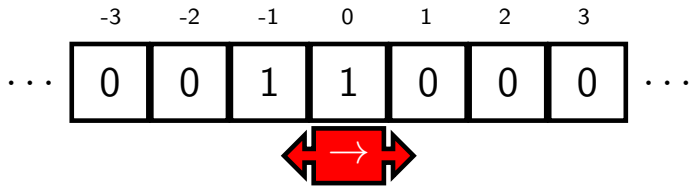
Converting to Turing Machine Dynamical Systems

Complete Turing machine: $T = (Q, \Sigma, \delta)$

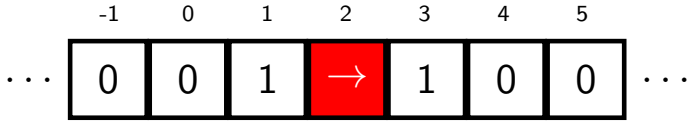
Total Transition function: $\delta : Q \times \Sigma \to Q \times \Sigma \times \{\blacktriangleleft, \bullet, \blacktriangleright\}$



TMT Configuration: $({}^{\omega}\boldsymbol{\Sigma}, \boldsymbol{Q}, \boldsymbol{\Sigma}^{\omega})$



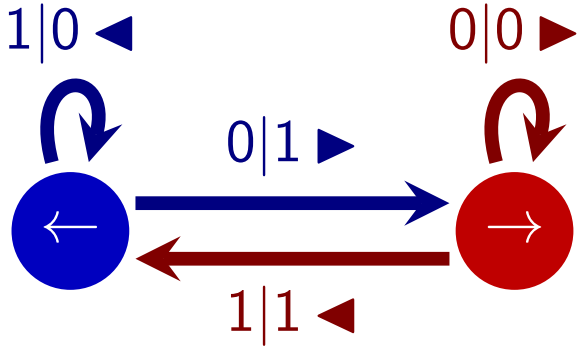
TMH Configuration: $\subset (\boldsymbol{\Sigma} \cup \boldsymbol{Q})^{\mathbb{Z}}$



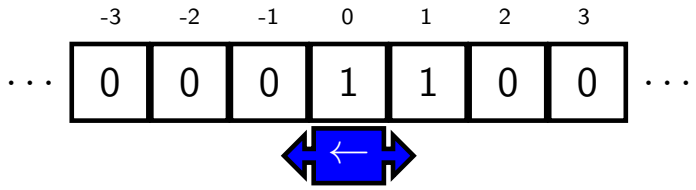
Converting to Turing Machine Dynamical Systems

Complete Turing machine: $T = (Q, \Sigma, \delta)$

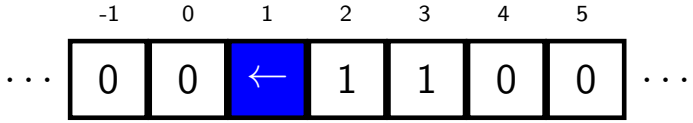
Total Transition function: $\delta : Q \times \Sigma \to Q \times \Sigma \times \{\blacktriangleleft, \bullet, \blacktriangleright\}$



TMT Configuration: $(^{\omega}\boldsymbol{\Sigma}, \boldsymbol{Q}, \boldsymbol{\Sigma}^{\omega})$



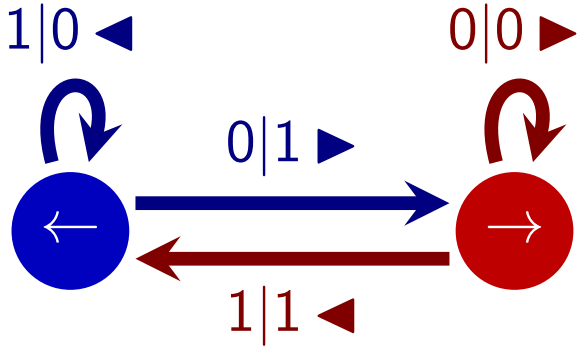
TMH Configuration: $\subset (\boldsymbol{\Sigma} \cup \boldsymbol{Q})^{\mathbb{Z}}$



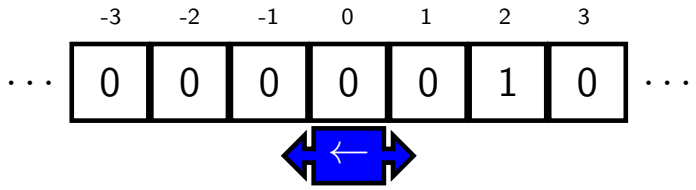
Converting to Turing Machine Dynamical Systems

Complete Turing machine: $T = (Q, \Sigma, \delta)$

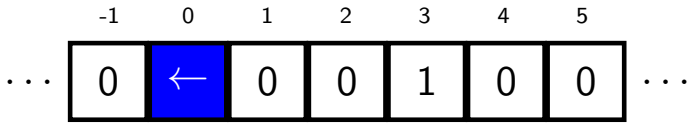
Total Transition function: $\delta : Q \times \Sigma \to Q \times \Sigma \times \{\blacktriangleleft, \bullet, \blacktriangleright\}$



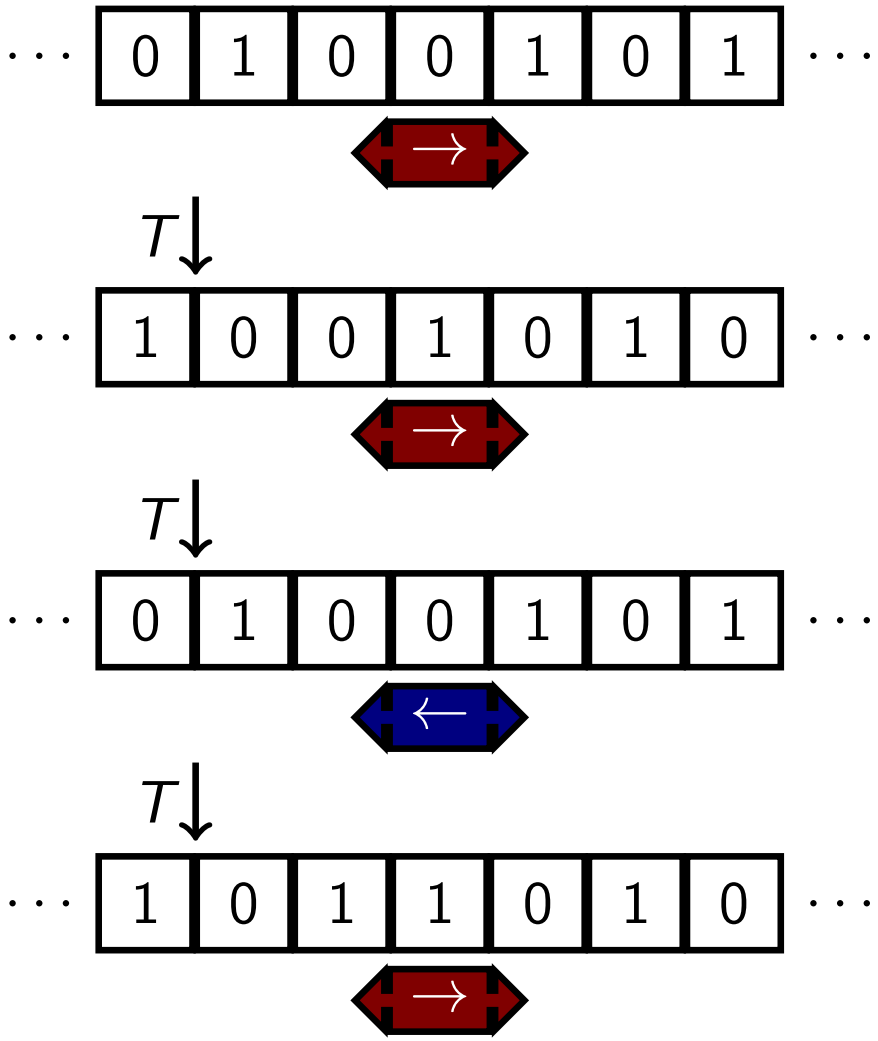
TMT Configuration: $(^{\omega}\boldsymbol{\Sigma}, \boldsymbol{Q}, \boldsymbol{\Sigma}^{\omega})$



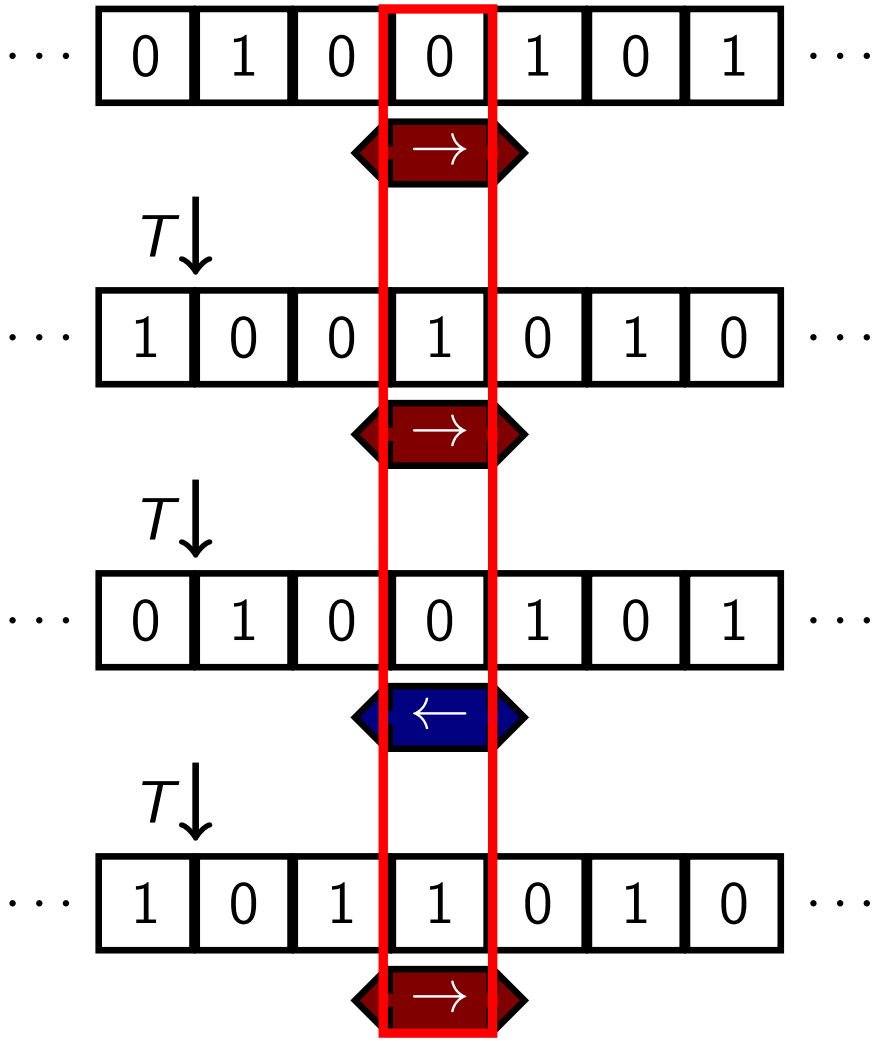
TMH Configuration: $\subset (\boldsymbol{\Sigma} \cup \boldsymbol{Q})^{\mathbb{Z}}$



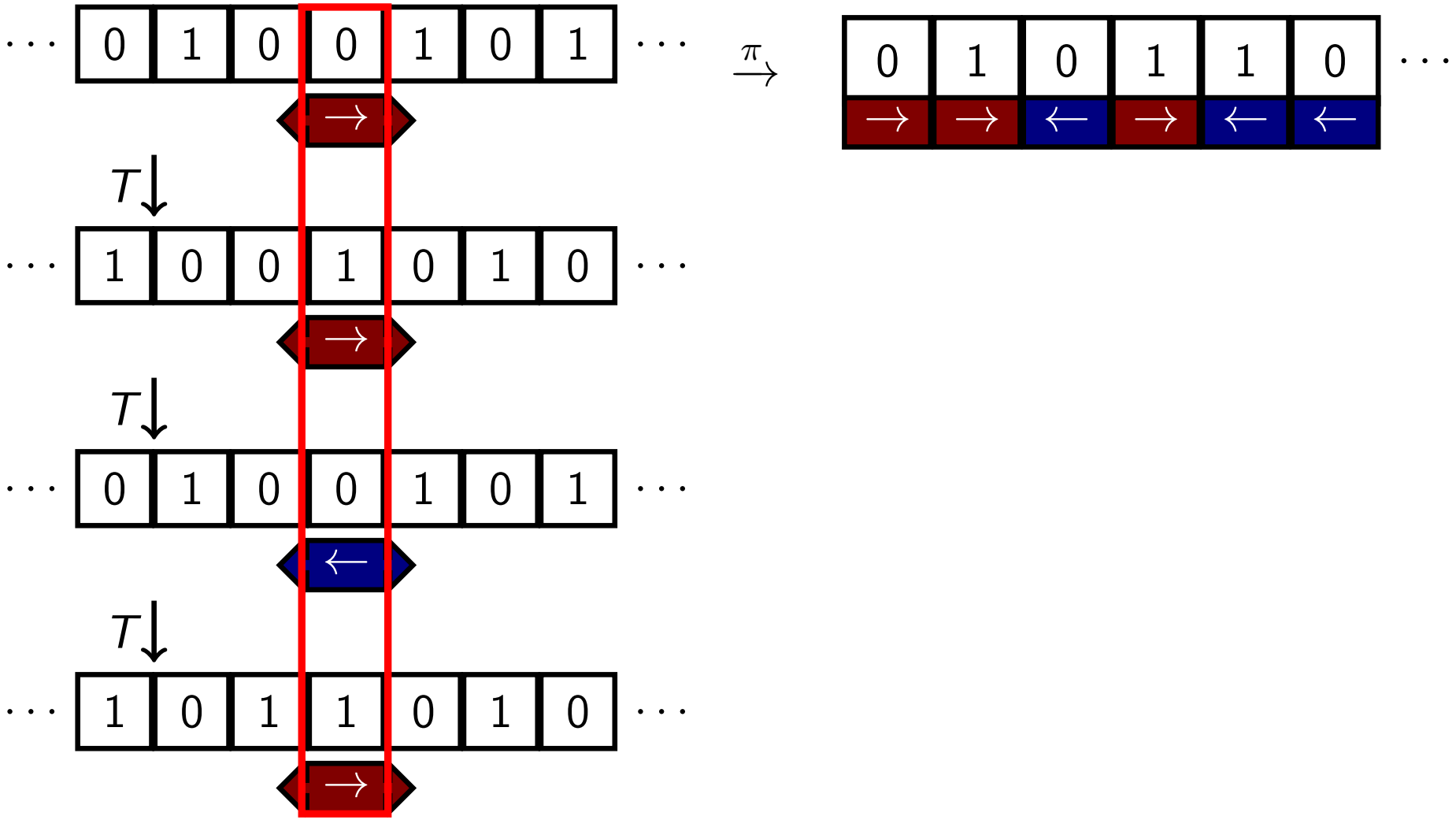
Column Factor of TMT (t-shift)



