

AVALIAÇÃO DA RIQUEZA DINÂMICA DE AUTÔMATOS CELULARES COM ASSINCRONIA POR PRIORIDADE DE VIZINHANÇA

Bernardo Siqueira Esteves dos Reis (IC) e Eurico Luiz Próspero Ruivo (Orientador)

Apoio: PIBIC Mackpesquisa

RESUMO

Os Autômatos Celulares (ACs) têm sido estudados não apenas pelo interesse matemático e computacional, mas também pela sua representação em modelos de sistemas complexos do mundo real (HOEKSTRA; KROC; SLOOT, 2010), como por exemplo na proliferação de doenças (MIKLER; VENKATACHALAM; ABBAS, 2005), dinâmica de fluídos (WOLF-GLADROW, 2000) e o crescimento urbano (ABURAS et al., 2016). Porém seus maiores estudos foram feitos com base no comportamento síncrono de suas células, portanto este projeto utiliza como base a atualização assíncrona dos ACs, que pode tanto reduzir a complexidade de uma regra, quanto solucionar problemas mais rapidamente e problemas que não podiam ser resolvidos. Este estudo analisa a forma de atualização por assincronia de prioridade de vizinhança, para avaliar a riqueza dinâmica dos ACs e observar como esta dinâmica é classificada nos modelos definidos por Wolfram (2002) e Li e Packard (1990).

Palavras-chave: Autômatos Celulares, assincronia, prioridade de vizinhança

ABSTRACT

Cellular Automata (CAs) have been studied not only for mathematical and computational interest, but also for their representation in real-world complex systems models (HOEKSTRA; KROC; SLOOT, 2010), such as in the proliferation of diseases (MIKLER; VENKATACHALAM; ABBAS, 2005), fluid dynamics (WOLF-GLADROW, 2000) and urban growth (ABURAS et al., 2016). However, most of their studies were based on the synchronous behavior of their cells, therefore this project uses as a basis the asynchronous update of CAs, which can either reduce the complexity of a rule, or solve problems faster than before and problems that could not be solved. This study analyzes the form of updating by neighborhood priority asynchrony to assess the dynamic richness of CAs and observe how this dynamic is classified in the models defined by Wolfram (2002) and Li and Packard (1990).

Keywords: Cellular Automata, asynchrony, neighborhood priority

1 INTRODUÇÃO

1.1 Problema de pesquisa

Nesta pesquisa tratamos dos Autômatos celulares (ACs) (KARI, 2005; WOLFRAM, 2002), que são estruturas dinâmicas localmente definidas, discretas no tempo, espaço e estado. São compostos por unidades chamadas de células, que são conectadas em uma malha regular (denominada reticulado) e sujeitas a uma função (ou regra) local que determina o próximo estado de cada unidade com base em seu estado atual e no estado das células vizinhas a ela.

Naturalmente, a regra que define o comportamento dinâmico das estruturas atua de forma síncrona sobre as células dos autômatos. Ou seja, as células leem as informações dos estados das células vizinhas e simultaneamente são atualizadas. Todavia, uma nova forma de atualizar as células têm ganhado atenção ao longo da última década, este é a atualização de assincronia por prioridade de vizinhança. Nesta nova forma, cada célula tem uma prioridade associada, a qual pode ser definida determinística ou de modo estocástico (FATÈS, 2014)

De forma computacional, certas propriedades dinâmicas das regras dos ACs são mais dependentes da sincronia do que da regra (BERSINI; DETOURS, 1994; FATÈS; MORVAN, 2005), levantando a pergunta se uma regra irá adquirir novas prioridades ao considerarmos esquemas de atualização assíncronos (VIELHABER, 2013). Sabe-se que é possível utilizar a assincronia para reduzir a complexidade de uma regra necessária para resolver problemas previamente resolvidos de forma síncrona, (RUIVO; OLIVEIRA, 2019), bem como eventualmente permitir a solução de problemas que não podiam ser resolvidos por meio de atualização síncrona das células (RUIVO; OLIVEIRA; PERROT, 2019).

Para os autômatos celulares a assincronia determinística é definida através da posição da célula no reticulado, impedindo a generalização da atualização para sistemas mais genéricos que não possuem uma caracterização da posição, que dificulta o crescimento de um esquema de atualização de configurações iniciais grandes. Neste projeto, será analisada a riqueza dinâmica do espaço de ACs elementares sob uma nova forma de definir a assincronia deterministicamente: atribuindo uma prioridade de atualização para cada possível vizinhança de uma célula.

