

Simulación de Eventos Discretos

Ricardo Kirkner

Tabla de contenidos

Parte I	3
Parte II.....	6
Bibliografía	19

Parte I

Identificación del sistema

Para este trabajo se ha elegido modelar y simular el comportamiento de un robot reconfigurable, basándonos en un trabajo realizado entre el *Department of Computer Science* del *Dartmouth College* y el *National Institute of Advanced Industrial Science & Technology* de Japón.

En dicho trabajo se estudia el comportamiento de robots modulares, en los cuales los módulos pueden desplazarse sobre la superficie de la estructura que forman. El objetivo del ese trabajo es encontrar algoritmos de planificación de reconfiguración que sean independientes de la arquitectura subyacente.

Para independizarse de la arquitectura, se definen tres tipos de movimientos abstractos que puede realizar un módulo sobre la superficie de la estructura. Estos movimientos son:

- movimiento rectilíneo en un plano
- transición convexa de un plano a otro
- transición cóncava de un plano a otro

En función de estas abstracciones se pueden razonar algoritmos descentralizados para la reconfiguración de cualquier sistema que sea capaz de realizar estos tipos de movimiento.

En su trabajo, los autores proponen que la habilidad de cambiar de forma puede producir movimientos al estilo de fluidos, en los cuales el robot puede adaptarse al terreno, permitiéndole trabajar adecuadamente en ambientes poco estructurados. Por esta razón sugieren el uso de autómatas celulares para resolver este problema.

Una suposición del trabajo es que los módulos pueden desplazarse de manera arbitraria sobre la superficie del robot, siempre y cuando no desconecten la estructura.

Modelización del sistema

En este trabajo, se desea modelar y simular un robot modular que sea capaz de desplazarse de izquierda a derecha en un espacio poco estructurado, evitando todos los obstáculos que se le presenten. El modelo representa un corte bidimensional de un espacio tridimensional. Por esta razón la visualización de la simulación permitirá ver al robot "de costado".

Debido a las simplificaciones y suposiciones tomadas del trabajo de referencia, para que el modelo pueda ser extendido a un espacio tridimensional completo, es necesario que el mismo sea homogéneo en la tercera dimensión, es decir todos los planos deben ser "parecidos" (no debe haber grandes diferencias entre la altura de los obstáculos entre un plano y otro, para una posición dada). Si se cumple esta restricción, para adaptar el modelo a tres dimensiones, solamente hará falta simular cada corte bidimensional del espacio mediante el modelo propuesto y luego concatenar todos los planos (ya que la suposición permite que los planos puedan ser tomados como independientes entre sí). En caso de no cumplirse la restricción, no es posible garantizar que el robot se mantenga conexo en las tres dimensiones.

En el modelo propuesto en este trabajo para la representación del robot cada módulo ocupa una celda del espacio celular. De acuerdo a esta suposición, se pueden definir 10 estados posibles para cada celda:

1. La celda se encuentra vacía

2. La celda se encuentra ocupada por un módulo del robot que está quieto.
3. La celda se encuentra ocupada por un obstáculo.
4. La celda se encuentra ocupada por un módulo del robot que se está moviendo hacia arriba.
5. La celda se encuentra ocupada por un módulo del robot que está yendo hacia arriba y a la derecha.
6. La celda se encuentra ocupada por un módulo del robot que se está moviendo hacia la derecha.
7. La celda se encuentra ocupada por un módulo del robot que está yendo hacia abajo y a la derecha.
8. La celda se encuentra ocupada por un módulo del robot que se está moviendo hacia abajo.
9. La celda se encuentra ocupada por un módulo del robot que se está moviendo hacia arriba y a la izquierda.
10. La celda se encuentra ocupada por un módulo del robot que se está moviendo hacia abajo y a la izquierda.

Estos estados se pueden clasificar de acuerdo a los movimientos abstractos propuestos en la introducción. Los estados 1, 2 y 3 no corresponden a ningún tipo de movimiento; los estados 4, 6 y 8 representan movimientos rectilíneos en el plano, y los estados 5, 7, 9 y 10 representan transiciones de un plano a otro.

Este modelo se corresponde con un comportamiento por etapas. Durante la primer etapa, cada módulo del robot decide que acción debe realizar. Durante la segunda etapa se realiza el movimiento estipulado en la primer etapa.

Importante: En este modelo se asume que cada módulo se puede desplazar únicamente por encima de otro módulo. Es decir, los módulos no pueden desplazarse directamente sobre los obstáculos.

Esta suposición se desprende del hecho de que no se desea modelar un robot que sea capaz de aferrarse a las paredes y trepar por ellas.

A continuación, se muestran gráficos de los movimientos posibles para un módulo del robot. En estos gráficos, las celdas de color rojo corresponden a módulos del robot quietos, las celdas azules corresponden a módulos en movimiento, celdas verdes representan obstáculos, y las celdas blancas corresponden a celdas que no están ocupadas. Para mayor claridad, se muestra una única celda activa.

