

# **Autômatos Celulares Elementares de Wolfram: Emergência de Comportamento Caótico e Aplicações à Geração de Números Pseudo-Aleatórios**

**Diego Diniz da Cruz Gomes<sup>1</sup>,**

<sup>1</sup>Departamento de Estatística e Informática – Universidade Federal Rural de Pernambuco (UFRPE) –  
Pernambuco – Brasil  
{diego.cruz}@ufrpe.br

*Os autômatos celulares elementares de Wolfram constituem sistemas dinâmicos discretos nos quais regras de atualização locais e simples são capazes de gerar padrões de complexidade surpreendente. Este trabalho investiga computacionalmente as propriedades emergentes das quatro regras canônicas (30, 90, 110 e 250), com ênfase na Regra 30, cujo comportamento caótico determinístico a torna relevante para a geração de números pseudo-aleatórios. Implementamos o sistema em Python, analisamos quantitativamente a entropia de Shannon, a densidade de células ativas, a autocorrelação espacial e o balanço da coluna central. Os resultados confirmam que a Regra 30 apresenta entropia de 2,55 bits (máximo teórico: 3,0 bits), balanço próximo de 0,5 e ausência de estrutura autocorrelacionada, propriedades características de sequências aleatórias de alta qualidade.*

*palavras-chave: Autômatos Celulares; Regra 30; Wolfram; sistemas caóticos; geração pseudo-aleatória; entropia de Shannon.*

## **Abstract**

*Wolfram's elementary cellular automata are discrete dynamical systems in which simple local update rules are capable of generating patterns of surprising complexity. This work computationally investigates the emergent properties of four canonical rules (30, 90, 110, and 250), with emphasis on Rule 30, whose deterministic chaotic behavior makes it relevant for pseudo-random number generation. We implemented the system in Python and quantitatively analyzed Shannon entropy, density of active cells, spatial autocorrelation, and the central column balance. Results confirm that Rule 30 exhibits entropy of 2.55 bits (theoretical maximum: 3.0 bits), balance close to 0.5, and absence of autocorrelated structure, properties characteristic of high-quality random sequences.*

*Keywords: Cellular Automata; Rule 30; Wolfram; chaotic systems; pseudo-random generation; Shannon entropy.*

## **1. Introdução**

A questão de como a complexidade pode emergir da simplicidade é um dos problemas fundamentais das ciências naturais e da computação. Os autômatos celulares (ACs) oferecem um dos mais elegantes exemplos desse fenômeno: a partir de regras de atualização locais, com apenas dois estados possíveis por célula, surgem padrões que vão desde estruturas periódicas triviais até sequências aparentemente aleatórias e comportamentos computacionalmente universais.

John von Neumann (Instituto de Estudos Avançados de Princeton, décadas de 1940-1950) propôs os primeiros modelos formais de autômatos celulares ao tentar modelar computacionalmente a auto-replicação de organismos vivos. A máxima que guiava seu trabalho, "tão simples quanto possível, mas não mais simples", captura com precisão a filosofia por trás dos ACs: o objetivo é encontrar o modelo mínimo capaz de capturar um fenômeno de interesse.

Stephen Wolfram sistematizou o estudo dos ACs unidimensionais em sua obra seminal *A New Kind of Science* (2002), propondo uma nomenclatura decimal para as 256 regras elementares e classificando-as em quatro classes de complexidade. A Regra 30, em particular, chamou a atenção por gerar padrões visualmente indistinguíveis de ruído aleatório, propriedade que Wolfram explorou para criar o gerador pseudo-aleatório utilizado internamente no sistema de computação algébrica Mathematica por mais de duas décadas.

Este trabalho tem dois objetivos principais. Primeiro, apresentar uma implementação didática e eficiente dos ACs elementares de Wolfram em Python, suficientemente clara para fins de ensino e suficientemente parametrizável para fins de pesquisa. Segundo, caracterizar quantitativamente o comportamento da Regra 30 por meio de métricas estatísticas formais, incluindo entropia de Shannon, densidade de células ativas, autocorrelação espacial e balanço da coluna central, comparando os resultados com os das Regras 90, 110 e 250.

