

CLASSIFICAÇÃO AUTOMÁTICA DE AUTÔMATOS CELULARES POR  
MÁQUINAS PARALELAS USANDO ENTROPIA DE TRANSIÇÕES DE  
ESTADOS

Matheus Côrtes Machado Agostini

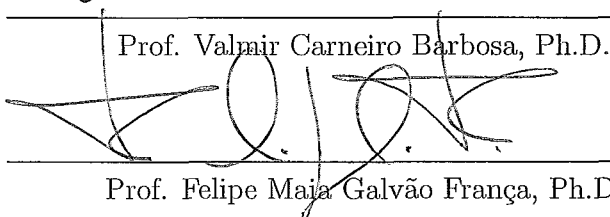
TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS  
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA  
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS  
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE  
EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:



---

Prof. Valmir Carneiro Barbosa, Ph.D.



---

Prof. Felipe Maia Galvão França, Ph.D.



---

Prof. Maria Clícia Stelling de Castro, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

ABRIL DE 2005

AGOSTINI, MATHEUS CÔRTEZ  
MACHADO

Classificação automática de autômatos  
celulares por máquinas paralelas usando en-  
tropia de transições de estados [Rio de  
Janeiro] 2005

XI, 74 p. 29,7 cm (COPPE/UFRJ,  
M.Sc., Engenharia de Sistemas e Com-  
putação, 2005)

Tese - Universidade Federal do Rio de  
Janeiro, COPPE

1. Autômatos Celulares
2. Entropia de Entradas
3. Computação Paralela

I. COPPE/UFRJ II. Título ( série )

Dedico aos meus pais,  
aos meus avós,  
à minha noiva, *Luciana Leonor*  
e ao meu irmão, *Lucas*

# Agradecimentos

Dificilmente eu conseguiria, com palavras, expressar toda gratidão que sinto neste momento no qual lembro de todas as pessoas que, de alguma forma, colaboraram para que este trabalho fosse concluído.

Em primeiro lugar agradeço ao bom Deus que sempre esteve ao meu lado em todos os momentos, concedendo luz e sabedoria de que eu precisava para enfrentar as provações desta vida.

Agradeço aos meus avós e meus pais que nunca deixaram de me apoiar e participar do meu caminhar, quer fosse fácil ou difícil, ainda que precisassem renunciar algo em favor de mim.

Agradeço ao Professor Valmir Carneiro Barbosa, por acreditar em mim e por me acolher, oferecendo-me um tema de pesquisa atual e interessante. Por estar sempre disposto a oferecer os estímulos necessários diante de cada situação. Por ser paciente e generoso, deixando transparecer seu caráter compreensivo frente aos meus problemas. Pela satisfação de trabalharmos juntos.

Agradeço à Professora Maria Clícia Stelling de Castro, pela amizade e pelo carinho que vêm desde a época da graduação e pelo constante incentivo ao interesse pelas ciências. Por ter tido participação especial na minha continuidade no desenvolvimento deste trabalho. Pela alegria de tê-la como amiga.

Agradeço aos professores do PESC pelos ensinamentos valiosos, de maneira especial ao Professor Felipe Maia Galvão França pelo interesse neste trabalho.

Agradeço à Luciana pelo amor, carinho e pela paciência, pois mesmo à distância, sempre procurava me manter firme no propósito de concluir este trabalho, aos amigos Fernando Marcos e Graziela, pela amizade desprendida e lealdade e por estarem sempre ao meu lado dando o apoio necessário pra qualquer momento desde os tempos da graduação.

Enternecido, agradeço a todos.



Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.).

CLASSIFICAÇÃO AUTOMÁTICA DE AUTÔMATOS CELULARES POR  
MÁQUINAS PARALELAS USANDO ENTROPIA DE TRANSIÇÕES DE  
ESTADOS

Matheus Côrtes Machado Agostini

Abril/2005

Orientador: Valmir Carneiro Barbosa

Programa: Engenharia de Sistemas e Computação

Este trabalho apresenta um estudo da classificação das regras de atualização de autômatos celulares de acordo com as quatro classes de Wolfram. Damos início abordando o conceito de entropia de entradas de um bloco espaço-temporal na evolução de um autômato celular e partindo deste conceito apresentamos uma nova medida de entropia baseada nas transições de estado das células. Nossa nova entropia é voltada para a classificação de regras de atualização por máquinas paralelas, sendo assim cuidadosa com os requisitos de comunicação necessários; para refletir este fato nós a chamamos de entropia de transições centrada na célula. Conduzimos o trabalho baseados em amplos experimentos computacionais envolvendo autômatos celulares uni e bi-dimensionais. Estes experimentos nos permitiram concluir que esta nova entropia possui forte capacidade discriminatória, por conseguinte oferecendo valioso auxílio no processo de classificação.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.).

AUTOMATIC CLASSIFICATION OF CELLULAR AUTOMATA BY  
PARALLEL MACHINES USING A STATE-TRANSITION ENTROPY

Matheus C3rtes Machado Agostini

April/2005

Advisor: Valmir Carneiro Barbosa

Department: Systems Engineering and Computer Science

This work presents a study of the classification of cellular-automaton update rules into Wolfram's four classes. We start with the notion of the input entropy of a spatiotemporal block in the evolution of a cellular automaton, and build on it by introducing a novel entropy measure based on state transitions by the cells. Our new entropy is targeted at the classification of update rules by parallel machines, being therefore mindful of the necessary communications requirements; we call it cell-centric transition entropy to reflect this fact. We report on extensive computational experiments on both one- and two-dimensional cellular automata. These experiments allow us to conclude that the new entropy possess strong discriminatory capabilities, therefore providing valuable aid in the classification process.

# Sumário

Agradecimentos	iv
Resumo	v
Abstract	vi
1 Introdução	1
2 Conceitos em autômatos celulares	5
2.1 Noções básicas . . . . .	5
2.2 Estratégias de classificação . . . . .	11
2.3 Entropia de entradas . . . . .	17
3 Classificação por entropia de transições	22
3.1 Heurística centrada na célula . . . . .	22
3.2 Entropia de transições . . . . .	26
3.3 Limite superior . . . . .	29
4 Experimentos e análise de resultados	31
4.1 Autômatos celulares infinitos . . . . .	31
4.2 O valor de $T$ . . . . .	36
4.3 Análise de resultados . . . . .	56
4.4 Discussão . . . . .	64
5 Considerações finais	67



# Lista de Figuras

2.1	Atualização de estados de um autômato celular uni-dimensional do tempo $t$ para o tempo $t + 1$ . . . . .	6
2.2	Vizinhanças de Moore e von Neumann . . . . .	7
2.3	Esquema de numeração para os autômatos uni-dimensionais . . . . .	9
2.4	Exemplos de padrões evolutivos de autômatos celulares de uma dimensão . . . . .	10
2.5	Janela de tamanho $T$ com $X_1 \times \dots \times X_d \times T$ estados de um autômato uni-dimensional . . . . .	20
3.1	Simulação paralela de autômatos celulares usando $N$ processadores . . . . .	28
4.1	Simulação de um autômato celular uni-dimensional infinito . . . . .	33
4.2	Conjunto $15 \times 10$ de células . . . . .	36
4.3	Média e variância da entropia de transições centrada na célula como uma função de $T$ - uni-dimensional . . . . .	39
4.4	Média e variância da entropia de transições centrada na célula como uma função de $T$ - bi-dimensional . . . . .	40
4.5	Padrões espaço-temporais de autômatos uni-dimensionais infinitos . . . . .	43
4.6	Padrões espaço-temporais de autômatos bi-dimensionais infinitos . . . . .	44
4.7	Padrões espaço-temporais de autômatos uni-dimensionais cilíndricos . . . . .	45
4.8	Padrões espaço-temporais de autômatos bi-dimensionais cilíndricos . . . . .	46
4.9	Padrões espaço-temporais em 3D von Neumann $15 \times 15$ infinito . . . . .	47
4.10	Padrões espaço-temporais em 3D Moore $15 \times 15$ infinito . . . . .	48
4.11	Padrões espaço-temporais em 3D von Neumann $15 \times 15$ cilíndrico . . . . .	49

4.12	Padrões espaço-temporais em 3D Moore 15x15 cilíndrico . . . . .	50
4.13	Padrões espaço-temporais em 3D von Neumann 30x30 infinito . . . . .	51
4.14	Padrões espaço-temporais em 3D Moore 30x30 infinito . . . . .	52
4.15	Padrões espaço-temporais em 3D von Neumann 30x30 cilíndrico . . . . .	53
4.16	Padrões espaço-temporais em 3D Moore 30x30 cilíndrico . . . . .	54
4.17	Distribuição das células de um autômato celular bi-dimensional ao longo de $N$ processadores . . . . .	58
4.18	Ocorrência de pares de média-variância para autômatos celulares unidimensionais . . . . .	60
4.19	Ocorrência de pares média-variância dentro de um conjunto de experimentos para autômatos celulares bi-dimensionais . . . . .	63

# Lista de Tabelas

4.1	Regras de atualização conhecidas . . . . .	38
4.2	Médias e variâncias extraídas de uma média de 5 simulações com estados iniciais escolhidos aleatoriamente . . . . .	55

# Capítulo 1

## Introdução

Autômatos celulares são sistemas dinâmicos, de tempo discreto, que podem simular sistemas naturais complexos contendo um grande número de elementos idênticos, simples e com interações locais. Cada um destes elementos é chamado célula e possui um conjunto finito de estados possíveis. O estado de uma célula particular é estabelecido de acordo com os estados preliminares de uma vizinhança de células ao seu redor. Os estados das células evoluem de maneira síncrona em passos de tempo discretos de acordo com regras idênticas.

Um grupo simples de autômatos celulares teve seu comportamento discutido em [29], onde foram fornecidas extensivas referências. Mostrou-se que apesar de sua construção simples, alguns autômatos celulares podem apresentar comportamentos complexos. Em [31, 13], podemos encontrar um sumário de resultados importantes e muitas outras referências. Um dos temas mais interessantes da investigação é a classificação de regras de atualização  $f$ , e por conseguinte do autômato celular baseado nesta, considerando sua complexidade.



A atenção inicial ao interesse nesta questão foi dada a partir do estudo dos autômatos celulares infinitos uni-dimensionais por Stephen Wolfram [30]. Esse estudo resultou na descoberta empírica de que, praticamente desconsiderando o estado inicial do autômato,  $f$  recai de forma consistente dentro de uma das quatro possíveis classes qualitativas: (i) a evolução do autômato atinge uma configuração homogênea, i.e., uma configuração na qual todas as células possuem o mesmo estado; (ii) a evolução leva a um ponto fixo não-homogêneo ou a um ciclo; (iii) a evolução leva a uma sucessão caótica de configurações; ou (iv) a evolução leva a estruturas espaço-temporais localizadas que são algumas vezes persistentes. Esta classificação qualitativa foi criada inicialmente para o caso uni-dimensional. Entretanto não existe, a princípio, um argumento pelo qual ela não possa ser aplicada aos casos de maiores dimensões. Na verdade, pouco tempo depois apareceram em [23] estudos similares para o caso bi-dimensional.

De modo natural, as regras de atualização da classe (iv) são associadas intuitivamente à realização de computação “complexa” pelos autômatos celulares que são construídos a partir dessas regras. Tem-se grande interesse na utilização desta computação como ferramenta de modelagem. Por esta razão, têm sido realizados grandes esforços na busca por métodos capazes de categorizar automaticamente as regras de atualização dentro das classes (i) a (iv). Formalmente, todos os esforços circundam o chamado conjunto limite de uma regra de atualização  $f$  no caso infinito. Este é o conjunto de configurações que resultam de todas as configurações iniciais possíveis depois de decorrido um longo tempo. Como resultado, toda propriedade

não trivial de um conjunto limite (i.e., propriedade que existe para pelo menos um autômato celular e não existe para pelo menos um outro) pode ser provada como sendo indecidível através da redução do problema de decidir se existe um conjunto limite com um único elemento [15], o qual é conhecido por ser indecidível [14].

Em consequência desta incerteza inerente, toda estratégia para a classificação de regras de atualização deve, necessariamente, ser de natureza heurística ou então resumir-se em uma heurística se houver algum uso prático. O objetivo deste trabalho é estudar uma heurística que pode ser agregada à simulação paralela de autômatos celulares visando analisar os padrões espaço-temporais que surgem, com objetivo de categorizar regras de atualização dentro das quatro classe de Wolfram. Assim sendo, a eficiência no uso mínimo de comunicação é essencial, o que nos leva ao que chamamos de heurística centrada na célula, ou seja, a heurística que depende o mínimo possível da troca de informações entre os processadores durante a simulação de um autômato celular em sistemas paralelos.

Tendo em vista uma aproximação do leitor com os conceitos fundamentais para o bom entendimento do tema, no Capítulo 2 são abordados os assuntos mais importantes de autômatos celulares e revisamos algumas heurísticas distintas que têm sido propostas para a classificação automática das regras de atualização. A partir de então, com uma familiaridade maior destes conceitos, discutimos no Capítulo 3 as chamadas medidas de entropia de entradas e em seguida apresentamos nossa heurística definida como centrada na célula. Após todas as considerações a respeito de nossa heurística, mostramos e analisamos no Capítulo 4 os resultados dos exper-

imentos computacionais feitos com autômatos celulares de uma e duas dimensões, e ainda estendemos a discussão sobre o assunto. Finalmente, no Capítulo 5, são feitas algumas considerações adicionais e observações relevantes.

# Capítulo 2

## Conceitos em autômatos celulares

### 2.1 Noções básicas

Autômatos celulares são sistemas dinâmicos, discretos no tempo, constituídos de elementos que podem assumir um número finito de estados. Estes elementos são denominados células. A evolução do autômato no tempo se dá pela atualização dos estados das células como resultado da interação com outras células. Os autômatos celulares têm adquirido um papel cada vez mais importante como uma ferramenta de modelagem em diversas áreas de pesquisa (cf., ex., [16, 2] e suas referências) desde sua introdução há cinco décadas por von Neumann [27]. Foram apontados até mesmo como uma abstração central na modelagem do processo fundamental da natureza [32].

Dados um conjunto  $S = \{0, \dots, s-1\}$  de estados possíveis e um inteiro  $t \geq 0$ , um autômato celular com  $n$  células evolui do tempo  $t$  para o tempo  $t+1$  por intermédio da atualização sincronizada de todos os  $n$  estados pela aplicação de um mapeamento determinístico  $F_f$  de  $S^n$  em  $S^n$ , como está ilustrado na Figura 2.1. Este mapeamento

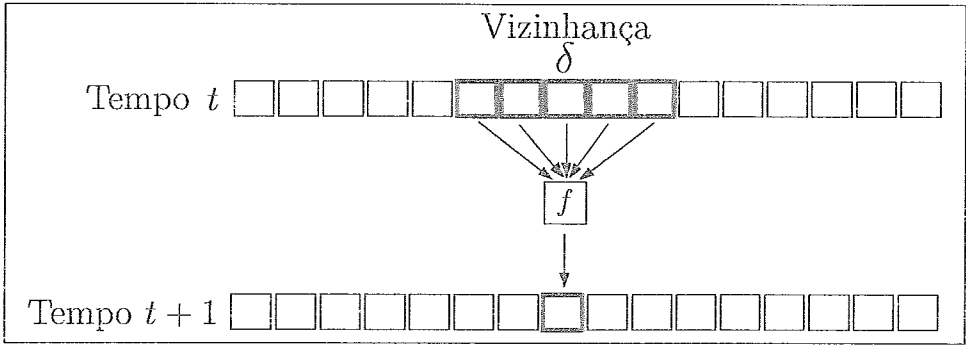


Figura 2.1: Atualização de estados de um autômato celular uni-dimensional do tempo  $t$  para o tempo  $t + 1$ , aplicando o mapeamento determinístico  $F_f$ , onde  $f$  é a regra de atualização local.

$F_f$  é de natureza global e depende da regra de atualização local  $f$ , que indica como cada estado individual deve ser atualizado a partir do estado da célula no tempo  $t$  bem como do estado das células adjacentes que estão dentro de uma vizinhança de tamanho  $\delta$ . Assim, a regra de atualização  $f$  é o mapeamento de  $S^{1+\delta}$  em  $S$ .

A vizinhança de um autômato celular é determinada, normalmente, por um reticulado multidimensional subjacente de acordo com diversos critérios possíveis. Um vizinho de uma célula relativo a uma certa dimensão do reticulado pode ser, por exemplo, uma das células que estejam afastadas  $r > 0$  posições da célula em questão ao longo desta dimensão. Porém, nenhuma posição afastada em nenhuma outra dimensão. O número de posições representado por  $r$  é referido sempre como o raio da regra de atualização naquela dimensão. Quando o raio  $r = 1$  em todas as dimensões, fica caracterizada a vizinhança conhecida como vizinhança de von Neumann (c.f., Figura 2.2). Entretanto, neste trabalho empregamos a mesma denominação para

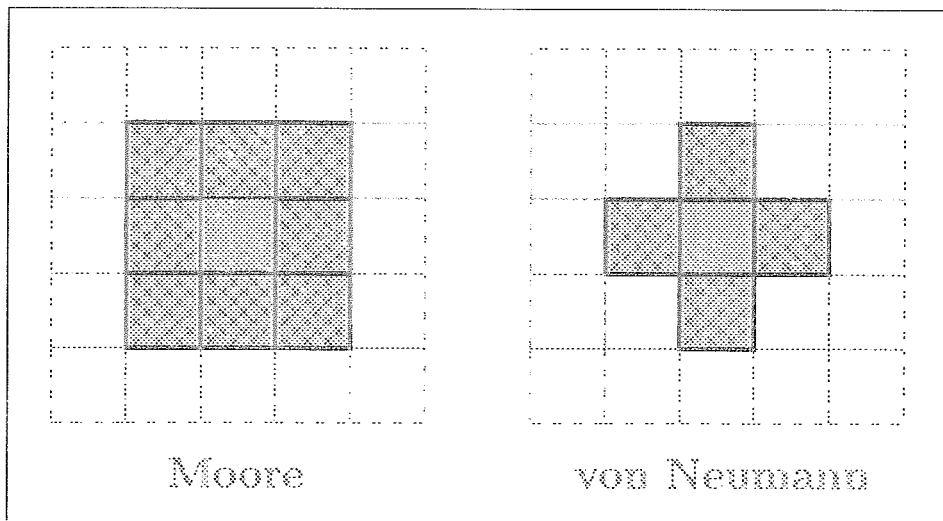


Figura 2.2: Vizinhanças de Moore e von Neumann, respectivamente. O raio  $r = 1$  em ambos os casos.

raios maiores. Poderíamos, ainda, permitir que duas células sejam vizinhas entre si sempre que uma puder alcançar a outra sem ultrapassar nenhuma fronteira ao longo de uma certa dimensão, onde esta fronteira é demarcada pelo raio  $r$  da regra de atualização naquela dimensão. Para o raio  $r = 1$ , neste último exemplo, esta vizinhança é chamada de vizinhança de Moore (c.f., Figura 2.2), contudo, novamente, empregamos a mesma denominação para raios maiores. É usual considerar que o reticulado possua bordas cilíndricas, isto é, permitir que toda célula tenha exatamente duas vizinhanças mais próximas ao longo de cada dimensão, para os casos em que  $n$  é finito. Ao tratarmos de autômatos celulares finitos, aqueles cujo  $n$  é finito,  $F_f$  leva, necessariamente, a um ponto fixo ou a um ciclo de configurações em  $S^n$ , ou seja,  $x$  tal que  $F_f(x) = x$  ou  $x_0, \dots, x_{p-1}$ , com  $p > 0$ , de modo que  $x_0 = F_f(x_{p-1})$ ,

$x_1 = F_f(x_0)$ , e assim sucessivamente [34]. Por outro lado, o caso dos autômatos celulares infinitos é bem mais complicado e intrigante, já que  $n$  é totalmente infinito e nenhuma periodicidade é garantida a partir da aplicação sucessiva de  $F_f$ .

