

Introducción a computación no-convencional y autómata celular

Genaro Juárez Martínez

<http://uncomp.uwe.ac.uk/genaro/>

Unconventional Computing Group
Faculty of Computing, Engineering and Mathematical Sciences
University of the West of England
Bristol, United Kingdom

iGEM-México
Marzo 27 del 2008.

Se presenta una introducción general de la teoría de la computación, el concepto de computación no-convencional, autómatas celulares (AC) y algunos recientes resultados para simular procesos con AC. Esta introducción se enfoca al problema de llegar a simular procesos biológicos con AC y su proyección a procesos computables en ambientes no-lineales, para el grupo de investigación iGEM-México.

Part I

Computación

Procesos computables

Un proceso computable es aquel que puede ser descrito iterativamente por una sucesión de pasos, originando dos implicaciones:

1. Que procesos pueden ser descritos?
2. Existen procesos que pueden estar bien definidos pero que no pueden ser descritos completamente?

es decir, que procesos son computables y no computables.

De esta manera, si tenemos un conjunto de procesos computables entonces todos ellos pueden ser descritos a través de un algoritmo.

Computación clásica

La *teoría de lenguajes formales* nos da un esquema para estudiar conjuntos de cadenas desde un alfabeto finito. Los lenguajes pueden ser vistos como entradas de algunas clases de máquinas o como el producto final de un sistema de sustituciones.

lenguaje	estructura
recursivamente enumerables	máquina de Turing
sensitivos de contexto	autómata de límite lineal
libres de contexto	autómata de pila
regular	autómata finito

Table: Clases de lenguajes y máquina asociada.

El modelo básico de estas máquinas para estos lenguajes y necesaria para toda computación, es la máquina de Turing. La importancia de asociar una máquina o sistema para determinar cada tipo de lenguaje nos lleva a una clasificación de lenguajes (jerarquía de Chomsky).

- ▶ Dentro del estudio para llevar a cabo computaciones es posible lograrlo desde la implementación de mecanismos abstractos como son: máquinas de Turing, sistemas Post, redes neuronales, máquinas finitas, compuertas lógicas, sistemas de Lyndermayer o cualquier sistema computacionalmente equivalente Turing.
- ▶ Un aspecto interesante en teoría de la computación es el estudio de sistemas *universales*, elementos universales, máquinas universales y finalmente sistemas mínimos universales.

¿Qué es computación no-convencional?

Un esquema de computación que actualmente es visto como no-convencional debe ser porque su desarrollo aún no es disponible o terminado [*Toffoli 1998*].

Computación no-convencional

Computación no-convencional podemos decir que es investigada y desarrollada en tres grandes ciencias. Además, la mencionada computación no-convencional es relacionada al naciente concepto de *computación natural* y sistemas complejos.

Tres ciencias convergen

1. Computación en física cuántica (computadora 7-qubit)
2. Computación química (computadora de reacción y difusión)
3. Computación biológica (computadora auto-ensamblada)

Computación no-convencional

Ilustramos la implementación e interpretación de una compuerta MAJORITY en un quantum-dot basado en AC. Cada célula cuántica cambia de polarización (estado) al recibir un estímulo.

célula cuántica - dot

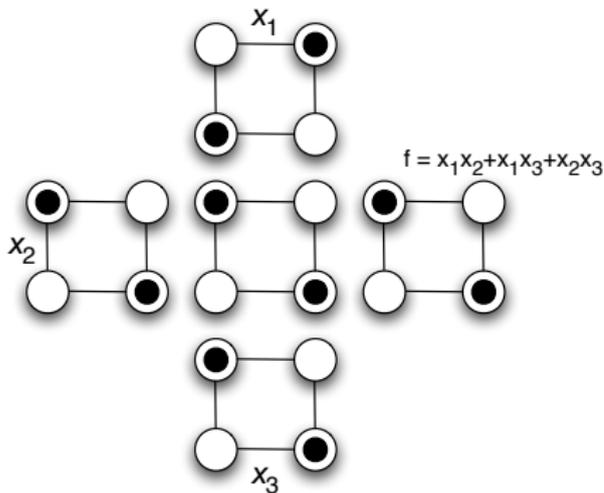
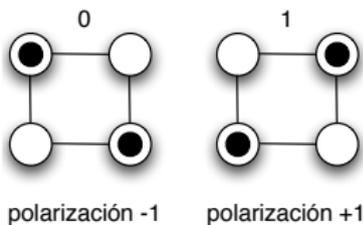


Figure: Implementación de una compuerta MAJORITY en un quantum-dot AC.

