Dinámica Simbólica sobre los grupos
Problema de dominó y aperiodicidad en $\mathbb{Z}$ y $\mathbb{Z}^2$.

Sebastián Barbieri Lemp

University of British Columbia

Escuela de invierno en grupos y dinámica en México
Enero, 2018

Una pincelada de computación

Para esta plática necesitaremos un poquitito de teoría de computación.

Idea

Supongamos que estamos enseñando un curso de cálculo y hablamos de una función.

Una pincelada de computación

Para esta plática necesitaremos un poquitito de teoría de computación.

Idea

Supongamos que estamos enseñando un curso de cálculo y hablamos de una función.

- Nosotros: $f : X \to Y$ es una asignación de un único elemento de Y a cada elemento de X.
- Estudiante: $f(x) = y$ es una **regla** explícita que asocia a x un valor particular: $\sin(x)$, e^x, $2x + 1$, etc.

Una pincelada de computación

Para esta plática necesitaremos un poquitito de teoría de computación.

Idea

Supongamos que estamos enseñando un curso de cálculo y hablamos de una función.

- Nosotros: $f : X \to Y$ es una asignación de un único elemento de Y a cada elemento de X.
- Estudiante: $f(x) = y$ es una **regla** explícita que asocia a x un valor particular: $\sin(x)$, e^x, $2x + 1$, etc.

¿Cómo formalizar la noción de regla? -Algoritmo.

Algoritmo

Informalmente, un algoritmo es un conjunto finito de reglas que se aplican secuencialmente sin limitación de tiempo ni de memoria.

Algoritmo

Informalmente, un algoritmo es un conjunto finito de reglas que se aplican secuencialmente sin limitación de tiempo ni de memoria.

Ejemplos

- Programas en C++, Java, Python, Sage, etc.

Algoritmo

Informalmente, un algoritmo es un conjunto finito de reglas que se aplican secuencialmente sin limitación de tiempo ni de memoria.

Ejemplos

▷ Programas en C++, Java, Python, Sage, etc.
▷ El proceso de diagonalizar una matriz con entradas racionales 1000×1000 con lápiz y papel sin cometer errores.

Algoritmo

Informalmente, un algoritmo es un conjunto finito de reglas que se aplican secuencialmente sin limitación de tiempo ni de memoria.

Ejemplos

$\triangleright$ Programas en C++, Java, Python, Sage, etc.
$\triangleright$ El proceso de diagonalizar una matriz con entradas racionales 1000×1000 con lápiz y papel sin cometer errores.
$\triangleright$ El proceso de encontrar la inversa de una matriz con entradas racionales.

Algoritmo

Informalmente, un algoritmo es un conjunto finito de reglas que se aplican secuencialmente sin limitación de tiempo ni de memoria.

Ejemplos

$\triangleright$ Programas en C++, Java, Python, Sage, etc.
$\triangleright$ El proceso de diagonalizar una matriz con entradas racionales 1000×1000 con lápiz y papel sin cometer errores.
$\triangleright$ El proceso de encontrar la inversa de una matriz con entradas racionales.

Una (de muchas) formalizaciones de este concepto es la máquina de Turing.

Máquinas de Turing

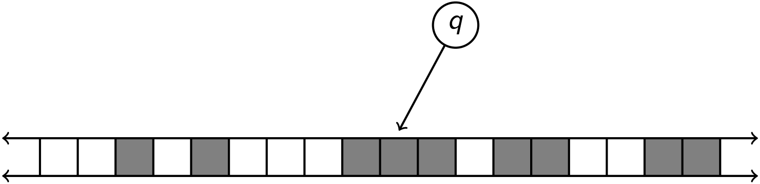
Una máquina de Turing T está esencialmente definida por:

- Un alfabeto Σ.
- Un conjunto de estados Q.
- Una regla $\delta_T : \Sigma \times Q \to \Sigma \times Q \times \{-1, 0, 1\}$.

Máquinas de Turing

Una máquina de Turing T está esencialmente definida por:

- Un alfabeto Σ.
- Un conjunto de estados Q.
- Una regla $\delta_T : \Sigma \times Q \to \Sigma \times Q \times \{-1, 0, 1\}$.

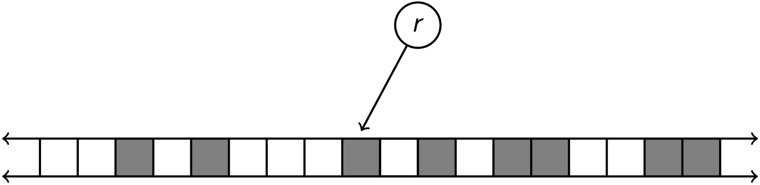


$$\delta_T(\blacksquare, q) = (\square, r, -1)$$

Máquinas de Turing

Una máquina de Turing T está esencialmente definida por:

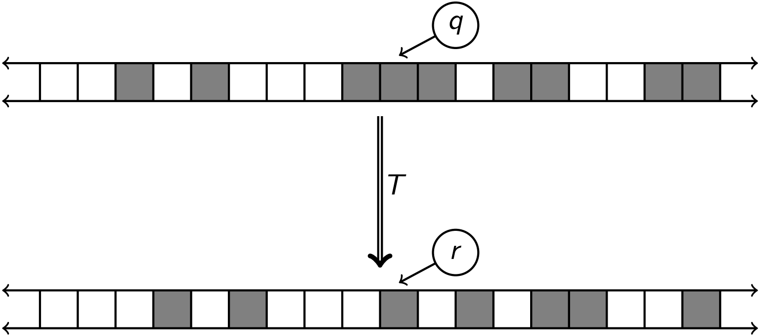
- Un alfabeto Σ.
- Un conjunto de estados Q.
- Una regla $\delta_T : \Sigma \times Q \to \Sigma \times Q \times \{-1, 0, 1\}$.



$$\delta_T(\blacksquare, q) = (\square, r, -1)$$

Esta regla define una función:

$$T : \Sigma^{\mathbb{Z}} \times Q \to \Sigma^{\mathbb{Z}} \times Q$$



Esta regla define una función:

$$T : \Sigma^{\mathbb{Z}} \times Q \to \Sigma^{\mathbb{Z}} \times Q$$

Tal que si $(x, q) \in \Sigma^{\mathbb{Z}} \times Q$ y $\delta_T(x_0, q) = (a, r, d)$ entonces:

$$T(x, q) = (\sigma_{-d}(\tilde{x}), r)$$

donde $\tilde{x}_0 = a$ y $\tilde{x}|_{\mathbb{Z}\setminus\{0\}} = x|_{\mathbb{Z}\setminus\{0\}}$.
También distinguimos dos estados especiales en Q. Un estado inicial q_I y un estado de parada q_F.

Máquinas de Turing

▷ asumamos que el alfabeto Σ contiene un símbolo especial $\sqcup$ representando el blanco.

$\triangleright$ asumamos que el alfabeto Σ contiene un símbolo especial $\sqcup$ representando el blanco.

Sea $w \in (\Sigma \setminus \{\sqcup\})^*$. Una MT T acepta a w si existe $n \in \mathbb{N}$ tal que

$$T(\sqcup^\infty.w\sqcup^\infty, q_I) \in \Sigma^{\mathbb{Z}} \times \{q_F\}.$$

Es decir, w es aceptada por T si comenzando en la configuración que contiene a w en el origen y símbolos blancos alrededor, se llega eventualmente al estado q_F.

Sea $L \subset (\Sigma \setminus \{\sqcup\})^*$ un lenguaje.

Definición

- Decimos que L es *recursivamente enumerable (RE)* si existe una MT T tal que: $w \in L$ si y solamente si w es aceptada por T.
- Si $(\Sigma \setminus \{\sqcup\})^* \setminus L$ es RE, decimos que L es *co-recursivamente enumerable (co-RE)*
- Si L es RE y co-RE, decimos que es *decidible*.

Ejemplos

- El lenguaje $L \subset \{0,1\}^*$ de los números naturales en base 2 que son divisibles por 8 es decidible.
- El lenguaje de codificaciones de gráficas finitas que contienen un ciclo es decidible.
- El lenguaje de los números naturales n en base 2 tal que $\underbrace{666\ldots6}_{n\text{-veces}}$ aparece en el desarrollo en binario de π es decidible.

¿Es mi SFT vacío?

Ejemplo

El lenguaje de conjuntos de patrones prohibidos $\mathcal{F}$ finitos que definen un $\mathbb{Z}^d$-SFT no vacío es co-RE.

¿Es mi SFT vacío?

Ejemplo

El lenguaje de conjuntos de patrones prohibidos $\mathcal{F}$ finitos que definen un $\mathbb{Z}^d$-SFT no vacío es co-RE.

Algoritmo

1. Instanciar $n \leftarrow 0$.
2. Actualizar $n \leftarrow n+1$ y construir $\Lambda_n = [-n, n]^d$.
3. Para cada $p \in \mathcal{A}^{\Lambda_n}$, verificar si aparece un patrón prohibido en p.
   - Si existe p sin patrones prohibidos, volver al paso 2.
   - Si en todo $p \in \mathcal{A}^{\Lambda_n}$ aparece un patrón prohibido, aceptar $\mathcal{F}$.

¿Es mi SFT vacío?

Ejemplo

El lenguaje de conjuntos de patrones prohibidos $\mathcal{F}$ finitos que definen un $\mathbb{Z}^d$-SFT no vacío es co-RE.

Algoritmo

1. Instanciar $n \leftarrow 0$.
2. Actualizar $n \leftarrow n + 1$ y construir $\Lambda_n = [-n, n]^d$.
3. Para cada $p \in \mathcal{A}^{\Lambda_n}$, verificar si aparece un patrón prohibido en p.
   - Si existe p sin patrones prohibidos, volver al paso 2.
   - Si en todo $p \in \mathcal{A}^{\Lambda_n}$ aparece un patrón prohibido, aceptar $\mathcal{F}$.

¿Es ese lenguaje decidible?

El caso de $\mathbb{Z}$

Ejercicio

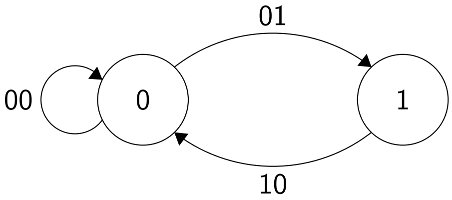
Todo $\mathbb{Z}$-SFT es conjugado al conjunto de paseos bi-infinitos en una gráfica finita.