Um esquema de numeração para os autômatos uni-dimensionais de raio  $r = 1$  foi proposto por Wolfram em [29]. Nele, os *bits* de saída são ordenados lexicograficamente, como na regra de atualização da Figura 2.3, e são lidos da direita para a esquerda para formar um número na base decimal entre 0 e 255. Por exemplo, a regra apresentada na Figura 2.3 (em binário: 10110101) é a regra de atualização 181. Para autômatos celulares uni-dimensionais de raio  $r > 1$ , é adotado este mesmo esquema de numeração, a partir de uma ordenação lexográfica dos *bits* de saída. Todavia, para raios maiores, o que se faz normalmente é traduzir o código binário da regra para a base hexadecimal. Por exemplo, a regra de raio  $r = 2$  dada pelos *bits* 10101010110010111000101011101101 é equivalente à regra AACB8AED. Nestes casos, o número de regras de atualização distintas é dado por  $2^{2^{1+2r}}$ .

No caso de autômatos celulares bi-dimensionais sob a vizinhança de von Neumann é adotada uma notação semelhante. Contudo, o *bit* mais significativo corresponde a  $00 \dots 0$ , o próximo a  $00 \dots 0 + 1$  e assim sucessivamente. Assim sendo, as entradas são lidas em uma ordem particular; *célula*, norte, leste, sul, oeste [28]. Nesse caso o número de regras distintas é dado por  $2^{2^{1+2(r_1+r_2)}}$ . As regras de atualização de autômatos celulares sob a vizinhança de Moore são, por sua vez, em geral totalísticas [23], ou seja, os *bits* de saída são obtidos através da soma dos *bits* da vizinhança e não pelo valor discreto de cada um. É adotada, neste caso, a notação

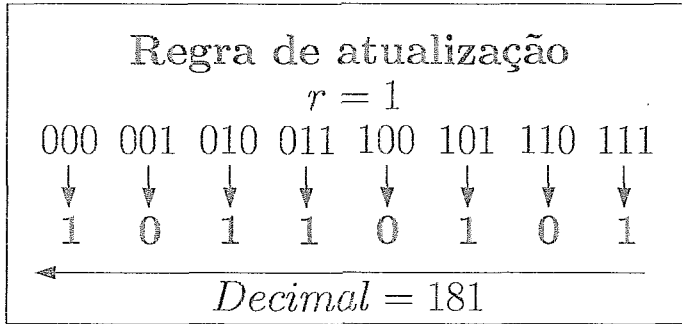


Figura 2.3: Exemplo de regra de atualização de raio  $r = 1$ , com vizinhança  $\delta = 3$  e 8 combinações de 3 bits, gerando 8 bits de saída ordenados lexicograficamente. O binário 10010101 é a regra de atualização 149 expressa em decimal.

do “Jogo da vida” de Conway [11, 5] como a seguir:  $b\bar{x}$  indica que o estado da célula move-se de 0 para 1 (“nascimento”) se a célula tem  $x$  vizinhos com estado 1;  $sx$  significa que o estado da célula permanece 1 (“sobrevivência”) se a célula tem  $x$  vizinhos com estado 1; em todos os demais casos não especificados o estado da célula se torna 0 [28](e.g.,  $b1b3b5s1s3s5$ ). O número de regras distintas, neste caso, é  $2^{2(1+2r_1)(1+2r_2)}$ .

As Figuras em 2.4 apresentam padrões evolutivos de autômatos celulares de uma dimensão, com  $n = 300$  células, à medida em que  $t$  aumenta. Nestes exemplos, há dois estados possíveis que são representados graficamente por pontos com coloração preta ( $s = 1$ ) e branca ( $s = 0$ ). Cada caso gera um padrão evolutivo diferente que pode ser classificado qualitativamente de acordo as quatro classes de Wolfram [30].



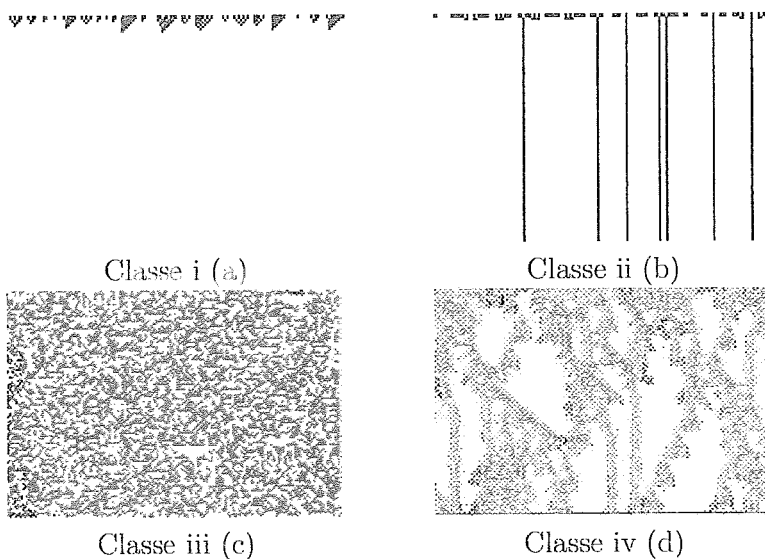


Figura 2.4: Exemplos de padrões evolutivos de autômatos celulares de uma dimensão. As Figuras (a)–(d) mostram a evolução no tempo (de cima para baixo) de autômatos celulares baseados em quatro regras de atualização diferentes a partir de estados iniciais escolhidos aleatoriamente. Nestes casos, temos dois estados possíveis  $S = \{0, 1\}$  e vizinhança  $\delta = 5$ , sendo que, no caso (a) a regra de atualização leva, depois de um certo número de passos, a um estado homogêneo único; no caso (b), leva a um conjunto de estruturas estáveis e distintas; no caso (c), a regra de atualização leva a um padrão caótico e instável; e no caso (d), leva a padrões complexos e muitas vezes de longa duração.

## 2.2 Estratégias de classificação

De modo geral, percebemos duas classes de estratégia fundamentais para a categorização de regras de atualização. A primeira classe abarca as técnicas que visam à extração das capacidades computacionais das regras de atualização considerando somente as próprias regras de atualização, desconsiderando qualquer simulação de autômatos celulares para a análise dos resultados dos padrões espaço-temporais. Nas abordagens deste tipo, o tamanho da vizinhança de uma célula é de fato  $\delta = 2r$ , pois elas concentraram-se em autômatos celulares uni-dimensionais.

Dentro desta classe o primeiro passo foi dado por Langton [17], que propôs a classificação de uma regra de atualização  $f$  dentro das quatro classes de Wolfram, examinando apenas um parâmetro, denominado  $\lambda_f$ . Este parâmetro expressa a probabilidade de que uma vizinhança aplicada em  $f$  leve a um estado invariável  $\sigma$ , conhecido como estado quiescente. Quando  $\lambda_f = 0$ , todas as vizinhanças levam ao estado mudo  $\sigma$  e nesse caso, à medida em que  $t$  cresce, o autômato celular evolui para um estado homogêneo ou ordenado. Em contrapartida, quando  $\lambda_f = 1$ , nenhuma vizinhança leva a  $\sigma$ , e assim o autômato evolui para um estado caótico ou desordenado. O parâmetro é dado por:

$$\lambda_f = 1 - \frac{q}{s^{1+2r}}. \quad (2.1)$$

Em (2.1),  $q$  é o número de linhas da regra de atualização de tamanho  $(1 + 2r)$  para as quais  $f$  tem  $\sigma$  como saída, onde  $\sigma \in S$  é o estado mudo.

O primeiro relato de uso do parâmetro  $\lambda_f$  indicou um comportamento tal como um parâmetro de ordem com relação à ocorrência de transição de fase: ao se gerar regras de atualização  $f$  com  $\lambda_f$  aumentando, encontra-se inicialmente o comportamento de classe (i), em seguida o de classe (ii), o de classe (iv) em torno de  $\lambda_f = 0.5$ , e finalmente o comportamento de classe (iii). Este comportamento parece indicar que a complexidade é encontrada em uma região no espaço  $\lambda_f$  conhecida como “fronteira do caos”. Entretanto, considerando algumas dificuldades em relação à existência e escolha de um estado mudo, logo surgiram críticas de várias fontes (cf., ex., [22, 21, 19]) em relação à existência e à natureza da transição de fase. Particularmente, foram encontradas próximo de  $\lambda_f = 0.5$  regras de atualização pertencentes a várias classes, não somente a classe (iv).

Foram introduzidas duas outras abordagens interessantes, que, também, não exigem simulações de autômatos celulares. O objetivo de investigar como a informação contida em uma regra de atualização  $f$  afeta o comportamento geral dos autômatos celulares construídos em  $f$  é compartilhado por ambas. Uma das abordagens é de natureza topológica, isto é, visa analisar o comportamento global dos autômatos celulares através da identificação de padrões espaço-temporais de tamanho finito [6]. Por outro lado, uma outra referência [8] é algorítmica e procura caracterizar as regras de atualização  $f$  sob a perspectiva da complexidade de Kolmogorov [18], isto é, a perspectiva da descrição mais simples de  $f$ . Apesar de cada abordagem possuir uma perspectiva própria, ambas estão claramente relacionadas à classificação de Wolfram. Outrossim, a última abordagem pode ser um ponto fundamental para

algumas incongruências que são inerentes ao parâmetro  $\lambda_f$ . Entretanto, é relevante destacar que, dado que as duas abordagens induzem a um sistema particular de classe, nenhuma delas refere-se claramente à classe (iv). Deixamos indicada a fonte original para detalhes adicionais.

Para categorizar as regras de atualização de autômatos celulares há, também, uma classe de estratégias inteiramente distinta que concentra-se na análise dos padrões espaço-temporais à medida em que estes vão se formando durante a evolução dos autômatos celulares a partir de uma configuração inicial. Agora, se torna naturalmente necessário considerar o fato de que os autômatos celulares observados são formalmente infinitos. Contudo, por ora deixamos de lado este conceito e qualquer dificuldade que ele possa causar na prática, pois tratamos deste assunto na Seção 4.1. Todavia, é importante citar que algumas abordagens de sucesso foram construídas a partindo do pressuposto de que o número de células é finito. Podemos oferecer como exemplo a “mecânica computacional” em [12], a qual para autômatos celulares uni-dimensionais baseia-se em máquinas de estados-finitos [24], derivadas de padrões na evolução do autômato celular, para delinear as características espaço-temporais fundamentais que são inerentes a cada regra de atualização.

Nesta classe, estão as abordagens que fundamentam-se em alguma forma de medida de entropia como base dos esforços de classificação, sendo estas, centrais ao nosso interesse. A abordagem inicial nesta direção apareceu no mesmo trabalho que introduziu as quatro classes de Wolfram [30]. Em essência, o que se faz é considerar a distribuição de probabilidade de blocos espaço-temporais à medida em que eles

ocorrem durante a evolução dos autômatos celulares para um regra fixa  $f$  e então usar essa distribuição para definir as entropias desejadas.

Para uma caracterização mais precisa, seja  $d > 0$  o número de dimensões dos autômatos celulares em questão, e sejam  $X_1, \dots, X_d$  o números de células contíguas ao longo de cada dimensão. Sendo  $t_+ \geq 0$  um número de passos de tempo sucessivos durante a evolução do autômato celular que emprega a regra de atualização  $f$ , queremos saber o comportamento do autômato celular, dado  $f$ , dentro do espaço de tempo compreendido em  $t_+$ . Para um passo de tempo  $t$ ,  $0 \leq t < t_+$ , temos que o bloco de estados  $X_1 \times \dots \times X_d$  no tempo  $t$  é obtido aplicando-se a regra de atualização  $f$  sobre o bloco de estados  $X_1 \times \dots \times X_d$  no tempo  $t - 1$ . Para melhor visualizar a aplicação da regra de atualização  $f$  sobre  $X_1 \times \dots \times X_d$ , consideramos a construção de uma tabela de regras baseada em  $f$ , isto é, uma tabela que reflita o mapeamento de  $S^{1+\delta}$  em  $S$ , onde cada entrada da tabela é formada por uma  $(1 + \delta)$ -tupla distinta que uma vizinhança de tamanho  $\delta$  pode assumir mais sua saída correspondente. Assim, para cada estado que é atualizado no bloco de estados  $X_1 \times \dots \times X_d$  é feito um acesso à tabela de regras.

Quando mantemos um histórico da frequência de acessos à tabela de regras no tempo  $t$  da evolução do autômato celular esta abordagem é extremamente útil, pois a partir deste histórico obtemos uma medida segura para a classificação do comportamento do autômato dentro do espaço de  $t_+$  passos de tempo. Esta medida compreende as entropias dos acessos às entradas da tabela de regras para cada passo de tempo  $t$  bem como a variância dessas quantidades dentro do espaço de tempo

compreendido em  $t_+$  [33]. Pode-se representar a freqüência de acessos à tabela por um histograma que distribui o total de  $X_1 \times \dots \times X_d$  acessos em cada passo de tempo da evolução sobre as  $s^{1+\delta}$  vizinhanças (i.e., as entradas da tabela).

A entropia de Shannon [25] desta distribuição de freqüência de acessos, entropia de entradas  $E$ , no tempo  $t$ , é dada por:

$$E_f^t(X_1, \dots, X_d) = - \sum_{i=1}^{s^{1+\delta}} \left( \frac{q_i^t}{X_1 \dots X_d} \right) \log_s \left( \frac{q_i^t}{X_1 \dots X_d} \right), \quad (2.2)$$

onde  $q_i^t$  expressa a quantidade de acessos à entrada  $i$  da tabela de regras no tempo  $t$ . Logo, a partir de (2.2) temos a média das entropias de entradas para  $t_+$  passos de tempo da execução:

$$\overline{E}_f(X_1, \dots, X_d) = - \frac{1}{t_+} \sum_{t=0}^{t_+-1} \sum_{i=1}^{s^{1+\delta}} \left[ \left( \frac{q_i^t}{X_1 \dots X_d} \right) \log_s \left( \frac{q_i^t}{X_1 \dots X_d} \right) \right]. \quad (2.3)$$

A entropia de entradas estabelecida em (2.2) expressa uma distribuição de probabilidade, já que podemos considerar  $\frac{q_i^t}{X_1 \dots X_d}$  uma aproximação da probabilidade de que a  $i$ -ésima entrada da tabela seja acessada no tempo  $t$ . Ou seja, a entropia pode nos mostrar se os acessos à tabela de regras estão mais concentrados ou mais uniformemente distribuídos (i.e., todas as entradas têm a mesma probabilidade de ocorrer no tempo  $t$ ). Dessa forma, o valor médio em (2.3) das entropias de entradas pode ser utilizado para classificar o comportamento do autômato celular no espaço de tempo compreendido em  $t_+$ . Detalhando, temos que valores baixos das entropias de entradas refletem o fato de que os acessos estão mais concentrados em algumas entradas. Em termos práticos isso mostra a inclinação do autômato celular para evoluir até um ponto fixo ou estado global ordenado. Já valores altos das

entropias de entradas expressam a tendência ao caos, porquanto, neste caso, os acessos estão mais uniformemente distribuídos. Contudo, faltaria categorizar através das entropias de entradas o comportamento de autômatos celulares complexos. Neste caso, a variância das entropias de entradas dentro do espaço de tempo compreendido em  $t_+$ , expressa em (2.4), nos fornece uma boa medida do comportamento do autômato celular ao longo de sua evolução:

$$\begin{aligned} \sigma^2(E_f(X_1, \dots, X_d)) \\ = \frac{1}{t_+} \sum_{t=0}^{t_+-1} [E_f^t(X_1, \dots, X_d) - \overline{E_f}(X_1, \dots, X_d)]^2. \end{aligned} \quad (2.4)$$

A complexidade que procuramos identificar é caracterizada pelo desequilíbrio das freqüências de acessos em alguns passos da evolução do autômato celular e pela uniformidade em outros. Isso é claramente refletido pela variância das entropias de entradas dentro do espaço de tempo compreendido em  $t_+$ .

Esta estratégia de classificação, introduzida por Wuensche [33], representa uma tentativa de fusão de algumas características importantes das classes de estratégias quantitativas e qualitativas; tanto pela análise estatística sobre a freqüência dos acessos à tabela de regras, que está fortemente ligada ao estudo de Langton sobre o potencial da regra de atualização  $f$  expresso através do parâmetro  $\lambda_f$ , quanto pelo exame do padrão espaço-temporal da evolução do autômato celular na tentativa de categorizá-lo dentro do espaço de tempo compreendido em  $t_+$ . Na Seção 2.3, abordamos esta estratégia com mais detalhes.

## 2.3 Entropia de entradas

Para que os dados coletados através da estratégia de classificação que utiliza entropia de entradas tratada na Seção 2.2 tenham algum significado classificatório, é necessário executar um número muito grande de passos de tempo do autômato devido à natureza estatística dessa estratégia. São necessárias, na verdade, algumas execuções com estados iniciais gerados de modo aleatório. Isto garante, na prática, que os blocos de estados observados estejam em locais diversos no traço espaço-temporal do autômato. Assim, temos uma classificação de seu comportamento global mais precisa.

O cálculo da entropia de (2.2), para um único passo de tempo, traduz o comportamento do autômato somente naquele tempo, sem considerar o que pode ter sucedido antes. Este fato constitui outro ponto importante que deve ser considerado. Caso pudesse refletir o comportamento do autômato em passos de tempo anteriores, esta informação, embora não sendo inútil, traria consigo um maior valor. Para tal, Wuensche apresentou a idéia de calcular a entropia de entradas para cada passo de tempo  $t$  considerando não somente  $t$ , porém, uma janela de  $T$  passos de tempo que inicia em  $T - t$  e termina em  $T - 1$ , onde  $0 < T < t_+$  sendo  $t_+ > 0$  o número de passos da execução do autômato [33]. A frequência de acessos às entradas da tabela de regras seria, deste modo, relativa não somente aos  $s^{1+\delta}$  acessos do tempo  $t$ , mas relativa aos  $s^{(1+\delta)T}$  acessos de todos os tempos dentro da janela, onde  $s$  é o número de estados diferentes que uma célula pode assumir e  $\delta$  o tamanho



da vizinhança da célula. A seguir, reescrevemos a entropia de entradas, a média das entropias e a variância, considerando  $T$ .

