Simulación de tránsito urbano con modelos Cell-DEVS

Alejandra Davidson
ad3n@dc.uba.ar

Gabriel A. Wainer
gabrielw@dc.uba.ar

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Pabellón I - Ciudad Universitaria
Buenos Aires (1428) - ARGENTINA

Abstract

En los últimos años el formalismo de autómatas celulares ha sido utilizado para simulación de tráfico de vehículos. Los A.C. imponen severas restricciones de precisión y performance. Este trabajo plantea un modelo para calles de 1 ó 2 carriles de igual dirección que incluye el comportamiento de los autos al llegar a las intersecciones, pero utilizando el paradigma de especificación Cell-DEVS. Éste permite la descripción de espacios celulares a través de su definición como modelos de eventos discretos, mejorando los problemas de los A.C. Mediante esta variación también se obtienen reglas de comportamiento más simples que facilitan su extensión para incorporar nuevos aspectos reales del problema al modelo. Además de lograr mayor precisión y mejor uso de los recursos durante la simulación.

1. Introducción

Cell_DEVS es un paradigma de especificación de modelos simulables, propuesto en [W98,WG98], que se utiliza para describir espacios de celdas ubicadas geométricamente. Cada celda se puede definir como:

$$C_{ij} = < I, X, S, Y, N, \delta_{int}, \delta_{ext}, \text{delay}, d, \tau, \lambda, D >$$

Cada celda presenta su estado (representado por un valor del conjunto **S**) que evoluciona en el tiempo de acuerdo a un conjunto de *reglas locales* de transición (representadas por la función τ). Para que puedan ser aplicadas en cada celda no se necesita conocer el estado de todo el espacio, sino el del sector que constituye su vecindad (representado por **N**). Estas celdas se acoplan (mediante la interfaz **I**) para formar un espacio Cell_DEVS completo a través de la relación de vecindad. La función de duración de vida (**D**) se usa para controlar el tiempo de duración del estado de una celda. A través de las funciones de transición interna (δ_{int}), de transición externa (δ_{ext}) y de salida (λ) se modela la demora (**d**) asociada a la celda. Ésta puede ser de 2 tipos:

a) de Transporte: permite que el cambio de estado ante un evento externo sea demorado durante el tiempo correspondiente a la demora.

b) Inerciales: permite que el cambio de estado ante un evento externo se haga efectivo sólo en el caso que se mantenga ese valor por el tiempo correspondiente a la demora; de no ser así, se descarta.

Los modelos atómicos pueden integrar un modelo acoplado Cell-DEVS, definido como:

$$GCC = < Xlist, Ylist, I, X, Y, n, \{t_1,...,t_n\}, \eta, N, C, B, Z, select >$$

Este modelo representa un espacio de dimensión **n**, donde $\{t_1,...,t_n\}$ es la cantidad de celdas en cada una de las dimensiones. Está formado por un conjunto de celdas atómicas (**C**), como las definidas anteriormente, conectadas entre sí mediante la definición de vecindad. Donde, **N** es el conjunto de celdas vecinas, η indica la cantidad de vecinas y la función **Z** es quien establece el acoplamiento entre ellas. También se define una interfaz del modelo (**I**) para que se pueda acoplar con otros, indicando además la lista de celdas de acoplamiento de salida (**Ylist**); la lista de acoplamiento de entrada (**Xlist**); el conjunto de eventos externos de entrada (*X*) y el de salida (*Y*). Luego, el conjunto **B** define el borde del espacio de celdas; y si es vacío, toda celda en el espacio tendrá el mismo comportamiento: las celdas en un borde se conectan con las que están el borde opuesto. Caso contrario, las celdas tendrán un comportamiento diferente a las otras del modelo. Por último, **select** es la función de selección ante eventos simultáneos.

El paradigma Cell-DEVS plantea una mejora para los autómatas celulares, representándolos mediante modelos de eventos discretos; logrando mayor precisión y mejor utilización de los recursos. Como la simulación de tráfico de vehículos es un área importante de aplicación de los autómatas celulares, aquí se presenta un modelo de calles y cruces basado en el formalismo Cell-DEVS. Cuya definición completa se encuentra en [DW99] e incluye calles de cualquier cantidad de carriles, con señales de tránsito, barreras de trenes, semáforos, circulación de camiones y otros elementos que condicionan el flujo de vehículos. En la sección 2.1 se especifican calles de 1 y 2 carriles con circulación de autos; definiendo separadamente un modelo genérico para sus intersecciones o cruces en la sección 2.2., siguiendo el modelo planteado en [CQL95], [CLQ96] y [CDL97]. Y se establece la interfaz entre ambos que permita su acoplamiento.