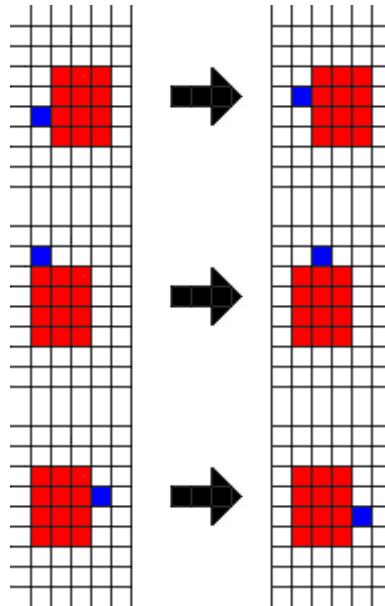


Figura 1. Movimiento en el plano

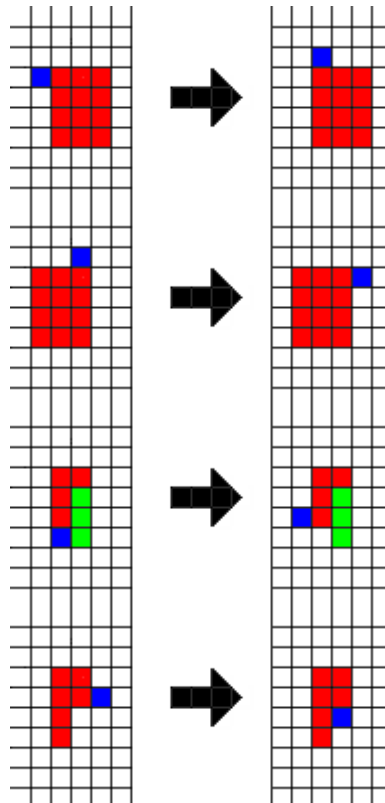


Figura 2. Transición convexa de un plano a otro

Se puede observar que de acuerdo a este tipo de movimientos, el robot se desplaza siguiendo un movimiento similar al de los fluidos, es decir, se ajusta al contorno de los obstáculos.

Parte II

Especificación del modelo Cell-DEVS

Para modelar este sistema se utilizó un modelo *acoplado Cell-DEVS cerrado con demoras de transporte*. La especificación formal del mismo se puede ver a continuación:

$$CC = \langle n, \{f, c\}, \eta, N, C, B, Z, \text{select} \rangle$$

donde

$$n = 2$$

$$f \in \mathbb{N}$$

$$c \in \mathbb{N}$$

$$\eta = 10$$

$$N = \{(-1,-1), (-1,0), (-1,1), (0,-1), (0,0), (0,1), (1,-2), (1,-1), (1,0), (1,1)\}$$

$$C = \{C_{ij} / i \in [1,f] \wedge j \in [1,c] \wedge C_{ij} \text{ es un modelo atómico con demora de transporte}\}$$

$$B = \emptyset$$

$$\text{select} = \{(-1,-1), (-1,0), (-1,1), (0,-1), (0,0), (0,1), (1,-2), (1,-1), (1,0), (1,1)\}$$

y Z está dada por:

$$P_{ij}^{X^1} \longleftarrow P_{i-1,j-1}^{Y^1}$$

$$P_{ij}^{X^2} \longleftarrow P_{i-1,j}^{Y^2}$$

$$P_{ij}^{X^3} \longleftarrow P_{i-1,j+1}^{Y^3}$$

$$P_{ij}^{X^4} \longleftarrow P_{i,j-1}^{Y^4}$$

$$P_{ij}^{X^5} \longleftarrow P_{ij}^{Y^5}$$

$$P_{ij}^{X^6} \longleftarrow P_{i,j+1}^{Y^6}$$

$$P_{ij}^{X^7} \longleftarrow P_{i+1,j-2}^{Y^7}$$

$$P_{ij}^{X^8} \longleftarrow P_{i+1,j-1}^{Y^8}$$

$$P_{ij}^{X^9} \longleftarrow P_{i+1,j}^{Y^9}$$

$$P_{ij}^{X^{10}} \longleftarrow P_{i+1,j+1}^{Y^{10}}$$

$$P_{ij}^{Y^1} \longrightarrow P_{i+1,j+1}^{X^1}$$

$$P_{ij}^{Y^2} \longrightarrow P_{i+1,j}^{X^2}$$

$$P_{ij}^{Y^3} \longrightarrow P_{i+1,j-1}^{X^3}$$

$$P_{ij}^{Y^4} \longrightarrow P_{i,j+1}^{X^4}$$

$$P_{ij}^{Y^5} \longrightarrow P_{ij}^{X^5}$$

$$P_{ij}^{Y^6} \longrightarrow P_{i,j-1}^{X^6}$$

$$P_{ij}^{Y^7} \longrightarrow P_{i-1,j+2}^{X^7}$$

$$P_{ij}^{Y^8} \longrightarrow P_{i-1,j+1}^{X^8}$$

$$P_{ij}^{Y^9} \longrightarrow P_{i-1,j}^{X^9}$$

$$P_{ij}^{Y^{10}} \longrightarrow P_{i-1,j-1}^{X^{10}}$$

El vecindario elegido para este modelo es un vecindario de Moore ligeramente modificado. Originalmente, se comenzó trabajando con un vecindario de Moore, pero

la información que proveía este vecindario no alcanzaba, por lo que se tuvo que agrandar el mismo.

