

ESTUDO EXPLORATÓRIO DAS PROPRIEDADES GLOBAIS DE AUTÔMATOS CELULARES COM ASSINCRONIA POR PRIORIDADE DE VIZINHANÇA

Ana Cecília Barbosa Alves e Eurico Luiz Prospero Ruivo (Orientador)

Resumo

Os Autômatos celulares (ACs) são sistemas evolutivos baseados em regras simples, formados por uma rede de células, um grid. Cada célula ocupa uma posição na rede e possui um determinado estado inicial que é alterado de acordo com as regras e com o estado das células vizinhas. Tem despertado grande interesse devido à capacidade de simular comportamentos auto-organizados complexos que são observados em sistemas naturais. Os estudos sobre Autômatos Celulares ganharam importância principalmente pelo sucesso em criar simulações e previsões de eventos naturais e comportamentos dos sistemas vivos. (GREMONINI; VICENTINI, 2008). Atualmente, a assincronia dos ACs é determinada a partir da sua posição, o que resulta na mesma evolução temporal, além da quantidade de atualizações dessemelhantes aumentar combinatoriamente com o tamanho do reticulado. Esse estudo analisa através da assincronia por prioridade de vizinhança como propriedades globais se comportam quando passamos a considerar esse tipo de assincronia, e também em relação aos autômatos celulares síncronos.

Palavras chaves: Autômatos Celulares, assincronia, propriedades globais, prioridade por vizinhança.

Abstract

Cellular Automata (CAs) are evolutionary systems based on simple rules, formed by a network of cells, a grid. Each cell occupies a position in the network and has a certain initial state that changes according to the rules and the state of neighboring cells. It has aroused great interest due to its ability to simulate complex self-organized behaviors that are observed in natural systems. Cellular Automata studies gained importance mainly due to their success in creating simulations and predictions of natural events and behavior of living systems. (GREMONINI; VICENTINI, 2008). Currently, the asynchrony of ACs is determined from their position, which results in the same temporal evolution, in addition to the amount of dissimilar updates increasing combinatorially with the size of the lattice. This study analyzes, through neighborhood priority asynchrony, how global properties behave when we consider this type of asynchrony, and also in relation to synchronous cellular automata.

Keywords: Cellular Automata, asynchrony, global properties, neighborhood priority.

1 Introdução

1.1 Problema de pesquisa

Os Autômatos celulares (ACs) (KARI, 2005; WOLFRAM, 2002) são sistemas evolutivos baseados em regras simples com comportamentos complexos. São formados por unidades denominadas células que conectadas formam um reticulado que está sujeito a uma regra (ou função) que determina o próximo estado de cada unidade considerando o seu estado atual e das células vizinhas.

O comportamento mais observado até o momento em autômatos celulares é o de atualização síncrona sobre as células. Isto é, as células recebem os estados de seus vizinhos e atualizam simultaneamente. No entanto, ao longo da última década, a atenção está se voltando cada vez mais à possibilidade de atualização assíncrona, ou seja, cada célula tem uma prioridade associada, a qual pode ser definida determinística ou estocástica (FATÈS, 2014).

Do ponto de vista computacional, algumas propriedades dinâmicas das regras de autômatos celulares dependem mais do sincronismo que da regra em si (BERSINI; DETOURS, 1994; FATÈS; MORVAN, 2005), levando à questão de quais podem ser as novas propriedades de uma regra quando são considerados esquemas de atualização assíncronos (VIELHABER, 2013). É certo, as atualizações assíncrona tem o potencial de reduzir a complexidade de uma regra usada para resolver um problema inicialmente solucionado de maneira síncrona (RUIVO; OLIVEIRA, 2019), como também a resolução de questionamentos até então não resolvidos por meio da atualização síncrona (RUIVO; OLIVEIRA; PERROT, 2019).

A assincronia determinística em autômatos celulares é definida a partir da localização da célula no gride, o que não permite a generalização de um esquema de atualização para estruturas mais gerais, na qual é desconsiderado a posição. Nesse estudo é proposto uma nova forma de definir a assincronia deterministicamente: atribuindo uma prioridade de atualização para cada possível vizinhança de uma célula.

Um autômato assíncrono celular é capaz de atualizar células individuais de forma independente, de tal forma que o novo estado de uma célula afeta o cálculo de estados em células vizinhas.

1.2 Justificativa

O processo de definir assincronia determinística em autômatos celulares depende da posição que cada célula ocupa no reticulado. O que resulta em atualizações semelhantes ao longo de extensas evoluções temporais, além da quantidade diferente de atualizações que

cresce proporcionalmente com o tamanho do reticulado. Esse estudo analisa através da assincronia por prioridade de vizinhança como propriedades globais se comportam quando passamos a considerar esse tipo de assincronia, e em comparação com autômatos celulares síncronos, com perspectiva para novos resultados teóricos, e potencial para servir como um novo paradigma de modelo de sistemas complexos naturais.

1.3 Objetivo

O questionamento principal em relação aos autômatos celulares é definir os problemas que podem ser solucionados e quais comportamentos são alcançados tendo como base as interações e trocas entre as células (MONTALVA-MEDEL; OLIVEIRA; GOLES, 2018). Ainda que de maneira generalista, é conhecido que uma solução depende de como o problema foi elaborado, o que é equivalente a determinar as restrições que um sistema de autômatos celulares deve cumprir, bem como determinar suas capacidades.