1.2 Justificativa

A forma padrão para definir a assincronia determinística nos autômatos celulares é dependente das posições das células no retículo, isto é o que mantém suas atualizações da

mesma forma ao longo de sua evolução no sistema. Esta dependência também leva em conta o fato que a quantidade de atualizações iniciais distintas cresce combinatoriamente com o tamanho do retículo.

1.3 Objetivo

Uma das dificuldades no estudo dos autômatos celulares é identificar quais os problemas do mundo real podem ser resolvidos e quais comportamentos globais podem ser alcançados através da troca e atualização local entre as células (MONTALVA-MEDEL; OLIVEIRA; GOLES, 2018). Embora seja muito difícil de construir um algoritmo que resolveria o problema de forma geral, a solução depende da formulação deste, o que no sistema dos ACs equivale a determinar as restrições e capacidades que eles devam cumprir.

Mesmo que esse é um tema bem explorado para ambos os autômatos celulares síncronos e para os com assincronia determinística definida por posição, a assincronia proposta neste trabalho consiste em outra visão para esta dificuldade nos estudos fundamentais.

Mais especificamente, estudou-se como a riqueza dinâmica em termos das classes definidas por Wolfram (2002) e Li e Packard (1990) se comporta no espaço dos autômatos celulares elementares quando passamos a considerar esse novo tipo de assincronia em relação aos ACs síncronos.

2 REFERENCIAL TEÓRICO

Acredita-se que desde os primeiros estudos sobre os autômatos celulares que eles podem explicar e modelar qualquer fenômeno do mundo real a partir de suas simples estruturas que agem localmente com suas propriedades matemático-computacionais. Isto pode ser comprovado como dito anteriormente pelo uso deles na simulação da propagação de doenças, dinâmica de fluidos e crescimento urbano

Um exemplo de suas capacidades matemático-computacionais podemos ver que há regras de casos mais simples como o dos autômatos celulares elementares (ACEs), que agem sobre um conjunto de apenas três bits, um exemplo disto é a Regra 110 e suas equivalentes dinâmicas, capazes de simular uma máquina de Turing universal (COOK, 2004). Outra regra que possui uma computabilidade universal é a regra de AC conhecida como “Conway’s Game of Life” (GARDNER, 1970), que age em uma base de nove bits.

Duas das questões colocadas em (WOLFRAM, 1985) sobre a dinâmica típica dos ACs são as seguintes, em tradução livre: “Que classificação geral sobre o comportamento dos autômatos celulares pode ser feita?” e “Como estão distribuídos os diferentes comportamentos ao longo de um espaço de regras de autômatos celulares?”. Wolfram (1985) respondeu parcialmente às duas questões fornecendo um esquema de classificação geral e explorando como a quantidade de regras em cada uma dessas classes varia de acordo com o espaço delas considerado.

Mais especificamente, a classificação dada por Wolfram (1984) utiliza a observação experimental das evoluções temporais típicas de cada regra e as separa em quatro classes de ACs:

- Classe I (C I): configurações pertencentes a evoluções temporais de regras desta classe tipicamente tornam-se homogêneas após um certo número de iterações. Ou seja, após um tempo de evolução todas as células de uma configuração atingem o mesmo estado de 0 ou 1.;
- Classe II (C II): evoluções temporais típicas de regras desta classe atingem um ponto-fixa ou um regime periódico após um transiente. Em outras palavras, quando é atingido a estabilidade da atualização a regra mantém um padrão evolutivo regular em sua configuração;
- Classe III (C III): padrões não periódicos e aparentemente caóticos surgem ao longo das evoluções temporais típicas de regras desta classe;
- Classe IV (C IV): regras pertencentes a esta classe geram configurações nas quais surgem tanto regiões periódicas com comportamento bem-definido quanto regiões com comportamento caótico, havendo interação entre tais estruturas. Generalizando seria uma mistura entre ambas as classes II e III.

Conjectura-se (WOLFRAM, 2002) que regras da Classe IV apresentam computabilidade universal. Apresentando tal propriedade, a Regra 110 do espaço dos ACEs é da Classe IV por exemplo.

