

Validez de los mosaicos basados en el autómata celular lineal regla 110*

Abdiel E. Cáceres González [†]

Universidad Juárez Autónoma de Tabasco, DACB

Idelfonso Izquierdo Márquez [‡]

Universidad Juárez Autónoma de Tabasco, DACB

Se encontrarán definiciones fundamentales de autómatas celulares y diagramas espacio-temporales, para entonces describir los mosaicos R_{110} , posteriormente ahondaremos en el análisis de estos patrones gráficos, analizando su contorno con el fin de analizar cubrimientos basándose solamente en su contorno, ya que es computacionalmente más barato que analizar la imagen.

Will be fundamental definitions of cellular automata and space-temporary diagrams, by then to describe the R_{110} tiles, later we will go deep in analysis of these graphical patterns, analyzing its contour with the purpose of only analyzing covers being based on its contour, since computationally he is cheaper than to analyze the image.

Palabras clave: Algoritmos de decisión, teoría de mosaicos, Autómata celular regla 110.

Keywords: Decision algorithms, Theory of tiles, Rule 110 cellular automaton.

1. Introducción

El estudio de los autómatas celulares ha sido una actividad dentro de las ciencias computacionales que desde mediados del siglo XX (1940's) ha captado la atención de muchos investigadores [1, 2]. Más recientemente, se han hecho muchos estudios relacionados con el autómata celular lineal conocido como "regla 110", $ac1-R_{110}$ para abreviar [7]. La razón es porque parece ser el autómata celular más simple, que muestra en sus evoluciones tal complejidad, que es posible hacer computación universal. El término "computación universal" se refiere a que por medio del $ac1-R_{110}$ es posible codificar cualquier algoritmo que resuelva problemas Turing computables, porque se ha demostrado que se puede codificar el $ac1-R_{110}$ de modo que sea equivalente a un sistema de Post [8], y también se ha demostrado que un sistema de Post es equivalente a una Máquina Universal de Turing [3, 4].

Por otro lado, las evoluciones del $ac1-R_{110}$ se pueden apreciar en el diagrama espacio-temporal, que es un arreglo de las evoluciones globales, colocando en el renglón superior la configuración inicial y apilando hacia abajo las siguientes configuraciones¹. En el diagrama espacio-temporal del $ac1-R_{110}$ se aprecian patrones

*Recibido el 15 de marzo de 2007 y aceptado el 30 de abril de 2007

[†]**Dirección postal:** Carr. Cunduacán-Jalpa Km 1, Cunduacán Tabasco, México. A.P. 24 C.P. 86690. Tel. (+52)914 336-0928. **Correo electrónico:** abdielc@acm.org

[‡]**Dirección postal:** Carr. Cunduacán-Jalpa Km 1, Cunduacán Tabasco, México. A.P. 24 C.P. 86690. Tel. (+52)914 336-0928. **Correo electrónico:** idelfonso.3@hotmail.com

¹Para una completa descripción de los diagramas espacio-temporales de los autómatas celulares

gráficos en forma de triángulos –véase la figura 1–. Estos patrones gráficos triangulares son el objeto de estudio de este artículo, y han sido estudiados desde finales de la década de 1990 [9, 12].