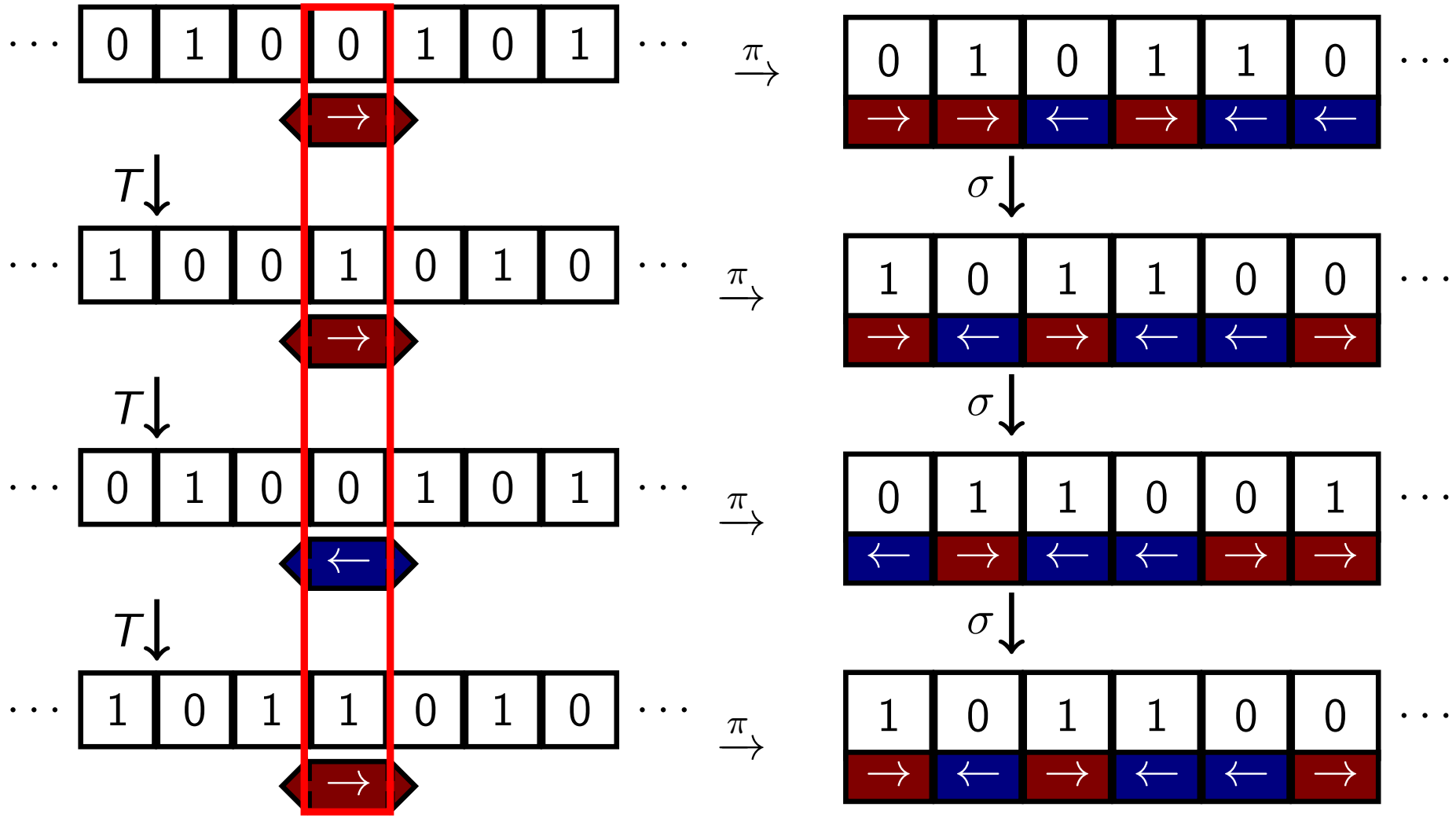
Column Factor of TMT (t-shift)



Column Factor of TMT (t-shift)



Column Factor of TMT (t-shift)



Interesting considerations:

- TMH and TMT [Kůrka '97] exists because $X = (Q, \mathbb{Z}, \Sigma^{\mathbb{Z}})$ is not compact, a serious draw back in order to be consider as topological systems.
- Both can be endowed with the cantor metric, being compact and perfect (no isolated points).
- Column factor of TMH can be consider, and also column factor of more than one cell (not in this talk).
- $\gamma : X \to X_t$ is onto and $\psi : X \to X_h$ is one-to-one.

$$\begin{tikzcd}
X_t \arrow[d, "T_t"] & X \arrow[l, "\gamma"'] \arrow[r, "\psi"] \arrow[d, "T"] & X_h \arrow[d, "T_h"] \\
X_t & X \arrow[l, "\gamma"'] \arrow[r, "\psi"] & X_h
\end{tikzcd}$$

Interesting considerations:

- TMH and TMT [Kůrka '97] exists because $X = (Q, \mathbb{Z}, \Sigma^{\mathbb{Z}})$ is not compact, a serious draw back in order to be consider as topological systems.
- Both can be endowed with the cantor metric, being compact and perfect (no isolated points).
- Column factor of TMH can be consider, and also column factor of more than one cell (not in this talk).
- $\gamma : X \to X_t$ is onto and $\psi : X \to X_h$ is one-to-one.

$$\begin{tikzcd}
X_t \arrow[d, "T_t"] & X \arrow[l, "\gamma"'] \arrow[r, "\psi"] \arrow[d, "T"] & X_h \arrow[d, "T_h"] \\
X_t & X \arrow[l, "\gamma"'] \arrow[r, "\psi"] & X_h
\end{tikzcd}$$

Interesting considerations:

- TMH and TMT [Kůrka '97] exists because $X = (Q, \mathbb{Z}, \Sigma^{\mathbb{Z}})$ is not compact, a serious draw back in order to be consider as topological systems.
- Both can be endowed with the cantor metric, being compact and perfect (no isolated points).
- Column factor of TMH can be consider, and also column factor of more than one cell (not in this talk).
- $\gamma : X \to X_t$ is onto and $\psi : X \to X_h$ is one-to-one.

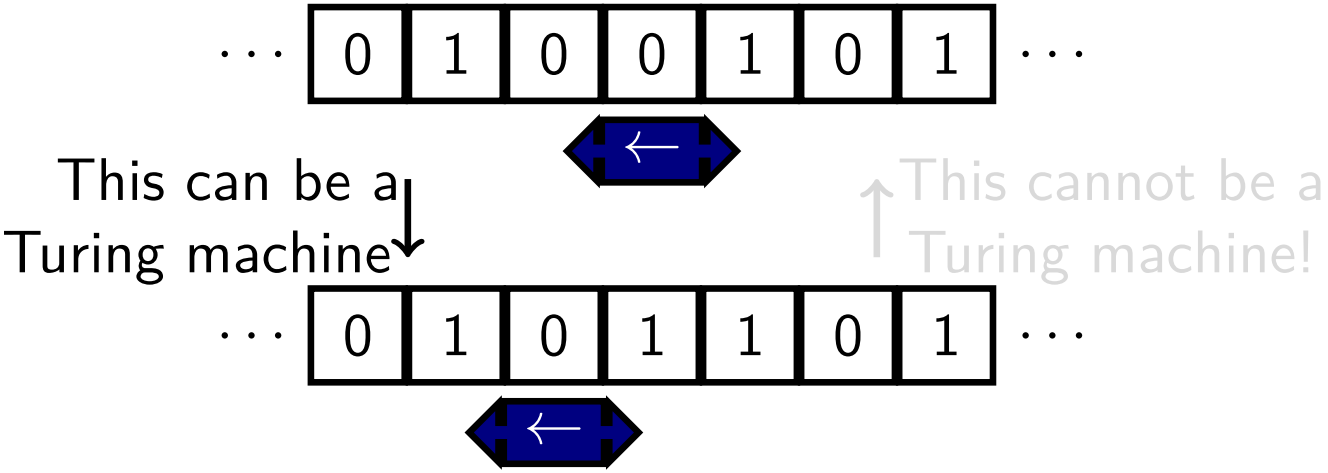
$$\begin{tikzcd}
X_t \arrow[d, "T_t"] & X \arrow[l, "\gamma"'] \arrow[r, "\psi"] \arrow[d, "T"] & X_h \arrow[d, "T_h"] \\
X_t & X \arrow[l, "\gamma"'] \arrow[r, "\psi"] & X_h
\end{tikzcd}$$

Interesting considerations:

- TMH and TMT [Kůrka '97] exists because $X = (Q, \mathbb{Z}, \Sigma^{\mathbb{Z}})$ is not compact, a serious draw back in order to be consider as topological systems.
- Both can be endowed with the cantor metric, being compact and perfect (no isolated points).
- Column factor of TMH can be consider, and also column factor of more than one cell (not in this talk).
- $\gamma : X \to X_t$ is onto and $\psi : X \to X_h$ is one-to-one.

$$\begin{tikzcd}
X_t \arrow[d, "T_t"] & X \arrow[l, "\gamma"'] \arrow[r, "\psi"] \arrow[d, "T"] & X_h \arrow[d, "T_h"] \\
X_t & X \arrow[l, "\gamma"'] \arrow[r, "\psi"] & X_h
\end{tikzcd}$$

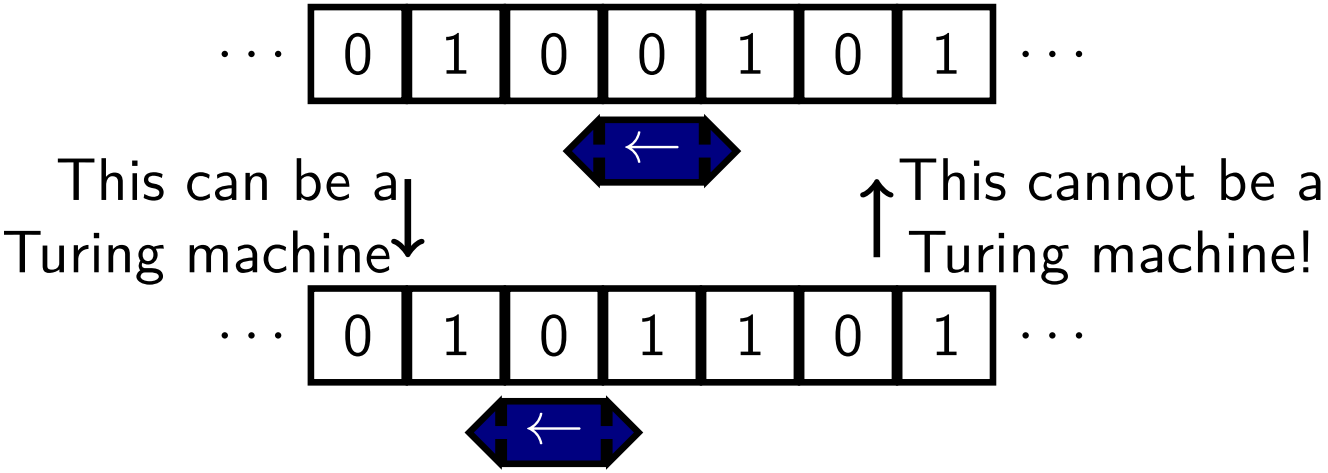
The problem with reversibility.



- Every state of a reversible Turing machine can just be attained from one direction. We consider $\mu : Q \to \{-1, 0, 1\}$.

- If $\varphi(w, i, r) = (w, i - \mu(r), r)$, then: $T^{-1} \circ \varphi \circ T = \varphi$

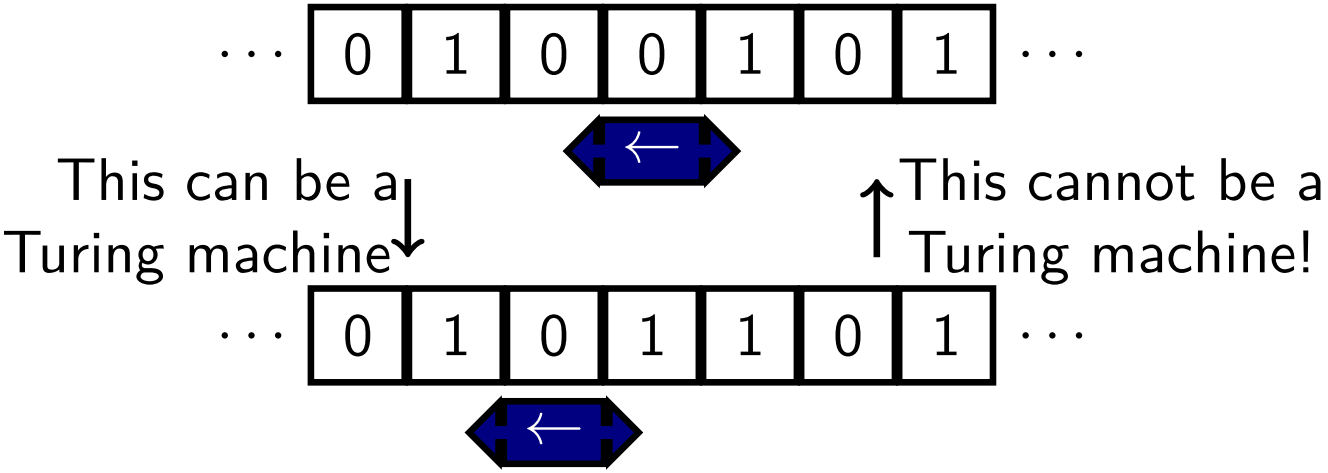
The problem with reversibility.



- Every state of a reversible Turing machine can just be attained from one direction. We consider $\mu : Q \to \{-1, 0, 1\}$.

- If $\varphi(w, i, r) = (w, i - \mu(r), r)$, then: $T^{-1} \circ \varphi \circ T = \varphi$

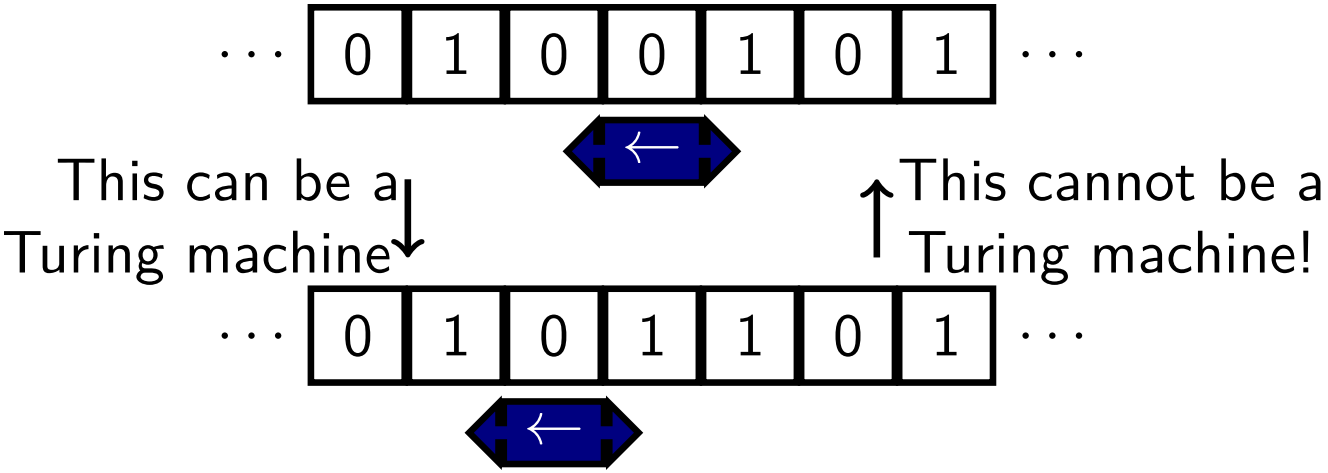
The problem with reversibility.



