

**UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC**

**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**ROSANA CORREA DA SILVA**

**TEORIA DE AUTÔMATOS CELULARES: ANÁLISE E SUAS APLICAÇÕES  
COMPUTACIONAIS**

**CRICIUMA, NOVEMBRO DE 2009**

**ROSANA CORREA DA SILVA**

**TEORIA DE AUTÔMATOS CELULARES: ANÁLISE E SUAS APLICAÇÕES  
COMPUTACIONAIS**

Trabalho de Conclusão de Curso apresentado  
para obtenção do Grau de Bacharel em  
Ciência da Computação da Universidade do  
Extremo Sul Catarinense.

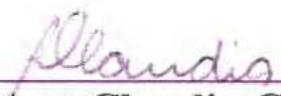
Orientadora: Prof<sup>ª</sup> MSc. Christine Vieira  
Scarpato

**CRICIUMA, NOVEMBRO DE 2009**

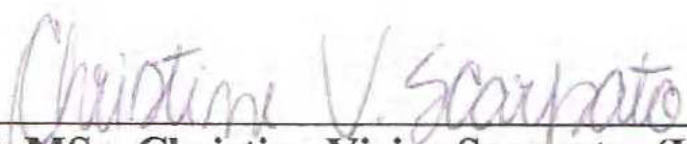
**ROSANA CORRÊA DA SILVA**

**Teoria de Autômatos Celulares: Análise e Suas Aplicações  
Computacionais**


Submetido ao corpo docente do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

  
\_\_\_\_\_  
**Profa. MSc. Ana Claudia Garcia Barbosa**  
Coordenadora do Curso de Ciência da Computação

Banca Examinadora:

  
\_\_\_\_\_  
**Profª. MSc. Christine Vieira Scarpato (UNESC)**  
Orientador

  
\_\_\_\_\_  
**Profº. Esp. Fabrício Giordani (UNESC)**

  
\_\_\_\_\_  
**Profª. MSc. Cristiane Raquel Woszezenki (UNESC)**

Dedico este trabalho a minha família, especialmente  
minha mãe Hilda, a todos os meus amigos e  
especialmente a Deus.

## AGRADECIMENTOS

À minha mãe Hilda, pelo carinho, paciência, forças e milagres para que eu pudesse ter uma boa formação educacional, pessoal e profissional.

Agradeço também ao meu tio João Dienes Cipriano Antunes que sempre me tratou como uma filha, sendo um dos principais responsáveis pela pessoa que sou hoje.

Da mesma forma, ao meu namorado Cleber, que me escutou horas falando sobre meu trabalho e parecia que ele entendia me deixando feliz.

Ao meu sobrinho, que mesmo atrapalhando a minha concentração com bagunças e choros, me faz sempre sorrir e ver que a vida vale a pena.

A meus irmãos que agradeço aqui pelo simples fato de existirem e pelos quais sou tão grata por tudo.

Deixo registrado aqui também meu agradecimento aos meus amigos tão especiais Emanuela Bento Lopes e Fernando Fraga Rodrigues pelo apoio nos momentos de desânimo, sempre me dando forças para continuar em frente, cada um num momento de minha vida.

A professora e orientadora Christine Vieira Scarpato pela paciência, compreensão, idéias e críticas que foram de grande valor para a conclusão deste trabalho.

Agradeço a todo o Departamento de Ciência da Computação da Unesc pelo incentivo e ajuda, principalmente ao professor Fabrício Giordani pela paciência e pelas dicas durante a implementação do trabalho.

Especialmente a Deus que é o único que sabe das dificuldades que tive para chegar até aqui, mas que sou grata pela força, saúde e por tudo que tenho em minha vida.

*“Para ter algo que nunca teve, precisa fazer algo que nunca fez.”*

*[ Chico Xavier ]*

## RESUMO

Autômatos Celulares têm sido utilizados como técnicas de processamento distribuído e granular em problemas clássicos da computação, tais como, criptografia, escalonamento de tarefas e reconhecimento de imagens, como também na simulação de comportamentos naturais, físicos, químicos, biológicos e sociais, como por exemplo, o espalhamento de uma epidemia em uma determinada região geográfica. Este trabalho tem como objetivo demonstrar a teoria de Autômatos Celulares e suas aplicações por meio de pesquisas referentes a definições sobre Máquina de Estados, Autômatos Finitos e Autômatos Celulares, além da identificação das diversas áreas de aplicação dos Autômatos Celulares. Com a teoria analisada e compreendida, é modelado um protótipo que simula o alastramento de um incêndio florestal implementando os seus conceitos básicos. Inicialmente, é criada a imagem de uma floresta fictícia, sendo esta dividida em um plano cartesiano, onde cada coordenada do plano tem uma célula alocada e o conjunto de todas as células forma o autômato que é a floresta. Ao final, é obtida uma interface constando o desenho da floresta e a opção de entrada das coordenadas, cujas coordenadas indicam o foco inicial do incêndio. O fogo se alastra pela floresta por meio de regras de transição imposta a cada uma das células durante as interações, sendo estas regras previamente definidas no código fonte do sistema, permitindo que o autômato evolua uniformemente, possibilitando uma análise do comportamento natural com a aplicação da teoria de Autômatos Celulares.

Palavras-chave: Autômato Celular, Teoria da Computação, Máquina de Estados, Autômato Finito.

## **ABSTRACT**

*Cellular Automata have been used as techniques of distributed processing and granular classical problems of computing, such as encryption, scheduling tasks and image recognition, as well as the simulation of behavior natural, physical, chemical, biological and social, such as the spreading of an epidemic in a particular geographic region. This paper aims to demonstrate the theory of Cellular Automata and its applications through research with respect to definitions of State Machine, Finite Automata and Cellular Automata, beyond the identification of several areas of application of Cellular Automata. With the theory analyzed and understood, is modeled a prototype that simulates the spread of a forest fire implementing their basic concepts. First you create an image of an artificial forest, which is divided into a cartesian plane, where each coordinate of the plan has allocated a cell and the set of all cells shape the automaton that is the forest. At the end is obtained interface design consisting of forest and the option to input the coordinates, the coordinates indicate the initial focus of the fire. The fire spreads through the forest through transition rules imposed on each of the cells in their interactions, and these rules previously defined in the system source code, allowing the robot to evolve smoothly, allowing an analysis of the natural behavior of the application of Cellular Automata theory.*

*Keywords: Cellular automata, Theory Computer Science, State Machine, Finite Automata.*

## LISTA DE FIGURAS

Figura 1. Diagrama AFD: estado inicial (esquerda) e final (direita).....	22
Figura 2. Diagrama AFD: transição .....	23
Figura 3. Diagrama AFD: representações alternativas para transições paralelas.....	23
Figura 4. Exemplo de AC celular unidimensional de raio 1 .....	25
Figura 5. Modelo proposto por Ulam a Von Neumann.....	28
Figura 6. AC de Von Neumann.....	29
Figura 7. Ciclo de um <i>glider</i> .....	30
Figura 8. Autômato representado em forma de vetor.....	31
Figura 9. Formas geométricas da célula .....	32
Figura 10. Representação de vários reticulados unidimensionais .....	32
Figura 11. Representação bidimensional.....	33
Figura 12. Representação tridimensional .....	33
Figura 13. Limite fixo.....	34
Figura 14. Limite periódico.....	34
Figura 15. Limite reflexivo.....	35
Figura 16. Limite adiabático.....	35
Figura 17. Regra de transição e espaço celular de um AC unidimensional de 2 estados.....	37
Figura 18. Evolução temporal do reticulado do AC unidimensional da Figura 17.....	37
Figura 19. Exemplos de interações.....	38
Figura 20. Diagrama de padrões espaço-temporais de um AC .....	40
Figura 21. Classificação dos ACs: (a) classe 1, (b) classe 2, (c) classe 3, (d) classe 4 .....	41
Figura 22. Vizinhança de Von Neumann (a) raio=1 e (b) raio=2 .....	43
Figura 23. Vizinhança de Moore (a) raio=1 e (b) raio=2 .....	43
Figura 24. Vizinhança aleatória.....	44
Figura 25. Exemplo de um <i>glider</i> de período 4 no <i>Life</i> .....	44

Figura 26. Estrutura periódica no <i>Life</i> semelhante a um relógio.....	45
Figura 27. Representação dos intervalos musicais .....	49
Figura 28. Programa Gauss 18 .....	50
Figura 29. Vizinhança local com indivíduos próximos e vizinhança aleatória com indivíduo a uma distância $L$ .....	51
Figura 30. Representação da floresta fictícia.....	56
Figura 31. Representação parcial das células na floresta .....	57
Figura 32. Regras de transição implementadas em código .....	59
Figura 33. Exemplificando limite adiabático no reticulado da floresta.....	60
Figura 34. Condições de evolução e encadeamento do AC .....	61
Figura 35. Incêndio iniciado a partir das coordenadas .....	63
Figura 36. Atualizando a matriz de estados do AC .....	63
Figura 37. Foco de incêndio com início na coordenada (43,38) .....	65
Figura 38. Evolução do AC demonstrando as sete primeiras iterações a partir da coordenada (43,38).....	65

## LISTA DE TABELAS

Tabela 1. Função programa (AFD) na forma de tabela de transições de um AF .....	24
---	----

## LISTA DE SIGLAS

AC	Autômato Celular
AF	Autômato Finito
AFD	Autômato Finito Determinístico
AFND	Autômato Finito Não-Determinístico
AFS	Autômato Finito com Saída
AIDS	Síndrome da Imunodeficiência Adquirida
CEFET-PR	Centro Federal de Educação Tecnológica do Paraná
EP-USP	Escola Politécnica da Universidade de São Paulo
JAVA	<i>Java Language Source Code File</i>
LR	Linguagens Regulares
MEF	Máquina de Estado Finito
SEF	Sistema de Estado Finito
UFF	Universidade Federal Fluminense
UML	<i>Unified Modeling Language</i>
UNESC	Universidade do Extremo Sul Catarinense
UNISINOS	Universidade do Vale do Rio dos Sinos
USP	Universidade de São Paulo

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>15</b>
1.1 OBJETIVO GERAL.....	16
1.2 OBJETIVOS ESPECIFICOS .....	16
1.3 JUSTIFICATIVA .....	16
1.4 ESTRUTURA DO TRABALHO.....	18
<b>2 MÁQUINAS DE ESTADOS.....</b>	<b>19</b>
2.1 RECONHECEDORES DE LINGUAGENS OU AUTÔMATOS FINITOS.....	20
<b>2.1.1 Diagrama de Transições.....</b>	<b>22</b>
<b>2.1.2 Tabelas de Transições .....</b>	<b>23</b>
<b>3 AUTÔMATOS CELULARES .....</b>	<b>25</b>
3.1 DEFINIÇÃO FORMAL DE AUTÔMATOS CELULARES .....	26
3.2 ORIGEM E EVOLUÇÃO DOS AUTÔMATOS CELULARES.....	27
3.3 ORGANIZAÇÃO DO AUTÔMATO CELULAR.....	31
3.4 DIMENSÕES DOS AUTÔMATOS CELULARES .....	38
<b>3.4.1 Autômatos Celulares Unidimensionais.....</b>	<b>39</b>
<b>3.4.1.1 Classificação dos Autômatos Celulares Unidimensionais.....</b>	<b>40</b>
<b>3.4.2 Autômatos Celulares Bidimensionais .....</b>	<b>42</b>
<b>3.4.3 Autômatos Celulares Tridimensionais.....</b>	<b>45</b>
3.5 ÁREAS DE APLICAÇÕES DOS AUTÔMATOS CELULARES.....	45
<b>4 TRABALHOS CORRELATOS .....</b>	<b>47</b>
<b>5. IMPLEMENTAÇÃO DO PROTÓTIPO PROPOSTO .....</b>	<b>52</b>
5.1 METODOLOGIA.....	52
5.2 DESENVOLVIMENTO E RESULTADOS OBTIDOS.....	54
<b>CONCLUSÃO.....</b>	<b>68</b>
<b>REFERÊNCIAS.....</b>	<b>70</b>

<b>BIBLIOGRAFIA RECOMENDADA.....</b>	<b>74</b>
<b>APÊNDICE A – DIAGRAMA DE ESTADOS DO AUTOMATO CELULAR.....</b>	<b>75</b>
<b>APÊNDICE B – REGRAS DE TRANSIÇÃO .....</b>	<b>76</b>
<b>APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação) .....</b>	<b>77</b>
<b>APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação) .....</b>	<b>78</b>
<b>APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação) .....</b>	<b>79</b>
<b>APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação) .....</b>	<b>80</b>
<b>APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação) .....</b>	<b>81</b>
<b>APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação) .....</b>	<b>82</b>
<b>APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação) .....</b>	<b>83</b>
<b>APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação) .....</b>	<b>84</b>
<b>APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação) .....</b>	<b>85</b>
<b>APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação) .....</b>	<b>86</b>
<b>APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação) .....</b>	<b>87</b>
<b>APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação) .....</b>	<b>88</b>
<b>APÊNDICE C – LIMITE ADIABÁTICO.....</b>	<b>89</b>

## 1 INTRODUÇÃO

Autômatos Celulares têm sido utilizados como técnicas de processamento distribuído e granular em problemas clássicos da computação, tais como, criptografia, escalonamento de tarefas e reconhecimento de imagens, entre outros.

A teoria de Autômatos Celulares vem de encontro à busca persistente dos pesquisadores do mundo tecnológico: solução de problemas complexos através de sistemas de simples implementação. Como fontes bibliográficas, existem pesquisas sobre Autômatos Celulares em diversas áreas sendo estas publicadas em forma de artigos e monografias tanto no Brasil como no exterior.

O Autômato Celular é utilizado em pesquisas da área da saúde, como por exemplo, no estudo da evolução da AIDS (Síndrome da Imunodeficiência Adquirida), no organismo das pessoas chamadas de soro-positivos (possuem o vírus). O software que implementou a evolução da AIDS utilizou-se da teoria de Autômato Celular, sendo capaz de mostrar a evolução da doença sem tratamento, detalhando todas as fases da doença (desde a infecção até a perda, por parte do corpo, do seu sistema imunológico), e ainda demonstra a reação do vírus e das células infectadas.

Os Autômatos Celulares também são utilizados para avaliar fenômenos naturais de dimensão espaço-temporal como: escoamento de água da chuva, dispersão de sementes, crescimento populacional, fenômenos urbanos, entre outros.

Atualmente, na UNESC não é possível encontrar pesquisas direcionadas a Autômatos Celulares, somente pequenos trabalhos extracurriculares solicitados pelo docente da área. Devido à escassez de pesquisas, bibliografias e produções de softwares baseados na teoria de Autômatos Celulares disponíveis nesta universidade, propõe-se com esta pesquisa, uma visão geral da área, sendo que não é foco do trabalho produzir um novo modelo ou novo algoritmo, mas sim um estudo das aplicações de Autômatos Celulares. A relevância está na

contribuição à comunidade acadêmica, onde esta poderá, em trabalhos futuros, encontrar uma fonte de informação concentrada, que aborda a área de Autômatos Celulares e alguns métodos de implementação dos mesmos. A fim de validar esse estudo será feita a implementação de um protótipo utilizando-se Autômatos Celulares.

### 1.1 OBJETIVO GERAL

Demonstrar a teoria de Autômatos Celulares e suas aplicações.

### 1.2 OBJETIVOS ESPECIFICOS

São objetivos específicos deste trabalho:

- a) entender a teoria de Autômatos Celulares;
- b) identificar as diversas áreas de atuação dos Autômatos Celulares;
- c) verificar as aplicações já existentes que utilizam os Autômatos Celulares;
- d) demonstrar o funcionamento do Autômato Celular;
- e) aplicar os princípios de Autômato Celular no desenvolvimento de um protótipo.

### 1.3 JUSTIFICATIVA

Sendo um tema pouco pesquisado por acadêmicos do curso de Ciência da Computação da UNESC e por ser uma área com diversas aplicações computacionais que não se restringe somente à área de ciências exatas, podendo ser encontradas aplicações nas áreas da saúde, biologia, geologia entre outros, é relevante que, a partir desse estudo seja,

descoberta a utilização e as teorias sobre Autômatos Celulares. O projeto tem o foco na contribuição para a comunidade acadêmica, onde em pesquisas futuras, poderá ser encontrada uma fonte de informação concentrada que aborda a área de Autômatos Celulares e alguns métodos de implementação, expandindo assim as possíveis aplicabilidades dos mesmos a outras áreas ainda não exploradas.

Apesar de sua simplicidade de implementação, foi observado que os Autômatos Celulares exibem uma diversidade de comportamentos dinâmicos, atraindo o interesse dos pesquisadores, cujas pesquisas podem servir de paradigma para novas tecnologias de hardware, especialmente para a nanotecnologia (ILACHINSKI , 2001).

Numa abordagem mais aplicada à área computacional, os Autômatos Celulares, por tratar-se de estruturas computacionais de implementação extremamente simples, mas capazes de exibir uma dinâmica complexa alcançando resultados satisfatórios em sistemas complexos, se tornaram importantes ferramentas para o estudo de sistemas naturais. Diversos pesquisadores utilizam-se de Autômatos Celulares para realizar a modelagem de fenômenos reais, permitindo a interdisciplinaridade entre áreas distintas, conseguindo assim importantes resultados dos modelos obtidos, tais como: simulação de sistemas biológicos (processos de reprodução), fenômenos físicos (difusão), sistemas sociais (formação de comunidades), entre outros (OLIVEIRA, 2003; PESSOA; GAGLIARDI; ALVES, 2004).

O conhecimento adquirido durante essa jornada contribuirá na aplicação dos resultados obtidos durante a pesquisa, relatando a teoria numa prática mais sucinta em forma de um sistema modelo.

## 1.4 ESTRUTURA DO TRABALHO

Esta pesquisa é composta por 5 capítulos, sendo seu objetivo detalhar a teoria de Autômatos Celulares e como demonstração a implementação de um protótipo.

O Capítulo 2 aborda máquinas de estados, onde é feita uma breve conceitualização sobre Autômatos Finitos e suas representações gráficas. O capítulo seguinte traz uma visão geral sobre o objeto de estudo que são os Autômatos Celulares: sua origem, definição formal, sua organização, suas dimensões e algumas aplicações em diversas áreas de estudos.

Os trabalhos correlatos são descritos no Capítulo 4.

O Capítulo 5 demonstra a aplicação da teoria de ACs aplicada em forma de uma simulação de alastramento de incêndio florestal, sendo neste capítulo vistas a interface, as regras utilizadas e a implementação do protótipo.

Finalmente, na conclusão são apresentadas as considerações finais, bem como as dificuldades encontradas e sugestões para trabalhos futuros.

## 2 MÁQUINAS DE ESTADOS

Sistemas de Estados Finitos (SEF) ou Máquinas de Estados Finitos (MEF) são máquinas abstratas que possuem entradas e saídas discretas<sup>1</sup>. Elas são organizadas para assumirem um número finito e pré-definido de estados e são limitadas por uma memória pré-fixada. A cada estado adotado pela MEF, ela somente armazena as informações anteriores essenciais para definir as ações para a próxima etapa do processo (MENEZES, 2005; VIEIRA, 2006).

Uma vez que o elemento de entrada e o estado antecedente são conhecidos, o próximo estado é previsível. A informação de saída é armazenada em cada estado, portanto, as saídas são determinadas pelos estados anteriores (GERSTING, 2001).

A justificativa para estudo deste modelo matemático é a associação com a essência de sistemas concretos tais como: elevadores, analisadores léxicos, processadores de texto (MENEZES, 2005), máquinas de vender jornais e refrigerantes, relógios digitais entre outros (VIEIRA, 2006).

Contudo, nem todas as máquinas que podem ser definidas como finitas podem também ser representadas como tais. Um exemplo seria o cérebro humano, que apesar da possibilidade dos neurônios serem aparentemente um número finito de bits, podendo ser representados por um número finito de estados, o alto número de combinações geraria um número muito maior de estados, sendo isto, considerado na prática, uma representação um tanto deficiente (MENEZES, 2005).

As MEFs podem ser divididas em duas categorias (VIEIRA, 2006):

---

<sup>1</sup> As variáveis de estado permanecem constantes ao longo do tempo, contudo seus valores são alterados em momentos bem definidos conhecidos como passo ou tempo de ocorrência de evento (PASCOAL; BECCENERI; MARIETTO, 2005).

a) **transdutores ou Autômatos Finitos com Saída (AFS)**: geram uma saída baseada em uma entrada e/ou um estado utilizando ações. Eles são utilizados para aplicações de controles, podendo ser representadas pelas máquinas de Moore e de Mealy;

b) **reconhecedores de linguagens ou Autômatos Finitos (AF)**: para cada entrada, existem duas saídas possíveis, onde a saída sinaliza se a entrada foi aceita ou rejeitada.

Na próxima seção será feita uma explanação sobre os reconhecedores de linguagem, cujo tema está mais voltado ao objeto de estudo.

## 2.1 RECONHECEDORES DE LINGUAGENS OU AUTÔMATOS FINITOS

Autômato Finito é um SEF, portanto possui um número finito de estados, sendo um modelo computacional do tipo seqüencial<sup>2</sup> que pode ser distinguido em (HOPCROFT; ULLMAN; MOTWANI, 2002; MENEZES, 2005):

a) **determinístico**: dependendo do estado momentâneo e do símbolo de entrada, o processo pode assumir um único estado;

b) **não-determinístico**: dependendo do estado momentâneo e do símbolo de entrada, o processo pode assumir um conjunto de estados possíveis.

O Autômato Finito Determinístico (AFD) pode ser definido formalmente numa quintupla ordenada (MENEZES, 2005):

$$M = (\Sigma, Q, \delta, q_0, F)$$

Na qual:

$\Sigma$  é um alfabeto de símbolos de entrada, ou simplesmente alfabeto de entrada;

$Q$  é um conjunto de estados possíveis do autômato o qual é finito;

---

<sup>2</sup> A execução da próxima componente depende da terminação da componente anterior (MENEZES, 2005).

$\delta$  é uma função programa ou simplesmente programa, ou ainda função de transição:

$$\delta: Q \times \Sigma \rightarrow Q$$

a qual é uma função parcial. Supondo que a função programa é definida para um estado  $P$  e um símbolo  $a$ , resultando no estado  $X$ , então:

$$\delta(P, a) = X$$

é uma transição do autômato;

$q_0$  é um elemento distinguido de  $Q$ , denominado estado inicial;

$F$  é um subconjunto de  $Q$ , denominado conjunto de estados finais.

Um AF é essencialmente formado de três partes (LEWIS; PAPADIMITRIOU, 2000; MENEZES, 2005):

a) **fita de entrada:** mecanismo de entrada que possui a informação a ser processada;

b) **unidade de controle ou cabeçote de leitura:** reflete o estado momentâneo do sistema, onde uma unidade de leitura acessa uma célula da fita de cada vez e, em intervalos regulares, movimenta-se unicamente para a direita;

c) **função programa ou função de transição:** o que comanda as leituras e define o novo estado do sistema que dependem do estado atual e do dado de entrada.

Ainda sobre a definição de AF (LEWIS; PAPADIMITRIOU, 2000; MENEZES, 2005), a fita é finita e dividida em células, onde cada célula terá apenas um símbolo da palavra a ser processada, cuja esta ocupa toda a fita. Cada símbolo pertence a um conjunto pré-definido chamado *alfabeto de entrada*. O processo de leitura de uma palavra  $w$  faz-se no emprego sucessivo da função de transição para cada símbolo de  $w$ , até que uma condição de parada seja satisfeita durante o processo. Uma restrição a ser levantada é a impossibilidade de gravar qualquer coisa sobre a fita, e esta não existe memória auxiliar limitado assim o processo de leitura (MENEZES, 2005).

O processo pode ser resumido da seguinte forma: quando um dado de entrada é lido, a cabeça da fita move-se em direção à célula à sua direita. Com isso, altera-se o estado do sistema, realizando este movimento seqüencialmente até o final da fita de entrada. O autômato finaliza sua análise aceitando ou rejeitando o que foi lido (LEWIS; PAPADIMITRIOU, 2000).