Ainda que esse tema seja amplamente estudado em Acs síncronos e com assincronia determinística definida por posição, o que propõe esse estudo consiste em outra perspectiva sobre esse assunto.

Buscou-se estudar como propriedades globais tais como conservabilidade, reversibilidade e bacias de atração se comportam quando passamos a considerar esse novo tipo de assincronia, tanto em relação à assincronia definida por meio de posição quanto em relação aos autômatos celulares síncronos. Em paralelo, foi desenvolvido outro projeto também sobre assincronia por vizinhança no entanto com um viés diferente: complexidade e riqueza dinâmica.

2 Referencial teórico

Desde os primeiros estudos de Neumann (1966) envolvendo autômatos celulares no contexto evoluções temporais, eles têm sido usados para simular fenômenos reais, como propagação de doenças (MIKLER; VENKATACHALAM; ABBAS, 2005), dinâmica dos fluidos (WOLF- GLADROW, 2000) e modelos de crescimento urbano (ABURAS et al, 2016), quanto sob o ponto de vista de suas propriedades matemático-computacionais (MITCHELL, 1998; WOLFRAM, 1984). Wolfram (2002) propõe que qualquer fenômeno global pode ser explicado por e simulado a partir de estruturas simples que interagem localmente, o que inclui os ACs.

É possível observar que até em autômatos celulares elementares (ACEs), ACs com conjunto de estados binário e comportamento localmente definido por funções que atuam sobre conjuntos de três bits, há regras, como a Regra 110 e suas equivalentes dinâmicas, capazes de simular uma máquina de Turing universal (COOK, 2004). Além delas, o AC

conhecido como “Conway’s Game of Life” (GARDNER, 1970), uma regra binária bidimensional definida localmente com base em nove entradas binárias (a célula central de uma matriz e suas vizinhas mais próximas), também apresenta computabilidade universal.

Ainda que algumas regras apresentem computabilidade universal, a codificação do problema e a sua resolução produzida por regras necessitam de acesso global ao conjunto de células denominado configuração. Dessa forma, uma questão que surge é: quais tarefas podem ser feitas por ACs considerando apenas o acesso e as atividades locais?

É certo que algumas propriedades dinâmicas de alguns ACs são resultados da atualização síncrona e não puramente da regra de interação local entre as células (VIELHABER, 2013). É possível visualizar esse fato na figura abaixo: a Regra 110 do espaço elementar que com atualização síncrona (esquerda) apresenta computabilidade universal (COOK, 2004) e é capaz de formar padrões complexos, deixa de ter esse comportamento quando passa a ter uma atualização totalmente assíncrona (direita) onde as células são atualizadas sequencialmente uma por vez.

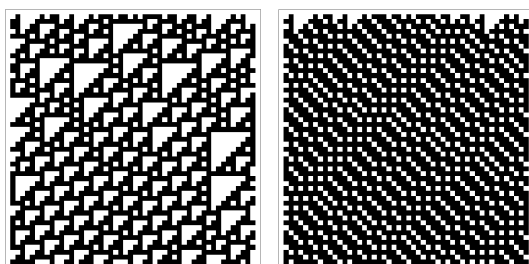


Figura 1: Evoluções temporais da Regra 110 do espaço elementar com atualização síncrona (esquerda) e totalmente assíncrona (direita). A dinâmica e sua complexidade podem mudar completamente quando atualizações assíncronas são consideradas.

Os esquemas de atualização assíncronos determinísticos são definidos pela prioridade de atualização de cada célula relativa à localização (ARACENA et al., 2011, 2013). Ou seja, para determinar um esquema de atualização assíncrona, é preciso saber a quantidade de células no reticulado. A quantidade de esquemas de atualização não equivalentes cresce pelo menos exponencialmente com a quantidade de células (ARACENA et al., 2011) e a maioria deles não podem ser generalizados para um reticulado de tamanho arbitrário. Esse tipo de assincronia será denominada assincronia posicional (AP).

É possível notar que em AP, a ordem de atualização das células no gride é constante ao longo da evolução temporal. Esse projeto então se propõe a buscar um novo modelo para assincronia para autômatos celulares, no qual a localização da célula é desprezível para o estado dela e o da vizinhança passa ser o que define essa prioridade. Esse tipo de assincronia será denominada assincronia por vizinhança (AV).

Formalmente, para definir que um esquema de atualização é assíncrono por vizinhança, é necessário detalhar a prioridade de atualização de cada possível vizinhança de uma célula, tornando essa prioridade relacionada à informação disponível para cada célula em vez da sua localização.

A partir de um reticulado, uma assincronia por vizinhança(AV) se difere de assincronia posicional (AP) nos seguintes aspectos:

- A ordem atualização de cada célula pode variar ao longo da evolução temporal, uma vez que as vizinhanças de cada célula podem mudar;
- É mais autônomo no sentido que é definida em termos de uma propriedade dinâmica (estados na vizinhança) em vez de uma propriedade estática (posição no reticulado);
- Um esquema de atualização com AV sempre pode ser generalizado para qualquer quantidade de células em um reticulado, uma vez que o tamanho da vizinhança é fixo;
- Um esquema de atualização com AV é totalmente definido por meio de uma propriedade local e não depende do padrão de conexão entre as células, podendo ser estudado, sob alguns aspectos, sem levar em conta o comportamento global.

