

PRÁCTICA 3

NOTAS AUXILIARES PARA SU CORRECTO DESARROLLO E IMPLEMENTACIÓN

La práctica tiene dos objetivos principales:

- Desarrollar un núcleo de computación de autómatas celulares unidimensional elementales, donde:
 - la regla (función de transición) esté codificada mediante un número natural, utilizando la técnica de síntesis de reglas de Wolfram (<https://mathworld.wolfram.com/TotalisticCellularAutomaton.html>), generalizada a un número de estados $n \geq 2$;
 - sea capaz de computar la evolución espacio-temporal del autómata celular elegido con output gráfico, durante el número de generaciones elegido (ver Figura 1).
- Integrar el núcleo de computación citado en un GUI derivado de los ya desarrollados en prácticas anteriores, de forma que todos los parámetros necesarios para configurar el autómata celular que se desea computar sean introducidos a través del GUI y el output gráfico se muestre también en el GUI.

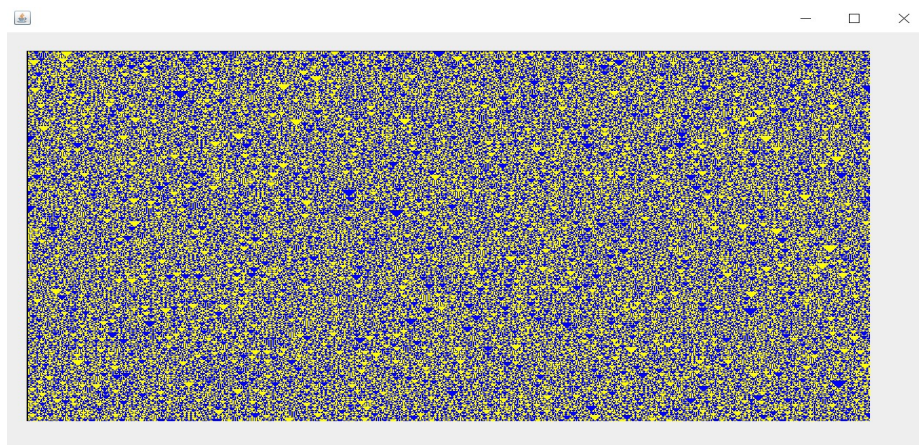


Figura 1: Autómata Celular Unidimensional en Output Gráfico.

¿Cuáles son las entradas que el GUI debería leer? El GUI debería tener campos de entrada para todos los elementos que se requieren para especificar y computar un AC 1-D, y que son:

- Número de estados por célula, mayor o igual que 2.
- Función de transición, codificada mediante un número natural, de acuerdo a las reglas de síntesis descrita en el URL especificado anteriormente, generalizado desde el modelo propuesto en la referencia para dos estados, al número de estados elegido por el usuario
- Tipo de configuración inicial: aleatoria (utilizando algún generador ya implementado en la práctica anterior, a escoger libremente), o célula central activa (a cualquier estado distinto de cero) y resto inactivas (estado cero).
- Condición de Frontera: nula o cilíndrica, a elegir por el usuario.
- Número de Generaciones.

¿Cómo dibujo la evolución espacio-temporal del autómata? Hay múltiples formas utilizando las clases que `java.swing` proporciona para dibujar. Probablemente la más simple sea la que se

ilustra en el código `automataCelularChorra.java` que se le proporciona en la carpeta de la práctica. No obstante lo anterior, se le anima a estudiar métodos alternativos.

¿Qué elementos de control debería tener el GUI ? Realmente pocos:

- Un botón para activar la simulación una vez leídos los datos necesarios.
- Un botón para limpiarlo todo y comenzar una simulación nueva.

¿Qué aspecto debería tener mi GUI? Algo similar a la interfaz que ilustra la Figura 2; esta imagen es **orientativa**, y **en ningún caso necesita seguirla de forma estricta**. Sea creativo.

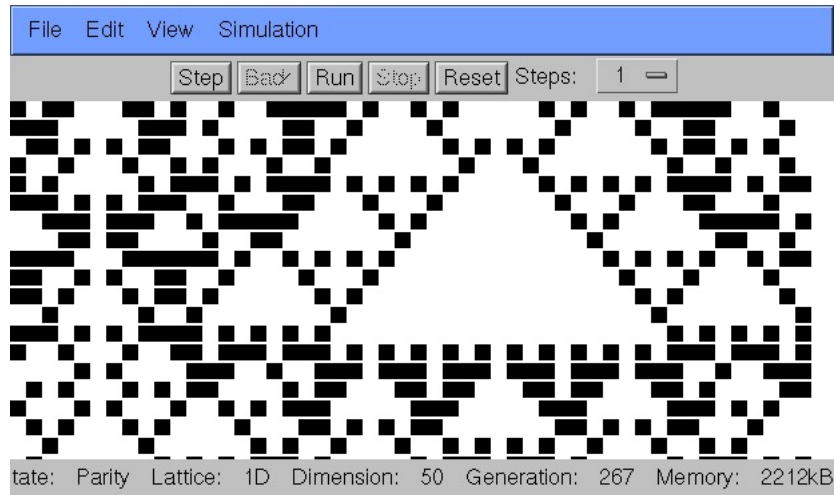


Figura 2: Posible Aspecto del GUI

¿Cómo sé si mi núcleo de computación está bien desarrollado? Bastará confrontar su simulador con unos cuantos autómatas celulares de prueba cuyo output gráfico es conocido (o bien observar el output en modo texto de unas cuantas generaciones para una regla previamente conocida). El URL ya indicado le será muy útil para ello. El “paper” original de Wolfram sobre autómatas celulares, publicado en la revista Nature en 1984, que se le proporciona en la carpeta de la práctica, también. En estas referencias aparecen autómatas celulares con información que incluye el output gráfico y la regla numérica que los codifica.