Um AF pode ser especificado graficamente de duas formas (HOPCROFT; ULLMAN; MOTWANI, 2002):

- a) diagrama de transições;
- b) tabela de transições.

### 2.1.1 Diagrama de Transições

Um diagrama de transição para um AF é um grafo direcionado e rotulado, onde cada estado é representado por um vértice também chamado de nó. O estado inicial possui uma seta que se origina de nenhum nó, os estados finais são representados por círculos diferenciados e os demais estados por um círculo com contorno simples conforme Figura 1. Os arcos ou arestas representam as transições, sendo que, entre dois estados  $P$  e  $Q$ , existirá uma aresta direcionada de  $P$  para  $Q$ , com rótulo  $a$ , conforme visualizado na Figura 2 (MENEZES, 2005).

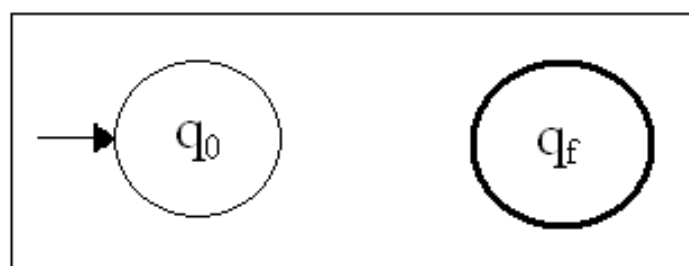


Figura 1. Diagrama AFD: estado inicial (esquerda) e final (direita)  
Fonte: MENEZES, P. (2005)

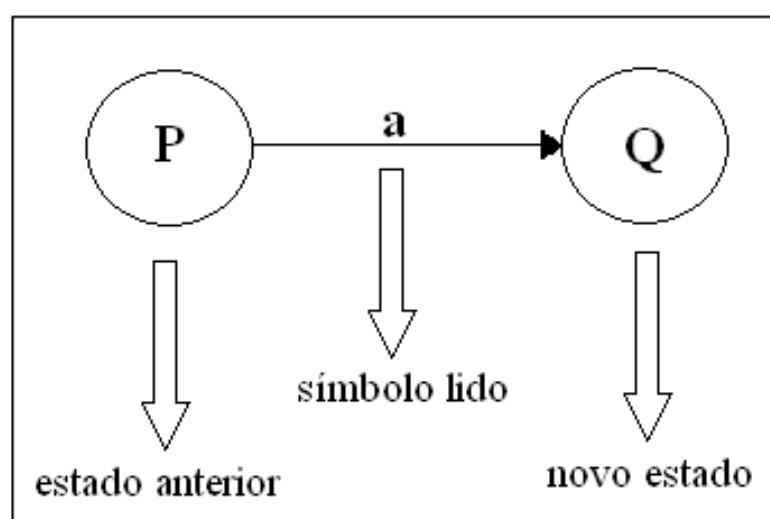


Figura 2. Diagrama AFD: transição  
Fonte: MENEZES, P. (2005)

Ao analisar a Figura 2 é possível identificar os estados  $P$  e  $Q$ . A aresta representa a transição entre  $P$  para  $Q$  rotulada de  $a$  (símbolo de entrada), isso significa que se o autômato está no estado  $P$  e lê o símbolo  $a$  o próximo estado será o  $Q$ .

Complementando: existem ainda as transições paralelas (mesmos nodos de origem e destino) podem alternativamente ser representados como na Figura 3 (supondo que  $\delta(P,a) = Q$  e  $\delta(P,b) = Q$ ).

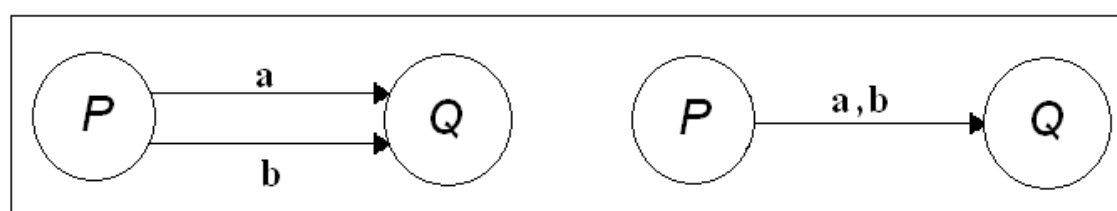


Figura 3. Diagrama AFD: representações alternativas para transições paralelas  
Fonte: MENEZES, P. (2005)

### 2.1.2 Tabelas de Transições

Uma forma alternativa e muito comum de representar uma função programa é como uma tabela de dupla entrada, sendo uma função que recebe dois argumentos (o símbolo lido e o estado) e retorna um valor (o próximo estado). No exemplo da Tabela 1 pode-se verificar os estados  $q_0$ ,  $q_1$ ,  $q_2$  e  $q_f$ , sendo  $q_0$  um estado inicial e  $q_f$  um estado final. A seta

indica o estado inicial do autômato, onde no estado  $q_0$  com o símbolo  $a$  teremos como retorno o estado  $q_1$  como próximo estado (MENEZES, 2005).

Tabela 1. Função programa (AFD) na forma de tabela de transições de um AF

	$\delta_1$	a	b
$\rightarrow$	$q_0$	$q_1$	$q_2$
	$q_1$	$q_f$	$q_2$
	$q_2$	$q_1$	$q_f$
	* $q_f$	$q_f$	$q_f$

Fonte: MENEZES, P. (2005)

Após uma definição dos conceitos básicos do AF e suas representações, dá-se seqüência a explanação do objeto de estudo deste projeto, Autômatos Celulares.

### 3 AUTÔMATOS CELULARES

Autômato Celular (AC) consiste de um espaço euclidiano<sup>3</sup> (MISSIO, BARROS, 2006) sendo um modelo dinâmico de dimensão espacial e temporal discretas (THIELO, 1999), consistindo num elevado número de células idênticas programadas. São implementados facilmente, permitindo a organização direta de seus padrões para a avaliação de seu comportamento, tornando-se assim, relevantes objetos de estudo e modelagem de sistemas complexos construídos e/ou naturais em diversas áreas (BAR\_YAM, 1997; GUTOWITZ, 1995; OLIVEIRA, 2003; WOLFRAM, 1983). Sua principal característica é a simplicidade da modelagem matemática aplicada em seu desenvolvimento (MISSIO, BARROS, 2006).

O AC é formado por uma malha de células (Figura 4) que evoluem a passos de tempo discretos. Cada célula possui um conjunto finito de estados. Seus estados futuros podem ser regrados pelo estado da célula central e pelo estado de suas células vizinhas. Este estado também faz parte do conjunto finito. A ligação da célula com sua vizinhança é pontual e semelhante nos diversos espaços de tempo (CASTRO; CASTRO, 2008; PESSOA; GAGLIARDI; ALVES, 2004).

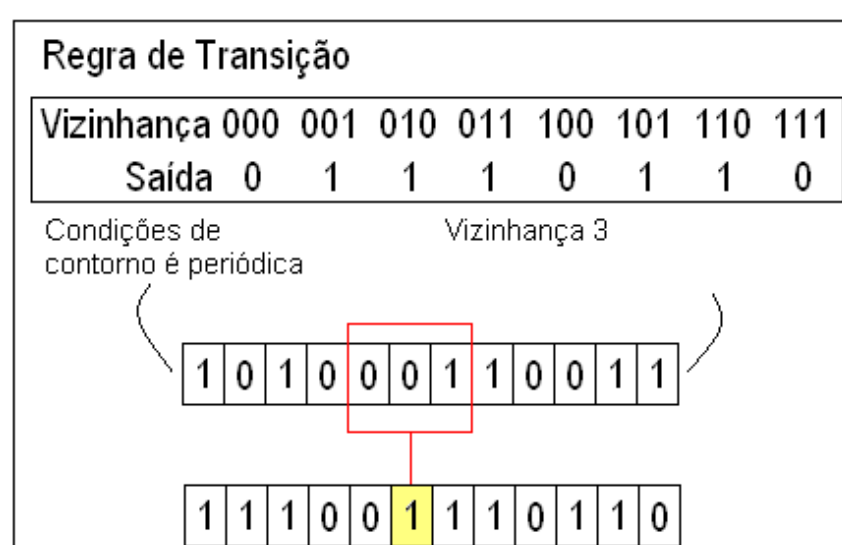


Figura 4. Exemplo de AC celular unidimensional de raio 1<sup>4</sup>  
 Fonte: MARIANO,A.; et al.(2007)

<sup>3</sup> Prática de ligação baseada no espaço do vizinho mais próximo (WERMUTH, 2007).

<sup>4</sup> Um AC dimensionado pelos seus dois vizinhos adjacentes, sendo a regra aplicada a célula central, onde a transição para um novo estado dependerá dos estados destes dois vizinhos (ALVES; GAGLIARDI, 2006).

A interação nas células da rede do AC acontece paralelamente em todas as células de forma encadeada e organizada (MARIANO et al, 2007).

Segundo Alves e Gagliardi (2006) as células alteram seu estado dependendo da condição da célula central e de sua vizinhança, sendo estes estados alterados por regras impostas previamente ao sistema. Estas regras podem ser:

a) **determinísticas**: quando as regras dependem da configuração do sistema dependente de uma vizinhança pré-estabelecida, neste caso o sistema deverá contemplar todas as vizinhanças possíveis;

b) **totalística**: a célula mudará de estado dependendo somente se um número  $x$  de vizinhos estiver em determinado estado, não importando os outros estados de outras células da vizinhança;

c) **probabilística**: não são estabelecidas regras para cada vizinhança possível, mas são previstos os resultados utilizando a probabilidade.

### 3.1 DEFINIÇÃO FORMAL DE AUTÔMATOS CELULARES

Um AC é definido pela quádrupla:  $AC=(L, S, N, f)$ , onde cada elemento é (BARONE et al, 2003):

$L$  uma rede regular formada pelas células;

$S$  um conjunto finito de estados;

$N$  um conjunto finito ( $|N| = n$ ) de índices vizinhos;

$f: S^n \rightarrow S$  uma função de transição (regras).

Quando as células preenchem completamente o espaço da rede, sendo esta periódica e de dimensão  $d$ , e durante as interações é obtido o mesmo reticulado, pode-se dizer que este AC é regular. Em alguns casos, a regra de transição não leva em consideração o estado corrente da célula, sendo consideradas somente suas vizinhas, assim, neste caso, o

estado atual da célula não fará parte da regra de transição (PASCOAL; BECCENERI; MARIETTO, 2005).

### 3.2 ORIGEM E EVOLUÇÃO DOS AUTÔMATOS CELULARES

Entre as décadas de 40 e 50 a humanidade presenciou um aperfeiçoamento súbito na área tecnológica, com destaque à área computacional, onde a possibilidade da representação de fenômenos matemáticos complexos ficou mais próxima de sua concretização com a utilização de ACs (MENDES, 2005).

John Von Neumann pode ser considerado um grande contribuidor da época, sendo o idealizador da definição de AC, a qual foi um pouco antes de sua morte em 1957. A intenção de Von Neumann era demonstrar que uma máquina seria capaz de se auto-reproduzir, para isso fez uma junção entre conceitos da biologia e da teoria dos autômatos (AGUIAR; COSTA, 2001; MENDES, 2005).

A utilização de ACs tem grande valia no apoio de modelagens que possuem fenômenos coletivos como: “quebra de simetria, caos, ordenamento de fases e turbulência”, aplicadas constantemente, por exemplo, nas áreas de biologia (reprodução) e física (difusão), possibilitando simulações que dependem de tempo e espaço (ALVES; GAGLIARDI, 2006).

O AC criado por Von Neumann, embasado em pesquisas de Alan Turing<sup>5</sup>, surgiu como uma forma de modelagem de fenômenos de certa complexidade que possuem um elevado número de variáveis que são difíceis de serem representadas por métodos tradicionais (JESUS; KAWANO, 2002).

---

<sup>5</sup> Matemático de origem inglesa, nascido em Londres em 1912 e morreu envenenado em 1954. Criador da Teoria da computação (SUPERINTERESSANTE, 2000).

Enquanto Von Neumann iniciava seus estudos, Stanislaw Ulam<sup>6</sup> pesquisava comportamentos de células dispostas bidimensionalmente, onde cada uma delas tinha seu estado alterado conforme o seu estado e o estado de suas células vizinhas dependendo da atuação das regras de transição (OLIVEIRA, 2003).

Ulam com seus estudos sugeriu a Von Neumann o conceito de uma rede artificial (observada na Figura 5), que aparentemente seria um “tabuleiro de xadrez”, onde cada campo constituiria uma célula com comportamento independente, contudo obedecendo a regras impostas a elas. Estas regras definiriam o próximo estado de cada célula durante sua evolução ocorrida no período de tempo pré-determinado (ARAUJO, 2007).

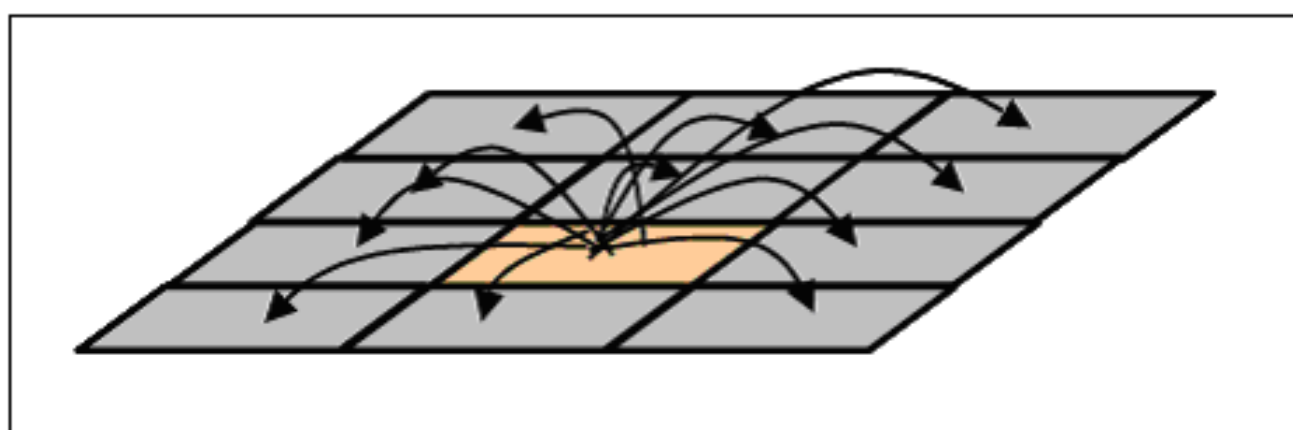


Figura 5. Modelo proposto por Ulam a Von Neumann  
Fonte: ARAUJO, D. (2007)

Conseqüente à sugestão de Ulam, Von Neumann concluiu sua intenção inicial, originou um AC, que solucionou o enigma da auto-reprodução, partindo de uma máquina universal de Turing encapsulada em uma rede de células, com 29 estados por célula e cinco células como vizinhança (OLIVEIRA, 2003), esta conhecida como vizinhança de Von Neumann. O autômato batizado por Von Neumann de *Mc* começou com uma rede considerada infinita, com cada célula num estado inativo. Os detalhes desta criação eram representados pelos 29 diferentes estados possíveis das células individuais. As diferentes combinações destes estados geriam o comportamento do organismo e definiam o mesmo (PASCOAL; BECCENERI; MARIETTO, 2005; AGUIAR et al, 2004).

---

<sup>6</sup> Matemático polaco-americano, nascido em 1909 e morreu em 1984. Um dos principais desenvolvedores da bomba de hidrogênio em *Los Alamos* (MACROCOSMO.COM, 2004).

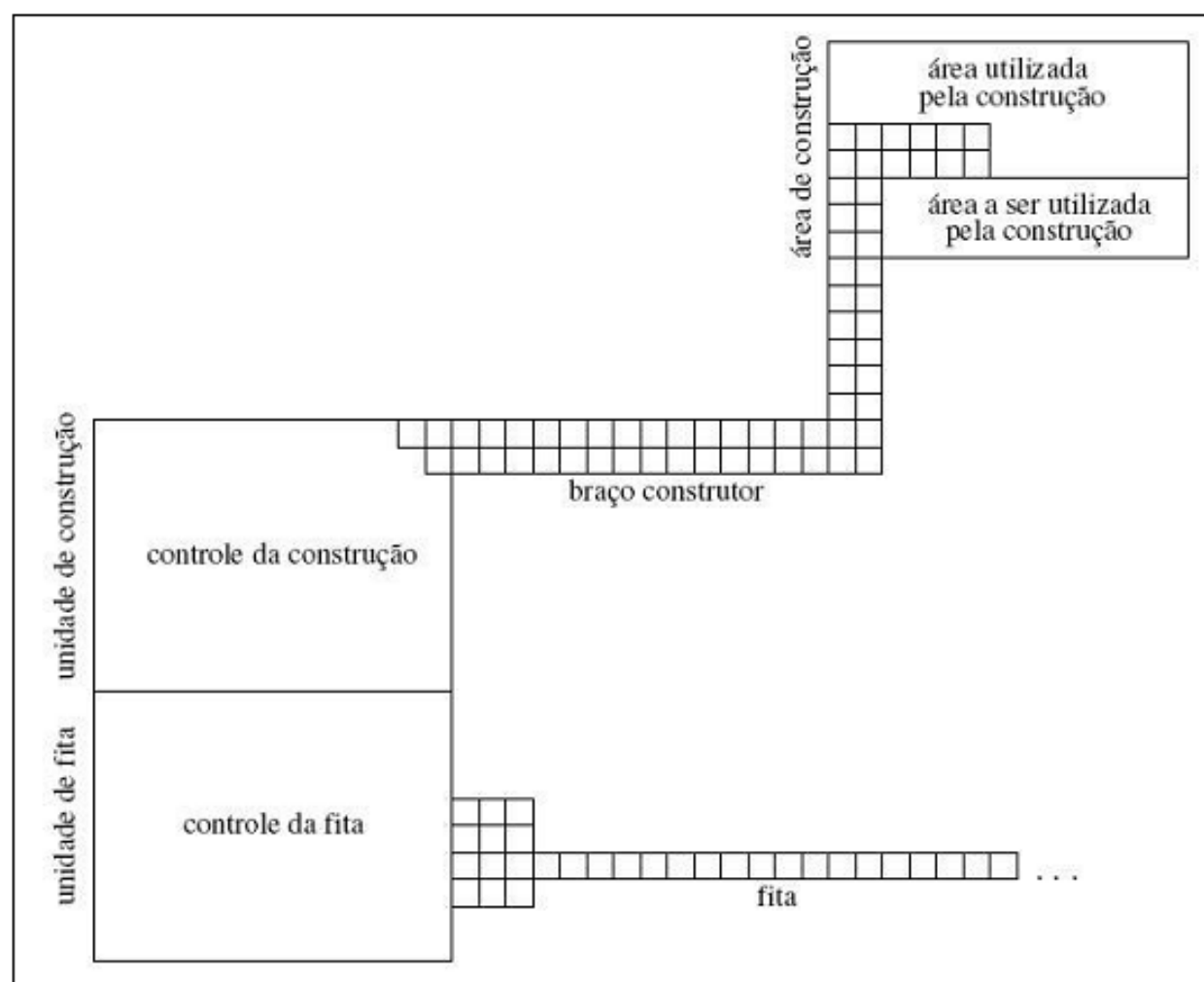


Figura 6. AC de Von Neumann

Fonte: PASCOAL, F.; BECCENERI, J.; MARIETTO, M., (2005)

A Figura 6 representa o autômetro  $M_c$  criado por Von Neumann sendo possível visualizar que este é composto pela primeira parte que forma o novo autômetro e uma segunda parte que é a fita, responsável por compreender a informação essencial para a formação do novo autômetro  $M$ . A construção do autômetro  $M$  é iniciada por meio da propagação de estados pelas células, formando sinais entre a unidade da fita e a unidade de construção formada de um controle de construção e um braço construtor. O  $M_c$  é um autômetro ( $M$ ) construtor auto-reprodutor, sendo chamado também de construtor universal, pois pode construir qualquer autômetro, bastando sua descrição estar na sua fita (PASCOAL; BECCENERI; MARIETTO, 2005; AGUIAR et al, 2004).

Segundo Nöth (2009) a máquina de Von Neumann nunca se tornou uma máquina real, mas projetos a partir dela foram realizados aprimorando os ACs. Outros projetos foram realizados utilizando conceitos da área de sistemas artificiais, estes com habilidades de auto-organização e auto-reprodução. Oriundos destes projetos são os conhecidos vírus de computador.

Após a morte de Von Neumann outros pesquisadores encontraram ACs menos complexos capazes de se auto-reproduzirem. Seguindo os conceitos de Von Neumann, no começo dos anos 70, John Conway modelou o *Game of Life*, que é um AC binário bidimensional, cuja regra de transição expressava que células no estado 0 são ditas como “mortas” e as células no estado 1 como “vivas”. Conway foi um dos principais responsáveis em difundir a teoria dos ACs nos meios acadêmicos com a modelagem do *Game of Life*, pois o AC de fácil compreensão demonstrou a capacidade de reproduzir fenômenos complexos e organizações semelhantes a organismos artificiais (OLIVEIRA, 2003).

O *Game of Life* é um autômato que utiliza o tipo de vizinhança definida por Moore<sup>7</sup> e com regras de transição relativamente simples, cujo comportamento não segue para um estado caótico<sup>8</sup>, entretanto podem desenvolver estruturas infinitas no espaço, confirmando sua computação universal e tais estruturas podem ser chamadas de *Gliders*<sup>9</sup> (MENDES, 2005). O ciclo de um glider pode ser visto na Figura 7 posteriormente.

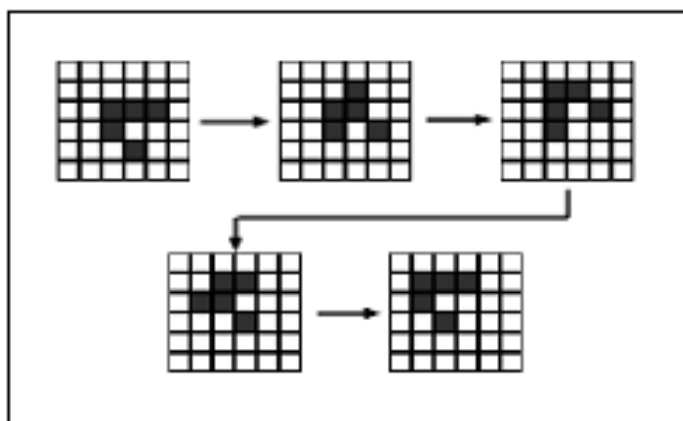


Figura 7. Ciclo de um *glider*  
Fonte: MENDES, R. (2005)

Uma regra simples de AC é exemplificada por Aguiar et al (2004) da seguinte forma:

$$a_i^{(t+1)} = a_{i-1}^{(t)} + a_{i+1}^{(t)} \text{ mod } 2 \quad (1)$$

<sup>7</sup> Vizinhança composta pelas células adjacentes e diagonais a célula central (PASCOAL; BECCENERI; MARIETTO, 2005).

<sup>8</sup> Estado caótico é quando as células do reticulado apresentam um comportamento aperiódico, não padronizado, instável ao longo da evolução do AC (MENDES, 2005).

<sup>9</sup> Objeto que se movimenta pela rede celular e que periodicamente repete seu formato em locais distintos, girando em torno de seu eixo principal (CASTRO; CASTRO, 2008).

onde  $i$  é a posição na célula no tempo  $t$ , denotado como  $a_i^{(t)}$ . O resultado final desta regra sempre será 0 ou 1, onde é considerado o resto da divisão por 2. Aguiar et al (2004) ainda faz uma representação de um AC proposto por Stephen Wolfram utilizando um vetor (Figura 8) de células cujos valores variam entre 0 e 1, possibilitando a aplicação da regra da equação 1.

