Topological Mixing and Linear Recurrence on SMART

Rodrigo Torres-Avilés

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

Online Logic Seminar, June 25, 2020
Content

1 Preliminars
   - Intuition
   - Math definitions
   - SMART machine and sons(?)
   - Investigative questions

2 Results
   - SMART is Top. Mixing
   - Undecidability of Top. Mixing in $t$-shifts
   - Weak Mixing and Embedding

3 Closing
   - Fundings
Traditional Turing Machine

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

Partial Transition function: \( \delta : Q \times \Sigma \cup \{\Lambda\} \rightarrow Q \times \Sigma \cup \{\Lambda\} \times \{\leftarrow, \bullet, \rightarrow\} \)

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

\[
\begin{array}{ccccccc}
-1 & 0 & 1 & 2 & 3 & 4 & 5 \\
\Lambda & 0 & 0 & 0 & 1 & \Lambda & \Lambda \\
\end{array}
\]
Traditional Turing Machine

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

Partial Transition function: \( \delta : Q \times \Sigma \cup \{\Lambda\} \rightarrow Q \times \Sigma \cup \{\Lambda\} \times \{\blackleftarrow, \bullet, \rightarrow\} \)

Configuration: \((Q, \mathbb{Z}, (\Sigma \cup \{\Lambda\})^\mathbb{Z})\)
Turing machine is a six-tuple: $T = (Q, \Sigma, q_0, F, \Lambda, \delta)$

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

Configuration: $(Q, \mathbb{Z}, (\Sigma \cup \{\Lambda\})^\mathbb{Z})$
Converting to Turing Machine with Moving Tape (TMT)

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

Partial Transition function: \( \delta : Q \times \Sigma \cup \{\Lambda\} \rightarrow Q \times \Sigma \cup \{\Lambda\} \times \{\leftarrow, \cdot, \rightarrow\} \)

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

\[
\begin{array}{cccccccc}
-1 & 0 & 1 & 2 & 3 & 4 & 5 & \\
\Lambda & 0 & 0 & 1 & 1 & \Lambda & \Lambda & \\
\end{array}
\]
Converting to Turing Machine with Moving Tape (TMT)

Turing machine is a three-tuple: \( T = (Q, \Sigma, \delta) \)
Partial Transition function: \( \delta : Q \times \Sigma \rightarrow Q \times \Sigma \times \{\leftarrow, \bullet, \rightarrow\} \)

Configuration: \( (Q, Z, \Sigma^Z) \)

\[
\begin{array}{cccccccc}
-1 & 0 & 1 & 2 & 3 & 4 & 5 \\
\cdots & 0 & 0 & 0 & 1 & 1 & 0 & 0 & \cdots
\end{array}
\]
Converting to Turing Machine with Moving Tape (TMT)

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

**Total** Transition function: \( \delta : Q \times \Sigma \rightarrow Q \times \Sigma \times \{\leftarrow, \circ, \rightarrow\} \)

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

\[
\begin{array}{cccccc}
0 & 0 & 0 & 1 & 1 & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}
\]
Converting to Turing Machine with Moving Tape (TMT)

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

**Total** Transition function: \( \delta : Q \times \Sigma \rightarrow Q \times \Sigma \times \{\leftarrow, \cdot, \rightarrow\} \)

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

\[
\begin{array}{ccccccc}
\cdots & 0 & 0 & 0 & 1 & 1 & 0 & 0 & \cdots \\
\end{array}
\]
Converting to Turing Machine with Moving Tape (TMT)

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

Total Transition function: \( \delta : Q \times \Sigma \rightarrow Q \times \Sigma \times \{\mathord{\lt}, \mathord{\bullet}, \mathord{\gt}\} \)

Configuration: \((\omega \Sigma, Q, \Sigma^\omega)\)
Converting to Turing Machine with Moving Tape (TMT)

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

**Total** Transition function: \( \delta : Q \times \Sigma \rightarrow Q \times \Sigma \times \{\leftarrow, \bullet, \rightarrow\} \)

Configuration: \((\omega \Sigma, Q, \Sigma^\omega)\)

\[
\begin{array}{cccccccc}
\cdots & 0 & 0 & 0 & 1 & 1 & 0 & 0 & \cdots \\
\end{array}
\]
Converting to Turing Machine with Moving Tape (TMT)

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

**Total** Transition function: \( \delta : Q \times \Sigma \rightarrow Q \times \Sigma \times \{\leftarrow, \bullet, \rightarrow\} \)