2. Definición de calles en base al formalismo Cell-DEVS

Para representar el tráfico urbano se debe considerar, por un lado, el comportamiento de los vehículos a lo largo de calles sin cruces. Por otro lado, se debe analizar cómo actúan en las esquinas, para luego definir una interfaz entre ambos que permita combinarlos y modelar la circulación de autos en calles con cruces. Así, se define un modelo para calles en la sección 2.1 y otro para cruces en la sección 2.2, de forma tal que se puedan acoplar a través de las celdas borde.

2.1. Calles

Una calle se define como una secuencia de *tramos*, donde cada uno representa un sector sin cruces y de mano única. El movimiento de los vehículos consiste en avanzar hacia adelante, ya sea recto o en diagonal, siempre y cuando tengan lugar. Si se trata de tramos de carril único (sección 2.1.1) el auto sólo se puede mover en línea recta, y si son 2 carriles (sección 2.1.2) además puede avanzar en una de las diagonales. Para obtener un comportamiento determinístico, en el segundo caso, se establece que un auto intente primero moverse sobre su propio carril y sólo si esto no es posible probará con el otro.

2.1.1. Calles de carril único

Un tramo de k celdas, con velocidad máxima de circulación de max (Km/h) y de carril único se define como un modelo Cell_DEVS de una dimensión con demora de transporte, cuya estructura se presenta en la Figura 1.

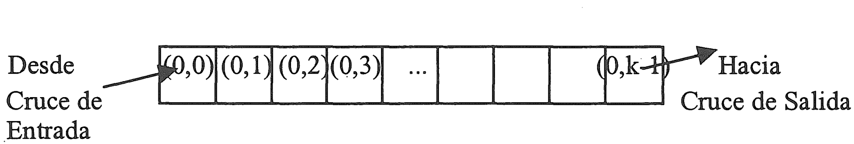

Figura 1 – Tramo de carril único

Cada celda en este espacio se define como:

$$C_{0j} = < I, X, S, Y, N, \delta_{int}, \delta_{ext}, \text{delay}, d, \tau, \lambda, D >$$

con

$I = <\eta, P^x, P^y>$, donde

$\eta = 3$;
$P^x = \{ (X_1, \text{binario}), (X_2, \text{binario}), (X_3, \text{binario}) \}$;
$P^y = \{ (Y_1, \text{binario}), (Y_2, \text{binario}), (Y_3, \text{binario}) \}$.

$X = Y = \{0, 1\}$;

S:

$$s = \begin{cases} 1 \text{ si hay un vehículo en la celda;} \\ 0 \text{ sino.} \end{cases}$$

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

(0,-1)	(0,0)	(0,1)

Figura 2 - Vecindario de la celda origen

delay = transport;

Las demoras de transporte se utilizan para modelar el tiempo que tarda un vehículo en salir de una celda e ingresar a la siguiente, y depende de su velocidad. Este valor (**d**) es generado mediante una función aleatoria de distribución probabilística normal, que recibe como parámetro el valor máximo que puede alcanzar la función (max), y que retorna un valor natural que representa la nueva velocidad del vehículo elegida en forma aleatoria de acuerdo a la distribución descripta. Esto permite modelar distintas velocidades para cada vehículo en distintos momentos.

λ, δ_{int} y δ_{ext} se comportan como las funciones definidas por el formalismo Cell-DEVS para demoras de transporte.

τ: S x N $\rightarrow$ S se define de la siguiente manera,

Nuevo estado	Estado del vecindario	Nombre de la regla
1	(0,-1) = 1 and (0,0) = 0	Llega_Desde_Atrás
0	(0,0) = 1 and (0,1) = 0	Sale_Hacia_Adelante
(0,0)	t /*en otro caso conserva estado */	Default

Donde, (i,j) $\in$ N y dentro de la regla representa el estado actual del vecino (i,j).

