

Máquinas de estados finitos composicionales

Sebastián Dopazzo
UTN–FRBA, SECTIP
GIAR
Buenos Aires, Argentina
sdopazo@frba.utn.edu.ar

Joaquín S. Toranzo Calderón
SADAF–CONICET
IIF
Buenos Aires, Argentina
jtoranzocalderon@frba.utn.edu.ar

Ramiro G. Rodríguez Colmeiro
UTN–FRBA, SECTIP
GIAR
Buenos Aires, Argentina
rrodriguezcolmeiro@frba.utn.edu.ar

Sergio Alberino
UTN–FRBA, SECTIP
GIAR
Buenos Aires, Argentina
salberino@frba.utn.edu.ar

Abstract—Se propone un método para definir máquinas de estados finitos (MEFs) composicionales, i.e. MEFs compuestas de otras MEFs. Esto permite orientar la generación de agentes basados en modelos hacia sistemas más robustos y fácilmente analizables. Se ejemplifica con una MEF definida manualmente y con su equivalente composicional.

Keywords—máquinas de estados finitos, exploración robótica, entornos discretos, diseño automático de agentes

I. INTRODUCCIÓN

Si bien los enfoques simbólicos o abstractos han gozado de cierta preeminencia en la literatura sobre inteligencia artificial aplicada a sistemas robóticos [1], desde hace tiempo hay un incremento en las motivaciones para desarrollar sistemas que se basen en agentes reactivos y situados respecto de su entorno más que a un modelo o constructo de éste [2]. Este tipo de sistemas tienen ciertas ventajas que se encuentran cada vez más alineadas con las nuevas arquitecturas computacionales, principalmente por la mayor capacidad para paralelizar procesos simples, los que permiten obtener soluciones a partir de sistemas distribuidos o multiagentes [2]. Las limitaciones de estos sistemas suelen surgir cuando la cantidad de agentes posibles es tal que no hay restricciones obvias a cómo seleccionarlos o combinarlos.

Este trabajo continúa este esfuerzo por obtener soluciones a partir de comportamientos globales exitosos que emergen de agentes situados de poca complejidad. Con el objetivo de resolver la exploración de entornos discretos, en un trabajo previo [3] se propusieron agentes reactivos basados en modelos [1, pp. 50-52], especialmente máquinas de estados finitos (MEFs) [4]. Éstas modelan el comportamiento de sistemas con números finitos de estados en el que cada uno tiene una salida correspondiente a distintas entradas. Esencialmente, las MEFs definen la lógica de aplicación, con la ventaja de consumir poca memoria y determinar sus acciones rápidamente [5, 6].

En ciertos tipos de entornos discretos, su exploración con MEFs se mostró prometedora, pero las limitaciones de estos sistemas surgieron rápidamente, requiriendo criterios de selección o diseño de MEFs para un estudio más profundo de las combinaciones más exitosas [3]. Aunque las MEFs ofrecen una estructura organizada para representar comportamientos discretos, su complejidad aumenta exponencialmente a medida que se expande el número de estados y transiciones (ver II). Este aumento de complejidad

se hace evidente en comportamientos donde las interacciones entre los estados y las entradas son numerosas.

Muestrear sobre toda MEF definible, dada una caracterización del entorno, es impracticable, pero tampoco existe un criterio general para restringir los diseños de MEFs. El diseño manual de MEFs involucra dificultades en la propuesta y evaluación de estos sistemas [3]. En este trabajo se propone un método para definir MEFs composicionales, i.e. MEFs que se componen de otras MEFs. Estas últimas describirán comportamientos muy sencillos, pero robustos (tales como “avanzar hasta que sea imposible” o “girar y avanzar un paso a la derecha”), por lo que su definición y análisis se vuelven eficientemente tratables. La tarea de diseño se realiza, luego, en la composición de MEFs más complejas que parten de estas otras ya estudiadas.