Part II

Autómata Celular

Autómata celular

Definición

Un *autómata celular* es un sistema dinámico discreto evolucionando en un arreglo regular infinito.

La dinámica es la siguiente. Tenemos un conjunto finito de estados (nuestro alfabeto) y una función local afectando una cantidad de estados dentro del arreglo regular, donde cada elemento del arreglo toma un valor del conjunto de estados y es llamada una *célula*. El arreglo regular es mejor conocido como el *espacio de evoluciones* dentro de la literatura de autómata celular (AC). La función local es importante porque determina el comportamiento del AC en estudio. La *función local* es determinada por una *célula central* y sus células vecinas, formando una *vecindad*. La vecindad es el número de argumentos que la función local recibe. Finalmente las transformaciones determinadas para cada vecindad diferente corresponden a un elemento del conjunto de estados. Como ingrediente final necesitamos una asignación de estados en el espacio de evoluciones que representa la *configuración inicial* del sistema.

Estructura general

Dinámica en AC

Tipos de AC

Los autómatas de Wolfram

Simulación de procesos con AC

Procesos con AC

Referencias

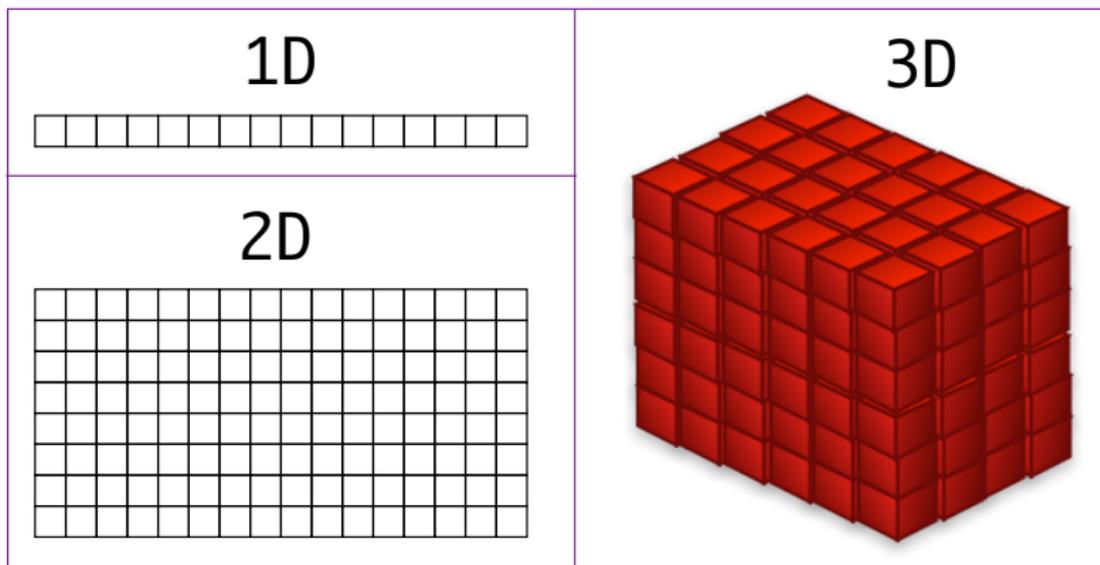
Software libre en AC

Entonces podemos definir formalmente un AC de la siguiente manera:

Definición

Un *autómata celular* es una 4-tupla $A = \langle \Sigma, u, \varphi, c_0 \rangle$ evolucionando en $d \in \mathbb{Z}^+$ dimensión, con un conjunto finito de estados Σ , una conexión local u tal que $u = \{x_{i,j}, \dots, n:d\}$, la función de transición $\varphi : u \rightarrow \Sigma$ y la condición inicial del sistema c_0 .