Configuration: \( (\omega \Sigma, Q, \Sigma^\omega) \)

\[ \begin{array}{cccccccc}
... & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & ... \\
\end{array} \]
Column Factor of TMT ($t$-shift)

\[
\begin{array}{cccccccc}
0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
\end{array}
\]

\[\xrightarrow{T}\]

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
\end{array}
\]

\[\xrightarrow{T}\]

\[
\begin{array}{cccccccc}
0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
\end{array}
\]

\[\xleftarrow{T}\]

\[
\begin{array}{cccccccc}
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
\end{array}
\]

\[\xrightarrow{T}\]

\[
\begin{array}{cccccccc}
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
\end{array}
\]
Column Factor of TMT ($t$-shift)
Column Factor of TMT (t-shift)

\[
\begin{array}{cccccccccc}
\cdots & 0 & 1 & 0 & 0 & 1 & 0 & 1 & \cdots \\
\downarrow T & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \cdots \\
\cdots & 1 & 0 & 0 & 1 & 0 & 1 & 0 & \cdots \\
\downarrow T & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \cdots \\
\cdots & 0 & 1 & 0 & 0 & 1 & 0 & 1 & \cdots \\
\downarrow T & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \uparrow \rightarrow & \cdots \\
\cdots & 1 & 0 & 1 & 1 & 0 & 1 & 0 & \cdots \\
\end{array}
\]
Column Factor of TMT (t-shift)
Content

1 Preliminaries
   - Intuition
   - Math definitions
   - SMART machine and sons(?)
   - Investigative questions

2 Results
   - SMART is Top. Mixing
   - Undecidability of Top. Mixing in $t$-shifts
   - Weak Mixing and Embedding

3 Closing
   - Fundings
Then, formally speaking...

- Considering Turing machine $T = (Q, \Sigma, \delta)$, the pair $(X = (\omega \Sigma, Q, \Sigma^\omega), T)$ is a Topological Dynamical System known as *Turing Machine with moving Tape*.
- $X$ is endowed with the cantor metric, being compact and perfect (no isolated points).
- $T$ (to apply a Turing instruction) is a continuous map.
- Considering $\pi(u, q, v) = (q, v_0)$, the subshift known as *t-shift* $(S_t, \sigma)$ is the factor $\tau : X \to S_t$, defined as $\tau(x) = (\pi(T^n(x)))_{n \in \mathbb{N}}$. 
Then, formally speaking...

- Considering Turing machine $T = (Q, \Sigma, \delta)$, the pair $(X = (\omega \Sigma, Q, \Sigma^\omega), T)$ is a Topological Dynamical System known as *Turing Machine with moving Tape*.
- $X$ is endowed with the cantor metric, being compact and perfect (no isolated points).
- $T$ (to apply a Turing instruction) is a continuous map.
- Considering $\pi(u, q, v) = (q, v_0)$, the subshift known as *t-shift* $(S_t, \sigma)$ is the factor $\tau : X \to S_t$, defined as $\tau(x) = (\pi(T^n(x)))_{n \in \mathbb{N}}$.
Then, formally speaking...

- Considering Turing machine $T = (Q, \Sigma, \delta)$, the pair $(X = (\omega \Sigma, Q, \Sigma^\omega), T)$ is a Topological Dynamical System known as *Turing Machine with moving Tape*.
- $X$ is endowed with the cantor metric, being compact and perfect (no isolated points).
- $T$ (to apply a Turing instruction) is a continuous map.
- Considering $\pi(u, q, v) = (q, v_0)$, the subshift known as *t-shift* $(S_t, \sigma)$ is the factor $\tau: X \to S_t$, defined as $\tau(x) = (\pi(T^n(x)))_{n \in \mathbb{N}}$.
Then, formally speaking...

- Considering Turing machine $T = (Q, \Sigma, \delta)$, the pair $(X = (\omega \Sigma, Q, \Sigma^\omega), T)$ is a Topological Dynamical System known as *Turing Machine with moving Tape*.

- $X$ is endowed with the cantor metric, being compact and perfect (no isolated points).

- $T$ (to apply a Turing instruction) is a continuous map.

- Considering $\pi(u, q, v) = (q, v_0)$, the subshift known as *t-shift* $(S_t, \sigma)$ is the factor $\tau : X \rightarrow S_t$, defined as $\tau(x) = (\pi(T^n(x)))_{n \in \mathbb{N}}$
Interesting questions already answer!