En un tramo de carril único sólo se permite que los vehículos avancen hacia la posición de adelante, siempre y cuando la misma esté vacía. Este movimiento (especificado con la función τ) queda definido por 3 reglas. La regla Llega_Desde_Atrás representa el movimiento recto del vehículo desde la celda de atrás hacia la origen, siempre y cuando ésta se encuentre vacía. La regla Sale_Hacia_Adelante representa el movimiento recto del vehículo que se encuentra en la celda origen hacia la de adelante. Y por último, la regla Default considera cualquier otro caso donde el estado de la celda no cambia.

El modelo acoplado correspondiente al tramo se define como:

$$TC1(k, max) = < Xlist, Ylist, I, X, Y, n, \{t_1,...,t_n\}, \eta, N, C, B, Z, select >$$

TC1(k, max) significa Tramo de 1 Carril de longitud k (celdas) con velocidad máxima de circulación de max (Km/h).

$n = 1$

$t_1 = k$

$\eta = 3$

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

La función **Z** se construye siguiendo la definición dada por el formalismo Cell_DEVS, instanciada con el vecindario de este espacio.

select = { (0,1), (0,0), (0,-1) }

$X = Y = \{ 0, 1 \}$

Ylist = **Xlist** = { (0,0), (0,k-1) }

I = $<P^x, P^y>$

$P^x = \{<X_{\eta+1}(0,0), \text{binario}>, <X_{\eta+1}(0,k-1), \text{binario}>\}$
$P^y = \{<Y_{\eta+1}(0,0), \text{binario}>, <Y_{\eta+1}(0,k-1), \text{binario}>\}$

Los ports se denominarán de la siguiente forma:

Port	Nombre
$X_{\eta+1}(0,0)$	x-c-hayauto
$X_{\eta+1}(0,k-1)$	x-c-haylugar
$Y_{\eta+1}(0,0)$	y-c-haylugar
$Y_{\eta+1}(0,k-1)$	y-c-hayauto

B = { (0,0), (0,k-1) }

La interface externa del modelo la conforman las celdas (0,0) y (0,k-1), pues cada una debe intercambiar información con los cruces correspondientes, para establecer su nuevo estado; y es por ello que forman parte del conjunto borde. Para la celda (0,0) se define el siguiente vecindario y comportamiento:

$\eta = 2$

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

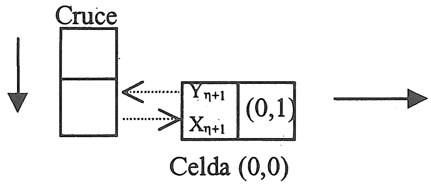

Figura 3- Vecindario y acoplamiento de la celda (0,0)

La función τ para esta celda se define igual que para el resto del espacio, a excepción de la regla Llega_Desde_Atrás que es cambiada por Llega_Desde_Cruce, que se especifica como:

Nuevo Estado	Estado del vecindario	Nombre de la regla
1	(portvalue(x-c-hayauto) = 1 and (0,0) = 0)	Llega_Desde_Cruce

La celda (0,0) se encuentra acoplada a la celda del cruce desde la que recibe los vehículos, como se puede ver en la Figura 3. Siempre que el port x-c-hayauto ($X_{\eta+1}$) esté en 1 representa que existe un vehículo que saldrá del cruce hacia la celda (0,0). Por lo tanto, la regla Llega_Desde_Cruce representa el movimiento de avance de un vehículo que se encuentra en el cruce (portvalue(x-c-hayauto) = 1) hacia la celda (0,0). Ésta envía su estado hacia el cruce a través del port y-c-haylugar, es decir 0 si está vacía y 1 en otro caso. Los demás parámetros del modelo para esta celda no cambian.