Algunos espacios de evoluciones



Estructura general

Dinámica en AC

Tipos de AC

Los autómatas de
Wolfram

Simulación de procesos con AC

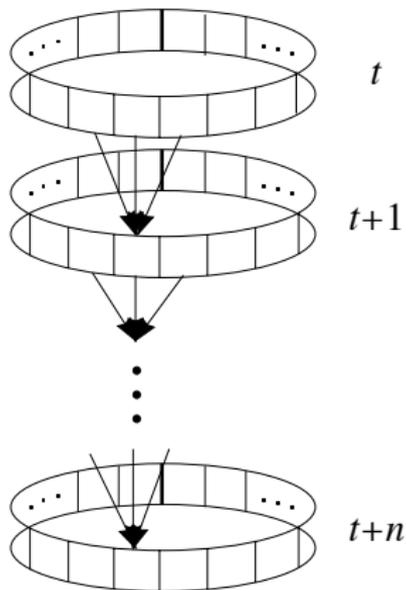
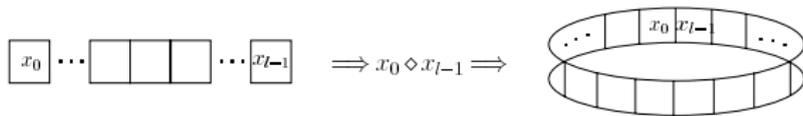
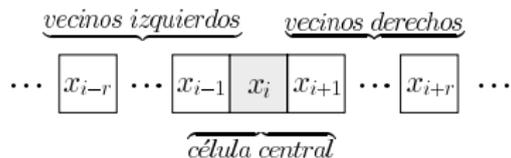
Procesos con AC

Referencias

Software libre en
AC

Figure: AC puede evolucionar en alguna d -dimensión.

Dinámica en una dimensión



Estructura general

Dinámica en AC

Tipos de AC

Los autómatas de
Wolfram

Simulación de
procesos con AC

Procesos con AC

Referencias

Software libre en
AC

Dinámica en dos dimensiones

En dos dimensiones tenemos un arreglo definido por las parejas ordenadas $\mathbb{Z} \times \mathbb{Z}$. De igual manera que una dimensión, cada una de las células $x_{i,j}$ toma un elemento del conjunto de estados Σ .

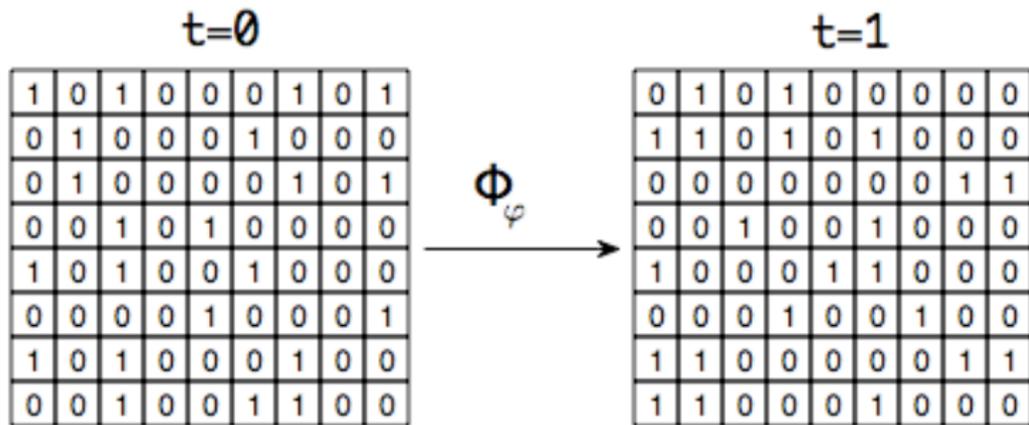
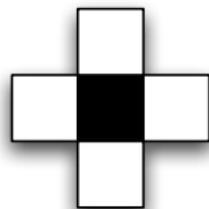


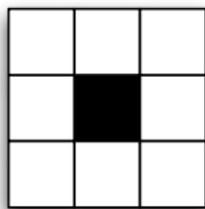
Figure: Espacio de evoluciones en dos dimensiones.