- Is there an aperiodic $t$-shift? $(\forall w \in S_t, \forall n \in \mathbb{N} : w \neq \sigma^n(w))$
  
  **Answer:** Yes! (SMART machine)

- Is there a minimal $t$-shift? $(\forall w \in S_t, \forall v \in \mathcal{L}(S_t) : v \sqsubseteq w^i)$
  
  **Answer:** Yes! (SMART machine)

- Is possible to decide if a $t$-shift is aperiodic and/or minimal?
  
  **Answer:** No

---

\[^1\text{This means } v \text{ is a subword or factor of } w: \exists i : v = w_iw_{i+1}...w_{i+|w|-1}.\]
Interesting questions already answer!

- Is there an aperiodic $t$-shift? $(\forall w \in S_t, \forall n \in \mathbb{N} : w \neq \sigma^n(w))$
  Answer: Yes! (SMART machine)

- Is there a minimal $t$-shift? $(\forall w \in S_t, \forall v \in L(S_t) : v \sqsubseteq w^i)$
  Answer: Yes! (SMART machine)

- Is possible to decide if a $t$-shift is aperiodic and/or minimal?
  Answer: No

---

This means $v$ is a subword or factor of $w$: $\exists i : v = w_i w_{i+1} \ldots w_{i+|w|-1}$. 
Interesting questions already answer!

- Is there an aperiodic $t$-shift? $(\forall w \in S_t, \forall n \in \mathbb{N} : w \neq \sigma^n(w))$
  Answer: Yes! (SMART machine)

- Is there a minimal $t$-shift? $(\forall w \in S_t, \forall v \in \mathcal{L}(S_t) : v \sqsubseteq w^i)$
  Answer: Yes! (SMART machine)

- Is possible to decide if a $t$-shift is aperiodic and/or minimal?
  Answer: No

---

\*This means $v$ is a subword or factor of $w$: $\exists i : v = w_i w_{i+1} \ldots w_{i+|w|-1}$.
# Content

1. Preliminars
   - Intuition
   - Math definitions
   - SMART machine and sons(?)
   - Investigative questions

2. Results
   - SMART is Top. Mixing
   - Undecidability of Top. Mixing in $t$-shifts
   - Weak Mixing and Embedding

3. Closing
   - Fundings
SMART machine
Known Behavior [COT* ’17]

- SMART machine $T$ is minimal, aperiodic and substitutive.

- State $b$ has the same behavior than $d$, but in the opposite direction. Same as $p$ with $q$.

- SMART machine pushes a one to the right

$$\exists n \in \mathbb{N} : T^n \left( \begin{array}{ccc} 1 & 0 & 0^k & 1 & 0 \\ p & 1 & 0 \\ \end{array} \right) = \left( \begin{array}{ccc} 1 & 0 & 0^{k+1} & 1 \\ p & 1 & 0 \\ \end{array} \right)$$

in a recursive way.
Known Behavior [COT* ’17]

- SMART machine $T$ is minimal, aperiodic and substitutive.
- State $b$ has the same behavior than $d$, but in the opposite direction. Same as $p$ with $q$.
- SMART machine pushes a one to the right
  \[ \exists n \in \mathbb{N} : T^n \begin{pmatrix} 1 & 0 & 0^k & 1 & 0 \\ p & 0^k & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0^{k+1} & 1 \\ p & 0^k & 1 & 0 \end{pmatrix} \] in a recursive way.
Known Behavior [COT* '17]

- SMART machine $T$ is minimal, aperiodic and substitutive.

- State $b$ has the same behavior than $d$, but in the opposite direction. Same as $p$ with $q$.

- SMART machine pushes a one to the right

  $\exists n \in \mathbb{N} : T^n \left( \begin{array}{ccc} 1 & 0 & 0^k & 1 & 0 \\ p & 0^k & 1 & 0 \end{array} \right) = \left( \begin{array}{ccc} 1 & 0 & 0^{k+1} & 1 \\ p & 0^k & 1 & 0 \end{array} \right)$

  in a recursive way.
Decidable results [GOT* ’15]

- **Innocuous**: Incomplete Turing machine which halts with the same tape content which it starts.

- **Embedding**: Insert an innocuous machine inside a Complete Turing machine.

- With certain cares, aperiodicity and minimality can depend on the aperiodicity and mortality\(^{\text{ii}}\) of the innocuous machine, respectively.

- Every known Minimal Turing machine is an embedding of SMART machine (or BinSmart, a two symbol version of the same dynamic).

\(^{\text{ii}}\)A Turing machine is *Mortal* if \((\forall x \in X)(\exists n \in \mathbb{N}) : T^n(x)\) is undef.
Decidable results [GOT* ’15]

- **Innocuous**: Incomplete Turing machine which halts with the same tape content which it starts.