Dado um dos blocos de estados de tamanho  $X_1 \times \cdots \times X_d \times T$ , para uma janela de  $T$  passos de tempo, começamos considerando a probabilidade de que ocorra, dentro do bloco, cada uma das possíveis  $s^{1+\delta}$  entradas para uma célula. Denotando a probabilidade de ocorrência da  $i$ -ésima entrada por  $Q_i$ ,  $1 \leq i \leq s^{1+\delta}$ , a entropia de entradas para o tempo  $t$  dentro da janela de tamanho  $T$  é estabelecida por

$$E_f(X_1, \dots, X_d, T) = - \sum_{i=1}^{s^{1+\delta}} Q_i \log_s Q_i. \quad (2.5)$$

Na prática, o uso da entropia de entradas começa fixando-se o valor de  $X_1, \dots, X_d$  e de  $T$  e então escolhendo as células  $X_1 \dots X_d$  a serem observadas durante a simulação do autômato celular construído em  $f$ . Cada simulação é executada a partir de uma configuração inicial gerada aleatoriamente e então prossegue por  $t_+$  passos de tempo, para algum  $t_+ \geq T - 1$ , produzindo uma nova configuração a cada passo de tempo. Para cada tempo  $t$  no intervalo  $[t_0, t_+]$  com  $T - 1 \leq t_0 \leq t_+$ , a probabilidade  $Q_i$  de que a  $i$ -ésima entrada ocorra dentro de um bloco de tamanho  $X_1 \times \cdots \times X_d \times T$  pode ser aproximada por  $\frac{q_i^t}{X_1 \times X_2 \times \cdots \times X_d \times T}$ , onde  $q_i^t$  é o número de ocorrências da  $i$ -ésima entrada dentro do bloco de estados que termina no tempo  $t$ . Assim, o valor da entropia que tem origem em (2.5) e refere-se ao bloco que termina no tempo  $t$  é

$$E_f^t(X_1, \dots, X_d, T) = - \sum_{i=1}^{s^{1+\delta}} \left( \frac{q_i^t}{X_1 \dots X_d T} \right) \log_s \left( \frac{q_i^t}{X_1 \dots X_d T} \right). \quad (2.6)$$

A média e a variância da grandeza em (2.6) para  $t = t_0, \dots, t_+$  são

$$\overline{E}_f(X_1, \dots, X_d, T) = \frac{1}{t_+ - t_0 + 1} \sum_{t=t_0}^{t_+} E_f^t(X_1, \dots, X_d, T) \quad (2.7)$$

e

$$\begin{aligned} \sigma^2(E_f(X_1, \dots, X_d, T)) \\ = \frac{1}{t_+ - t_0 + 1} \sum_{t=t_0}^{t_+} [E_f^t(X_1, \dots, X_d, T) - \overline{E}_f(X_1, \dots, X_d, T)]^2, \end{aligned} \quad (2.8)$$

respectivamente. O primeiro valor de entropia é calculado, naturalmente, para o tempo  $t = T$ , pois é nesse tempo onde ocorre o primeiro bloco de estados de tamanho  $X_1 \times \dots \times X_d \times T$ . A Figura 2.5 exemplifica a forma com que a janela de tamanho  $T$  desloca-se a partir do tempo  $t = T$  até o final da execução de um autômato uni-dimensional. A cada passo de tempo da evolução do autômato, a janela avança uma linha, desprezando os acessos feitos à tabela de regras no tempo  $t - T - 1$  (i.e., primeira linha da janela no tempo  $t - 1$ ) e incorporando os acessos do tempo atual. Nesse momento, é calculada a entropia de entradas para este tempo.

Após termos estas grandezas ponderadas para um determinado número de simulações dadas configurações iniciais geradas aleatoriamente, elas podem ser usadas para revelar a qual classe de Wolfram a regra de atualização  $f$  pertence, conforme pudemos observar na Seção 2.2. A grosso modo, o cálculo da entropia de entradas é “suavizado” quando feito dentro de uma janela de passos de tempo. Todavia, isso nos leva a uma reflexão sobre qual seria o tamanho ideal da janela. Precisamos saber se qualquer valor de  $T$  fornece uma boa classificação ou se deve existir um valor capaz de otimizar o processo de classificação. Esta questão é discutida no

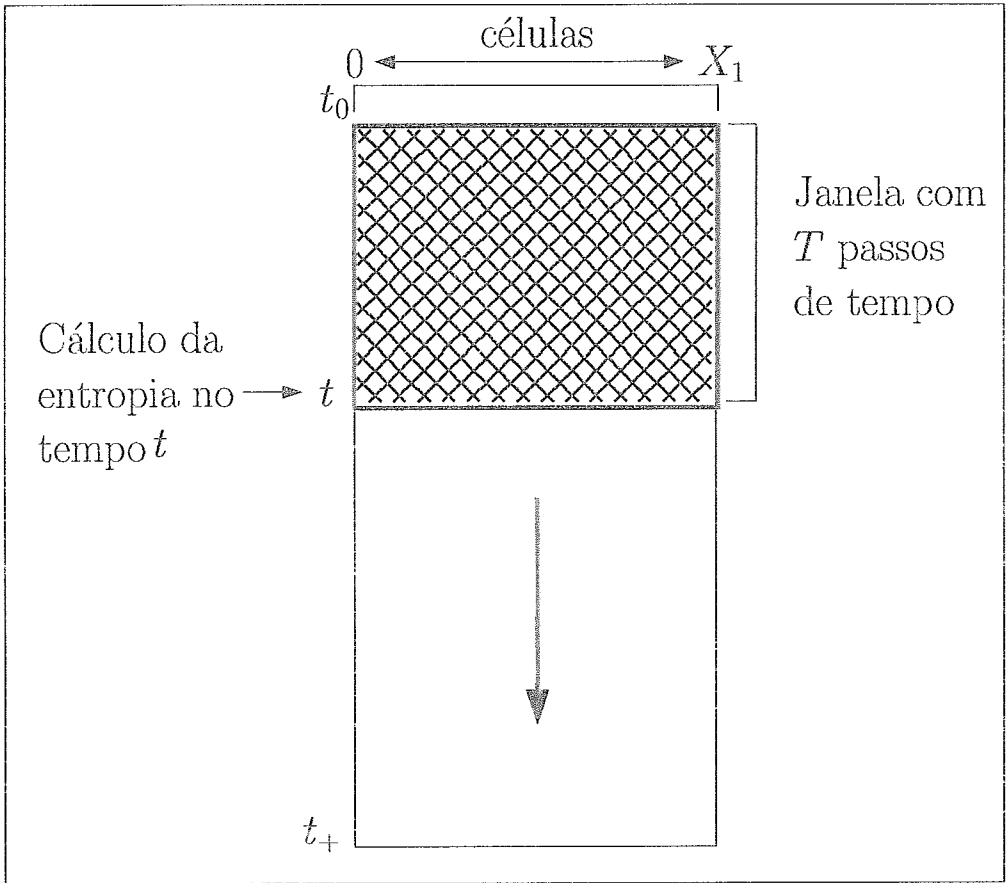


Figura 2.5: Janela de tamanho  $T$  com  $X_1 \times \dots \times X_d \times T$  estados de um autômato uni-dimensional. A cada passo de tempo da evolução do autômato, a janela avança um linha, desprezando os acessos feitos à tabela de regras no tempo  $t - T - 1$  e incorporando os acessos do tempo atual. Nesse momento, é calculada a entropia para este tempo.

Capítulo 4 sendo de grande importância em nosso estudo.

Os primeiros experimentos de Wuensche concentraram-se apenas na classificação de autômatos celulares uni-dimensionais. Nestes experimentos descobriu-se que médias e variâncias baixas caracterizam uma regra de atualização de classe (i) ou (ii), de comportamento ordenado, ao passo que médias altas e variâncias baixas revelam uma regra de atualização classe (iii), cujo comportamento é caótico. O comportamento da classe (iv), complexo, é indicado por valores intermediários da média e por variâncias altas [33].

Apresentamos, neste capítulo, a idéia central da construção de autômatos celulares, bem como o mapeamento global  $F_f$ , onde  $f$  é a regra de atualização local que indica como cada estado individual do conjunto  $S = \{0, \dots, s - 1\}$  de estados possíveis para cada célula do autômato pode ser assumido pelas células. Identificamos ainda duas classes de estratégias fundamentais para a classificação dessas regras de atualização: as técnicas quantitativas, que visam extrair as capacidades computacionais das regras de atualização considerando somente as próprias regras; e as técnicas qualitativas, concentradas na análise dos padrões espaço-temporais à medida em que estes vão se formando ao longo da evolução do autômato celular. O foco do nosso interesse reside nas estratégias desta última classe, que baseiam-se em alguma forma de entropia, ou seja, o que se faz é considerar a distribuição de probabilidade dos blocos espaço-temporais conforme eles surgem na evolução do autômato. Mostramos ainda o conceito de entropia de entradas introduzido por Wuensche, o qual embasará o método de classificação proposto por este trabalho.

## Capítulo 3

# Classificação por entropia de transições

### 3.1 Heurística centrada na célula

O cálculo da média e variância da entropia de entradas, conforme (2.7) e (2.8) respectivamente, exige a simulação do autômato celular que é baseado em  $f$  por  $t_+$  passos de tempo e requer também que as grandezas obtidas por (2.6) sejam reunidas, à medida em que a janela de tamanho  $X_1 \times \dots \times X_d \times T$  desliza ao longo da evolução do autômato. Pode ocorrer que as células  $X_1 \dots X_d$  não se encontrem no mesmo processador quando a simulação é realizada em paralelo. Dessa forma, computar as entropias de entradas, da maneira que foi apresentada em (2.6) para todos os valores de  $t$ , requer uma quantidade considerável de comunicação envolvendo os processadores que alocam as células.

A simulação paralela de um autômato celular consiste basicamente em distribuir suas  $X_1 \times \dots \times X_d$  células entre os  $N$  processadores disponíveis para esta computação. Cada processador deve possuir, então, uma cópia da tabela de regras e deve conter,

também, uma estrutura de dados capaz de armazenar o histórico de acessos às  $s^{(1+\delta)}$  entradas da tabela que acontecem dentro da janela de tamanho  $T$  passos de tempo. Além do mais, deve ser escolhido um processador para colher todas as informações do histórico de acessos de cada processador participante da computação. Este processador mestre é o responsável por calcular as entropias de entradas, a média dessas entropias ao longo dos  $t_+$  passos da evolução do autômato e também a variância.

Dado que um processador, digamos  $P$ , tenha sido escolhido para reunir todas as informações necessárias para a computação da média e da variância, temos que o número de inteiros que precisam ser transmitidos a  $P$  (i.e., número de acessos a cada entrada da tabela) é  $(N - 1) \times s^{1+\delta} \times t_+$ . De modo geral, para cada processador participante e cada tempo  $t$ , os inteiros que precisam ser transmitidos são os  $s^{1+\delta}$  inteiros, onde cada um corresponde à soma de todos os acessos a uma entrada na tabela dentro da janela que termina no tempo  $t$ . Portanto, o número total de inteiros necessários para os cálculos desejados ao longo dos  $t_+$  passos da evolução do autômato é da ordem de:

$$O(N \times t_+ \times s^{(1+\delta)}) \tag{3.1}$$

A simulação paralela de autômatos celulares, da forma como foi abordada, exige muitos recursos na prática. Dentre estes, um grande poder de processamento, necessário para computar a aplicação da regra de atualização  $f$  e uma grande capacidade de armazenamento, necessária para guardar os estados das células bem

como as estruturas de dados utilizadas no cálculo da entropia. A quantidade de comunicação exigida também é bem expressiva, já que, além das informações do histórico de acessos que são enviadas ao processador mestre (3.1) existe a troca de mensagens contendo os estados remotos necessários para a simulação do autômato.

O ponto mais importante dos requisitos de comunicação de (3.1) é que o logaritmo que aparece em (2.6) pode ser calculado somente após as contribuições a  $q_i^t$  terem sido consideradas para todas as  $X_1 \dots X_d$  células, em particular para todas as células alocadas fora do processador  $P$ , cujos acessos são armazenados no histórico de acessos dos processadores participantes. Segundo [20], para obtermos uma aproximação “centrada na célula” para (2.6) que permita que os requisitos de comunicação sejam drasticamente reduzidos, o passo inicial é examinar mais cautelosamente o argumento deste logaritmo e perceber que  $\frac{q_i^t}{X_1 \dots X_d}$  é o número médio de ocorrências da  $i$ -ésima entrada por célula dentro da janela de tamanho  $X_1 \times \dots \times X_d \times T$  estados.

Seja  $q_i^{c,t}$  o número de ocorrências da  $i$ -ésima entrada para a  $c$ -ésima célula dentro da janela que termina no tempo  $t$ . Logo,  $q_i^t = \sum_{c=1}^{X_1 \dots X_d} q_i^{c,t}$  e então (2.6) pode ser reescrita como:

$$\begin{aligned}
 E_f^t(X_1, \dots, X_d, T) &= - \sum_{i=1}^{s^{1+\delta}} \frac{\sum_{c=1}^{X_1 \dots X_d} q_i^{c,t}}{X_1 \dots X_d T} \log_s \left( \frac{\sum_{c=1}^{X_1 \dots X_d} q_i^{c,t}}{X_1 \dots X_d T} \right) \\
 &= - \frac{1}{X_1 \dots X_d} \sum_{c=1}^{X_1 \dots X_d} \sum_{i=1}^{s^{1+\delta}} \left( \frac{q_i^{c,t}}{T} \right) \log_s \left( \frac{\sum_{c'=1}^{X_1 \dots X_d} q_i^{c',t}}{X_1 \dots X_d T} \right). \quad (3.2)
 \end{aligned}$$

Sempre que conveniente, podemos definir a entropia de entradas centrada na célula pela aproximação de  $q_i^{c,t}$  por sua média sobre todas as células na janela. Ou

seja, usamos a aproximação

$$\frac{\sum_{c'=1}^{X_1 \dots X_d} q_i^{c',t}}{X_1 \dots X_d} \approx q_i^{c,t} \quad (3.3)$$

no argumento para o logaritmo em (3.2) para todo  $c$  tal que  $1 \leq c \leq X_1 \dots X_d$ . A partir de então obtemos

$$C_f^t(X_1, \dots, X_d, T) = -\frac{1}{X_1 \dots X_d} \sum_{c=1}^{X_1 \dots X_d} \sum_{i=1}^{s^{1+\delta}} \left( \frac{q_i^{c,t}}{T} \right) \log_s \left( \frac{q_i^{c,t}}{T} \right), \quad (3.4)$$

que é a entropia de entradas centrada na célula da janela com  $X_1 \times \dots \times X_d \times T$  estados que termina no tempo  $t$  [20].

Considerando a entropia de entradas centrada na célula, analisamos o volume de comunicação direcionada ao processador especial  $P$  durante uma simulação paralela de um autômato celular. Claramente, um processador  $Q \neq P$  alocando  $X_Q$  células pode calcular sua porção do somatório duplo em (3.4) para cada  $t$  (e.g., com  $c$  variando sobre  $X_Q$  células) sem nenhuma comunicação. Se  $N$  denota o número de processadores, então  $P$  tem que receber  $(N-1) \times t_+$  números de ponto flutuante para a simulação completa, sendo que cada número é a entropia de entradas centrada na célula para cada tempo  $t$ . Sendo que esta entropia é a parcela pertencente à parte do autômato que está alojada naquele processador. Assim, se desconsiderarmos qualquer diferença entre o envio de números inteiros e números de ponto flutuante, o uso da aproximação centrada na célula para a entropia de entradas economiza uma quantidade considerável de comunicação, reduzindo os requisitos em (3.1) para  $O(N \times t_+)$ .

Na prática, esta redução por um fator de  $O(s^{(1+\delta)})$  dos requisitos de comu-



nicação traz grandes benefícios para a simulação paralela de autômatos celulares. Isto porque, a partir desta nova abordagem, cada processador é capaz de calcular sua parcela da entropia sem a necessidade de comunicação. Além disso, a árdua tarefa do processador mestre de colher as informações a cada passo de tempo e calcular a entropia globalmente é postergada para o final da simulação, o que é um ganho expressivo no tempo total de computação.

## 3.2 Entropia de transições

Observamos que a forma funcional de (3.4) é sugestiva de inúmeras possibilidades de critérios diferentes que podem ser experimentados na tentativa de classificação das regras de atualização de autômatos celulares. A possibilidade que consideramos neste trabalho é a seguinte. Para uma célula qualquer  $c$  dentro do bloco  $X_1 \times \dots \times X_d \times T$  que termina no tempo  $t$ , seja  $\tau^{c,t}$  o número de transições de estado dentro do bloco que provoca mudança de estado na célula durante a simulação. Este valor não depende explicitamente das entradas na tabela de regras que levam aos estados das células, assim, considerando que existam  $T-1$  transições de estado por célula dentro do bloco, ao calcularmos a entropia centrada na célula correspondente a partir de (3.4), obtemos

$$T_f^t(X_1, \dots, X_d, T) = -\frac{1}{X_1 \dots X_d} \sum_{c=1}^{X_1 \dots X_d} \left( \frac{\tau^{c,t}}{T-1} \right) \log_s \left( \frac{\tau^{c,t}}{T-1} \right). \quad (3.5)$$

Chamamos a grandeza em (3.5) de entropia de transições centrada na célula relativa ao bloco que termina no tempo  $t$ . Naturalmente, esta medida compartilha

com a entropia de entradas centrada na célula todas as características relevantes relacionadas à simulação paralela de autômatos celulares. A diferença básica é que o histórico relevante a ser mantido por cada processador durante a simulação paralela, conforme mencionado na Seção 3.1, agora é o histórico de transições que acontecem dentro da janela de tamanho  $T$  passos de tempo, em oposição ao histórico de acessos às  $s^{(1+\delta)}$  entradas da tabela dentro de uma mesma janela. A Figura 3.1 ilustra como a simulação paralela é realizada e mostra, também, as mensagens que são enviadas pelos  $N - 1$  processadores participantes para o processador mestre. Todavia, não apresentamos nesta ilustração as mensagens contendo o conjunto de estados necessários ao cálculo do próximo passo de tempo da evolução. Tratamos desse assunto no Capítulo 4.

A questão essencial é, naturalmente, saber se a entropia de transições centrada na célula definida em (3.5) possui, ainda, capacidade discriminatória semelhante àquela da entropia de entradas. Como há de ficar aparente no Capítulo 4, ela também oferece indícios interessantes na categorização da regra de atualização subjacente quando analisada sob a perspectiva de sua média  $\overline{T}_f(X_1, \dots, X_d, T)$  e de sua variância  $\sigma^2(T_f(X_1, \dots, X_d, T))$ , definidas em analogia a (2.7) e (2.8), respectivamente.

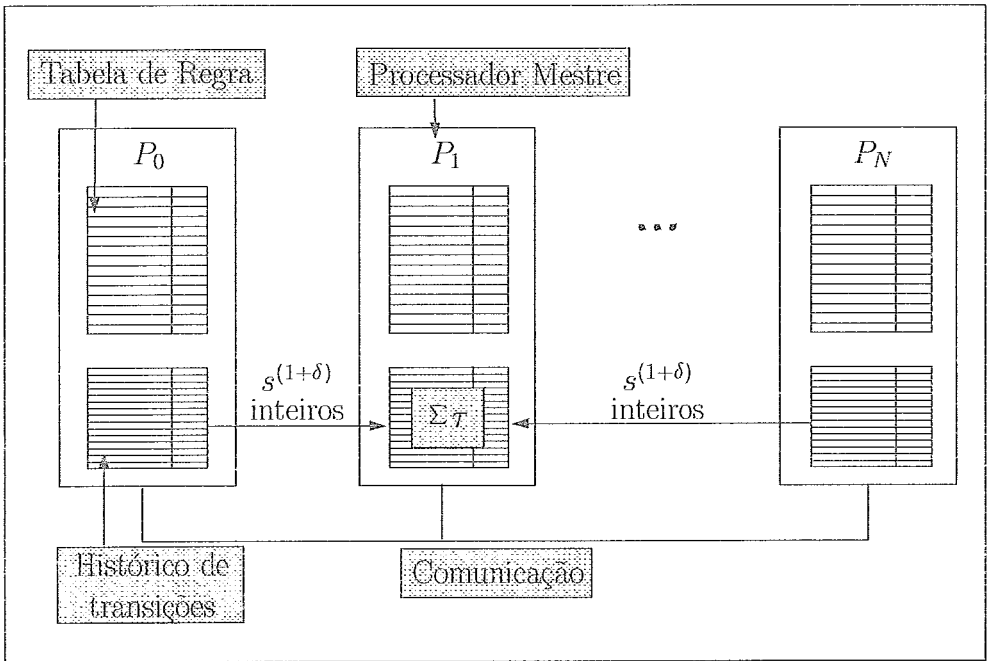


Figura 3.1: Simulação paralela de autômatos celulares usando  $N$  processadores. Cada processador contém uma cópia da tabela de regras e também uma estrutura de dados para armazenar o histórico de transições. O histórico de transições é enviado ao processador mestre que é o responsável pelo cálculo das entropias, da média e da variância.

