

# Três Experimentos com Atualização Assíncrona por Prioridade de Vizinhança em Autômatos Celulares Elementares

Marcelo Vironda Rozanti

Felipe Stefanelli de Aguiar Silva

Prof. Dr. Pedro Paulo Balbi de Oliveira

Faculdade de Computação e Informática - Universidade Presbiteriana Mackenzie

São Paulo, SP – Brasil

**Abstract.** *Cellular Automata are discrete computational systems that have proved useful as general models of complexity and representations of dynamics on a variety of scientific fields. These systems are essentially computed using local functions and can even be implemented in physical structures. Many of them can compute functions and solve algorithmic problems. The present project attempts to explore a fundamental subset of them, called Elementary Cellular Automata with a specific kind of neighbourhood-priority-based asynchronous updating in the hopes of finding either number-conserving, density-classifying or parity-solving rules. A implementação desenvolvida está disponível em <https://github.com/mvrozanti/TCC>.*

**Resumo.** *Autômatos Celulares são sistemas computacionais discretos que se têm provado úteis como modelos genéricos de complexidade e representação de diversas dinâmicas em uma variedade de áreas científicas. Estes sistemas são essencialmente computados usando funções locais e podem até ser implementados em estruturas físicas. Muitos deles podem computar funções e resolver problemas algorítmicos. O presente projeto explora um conjunto fundamental deles, chamados Autômatos Celulares Elementares através de um tipo específico de atualização assíncrona, baseada em prioridade da vizinhança, à procura de regras que satisfaçam conservabilidade numérica, e que resolvam os problemas de decisão de determinação de maioria e de paridade. The implementation developed is hosted at <https://github.com/mvrozanti/TCC>.*

## 1. Introdução

Autômatos Celulares (ACs) são uma categoria de sistemas discretos. Por sua própria simplicidade, estes sistemas têm ocupado uma posição privilegiada no estudo de complexidade nos mais diversos campos da ciência, de biologia teórica a economia, entre muitos outros. O desenvolvimento de ACs é tipicamente atribuído a John von Neumann através de suas tentativas de criar um modelo abstrato de autoreprodução biológica. Ao fim de 1950, foi notado que ACs poderiam ser vistos como computadores paralelos [Wolfram 2002, p. 876]. Apesar de falta de investigação científica até 1970, um exemplo de AC se tornou muito famoso por seu comportamento complexo e regras simplesmente

descritas inventadas por John Conway, *The Game of Life*, que se popularizou após sua aparição na revista *Scientific American*.

Desde então, múltiplos estudos compreensivos foram realizados por cientistas ao redor do mundo, destacando os trabalhos feitos por Stephen Wolfram na década de 1980, culminando na publicação do livro *A New Kind of Science* em que Wolfram apresenta uma coleção de resultados a respeito de ACs, com uma série de descobertas revolucionárias. Esta é uma pesquisa exploratória, dada a natureza desconhecida das propriedades de Autômatos Celulares Elementares com o tipo de atualização mencionado na subseção 2.3. Baseada em medidas objetivas e dados verificáveis, esta pesquisa foi voltada para encontrar caminhos, formas, maneiras e procedimentos para atingir um determinado fim, buscando definir um processo ou uma ferramenta que leve à solução do problema proposto. Quanto aos meios, foram utilizados os recursos mencionados na Bibliografia.

A seguir, são apresentados conceitos e experimentos relacionados a Autômatos Celulares Elementares. Apesar de tradicionalmente executados sincronamente, neste trabalho para todas as execuções é usada uma nova estratégia assíncrona de atualização.

### **1.1. Contextualização e Relevância**

A motivação inicial para o estudo de ACs por parte de von Neumann não era matemática mas tinha como diretriz encontrar uma forma viável de tratar do problema de como fazer máquinas se reproduzirem. Hoje é sabido que esses sistemas são aplicáveis para resolver problemas relativos a classificação de densidade, simulação de partículas, compressão de dados, design estrutural, representação de comportamentos dinâmicos como trânsito em cidades ou movimentação de seres vivos entre muitos outros.

Há uma infinidade de ACs passíveis de construção para modelagem ou solução de problemas. Um deles é a modelagem de tráfego de veículos, como é notado por [Rosenblueth and Gershenson 2011, p. 307]: “Autômatos Celulares Elementares”, discutidos nas próximas seções, “podem modelar tráfego também. Podemos presumir que um veículo se move para frente ou não dependendo apenas na presença ou ausência de outro veículo em sua frente. Supondo que um veículo se move uma célula para a direita se e apenas se uma a célula à direita está vazia, então a regra 184 corresponde ao tráfego se movendo à direita.”