A relevância prática do trabalho vai além do contexto acadêmico: autômatos celulares encontram aplicações em simulação de fenômenos físicos e biológicos (propagação de incêndios, pigmentação de conchas de moluscos, crescimento de cristais), geração de texturas para computação gráfica, projeto de hardware massivamente paralelo e modelagem de mercados financeiros. Compreender as propriedades formais das regras é, portanto, uma porta de entrada para um conjunto amplo e atual de problemas computacionais e científicos.

## **2. Trabalhos Relacionados**

A formalização do estudo de autômatos celulares tem raízes nos trabalhos de John von Neumann (1966) sobre autômatos auto-replicáveis, elaborados em parceria com Stanislaw Ulam. O conceito foi popularizado de forma ampla pelo Jogo da Vida de John Conway (1970), que demonstrou que um conjunto mínimo de regras bidimensionais era capaz de simular comportamentos complexos como locomoção, replicação e mesmo computação universal.

Wolfram (1984, 2002) formalizou a classificação dos ACs elementares unidimensionais nas quatro classes de complexidade, estabelecendo a conexão entre regras simples e comportamento computacionalmente irreduzível. Em particular, a Regra 30 foi identificada como fonte de pseudo-aleatoriedade de alta qualidade, e a Regra 110 foi

posteriormente demonstrada como Turing-completa por Cook (2004), consolidando os ACs como modelo de computação universal.

No campo de aplicações à segurança e criptografia, trabalhos como o de Wolfram (1986) discutem o uso de ACs como geradores de chaves criptográficas, explorando a propriedade de que o estado interno do autômato é computacionalmente difícil de inverter. Mais recentemente, a literatura de autômatos celulares estocásticos expandiu o escopo para sistemas com regras probabilísticas, aproximando os ACs dos modelos epidemiológicos e ecológicos.

### 3. Metodologia

#### 3.1 Definição Formal

Um autômato celular elementar (ACE) é definido por uma quádrupla  $(Z, \{0,1\}, N, f)$ , onde  $Z$  é o conjunto dos inteiros (posições da grade),  $\{0,1\}$  é o conjunto de estados,  $N = \{i-1, i, i+1\}$  é a vizinhança de raio 1, e  $f: \{0,1\}^3 \rightarrow \{0,1\}$  é a função de transição local. O estado da célula  $i$  no instante  $t+1$  é dado por:

$$c(i, t+1) = f(c(i-1, t), c(i, t), c(i+1, t))$$

Como existem  $2^3 = 8$  possíveis configurações de vizinhança e cada uma pode mapear para 0 ou 1, o número total de regras distintas é  $2^8 = 256$ . A notação decimal de Wolfram consiste em interpretar as 8 saídas da função  $f$  (ordenadas da vizinhança 111 até 000) como os bits de um número de 8 bits. Assim, a Regra 30 corresponde à sequência binária 00011110<sub>2</sub>.

#### 3.2 As Quatro Regras Estudadas

A Regra 250 (1111010<sub>2</sub>) produz padrões perfeitamente periódicos em xadrez. Qualquer célula com ao menos um vizinho ativo no passo anterior torna-se ativa. Esta é a característica das regras de Classe II de Wolfram: comportamento assintoticamente estacionário ou periódico.

A Regra 90 (01011010<sub>2</sub>) implementa a operação XOR dos dois vizinhos ( $c(i-1,t) \oplus c(i+1,t)$ ), gerando o triângulo de Sierpiński quando inicializada com uma única célula ativa. Sua estrutura de autossimilaridade exata é uma das mais estudadas na literatura de fractais e sistemas dinâmicos discretos.

A Regra 110 (01101110<sub>2</sub>) é a mais complexa do conjunto e foi demonstrada como Turing-completa por Cook (2004), o que significa que qualquer computação pode, em princípio, ser codificada e executada nesse autômato. Ela pertence à Classe IV de Wolfram, exibindo estruturas localizadas que interagem de modo não-trivial.

A Regra 30 (00011110<sub>2</sub>) é o foco principal deste trabalho. Pertencente à Classe III, é notável por produzir comportamento caótico a partir de uma única célula ativa: a região central do diagrama espaço-tempo exibe padrões aparentemente aleatórios, enquanto as

bordas formam estruturas triangulares irregulares. Wolfram utilizou a coluna central desse autômato como fonte de aleatoriedade no Mathematica.