### 3.3 Limite superior

Podemos estabelecer o limite superior da entropia de transições centrada na célula de (3.5) da seguinte maneira. Consideremos a expressão

$$-\left(\frac{\tau^{c,t}}{T-1}\right) \log_s \left(\frac{\tau^{c,t}}{T-1}\right), \quad (3.6)$$

extraída de (3.5) e façamos  $x = \tau^{c,t}/(T-1)$ . Assim obtemos a função

$$f(x) = -x \log_s x, \quad (3.7)$$

da qual devemos encontrar seu máximo. Para tanto, começamos lembrando que, segundo as propriedades dos logaritmos,  $\log_s x = \frac{\ln x}{\ln s}$ . Agora podemos reescrever (3.7) como

$$f(x) = -x \frac{\ln x}{\ln s}. \quad (3.8)$$

Calculando a derivada de primeira ordem de (3.8) temos

$$f'(x) = \frac{-1 - \ln x}{\ln s}. \quad (3.9)$$

Sabemos que  $f(x)$  é máxima quando  $f'(x) = 0$  e para que isto se verifique em (3.9) basta fazer

$$\ln x = -1. \quad (3.10)$$

Outra vez usando das propriedades dos logaritmos, concluímos que

$$x = e^{-1}, \quad (3.11)$$

é o valor que maximiza (3.7). Agora basta atribuir este valor a  $\tau^{c,t}/(T-1)$  para todos os valores apropriados de  $c$ , produzindo assim

$$T_f^t(X_1, \dots, X_d, T) \leq \frac{e^{-1}}{\ln s}. \quad (3.12)$$

Alguns dos resultados expostos no Capítulo 4 são baseados nas regras de atualização totalísticas tratadas no Capítulo 2, ou seja, regras de atualização cujos resultados não dependem da análise do estado individual da célula e dos estados de seus vizinhos. Porém, da análise do estado da célula e da soma dos estados dos seus vizinhos. Todavia, para tais regras de atualização, e considerando o caso  $s = 2$  apenas, o limite dado por (3.12) permanece correto e fornece um valor aproximado de 0.53.

Utilizando a noção de entropia de entradas introduzida por Wuensche, abordamos neste capítulo a classificação das regras de atualização através da análise qualitativa dos padrões espaço-temporais gerados através da execução do autômato. A partir desta idéia, introduzimos a nossa heurística de entropia de transições centrada na célula, com a qual procuramos elaborar um meio de classificação que mantivesse a capacidade discriminatória da entropia de entradas e que pudesse amenizar a grande demanda de comunicação necessária para a classificação de regras de atualização de autômatos celulares simulados em paralelo. Mostramos e discutimos, no capítulo seguinte, todos os experimentos que foram realizados com objetivo de provar nossa heurística centrada na célula.

# Capítulo 4

## Experimentos e análise de resultados

### 4.1 Autômatos celulares infinitos

Uma primeira decisão crucial ao se planejar a simulação de autômatos celulares infinitos é determinar quais células contíguas devem ser observadas ao longo de cada dimensão  $d$  e também o número de passos  $t_+$  durante os quais deve durar a simulação. Isso porque um autômato celular infinito não pode ser simulado em sua totalidade devido às restrições de tempo de computação e armazenamento, tampouco pode uma porção dele ser simulada durante um número indefinido de passos. Escolher um número finito de células para observação implica no problema de como lidar com as fronteiras da região observada, visto que estas fronteiras afetam a simulação mas não podem ser estendidas indefinidamente. Poderíamos adotar fronteiras cilíndricas artificiais ou alimentar as células fronteiriças com valores escolhidos aleatoriamente a cada passo de tempo da simulação. Entretanto, em princípio, isto não bastaria, uma vez que teria impacto direto na natureza infinita e determinística de um autômato

celular.

A solução vem primeiramente da escolha do valor  $t_+$  do número de passos durante os quais o autômato celular é observado na simulação, assim como o número  $\ell_k$  de células contíguas que são observadas ao longo da  $k$ -ésima dimensão,  $1 \leq k \leq d$ . A partir dessas decisões é que tratamos o problema. Mostramos, inicialmente, a solução para o caso de autômatos celulares uni-dimensionais e generalizaremos para dimensões maiores.

Considere  $\ell_1$  o número de células adjacentes que desejamos observar ao longo dos  $t_+$  passos da simulação de um autômato celular uni-dimensional infinito com vizinhança de raio  $r_1$ . Partimos da análise do último passo de tempo da simulação,  $t_+$ . Supondo  $c_1$  a primeira célula ao longo de  $\ell_1$ , da esquerda para a direita, e  $c_{\ell_1}$  a última célula. Logo, para a obtenção dos estados de  $c_1$  e  $c_{\ell_1}$  no passo de tempo  $t_+$  devemos considerar os estados das suas células vizinhas no tempo  $t_+ - 1$ . No caso de  $c_1$ , é necessário considerar  $r_1$  células vizinhas, à sua esquerda, enquanto no caso de  $c_{\ell_1}$ ,  $r_1$  células vizinhas à direita. Contudo,  $c_1$  e  $c_{\ell_1}$  são células fronteiriças, logo é necessário que no tempo  $t_+ - 1$  sejam inseridas artificialmente  $r_1$  células vizinhas à esquerda de  $c_1$  e igual quantidade de células à direita de  $c_{\ell_1}$ , dado que esta é a única forma possível de computar os estados de  $c_1$  e  $c_{\ell_1}$  no passo de tempo  $t_+$ .

Todavia, as células inseridas artificialmente no autômato no tempo  $t_+ - 1$ , também, requerem que seus estados nesse passo de tempo sejam computados a partir de suas vizinhanças no tempo  $t_+ - 2$ . Deste modo,  $r_1$  células são inseridas, novamente, à esquerda e à direita dessas células “artificiais” que são fronteiriças no

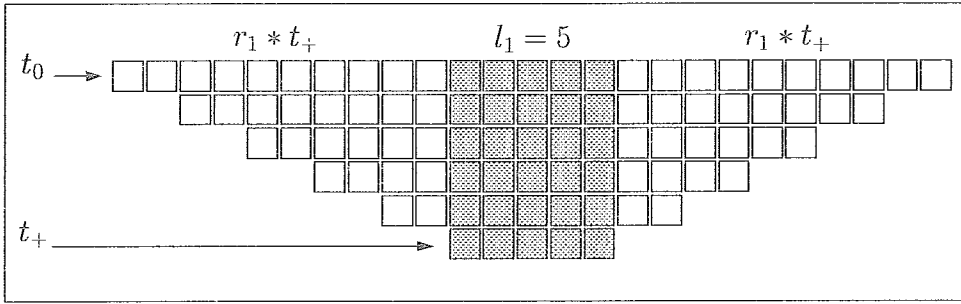


Figura 4.1: Simulação de um autômato celular uni-dimensional infinito com vizinhança de raio  $r_1 = 2$  e com  $\ell_1 = 5$  células contíguas sob observação ao longo de  $t_+ = 6$  passos de tempo. Neste caso, o número total de células no tempo  $t_0$  é de 25 células.

tempo  $t_+ - 1$ . Podemos observar claramente que este processo se repetirá regressivamente até o passo de tempo inicial  $t_0$ . Ou seja, serão inseridas  $2 \times r_1$  células a cada passo de tempo desde  $t_+ - 1$  até  $t_0$ . Portanto, para observar as  $\ell_1$  células contíguas de um autômato celular uni-dimensional infinito, com vizinhança de raio  $r_1$ , ao longo de  $t_+$  passos de tempo são necessárias

$$(2 \times r_1) \times (t_+ - 1) + \ell_1 \quad (4.1)$$

células no tempo  $t_0$ .

Na Figura 4.1 vemos a simulação de um autômato celular uni-dimensional infinito com vizinhança de raio  $r_1 = 2$  e com  $\ell_1 = 5$  células adjacentes sob observação ao longo de  $t_+ = 6$  passos de tempo. Neste caso, o número total de células no tempo  $t_0$  é de 25 células.

Precisamos generalizar a solução dada no caso uni-dimensional para autômatos com  $d > 1$ . Para isso, começamos assumindo uma vizinhança de von Neumann



e então dividimos a vizinhança  $\delta$  da célula considerando cada dimensão; ou seja, escrevemos  $\delta = 2(r_1 + \dots + r_d)$ , onde cada  $r_k$  é o raio da regra de atualização da  $k$ -ésima dimensão. Para obter o estado de uma célula fronteira ao longo da  $k$ -ésima dimensão no tempo  $t_+$ , como se ela realmente estivesse embutida em um autômato celular infinito, precisamos dos estados das  $r_k$  células “artificiais”, além da fronteira daquela dimensão, no tempo  $t_+ - 1$ . Dado que o número de células fronteiriças na  $k$ -ésima dimensão no tempo  $t_+$  é

$$2 \prod_{\substack{l=1 \\ l \neq k}}^d \ell_l, \quad (4.2)$$

portanto, de forma análoga a (4.1), o número total de células no tempo  $t_+ - 1$  é

$$\ell_1 \dots \ell_d + 2 \sum_{k=1}^d r_k \prod_{\substack{l=1 \\ l \neq k}}^d \ell_l \leq \prod_{k=1}^d (\ell_k + 2r_k). \quad (4.3)$$

Para  $d = 1$ , observamos que, a igualdade em (4.3) nos leva a (4.1). Entretanto, é importante considerar o limite superior, pois ele generaliza o número de células necessárias até o tempo  $t = 0$ . O número de células iniciais necessárias, nesse caso, não excede

$$\prod_{k=1}^d [\ell_k + 2r_k(t_+ - 1)] \quad (4.4)$$

O limite superior dado por (4.4) é um exagero para  $d > 1$  sob a vizinhança de von Neumann, dado que ele corresponde a um conjunto de células de tamanho  $(\ell_1 + 2r_1 t_+) \times \dots \times (\ell_d + 2r_d t_+)$ . Entretanto, queremos apenas mostrar que é perfeitamente viável determinar o número de estados iniciais que são necessários a uma simulação envolvendo exatamente aquele número de células, independente de todas as implicações inerentes a uma simulação paralela envolvendo tais células.

De agora em diante, temos um conjunto de células que pode ser simulado através do tempo  $t_+$  na certeza de que o comportamento observado no conjunto  $\ell_1 \dots \ell_d$  de células é perfeitamente compatível com a suposição de um autômato celular infinito e também do caráter determinístico de sua regra de atualização. Agora, as fronteiras são tratadas de forma abstrata, apesar de ainda existirem em relação ao prolongamento do conjunto de células sob observação. Assim, é indiferente o uso de fronteiras cilíndricas ou de estados escolhidos aleatoriamente para preencher as entradas necessárias para as células “artificiais” inseridas, já que somente os estados das células sob observação após o tempo  $t_+$  podem ser afetados pelo efeito de qualquer escolha.

Se considerarmos uma vizinhança de Moore, temos  $\delta = (1 + 2r_1) \dots (1 + 2r_d) - 1$ , portanto o número de estados necessários no tempo  $t_+ - 1$  é exatamente o limite superior apresentado em (4.3). Neste caso, o número de células dadas por (4.4) deixa de ser um exagero para expressar o que é necessário com exatidão.

Apresentamos, na Figura 4.2, uma ilustração destas situações no caso bi-dimensional, quando  $\ell_1 = 3$  e  $\ell_2 = 4$  com  $r_1 = 2$  e  $r_2 = 1$ . Na figura é mostrado o conjunto de células necessárias no caso dos estados das células sombreadas estarem sob observação para os passos futuros no tempo  $t_+ = 3$ , como se as células fossem parte de um autômato celular infinito e determinístico. O conjunto de células em branco representa as células necessárias no caso de uma vizinhança de von Neumann. O conjunto de células necessárias no caso de uma vizinhança de Moore é composto pela união das células em branco com as células tracejadas. Independente do tipo de

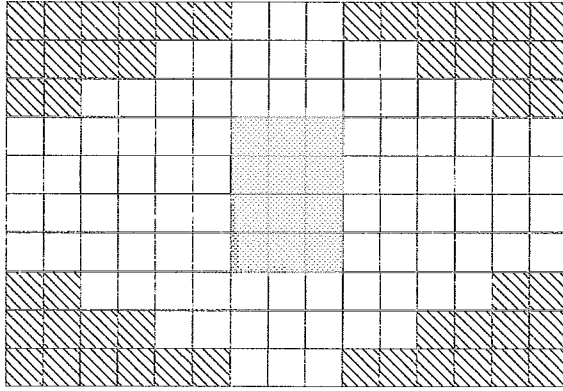


Figura 4.2: Conjunto  $15 \times 10$  de células para as quais os estados iniciais são necessários no caso bi-dimensional de  $\ell_1 = 3$ ,  $\ell_2 = 4$ ,  $r_1 = 2$ , e  $r_2 = 1$ , de forma que as células sombreadas possam ser observadas corretamente para  $t_+ = 3$ . O conjunto de células em branco contém as células necessárias no caso de uma vizinhança de von Neumann. A soma das células em branco mais as células tracejadas compõe o conjunto de células necessárias no caso de uma vizinhança de Moore ser usada.

vizinhança usada, empregamos o último conjunto de células, deste ponto em diante.

## 4.2 O valor de $T$

Ao simularmos um autômato celular com o objetivo de computar a entropia de entradas da Seção 2.3 ou a entropia de transições centrada na célula da Seção 3.2 dentro de uma janela de tamanho  $X_1 \times \dots \times X_d \times T$ , o núcleo do conjunto de células observadas é tal que  $\ell_1 = X_1, \dots, \ell_d = X_d$ . Nesta seção, investigamos a escolha de  $T$  que melhor favorece a capacidade classificatória de nossa heurística centrada na célula considerando as classes de Wolfram. Consideramos  $S = \{0, 1\}$ , i.e.,  $s = 2$ , de agora em diante.

Nosso objetivo é executar um conjunto inicial de experimentos com  $t_+ = 500$

em apenas um processador e analisar suas saídas com a finalidade de encontrar o valor de  $T$  mais adequado para ser usado nos demais experimentos principais. Esses experimentos iniciais se dividem em quatro casos, considerando:  $d = 1$  e  $r_1 = 2$ ,  $d = 1$  e  $r_1 = 3$ ,  $d = 2$  sob a vizinhança de von Neumann com  $r_1 = r_2 = 1$ , e o último para  $d = 2$  sob a vizinhança de Moore com  $r_1 = r_2 = 1$ . Em cada caso a primeira execução corresponde a uma regra de atualização classe (i), a segunda corresponde a uma regra classe (ii) e assim sucessivamente. Na Tabela 4.1 estão detalhadas as regras, já conhecidas, que são usadas nesses testes.

As regras de atualização são especificadas na Tabela 4.1 de acordo com as convenções adotadas no Capítulo 2: Para os experimentos de uma dimensão, cada regra está na forma hexadecimal. O caso de duas dimensões sob vizinhança de von Neumann, cada regra está na forma hexadecimal mencionada neste trabalho e sugerida em [28]. O caso de duas dimensões sob Moore compreende apenas regras totalísticas. Para estas regras, adotamos a notação do “Jogo da vida” de Conway.

Em cada execução, o autômato celular é simulado para  $T = 5, 10, \dots, 250$ , na tentativa de definirmos um valor que maximize a separação entre as classes. Para cada simulação são calculadas a média e a variância da Seção 3.2, isto é,  $\overline{T_f}$  e  $\sigma^2(T_f)$ , onde  $f$  é a regra de atualização em questão. Todas as simulações compartilham o mesmo valor de  $d$ ,  $X_1, \dots, X_d$ ,  $r_1, \dots, r_d$  e partem da mesma configuração inicial, que é gerada no início para aquele grupo de simulações escolhendo-se aleatoriamente os estados das  $(X_1 + 2r_1t_+) \dots (X_d + 2r_d t_+)$  células envolvidas.

Os resultados destes experimentos iniciais são apresentados nas Figuras 4.3 e 4.4,

Tabela 4.1: Regras de atualização usadas para plotar as Figuras 4.3 e 4.4.

Experimento	Classe	Regra de atualização
$d = 1,$	(i)	1d000a20
$r_1 = 2,$	(ii)	01dc3610
Figuras 4.3(a–d)	(iii)	994a6a65
	(iv)	6c1e53a8
$d = 1,$	(i)	1df0000000f000000000000000000020
$r_1 = 3,$	(ii)	7fdc3610fc48472c01dc361001dc3660
Figuras 4.3(e–h)	(iii)	994f6a65994a6a65a94a6a65994a6a99
	(iv)	3b469c0ee4f7fa96f93b4d32b09ed0e0
$d = 2,$	(i)	00000601
$r_1 = r_2 = 1,$	(ii)	06900600
von Neumann,	(iii)	69969669, Fredkin2 [28]
Figuras 4.4(a–d)	(iv)	6db6fac8, Crystal2 [28]
$d = 2,$	(i)	b3b6b7 s3s6s7s8
$r_1 = r_2 = 1,$	(ii)	b3 s2s5s6
Moore,	(iii)	b1b3b5 s1s3s5
Figuras 4.4(e–h)	(iv)	b3 s2s3

as quais referem-se ao comportamento da entropia de transições centrada na célula para os casos uni e bi-dimensional, respectivamente, à medida em que  $T$  varia.