Para la celda (0,k-1) también se define un vecindario y comportamiento diferente al resto. Aquí:

$\eta = 2$

N = { (0,-1), (0,0) }

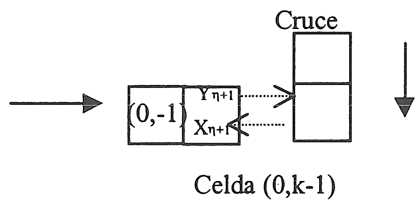

Figura 4- Vecindario y acoplamiento de la celda (0,k-1)

La función τ para esta celda se define igual que para el resto del espacio, a excepción de la regla Sale_Hacia_Adelante que es cambiada por Sale_Hacia_Cruce, que se especifica como:

Nuevo Estado	Estado del vecindario	Nuevo Estado de los Ports	Delay	Nombre de la regla
0	((0,0) = 1 and portvalue(x-c-haylugar) = 0)	send(1, y-c-hayauto)	inercial	Sale_Hacia_Cruce

Donde, la función *send* permite definir el nuevo estado de los ports externos seleccionados. Es decir, send (v, P_{j1}^{y}, P_{j2}^{y}, ... , P_{jn}^{y}) representa que el nuevo estado de los ports {P_{j1}^{y}, P_{j2}^{y}, ... , P_{jn}^{y}} es v, donde v $\in$ {0, 1}. Además, cuando no se define send en forma explícita se asume que *todos* los ports de salida son actualizados con el nuevo valor del estado de la celda.

La celda (0,k-1) se encuentra acoplada con 1 celda del cruce, como se puede ver en la Figura 4. El port x-c-haylugar informa del estado de la celda de ingreso y de la anterior, y así la regla Sale_Hacia_Cruce representa el avance del vehículo desde la celda (0,k-1) hacia la de adelante (dentro del cruce), siempre y cuando ésta y su anterior se encuentren vacías (portvalue(x-c-haylugar) = 0), modelando que los autos dentro del cruce tienen prioridad sobre los que quieren ingresar. Esta regla se define con demora inercial porque representa el comportamiento de los vehículos que llegan a las esquinas; avanzando sólo si durante un cierto tiempo (demora inercial) tienen el camino libre, caso contrario esperan hasta tener lugar.

De esta forma los autos que circulan dentro del cruce tienen prioridad de avanzar sobre aquellos que desean ingresar y estos últimos no lo hacen si no tienen el camino libre por el tiempo que les lleva realizar la maniobra. La celda (0,k-1) actualiza el port de salida con 1 si existe un auto en la celda que está habilitado (las celdas del cruce permanecieron vacías durante la demora inercial correspondiente) para avanzar al cruce, caso contrario su valor es 0. Los demás parámetros del modelo para esta celda no cambian.

2.1.2. Calles de dos carriles de igual dirección

El modelo para calles de 2 carriles es una extensión del planteado para tramos de carril único, permitiendo que los vehículos además de moverse a la celda de adelante, lo puedan hacer hacia la que tienen adelante en diagonal. Un tramo de k celdas, con velocidad máxima de circulación de max (Km/h) y de 2 carriles se define como un modelo Cell_DEVS de dos dimensiones con demora de transporte, cuya estructura se presenta en la Figura 5.

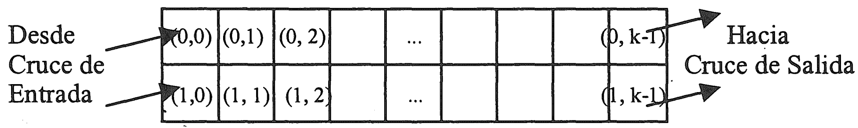

Figura 5 – Tramo de 2 carriles

Las celdas de cada fila de este espacio tienen comportamiento distinto, determinado por la conformación de su vecindario, y serán especificadas por separado. Es decir, los autos de las celdas de la primera fila sólo se pueden mover hacia adelante y en diagonal derecha; mientras que los de la otra lo pueden hacer hacia adelante y en diagonal izquierda.

Las celdas de la primera fila (carril 0) del espacio se definen igual que las de carril único a excepción de su interface (I), su vecindario (N) y las reglas de comportamiento (τ), que se especifican como:

$I = < \eta, P^x, P^y >$, donde

$\eta = 6$;

$P^x = \{ (X_1, \text{binario}), (X_2, \text{binario}), (X_3, \text{binario}), (X_4, \text{binario}), (X_5, \text{binario}), (X_6, \text{binario}) \}$;

$P^y = \{ (Y_1, \text{binario}), (Y_2, \text{binario}), (Y_3, \text{binario}), (Y_4, \text{binario}), (Y_5, \text{binario}), (Y_6, \text{binario}) \}$.