ACs são também capazes de resolver problemas globais de decisão. Dois exemplos emblemáticos são o Problema da Maioria, também chamado de Problema da classificação de densidade e o Problema da paridade. Ambos problemas dependem que as funções locais preservem a informação sobre a vizinhança ao longo do processamento global, de modo que a evolução eventualmente gere um reticulado totalmente preenchido de células com um único estado (no contexto de Autômatos Celulares Elementares, 0 ou 1).

### **1.2. Objetivos do Estudo**

A motivação para o presente trabalho é estudar três importantes propriedades nos ACEs executados com a estratégia definida na subseção 2.3:

- Conservabilidade numérica;
- Capacidade de classificação de densidade;
- Capacidade de resolver o problema da paridade.

O primeiro objetivo trata-se de verificar se a regra 184, que é conservativa em modo síncrono, se mantém conservativa em algum esquema de atualização assíncrono, conforme a definição em de conservabilidade detalhada na subseção 2.4.

Considerando que é fato conhecido que nenhum AC síncrono é capaz de classificar densidade perfeitamente, o objetivo do segundo experimento foi verificar se algum ACE poderia adquirir essa capacidade em modo assíncrono.

É também um fato conhecido que o problema da paridade, detalhado na subseção 2.6, pode ser resolvido por ACEs executados em modo síncrono. O objetivo do terceiro experimento foi verificar se ACEs executados com a estratégia descrita na subseção 2.3 ainda seriam capazes de resolver este problema e em quais esquemas de atualização assíncrona por prioridade da vizinhança.

Três objetivos instrumentais fundamentais para a conclusão deste trabalho foram:

- Definir AC e ACE;
- Definir formalmente a estratégia de atualização;
- Definir e identificar a propriedade e problemas estudados;

## 2. Referencial Teórico

Em relação ao referencial teórico estudado neste TCC, foi de suma importância entender os seguintes tópicos:

### 2.1. Autômatos Celulares

Como diz [Wolfram 1983], “Sistemas matemáticos de simples construção são capazes de comportamento altamente complexo com muitas características”. ACs são um exemplo desse tipo de sistemas. ACs são compostos por células, cada uma com um estado. Eles podem ter  $d$  dimensões e  $q$  estados finitos, os quais devem ser estipulados previamente e são necessariamente computados através de funções locais, isto é, através da vizinhança entre as células.

Wolfram diz ainda, “Autômatos celulares são simples idealizações matemáticas de sistemas naturais. Eles consistem de um reticulado de locais idênticos discretos, com cada local tomando um conjunto finito de valores inteiros. Os valores dos locais evoluem em fases temporais”, também chamadas de *time steps*, “discretas de acordo com regras determinísticas que especificam o valor de cada local em termos dos valores dos locais adjacentes”. É importante notar que todo AC depende de uma *configuração inicial*, que é um reticulado de necessariamente  $d$ -dimensões com cada célula com um estado pertencendo ao conjunto de  $q$  estados.

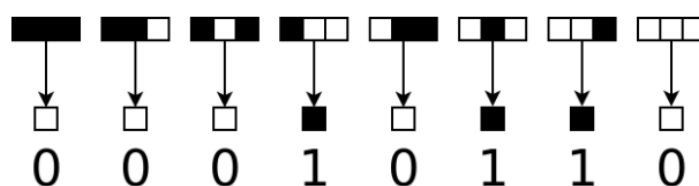
### 2.2. Autômatos Celulares Elementares

ACEs formam a família de ACs com  $d=1$ ,  $q=2$  e  $r=1$ , ou seja, são unidimensionais, com células binárias e vizinhanças definidas como 3 células contíguas. O reticulado é necessariamente ordenado e é dito ter comprimento  $n$ , isto é, tem  $n$  células em cada reticulado da evolução e permanece constante até o fim das evoluções.

Assim como ACs, as regras de transição de um ACE definem o novo estado de uma determinada célula em função de sua vizinhança. O comportamento da aplicação das

regras de transição nas bordas do reticulado pode ser especificado de várias formas, muitas vezes como conexo com a borda oposta, tornando o reticulado efetivamente circular e contínuo, que é o tipo estudado neste trabalho.

Uma definição direta das regras de transição consiste na enumeração de todos os estados possíveis da própria célula, das células vizinhas e do próximo estado, contendo a informação necessária para computar o estado futuro das células em função de sua vizinhança e, conseqüentemente, da evolução como um todo. A regra do AC corresponde a uma função local, formada por transições de estado. A ação dela sobre cada célula de uma configuração pode ser vista como resultado da ação de uma função global sobre o reticulado, que é distinta da função local.