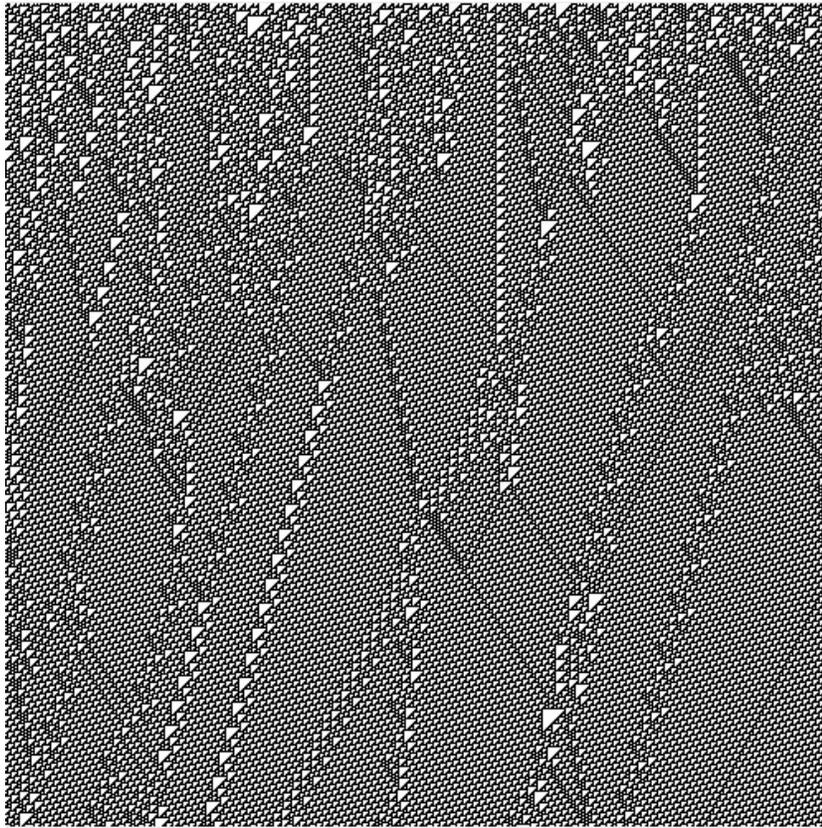


Figura 1. Diagrama espacio-temporal del $ac1-R_{110}$, iniciando con una configuración global aleatoria (distribución uniforme) de 500 células y a través de 500 generaciones

En esos estudios [9, 12], se ha relacionado los diagramas espacio-temporales del $ac1-R_{110}$ con un rompecabezas armado de muchas piezas, algunas de ellas iguales, pero que en conjunto pueden cubrir el espacio plano. Este punto de vista, el de cubrir el plano con los “*mosaicos que son obtenidos del diagrama espacio-temporal del $ac1-R_{110}$* ” ($mosaico-R_{110}$, $mosaicos-R_{110}$), ha causado que los $mosaicos-R_{110}$ tengan relación con otros problemas computacionales como el Problema de Heesch [5]; el problema del dominó, conocido también como La Conjetura de Wang [6].

En este artículo, estudiaremos cómo codificar los $mosaicos-R_{110}$, de modo que sea computacionalmente más barato almacenarlos y manipularlos. Estudiaremos entonces lo que llamaremos *contornos regla-110* ($contorno-R_{110}$ ó $contornos-R_{110}$ para abreviar). Que brevemente, son secuencias de cadenas que describen el contorno de un patrón en \mathbb{Z}^2 , garantizando que al considerar solamente el contorno de la figura, no se pierda información del interior.

lineales, refiérase a [2]

Para entender cómo se componen los mosaicos, agregaremos una nueva definición a la colección de definiciones relacionadas con **triángulos- R_{110}** definidos en [12], la de los *triángulos completos*, que nos será de mucha ayuda para determinar el contorno más grande que se puede encontrar en un diagrama espacio-temporal determinado (del **ac1- R_{110}**). Hacia el final del artículo describiremos un algoritmo para obtener todos los contornos de un mosaico con la ayuda de una máquina de Moore².

2. Definiciones fundamentales

Esta sección se ha puesto con el fin de describir algunos elementos que serán utilizados con frecuencia en este artículo, si el lector está familiarizado con estos elementos, puede seguir en la siguiente sección, o bien, tomar esta sección como referencia y regresar a esta siempre que sea necesario.

2.1 Autómata celular lineal regla 110

Los autómatas celulares lineales (**ac1**) son descripciones de sistemas dinámicos discretos, definidos en un espacio compuesto por una sucesión finita de celdas conocidas como células; en principio la sucesión de células es infinita en ambas direcciones, sin embargo por necesidades computacionales se considera un espacio finito.

Un **ac1** (G, T) es un sistema compuesto por una secuencia $G = (s_i)_{0 \leq i \leq n-1}$ que evoluciona a través del tiempo T en pasos discretos. Cada s_i es la i -ésima **célula** de G , donde $s_i \in K$, a su vez $K = \{0, 1, \dots, k-1\}$ es un conjunto finito, numerable y normalmente $|K| \ll \infty$, K se denomina el **conjunto de estados** de G .

La dinámica de los autómatas celulares comienza con la aplicación de una función que transforma un estado global del autómata celular en otro (posiblemente el mismo) Φ , cuya firma es $\Phi : K^n \rightarrow K^n$, donde K^n es el conjunto de todas las secuencias de n estados, tomados del conjunto K . La función Φ se conoce como **función de evolución global**.

