Modelación Basada en Agentes

Dr. Felipe Contreras

14 de febrero de 2018

1. Antecedentes
   - Sistemas Complejos
   - Mapeos Discretos
   - Autómatas Celulares

2. Modelos Basados en Agentes

Sistemas Complejos

Características

- Sistema: Conjunto de elementos o partes conectadas entre sí, que llevan acabo cierta función
- Complejo $\neq$ Complicado
- Presentan auto-organización
- Exhiben propiedades emergentes ("el todo es más que la suma de sus partes")
  1. Muchos elementos
  2. Las interacciones son dinámicas
  3. Elementos influyen y son influidos por los demás
  4. Las interacciones son no lineales (pequeñas "causas", pueden tener "efectos" grandes)
  5. Las interacciones son recursivas
  6. Son abiertos
  7. Operan lejos del equilibrio
  8. Tienen una historia
  9. Los elementos actúan con información local

Características

- Sistema: Conjunto de elementos o partes conectadas entre sí, que llevan acabo cierta función
- Complejo $\neq$ Complicado
- Presentan auto-organización
- Exhiben propiedades emergentes ("el todo es más que la suma de sus partes")
  1. Muchos elementos
  2. Las interacciones son dinámicas
  3. Elementos influyen y son influidos por los demás
  4. Las interacciones son no lineales (pequeñas "causas", pueden tener "efectos" grandes)
  5. Las interacciones son recursivas
  6. Son abiertos
  7. Operan lejos del equilibrio
  8. Tienen una historia
  9. Los elementos actúan con información local

Características

- Sistema: Conjunto de elementos o partes conectadas entre sí, que llevan acabo cierta función
- Complejo $\neq$ Complicado
- Presentan auto-organización
- Exhiben propiedades emergentes ("el todo es más que la suma de sus partes")
  1. Muchos elementos
  2. Las interacciones son dinámicas
  3. Elementos influyen y son influidos por los demás
  4. Las interacciones son no lineales (pequeñas "causas", pueden tener "efectos" grandes)
  5. Las interacciones son recursivas
  6. Son abiertos
  7. Operan lejos del equilibrio
  8. Tienen una historia
  9. Los elementos actúan con información local

Características

- Sistema: Conjunto de elementos o partes conectadas entre sí, que llevan acabo cierta función
- Complejo $\neq$ Complicado
- Presentan auto-organización
- Exhiben propiedades emergentes ("el todo es más que la suma de sus partes")
  1. Muchos elementos
  2. Las interacciones son dinámicas
  3. Elementos influyen y son influidos por los demás
  4. Las interacciones son no lineales (pequeñas "causas", pueden tener "efectos" grandes)
  5. Las interacciones son recursivas
  6. Son abiertos
  7. Operan lejos del equilibrio
  8. Tienen una historia
  9. Los elementos actúan con información local

Características

- Sistema: Conjunto de elementos o partes conectadas entre sí, que llevan acabo cierta función
- Complejo $\neq$ Complicado
- Presentan auto-organización
- Exhiben propiedades emergentes ("el todo es más que la suma de sus partes")
  1. Muchos elementos
  2. Las interacciones son dinámicas
  3. Elementos influyen y son influidos por los demás
  4. Las interacciones son no lineales (pequeñas "causas", pueden tener "efectos" grandes)
  5. Las interacciones son recursivas
  6. Son abiertos
  7. Operan lejos del equilibrio
  8. Tienen una historia
  9. Los elementos actúan con información local

Características

- Sistema: Conjunto de elementos o partes conectadas entre sí, que llevan acabo cierta función
- Complejo $\neq$ Complicado
- Presentan auto-organización
- Exhiben propiedades emergentes ("el todo es más que la suma de sus partes")
  1. Muchos elementos
  2. Las interacciones son dinámicas
  3. Elementos influyen y son influidos por los demás
  4. Las interacciones son no lineales (pequeñas "causas", pueden tener "efectos" grandes)
  5. Las interacciones son recursivas
  6. Son abiertos
  7. Operan lejos del equilibrio
  8. Tienen una historia
  9. Los elementos actúan con información local

Características

- Sistema: Conjunto de elementos o partes conectadas entre sí, que llevan acabo cierta función
- Complejo $\neq$ Complicado
- Presentan auto-organización
- Exhiben propiedades emergentes ("el todo es más que la suma de sus partes")
  1. Muchos elementos
  2. Las interacciones son dinámicas
  3. Elementos influyen y son influidos por los demás
  4. Las interacciones son no lineales (pequeñas "causas", pueden tener "efectos" grandes)
  5. Las interacciones son recursivas
  6. Son abiertos
  7. Operan lejos del equilibrio
  8. Tienen una historia
  9. Los elementos actúan con información local

Características

- Sistema: Conjunto de elementos o partes conectadas entre sí, que llevan acabo cierta función
- Complejo $\neq$ Complicado
- Presentan auto-organización
- Exhiben propiedades emergentes ("el todo es más que la suma de sus partes")
  1. Muchos elementos
  2. Las interacciones son dinámicas
  3. Elementos influyen y son influidos por los demás
  4. Las interacciones son no lineales (pequeñas "causas", pueden tener "efectos" grandes)
  5. Las interacciones son recursivas
  6. Son abiertos
  7. Operan lejos del equilibrio
  8. Tienen una historia
  9. Los elementos actúan con información local

Características

- Sistema: Conjunto de elementos o partes conectadas entre sí, que llevan acabo cierta función
- Complejo $\neq$ Complicado
- Presentan auto-organización
- Exhiben propiedades emergentes ("el todo es más que la suma de sus partes")
  1. Muchos elementos
  2. Las interacciones son dinámicas
  3. Elementos influyen y son influidos por los demás
  4. Las interacciones son no lineales (pequeñas "causas", pueden tener "efectos" grandes)
  5. Las interacciones son recursivas
  6. Son abiertos
  7. Operan lejos del equilibrio
  8. Tienen una historia
  9. Los elementos actúan con información local

Características

- Sistema: Conjunto de elementos o partes conectadas entre sí, que llevan acabo cierta función
- Complejo $\neq$ Complicado
- Presentan auto-organización
- Exhiben propiedades emergentes ("el todo es más que la suma de sus partes")
  1. Muchos elementos
  2. Las interacciones son dinámicas
  3. Elementos influyen y son influidos por los demás
  4. Las interacciones son no lineales (pequeñas "causas", pueden tener "efectos" grandes)
  5. Las interacciones son recursivas
  6. Son abiertos
  7. Operan lejos del equilibrio
  8. Tienen una historia
  9. Los elementos actúan con información local

Características

- Sistema: Conjunto de elementos o partes conectadas entre sí, que llevan acabo cierta función
- Complejo $\neq$ Complicado
- Presentan auto-organización
- Exhiben propiedades emergentes ("el todo es más que la suma de sus partes")
  1. Muchos elementos
  2. Las interacciones son dinámicas
  3. Elementos influyen y son influidos por los demás
  4. Las interacciones son no lineales (pequeñas "causas", pueden tener "efectos" grandes)
  5. Las interacciones son recursivas
  6. Son abiertos
  7. Operan lejos del equilibrio
  8. Tienen una historia
  9. Los elementos actúan con información local

Características

- Sistema: Conjunto de elementos o partes conectadas entre sí, que llevan acabo cierta función
- Complejo $\neq$ Complicado
- Presentan auto-organización
- Exhiben propiedades emergentes ("el todo es más que la suma de sus partes")
  1. Muchos elementos
  2. Las interacciones son dinámicas
  3. Elementos influyen y son influidos por los demás
  4. Las interacciones son no lineales (pequeñas "causas", pueden tener "efectos" grandes)
  5. Las interacciones son recursivas
  6. Son abiertos
  7. Operan lejos del equilibrio
  8. Tienen una historia
  9. Los elementos actúan con información local

Características

- Sistema: Conjunto de elementos o partes conectadas entre sí, que llevan acabo cierta función
- Complejo $\neq$ Complicado
- Presentan auto-organización
- Exhiben propiedades emergentes ("el todo es más que la suma de sus partes")
  1. Muchos elementos
  2. Las interacciones son dinámicas
  3. Elementos influyen y son influidos por los demás
  4. Las interacciones son no lineales (pequeñas "causas", pueden tener "efectos" grandes)
  5. Las interacciones son recursivas
  6. Son abiertos
  7. Operan lejos del equilibrio
  8. Tienen una historia
  9. Los elementos actúan con información local

Mapeos Discretos

Mapeos discretos

- $y = f(x)$
- $x_1 = f(x_0)$
- $x_2 = f(x_1), x_3 = f(x_2), x_4 = f(x_3), ...$

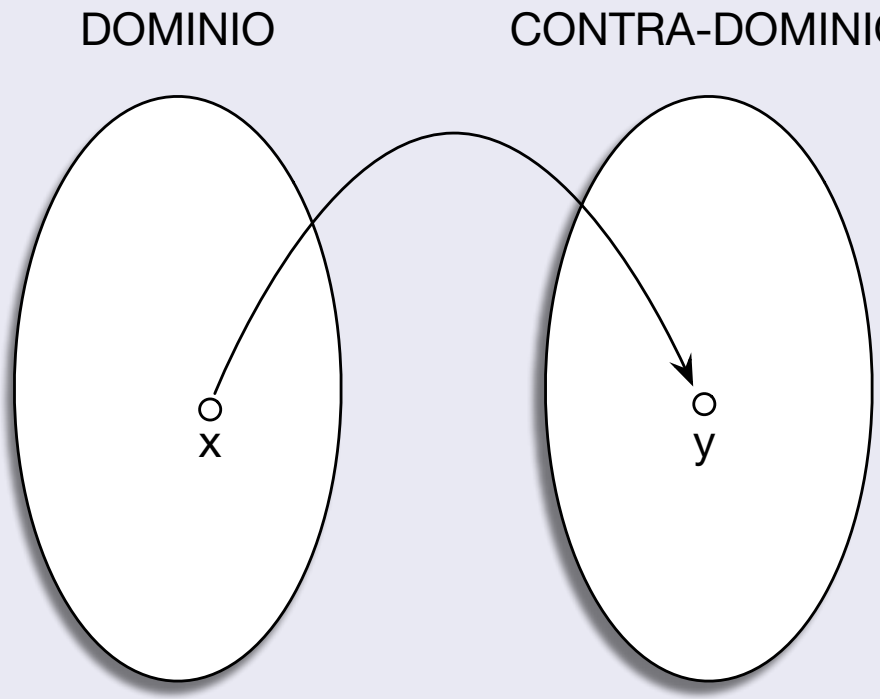
Mapeos discretos

- $y = f(x)$
- $x_1 = f(x_0)$
- $x_2 = f(x_1), x_3 = f(x_2), x_4 = f(x_3), ...$

Mapeos discretos

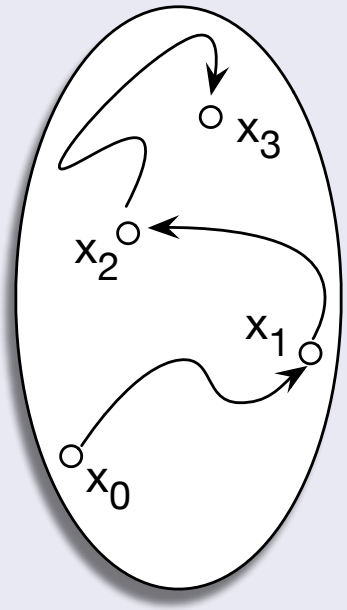
- $y = f(x)$
- $x_1 = f(x_0)$
- $x_2 = f(x_1), x_3 = f(x_2), x_4 = f(x_3), ...$

Representación gráfica



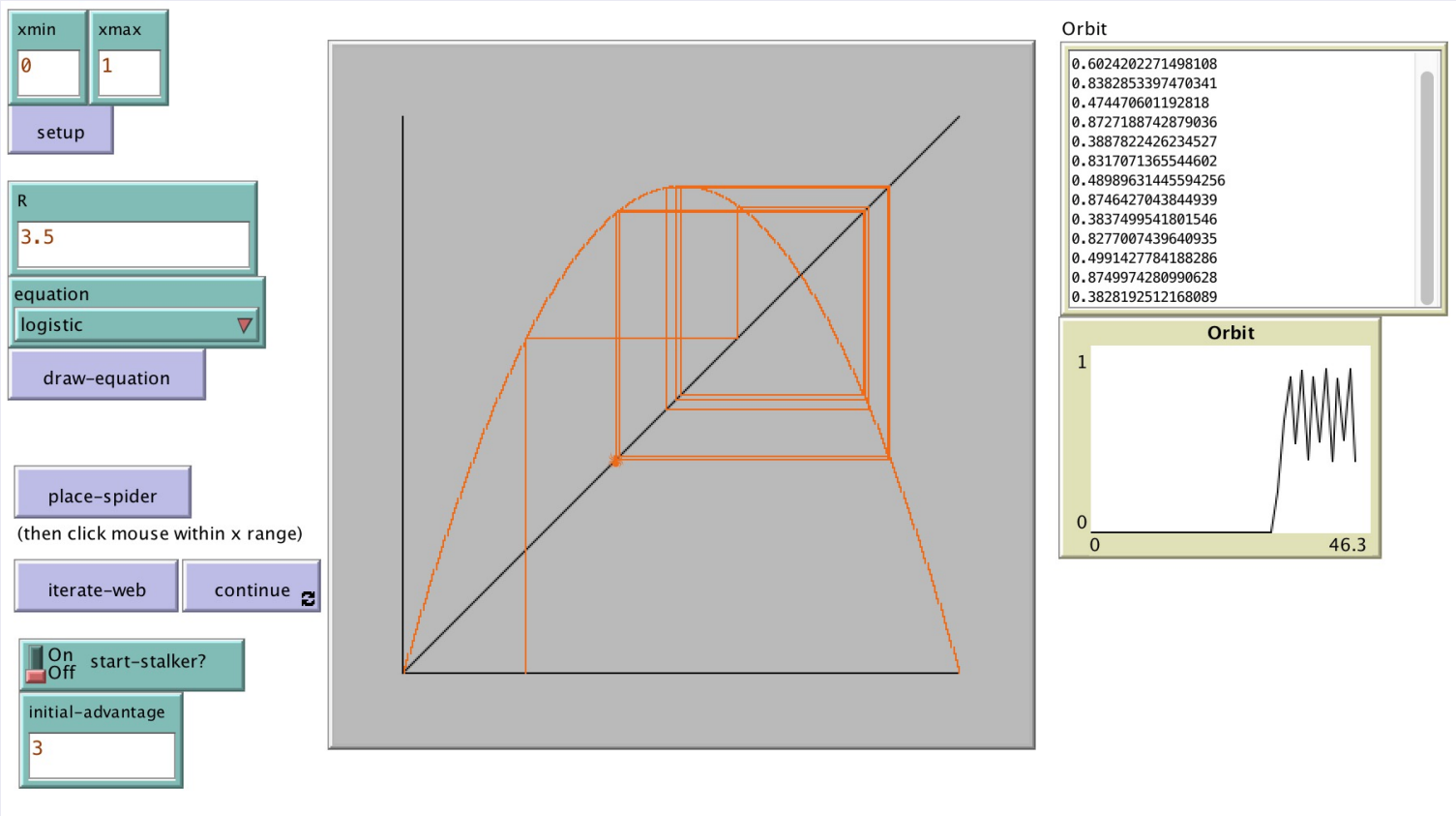
Representación gráfica

DOMINIO Y CONTRA-DOMINIO



Representación gráfica

Programa "logistica"



