

PRÁCTICA 4: NOTAS AUXILIARES PARA SU CORRECTO DESARROLLO E IMPLEMENTACIÓN

La práctica tiene por objetivo el poder caracterizar el grado de caos (desorden, incertidumbre) que la evolución de los autómatas celulares unidimensionales genera; para ello, utilizaremos tres medidas que permitan caracterizar ese desorden: distancia de Hamming, entropía espacial de configuración y entropía temporal celular.

DISTANCIA DE HAMMING

Definición: dadas dos configuraciones de un autómata celular C_1, C_2 , la distancia de Hamming interconfiguración $d_H(C_1, C_2)$ se define como el número de dígitos (células) en que varían. Sea consciente de que d_H cumple con las tres propiedades que definen matemáticamente una distancia (repáselas), y que por tanto el espacio de configuraciones de un autómata celular, junto con d_H , adquieren estructura de espacio métrico.

Ejemplo: dadas dos configuraciones de un autómata celular binario 0001 y 1101, su distancia de Hamming es 2.

¿Qué debe hacer vuestra práctica 4 con la distancia de Hamming? El programa efectuará la simulación del autómata celular a partir de su configuración inicial (lo que ya está implementado en la práctica 3), irá calculando la distancia de Hamming interconfiguración, y construyendo la curva de distancias, en función del número de generación.

Ejemplo: sean cuatro configuraciones C_1, C_2, C_3, C_4 de un autómata celular con 100 células, cada una sucesora de la anterior, y obtenida de ella aplicando la función de transición (en nuestro caso, codificada numéricamente). Tras calcular las distancias de Hamming interconfiguración, se encuentra que $d_H(C_1, C_2) = 67, d_H(C_2, C_3) = 31, d_H(C_3, C_4) = 13$. La curva de distancia de Hamming es:

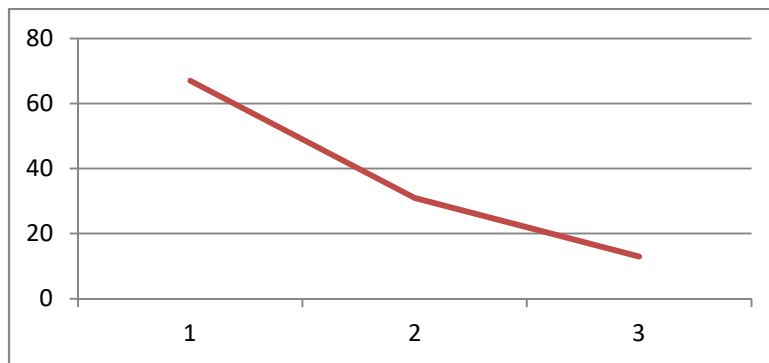


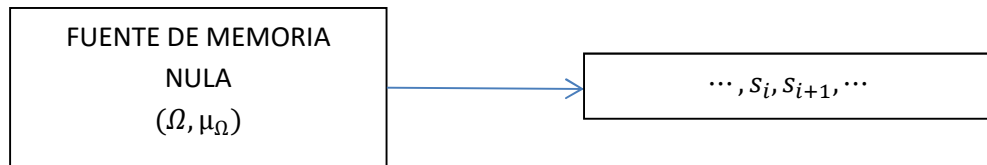
Figura 1: Curva de Distancia de Hamming

¿Cómo se interpreta esto? Autómatas celulares con curvas de distancia de Hamming altas (desde un punto de vista relativo, dado un número de células concreto) serán en principio mejores desde el punto de vista de la incertidumbre que su evolución en el tiempo genera.

ENTROPÍA

Definición (fuente de memoria nula): Una fuente de memoria nula es un par $F = (\Omega, \mu_\Omega)$ donde Ω es un alfabeto finito de símbolos $\{s_1, s_2, \dots, s_n\}$ y $\mu_\Omega: \Omega \rightarrow [0, 1]$ es una distribución de probabilidad, donde $\mu_\Omega(s_i)$ es la probabilidad del símbolo s_i .