Li e Packard (1990) forneceram uma partição mais fina dessas regras, subdividindo a Classe II em duas classes. Mais precisamente, de acordo com a classificação de Li-Packard, uma regra pode ser:

- Nula: como as regras da Classe I de Wolfram;
- Ponto-fixo: como as regras da Classe II de Wolfram que evoluem para um regime de ponto-fixo;
- Periódica: como as regras da Classe II de Wolfram que evoluem para um regime periódico;
- Caótica: como as regras da Classe III de Wolfram;
- Complexa: como as regras da Classe IV de Wolfram.

Langton (1990) sugere que há regras com capacidade computacional de armazenamento e processamento de informações em uma região dos espaços de regras denominada beira do caos, localizada entre regras com comportamento ordenado e previsível como de classe II e regras com comportamento aparentemente caótico como a classe III.

Apesar da computabilidade universal ser uma das características de algumas regras, a solução do problema e a leitura da saída produzida por regras como as citadas acima necessitam de um acesso normalizado, chamado de configuração. Portanto é levantada uma questão: quais tarefas podem ser executadas por ACs considerando-se apenas o acesso e atividade locais sobre a configuração?

De fato, certas propriedades dinâmicas de alguns ACs não são efetivamente apenas resultados da atividade local da configuração inicial, mas também resultantes do esquema de atualização síncrono (VIELHABER, 2013). A Figura 1 ilustra esse fato: o espaço elementar da atualização síncrona (esquerda) da Regra 110 apresenta computabilidade universal (COOK, 2004) sendo capaz de formar padrões complexos e caóticos, deixa de ter esse comportamento quando passa a ter uma atualização totalmente assíncrona (direita) onde as células são atualizadas sequencialmente uma por vez.

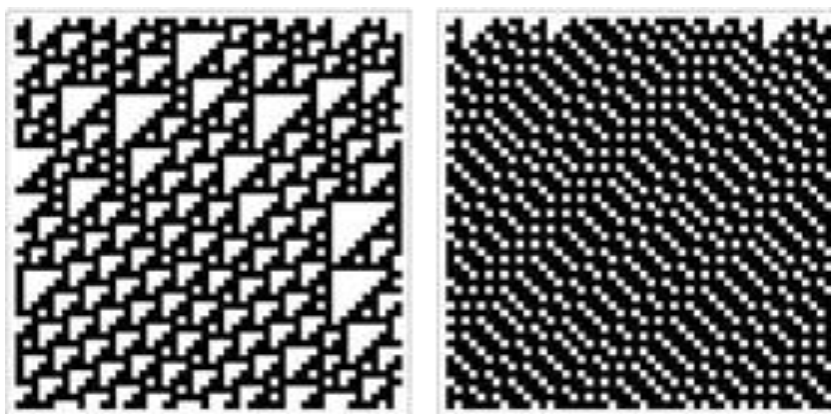


Figura 1: Evoluções temporais da Regra 110 no espaço elementar com atualização síncrona (esquerda) e totalmente assíncrona (direita). A dinâmica e sua complexidade podem mudar completamente quando atualizações assíncronas são consideradas.

Os esquemas de atualização assíncronos determinísticos são definidos a partir da prioridade da atualização de cada regra em relação com sua posição (ARACENA et al., 2011, 2013). Isto é, para especificar um esquema de atualização assíncrona, é necessário antes conhecer a quantidade de células em um reticulado. Esta quantidade também influencia no mínimo exponencialmente na quantidade de esquema de atualizações iniciais assíncronas (ARACENA et al., 2011) sendo cada um deles único, logo, não podendo ser generalizados para um reticulado de tamanho arbitrário. Esse tipo de assincronia será denominado assincronia posicional (AP).

Classicamente, com AP a ordem em que as células são atualizadas no reticulado é fixa ao longo da evolução temporal. O que este projeto propõe é investigar um novo modelo para assincronia para autômatos celulares, um modelo no qual a posição da célula passa a ser irrelevante enquanto o estado dela e de seus vizinhos passa a ser o que define essa prioridade. Esse tipo de assincronia será denominado assincronia por vizinhança (AV).

Mais precisamente, para definir um esquema de atualização com AV, é necessário especificar a prioridade de atualização de cada possível vizinhança de uma célula, fazendo com que essa prioridade esteja relacionada à informação disponível para cada agente (célula) em vez de sua posição.

Dado um reticulado, AV difere de 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 orgânica 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. Diferente do AP, uma vez que este não pode ser generalizado;
- 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.

O modelo de atualização proposto se inspira em duas regras das interações locais: ação e reação, e o fato que interações locais mais rápidas podem inibir ou modificar reações mais lentas.