Tipos de vecindad

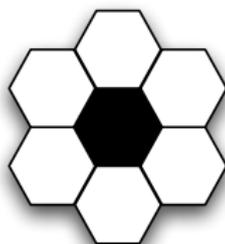
La regla de evolución puede encontrar algunas variantes a través de la historia. Sin embargo, solo consideramos las más conocidas. Iniciamos con la *vecindad de von Neumann* definida para una célula central y sus vecinos ortogonales (figura a). La *vecindad de Moore* que además de los vecinos ortogonales se consideran los vecinos diagonales (figura b). Por último la *vecindad hexagonal* (figura c).



(a)



(b)



(c)

Figure: Tipos de vecindad en dos dimensiones.

Clases de Wolfram

Wolfram realiza todo un estudio sistemático en AC de una dimensión. Una de sus principales aportaciones son las conocidas *clases de Wolfram*. Karel Culik II define estas clases de la siguiente manera:

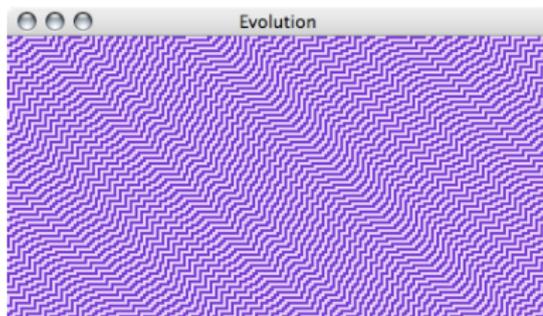
Clasificación

- ▶ Un AC es de clase I si existe un estado estable $x_i \in \Sigma$ tal que todas las configuraciones finitas x_i evolucionan a la *configuración homogénea* de x_i .
- ▶ Un AC es de clase II si existe un estado estable $x_i \in \Sigma$ tal que cualquier configuración finita tiene una evolución que llega a ser periódica.
- ▶ Un AC es de clase III si existe un estado estable $x_i \in \Sigma$ tal que para alguna pareja de configuraciones finitas x_i de c_i y c_j , es decidible si c_i evoluciona a c_j .
- ▶ Un AC es de clase IV si incluye todo AC.

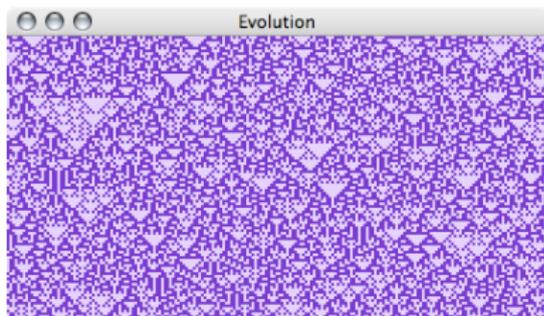
Clases de Wolfram



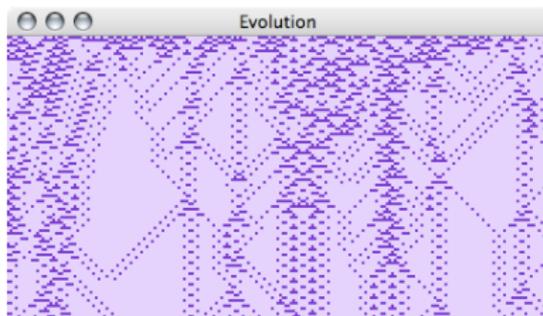
Regla 0



Regla 15



Regla 90



Regla 54

Figure: Cuatro clases en AC elemental de una dimensión.