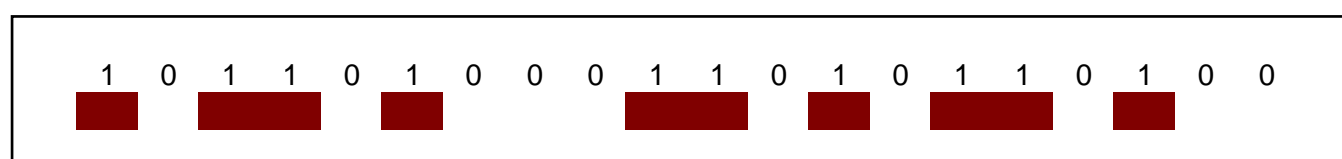


Figura 8. Autômato representado em forma de vetor  
Fonte: AGUIAR, M. S. (2004).

A equação 1 exemplifica como uma regra de transição pode ser definida para determinar os estados das células ao longo de sua evolução. Já a Figura 8 demonstra como estes estados podem ser vistos ilustrativamente num instante  $t$  e como podem ser representados nos instantes seguintes, aplicando a regra da equação 1.

### 3.3 ORGANIZAÇÃO DO AUTÔMATO CELULAR

O AC é um conjunto de *automata* (células) que interagem entre si em função de uma coleção finita de condições pré-definidas (LI; PACKARD, 1990). A complexidade não está na representatividade de equações complexas, mas no desenvolvimento do autômato que, durante sua evolução, apresenta comportamento complexo apesar de cada uma de suas células apresentarem estados regidos por condições simples pré-definidos. (PASCOAL; BECCENERI; MARIETO, 2005). Estas condições pré-definidas (regras) serão os passos que o AC irá tomar para a resolução de um dado problema e qualquer alteração nestas, pode alterar conseqüentemente a evolução do AC, prejudicando a solução do problema. As condições de transição são baseadas em várias características tanto da célula (menor unidade do AC) quanto do próprio AC (ALVES, GAGLIARDI, 2006).

Uma *automata* pode adotar diversas formas geométricas, dentre elas pode-se citar: quadrangular, triangular e hexagonal (Figura 9). Geralmente a forma quadrangular, sendo a mais simples, é a empregada na maioria das simulações e pesquisas (PASCOAL; BECCENERI; MARIETTO, 2005; PESSOA; GAGLIARDI; ALVES, 2004; ALVES, GAGLIARDI, 2006).

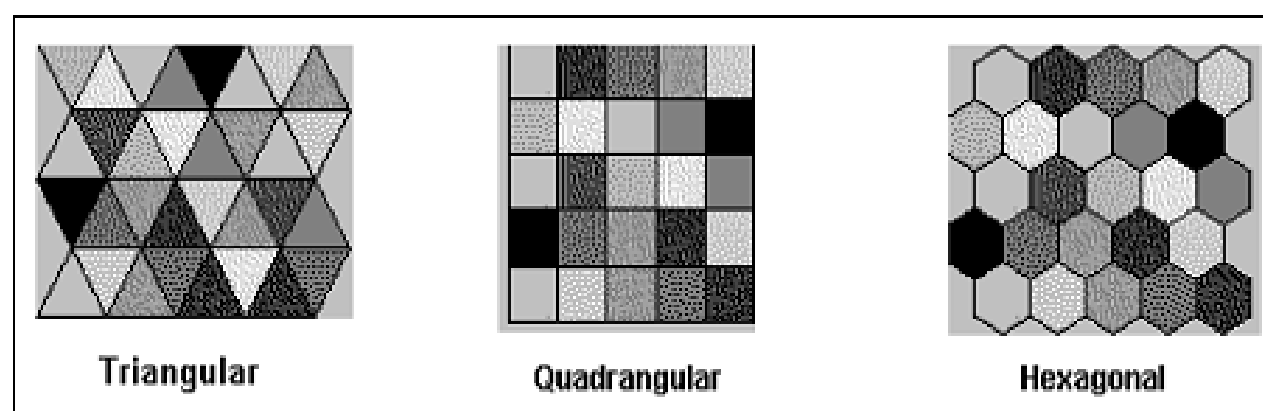


Figura 9. Formas geométricas da célula  
Fonte: PASCOAL, F.; ; BECCENERI, J.; MARIETTO, M. (2005)

Um AC pode ser disposto em uma, duas ou três dimensões denominadas de espaço ou rede unidimensional, bidimensional e tridimensional. No autômato unidimensional, as células estão distribuídas linearmente, ou seja, o reticulado é um vetor, onde o deslocamento acontece em uma direção (Figura 10). No bidimensional, temos uma distribuição no plano, sendo que neste tipo de dimensão é possível variar formas diferentes para as células (Figura 11). O tridimensional (Figura 12) é uma distribuição espacial, cujas células podem assumir formas diferentes assim como no bidimensional (BARONE et al, 2003; PESSOA; GAGLIARDI; ALVES, 2004; ALVES, GAGLIARDI, 2006).

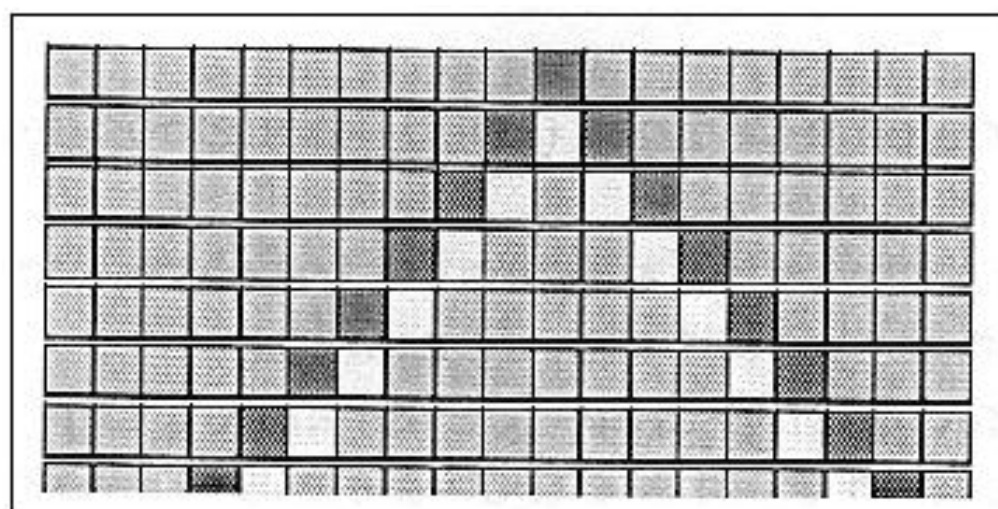


Figura 10. Representação de vários reticulados unidimensionais  
Fonte: BARONE, D. et al (2003)

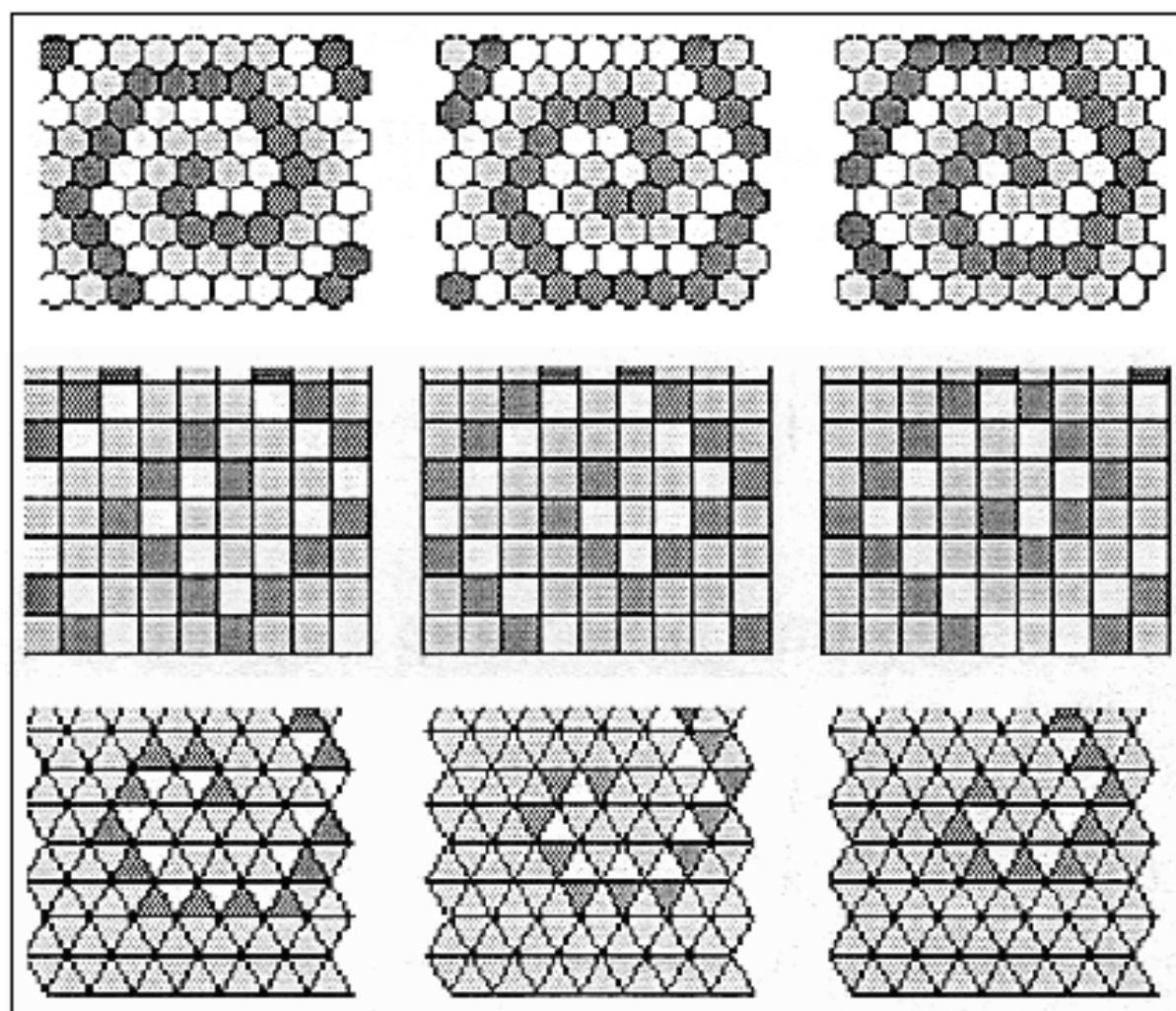


Figura 11. Representação bidimensional  
 Fonte: BARONE, D. et al (2003)

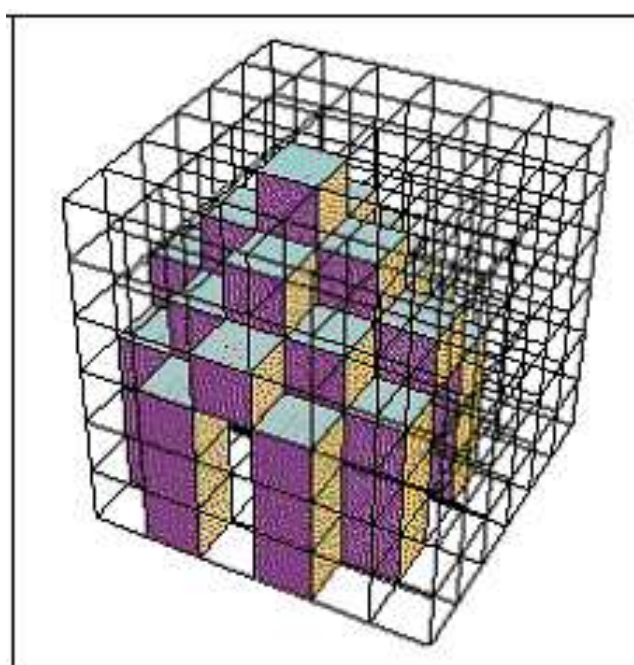


Figura 12. Representação tridimensional  
 Fonte: BARONE, D. et al (2003)

Teoricamente, a rede do AC pode ser finita (simulações computacionais) ou infinita (sem limitações laterais). Contudo, redes finitas devem possuir regras de limite para prever o que acontecerá nas bordas, já que possuem vizinhanças incompletas (PESSOA; GAGLIARDI; ALVES, 2004; ALVES, GAGLIARDI, 2006). Este problema pode ser solucionado com as perspectivas seqüentes (LEITE; LINS; CERQUEIRA, 2008; PASCOAL; BECCENERI; MARIETTO, 2005; MENDES, 2005):

- a) **limite fixo**: células "invisíveis" estariam ao redor da rede com um estado predefinido e esses estados são mantidos durante todo instante da interação (Figura 13);

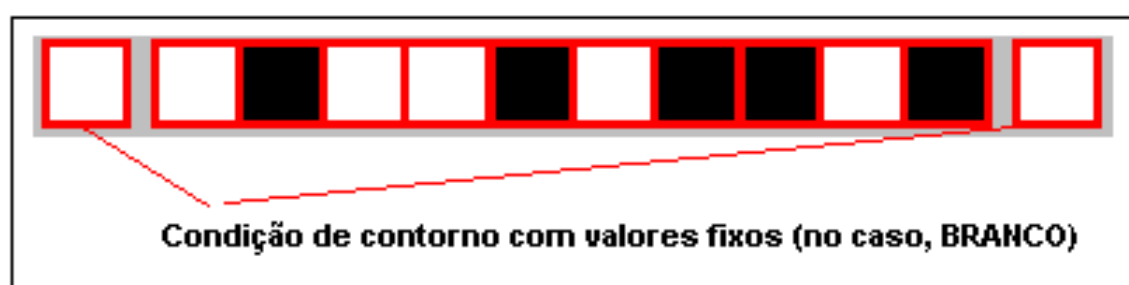


Figura 13. Limite fixo  
Fonte: LEITE, I; LINS, J.; CERQUEIRA, M. (2008)

- b) **limite periódico ou cíclico**: células da borda são vizinhas das células da borda oposta, o qual é representado quando há uma extensão da grade de forma que cada célula da borda tenha uma célula vizinha (Figura 14);

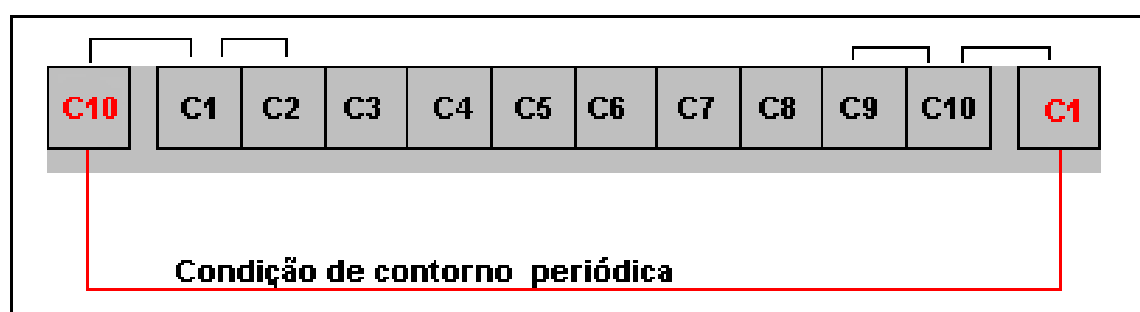


Figura 14. Limite periódico  
Fonte: LEITE, I; LINS, J.; CERQUEIRA, M. (2008)

- c) **limite reflexivo**: as células da borda são refletidas internamente a grade, considerando no lugar da célula ausente na vizinhança uma cópia de uma célula interna desta mesma vizinhança, não necessariamente esta cópia seja a célula de fronteira (Figura 15);

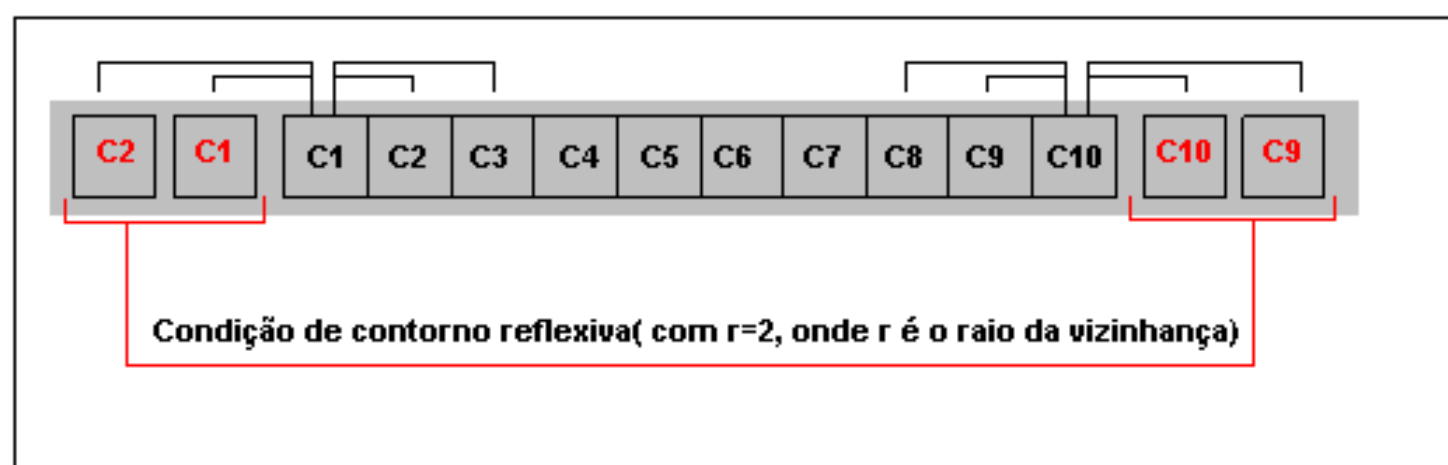


Figura 15. Limite reflexivo  
 Fonte: LEITE, I; LINS, J.; CERQUEIRA, M. (2008)

d) **limite adiabático**: o valor da própria célula da fronteira é repetido nas células “invisíveis” (Figura 16).

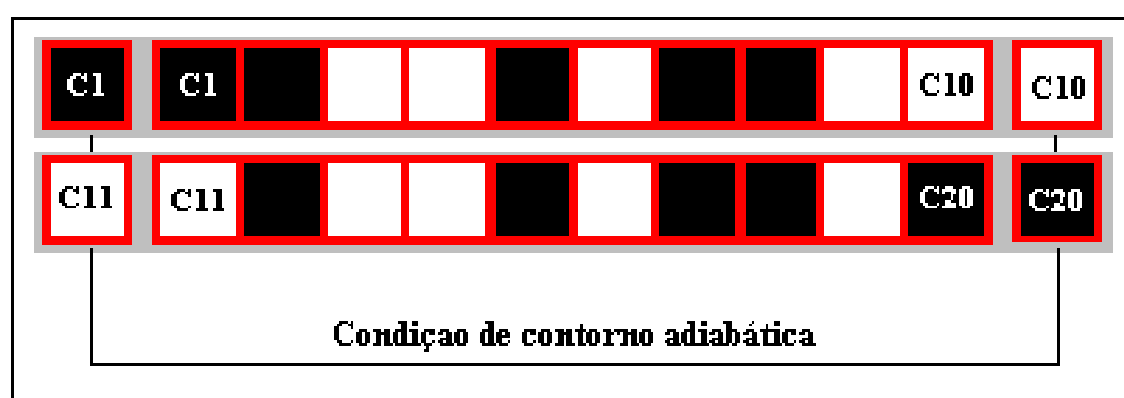


Figura 16. Limite adiabático  
 Fonte: MENDES, R. (2005)

O limite periódico normalmente é aplicado quando a rede de células é finita formando uma rede circular no caso de ACs unidimensionais (VILLELA; CARVALHO, 2007).

Visto que um AC não é infinito, é necessário definir as células que fazem parte das extremidades do AC em questão. A solução para isto é criar regras diferenciadas para estas células, determinando assim comportamento e condições diferenciadas para estas extremas (MENDES, 2005).

O comportamento de um autômato é função de sua configuração inicial. Portanto, a análise desta é essencial para a previsão da forma como ele irá desenvolver durante a interação. Os valores iniciais das variáveis de estado são referenciados como configurações iniciais. Partindo destas condições iniciais, as células evoluem a cada passo de tempo, mudando seus estados através das regras a elas atribuídas (LANZER, 2004).

Ao início da interação a célula pode ter seu comportamento alterado a cada passo de tempo. O conjunto destes comportamentos reúne todos os estados que uma célula pode assumir, sendo que as regras que se aplicam a cada célula dependem do seu estado no momento (BARONE et al, 2003). Os estados são alterados conforme um conjunto de condições de evolução, sendo estas impostas de forma paralela e sincronizada a todas as células (VASCONCELOS, 2005). Por exemplo, no caso de um incêndio florestal pode-se prever alguns estados possíveis de uma célula, tais como: árvore em fogo, mato em fogo e tudo queimado. O conjunto de estados pode ser formado por número (0, 1, 2...) ou propriedades (PASCOAL; BECCENERI; MARIETTO, 2005), como no caso do incêndio florestal.

O número de estados possíveis por célula iniciou-se com 29 estados, idéia iniciada com Von Neumann, mas a maioria dos ACs analisada na atualidade é representada por um número inferior a 29, sendo os binários os mais utilizados (OLIVEIRA, 2003).

Um AC é um modelo matemático dinâmico (THIELO, 1999). Para o funcionamento desejado são introduzidas as condições de evolução. Estas têm por finalidade determinar o estado das células para a próxima iteração. A geração do próximo estado é definida pelo estado da célula mais as suas vizinhas, aquelas que foram pré-determinadas anteriormente. As regras aplicadas atualizam os estados de todas as células simultaneamente (sincronizadamente) baseados em valores (momentâneos) das variáveis de sua vizinhança. Se todas as células tiverem o mesmo conjunto de regras, o AC pode ser dito como homogêneo (PASCOAL; BECCENERI; MARIETTO, 2005).

O novo estado de uma célula depende de seu estado atual e dos estados das células de sua vizinhança pré-definida, assim levando em consideração estas variáveis é aplicada uma regra determinando o novo estado da célula em questão (MENDES, 2005).

A Figura 17 apresenta um AC unidimensional que possui 2 estados ( $k = 2$ ), cujo conjunto de estados  $\Sigma$  pode assumir valores 0 e 1 ( $\Sigma = \{0,1\}$ ), com um reticulado formado por

10 células ( $N = 10$ ). A vizinhança  $\eta_i$  de cada célula  $i$  consiste nela mesma e nos dois vizinhos mais próximos: células  $i-1$  e  $i+1$ . As condições de limite periódicas, ou seja, a célula 1 é considerada como a vizinha da célula 10, reciprocamente. Na Figura 17 são demonstrados os estados das células no reticulado no tempo inicial ( $t = 0$ ) e também a regra de transição do AC formada pelas oito vizinhanças possíveis (regras de transição) e seus bits de atualização. Por exemplo, o valor de atualização em  $t = 1$  da terceira célula do reticulado é 0, uma vez que a vizinhança desta célula em  $t = 0$  é 001 e a regra estabelece a transição 001  $\rightarrow$  0 (AGUIAR; COSTA, 2001; OLIVEIRA, 2003).