El caso de $\mathbb{Z}$

Ejercicio

Todo $\mathbb{Z}$-SFT es conjugado al conjunto de paseos bi-infinitos en una gráfica finita.

Example: Ejemplo, si $\mathcal{F} = \{11\}$.

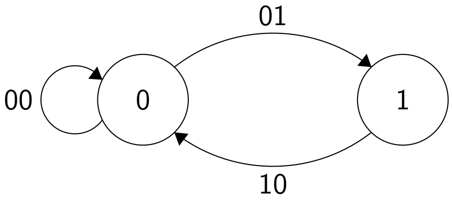


El caso de $\mathbb{Z}$

Ejercicio

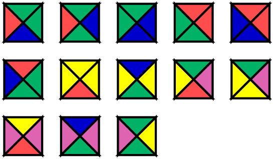
Todo $\mathbb{Z}$-SFT es conjugado al conjunto de paseos bi-infinitos en una gráfica finita.

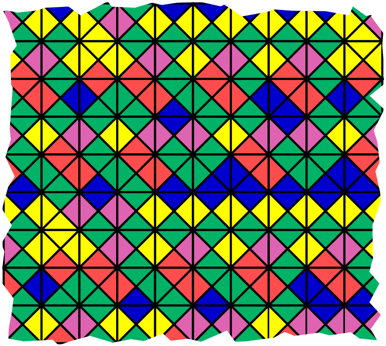
Example: Ejemplo, si $\mathcal{F} = \{11\}$.



Cómo la gráfica es finita, el SFT es no vacío si y solamente si la gráfica contiene un ciclo. En consecuencia la respuesta es **sí** para el caso de $\mathbb{Z}$.

Recuerdo: Teselado de Wang



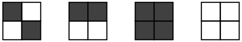


Recuerdo: Teselado de Wang

Ejercicio

Todo $\mathbb{Z}^2$-SFT es conjugado a un conjunto de teselaciones por baldosas de Wang.

Patrones prohibidos de tamaño k

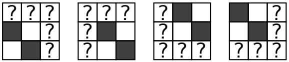


Recuerdo: Teselado de Wang

Ejercicio

Todo $\mathbb{Z}^2$-SFT es conjugado a un conjunto de teselaciones por baldosas de Wang.

Patrones prohibidos de tamaño $2n+1$



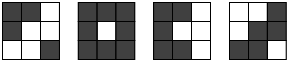
etc. . .

Recuerdo: Teselado de Wang

Ejercicio

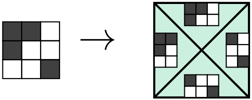
Todo $\mathbb{Z}^2$-SFT es conjugado a un conjunto de teselaciones por baldosas de Wang.

Patrones admisibles de tamaño $2n+1$



etc. . .

Baldosas de Wang



Problema de Dominó

Por la equivalencia anterior, podemos limitarnos a estudiar:

Problema de Dominó

Sea τ un conjunto de baldosas de Wang. ¿Existe un algoritmo que determine si teselan el plano?

Problema de Dominó

Por la equivalencia anterior, podemos limitarnos a estudiar:

Problema de Dominó

Sea τ un conjunto de baldosas de Wang. ¿Existe un algoritmo que determine si teselan el plano?

Demostraremos que la respuesta es **NO**. Para ello, debemos estudiar como mostrar resultados negativos de ese estilo.

Probando que algo no es decidible

¿ Cómo demostrar que un lenguaje no es decidible? – Por contradicción y reducciones.

Problema de la parada (contradicción)

Sea $\mathtt{PP}$ el lenguaje de las codificaciones de una MT T y palabras w tal que T acepta a w.

Probando que algo no es decidible

¿ Cómo demostrar que un lenguaje no es decidible? – Por contradicción y reducciones.

Problema de la parada (contradicción)

Sea $\mathtt{PP}$ el lenguaje de las codificaciones de una MT T y palabras w tal que T acepta a w.

- Supongamos existe una MT $\mathcal{M}$ que acepta $(\langle T \rangle, w)$ si y solamente **no** pertenece a $\mathtt{PP}$. (Asumimos es co-RE)

Probando que algo no es decidible

¿ Cómo demostrar que un lenguaje no es decidible? – Por contradicción y reducciones.

Problema de la parada (contradicción)

Sea $\mathtt{PP}$ el lenguaje de las codificaciones de una MT T y palabras w tal que T acepta a w.

- Supongamos existe una MT $\mathcal{M}$ que acepta $(\langle T \rangle, w)$ si y solamente **no** pertenece a $\mathtt{PP}$. (Asumimos es co-RE)
- Consideremos la máquina $\mathcal{N}$ que en la entrada $\langle T \rangle$ simula $\mathcal{M}$ en la entrada $(\langle T \rangle, \langle T \rangle)$.

Probando que algo no es decidible

¿ Cómo demostrar que un lenguaje no es decidible? – Por contradicción y reducciones.

Problema de la parada (contradicción)

Sea $\mathtt{PP}$ el lenguaje de las codificaciones de una MT T y palabras w tal que T acepta a w.