**Figura 1. Ilustração das regras de transição para a regra 22 no código Wolfram.**

Os ACEs são enumerados por suas regras de transição, de 0 a 255, o que é conhecido como número de Wolfram. Segundo [Wolfram 2002, p. 24], “A linha superior em cada caixa dá uma das possíveis combinações de cores para uma célula e sua vizinhança imediata. A linha inferior então especifica qual será a cor da célula central no próximo passo em cada um desses casos.”

Uma definição direta de regra de transição consiste na enumeração de todas as combinações de estado possíveis da vizinhança de uma célula, e do próximo estado associado a elas, o que define a função local a ser usada na evolução temporal do reticulado.

### **2.3. Atualização Assíncrona por Prioridade da Vizinhança**

Parte imprescindível da definição de um AC é a ordem de atualização das células no reticulado. Apesar de os ACs serem fundamentalmente concebidos com o comportamento de atualização síncrona, é possível se utilizar das mais diversas e criativas estratégias de atualização.

No presente trabalho foi empregada uma nova estratégia de atualização: por "prioridade da vizinhança", em que os esquemas associados contém a prioridade da atualização de cada uma das 8 possíveis vizinhanças do espaço elementar, as quais deverão ser seguidas deterministicamente durante as evoluções temporais.

Esse tipo de atualização requer uma distinção das iterações globais e locais (microiterações): as prioridades de vizinhanças são iteradas sequencialmente de menor a maior e para cada prioridade, o reticulado é varrido e atualizam-se apenas as células que tenham vizinhança com a prioridade atual, mantendo-se o estado das vizinhanças que não estejam na prioridade atual da iteração. Ao fim da varredura do reticulado, o processo se repete para a próxima prioridade, agora iterando sobre o reticulado gerado no processo descrito anteriormente, com uma prioridade menor. O reticulado resultante após iterar sobre todas as prioridades é chamado de iteração global (ou macroiteração) e é incluído

nas renderizações, o que não acontece com as microiterações.

Esse modo de atualização inclui também a atualização síncrona, que pode ser simulada quando as prioridades do esquema têm mesmo valor (11111111, 22222222, ..., etc). As evoluções podem ser identificadas como um par  $\langle R, E \rangle$ , onde  $R$  é a regra e  $E$  é um esquema de prioridade.

É importante notar que diferentes esquemas de prioridade podem ser equivalentes: quando esquemas de prioridade equivalentes são aplicados a uma mesma regra, são geradas evoluções iguais. Um exemplo entre esquemas equivalentes pode ser entre os esquemas 11111112 e 11111113, em que as prioridades da última transição são nominalmente diferentes, mas efetivamente iguais. Outro caso é a equivalência entre os oito esquemas síncronos: 11111111, 22222222, 33333333, etc. Para experimentos feitos neste trabalho, foram filtrados esses dois casos, de modo que não houvesse redundância nas execuções, reduzindo o custo computacional.

Além desses dois tipos de equivalência entre esquemas, é possível haver equivalência entre esquemas apenas quando são pareados a uma determinada regra, em função de suas transições não-ativas, como é o caso do que acontece no primeiro experimento, discutido na seção 4.

## 2.4. Conservabilidade Numérica

Segundo [Boccara and Fukś 2002], “Uma regra de AC  $f$  de  $q$  estados, unidimensional, de comprimento  $n$  é conservativa se, para todas as configurações cíclicas de tamanho  $L \geq n$  satisfaz”:

$$f(x_1, x_2, \dots, x_{n-1}, x_n) + f(x_2, x_3, \dots, x_n, x_{n+1}) + \dots \\ + f(x_L, x_1, \dots, x_{n-2}, x_{n-1} = x_1 + x_2 + \dots + x_L$$

No contexto de ACEs, uma evolução é dita conservativa caso, para qualquer configuração inicial, a soma dos estados presentes numa configuração é sempre a mesma para todas as iterações. Como é mencionado na subseção 1.1, ACs conservativos podem ser usados para modelar ambientes em que há conservabilidade, como trânsito automobilístico, simulação de partículas, entre outros. A regra 184 é uma regra que, em execução síncrona, gera evoluções conservativas pela definição anterior de conservabilidade.