Informalmente, una fuente de memoria nula puede imaginarse como una caja negra que emite símbolos del alfabeto al exterior, que un observador recibe. El observador desconoce qué símbolo aparecerá en el instante siguiente, y por tanto tiene un cierto grado de incertidumbre acerca de qué símbolo será emitido a continuación.



Objetivo: determinar numéricamente el grado de incertidumbre del observador acerca del siguiente símbolo emitido por la fuente F . O dicho de otra forma (que es equivalente) cuánta información le aporta al observador la aparición de un símbolo. Todo esto se mide mediante el concepto de entropía, que construye una función específicamente diseñada para medir el grado de incertidumbre del observador externo.

Definición (entropía de información): dada una fuente de memoria nula X , su entropía viene definida por la siguiente ecuación:

$$H(X) = - \sum_{i=1}^n P(x_i) \log_b P(x_i)$$

En ella, $P(x_i)$ es la probabilidad del i -ésimo símbolo del alfabeto de la fuente. La base logarítmica b se suele escoger igual a 2, a 10 ó a e , si bien la más frecuente es la base 2.

Ejemplo 1: Dada una fuente de memoria nula $F = (\{0, 1\}, \mu_\Omega)$ donde $\mu_\Omega(0) = 0.5$ y $\mu_\Omega(1) = 0.5$, la entropía de la fuente es: $H(F) = -(\mu_\Omega(0) \log_2 \mu_\Omega(0) + \mu_\Omega(1) \log_2 \mu_\Omega(1)) = 1$

Interpretación: la incertidumbre del observador es máxima (entropía igual a 1, cada símbolo emitido por la fuente le aporta un bit de información) ya que todos los símbolos son equiprobables.

Ejemplo 2: Dada una fuente de memoria nula $F = (\{0, 1\}, \mu_\Omega)$ donde $\mu_\Omega(0) = 0.7$ y $\mu_\Omega(1) = 0.3$, la entropía de la fuente es:

$$H(F) = -(\mu_\Omega(0) \log_2 \mu_\Omega(0) + \mu_\Omega(1) \log_2 \mu_\Omega(1)) = 0.81$$

Interpretación: la incertidumbre del observador es menor que antes (entropía igual a 0.81, y cada símbolo aporta menos de un bit de información) ya que un símbolo es mucho más probable que otro.

Ejemplo 3: Dada una fuente de memoria nula $F = (\{0, 1\}, \mu_\Omega)$ donde $\mu_\Omega(0) = 0.95$ y $\mu_\Omega(1) = 0.05$, la entropía de la fuente es: $H(F) = -(\mu_\Omega(0) \log_2 \mu_\Omega(0) + \mu_\Omega(1) \log_2 \mu_\Omega(1)) = 0.29$

Interpretación: la incertidumbre del observador es muy baja (entropía igual a 0.29, y cada símbolo aporta muy poca información) ya que un símbolo es mucho más probable que el otro.

Nota: hay una calculadora de entropía disponible on-line en <https://planetcalc.com/2476/>

Gráfica: variación de la entropía para un alfabeto con dos símbolos; en ordenadas, la probabilidad $P(X)$ de uno de los símbolos en orden creciente (la del otro símbolo sería $1-P(X)$):

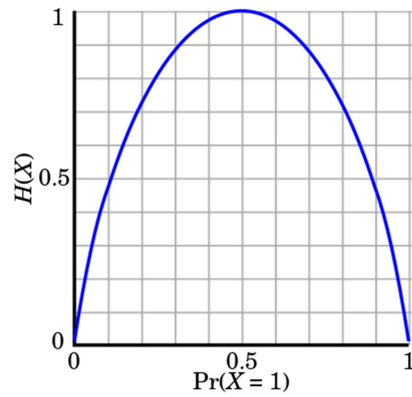


Figura 2: Curva de Entropía

Vemos como la entropía es máxima cuando ambos símbolos son equiprobables.