Ainda que este primeiro conjunto de experimentos seja desprovido de significado estatístico, dado que foi baseado em execuções a partir de uma única configuração inicial, ele fornece uma indicação inicial da capacidade discriminatória da nossa heurística centrada na célula. De fato, um exame de todos os gráficos de médias e variâncias nas Figuras 4.3 e 4.4 revela que, com poucas exceções, as classes (i)–(iv) podem, no pior caso, ser discriminadas por uma ordem de grandeza através da média ou da variância da entropia de transições centrada na célula para a maioria dos valores de  $T$ . Por exemplo, ao compararmos as Figuras 4.3(a) e (b) temos a indicação de que a entropia média fornece uma boa discriminação entre as quatro classes,

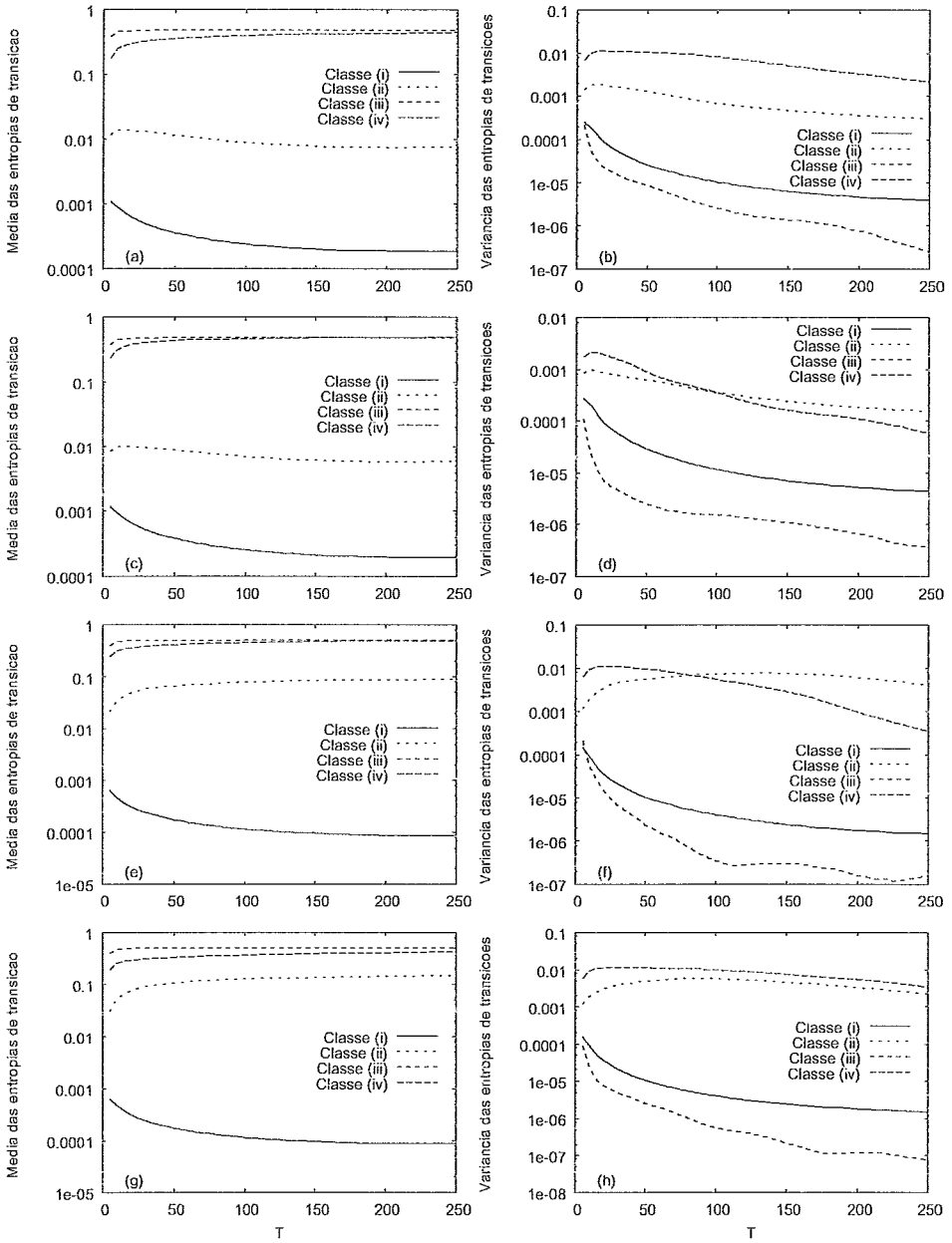


Figura 4.3: A média  $\overline{(T_f)}$  e a variância  $(\sigma^2(T_f))$  da entropia de transições centrada na célula como uma função de  $T$  sob quatro regras de atualização diferentes, uma para cada classe, de (i) até (iv), para  $d = 1$ . São mostrados os casos de 150 células com  $r_1 = 2$  (a e b), o caso de 300 células com  $r_1 = 2$  (c e d), o caso de 150 células com  $r_1 = 3$  (e e f) e o caso de 300 células com  $r_1 = 3$  (g e h).

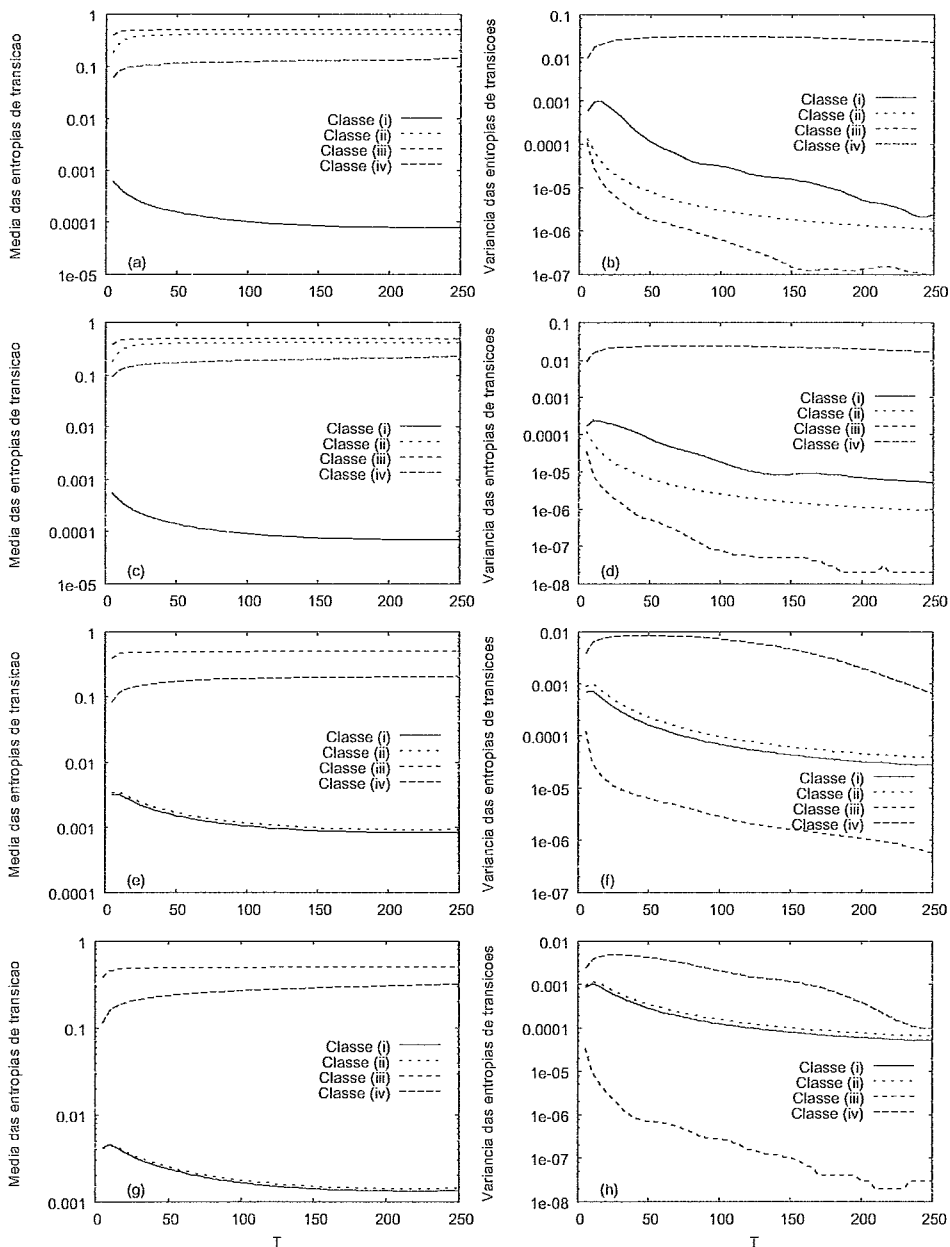


Figura 4.4: A média ( $\overline{T_f}$ ) e a variância ( $\sigma^2(T_f)$ ) da entropia de entradas centrada na célula como uma função de  $T$  sob quatro regras de atualização diferentes, uma para cada classe, de (i) até (iv), para  $d = 2$ . São mostrados o caso de  $(15 \times 15)$  células usando vizinhança de von Neumann (a e b), o caso de  $(30 \times 30)$  células sob von Neumann (c e d), o caso de  $(15 \times 15)$  células sob Moore (e e f) e o caso de  $(30 \times 30)$  células sob a vizinhança de Moore (g e h). Em todos os casos  $r_1 = r_2 = 1$ .

exceto entre as classes (iii) e (iv), que no entanto podem ser facilmente separadas pela variância. As exceções residem nos casos bi-dimensionais com vizinhança de Moore, nos quais nossa heurística não é capaz de distinguir bem as classes (i) e (ii).

Mais adiante, após termos fornecido dados adicionais, retornaremos a esta discussão da capacidade discriminatória de nossa heurística. Todavia, retornando ao objetivo original destes experimentos iniciais, podemos observar pelas Figuras 4.3 e 4.4 que há várias possibilidades para a escolha do valor de  $T$  que garantem uma boa separação entre as classes. É importante notar que ao escolher o menor valor possível surge a vantagem de aliviar a demanda de processamento para a computação da entropia. Considerando esse fato, nossa escolha daqui por diante é usar  $T = 25$ , pois este valor é um bom discriminador e traz ainda a vantagem de favorecer a velocidade de processamento de nossos experimentos.

Com a finalidade de complementar estes experimentos iniciais, mostramos nas Figuras 4.5, 4.6, 4.7 e 4.8, alguns padrões espaço-temporais gerados a partir da evolução dos autômatos celulares baseados nas regras de atualização da Tabela 4.1. Em todos os casos, representamos o estado 0 pela cor branca e o estado 1 pela cor preta.

O primeiro conjunto de padrões corresponde a autômatos celulares infinitos, ou seja, são mostrados os estados das células que contribuíram para o cálculo da entropia. Este conjunto é mostrado nas Figuras 4.5 e 4.6 para os casos uni e bi-dimensional, respectivamente.



Um segundo conjunto de padrões espaço-temporais é mostrado baseado na evolução dos mesmos estados iniciais, porém, com fronteiras cilíndricas. Este é mostrado nas Figuras 4.7 e 4.8 para os casos uni e bi-dimensional, respectivamente. Estes padrões estão relacionados com os dados apresentados na Tabela 4.2.

O terceiro conjunto de padrões, desta vez em 3 dimensões, é mostrado nas Figuras 4.9 a 4.16. Este conjunto representa tridimensionalmente os padrões apresentados nas Figuras 4.6 e 4.8. A geração desses padrões tridimensionais foi possibilitada através de um programa criado em linguagem C com OpenGL (*Open Graphics Library*) como parte dos requisitos deste trabalho. O programa é capaz de exibir em 3 dimensões, os padrões que estão originalmente armazenados em formato de texto simples e permite ao usuário mover a imagem como desejar para que possa visualizar cada detalhe ao longo de toda a evolução do autômato. É importante lembrar que o estado 0 é representado pelo espaço vazio e o estado 1 pelo cubo.

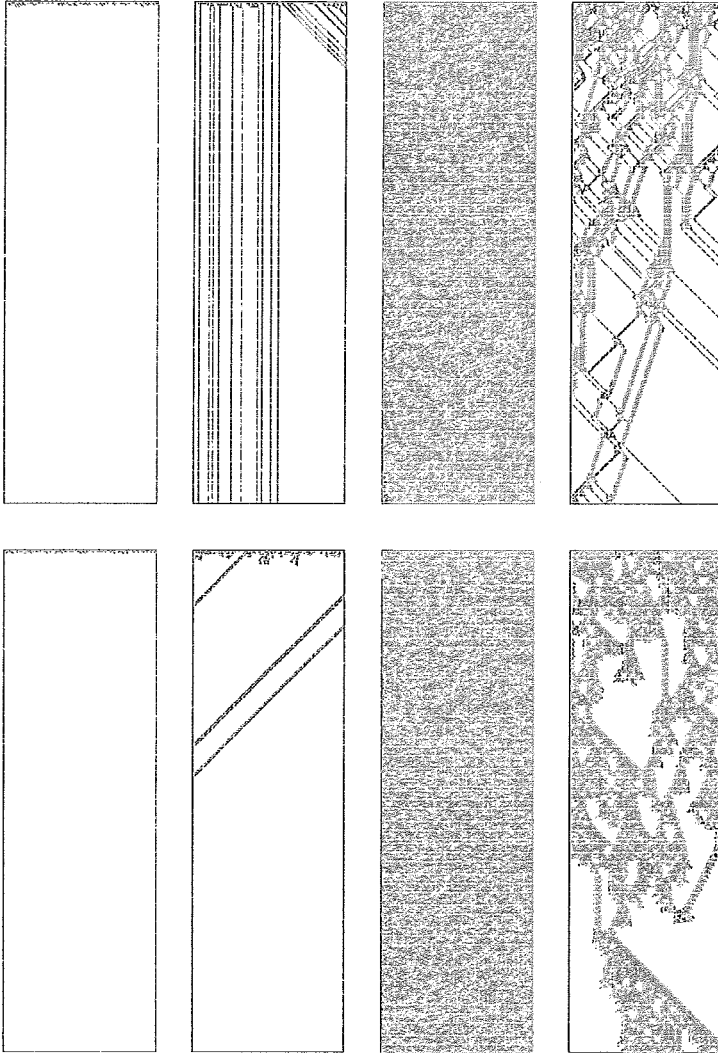


Figura 4.5: Padrões espaço-temporais para as regras de atualização apresentadas na Tabela 4.1, para  $d = 1$  e infinito, com 150 células observadas ao longo de 500 passos de tempo. Cada caso mostra os estados das células horizontalmente para cada passo de tempo; sendo que o tempo aumenta do topo para baixo. O primeiro caso mostrado corresponde a  $r_1 = 2$  e o segundo a  $r_1 = 3$ . Em cada linha, da esquerda para a direita, temos as regras de atualização pertencentes as classes (i)–(iv).

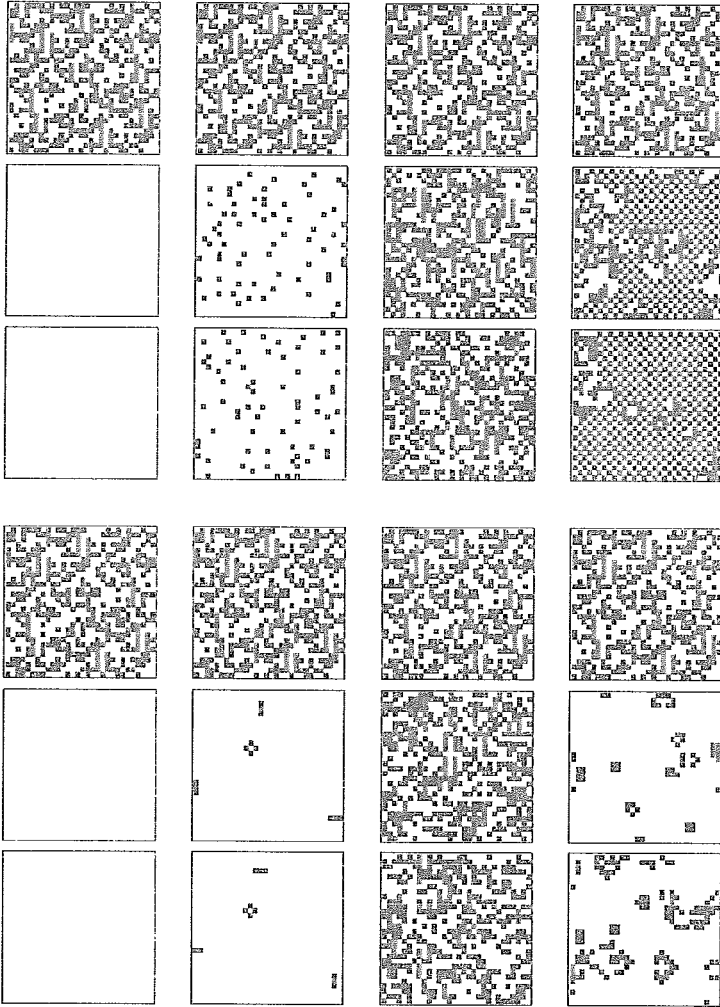


Figura 4.6: Padrões espaço-temporais para as regras de atualização apresentadas na Tabela 4.1, para  $d = 2$  e infinito, com  $30 \times 30$  células observadas. Cada quadro mostra uma configuração durante a evolução do autômato. Os quadros mais ao topo correspondem a regras de atualização de von Neumann e os quadros na parte inferior são relativos a regras de atualização de Moore. Em cada trio de linhas, a primeira linha corresponde a  $t = 0$ , a linha do meio a  $t = 125$  e última corresponde a  $t = 250$ . Em cada coluna, da esquerda para a direita, está uma regra de atualização pertencente às classes (i)–(iv), respectivamente.

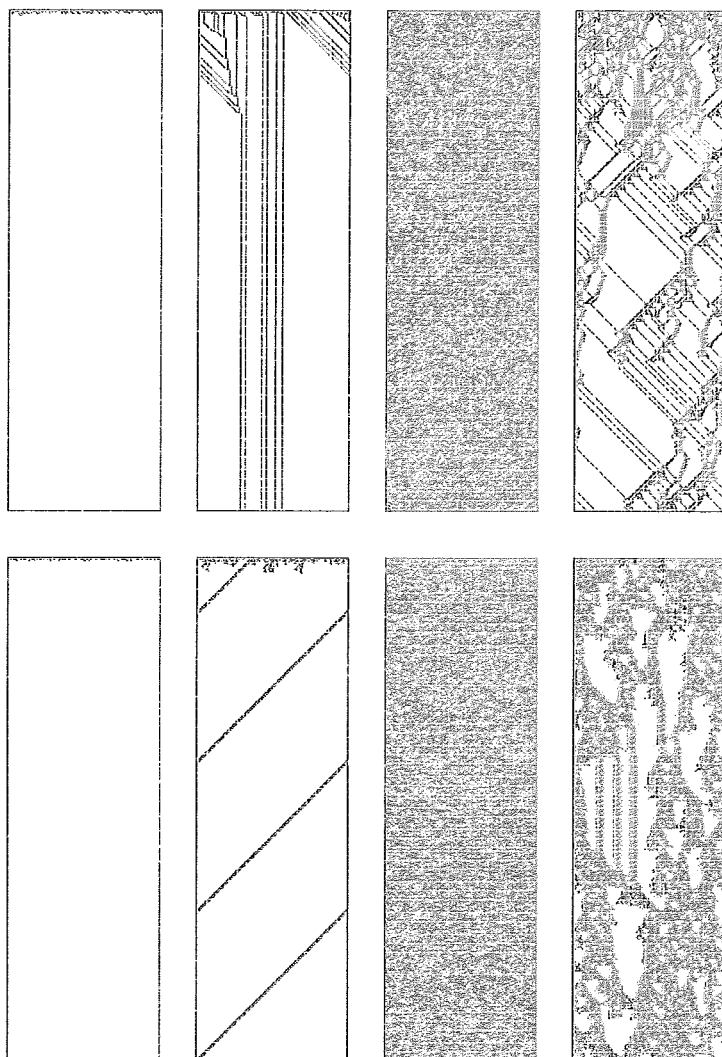


Figura 4.7: Padrões espaço-temporais para as regras de atualização apresentadas na Tabela 4.1, para  $d = 1$  e cilíndrico, com 150 células observadas ao longo de 500 passos de tempo. Cada caso mostra os estados das células horizontalmente para cada passo de tempo; sendo que o tempo aumenta do topo para baixo. O primeiro caso mostrado corresponde a  $r_1 = 2$  e o segundo a  $r_1 = 3$ . Em cada linha, da esquerda para a direita, temos as regras de atualização pertencentes as classes (i)–(iv).