La función Φ se determina mediante la aplicación de una función ϕ conocida como **función de evolución local**, cuya firma es $\phi : K^3 \rightarrow K$, cada célula $s_i \in G$ tiene asociado un conjunto de células llamado la **vecindad** de s_i . La vecindad ν de s_i , está determinada por la célula inmediata anterior, la célula s_i , y la célula inmediata siguiente, lo que podemos describir con la función vecindad $\nu : K \rightarrow K^3$. De modo que para cada $s_i \in G$:

$$\nu(s_i) = \begin{cases} (s_{n-1}, s_1, s_2) & \text{si } i = 0; \\ (s_{i-1}, s_i, s_{i+1}) & \text{si } 0 < i < n-1; \\ (s_{n-2}, s_{n-1}, s_1) & \text{si } i = n-1; \end{cases}$$

Esta definición de la función vecindad, establece las condiciones periódicas en las fronteras del espacio celular G .

De este modo, la dinámica del autómata celular se establece considerando un

²Para leer la descripción original de la máquina de Moore, revise la referencia [13]

espacio celular inicial $G^{(0)}$ y haciendo

$$\begin{aligned} G^{(1)} &= \Phi(G^{(0)}) = \Phi^1(G^{(0)}) = (\phi \circ \nu(s_i^{(0)}))_{0 \leq i \leq n-1} \\ G^{(2)} &= \Phi(G^{(1)}) = \Phi^2(G^{(0)}) = (\phi \circ \nu(s_i^{(1)}))_{0 \leq i \leq n-1} \\ G^{(3)} &= \Phi(G^{(2)}) = \Phi^3(G^{(0)}) = (\phi \circ \nu(s_i^{(2)}))_{0 \leq i \leq n-1} \\ &\vdots \\ G^{(t)} &= \Phi(G^{(t-1)}) = \Phi^t(G^{(0)}) = (\phi \circ \nu(s_i^{(t-1)}))_{0 \leq i \leq n-1} \end{aligned}$$

Cada una de las aplicaciones de Φ se conoce como **evoluci3n**, y comunmente decimos que el aut3mata celular lineal ha evolucionado un paso, y en el mismo sentido, decimos **evolucionar** a la acci3n de aplicar Φ a una configuraci3n global G^t , para cualquier $t \geq 0$

Donde $G^{(t)}$ indica la configuraci3n global del aut3mata celular lineal en el paso de tiempo t ; y del mismo modo, $s_i^{(t)} \in K$, indica el estado de la c3lula i en el paso de tiempo t . Sea ahora $K = \{0, 1\}$ el conjunto de estados de las c3lulas del aut3mata celular, y ϕ la funci3n

$$\phi = \begin{pmatrix} 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Entonces decimos que $\Phi_{r_{110}}$ es la funci3n global de un aut3mata celular regla 110³.

Al tomar un espacio celular de tama1o finito $n = 500$; un estado global inicial del **ac1** como

$$G^{(0)} = (s_0, s_1, \dots, s_{n-1});$$

donde cada $s_i = \text{rand}(2)$, $0 \leq i < n$ con una distribuci3n de probabilidades uniforme, esperando que aproximadamente la mitad de las c3lulas tenga el valor 0 y el resto tenga el valor 1; y evolucionar el **ac1** hasta $t = 500$, podemos observar comportamientos similares al que se muestra en la figura 1.

La figura 1 es un diagrama espacio-temporal de las evoluciones del **ac1-R₁₁₀**, es decir, una gr3fica que muestra una a una, las configuraciones del **ac1-R₁₁₀** que fueron consideradas desde un tiempo inicial $t = 0$ hasta un tiempo final $t = 500$ en este caso. El diagrama espacio-temporal muestra la historia del **ac1-R₁₁₀**.

2.2 Mosaicos regla 110 (mosaicos-R₁₁₀)

Al observar con m3s detalle el diagrama espacio-temporal –v3ase la figura 2–, notamos que es un plano cubierto de figuras pseudo triangulares⁴ de diferentes tama1os, pero que conservan la misma orientaci3n y forma, haciendo que este **ac1** en particular, sea objeto de an3lisis, con el fin de responder a cuestionamientos tales como: 3cu3l es la causa de que estos tri3ngulos ocurran? y 3tendr3 alg3n significado computacional?

³El n3mero 110 viene porque los d3gitos ‘01110110’ del codominio de la funci3n se pueden leer como una palabra en binario que se traduce como ‘110’ en decimal

⁴En realidad no son tri3ngulos, pues est3n presentes en un espacio bidimensional discreto, lo que hace que su lado de hipotenusa se vea como “escalera”, adem3s de que estas figuras est3n semi-bordeadas por todos lados menos por su diagonal, a pesar de esto, acordaremos llamarlas “tri3ngulos”, mas bien por conservar la nomenclatura usual en esta 3rea.