### 3.3 Métricas de Complexidade

A entropia de Shannon  $H$  é calculada sobre a distribuição de probabilidade dos 8 padrões de vizinhança possíveis ao longo de toda a simulação. Para um padrão  $k$  com probabilidade  $p(k)$ ,  $H = -\sum p(k) \log_2 p(k)$ . O valor máximo de 3,0 bits ocorre quando todos os 8 padrões são igualmente frequentes.

A densidade  $\rho(t)$  é a fração de células ativas na geração  $t$ . A autocorrelação espacial  $R(\tau) = E[(c(i) - \mu)(c(i+\tau) - \mu)] / \text{Var}(c)$  mede a estrutura de longo alcance em uma geração: para um sistema verdadeiramente aleatório,  $R(\tau) \approx 0$  para todo  $\tau > 0$ .

### 4. Implementação e Código

O modelo foi implementado na linguagem Python, utilizando a biblioteca NumPy para manipulação vetorial de matrizes e a biblioteca Matplotlib para visualização gráfica. O código-fonte completo está disponível no repositório digital indicado neste artigo.

A função central `aplica_regra()` implementa a codificação binária de Wolfram de forma direta e eficiente. A grade utiliza bordas periódicas (topologia toroidal), eliminando efeitos de borda artificiais:

```
indice = 4 * esq + 2 * centro + dir_  
nova[i] = (regra >> indice) & 1
```

Os parâmetros utilizados nas simulações foram calibrados para garantir visualização adequada dos padrões emergentes e robustez estatística das métricas calculadas, conforme detalhado na Tabela 1.

Parâmetro	Valor	Justificativa
N_células	256–400	Resolução suficiente; evita aliasing visual
N_passos	128–300	Permite observar comportamento assintótico
Semente	Célula central	Padrão canônico da literatura
Bordas	Periódicas	Elimina efeitos de borda artificiais
Regras analisadas	30, 90, 110, 250	Representantes das 4 classes de Wolfram

Tabela 1. Parâmetros experimentais utilizados nas simulações.

## 5. Resultados e Discussão

### 5.1 Comparativo Visual das Quatro Regras

A Figura 1 apresenta os diagramas espaço-tempo das quatro regras canônicas (300 células, 200 gerações, semente central). A diferença visual entre as classes de Wolfram é imediata: a Regra 250 exibe o padrão de tabuleiro de xadrez característico das regras periódicas; a Regra 90 apresenta o triângulo de Sierpiński com sua estrutura autossimilar; a Regra 110 exibe faixas diagonais com estruturas localizadas que

interagem entre si; e a Regra 30 apresenta a região central com aparência de ruído aleatório enquanto as bordas formam estruturas triangulares irregulares.