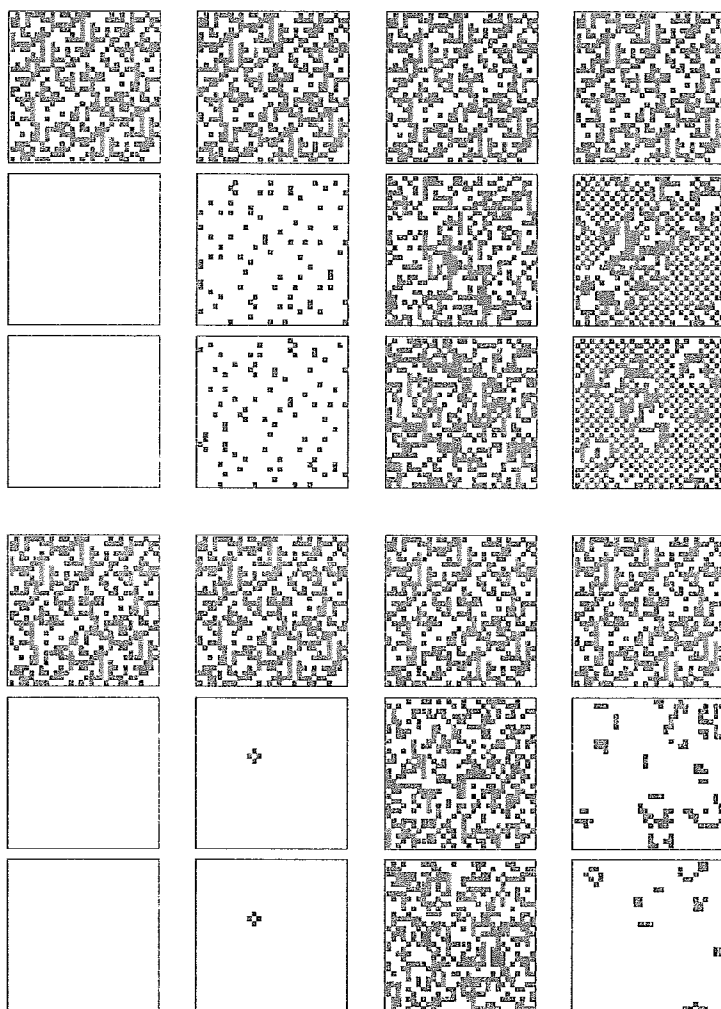


Figura 4.8: Padrões espaço-temporais para as regras de atualização apresentadas na Tabela 4.1, para  $d = 2$  e cilíndrico, com  $30 \times 30$  células observadas. Cada quadro mostra uma configuração durante a evolução do autômato. Os quadros mais ao topo correspondem a regras de atualização de von Neumann e os quadros na parte inferior são relativos a regras de atualização de Moore. Em cada trio de linhas, a primeira linha corresponde a  $t = 0$ , a linha do meio a  $t = 125$  e última corresponde a  $t = 250$ . Em cada coluna, da esquerda para a direita, está uma regra de atualização pertencente às classes (i)–(iv), respectivamente.

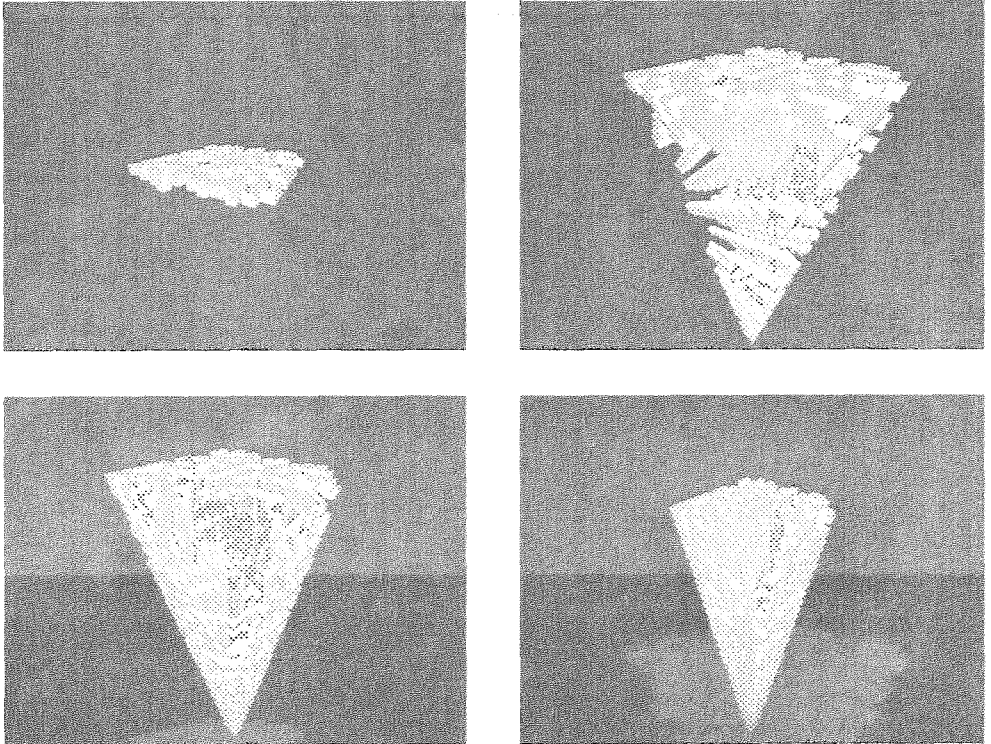


Figura 4.9: Padrões espaço-temporais para as regras de atualização apresentadas na Tabela 4.1, para  $d = 2$  e infinito, com  $15 \times 15$  células observadas. Cada quadro mostra uma configuração durante a evolução do autômato e correspondem a regras de atualização de von Neumann. Em cada quadro da esquerda para a direita e de cima para baixo, está uma regra de atualização pertencente às classes (i)- (iv), respectivamente.

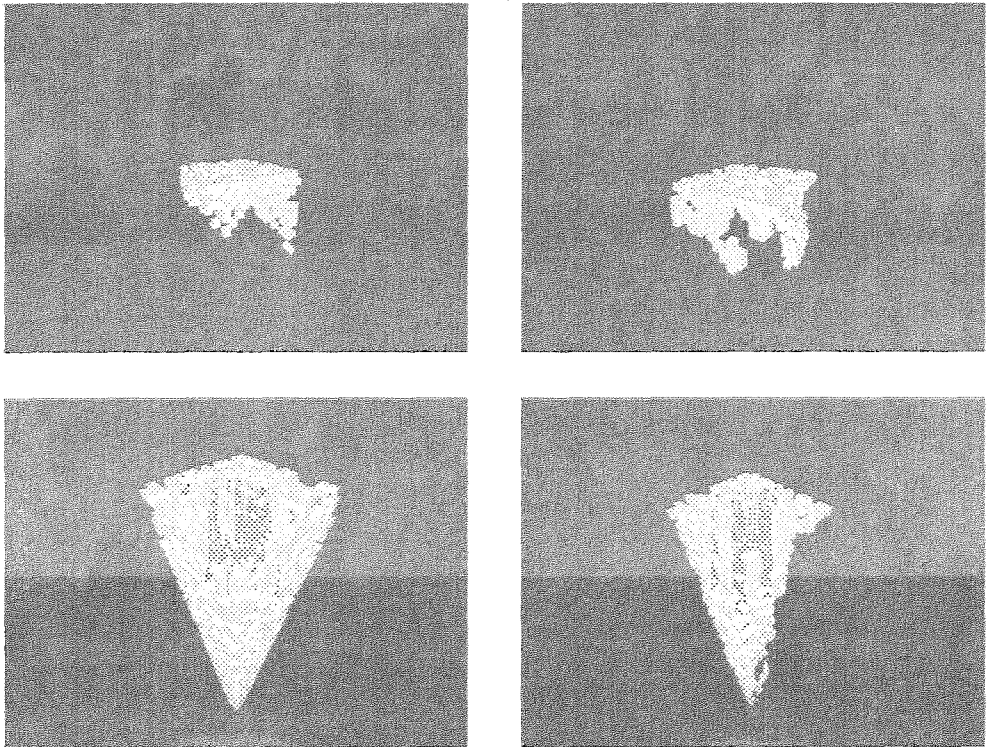


Figura 4.10: Padrões espaço-temporais para as regras de atualização apresentadas na Tabela 4.1, para  $d = 2$  e infinito, com  $15 \times 15$  células observadas. Cada quadro mostra uma configuração durante a evolução do autômato e correspondem a regras de atualização de Moore. Em cada quadro da esquerda para a direita e de cima para baixo, está uma regra de atualização pertencente às classes (i)–(iv), respectivamente.

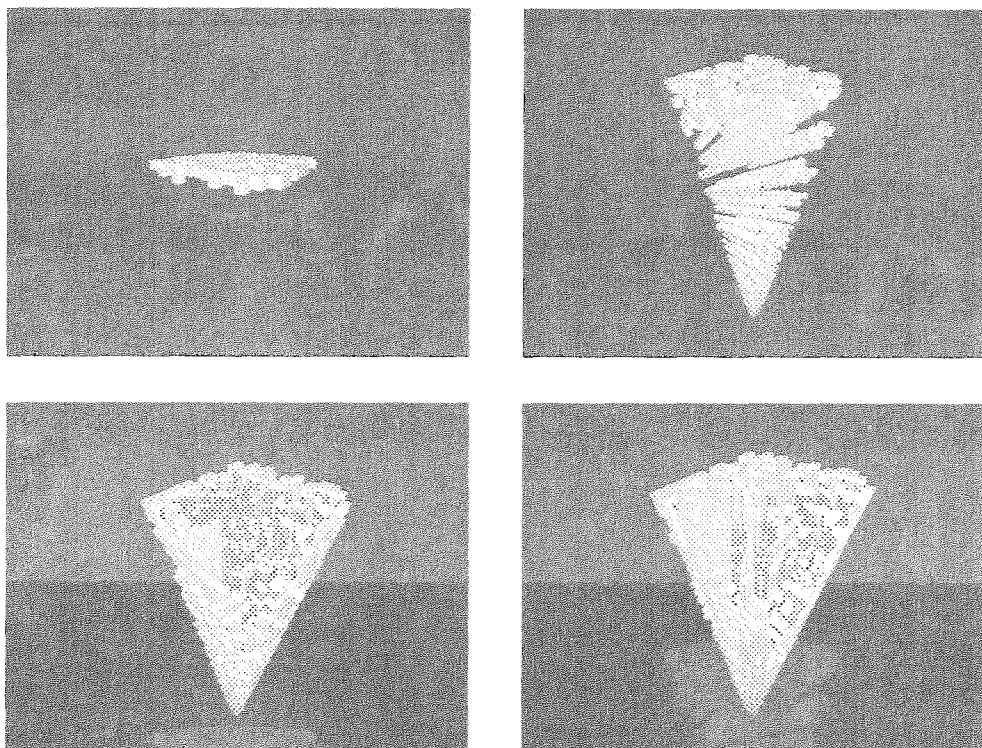


Figura 4.11: Padrões espaço-temporais para as regras de atualização apresentadas na Tabela 4.1, para  $d = 2$  e cilíndrico, com  $15 \times 15$  células observadas. Cada quadro mostra uma configuração durante a evolução do autômato e correspondem a regras de atualização de von Neumann. Em cada quadro da esquerda para a direita e de cima para baixo, está uma regra de atualização pertencente às classes (i)-(iv), respectivamente.



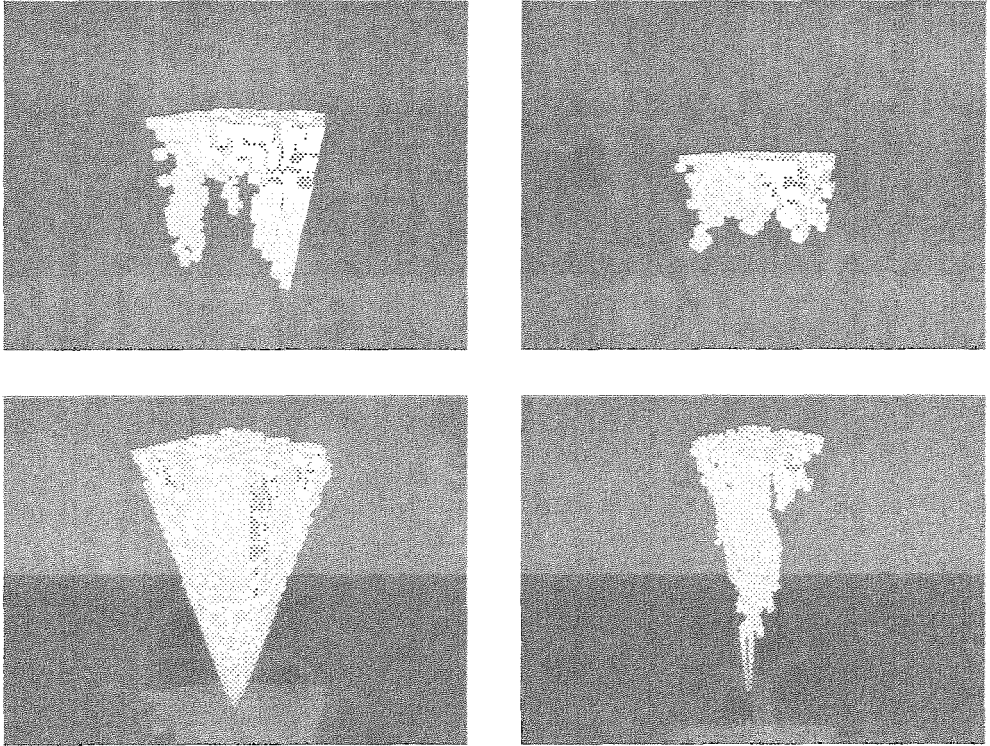


Figura 4.12: Padrões espaço-temporais para as regras de atualização apresentadas na Tabela 4.1, para  $d = 2$  e cilíndrico, com  $15 \times 15$  células observadas. Cada quadro mostra uma configuração durante a evolução do autômato e correspondem a regras de atualização de Moore. Em cada quadro da esquerda para a direita e de cima para baixo, está uma regra de atualização pertencente às classes (i)-(iv), respectivamente.

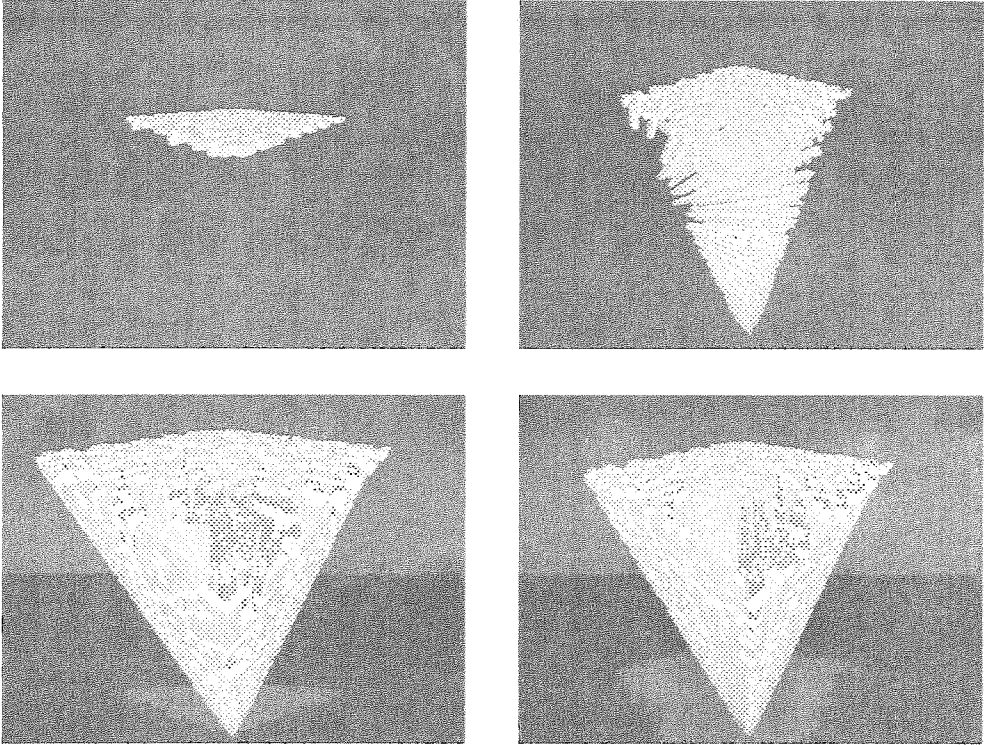


Figura 4.13: Padrões espaço-temporais para as regras de atualização apresentadas na Tabela 4.1, para  $d = 2$  e infinito, com  $30 \times 30$  células observadas. Cada quadro mostra uma configuração durante a evolução do autômato e correspondem a regras de atualização de von Neumann. Em cada quadro da esquerda para a direita e de cima para baixo, está uma regra de atualização pertencente às classes (i)–(iv), respectivamente.

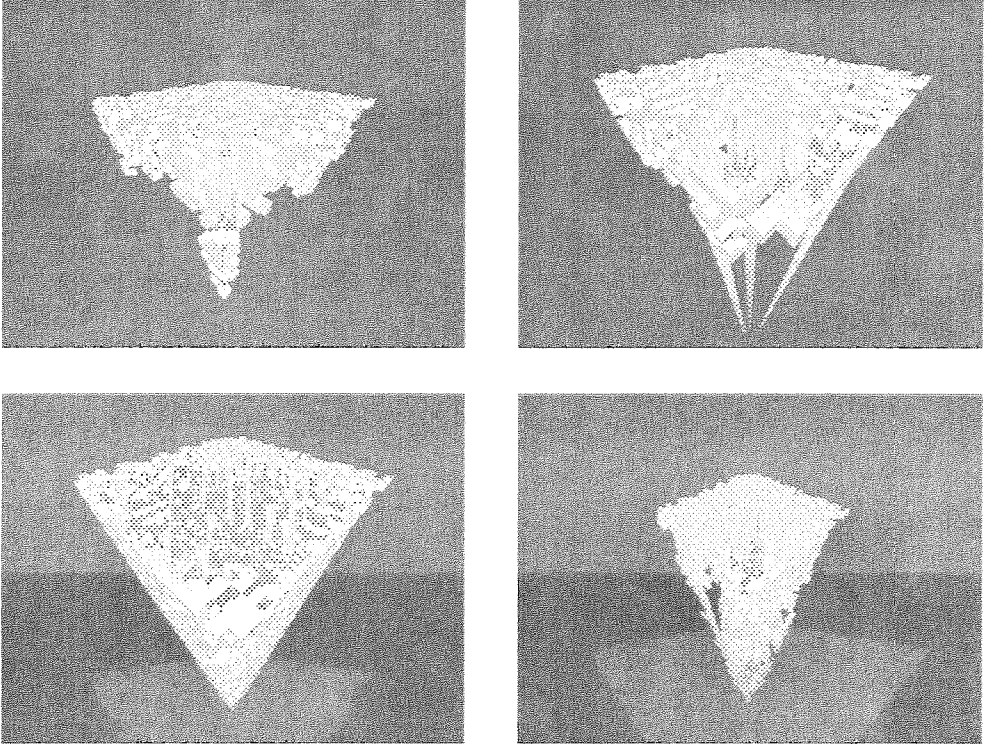


Figura 4.14: Padrões espaço-temporais para as regras de atualização apresentadas na Tabela 4.1, para  $d = 2$  e infinito, com  $30 \times 30$  células observadas. Cada quadro mostra uma configuração durante a evolução do autômato e correspondem a regras de atualização de Moore. Em cada quadro da esquerda para a direita e de cima para baixo, está uma regra de atualização pertencente às classes (i)-(iv), respectivamente.