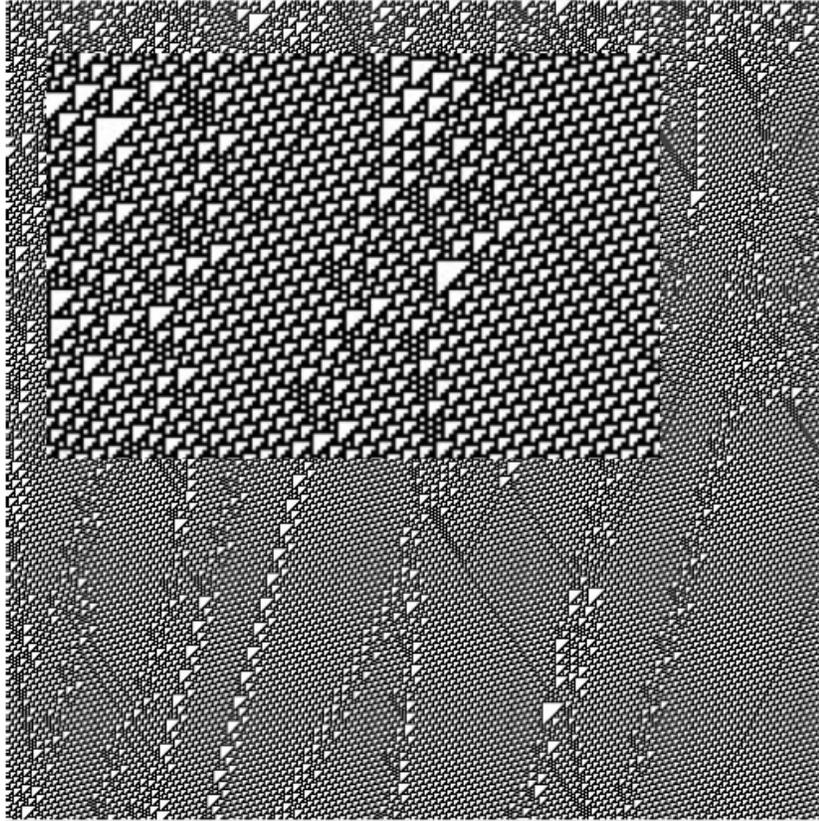


Figura 2. Acercamiento en el diagrama espacio-temporal del $ac1-R_{110}$

De modo que podemos ver los diagramas espacio-temporales como una imagen digital [14], es decir, un área rectangular compuesta por elementos pictóricos que pueden ser simbólicamente 1 ó 0.

Consideremos ahora el conjunto Ψ^n que contiene todos los diagramas espacio-temporales del $ac1-R_{110}$ con un espacio de n células y un tiempo de evolución de $t = n$. Con esto, logramos coleccionar imágenes digitales de tamaño $n \times n$ puntos binarios.

En consecuencia, llamemos Ψ^* al conjunto de todos los diagramas espacio temporales de longitud finita $n \geq 3$. El tamaño mínimo debe ser 3 puesto que se requieren de 3 células para que la función de vecindad ν esté definida.

Una línea de investigación de hace algunos años atrás [15] es describir un procedimiento efectivo que permita decidir si se puede cubrir el plano con un conjunto de figuras triangulares, de modo que las figuras no se roten ni se traslapen y que, por supuesto, no dejen huecos. En otras palabras, crear un cubrimiento del plano Γ de modo que para algún diagrama espacio-temporal $\psi \in \Psi^n$, se cumpla que

$$\psi \subseteq \Gamma,$$

lo que se puede interpretar como “el diagrama espacio-temporal está contenido en el cubrimiento Γ ”.

En principio, llamaremos **triángulo- R_{110}** a las figuras triangulares del diagrama espacio-temporal del **ac1- R_{110}** , descritas informalmente de la siguiente manera:

Un **triángulo- R_{110}** de tamaño n lo denotaremos como t_n , y es una figura en el plano que consta de cuatro partes, y cada una de estas partes se ilustra en la figura 3 parte (B), pero las podemos describir como:

1. 1 célula de estado 0 llamada **el origen** del triángulo
2. una secuencia de n células de estado 0 extendidas horizontalmente, llamada la **frontera superior** del triángulo.
3. una secuencia de n células de estado 0 extendidas verticalmente, llamada la **frontera izquierda** del triángulo.
4. Un conjunto triangular de células de estado 1 de tamaño n , que se extiende de mayor a menor y de arriba hacia abajo, lo que conoceremos como el **cuerpo** del triángulo. Esta sección da pie a llamar **diagonal** a las células que forman la hipotenusa del triángulo.