- **Embedding**: Insert a innocuous machine inside a Complete Turing machine.

- With certain cares, aperiodicity and minimality can depend on the aperiodicity and mortality\(^{ii}\) of the innocuous machine, respectively.

- Every known Minimal Turing machine is an embedding of SMART machine (or BinSmart, a two symbol version of the same dynamic).

\(^{ii}\)A Turing machine is *Mortal* if \((\forall x \in X)(\exists n \in \mathbb{N}) : T^n(x)\) is undef.
Decidable results \([\text{GOT}^* \ '15]\)

- **Innocuous**: Incomplete Turing machine which halts with the same tape content which it starts.

- **Embedding**: Insert a innocuous machine inside a Complete Turing machine.

- With certain cares, aperiodicity and minimality can depend on the aperiodicity and mortality\(^{ii}\) of the innocuous machine, respectively.

- Every known Minimal Turing machine is an embedding of SMART machine (or BinSmart, a two symbol version of the same dynamic).

\(^{ii}\)A Turing machine is *Mortal* if \((\forall x \in X)(\exists n \in \mathbb{N}) : T^n(x)\) is undef.
Decidable results [GOT* '15]

- **Innocuous**: Incomplete Turing machine which halts with the same tape content which it starts.
- **Embedding**: Insert a innocuous machine inside a Complete Turing machine.

With certain cares, aperiodicity and minimality can depend on the aperiodicity and mortality\(^\text{ii}\) of the innocuous machine, respectively.

Every known Minimal Turing machine is an embedding of SMART machine (or BinSmart, a two symbol version of the same dynamic).

\(^\text{ii}\)A Turing machine is *Mortal* if \((\forall x \in X)(\exists n \in \mathbb{N}) : T^n(x)\) is undef.
Embedding

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

Machine \( I \) needs to be *innocuous*: \( (\forall i \in \{1, \ldots, l\}) : (u, r_i, v) \) evolves indefinitely, or \( \exists t \in \mathbb{N} : T^t(u, r_i, v) = (u, r'_i, v) \) and it is a halting configuration.
Embedding

Machine $H$:
Embedding

\[ r \rightarrow r_1 \rightarrow r_2 \rightarrow \ldots \rightarrow r_j \rightarrow r'_1 \rightarrow r'_2 \rightarrow \ldots \rightarrow r'_{j-1} \rightarrow r'_j \rightarrow q \]
Embedding

Diagram showing nodes labeled as $r$, $r_1$, $r_2$, ..., $r_j$, and $q$. There are dashed lines connecting these nodes, indicating a network structure. The diagram suggests a flow or transition from $r$ to $q$ through the intermediate nodes.
Complexity

- If machine $H$ is Minimal, then $H_I^{iii}$ is Minimal if and only if $I$ is mortal.

- The time machine $H_I$ takes inside machine $I$ before returning to machine $H$ is bounded by $j \ast t_I^{iv}$.

- Therefore, exists an infinite amount of Minimal Turing machines, embedding mortal machines into SMART (or binSMART).

---

$^{iii}$Machine $I$ embedded into machine $H$

$^{iv}$Every Mortal Turing machine is uniformly mortal
Complexity

- If machine $H$ is Minimal, then $H^\text{iii}_I$ is Minimal if and only if $I$ is mortal.

- The time machine $H_I$ takes inside machine $I$ before returning to machine $H$ is bounded by $j \ast t^\text{iv}_I$.

- Therefore, exists an infinite amount of Minimal Turing machines, embedding mortal machines into SMART (or binSMART).

---

$^\text{iii}$ Machine $I$ embedded into machine $H$

$^\text{iv}$ Every Mortal Turing machine is uniformly mortal
Complexity

- If machine $H$ is Minimal, then $H_I^{iii}$ is Minimal if and only if $I$ is mortal.

- The time machine $H_I$ takes inside machine $I$ before returning to machine $H$ is bounded by $j * t_I^{iv}$.

- Therefore, exists an infinite amount of Minimal Turing machines, embedding mortal machines into SMART (or binSMART).