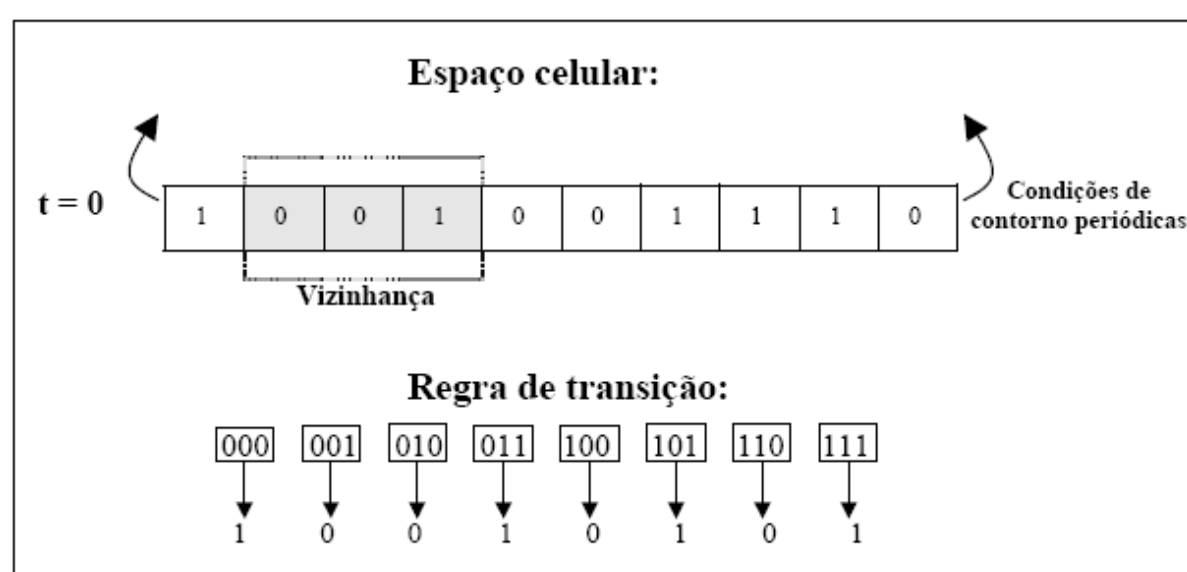


Figura 17. Regra de transição e espaço celular de um AC unidimensional de 2 estados  
Fonte: OLIVEIRA, G. (2003)

A Figura 18 apresenta a atualização dos estados das células do reticulado apresentado na Figura 17, ao se aplicar a regra de transição por 4 passos de tempo.

<b>t = 0</b>	1	0	0	1	0	0	1	1	1	0
<b>t = 1</b>	0	0	0	0	0	0	1	1	0	1
<b>t = 2</b>	0	1	1	1	1	0	1	0	1	0
<b>t = 3</b>	0	1	1	1	0	1	0	1	0	0
<b>t = 4</b>	0	1	1	0	1	0	1	0	0	1

Figura 18. Evolução temporal do reticulado do AC unidimensional da Figura 17.  
Fonte: OLIVEIRA, G. (2003)

A interação acontece quando há a leitura do estado atual da célula e de suas vizinhas e seguidamente são executadas as regras que lhes foram atribuídas. A interação

acontece dependendo do conjunto de regras, e isto se dá em cada célula, e quando concluído as regras atribuídas ocorre a próxima interação. Pode-se ver um exemplo ilustrativo na Figura 19, usando o conjunto de regras do *Game of life*<sup>10</sup> (PASCOAL; BECCENERI; MARIETTO, 2005).

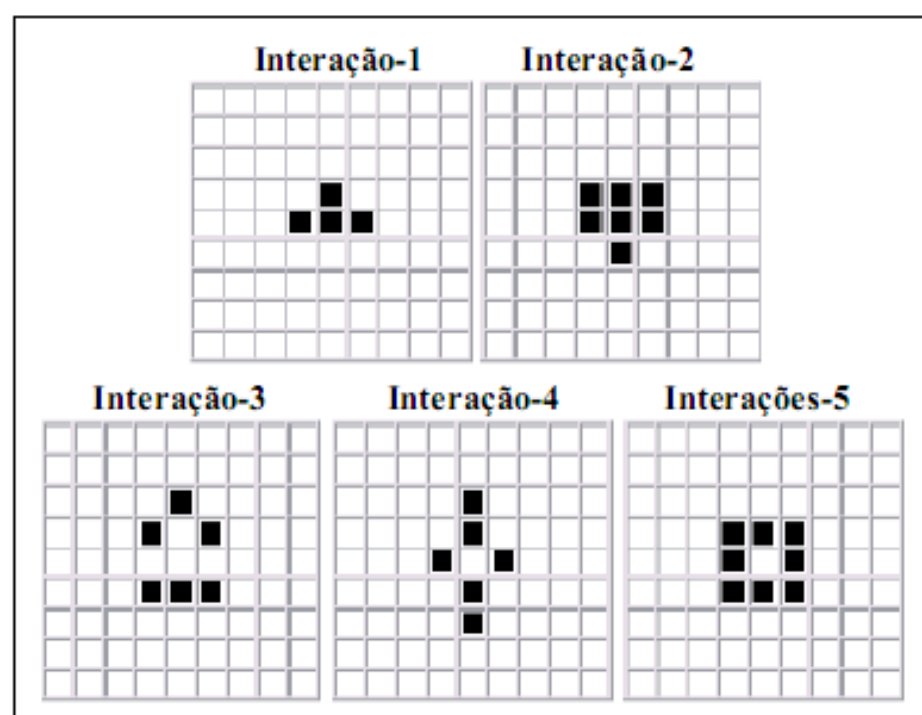


Figura 19. Exemplos de interações  
Fonte: PASCOAL, F. ; BECCENERI, J.; MARIETTO, M. (2005)

As interações ilustradas no AC bidimensional da Figura 19 consideram que a vizinhança da célula central é formada pelas células adjacentes e pelas células das diagonais, sendo as regras do *Game of life* empregadas em todas as células do autômato.

As regras de transição podem ser numerosas dependendo da dimensão do AC, pois quanto mais dimensões, maior será o número de regras possíveis, sendo considerado neste argumento um conjunto de dois estados possíveis para cada célula do AC.

### 3.4 DIMENSÕES DOS AUTÔMATOS CELULARES

Um AC pode ser disposto em rede unidimensional, bidimensional e tridimensional. No autômato unidimensional, as células são distribuídas uma a uma formando

<sup>10</sup> Uma célula viva com menos de dois vizinhos vivos, ela morre. Uma célula viva com mais de três vizinhos vivos, ela morre. Uma célula viva ou morta com três vizinhos vivos se torna uma célula viva. Uma célula com dois vizinhos vivos, continua no mesmo estado (PASCOAL; BECCENERI; MARIETTO, 2005).

um vetor e o movimento acontece somente em uma direção. No bidimensional, temos uma distribuição no plano, considerando as posições leste, oeste, norte e sul ou direita, esquerda, acima e abaixo. O tridimensional é uma distribuição espacial, semelhante a simetria de um cubo, cujas células podem assumir formas diferentes assim como no bidimensional (BARONE et al, 2003; PESSOA; GAGLIARDI; ALVES, 2004; ALVES, GAGLIARDI, 2006).

### 3.4.1 Autômatos Celulares Unidimensionais

Este é o autômato que possui uma única dimensão é considerado o mais simples, onde as células estão distribuídas linearmente, ou seja, o reticulado é um vetor com valores 0 e 1. As células do AC evoluem a passo de tempo, sendo seu próximo estado determinado pelo estado da própria célula e de sua vizinhança que são as células adjacentes de cada lado. Com apenas dois estados possíveis por célula, e a vizinhança formada por 3 células, existem 8 ( $2^3$ ) padrões possíveis para uma vizinhança, com isso 256 ( $2^8$ ) regras possíveis (BARONE et al, 2003).

Nos ACs unidimensionais as células estão ligadas a  $r$  células locais, onde  $r$  é o raio de sua vizinhança, com isso cada célula tem uma vizinhança definida por  $2r + 1$  vizinhos (VILLELA; CARVALHO, 2007).

As interações do AC unidimensionais podem ser visualizadas por gráficos espaço-temporais, nos quais os estados da rede são plotados em função do tempo. A Figura 20 apresenta um gráfico espaço-temporal de um AC, cuja regra de transição no reticulado com  $N = 100$  células (eixo horizontal) e uma interação por 100 passos de tempo (eixo vertical). Visando uma melhor análise do desenvolvimento do autômato, tanto neste caso (Figura 20) como em outras bibliografias, nos gráficos espaço-temporais as células no estado 0 são representadas pela cor branca e no estado 1 são representadas pela cor preta (OLIVEIRA,

2003). A cor cinza é pouca concentração da cor preta, pois as células nos gráficos são extremamente pequenas, dando a impressão de cor cinza.

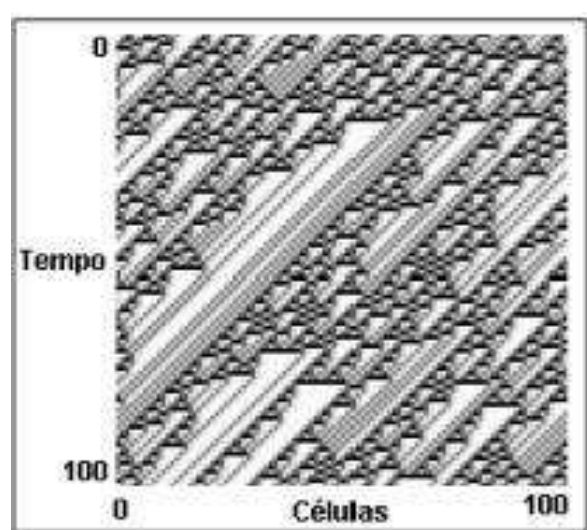


Figura 20. Diagrama de padrões espaço-temporais de um AC  
Fonte: OLIVEIRA, G.(2003)

### 3.4.1.1 Classificação dos Autômatos Celulares Unidimensionais

A classificação de ACs surgiu a partir de estudos de ACs infinitos unidimensionais que, na prática, desconsiderando o estado inicial do autômato, estes apresentam comportamentos constantes (VILLELA; CARVALHO, 2007), que segundo Wolfram (1984) e reafirmado por Aguiar e Costa (2001), Li e Packard (1990) e Mendes (2005), podem ser distintos em quatro classes qualitativas, que definem como se dá a evolução das células em função das regras de atualização das células, a partir das configurações iniciais. São elas:

- a) **classe 1 - pontos limites:** um autômato, após um número finito de interações, alcança um único estado, ou seja, a evolução leva a um estado homogêneo, no qual todas as células eventualmente alcançam o mesmo valor (Figura 21 (a));
- b) **classe 2 - ciclo limite:** o autômato geralmente cria imagens que se repetem periodicamente, com poucos períodos ou imagens estáveis (Figura 21 (b));
- c) **classe 3 - caótica:** o autômato conduz a padrões aperiódicos. Grandes alterações ao longo das interações podem ser afetadas por definições iniciais do autômato (Figura 21(c));

d) **classe 4 - comportamento complexo:** o autômato após um número finito de interações, normalmente morre (Figura 21 (d)). Contudo, as aparições de padrões estáveis podem ser visualizadas algumas vezes como os gerados pela segunda classe (ciclo limite).

O AC pode ser definido de forma quantitativa, conforme demonstra Aguiar et al (2004), onde é mensurado a previsibilidade das conseqüências das transições do AC, visto que é conhecido o estado inicial:

a) **classe 1:** a evolução acaba num estado singular e similar (Figura 21(a));

b) **classe 2:** os resultados de uma célula em particular propagam-se até certo ponto de sua vizinhança, assim a alteração do valor inicial de uma célula somente tem efeito numa determinada parte do AC (Figura 21 (b));

c) **classe 3:** neste caso a alteração do valor inicial de uma célula continuará afetando todo o AC nos próximos espaços de tempo (Figura 21 (c)).

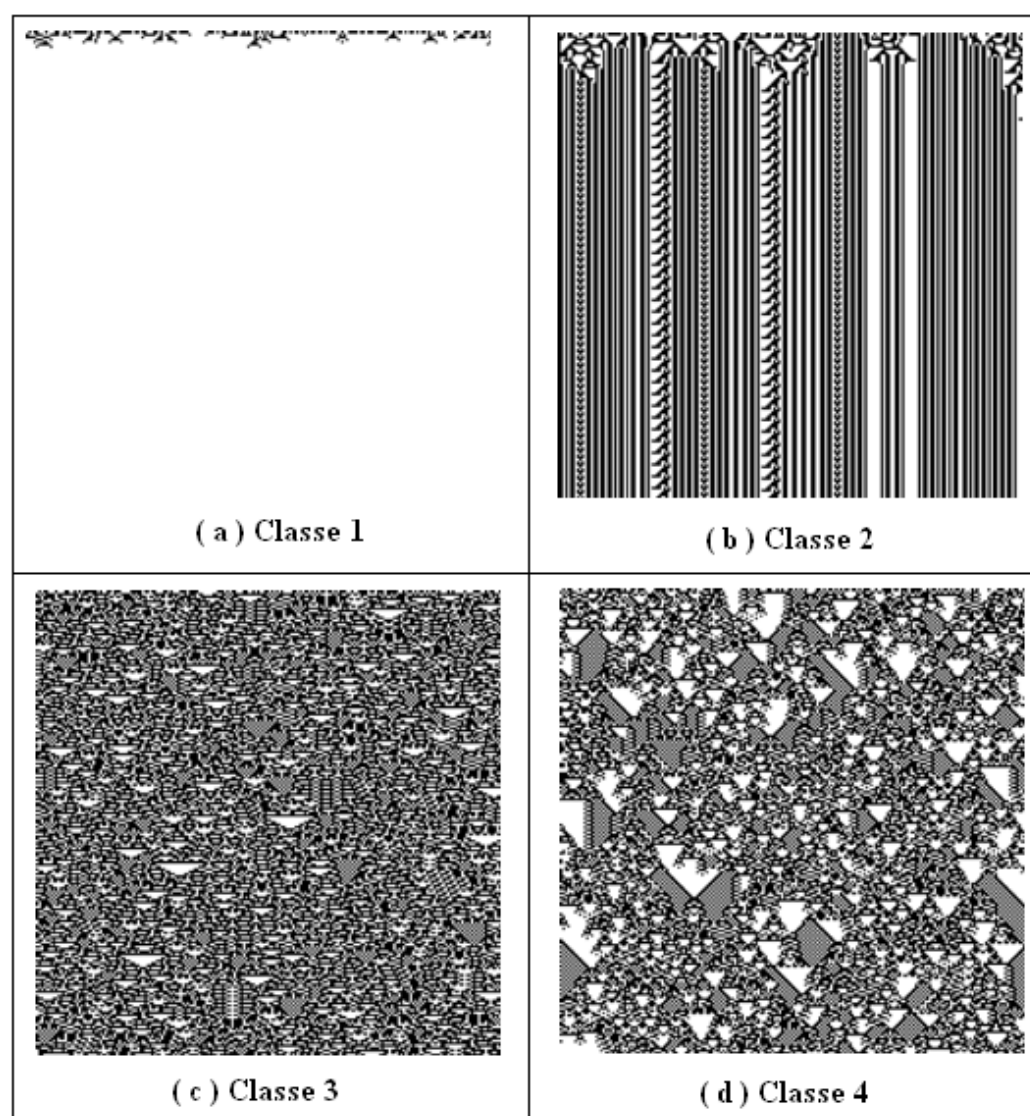


Figura 21. Classificação dos ACs: (a) classe 1, (b) classe 2, (c) classe 3, (d) classe 4  
Fonte: MENDES, R. (2005)

Wolfram (1984) também mostrou que mesmo os modelos de ACs mais simples, que são os unidimensionais binários e de vizinhança 3, exibem dinâmicas pertencentes às quatro classes.

Devido ao comportamento comum dos ACs, independentes de regras ou estados, os mesmos sempre tenderão seu comportamento a uma das quatro classes.

### 3.4.2 Autômatos Celulares Bidimensionais

Em um AC de apenas duas dimensões as células são postas lado a lado formando um plano. Ele implica numa vizinhança celular com alguns elementos, ficando a célula de transição no centro destes elementos. (BARONE et al, 2003). Assim, o cálculo do novo estado de uma célula considera nove bits no processo e como cada célula pode assumir no mínimo dois estados diferentes, existem 512 ( $2^9$ ) padrões possíveis de estados, fornecendo  $1,34 \times 10^{154}$  ( $2^{512}$ ) regras de transição possíveis (JESUS; KAWANO, 2002).

Após definida a dimensão da rede, é necessário indicar as direções para as quais as células irão evoluir, visto que as regras de transição dependem dos estados da célula e de suas vizinhas. Contudo, para que isso aconteça de forma uniforme e padrão é necessário ter uma vizinhança pré-determinada. Segundo Mendes (2005), Pascoal; Becceneri; Marietto (2005) e Alves e Gagliardi (2006), as vizinhanças para espaços bidimensionais mais comuns são:

- a) **Von Neumann:** composta por uma célula acima, abaixo, à direita e outra à esquerda de cada célula, sendo que a vizinhança na Figura 22 (a) possui o raio = 1, podendo ser também encontrada com raio = 2 conforme é demonstrado na Figura 22 (b);

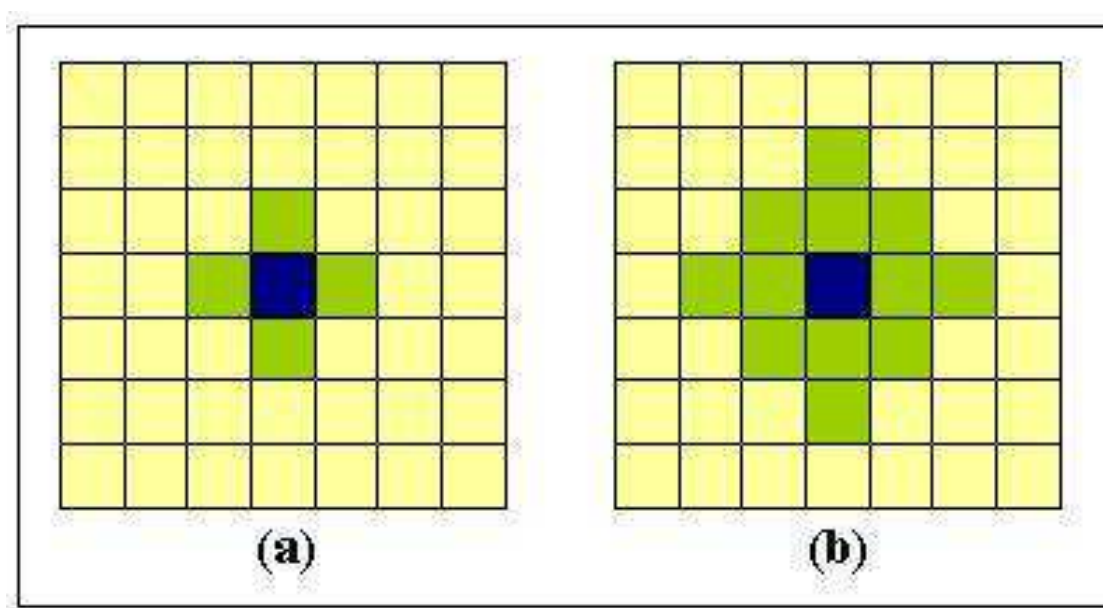


Figura 22. Vizinhança de Von Neumann (a) raio=1 e (b) raio=2  
 Fonte: PASCOAL, F.; BECCENERI, J.; MARIETTO, M. (2005)

b) **Moore**: além das células consideradas no modelo de Von Neumann também são consideradas as células presentes nas diagonais, com possibilidade igual de raio = 1 mostrado na Figura 23 (a) ou raio = 2 demonstrado na Figura 23 (b);

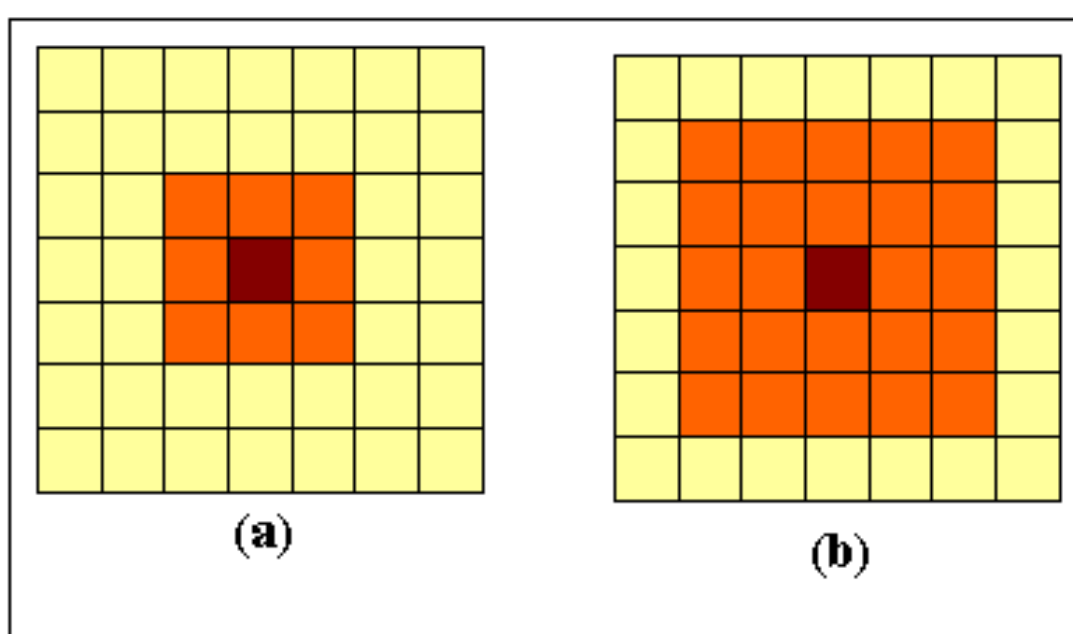


Figura 23. Vizinhança de Moore (a) raio=1 e (b) raio=2  
 Fonte: PASCOAL, F. ; BECCENERI, J.; MARIETTO, M. (2005)

c) **arbitrária ou aleatória**: as células vizinhas ficam espalhadas por toda rede conforme visualizado na Figura 24.

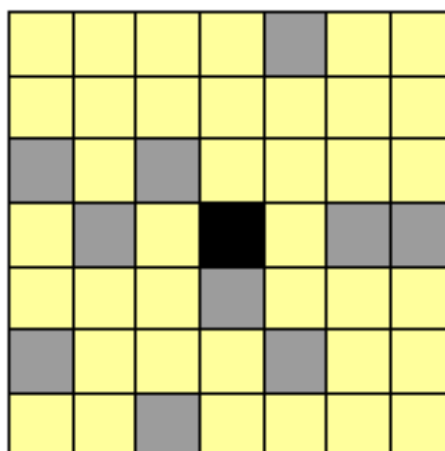


Figura 24. Vizinhança aleatória  
 Fonte: PASCOAL, F. ; BECCENERI, J.; MARIETTO, M. (2005)

O *Game of Life*, proposto por John Conway no final dos anos 60, trata-se de um AC bidimensional utilizando a vizinhança de Moore (BERLEKAMP; CONWAY; GUY, 1992). As transições do *Life* seguem as seguintes regras: Uma célula viva com menos de dois vizinhos vivos, ela morre. Uma célula viva com mais de três vizinhos vivos, ela morre. Uma célula viva ou morta com três vizinhos vivos se torna uma célula viva. Uma célula com dois vizinhos vivos continua no mesmo estado (PASCOAL; BECCENERI; MARIETTO, 2005).

O *Game of Life* é constituído de um reticulado bidimensional binário, começando com uma configuração inicial com um número finito de 1s. Este AC é muito conhecido porque com frequência dá origem a padrões interessantes e complexos no espaço de células, apesar de sua regra de transição simples. Por exemplo, é muito fácil construir configurações iniciais que irão produzir estruturas localizadas simples e propagantes chamadas *gliders*. A Figura 25 apresenta um exemplo de um *glider*, que se move um quadrado diagonalmente a cada 4 passos de tempo.