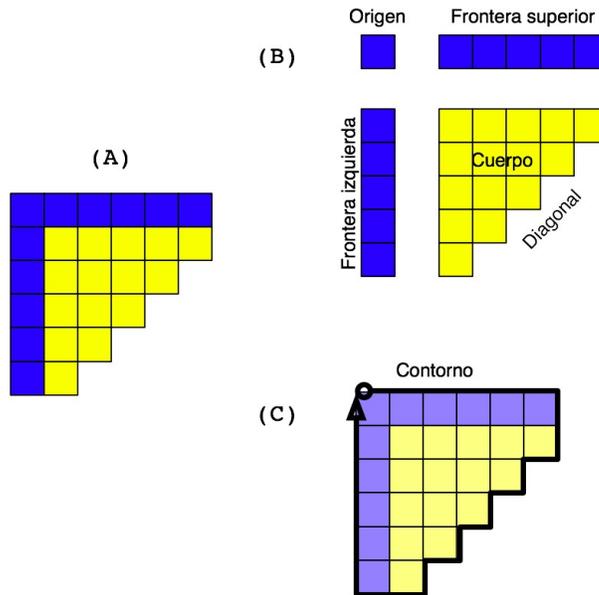


Figura 3. (A) **triángulo- R_{110}** t_5 ; (B) Descomposición en sus secciones principales: La célula origen, las fronteras superior e izquierda, su diagonal y el cuerpo del triángulo; (C) El contorno del **triángulo- R_{110}** , Convencionalmente se dirige en el sentido de las manecillas del reloj, pero no es una regla.

Un diagrama espacio-temporal es entonces un plano de extensión finita que está completamente cubierta por **triángulos- R_{110}** , aunque al parecer algunos de ellos están incompletos, situación que se detalla más adelante en este artículo. En la figura 4 se muestra un detalle del diagrama espacio-temporal de la figura 1, en donde hemos

destacado un triángulo t_{10} , un t_5 y un t_6 ; podemos, sin embargo, descubrir triángulos de otros tamaños.

En especial el t_0 es un triángulo que está compuesto de una célula origen y tanto sus fronteras superior e izquierda son de longitud 0, y el cuerpo del t_0 también es de tamaño 0, de modo que un t_0 se compone únicamente de una célula, la del origen.

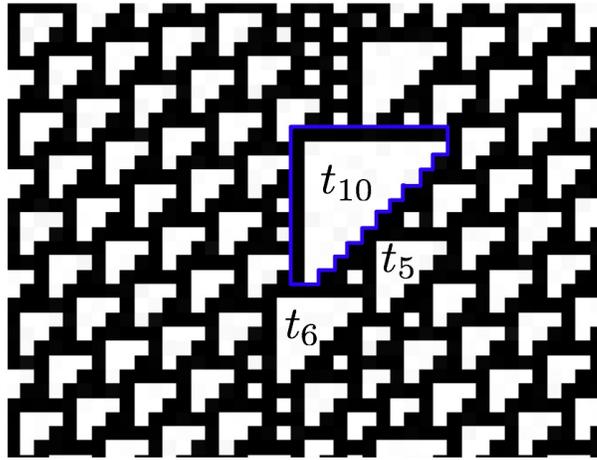


Figura 4. Detalle del diagrama espacio-temporal del $a1-R_{110}$ identificando tres triángulos- R_{110} : el contorno de un t_{10} , un t_5 y un t_6

2.3 Contornos R_{110}

Cada triángulo- R_{110} tiene una extensión en el espacio bidimensional discreto que está enmarcada por una división conceptual que llamaremos **contorno**. En la figura 3 describimos el contorno del t_5 que utilizamos como ejemplo, y en la figura 4, el t_{10} aparece con su contorno destacado.

Para codificar el contorno, utilizaremos el conjunto $D = \{\mathbf{n}, \mathbf{s}, \mathbf{e}, \mathbf{w}\}$ para designar la dirección de cada segmento de línea del contorno, comenzando desde la parte más arriba y más a la izquierda del triángulo y continuando en el sentido de las manecillas del reloj, como se ilustra en la parte (C) de la figura 3.