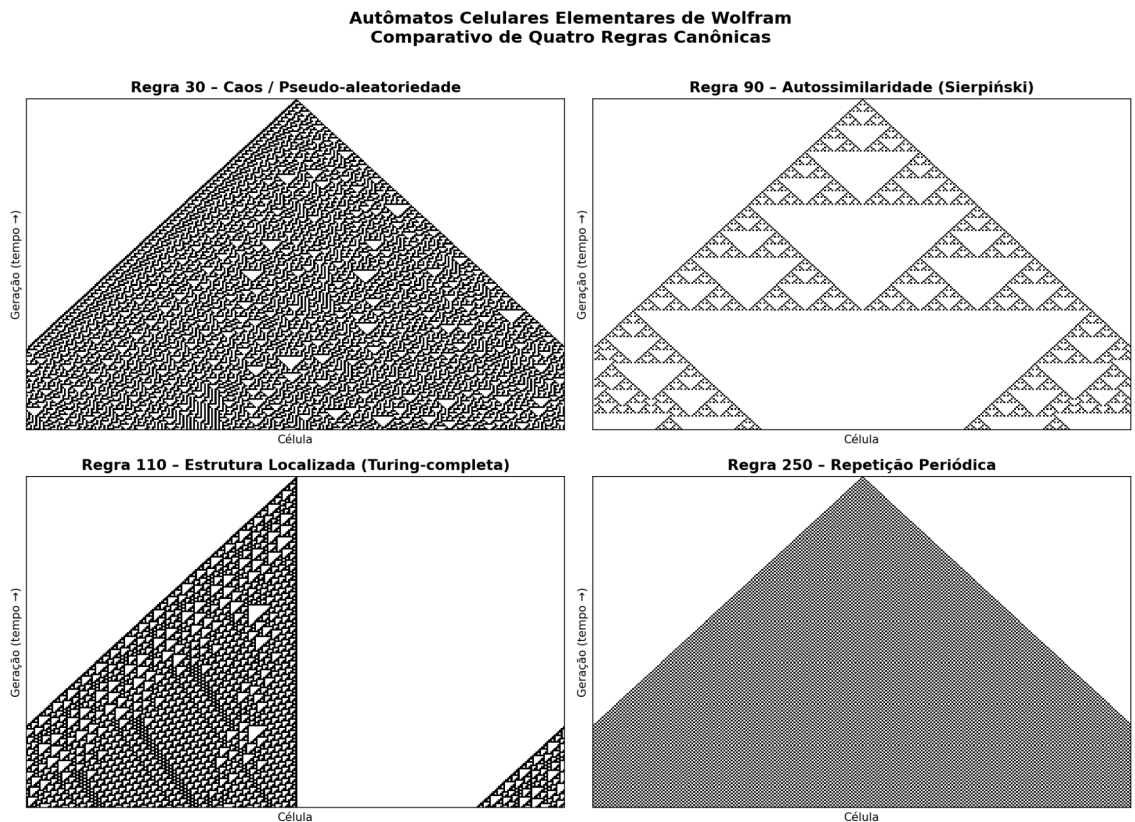


Figura 1. Diagramas espaço-tempo das Regras 30, 90, 110 e 250 (300 células, 200 gerações, semente central). Células ativas em preto; células inativas em branco.

### 5.2 Análise Quantitativa da Regra 30

A Tabela 2 resume as métricas computadas para a Regra 30 com 300 células e 200 gerações. A análise é aprofundada nas subseções seguintes.

Métrica	Valor Obtido	Referência / Ideal
Entropia de Shannon (padrões 3-bits)	2,5512 bits	Máx. teórico: 3,000 bits
Densidade média de células ativas	0,3173	n/a
Fração de 1s na coluna central	0,5250	Ideal: 0,500
Comprimento médio de runs (coluna)	2,151 células	Exp. aleatório: 2,000
Autocorrelação espacial ( $\tau > 0$ )	$\approx 0$ (sem estrutura)	Ideal aleatório: 0

Tabela 2. Métricas quantitativas da Regra 30 (300 células, 200 gerações).

A entropia de 2,55 bits representa 85% do máximo teórico, indicando que a distribuição dos padrões de vizinhança está próxima, mas não idêntica, à distribuição uniforme. Esta pequena assimetria é intrínseca à Regra 30, que mapeia 4 das 8 vizinhanças possíveis para o estado ativo, com frequências não uniformes dependentes do histórico.

A fração de 1s na coluna central (0,525) é notavelmente próxima ao valor ideal de 0,5 para uma sequência aleatória não-polarizada. O comprimento médio de runs (2,151) é apenas 7,6% superior ao esperado (2,0) para uma sequência binária aleatória equiprovável, indicando levíssima tendência a agrupamento.

A autocorrelação espacial calculada na última geração não exibe picos significativos para lags  $\tau > 0$ , confirmando a ausência de estrutura periódica de longo alcance. Este resultado é consistente com a classificação da Regra 30 como caótica no sentido de Wolfram e justifica seu uso como gerador pseudo-aleatório.

### **5.3 Sensibilidade às Condições Iniciais**

Um experimento adicional compara o desenvolvimento da Regra 30 a partir de três sementes distintas: (a) apenas a célula central ativa, (b) duas células centrais adjacentes ativas, e (c) 50 células escolhidas aleatoriamente. Os três casos produzem padrões globalmente diferentes, mas com a mesma estrutura estatística, ou seja, alta entropia e baixa autocorrelação, confirmando que a aleatoriedade emergente é uma propriedade intrínseca da regra e não um artefato da semente inicial.

Esta sensibilidade às condições iniciais, aliada ao determinismo da regra, classifica a Regra 30 como um sistema caótico no sentido da teoria clássica: trajetórias vizinhas no espaço de estados divergem exponencialmente, o que constitui o análogo discreto do efeito borboleta dos sistemas dinâmicos contínuos.