Características

- Converge (a un punto)
- No converge: tiene ciclo límite
- No converge: órbita densa

Características

- Converge (a un punto)
- No converge: tiene ciclo límite
- No converge: órbita densa

Características

- Converge (a un punto)
- No converge: tiene ciclo límite
- No converge: órbita densa

Autómatas Celulares

Definición (genérica)

- Un AC consiste de autómatas (llamados también celdas o sitios) idénticos, dispuestos uniformemente en los puntos de una látice D-dimensional de un espacio discreto. Normalmente D=1, 2, o 3
- Cada autómata es una variable dinámica y su cambio temporal esta dado por la expresión:

$$s_{t+1}(x) = F(s_t(x + x_0), s_t(x + x_1), \ldots, s_t(x + x_{n-1}))$$

- $s_t(x)$ es el estado de un autómata localizado en x en el tiempo t
- F es la función de transición de estado
- y $N = \{x_0, x_1, \ldots, x_{n-1}\}$ es la *vecindad*
- por lo general se aplica la misma F y la misma vecindad uniformemente a todas las posiciones espaciales

Definición (genérica)

- Un AC consiste de autómatas (llamados también celdas o sitios) idénticos, dispuestos uniformemente en los puntos de una látice D-dimensional de un espacio discreto. Normalmente D=1, 2, o 3
- Cada autómata es una variable dinámica y su cambio temporal esta dado por la expresión:

$$s_{t+1}(x) = F(s_t(x + x_0), s_t(x + x_1), \ldots, s_t(x + x_{n-1}))$$

- $s_t(x)$ es el estado de un autómata localizado en x en el tiempo t
- F es la función de transición de estado
- y $N = \{x_0, x_1, \ldots, x_{n-1}\}$ es la *vecindad*
- por lo general se aplica la misma F y la misma vecindad uniformemente a todas las posiciones espaciales

Definición (genérica)

- Un AC consiste de autómatas (llamados también celdas o sitios) idénticos, dispuestos uniformemente en los puntos de una látice D-dimensional de un espacio discreto. Normalmente D=1, 2, o 3
- Cada autómata es una variable dinámica y su cambio temporal esta dado por la expresión:

$$s_{t+1}(x) = F(s_t(x + x_0), s_t(x + x_1), \ldots, s_t(x + x_{n-1}))$$

- $s_t(x)$ es el estado de un autómata localizado en x en el tiempo t
- F es la función de transición de estado
- y $N = \{x_0, x_1, \ldots, x_{n-1}\}$ es la *vecindad*
- por lo general se aplica la misma F y la misma vecindad uniformemente a todas las posiciones espaciales

Definición (genérica)

- Un AC consiste de autómatas (llamados también celdas o sitios) idénticos, dispuestos uniformemente en los puntos de una látice D-dimensional de un espacio discreto. Normalmente D=1, 2, o 3
- Cada autómata es una variable dinámica y su cambio temporal esta dado por la expresión:

$$s_{t+1}(x) = F(s_t(x + x_0), s_t(x + x_1), \ldots, s_t(x + x_{n-1}))$$

- $s_t(x)$ es el estado de un autómata localizado en x en el tiempo t
- F es la función de transición de estado
- y $N = \{x_0, x_1, \ldots, x_{n-1}\}$ es la *vecindad*
- por lo general se aplica la misma F y la misma vecindad uniformemente a todas las posiciones espaciales

Definición (genérica)

- Un AC consiste de autómatas (llamados también celdas o sitios) idénticos, dispuestos uniformemente en los puntos de una látice D-dimensional de un espacio discreto. Normalmente D=1, 2, o 3
- Cada autómata es una variable dinámica y su cambio temporal esta dado por la expresión:

$$s_{t+1}(x) = F(s_t(x + x_0), s_t(x + x_1), \ldots, s_t(x + x_{n-1}))$$

- $s_t(x)$ es el estado de un autómata localizado en x en el tiempo t
- F es la función de transición de estado
- y $N = \{x_0, x_1, \ldots, x_{n-1}\}$ es la *vecindad*
- por lo general se aplica la misma F y la misma vecindad uniformemente a todas las posiciones espaciales

Definición (genérica)

- Un AC consiste de autómatas (llamados también celdas o sitios) idénticos, dispuestos uniformemente en los puntos de una látice D-dimensional de un espacio discreto. Normalmente D=1, 2, o 3
- Cada autómata es una variable dinámica y su cambio temporal esta dado por la expresión:

$$s_{t+1}(x) = F(s_t(x + x_0), s_t(x + x_1), \ldots, s_t(x + x_{n-1}))$$

- $s_t(x)$ es el estado de un autómata localizado en x en el tiempo t
- F es la función de transición de estado
- y $N = \{x_0, x_1, \ldots, x_{n-1}\}$ es la *vecindad*
- por lo general se aplica la misma F y la misma vecindad uniformemente a todas las posiciones espaciales

Definición (genérica)

- Fueron inicialmente desarrolladas por John von Newmann y su colaborador Stanislaw Ulam
- Constituyen una forma de describir dinámicas espacio-temporales altamente no lineales de una manera simple y concisa
- Se utilizan para diversos campos como la dinámica molecular, hidrodinámica, propiedades físicas de materiales, procesos químicos de reacción-difusión, crecimiento y morfogénesis de organismos vivos, etc. [Sayama p.185 y ss]

Definición (genérica)

- Fueron inicialmente desarrolladas por John von Newmann y su colaborador Stanislaw Ulam
- Constituyen una forma de describir dinámicas espacio-temporales altamente no lineales de una manera simple y concisa
- Se utilizan para diversos campos como la dinámica molecular, hidrodinámica, propiedades físicas de materiales, procesos químicos de reacción-difusión, crecimiento y morfogénesis de organismos vivos, etc. [Sayama p.185 y ss]

Definición (genérica)

- Fueron inicialmente desarrolladas por John von Newmann y su colaborador Stanislaw Ulam
- Constituyen una forma de describir dinámicas espacio-temporales altamente no lineales de una manera simple y concisa
- Se utilizan para diversos campos como la dinámica molecular, hidrodinámica, propiedades físicas de materiales, procesos químicos de reacción-difusión, crecimiento y morfogénesis de organismos vivos, etc. [Sayama p.185 y ss]

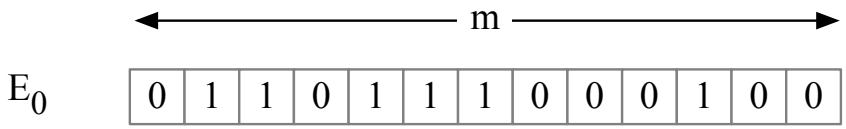
AC 1D: Definición práctica

- Vocabulario σ de n símbolos
- Organización de m de estos símbolos en un estado inicial E_0
- Tamaño de vecindad o radio ρ
- Condiciones en la frontera (cíclica, terminación, valor único)
- Regla de evolución (función de mapeo)

$$\sigma = \{0, 1\},\ n = 2$$

AC 1D: Definición práctica

- Vocabulario σ de n símbolos
- Organización de m de estos símbolos en un estado inicial E_0
- Tamaño de vecindad o radio ρ
- Condiciones en la frontera (cíclica, terminación, valor único)
- Regla de evolución (función de mapeo)



AC 1D: Definición práctica

- Vocabulario σ de n símbolos
- Organización de m de estos símbolos en un estado inicial E_0
- Tamaño de vecindad o radio ρ
- Condiciones en la frontera (cíclica, terminación, valor único)
- Regla de evolución (función de mapeo)

$$\rho = 3$$

AC 1D: Definición práctica

- Vocabulario σ de n símbolos
- Organización de m de estos símbolos en un estado inicial E_0
- Tamaño de vecindad o radio ρ
- Condiciones en la frontera (cíclica, terminación, valor único)
- Regla de evolución (función de mapeo)

E_0

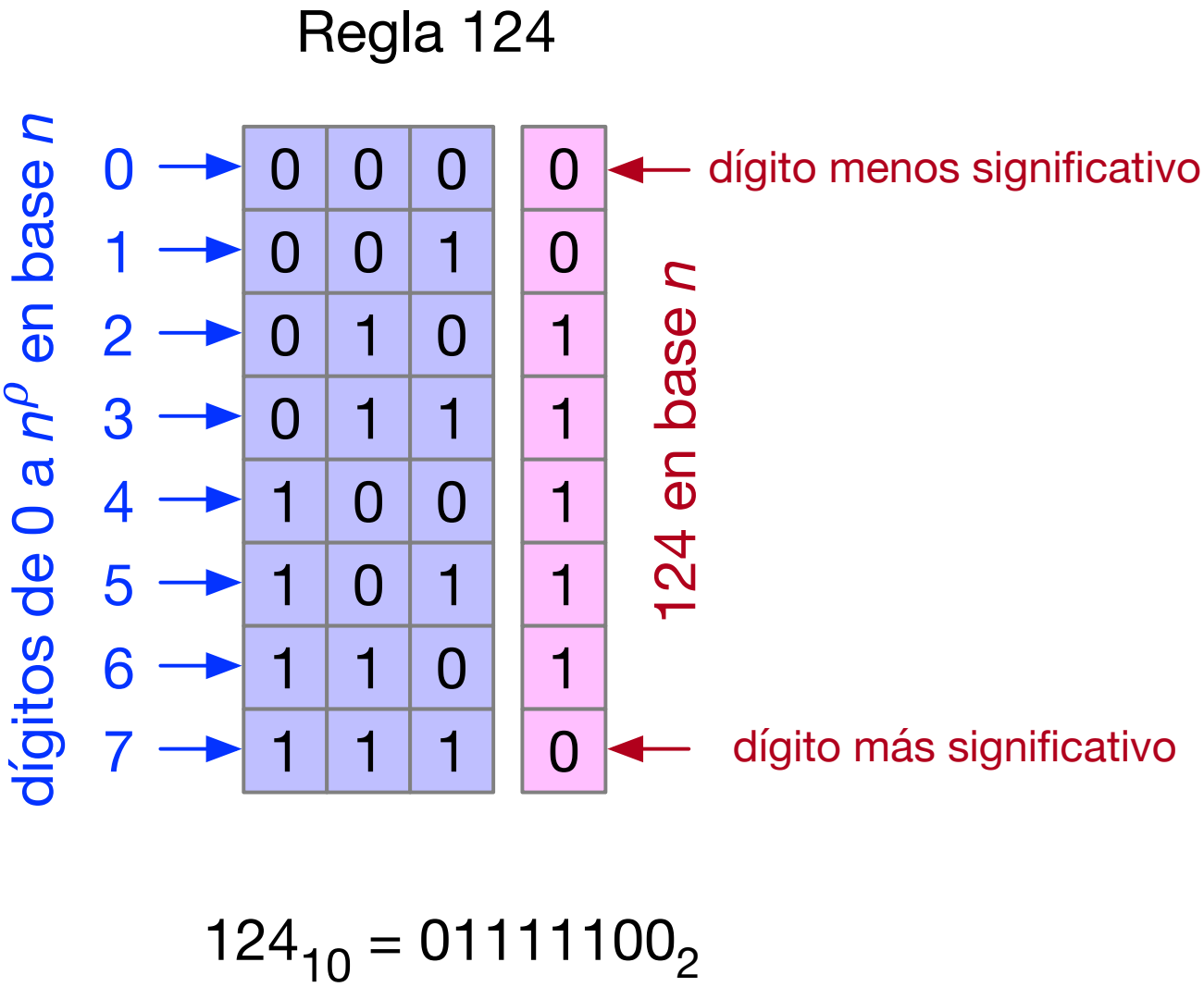
AC 1D: Definición práctica

- Vocabulario σ de n símbolos
- Organización de m de estos símbolos en un estado inicial E_0
- Tamaño de vecindad o radio ρ
- Condiciones en la frontera (cíclica, terminación, valor único)
- Regla de evolución (función de mapeo)

Regla 124

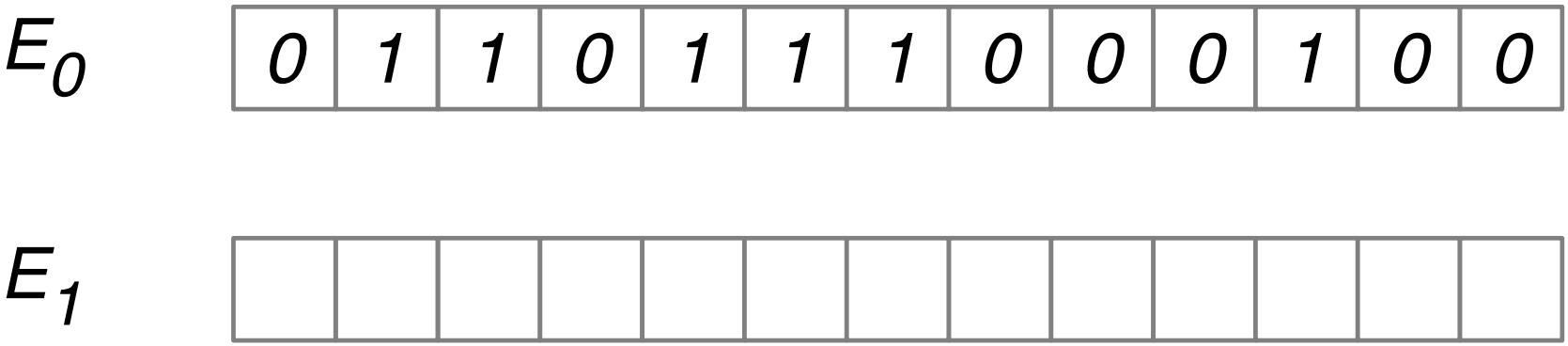
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Regla de evolución