La longitud de cada segmento del contorno es de una unidad, justo del tamaño de cada célula, de modo que por ejemplo, un t_0 se compone únicamente de 4 segmentos que rodean a la única célula que lo compone.

Con esta línea de pensamiento, diremos que un contorno es una secuencia $c = (c_1, c_2, \dots, c_l)$, donde cada $c_i \in D$ y $l < \infty$. Los contornos- R_{110} tiene la propiedad:

1. Son curvas cerradas, esto se puede verificar comprobando el número de \mathbf{n} 's, \mathbf{s} 's, \mathbf{e} 's y \mathbf{w} 's que tiene la secuencia. Si hacemos que $\text{num}(\mathbf{n})$ sea el número de \mathbf{n} 's que ocurren, y del mismo modo $\text{num}(\mathbf{s})$ la cantidad de \mathbf{s} 's; $\text{num}(\mathbf{e})$ la cantidad de \mathbf{e} 's y $\text{num}(\mathbf{w})$ el número de \mathbf{w} 's, entonces

$$\text{num}(\mathbf{s}) + \text{num}(\mathbf{e}) = \text{num}(\mathbf{w}) + \text{num}(\mathbf{n})$$

- Se pueden distinguir tres partes en la secuencia, una de ellas poblada de e 's, que indican la extensión del **triángulo- R_{110}** ; la parte central compuesta por sw 's que forman la diagonal; y finalmente una serie de n 's que forman la frontera izquierda.

Para ilustrar los contornos, el contorno del t_{10} de la figura 4 está codificado como $c_{t_{10}} = eeeeeeeeeesswswswswswswswswswswswnnnnnnnnnnn$, el contorno de un t_0 es $c_{t_0} = eswn$

2.4 Extensión natural de los mosaicos- R_{110} y los contornos- R_{110}

Hasta ahora describimos tanto los mosaicos como los contornos de manera tal que el interior de ellos describe únicamente un **triángulo- R_{110}** . Sin embargo podemos describir una extensión natural de ellos, permitiendo que los mosaicos cubran más espacio, involucrando más de un **triángulo- R_{110}** .

Llamaremos **mosaico- R_{110}** (compuesto) a una región del diagrama espacio-temporal de un **acl- R_{110}** que se compone de más de un **triángulo- R_{110}** . En este sentido, los **mosaicos- R_{110}** que se componen solamente de un triángulo (de cualquier tamaño) los llamaremos **mosaico- R_{110}** (unitarios), y en general, **mosaico- R_{110}** será una región del diagrama espacio-temporal que se compone de uno o más **triángulo- R_{110}** .

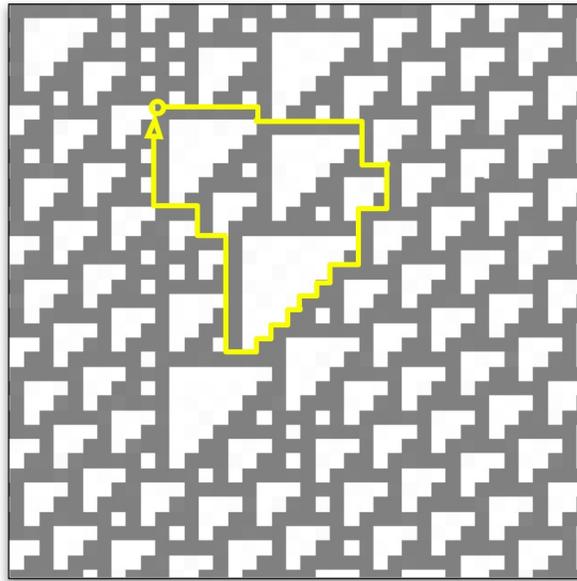


Figura 5. Contorno de un **mosaico- R_{110}** que se compone de 15 **triángulos- R_{110}** , incluyendo los t_0 's, que en la figura se aprecian con cuadros en tono más obscuro, por convención, los t_0 siempre ocurren en la diagonal de otros triángulos de mayor tamaño.

Los siguientes conjuntos serán de mucha utilidad en el transcurso del artículo.