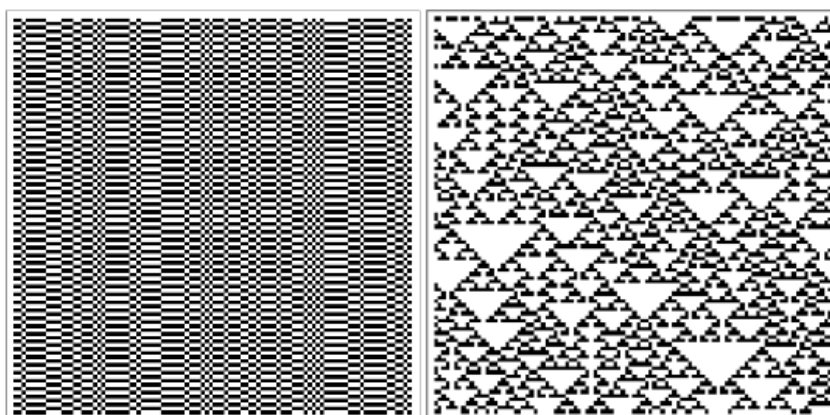


Figura 2: Evoluções temporais da Regra 51 no espaço elementar com atualização síncrona (esquerda) e assíncrona por prioridade de vizinhança (direita). Pode ser visto que uma regra bem periódica síncrona pode se tornar caótica com a assincronia.

Para aprendizado de máquina nesta pesquisa é utilizado o modelo algorítmico de rede neural do Perceptron de Multicamadas (Multilayer Perceptron ou MLP). Esta rede é composta por diversas camadas com determinado número de neurônios (nós), são elas uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Estas camadas são conectadas em um grafo direcionado para prever uma saída por meio da técnica de aprendizado supervisionado chamada Retropropagação de Erro (Backpropagation) que computa o gradiente da função de perda diminuindo os erros da previsão. O MLP foi estudado e feito em conjunto com (PASSOS, 2021).

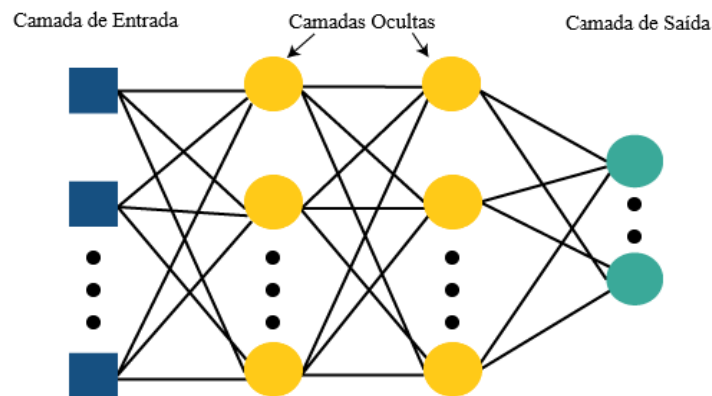


Figura 3: Demonstração de uma rede de MLP genérica com uma camada de entrada, duas camadas ocultas e uma camada de saída, todas com valores indeterminados de nós.

3 METODOLOGIA

Para os experimentos computacionais a seguir, foi utilizado 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. A classificação das regras, a partir de entradas iniciais aleatórias, para serem utilizadas no treino do algoritmo MLP foi feita utilizando o Wolfram Alpha.

Para a análise da progressão dos ACs são puladas uma certa quantidade de linhas, pois em alguns espaços amostrais as regras, geralmente regras de classe 2, precisam de um tempo para se estabilizarem.

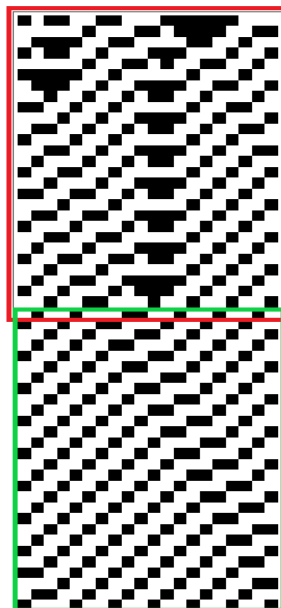


Figura 4: Exemplo de execução da regra 131, em vermelho o trecho transiente que será pulado e em verde o trecho que será analisado

A fim de classificar as regras, é extraído da progressão dos ACs texturas do intervalo de uma configuração das regras dos ACs. Estas são calculadas a partir de um quadrado de tamanho $\underline{d}$ por $\underline{d}$ e deste quadrado é obtido um valor binário (células pretas são representadas por 1 e as brancas como 0) que é convertido para decimal, sendo este o valor da textura.