---

$^{iii}$Machine $I$ embedded into machine $H$

$^{iv}$Every Mortal Turing machine is uniformly mortal
1 Preliminaries
   - Intuition
   - Math definitions
   - SMART machine and sons(?)
   - Investigative questions

2 Results
   - SMART is Top. Mixing
   - Undecidability of Top. Mixing in $t$-shifts
   - Weak Mixing and Embedding

3 Closing
   - Fundings
Questions not answered yet

- Is there a minimal $t$-shift which is topological mixing?
  $$(\forall u, v \in \mathcal{L}(S_t), \exists n \in \mathbb{N}, \forall N > n, \exists w \in \mathcal{L}(S_t) : |uw| = N \land uwv \in \mathcal{L}(S_t))$$

- Is top. mixing decidable in $t$-shifts?\(^v\)

- Is weak mixing decidable in minimal $t$-shifts?
  $$((S_t \times S_t, \sigma \times \sigma) \text{ is transitive}^{vi}.)$$

\(\quad ^v\text{An easy example of a non-minimal top. mixing } t\text{-shift is the full shift machine.}\)

\(\quad ^{vi}(\exists w \in S_t, \forall v \in \mathcal{L}(S_t) : v \subseteq w)\)
Questions not answered yet

- Is there a minimal $t$-shift which is topological mixing?
  \[
  (\forall u, v \in \mathcal{L}(S_t), \exists n \in \mathbb{N}, \forall N > n, \exists w \in \mathcal{L}(S_t) : |uw| = N \land uwv \in \mathcal{L}(S_t))
  \]

- Is top. mixing decidable in $t$-shifts?\(^v\)

- Is weak mixing decidable in minimal $t$-shifts?
  \[
  ((S_t \times S_t, \sigma \times \sigma) \text{ is transitive}^v)
  \]

\(^v\) An easy example of a non-minimal top. mixing $t$-shift is the full shift machine.

\(^vi\) \[
(\exists w \in S_t, \forall v \in \mathcal{L}(S_t) : v \subseteq w)
\]
Questions not answered yet

- Is there a minimal $t$-shift which is topological mixing?
  \[(\forall u, v \in \mathcal{L}(S_t), \exists n \in \mathbb{N}, \forall N > n, \exists w \in \mathcal{L}(S_t) : |uw| = N \land uwv \in \mathcal{L}(S_t))\]

- Is top. mixing decidable in $t$-shifts?\(^v\)

- Is weak mixing decidable in minimal $t$-shifts?
  \[((S_t \times S_t, \sigma \times \sigma) \text{ is transitive}^{vi}.)\]

\(^v\)An easy example of a non-minimal top. mixing $t$-shift is the full shift machine.

\(^{vi}(\exists w \in S_t, \forall v \in \mathcal{L}(S_t) : v \sqsubseteq w)\)
Content

1 Preliminars
   - Intuition
   - Math definitions
   - SMART machine and sons(?)
   - Investigative questions

2 Results
   - SMART is Top. Mixing
     - Undecidability of Top. Mixing in t-shifts
     - Weak Mixing and Embedding

3 Closing
   - Fundings
Considerations before the proof

- We center in top. mixing in TMT, as $t$-shift is indeed a factor.
- Let us declare $X = \left( \ast \Sigma \times Q \times \Sigma^* \right)$. In this way, the open sets in $X$ are defined by elements $u \in X$, as:
  
  \[ [u] = \{ x \in X : u \sqsubseteq x \} \]

- The return time in a Turing machine $T$ for a given $u \in X$ is
  \[ R_T(u) = \max\{ n : (\forall w \sqsubseteq u)n = \min\{ m > 0 : T^m(w) \sqsubseteq u \} \} \]
  If $T$ is minimal, then $R_T(u)$ is finite for all $u$. This concept can also be applied in subshifts.

\[ \text{vii This means that, if } u = (v, r, v'), \text{ then } (\exists w \in \omega \Sigma, \exists w' \in \omega \Sigma) \text{ such that } x = (wv, r, v'w') \]
Considerations before the proof

- We center in top. mixing in TMT, as $t$-shift is indeed a factor.
- Let us declare $X = (\Sigma \times Q \times \Sigma^*)$. In this way, the open sets in $X$ are defined by elements $u \in X$, as:

$$[u] = \{ x \in X : u \sqsubseteq x^v \}$$

- The *return time* in a Turing machine $T$ for a given $u \in X$ is

$$R_T(u) = \max\{ n : (\forall w \sqsubseteq u)n = \min\{ m > 0 : T^m(w) \sqsubseteq u \} \}.$$  
If $T$ is minimal, then $R_T(u)$ is finite for all $u$. This concept can also be applied in subshifts.