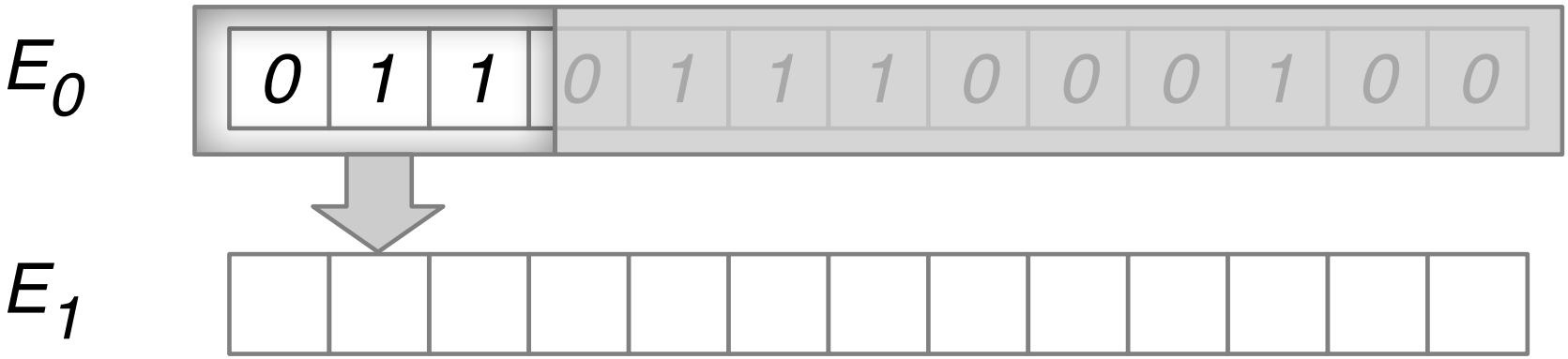


$$124_{10} = 01111100_2$$

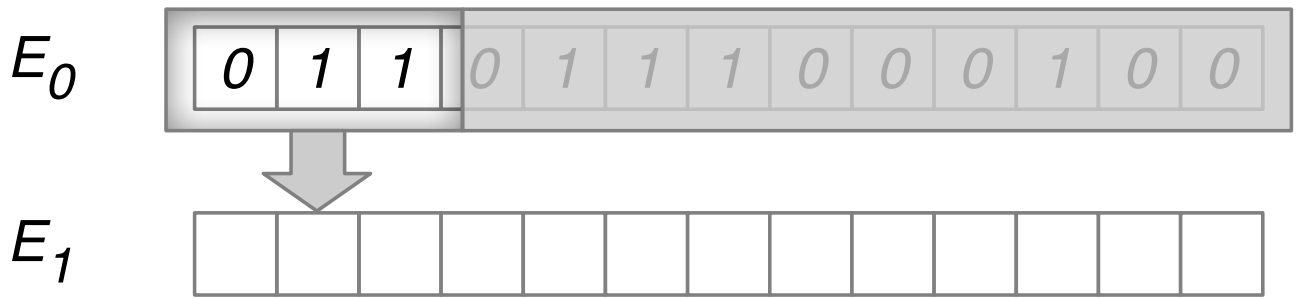
Aplicación de la regla



Aplicación de la regla



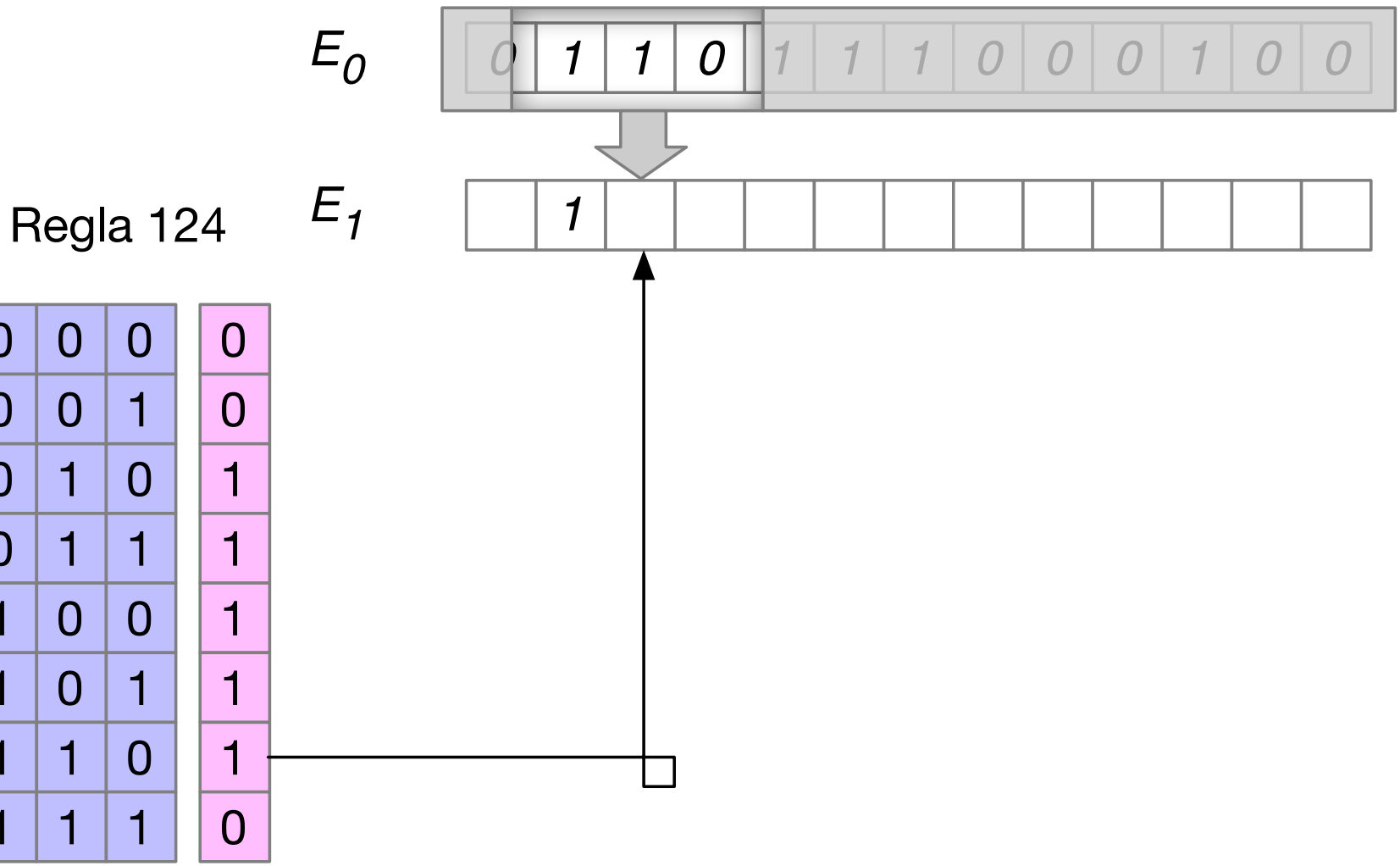
Aplicación de la regla



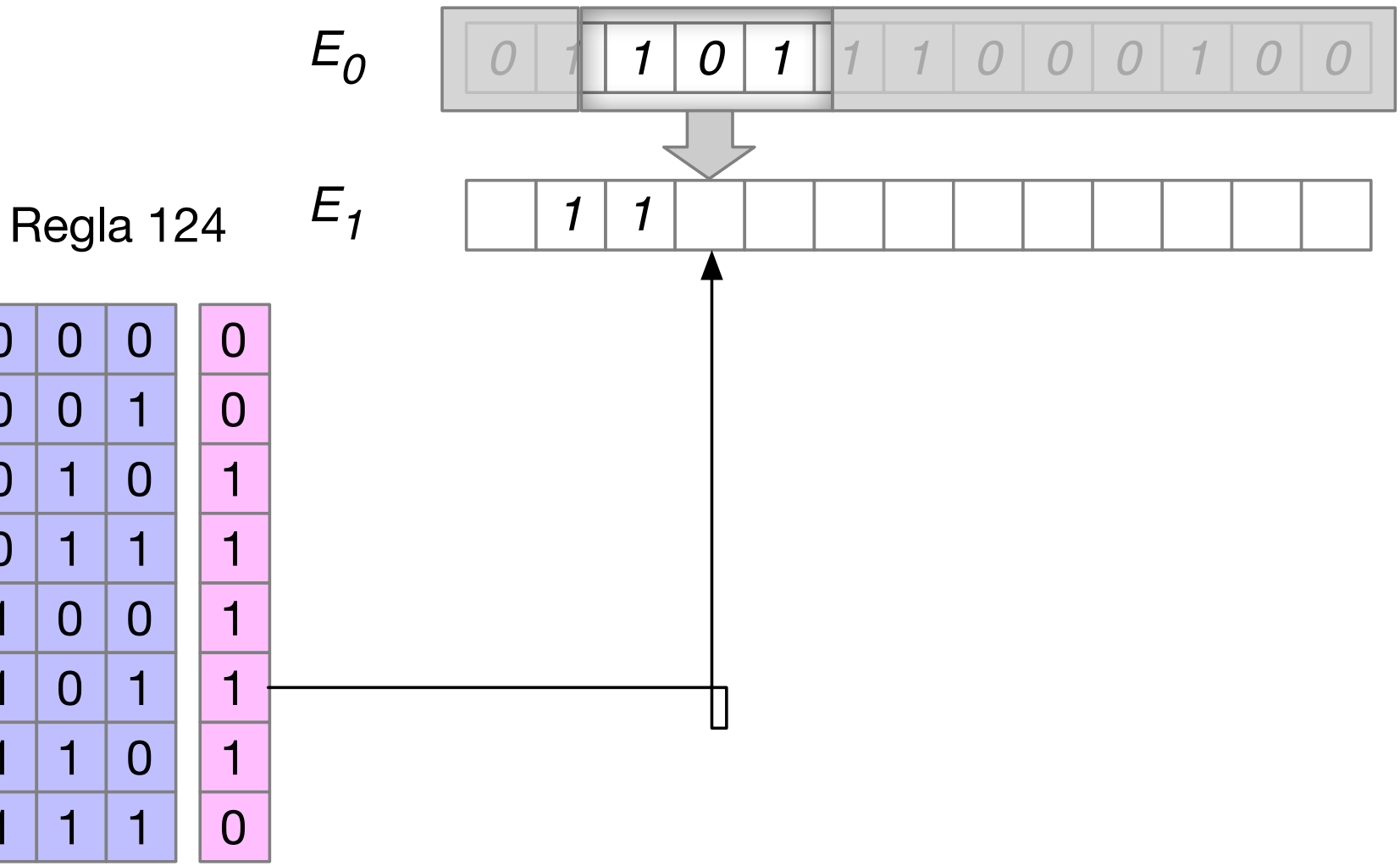
Regla 124

0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

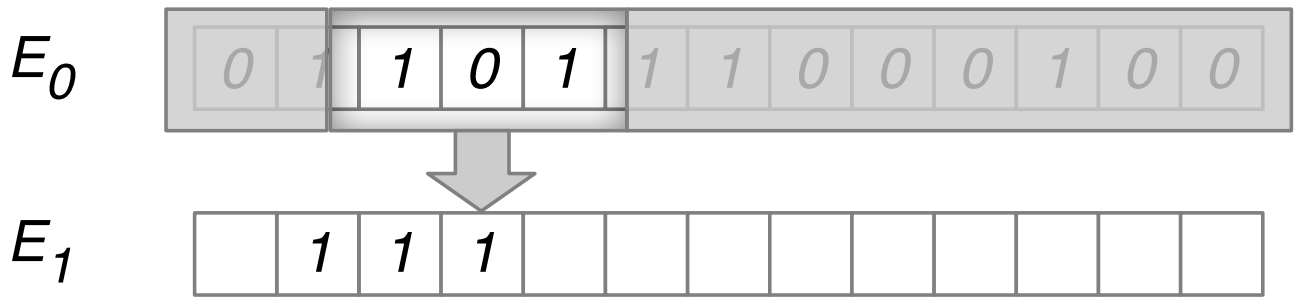
Aplicación de la regla



Aplicación de la regla

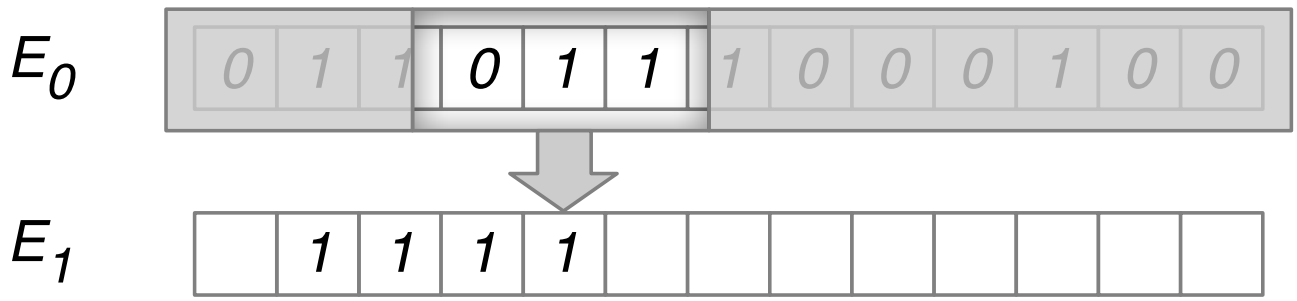


Aplicación de la regla



0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

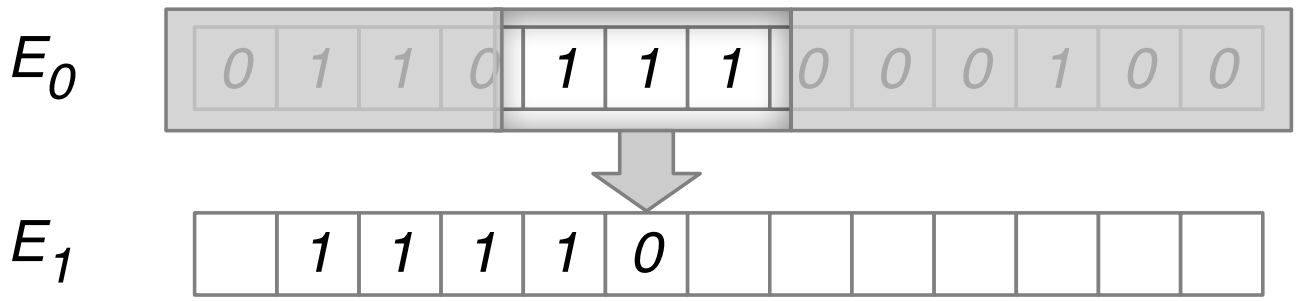
Aplicación de la regla



Regla 124

0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

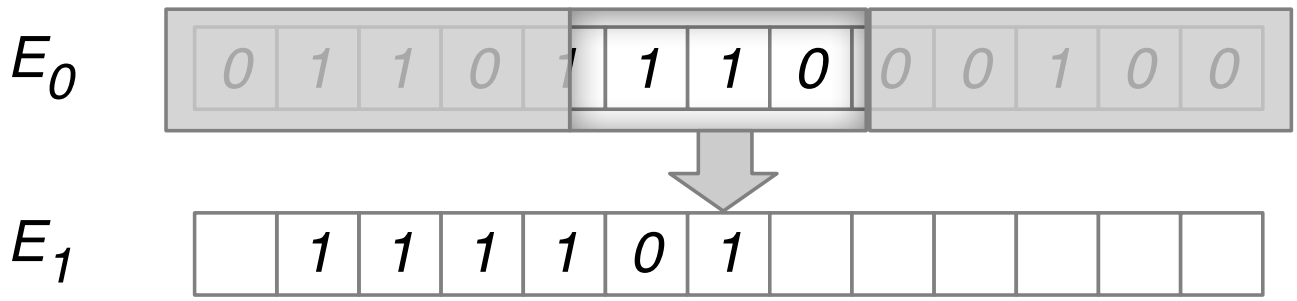
Aplicación de la regla



Regla 124

0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Aplicación de la regla



Regla 124

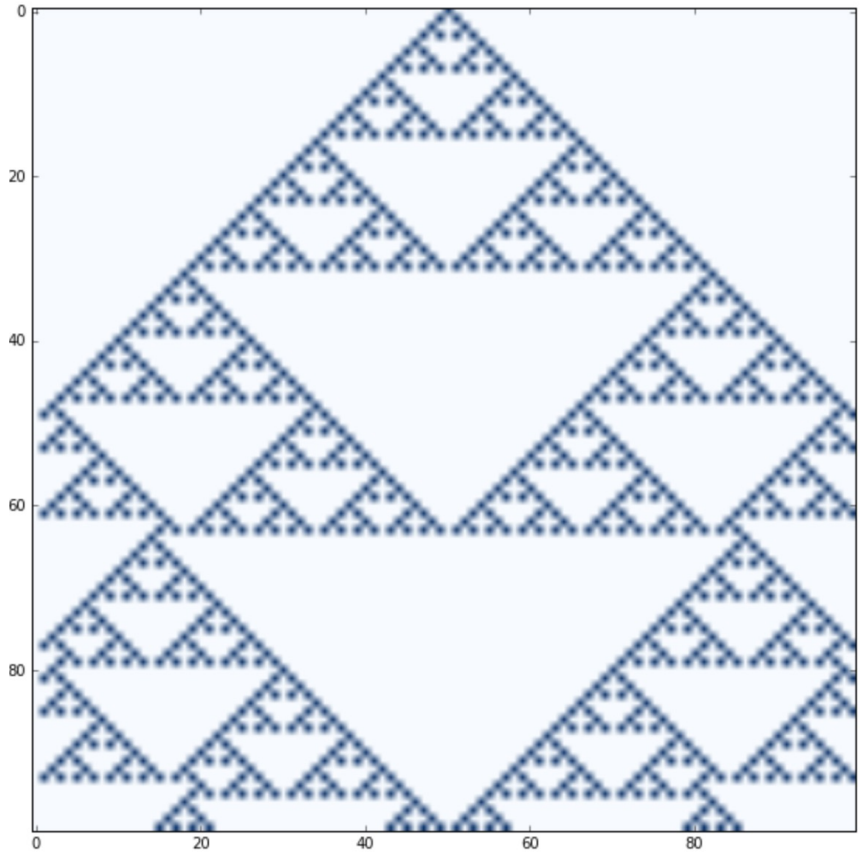
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Aplicación de la regla

¿cómo quedan las demás?, ¿cómo funciona la condición de frontera cíclica?