El vecindario que finalmente fue utilizado es

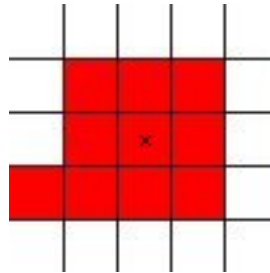


Figura 3. Vecindario

La celda agregada resulta necesaria para poder determinar el momento en que el módulo del robot debe retroceder, para ocupar una posición que le permita al robot "llenar" las celdas debajo de si. Este movimiento corresponde a los módulos que se encuentren en el estado 10 anteriormente enunciado.

Este vecindario provee la suficiente cantidad de información para permitir al robot desplazarse de la manera pretendida. Nótese que si se utilizara un vecindario más pequeño, se perdería información necesaria para determinar la correcta evolución del modelo.

Para especificar el comportamiento de cada célula, se utiliza la especificación formal del modelo de célula con demora de transporte. Dicha especificación se presenta a continuación.

$$TDC = \langle X, Y, S, N, \text{delay}, \delta_{\text{int}}, \delta_{\text{ext}}, \tau, \lambda, D \rangle$$

donde

$$X = \emptyset$$

$$Y = \emptyset$$

$$S = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$N = \{(-1,-1), (-1,0), (-1,1), (0,-1), (0,0), (0,1), (1,-2), (1,-1), (1,0), (1,1)\}$$

$$\text{delay} = \text{transport}$$

$$d = 100\text{ms}$$

y

$\tau: N \rightarrow S$ está dada por:

Tabla 1. Movimiento hacia arriba

Estado	Condición	Comentario
S = 3	$(-1,-1)=0$ y $(-1,0)=0$ y $(-1,1)=1$ y $(0,-1)=0$ y $(0,0)=1$ y $(0,1)=1$	Activamos el movimiento
S = 1	$(0,-1)=0$ y $(0,0)=0$ y $(0,1)=1$ y $(1,-1)=0$ y $(1,0)=3$ y $(1,1)=1$	Realizamos el movimiento
S = 0	$(-1,-1)=0$ y $(-1,0)=0$ y $(-1,1)=1$ y $(0,-1)=0$ y $(0,0)=3$ y $(0,1)=1$	Desactivamos el movimiento

Estado	Condición	Comentario
S = 1	$(-1,-1)=0$ y $(-1,0)=0$ y $(-1,1)>2$ y $(0,-1)=0$ y $(0,0)=3$ y $(0,1)=1$	Cancelamos el movimiento

Tabla 2. Movimiento hacia arriba y a la derecha

Estado	Condición	Comentario
S = 4	$(-1,-1)=(0 \text{ ó } 2)$ y $(-1,0)=0$ y $(-1,1)=0$ y $(0,-1)=(0 \text{ ó } 2)$ y $(0,0)=1$ y $(0,1)=1$ y $(1,1)!=2$	Activamos el movimiento
S = 4	$(-1,-1)=(0 \text{ ó } 2)$ y $(-1,0)=0$ y $(-1,1)=0$ y $(0,-1)=(0 \text{ ó } 2)$ y $(0,0)=1$ y $(0,1)=1$ y $(1,-1)=(0 \text{ ó } 2)$ y $(1,0)=(0 \text{ ó } 2)$ y $(1,1)=(0 \text{ ó } 2)$	Activamos el movimiento
S = 1	$(0,-1)=0$ y $(0,0)=0$ y $(1,-1)=4$ y $(1,0)=1$	Realizamos el movimiento
S = 0	$(-1,-1)=(0 \text{ ó } 2)$ y $(-1,0)=0$ y $(-1,1)=0$ y $(0,-1)=(0 \text{ ó } 2)$ y $(0,0)=4$ y $(0,1)=1$	Desactivamos el movimiento
S = 1	$(-1,-1)=(0 \text{ ó } 2)$ y $(-1,0)=0$ y $(-1,1)=0$ y $(0,-1)=(0 \text{ ó } 2)$ y $(0,0)=4$	Cancelamos el movimiento

Tabla 3. Movimiento hacia la derecha

Estado	Condición	Comentario
S = 5	$(-1,-1)=(0 \text{ ó } 2)$ y $(-1,0)=0$ y $(-1,1)=0$ y $(0,0)=1$ y $(0,1)=0$ y $(1,-1)!=2$ y $(1,0)=1$ y $(1,1)=1$	Activamos el movimiento
S = 5	$(-1,-1)=(0 \text{ ó } 2)$ y $(-1,0)=0$ y $(-1,1)=0$ y $(0,-1)=(0 \text{ ó } 2)$ y $(0,0)=1$ y $(0,1)=0$ y $(1,0)=1$ y $(1,1)=1$	Activamos el movimiento
S = 1	$(0,-1)=5$ y $(0,0)=0$ y $(1,-1)=1$ y $(1,0)=1$	Realizamos el movimiento
S = 0	$(-1,-1)=(0 \text{ ó } 2)$ y $(-1,0)=0$ y $(-1,1)=0$ y $(0,0)=5$ y $(0,1)=0$ y $(1,0)=1$ y $(1,1)=1$	Desactivamos el movimiento
S = 1	$(-1,-1)=0$ y $(-1,0)=0$ y $(-1,1)=0$ y $(0,0)=5$ y $(0,1)=0$ y $(1,0)=1$	Cancelamos el movimiento