- Supongamos existe una MT $\mathcal{M}$ que acepta $(\langle T\rangle, w)$ si y solamente **no** pertenece a $\mathtt{PP}$. (Asumimos es co-RE)
- Consideremos la máquina $\mathcal{N}$ que en la entrada $\langle T\rangle$ simula $\mathcal{M}$ en la entrada $(\langle T\rangle, \langle T\rangle)$.
- Supongamos $\mathcal{N}$ acepta a $\langle \mathcal{N}\rangle$. Entonces por definición $\mathcal{M}$ acepta a $(\langle \mathcal{N}\rangle, \langle \mathcal{N}\rangle)$, lo cual significa que $\langle \mathcal{N}\rangle$ no es aceptada por $\mathcal{N}$.

Probando que algo no es decidible

¿ Cómo demostrar que un lenguaje no es decidible? – Por contradicción y reducciones.

Problema de la parada (contradicción)

Sea $\mathtt{PP}$ el lenguaje de las codificaciones de una MT T y palabras w tal que T acepta a w.

- Supongamos existe una MT $\mathcal{M}$ que acepta $(\langle T \rangle, w)$ si y solamente **no** pertenece a $\mathtt{PP}$. (Asumimos es co-RE)
- Consideremos la máquina $\mathcal{N}$ que en la entrada $\langle T \rangle$ simula $\mathcal{M}$ en la entrada $(\langle T \rangle, \langle T \rangle)$.
- Supongamos $\mathcal{N}$ acepta a $\langle \mathcal{N} \rangle$. Entonces por definición $\mathcal{M}$ acepta a $(\langle \mathcal{N} \rangle, \langle \mathcal{N} \rangle)$, lo cual significa que $\langle \mathcal{N} \rangle$ no es aceptada por $\mathcal{N}$.
- Entonces $\mathcal{N}$ no acepta a $\langle \mathcal{N} \rangle$, pero eso significa que $\mathcal{M}$ no acepta a $(\langle \mathcal{N} \rangle, \langle \mathcal{N} \rangle)$, lo cual significa que $\langle \mathcal{N} \rangle$ es aceptada por $\mathcal{N}$.

Probando que algo es indecidible

¿Cómo demostrar que un lenguaje no es decidible? – Por contradicción y reducciones.

Problema de la parada en entrada vacía (reducción)

Sea PPV el lenguaje de las codificaciones de una MT T tal que T acepta la entrada vacía.

Probando que algo es indecidible

¿Cómo demostrar que un lenguaje no es decidible? – Por contradicción y reducciones.

Problema de la parada en entrada vacía (reducción)

Sea $\mathtt{PPV}$ el lenguaje de las codificaciones de una MT T tal que T acepta la entrada vacía.

- Supongamos existe una MT $\mathcal{M}$ que acepta $\langle T \rangle$ si y solamente **no** pertenece a $\mathtt{PPV}$.

Probando que algo es indecidible

¿Cómo demostrar que un lenguaje no es decidible? – Por contradicción y reducciones.

Problema de la parada en entrada vacía (reducción)

Sea PPV el lenguaje de las codificaciones de una MT T tal que T acepta la entrada vacía.

- Supongamos existe una MT $\mathcal{M}$ que acepta $\langle T \rangle$ si y solamente **no** pertenece a PPV.
- Consideremos la máquina $\mathcal{N}$ que en la entrada $(\langle T \rangle, w)$ hace lo siguiente:
  - escribe en la cinta la codificación de una MT T' que en entrada vacía escribe w y luego ejecuta T.
  - Ejecuta $\mathcal{M}$ en $\langle T' \rangle$.

Probando que algo es indecidible

¿Cómo demostrar que un lenguaje no es decidible? – Por contradicción y reducciones.

Problema de la parada en entrada vacía (reducción)

Sea $\mathtt{PPV}$ el lenguaje de las codificaciones de una MT T tal que T acepta la entrada vacía.

- Supongamos existe una MT $\mathcal{M}$ que acepta $\langle T \rangle$ si y solamente **no** pertenece a $\mathtt{PPV}$.
- Consideremos la máquina $\mathcal{N}$ que en la entrada $(\langle T \rangle, w)$ hace lo siguiente:
  - escribe en la cinta la codificación de una MT T' que en entrada vacía escribe w y luego ejecuta T.
  - Ejecuta $\mathcal{M}$ en $\langle T' \rangle$.
- $\mathcal{N}$ acepta $(\langle T \rangle, w)$ si y solamente si $\mathcal{M}$ acepta $\langle T' \rangle$. Por definición, esto es si T no acepta w. Pero tal máquina no existe ya que $\mathtt{PP}$ no es co-RE.

Reducciones y más reducciones

Teorema [Hooper, 66]. Problema de la inmortalidad de MT

El lenguaje de las codificaciones de MT para las cuales existe una configuración inicial no aceptada no es decidible.

Detalle: No se requieren símbolos blancos, no se requiere comenzar a un estado inicial fijo.

Reducciones y más reducciones

Teorema [Hooper, 66]. Problema de la inmortalidad de MT

El lenguaje de las codificaciones de MT para las cuales existe una configuración inicial no aceptada no es decidible.

Detalle: No se requieren símbolos blancos, no se requiere comenzar a un estado inicial fijo.

$\triangleright$ Primero mostraremos que un problema de inmortalidad de funciones afines por pedazos no es decidible. Por reducción al problema de inmortalidad de MT.