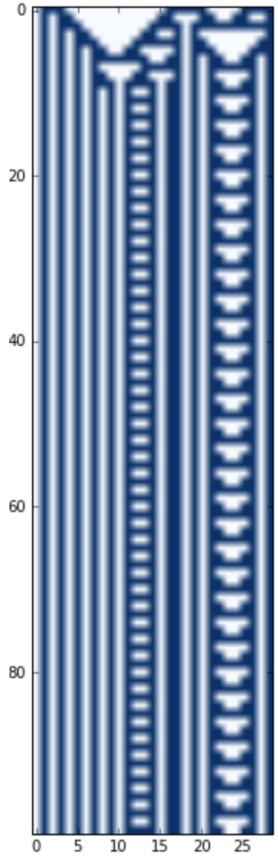
Ejemplos (Programa "Autómatas Celulares")

Regla 90, E_0="central"



Ejemplos (Programa "Autómatas Celulares")

Regla 94, E_0="01110000000000001111100001111"



Ejemplos (Programa "Autómatas Celulares")

Regla 135, E_0="azar"

Clasificación de Wolfram

- Uniforme
- Cíclico
- Aleatorio
- Complejo

Clasificación de Wolfram

- Uniforme
- Cíclico
- Aleatorio
- Complejo

Clasificación de Wolfram

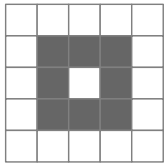
- Uniforme
- Cíclico
- Aleatorio
- Complejo

Clasificación de Wolfram

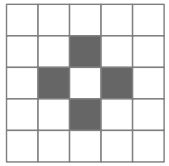
- Uniforme
- Cíclico
- Aleatorio
- Complejo

Autómatas en 2D: Regla de la mayoría

- La vida o muerte de la celda central está dictada por el valor de la mayoría de las celdas de su vecindad de Moore



Vecindad de Moore

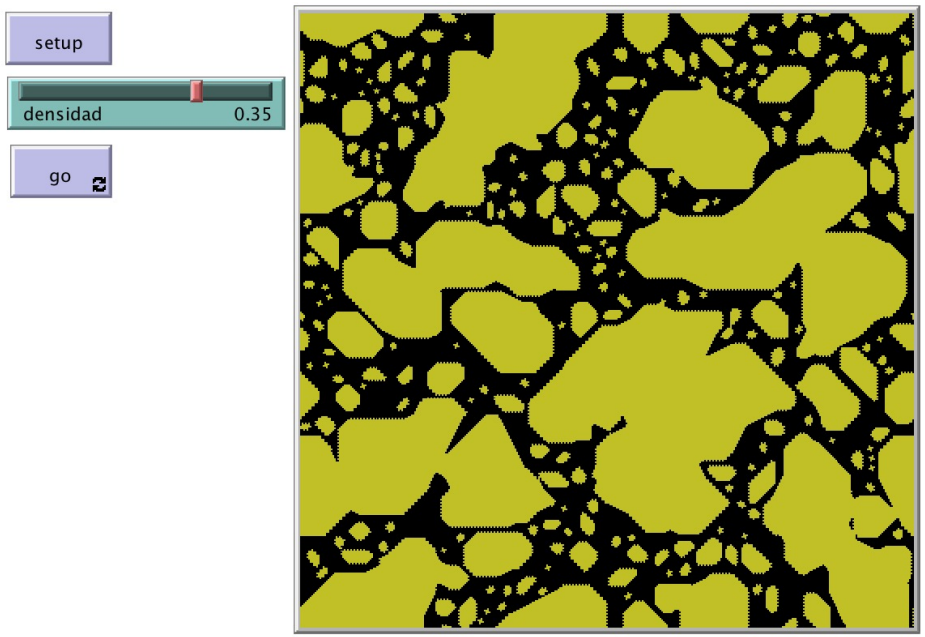


Vecindad de von Neumann

Autómatas en 2D: Regla de la mayoría

- La vida o muerte de la celda central está dictada por el valor de la mayoría de las celdas de su vecindad de Moore

Programa "Manchas"



Autómatas en 2D: El juego de la vida

- El conjunto de símbolos es $\sigma = \{0, 1\}$, significando 0=“muerta”, 1=“viva”
- E_0, y todos los demás estados, están dispuestos en una parrilla 2D de celdas
- La vecindad mide $\rho = (3, 3)$, es un cuadro de 3×3 símbolos
- La regla de evolución para el siguiente estado, asigna a la celda central el valor:
  - “viva”, si la celda central esta “muerta” y hay exactamente 3 vecinos vivos
  - “muerta”, si la celda central está “viva” y más de 3 (sobrepoblación) o menos de 2 (soledad) vecinos están vivos
  - En cualquier otro caso, la celda mantiene su símbolo

Autómatas en 2D: El juego de la vida

- El conjunto de símbolos es $\sigma = \{0, 1\}$, significando 0=“muerta”, 1=“viva”
- E_0, y todos los demás estados, están dispuestos en una parrilla 2D de celdas
- La vecindad mide $\rho = (3, 3)$, es un cuadro de 3×3 símbolos
- La regla de evolución para el siguiente estado, asigna a la celda central el valor:
  - “viva”, si la celda central esta “muerta” y hay exactamente 3 vecinos vivos
  - “muerta”, si la celda central está “viva” y más de 3 (sobrepoblación) o menos de 2 (soledad) vecinos están vivos
  - En cualquier otro caso, la celda mantiene su símbolo

Autómatas en 2D: El juego de la vida

- El conjunto de símbolos es $\sigma = \{0, 1\}$, significando 0=“muerta”, 1=“viva”
- E_0, y todos los demás estados, están dispuestos en una parrilla 2D de celdas
- La vecindad mide $\rho = (3, 3)$, es un cuadro de 3×3 símbolos
- La regla de evolución para el siguiente estado, asigna a la celda central el valor:
  - “viva”, si la celda central esta “muerta” y hay exactamente 3 vecinos vivos
  - “muerta”, si la celda central está “viva” y más de 3 (sobrepoblación) o menos de 2 (soledad) vecinos están vivos
  - En cualquier otro caso, la celda mantiene su símbolo

Autómatas en 2D: El juego de la vida

- El conjunto de símbolos es $\sigma = \{0, 1\}$, significando 0="muerta", 1="viva"
- E_0, y todos los demás estados, están dispuestos en una parrilla 2D de celdas
- La vecindad mide $\rho = (3, 3)$, es un cuadro de 3×3 símbolos
- La regla de evolución para el siguiente estado, asigna a la celda central el valor:
  - "viva", si la celda central esta "muerta" y hay exactamente 3 vecinos vivos
  - "muerta", si la celda central está "viva" y más de 3 (sobrepoblación) o menos de 2 (soledad) vecinos están vivos
  - En cualquier otro caso, la celda mantiene su símbolo

Autómatas en 2D: El juego de la vida

- El conjunto de símbolos es $\sigma = \{0, 1\}$, significando 0=“muerta”, 1=“viva”
- E_0, y todos los demás estados, están dispuestos en una parrilla 2D de celdas
- La vecindad mide $\rho = (3, 3)$, es un cuadro de 3×3 símbolos
- La regla de evolución para el siguiente estado, asigna a la celda central el valor:
  - “viva”, si la celda central esta “muerta” y hay exactamente 3 vecinos vivos
  - “muerta”, si la celda central está “viva” y más de 3 (sobrepoblación) o menos de 2 (soledad) vecinos están vivos
  - En cualquier otro caso, la celda mantiene su símbolo

Autómatas en 2D: El juego de la vida

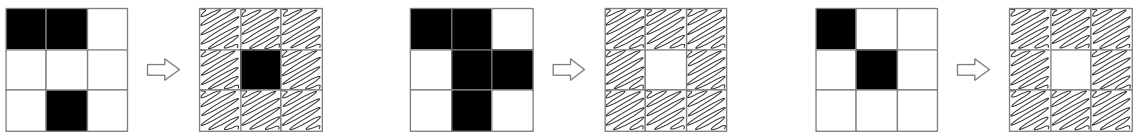
- El conjunto de símbolos es $\sigma = \{0, 1\}$, significando 0="muerta", 1="viva"
- E_0, y todos los demás estados, están dispuestos en una parrilla 2D de celdas
- La vecindad mide $\rho = (3, 3)$, es un cuadro de 3×3 símbolos
- La regla de evolución para el siguiente estado, asigna a la celda central el valor:
  - "viva", si la celda central esta "muerta" y hay exactamente 3 vecinos vivos
  - "muerta", si la celda central está "viva" y más de 3 (sobrepoblación) o menos de 2 (soledad) vecinos están vivos
  - En cualquier otro caso, la celda mantiene su símbolo

Autómatas en 2D: El juego de la vida

- El conjunto de símbolos es $\sigma = \{0, 1\}$, significando 0="muerta", 1="viva"
- E_0, y todos los demás estados, están dispuestos en una parrilla 2D de celdas
- La vecindad mide $\rho = (3, 3)$, es un cuadro de 3×3 símbolos
- La regla de evolución para el siguiente estado, asigna a la celda central el valor:
  - "viva", si la celda central esta "muerta" y hay exactamente 3 vecinos vivos
  - "muerta", si la celda central está "viva" y más de 3 (sobrepoblación) o menos de 2 (soledad) vecinos están vivos
  - En cualquier otro caso, la celda mantiene su símbolo

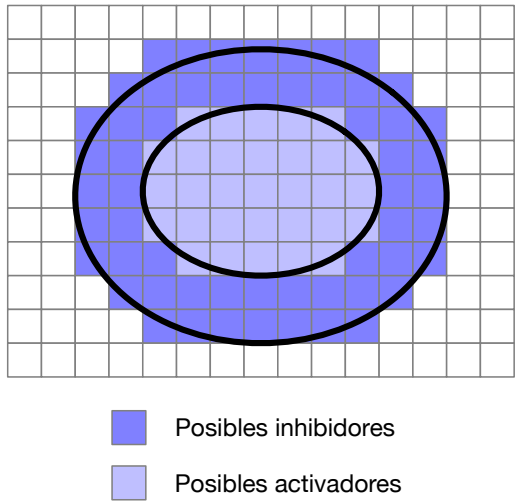
Autómatas en 2D: El juego de la vida

- El conjunto de símbolos es $\sigma = \{0, 1\}$, significando 0=“muerta”, 1=“viva”
- E_0, y todos los demás estados, están dispuestos en una parrilla 2D de celdas
- La vecindad mide $\rho = (3, 3)$, es un cuadro de 3×3 símbolos
- La regla de evolución para el siguiente estado, asigna a la celda central el valor:
  - “viva”, si la celda central esta “muerta” y hay exactamente 3 vecinos vivos
  - “muerta”, si la celda central está “viva” y más de 3 (sobrepoblación) o menos de 2 (soledad) vecinos están vivos
  - En cualquier otro caso, la celda mantiene su símbolo



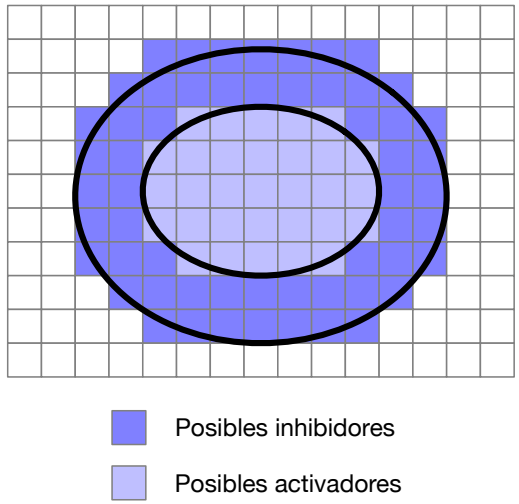
Autómatas en 2D: Patrones de Turing

- Dos regiones elípticas, concéntricas a la celda central, cuya vida determinan
- Las celdas vivas en la elipse interna constituyen los activadores (A)
- Las celdas fuera de la elipse interna pero dentro de la externa constituyen los inhibidores (I)
- Hay un factor w que dice que tan potentes son los inhibidores respecto a los activadores ($w = 2$, significa que son el doble de potentes)
- Calcular $F = A - w * I$
  - Si $F > 0$, la celda central vive
  - Si $F < 0$, la celda central muere
  - Si $F = 0$, la celda central no cambia su valor



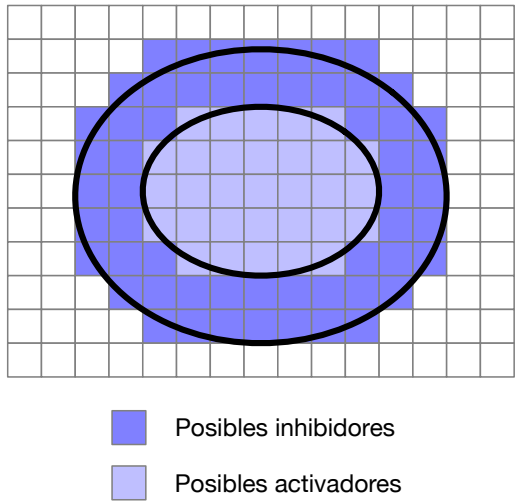
Autómatas en 2D: Patrones de Turing

- Dos regiones elípticas, concéntricas a la celda central, cuya vida determinan
- Las celdas vivas en la elipse interna constituyen los activadores (A)
- Las celdas fuera de la elipse interna pero dentro de la externa constituyen los inhibidores (I)
- Hay un factor w que dice que tan potentes son los inhibidores respecto a los activadores ($w = 2$, significa que son el doble de potentes)
- Calcular $F = A - w * I$
  - Si $F > 0$, la celda central vive
  - Si $F < 0$, la celda central muere
  - Si $F = 0$, la celda central no cambia su valor