Tabla 4. Movimiento hacia abajo y a la derecha

Estado	Condición	Comentario
S = 6	$(-1,-1)=0$ y $(-1,0)=0$ y $(-1,1)=0$ y $(0,0)=1$ y $(0,1)=0$ y $(1,-1)!=2$ y $(1,0)=1$ y $(1,1)=0$	Activamos el movimiento
S = 1	$(-1,-1)=6$ y $(-1,0)=0$ y $(0,-1)=1$ y $(0,0)=0$	Realizamos el movimiento
S = 0	$(-1,-1)=0$ y $(-1,0)=0$ y $(-1,1)=0$ y $(0,0)=6$ y $(0,1)=0$ y $(1,0)=1$ y $(1,1)=0$	Desactivamos el movimiento

Tabla 5. Movimiento hacia abajo

Estado	Condición	Comentario
S = 7	$(0,-1)=1$ y $(0,0)=1$ y $(0,1)=(0 \text{ ó } 2)$ y $(1,-1)=1$ y $(1,0)=0$ y $(1,1)=(0 \text{ ó } 2)$	Activamos el movimiento

Estado	Condición	Comentario
S = 1	$(-1,0)=7$ y $(0,-1)=1$ y $(0,0)=0$	Realizamos el movimiento
S = 0	$(0,0)=7$ y $(0,1)=(0 \text{ ó } 2)$ y $(1,-1)=1$ y $(1,0)=0$ y $(1,1)=(0 \text{ ó } 2)$	Desactivamos el movimiento

Tabla 6. Movimiento hacia arriba y a la izquierda

Estado	Condición	Comentario
S = 8	$(-1,-1)=0$ y $(-1,0)=1$ y $(-1,1)=(1 \text{ ó } 2)$ y $(0,-1)=0$ y $(0,0)=1$ y $(0,1)=(0 \text{ ó } 2)$ y $(1,-1)=(0 \text{ ó } 2)$ y $(1,0)=(0 \text{ ó } 2)$ y $(1,1)=(0 \text{ ó } 2)$	Activamos el movimiento
S = 1	$(0,-1)=0$ y $(0,0)=0$ y $(0,1)=1$ y $(1,-1)=0$ y $(1,0)=0$ y $(1,1)=8$	Realizamos el movimiento
S = 0	$(-1,-1)=0$ y $(-1,0)=1$ y $(-1,1)=(1 \text{ ó } 2)$ y $(0,-1)=0$ y $(0,0)=8$ y $(0,1)=(0 \text{ ó } 2)$ y $(1,-1)=(0 \text{ ó } 2)$ y $(1,0)=(0 \text{ ó } 2)$ y $(1,1)=(0 \text{ ó } 2)$	Desactivamos el movimiento

Tabla 7. Movimiento hacia abajo y a la izquierda

Estado	Condición	Comentario
S = 9	$(0,-1)=1$ y $(0,0)=1$ y $(0,1)=0$ y $(1,-2)=(1 \text{ ó } 2)$ y $(1,-1)=0$ y $(1,0)=0$ y $(1,1)=0$	Activamos el movimiento
S = 1	$(-1,0)=1$ y $(-1,1)=9$ y $(0,-1)=(1 \text{ ó } 2)$ y $(0,0)=0$ y $(0,1)=0$	Realizamos el movimiento
S = 0	$(0,-1)=1$ y $(0,0)=9$ y $(0,1)=0$ y $(1,-2)=(1 \text{ ó } 2)$ y $(1,-1)=0$ y $(1,0)=0$ y $(1,1)=0$	Desactivamos el movimiento

Tabla 8. Regla default

Estado	Condición
S = $\{(0,0)\}$	true

Esta última regla ocasiona que todas las celdas que no cumplen con ninguna otra regla, permanezcan en su estado original.

Resultados de la simulación

A continuación se muestran ejemplos de una corrida de la simulación en un entorno poco estructurado.