Teorema [Hooper, 66]. Problema de la inmortalidad de MT

El lenguaje de las codificaciones de MT para las cuales existe una configuración inicial no aceptada no es decidible.

Detalle: No se requieren símbolos blancos, no se requiere comenzar a un estado inicial fijo.

▷ Primero mostraremos que un problema de inmortalidad de funciones afines por pedazos no es decidible. Por reducción al problema de inmortalidad de MT.

▷ Desde ese problema deduciremos que el problema de dominó no es decidible. La prueba que presentaremos es de [Kari, 07]. Aunque esencialmente ya se conocía a partir de otra construcción de Kari del '92.

Inmortalidad de funciones racionales afines por pedazos.

▷ Sea $U_i \subset \mathbb{R}^2$ un conjunto de la forma $[n, n+1] \times [m, m+1]$ para $n, m \in \mathbb{Z}$. Consideremos una función $f_i : U_i \to \mathbb{R}^2$ tal que $f_i(x) = A_i x + b_i$ para $A_i \in \mathcal{M}_{2\times 2}(\mathbb{Q})$ y $b_i \in \mathbb{Q}^2$.

Inmortalidad de funciones racionales afines por pedazos.

$\triangleright$ Sea $U_i \subset \mathbb{R}^2$ un conjunto de la forma $[n, n+1] \times [m, m+1]$ para $n, m \in \mathbb{Z}$. Consideremos una función $f_i : U_i \to \mathbb{R}^2$ tal que $f_i(x) = A_i x + b_i$ para $A_i \in \mathcal{M}_{2\times 2}(\mathbb{Q})$ y $b_i \in \mathbb{Q}^2$.

$\triangleright$ Sea I un conjunto finito y f_i funciones de la forma anterior tal que los U_i son disjuntos. Una función $f : \bigcup_{i\in I} U_i \to \mathbb{R}^2$ tal que $f(x) = f_i(x)$ para $x \in U_i$ es una **función racional afín por pedazos**.

Inmortalidad de funciones racionales afines por pedazos.

▷ Sea $U_i \subset \mathbb{R}^2$ un conjunto de la forma $[n, n+1] \times [m, m+1]$ para $n, m \in \mathbb{Z}$. Consideremos una función $f_i : U_i \to \mathbb{R}^2$ tal que $f_i(x) = A_i x + b_i$ para $A_i \in \mathcal{M}_{2\times 2}(\mathbb{Q})$ y $b_i \in \mathbb{Q}^2$.

▷ Sea I un conjunto finito y f_i funciones de la forma anterior tal que los U_i son disjuntos. Una función $f : \bigcup_{i\in I} U_i \to \mathbb{R}^2$ tal que $f(x) = f_i(x)$ para $x \in U_i$ es una **función racional afín por pedazos**.

▷ Un punto $x \in \bigcup_{i\in I} U_i$ se denomina *inmortal* si $f^n(x) \in \bigcup_{i\in I} U_i$ para todo $n \in \mathbb{N}$.

Inmortalidad de funciones racionales afines por pedazos

Teorema [Kari, 07]. Problema de la inmortalidad de MT

El lenguaje de las codificaciones de funciones racionales afines por pedazos que admiten un punto inmortal no es decidible.

Inmortalidad de funciones racionales afines por pedazos

Teorema [Kari, 07]. Problema de la inmortalidad de MT

El lenguaje de las codificaciones de funciones racionales afines por pedazos que admiten un punto inmortal no es decidible.

[Prueba en la pizarra]

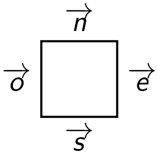
El problema de dominó.

▷ Reduciremos el problema de dominó al problema de inmortalidad de funciones racionales afines por pedazos.

El problema de dominó.

▷ Reduciremos el problema de dominó al problema de inmortalidad de funciones racionales afines por pedazos.

Sean $\overrightarrow{n}, \overrightarrow{o}, \overrightarrow{s}, \overrightarrow{e} \in \mathbb{R}^2$ y f una función afín. Consideremos la baldosa:

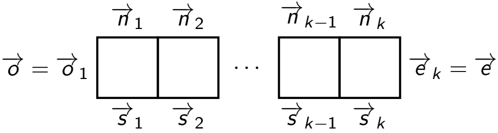


Decimos que la baldosa calcula f si

$$f(\overrightarrow{n}) + \overrightarrow{o} = \overrightarrow{s} + \overrightarrow{e}$$

El problema de dominó.

▷ La idea es que si pegamos varias baldosas horizontalmente tendremos:



Cómo f es afín y dado que $\vec{e}_i = \vec{o}_{i+1}$ para $i = 1 \dots k-1$ tenemos que:

$$f\left(\frac{\vec{n}_1 + \cdots + \vec{n}_k}{k}\right) + \frac{1}{k}\vec{o} = \frac{\vec{s}_1 + \cdots + \vec{s}_k}{k} + \frac{1}{k}\vec{e}.$$

El problema de dominó.

Tomemos $f_i : [n, n+1] \times [m, m+1] \to \mathbb{R}^2$ tal que $f_i(x) = M_i x + b_i$. Queremos codificarla como baldosas de Wang.

El problema de dominó.

Tomemos $f_i : [n, n+1] \times [m, m+1] \to \mathbb{R}^2$ tal que $f_i(x) = M_i x + b_i$. Queremos codificarla como baldosas de Wang.