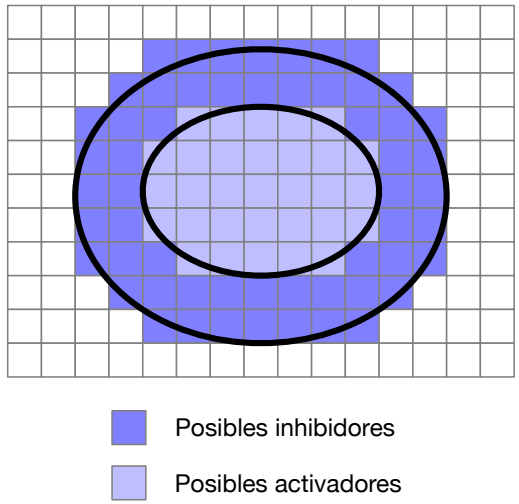
Autómatas en 2D: Patrones de Turing

- Dos regiones elípticas, concéntricas a la celda central, cuya vida determinan
- Las celdas vivas en la elipse interna constituyen los activadores (A)
- Las celdas fuera de la elipse interna pero dentro de la externa constituyen los inhibidores (I)
- Hay un factor w que dice que tan potentes son los inhibidores respecto a los activadores ($w = 2$, significa que son el doble de potentes)
- Calcular $F = A - w * I$
  - Si $F > 0$, la celda central vive
  - Si $F < 0$, la celda central muere
  - Si $F = 0$, la celda central no cambia su valor



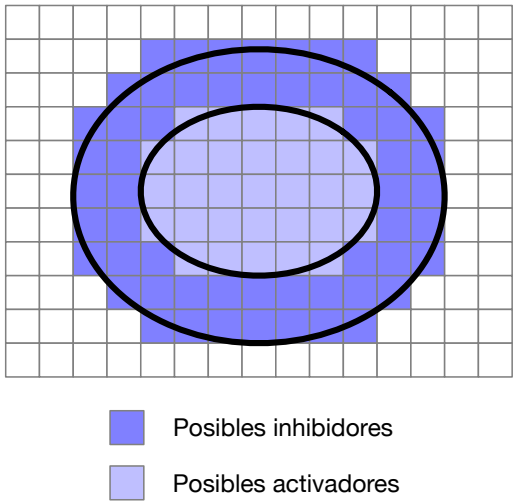
Autómatas en 2D: Patrones de Turing

- Dos regiones elípticas, concéntricas a la celda central, cuya vida determinan
- Las celdas vivas en la elipse interna constituyen los activadores (A)
- Las celdas fuera de la elipse interna pero dentro de la externa constituyen los inhibidores (I)
- Hay un factor w que dice que tan potentes son los inhibidores respecto a los activadores ($w = 2$, significa que son el doble de potentes)
- Calcular $F = A - w * I$
  - Si $F > 0$, la celda central vive
  - Si $F < 0$, la celda central muere
  - Si $F = 0$, la celda central no cambia su valor



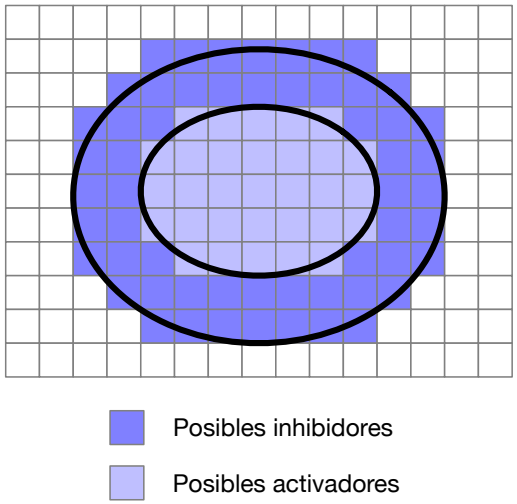
Autómatas en 2D: Patrones de Turing

- Dos regiones elípticas, concéntricas a la celda central, cuya vida determinan
- Las celdas vivas en la elipse interna constituyen los activadores (A)
- Las celdas fuera de la elipse interna pero dentro de la externa constituyen los inhibidores (I)
- Hay un factor w que dice que tan potentes son los inhibidores respecto a los activadores ($w = 2$, significa que son el doble de potentes)
- Calcular $F = A - w * I$
  - Si $F > 0$, la celda central vive
  - Si $F < 0$, la celda central muere
  - Si $F = 0$, la celda central no cambia su valor



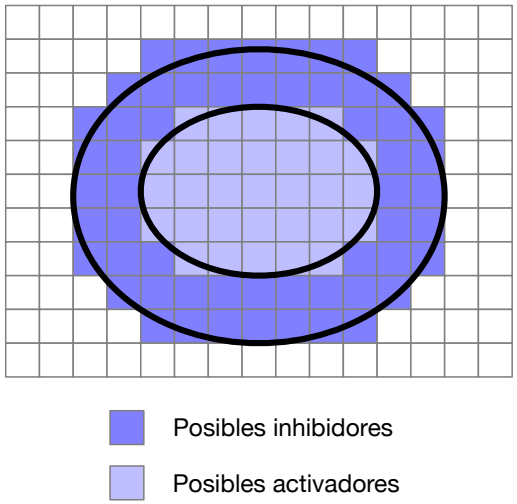
Autómatas en 2D: Patrones de Turing

- Dos regiones elípticas, concéntricas a la celda central, cuya vida determinan
- Las celdas vivas en la elipse interna constituyen los activadores (A)
- Las celdas fuera de la elipse interna pero dentro de la externa constituyen los inhibidores (I)
- Hay un factor w que dice que tan potentes son los inhibidores respecto a los activadores ($w = 2$, significa que son el doble de potentes)
- Calcular $F = A - w * I$
  - Si $F > 0$, la celda central vive
  - Si $F < 0$, la celda central muere
  - Si $F = 0$, la celda central no cambia su valor



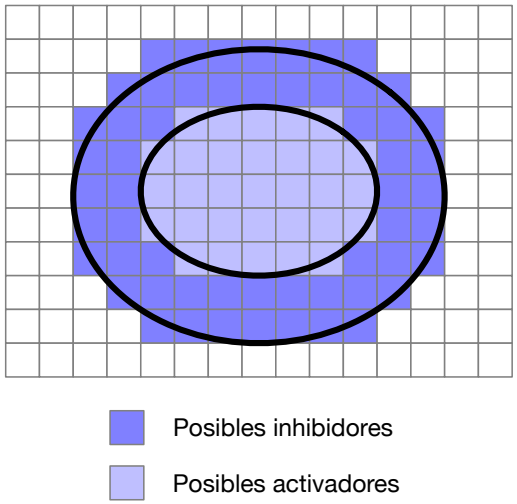
Autómatas en 2D: Patrones de Turing

- Dos regiones elípticas, concéntricas a la celda central, cuya vida determinan
- Las celdas vivas en la elipse interna constituyen los activadores (A)
- Las celdas fuera de la elipse interna pero dentro de la externa constituyen los inhibidores (I)
- Hay un factor w que dice que tan potentes son los inhibidores respecto a los activadores ($w = 2$, significa que son el doble de potentes)
- Calcular $F = A - w * I$
  - Si $F > 0$, la celda central vive
  - Si $F < 0$, la celda central muere
  - Si $F = 0$, la celda central no cambia su valor



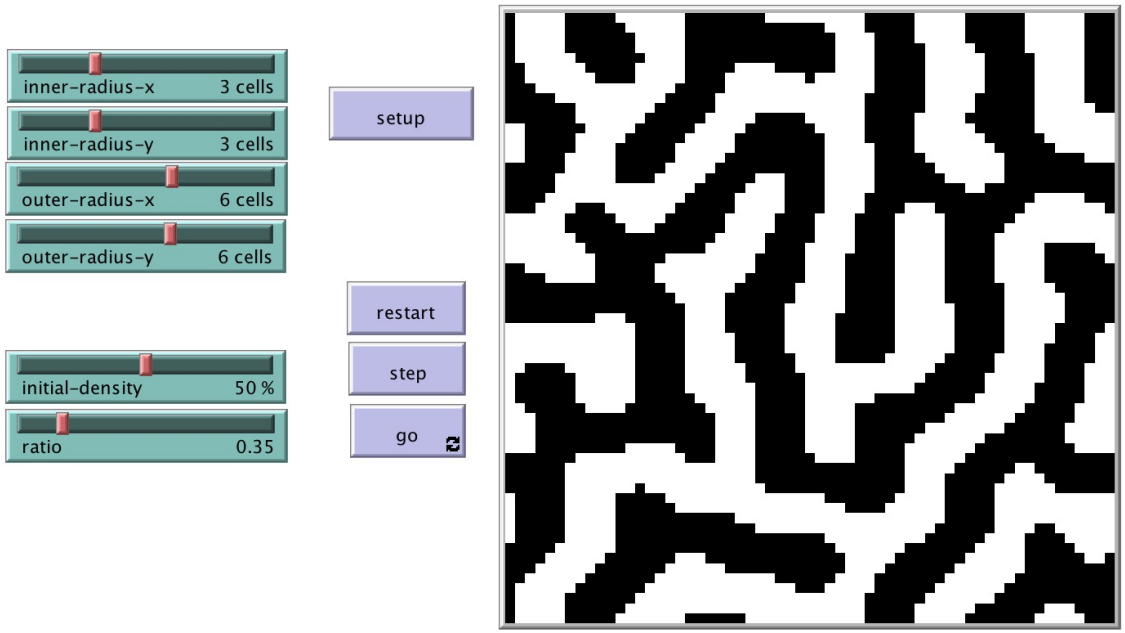
Autómatas en 2D: Patrones de Turing

- Dos regiones elípticas, concéntricas a la celda central, cuya vida determinan
- Las celdas vivas en la elipse interna constituyen los activadores (A)
- Las celdas fuera de la elipse interna pero dentro de la externa constituyen los inhibidores (I)
- Hay un factor w que dice que tan potentes son los inhibidores respecto a los activadores ($w = 2$, significa que son el doble de potentes)
- Calcular $F = A - w * I$
  - Si $F > 0$, la celda central vive
  - Si $F < 0$, la celda central muere
  - Si $F = 0$, la celda central no cambia su valor



Autómatas en 2D: Patrones de Turing

Programa "Fur" (biblioteca de modelos de Netlogo)



Autómatas en 2D: Patrones de Turing: Observaciones

- Nota como para ciertos valores de los parámetros, se observa la formación de patrones de "manchas", los cuales no dependen directamente de las reglas o el estado inicial de las celdas.
- Nota como en cierto momento estas "manchas", se estabilizan en cierta forma. Aunque algunas manchas, o fronteras de las mismas, podrían no llegar a estabilizarse.
- Nota también que esto aún puede considerarse autómata celular... discute por qué.

Autómatas en 2D: Patrones de Turing: Observaciones

- Nota como para ciertos valores de los parámetros, se observa la formación de patrones de "manchas", los cuales no dependen directamente de las reglas o el estado inicial de las celdas.
- Nota como en cierto momento estas "manchas", se estabilizan en cierta forma. Aunque algunas manchas, o fronteras de las mismas, podrían no llegar a estabilizarse.
- Nota también que esto aún puede considerarse autómata celular... discute por qué.

Autómatas en 2D: Patrones de Turing: Observaciones

- Nota como para ciertos valores de los parámetros, se observa la formación de patrones de "manchas", los cuales no dependen directamente de las reglas o el estado inicial de las celdas.
- Nota como en cierto momento estas "manchas", se estabilizan en cierta forma. Aunque algunas manchas, o fronteras de las mismas, podrían no llegar a estabilizarse.
- Nota también que esto aún puede considerarse autómata celular... discute por qué.

Percolación: definición

- Se le llama percolación al traslado de alguna substancia a través de un medio "poroso". La substancia avanza siguiendo posiciones adyacentes de dicho medio.
- Ejemplos de esto son: en la "cafetera de gotitas", el café que atraviesa el filtro se dice que ha percolado. El petróleo u otros líquidos que se extraen de la tierra, pueden percolar a través de arena o rocas.
- Es relativamente sencilla su modelación mediante autómatas celulares...
- Ver programas "Fire" y "Percolation", en Netlogo.

Percolación: definición

- Se le llama percolación al traslado de alguna substancia a través de un medio "poroso". La substancia avanza siguiendo posiciones adyacentes de dicho medio.
- Ejemplos de esto son: en la "cafetera de gotitas", el café que atraviesa el filtro se dice que ha percolado. El petróleo u otros líquidos que se extraen de la tierra, pueden percolar a través de arena o rocas.
- Es relativamente sencilla su modelación mediante autómatas celulares...
- Ver programas "Fire" y "Percolation", en Netlogo.

Percolación: definición

- Se le llama percolación al traslado de alguna substancia a través de un medio "poroso". La substancia avanza siguiendo posiciones adyacentes de dicho medio.
- Ejemplos de esto son: en la "cafetera de gotitas", el café que atraviesa el filtro se dice que ha percolado. El petróleo u otros líquidos que se extraen de la tierra, pueden percolar a través de arena o rocas.
- Es relativamente sencilla su modelación mediante autómatas celulares...
- Ver programas "Fire" y "Percolation", en Netlogo.

Percolación: definición

- Se le llama percolación al traslado de alguna substancia a través de un medio "poroso". La substancia avanza siguiendo posiciones adyacentes de dicho medio.
- Ejemplos de esto son: en la "cafetera de gotitas", el café que atraviesa el filtro se dice que ha percolado. El petróleo u otros líquidos que se extraen de la tierra, pueden percolar a través de arena o rocas.
- Es relativamente sencilla su modelación mediante autómatas celulares...
- Ver programas "Fire" y "Percolation", en Netlogo.

Percolación: modelo

- Una celda puede tener uno de tres estados: "desocupada", "ocupada" o "utilizada".
- El espacio disponible (rectángulo de celdas) se marcan unas celdas como desocupadas y otras como ocupadas, dependiendo de un valor llamado la "densidad".
- Luego de esto, en el llamado 1er estado, se marcan como "utilizadas", las celdas ocupadas de la columna más a la izquierda[1].
- En el siguiente estado, se marcan como "utilizadas", las celdas del siguiente renglón que estén ocupadas y sean adyacentes[2] a celdas utilizadas del estado anterior. Y así sucesivamente.
- Se dice que "percola" si al menos una celda de la columna de la derecha se torna "utilizada".

[1] percolación hacia la derecha, o el renglón de celdas ocupadas de más arriba, para percolar hacia abajo

[2] dentro de una vecindad de Moore, por ejemplo

Percolación: modelo

- Una celda puede tener uno de tres estados: “desocupada”, “ocupada” o “utilizada”.
- El espacio disponible (rectángulo de celdas) se marcan unas celdas como desocupadas y otras como ocupadas, dependiendo de un valor llamado la “densidad”.
- Luego de esto, en el llamado 1er estado, se marcan como “utilizadas”, las celdas ocupadas de la columna más a la izquierda[1].
- En el siguiente estado, se marcan como “utilizadas”, las celdas del siguiente renglón que estén ocupadas y sean adyacentes[2] a celdas utilizadas del estado anterior. Y así sucesivamente.
- Se dice que “percola” si al menos una celda de la columna de la derecha se torna “utilizada”.

[1] percolación hacia la derecha, o el renglón de celdas ocupadas de más arriba, para percolar hacia abajo

[2] dentro de una vecindad de Moore, por ejemplo

Percolación: modelo

- Una celda puede tener uno de tres estados: "desocupada", "ocupada" o "utilizada".
- El espacio disponible (rectángulo de celdas) se marcan unas celdas como desocupadas y otras como ocupadas, dependiendo de un valor llamado la "densidad".
- Luego de esto, en el llamado 1er estado, se marcan como "utilizadas", las celdas ocupadas de la columna más a la izquierda[1].
- En el siguiente estado, se marcan como "utilizadas", las celdas del siguiente renglón que estén ocupadas y sean adyacentes[2] a celdas utilizadas del estado anterior. Y así sucesivamente.
- Se dice que "percola" si al menos una celda de la columna de la derecha se torna "utilizada".