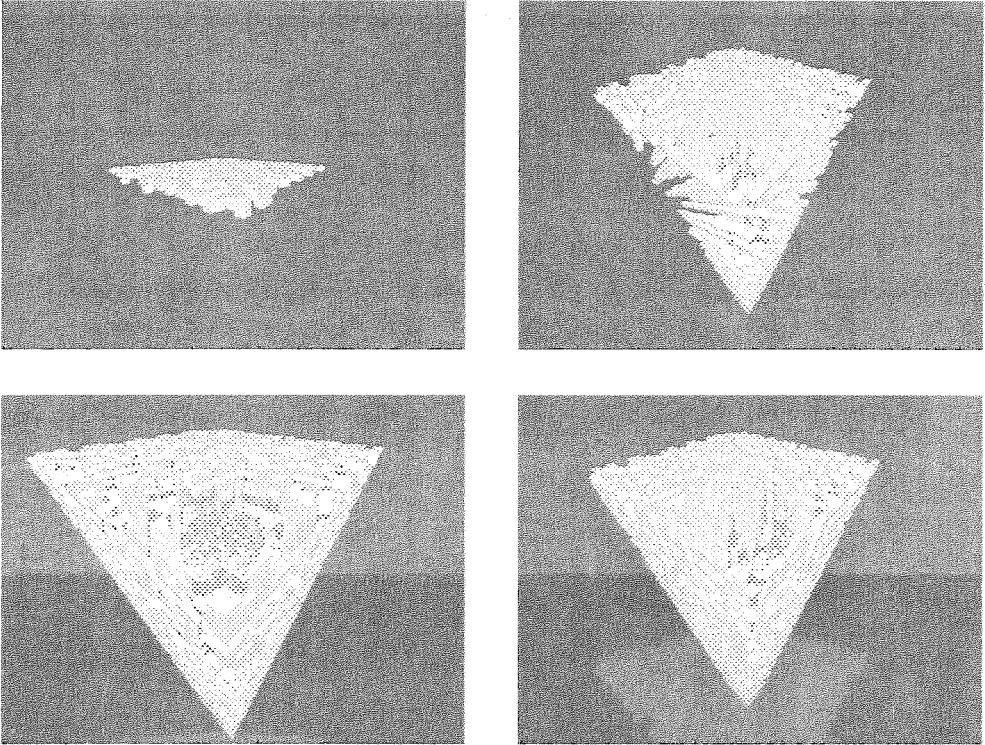


Figura 4.15: Padrões espaço-temporais para as regras de atualização apresentadas na Tabela 4.1, para  $d = 2$  e cilíndrico, com  $30 \times 30$  células observadas. Cada quadro mostra uma configuração durante a evolução do autômato e correspondem a regras de atualização de von Neumann. Em cada quadro da esquerda para a direita e de cima para baixo, está uma regra de atualização pertencente às classes (i)–(iv), respectivamente.

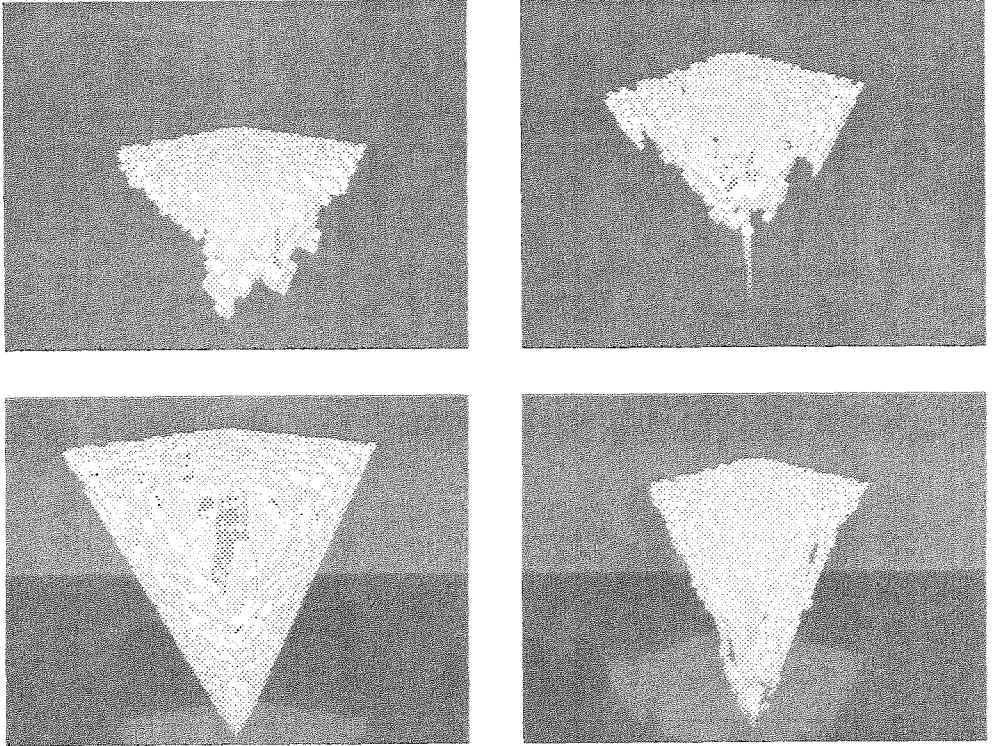


Figura 4.16: Padrões espaço-temporais para as regras de atualização apresentadas na Tabela 4.1, para  $d = 2$  e cilíndrico, com  $30 \times 30$  células observadas. Cada quadro mostra uma configuração durante a evolução do autômato e correspondem a regras de atualização de Moore. Em cada quadro da esquerda para a direita e de cima para baixo, está uma regra de atualização pertencente às classes (i)–(iv), respectivamente.

Tabela 4.2: Médias e variâncias extraídas de uma média de 5 simulações com estados iniciais escolhidos aleatoriamente, com fronteiras cilíndricas para  $t_+ = 500$  e  $T = 25$ . Os experimentos foram organizados da seguinte forma: (I)  $d = 1$ , 150 células,  $r_1 = 2$ ; (II)  $d = 1$ , 300 células,  $r_1 = 2$ ; (III)  $d = 1$ , 150 células,  $r_1 = 3$ ; (IV)  $d = 1$ , 300 células,  $r_1 = 3$ ; (V)  $d = 2$ , vizinhança de von Neumann,  $15 \times 15$  células,  $r_1 = r_2 = 1$ ; (VI)  $d = 2$ , vizinhança de von Neumann,  $30 \times 30$  células,  $r_1 = r_2 = 1$ ; (VII)  $d = 2$ , vizinhança de Moore,  $15 \times 15$  células,  $r_1 = r_2 = 1$ ; (VIII)  $d = 2$ , vizinhança de Moore,  $30 \times 30$  células,  $r_1 = r_2 = 1$ . As regras de atualização são aquelas dadas na Tabela 4.1, para as classes de (i) a (iv). Os números são truncados em seis casas decimais.

Experimento	$\overline{T_f}$			
	(i)	(ii)	(iii)	(iv)
I	0.000642	0.006438	0.474852	0.340475
II	0.000643	0.003924	0.474737	0.389267
III	0.000307	0.137617	0.489661	0.355899
IV	0.000310	0.102894	0.490509	0.290770
V	0.000237	0.335283	0.469148	0.231000
VI	0.000237	0.366104	0.484582	0.185903
VII	0.002447	0.002894	0.484342	0.042994
VIII	0.002582	0.003393	0.484713	0.094744
	$\sigma^2(T_f)$			
	(i)	(ii)	(iii)	(iv)
I	0.000080	0.000779	0.000010	0.005762
II	0.000079	0.000433	0.000007	0.002972
III	0.000031	0.000544	0.000009	0.006885
IV	0.000032	0.000585	0.000004	0.005747
V	0.000019	0.000005	0.000001	0.012932
VI	0.000019	0.000003	0.000000	0.016717
VII	0.000473	0.000532	0.000010	0.010095
VIII	0.000497	0.000609	0.000002	0.009627

### 4.3 Análise de resultados

Em suma, a simulação em paralelo de um autômato celular corresponde ao projeto de um algoritmo distribuído síncrono simples, da forma como está descrito em [4], empregando a técnica de  $\alpha$ -sincronização [1]. Podemos encontrar muitas referências sobre o assunto em [3, 7, 26]. Nosso simulador paralelo foi projetado e implementado usando igual estrutura tanto para autômatos celulares uni-dimensionais quanto para os bi-dimensionais.

A infra-estrutura utilizada nos experimentos consiste de um *cluster* com oito computadores pessoais (PCs), tendo todos eles um processador Intel Pentium 4 de 1.8 GHz com 1 *gigabyte* de memória. Os oito computadores são interconectados por um *switch* gigabit-ethernet. O simulador foi desenvolvido completamente na linguagem C, utilizando a biblioteca de passagem de mensagens MPI (*Message Passing Interface*). Cada simulação é iniciada pelo particionamento do autômato nos  $N = 8$  processadores disponíveis. Cada processador tem a capacidade de comunicação direta com todos os demais. Para  $d = 1$ , as células são particionadas entre os processadores de forma que cada processador receba um conjunto de células adjacentes para simular; para  $d = 2$ , o autômato é subdividido em retângulos de células adjacentes.

A distribuição das células para o caso bi-dimensional é ilustrada na Figura 4.17. Ao longo das duas dimensões temos um conjunto de células adjacentes para cada processador. Além das células pertencentes àquele processador, também, é

necessário armazenar cópias das células vizinhas necessárias para o cálculo dos estados das células que residem na fronteira entre um processador e outro, em cada dimensão, da maneira como está representada pela linha pontilhada. Portanto, é necessário obter os estados das células vizinhas delimitadas antes de ser aplicada a regra de atualização. Cabe ressaltar que o mesmo processo deve acontecer para todos os processadores e nas duas dimensões. Outra importante observação refere-se à maneira como as células fronteiriças do autômato são atualizadas. São dois os possíveis casos: no primeiro, escolhemos estados aleatórios para células complementares; no segundo caso, formamos um torus, associando os estados das células da base do autômato com as células do topo e as células mais à esquerda do processador  $P_1$  com as células mais à direita do processador  $P_N$ .

É importante observar que a mesma relação de vizinhança estabelecida entre as células pelo reticulado que define o autômato celular é sustentada na vizinhança entre processadores. Ou seja, dois processadores são vizinhos sempre que pelo menos uma célula que um deles aloja seja uma vizinha de uma célula abrigada pelo outro.

A simulação, propriamente dita, começa em cada processador com a escolha aleatória do estado inicial para cada uma das células atribuídas a este processador e a troca de cópias dos estados iniciais das células que estão na fronteira entre um processador e outro. O processador inicia então as iterações à medida em que  $t$  aumenta a partir de 0 até  $t_+$ . Em cada iteração novos estados são calculados e cópias das células em fronteira são trocadas entre os processadores. Também, são calculadas as porções da entropia definida em (3.5) correspondente às células que



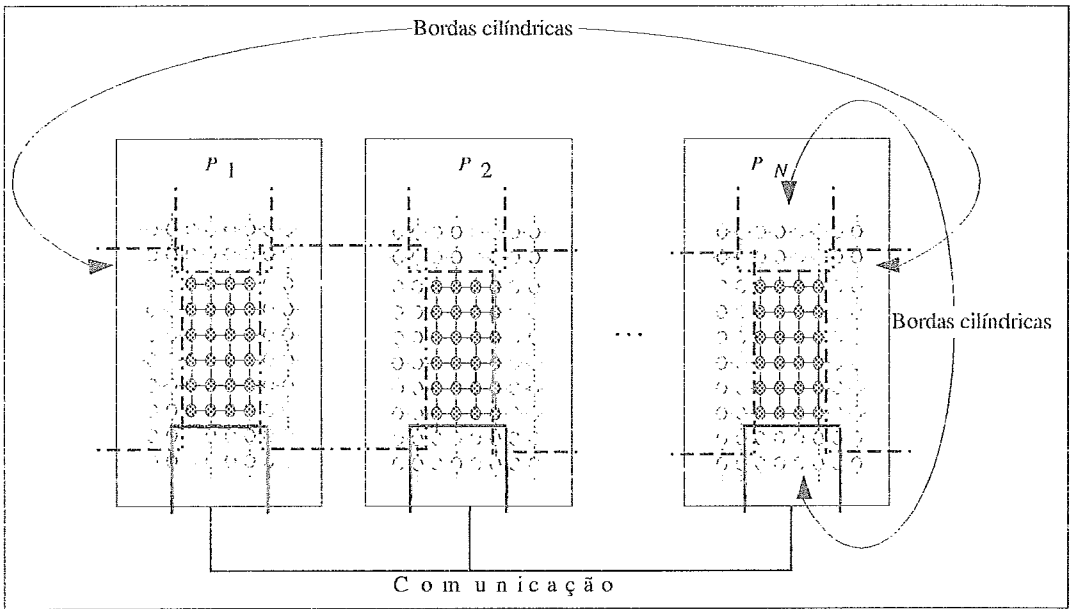


Figura 4.17: Distribuição das células de um autômato celular bi-dimensional ao longo de  $N$  processadores. Cada processador tem a capacidade de comunicação direta com todos os demais. As células cujos estados são necessários para o cálculo dos estados das células que residem na fronteira entre um processador e outro são copiadas.

estão sendo observadas. Ao final, cada processador que abriga pelo menos uma célula observada envia os seus  $2(t_+ - T + 2)$  resultados de entropia para o processador mestre, previamente escolhido, para o cálculo global da média e variância ( $\overline{T_f}$  e  $\sigma^2(T_f)$ ).

### **Autômatos celulares uni-dimensionais**

A configuração para nossos experimentos uni-dimensionais está baseada em  $r_1 = 2$  ou  $r_1 = 3$ . Para  $r_1 = 2$ , temos  $X_1 = 2000$ ,  $t_+ = 500$ , e  $T = 25$ . O número total de células a simular é então  $X_1 + 2r_1t_+ = 4000$ , portanto o número de células observadas é metade do total. Para  $r_1 = 3$ , temos  $X_1 = 2400$ ,  $t_+ = 400$ , e  $T = 25$ . Nesse caso, o número total de células na simulação é 4800, e novamente as células observadas são metade do número total de células. Tais valores, foram escolhidos estritamente por motivos práticos de manter os tempos de simulação dentro de um limite aceitável.

No caso uni-dimensional, o número de regras de atualização diversas é dado por  $2^{2^{1+2r_1}}$ , isto é,  $2^{32}$  para  $r_1 = 2$  e  $2^{128}$  para  $r_1 = 3$ . Basta recordarmos o espaço de regras de atualização abordado no Capítulo 2. Nossos resultados estão baseados em 50000 regras de atualização escolhidas aleatoriamente entre todas as possibilidades e são mostradas na Figura 4.18 como gráficos da variância  $\sigma^2(T_f)$  contra a média  $\overline{T_f}$ . O ponto correspondente à regra de atualização uni-dimensional da Tabela 4.1 não é mostrado explicitamente, contudo é destacado por indicação nas coordenadas da classe à qual a regra de atualização pertence.

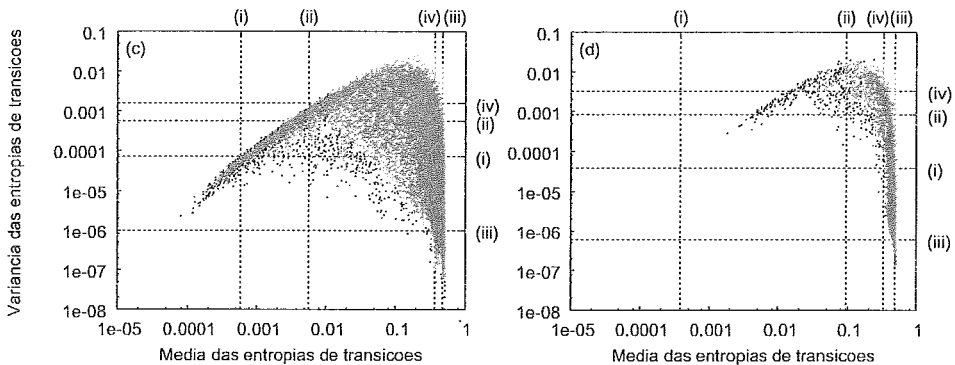


Figura 4.18: Ocorrência de pares de média-variância para autômatos celulares uni-dimensionais. Os dados são mostrados para a entropia de transições centrada na célula com  $r_1 = 2$  (a) e  $r_1 = 3$  (b) como gráficos de  $\sigma^2(T_f)$  contra  $\overline{T_f}$ . Cada gráfico contém 50000 pontos, cada ponto corresponde a uma regra de atualização escolhida aleatoriamente e a uma média sobre 5 configurações iniciais escolhidas aleatoriamente. As regras de atualização uni-dimensionais da Tabela 4.1 também são mostradas no mesmo experimento, mas não como pontos e sim apontadas com uma indicação nas coordenadas das classes (i)–(iv) à qual elas pertencem.

Uma informação crucial que não foi mencionada nos gráficos da Figura 4.18 refere-se à densidade dos pontos em alguma região particular da média e variância. Tratamos deste assunto a seguir. Em princípio, para cada um dos gráficos escolhemos um valor para a entropia média que separa as regras de atualização marcadas como (i) e (ii) daquelas classificadas como (iii) e (iv). Nas Figuras 4.18(a) e (b), esta entropia média pode ser considerada como sendo 0.1. Ao selecionar este valor particionamos o gráfico em duas regiões e para cada uma selecionamos uma variância que pode ser usada para separar as regras de atualização rotuladas como (i) e (ii) à esquerda, e outra que da mesma forma pode ser usada para regras marcadas como (iii) e (iv). Nossas escolhas nos lados esquerdo e direito são 0.0001 e 0.001, respectivamente. Ao final, em cada gráfico, temos uma partição em quatro regiões, cada

uma contendo exatamente uma das regras de atualização rotuladas (i)–(iv).

Podemos então, agora, fornecer as informações sobre a distribuição das regras de atualização. Na parte (a), 2.54% das regras de atualização estão dentro da região (i), 9.30% na região (ii), 83.47% na região (iii) e 4.69% na região (iv) e na parte (b) não há qualquer regra de atualização dentro da região (i), 0.80% das regras de atualização estão dentro da região (ii), 97.59% na região (iii), e 1.61% na (iv). A predominância das regras de atualização classe (iii), bem como a relativa raridade das regras de atualização de classe (iv) são então confirmadas, conforme pudemos observar.

### **Autômatos celulares bi-dimensionais**

Para a realização dos experimentos bi-dimensionais usamos  $r_1 = r_2 = 1$  em todos os casos. Independente do tipo de vizinhança, von Neumann ou Moore, nossos experimentos têm  $X_1 = X_2 = 100$ ,  $t_+ = 50$ , e  $T = 25$ . Assim, o número total de células a ser simulado é  $(X_1 + 2r_1t_+)(X_2 + 2r_2t_+) = 40000$ , portanto o número de células observadas é um quarto desse total. Novamente, os valores para as variáveis dos experimentos foram escolhidos estritamente por motivos práticos de manter os tempos de simulação dentro de um limite aceitável.