Para $x \in \mathbb{R}^2$, denotemos

$$A_k(x) = \lfloor kx \rfloor$$

$$B_k(x) = A_k(x) - A_{k-1}(x) = \lfloor kx \rfloor - \lfloor (k-1)x \rfloor.$$

El problema de dominó.

Tomemos $f_i : [n, n+1] \times [m, m+1] \to \mathbb{R}^2$ tal que $f_i(x) = M_i x + b_i$. Queremos codificarla como baldosas de Wang.

Para $x \in \mathbb{R}^2$, denotemos

$$A_k(x) = \lfloor kx \rfloor$$

$$B_k(x) = A_k(x) - A_{k-1}(x) = \lfloor kx \rfloor - \lfloor (k-1)x \rfloor.$$

Notemos que:

▷ $(B_k(x))_{k \in \mathbb{Z}}$ es una secuencia de valores enteros en $\{(n, m), (n, m+1), (n+1, m), (n+1, m+1)\}$.

$$\triangleright \lim_{k \to \infty} \frac{\sum_{j \in [-k,k]} B_j(x)}{2k+1} = x.$$

El problema de dominó.

Tomemos $f_i : [n, n+1] \times [m, m+1] \to \mathbb{R}^2$ tal que $f_i(x) = M_i x + b_i$. Queremos codificarla como baldosas de Wang.

El problema de dominó.

Tomemos $f_i : [n, n+1] \times [m, m+1] \to \mathbb{R}^2$ tal que $f_i(x) = M_i x + b_i$. Queremos codificarla como baldosas de Wang.

Construyamos las siguientes baldosas $x \in \mathbb{R}^2$ y $k \in \mathbb{Z}$.

$$
\begin{array}{ccc}
 & B_k(x) & \\
\begin{array}{c} f_i(A_{k-1}(x)) - A_{k-1}(f_i(x)) \\ +(k-1)b \end{array} & \boxed{} & \begin{array}{c} f_i(A_k(\vec{x})) - A_k(f_i(x)) \\ +kb \end{array} \\
 & B_k(f_i(x)) &
\end{array}
$$

El problema de dominó.

$$
\begin{array}{ccc}
 & B_k(x) & \\
\begin{array}{c} f_i(A_{k-1}(x)) - A_{k-1}(f_i(x)) \\ +(k-1)b \end{array} & \boxed{} & \begin{array}{c} f_i(A_k(\vec{x})) - A_k(f_i(x)) \\ +kb \end{array} \\
 & B_k(f_i(x)) &
\end{array}
$$

Propiedades [ejercicio]:

- La baldosa calcula f, es decir: $f(\vec{n}) + \vec{o} = \vec{s} + \vec{e}$.
- Los cuatro valores están acotados uniformemente para (x, k). Es decir, existe una cantidad finita de baldosas dada la función f_i.
- Se pueden construir efectivamente desde una descripción de f_i.

El problema de dominó.

Sea f una función afín racional por pedazos y sea τ_f el juego de baldosas finito asociado. Debemos mostrar que f admite un punto inmortal si y solamente si τ_f tesela el plano.

El problema de dominó.

Sea f una función afín racional por pedazos y sea τ_f el juego de baldosas finito asociado. Debemos mostrar que f admite un punto inmortal si y solamente si τ_f tesela el plano.

[Prueba en la pizarra]

El problema de dominó.

Acabamos de probar (mod detalles) que:

Teorema: Berger 66'

El problema de dominó no es decidible.

A continuación analizaremos otro problema relacionado...

Definiciones

Sea $G \curvearrowright X$ una acción de grupo.

Definición

$x \in X$ se dice *periódico* si $\mathrm{Stab}(x) := \{g \in G \mid gx = x\} \neq \{1_G\}$.

Definiciones

Sea $G \curvearrowright X$ una acción de grupo.

Definición

$x \in X$ se dice *periódico* si $\mathrm{Stab}(x) := \{g \in G \mid gx = x\} \neq \{1_G\}$.

Definición

Una acción $G \curvearrowright X$ se dice *libre* si no tiene puntos periódicos.

Definiciones

Sea $G \curvearrowright X$ una acción de grupo.

Definición

$x \in X$ se dice *periódico* si $\mathrm{Stab}(x) := \{g \in G \mid gx = x\} \neq \{1_G\}$.

Definición

Una acción $G \curvearrowright X$ se dice *libre* si no tiene puntos periódicos.

Definición

Un subshift se dice *fuertemente aperiódico* si la acción del shift es libre.

Aperiodicidad

Consideremos la clase de $\mathbb{Z}^d$-SFTs.

Pregunta

¿Existe un $\mathbb{Z}^d$-SFT no vacío que sea fuertemente aperiódico?

Aperiodicidad

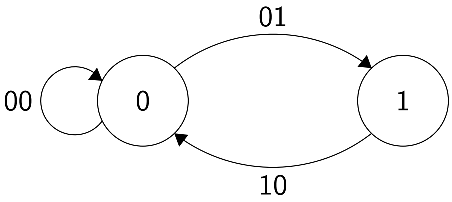
Consideremos la clase de $\mathbb{Z}^d$-SFTs.