---

$vii$ This means that, if $u = (v, r, v')$, then $(\exists w \in \Sigma, \exists w' \in \Sigma)$ such that $x = (wv, r, v'w')$
Considerations before the proof

- We center in top. mixing in TMT, as $t$-shift is indeed a factor.
- Let us declare $X = (*\Sigma \times Q \times \Sigma^*)$. In this way, the open sets in $X$ are defined by elements $u \in X$, as:

  $$[u] = \{ x \in X : u \sqsubseteq x^{\text{vii}} \}$$

- The *return time* in a Turing machine $T$ for a given $u \in X$ is

  $$R_T(u) = \max \{ n : (\forall w \sqsubseteq u)n = \min \{ m > 0 : T^m(w) \sqsupseteq u \} \}.$$ 

  If $T$ is minimal, then $R_T(u)$ is finite for all $u$. This concept can also be applied in subshifts.

\[ \text{vii} \text{This means that, if } u = (v, r, v'), \text{ then } (\exists w \in \omega \Sigma, \exists w' \in \omega \Sigma) \text{ such that } x = (wv, r, v'w') \]
Main Theorem

Theorem

SMART machine has a topological Mixing TMT.
Proof Sketch

**Topological Mixing:** \((\forall u, v)(\exists n)(\forall N > n)(\exists u' \supseteq u) : T^N(u') \supseteq v\)
Proof Sketch

Topological Mixing: $(\forall u, v)(\exists n)(\forall N > n)(\exists u' \sqsubseteq u) : T^N(u') \sqsubseteq v$
Proof Sketch

Topological Mixing: $(\forall u, v)(\exists n)(\forall N > n)(\exists u' \supseteq u) : T^N(u') \supseteq v$

Let us use: $\forall u, v \in (\Sigma^* \times Q \times \Sigma^*)$, $\exists k$ big enough such that:

\[ T^{t_u} \]

\[ T^{t_v} \]
Proof Sketch

Topological Mixing: \((\forall u, v)(\exists n)(\forall N > n)(\exists u' \sqsupseteq u) : T^N(u') \sqsubseteq v\)

Now, \(m = R_T(\, b, 00^k\,), \forall N' > J(k, m)^{\text{viii}}, \exists i \leq m \text{ and } w, w' \in \Sigma^*:\)

\[
\begin{array}{ccccccc}
1^\omega & 2^{m-i} & 1^i & 0 & 0^k & 2^{m+1} & 1^\omega \\
\downarrow & & & & & T^{t_u} & \\
... & u & ... \\
\end{array}
\quad \rightarrow \quad
\begin{array}{ccccccc}
1^\omega & w & 0 & 0^k & w' & 1^\omega \\
\downarrow & & & & & T^{t_v} & \\
... & v & ... \\
\end{array}
\]

\[J(k, m) = \frac{3^{k+3}}{4}(3^{2m} - 1) - m\]
Proof Sketch

Topological Mixing: $(\forall u, v)(\exists n)(\forall N > n)(\exists u' \supseteq u) : T^N(u') \supseteq v$

Now, $m = R_T(b, 00^k), \forall N' > J(k, m)^{\text{viii}}, \exists i \leq m$ and $w, w' \in \Sigma^*$:

$$1^\omega \ 2^{m-i} \ 1^i \ 0 \ 0^k \ 2^{m+1} \ 1^\omega \quad \xrightarrow{T^{N'}} \quad 1^\omega \ w \ 0 \ 0^k \ w' \ 1^\omega$$

Therefore, $n = J(k, m)$, $N = N' - t_u + t_v$ and $u' = T^{t_u}(1^\omega 2^{m-i} 1^i, b, 00^k 2^{m+1} 1^\omega)$.

$$J(k, m) = \frac{3^{k+3}}{4}(3^{2m} - 1) - m$$
Content

1 Preliminars
   • Intuition
   • Math definitions
   • SMART machine and sons(?)
   • Investigative questions

2 Results
   • SMART is Top. Mixing
   • Undecidability of Top. Mixing in $t$-shifts
   • Weak Mixing and Embedding

3 Closing
   • Fundings
Theorem

Every Mixing notion is undecidable in t-shifts.
Proof Sketch

- Consider full shit machine
  
  \[ H = (\{r_0\}, \{0, 1, 2\}, \forall i : \delta(r_0, i) = (r_0, i, \uparrow)) \].

- Consider \( I \) any binary innocuous machine.

- Replace instruction \( \delta(r_0, 2) = (r_0, 2, \uparrow) \) to form the embedding \( H_I \).

- As machine \( H_I \) moves freely to the right with symbol 0 (or 1), then \( H_I \) is top. mixing if and only if \( I \) is aperiodic\(^{\text{ix}} \).