3 Procedimentos metodológicos

Apesar da simplicidade estrutural dos autômatos celulares, entender como seus comportamentos globais emergem tem sido uma tarefa dificilmente alcançada apenas por meios analíticos. Portanto, a pesquisa nesse tópico se baseia tanto em métodos computacionais e de simulação quanto em métodos matemáticos formais. Tal perspectiva conjunta é o fundamento dos métodos deste projeto. A seguir detalhamos os métodos que foram aplicados às atividades descritas na Seção 4.

Quanto aos experimentos computacionais, o recurso fundamental foi o software Wolfram Mathematica, em particular sua Wolfram language e a biblioteca de funções de autômatos celulares e redes de autômatos CAMaT.nb, que tem sido continuamente desenvolvida por pesquisadores da área.

Experimentos computacionais foram a ferramenta fundamental dos objetivos neste projeto, conforme detalhado na próxima seção.

Importante para o entendimento desse projeto o conceito de vizinhança ativa. Neste trabalho foram considerados ACs elementares, ou seja, com conjunto de estados binário e vizinhança unidimensional de raio 1, isto é, a vizinhança de uma célula é formada por ela própria e por seus vizinhos imediatos à esquerda e à direita.

Dada uma regra local f de um AC elementar e uma vizinhança $(x_1, x_2, x_3) \in \{0, 1\}^3$, tal vizinhança é dita ativa quando $f(x_1, x_2, x_3) = 1 - x_2$, isto é, a vizinhança é dita ativa quando a regra local faz com que sua célula central mude de estado (de 0 para 1 ou de 1 para 0) ao ser iterada.

O conceito de vizinhança ativa está correlacionado com a quantidade de updates distintos por regras, isto é, como a vizinhança ativa interfere no estado do seu vizinho e dependendo da prioridade para que a atualização ocorra são inúmeras as saídas para cada regra. Sendo a vizinhança ativa fator primordial para explicar o alto número de updates para regras com mais vizinhanças ativas.

Exemplo: Uma regra com duas vizinhanças ativas - Regra 12

{1, 1, 1}	0
{1, 1, 0}	0
{1, 0, 1}	0
{1, 0, 0}	0
{0, 1, 1}	1
{0, 1, 0}	1
{0, 0, 1}	0
{0, 0, 0}	0

Possibilidades de atualização:

$\{\{1,1,0,0,0,0,0,0\}, \{1,2,0,0,0,0,0,0\}, \{2,1,0,0,0,0,0,0\}\}$

Figura 2: Na figura é possível visualizar que com duas vizinhanças ativas existem três possibilidades de updates para regra 12.

{1, 1, 1}	0	{1, 0, 1, 0, 0, 0, 1, 0},
{1, 1, 0}	1	{1, 0, 1, 0, 0, 0, 2, 0},
{1, 0, 1}	1	{1, 0, 2, 0, 0, 0, 1, 0},
{1, 0, 0}	0	{1, 0, 2, 0, 0, 0, 2, 0},
{0, 1, 1}	1	{2, 0, 1, 0, 0, 0, 1, 0},
{0, 1, 0}	1	{2, 0, 1, 0, 0, 0, 2, 0},
{0, 0, 1}	1	{2, 0, 2, 0, 0, 0, 1, 0},
{0, 0, 0}	0	{1, 0, 2, 0, 0, 0, 3, 0},
		{1, 0, 3, 0, 0, 0, 2, 0},
		{2, 0, 1, 0, 0, 0, 3, 0},
		{2, 0, 3, 0, 0, 0, 1, 0},
		{3, 0, 1, 0, 0, 0, 2, 0},
		{3, 0, 2, 0, 0, 0, 1, 0}

Figura 3: A esquerda é possível visualizar a quantidade de vizinhanças ativas na Regra 110, e a direita as atualizações possíveis.

$$\left(\begin{array}{cc} \{1, 1, 1\} & 1 \\ \{1, 1, 0\} & 0 \\ \{1, 0, 1\} & 1 \\ \{1, 0, 0\} & 0 \\ \{0, 1, 1\} & 0 \\ \{0, 1, 0\} & 0 \\ \{0, 0, 1\} & 1 \\ \{0, 0, 0\} & 0 \end{array} \right)$$

Figura 4: Vizinhanças ativas na regra 162 - Evolução temporal da regra 162 com o esquema de atualização {0, 1, 1, 0, 2, 1, 1, 0} em um vetor de configuração de tamanho sete qualquer:

Configuração inicial: {0,1,1,0,1,1,0,1}

Tabela 1 : Evolução temporal da regra 162 com update {0, 1, 1, 0, 2, 1, 1, 0}

Configuração Inicial	0	1	1	0	1	1	0	1
Passo 1	1	0	0	1	0	0	1	0
Passo 2	0	0	1	0	0	1	0	0

A leitura de um vetor de atualização é feita considerando que cada número no vetor representa o grau de prioridade de atualização daquela célula. Por exemplo, o número 2 do vetor do exemplo acima significa que o número central da quinta posição é o autômato que primeiro deve ser atualizado, deve-se então visualizar as vizinhanças ativas da figura 17 para verificar se é uma vizinhança é ativa ou não. Em seguida, é necessário ver os autômatos que possuem grau de atualização com prioridade 1, e repetir o processo citado anteriormente e assim por diante, até que todos os autômatos tenham sido atualizados. A principal diferença entre a atualização dos autômatos síncronos e assíncronos é que nos síncronos a atualização ocorre simultaneamente em todos os autômatos, ou seja, não existe uma prioridade de atualização para cada célula.