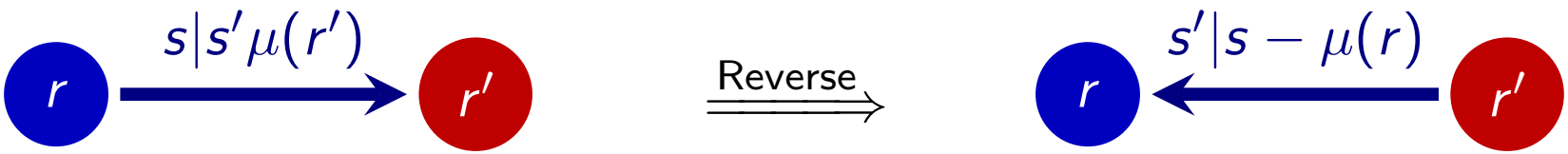
- Every state of a reversible Turing machine can just be attained from one direction. We consider $\mu : Q \to \{-1, 0, 1\}$.

- If $\varphi(w, i, r) = (w, i - \mu(r), r)$, then: $T^{-1} \circ \varphi \circ T = \varphi$

The problem with reversibility.

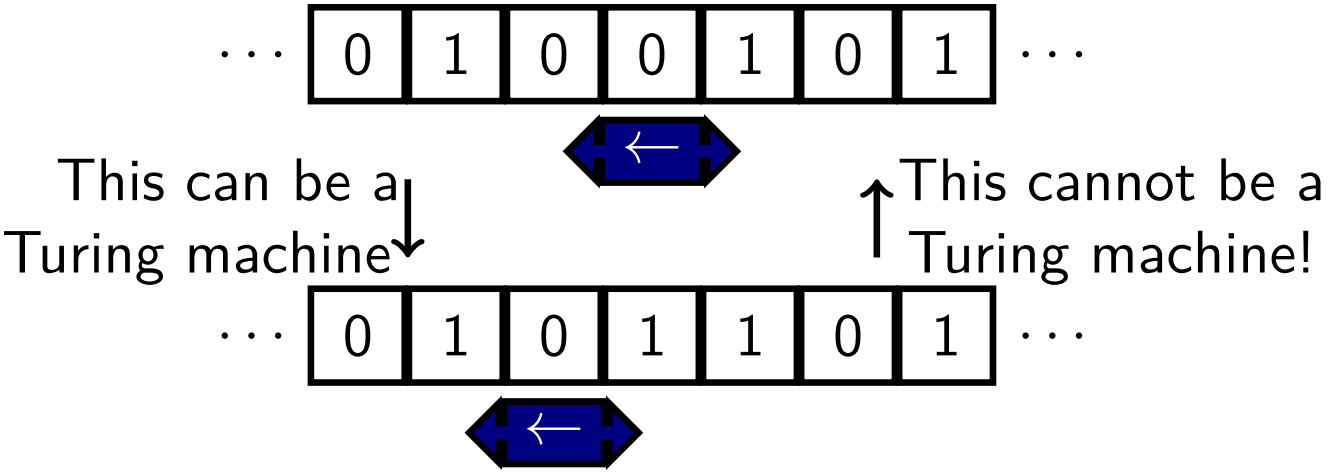


- Every state of a reversible Turing machine can just be attained from one direction. We consider $\mu : Q \to \{-1, 0, 1\}$.



- If $\varphi(w, i, r) = (w, i - \mu(r), r)$, then: $T^{-1} \circ \varphi \circ T = \varphi$

The problem with reversibility.

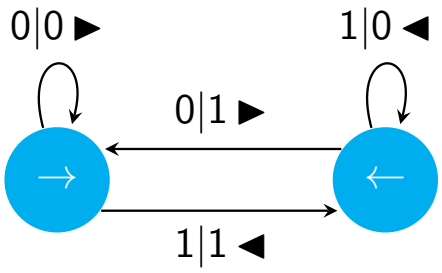


- Every state of a reversible Turing machine can just be attained from one direction. We consider $\mu : Q \to \{-1, 0, 1\}$.

- If $\varphi(w, i, r) = (w, i - \mu(r), r)$, then: $T^{-1} \circ \varphi \circ T = \varphi$

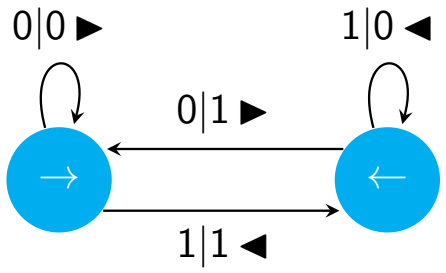
Another way to construct T^{-1}

1.-

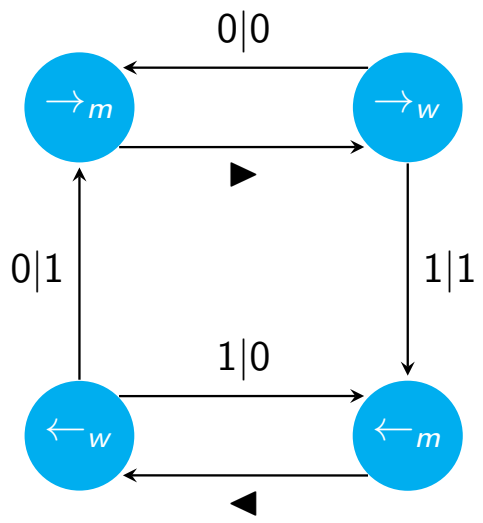


Another way to construct T^{-1}

1.-

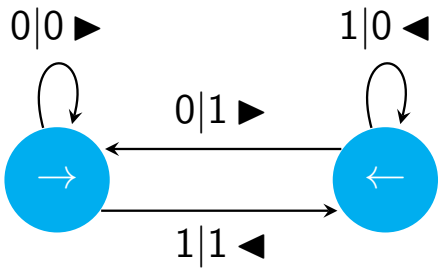


2.-

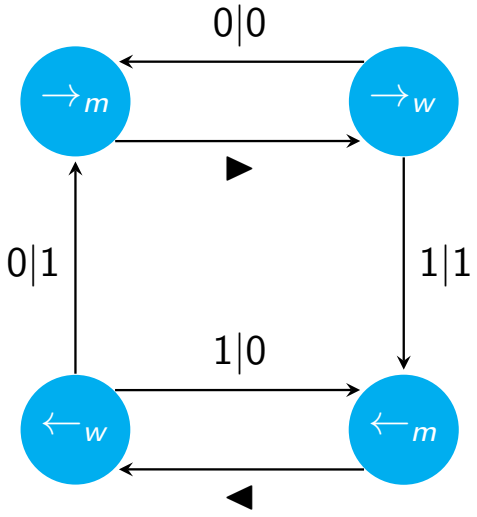


Another way to construct T^{-1}

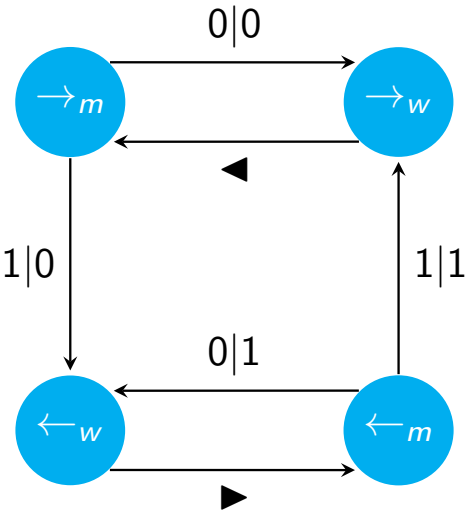
1.-



2.-

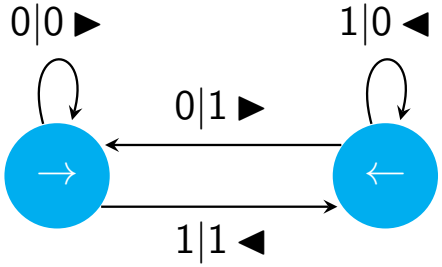


3.-

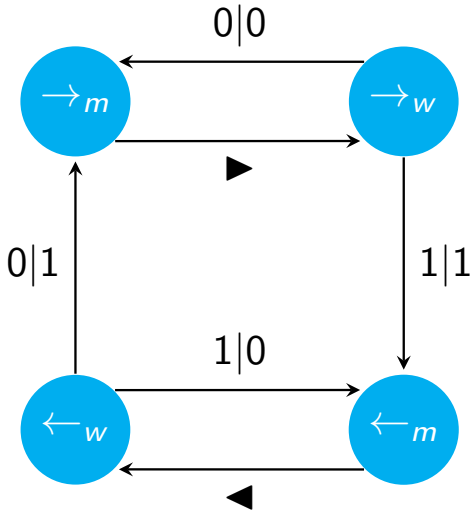


Another way to construct T^{-1}

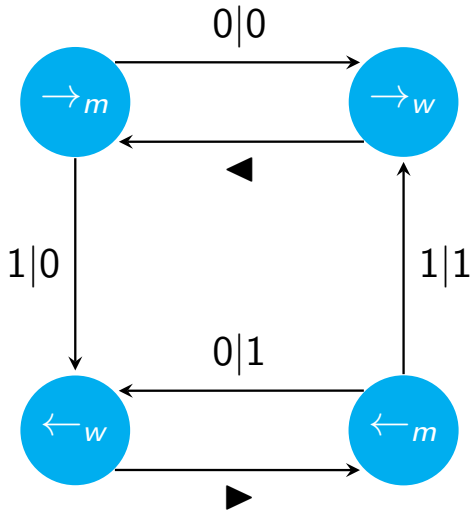
1.-



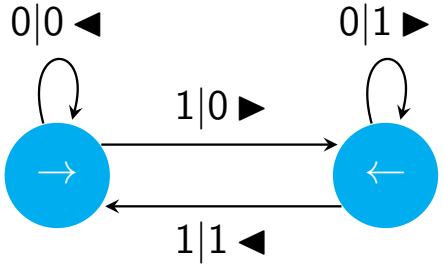
2.-



3.-



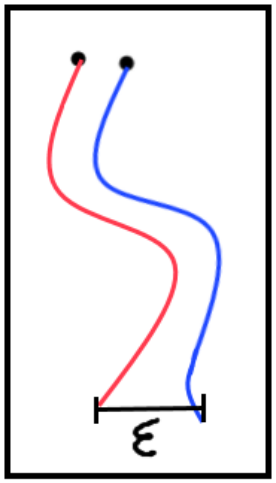
4.-

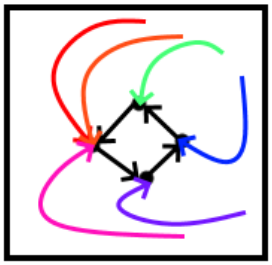


Content

Results in [Kůrka '97]

For every Turing machine, $\exists x \in X$ whose orbit never get the head to position -1. This roughly implies:



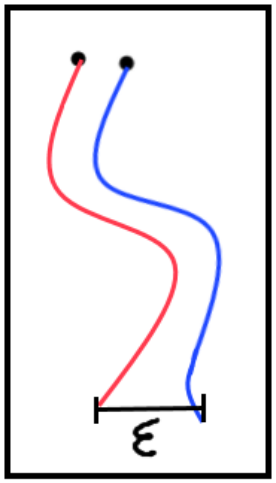


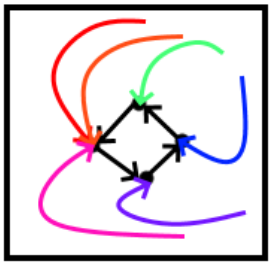
- Nor TMH neither TMT are **positively expansive**[a] $((\exists \epsilon > 0)(\forall x \neq y \in X)(\exists n > 0) d(T^n(x), T^n(y)) > \epsilon)$
- No bijective TMH is **expansive** $((\exists \epsilon > 0)(\forall x \neq y \in X)(\exists n \in Z) d(T^n(x), T^n(y)) > \epsilon)$
- Nor TMH neither TMT have **attracting periodic points**.

[a] Note that P.E. implies sensitivity, but it is possible to have sensitive Turing machines.

Results in [Kůrka '97]

For every Turing machine, $\exists x \in X$ whose orbit never get the head to position -1. This roughly implies:



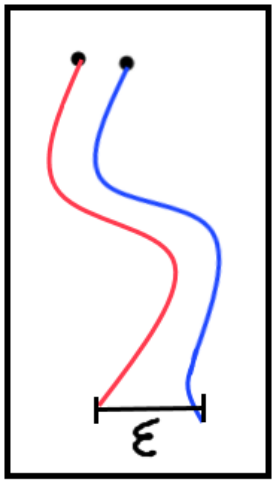


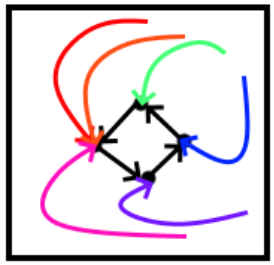
- Nor TMH neither TMT are **positively expansive**[a] $((\exists \epsilon > 0)(\forall x \neq y \in X)(\exists n > 0) d(T^n(x), T^n(y)) > \epsilon)$
- No bijective TMH is **expansive** $((\exists \epsilon > 0)(\forall x \neq y \in X)(\exists n \in Z) d(T^n(x), T^n(y)) > \epsilon)$
- Nor TMH neither TMT have **attracting periodic points**.

[a] Note that P.E. implies sensitivity, but it is possible to have sensitive Turing machines.

Results in [Kůrka '97]

For every Turing machine, $\exists x \in X$ whose orbit never get the head to position -1. This roughly implies:



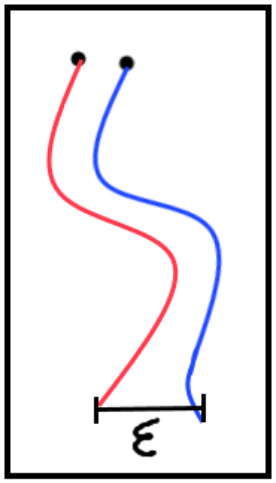


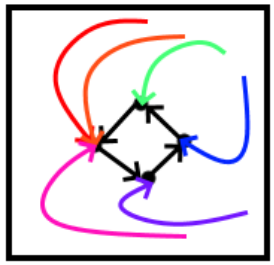
- Nor TMH neither TMT are **positively expansive**[a] $((\exists \epsilon > 0)(\forall x \neq y \in X)(\exists n > 0) d(T^n(x), T^n(y)) > \epsilon)$
- No bijective TMH is **expansive** $((\exists \epsilon > 0)(\forall x \neq y \in X)(\exists n \in Z) d(T^n(x), T^n(y)) > \epsilon)$
- Nor TMH neither TMT have **attracting periodic points**.

[a] Note that P.E. implies sensitivity, but it is possible to have sensitive Turing machines.

Results in [Kůrka '97]

For every Turing machine, $\exists x \in X$ whose orbit never get the head to position -1. This roughly implies:





- Nor TMH neither TMT are **positively expansive**[a] $((\exists \epsilon > 0)(\forall x \neq y \in X)(\exists n > 0) d(T^n(x), T^n(y)) > \epsilon)$
- No bijective TMH is **expansive** $((\exists \epsilon > 0)(\forall x \neq y \in X)(\exists n \in Z) d(T^n(x), T^n(y)) > \epsilon)$
- Nor TMH neither TMT have **attracting periodic points**.