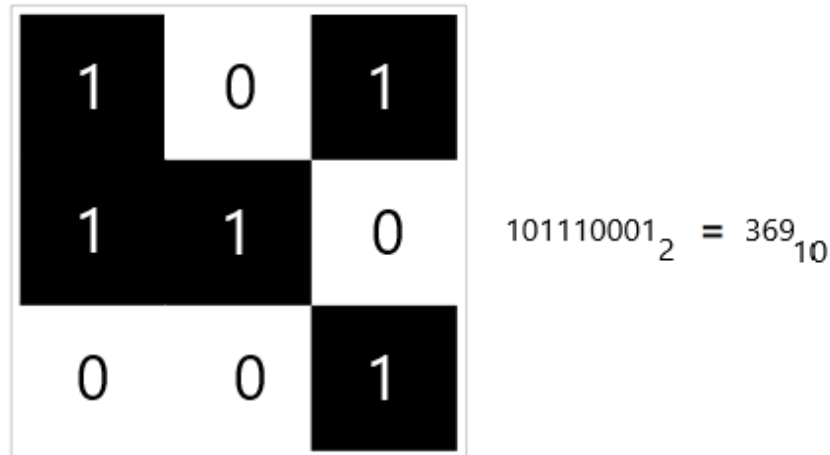


Figura 5: Exemplo de extração de textura com tamanho 3 por 3

O cálculo e contabilização das texturas é feito através de uma função desenvolvida por (PASSOS, 2021), que desloca 1 elemento por vez até o fim da progressão do AC e armazena a quantidade de texturas em um vetor de 512 posições.

A partir desses vetores, foram retiradas 14 características:

- Maior frequência, Menor frequência;
- Maior diferença entre frequências (diferentes de zero);
- Diferença entre a maior frequência e a segunda maior;
- Média desconsiderando zero;
- Mediana desconsiderando zero;
- Maior valor da Lista de Moda (diferente de zero);
- Desvio Padrão;
- Quantidade de frequências diferentes;
- Quantidade de frequências diferentes de zero;
- Menor valor do vetor normalizado diferente de zero;
- Maior valor da moda (diferente de zero) vetor normalizado;
- Menor valor da moda vetor normalizado.

Estas foram extraídas para serem implementadas no treinamento do modelo MLP e utilizadas em (PASSOS, 2021). Utilizamos este algoritmo para prever a classe dos ACs a partir das características do vetor contador. Na construção do MLP foram usadas 5 camadas, uma de input com 14 nós, três ocultas com respectivamente 24, 14 e 7 nós e uma camada de output de 4 nós que representam a probabilidade de cada uma das 4 regras de ACs.

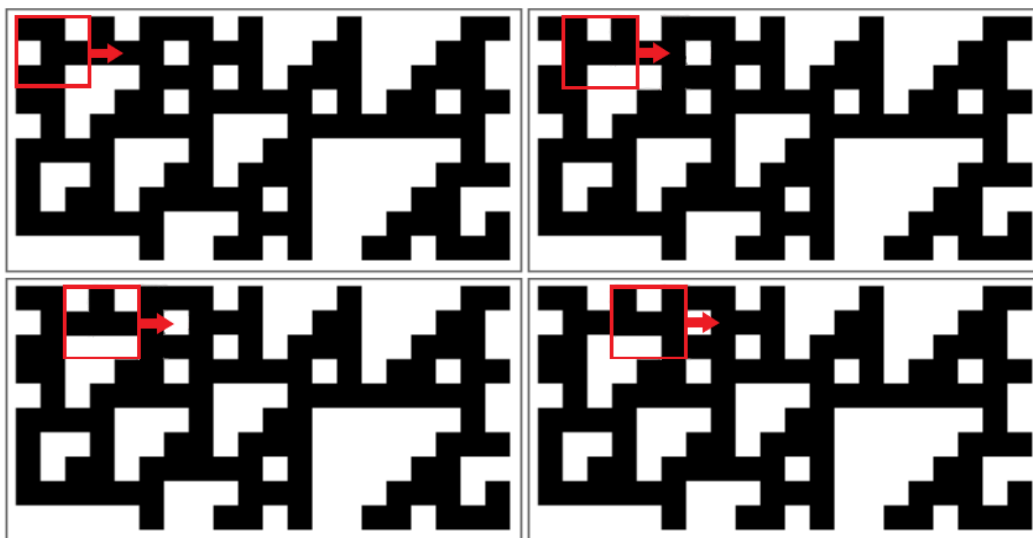


Figura 6: Exemplo da progressão de coleta das texturas.