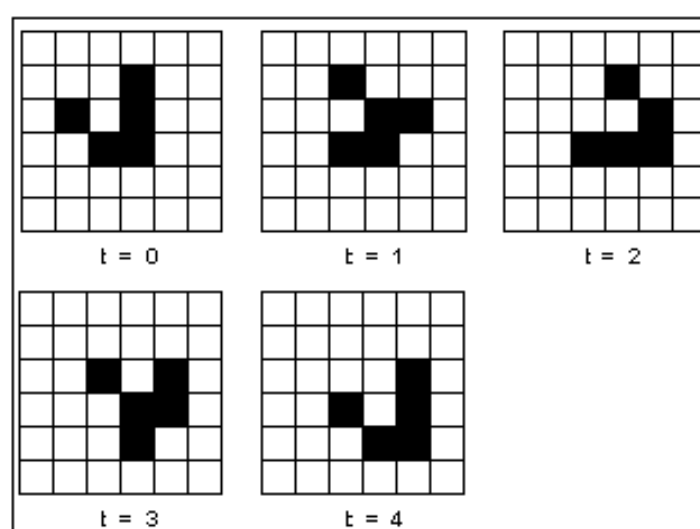


Figura 25. Exemplo de um *glider* de período 4 no *Life*  
 Fonte: OLIVEIRA, G. (2003)

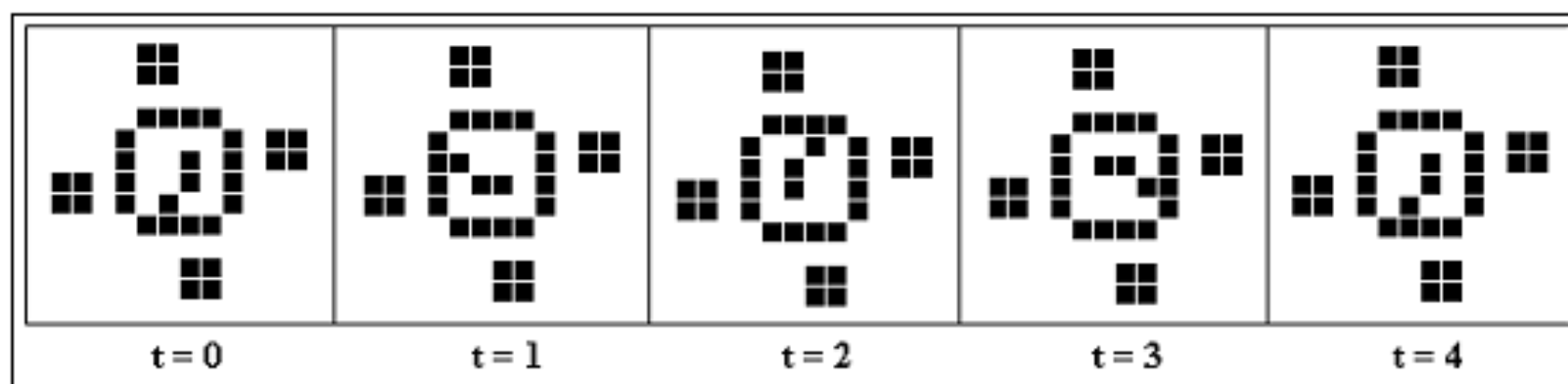


Figura 26. Estrutura periódica no *Life* semelhante a um relógio  
 Fonte: OLIVEIRA, G. (2003)

A Figura 26 apresenta uma outra estrutura periódica interessante encontrada no *Life* que se assemelha a um relógio (OLIVEIRA, 2003).

### 3.4.3 Autômatos Celulares Tridimensionais

O AC tridimensional é uma distribuição espacial, onde também as células podem assumir formas diferentes assim como os ACs bidimensionais. Sua vizinhança é composta por 26 células considerando a célula central e todas as células adjacentes que ficam ao redor da célula central (BARONE et al, 2003). Estes autômatos têm pouca bibliografia motivada por sua complexidade de implementação e pela sua semelhança aos bidimensionais.

## 3.5 ÁREAS DE APLICAÇÕES DOS AUTÔMATOS CELULARES

A célula, menor unidade do AC, contém propriedades simples e realiza poucas tarefas, mas o que a diferencia é sua forte ligação com as células de sua rede na troca de informações. Contudo, cada célula se comporta independentemente tendo em comum somente o resultado final da interação (SIPPER, 1997). Apesar de sua simplicidade, o AC pode ser aplicado em diversas áreas (OLIVEIRA, 2003) dentre as quais podemos citar:

- a) simulação de propagação de incêndios florestais prevendo resultados finais, analisando por meio dos ACs o fogo em si e a topografia do terreno incendiado (GONCALVES; CENTENO; CELLERON, 2003);
- b) modelagem de sistemas biológicos (JESUS; KAWANO, 2002);
- c) evolução de estruturas e modelos no crescimento de organismo (JESUS; KAWANO, 2002);
- d) simulação do deslocamento de nuvens em imagens de satélite, capaz de prever o comportamento de nuvens em um curto período de tempo (GONCALVES; CENTENO; CELLERON, 2003);
- e) processamento de imagens e identificação de modelos visuais (JESUS; KAWANO, 2002);
- f) no ensino de cálculo integral e diferencial na área de exatas para cursos de ensino superior (UEHARA; SILVEIRA, 2008);
- g) geração de notas musicais (MIRANDA, 1995);
- h) no escalonamento de tarefas onde é utilizado mais de um processador (SWIECICKA SEREDYNSKY, 2000);
- i) simulação de espalhamento de epidemias em determinadas áreas geográficas (EMMENDORFER; RODRIGUES, 2000; PEIXOTO; BARROS, 2004); entre outras aplicações.

Os ACs são utilizados em diversas áreas de estudo a fim de simular fenômenos naturais com o intuito de prever futuros comportamentos sejam eles físicos ou químicos tanto de animais ou de vegetais para que o ser humano possa interferir a seu favor, evitando possíveis prejuízos a sociedade.

#### 4 TRABALHOS CORRELATOS

Os ACs estão sendo aplicados em pesquisas e simulações em diversas áreas, segue alguns exemplos:

- a) a Universidade Federal Fluminense (UFF) realizou a pesquisa sobre a evolução de grãos e a distribuição de seu tamanho final utilizando ACs, cuja finalidade do projeto é comparar o método dos AC com evolução microestrutural real do grão. Durante o estudo percebeu-se uma pequena, mas tolerável divergência entre o experimento real e a simulação por AC sendo que era esperado, pois o método é considerado somente uma aproximação do fenômeno observado (ASSIS, 2006);
- b) o Centro Federal de Educação Tecnológica do Paraná (CEFET – PR) juntamente com a *Université Toulouse le Mirail – Maison de La Recherche* (França) realizaram a pesquisa utilizando ACs a fim de simular ambientes naturais como uma floresta visando futuros planejamentos e decisões ambientais. A intenção é, a partir de imagens digitais captadas em tempo real por satélites, poder prever o que acontecerá com uma determinada floresta futuramente. Durante o desenvolvimento foi realizada uma regressão de imagem comparando uma determinada floresta atualmente com sua imagem de 1994. O resultado foi validado, pois a distribuição geográfica da floresta foi satisfatória (GONCALVES; CENTENO; CELLERON, 2003);
- c) a Universidade de São Paulo (USP) realizou uma pesquisa com intuito de analisar e simular o comportamento de AC no estudo da dispersão e propagação de ondas. O AC, neste caso, foi utilizado como um recurso matemático para a demonstração de fenômenos de propagação e difusão a fim

de exemplificar um fenômeno complexo aplicando a teoria de ACs (JESUS; KAWANO, 2002);

- d) na Universidade do Vale do Rio dos Sinos (UNISINOS) foi desenvolvida uma pesquisa sobre ACs binários, unidimensionais, cuja intenção desta é encontrar uma solução para o problema da paridade, que consiste em detectar se a quantidade de 1s em uma configuração binária é um número par ou ímpar. Contudo ao término da pesquisa observou-se que a metodologia utilizada pode não ter sido suficiente para atingir o objetivo final da pesquisa, pois as técnicas apesar de não eficazes podem ser ampliadas para assim chegar no resultado desejado (OLIVEIRA;VAIANO, 2005).
- e) na Escola Politécnica da Universidade de São Paulo (EP-USP) a partir de testes no autômato criado por Gutowitz (1995) que utilizava ACs irreversíveis<sup>11</sup> com sensibilidade unidirecional para geração de chaves criptográficas, descobriu-se que a forma unidirecional estava causando problemas, gerando textos criptografados similares a partir do texto puro, podendo haver quebra do código criptografado. Contudo, após vários testes o autômato unidirecional foi modificado para bidirecional sanando o problema de similaridade. O AC irreversível de Gutowitz (1995) realiza a criptografia considerando o texto a ser criptografado como a rede final formada de um autômato (*backward*). Após a aplicação das regras de transição por várias iterações é obtida, o que podemos chamar de rede inicial e esta apresentada uma configuração de classe 3 – caótico (visto na seção 3.3). A obtenção do texto original é alcançada aplicando a mesma regra (*forward*) sobre a rede

---

<sup>11</sup> Segundo Oliveira, Coelho e Monteiro (2003, p.02), “são aqueles que, dado um reticulado qualquer, aplicando-se a regra de transição sobre ele, chega-se a um novo reticulado e deste não há regra que garanta a sua volta ao reticulado anterior”.

inicial que é o texto criptografado (OLIVEIRA; COELHO; MONTEIRO, 2003).

- f) em Miranda (1995) é apresentado *CAMUS (Cellular Automata Musical)* como uma técnica aplicando AC. *CAMUS* usa o plano cartesiano a fim de representar uma tríade (acorde de três notas musicais). O eixo  $x$  da Figura 27 expressa o primeiro período de tríade, e o eixo  $y$  expressa o segundo período. A criação musical é iniciada ajustando o AC num formato aleatório. Quando o autômato “vive” durante um certo passo de tempo, as coordenadas  $x$  e  $y$  da célula são examinadas gerando os períodos, compondo a tríade.

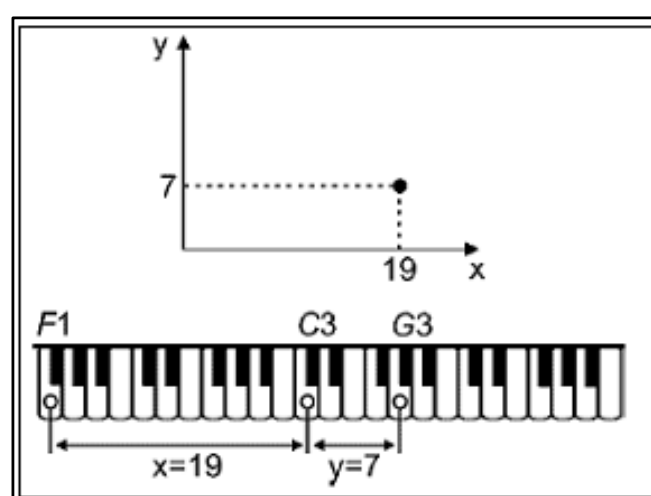


Figura 27. Representação dos intervalos musicais

Fonte: PASCOAL, F. ; BECCENERI, J.; MARIETTO, M. (2005)

- g) em Swiecicka e Seredynsky (2000) é utilizado um AC para descobrir a melhor distribuição de tarefas entre dois processadores de uma estrutura paralela, de acordo com um programa pré-especificado. O programa discrimina as tarefas a serem executadas, a seqüência de execução, o período que cada uma ocupará o processador e o tempo de acesso quando tarefas estão reservadas em processadores diferentes. A Figura 28 apresenta um programa chamado *Gauss 18* que manipula 18 tarefas quaisquer. Os círculos enumerados são interpretados como tarefas. As setas entre os círculos mostram as seqüências de execução das tarefas. Ao lado esquerdo de cada círculo tem-se o tempo total de execução de cada tarefa. Por exemplo, a tarefa 14 só pode ser iniciada após as execuções das tarefas 10 e 11. O número ao lado de cada seta é valor de

acesso entre as tarefas, caso estejam reservadas em processadores diferentes. Assim, se a tarefa 10 estiver alocada no mesmo processador da tarefa 14, esta última pode ser iniciada imediatamente após a realização de 10. Porém, se elas estiverem alocadas em processadores distintos, a tarefa 14 só poderá ser realizada 8 unidades de tempo após o término da tarefa 10. O escalonamento é realizado através de uma rede de dimensão equivalente à quantidade de tarefas a serem executadas, como na Figura 28 são 18 células para as 18 tarefas (OLIVEIRA, 2003).

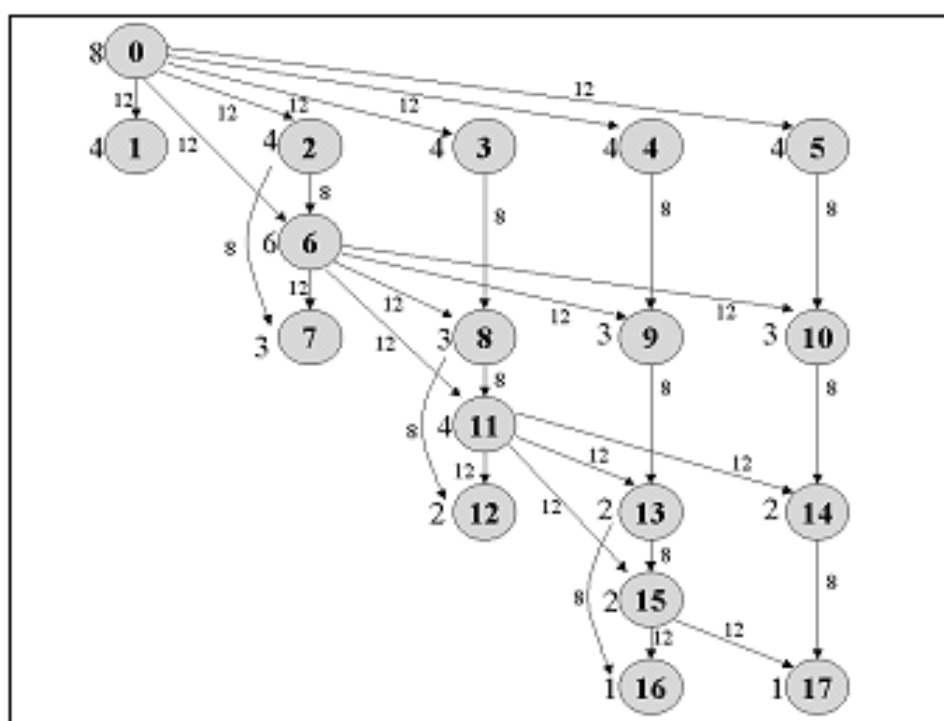


Figura 28. Programa Gauss 18  
Fonte: OLIVEIRA, G. (2003)

h) em Emmendorfer e Rodrigues (2000) e Peixoto e Barros (2004) no modelo SIR<sup>12</sup> simulado no espalhamento de epidemia foi considerado dois tipos de contatos entre os indivíduos: vizinhança de Moore com raio igual a 1, onde a vizinhança é composta pelos 8 vizinhos ao redor da célula; e vizinhança aleatória que são indivíduos que ficam a uma determinada distância  $L$  do indivíduo analisado (Figura 29). A probabilidade de qualquer pessoa ser vítima da epidemia vai depender do número de outros contatos locais ou não-locais

<sup>12</sup> A população que pode ser infectada pela epidemia é dividida em: pessoa que está sujeita a epidemia (S), pessoa infectada (I) e recuperada (R) da epidemia. Sendo regras neste sistema que a pessoa infectada se torna infectante e a recuperada fica imune a epidemia. (EMMENDORFER; RODRIGUES, 2000; PEIXOTO; BARROS, 2004).

infectados, podendo recuperar-se posteriormente ou morrer devido a complicações no quadro clínico. Contudo, para uma pessoa (célula) ser infectada e ser transmissora de tal doença vai depender do número de infectantes que ela tenha a sua volta, nisto é considerado o conceito de vizinhança da célula. Como a população é estável, quando uma pessoa (célula) morre, seguidamente esta pessoa é substituída por outra pessoa sujeita a contrair a doença (EMMENDORFER; RODRIGUES, 2000; PEIXOTO; BARROS, 2004).

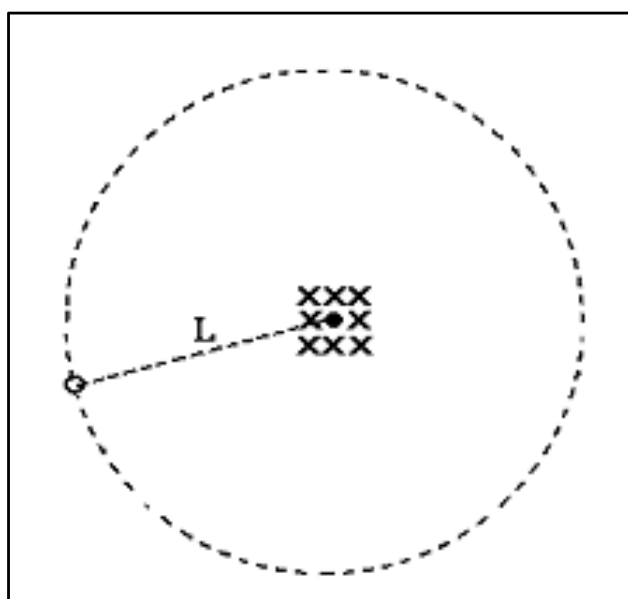


Figura 29. Vizinhança local com indivíduos próximos e vizinhança aleatória com indivíduo a uma distância  $L$ .  
Fonte: EMMENDORFER, L.R; RODRIGUES, L.A.D., (2000)

## 5. IMPLEMENTAÇÃO DO PROTÓTIPO PROPOSTO

O protótipo implementado trata de um alastramento de incêndio florestal aplicando a teoria de ACs. O vasto campo de aplicabilidade da teoria de ACs permite simulações de fenômenos naturais que antes somente eram imaginados agora podem ser representados, se não perfeitamente, mas de modo muito semelhante à realidade. Essas simulações permitem a interdisciplinaridade entre áreas distintas utilizando cálculos exatos em áreas como biologia, ciências sociais, entre outros. Uma teoria de simples implementação, composta por regras de fácil compreensão e manipulação, mas que durante as transições apresenta um elevado grau de complexidade nos seus estados transitórios até seu resultado final.

A teoria de ACs vem de encontro à busca persistente dos pesquisadores do mundo tecnológico: solução de problemas complexos através de sistemas de simples implementação.

A modelagem de um AC necessita a verificação de todas as hipóteses de estado das células, assim como todas as vizinhanças possíveis que ela pode ter, gerando um trabalho que se atém a detalhes minuciosos, mas com um resultado final simples implementação e ajustes para atingir o objetivo almejado. A obtenção do AC desejado necessita estudo sobre suas formas de representação assim o levantamento de todas características do ambiente a ser modelado para que a evolução do AC não apresente desvios fora da realidade.

### 5.1 METODOLOGIA

A metodologia utilizada até a implementação do protótipo foi baseada nas seguintes etapas:

- a) levantamento bibliográfico referente a Máquina de Estados, AFs e ACs;

- b) estudo sobre as áreas de aplicação dos ACs;
- c) definição do protótipo a ser implementado utilizando ACs;
- d) modelagem da interface utilizando uma Linguagem de Modelagem Unificada (*Unified Modeling Language – UML*);
- e) implementação do protótipo definido;
- f) realização de testes e correção de erros.

A necessidade de conhecimento se faz obrigatória para a implementação da teoria de ACs. Inicialmente e para atender a primeira etapa foi realizado um levantamento bibliográfico referente a máquinas de estados, AFs e ACs. A compreensão teórica é primordial para implementação prática, sendo a base em todo o desenvolvimento do protótipo.

A segunda etapa da metodologia foi demonstrada na seção 3.5 (áreas de aplicações dos ACs) e na seção 4 (trabalhos correlatos) por meio de exemplos de trabalhos já realizados aplicando a teoria de ACs permitindo o alinhamento das idéias e seqüentemente a definição do protótipo no qual é aplicada a teoria examinada.

A aplicabilidade da teoria de ACs em diversas áreas, possibilita a modelagem de diversos fenômenos físicos, biológicos, sociais, naturais, entre outros. Incêndios acidentais ou criminosos são constantes em região de densa mata atlântica, atrapalhando a vida selvagem e da sociedade que vive a seu redor. Assim, como forma de representar um problema da sociedade, na etapa seguinte foi definido o protótipo que simula o alastramento de incêndio numa floresta. A floresta definida é fictícia, mas futuramente a mesma teoria ou melhorada pode ser aplicada em uma floresta real. Considerando uma floresta sem seres vivos tais como: mamíferos, aves, peixes, entre outros, a floresta implementada é constituída por células que podem adotar os seguintes estados:

- a) mata verde nativa;
- b) rocha ou pedra nativa;

- c) rio, lagoa ou riacho;
- d) fogo ou mata queimando;
- e) cinza ou mata queimada;

A modelagem do AC foi feita utilizando a linguagem *UML* (Apêndice A), sendo construído um diagrama de estados que demonstra a transição de estados das células que formam o AC. Durante uma interação cada célula do AC, uma a uma e de forma encadeada, é submetida a verificação de seu estado: mata verde, rocha, rio, mata queimando ou cinzas/mata queimada e imediatamente é verificado também o estado de suas células vizinhas a fim de verificar qual regra pode ser aplicável, ocasionando ou não a mudança de estado da célula central.

Após o levantamento bibliográfico e a modelagem em *UML*, partiu-se para as etapas seguintes que são: implementação do protótipo e a realização de testes e correção de erros, onde são aplicados os conceitos levantados sobre a teoria de ACs.

## 5.2 DESENVOLVIMENTO E RESULTADOS OBTIDOS

O protótipo foi desenvolvido no ambiente Java *JDK 6 Update 16* utilizando o *NetBeans 6.7.1* e o sistema operacional utilizado foi o Microsoft® Windows XP. A utilização de tal ambiente é motivada pela disponibilidade gratuita, sendo uma ferramenta utilizada pelos acadêmicos da universidade, permitindo assim que outros pesquisadores possam estudar e dar continuidade ao trabalho iniciado.

O desenvolvimento do protótipo tem como base o estudo e a aplicabilidade da teoria de ACs que, apesar de demonstrar regras de grande simplicidade, permite a representatividade complexa de fenômenos naturais, físicos, químicos, sociais, entre outros.

Tal *software* foi baseado nas seguintes características de um AC:

- a) **forma geométrica da célula:** quadrangular;

- b) **dimensão do AC:** bidimensional;
- c) **vizinhança:** Von Neumann;
- d) **tamanho da vizinhança:** raio igual a 1, sendo as células adjacentes a célula central;
- e) **limite de vizinhança das bordas:** limite adiabático;
- f) **transição de estado:** depende do estado da célula central e dos estados das quatro células da vizinhança.

Para descrever a dinâmica de alastramento de incêndio florestal, a floresta fictícia (Figura 30) modelada foi dividida em 81 células ao longo de sua largura por 81 células ao longo de sua altura, totalizando 6.561 células que apresentam os seguintes estados:

- a) **mata verde nativa:** estado representado pelo número 0 e pela cor verde;
- b) **rochas ou pedras nativas:** estados representados pelo número 1 e pela cor marron;
- c) **rios, lagoas ou riachos:** estados representados pelo número 2 e pela cor azul;
- d) **fogo ou mata queimando:** estados representados pelo número 3 e pela cor vermelha;
- e) **cinzas ou mata queimada:** estados representados pelo número 4 e pela cor cinza.