[1] percolación hacia la derecha, o el renglón de celdas ocupadas de más arriba, para percolar hacia abajo

[2] dentro de una vecindad de Moore, por ejemplo

Percolación: modelo

- Una celda puede tener uno de tres estados: "desocupada", "ocupada" o "utilizada".
- El espacio disponible (rectángulo de celdas) se marcan unas celdas como desocupadas y otras como ocupadas, dependiendo de un valor llamado la "densidad".
- Luego de esto, en el llamado 1er estado, se marcan como "utilizadas", las celdas ocupadas de la columna más a la izquierda[1].
- En el siguiente estado, se marcan como "utilizadas", las celdas del siguiente renglón que estén ocupadas y sean adyacentes[2] a celdas utilizadas del estado anterior. Y así sucesivamente.
- Se dice que "percola" si al menos una celda de la columna de la derecha se torna "utilizada".

[1] percolación hacia la derecha, o el renglón de celdas ocupadas de más arriba, para percolar hacia abajo

[2] dentro de una vecindad de Moore, por ejemplo

Percolación: modelo

- Una celda puede tener uno de tres estados: "desocupada", "ocupada" o "utilizada".
- El espacio disponible (rectángulo de celdas) se marcan unas celdas como desocupadas y otras como ocupadas, dependiendo de un valor llamado la "densidad".
- Luego de esto, en el llamado 1er estado, se marcan como "utilizadas", las celdas ocupadas de la columna más a la izquierda[1].
- En el siguiente estado, se marcan como "utilizadas", las celdas del siguiente renglón que estén ocupadas y sean adyacentes[2] a celdas utilizadas del estado anterior. Y así sucesivamente.
- Se dice que "percola" si al menos una celda de la columna de la derecha se torna "utilizada".

[1] percolación hacia la derecha, o el renglón de celdas ocupadas de más arriba, para percolar hacia abajo

[2] dentro de una vecindad de Moore, por ejemplo

Percolación: variantes

- Los modelos de Netlogo presentan unas variantes a esta definición
- En “fire”, toda celda utilizada “contagia” a sus vecinas ocupadas, que pueden estar en columnas anteriores o siguientes.
- En “percolation”, no hay celdas desocupadas, pero una utilizada “contagia” (o no) a cada una de sus dos vecinas del siguiente renglón, con cierta probabilidad, llamada “porosidad”.

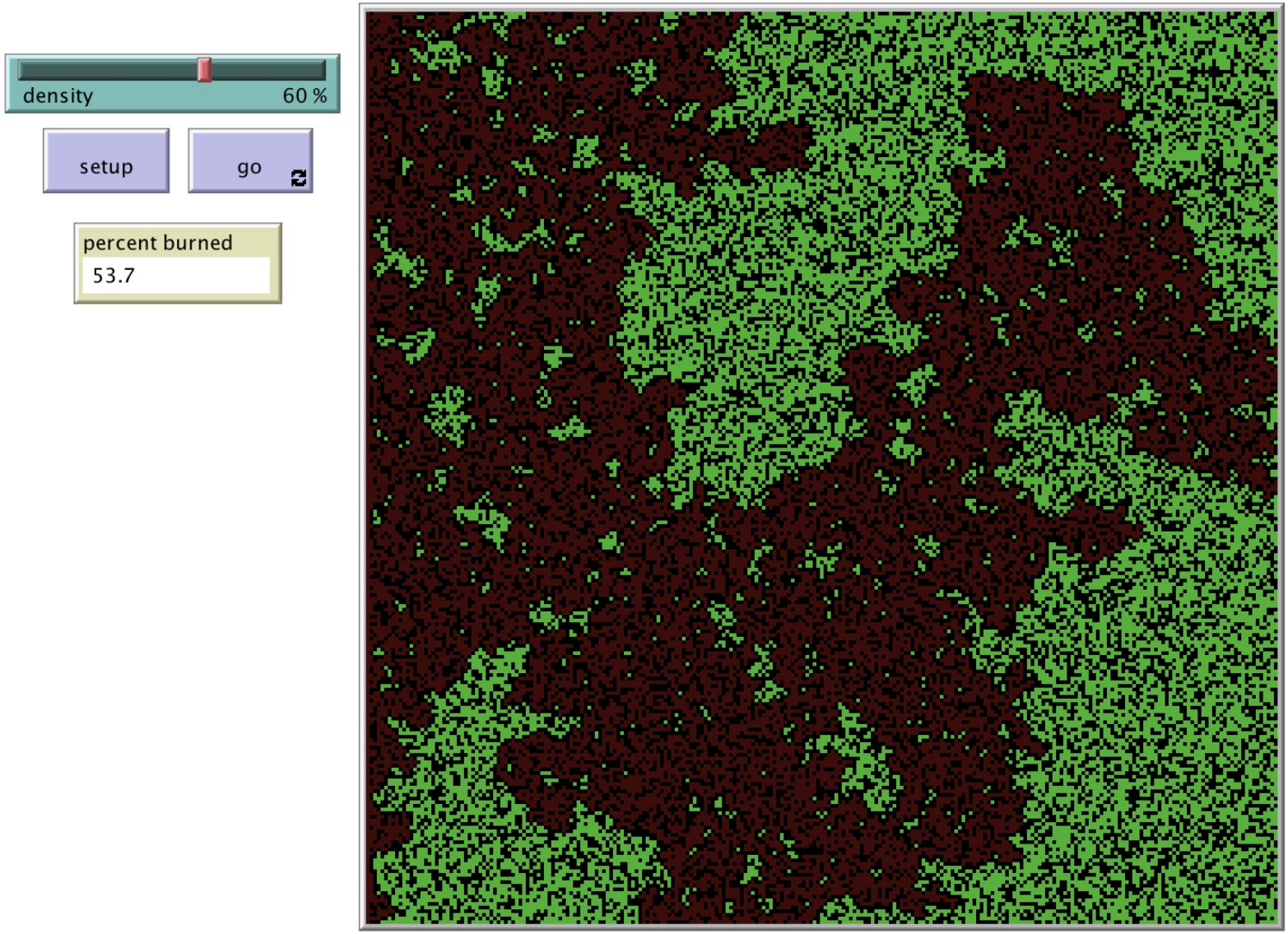
Percolación: variantes

- Los modelos de Netlogo presentan unas variantes a esta definición
- En “fire”, toda celda utilizada “contagia” a sus vecinas ocupadas, que pueden estar en columnas anteriores o siguientes.
- En “percolation”, no hay celdas desocupadas, pero una utilizada “contagia” (o no) a cada una de sus dos vecinas del siguiente renglón, con cierta probabilidad, llamada “porosidad”.

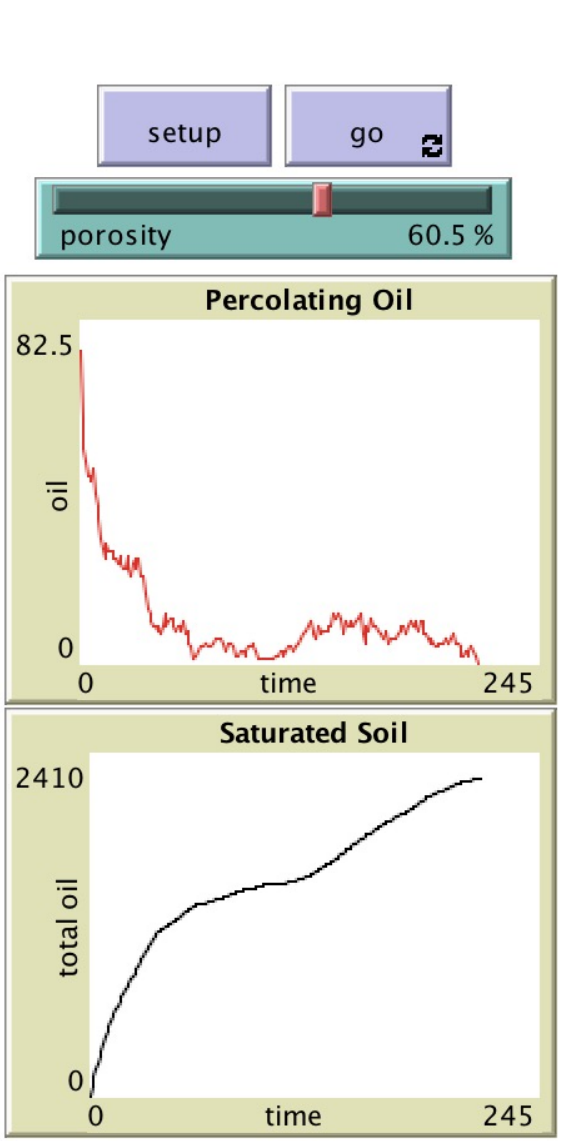
Percolación: variantes

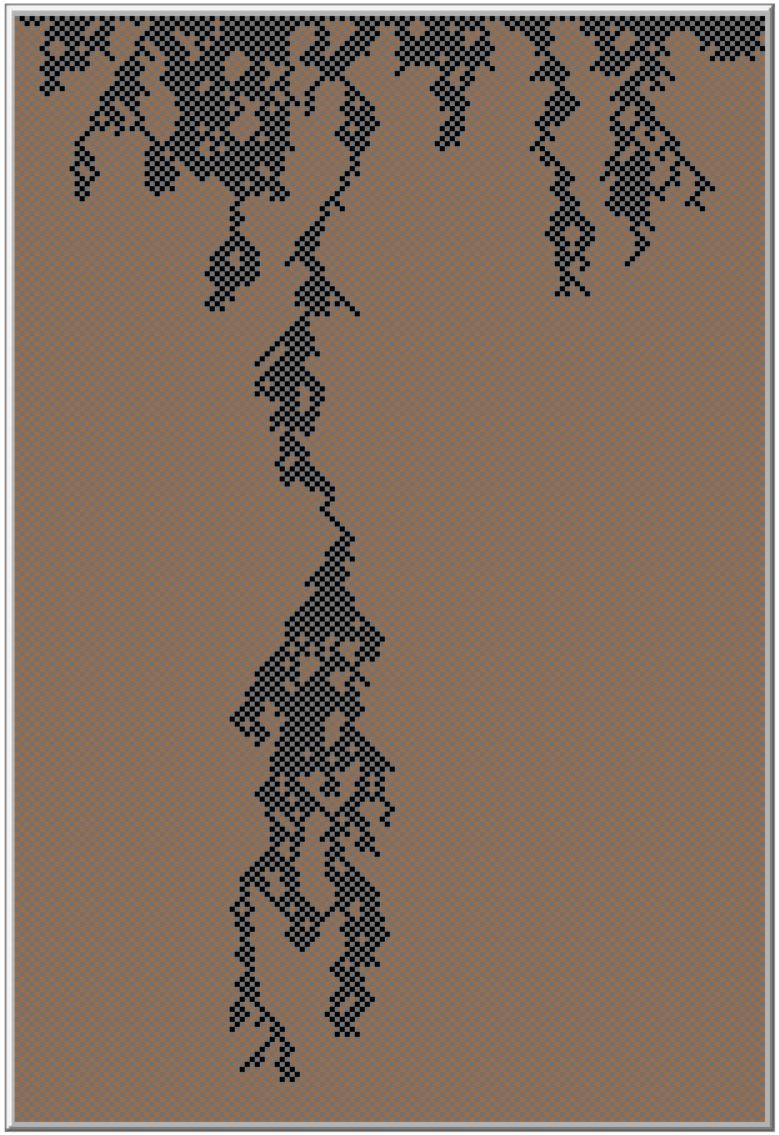
- Los modelos de Netlogo presentan unas variantes a esta definición
- En “fire”, toda celda utilizada “contagia” a sus vecinas ocupadas, que pueden estar en columnas anteriores o siguientes.
- En “percolation”, no hay celdas desocupadas, pero una utilizada “contagia” (o no) a cada una de sus dos vecinas del siguiente renglón, con cierta probabilidad, llamada “porosidad”.

Percolación: ejemplos



Percolación: ejemplos





Percolación: obaservaciones

- Variando la densidad, pudiera no "percolar". Dicho de otra forma, habrá un estado en que no haya suficientes celdas ocupadas adyacentes, a celdas utilizadas en el estado anterior.
- Se puede localizar el punto exacto[3] de densidad en que ya se logra percolar[4].
- Observa que con densidad cero, no percola, y con densidad uno, si. Por lo que es posible encontrar el "punto a partir del cual, el comportamiento cualitativo del sistema cambia".
- Nota también que esto aún puede considerarse autómata celular... discute por qué.

[3] en realidad, como interviene el azar, esto depende de la precisión

[4] de nuevo, como interviene el azar, hay que realizar varios experimentos con la misma densidad, para ver si se obtiene el mismo resultado

Percolación: obaservaciones

- Variando la densidad, pudiera no "percolar". Dicho de otra forma, habrá un estado en que no haya suficientes celdas ocupadas adyacentes, a celdas utilizadas en el estado anterior.
- Se puede localizar el punto exacto[3] de densidad en que ya se logra percolar[4].
- Observa que con densidad cero, no percola, y con densidad uno, si. Por lo que es posible encontrar el "punto a partir del cual, el comportamiento cualitativo del sistema cambia".
- Nota también que esto aún puede considerarse autómata celular... discute por qué.

[3] en realidad, como interviene el azar, esto depende de la precisión

[4] de nuevo, como interviene el azar, hay que realizar varios experimentos con la misma densidad, para ver si se obtiene el mismo resultado

Percolación: obaservaciones

- Variando la densidad, pudiera no "percolar". Dicho de otra forma, habrá un estado en que no haya suficientes celdas ocupadas adyacentes, a celdas utilizadas en el estado anterior.
- Se puede localizar el punto exacto[3] de densidad en que ya se logra percolar[4].
- Observa que con densidad cero, no percola, y con densidad uno, si. Por lo que es posible encontrar el "punto a partir del cual, el comportamiento cualitativo del sistema cambia".
- Nota también que esto aún puede considerarse autómata celular... discute por qué.

[3] en realidad, como interviene el azar, esto depende de la precisión

[4] de nuevo, como interviene el azar, hay que realizar varios experimentos con la misma densidad, para ver si se obtiene el mismo resultado

Percolación: obaservaciones

- Variando la densidad, pudiera no "percolar". Dicho de otra forma, habrá un estado en que no haya suficientes celdas ocupadas adyacentes, a celdas utilizadas en el estado anterior.
- Se puede localizar el punto exacto[3] de densidad en que ya se logra percolar[4].
- Observa que con densidad cero, no percola, y con densidad uno, si. Por lo que es posible encontrar el "punto a partir del cual, el comportamiento cualitativo del sistema cambia".
- Nota también que esto aún puede considerarse autómata celular... discute por qué.

[3] en realidad, como interviene el azar, esto depende de la precisión

[4] de nuevo, como interviene el azar, hay que realizar varios experimentos con la misma densidad, para ver si se obtiene el mismo resultado

Modelos Basados en Agentes

Agentes

- Agente: elemento individual autónomo de una simulación computacional
- Tiene:
  - Movimiento dentro de un ambiente ("mundo" en Netlogo)
  - Características como memoria (variables)
  - La autonomía de acción (funciones) que se le dé
  - Capacidad de interactuar con su ambiente y otros agentes
  - Los estados son menos claros, pero puede inspeccionarse el estado del sistema en un momento dado
- En general tienen menos restricciones que los autómatas
- ¿Un agente generaliza los autómatas? o ¿todo se puede modelar con autómatas?

