Tutoría 12
Física Computacional I
Grado en Física

Javier Carrasco Serrano, javcarrasco@madrid.uned.es

Tema 15: Autómatas Celulares Elementales

Adobe Acrobat Document

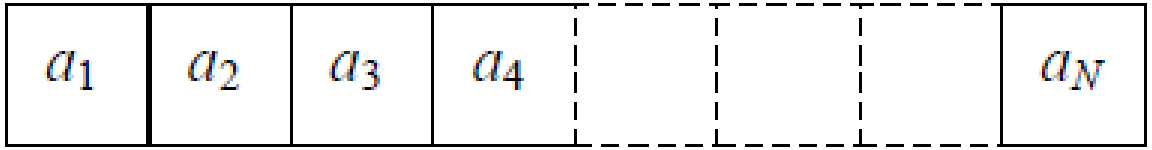
15

15.1. Introducción a los autómatas celulares

- Sistemas dinámicos discretos y deterministas (determinados por la condición inicial) → conjunto finito de elementos idénticos (células o celdas del autómata). El estado de una celda está determinado por el estado anterior de sus celdas "vecinas".
- Reproducen comportamientos complejos a partir de reglas y conjuntos simples → conocer y controlar el proceso.
- Características fundamentales: espacio discreto y homogéneo (dividido en celdas); número discreto y finito de estados; tiempo discreto; actualización simultánea de las celdas; reglas deterministas o estocásticas (probabilidad); reglas locales (dependen de sus vecinos).
- Aplicaciones: plantas (intercambio CO2-O2), redes neuronales, física de superficies (cristalinos, átomos → reacciones químicas y crecimientos), dinámica de poblaciones (depredador-presa), procesos de reacción-difusión (pigmentaciones, desertificación), propagación de ondas en medios excitables (sistema nervioso, red neuronal), física de fluidos (turbulencias),…

15.2. Autómatas celulares elementales

- Mecánica estadística → determinar mediante estadística comportamientos medios de sistemas macroscópicos (grandes con muchos elementos) a partir de elementos microscópicos del sistema. Auto-organización → evolución a sistemas ordenados (no esperados).
- Autómates Celulares Elementales (ACE) → fila de N celdas determinadas por su posición; estado 0 ó 1; celdas vecinas (a_{i-1}, a_i, a_{i+1}); tiempo discreto → regla de actualización R; condición de contorno (a_1 y a_N son vecinas).
- Si una configuración se repite → estado estacionario. Reglas periódicas, reglas fractales.
- Representación binaria de la regla: (10010110) → 1 ó 0 a cada posibilidad (1,1,1),…,(0,0,0).
- Ensamblado con operadores << (desplaza) y | (concatena) + desplazamiento bits de estados con la regla → bit shift.



$$a_i[t+1] = R\left(a_{i-1}[t], a_i[t], a_{i+1}[t]\right)$$

Listado 15.1 → regla 54 (54 en binario). Ejercicio 15.2, 15.3 (uso aleatorios inicialización).

15.3. Propiedades globales

- Reglas complejas → comportamiento caótico.
- Distancia entre dos estados (s) → distancia de Hamming: diferencia de los dos estados (bit a bit; celda a celda; contar iguales y diferentes) → crece con t en reglas complejas (y son reglas muy similares a las simples).
- Irreversibilidad: si diferentes estados evolucionan hacia el mismo (no hay regla inversa) → el número de estados visitados disminuye en t; y el de no visitados aumenta.
- Reversibilidad: si los estados no se "cruzan".
- Auto-organización: selección de ciertos estados a partir de configuraciones iniciales aleatorias.
- Entropía: medida del desorden (o información, estados posibles) del sistema.

$$S(t) = -\frac{1}{N}\sum_i p_i(t)\log_2 p_i(t)$$

- Universalidades: generalizar a ACEs de más de dos estados y más de tres vecinos → agrupaciones de reglas → clasificar comportamientos por grupos → clases de universalidades: 1 (homogéneas), 2 (estacionario no homogéneo), 3 (patrón caótico) y 4 (comportamiento caótico).