[a] Note that P.E. implies sensitivity, but it is possible to have sensitive Turing machines.

More results in [Kůrka '97]

- Every TMH has **zero topological entropy**.
- It is **conjectured** that every TMT **has a periodic point**.

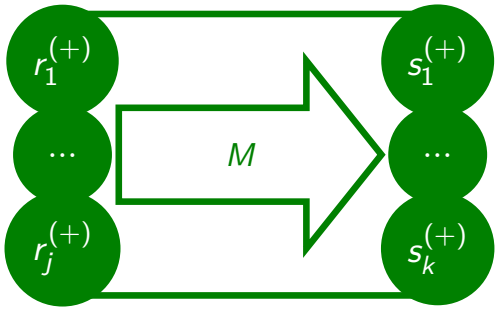
More results in [Kůrka '97]

- Every TMH has **zero topological entropy**.
- It is **conjectured** that every TMT **has a periodic point**.

Content

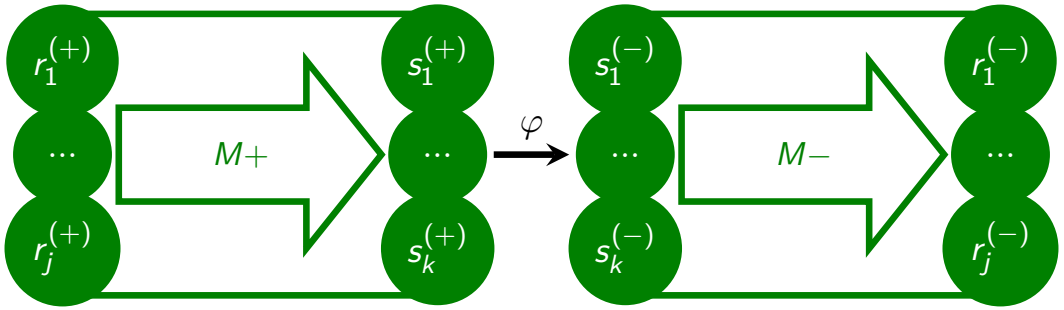
Useful technique to be used: Reversing the time

Consider incomplete and reversible TM M, with j defective pairs (state and symbol not in the image of δ), and k halting states.



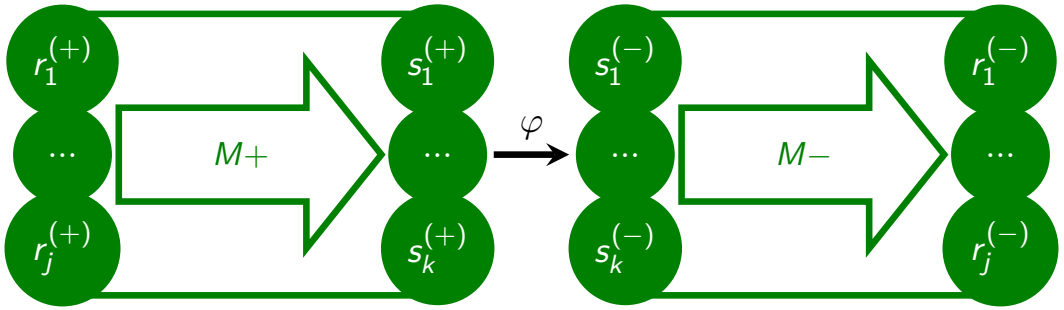
Useful technique to be used: Reversing the time

We attach M^{-1}, connecting every $\delta(s_i, a) = \mathit{undef}$ with $\delta(s_i^{(+)}, a) = (s_i^{(-)}, a, -\mu(s_i))$



Useful technique to be used: Reversing the time

We attach M^{-1}, connecting every $\delta(s_i, a) = undef$ with
$$\delta(s_i^{(+)}, a) = (s_i^{(-)}, a, -\mu(s_i))$$



This machine, called I_M, has the property of being **Innocuous**: If the machine starts at $(r_l^{(+)}, i, w)$, it will halt at $(r_l^{(-)}, i - \mu(r_l), w)$, or compute indefinitely.

Context

- We center on TMT topological entropy.
- It is defined by its t-shift:

$$\mathcal{H}(T_t) = \lim_n \frac{\log |\mathcal{L}_n(S_T)^{\mathrm{i}}|}{n}$$

- An early work [Blondel et al., '04] showed the undecidability of topological entropy of TMT, but considering multiple tapes (based in product between Turing machines).

[i] $\mathcal{L}_n(S) = \{w \in \mathcal{L}(S) : |w| = n\}$

Context

- We center on TMT topological entropy.
- It is defined by its t-shift:

$$\mathcal{H}(T_t) = \lim_n \frac{\log |\mathcal{L}_n(S_T)^{\mathrm{i}}|}{n}$$

- An early work [Blondel et al., '04] showed the undecidability of topological entropy of TMT, but considering multiple tapes (based in product between Turing machines).

[i] $\mathcal{L}_n(S) = \{w \in \mathcal{L}(S) : |w| = n\}$

Context

- We center on TMT topological entropy.
- It is defined by its t-shift:

$$\mathcal{H}(T_t) = \lim_n \frac{\log |\mathcal{L}_n(S_T)^{\mathrm{i}}|}{n}$$

- An early work [Blondel et al., '04] showed the undecidability of topological entropy of TMT, but considering multiple tapes (based in product between Turing machines).

[i] $\mathcal{L}_n(S) = \{w \in \mathcal{L}(S) : |w| = n\}$

Computability of Entropy

- Jeandel [Jeandel '14] proved that topological entropy for one tape TMT can be computable with error $\epsilon > 0$.
- Using results from [Brudno '83]:

  $\exists x \in X : \mathcal{H}(T_t) = \lim_n \frac{K(\mathcal{L}_n(x))}{n}$[ii]
- It is proven that top. entropy is reached as a maximum in the set of configurations $c \in C^+ \subseteq X_t$ such that:
  - $\forall n > 0 : \rho(c, n) > 0$.
  - Every cell is visited a finite amount of time.

[ii] Kolgomorov Complexity.

Computability of Entropy

- Jeandel [Jeandel '14] proved that topological entropy for one tape TMT can be computable with error $\epsilon > 0$.
- Using results from [Brudno '83]:
  $\exists x \in X : \mathcal{H}(T_t) = \lim_n \frac{K(\mathcal{L}_n(x))^{\text{ii}}}{n}$
- It is proven that top. entropy is reached as a maximum in the set of configurations $c \in C^+ \subseteq X_t$ such that:
  - $\forall n > 0 : \rho(c, n) > 0$.
  - Every cell is visited a finite amount of time.

[ii] Kolgomorov Complexity.

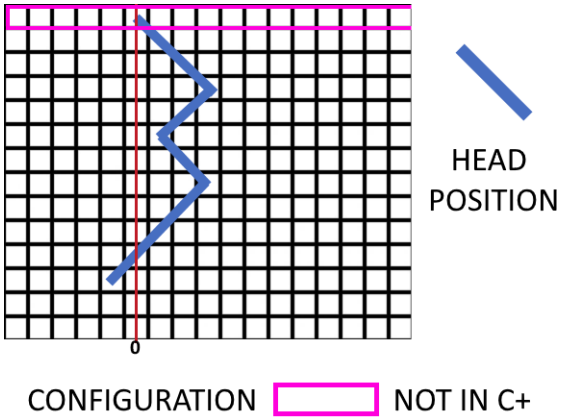
Computability of Entropy

- Jeandel [Jeandel '14] proved that topological entropy for one tape TMT can be computable with error $\epsilon > 0$.
- Using results from [Brudno '83]:
  $\exists x \in X : \mathcal{H}(T_t) = \lim_n \frac{K(\mathcal{L}_n(x))}{n}$[ii]
- It is proven that top. entropy is reached as a maximum in the set of configurations $c \in C^+ \subseteq X_t$ such that:
  - $\forall n > 0 : \rho(c, n) > 0$.
  - Every cell is visited a finite amount of time.

[ii] Kolgomorov Complexity.

Computability of Entropy

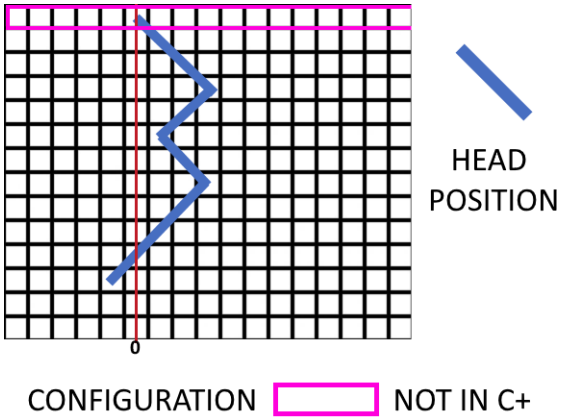
- Jeandel [Jeandel '14] proved that topological entropy for one tape TMT can be computable with error $\epsilon > 0$.
- Using results from [Brudno '83]:
  $\exists x \in X : \mathcal{H}(T_t) = \lim_n \frac{K(\mathcal{L}_n(x))}{n}$[ii]
- It is proven that top. entropy is reached as a maximum in the set of configurations $c \in C^+ \subseteq X_t$ such that:
  - $\forall n > 0 : \rho(c, n) > 0$.
  - Every cell is visited a finite amount of time.



[ii] Kolgomorov Complexity.

Computability of Entropy

- Jeandel [Jeandel '14] proved that topological entropy for one tape TMT can be computable with error $\epsilon > 0$.
- Using results from [Brudno '83]:
  $\exists x \in X : \mathcal{H}(T_t) = \lim_n \frac{K(\mathcal{L}_n(x))}{n}$[ii]
- It is proven that top. entropy is reached as a maximum in the set of configurations $c \in C^+ \subseteq X_t$ such that:
  - $\forall n > 0 : \rho(c, n) > 0$.
  - Every cell is visited a finite amount of time.



[ii] Kolgomorov Complexity.

How to compute the entropy

- To calculate entropy considering first n cells, it is constructed a Finite Automaton considering all possible paths in C^+ (finite) and calculate its (Kolgomorov) complexity.
- This finite automaton has the *diamond property*: given two vertices a, b and a word u, there is at most one path from a to b labeled by u.
- Based in the work of Brudno, it is possible to calculate a finite automaton complexity.
- Also, Given Kolmogorov complexity properties, it is possible to approximate $\mathcal{H}(T_t)$ from above.

How to compute the entropy

- To calculate entropy considering first n cells, it is constructed a Finite Automaton considering all possible paths in C^+ (finite) and calculate its (Kolgomorov) complexity.
- This finite automaton has the *diamond property*: given two vertices a, b and a word u, there is at most one path from a to b labeled by u.
- Based in the work of Brudno, it is possible to calculate a finite automaton complexity.
- Also, Given Kolmogorov complexity properties, it is possible to approximate $\mathcal{H}(T_t)$ from above.

How to compute the entropy

- To calculate entropy considering first n cells, it is constructed a Finite Automaton considering all possible paths in C^+ (finite) and calculate its (Kolgomorov) complexity.
- This finite automaton has the *diamond property*: given two vertices a, b and a word u, there is at most one path from a to b labeled by u.
- Based in the work of Brudno, it is possible to calculate a finite automaton complexity.
- Also, Given Kolmogorov complexity properties, it is possible to approximate $\mathcal{H}(T_t)$ from above.

How to compute the entropy

- To calculate entropy considering first n cells, it is constructed a Finite Automaton considering all possible paths in C^+ (finite) and calculate its (Kolgomorov) complexity.
- This finite automaton has the *diamond property*: given two vertices a, b and a word u, there is at most one path from a to b labeled by u.
- Based in the work of Brudno, it is possible to calculate a finite automaton complexity.
- Also, Given Kolmogorov complexity properties, it is possible to approximate $\mathcal{H}(T_t)$ from above.

But then, is it possible to decide if $\mathcal{H}(T_t) = 0$?

Theorem (Gajardo, Ollinger, TA '15)

To decide if a (reversible and) complete Turing machine has a entropy of 0 *is undecidable.*

Proof Sketch

- A simple simulation of a 2 Reversible Counter Machine C in the right side of the tape is done:
  - Input $<_s 1^n|1^m>$ represents configuration (s, n, m) of C.
  - For a successful computation, we need enough work space (represented by symbols 1 in the tape).
- We distinguish 4 types of configurations:
  - Configurations with garbage in the tape.
  - Configurations with unbounded searches of symbols: $|, >, <_s$.
  - Configurations with successful infinite computation of C.
  - Configurations with successful finite computation of C.
- If the computation finishes, we allow to initiate a new computation starting with empty counters, searching to the right the $|$ symbol.

Proof Sketch

- A simple simulation of a 2 Reversible Counter Machine C in the right side of the tape is done:
  - Input $<_s 1^n|1^m>$ represents configuration (s, n, m) of C.
  - For a successful computation, we need enough work space (represented by symbols 1 in the tape).
- We distinguish 4 types of configurations:
  - Configurations with garbage in the tape.
  - Configurations with unbounded searches of symbols: $|, >, <_s$.
  - Configurations with successful infinite computation of C.
  - Configurations with successful finite computation of C.
- If the computation finishes, we allow to initiate a new computation starting with empty counters, searching to the right the $|$ symbol.

Proof Sketch

- A simple simulation of a 2 Reversible Counter Machine C in the right side of the tape is done:
  - Input $<_s 1^n|1^m>$ represents configuration (s, n, m) of C.
  - For a successful computation, we need enough work space (represented by symbols 1 in the tape).
- We distinguish 4 types of configurations:
  - Configurations with garbage in the tape.
  - Configurations with unbounded searches of symbols: $|, >, <_s$.
  - Configurations with successful infinite computation of C.
  - Configurations with successful finite computation of C.
- If the computation finishes, we allow to initiate a new computation starting with empty counters, searching to the right the $|$ symbol.

