

Implementación del Juego de la vida de Conway en PowerDEVS

Lucio Mansilla

26 de julio de 2023

1. Introducción / Problema

El Juego de la Vida de Conway, comúnmente conocido como "Juego de la Vida", es un autómata celular que fue propuesto por el matemático británico John Horton Conway en 1970. Los autómatas celulares son modelos matemáticos para sistemas dinámicos que evolucionan en pasos de tiempo discretos (generaciones). A pesar de su simplicidad aparente, tienen la capacidad de simular sistemas complejos y mostrar comportamientos emergentes, siendo utilizados en diversos contextos, desde la física hasta la teoría de computación.

En el Juego de la Vida, cada celda en una cuadrícula bidimensional puede estar en uno de dos estados: "viva" o "muerta". Las celdas interactúan con sus ocho vecinos adyacentes en horizontal, vertical y diagonal, transicionando entre los estados de vida y muerte de acuerdo a las siguientes reglas de evolución:

- Cualquier celda viva con dos o tres vecinos vivos sobrevive para la siguiente generación.
- Cualquier celda viva con menos de dos vecinos vivos muere por soledad para la siguiente generación.
- Cualquier celda viva con más de tres vecinos vivos muere por superpoblación para la siguiente generación.
- Cualquier celda muerta con exactamente tres vecinos vivos nace para la siguiente generación.

Aunque estas son las reglas originales, existen muchas variantes del Juego de la Vida con diferentes reglas. Además, es posible observar la aparición de diversos patrones, algunos estáticos, otros oscilan entre varios estados y otros se desplazan por el tablero. Estos patrones serán el objeto de estudio en las secciones posteriores de este informe.

El propósito de este proyecto es explorar la dinámica del juego de una manera visual e interactiva, para ello se implementará el mismo en PowerDEVS, una herramienta de simulación de eventos discretos basada en la teoría DEVS (Discrete Event System Specification).

2. Especificación DEVS de una celda

Puesto que el Juego de la Vida es un autómata celular, es natural modelarlo como un sistema DEVS. En particular, cada celda en la cuadrícula se modela como un DEVS atómico, que se comunica con sus vecinos a través de sus puertos de entrada y salida.

El DEVS que representa a una célula está definido como

$$C = \langle X, Y, S, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, ta \rangle$$

donde

- $X = \{R\}$ es el conjunto de entradas.

3. Implementación en PowerDEVS

4. Patrones

El Juego de la Vida es conocido por la variedad de patrones que pueden surgir de sus reglas simples. Los patrones son configuraciones de células que se repiten después de un número fijo de generaciones, y pueden clasificarse en varias categorías según su comportamiento.

4.1. Patrones estáticos

Los patrones estáticos, también conocidos como "still lifes", son configuraciones de células que no cambian de una generación a la siguiente. Un ejemplo clásico de un patrón estático es el "bloque", una cuadrícula de 2x2 células vivas.

4.2. Osciladores

Los osciladores son patrones que se repiten después de un número fijo de generaciones, llamado su período. Un ejemplo famoso es el "blinker", una línea recta de tres células vivas que oscila entre una orientación horizontal y vertical.

4.3. Naves espaciales

Las naves espaciales son patrones que se trasladan a través de la cuadrícula mientras oscilan. El ejemplo más conocido es la "nave ligera" (o "glider".^{en} inglés), que se desplaza diagonalmente a través de la cuadrícula mientras oscila entre cuatro configuraciones diferentes.

4.4. Patrones complejos

Además de estos patrones simples, también existen patrones más complejos en el Juego de la Vida. Algunos de estos pueden "disparar" naves espaciales, otros pueden construir patrones adicionales, y otros pueden incluso comportarse como máquinas de Turing universales, lo que significa que pueden computar cualquier función computable.

Estos patrones ilustran la increíble diversidad de comportamientos que pueden surgir del Juego de la Vida. En las siguientes secciones, veremos cómo estos patrones pueden ser generados e investigados usando PowerDEVS.

5. Conclusiones