II. MÁQUINAS DE ESTADOS FINITOS

Dado el uso pretendido de las MEFs a agentes o plataformas robóticas que empleen pocos recursos, una forma de conceptualizar las MEFs [4] es como sistemas que evalúan variables de su entorno como entradas del sistema (E), que se encuentran en un estado u otro de un conjunto dado (Q), y cada combinación de estado posible y entrada posible, realizan una acción de un conjunto posible (S), pudiendo mantenerse en el mismo estado o debiendo pasar a otro. Formalmente, una máquina de estados finitos M se define como la estructura $M = \langle E, S, Q, ft, fs \rangle$, donde ft y fs son, respectivamente, la función de transición entre estados y la función de salida del sistema. Es decir, ft indica para cada estado y cada combinación de entradas posible, a qué estado pasa el sistema (si es que cambia de estado), y fs indica qué acción realiza (para cada combinación de estados y entradas).

En este trabajo se continuará la propuesta de [3], donde se propone que S tenga como acciones posibles moverse hacia adelante una celda del entorno discreto en el que se encuentra el agente, girar hacia la derecha o la izquierda 90°, o quedarse quieto en el lugar, denotadas respectivamente por los símbolos A, D, I y N. Las entradas se corresponderán con lo determinado por sensores de proximidad en cada dirección, formando una estructura <sensor frontal, sensor derecho, sensor trasero, sensor izquierdo>, de manera que un ‘0’ representa que se puede acceder a la celda en la dirección del sensor, y un ‘1’ indica que no. Por ejemplo, la secuencia <1000> indica que excepto al frente, el agente está libre de moverse en toda dirección, y la secuencia <0110> indica que podría moverse sólo hacia delante y hacia la izquierda.

Considérese una MEF cuyo comportamiento consiste en ubicar una pared del entorno a su derecha y luego rodearla. Ubicada la pared, avanzaría mientras pueda o giraría buscando un espacio por el cual avanzar, pero en cuanto pierda la pared por avanzar, deberá girar y avanzar hacia la derecha, donde eventualmente encontrará la pared que venía siguiendo. Esta MEF es fácil de definir sólo con un par de estados, y si la meta de la exploración es adyacente a la pared que se está siguiendo, será efectiva [3]. Pero ese no es siempre el caso, y es conveniente contar con otras MEFs que propongan estrategias superiores.

En la Tabla 1 se define una MEF más compleja **M**. Su comportamiento busca representar una exploración que ubique celdas del entorno donde haya más de un camino para tomar, y por explorarlos de a uno, de manera que si estuvieran bloqueados (la entrada fuera <1110>), se regresaría a la derivación para explorar otro camino. Sin embargo, tomar un camino podría llevar a otra derivación, y registrar todas las derivaciones posibles requeriría una gran cantidad de estados indeterminados, por lo que este comportamiento reinicia su comportamiento en cada nueva derivación. Puede observarse que la cantidad de transiciones posibles es suficientemente grande como para dificultar su análisis.

Sin embargo, **M** puede definirse a partir de otras MEFs más básicas, con comportamientos fácilmente definibles. En lugar de considerar estados aislados y todas las transiciones necesarias desde ese estado hacia los otros que se fueran a usar, la propuesta de este trabajo es definir en base a pocos estados un comportamiento que tenga condiciones de finalización—i.e. condiciones bajo las cuales se considere que el comportamiento descrito ha concluido. De ese modo, se definen instancias de clases de MEFs, que varíen en las transiciones que realizan a partir de las condiciones de finalización.