Proof Sketch

- Then, we use **Reversing the Time**.
- Configurations with garbage will generate periodic points, only seeing a finite amount of tape.
- Unbounded searches generate periodic points in TMT, but only seeing one symbol.
- Infinite correct computations will visit cell 0 infinitely often.
- Therefore, the unique configuration with entropy greater than 0 exists if C with empty counters halts:

$$x = ({}^{\omega}1, q_0, <_{s_0} \mid > 1^{n_1}|1^{n_2}|1^{n_3}|...)$$

Proof Sketch

- Then, we use **Reversing the Time**.
- Configurations with garbage will generate periodic points, only seeing a finite amount of tape.
- Unbounded searches generate periodic points in TMT, but only seeing one symbol.
- Infinite correct computations will visit cell 0 infinitely often.
- Therefore, the unique configuration with entropy greater than 0 exists if C with empty counters halts:

$$x = ({}^{\omega}1, q_0, <_{s_0} | > 1^{n_1}|1^{n_2}|1^{n_3}|...)$$

Proof Sketch

- Then, we use **Reversing the Time**.
- Configurations with garbage will generate periodic points, only seeing a finite amount of tape.
- Unbounded searches generate periodic points in TMT, but only seeing one symbol.
- Infinite correct computations will visit cell 0 infinitely often.
- Therefore, the unique configuration with entropy greater than 0 exists if C with empty counters halts:

$$x = ({}^{\omega}1, q_0, <_{s_0} | > 1^{n_1}|1^{n_2}|1^{n_3}|...)$$

Proof Sketch

- Then, we use **Reversing the Time**.
- Configurations with garbage will generate periodic points, only seeing a finite amount of tape.
- Unbounded searches generate periodic points in TMT, but only seeing one symbol.
- Infinite correct computations will visit cell 0 infinitely often.
- Therefore, the unique configuration with entropy greater than 0 exists if C with empty counters halts:

$$x = ({}^{\omega}1, q_0, <_{s_0} | > 1^{n_1}|1^{n_2}|1^{n_3}|...)$$

Proof Sketch

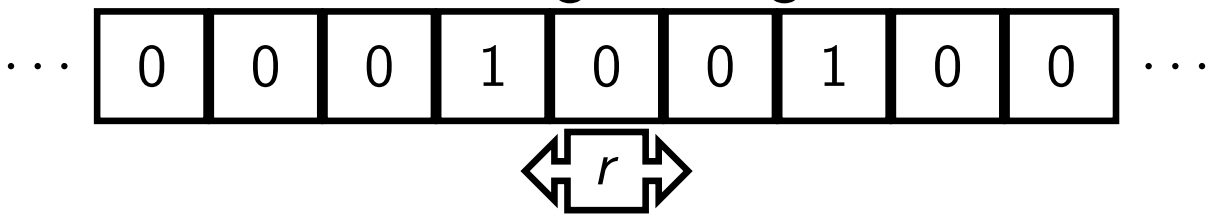
- Then, we use **Reversing the Time**.
- Configurations with garbage will generate periodic points, only seeing a finite amount of tape.
- Unbounded searches generate periodic points in TMT, but only seeing one symbol.
- Infinite correct computations will visit cell 0 infinitely often.
- Therefore, the unique configuration with entropy greater than 0 exists if C with empty counters halts:

$$x = ({}^{\omega}1, q_0, <_{s_0} \mid > 1^{n_1} | 1^{n_2} | 1^{n_3} | ...)$$

Content

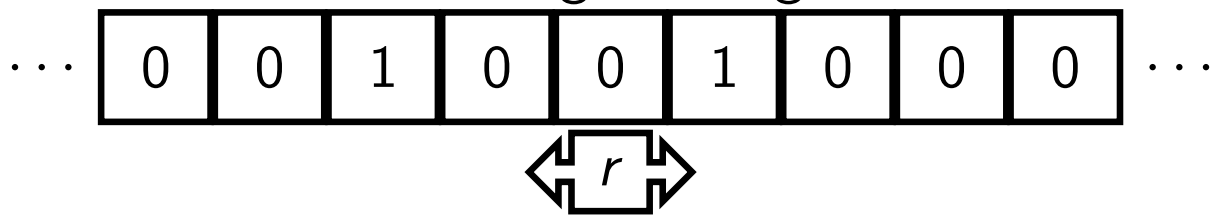
How to aperiodic: Idea

Idea: go the right.



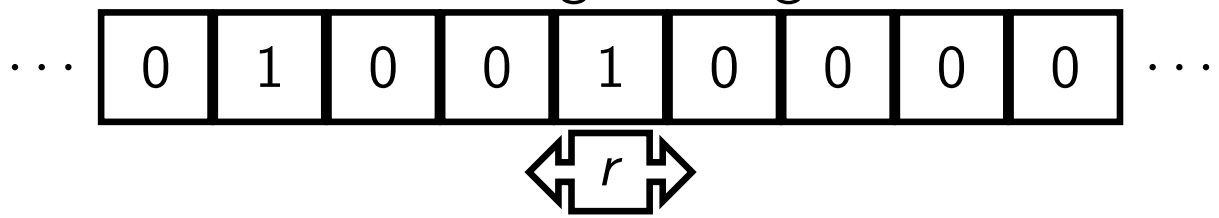
How to aperiodic: Idea

Idea: go the right.



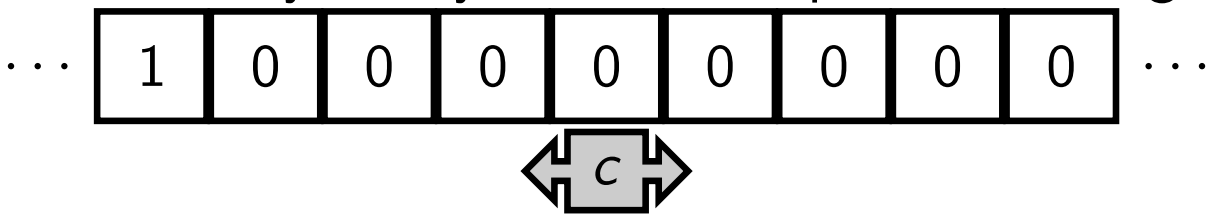
How to aperiodic: Idea

Idea: go the right.



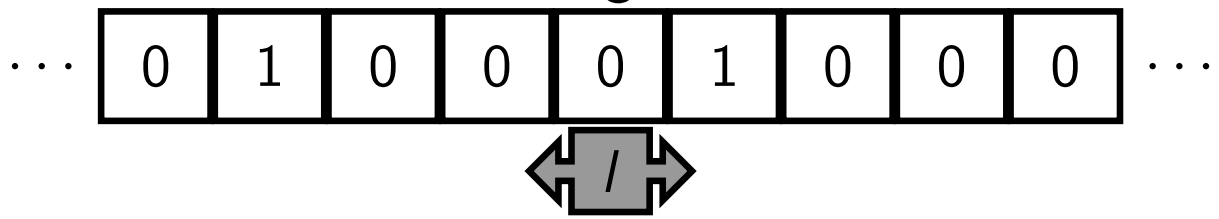
How to aperiodic: Idea

Idea: Carry the symbol 1 one space to the right.



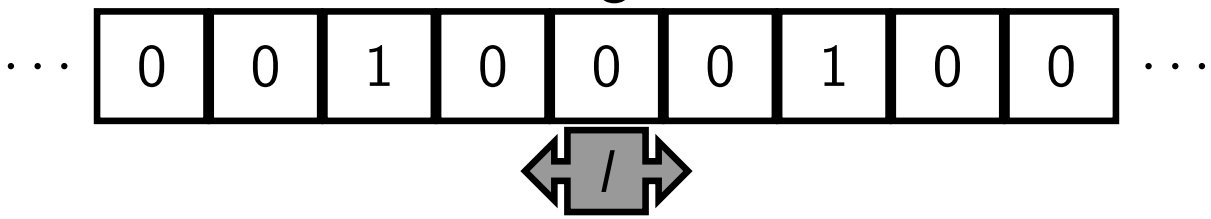
How to aperiodic: Idea

Idea: Returning to 1 at the left.



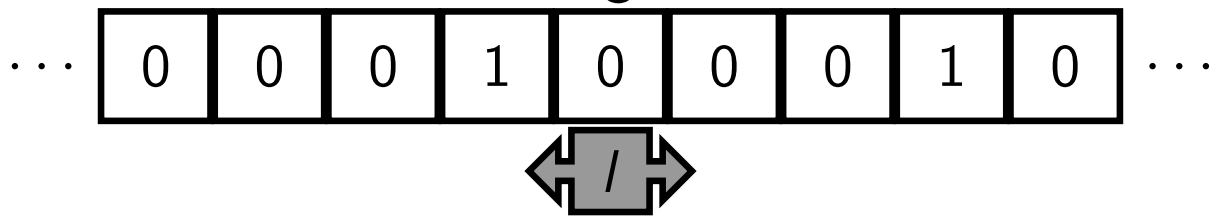
How to aperiodic: Idea

Idea: Returning to 1 at the left.



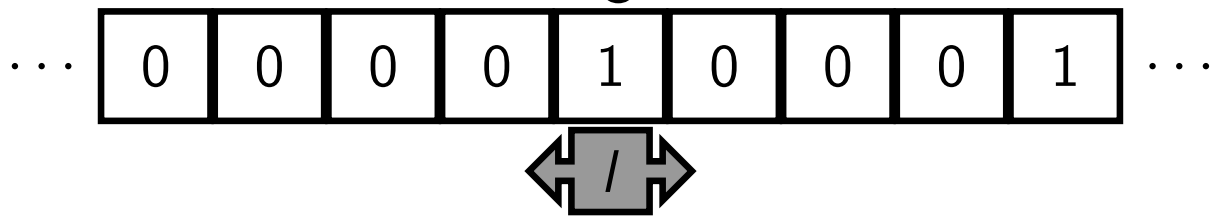
How to aperiodic: Idea

Idea: Returning to 1 at the left.



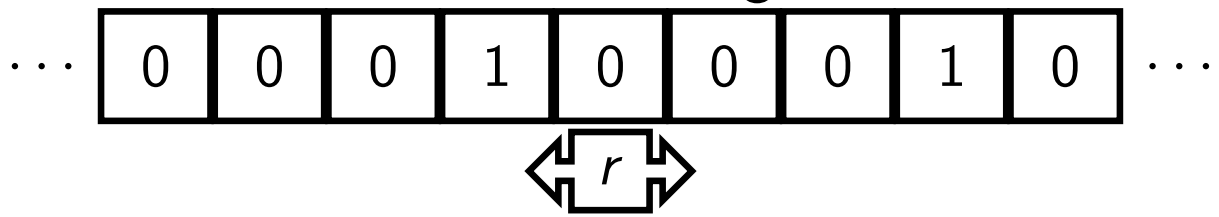
How to aperiodic: Idea

Idea: Returning to 1 at the left.



How to aperiodic: Idea

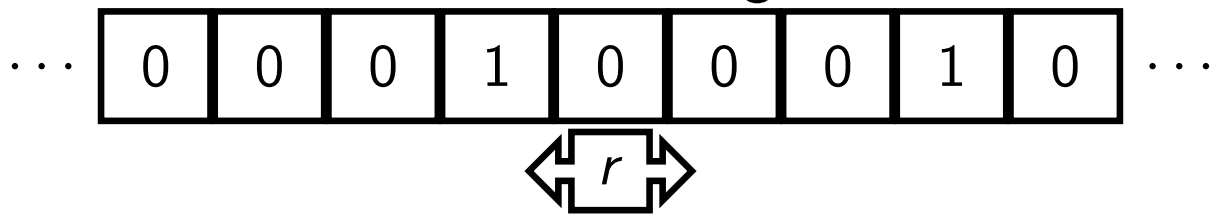
Idea: Start again.



How to aperiodic: Idea

Problem: If configuration has no 1 to the left or right.

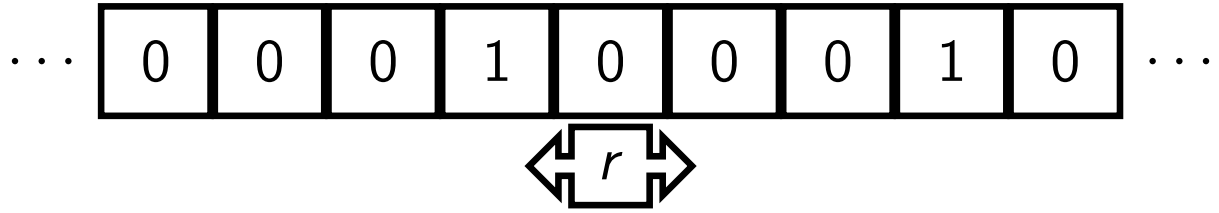
Idea: Start again.



How to Aperiodic: Solution

Hopper Call! Roughly speaking, this make a bounded call inside the search space, pinging a 1 to the right, leaving a 1 in the original position.

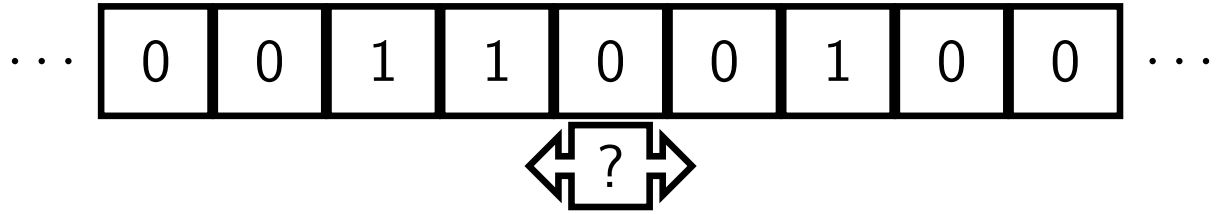
Start a Hopper Call.



How to Aperiodic: Solution

Hopper Call! Roughly speaking, this make a bounded call inside the search space, pinging a 1 to the right, leaving a 1 in the original position.

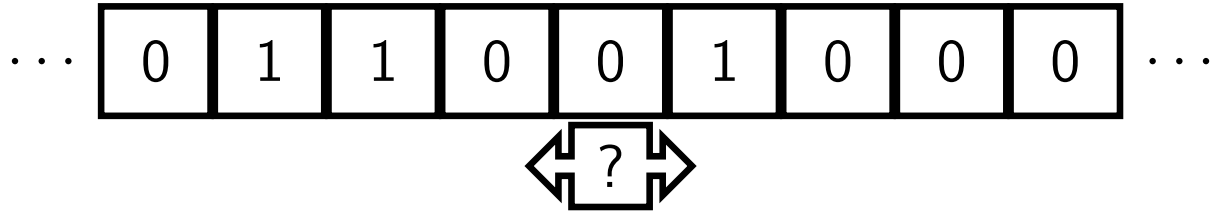
Start a Hopper Call.



How to Aperiodic: Solution

Hopper Call! Roughly speaking, this make a bounded call inside the search space, pinging a 1 to the right, leaving a 1 in the original position.

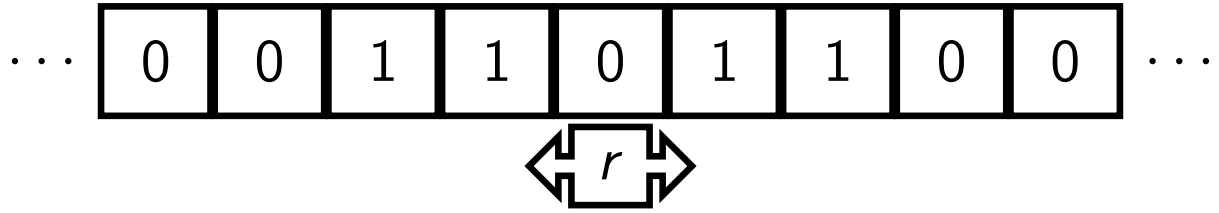
Start a Hopper Call.



How to Aperiodic: Solution

Hopper Call! Roughly speaking, this make a bounded call inside the search space, pinging a 1 to the right, leaving a 1 in the original position.

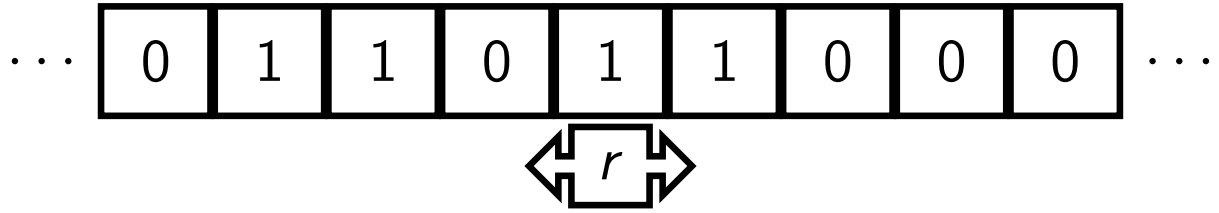
Work normally.



How to Aperiodic: Solution

Hopper Call! Roughly speaking, this make a bounded call inside the search space, pinging a 1 to the right, leaving a 1 in the original position.