Procesos con AC en biología

Simulación de un patrón como luminiscencia basado en la regla de difusión. El fenómeno es reproducido a través de pequeñas colonias de células vivas convergiendo en osciladores de período dos.

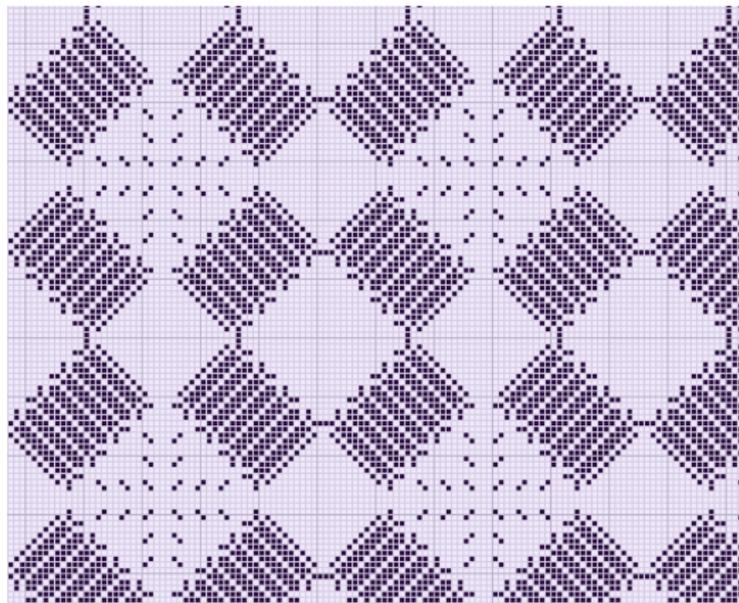


Figure: Simulando un patrón como luminiscencia con la regla de difusión.

Procesos con AC en química o biología

Simulación de patrones espirales con AC en dos dimensiones, hexagonal y con tres estados: *activador*, *inhibidor* y *refractorio*.



Figure: DDLAB simulación definiendo patrones espirales.

Procesos con AC en computación

Avances en AC con tres estados simulando compuertas lógicas a través de un sistema de partículas móviles y estacionarias.