APLICACIÓN DE LA ENTROPÍA SOBRE AUTÓMATAS CELULARES UNIDIMENSIONALES

Veamos cómo aplicar esto para medir el grado de desorden (caos, incertidumbre) en un autómata celular unidimensional:

Entropía Espacial de Configuración: dada una configuración de un autómata celular, que no es más que un array finito de n células que toman valores en el espacio de estados, es posible extraer una distribución de probabilidad sin más que contar el número de células que hay en cada estado, y dividir por n . A partir de esta distribución, el cálculo de la entropía de la configuración es inmediato.

Ejemplo: Dada la configuración C_0 de un autómata celular binario **1101100100** vemos que $\mu_{\Omega}(0) = \mu_{\Omega}(1) = \frac{5}{10} = 0.5$, de manera que la entropía espacial de la configuración es $H(C_0) = 1$

Curva de Entropía Espacial: Sean ahora cuatro configuraciones (cada una sucesora de la anterior, y obtenida por aplicación de la función de transición) de un autómata celular unidimensional C_1, C_2, C_3, C_4 con entropías espaciales computadas según el Ejemplo anterior, e iguales a 0.98, 0.77, 0.32, 0.55. La curva de entropía es:

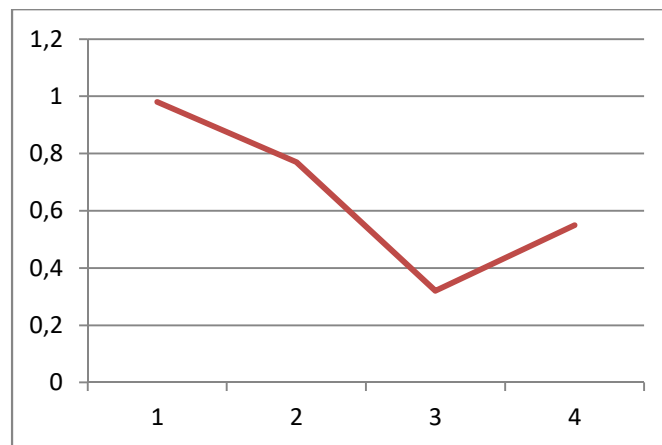


Figura 3: Curva de Entropía Espacial

Interpretación: En un autómata celular que conserve (o aumente) el grado de desorden presente en la configuración inicial (aleatoria) la curva será plana y estará cerca del límite superior que es 1. Son los autómatas que nos interesan. Un autómata donde la curva es descendente hacia valores de entropía pequeños, no nos interesará, ya que no genera incertidumbre.

¿Qué debe hacer vuestra práctica 4 con la entropía espacial? Exactamente el proceso descrito en este apartado. Ir calculando configuraciones de un autómata celular elegido por el usuario, calculando sus entropías espaciales, y construyendo la curva de entropía espacial.

Entropía Temporal Celular: sea ahora un autómata celular binario, del que computamos n configuraciones (o n generaciones temporales) dadas por C_1, C_2, \dots, C_n , y consideremos los estados adoptados por una célula concreta a_i . Resulta evidente que esos n estados forman una sucesión de bits, de la que nuevamente podemos extraer una distribución de probabilidad, y una entropía. Esa entropía $H(a_i)$ es la entropía espacial de la célula a_i .

Ejemplo: Sea la siguiente secuencia de configuraciones de un autómata celular binario:

Generación	a_0	a_1	a_2	a_3
1	0	0	1	1
2	1	0	1	0
3	1	1	1	1
4	1	1	1	0

Tabla 1: Configuraciones

Si ahora computamos la entropía temporal celular de a_1 , vemos que $H(a_1) = 1$; si hacemos lo propio para a_2 , resulta que $H(a_2) = 0$.

¿Qué debe hacer vuestra práctica 4 con la entropía temporal celular? El usuario debe poder escoger una célula concreta del array, tras lo cual el programa computará y mostrará su entropía temporal.