TABLA I. DEFINICIÓN DE **M**: EXPLORACIÓN DE PASILLOS

Entradas	Estados de M						
	<i>Sa</i>	<i>Sb</i>	<i>Sc</i>	<i>Sd</i>	<i>Se</i>	<i>Sf</i>	<i>Sg</i>
0000	A Sb	A Sb	A Sd	A Sb	I Sb	A Sb	A Sb
0001	A Sb	A Sb	A Sd	A Sb	D Sf	A Sb	A Sb
0010	A Sa	A Sb	A Sd	A Sd	A Se	A Sf	A Sg
0011	A Sa	A Sb	A Sd	A Sd	A Se	A Sf	A Sg
0100	A Sb	A Sb	A Sd	A Sb	I Sd	A Sb	A Sb
0101	A Sa	A Sb	A Sd	A Sd	A Se	A Sf	A Sg
0110	A Sa	A Sb	A Sd	A Sd	A Se	A Sf	A Sg
0111	A Sa	A Sb	A Sd	A Sd	A Se	A Se	A Sg
1000	I Sc	I Sc	N Sc	I Sc	I Sc	I Sc	I Sc
1001	D Sa	D Sb	N Sc	D Sd	D Se	D Sf	D Sg
1010	I Sa	I Sb	N Sc	I Sd	I Se	I Sf	I Sg
1011	D Sa	D Sb	N Sc	D Sd	D Se	D Sf	D Sg
1100	I Sa	I Sb	N Sc	I Sd	I Se	I Sf	I Sg
1101	I Sa	I Se	N Sc	I Sg	I Se	I Sf	I Sg
1110	I Sa	I Sb	N Sc	I Sd	I Se	I Sf	I Sg
1111	N Sa	N Sb	N Sc	N Sd	N Se	N Sf	N Sg

Por ejemplo, en la Tabla 2 se encuentran tres MEFs esquemáticas **Mx**, **My** y **Mz**. Si se comparan los estados de **M** en la Tabla 1, puede verse que *Se* es instancia de **My**, que *Sc* es instancia de **Mz**, y que *Sa*, *Sb*, *Sd*, *Sf* y *Sg* son instancias de **Mx**; estas últimas sólo se diferencian en el estado del que parten y los estados con los que instancian las variables que figuran como condiciones de salida—i.e. *C1*, *C2*, *C3* y *C4*.

III. TRABAJOS FUTUROS

La implementación del procedimiento presentado, i.e. un método para definir MEFs composicionales, permitirá extender los resultados de [3], principalmente en el análisis de nuevas MEFs que puedan ser tomadas en conjunto para resolver tareas de exploración.

En segundo lugar, aunque el diseño de MEFs de base sea manual, el diseño de MEFs composicionales puede automatizarse. Se espera integrar el procedimiento presentado a sistemas automáticos basados en algoritmos genéticos y aprendizaje por refuerzo.

REFERENCIAS

- [1] S. J. Russell y P. Norvig, *Artificial Intelligence: A Modern Approach*, 3ed., Prentice Hall, 2009.
- [2] R. A. Brooks, "Intelligence Without Reason" en L. Steels y R. A. Brooks (eds.), *The Artificial Life Route to Artificial Intelligence: Building Embodied Situated Agents*, Lawrence Erlbaum Associates Inc., 1995, pp. 25-81.
- [3] J. S. Toranzo Calderón, R. G. Rodríguez Colmeiro y S. Alberino, "Análisis de navegación de agentes reactivos en entornos discretos", *Proyecciones*, 2024, 22 (1), pp. 11-22.
- [4] P. Isasi, P. Martínez y D. Borrajo, *Lenguajes, gramáticas y autómatas. Un enfoque práctico*, Addison-Wesley, 1997.
- [5] S. Alberino, F. Sampieri, P. Folino, J. C. Gómez y C. Verrastro, "Programación de robots utilizando hoja de cálculo" en *Actas de las VIII Jornadas Argentinas de Robótica*, Buenos Aires, Argentina, 2015.
- [6] S. Alberino, J. S. Toranzo Calderón, J. C. Gómez y C. Verrastro, "Método de programación para robótica didáctica" en *Actas de las X Jornadas Argentinas de Robótica*, Neuquén, Argentina, 2019.

TABLA II. DEFINICIONES DE **Mx**, **My** y **Mz**: ESQUEMAS DE MEFs CUYAS INSTANCIAS SIRVEN PARA COMPONER A **M**.

Entradas	MEFs esquemáticas		
	<i>Mx</i>	<i>My</i>	<i>Mz</i>
0000	A C1	I C1	A C1
0001	A C1	D C2	A C1
0010	A x	A y	A C1
0011	A x	A y	A C1
0100	A C1	I C3	A C1
0101	A x	A y	A C1
0110	A x	A y	A C1
0111	A C2	A y	A C1
1000	I C3	D C4	N z
1001	D x	D y	N z
1010	I x	I y	N z
1011	D x	D y	N z
1100	I x	I y	N z
1101	I C4	I y	N z
1110	I x	I y	N z
1111	N x	N y	N z