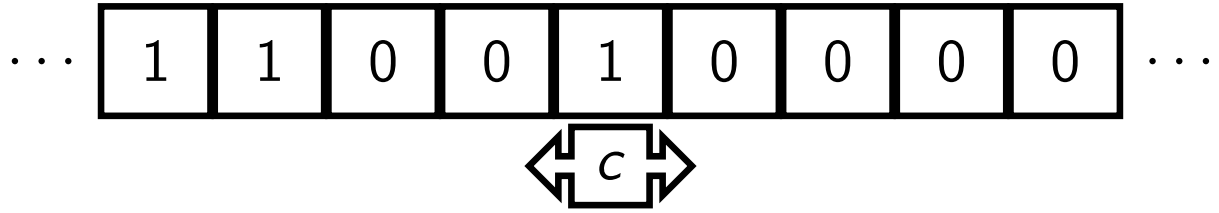
Work normally.



How to Aperiodic: Solution

Hopper Call! Roughly speaking, this make a bounded call inside the search space, pinging a 1 to the right, leaving a 1 in the original position.

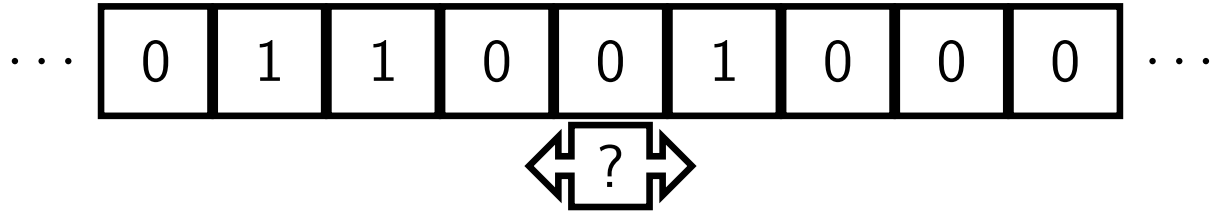
And when you have no more space...



How to Aperiodic: Solution

Hopper Call! Roughly speaking, this make a bounded call inside the search space, pinging a 1 to the right, leaving a 1 in the original position.

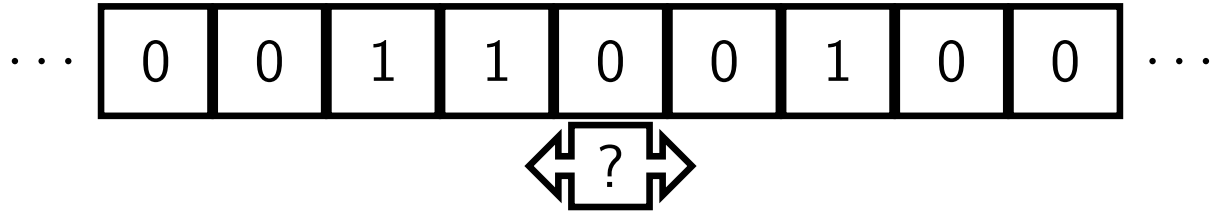
Return the Hooper Call.



How to Aperiodic: Solution

Hopper Call! Roughly speaking, this make a bounded call inside the search space, pinging a 1 to the right, leaving a 1 in the original position.

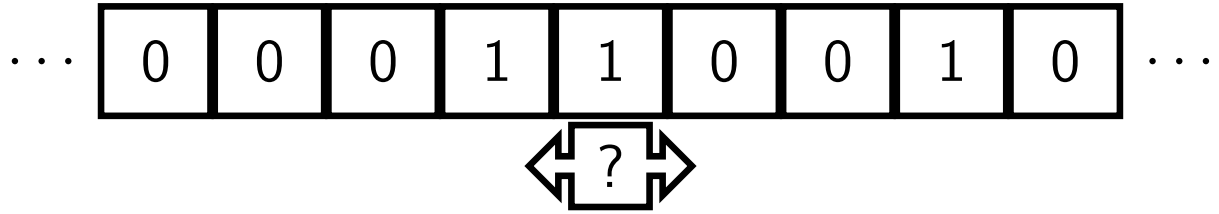
Return the Hooper Call.



How to Aperiodic: Solution

Hopper Call! Roughly speaking, this make a bounded call inside the search space, pinging a 1 to the right, leaving a 1 in the original position.

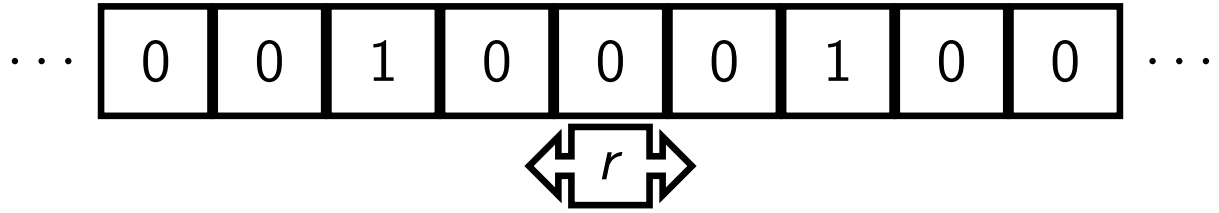
Return the Hooper Call.



How to Aperiodic: Solution

Hopper Call! Roughly speaking, this make a bounded call inside the search space, pinging a 1 to the right, leaving a 1 in the original position.

And you have advanced one space.



Hooper's Call

- It was introduced by [Hooper '66], to prove the undecidability of wheter a Turing machine was **Mortal**.
- An incomplete Turing machine is called **Mortal** if it halts in every possible configuration[iii].
- This study was further completed by [Kari, Ollinger '08], proving undecidability of mortality in reversible Turing machines, and later on by [Jeandel '12], by characterizing immortal configurations (which does not halt).

[iii] Not to confuse with the totality problem, where it is required to halt in every *finite* input over a tape full of blank symbols.

Hooper's Call

- It was introduced by [Hooper '66], to prove the undecidability of wheter a Turing machine was **Mortal**.
- An incomplete Turing machine is called **Mortal** if it halts in every possible configuration[iii].
- This study was further completed by [Kari, Ollinger '08], proving undecidability of mortality in reversible Turing machines, and later on by [Jeandel '12], by characterizing immortal configurations (which does not halt).

[iii] Not to confuse with the totality problem, where it is required to halt in every *finite* input over a tape full of blank symbols.

Hooper's Call

- It was introduced by [Hooper '66], to prove the undecidability of wheter a Turing machine was **Mortal**.
- An incomplete Turing machine is called **Mortal** if it halts in every possible configuration[iii].
- This study was further completed by [Kari, Ollinger '08], proving undecidability of mortality in reversible Turing machines, and later on by [Jeandel '12], by characterizing immortal configurations (which does not halt).

[iii] Not to confuse with the totality problem, where it is required to halt in every *finite* input over a tape full of blank symbols.

Disproving Kůrka's conjecture

Theorem (Blondel, Cassaigne, Nichitu '02)

There exists a complete Turing machine aperiodic in TMT.

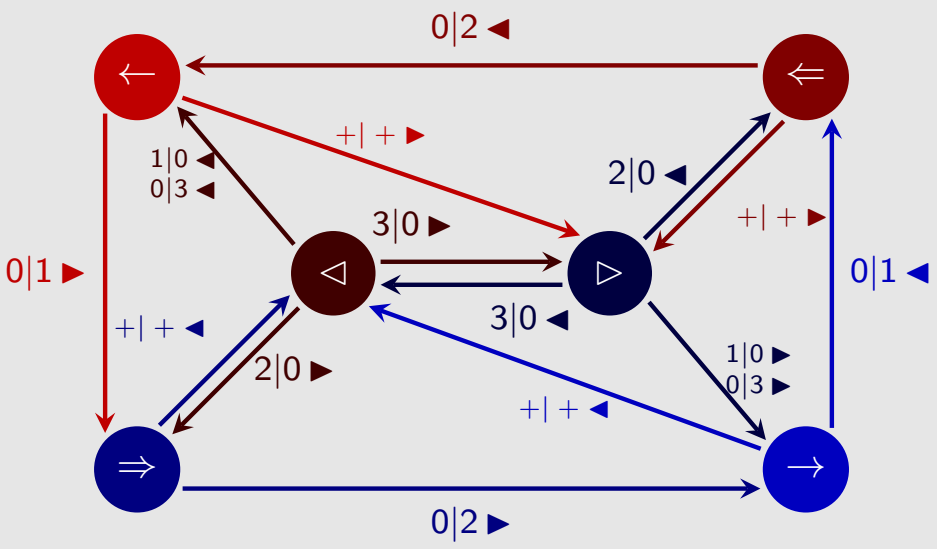
Proof.

Disproving Kůrka's conjecture

Theorem (Blondel, Cassaigne, Nichitu '02)

There exists a complete Turing machine aperiodic in TMT.

Proof.



□

Disproving Kůrka's conjecture

- The machine presented is a compressed version of the one explained in Blondel et al. work.
- As can be seen in the diagram, the machine is not reversible.
- In [Kari, Ollinger '08], a reversible, but incomplete, Aperiodic Turing machine was presented.
- Using *reversing the time*, it was proved that **aperiodicity is undecidable in incomplete and reversible Turing machines**.
- Finally, it was **conjectured** the **existence and undecidability of aperiodicity in complete and reversible Turing machines**.

Disproving Kůrka's conjecture

- The machine presented is a compressed version of the one explained in Blondel et al. work.
- As can be seen in the diagram, the machine is not reversible.
- In [Kari, Ollinger '08], a reversible, but incomplete, Aperiodic Turing machine was presented.
- Using *reversing the time*, it was proved that **aperiodicity is undecidable in incomplete and reversible Turing machines**.
- Finally, it was **conjectured** the **existence and undecidability of aperiodicity in complete and reversible Turing machines**.

Disproving Kůrka's conjecture

- The machine presented is a compressed version of the one explained in Blondel et al. work.
- As can be seen in the diagram, the machine is not reversible.
- In [Kari, Ollinger '08], a reversible, but incomplete, Aperiodic Turing machine was presented.
- Using *reversing the time*, it was proved that **aperiodicity is undecidable in incomplete and reversible Turing machines**.
- Finally, it was **conjectured** the **existence and undecidability of aperiodicity in complete and reversible Turing machines**.

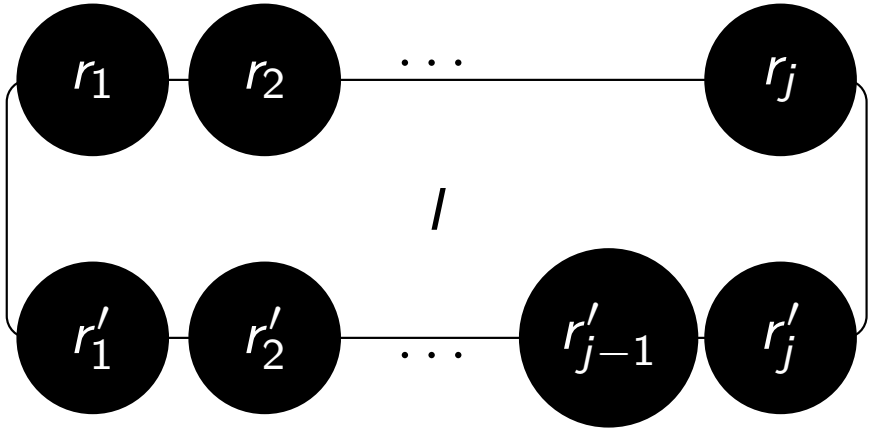
Disproving Kůrka's conjecture

- The machine presented is a compressed version of the one explained in Blondel et al. work.
- As can be seen in the diagram, the machine is not reversible.
- In [Kari, Ollinger '08], a reversible, but incomplete, Aperiodic Turing machine was presented.
- Using *reversing the time*, it was proved that **aperiodicity is undecidable in incomplete and reversible Turing machines**.
- Finally, it was **conjectured** the **existence and undecidability of aperiodicity in complete and reversible Turing machines**.

Disproving Kůrka's conjecture

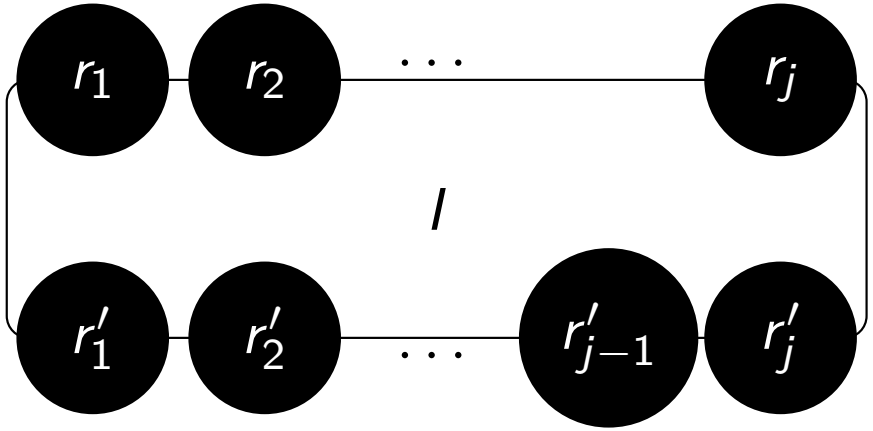
- The machine presented is a compressed version of the one explained in Blondel et al. work.
- As can be seen in the diagram, the machine is not reversible.
- In [Kari, Ollinger '08], a reversible, but incomplete, Aperiodic Turing machine was presented.
- Using *reversing the time*, it was proved that **aperiodicity is undecidable in incomplete and reversible Turing machines**.
- Finally, it was **conjectured** the **existence and undecidability of aperiodicity in complete and reversible Turing machines**.

Useful technique to be used: Embedding



- Consider $\forall i \in \{1, ..., j\}$: r'_i: Starting states. r_i: Halting states.

Useful technique to be used: Embedding

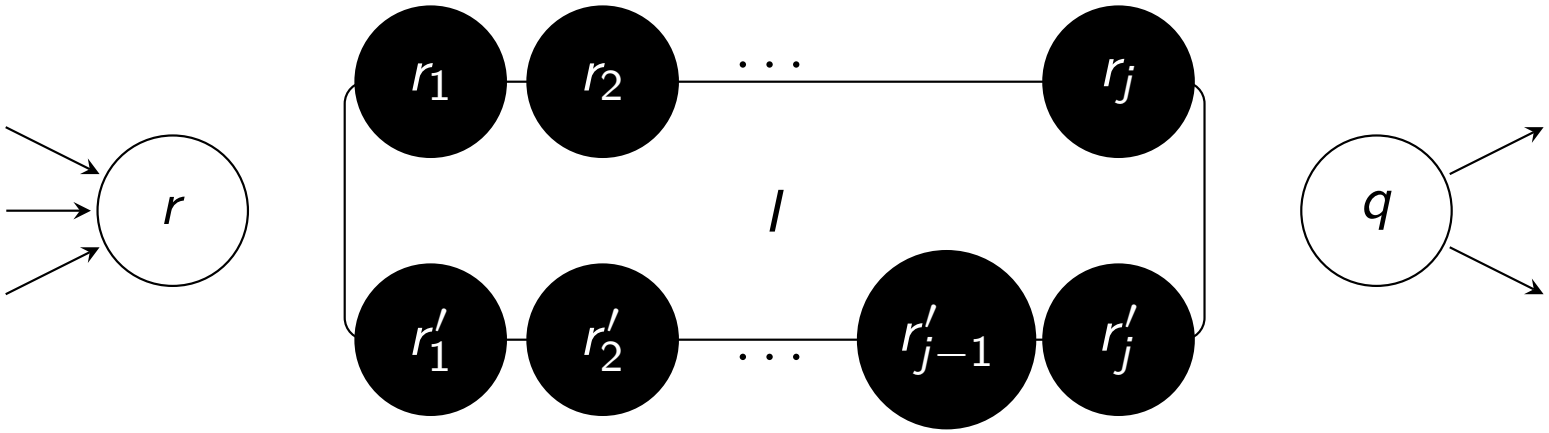


- Consider $\forall i \in \{1, ..., j\}$: r'_i: Starting states. r_i: Halting states.
- Machine I needs to be *innocuous*, which can be attained by using *reversing the time*.