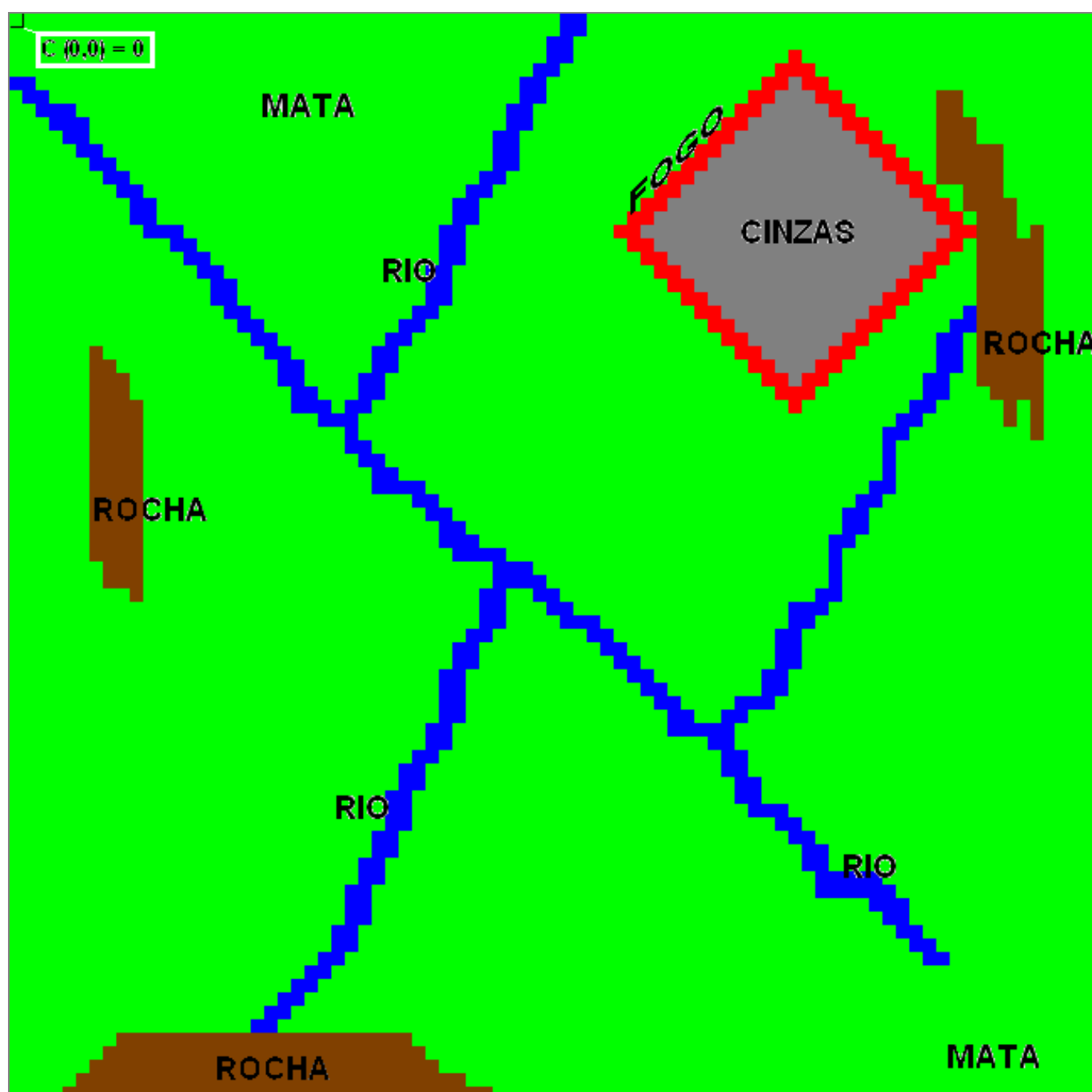


Figura 30. Representação da floresta fictícia

As 6.561 células da rede fazem parte de uma matriz, sendo cada célula identificada por uma coordenada [linha, coluna] e cada coordenada tem um valor que é chamado de *estado da célula*. Exemplificando: a célula da linha 0 e da coluna 0 na Figura 30 apresenta estado 0, por isso seu estado apresenta-se na cor verde que representa mata verde. Uma pequena parcela das células do reticulado podem ser vistas representadas na Figura 31.

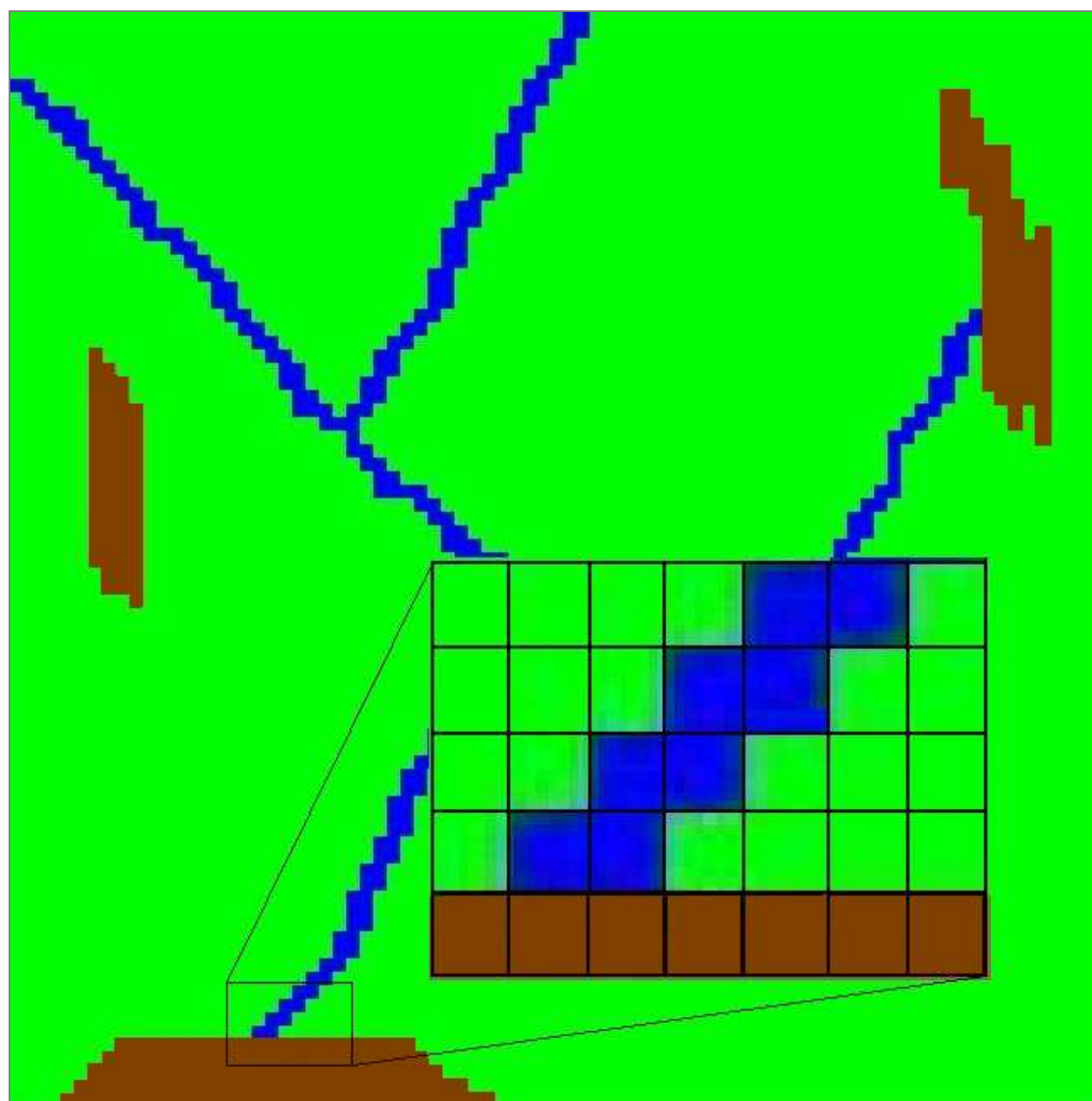


Figura 31. Representação parcial das células na floresta

Durante as transições, todas as células da rede, uma a uma e de forma encadeada, são submetidas a verificação de seu estado e, a partir disto, é verificado o estado de suas células da vizinhança. Uma interação é iniciada:

- a) verificando primeiramente o estado da célula na posição (0,0);
- b) depois analisando a vizinhança da célula da posição (0,0);
- c) aplicando a regra de transição condizente com o estado da posição (0,0) e de suas vizinhas,
- d) alterando ou não o estado da célula da posição (0,0);

Após terminada a verificação da célula da posição (0,0), a mesma rotina é aplicada para as células das posições (0,1), (0,2), (0,3)... até a posição (80,80). Quando é verificado, analisado e alterado ou não o estado da última célula e terminada uma interação do AC, concluindo sua primeira evolução, dentre muitas que virão na seqüência. As células

continuarão evoluindo até que uma condição de parada determinada pelo sistema ou pelo usuário seja executada.

Considerando todos os estados e vizinhanças possíveis, foram descritas todas as regras do AC que estão representadas do Apêndice B, totalizando 3.125 regras de transição.

No caso do protótipo implementado, as regras para a transição de estado das células podem ser classificadas como determinísticas, pois a evolução do AC depende de um único conjunto de regras que contempla todas as transições possíveis do AC. A condição de existir um único conjunto de regras caracteriza um AC como homogêneo, pois todas as células do reticulado são submetidas ao mesmo conjunto de regras (Apêndice B). Com a análise de todas as regras geradas, as mesmas foram resumidas às seguintes condições:

- a) **quando o estado da célula central for mata nativa (0):** se houver qualquer vizinho em estado de fogo, esta célula também passará para estado de fogo, caso contrário ela se manterá em estado de mata nativa;
- b) **quando o estado da célula central for rocha (1):** a mesma se manterá neste estado, pois rocha não sofre interferência do fogo;
- c) **quando o estado da célula central for rio (2):** a mesma se manterá neste estado, pois água não sofre interferência do fogo;
- d) **quando o estado da célula central for fogo (3):** se houver qualquer vizinho em estado de mata nativa, esta célula se manterá em estado de fogo, caso contrário seu estado será alterado para mata queimada ou cinza ;
- e) **quando o estado da célula central for cinza (4):** a mesma se manterá neste estado, pois as cinzas já são resultado da interferência do fogo.

A Figura 32 mostra um pedaço do código fonte que representa as regras detalhadas acima.

```

switch (acCentral) {
  case 0:
    if ((vizsup==3)|| (vizinf==3)|| (vizdir==3)|| (vizesq==3)) {
      acNovo[lin][col] = 3;
    }
    else
      acNovo[lin][col] = 0;
    break;

  case 1: acNovo[lin][col] = 1;
    break;

  case 2: acNovo[lin][col] = 2;
    break;

  case 3:
    // Se qualquer dos quatro vizinhos é mata
    if ((vizsup==0)|| (vizinf==0)|| (vizdir==0)|| (vizesq==0)) {
      acNovo[lin][col] = 3;
    }
    else { // Se dois vizinhos forem fogo e os outros forem inertes (água, rocha)
      // Se os quatro vizinhos forem fogo
      // Se todos os vizinhos forem inertes (água, rocha ou cinza)
      // Se um dos vizinhos for fogo e os outros forem inertes (água, rocha ou cinza)
      // Se dois vizinhos forem fogo e os outros forem inertes (cinza)
      // Se três vizinhos forem fogo e os outros inertes (rocha, água, cinza)
      // Se dois vizinhos forem fogo e outros forem inertes (rocha, água e cinza)
      acNovo[lin][col] = 4;
    }
    break;

  case 4: acNovo[lin][col] = 4;
    break;
}

```

Figura 32. Regras de transição implementadas em código

O problema que se tem na aplicação das regras são os limites de borda do AC que o torna finito. Para sanar este problema é utilizado o conceito de limite adiabático, ou seja, o estado da célula vizinha das células que ficam nas bordas, é o mesmo estado da célula em questão. Exemplificando tem-se a Figura 33 que demonstra o limite de borda. Neste caso está sendo exemplificado a formação de vizinhança do que se considera como o canto inferior esquerdo da floresta.

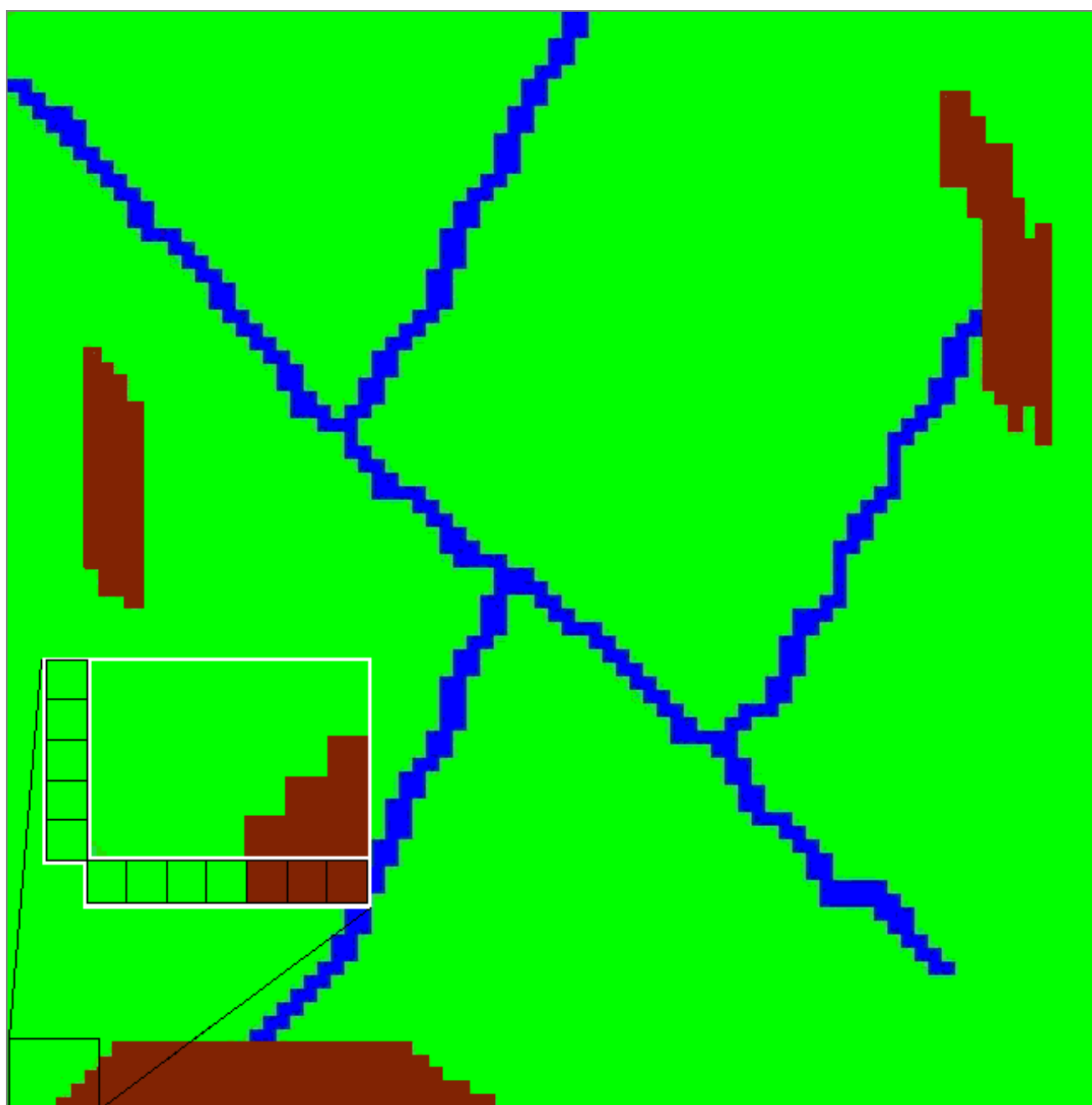


Figura 33. Exemplificando limite adiabático no reticulado da floresta

Pode-se observar que as sete células refletidas na parte inferior possuem as mesmas cores (estados) das células do autômato, permitindo assim a verificação também desta vizinhança. A implementação do limite adiabático pode ser visto no Apêndice C, sendo considerado os quatro cantos do autômato que não possuem vizinhança acima e à direita, acima e à esquerda, abaixo e à direita ou abaixo e à esquerda; e também os lados do autômato que não possuem vizinhança à direita, à esquerda, acima ou abaixo. Nestes casos aplicando o conceito de limite adiabático conforme pode ser visto no código implementado no Apêndice C.

Definidos os estados, as regras de transição e a organização das células e do AC, foi feito o encadeamento que demonstra a evolução do AC na Figura 34.

O código demonstrado na Figura 34 mostra as condições para a evolução do AC. Inicialmente, a matriz é atualizada com as coordenadas do início do incêndio (linha 247 da

Figura 34), com isso a imagem da floresta é redesenhada com o primeiro ponto de fogo o qual foi determinado pelo usuário (linha 251 da Figura 34). Para manter o AC evoluindo foi criada uma *thread* (um processo dividido a si mesmo em duas ou mais tarefas que podem ser executadas paralelamente) possibilitando que, enquanto o AC evolui, ele pode ser interrompido pelas ações de PARAR ou PAUSAR executadas pelo usuário ou o AC será finalizado quando todas as células atingidas pelo *fogo* estiverem em estado de *cinzas*. Mas enquanto o AC não é finalizado ou interrompido, as interações vão ocorrendo. Todas as células são analisadas e seus estados são atualizados por meio de uma função (linha 258 da Figura 34). Com a atualização de todas as células do AC, sua imagem é redesenhada (linha 259 da Figura 34) de acordo com os novos estados. A matriz de estados da interação anterior é atualizada com os novos estados da matriz atual por meio de uma função (linha 260 da Figura 34).

```

247 boolean ok = automato.RecriarMatriz(linha,coluna);
248
249 if (ok)
250 {
251     automato.CriarImagem();
252     new Thread() {
253         @Override
254         public void run() {
255             while ((!pausar) &&(!parar) &&(aindaFogo)) {
256
257                 aindaFogo = automato.aindaHaFogo();
258                 automato.percorrerAutomato();
259                 automato.CriarImagem();
260                 automato.atualizarAutomato();
261             }
262             if (!aindaFogo) {
263
264                 txtArea.setForeground(Color.black);
265                 int perc = automato.areaQueimada();
266                 txtArea.setText("Percentual da floresta queimada: "+perc+"%");
267
268                 btnIniciarSimulacao.setEnabled(true);
269                 btnPausar.setEnabled(false);
270                 btnContinuar.setEnabled(false);
271                 btnParar.setEnabled(false);
272                 btnLimpar.setEnabled(true);
273                 btnFechar.setEnabled(true);
274             }
275         }
276     }.start();
277 }

```

Figura 34. Condições de evolução e encadeamento do AC

Ao finalizar o AC, uma mensagem é apresentada ao usuário mostrando o percentual de floresta que foi queimada (linha 266 da Figura 34).

Quando o *software* é inicializado, o desenho da floresta é feito por meio do carregamento da matriz que possui as 6.561 coordenadas/células. Cada célula da floresta tem um estado que pode variar entre: mata nativa (0), representado pela cor verde; rocha (1), representado pela cor marron; rio (2), representado pela cor azul. Estes estados já são pré-definidos pelo sistema, não sendo possível ser alterado pelo usuário. Com o início da evolução, aparecem fogo/mata queimando (3) representado pela cor vermelha e cinza/mata queimada (4) representado pela cor cinza, que farão parte da evolução do AC na simulação do alastramento do fogo.

O foco do incêndio é iniciado por meio da entrada das coordenadas inseridas pelo usuário (Figura 35), pois o mesmo escolhe onde o fogo deve iniciar, a fim de avaliar seu comportamento ao longo do tempo na evolução do AC.

O fogo irá se alastrar até o momento que o usuário interrompa a evolução do fogo na floresta por meio dos botões PARAR, PAUSAR ou FECHAR (Figura 35) o *software* ou até que o fogo tome conta de tudo que estiver ao seu alcance, transformando em cinzas e quando não tiver mais nada para queimar eles será finalizado.

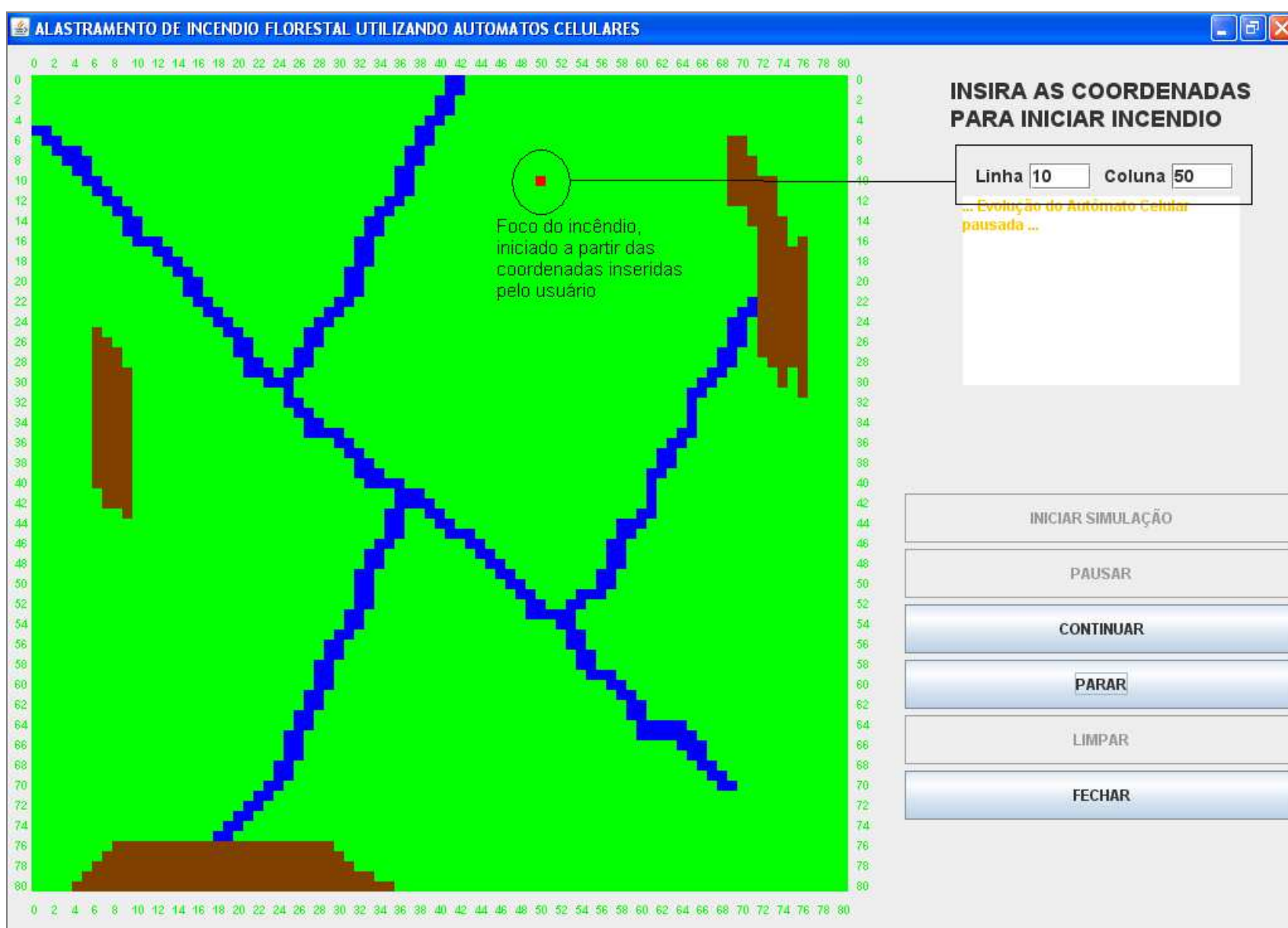


Figura 35. Incêndio iniciado a partir das coordenadas

Ao inserir as coordenadas para iniciar o incêndio, o usuário pressiona o botão INICIAR SIMULAÇÃO. Ao pressionar o botão, o *software* captura as coordenadas, verifica se o estado da célula de tal coordenada é diferente de *rocha* ou *água* (Figura 36), pois os mesmos não pegam fogo. Caso o estado da célula da coordenada indicada pelo usuário for diferente de *rocha* ou *água* é atualizada a matriz, sendo que a célula da coordenada informada pelo usuário passa a ter o estado de fogo (3) e consequentemente é atualizado o desenho da floresta mostrando o início do foco, iniciando a dinâmica de alastramento de incêndio.

```
// Recebe as coordenadas do inicio do fogo e atualiza a matriz de estados do AC
public boolean RecriarMatriz (int lin, int col) {
    if ((acAtual[lin][col]==1) || (acAtual[lin][col]==2) || (acAtual[lin][col]==4)) {
        return false;
    }
    else {
        acAtual[lin][col] = 3;
        return true;
    }
}
```

Figura 36. Atualizando a matriz de estados do AC

Atualizada a matriz de estados, o sistema fica atualizando o estado de todas as células do AC a cada interação, submetendo todas as células ao mesmo conjunto de regras de transição. A evolução do AC acontecerá até que o fogo tome conta de tudo que estiver ao seu alcance transformando em cinza ou até o usuário pause ou pare o sistema.