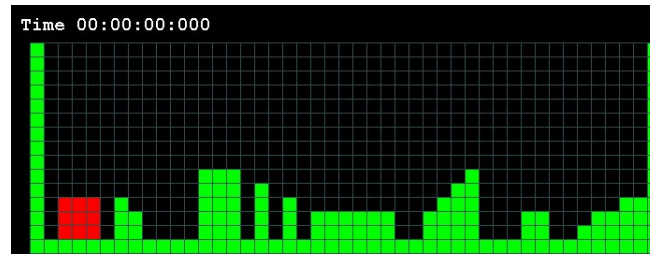


Figura 4. Estado inicial

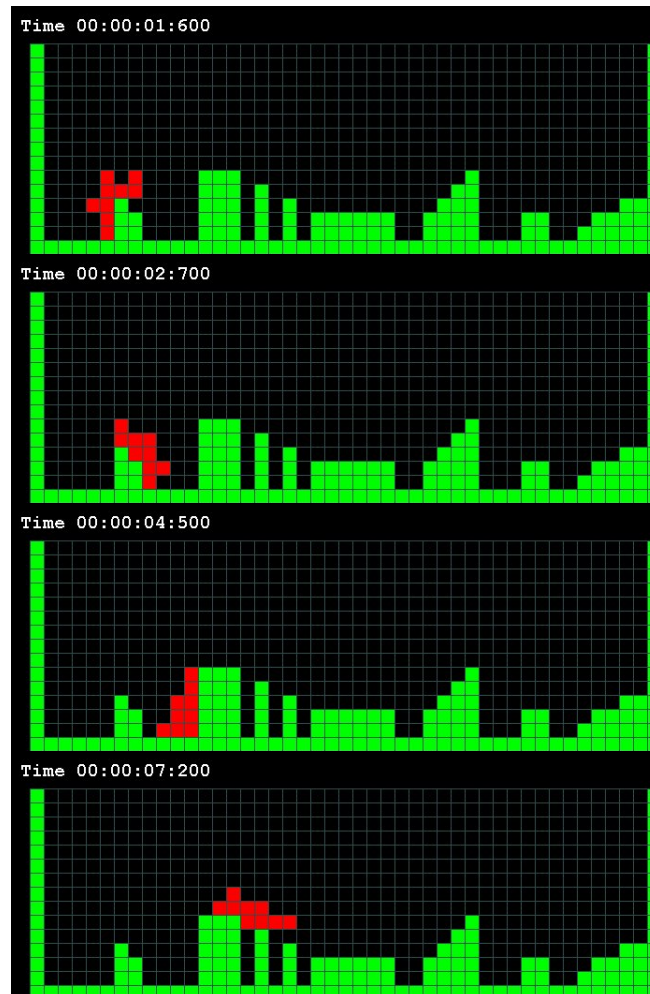


Figura 5. Estados intermedios

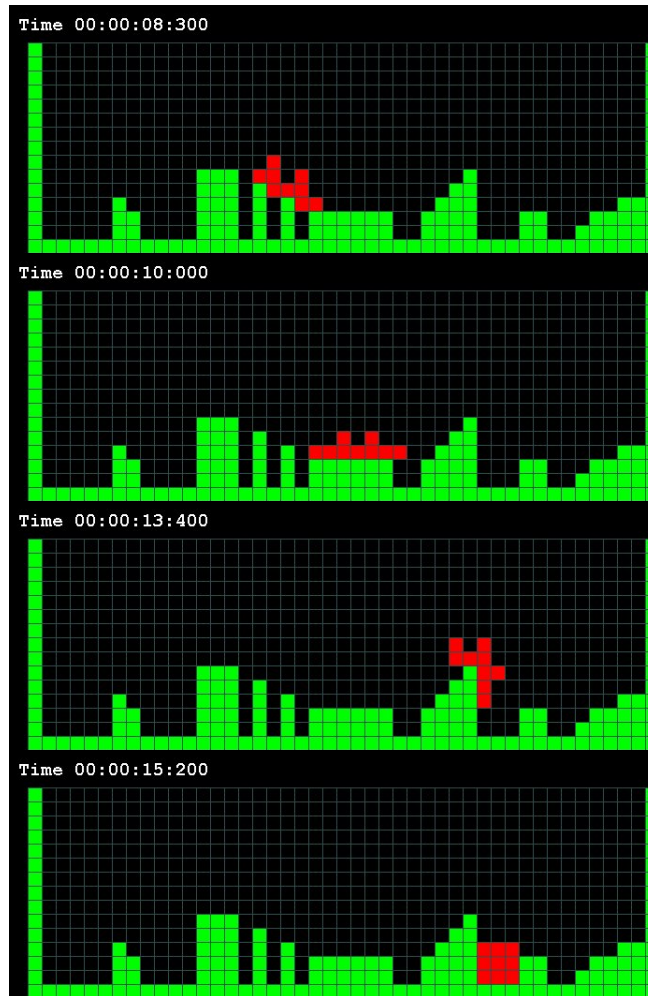


Figura 6. Estados intermedios

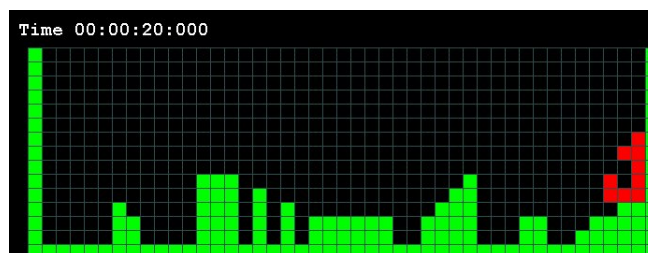


Figura 7. Estado final

Como se puede observar en estos ejemplos, el robot es capaz de sobrepasar los obstáculos, manteniéndose siempre conectado.

Otros ejemplos del comportamiento, en distintos escenarios se proveen a continuación.

