Tarea 1: Juego de la vida de Conway

PROFESOR

Johansell Villalobos Cubillo

2025-06-04

Índice

1. Descripción General

El **Juego de la Vida** es un *autómata celular* propuesto por el matemático británico **John Horton Conway** en 1970. Se trata de una simulación matemática en una **rejilla bidimensional** donde cada celda puede estar en uno de dos estados: **viva** o **muerta**. A lo largo de pasos discretos de tiempo (llamados *generaciones*), el sistema evoluciona de acuerdo con reglas locales muy simples, basadas en el estado de las celdas vecinas.

2. Reglas del Juego

Cada celda tiene 8 vecinas (adyacentes en horizontal, vertical y diagonal). En cada generación, el estado de cada celda se actualiza simultáneamente según las siguientes reglas:

1. **Superpoblación:** Una celda viva con más de tres vecinos vivos muere.
2. **Soledad:** Una celda viva con menos de dos vecinos vivos muere.
3. **Supervivencia:** Una celda viva con dos o tres vecinos vivos permanece viva.
4. **Reproducción:** Una celda muerta con exactamente tres vecinos vivos se convierte en celda viva.

3. Comportamientos Emergentes

A pesar de la simplicidad de sus reglas, el Juego de la Vida exhibe comportamientos altamente complejos y sorprendentes:

- **Osciladores:** Patrones que se repiten tras un número fijo de generaciones (e.g., *Blinker*).
- **Naves espaciales:** Patrones que se desplazan a través de la rejilla (e.g., *Glider*).
- **Estructuras estáticas:** Patrones que permanecen inalterados (e.g., *Block*).
- **Computación:** Es un sistema **Turing completo**, capaz de simular cualquier algoritmo computacional.

4. Importancia

El Juego de la Vida ha sido ampliamente estudiado en disciplinas como:

- Teoría de sistemas complejos y autoorganización.
- Vida artificial y biología teórica.
- Computación teórica y modelos de universalidad.
- Filosofía de la mente y emergencia.

Este modelo ilustra cómo reglas locales simples pueden dar lugar a comportamientos globales inesperadamente complejos, siendo un excelente ejemplo de *emergencia* en sistemas dinámicos.

5. Objetivo

El propósito de esta tarea es consolidar conceptos de programación orientada a objetos, visualización de datos y análisis de rendimiento en el contexto de programación paralela. Los estudiantes implementarán el algoritmo del Juego de la Vida de Conway en Python, realizarán visualizaciones de su evolución y medirán empíricamente.

6. Instrucciones

6.1. Implementación orientada a objetos

- Desarrolle una clase `GameOfLife` en Python que contenga al menos los siguientes métodos:
  - `__init__(self, rows, cols, initial_state=None)`: para inicializar el tablero.
  - `step(self)`: que actualiza el estado del tablero según las reglas de Conway.
  - `run(self, steps)`: para ejecutar múltiples iteraciones del juego.
  - `get_state(self)`: que devuelve el estado actual del tablero.
- La implementación debe poder funcionar con un estado inicial aleatorio o uno provisto manualmente.

6.2. Visualización

- Utilice `matplotlib` o `matplotlib.animation` para visualizar la evolución del juego.
- Cree animaciones o secuencias de imágenes para representar gráficamente la evolución de patrones clásicos (como *Glider*, *Blinker*, *Toad*, etc.).
- La visualización debe ser capaz de funcionar en diferentes tamaños de grilla (e.g., 32x32, 128x128, 512x512).

6.3. Medición de rendimiento y complejidad empírica

- Realice pruebas de rendimiento empíricas variando el tamaño de la grilla (por ejemplo: 32x32, 64x64, 128x128, ..., 1024x1024).
- Para cada tamaño, mida el tiempo promedio de ejecución por iteración del juego.
- Presente una gráfica de tiempo vs tamaño de entrada (número de celdas) y compare con curvas teóricas de complejidad ($O(n)$, $O(n \log n)$, $O(n^2)$, etc.).
- Incluya al menos una visualización log-log.

6.4. Análisis y discusión

- Analice los resultados obtenidos. ¿Cómo escala su implementación? ¿Qué limitaciones o cuellos de botella observa?
- Compare la eficiencia relativa de su versión paralela frente a la secuencial.

6.5. Entrega

- Código bien documentado en Python.
- Incluir un `README.md` explicando cómo ejecutar su simulación, cómo generar las visualizaciones y cómo reproducir los experimentos. Capturas de pantalla o animaciones del juego. Gráficas de rendimiento. Discusión de resultados y conclusiones.