---

\(^{\text{ix}}\)We are referring to aperiodic seeing a finite amount of tape. If it is not, \( H_I \) is not even transitive.
Proof Sketch

- Consider full shit machine
  \[ H = (\{r_0\}, \{0, 1, 2\}, \forall i : \delta(r_0, i) = (r_0, i, \triangleright)). \]

- Consider \( I \) any binary innocuous machine.

- Replace instruction \( \delta(r_0, 2) = (r_0, 2, \triangleright) \) to form the embedding \( H_I \).

- As machine \( H_I \) moves freely to the right with symbol 0 (or 1), then \( H_I \) is top. mixing if and only if \( I \) is aperiodic\(^{ix} \).

\(^{ix}\)We are referring to aperiodic seeing a finite amount of tape. If it is not, \( H_I \) is not even transitive.
Proof Sketch

- Consider full shit machine
  \[ H = (\{r_0\}, \{0, 1, 2\}, \forall i : \delta(r_0, i) = (r_0, i, \Rightarrow)) \].
- Consider \( I \) any binary innocuous machine.
- Replace instruction \( \delta(r_0, 2) = (r_0, 2, \Rightarrow) \) to form the embedding \( H_I \).
- As machine \( H_I \) moves freely to the right with symbol 0 (or 1), then \( H_I \) is top. mixing if and only if \( I \) is aperiodic\(^{ix}\).

\(^{ix}\) We are referring to aperiodic seeing a finite amount of tape. If it is not, \( H_I \) is not even transitive.
Proof Sketch

- Consider full shit machine
  \[ H = (\{r_0\}, \{0, 1, 2\}, \forall i : \delta(r_0, i) = (r_0, i, \triangleright)). \]

- Consider \( I \) any binary innocuous machine.

- Replace instruction \( \delta(r_0, 2) = (r_0, 2, \triangleright) \) to form the embedding \( H_I \).

- As machine \( H_I \) moves freely to the right with symbol 0 (or 1), then \( H_I \) is top. mixing if and only if \( I \) is aperiodic\(^ix\).

\(^ix\)We are referring to aperiodic seeing a finite amount of tape. If it is not, \( H_I \) is not even transitive.
Content

1 Preliminars
- Intuition
- Math definitions
- SMART machine and sons(?)
- Investigative questions

2 Results
- SMART is Top. Mixing
- Undecidability of Top. Mixing in $t$-shifts
- Weak Mixing and Embedding

3 Closing
- Fundings
x ∈ Σ^N is \textit{linearly recurrent} if ∃ K ∈ N, ∀ u ⊑ x : R_x(u) ≤ K |u|.

A minimal subshift S is said to be \textit{linearly recurrent} if it posses an element w ∈ S which is linearly recurrent.

In particular, if a minimal subshift S is linearly recurrent, then ∀ w ∈ S : w is linearly recurrent.
Linearly Recurrent

- \( x \in \Sigma^\mathbb{N} \) is *linearly recurrent* if \( \exists K \in \mathbb{N}, \forall u \sqsupseteq x : R_x(u) \leq K|u| \).

- A minimal subshift \( S \) is said to be *linearly recurrent* if it posses an element \( w \in S \) which is linearly recurrent.

- In particular, if a minimal subshift \( S \) is linearly recurrent, then \( \forall w \in S : w \) is linearly recurrent.
A minimal subshift $S$ is said to be linearly recurrent if it possesses an element $w \in S$ which is linearly recurrent.

In particular, if a minimal subshift $S$ is linearly recurrent, then $\forall w \in S : w$ is linearly recurrent.
Every minimal subshift product of a substitution is linearly recurrent [Durant].

Weak Mixing is decidable in linearly recurrent subshifts [Durant].

SMART machine has a minimal and substitutive $t$-shift [COT* 17’], therefore it is linearly recurrent.

Interestingly, every known minimal $t$-shift is an embedding of a Linearly Recurrent Turing machine with a Mortal machine.
Every minimal subshift product of a substitution is linearly recurrent [Durant].

Weak Mixing is decidable in linearly recurrent subshifts [Durant].

SMART machine has a minimal and substitutive $t$-shift [COT* 17'], therefore it is linearly recurrent.

Interestingly, every known minimal $t$-shift is an embedding of a Linearly Recurrent Turing machine with a Mortal machine.
Every minimal subshift product of a substitution is linearly recurrent [Durant].

Weak Mixing is decidable in linearly recurrent subshifts [Durant].

SMART machine has a minimal and substitutive $t$-shift [COT* 17'], therefore it is linearly recurrent.

Interestingly, every known minimal $t$-shift is an embedding of a Linearly Recurrent Turing machine with a Mortal machine.
Every minimal subshift product of a substitution is linearly recurrent [Durant].