Para o treino do algoritmo MLP, configuração definida em (PASSOS, 2021), não foram utilizadas todas as regras dos ACs pois são desbalanceadas em número entre as classes, logo para o modelo não ficar enviesado utilizamos 40 regras diferentes, separando 10 regras de cada classe de AC, e para termos mais exemplos de treino criamos 3 exemplos de cada regra, totalizando 120 elementos para o treinamento do modelo preditivo. Foram extraídos da atualização, utilizando entradas aleatórias, de cada um dos exemplos uma textura de tamanho 50 por 50. Após o treino do algoritmo utilizou-se todas as 256 regras para o teste de previsão de classe e suas características de textura. Como resultado, o modelo atingiu 93% de acurácia com margem de erro de 1%.

Os testes anteriores foram feitos com ACs síncronos, portanto, para o próximo estudo utilizou-se os ACs com assincronia por prioridade de vizinhança, para analisar a pertinência do percentual de cada regra a cada classe definida por Wolfram. Para os ACs de caso assíncrono a função criadora dos vetores de textura foi modificada para receber um esquema de atualização assíncrono. Portanto utilizam-se todos os esquemas iniciais de atualização possíveis com assincronia por prioridade de vizinhança de cada regra para criar todos os vetores de características. A partir de sua criação, foi utilizado o modelo de MLP para prever a classe de cada vetor. Para atingir valores mais concisos, foram feitos 10 testes para cada regra e após isto uma média aritmética de cada porcentagem.

4 RESULTADO E DISCUSSÃO

Os resultados das previsões feitas podem ser observados nas tabelas a seguir, de acordo com sua classe síncrona.

A classificação dos resultados na tabela foi feita através de uma paleta de verdes, em verde escuro são as classes com maior peso de porcentagem e em verde claro as classes de menor peso.

Tabela 1. Proporção relativa da Previsão de Classe Assíncrona das regras classe 2

Regra	Classe Síncrona	Classe 1 (%)	Classe 2 (%)	Classe 3 (%)	Classe 4 (%)
1	2	0	54,64	25,99	19,37
2	2	26,99	72,72	0,18	0,11
3	2	0,12	66,47	20,98	12,42
4	2	0	100	0	0
5	2	0	100	0	0
6	2	6,67	92,89	0,27	0,27
7	2	0	96,37	2,55	0,27
9	2	0	49,07	24,26	26,27
10	2	4,53	95,47	0	0
11	2	0,04	77,19	16,08	6,69
12	2	0	100	0	0
13	2	0	100	0	0
14	2	0	100	0	0
15	2	0	99,73	0,27	0
19	2	0,01	73,70	12,91	13,38
23	2	0,01	93,45	2,91	3,63
24	2	33,87	61,69	2,67	1,87
25	2	0,04	69,50	20,63	9,83
26	2	2,69	67,98	22,85	6,47

27	2	0,02	71,52	18,92	9,54
28	2	0	100	0	0
29	2	0	98,93	0	1,07
33	2	0,04	35,68	37,19	27,09
34	2	23,51	71,59	3,88	1,02
35	2	0,02	66,76	19,67	13,55
36	2	1,07	98,93	0	0
37	2	0	44,73	33,46	21,81
38	2	5,29	87,54	3,14	4,03
41	2	0,07	49,17	38,22	12,53
42	2	2,88	96,98	0,07	0,07
43	2	0	60,09	25,75	14,15
44	2	0	100	0	0
46	2	0	93,33	4,00	2,67
50	2	6,51	74,59	10,49	8,41
51	2	0,18	67,66	19,67	12,49
56	2	2,89	96,97	0,07	0,07
57	2	0,02	43,26	30,11	26,61
58	2	1,22	59,96	25,49	13,32
62	2	0,11	69,72	14,49	15,68
72	2	26,67	73,33	0	0
74	2	0	100	0	0
76	2	0	100	0	0
77	2	0	100	0	0
78	2	0	100	0	0
94	2	0	100	0	0
104	2	16,92	83,08	0	0