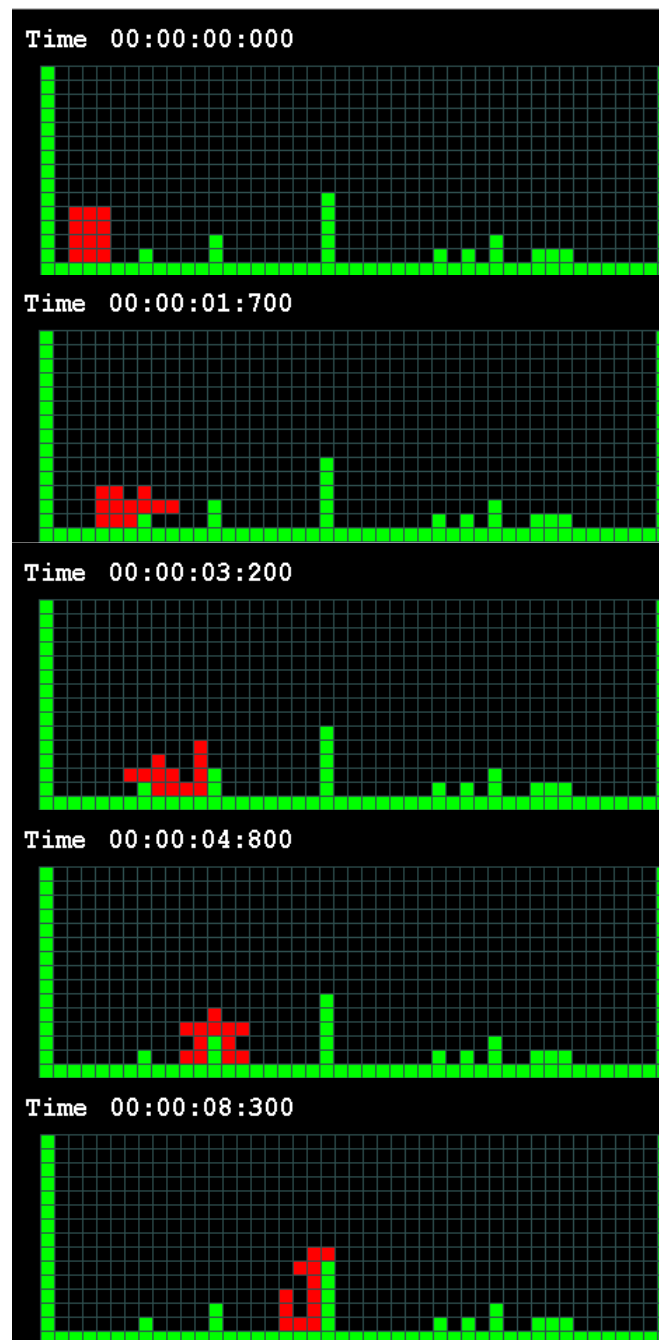


Figura 8. Pocos obstáculos

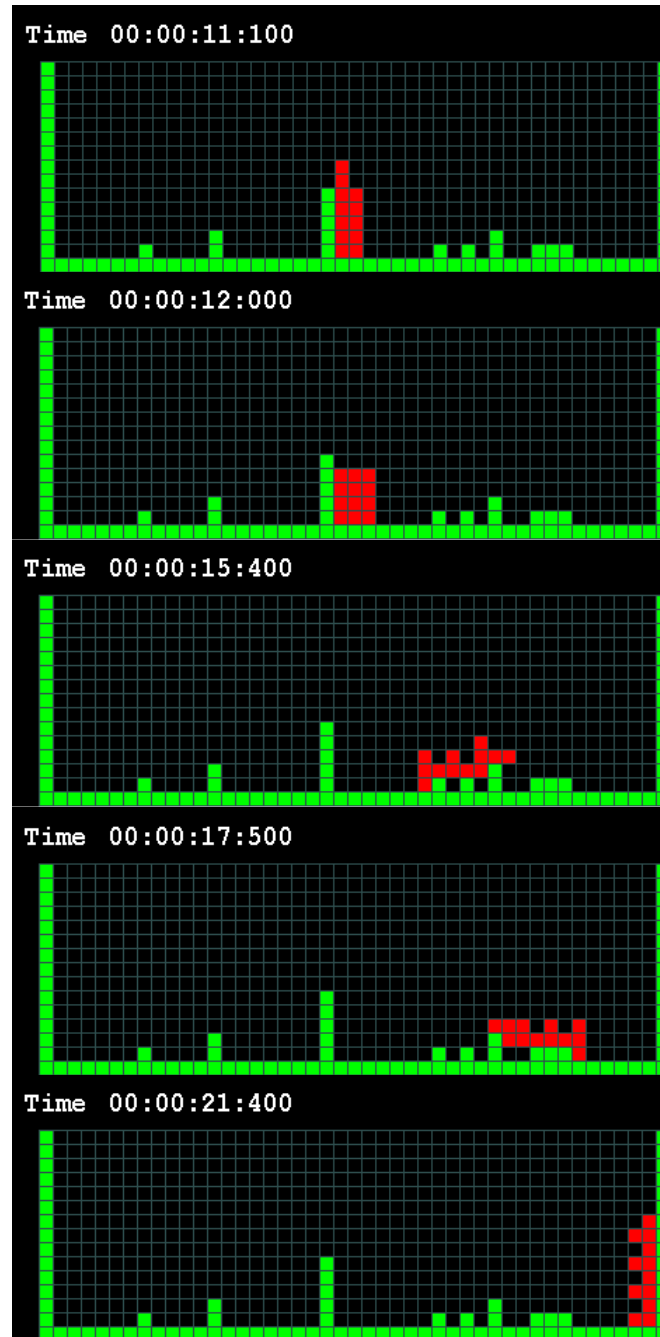


Figura 9. Pocos obstáculos (cont.)