¿Podéis necesitar cambiar de base logarítmica en el programa? Pues puede ser. En general, cualquier calculadora/lenguaje de programación ofrecen logaritmos en base 10 y en base e. Sin embargo, quizás sean necesarias bases alternativas (por ejemplo, 2). Sobre esto, la red es “vasta e infinita” pero en todo caso, aquí va un corto tutorial que servirá:

[Cambio de base del logaritmo, con ejemplos - Didactalia: material educativo](#)

¿Cómo se dibuja una gráfica con Java? Como con todo lo relacionado con las interfaces gráficas, ni hay una forma única, ni mucho menos un estándar. En la carpeta de la práctica se ofrecen dos ejemplos de código que ilustran cómo dibujar una gráfica a partir de una nube discreta de datos (vuestro caso) y a partir de una función continua predefinida en el lenguaje ($\sin(x)$).

INPUTS Y OUTPUTS DEL GUI

Los inputs del GUI deben ser los siguientes:

- Un autómata celular a estudiar, elegido por el usuario escogiendo
 - número de estados (limitado a los fijados en la práctica anterior).
 - radio de vecinos.
 - regla codificada numéricamente.

- Estado inicial aleatorio o célula central activa.
 - Condición de frontera nula o cilíndrica.
- Análisis que se desea desarrollar (idealmente, todos) sobre el autómata celular escogido; esto es, curva de distancia de Hamming, curva de entropía espacial, o entropía temporal de una célula.
- Esto implica reutilizar la parte del código desarrollada para esto en la práctica número 3, y obviamente el núcleo de computación. Tal núcleo debería ser aumentado con las capacidades necesarias para el cálculo de las medidas de incertidumbre.
- La Figura 4 ilustra un posible formato de input, que incluye algunos campos que en esta ocasión se piden (como la condición de frontera

The screenshot shows a window titled "Simulación de autómata 1D". It contains several input fields and checkboxes:

- Número de estados:** 2
- Función de transición:** 30
- Tipo de configuración inicial:** ('Aleatoria' o 'Solo célula central activa') - Solo célula central activa
- Condición de frontera:** ('Nula' o 'Cilíndrica') - Nula
- Número de generaciones:** 400
- Semilla:** 1
- Número de células:** 500
- Buttons:** Iniciar, Reiniciar
- Checkboxes:**
 - ☒ Calcular distancia de Hamming
 - ☒ Calcular entropía espacial
 - ☒ Calcular entropía temporal
- Other fields:** Célula: 200
- Footer:** Introduzca parámetros

Figura 4: Ejemplo de Input al GUI

A partir de lo anterior, el output del GUI será:

- La curva de distancia de Hamming.
- La curva de entropía especial.
- La entropía temporal de una célula.
- La Figura 5 ilustra un posible ejemplo de output:

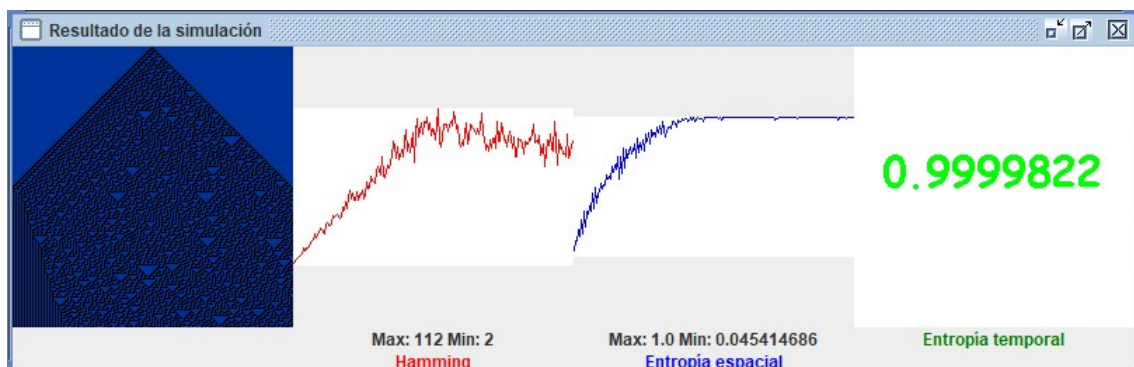


Figura 5: Ejemplo de Output del GUI