108	2	0	100	0	0
130	2	25,60	74,13	0	0,27
132	2	0	100	0	0
134	2	7,69	90,77	0	1,54
138	2	0	100	0	0
140	2	0	100	0	0
142	2	0	100	0	0
152	2	30,77	69,23	0	0
154	2	7,57	88,85	2,93	0,85
156	2	0	100	0	0
162	2	25,14	73,90	0,70	0,26
164	2	1,54	98,46	0	0
170	2	1,60	98,40	0	0
172	2	0	100	0	0
178	2	8,89	83,89	4,43	2,79
200	2	0	100	0	0
204	2	0	100	0	0
232	2	0	100	0	0

Tabela 2, Proporção relativa da Previsão de Classe Assíncrona das regras classe 3

Regra	Classe Síncrona	Classe 1 (%)	Classe 2 (%)	Classe 3 (%)	Classe 4 (%)
18	3	7,79	74,54	10,23	7,44
22	3	1,94	85,75	6,75	5,56
30	3	0,13	89,87	4,53	5,47
45	3	0,13	64,93	18,93	15,99
60	3	0,27	70,00	14,13	15,60
73	3	0,77	67,69	13,85	17,69

90	3	1,20	18,27	52,13	28,40
105	3	0,13	38,41	39,33	22,13
106	3	0,67	65,73	17,17	16,53
122	3	0,90	26,17	47,30	25,62
126	3	0,53	40,13	42,81	16,53
146	3	6,59	81,05	7,65	4,71
150	3	1,20	88,13	5,33	5,34

Tabela 3, Proporção relativa da Previsão de Classe Assíncrona das regras classe 4

Regra	Classe Síncrona	Classe 1 (%)	Classe 2 (%)	Classe 3 (%)	Classe 4 (%)
54	4	1,52	72,40	10,97	15,10
110	4	0,02	29,99	32,30	37,69

Podemos observar por meio das tabelas que as classes previstas para os ACs assíncronos são predominantes as de classe 2. Porém as regras 90 e 122 são duas exceções, pois possuem predominância na classe 3, isto ocorre pela natureza aleatória destas. Temos também regras que são muito divididas em relação à porcentagem, como por exemplo 1, 9, 33, 37, 41, 57, 105, 122, 126 e 110.

Nota-se que as regras com uma porcentagem de previsão acima de 90% tendem a possuir menos atualizações possíveis por regra, isto ocorre em 92% dos casos. Mesmo que haja casos como por exemplo a regra 9 que possui apenas 75 esquemas e porcentagens distribuídas quase uniformemente entre as classes, ainda podemos inferir que quanto mais esquemas iniciais por regra, mais chances de as previsões serem mais diversificadas entre as classes.

As próximas figuras demonstram um exemplo de cada classe a partir da evolução do AC de regra 22 com assincronia por propriedade de vizinhança e observa-se que uma mesma regra pode possuir diferentes evoluções e classes de acordo com suas propriedades iniciais.