Como se puede observar en las figuras 6 y 10, luego de superar un obstáculo de altura considerable (recordar que el robot es capaz de superar obstáculos que superen en altura al robot por un módulo como máximo), si el robot tiene espacio suficiente, se vuelve a reconfigurar de forma de obtener una estructura rectangular.

Otra observación que se puede extraer de las figuras 5, 6 y 10, es que cuando los obstáculos están separados por espacios pequeños (de una o dos celdas de ancho), el robot no desciende por estos huecos, sino que permanece "colgando" en el aire, hasta lograr apoyarse sobre el próximo obstáculo. Este comportamiento es imprescindible, ya que si el robot descendiera por estos "huecos", no sería capaz de volver a salir de

ellos (podría interpretarse que el robot es capaz de "ver" que hay un apoyo cerca, y "estirarse" para alcanzarlo).

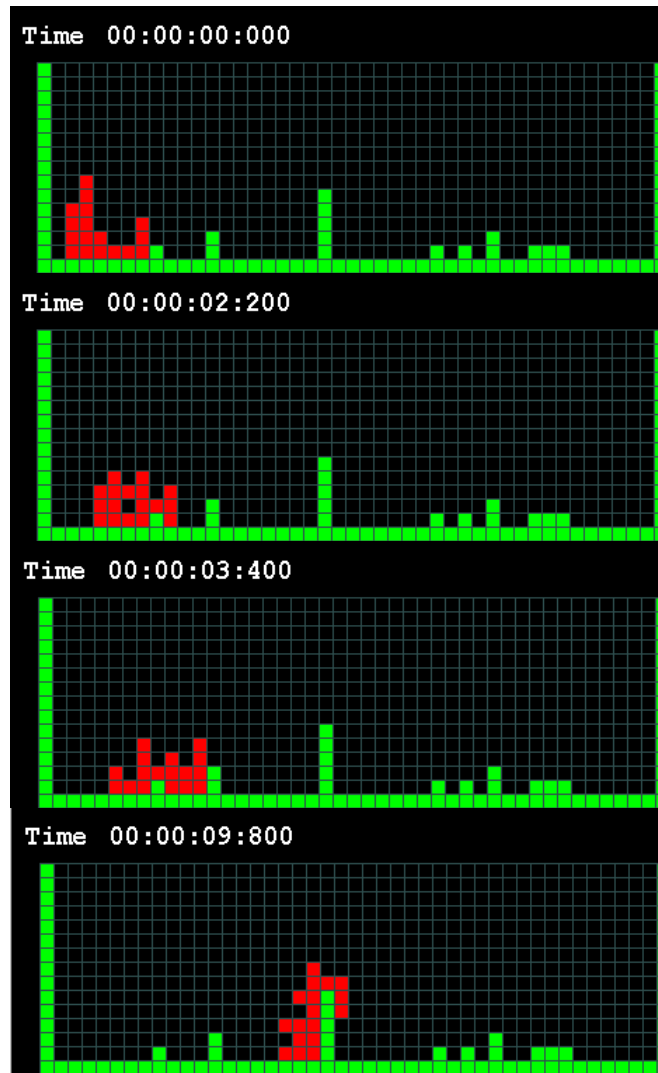


Figura 10. Robot con forma arbitraria

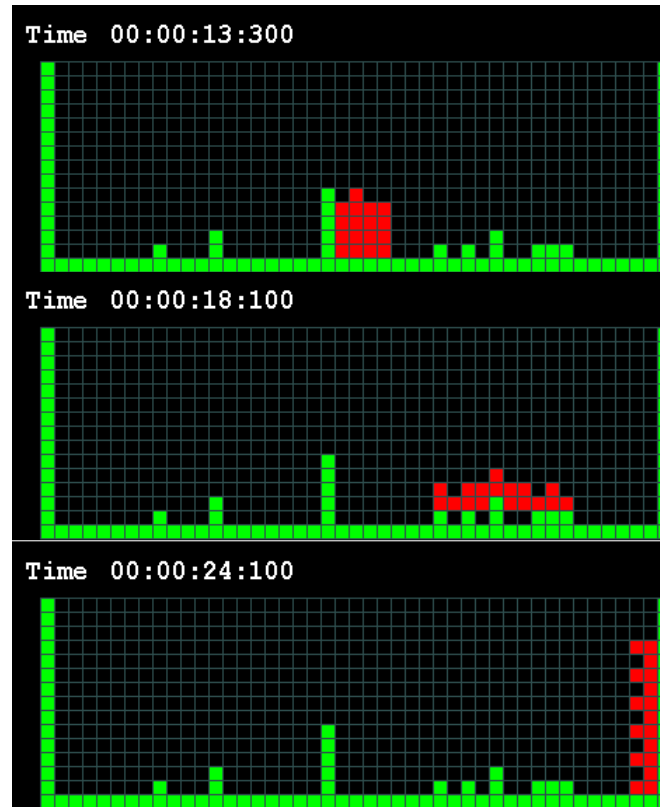


Figura 11. Robot con forma arbitraria (cont.)