Agentes

- Agente: elemento individual autónomo de una simulación computacional
- Tiene:
  - Movimiento dentro de un ambiente ("mundo" en Netlogo)
  - Características como memoria (variables)
  - La autonomía de acción (funciones) que se le dé
  - Capacidad de interactuar con su ambiente y otros agentes
  - Los estados son menos claros, pero puede inspeccionarse el estado del sistema en un momento dado
- En general tienen menos restricciones que los autómatas
- ¿Un agente generaliza los autómatas? o ¿todo se puede modelar con autómatas?

Agentes

- Agente: elemento individual autónomo de una simulación computacional
- Tiene:
  - Movimiento dentro de un ambiente ("mundo" en Netlogo)
  - Características como memoria (variables)
  - La autonomía de acción (funciones) que se le dé
  - Capacidad de interactuar con su ambiente y otros agentes
  - Los estados son menos claros, pero puede inspeccionarse el estado del sistema en un momento dado
- En general tienen menos restricciones que los autómatas
- ¿Un agente generaliza los autómatas? o ¿todo se puede modelar con autómatas?

Agentes

- Agente: elemento individual autónomo de una simulación computacional
- Tiene:
  - Movimiento dentro de un ambiente ("mundo" en Netlogo)
  - Características como memoria (variables)
  - La autonomía de acción (funciones) que se le dé
  - Capacidad de interactuar con su ambiente y otros agentes
  - Los estados son menos claros, pero puede inspeccionarse el estado del sistema en un momento dado
- En general tienen menos restricciones que los autómatas
- ¿Un agente generaliza los autómatas? o ¿todo se puede modelar con autómatas?

Agentes

- Agente: elemento individual autónomo de una simulación computacional
- Tiene:
  - Movimiento dentro de un ambiente ("mundo" en Netlogo)
  - Características como memoria (variables)
  - La autonomía de acción (funciones) que se le dé
  - Capacidad de interactuar con su ambiente y otros agentes
  - Los estados son menos claros, pero puede inspeccionarse el estado del sistema en un momento dado
- En general tienen menos restricciones que los autómatas
- ¿Un agente generaliza los autómatas? o ¿todo se puede modelar con autómatas?

Agentes

- Agente: elemento individual autónomo de una simulación computacional
- Tiene:
  - Movimiento dentro de un ambiente ("mundo" en Netlogo)
  - Características como memoria (variables)
  - La autonomía de acción (funciones) que se le dé
  - Capacidad de interactuar con su ambiente y otros agentes
  - Los estados son menos claros, pero puede inspeccionarse el estado del sistema en un momento dado
- En general tienen menos restricciones que los autómatas
- ¿Un agente generaliza los autómatas? o ¿todo se puede modelar con autómatas?

Agentes

- Agente: elemento individual autónomo de una simulación computacional
- Tiene:
  - Movimiento dentro de un ambiente ("mundo" en Netlogo)
  - Características como memoria (variables)
  - La autonomía de acción (funciones) que se le dé
  - Capacidad de interactuar con su ambiente y otros agentes
  - Los estados son menos claros, pero puede inspeccionarse el estado del sistema en un momento dado
- En general tienen menos restricciones que los autómatas
- ¿Un agente generaliza los autómatas? o ¿todo se puede modelar con autómatas?

Agentes

- Agente: elemento individual autónomo de una simulación computacional
- Tiene:
  - Movimiento dentro de un ambiente ("mundo" en Netlogo)
  - Características como memoria (variables)
  - La autonomía de acción (funciones) que se le dé
  - Capacidad de interactuar con su ambiente y otros agentes
  - Los estados son menos claros, pero puede inspeccionarse el estado del sistema en un momento dado
- En general tienen menos restricciones que los autómatas
- ¿Un agente generaliza los autómatas? o ¿todo se puede modelar con autómatas?

Agentes

- Agente: elemento individual autónomo de una simulación computacional
- Tiene:
  - Movimiento dentro de un ambiente ("mundo" en Netlogo)
  - Características como memoria (variables)
  - La autonomía de acción (funciones) que se le dé
  - Capacidad de interactuar con su ambiente y otros agentes
  - Los estados son menos claros, pero puede inspeccionarse el estado del sistema en un momento dado
- En general tienen menos restricciones que los autómatas
- ¿Un agente generaliza los autómatas? o ¿todo se puede modelar con autómatas?

Introducción a Netlogo

- Lenguaje para Sistemas Complejos desarrollado por Uri Wilensky
- Puede verse como una generalización del lenguaje "Logo" (inicialmente desarrollado en el MIT)
- Permite manejar "agentes", que son principalmente las "tortugas" (que pueden adquirir personalidades), parches (estáticos, pero fuera de eso, son agentes también), y "ligas" (aristas entre tortugas, que permiten identificar las relacionadas, de ser necesario)
- Para programas chicos y medianos, su programación es bastante sencilla, para programas grandes, lo mejor es buscar otro lenguaje más avanzado (Python, Java, C, C++)

Introducción a Netlogo

- Lenguaje para Sistemas Complejos desarrollado por Uri Wilensky
- Puede verse como una generalización del lenguaje "Logo" (inicialmente desarrollado en el MIT)
- Permite manejar "agentes", que son principalmente las "tortugas" (que pueden adquirir personalidades), parches (estáticos, pero fuera de eso, son agentes también), y "ligas" (aristas entre tortugas, que permiten identificar las relacionadas, de ser necesario)
- Para programas chicos y medianos, su programación es bastante sencilla, para programas grandes, lo mejor es buscar otro lenguaje más avanzado (Python, Java, C, C++)

Introducción a Netlogo

- Lenguaje para Sistemas Complejos desarrollado por Uri Wilensky
- Puede verse como una generalización del lenguaje "Logo" (inicialmente desarrollado en el MIT)
- Permite manejar "agentes", que son principalmente las "tortugas" (que pueden adquirir personalidades), parches (estáticos, pero fuera de eso, son agentes también), y "ligas" (aristas entre tortugas, que permiten identificar las relacionadas, de ser necesario)
- Para programas chicos y medianos, su programación es bastante sencilla, para programas grandes, lo mejor es buscar otro lenguaje más avanzado (Python, Java, C, C++)

Introducción a Netlogo

- Lenguaje para Sistemas Complejos desarrollado por Uri Wilensky
- Puede verse como una generalización del lenguaje "Logo" (inicialmente desarrollado en el MIT)
- Permite manejar "agentes", que son principalmente las "tortugas" (que pueden adquirir personalidades), parches (estáticos, pero fuera de eso, son agentes también), y "ligas" (aristas entre tortugas, que permiten identificar las relacionadas, de ser necesario)
- Para programas chicos y medianos, su programación es bastante sencilla, para programas grandes, lo mejor es buscar otro lenguaje más avanzado (Python, Java, C, C++)

Elementos de Netlogo

- Pestaña “Ejecutar”
  - “Añadir” (“Botón”, “Deslizador”, “Interruptor”, “Seleccionador”, “Entrada”, “Monitor”, “Gráfico”, “Salida”, “Nota”)
  - Velocidad
  - ticks
  - Actualizar de la Vista (“continuamente”, “manualmente - ticks”)
  - Configuración
  - Mundo
  - Terminal de Instrucciones, Borrar, observador>
- Pestaña “Información” (Documentación del programa)
- Pestaña “Código” (Buscar, Comprobar, Procedimientos, Sangrado automático)
- Menús

Elementos de Netlogo

- Pestaña “Ejecutar”
  - “Añadir” (“Botón”, “Deslizador”, “Interruptor”, “Seleccionador”, “Entrada”, “Monitor”, “Gráfico”, “Salida”, “Nota”)
  - Velocidad
  - ticks
  - Actualizar de la Vista (“continuamente”, “manualmente - ticks”)
  - Configuración
  - Mundo
  - Terminal de Instrucciones, Borrar, observador>
- Pestaña “Información” (Documentación del programa)
- Pestaña “Código” (Buscar, Comprobar, Procedimientos, Sangrado automático)
- Menús

Elementos de Netlogo

- Pestaña “Ejecutar”
  - “Añadir” (“Botón”, “Deslizador”, “Interruptor”, “Seleccionador”, “Entrada”, “Monitor”, “Gráfico”, “Salida”, “Nota”)
  - Velocidad
  - ticks
  - Actualizar de la Vista (“continuamente”, “manualmente - ticks”)
  - Configuración
  - Mundo
  - Terminal de Instrucciones, Borrar, observador>
- Pestaña “Información” (Documentación del programa)
- Pestaña “Código” (Buscar, Comprobar, Procedimientos, Sangrado automático)
- Menús

Elementos de Netlogo

- Pestaña “Ejecutar”
  - “Añadir” (“Botón”, “Deslizador”, “Interruptor”, “Seleccionador”, “Entrada”, “Monitor”, “Gráfico”, “Salida”, “Nota”)
  - Velocidad
  - ticks
  - Actualizar de la Vista (“continuamente”, “manualmente - ticks”)
  - Configuración
  - Mundo
  - Terminal de Instrucciones, Borrar, observador>
- Pestaña “Información” (Documentación del programa)
- Pestaña “Código” (Buscar, Comprobar, Procedimientos, Sangrado automático)
- Menús

Elementos de Netlogo

- Pestaña “Ejecutar”
  - “Añadir” (“Botón”, “Deslizador”, “Interruptor”, “Seleccionador”, “Entrada”, “Monitor”, “Gráfico”, “Salida”, “Nota”)
  - Velocidad
  - ticks
  - Actualizar de la Vista (“continuamente”, “manualmente - ticks”)
  - Configuración
  - Mundo
  - Terminal de Instrucciones, Borrar, observador>
- Pestaña “Información” (Documentación del programa)
- Pestaña “Código” (Buscar, Comprobar, Procedimientos, Sangrado automático)
- Menús

Elementos de Netlogo

- Pestaña “Ejecutar”
  - “Añadir” (“Botón”, “Deslizador”, “Interruptor”, “Seleccionador”, “Entrada”, “Monitor”, “Gráfico”, “Salida”, “Nota”)
  - Velocidad
  - ticks
  - Actualizar de la Vista (“continuamente”, “manualmente - ticks”)
  - Configuración
  - Mundo
  - Terminal de Instrucciones, Borrar, observador>
- Pestaña “Información” (Documentación del programa)
- Pestaña “Código” (Buscar, Comprobar, Procedimientos, Sangrado automático)
- Menús

Elementos de Netlogo

- Pestaña “Ejecutar”
  - “Añadir” (“Botón”, “Deslizador”, “Interruptor”, “Seleccionador”, “Entrada”, “Monitor”, “Gráfico”, “Salida”, “Nota”)
  - Velocidad
  - ticks
  - Actualizar de la Vista (“continuamente”, “manualmente - ticks”)
  - Configuración
  - Mundo
  - Terminal de Instrucciones, Borrar, observador>
- Pestaña “Información” (Documentación del programa)
- Pestaña “Código” (Buscar, Comprobar, Procedimientos, Sangrado automático)
- Menús

Elementos de Netlogo

- Pestaña “Ejecutar”
  - “Añadir” (“Botón”, “Deslizador”, “Interruptor”, “Seleccionador”, “Entrada”, “Monitor”, “Gráfico”, “Salida”, “Nota”)
  - Velocidad
  - ticks
  - Actualizar de la Vista (“continuamente”, “manualmente - ticks”)
  - Configuración
  - Mundo
  - Terminal de Instrucciones, Borrar, observador>
- Pestaña “Información” (Documentación del programa)
- Pestaña “Código” (Buscar, Comprobar, Procedimientos, Sangrado automático)
- Menús

Elementos de Netlogo

- Pestaña “Ejecutar”
  - “Añadir” (“Botón”, “Deslizador”, “Interruptor”, “Seleccionador”, “Entrada”, “Monitor”, “Gráfico”, “Salida”, “Nota”)
  - Velocidad
  - ticks
  - Actualizar de la Vista (“continuamente”, “manualmente - ticks”)
  - Configuración
  - Mundo
  - Terminal de Instrucciones, Borrar, observador>
- Pestaña “Información” (Documentación del programa)
- Pestaña “Código” (Buscar, Comprobar, Procedimientos, Sangrado automático)
- Menús

Elementos de Netlogo

- Pestaña “Ejecutar”
  - “Añadir” (“Botón”, “Deslizador”, “Interruptor”, “Seleccionador”, “Entrada”, “Monitor”, “Gráfico”, “Salida”, “Nota”)
  - Velocidad
  - ticks
  - Actualizar de la Vista (“continuamente”, “manualmente - ticks”)
  - Configuración
  - Mundo
  - Terminal de Instrucciones, Borrar, observador>
- Pestaña “Información” (Documentación del programa)
- Pestaña “Código” (Buscar, Comprobar, Procedimientos, Sangrado automático)
- Menús

Elementos de Netlogo

- Pestaña “Ejecutar”
  - “Añadir” (“Botón”, “Deslizador”, “Interruptor”, “Seleccionador”, “Entrada”, “Monitor”, “Gráfico”, “Salida”, “Nota”)
  - Velocidad
  - ticks
  - Actualizar de la Vista (“continuamente”, “manualmente - ticks”)
  - Configuración
  - Mundo
  - Terminal de Instrucciones, Borrar, observador>
- Pestaña “Información” (Documentación del programa)
- Pestaña “Código” (Buscar, Comprobar, Procedimientos, Sangrado automático)
- Menús

Mundo

- Es el ambiente donde se mueven los agentes
- Está compuesto de cuadritos contiguos llamados "parcelas" o "parches"
- Consiste de agentes estáticos, ¿generalización de autómatas?
- Cada parche tiene coordenadas (enteras para su centro)
- En "configuración" (o "editar" con el botón derecho) se pueden modificar sus dimensiones, tamaño de parches, inicio y dirección de coordenadas, condiciones de frontera, etc.
- Se puede desplazar con el ratón, pero es único.
- El origen de sus coordenadas, el parche $(0, 0)$, está en el centro.
- Las coordenadas x y y crecen hacia la derecha y hacia arriba, y decrecen en los otros sentidos.

Mundo

- Es el ambiente donde se mueven los agentes
- Está compuesto de cuadritos contiguos llamados "parcelas" o "parches"
- Consiste de agentes estáticos, ¿generalización de autómatas?
- Cada parche tiene coordenadas (enteras para su centro)
- En "configuración" (o "editar" con el botón derecho) se pueden modificar sus dimensiones, tamaño de parches, inicio y dirección de coordenadas, condiciones de frontera, etc.
- Se puede desplazar con el ratón, pero es único.
- El origen de sus coordenadas, el parche $(0, 0)$, está en el centro.
- Las coordenadas x y y crecen hacia la derecha y hacia arriba, y decrecen en los otros sentidos.

Mundo

- Es el ambiente donde se mueven los agentes
- Está compuesto de cuadritos contiguos llamados "parcelas" o "parches"
- Consiste de agentes estáticos, ¿generalización de autómatas?
- Cada parche tiene coordenadas (enteras para su centro)
- En "configuración" (o "editar" con el botón derecho) se pueden modificar sus dimensiones, tamaño de parches, inicio y dirección de coordenadas, condiciones de frontera, etc.
- Se puede desplazar con el ratón, pero es único.
- El origen de sus coordenadas, el parche $(0, 0)$, está en el centro.
- Las coordenadas x y y crecen hacia la derecha y hacia arriba, y decrecen en los otros sentidos.

Mundo

- Es el ambiente donde se mueven los agentes
- Está compuesto de cuadritos contiguos llamados "parcelas" o "parches"
- Consiste de agentes estáticos, ¿generalización de autómatas?
- Cada parche tiene coordenadas (enteras para su centro)
- En "configuración" (o "editar" con el botón derecho) se pueden modificar sus dimensiones, tamaño de parches, inicio y dirección de coordenadas, condiciones de frontera, etc.
- Se puede desplazar con el ratón, pero es único.
- El origen de sus coordenadas, el parche $(0, 0)$, está en el centro.
- Las coordenadas x y y crecen hacia la derecha y hacia arriba, y decrecen en los otros sentidos.