Pregunta

¿Existe un $\mathbb{Z}^d$-SFT no vacío que sea fuertemente aperiódico?

Caso fácil $d = 1$.

no.



Aperiodicidad: Caso $\mathbb{Z}^2$

Asumamos primero que la respuesta es también no. Es decir, que todo $\mathbb{Z}^2$-SFT admite un punto periódico.

$\triangleright$ En particular, sería válido para teselados de Wang.

Aperiodicidad: Caso $\mathbb{Z}^2$

Asumamos primero que la respuesta es también no. Es decir, que todo $\mathbb{Z}^2$-SFT admite un punto periódico.

$\triangleright$ En particular, sería válido para teselados de Wang.

Lema

Si un teselado de Wang X admite un punto periódico x, entonces también admite una órbita finita.

Aperiodicidad: Caso $\mathbb{Z}^2$

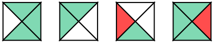
Lema

Si un teselado de Wang X admite un punto periódico x, entonces también admite una órbita finita.

Prueba

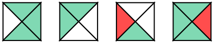
- Supongamos $x \in X$ tiene periodo $v \in \mathbb{Z}^2 \setminus \{\vec{0}\}$. Consideremos el conjunto $Y \subset X$ de configuraciones v-periódicas en X.
- Y es un SFT no vacío. Se obtiene al prohibir en X los patrones con soporte $\{\vec{0}, v\}$ que admiten símbolos distintos.
- La acción de $\mathbb{Z}^2$ no es fiel en Y, luego el sistema define una acción de $\mathbb{Z}^2/v\mathbb{Z} \cong \mathbb{Z} \oplus F$ para un grupo abeliano finito F.
- Como $[\mathbb{Z} \oplus F : \mathbb{Z}] < \infty$ la subacción de $\mathbb{Z}$ sobre este subshift define un $\mathbb{Z}$-SFT, que admite un periodo $u \in v^{\perp}$.

Sea τ un conjunto de baldosas de Wang.

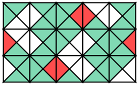


Aperiodicidad y dominós

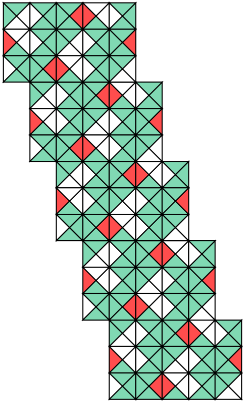
Sea τ un conjunto de baldosas de Wang.



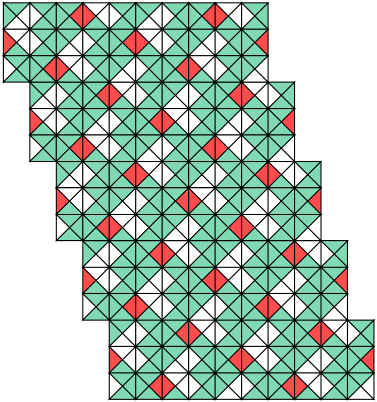
Si comenzamos a generar todos los rectángulos admisibles y eventualmente encontramos uno cuyas aristas coinciden (mod rotación) podemos utilizarlo para generar un teselado del plano.