Inicialmente, foram calculados através da verificação de propriedade e por atualizado por regras com configurações de tamanho sete, a princípio, os esquemas de atualização que conservam a paridade (Soma módulo 2) e a conservação numérica de estados dependendo da quantidade de vizinhanças ativas - alteram o estado das células vizinhas .

Posteriormente, verificou-se que as regras que mantiveram a paridade e os estados

para configurações de tamanho sete, mantêm o comportamento até pelo menos configurações de tamanho vinte.

Criou-se uma função, escrita em Wolfram Language, que permite a execução de autômatos celulares assíncronos com atualizações definidas por prioridade de vizinhanças. De posse dessa função, foi possível criar uma função que recebe uma regra desse tipo, um tamanho de configuração e cria o grafo representando as bacias de atração para essas entradas - função Bacia.

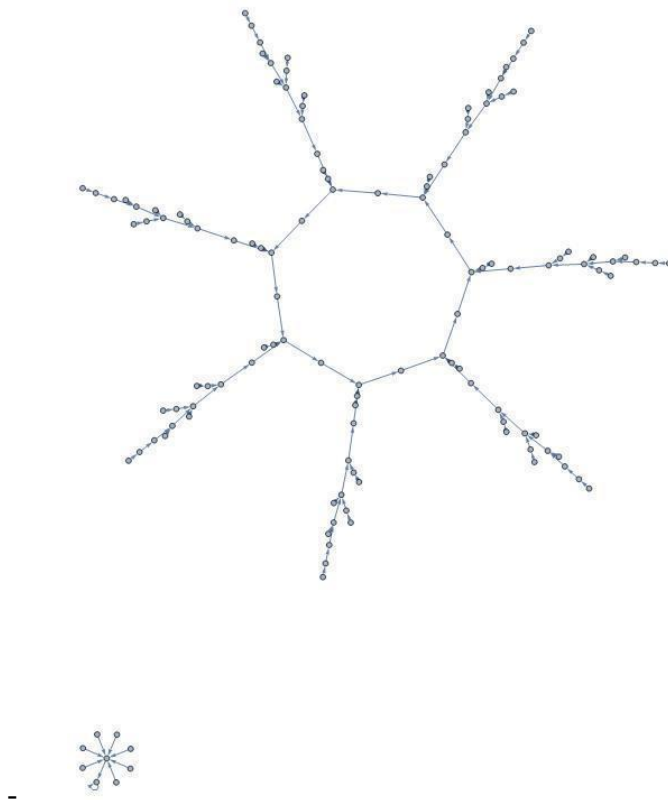


Figura 5: Na imagem é possível observar o resultado da função de Bacia, com a regra 110, com os casos de atualização $\{1,0,2,0,0,3,0\}$ com configurações de tamanho sete.

A bacia de atração pode ser usada então para verificar se vértices numa mesma componente conexa compartilham alguma propriedade. Verificou-se a conservação de paridade e da conservação numérica de estados para todas as regras com até seis vizinhanças ativas. Então foram executados os seguintes passos para calcular informações da bacia de atração a partir dos updates de cada regra :

- Para calcular a quantidade de atratores diferentes por regra foi necessário unir os dados das bacias que eram iguais e em seguida calculou-se o número de bacias de atração diferentes por regras;
- Posteriormente, calculou-se o maior componente da bacia de atração e o menor;
- Por fim, foi calculada a média da quantidade de vértices por componente e o

desvio-padrão da quantidade de vértices por componente, ambos foram gerados a partir de uma média simples.

4 Resultados

Por meio do processo descrito na Seção 3, foi possível verificar que os esquemas que conservam os estados numericamente em todos os casos de atualização também preservam a paridade.

Além de que os casos que tiveram suas propriedades conservadas em regras atuando sobre configurações de tamanho sete conservam também pelo menos até o tamanho vinte.

Tabela 2 : Esquemas por número de vizinhanças ativas por regras que conservam os estados

Vizinhanças Ativas	Regras			
5	162 {0, 1, 1, 0, 2, 1, 1, 0}	176 {0, 2, 1, 1, 1, 1, 0, 0}	186 {0, 1, 1, 2, 0, 1, 1, 0}	242 {0, 0, 1, 1, 1, 1, 2, 0}
6	178 {0, 1, 1, 2, 2, 1, 1, 0}, {{0, 2, 1, 1, 1, 1, 2, 0}}, {{0, 1, 1, 2, 3, 1, 1, 0}}, {{0, 1, 1, 3, 2, 1, 1, 0}}, {{0, 2, 1, 1, 1, 1, 3, 0}}, {{0, 3, 1, 1, 1, 1, 2, 0}}			

Tabela 3: Esquemas por número de vizinhanças ativas por regras que conservam a paridade

Vizinhanças Ativas	Regras			
5	162 {0, 1, 1, 0, 2, 1, 1, 0}	176 {0, 2, 1, 1, 1, 1, 0, 0}	186 {0, 1, 1, 2, 0, 1, 1, 0}	242 {0, 0, 1, 1, 1, 1, 2, 0}
6	178 {{0, 1, 1, 2, 2, 1, 1, 0}}, {{0, 2, 1, 1, 1, 1, 2, 0}}, {{0, 1, 1, 2, 3, 1, 1, 0}}, {{0, 1, 1, 3, 2, 1, 1, 0}}, {{0, 2, 1, 1, 1, 1, 3, 0}}, {{0, 3, 1, 1, 1, 1, 2, 0}}			

A partir dos resultados apresentados, é possível observar que as regras que conservam os casos que tiveram suas propriedades conservadas em regras atuando sobre configurações de tamanho sete conservam também pelo menos até o tamanho vinte, além de que é possível concluir que as regras que conservam os estados numericamente e paridade

possuem os mesmos casos de atualizações.

