

Análisis de patrones en autómatas celulares Wolfram regla 30

Carlos Manuel Rodríguez Martínez

13 de octubre de 2015

1. Análisis de la fila central del autómata celular regla 30

El autómata celular elemental de regla 30 se construye a partir de las reglas que se muestran en la figura 1. Al utilizar una condición inicial con un recuadro del mapa con valor 1 y evolucionar el sistema se obtiene un mapa como el que se muestra en la figura 2.



Figura 1: Reglas para construir el autómata celular 30. [1]

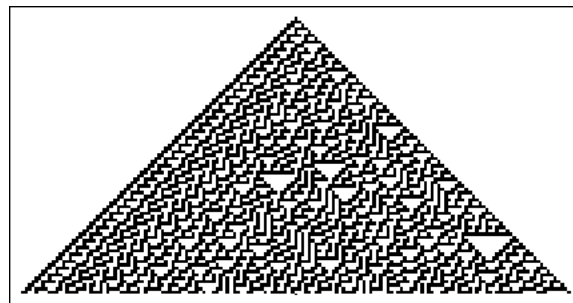
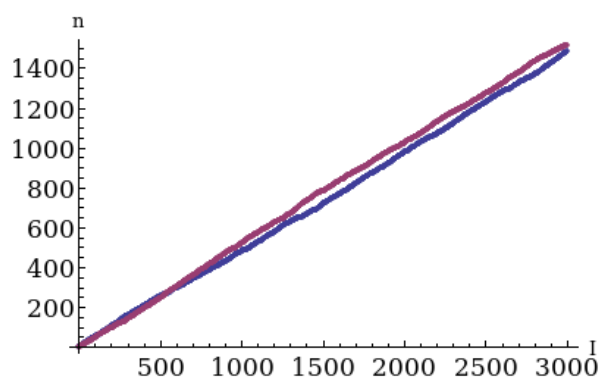
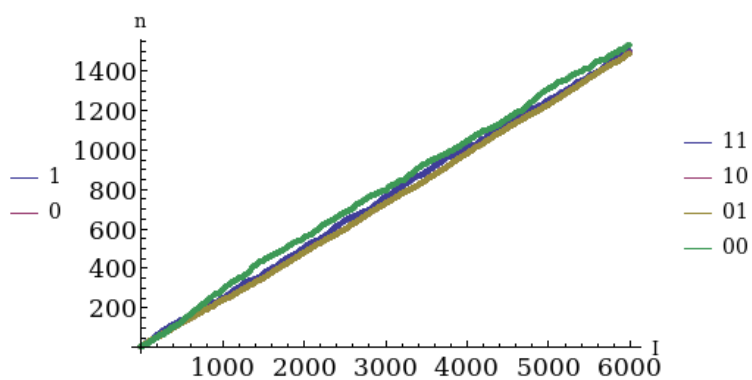


Figura 2: Autómata celular.

La regla 30 del autómata celular elemental tiene la propiedad de que la aparición de ceros y unos es totalmente aleatoria, lo cual lo hace un buen generador de números aleatorios. Se analizó la frecuencia de aparición de unos y ceros en la fila central.

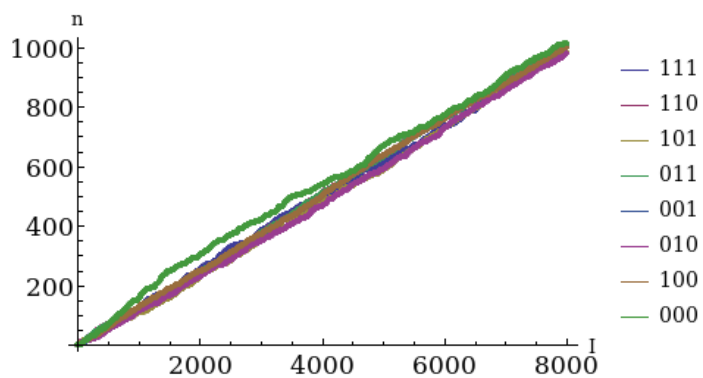


(a) Frecuencia de 0 y 1.



(b) Frecuencia de 00, 01, 10, 11.

Figura 3: Frecuencia de aparición de patrones respecto a iteraciones en AC.



(a) Frecuencia de patrones de 3 dígitos.

Figura 4: Frecuencia de aparición de patrones respecto a iteraciones en AC.

El código utilizado para generar las gráficas es el siguiente.