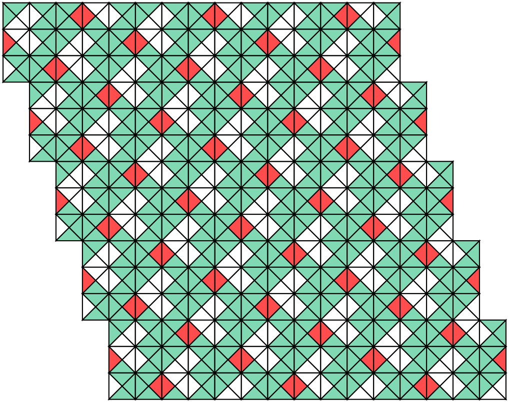
Aperiodicidad y dominós



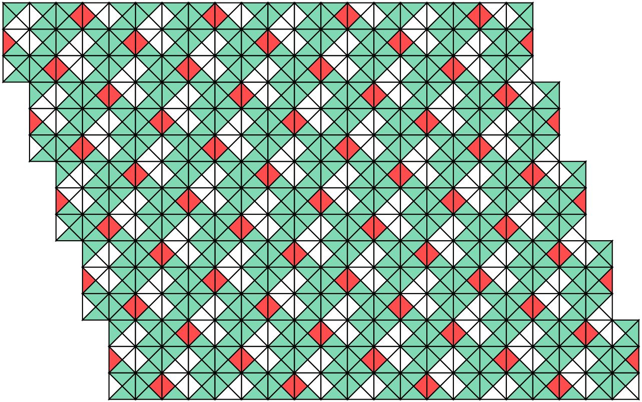
Aperiodicidad y dominós



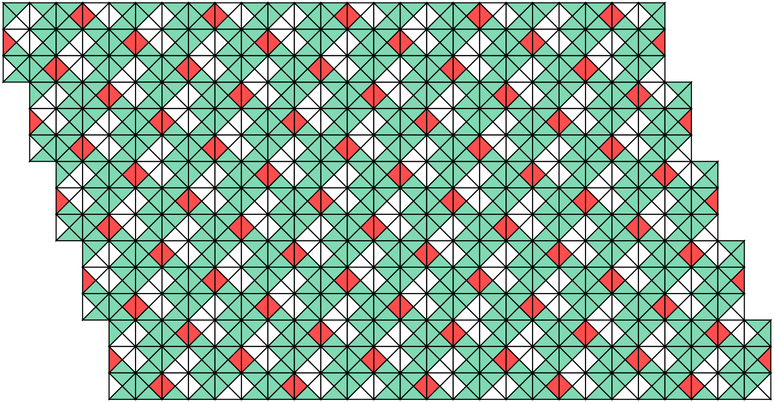
Aperiodicidad y dominós



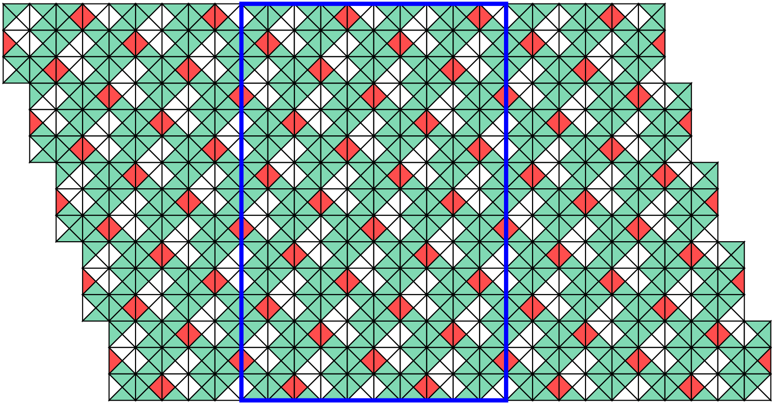
Aperiodicidad y dominós



Aperiodicidad y dominós



Aperiodicidad y dominós



Lema

Si todo teselado de Wang X admite una órbita finita. Entonces el problema de dominó es decidible.

Sabemos que es co-RE, basta mostrar que es RE.

Aperiodicidad y dominós

Lema

Si todo teselado de Wang X admite una órbita finita. Entonces el problema de dominó es decidible.

Sabemos que es co-RE, basta mostrar que es RE.

Algoritmo

1. Instanciar $n \leftarrow 0$.
2. Actualizar $n \leftarrow n + 1$ y construir $\Lambda_n = [-n, n]^d$.
3. Para cada teselado de baldosas válido $p \in \mathcal{A}^{\Lambda_n}$:
   - Verificar si sus aristas coinciden (mod rotación). Si es así, aceptar.
4. Volver al paso 2.

Pero... ¿no habíamos mostrado que no es decidible?

Aperiodicidad y dominós

Teorema [Berger, 66]

Existen $\mathbb{Z}^2$-SFTs fuertemente aperiódicos.

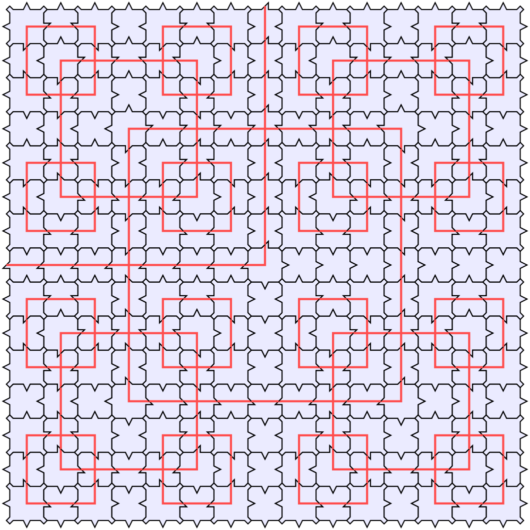
Aperiodicidad y dominós

Teorema [Berger, 66]

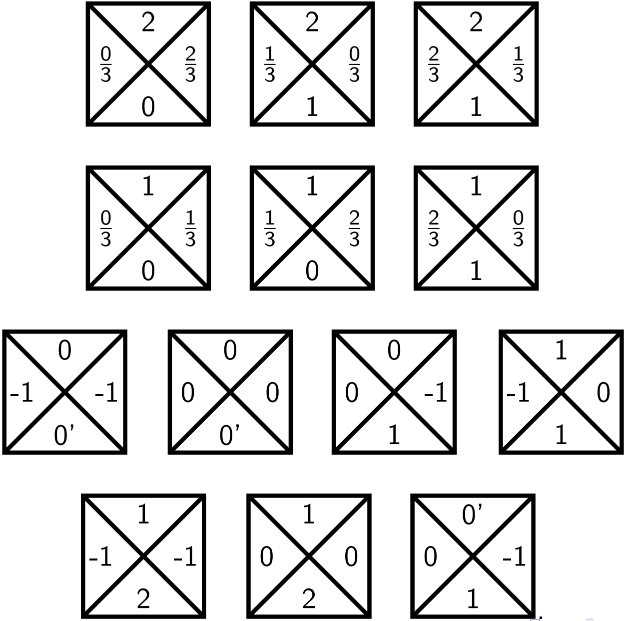
Existen $\mathbb{Z}^2$-SFTs fuertemente aperiódicos.

De hecho, se pueden hacer construcciones explícitas.

Teselado de Robinson



Teselado de Kari (y Culik)



Teselado de Jeandel Rao (11 baldosas)