Weak Mixing is decidable in linearly recurrent subshifts [Durant].

SMART machine has a minimal and substitutive $t$-shift [COT* 17'], therefore it is linearly recurrent.

Interestingly, every known minimal $t$-shift is an embedding of a Linearly Recurrent Turing machine with a Mortal machine.
Lemma

*Embedding a Mortal Turing machine on a Linearly Recurrent Turing machine is Linearly Recurrent.*
Proof Sketch

- Consider minimal and linearly recurrent machine $H$.
- Consider $I$ any Innocuous mortal machine with $j$ pairs.
- Consider any $u \in \mathcal{L}(S_{H_I})$ and consider $w \supseteq u$ any extension of $u$ that does not repeat $u$ ($u$ is prefix of $w$ and $|w|_u = 1$). As $H_I$ is minimal, we know that $|w|$ is always finite.
Proof Sketch

- Consider minimal and linearly recurrent machine $H$.
- Consider any Innocuous mortal machine with $j$ pairs.
- Consider any $u \in \mathcal{L}(S_{H_I})$ and consider $w \supseteq u$ any extension of $u$ that does not repeat $u$ ($u$ is prefix of $w$ and $|w|_u = 1$). As $H_I$ is minimal, we know that $|w|$ is always finite.
Proof Sketch

- Consider minimal and linearly recurrent machine $H$.
- Consider $I$ any innocuous mortal machine with $j$ pairs.
- Consider any $u \in \mathcal{L}(S_{H_I})$ and consider $w \supseteq u$ any extension of $u$ that does not repeat $u$ ($u$ is prefix of $w$ and $|w|_u = 1$). As $H_I$ is minimal, we know that $|w|$ is always finite.
Proof Sketch

- Consider minimal and linearly recurrent machine $H$.
- Consider $I$ any Innocuous mortal machine with $j$ pairs.
- Consider any $u \in \mathcal{L}(S_{H_I})$ and consider $w \supseteq u$ any extension of $u$ that does not repeat $u$ ($u$ is prefix of $w$ and $|w|_u = 1$). As $H_I$ is minimal, we know that $|w|$ is always finite.

$$w = \text{[Visualization of a string with blue and red segments]}$$

As we stated, the time it takes inside $I$ (in red) is bounded by $j \ast t_I$ in every call.
Proof Sketch

- Consider minimal and linearly recurrent machine $H$.
- Consider $I$ any Innocuous mortal machine with $j$ pairs.
- Consider any $u \in \mathcal{L}(S_{H_I})$ and consider $w \sqsupseteq u$ any extension of $u$ that does not repeat $u$ ($u$ is prefix of $w$ and $|w|_u = 1$). As $H_I$ is minimal, we know that $|w|$ is always finite.

If we take out the red parts, we get $w' \in \mathcal{L}(S_H)$. We know that every possible $w'$ is bounded by some $K \times |u'|$ ($u$ without red parts).
Proof Sketch

- Consider minimal and linearly recurrent machine $H$.
- Consider $I$ any Innocuous mortal machine with $j$ pairs.
- Consider any $u \in \mathcal{L}(S_{H_I})$ and consider $w \supseteq u$ any extension of $u$ that does not repeat $u$ ($u$ is prefix of $w$ and $|w|^u = 1$). As $H_I$ is minimal, we know that $|w|$ is always finite.

$$w = \begin{array}{ccccccccccc}
\text{Blue} & \text{Blue} & \text{Blue} & \text{Red} & \text{Red} & \text{Red} & \text{...} & \text{Blue} & \text{Blue} & \text{Red} & \text{Red} & \text{Red} & \text{...} & \text{Blue} \\
\end{array}$$

Therefore, every $w \leq K \times |u| \times j \times t_I = (K \times j \times t_I) \times |u|$. Ergo, $H_I$ is linearly recurrent.
Conjecture

Weak Mixing is decidable in minimal $t$-shifts
Content

1. Preliminars
   - Intuition
   - Math definitions
   - SMART machine and sons(?)
   - Investigative questions

2. Results
   - SMART is Top. Mixing
   - Undecidability of Top. Mixing in $t$-shifts
   - Weak Mixing and Embedding

3. Closing
   - Fundings
Fundings

CONICYT
Ministerio de Educación
Gobierno de Chile

FONDECYT
Fondo Nacional de Desarrollo Científico y Tecnológico

ecos

Dirección General de Investigación, Desarrollo e Innovación

LABORATOIRE D'INFORMATIQUE
FONDAMENTALE D'ORLEANS