Mundo

- Es el ambiente donde se mueven los agentes
- Está compuesto de cuadritos contiguos llamados "parcelas" o "parches"
- Consiste de agentes estáticos, ¿generalización de autómatas?
- Cada parche tiene coordenadas (enteras para su centro)
- En "configuración" (o "editar" con el botón derecho) se pueden modificar sus dimensiones, tamaño de parches, inicio y dirección de coordenadas, condiciones de frontera, etc.
- Se puede desplazar con el ratón, pero es único.
- El origen de sus coordenadas, el parche $(0, 0)$, está en el centro.
- Las coordenadas x y y crecen hacia la derecha y hacia arriba, y decrecen en los otros sentidos.

Mundo

- Es el ambiente donde se mueven los agentes
- Está compuesto de cuadritos contiguos llamados "parcelas" o "parches"
- Consiste de agentes estáticos, ¿generalización de autómatas?
- Cada parche tiene coordenadas (enteras para su centro)
- En "configuración" (o "editar" con el botón derecho) se pueden modificar sus dimensiones, tamaño de parches, inicio y dirección de coordenadas, condiciones de frontera, etc.
- Se puede desplazar con el ratón, pero es único.
- El origen de sus coordenadas, el parche $(0, 0)$, está en el centro.
- Las coordenadas x y y crecen hacia la derecha y hacia arriba, y decrecen en los otros sentidos.

Mundo

- Es el ambiente donde se mueven los agentes
- Está compuesto de cuadritos contiguos llamados "parcelas" o "parches"
- Consiste de agentes estáticos, ¿generalización de autómatas?
- Cada parche tiene coordenadas (enteras para su centro)
- En "configuración" (o "editar" con el botón derecho) se pueden modificar sus dimensiones, tamaño de parches, inicio y dirección de coordenadas, condiciones de frontera, etc.
- Se puede desplazar con el ratón, pero es único.
- El origen de sus coordenadas, el parche $(0, 0)$, está en el centro.
- Las coordenadas x y y crecen hacia la derecha y hacia arriba, y decrecen en los otros sentidos.

Mundo

- Es el ambiente donde se mueven los agentes
- Está compuesto de cuadritos contiguos llamados "parcelas" o "parches"
- Consiste de agentes estáticos, ¿generalización de autómatas?
- Cada parche tiene coordenadas (enteras para su centro)
- En "configuración" (o "editar" con el botón derecho) se pueden modificar sus dimensiones, tamaño de parches, inicio y dirección de coordenadas, condiciones de frontera, etc.
- Se puede desplazar con el ratón, pero es único.
- El origen de sus coordenadas, el parche $(0, 0)$, está en el centro.
- Las coordenadas x y y crecen hacia la derecha y hacia arriba, y decrecen en los otros sentidos.

Programas (códigos) iniciales

- Por lo general en la ventana de código se agregan dos procedimientos: “setup” y “go”
- “setup” ajusta las condiciones iniciales para el mundo y demás agentes
- “go” pone en marcha cualquier proceso que requiera la simulación
- Además de esto se suelen agregar deslizadores o cuadros de entrada para introducir o modificar los parámetros de la simulación
- El código mínimo para el setup es el siguiente:

```
to setup
    clear-all  ; se abrevia: ca
    ; agrega aquí inicialización de variables,
    ;  creación de tortugas, etc.
    ; nota los espacios a la izq. de estas líneas
    ;  indicando que están ''dentro'' de setup
    reset-ticks  ; inicializa el contador de pasos (ticks)
end
```

Programas (códigos) iniciales

- Por lo general en la ventana de código se agregan dos procedimientos: “setup” y “go”
- “setup” ajusta las condiciones iniciales para el mundo y demás agentes
- “go” pone en marcha cualquier proceso que requiera la simulación
- Además de esto se suelen agregar deslizadores o cuadros de entrada para introducir o modificar los parámetros de la simulación
- El código mínimo para el setup es el siguiente:

```
to setup
    clear-all  ; se abrevia: ca
    ; agrega aquí inicialización de variables,
    ;  creación de tortugas, etc.
    ; nota los espacios a la izq. de estas líneas
    ;  indicando que están ''dentro'' de setup
    reset-ticks  ; inicializa el contador de pasos (ticks)
end
```

Programas (códigos) iniciales

- Por lo general en la ventana de código se agregan dos procedimientos: "setup" y "go"
- "setup" ajusta las condiciones iniciales para el mundo y demás agentes
- "go" pone en marcha cualquier proceso que requiera la simulación
- Además de esto se suelen agregar deslizadores o cuadros de entrada para introducir o modificar los parámetros de la simulación
- El código mínimo para el setup es el siguiente:

```
to setup
    clear-all  ; se abrevia: ca
    ; agrega aquí inicialización de variables,
    ;  creación de tortugas, etc.
    ; nota los espacios a la izq. de estas líneas
    ;  indicando que están ''dentro'' de setup
    reset-ticks  ; inicializa el contador de pasos (ticks)
end
```

Programas (códigos) iniciales

- Por lo general en la ventana de código se agregan dos procedimientos: “setup” y “go”
- “setup” ajusta las condiciones iniciales para el mundo y demás agentes
- “go” pone en marcha cualquier proceso que requiera la simulación
- Además de esto se suelen agregar deslizadores o cuadros de entrada para introducir o modificar los parámetros de la simulación
- El código mínimo para el setup es el siguiente:

```
to setup
    clear-all  ; se abrevia: ca
    ; agrega aquí inicialización de variables,
    ;  creación de tortugas, etc.
    ; nota los espacios a la izq. de estas líneas
    ;  indicando que están ''dentro'' de setup
    reset-ticks  ; inicializa el contador de pasos (ticks)
end
```

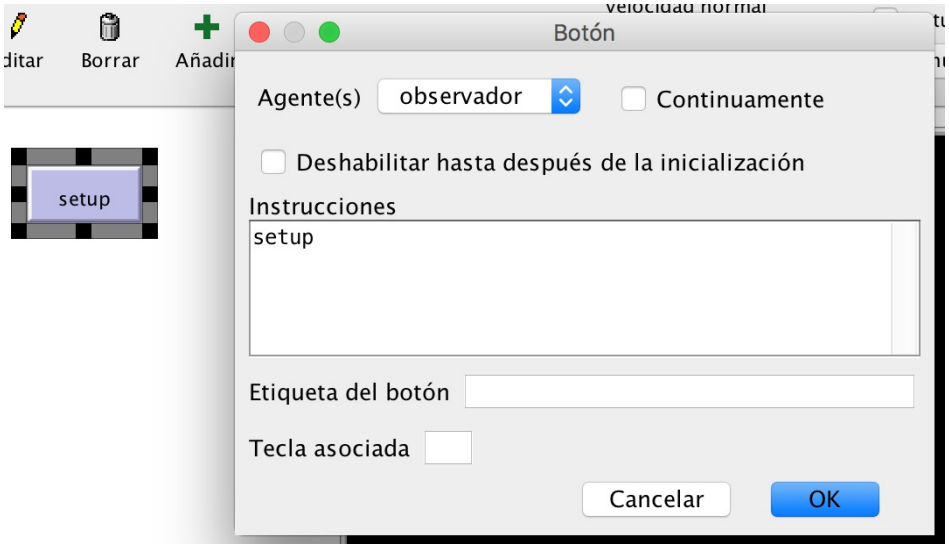
Programas (códigos) iniciales

- Por lo general en la ventana de código se agregan dos procedimientos: "setup" y "go"
- "setup" ajusta las condiciones iniciales para el mundo y demás agentes
- "go" pone en marcha cualquier proceso que requiera la simulación
- Además de esto se suelen agregar deslizadores o cuadros de entrada para introducir o modificar los parámetros de la simulación
- El código mínimo para el setup es el siguiente:

```
to setup
    clear-all  ; se abrevia: ca
    ; agrega aquí inicialización de variables,
    ;  creación de tortugas, etc.
    ; nota los espacios a la izq. de estas líneas
    ;  indicando que están ''dentro'' de setup
    reset-ticks  ; inicializa el contador de pasos (ticks)
end
```

Programas iniciales

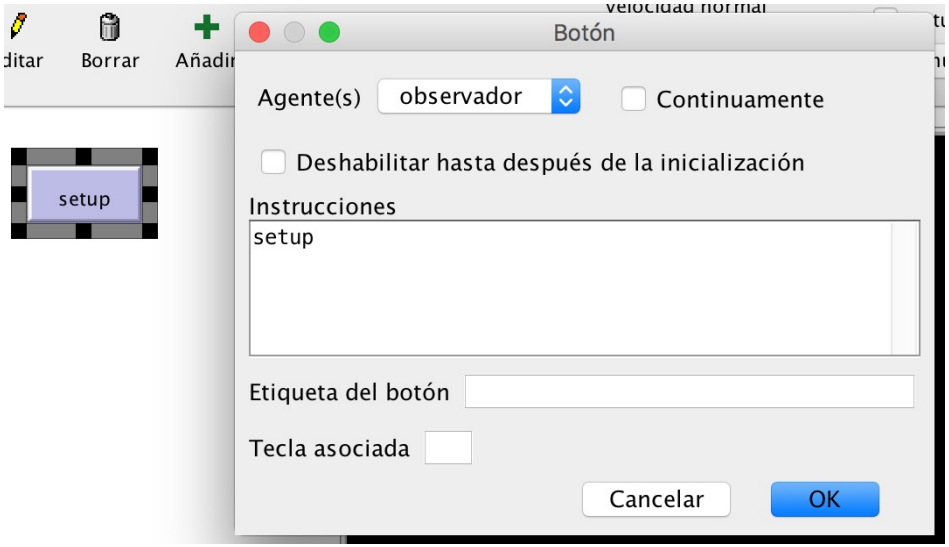
- Luego de teclear este código, hay que hacer que se ejecute, por ejemplo con un botón cuyo único contenido es la llamada a la función "setup":



- Para agregar botones basta seleccionar "Botón" en el menú, presionar "Añadir" y dar click en alguna parte blanca junto al área del mundo (el "mundo" es el rectángulo negro de la pestaña "Ejecutar").

Programas iniciales

- Luego de teclear este código, hay que hacer que se ejecute, por ejemplo con un botón cuyo único contenido es la llamada a la función "setup":



- Para agregar botones basta seleccionar "Botón" en el menú, presionar "Añadir" y dar click en alguna parte blanca junto al área del mundo (el "mundo" es el rectángulo negro de la pestaña "Ejecutar").

Píntalo de rojo

- Para que todos los parches realicen un mismo conjunto de acciones, utilizamos "ask": (después de teclear este código, agrega los botones necesarios)

```
to setup
  ca
  reset-ticks
end

to go
  ask  patches [
    set  pcolor  red
 ]
end
```

- ¿Qué pasa cuando presionas "go", pero antes haces que se ejecute "mas lento"?, ¿Siempre es igual?

Píntalo de rojo

- Para que todos los parches realicen un mismo conjunto de acciones, utilizamos "ask": (después de teclear este código, agrega los botones necesarios)

```
to setup
  ca
  reset-ticks
end

to go
  ask  patches [
    set  pcolor  red
 ]
end
```

- ¿Qué pasa cuando presionas "go", pero antes haces que se ejecute "mas lento"?, ¿Siempre es igual?

Tortugas borrachas

- Veamos la creación y movimiento de tortugas

```
to setup
  ca
  create-turtles 10 [   ; abrev. crt
    pen-down    ; abrev. pd
  ]
  reset-ticks
end

to go
  ask turtles [
    set heading random 360
    forward 1   ; abrev. fd
  ]
  tick
end
```

- Esto se puede leer: “crea 10 tortugas, cada una: baja su pluma”, “pídele a las tortugas: que apunten a un ángulo al azar en [0, 360); que avancen un paso”, al terminar con las tortugas: “avanza el reloj”

Tortugas borrachas

- Veamos la creación y movimiento de tortugas

```
to setup
  ca
  create-turtles 10 [   ; abrev. crt
    pen-down    ; abrev. pd
  ]
  reset-ticks
end

to go
  ask turtles [
    set heading random 360
    forward 1   ; abrev. fd
  ]
  tick
end
```

- Esto se puede leer: “crea 10 tortugas, cada una: baja su pluma”, “pídele a las tortugas: que apunten a un ángulo al azar en [0, 360); que avancen un paso”, al terminar con las tortugas: “avanza el reloj”

Tortugas borrachas

- Cada tortuga es un caminante al azar. Haz las siguiente prueba:
- Cambia el tamaño y forma del mundo: 200x200, parcela de 1 pixel
- Aumenta la cantidad de tortugas a 10000
- Quítale el pen-down y haz que todas sean blancas
- Modifica el botón “go” para que trabaje “continuamente”
- Cambia el menú de actualización a “manualmente (ticks)”
-

Tortugas borrachas

- Cada tortuga es un caminante al azar. Haz las siguiente prueba:
- Cambia el tamaño y forma del mundo: 200x200, parcela de 1 pixel
- Aumenta la cantidad de tortugas a 10000
- Quítale el pen-down y haz que todas sean blancas
- Modifica el botón “go” para que trabaje “continuamente”
- Cambia el menú de actualización a “manualmente (ticks)”
-

Tortugas borrachas

- Cada tortuga es un caminante al azar. Haz las siguiente prueba:
- Cambia el tamaño y forma del mundo: 200x200, parcela de 1 pixel
- Aumenta la cantidad de tortugas a 10000
- Quítale el pen-down y haz que todas sean blancas
- Modifica el botón “go” para que trabaje “continuamente”
- Cambia el menú de actualización a “manualmente (ticks)”
-

Tortugas borrachas

- Cada tortuga es un caminante al azar. Haz las siguiente prueba:
- Cambia el tamaño y forma del mundo: 200x200, parcela de 1 pixel
- Aumenta la cantidad de tortugas a 10000
- Quítale el pen-down y haz que todas sean blancas
- Modifica el botón “go” para que trabaje “continuamente”
- Cambia el menú de actualización a “manualmente (ticks)”
-

Tortugas borrachas

- Cada tortuga es un caminante al azar. Haz las siguiente prueba:
- Cambia el tamaño y forma del mundo: 200x200, parcela de 1 pixel
- Aumenta la cantidad de tortugas a 10000
- Quítale el pen-down y haz que todas sean blancas
- Modifica el botón “go” para que trabaje “continuamente”
- Cambia el menú de actualización a “manualmente (ticks)”
-

Tortugas borrachas

- Cada tortuga es un caminante al azar. Haz las siguiente prueba:
- Cambia el tamaño y forma del mundo: 200x200, parcela de 1 pixel
- Aumenta la cantidad de tortugas a 10000
- Quítale el pen-down y haz que todas sean blancas
- Modifica el botón “go” para que trabaje “continuamente”
- Cambia el menú de actualización a “manualmente (ticks)”

Tortugas borrachas

- Cada tortuga es un caminante al azar. Haz las siguiente prueba:
- Cambia el tamaño y forma del mundo: 200x200, parcela de 1 pixel
- Aumenta la cantidad de tortugas a 10000
- Quítale el pen-down y haz que todas sean blancas
- Modifica el botón “go” para que trabaje “continuamente”
- Cambia el menú de actualización a “manualmente (ticks)”
-