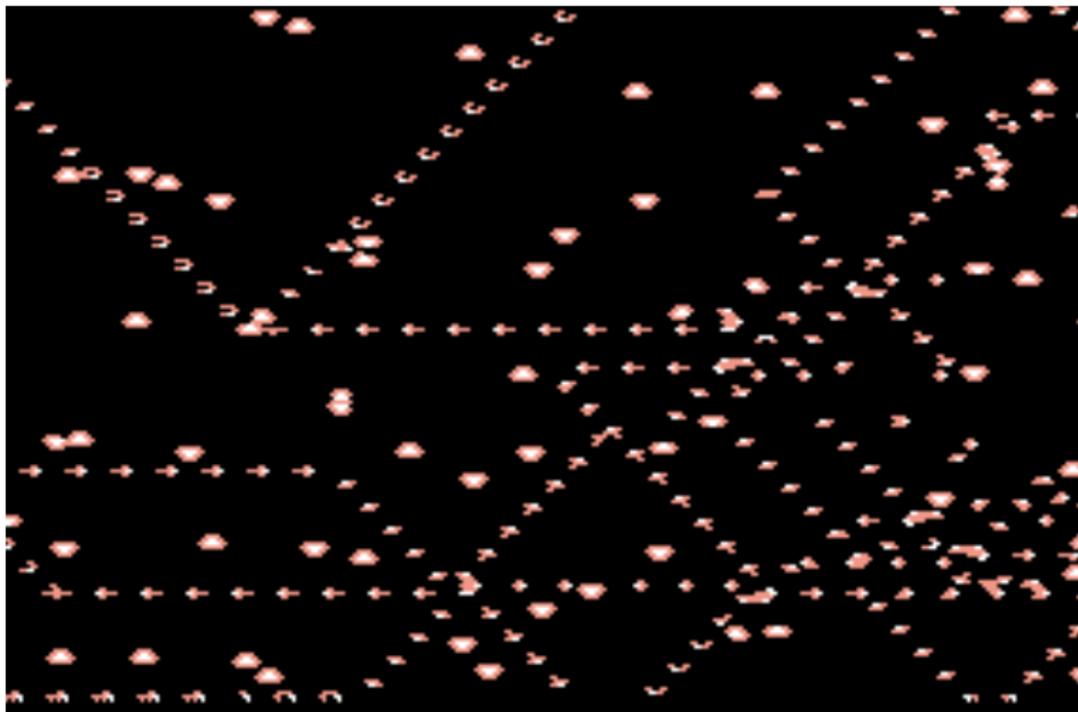


Figure: Regla espiral – un AC hexagonal bidimensional.

Estructura general

Dinámica en AC

Tipos de AC

Los autómatas de
Wolfram

Simulación de
procesos con AC

Procesos con AC

Referencias

Software libre en
AC



John E. Hopcroft and Jeffrey D. Ullman
*Introduction to Automata Theory Languages, and
Computation*
Addison-Wesley Publishing Company, 1987.



Marvin Minsky
Computation: Finite and Infinite Machines
Prentice Hall, 1967.



Andrew Adamatzky
Computing in Nonlinear Media and Automata Collectives
Institute of Physics Publishing, Bristol and Philadelphia, 2001.



Tommaso Toffoli and Norman Margolus
Cellular Automata Machines
The MIT Press, Cambridge, Massachusetts, 1987.

Estructura general

Dinámica en AC

Tipos de AC

Los autómatas de
Wolfram

Simulación de
procesos con AC

Procesos con AC

Referencias

Software libre en
AC

Referencias



Stephen Wolfram

A New Kind of Science

Wolfram Media, Inc., Champaign, Illinois, 2002.



A. Deutsch and S. Dormann

Cellular Automaton Modeling of Biological Pattern Formation

Birkhäuser Boston, 2004.



Tommaso Toffoli

Non-Conventional Computers

Encyclopedia of Electrical and Electronics Engineering (J. Webster ed.), Wiley & Sons (1998).



Genaro Juárez Martínez, Andrew Adamatzky and Ben De Lacy Costello

On logical gates in precipitating medium: cellular automaton model

arXiv:0802.3891v1 [nlin.CG], February 2008.

Referencias

-  Andrew Adamatzky and Andrew Wuensche
Computing in 'spiral rule' reaction-diffusion cellular automaton
Complex Systems 16/4, 2006.
-  Andrew Adamatzky, Genaro Juárez Martínez, and Juan C. Seck Tuoh Mora
Phenomenology of reaction-diffusion binary-state cellular automata
Int. J. of Bifurcation and Chaos 16/10:1–21, 2006.
-  Genaro Juárez Martínez, Andrew Adamatzky and Harold V. McIntosh
Localization dynamic in binary two-dimensional cellular automaton: Diffusion Rule
J. of Cellular Automata, in press.
-  Genaro Juárez Martínez and Pablo Padilla Longoria
Biology patterns constructions with cellular automata in elaboration.

Software libre para trabajar en AC

- ▶ Sistemas “OSXCA”: plataforma Macintosh
<http://uncomp.uwe.ac.uk/genaro/OSXCASystems.html>
- ▶ Andrew Wuensche, sistema “Discrete Dynamics Lab” (DDLAB): plataformas Macintosh, Linux y Windows
<http://www.ddlab.org/>
- ▶ Harold V. McIntosh, sistemas “NXLCAU”: plataformas NeXTSTEP y OpenStep
<http://delta.cs.cinvestav.mx/~mcintosh/oldweb/software.html>
- ▶ Andrew Trevorrow y Tomas Rokicki, sistema “Golly”: plataformas Macintosh, Linux y Windows
<http://golly.sourceforge.net/>