Os esquemas da tabela 1 e 2 são considerados shifts, como pode ser observado abaixo:

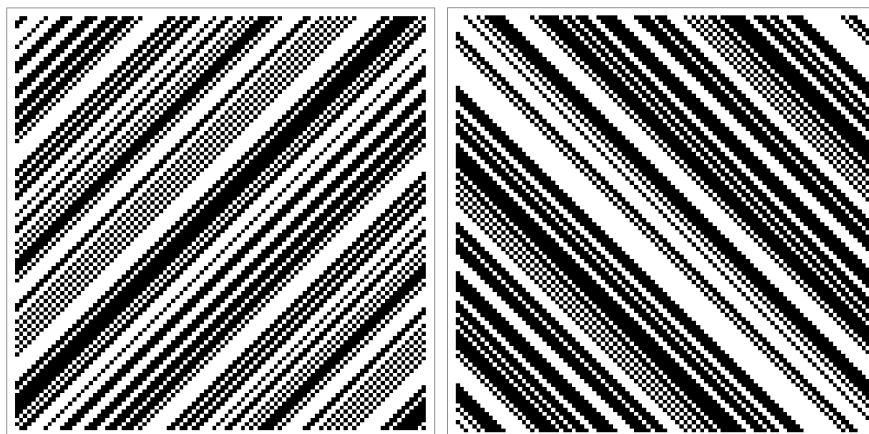


Figura 6 e 7: Evolução temporal das regras 162 $\text{update}\{0, 1, 1, 0, 2, 1, 1, 0\}$ - a esquerda - e 176 $\text{update}\{0, 2, 1, 1, 1, 1, 0, 0\}$ - na direita.

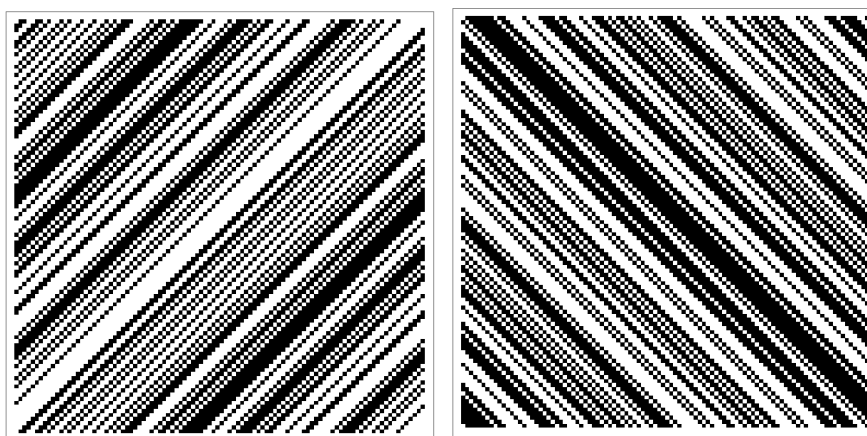


Figura 8 e 9 : Evolução temporal das regras 186 $\text{update}\{0, 1, 1, 2, 0, 1, 1, 0\}$ - a esquerda - e 242 $\text{update}\{0, 0, 1, 1, 1, 1, 2, 0\}$ - na direita.

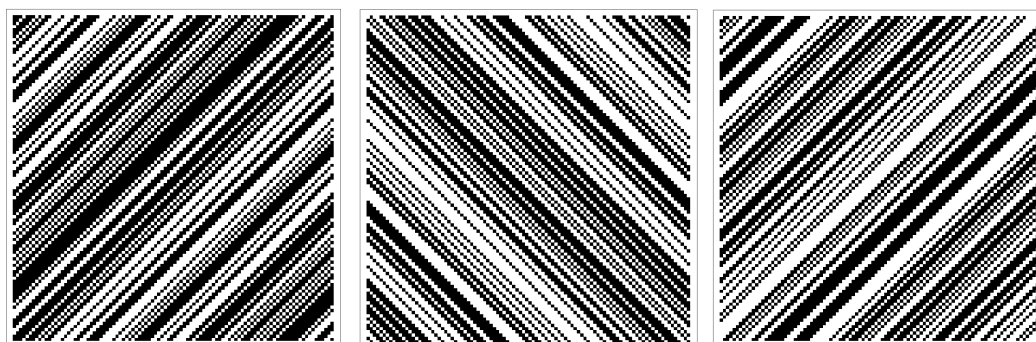




Figura 10, 11, 12, 13, 14 e 15: Evolução temporal das regras 178 update
 $\{0, 1, 1, 2, 2, 1, 1, 0\}$, $\{0, 2, 1, 1, 1, 1, 2, 0\}$, $\{0, 1, 1, 2, 3, 1, 1, 0\}$, $\{0, 1, 1, 3, 2, 1, 1, 0\}$,
 $\{0, 2, 1, 1, 1, 1, 3, 0\}$, $\{0, 3, 1, 1, 1, 1, 2, 0\}$ - da esquerda para direita -.