Listing 1: Código para patrones de dos dígitos

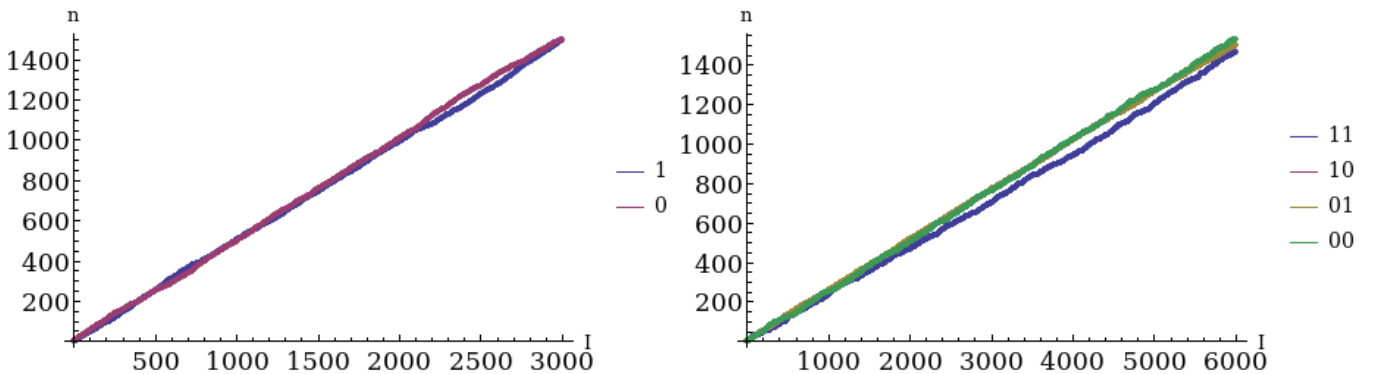
```

1 CountPattern[ list_, pat_] := Count[Partition[ list , Length[pat], 1], pat]
2 lista11 = {};
3 lista10 = {};
4 lista01 = {};
5 lista00 = {};
6 maxiter = 6000;
7 ac = CellularAutomaton[30, {{1}, 0}, maxiter ];
8 aclist = ac[[All, Ceiling[Length[ac] [[1]]/2]]];
9 Do[
10   list = aclist [[1 ;; iter ]];
11   v11 = CountPattern[list, {1, 1}];
12   v10 = CountPattern[list, {1, 0}];
13   v01 = CountPattern[list, {0, 1}];
14   v00 = CountPattern[list, {0, 0}];
15   AppendTo[lista11, v11];
16   AppendTo[lista10, v10];
17   AppendTo[lista01, v01];
18   AppendTo[lista00, v00];
19   , {iter, 1, maxiter}
20 ]
21 ListPlot[{ lista11, lista10, lista01, lista00 },
22 PlotLegends -> LineLegend[{"11", "10", "01", "00"}],
23 AxesLabel -> {Style["l", 12], Style["n", 12]},
24 BaseStyle -> {FontSize -> Scaled[.045]}, ImageSize -> {400, 300}}

```

2. Comparación con generador de números aleatorios

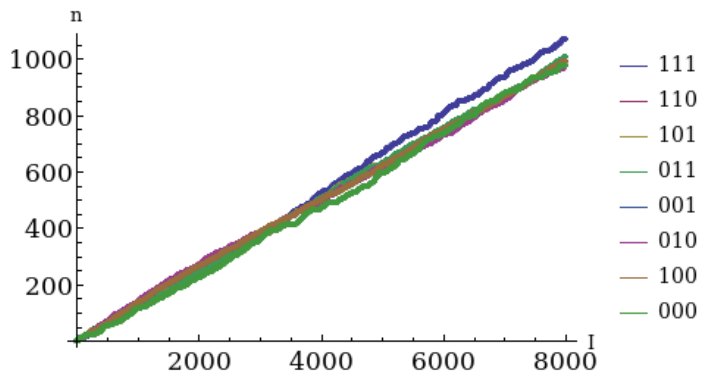
Se realizó una comparación con el algoritmo de generación de números aleatorios conocido como *Mersenne-Twister*. Este algoritmo es uno de los más usados por su velocidad y alto periodo de frecuencia ($2^{19937} - 1$). No se encontraron diferencias significativas al hacer el conteo de la frecuencia de aparición de patrones en el *Mersenne-Twister* respecto al autómata celular.



(a) Frecuencia de 0 y 1.

(b) Frecuencia de 00, 01, 10, 11.

Figura 5: Frecuencia de aparición de patrones respecto a iteraciones en MT.



(a) Frecuencia de patrones de 3 dígitos.

Figura 6: Frecuencia de aparición de patrones respecto a iteraciones en MT.

3. Referencias

- [1] Weisstein, Eric W. «Elementary Cellular Automaton.» From MathWorld—A Wolfram Web Resource.
<http://mathworld.wolfram.com/ElementaryCellularAutomaton.html>