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

(0,-1)	(0,0)	(0,1)	carril 0
(-1,-1)	(-1,0)	(-1,1)	carril 1

Figura 6 - Vecindario de la celda origen (carril 0)

La función τ para estas celda se define igual que para el tramo de carril único, pero se agregan las reglas que modelan el cambio de carril, que son:

Nuevo Estado	Estado del vecindario	Nombre de la regla
1	(0,0) = 0 and (0,-1) = 0 and (-1,-1) = 1 and (-1,0) = 1	Llega_Desde_CarrilDer
0	(0,0) = 1 and (-1,1) = 0 and (-1,0) = 0	Sale_Hacia_CarrilDer

Para los movimientos en diagonal se debe considerar que un vehículo primero intenta moverse derecho y que tiene prioridad de acceso a las posiciones de su propio carril frente a los coches de otros carriles. Así, la regla Llega_Desde_CarrilDer representa que un vehículo avanza hacia la celda origen desde atrás en diagonal y pide que no haya nadie que pueda acceder al mismo lugar desde atrás derecho ((0,-1) = 0) y además que el auto que quiere ocuparla no pueda avanzar sobre su carril ((-1,-1) = 1 and (-1,0) = 1). Luego la regla Sale_Hacia_CarrilDer_ConPrioridad representa que el vehículo de la celda origen puede avanzar en diagonal siempre y cuando no haya un auto que avance derecho hacia la celda destino ((-1,0) = 0).

Las celdas de la segunda fila (carril 1) del espacio sólo difieren de las anteriores en la definición del vecindario y las reglas de movimiento:

N = { (0,0), (0,1), (1,0), (1,1), (0,-1), (1,-1) };

(1,-1)	(1,0)	(1,1)	carril 0
(0,-1)	(0,0)	(0,1)	carril 1

Figura 7 - Vecindario de la celda origen (carril 1)

La función τ para estas celda se define igual que para el tramo de carril único, pero se agregan las reglas que modelan el cambio de carril, que son:

Nuevo Estado	Estado del vecindario	Nombre de la regla
1	((0,0) = 0 and (0,-1) = 0 and (1,-1) = 1 and (1,0) = 1)	Llega_Desde_CarrilIz
0	((0,0) = 1 and (1,1) = 0 and (1,0) = 0)	Sale_Hacia_CarrilIz

La regla Llega_Desde_CarrilIz representa que un vehículo avanza hacia la celda origen desde atrás en diagonal y pide que no haya nadie que pueda acceder al mismo lugar desde atrás derecho ((0,-1) = 0) y además que el auto que quiere ocuparla no pueda avanzar sobre su carril ((1,-1) = 1 and (1,0) = 1). La regla Sale_Hacia_CarrilIz representa que el vehículo de la celda origen puede avanzar en diagonal siempre y cuando no haya un auto que avance derecho hacia la celda destino ((1,0) = 0).

El modelo acoplado correspondiente al tramo se define como:

$$TC2(k, max) = < Xlist, Ylist, I, X, Y, n, \{t_1,...,t_n\}, \eta, N, C, B, Z, select >$$

TC2(k, max) significa Tramo de 2 Carriles de longitud k (celdas) cada uno, con velocidad máxima max (en Km/h).

$n = 2$

$t_1 = 2$

$t_2 = k$

$\eta = 6$

El vecindario (N) para las celdas del espacio ha sido descripto al definir el modelo de celda de cada carril. La función Z se construye siguiendo la definición dada por el formalismo Cell_DEVS, instanciada con el vecindario de este espacio.

select = { (0,1), (1,1), (0,0), (1,0), (0,-1), (1,-1) }

$X = Y = \{ 0, 1 \}$

Ylist = **Xlist** = { (0,0), (1,0), (0,k-1), (1,k-1) }

I = $<P^x, P^y>$

$P^x = \{<X_{\eta+1}(0,0), \text{binario}>, <X_{\eta+1}(1,0), \text{binario}>, <X_{\eta+1}(0,k\text{-}1), \text{binario}>, <X_{\eta+1}(1,k\text{-}1), \text{binario}>\}$
$P^y = \{<Y_{\eta+1}(0,0), \text{binario}>, <Y_{\eta+1}(1,0), \text{binario}>, <Y_{\eta+1}(0,k\text{-}1), \text{binario}>, <Y_{\eta+1}(1,k\text{-}1), \text{binario}>\}$