Retornando ao espaço de regras abordado no Capítulo 2, temos que no caso bi-dimensional com vizinhança de von Neumann existem  $2^{2^{1+2(r_1+r_2)}} = 2^{32}$  regras de atualização diversas. Ao considerarmos a vizinhança de Moore e ainda somente regras de atualização totalísticas, o número de regras de atualização distintas é

$2^{2(1+2r_1)(1+2r_2)} = 2^{18}$ . Outra vez, em ambos os casos, nossos resultados são baseados em 50000 regras de atualização escolhidas aleatoriamente nos conjuntos correspondentes. Devemos observar que restringir as regras de atualização da vizinhança de Moore dentro do conjunto de regras de atualização totalísticas é um meio de assegurar que estas 50000 amostras tenham alguma representatividade estatística. Na ausência desta restrição, o número de possíveis regras de atualização torna-se  $2^{2(1+2r_1)(1+2r_2)}$ . Este número, com valores para  $r_1$  e  $r_2$  como adotamos inicialmente, é  $2^{512}$ , cem ordens de magnitude maior que o número de regras de atualização diversas em quaisquer dos demais experimentos.

Estes casos são apresentados na Figura 4.19 a exemplo da Figura 4.18. As indicações marginais (i)–(iv) fornecem as coordenadas cujos pontos correspondem às regras de atualização da Tabela 4.1 para autômatos celulares bi-dimensionais. A parte (a) da figura refere-se à vizinhança de von Neumann e a parte (b) refere-se à vizinhança de Moore.

Outra vez, as informações adicionais referentes à densidade de pontos nos gráficos são adiadas de modo a seguirmos a mesma metodologia que empregamos nos experimentos uni-dimensionais. Refazendo os mesmos passos, primeiro selecionamos um valor de entropia média para cada gráfico afim de separar as regras de atualização classificadas como (i) ou como (ii) daquelas rotuladas como (iii) ou (iv), e em seguida selecionamos um valor de variância para separar cada par de regras de atualização marcadas. Como os gráficos na Figura 4.19 indicam, pode ser uma tarefa mais difícil que nos casos uni-dimensionais, uma vez que agora é mais comum

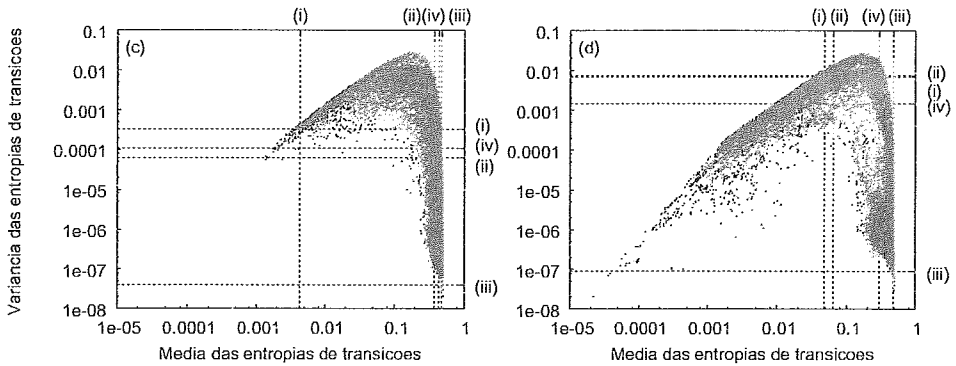


Figura 4.19: Ocorrência de pares média-variância dentro de um conjunto de experimentos para autômatos celulares bi-dimensionais. Os dados são mostrados para a entropia de transições centrada na célula com as vizinhanças de von Neumann (a) e Moore (b) como gráficos de  $\sigma^2(T_f)$  contra  $\overline{T_f}$ . Cada gráfico contém 50000 pontos, cada ponto corresponde a uma regra de atualização escolhida aleatoriamente. As regras de atualização bi-dimensionais da Tabela 4.1, também, são apresentadas como uma indicação nas coordenadas da classe (i)–(iv) a qual elas pertencem.

encontrar dois ou mais rótulos juntos ao longo de um eixo. Iniciamos com a parte (a), que trata da entropia de transições centrada na célula sob a vizinhança de von Neumann. Se selecionarmos 0.4 como o primeiro separador, e em seguida selecionarmos 0.0002 a esquerda e 0.0001 a direita, então 6.67% das regras de atualização estão dentro da região da classe (i), enquanto 16.07% das regras de atualização estão na região de classe (ii), 73.27% na região de classe (iii), e 3.98% na região classe (iv). Na parte (b), selecionamos os separadores 0.1, 0.0072, e 0.0001, e obtemos para as regiões (i)–(iv) as porcentagens 19.02%, 1.82%, 70.74%, e 8.42%, respectivamente, em relação à entropia de transições centrada na célula sob a vizinhança de Moore. Novamente, como no caso uni-dimencional, as indicações são claras de que temos relativa predominância e raridade de regras de classe (iii) e (iv), respectivamente.

## 4.4 Discussão

Os dados mostrados nas Figuras 4.18 e 4.19 para autômatos celulares uni e bi-dimensionais, respectivamente, tendem a exibir o seguinte comportamento em relação às classes (i)–(iv). Quando demonstrados em escala logarítmica dupla, eles aparecem, do modo geral, agrupados como um bumerangue cujo percurso, partindo da ponta onde estão localizados os menores valores das médias e variâncias, nos leva às regras de atualização de classe (i), depois as de classe (ii), as de classe (iv), próxima ao arqueamento médio, e finalmente as de classe (iii) após o arqueamento. As médias de entropias crescem com taxas variadas ao longo da transversal, enquanto as variâncias crescem inicialmente, mas caem novamente após do arqueamento médio.

Os casos uni-dimensionais, descritos na Figura 4.18, indicam claramente que as oito regras de atualização uni-dimensionais da Tabela 4.1 podem ser separadas por pelo menos uma ordem de magnitude da entropia média ou da variância, e até podem ser separadas por ambos. O mesmo não acontece com os casos mostrados na Figura 4.19, que precisam ser examinados em mais detalhes. Na parte (a), que corresponde à entropia de transições centrada na célula sob vizinhança de von Neumann, observamos uma mistura entre as classes (ii) e (iv). Na parte (b), correspondente à entropia de transições e vizinhança de Moore é difícil fazer distinção entre as classes (i) e (ii).

Nós observamos duas principais tendências subjacentes a estas misturas de classe mencionadas para os casos bi-dimensionais. A primeira delas refere-se à parte (a)

da Figura 4.19 onde ocorre uma mistura entre as classes (ii) e (iv) sob vizinhança de von Neumann. O que pode estar acontecendo é que o valor de 50 escolhido para  $t_+$  é insuficiente para o autômato estabelecer um comportamento mais típico de classe (ii) (e conseqüentemente valores para média e variância compatíveis com esta classe). A segunda tendência, referente à parte (b) da Figura 4.19, refere-se às misturas entre as classes (i) e (ii) sob vizinhança de Moore, mas pode também estar relacionada ao caso da parte (a), que discutimos anteriormente. Nossa expectativa de que a classificação de Wolfram continue valendo para os casos bi-dimensionais estão embasadas nas próprias investigações de Wolfram sobre autômatos celulares bi-dimensionais [23]. Porém, isto tem sido contestado pela hipótese de que o esquema de classificação é falho no reconhecimento do comportamento complexo em regras de atualização totalísticas bi-dimensionais, onde são encontradas estruturas denominadas *gliders*, ou seja, estruturas que são vistas “deslizando” através do reticulado bi-dimensional à medida em que o tempo aumenta [9, 10]. Se é este o caso, o problema em questão não é a mistura das classes (i) e (ii), mas a separação entre elas, pois como nenhuma das duas exhibe *gliders*, elas deveriam ser então classificadas igualmente em uma única classe.

Contudo, além desses pequenos conflitos e independente de qual tendência prevaleça, entendemos os resultados dos nossos experimentos, conforme as Figuras 4.18 e 4.19, como o desenvolvimento de uma metodologia para a classificação das regras de atualização dos autômatos celulares, que em muitos sentidos confirma as conclusões iniciais de [33].



Primeiramente, temos que examinar as entropias médias frente ao limite fornecido em (3.12). Neste caso, atingir o limite superior de aproximadamente 0.53 parece não depender de encontrar por acaso alguma regra particular. Na verdade, regras de atualização cuja entropia de transições média chega bem próxima ao limite ocorrem com frequência como pode ser visto nas figuras. Ao usar, então, nossa entropia de transições, um primeiro passo útil é comparar a entropia média da regra de atualização com este limite superior: se for próximo o bastante, digamos  $\geq 0.50$ , é quase certeza a regra de atualização ser de classe (iii).

Além deste teste inicial frente ao limite superior conhecido, o que falta da metodologia mencionada anteriormente é, essencialmente, a construção de relacionamentos entre regras de atualização dadas as suas entropias médias e variâncias, utilizando a abordagem relativizada. A essência aqui é que a classificação é um produto da comparação. Isso mostra a importância fundamental das regras de atualização como as apresentadas na Tabela 4.1, para as quais somos capazes de fornecer um classificação desejada a priori para que possam funcionar como sementes no processo de classificação mais amplo.

# Capítulo 5

## Considerações finais

É importante mencionar que a principal proposta deste trabalho é apresentar uma metodologia de classificação automática, paralela e precisa, na medida do possível, de autômatos celulares. Para atingir este objetivo propusemos uma nova entropia centrada na célula, deveras simples, mas que se mostrou satisfatória e compatível com outras soluções encontradas na literatura.

Partimos do conceito de entropia de entradas e sobre este desenvolvemos nossa heurística de entropia de transições centrada na célula, objetivando a classificação das regras de atualização de autômatos celulares simulados em paralelo. Para tal simulação utilizamos o modelo de passagem de mensagens. Foram realizados vários experimentos com autômatos uni e bi-dimensionais para nossa nova medida dentro do contexto de classificação adotado: as quatro classes de Wolfram. Através dos resultados destes experimentos provamos que a nova heurística definida para a classificação das regras de atualização que regem a evolução dos autômatos celulares possui uma satisfatória capacidade discriminatória no caso de autômatos uni-dimensionais. No

caso de autômatos bi-dimensionais ela é, também, um bom discriminador. Além disso, ajuda a amparar as sugestões de outros autores, i.e. [20], de que um esquema melhor de classificação talvez seja necessário para eliminarmos as misturas de classes apresentadas na Seção 4.4.

Os resultados dos nossos experimentos foram obtidos a partir de uma implementação paralela em um *cluster* (formado por PCs) de um simulador com um módulo embutido, utilizado para computar a entropia de transições centrada na célula.

Cada um dos oito experimentos principais expostos no Capítulo 4 compreendia, no *cluster*, 5 execuções independentes, partindo de estados iniciais aleatórios, para cada uma das 50000 regras de atualização, também, escolhidas aleatoriamente. Tais experimentos chegavam a consumir 5 dias, aproximadamente, para terminar. Este tempo dependia da categoria das regras de atualização que estava sendo usada, ou seja, uni-dimensional com dois raios possíveis, bi-dimensional sob a vizinhança de von Neumann e bi-dimensional sob a vizinhança de Moore. Os experimentos uni-dimensionais com raio  $r = 2$  consumiam o menor tempo, enquanto que os experimentos bi-dimensionais sob vizinhança de Moore, o maior tempo.

O fato de nossas simulações tratarem de autômatos celulares infinitos, conforme explicado no Capítulo 4, é naturalmente uma fonte de considerável desequilíbrio de carga entre os processadores, pois os processadores tendem a realizar números de cálculos diferentes ao longo da evolução do autômato. Dado que nosso objetivo era testar a capacidade discriminatória de nossa heurística, nós não empregamos ne-

nhuma técnica para para reduzir este desbalanceamento de carga. Todavia, reconhecemos que esta questão necessita de uma atenção especial ao simular autômatos celulares em larga escala, caso o impacto de limites infinitos seja considerado.

Devemos considerar dois tipos de desequilíbrio de carga existentes. O primeiro está no fato de que somente aqueles processadores que alojam células observadas fazem computação relacionada ao cálculo de entropia. Isto é, quanto mais células observadas tiver um processador, maior o volume de computação ele deve ter. O segundo tipo de desequilíbrio é consequência das células que não são observadas, mas que são necessárias para fornecer a ilusão de um autômato celular infinito. Na verdade, estas células não precisam ser simuladas mais que  $T_+$  passos de tempo. À medida em que o tempo passa, menor é a quantidade de células que não são observadas e precisam ser simuladas, como explicamos na Seção 4.1. Uma vez considerados estes desequilíbrios de carga, é possível utilizar uma ampla gama de políticas para gerar um balanceamento de carga mais uniforme entre os processadores.

A real necessidade de haver todo processamento extra, necessário para promover a simulação de um autômato celular infinito, é um outro aspecto importante relativo ao desempenho da simulação que merece consideração. Quando um novo esquema de classificação está sendo testado, esta abordagem, decerto, parece ser a mais indicada. Porém, uma vez estabelecido o novo esquema, o uso da abstração de autômato infinito possa, talvez, ser descartado em favor da utilização de bordas cilíndricas. Com esta proposta em mente, nós fizemos alguns experimentos e os resultados colhidos, mostrados na Tabela 4.2. Eles revelaram que nossa heurística centrada na

célula manteve capacidade discriminatória igual à encontrada nos resultados dos experimentos do Capítulo 4. Ainda que, ocasionalmente, os valores das médias e variâncias encontrados não fossem exatamente os mesmos, em consequência da própria diferença na quantidade de células das duas abordagens. Se observarmos as Figuras 4.7 e 4.8, que mostram os modelos espaço-temporais de alguns autômatos celulares com bordas cilíndricas, encontramos as mesmas características que associamos às classes (i) a (iv), independente da periodicidade artificial que aparece em alguns casos devido ao uso de bordas finitas. Todavia, esta periodicidade parece não ter qualquer efeito na entropia de transições centrada na célula, como pode ser observado na Tabela 4.2. Deste fato vem uma consequência direta, qual seja, autômatos celulares consideravelmente grandes podem ser simulados com o mesmo esforço global de processamento, otimizando, assim, o equilíbrio de carga discutido anteriormente. A simulação de autômatos celulares bi-dimensionais com valores maiores para  $t_+$  torna-se, da mesma maneira, mais viável, o que pode levar a um esclarecimento a respeito da mistura entre as classes (ii) e (iv) na classificação de autômatos celulares bi-dimensionais sob a vizinha de von Neumann, mencionada na Seção 4.4.

# Referências Bibliográficas

- [1] B. Awerbuch. Complexity of network synchronization. *Journal of the ACM*, 32:804–823, 1985.
- [2] S. Bandini, G. Mauri, and R. Serra. Cellular automata: from a theoretical parallel computational model to its application to complex systems. *Parallel Computing*, 27:539–553, 2001.
- [3] V. C. Barbosa. *Massively Parallel Models of Computation*. Ellis Horwood, Chichester, UK, 1993.
- [4] V. C. Barbosa. *An Introduction to Distributed Algorithms*. The MIT Press, Cambridge, MA, 1996.
- [5] E. R. Berlekamp, J. H. Conway, and R. K. Guy. *Winning Ways for Your Mathematical Plays*, volume 2. Academic Press, London, UK, 1982.
- [6] G. Braga, G. Cattaneo, P. Flocchini, and C. Quaranta Vogliotti. Pattern growth in elementary cellular automata. *Theoretical Computer Science*, 145:1–26, 1995.
- [7] P. Brinch Hansen. Parallel cellular automata: a model program for computational science. *Concurrency: Practice and Experience*, 5:425–448, 1993.

- [8] J.-C. Dubacq, B. Durand, and E. Formenti. Kolmogorov complexity and cellular automata classification. *Theoretical Computer Science*, 259:271–285, 2001.
- [9] D. Eppstein. Searching for spaceships. In R. J. Nowakowski, editor, *More Games of No Chance*, pages 433–453. Cambridge University Press, Cambridge, UK, 2002.
- [10] D. Eppstein. Which “Life”-like systems have gliders? <http://www.ics.uci.edu/~eppstein/ca>, weblink.
- [11] M. Gardner. Mathematical games: the fantastic combinations of John Conway’s new solitaire game ‘Life’. *Scientific American*, 223:120–123, 1970.
- [12] J. E. Hanson and J. P. Crutchfield. Computational mechanics of cellular automata: an example. *Physica D*, 103:169–189, 1997.
- [13] A. Ilachinski. *Cellular Automata*. World Scientific, Singapore, 2001.
- [14] J. Kari. The nilpotency problem of one-dimensional cellular automata. *SIAM Journal on Computing*, 21:571–586, 1992.
- [15] J. Kari. Rice’s theorem for the limit sets of cellular automata. *Theoretical Computer Science*, 127:229–254, 1994.
- [16] M. Kutrib, R. Vollmar, and Th. Worsch. Introduction to the special issue on cellular automata. *Parallel Computing*, 23:1567–1576, 1997.
- [17] C. G. Langton. Computation at the edge of chaos: phase transitions and emergent computation. *Physica D*, 42:12–37, 1990.

- [18] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, New York, NY, second edition, 1997.
- [19] M. Markus, T. Hahn, and I. Kusch. A novel quantification of cellular automata. *Parallel Computing*, 23:1635–1642, 1997.
- [20] F. M. N. Miranda. *Uma nova entropia de entradas para a classificação automática de autômatos celulares usando simulação paralela*. Tese de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2004.
- [21] M. Mitchell, J. P. Crutchfield, and P. T. Hraber. Dynamics, computation, and the “edge of chaos”: a re-examination. In G. Cowan, D. Pines, and D. Meltzer, editors, *Complexity: Metaphors, Models, and Reality*, pages 497–513. Addison-Wesley, Reading, MA, 1994.
- [22] M. Mitchell, P. T. Hraber, and J. P. Crutchfield. Revisiting the edge of chaos: evolving cellular automata to perform computations. *Complex Systems*, 7:89–130, 1993.
- [23] N. H. Packard and S. Wolfram. Two-dimensional cellular automata. *Journal of Statistical Physics*, 38:901–946, 1985.
- [24] J. E. Savage. *Models of Computation*. Addison-Wesley, Reading, MA, 1998.
- [25] C.E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423, 1948.



- [26] D. Talia. Cellular processing tools for high-performance simulation. *Computer*, 33:44–52, 2000.
- [27] J. von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, Urbana, IL, 1966.
- [28] M. Wójtowicz. Cellular automata rules lexicon. [http://psoup.math.wisc.edu/mcell/ca\\_rules.html](http://psoup.math.wisc.edu/mcell/ca_rules.html), weblink.
- [29] S. Wolfram. Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55:601–644, 1983.
- [30] S. Wolfram. Universality and complexity in cellular automata. *Physica D*, 10:1–35, 1984.
- [31] S. Wolfram. *Cellular Automata and Complexity*. Addison-Wesley, Reading, MA, 1994.
- [32] S. Wolfram. *A New Kind of Science*. Wolfram Media, Champaign, IL, 2002.
- [33] A. Wuensche. Classifying cellular automata automatically: finding gliders, filtering, and relating space-time patterns, attractor basins, and the  $Z$  parameter. *Complexity*, 4:47–66, 1999.
- [34] A. Wuensche and M. Lesser. *The Global Dynamics of Cellular Automata*. Addison-Wesley, Reading, MA, 1992.

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO  
COPPE

CLASSIFICAÇÃO AUTOMÁTICA DE AUTÔMATOS  
CELULARES POR MÁQUINAS PARALELAS USANDO  
ENTROPIA DE TRANSIÇÕES DE ESTADOS

MATHEUS CÔRTEZ MACHADO AGOSTINI