Os dados da tabela a seguir são referentes às medidas de bacia de atração das regras de 1 a 255, que revelam quantidade de variação, primeira coluna, quanto a regra fica mais rica - riqueza, se os componentes: se estão bem distribuídos, isto é, se um comportamento está sendo mais típico que outro, e a partir dela nesse estudo não foi possível concluir se existe um padrão ou não entre as regras.

Tabela 4: Dados sobre bacia de atração das regras de 1 a 255***

Regra	A quantidade de bacias de atração diferentes por regra.	Tamanho da maior componente	Tamanho da menor componente	A média da quantidade de vértices por componente	O desvio-padrão da quantidade de vértices por componente
1	5	15	17	46.9372	67.8723
2	45	22	10	39.3882	61.7989
3	112	29	6	21.3256	56.2445
4	7	29	6	1.92308	4.41379
5	4	29	9	2.6	14.36
6	20	36	5	10.7733	30.5101

7	52	36	5	2.67098	27.6864
8	1	1	128	128	128
9	3	8	17	21.4	35.84
10	9	6	35	15.8267	42.6951
11	15	6	35	15.8706	57.3792
12	2	29	6	2	4.41379
13	1	8	18	2	16
14	2	5	70	2	29.5385
15	4	10	14	2	41.472
16	45	22	10	39.3882	61.7989
17	112	29	6	21.3256	56.2445
18	180	23	14	15.2473	36.7904
19					
20	20	36	5	10.7733	30.5101
21	52	36	5	2.67098	27.6864
22	71	30	7	3.73567	21.2755
23	104	36	6	3.21162	24.7614
24	14	22	10	46.3333	68.9164
25	43	29	6	11.366	33.0036
26	50	15	12	11.8965	45.6695

27	111	29	8	11.3474	48.3983
28	6	36	5	2	9.8561
29	7	36	5	2	13.0043
30	22	22	9	2.45333	23.5623
31	52	36	5	2.67098	27.6864
32	9	16	58	102.61	107.063
33	32	29	11	25.8495	56.6851
34	236	36	7	33.7527	54.5624
35					
36	21	30	7	2.10667	7.40548
37	17	29	9	3.62292	38.0847
38	61	36	7	10.3974	34.6581
39	111	29	8	11.3474	48.3983
40	7	16	58	90.04	97.3298
41	17	17	11	11.1423	34.5035
42	67	15	14	14.9723	41.1788
43	57	11	14	16.3335	54.9525
44	9	30	7	2.23077	6.86035
45	7	17	12	3.62667	35.6735
46	9	6	70	4.89333	38.2578

47	15	6	35	15.8706	57.3792
48	236	36	7	33.7527	54.5624
49					
50					
51					
52	61	36	7	10.3974	34.6581
53	111	29	8	11.3474	48.3983
54	169	30	9	6.99915	30.7643
55					
56	79	36	7	34.8115	56.4288
57	86	31	6	12.9379	34.5974
58	197	18	11	14.5159	46.6432
59					
60	21	36	8	6	30.2364
61	43	29	6	11.366	33.0036
62	45	18	11	9.90203	46.2963
63	112	29	6	21.3256	56.2445
64	1	1	128	128	128
65	3	8	17	21.4	35.84
66	14	22	10	46.3333	68.9164

67	43	29	6	11.366	33.0036
68	2	29	6	2	4.41379
69	1	8	18	2	16
70	6	36	5	2	9.8561
71	7	36	5	2	13.0043
72	1	15	79	1	8.53333
73	1	22	44	1	5.81818
74	3	4	77	20.8462	56.6154
75	7	17	12	3.62667	35.6735
76					
77	1	29	8	1	4.41379
78	1	8	18	2	16
79	1	8	18	2	16
80	9	6	35	15.8267	42.6951
81	15	6	35	15.8706	57.3792
82	50	15	12	11.8965	45.6695
83	111	29	8	11.3474	48.3983
84	2	5	70	2	29.5385
85	4	10	14	2	41.472
86	22	22	9	2.45333	23.5623

87	52	36	5	2.67098	27.6864
88	3	4	77	20.8462	56.6154
89	7	17	12	3.62667	35.6735
90	6	10	14	18.7067	58.752
91	17	29	9	3.62292	38.0847
92	1	8	18	2	16
93	1	8	18	2	16
94	2	22	9	2.46154	13.6503
95	4	29	9	2.6	14.36
96	7	16	58	90.04	97.3298
97	17	17	11	11.1423	34.5035
98	79	36	7	34.8115	56.4288
99	86	31	6	12.9379	34.5974
100	9	30	7	2.23077	6.86035
101	7	17	12	3.62667	35.6735
102	21	36	8	6	30.2364
103	43	29	6	11.366	33.0036
104	3	22	72	3.15385	12.0839
105	2	15	14	2	9.27289
106	8	6	84	18.5733	54.6133

107	17	17	11	11.1423	34.5035
108	1	43	9	1	2.97674
109	1	22	44	1	5.81818
110	1	2	119	9	64
111	3	8	17	21.4	35.84
112	67	15	14	14.9723	41.1488
113	57	11	14	16.3335	54.9525
114	197	18	11	14.5159	46.6432
115					
116	9	6	70	4.89333	38.2578
117	15	6	35	15.8706	57.3792
118	45	18	11	9.90203	46.2963
119	112	29	6	21.3256	56.2445
120	8	6	84	18.5733	54.6133
121	17	17	11	11.1423	34.5035
122	10	6	84	27.4713	68.2193
123	32	29	11	25.8495	56.6851
124	1	2	119	9	64
125	3	8	17	21.4	35.84
126	2	2	119	29.6267	75.0933