Estos ports serán denotados de la siguiente forma:

Port	Nombre
$X_{\eta+1}(i,0),\ 0 \le i \le 1$	x-c-hayauto
$X_{\eta+1}(i,k\text{-}1),\ 0 \le i \le 1$	x-c-haylugar
$Y_{\eta+1}(i,0),\ 0 \le i \le 1$	y-c-haylugar
$Y_{\eta+1}(i,k\text{-}1),\ 0 \le i \le 1$	y-c-hayauto

B = { (0, k-1), (1, k-1), (0,0), (1,0) }

La interface de este modelo la conforman las celdas de la primera y última columna del espacio, pues cada una debe intercambiar información con los modelos de los cruces correspondientes. La utilización de los ports externos y las reglas de avance/llegada hacia/desde el cruce son las mismas que fueran definidas para el tramo de carril único.

El comportamiento para las celdas borde es distinto al definido para el resto del espacio. Para la celda (0,0) se define el siguiente vecindario y comportamiento:

$\eta = 4$

N = { (0,0), (0,1), (-1,0), (-1,1) }

La función τ para esta celda se define igual que para las del carril 0, a excepción de las reglas Llega_Desde_Atrás y Llega_Desde_CarrilDer que son cambiadas por Llega_Desde_Cruce; pues sólo puede recibir vehículos que salen desde el cruce. Los demás parámetros del modelo para esta celda no cambian.

Para la celda (1,0) también se define un vecindario y comportamiento diferente al resto. Aquí:

$\eta = 4$

N = { (0,0), (0,1), (1,0), (1,1) }

La función τ para esta celda se define igual que para las del carril 1, a excepción de las reglas Llega_Desde_Atrás y Llega_Desde_CarrilIz que son cambiadas por Llega_Desde_Cruce; pues sólo puede recibir vehículos que salen desde el cruce. Los demás parámetros del modelo para esta celda no cambian.

Para la celda (0,k-1) se define el siguiente vecindario y comportamiento:

$\eta = 4$

N = { (-1,0), (-1,-1), (0,0), (0,-1) }

La función τ para esta celda se define igual que para las del carril 0, a excepción de las reglas Sale_Hacia_CarrilDer y Sale_Hacia_Adelante que son cambiadas por Sale_Hacia_Cruce. Los demás parámetros del modelo para esta celda no cambian.

Para la celda (1,k-1) también se define un vecindario y comportamiento diferente al resto. Aquí:

$\eta = 4$

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

La función τ para esta celda se define igual que para las del carril 1, a excepción de las reglas Sale_Hacia_CarrilIz y Sale_Hacia_Adelante que son cambiadas por Sale_Hacia_Cruce. Los demás parámetros del modelo para esta celda no cambian.

2.2. Cruces

Para modelar las intersecciones o cruces de las calles se define un espacio Cell-DEVS que se representa como un anillo de celdas acoplado con carriles de tramos, siguiendo la propuesta que fuera planteada en [CQL95], [CLQ96] y [CDL97]. Las reglas de comportamiento de los vehículos establecen que un auto dentro de la intersección (en el anillo) tiene prioridad de acceso a una posición sobre cualquier otro vehículo que está fuera de ella. Dentro del cruce, un vehículo gira en sentido contrario a las agujas del reloj o sale hacia un tramo. Para modelar la elección del enlace de salida, se utiliza una función aleatoria local a la celda. Cada vez que un auto pasa por una salida, si hay lugar en el tramo, el vehículo chequea el valor que devuelve esta función para saber si debe seguir girando o salir. Caso contrario, el vehículo permanecerá en el cruce.

Un cruce con velocidad máxima max (Km/h) y k celdas, de las cuales las posiciones del conjunto In actúan como entradas hacia el cruce y las de Out son salidas del mismo; se define como un modelo Cell_DEVS de una dimensión con demora de transporte y bordes conectados, cuya estructura se presenta en la Figura 8.