Durante a evolução do AC o usuário tem a opção de pausar, parar ou fechar o sistema. Ao pausar o sistema o autômato fica estático deixando ativas as opções CONTINUAR ou PARAR a evolução ou ainda simplesmente FECHAR o sistema.

Pressionando o botão CONTINUAR o fogo continuará se alastrando a partir de onde foi pausado, o que permite ao usuário, pausar e continuar várias vezes o AC durante sua evolução, possibilitando a análise dos resultados obtidos.

Quando ativado o botão PARAR, a simulação é interrompida, mantendo a última imagem da floresta queimada e também a última atualização da matriz de estados do AC. Esta ação permite ao usuário iniciar uma nova simulação, possibilitando inserir novas coordenadas para iniciar novamente a simulação do incêndio florestal.

Quando a simulação é reiniciada, primeiramente é recriada a imagem original da floresta e posteriormente são capturadas as novas coordenadas, atualizando a matriz e o foco do incêndio, assim reinicializando o alastramento do incêndio.

Com intuito de entender melhor a evolução do AC será exemplificado o alastramento do fogo a partir da coordenada (43,38), conforme podem ser vistas nas Figuras 37 e 38.

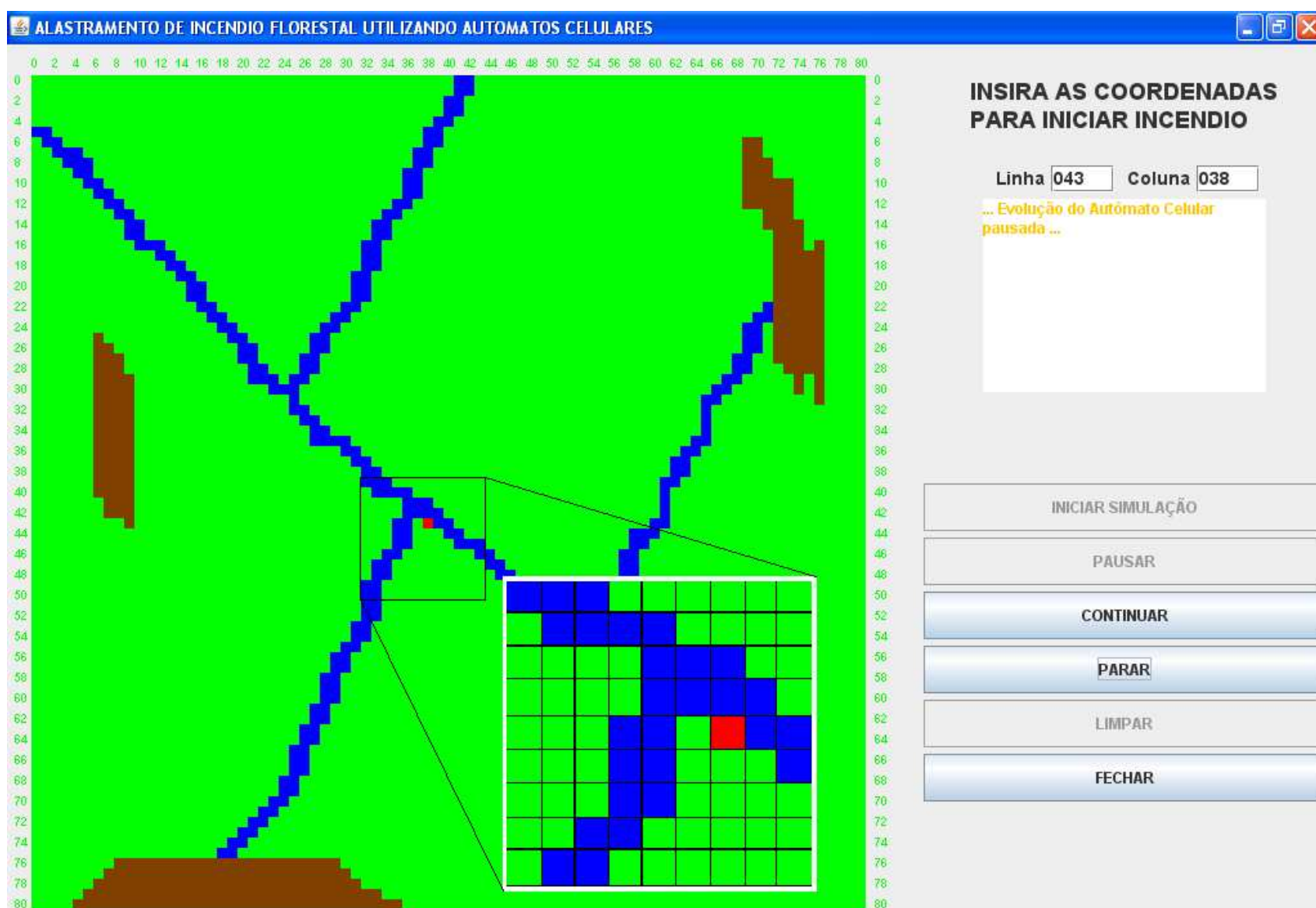


Figura 37. Foco de incêndio com início na coordenada (43,38)

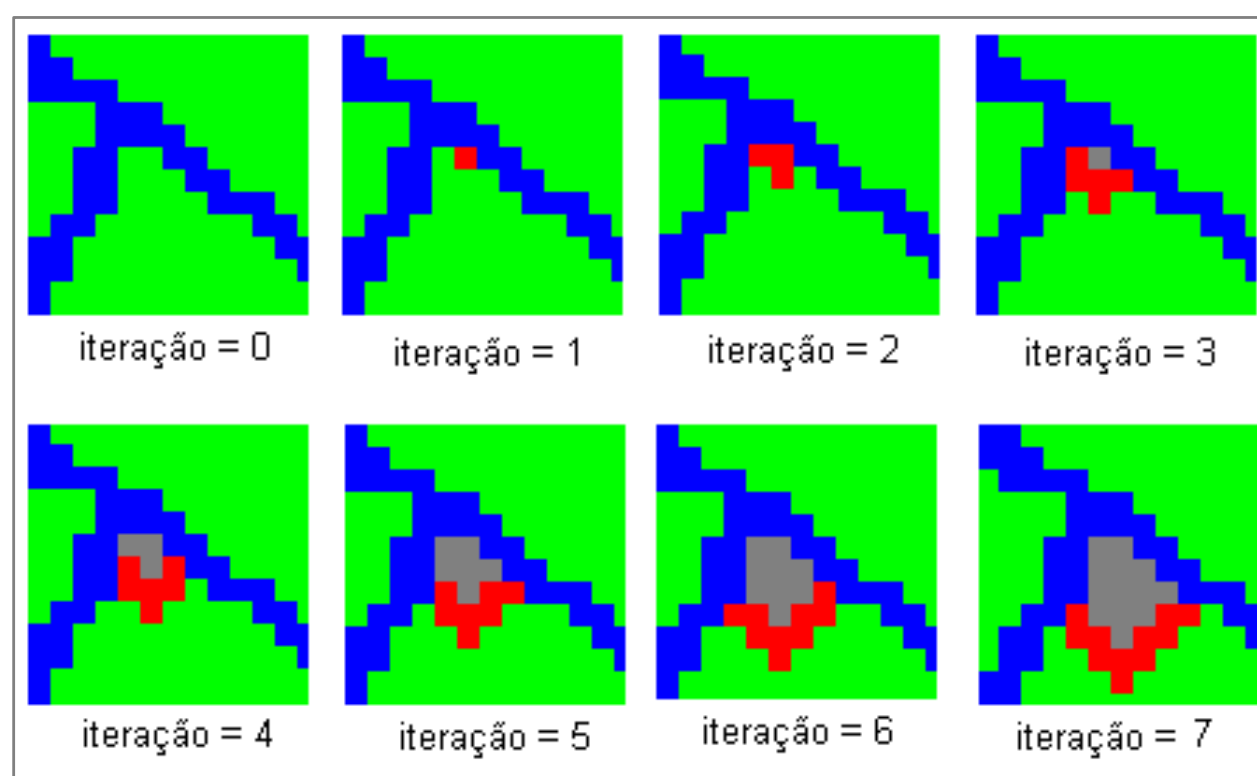


Figura 38. Evolução do AC demonstrando as sete primeiras iterações a partir da coordenada (43,38)

As quatro primeiras iterações da Figura 38 são descritas da seguinte forma:

- a) **iteração 0** : imagem inicial da floresta, sem foco de incêndio;
- b) **iteração 1**: imagem da floresta atualizada com a coordenada (43,38) com estado = fogo;

- c) **iteração 2:** nesta iteração são aplicadas as regras de transição 1504 na célula que fica esquerda do foco (Apêndice B) que alastra o fogo para o “lado” e a regra 1876 na célula abaixo do foco do incêndio (Apêndice B) que alastra o fogo para “baixo”, sendo que todas as outras células se mantêm nos mesmos estados, apesar de todas serem verificadas e aplicadas as regras de transições. As células à esquerda e abaixo foram as únicas que tiveram seus estados alterados nesta interação. A regra 1504 diz que quando a célula central estiver estado 0 (mata), suas vizinhas superior e esquerda estiverem em estado 2 (rio) , a vizinha da direita estiver em 3 (fogo) e a inferior for 0 (mata), a célula central assumirá o estado de fogo ou mata queimando. Já a transição 1876, cuja célula central e as vizinhas da direita, esquerda e inferior forem 0 (mata), mas possui a célula vizinha superior em estado de fogo, a célula central assumirá o novo estado de mata queimando (fogo);
- d) **iteração 3:** nesta iteração é aplicada a regra 1718 (Apêndice B) que altera o estado da célula onde iniciou o incêndio (coordenada 43,38) em cinzas e as regras de transição 2129, 1628 e 1876 (Apêndice B) são aplicadas as demais células que graficamente mostram o alastramento do fogo para frente e para os lados, enquanto que as outras células do reticulado se mantêm em seu estado, já que não tiveram influência do fogo.

A cada interação cada célula é submetida a apenas uma regra de transição

Pode-se observar, com este exemplo, que as células têm seus estados modificados de forma síncrona. São todas verificadas de forma encadeada e sequencial, uma a uma, e caso alguma regra aplique a mudança de estado, a mesma é feita de imediato. O AC obtido é regular, pois sua estrutura se mantém constante ao longo das iterações, tendo o tamanho da floresta inicial igual ao tamanho final, somente com alguns estados modificados entre fogo e

cinzas. É também um AC temporal, porque o mesmo evolui a passos de tempos pré-determinados igualmente em todo o reticulado.

A propagação do fogo é estritamente local, pois depende somente do estado da célula central e suas vizinhas adjacentes para alterar ou não seu estado e seu comportamento global permite uma análise do AC como um todo ao longo de sua evolução. Nesta simulação é desconsiderado possíveis ventos que podem alistar o fogo para lugares distantes atravessando rios, rochas e cinzas.

## CONCLUSÃO

É perceptível em todas as áreas de conhecimento que os conceitos entre as áreas se misturam gerando novos conceitos e novas aplicações. Baseada neste princípio foi criada a teoria de ACs, onde a área computacional se fundiu com a teoria de origem da vida, criando inicialmente um autômato auto-reprodutor e que ao longo do tempo foi lapidada originando o que conhecemos hoje de teoria dos ACs. Simulações de comportamentos naturais realizadas no meio acadêmico permitem ao homem prever desvios que possam prejudicar toda uma sociedade. A teoria de ACs vem de encontro as necessidades do ser humano de cada vez mais antecipar ações, evitando prejuízos futuros.

A pesquisa sobre esta área fascinante tem o intuito de alavancar pesquisas dentro de nossa universidade, tendo como ponto inicial, e já concluído, o levantamento de bibliografias e a demonstração dos conceitos básicos do AC.

Após finalizado o estudo e demonstrada por meio da implementação e execução do protótipo a teoria dos ACs, nota-se que todos os objetivos deste trabalho foram alcançados. A universidade tem disponível uma bibliografia concentrada dando uma visão ampla dessa nova área de estudo pertencente a teoria da computação. Isso permite que outros trabalhos sejam iniciados e concluídos de forma a contribuir com novos softwares ou novas teorias que venham a colaborar na melhoria de problemas educacionais ou sociais.

A maior dificuldade foi implementar graficamente o AC na linguagem Java, já que não foi encontrado nenhum código disponível nessa linguagem orientada a objetos e outros autores citados nas bibliografias, por meio de contato pessoal em forma de e-mails, relatam que tentaram implementar em Java e não obtiveram sucesso. O único código adquirido durante as pesquisas foi em C Builder o qual estava disponível em um fórum na *web*. A dificuldade foi superada com estudos e auxílio do professor da área de programação orientada a objetos do curso de Ciência da Computação.

Proponho como trabalhos futuros nesta área o estudo e a aplicabilidade dos tipos autômatos celulares tais como os irreversíveis, programáveis, naturais, dentre outros vistos superficialmente durante as leituras. Também é válida a idéia de criação de softwares aplicáveis à área de exatas e outras áreas, contribuindo no ensino de crianças, jovens e adultos.

## REFERÊNCIAS

AGUIAR, M. S.; COSTA, A. C. R. **Autômatos celulares para análise da monotonicidade da declividade de áreas geológicas**. Porto Alegre, 2001.

AGUIAR, M. S. ; COSTA, Fábila Amorim da ; DIMURO, Graçaliz Pereira ; COSTA, Antônio Carlos da Rocha . **Modelos matemáticos baseados em autômatos celulares**. São Carlos: SBMAC, 2004.

ALVES, Domingos ; GAGLIARDI, H. F. . **Técnicas de modelagem de processos epidêmicos e evolucionários**. 26. ed. São Carlos: SBMAC, 2006

ARAUJO, Dalton Francisco de. **Modelos computacionais para simulação do processo de expansão da esquistossomose na área litorânea de Pernambuco**. 2007

ASSIS, W. et al. **Estudo da evolução de grãos individuais e distribuição de tamanhos finais pelo método dos autômatos celulares**. Volta Redonda, 2006.

BARONE, D. A. C., SIMOES, E. V., FERRUGEM, A.,BAZZAN, A. L., YEPES, I., FRANZEN, E., SILVEIRA, S. R. **Sociedades artificiais - A Nova Fronteira da Inteligência Das Maquinas**. Porto Alegre RS: Bookman., 2003.

BAR-YAM, Y. **Dynamics of complex systems**. Massachusetts: Addison-Wesley, 1997.

BERLEKAMP, E.; CONWAY, J.; GUY, R. **Winning ways for your mathematical plays**. Vol.2. A K Peters Ltda, 1992.

CASTRO, Marcia Luciana Aguená; CASTRO, Reinaldo de Oliveira. **Autômatos celulares: implementações de Von Neumann, Conway e Wolfram**. Revista de ciências exatas e tecnologia: Anhanguera educacional S.A.. São Paulo, 2008.

EMMENDORFER, L. R. ; RODRIGUES, Luiz Alberto Diaz . **Um modelo de autômatos celulares para o espalhamento geográfico de epidemias: tendências em matemática aplicada e computacional**. São Carlos, 2000.

GERSTING, Judith L. **Fundamentos matemáticos para a ciência da computação**. Trad: Lúcio Leão Filho, Manoel Martins Filho. 4<sup>a</sup>. ed. Rio de Janeiro: LTC, 2001.

GONCALVES, R. M; CENTENO, T.M.; CELLERON, G. **Modelagem preditiva a partir de uma sequência de imagens de satélite utilizando autômatos celulares**. Curitiba, 2003.

GUTOWITZ, Howard. **Cryptography with dynamical systems**. Paris: Laboratoire d'Electronique, 1995.

HOPCROFT, John E.; ULLMAN, Jeffrey D.; MOTWANI, Rajeev. **Introdução à teoria de autômatos, linguagens e computação**. Rio de Janeiro: Elsevier, 2002.

ILACHINSKI, Andrew. **Cellular Automata: a discrete universe**. Singapore, World Scientific, 2001.

JESUS, Ricardo Alves de; KAWANO, Alexandre. **Aplicação de autômatos celulares na propagação de ondas**. São Paulo: Epusp, 2002.

LANZER, André Thiago de Souza. **Um modelo de simulação de autômatos celulares para avaliação de condições de biodiversidade e resiliência na exploração de florestas naturais**. Florianópolis, 2004.

LEITE, I. O. B.; LINS, J., C., S., CERQUEIRA, M., G., C., **Autômatos celulares**. Disponível em: <http://www.di.ufpe.br/~iobl/monografia/index.html>. Acesso em: 01 jun. 2008.

LEWIS, Harry R.; PAPADIMITRIOU, Christos H. **Elementos de teoria da computação**. 2. ed. Porto Alegre: Bookman, 2000.

LI, Wentian; PACKARD, Norman. **The structure of the elementary cellular automata rule space**. Santa Fé, USA, 1990.

MACROCOSMO.COM, Revista eletrônica. **O mistério do big-bang**. São Paulo, ed. 5, abr 2004. Disponível em: <http://www.revistamacrocosmo.com>. Acesso em 10 ago 2009.

MARIANO, A. ; DANTAS, A. ; MACEDO, H. ; OLIVEIRA, G. M. B. . **Novas abordagens na evolução de autômatos celulares unidimensionais na tarefa de sincronização**. V Conferência de Estudos em Engenharia Elétrica, Uberlândia, 2007.

MENDES, Rafael de Souza. **Cel – linguagem de programação celular**. Pelotas, 2005.

MENEZES, Paulo Fernando Blauth. **Linguagens formais e autômatos**. 5. ed. Porto Alegre: Sagra Luzzatto, 2005.

MIRANDA, E. **Granular synthesis of sounds by means of a cellular automata**. 1995.

MISSIO, Maristela; BARROS, Laécio Carvalho de. **Um estudo de autômatos celulares com parâmetros fuzzy para a dispersão da febre aftosa em bovinos no Mato Grosso do Sul**. São Paulo: UNICAMP, 2006.

NÖTH, Winfried. **Máquinas semióticas**. Disponível em: <http://www.revistas.univerciencia.org/index.php/galaxia/article/viewArticle/1265>. Acesso em: 10 ago 2009.

OLIVEIRA, G. M. B.. **Autômatos celulares: aspectos dinâmicos e computacionais**. III Jornada de Mini-cursos em Inteligência Artificial (MCIA). : Sociedade Brasileira de Computação, 2003.

OLIVEIRA G. M. B.; COELHO, A.R.; MONTEIRO, L.H.A. **Criptografia baseada em autômatos celulares com sensibilidade bidirecional**. São Paulo, 2003.

OLIVEIRA, Pedro P.B. de; VAIANO, Robson B. **Buscando um autômato celular para resolver o problema de paridade**. São Paulo, 2005.

PASCOAL, Fabio Sant'Ana; BECCENERI, José Carlos; MARIETTO, Maria das Graças Bruno. **Sociedade artificial fight4life: autômato celular modelando vida artificial**. Instituto Nacional de Pesquisas Espaciais. São José dos Campos, 2005.

PEIXOTO, M. S. ; BARROS, L. C. . **Um estudo de autômatos celulares para o espalhamento geográfico de epidemias com parâmetro fuzzy: tendências em matemática aplicada e computacional**. São Paulo, 2004.

SIPPER, Moshe. **Evolution of parallel cellular machines: The Cellular Programming Approach**. Lausanne: Springer, 1997.

SWIECICKA, A.; SEREDYNSKY, F. **Cellular automata approach to scheduling problem**. Parallel Computing in Electrical Engineering. Proceedings International Conference on, 2000.

SUPERINTERESSANTE. Ed. 159. Rio de Janeiro: Abril, 2000.

THIELO, M. R. **Alguns aspectos teóricos e experimentais sobre sistemas dinâmicos e não-linearidades**. Instituto de Informática: Universidade Federal do Rio Grande do Sul. Porto Alegre. RS, 1999.

UEHARA, Oscar Kiyokazu; SILVEIRA, Ismar Frango. **Aplicação de autômatos celulares no ensino de cálculo diferencial e integral em cursos de computação**. In: WEI – Workshop sobre educação em computação, 2008, Belem do Pará. Anais do XXVIII Congresso da SBC, 2008. p. 238 - 241.

VASCONCELOS. M.H.S. **Computação evolutiva e autômatos celulares**. Universidade Católica de Pelotas. Pelotas, 2005.

VIEIRA, Newton José. **Introdução aos fundamentos da computação: Linguagens e Máquinas**. São Paulo: Thomson, 2006.

VILLELA, S. M. ; CARVALHO, L. A. V. . **O resfriamento simulado no projeto ótimo de autômatos celulares para a geração de chaves em criptografia de fluxo**. ENIA VI Encontro Nacional de Inteligência Artificial. XXVII Congresso da Sociedade Brasileira de Computação – CSBC. Rio de Janeiro, 2007.

WERMUTH, Silvia Letícia Pozzebon. **Modelagem matemática e computacional de um sistema de apoio à decisão na indústria**. Santa Cruz do Sul: Universidade de Santa Cruz do Sul – UNISC, 2007.

WOLFRAM, S. **Cellular automata**. Los Alamos Science, 1983.

WOLFRAM, S. **Universality and complexity in cellular automata**. Physica D, 1984.

**BIBLIOGRAFIA RECOMENDADA**

CHAN, Mark C.; GRIFFITH, Steven W.; IASI, Anthony F. **Java: 1001 dicas de programação.** São Paulo: Makron Books, 1999.

DEITEL, H.M; DEITEL, P. J. **Java: como programar.** São Paulo: Pearson Education do Brasil, 2005.

DIVERIO, Tiarajú Asmuz; MENEZES, Paulo Blauth. **Teoria da computação: máquinas universais e computabilidade.** 2.ed. Porto Alegre: Sagra Luzzatto, 2000.

FOWLER, Martin; SCOTT, Kendal. **UML essencial: um breve guia para a linguagem – padrão de modelagem de objetos.** Tradução: Vera Pezerico e Christian Thomas Price. 2 ed. Porto Alegre: Bookman, 2000.

GRAMB, Tino; et al. **Non-standard computation: Molecular Computation - Cellular Automata - Evolutionary Algorithms - Quantum Computers.** Weinheim: Wiley-VCH, 1998.