Ejercicio 15.4, 15.5, Listado 15.2, 15.3 (representación de simulaciones / reglas), Ejercicio 15.6 - 15.13

Prueba evaluable de programación con Maxima

Corrección Maxima

- Solución del Equipo Docente a principio de Julio → PEC de Septiembre “generalización” de la de Junio, por lo que entender la solución dada por el Equipo Docente es un buen primer paso.
- Errores comunes:
  - Comentarios: ¿cómo introducir los inputs (especialmente si no tienen la forma propuesta)?
  - Batchload no ejecuta (comentarios abiertos, falta de ; entre funciones), sesión de Maxima, programas (que no funciones).
  - EDO lineal y no lineal. ¿Qué es una EDO no lineal? Ejemplo: $e^{f''(x)} + \sin f'(x) + \frac{1}{f(x)^3} = 0$.
  - Condiciones iniciales: sustituir pero no evaluar, cambiar signo a la segunda condición inicial, actualizar mal v_0.
  - ¿Sustituciones x por y ó por x(y)?
  - Función ejercicio 4 con función normal (sin necesidad de block).
  - Sustituir d/dx por d/dx(y) en lugar de por d/dy.

Prueba evaluable de programación con C

Adobe Acrobat
Document

Algunas indicaciones foros generales

- Longitud ejercicio 3:

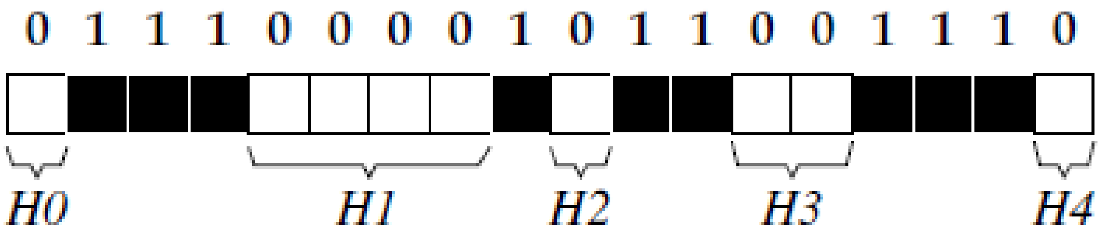
https://2019.cursosvirtuales.uned.es/dotlrn/grados/asignaturas/61041094-19/uforums/thread-view?message_id=43339725

- MUY IMPORTANTE: se pueden estar haciendo asunciones erróneas por no interpretar bien el enunciado. Por eso es muy importante la memoria, porque debe quedar claro que entendéis lo que estáis programando. Puede que hagáis programas estupendos que en algún punto se desvíen un poco de lo que se os está pidiendo porque lo estáis entendiendo de otra manera, por ese es muy importante que si tenéis dudas sobre lo que se os está pidiendo, pongáis cómo interpretáis el enunciado y qué es lo que vais a buscar y obtenéis con vuestros programas.
- Histograma:

https://2019.cursosvirtuales.uned.es/dotlrn/grados/asignaturas/61041094-19/uforums/thread-view?message_id=43567076

Prueba C – Ejercicio 1

- 0 no ocupado, 1 ocupado.
- Disposición aleatoria en un array.
- Porosidad = número 0’s / número 1’s
- ¿Qué se pide? Poroso longitud L y porosidad r dados por el usuario. Guardar en archivo texto y representar con gnuplot. Utilizar simulaciones con L = 250, 1250; r = 0.25, 1, 4. Es decir, tendríamos 6 simulaciones (L,r) = {(250, 0.25), (250, 1), (250, 4), (1250, 0.25), (1250, 1), (1250, 4)}
- ¿Qué quiere decir r=0.25, r=1, r=4?
- Resultado en una tabla de 2x3 gráficas.
- ¿Corte del aleatorio generado en > ó en >=?
- **PROBABILIDAD DE QUE UN SITIO ESTÉ OCUPADO:** $p = \frac{1}{(1+r)} \rightarrow p = ¿?$