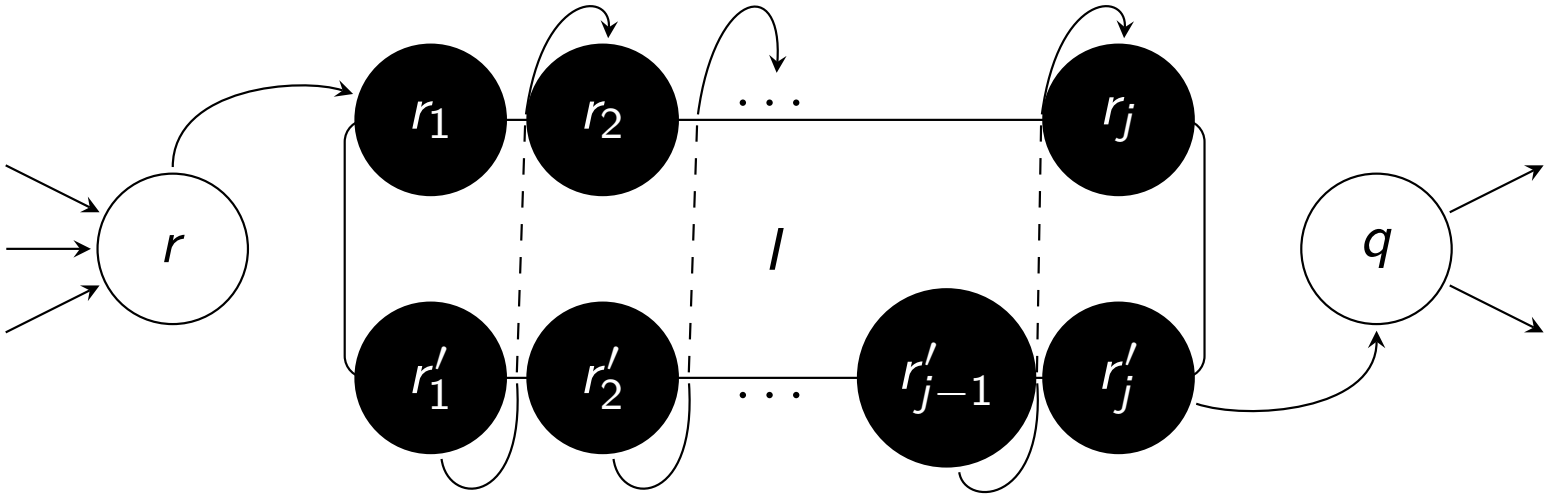
Useful technique to be used: Embedding

- Machine H:

Useful technique to be used: Embedding



Useful technique to be used: Embedding



Proving Kari & Ollinger conjecture

Theorem (Cassaigne, Ollinger, TA '17)

There exists an aperiodic complete and reversible TM in TMT with three symbols. It is called SMART machine.

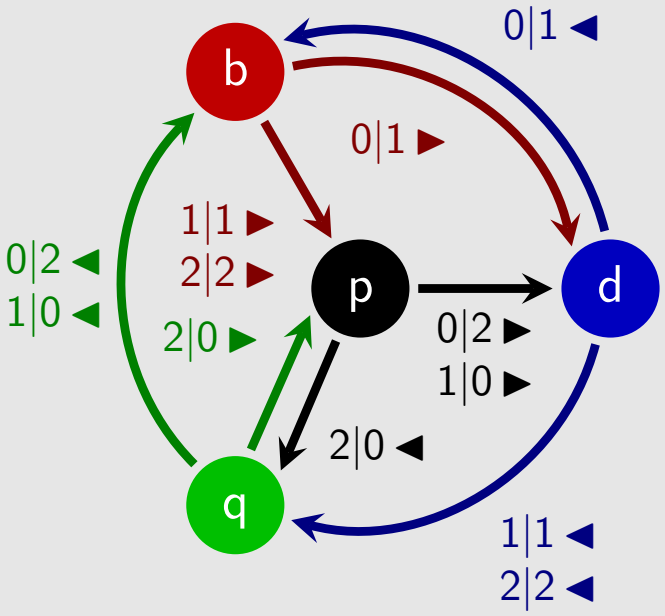
Proof.

Proving Kari & Ollinger conjecture

Theorem (Cassaigne, Ollinger, TA '17)

There exists an aperiodic complete and reversible TM in TMT with three symbols. It is called SMART machine.

Proof.



SMART machine

- This machine uses the same idea of Hooper's call pushing a positive symbol to the right. But this machine also does it to the left, in a recursive way.
- This machine is symmetric in the space and time.
- Also, its t-shift is also a **substitution**. (the closure of a replace from words to words).
- It is also the first example of the existence of...

SMART machine

- This machine uses the same idea of Hooper's call pushing a positive symbol to the right. But this machine also does it to the left, in a recursive way.
- This machine is symmetric in the space and time.
- Also, its t-shift is also a **substitution**. (the closure of a replace from words to words).
- It is also the first example of the existence of...

SMART machine

- This machine uses the same idea of Hooper's call pushing a positive symbol to the right. But this machine also does it to the left, in a recursive way.
- This machine is symmetric in the space and time.
- Also, its t-shift is also a **substitution**. (the closure of a replace from words to words).
- It is also the first example of the existence of...

SMART machine

- This machine uses the same idea of Hooper's call pushing a positive symbol to the right. But this machine also does it to the left, in a recursive way.
- This machine is symmetric in the space and time.
- Also, its t-shift is also a **substitution**. (the closure of a replace from words to words).
- It is also the first example of the existence of...

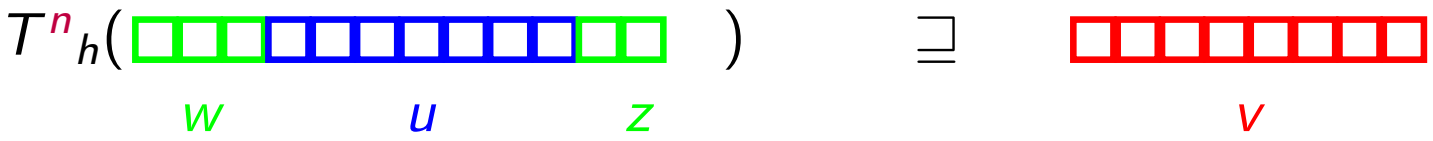
Topological Transitivity in TMH

For every $u,v \in (\Sigma \times Q)^* : |u|_Q \leq 1, |v|_Q \leq 1$

- Using *embedding*, it is possible to prove undecidability of this property with a reduction of the aperiodicity problem for reversible machines.

Topological Transitivity in TMH

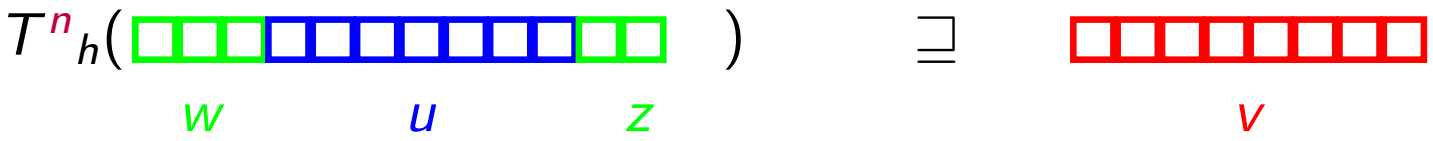
Exist $w, z \in (\Sigma \times Q)^*$ and $n \in \mathbb{N}$ such that:



- Using *embedding*, it is possible to prove undecidability of this property with a reduction of the aperiodicity problem for reversible machines.

Topological Transitivity in TMH

Exist $w, z \in (\Sigma \times Q)^*$ and $n \in \mathbb{N}$ such that:



- Using *embedding*, it is possible to prove undecidability of this property with a reduction of the aperiodicity problem for reversible machines.

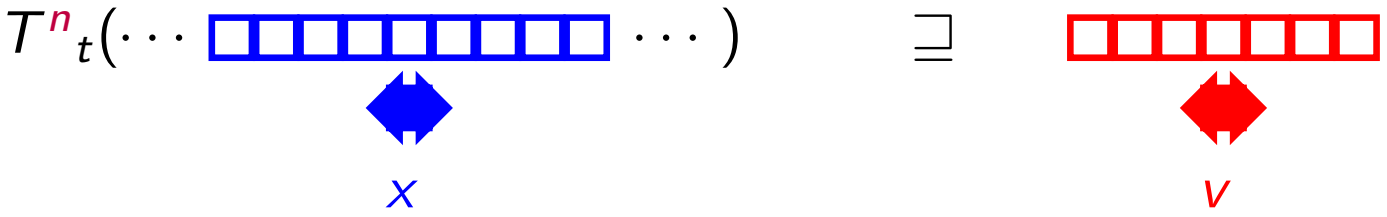
Topological Minimality in TMT

For all $x \in X_t$ and $v \in \Sigma^* \times Q \times \Sigma^*$

- For minimality, we need no periodic points. Then, it is expected than our first minimal machine is discover within the search of aperiodic Turing machines.
- Using *embedding*, it is possible to prove undecidability of this property with a reduction of the mortality problem for reversible machines.

Topological Minimality in TMT

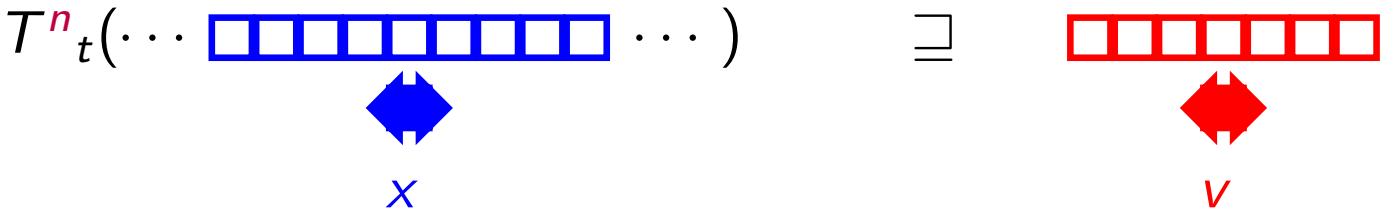
Exists $n \in \mathbb{N}$ such that:



- For minimality, we need no periodic points. Then, it is expected than our first minimal machine is discover within the search of aperiodic Turing machines.
- Using *embedding*, it is possible to prove undecidability of this property with a reduction of the mortality problem for reversible machines.

Topological Minimality in TMT

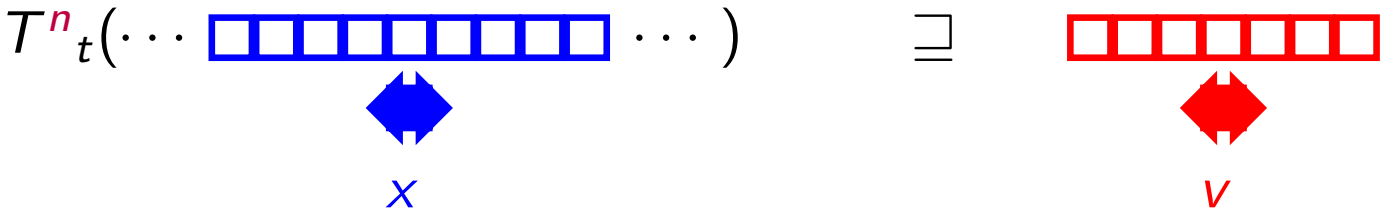
Exists $n \in \mathbb{N}$ such that:



- For minimality, we need no periodic points. Then, it is expected than our first minimal machine is discover within the search of aperiodic Turing machines.
- Using *embedding*, it is possible to prove undecidability of this property with a reduction of the mortality problem for reversible machines.

Topological Minimality in TMT

Exists $n \in \mathbb{N}$ such that:



- For minimality, we need no periodic points. Then, it is expected than our first minimal machine is discover within the search of aperiodic Turing machines.
- Using *embedding*, it is possible to prove undecidability of this property with a reduction of the mortality problem for reversible machines.

Content

The world of transitive t-shift

- Let us consider the following property:
- Blocking words [Gajardo, Ollinger, **TA** '12]:

$$(\exists(,r,u) \in (\Sigma^* \times Q \times \Sigma^*))(\forall w \in \Sigma^\omega)(\forall n \in \mathbb{N}) : \rho((,r,uw),n) \geq 0$$

- Blocking word problem was proven undecidable using a reduction from the halting of 2 reversible counter machine with empty counters.

The world of transitive t-shift

- Let us consider the following property:
- Blocking words [Gajardo, Ollinger, **TA** '12]:

$$(\exists(,r,u) \in (\Sigma^* \times Q \times \Sigma^*))(\forall w \in \Sigma^\omega)(\forall n \in \mathbb{N}) : \rho((,r,uw),n) \geq 0$$

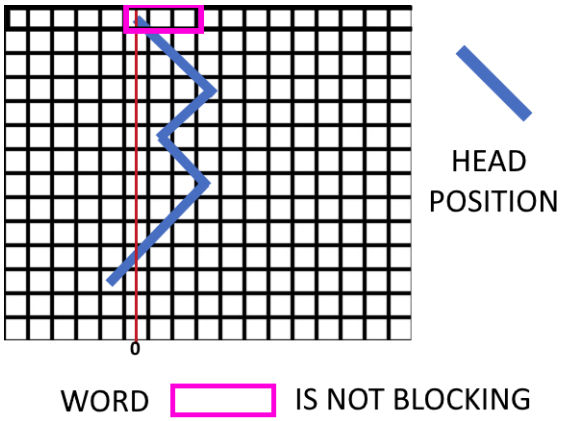
- Blocking word problem was proven undecidable using a reduction from the halting of 2 reversible counter machine with empty counters.