## 2.5. Classificação de Densidade

Segundo [Capcarrere et al. 1996, p. 1], “um exemplo dessa computação emergente é utilizar um AC para determinar a densidade global de bits em uma configuração de estado inicial. Conhecido como o problema de classificação de densidade, o AC unidimensional de dois estados é apresentado com uma configuração inicial arbitrária e deve convergir no tempo para um estado de apenas 1s se a configuração inicial contiver uma densidade de 1s  $> 0,5$ , e de apenas 0s se essa densidade  $< 0,5$ ; para uma densidade inicial de 0,5, o comportamento do AC é indefinido”. Em outras palavras, capacidade de classificação de densidade se dá quando o reticulado cíclico de um ACE torna-se totalmente preenchido pelo estado predominante na configuração inicial.

## 2.6. Problema da Paridade

Segundo [Betel et al. 2013, p. 1], “Consideramos o problema de paridade em autômatos celulares unidimensionais, binários e circulares: se a configuração inicial contém um número ímpar de 1s, a reticulado inteira deve convergir para 1s; caso contrário, deve convergir para 0s”. Assim como o problema da maioria, “é fácil ver que o problema é mal-definido para reticulados de tamanho ímpar (que, por definição, nunca seriam capazes de convergir para 1)”. É possível entender, então, que a solução do problema da paridade se dá quando o reticulado cíclico de um ACE torna-se totalmente preenchido pelo estado de quantidade ímpar na configuração inicial.

## 3. Metodologia da Pesquisa

No que tange à Metodologia empregada neste TCC, o trabalho teve início com uma revisão da literatura específica sobre o tema da pesquisa. Esta pesquisa abrange conceitos fundamentais de teoria da computação e o “estado da arte” em termos de análise de conservabilidade.

Este alicerce teórico foi obtido através de autores como: Wolfram; Boccara; entre outros, além de pesquisas em periódicos científicos, sites, publicações em empresas, teses e dissertações em universidades e publicações de associações técnicas. A leitura, análise e comparação da fundamentação teórica tiveram início com ACEs síncronos, modificados em seguida para viabilizar o tipo de atualização mencionado na subseção 2.3.

A capacidade de classificação de densidade e a capacidade de resolução do problema da paridade foram medidas através da contagem de configurações iniciais em que cada par  $\langle R, E \rangle$  resolve o problema em questão.

### 3.1. Organização e Delimitação do Estudo

A pesquisa se limita a ACEs com o tipo de atualização evidenciado na subseção 2.2, com comprimento de reticulados testados até  $n=15$  e com  $r=1$ .

Para o primeiro experimento, foram testados esquemas de atualização *independentes* para o ACE 184, i.e., os esquemas cujas dinâmicas são definidas pelas transições *ativas* de estado da regra, as que mudam o estado da célula central.

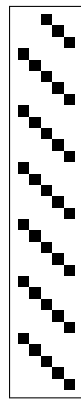
Os três experimentos foram de natureza computacional. Foram testados comprimentos de reticulado ( $n$ ) sucessivamente maiores, de  $n=5$  em diante. Para o primeiro experimento, foram testados esquemas de atualização *independentes* para o ACE 184, i.e., os esquemas cujas dinâmicas são definidas pelas transições *ativas* de estado da regra, as que mudam o estado da célula central. Para o segundo e o terceiro experimento, foram escolhidas regras *balanceadas* (as que têm quatro transições de estado para cada um dos estados, 0 e 1) e que tivessem no máximo 75 esquemas independentes por regra; o balanceamento se deve a uma condição necessária do problema e a restrição em 75 esquemas para diminuir as demandas computacionais.

## 4. Resultados dos Experimentos

### Experimento 1

É sabido que há cinco regras que apresentam conservabilidade numérica em suas evoluções: 170, 184, 204, 226 e 240. A motivação para o primeiro experimento foi descobrir se, no esquema de atualização usado, essa propriedade é mantida para a regra 184 (que é equivalente à regra 226), que foi escolhida por ser a mais rica do ponto de vista dinâmico dentre as cinco, empatada com a 226.

Em relação à conservabilidade, a execução da regra 184 com todas as configurações iniciais de comprimento até  $n=15$ , e com todos os seus esquemas de prioridade independentes, resultou num conjunto de esquemas assíncronos em que a regra permanece conservativa, como é mostrado na figura 2. Nota-se, entretanto, que todos correspondem ao modo síncrono, já que eles só diferem entre si nas prioridades das transições inativas, e as prioridades das ativas são as mesmas em cada esquema.