127	5	15	17	46.9372	67.8723
128	1	2	127	1	64
129	2	2	119	29.6267	75.0933
130	21	23	12	1	33.3801
131	45	18	11	9.90203	46.2963
132	2	30	8	1	4.26667
133	2	22	9	2.46154	13.6503
134	9	37	6	1	20.0726
135	22	22	9	2.45333	23.5623
136	1	2	127	1	64
137	1	2	119	9	64
138	3	9	28	1	18.5201
139	9	6	70	4.89333	38.2578
140					
141	1	8	18	2	16
142	1	8	70	1	16
143	2	5	70	2	29.5385
144	21	23	12	1	33.3801
145	45	18	11	9.90203	46.2963
146	79	24	21	1.07948	19.9981

147	169	30	9	6.99915	30.7643
148	9	37	6	1	20.0726
149	22	22	9	2.45333	23.5623
150	13	31	7	1	13.0297
151	71	30	7	3.73567	21.2755
152	7	23	12	1	31.5989
153	21	36	8	6	30.2364
154	30	23	10	1.46667	21.904
155	61	36	7	10.3974	34.6581
156	2	37	6	1	7.04705
157	6	36	5	2	9.8561
158	9	37	6	1	20.0726
159	20	36	5	10.7733	30.5101
160	6	6	99	7.53333	59.4489
161	10	6	84	24.4713	68.2193
162	124	23	7	3.41774	30.8768
163	197	18	11	14.5159	46.6432
164	7	18	10	1	7.86325
165	6	10	14	18.7067	58.752
166	30	23	10	1.46667	21.904

167	50	15	12	11.8965	45.6695
168	6	6	99	10.6923	55.7949
169	8	6	84	18.5733	54.6133
170	15	20	7	2.32	18.1918
171	67	15	14	14.9723	56.6154
172	3	18	12	1	7.7037
173	3	4	77	20.8462	56.6154
174	3	9	28	1	18.5201
175	9	6	35	15.8267	42.6951
176	124	23	7	3.41774	30.8768
177	197	18	11	14.5159	46.6432
178	201	25	7	2.58766	23.4256
179					
180	30	23	10	1.46667	21.904
181	50	15	12	11.8965	45.6695
182	79	24	21	1.07948	19.9981
183	180	23	14	15.2473	36.7904
184	19	23	12	5.88	41.6734
185	79	36	7	34.8115	56.4288
186	124	23	7	3.41774	30.8768

187	236	36	7	33.7527	56.5624
188	7	23	12	1	31.5989
189	14	22	10	46.3333	68.9164
190	21	23	12	1	33.3801
191	45	22	10	39.3882	61.7989
192	1	2	127	1	64
193	1	2	119	9	64
194	7	23	12	1	31.5989
195	21	36	8	6	30.2364
196					
197	1	8	18	2	16
198	2	37	6	1	7.04705
199	6	36	5	2	9.8561
200					
201	1	43	9	1	2.97674
202	3	18	12	1	7.7037
203	9	30	7	2.23077	6.86035
204					
205					
206					

207	2	29	6	2	4.41379
208	3	9	28	1	18.5201
209	9	6	70	4.89333	38.2578
210	30	23	10	1.46667	21.904
211	61	36	7	10.3974	34.6581
212	1	8	70	1	16
213	2	5	70	2	29.5385
214	9	37	6	1	20.0726
215	20	36	5	10.7733	30.5101
216	3	18	12	1	7.7037
217	9	30	7	2.23077	6.86035
218	7	18	10	1	7.86325
219	21	30	7	2.10667	7.40548
220					
221	2	29	6	2	4.41379
222	2	30	8	1	4.26667
223	7	29	6	1.92308	4.41379
224	6	6	99	10.6923	55.7949
225	8	6	84	18.5733	54.6133
226	19	23	12	5.88	41.6734

227	79	36	7	34.8115	56.4288
228	3	18	12	1	7.7037
229	3	4	77	20.8462	56.6154
230	7	23	12	1	31.5989
231	14	22	10	46.3333	68.9164
232	2	30	29	2	4.26667
233	3	22	72	3.15385	12.0839
234	4	6	99	10.6923	55.7949
235	7	16	58	90.04	97.3298
236					
237	1	15	79	1	8.53333
238	1	2	127	1	64
239	1	1	128	128	129
240	15	20	7	2.32	18.1918
241	61	15	14	14.9723	41.1788
242	124	23	7	3.41774	30.8768
243	236	36	7	33.7527	54.5624
244	3	9	28	1	18.5201
245	9	6	35	15.8267	42.6951
246	21	23	12	1	33.3801

247	45	22	10	39.3882	61.7989
248	6	6	99	10.6923	55.7949
249	7	16	58	90.04	97.3298
250	6	6	99	7.53333	59.4489
251	9	16	58	102.61	107.063
252	1	2	127	1	64
253	1	1	128	128	128
254	1	2	127	1	64
255	1	1	128	128	128

*** A tabela possui regras sem medidas, pois nesse estudo não foi possível executá-las.

5 Considerações Finais

A partir dos dados da tabela 2 e 3 não é possível encontrar nenhuma nova regra a partir dos estudos já existentes que conservam a paridade e os estados numericamente, sendo apenas shifts.