The world of transitive t-shift

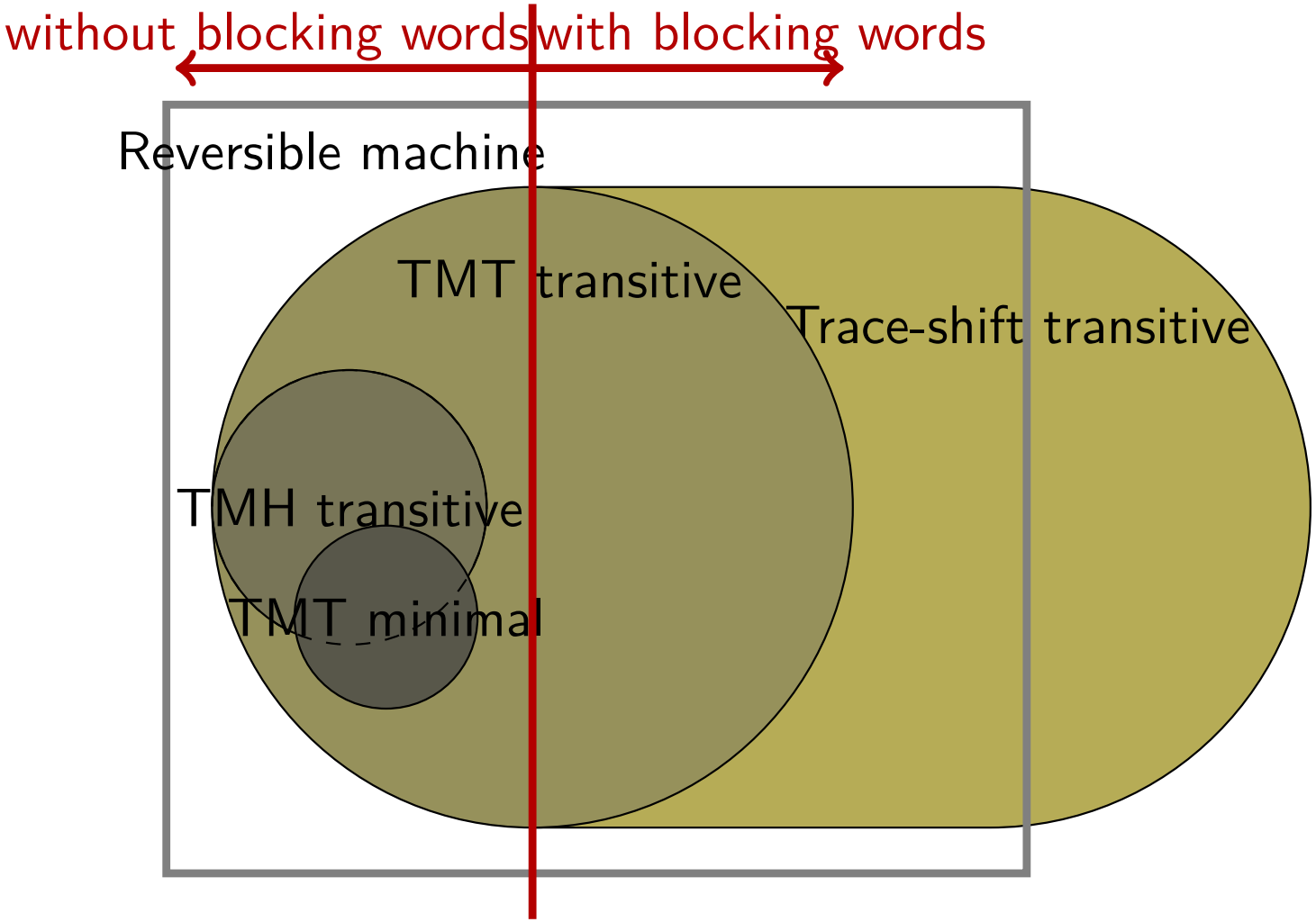
- Let us consider the following property:
- Blocking words [Gajardo, Ollinger, **TA** '12]:

$$(\exists(,r,u) \in (\Sigma^* \times Q \times \Sigma^*))(\forall w \in \Sigma^\omega)(\forall n \in \mathbb{N}) : \rho((,r,uw),n) \geq 0$$

- Blocking word problem was proven undecidable using a reduction from the halting of 2 reversible counter machine with empty counters.



The world of transitive t-shift



Embedding properties

- Every subset in the map has countable many Turing machines.
- Embedding technique is proven to kept some properties from the machines that formed it. Consider H the complete machine, I the innocuous machine, and H_I the formed machine. Then:
  - H_I is reversible $\leftrightarrow$ H is reversible.
  - If I is aperiodic: H is aperiodic $\rightarrow$ H_I is aperiodic.
  - If I is aperiodic and with smaller alphabet: H_I is transitive $\leftrightarrow$ H is transitive.
  - If I is mortal: H_I is minimal $\leftrightarrow$ H is minimal.
  - If I is immortal, then H_I is never minimal[iv].

[iv] This implies that Minimality is undecidable, even in the presence of Transitivity

Embedding properties

- Every subset in the map has countable many Turing machines.
- Embedding technique is proven to kept some properties from the machines that formed it. Consider H the complete machine, I the innocuous machine, and H_I the formed machine. Then:
  - H_I is reversible $\leftrightarrow$ H is reversible.
  - If I is aperiodic: H is aperiodic $\rightarrow$ H_I is aperiodic.
  - If I is aperiodic and with smaller alphabet: H_I is transitive $\leftrightarrow$ H is transitive.
  - If I is mortal: H_I is minimal $\leftrightarrow$ H is minimal.
  - If I is immortal, then H_I is never minimal[iv].

[iv] This implies that Minimality is undecidable, even in the presence of Transitivity

Embedding properties

- Every subset in the map has countable many Turing machines.
- Embedding technique is proven to kept some properties from the machines that formed it. Consider H the complete machine, I the innocuous machine, and H_I the formed machine. Then:
  - H_I is reversible $\leftrightarrow$ H is reversible.
  - If I is aperiodic: H is aperiodic $\rightarrow$ H_I is aperiodic.
  - If I is aperiodic and with smaller alphabet: H_I is transitive $\leftrightarrow$ H is transitive.
  - If I is mortal: H_I is minimal $\leftrightarrow$ H is minimal.
  - If I is immortal, then H_I is never minimal[iv].

[iv] This implies that Minimality is undecidable, even in the presence of Transitivity

Embedding properties

- Every subset in the map has countable many Turing machines.
- Embedding technique is proven to kept some properties from the machines that formed it. Consider H the complete machine, I the innocuous machine, and H_I the formed machine. Then:
  - H_I is reversible $\leftrightarrow$ H is reversible.
  - If I is aperiodic: H is aperiodic $\rightarrow$ H_I is aperiodic.
  - If I is aperiodic and with smaller alphabet: H_I is transitive $\leftrightarrow$ H is transitive.
  - If I is mortal: H_I is minimal $\leftrightarrow$ H is minimal.
  - If I is immortal, then H_I is never minimal[iv].

[iv] This implies that Minimality is undecidable, even in the presence of Transitivity

Embedding properties

- Every subset in the map has countable many Turing machines.
- Embedding technique is proven to kept some properties from the machines that formed it. Consider H the complete machine, I the innocuous machine, and H_I the formed machine. Then:
  - H_I is reversible $\leftrightarrow$ H is reversible.
  - If I is aperiodic: H is aperiodic $\rightarrow$ H_I is aperiodic.
  - If I is aperiodic and with smaller alphabet: H_I is transitive $\leftrightarrow$ H is transitive.
  - If I is mortal: H_I is minimal $\leftrightarrow$ H is minimal.
  - If I is immortal, then H_I is never minimal[iv].

[iv] This implies that Minimality is undecidable, even in the presence of Transitivity

Embedding properties

- Every subset in the map has countable many Turing machines.
- Embedding technique is proven to kept some properties from the machines that formed it. Consider H the complete machine, I the innocuous machine, and H_I the formed machine. Then:
  - H_I is reversible $\leftrightarrow$ H is reversible.
  - If I is aperiodic: H is aperiodic $\rightarrow$ H_I is aperiodic.
  - If I is aperiodic and with smaller alphabet: H_I is transitive $\leftrightarrow$ H is transitive.
  - If I is mortal: H_I is minimal $\leftrightarrow$ H is minimal.
  - If I is immortal, then H_I is never minimal[iv].

[iv] This implies that Minimality is undecidable, even in the presence of Transitivity

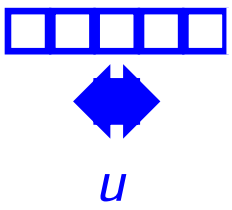
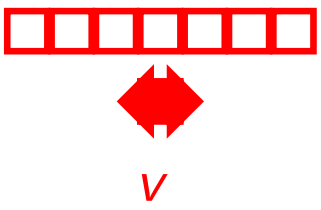
Embedding properties

- Every subset in the map has countable many Turing machines.
- Embedding technique is proven to kept some properties from the machines that formed it. Consider H the complete machine, I the innocuous machine, and H_I the formed machine. Then:
  - H_I is reversible $\leftrightarrow$ H is reversible.
  - If I is aperiodic: H is aperiodic $\rightarrow$ H_I is aperiodic.
  - If I is aperiodic and with smaller alphabet: H_I is transitive $\leftrightarrow$ H is transitive.
  - If I is mortal: H_I is minimal $\leftrightarrow$ H is minimal.
  - If I is immortal, then H_I is never minimal[iv].

[iv] This implies that Minimality is undecidable, even in the presence of Transitivity

Topological Mixing

$\forall u, v \in \Sigma^* \times Q \times \Sigma^*$ and $\forall N \geq n$ (which depends in u and v)

$T^N{}_t($  $)$ 

- Considering a combinatorial proof, it is possible to prove that **SMART machine is top. mixing**.
- Considering *embedding*, it is possible to prove that **Every mixing notion is undecidable for Turing machines**, with a reduction of the aperiodicity problem.
- Also, **Topological Weak-Mixing**[v] **is decidable in every embedding of known Minimal Turing machines**.

[v] $(X_t \times X_t, T_t \times T_t)$ is transitive

Topological Mixing

Exist $w, z \in \Sigma^*$ such that:

- Considering a combinatorial proof, it is possible to prove that **SMART machine is top. mixing**.
- Considering *embedding*, it is possible to prove that **Every mixing notion is undecidable for Turing machines**, with a reduction of the aperiodicity problem.
- Also, **Topological Weak-Mixing[v] is decidable in every embedding of known Minimal Turing machines**.

[v] $(X_t \times X_t, T_t \times T_t)$ is transitive

Topological Mixing

Exist $w, z \in \Sigma^*$ such that:

- Considering a combinatorial proof, it is possible to prove that **SMART machine is top. mixing**.
- Considering *embedding*, it is possible to prove that **Every mixing notion is undecidable for Turing machines**, with a reduction of the aperiodicity problem.
- Also, **Topological Weak-Mixing**[v] **is decidable in every embedding of known Minimal Turing machines**.

[v] $(X_t \times X_t, T_t \times T_t)$ is transitive

Topological Mixing

Exist $w, z \in \Sigma^*$ such that:

- Considering a combinatorial proof, it is possible to prove that **SMART machine is top. mixing**.
- Considering *embedding*, it is possible to prove that **Every mixing notion is undecidable for Turing machines**, with a reduction of the aperiodicity problem.
- Also, **Topological Weak-Mixing[v] is decidable in every embedding of known Minimal Turing machines**.

[v] $(X_t \times X_t, T_t \times T_t)$ is transitive

Topological Mixing

Exist $w, z \in \Sigma^*$ such that:

- Considering a combinatorial proof, it is possible to prove that **SMART machine is top. mixing**.
- Considering *embedding*, it is possible to prove that **Every mixing notion is undecidable for Turing machines**, with a reduction of the aperiodicity problem.
- Also, **Topological Weak-Mixing**[v] **is decidable in every embedding of known Minimal Turing machines**.

[v] $(X_t \times X_t, T_t \times T_t)$ is transitive

Content

Discussion

- This talk was a (non-exhaustive) compilation of some of the work done in Turing machine Topological Dynamical Properties.
- Much of the work adjacent to the topological properties were not touch, as the interesting work of Guillon, Salo, Gajardo and many others (apologies in advance).
- Gajardo et al. have results considering Sensitivity, Equicontinuity and Soficity.
- An interesting result of [Guillon, Salo '17] is that aperiodic machines as a sub-linear movement in the space.

Discussion

- This talk was a (non-exhaustive) compilation of some of the work done in Turing machine Topological Dynamical Properties.
- Much of the work adjacent to the topological properties were not touch, as the interesting work of Guillon, Salo, Gajardo and many others (apologies in advance).
- Gajardo et al. have results considering Sensitivity, Equicontinuity and Soficity.
- An interesting result of [Guillon, Salo '17] is that aperiodic machines as a sub-linear movement in the space.

Discussion

- This talk was a (non-exhaustive) compilation of some of the work done in Turing machine Topological Dynamical Properties.
- Much of the work adjacent to the topological properties were not touch, as the interesting work of Guillon, Salo, Gajardo and many others (apologies in advance).
- Gajardo et al. have results considering Sensitivity, Equicontinuity and Soficity.
- An interesting result of [Guillon, Salo '17] is that aperiodic machines as a sub-linear movement in the space.

Discussion

- This talk was a (non-exhaustive) compilation of some of the work done in Turing machine Topological Dynamical Properties.
- Much of the work adjacent to the topological properties were not touch, as the interesting work of Guillon, Salo, Gajardo and many others (apologies in advance).
- Gajardo et al. have results considering Sensitivity, Equicontinuity and Soficity.
- An interesting result of [Guillon, Salo '17] is that aperiodic machines as a sub-linear movement in the space.

Open questions

- There exists plenty of space to work in this field.
- Direct questions arise as: There exists a minimal non-linearly recurrent Turing machine?
  - In linearly recurrent systems, word u can be seen in the trace again at a maximum of $K * |u|$ space, with K depending on the system.
  - If it would not exists such a Turing machine, Topological Weak-Mixing is decidable in Minimal Turing machines.
- Other interesting questions is about the arithmetical hierarchy of some topological properties (which are also related to the previous question).
- Finally, a Rice like theorem considering one tape (for multiple tapes, Blondel et al. already answer the question).

Open questions

- There exists plenty of space to work in this field.
- Direct questions arise as: There exists a minimal non-linearly recurrent Turing machine?
  - In linearly recurrent systems, word u can be seen in the trace again at a maximum of $K * |u|$ space, with K depending on the system.
  - If it would not exists such a Turing machine, Topological Weak-Mixing is decidable in Minimal Turing machines.
- Other interesting questions is about the arithmetical hierarchy of some topological properties (which are also related to the previous question).
- Finally, a Rice like theorem considering one tape (for multiple tapes, Blondel et al. already answer the question).

Open questions

- There exists plenty of space to work in this field.
- Direct questions arise as: There exists a minimal non-linearly recurrent Turing machine?
  - In linearly recurrent systems, word u can be seen in the trace again at a maximum of $K * |u|$ space, with K depending on the system.
  - If it would not exists such a Turing machine, Topological Weak-Mixing is decidable in Minimal Turing machines.
- Other interesting questions is about the arithmetical hierarchy of some topological properties (which are also related to the previous question).
- Finally, a Rice like theorem considering one tape (for multiple tapes, Blondel et al. already answer the question).

Open questions

- There exists plenty of space to work in this field.
- Direct questions arise as: There exists a minimal non-linearly recurrent Turing machine?
  - In linearly recurrent systems, word u can be seen in the trace again at a maximum of $K * |u|$ space, with K depending on the system.
  - If it would not exists such a Turing machine, Topological Weak-Mixing is decidable in Minimal Turing machines.
- Other interesting questions is about the arithmetical hierarchy of some topological properties (which are also related to the previous question).
- Finally, a Rice like theorem considering one tape (for multiple tapes, Blondel et al. already answer the question).

Content

Fundings

CONICYT
Ministerio de Educación
Gobierno de Chile
FONDECYT
Fondo Nacional de Desarrollo
Científico y Tecnológico

ecos
Coopération & Partenariat
France – Amérique latine

Dirección General de Investigación,
Desarrollo e Innovación

LABORATOIRE
D'INFORMATIQUE
LIFO
FONDAMENTALE
D'ORLEANS