Prueba C – Ejercicio 1

Prueba C – Ejercicio 2

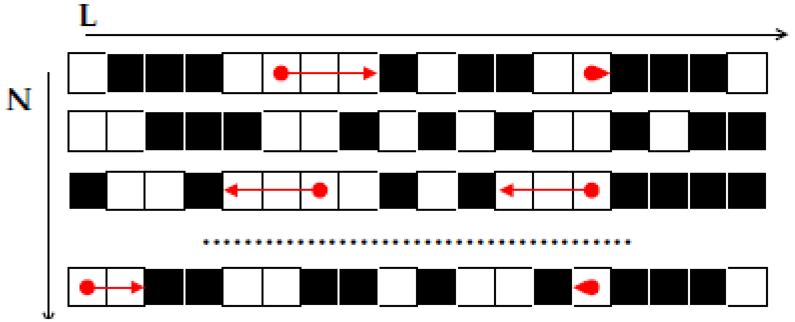
- Longitud de un hueco: número de 0's consecutivos.
- ¿Cómo calcular el número de 0's consecutivos?
- Histograma huecos: número de huecos de tamaño 1, de tamaño 2, … , de tamaño L.
- Histograma: Listado 12.3.
- Con el histograma se puede saber/estimar la probabilidad de que un 0 dado pertenezca a un hueco de longitud l. ¿Cómo? ¿Salida del histograma en número o en porcentaje?
- ¿Qué se pide? Calcular el histograma de la longitud de los huecos de tamaño 1, 2, … , L/2, dados L, r y N (número de cadenas a simular). Utilizar N = 1024, L = 64, 1024, r = 0.25, 1, 4.
- Distribución del histograma ("en el infinito"):

$$h(l, r) = r^l / (1 + r)^{2+l}$$

- ¿Qué sucede para los distintos tamaños (L)?

Prueba C – Ejercicio 2

Prueba C – Ejercicio 3



- Valores medios → media.
- Distancia que se puede recorrer en un 0 elegido al azar (hacia un lado o hacia el otro) hasta chocar con un 1.
- ¿Qué información nos da el histograma que podamos utilizar?
  - Número de cadenas de cada tamaño.
  - Tamaño de cada una de las cadenas → ¿qué sitios puede ocupar el 0?, ¿cuánto recorre?
- ¿Y el histograma teórico?
- Distancia media ≡ longitud de correlación
- ¿Qué se pide? ¿Ser un bruto? Es una posibilidad. ¿Es la única? ¿Es la más eficiente?
- Se pide recorrer cada cadena, y cada vez que se encuentre un 0, contar los 0s que hay a su derecha y a su izquierda.
- Hacerlo para L = 1024, N = 1024, r = 0.1, 0.25, 0.5, 1, 2, 4, 9 → ¿p?
- Representar longitud de correlación (distancia media) en función de r.
- ¿Depende de L? → probar con L = 32 → ¿qué quiere decir?

Prueba C – Ejercicio 3

Prueba C – Ejercicio 4

- Distancias medias mediante Monte-Carlo.
- Hacerlo para L = 1024, N = 1024, r = 0.1, 0.25, 0.5, 1, 2, 4, 9 → ¿p?
- Elegir M=N/16 de los 0s de cada cadena (al azar). Elegir el sentido (izquierda, derecha) al azar. ¿Cómo? Calcular la distancia hasta el primer 1. Hacer la media de las distancias calculadas y comparar con el ejercicio anterior. ¿Conclusiones? ¿Es Monte-Carlo robusto? ¿Merece la pena (recursos) una búsqueda exhaustiva?

Prueba C – Ejercicio 4

Gracias!