$\mathcal{M}_{r110}^{(0)}$: El conjunto de todos los *mosaicos- R_{110}* con 0 triángulos

$\mathcal{M}_{r110}^{(1)}$: El conjunto de todos los *mosaicos- R_{110}* con 1 triángulo

⋮

$\mathcal{M}_{r110}^{(*)}$: El conjunto de todos los *mosaicos- R_{110}* con 0 o más triángulos

3. Validez de los *mosaicos- R_{110}*

Teorema 1. Un *contornos- R_{110}* es válido si puede ser completamente cubierto con uno o más *mosaicos- R_{110}* sin traslapes, rotaciones, escalamientos.

Prueba. Este teorema será probado con el algoritmo de la página 33, que se encarga de decidir si un contorno es válido o no, es decir, que la respuesta de este algoritmo es 1 si el contorno es válido y 0 si el contorno no es válido. ■

3.1 Descripción de la prueba de validez

Primero, suponemos que las cadenas dadas como parámetros para este algoritmo, representan curvas cerradas en el plano discreto, de modo que los llamaremos *contornos*.

Consideremos un contorno $c = \delta_{i_0} \delta_{i_1} \dots \delta_{i_{k-1}}$, donde cada $\delta_i \in D$, y deseamos saber si ese contorno es un contorno válido, de modo que aplicamos la función `esValido(c)`, el resultado es o bien `#t` si es válido o `#f` si el contorno c no es válido.

El argumento principal del algoritmo es ir quitando triángulos, de modo que si al final, cuando ya no se puedan seguir eliminando triángulos, el contorno resultante es la cadena vacía, entonces se comprueba la validez del contorno inicial; por otro lado, si cuando ya no se puedan eliminar más triángulos del contorno, la cadena resultante no es vacía, se comprueba que el contorno no es válido.

Decidir qué triángulo quitar y en qué posición del contorno debe hacerse, es trabajo de una Máquina de Mealy-Moore (porque tiene acciones tanto en los estados como en las aristas) que reconoce la primera ocurrencia de la cadena `neks`.

```
-----
Algoritmo: esValido !- Decide si una cadena en D* es un contorno r110
Datos de entrada:
  contorno: Cadena de caracteres !- caracteres en D={n,s,e,w}
Datos de salida:
  Valido: Booleano !- #T si es valido, #F si no es valido.
-----
Inicio
Hasta terminar
  contorno<-reinicia-a-posicion-maxima(contorno)
```

```

triangulo-obtenido<-obtener-triangulo(contorno)
contorno<-elimina-triangulo(contorno,triangulo-obtenido)
si contorno=NULLO entonces esValido<-#t de otro modo esValido=#f
Fin
-----

```

3.2 Las subsecuencias $ne^k s$ determinan triángulos completos

La subsecuencia $ne^k s$ es sumamente importante para determinar el tamaño del triángulo a remover. La cadena $ne^k s$ es única para cada triángulo, de acuerdo a la siguiente afirmación.

Teorema 2. Si c es un contorno, de tal modo que $c = a \cdot ne^k s \cdot b$, para dos secuencias $a, b \in \{n, s, e, w\}^*$, entonces se determina con precisión el triángulo de tamaño k .

Prueba. La cadena de k direcciones e significan un tramo horizontal, de acuerdo a la regla de evolución del autómata celular regla 110, establece que la alineación de triángulos por el lado superior es imposible, pues así lo muestra la relación $(000 \rightarrow 1) \in \phi$ (véase la página 28). De modo que la longitud de un tramo horizontal establece el tamaño del triángulo, luego, por la construcción del triángulo, queda perfectamente determinada su altura, también de longitud k , y el lado diagonal con secuencia $s(sw)^{k-1}w$, de modo que el contorno del triángulo resultante es $e^k s(sw)^{k-1}wn^k$. ■

4. Discusión final

Existe, sin embargo una clase de subsecuencias que podríamos llamar *contornos nulos*, que informalmente son aquellas secuencias de direcciones que encierra un área nula, como por ejemplo el contorno $neeeewws$. Esta clase de secuencias son muy comunes en la prueba de validez, sin embargo no hay problema con ellos, pues son *anulables*. No ocurre lo mismo con contornos de la forma $\alpha_1 \cdot ne^{k_1} \beta e^{k_2} s \cdot \alpha_2$, donde las α 's y β con secuencias en $\{e, n, s, w, \}$, pero además, β es anulable. Esto ocasiona que no se determine con precisión el triángulo, porque existe una subsecuencia anulable que destruye la premisa del teorema 2.

El estudio de los *mosaicos- R_{110}* involucra distintas áreas de conocimiento tanto en las ciencias computacionales como en matemáticas, principalmente con geometría, matemáticas discretas y topología digital. Desde la perspectiva de cada una de esas áreas se pueden hallar temas de estudio.