**Figura 2. Todos os esquemas assíncronos conservativos da regra 184 são equivalentes.**

Regra	Esquema
184	<u>3</u> 2222221
	<u>3</u> 1111112
	<u>2</u> 3333331
	<u>2</u> 1111113
	<u>1</u> 3333332
	<u>1</u> 2222223
	<u>2</u> 2222221
	<u>2</u> 1111112
	<u>2</u> 1111111
	<u>1</u> 2222222
	<u>1</u> 2222221
	<u>1</u> 1111112
	<u>1</u> 1111111

As prioridades com underscore significam que qualquer prioridade pode ser colocada em seu lugar. Isto se deve a, nesses índices, as transições de estado serem não-ativas e, portanto, serem irrelevantes em que ordem são atualizadas. No entanto, é possível ver que as prioridades das transições de estado ativas são iguais entre si, que são todas representações equivalentes da execução síncrona.

### Experimentos 2 e 3

A motivação para os dois últimos experimentos a respeito do problema da maioria e problema da paridade foi estudar se os ACEs com atualização assíncrona por prioridade eram capazes de resolver esses problemas de decisão emblemáticos no contexto de ACs.

A execução das regras e esquemas de prioridade independentes selecionadas até comprimento  $n=7$  bastou para mostrar que nenhuma das regras elementares testadas foi capaz de classificar a densidade nem resolver o problema da paridade totalmente, qualquer que fosse o esquema de atualização independente utilizado.

A seguir são apresentadas as regras a quantidade de esquemas testados para elas no presente trabalho.

**Tabela 1. Primeiro conjunto de regras e quantidade de esquemas testados**

Regra	Qtd. Esquemas
29	75
30	75
45	75
46	75
60	75
71	75
75	75
77	3
78	3
85	75
86	75
89	75
90	75
92	3
101	75
102	75
105	75
106	75
108	3
116	75
120	75
135	75
139	75
141	3
142	3
149	75
150	75
153	75
154	75
156	3
165	75
166	75
169	75
170	75
172	3
180	75
184	75
195	75
197	3
198	3
201	3
...	...



...	...
202	3
204	1
209	75
210	75
212	3
216	3
225	75
226	75
228	3
232	3

**Tabela 2. Segundo conjunto de regras e quantidade de esquemas testados**

Regra	Qtd. Esquemas
23	4683
27	4683
39	4683
43	4683
53	4683
54	4683
57	4683
58	4683
83	4683
99	4683
113	4683
114	4683
147	4683
163	4683
177	4683
178	4683

## 5. Considerações Finais

Esta pesquisa é um passo no estudo sobre ACEs com atualização assíncrona por prioridade da vizinhança. Para tanto, realizamos os três experimentos acima descritos.

O primeiro experimento mostrou que a regra 184 permanece conservativa apenas em esquemas de atualização equivalentes ao modo síncrono. Já o segundo experimento revelou que, para os esquemas de atualização testados, não havia pares  $\langle R, E \rangle$  que resolvessem o problema da maioria. Por fim, o terceiro experimento demonstrou que não havia pares  $\langle R, E \rangle$  que resolvessem o problema da paridade.

É importante apontar que as conclusões mostradas aqui não incluíram todos os esquemas de atualização possíveis pelo alto custo computacional e que é necessário notar que esta estratégia de atualização é recente e ainda há muitos testes a serem feitos, seja para identificar propriedades estáticas ou verificar a capacidade de resolver problemas de decisão.

A única regra que não foi analisada no contexto dos problemas de decisão discutidos foi a regra 51 (chamada regra inversora, que muda 0s para 1s e vice-versa), e que contém a máxima quantidade possível de esquemas de atualização independentes (545.835 esquemas).

Estudamos três importantes propriedades nos ACEs executados com a estratégia de atualização assíncrona por prioridade da vizinhança: conservabilidade numérica, capacidade de classificação de densidade e capacidade de resolver o problema da maioria.

## Referências

- Betel, H., de Oliveira, P. P., and Flocchini, P. (2013). On the parity problem in one-dimensional cellular automata. *Natural Computing*, 12(3):323-337.
- Boccara, N. and Fukś, H. (2002). Number-conserving cellular automaton rules. *Fundamenta Informaticae*, 52(1-3):1-13.
- Capcarrere, M. S., Sipper, M., and Tomassini, M. (1996). Two-state,  $r=1$  cellular automaton that classifies density. *Physical review letters*, 77(24):4969.
- Rosenblueth, D. A. and Gershenson, C. (2011). A model of city traffic based on elementary cellular automata. *Complex Systems*, 19(4):305.
- Wolfram, S. (1983). Cellular automata. *Los Alamos Science*, 9.
- Wolfram, S. (2002). *A new kind of science*, volume 5. Wolfram media Champaign, IL.