A tabela 4 apresentada na seção de resultados finais foi obtida com um intuito de fornecer dados a serem explorados em trabalhos futuros, a exemplo do que foi feito em (DE OLIVEIRA ET AL., 2016).

A variedade e os padrões observados nesta pesquisa permitem entender o potencial que ACs assíncronos possuem, pois a partir de um sistema simples é possível melhorar a maneira e/ou transformar com que problemas do mundo contemporâneo são resolvidos.

6 Agradecimentos

Agradeço imensamente ao professor e orientador Eurico Ruivo, que me auxiliou e orientou no desenvolvimento deste projeto e possibilitou a mim a experiência da iniciação científica.

Quero também agradecer aos colegas com quem trabalhei durante a execução dessa pesquisa, ao PIBIC Mackenzie pelo apoio através da bolsa de iniciação científica e à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo projeto STIC-AmSud

7 Referências

- ABURAS, M. M. et al. The simulation and prediction of spatio-temporal urban growth trends using cellular automata models: A review. *International Journal of Applied Earth Observation and Geoinformation*, Elsevier, v. 52, p. 380–389, 2016.
- ARACENA, J. et al. On the number of update digraphs and its relation with the feedback arc sets and tournaments. *Discrete Applied Mathematics*, Elsevier, v. 161, n. 10, p. 1345–1355, 2013.
- ARACENA, J. et al. Combinatorics on update digraphs in Boolean networks. *Discrete Applied Mathematics*, v. 159, n. 6, p. 401–409, 2011.
- BERSINI, H.; DETOURS, V. Asynchrony induces stability in cellular automata based models. In: MIT PRESS, MA. *Artificial Life IV*. [S.l.], 1994. p. 382–387.
- COOK, M. Universality in elementary cellular automata. *Complex Systems*, v. 15, n. 1, p. 1–40, 2004.
- FATÈS, N. A guided tour of asynchronous cellular automata. *Journal of Cellular Automata*, v. 9, n. 5-6, p. 387–416, 2014.
- FATÈS, N. A.; MORVAN, M. An experimental study of robustness to asynchronism for elementary cellular automata. *Complex Systems*, v. 16, n. 1, p. 1–27, 2005.
- GARDNER, M. Mathematical games: The fantastic combinations of John Conway's new solitaire game "Life". *Scientific American*, v. 223, n. 4, p. 120–123, 1970.
- HOEKSTRA, A.; KROC, J.; SLOOT, P. Introduction to modeling of complex systems using cellular automata. *Simulating Complex Systems by Cellular Automata*, Springer, p. 1–16, 2010.
- KARI, J. Theory of cellular automata: A survey. *Theoretical Computer Science*, Elsevier, v. 334, n. 1–3, p. 3–33, 2005.
- LANGTON, C. Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, Elsevier, v. 42, n. 1-3, p. 12–37, 1990.
- MIKLER, A. R.; VENKATACHALAM, S.; ABBAS, K. Modeling infectious diseases using global stochastic cellular automata. *Journal of Biological Systems*, World Scientific, v. 13, n. 04, p. 421–439, 2005.
- MITCHELL, M. Computation in cellular automata : A selected review lattice. *NonStandard Computation*, p. 1–42, 1998.
- MONTALVA-MEDEL, M.; OLIVEIRA, P. P. de; GOLES, E. A portfolio of classification problems

by one-dimensional cellular automata, over cyclic binary configurations and parallel update. *Natural Computing*, Springer, v. 17, n. 3, p. 663–671, 2018.

NEUMANN, J. V. Theory of self-reproducing automata. *IEEE Transactions on Neural Networks*, v. 21, n. 1, p. 3–14, 1966.

RUIVO, E. L. P.; OLIVEIRA, P. P. B. de; PERROT, K. An asynchronous solution to the synchronisation problem for binary one-dimensional cellular automata. Submitted manuscript, 2019.

RUIVO, E. L. P.; OLIVEIRA, P. P. de. A perfect solution to the parity problem with elementary cellular automaton 150 under asynchronous update. *Information Sciences*, v. 493, p. 138–151, 2019. ISSN 0020-0255.

VIELHABER, M. Computation of functions on n bits by asynchronous clocking of cellular automata. *Natural Computing*, Springer, v. 12, n. 3, p. 307–322, 2013.

WOLF-GLADROW, D. A. *Lattice-Gas Cellular Automata and Lattice Boltzmann Models*. [S.I.]: Springer, 2000.

WOLFRAM, S. Computation theory of cellular automata. *Communications in Mathematical Physics*, v. 96, n. 1, p. 15–57, 1984.

WOLFRAM, S. *A New Kind of Science*. [S.I.]: Wolfram Media, 2002.

DE OLIVEIRA, P. P., Ruivo, E. L., Costa, W. L., Miki, F. T., & Trafaniuc, V. V. (2016). Advances in the study of elementary cellular automata regular language complexity. *Complexity*, 21(6), 267-279.

WOLFRAM, S. (2202). *A New Kind of Science*. Disponível em: <https://www.wolframscience.com/nks/> Acesso em : 27 de Maio 2021.

DE OLIVEIRA, P. P. B. CAMAT.nb - Cellular automata package for Mathematica, Software de distribuição restrita, disponibilizada pelo autor, 2018.

Contatos: ana-cecilia09@live.com e eurico.ruivo@mackenzie.br