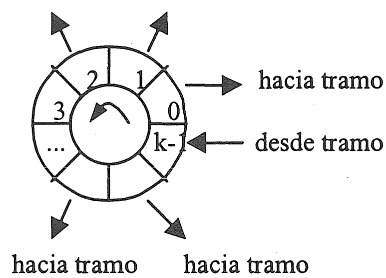

Figura 8 - Cruce

Las celdas de este espacio se definen igual que las de carril único a excepción de las reglas de comportamiento (τ), que se especifican luego de introducir el modelo acoplado. Esto se debe a que cada una de ellas se interconecta con algún tramo que influye sobre su comportamiento.

El modelo acoplado correspondiente al cruce c se define como:

$$\text{Cruce}(k, \text{In}, \text{Out}) = < \text{Xlist}, \text{Ylist}, I, X, Y, n, \{t_1,\ldots,t_n\}, \eta, N, C, B, Z, \text{select} >$$

Cruce(k, In, Out) significa que es un Cruce de longitud k (celdas), donde las posiciones de In actúan como entradas hacia el cruce y las de Out son salidas del mismo.

$n = 1$

$t_1 = k$

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

La función $\mathbb{Z}$ se construye siguiendo la definición dada por el formalismo Cell_DEVS, instanciada con el vecindario de este espacio.

$X = Y = \{ 0, 1 \}$

Ylist = **Xlist** = $\{ (0,i) \ / \ 0 \leq i < k \}$

$\mathbb{I} = <P^x, P^y>$

$P^x = \{<X_{\eta+1}(0,i), \text{binario}> / \ 0 \leq i < k \}$
$P^y = \{<Y_{\eta+1}(0,i), \text{binario}> / \ 0 \leq i < k \}$

Estos ports serán denotados de la siguiente forma:

Port	Nombre
$X_{\eta+1}(0,i)$, $i \in$ In	x-t-hayauto
$X_{\eta+1}(0,i)$, $i \in$ Out	x-t-haylugar
$Y_{\eta+1}(0,i)$, $i \in$ In	y-t-haylugar
$Y_{\eta+1}(0,i)$, $i \in$ Out	y-t-hayauto

$\mathbb{B} = \{\emptyset\}$

select = { (0,1), (0,0), (0,-1) }

τ: S x N $\rightarrow$ S,
Para definir esta función se debe tener en cuenta que el comportamiento de las celdas difiere si se trata de celdas de ingreso o de salida del cruce, por lo tanto se define un τ distinto para cada caso.

Las *celdas de salida del cruce* son aquellas del conjunto $\{ (0,i) \ / \ i \in \text{Out} \}$ y su función τ se define como:

Nuevo Estado	Estado del vecindario	Nuevo Estado de los Ports
1	(0,0) = 0 and (0,-1) = 1 and (portvalue(x-t-haylugar) = 1 or (portvalue(x-t-haylugar) = 0 and random < p_{salir})) /* Llega auto que permanecerá dentro del cruce.	send(0, y-t-hayauto)
0	(0,0) = 0 and (0,-1) = 1 and portvalue(x-t-haylugar) = 0 and random $\geq p_{salir}$ /* Llega auto que abandonará el cruce.	send(1, y-t-hayauto)
0	(0,0) = 1 and (0,1) = 0	send(0, y-t-hayauto)
(0,0)	t /*En otro caso la celda conserva el estado	send(0, y-t-hayauto)

Donde, *random* es una función aleatoria con una distribución determinada, que cuando devuelve un valor mayor que la constante p_{salir} provoca la salida del vehículo de la celda origen hacia el tramo acoplado.

El comportamiento de las celdas de salida se define con 4 reglas. La primera representa el avance de un vehículo desde la celda anterior a la origen y se toma la decisión de permanecer en cruce, ya sea porque la celda del tramo acoplado está ocupada (portvalue(x-t-haylugar) = 1) o porque la función aleatoria devolvió un valor menor a la probabilidad de salir. Se actualiza con 0 el port del tramo de salida (y-t-hayauto) pues no debe detectar la presencia del vehículo que no avanzará hacia él. La segunda regla representa el avance de un vehículo desde la celda anterior a la origen tomando la decisión de salir del cruce, para que esto sea posible la celda del tramo acoplado debe estar vacía y además la función aleatoria debe ser mayor o igual a la probabilidad de salir. Se actualiza con 1 el port y-t-hayauto para representar la salida del vehículo. La tercera regla representa que el vehículo de la celda de origen avanza hacia la siguiente celda dentro del cruce y la cuarta que la celda conserva su estado en cualquier otro caso no contemplado en las condiciones anteriores.