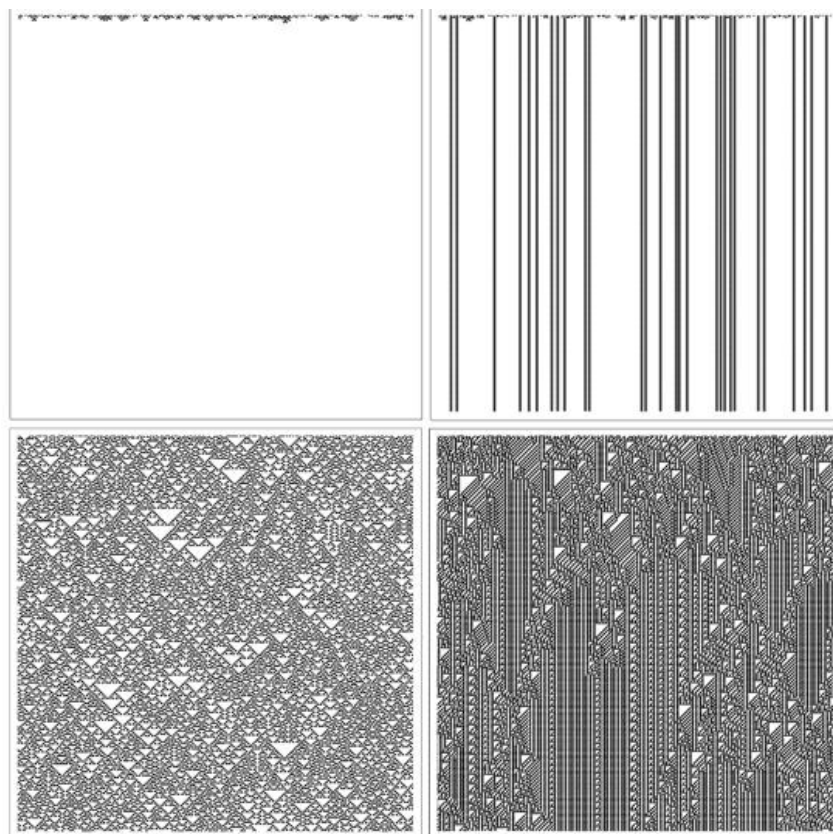


Figura 7: Evoluções temporais da regra 22 no espaço elementar com atualização assíncrona por prioridade de vizinhança representando as quatro classes de ACs. Evolução de classe 1 da regra 22 (esquerda superior). Evolução de classe 2 da regra 22 (direita superior). Evolução de classe 3 da regra 22 (esquerda inferior). Evolução de classe 4 da regra 22 (direita inferior).

Parte da previsão das classes obtidas é resultado de particularidades das execuções. Por exemplo isto ocorre na regra 110 devido ao comportamento típico desta ser como no caso síncrono para a maioria das configurações iniciais. Sendo assim, qualquer classificação obtida aqui corresponde a uma classificação do caso síncrono para essa regra.

Entretanto observa-se que houve uma divisão da classificação obtida. Isso se deve ao fato de, eventualmente, a Regra 110 entrar em um regime não-complexo a depender da quantidade de iterações e é por conta disto que para a análise da progressão são puladas uma certa quantidade de linhas, porém o transiente pode variar de tamanho interferindo no aprendizado e teste do algoritmo.

5 CONSIDERAÇÕES FINAIS

A partir dos resultados acima, observa-se que os autômatos celulares assíncronos se comportam tanto de forma previsível se relacionarmos quantos esquemas iniciais cada regra possui, como de forma aleatória se perceber que mesmo pelas similaridades em casos síncronos os seus desenvolvimentos podem ser muito diferentes nos casos assíncronos estudados.

Os ACs de casos assíncronos em relação aos síncronos possuem uma variabilidade maior, visto que, durante o estudo pode ser observado que uma regra com assincronia pode perder a sua característica de classe e outras acabam mantendo, e por conta disto que é muito interessante o estudo dos ACs assíncronos. Pois a partir de sistemas simples podemos resolver problemas complexos do mundo real.

6 AGRADECIMENTOS

Agradeço muito ao professor e orientador Eurico Ruivo, que me ajudou a desenvolver este projeto, e proporcionou a mim a oportunidade de trabalhar na iniciação científica.

Gostaria de agradecer aos meus colegas com quem trabalhei em conjunto durante a pesquisa, e juntamente agradeço o apoio da bolsa de iniciação científica que foi concedida pelo MackPesquisa – Fundo Mackenzie de Pesquisa.

Também agradecemos à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo projeto STIC-AmSud (CoDANet) nº 88881.197456/2018-01.

7 REFERÊNCIAS

ABURAS, M. M. e43t al. The simulation and prediction of spatio-temporal urban growth trends using c44ellular 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.I.], 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.
- LI, W.; PACKARD, N. The structure of the elementary cellular automata rule space. *Structure*, v. 4, p. 281–297, 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.

NOGUEIRA, M. A. Classificação automática do comportamento dinâmico de autômatos celulares binários. Tese (Doutorado) — Universidade Presbiteriana Mackenzie, 2019.

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.l.]:

Springer, 2000.

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

WOLFRAM, S. Twenty problems in the theory of cellular automata. Physica Scripta, Institute of Physics Publishing, T9, n. 3, p. 170–183, 1985.

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

PASSOS, A. M. Classificação de Complexidade de Autômatos Celulares por Meio de Análise de Vetores de Textura, 2021.

OLIVEIRA, P. P. de. Cellular Automata Mathematica Toolbox – CAMaT, Universidade Presbiteriana Mackenzie, 2019.

DE OLIVEIRA, P. P. B. CAMaT.nb - Cellular automata package for Mathematica, Software de distribuição restrita, disponibilizada pelo orientador, 2019.

Contatos: bernardo.sereis@gmail.com e eurico.ruivo@mackenzie.br