En este ejemplo se puede observar que no es necesario que la posición inicial del robot posea una estructura rectangular, sino que mientras que sea conexa, puede tomar una forma arbitraria.

Otra observación que se puede realizar es que es posible que se formen "agujeros" en el cuerpo del robot, sin que por ello el robot se desconecte. Además, el hecho de no ser un bloque compacto no le impide avanzar, sino que logra reconfigurarse para poder seguir moviéndose.

Nuevamente se puede observar que luego de superar un obstáculo de altura considerable, el robot se reconfigura de manera rectangular.

Una limitación de este conjunto de reglas es que, debido al vecindario utilizado, hay situaciones en las que el robot "pierde la noción de gravedad" y se despega del suelo antes de haberse "aferrado" a algún obstáculo. Esto ocurre cuando los obstáculos superan en altura al robot por más de un módulo.



Figura 12. Ejemplo de "ignorancia" de la gravedad

A continuación se puede observar que las reglas que se utilizan para generar el comportamiento son altamente sensibles a variaciones en el entorno que no fueron tenidas en cuenta. En este caso, se introduce una ligera modificación en un obstáculo, para la cuál no fue prevista ninguna regla. Esto produce que el robot no sea capaz de superar dicho obstáculo. Luego de alcanzar el estado que se muestra en el último gráfico, la simulación se detiene, ya que el modelo no tiene reglas que le permitan superar este tipo de situaciones.



Figura 13. Variaciones ligeras en la forma de los obstáculos

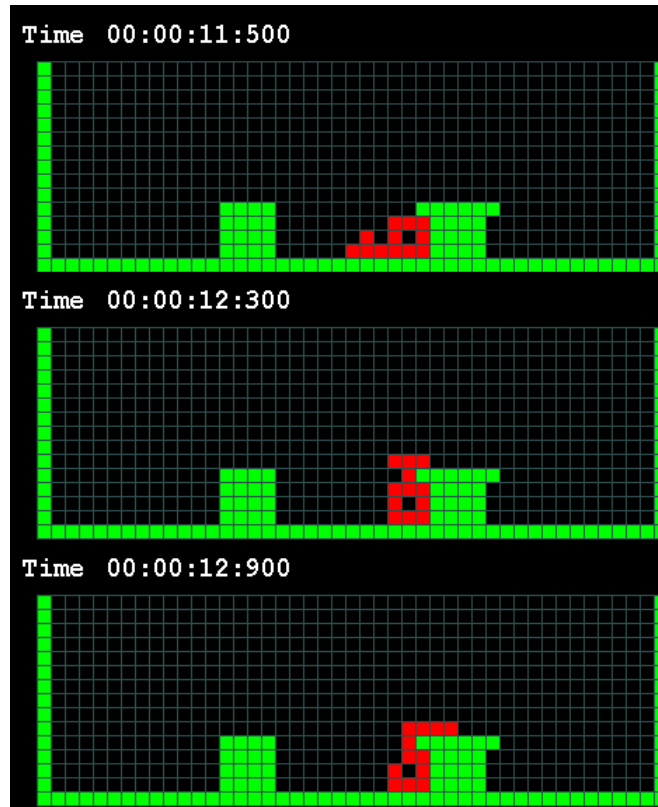


Figura 14. Variaciones ligeras en la forma de los obstáculos (cont.)

Conclusiones

Durante el trabajo realizado, una de las primeras cosas que pudo ser observada es que la construcción de reglas para un autómata celular es una labor cíclica e incremental, y que a medida que se obtiene mayor experiencia en el modelado mediante autómatas celulares, la construcción de estas reglas se va volviendo cada vez más intuitiva.

Sin embargo, para obtener un comportamiento completo, es necesario analizar todas las posibles situaciones, para poder tener un conjunto de reglas que cubra todas las posibilidades en las que se pueda encontrar el sistema. Para sistemas complejos, encontrar este conjunto puede ser una tarea bastante difícil.

Por otro lado, en ciertas ocasiones, el modelado mediante autómatas celulares resulta más complejo que el sistema real. Por ejemplo, en el caso que nos concierne, es necesario construir reglas para "prender" y "apagar" celdas, para simular el movimiento, teniendo en cuenta muchas situaciones posibles del entorno, cuando en el robot real, estas situaciones están restringidas directamente por la realidad (o sea, en el autómata celular hay que tener en cuenta que las celdas no empiecen a "reproducirse", cuando en el sistema real, no hace falta preocuparse de ello, ya que no podría suceder nunca).

Pudimos observar que el formalismo *Cell-DEVS* permite extender la semántica de los autómatas celulares de una manera muy sencilla. Por ejemplo, cambiando simplemente el tiempo de actualización de la celda para algunas reglas, es posible obtener un movimiento en una fase, utilizando reglas que originalmente fueron escritas para realizar este movimiento en dos fases.

Este tipo de variaciones, permite modelar con mayor grado de detalle sistemas que no se podrían modelar tan fácilmente utilizando autómatas celulares.

Bibliografía

[DecentralizedControl] Zack Butler, Keith Kotay, Daniela Rus, y Kohji Tomita.