### **5.4 Estabilidade do Sistema e Comparação com Outras Regras**

Em contraste com a Regra 30, a Regra 90 exibe forte autocorrelação de longo alcance (estrutura fractal) e entropia consideravelmente menor, pois a operação XOR impõe uma correlação binária precisa entre os vizinhos. A Regra 110, apesar de sua complexidade computacional (Turing-completa), apresenta densidade de células ativas e padrões de autocorrelação intermediários, consistentes com a Classe IV de Wolfram, que se situa na fronteira entre ordem e caos, denominada por Langton (1990) como "borda do caos".

A Regra 250, por sua vez, estabiliza rapidamente em um padrão de xadrez com entropia próxima de zero (todos os padrões de vizinhança apresentam distribuição desequilibrada), confirmando seu caráter de Classe II. Estas diferenças qualitativas são imediatamente visíveis nos diagramas espaço-tempo e são corroboradas pelas métricas quantitativas.

## **6. Aplicações Práticas**

### **6.1 Geração de Números Pseudo-Aleatórios**

O uso mais direto da Regra 30 é a geração de sequências pseudo-aleatórias a partir da coluna central do diagrama espaço-tempo. Wolfram incorporou esse mecanismo no sistema Mathematica, onde a função `Random[]` utilizou a Regra 30 como núcleo por mais de vinte anos. A vantagem em relação a geradores lineares congruenciais clássicos

está na ausência de padrões detectáveis de curto período e na resistência a ataques de adivinhação do estado interno.

## **6.2 Pigmentação Biológica**

Padrões em conchas de moluscos como a *Conus textile* apresentam estrutura visual notavelmente similar ao diagrama espaço-tempo da Regra 30. Wolfram (2002) argumenta que o mecanismo de pigmentação celular na borda de crescimento da concha pode ser modelado por um autômato celular local, no qual a regra de desenvolvimento de cada célula pigmentar depende apenas do estado das células adjacentes no passo de crescimento anterior.

## **6.3 Simulação de Sistemas Físicos e Biológicos**

Além das aplicações diretamente derivadas da Regra 30, os ACEs têm sido aplicados à modelagem de propagação de incêndios florestais, crescimento de cristais, fenômenos de magnetismo (modelo de Ising), fluxo turbulento e padrões de quebra em materiais. A universalidade computacional da Regra 110 abre perspectivas para o projeto de computadores massivamente paralelos onde a lógica de processamento é implementada via colisões entre estruturas localizadas.

## **7. Conclusão**

Este trabalho demonstrou computacionalmente que regras de atualização locais de extrema simplicidade, no caso uma função de apenas três bits, são suficientes para gerar comportamento caótico de alta complexidade estatística. A Regra 30 de Wolfram, em particular, exibe entropia próxima ao máximo teórico (85%), balanço binário quase ideal e ausência de autocorrelação espacial, tornando-a adequada como fonte de aleatoriedade em aplicações de engenharia e ciência da computação.

A comparação entre as quatro regras canônicas (30, 90, 110 e 250) ilustra de forma concreta as quatro classes de complexidade propostas por Wolfram, desde o comportamento periódico trivial até a complexidade computacional universal. Estas observações reforçam a tese central de que a complexidade não requer causas complexas, pois ela emerge naturalmente de sistemas simples e determinísticos.

Como trabalhos futuros, propõe-se (a) a extensão para autômatos celulares bidimensionais (Game of Life e variantes); (b) a análise formal do período e da profundidade computacional das 256 regras elementares; (c) testes estatísticos padronizados (NIST SP 800-22) da qualidade da sequência pseudo-aleatória gerada pela Regra 30; e (d) a implementação de autômatos celulares contínuos para modelagem de sistemas de reação-difusão, como os padrões de Turing.

## **Referências Bibliográficas**

Cook, M. (2004). Universality in elementary cellular automata. *Complex Systems*, 15(1), 1–40.

Langton, C. G. (1990). Computation at the edge of chaos. *Physica D: Nonlinear Phenomena*, 42(1–3), 12–37.

Princeton University COS 126 (2002). *Lecture P4: Cellular Automata*. Computer Science Department.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27(3), 379–423.

von Neumann, J. (1966). *Theory of Self-Reproducing Automata*. University of Illinois Press.

Wolfram, S. (1984). Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1–2), 1–35.

Wolfram, S. (2002). *A New Kind of Science*. Wolfram Media, Champaign, IL.