Las *celdas de ingreso al cruce* son aquellas del conjunto { (0,i) / i ∈ In } y su función τ se define como:

Nuevo Estado	Estado del vecindario	Nuevo Estado de los Ports
1	(0,0) = 0 and ((0,-1) = 1 or portvalue(x-t-hayauto) = 1)	send(1, y-t-haylugar)
0	(0,0) = 1 and (0,1) = 0 and (0,-1) = 0 /* No hay un auto con prioridad dentro del cruce	send(0, y-t-haylugar)
0	(0,0) = 1 and (0,1) = 0 and (0,-1) = 1 /* Hay un auto con prioridad dentro del cruce	send(1, y-t-haylugar)
(0,0)	t /*En otro caso la celda conserva el estado	-

Las celdas de ingreso al cruce pueden recibir vehículos desde su vecina de atrás o desde el tramo acoplado (port x-t-hayauto). La primera regla modela que si alguna de ellas tiene un vehículo y la celda de origen está vacía entonces avanzará, actualizando el port del tramo con 1 pues la celda está ocupada. La segunda y tercera regla representan que la celda origen se vacía si la de adelante lo está; diferenciándose en el estado de la celda de atrás. Si esta última está vacía entonces se actualiza el port con valor 0 pues hay suficiente lugar para que ingrese un vehículo desde el tramo, caso contrario ((0,-1) = 1) el estado del port se actualiza con 1 porque hay un auto dentro del cruce que tiene prioridad de acceder a la celda origen y los coches del tramo no pueden entrar en la intersección. Esto significa que el port y-t-haylugar se actualiza con 0 sólo cuando la celda origen y la anterior están vacías, permitiendo que un auto ingrese desde el tramo sólo si no hay otro dentro del cruce que quiera ocupar la misma posición. La cuarta regla representa que la celda conserva su estado en cualquier otro caso no contemplado en las condiciones anteriores.

3. Conclusiones

El modelo para calles y cruces planteado sobre el paradigma Cell-DEVS resulta en un conjunto de reglas de comportamiento simples, formadas por expresiones booleanas, que pueden ser comprendidas, modificadas y ejecutadas sin demasiado esfuerzo. Además la utilización de la demora de transporte para modelar la velocidad de los vehículos, colabora en la reducción de la complejidad de las reglas. Luego, es factible una extensión del modelo para incorporar más aspectos que influyen sobre el movimiento de los autos, como ser semáforos, baches, señales de tránsito, etc.

4. Referencias

[CDL97] CHOPARD, B.; DUPUIS, A..; LUTHI, P. "A Cellular Automata Model for Urban Traffic and its aplications to the city of Genova". Proceedings of Traffic and Granular Flow. 1997.

[CLQ96] CHOPARD, B.; QUELOZ, P.; LUTHI, P. "Cellular Automata Model of Car Traffic in two-dimensional street networks". J. Phys. A, vol 29, pp. 2325-2336, 1996.

[CQL95] CHOPARD, B.; QUELOZ, P.; LUTHI, P. "Traffic Models of a 2D road network". Proceedings of the 3rd CM users' Meeting. October 1995. Parma. Italy.

[DW99] DAVIDSON, A.; WAINER, G. "Definición de un lenguaje de especificación para simulación de tráfico urbano siguiendo el paradigma Cell-DEVS". Technical Report 99-003, Departamento de Computación, FCEN/UBA. 1999.

[W98] WAINER, G. "Discrete-Events cellular models with explicit delays". Tesis de Doctorado. Université d'Aix-Marseille III. France. 1998.

[WG98] WAINER, G.; GIAMBIASI, N. "Specification, modelling and simulation of timed Cell-DEVS models". Technical Report 97-007, Departamento de Computación, FCEN/UBA. Submitted to publication. 1998.

[Zei76] ZEIGLER, B. "Theory of modeling and simulation". Wiley, 1976.