Uno de los temas en estudio es determinar un algoritmo que reciba como entrada la descripción de un *mosaico- R_{110}* , dada por su contorno; y como resultado, devuelva una configuración inicial de un *ac1- R_{110}* , de tal modo que el *mosaico- R_{110}* propuesto, ocurra dentro del diagrama espacio-temporal del *ac1- R_{110}* cuando inicia con la configuración global inicial obtenida en la respuesta. Aunque en [12] se describe una manera de hacerlo, utilizando los diagramas de deBruijn [16] que es una herramienta basada en grafos útil para determinar las preimágenes de las evoluciones de los autómatas celulares, sin embargo esta idea aún no se ha implementado.

Otro tema en estudio es crear un procedimiento efectivo para determinar si un

mosaico de tamaño k , donde k es el número de triángulos- R_{110} que lo compone, es capaz de cubrir por completo el plano discreto, haciendo uso únicamente de copias del mismo mosaico. Este problema hace recordar el “Problema de Heesch” [16] y el número de Heesch para un determinado mosaico. El número de Heesch es la cantidad de capas que es posible colocar al rededor de un mosaico, de modo que un mosaico con un número de Heesch infinito, significa que cubre el plano por completo, y en cambio, un mosaico con número de Heesch igual a 0, significa que no se puede crear ni siquiera una capa al rededor de este mosaico.

Referencias

- [1] Von Neumann, John. *The Theory of Self-reproducing Automata*, A. Burks, ed., Univ. of Illinois Press, Urbana, IL. (1966)
- [2] Wikipedia contributors, *Cellular automaton*, Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Cellular_automaton&oldid=104993332. Accesado el 2 de febrero de 2007.
- [3] Davis, Martin. *Computability and Unsolvability*. McGraw-Hill Book Company, Inc, New York . (1958)
- [4] Marvin, Minsky. *Computation: Finite and Infinite Machines*, Prentice-Hall, Inc., N.J., 1967
- [5] Eppstein, David. *The Geometry Junkyard-The Heesch Problem*. Página de Internet disponible en URL: <http://www.ics.uci.edu/~eppstein/junkyard/heesch/>, accedido el 2 de febrero de 2007, actualizada el 14 de enero de 2007.
- [6] Weisstein, Eric W. *Wang's Conjecture*. From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/WangsConjecture.html>. Accesado el 2 de febrero de 2007.
- [7] Wikipedia contributors, *Rule 110 cellular automaton*, Wikipedia, The Free Encyclopedia, 24 January 2007, 03:30 UTC, http://en.wikipedia.org/w/index.php?title=Rule_110_cellular_automaton&oldid=102828255. Accesado el 2 de febrero de 2007.
- [8] Cook, Mattew. *Universality in Elementary Cellular Automata*, Complex Systems, Volume 15, Number 1, p. 1-40, 2004.
- [9] McIntosh, Harold V. *Rule 110 as it relates to the presence of gliders*, <http://delta.cs.cinvestav.mx/~mcintosh/comun/RULE110W/rule110.pdf>, January 1999.
- [10] McIntosh, Harold V. *Rule 110*. Universidad Autónoma de Puebla, México, Tech. Report available at <http://delta.cs.cinvestav.mx/mcintosh/oldweb/pautomata.html>, 1999.
- [11] McIntosh, Harold V. *Linear Cellular Automata Via de Bruijn Diagrams*. Universidad Autónoma de Puebla, México, Tech. Report available at <http://cellular.ci.ulsat.mx/comun/cf/debruijn.pdf> [Aug 10, 1999]
- [12] Caceres-González, Abdiel E. *Máquina celular de computación basada en mosaicos Regla 110*. Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Departamento de Ingeniería Eléctrica, Sección de Computación. 2005

- [13] Moore, Edward F. *Gedanken-experiments on Sequential Machines*, pp 129–153, Automata Studies, Annals of Mathematical Studies, no. 34, Princeton University Press, Princeton, N. J., 1956.
- [14] Rosenfeld, A. *Picture languages*. Academic Press, Inc., USA, 1979.
- [15] Palash, Sarkar. *A brief history of cellular automata*. ACM Computing Surveys, 32(1):pp80–107, 2000.
- [16] Heesch, H. *Reguläres parkettierungsproblem*. Westdeutscher Verlag. Cited in Tiling and Patterns by B. Grunbaum and Geoffrey C. Shephard. 1986 (1968).