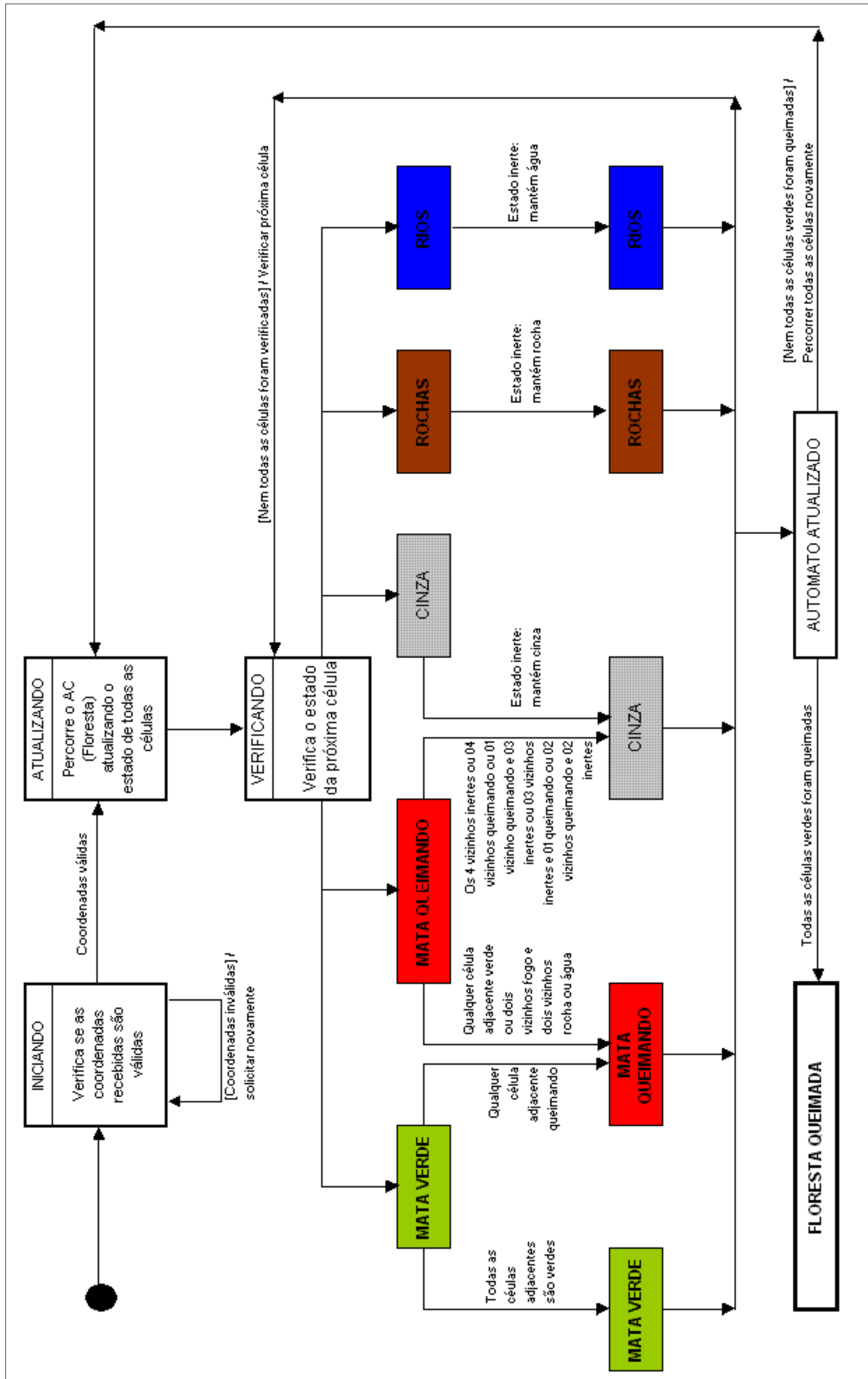
SCHATTEN, Alexander. **Cellular automata tutorial.** Disponível em: <<http://www.schatten.info/info/ca/ca.html>>. Acesso em: 01 jun. 2008.

SHALIZ, Cosma Rohilla. **Causal architecture, complexity and self-organization in time series and cellular automata.** 2001. Disponível em <[www.cscs.umich.edu/~crshalizi/thesis/single-spaced-thesis.pdf](http://www.cscs.umich.edu/~crshalizi/thesis/single-spaced-thesis.pdf)>. Acesso em 24 mar. 2008

SOMMERVILLE, Ian. **Engenharia de software.** São Paulo: Addison-Wesley, 2003.

VIEIRA, Luiz Filipe Menezes; VIEIRA, Marcos Augusto Menezes; VIEIRA, Newton José. **Language Emulator, uma ferramenta de auxílio no ensino de Teoria da Computação.** Disponível em <<http://www.inf.pucpcaldas.br/eventos/weimig2003/ArtigosWEIMIG2003/WEIMIG2003LuizFilipeMenezes.pdf>>. Acesso em 24 mar. 2008.

APÊNDICE A – DIAGRAMA DE ESTADOS DO AUTOMATO CELULAR



# APÊNDICE B – REGRAS DE TRANSIÇÃO

Regra 01	Regra 02	Regra 03	Regra 04	Regra 05	Regra 126	Regra 127	Regra 128	Regra 129	Regra 130
Regra 06	Regra 07	Regra 08	Regra 09	Regra 10	Regra 131	Regra 132	Regra 133	Regra 134	Regra 135
Regra 11	Regra 12	Regra 13	Regra 14	Regra 15	Regra 136	Regra 137	Regra 138	Regra 139	Regra 140
Regra 16	Regra 17	Regra 18	Regra 19	Regra 20	Regra 141	Regra 142	Regra 143	Regra 144	Regra 145
Regra 21	Regra 22	Regra 23	Regra 24	Regra 25	Regra 146	Regra 147	Regra 148	Regra 149	Regra 150
Regra 26	Regra 27	Regra 28	Regra 29	Regra 30	Regra 151	Regra 152	Regra 153	Regra 154	Regra 155
Regra 31	Regra 32	Regra 33	Regra 34	Regra 35	Regra 156	Regra 157	Regra 158	Regra 159	Regra 160
Regra 36	Regra 37	Regra 38	Regra 39	Regra 40	Regra 161	Regra 162	Regra 163	Regra 164	Regra 165
Regra 41	Regra 42	Regra 43	Regra 44	Regra 45	Regra 166	Regra 167	Regra 168	Regra 169	Regra 170
Regra 46	Regra 47	Regra 48	Regra 49	Regra 50	Regra 171	Regra 172	Regra 173	Regra 174	Regra 175
Regra 51	Regra 52	Regra 53	Regra 54	Regra 55	Regra 176	Regra 177	Regra 178	Regra 179	Regra 180
Regra 56	Regra 57	Regra 58	Regra 59	Regra 60	Regra 181	Regra 182	Regra 183	Regra 184	Regra 185
Regra 61	Regra 62	Regra 63	Regra 64	Regra 65	Regra 186	Regra 187	Regra 188	Regra 189	Regra 190
Regra 66	Regra 67	Regra 68	Regra 69	Regra 70	Regra 191	Regra 192	Regra 193	Regra 194	Regra 195
Regra 71	Regra 72	Regra 73	Regra 74	Regra 75	Regra 196	Regra 197	Regra 198	Regra 199	Regra 200
Regra 76	Regra 77	Regra 78	Regra 79	Regra 80	Regra 201	Regra 202	Regra 203	Regra 204	Regra 205
Regra 81	Regra 82	Regra 83	Regra 84	Regra 85	Regra 206	Regra 207	Regra 208	Regra 209	Regra 210
Regra 86	Regra 87	Regra 88	Regra 89	Regra 90	Regra 211	Regra 212	Regra 213	Regra 214	Regra 215
Regra 91	Regra 92	Regra 93	Regra 94	Regra 95	Regra 216	Regra 217	Regra 218	Regra 219	Regra 220
Regra 96	Regra 97	Regra 98	Regra 99	Regra 100	Regra 221	Regra 222	Regra 223	Regra 224	Regra 225
Regra 101	Regra 102	Regra 103	Regra 104	Regra 105	Regra 226	Regra 227	Regra 228	Regra 229	Regra 230
Regra 106	Regra 107	Regra 108	Regra 109	Regra 110	Regra 231	Regra 232	Regra 233	Regra 234	Regra 235
Regra 111	Regra 112	Regra 113	Regra 114	Regra 115	Regra 236	Regra 237	Regra 238	Regra 239	Regra 240
Regra 116	Regra 117	Regra 118	Regra 119	Regra 120	Regra 241	Regra 242	Regra 243	Regra 244	Regra 245
Regra 121	Regra 122	Regra 123	Regra 124	Regra 125	Regra 246	Regra 247	Regra 248	Regra 249	Regra 250

## APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação)

Regra 251 	Regra 252 	Regra 253 	Regra 254 	Regra 255 	Regra 376 	Regra 377 	Regra 378 	Regra 379 	Regra 380 
Regra 256 	Regra 257 	Regra 258 	Regra 259 	Regra 260 	Regra 381 	Regra 382 	Regra 383 	Regra 384 	Regra 385 
Regra 261 	Regra 262 	Regra 263 	Regra 264 	Regra 265 	Regra 386 	Regra 387 	Regra 388 	Regra 389 	Regra 390 
Regra 266 	Regra 267 	Regra 268 	Regra 269 	Regra 270 	Regra 391 	Regra 392 	Regra 393 	Regra 394 	Regra 395 
Regra 271 	Regra 272 	Regra 273 	Regra 274 	Regra 275 	Regra 396 	Regra 397 	Regra 398 	Regra 399 	Regra 400 
Regra 276 	Regra 277 	Regra 278 	Regra 279 	Regra 280 	Regra 401 	Regra 402 	Regra 403 	Regra 404 	Regra 405 
Regra 281 	Regra 282 	Regra 283 	Regra 284 	Regra 285 	Regra 406 	Regra 407 	Regra 408 	Regra 409 	Regra 410 
Regra 286 	Regra 287 	Regra 288 	Regra 289 	Regra 290 	Regra 411 	Regra 412 	Regra 413 	Regra 414 	Regra 415 
Regra 291 	Regra 292 	Regra 293 	Regra 294 	Regra 295 	Regra 416 	Regra 417 	Regra 418 	Regra 419 	Regra 420 
Regra 296 	Regra 297 	Regra 298 	Regra 299 	Regra 300 	Regra 421 	Regra 422 	Regra 423 	Regra 424 	Regra 425 
Regra 301 	Regra 302 	Regra 303 	Regra 304 	Regra 305 	Regra 426 	Regra 427 	Regra 428 	Regra 429 	Regra 430 
Regra 306 	Regra 307 	Regra 308 	Regra 309 	Regra 310 	Regra 431 	Regra 432 	Regra 433 	Regra 434 	Regra 435 
Regra 311 	Regra 312 	Regra 313 	Regra 314 	Regra 315 	Regra 436 	Regra 437 	Regra 438 	Regra 439 	Regra 440 
Regra 316 	Regra 317 	Regra 318 	Regra 319 	Regra 320 	Regra 441 	Regra 442 	Regra 443 	Regra 444 	Regra 445 
Regra 321 	Regra 322 	Regra 323 	Regra 324 	Regra 325 	Regra 446 	Regra 447 	Regra 448 	Regra 449 	Regra 450 
Regra 326 	Regra 327 	Regra 328 	Regra 329 	Regra 330 	Regra 451 	Regra 452 	Regra 453 	Regra 454 	Regra 455 
Regra 331 	Regra 332 	Regra 333 	Regra 334 	Regra 335 	Regra 456 	Regra 457 	Regra 458 	Regra 459 	Regra 460 
Regra 336 	Regra 337 	Regra 338 	Regra 339 	Regra 340 	Regra 461 	Regra 462 	Regra 463 	Regra 464 	Regra 465 
Regra 341 	Regra 342 	Regra 343 	Regra 344 	Regra 345 	Regra 466 	Regra 467 	Regra 468 	Regra 469 	Regra 470 
Regra 346 	Regra 347 	Regra 348 	Regra 349 	Regra 350 	Regra 471 	Regra 472 	Regra 473 	Regra 474 	Regra 475 
Regra 351 	Regra 352 	Regra 353 	Regra 354 	Regra 355 	Regra 476 	Regra 477 	Regra 478 	Regra 479 	Regra 480 
Regra 356 	Regra 357 	Regra 358 	Regra 359 	Regra 360 	Regra 481 	Regra 482 	Regra 483 	Regra 484 	Regra 485 
Regra 361 	Regra 362 	Regra 363 	Regra 364 	Regra 365 	Regra 486 	Regra 487 	Regra 488 	Regra 489 	Regra 490 
Regra 366 	Regra 367 	Regra 368 	Regra 369 	Regra 370 	Regra 491 	Regra 492 	Regra 493 	Regra 494 	Regra 495 
Regra 371 	Regra 372 	Regra 373 	Regra 374 	Regra 375 	Regra 496 	Regra 497 	Regra 498 	Regra 499 	Regra 500 

# APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação)

Regra 501	Regra 502	Regra 503	Regra 504	Regra 505	Regra 626	Regra 627	Regra 628	Regra 629	Regra 630
Regra 506	Regra 507	Regra 508	Regra 509	Regra 510	Regra 631	Regra 632	Regra 633	Regra 634	Regra 635
Regra 511	Regra 512	Regra 513	Regra 514	Regra 515	Regra 636	Regra 637	Regra 638	Regra 639	Regra 640
Regra 516	Regra 517	Regra 518	Regra 519	Regra 520	Regra 641	Regra 642	Regra 643	Regra 644	Regra 645
Regra 521	Regra 522	Regra 523	Regra 524	Regra 525	Regra 646	Regra 647	Regra 648	Regra 649	Regra 650
Regra 526	Regra 527	Regra 528	Regra 529	Regra 530	Regra 651	Regra 652	Regra 653	Regra 654	Regra 655
Regra 531	Regra 532	Regra 533	Regra 534	Regra 535	Regra 656	Regra 657	Regra 658	Regra 659	Regra 660
Regra 536	Regra 537	Regra 538	Regra 539	Regra 540	Regra 661	Regra 662	Regra 663	Regra 664	Regra 665
Regra 541	Regra 542	Regra 543	Regra 544	Regra 545	Regra 666	Regra 667	Regra 668	Regra 669	Regra 670
Regra 546	Regra 547	Regra 548	Regra 549	Regra 550	Regra 671	Regra 672	Regra 673	Regra 674	Regra 675
Regra 551	Regra 552	Regra 553	Regra 554	Regra 555	Regra 676	Regra 677	Regra 678	Regra 679	Regra 680
Regra 556	Regra 557	Regra 558	Regra 559	Regra 560	Regra 681	Regra 682	Regra 683	Regra 684	Regra 685
Regra 561	Regra 562	Regra 563	Regra 564	Regra 565	Regra 686	Regra 687	Regra 688	Regra 689	Regra 690
Regra 566	Regra 567	Regra 568	Regra 569	Regra 570	Regra 691	Regra 692	Regra 693	Regra 694	Regra 695
Regra 571	Regra 572	Regra 573	Regra 574	Regra 575	Regra 696	Regra 697	Regra 698	Regra 699	Regra 700
Regra 576	Regra 577	Regra 578	Regra 579	Regra 580	Regra 701	Regra 702	Regra 703	Regra 704	Regra 705
Regra 581	Regra 582	Regra 583	Regra 584	Regra 585	Regra 706	Regra 707	Regra 708	Regra 709	Regra 710
Regra 586	Regra 587	Regra 588	Regra 589	Regra 590	Regra 711	Regra 712	Regra 713	Regra 714	Regra 715
Regra 591	Regra 592	Regra 593	Regra 594	Regra 595	Regra 716	Regra 717	Regra 718	Regra 719	Regra 720
Regra 596	Regra 597	Regra 598	Regra 599	Regra 600	Regra 721	Regra 722	Regra 723	Regra 724	Regra 725
Regra 601	Regra 602	Regra 603	Regra 604	Regra 605	Regra 726	Regra 727	Regra 728	Regra 729	Regra 730
Regra 606	Regra 607	Regra 608	Regra 609	Regra 610	Regra 731	Regra 732	Regra 733	Regra 734	Regra 735
Regra 611	Regra 612	Regra 613	Regra 614	Regra 615	Regra 736	Regra 737	Regra 738	Regra 739	Regra 740
Regra 616	Regra 617	Regra 618	Regra 619	Regra 620	Regra 741	Regra 742	Regra 743	Regra 744	Regra 745
Regra 621	Regra 622	Regra 623	Regra 624	Regra 625	Regra 746	Regra 747	Regra 748	Regra 749	Regra 750

# APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação)

Regra 751	Regra 752	Regra 753	Regra 754	Regra 755	Regra 876	Regra 877	Regra 878	Regra 879	Regra 880
Regra 756	Regra 757	Regra 758	Regra 759	Regra 760	Regra 881	Regra 882	Regra 883	Regra 884	Regra 885
Regra 761	Regra 762	Regra 763	Regra 764	Regra 765	Regra 886	Regra 887	Regra 888	Regra 889	Regra 890
Regra 766	Regra 767	Regra 768	Regra 769	Regra 770	Regra 891	Regra 892	Regra 893	Regra 894	Regra 895
Regra 771	Regra 772	Regra 773	Regra 774	Regra 775	Regra 896	Regra 897	Regra 898	Regra 899	Regra 900
Regra 776	Regra 777	Regra 778	Regra 779	Regra 780	Regra 901	Regra 902	Regra 903	Regra 904	Regra 905
Regra 781	Regra 782	Regra 783	Regra 784	Regra 785	Regra 906	Regra 907	Regra 908	Regra 909	Regra 910
Regra 786	Regra 787	Regra 788	Regra 789	Regra 790	Regra 911	Regra 912	Regra 913	Regra 914	Regra 915
Regra 791	Regra 792	Regra 793	Regra 794	Regra 795	Regra 916	Regra 917	Regra 918	Regra 919	Regra 920
Regra 796	Regra 797	Regra 798	Regra 799	Regra 800	Regra 921	Regra 922	Regra 923	Regra 924	Regra 925
Regra 801	Regra 802	Regra 803	Regra 804	Regra 805	Regra 926	Regra 927	Regra 928	Regra 929	Regra 930
Regra 806	Regra 807	Regra 808	Regra 809	Regra 810	Regra 931	Regra 932	Regra 933	Regra 934	Regra 935
Regra 811	Regra 812	Regra 813	Regra 814	Regra 815	Regra 936	Regra 937	Regra 938	Regra 939	Regra 940
Regra 816	Regra 817	Regra 818	Regra 819	Regra 820	Regra 941	Regra 942	Regra 943	Regra 944	Regra 945
Regra 821	Regra 822	Regra 823	Regra 824	Regra 825	Regra 946	Regra 947	Regra 948	Regra 949	Regra 950
Regra 826	Regra 827	Regra 828	Regra 829	Regra 830	Regra 951	Regra 952	Regra 953	Regra 954	Regra 955
Regra 831	Regra 832	Regra 833	Regra 834	Regra 835	Regra 956	Regra 957	Regra 958	Regra 959	Regra 960
Regra 836	Regra 837	Regra 838	Regra 839	Regra 840	Regra 961	Regra 962	Regra 963	Regra 964	Regra 965
Regra 841	Regra 842	Regra 843	Regra 844	Regra 845	Regra 966	Regra 967	Regra 968	Regra 969	Regra 970
Regra 846	Regra 847	Regra 848	Regra 849	Regra 850	Regra 971	Regra 972	Regra 973	Regra 974	Regra 975
Regra 851	Regra 852	Regra 853	Regra 854	Regra 855	Regra 976	Regra 977	Regra 978	Regra 979	Regra 980
Regra 856	Regra 857	Regra 858	Regra 859	Regra 860	Regra 981	Regra 982	Regra 983	Regra 984	Regra 985
Regra 861	Regra 862	Regra 863	Regra 864	Regra 865	Regra 986	Regra 987	Regra 988	Regra 989	Regra 990
Regra 866	Regra 867	Regra 868	Regra 869	Regra 870	Regra 991	Regra 992	Regra 993	Regra 994	Regra 995
Regra 871	Regra 872	Regra 873	Regra 874	Regra 875	Regra 996	Regra 997	Regra 998	Regra 999	Regra 1000

## APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação)

Regra 1001 	Regra 1002 	Regra 1003 	Regra 1004 	Regra 1005 	Regra 1126 	Regra 1127 	Regra 1128 	Regra 1129 	Regra 1130 
Regra 1006 	Regra 1007 	Regra 1008 	Regra 1009 	Regra 1010 	Regra 1131 	Regra 1132 	Regra 1133 	Regra 1134 	Regra 1135 
Regra 1011 	Regra 1012 	Regra 1013 	Regra 1014 	Regra 1015 	Regra 1136 	Regra 1137 	Regra 1138 	Regra 1139 	Regra 1140 
Regra 1016 	Regra 1017 	Regra 1018 	Regra 1019 	Regra 1020 	Regra 1141 	Regra 1142 	Regra 1143 	Regra 1144 	Regra 1145 
Regra 1021 	Regra 1022 	Regra 1023 	Regra 1024 	Regra 1025 	Regra 1146 	Regra 1147 	Regra 1148 	Regra 1149 	Regra 1150 
Regra 1026 	Regra 1027 	Regra 1028 	Regra 1029 	Regra 1030 	Regra 1151 	Regra 1152 	Regra 1153 	Regra 1154 	Regra 1155 
Regra 1031 	Regra 1032 	Regra 1033 	Regra 1034 	Regra 1035 	Regra 1156 	Regra 1157 	Regra 1158 	Regra 1159 	Regra 1160 
Regra 1036 	Regra 1037 	Regra 1038 	Regra 1039 	Regra 1040 	Regra 1161 	Regra 1162 	Regra 1163 	Regra 1164 	Regra 1165 
Regra 1041 	Regra 1042 	Regra 1043 	Regra 1044 	Regra 1045 	Regra 1166 	Regra 1167 	Regra 1168 	Regra 1169 	Regra 1170 
Regra 1046 	Regra 1047 	Regra 1048 	Regra 1049 	Regra 1050 	Regra 1171 	Regra 1172 	Regra 1173 	Regra 1174 	Regra 1175 
Regra 1051 	Regra 1052 	Regra 1053 	Regra 1054 	Regra 1055 	Regra 1176 	Regra 1177 	Regra 1178 	Regra 1179 	Regra 1180 
Regra 1056 	Regra 1057 	Regra 1058 	Regra 1059 	Regra 1060 	Regra 1181 	Regra 1182 	Regra 1183 	Regra 1184 	Regra 1185 
Regra 1061 	Regra 1062 	Regra 1063 	Regra 1064 	Regra 1065 	Regra 1186 	Regra 1187 	Regra 1188 	Regra 1189 	Regra 1190 
Regra 1066 	Regra 1067 	Regra 1068 	Regra 1069 	Regra 1070 	Regra 1191 	Regra 1192 	Regra 1193 	Regra 1194 	Regra 1195 
Regra 1071 	Regra 1072 	Regra 1073 	Regra 1074 	Regra 1075 	Regra 1196 	Regra 1197 	Regra 1198 	Regra 1199 	Regra 1200 
Regra 1076 	Regra 1077 	Regra 1078 	Regra 1079 	Regra 1080 	Regra 1201 	Regra 1202 	Regra 1203 	Regra 1204 	Regra 1205 
Regra 1081 	Regra 1082 	Regra 1083 	Regra 1084 	Regra 1085 	Regra 1206 	Regra 1207 	Regra 1208 	Regra 1209 	Regra 1210 
Regra 1086 	Regra 1087 	Regra 1088 	Regra 1089 	Regra 1090 	Regra 1211 	Regra 1212 	Regra 1213 	Regra 1214 	Regra 1215 
Regra 1091 	Regra 1092 	Regra 1093 	Regra 1094 	Regra 1095 	Regra 1216 	Regra 1217 	Regra 1218 	Regra 1219 	Regra 1220 
Regra 1096 	Regra 1097 	Regra 1098 	Regra 1099 	Regra 1100 	Regra 1221 	Regra 1222 	Regra 1223 	Regra 1224 	Regra 1225 
Regra 1101 	Regra 1102 	Regra 1103 	Regra 1104 	Regra 1105 	Regra 1226 	Regra 1227 	Regra 1228 	Regra 1229 	Regra 1230 
Regra 1106 	Regra 1107 	Regra 1108 	Regra 1109 	Regra 1110 	Regra 1231 	Regra 1232 	Regra 1233 	Regra 1234 	Regra 1235 
Regra 1111 	Regra 1112 	Regra 1113 	Regra 1114 	Regra 1115 	Regra 1236 	Regra 1237 	Regra 1238 	Regra 1239 	Regra 1240 
Regra 1116 	Regra 1117 	Regra 1118 	Regra 1119 	Regra 1120 	Regra 1241 	Regra 1242 	Regra 1243 	Regra 1244 	Regra 1245 
Regra 1121 	Regra 1122 	Regra 1123 	Regra 1124 	Regra 1125 	Regra 1246 	Regra 1247 	Regra 1248 	Regra 1249 	Regra 1250 

# APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação)

Regra 1251	Regra 1252	Regra 1253	Regra 1254	Regra 1255	Regra 1376	Regra 1377	Regra 1378	Regra 1379	Regra 1380
Regra 1256	Regra 1257	Regra 1258	Regra 1259	Regra 1260	Regra 1381	Regra 1382	Regra 1383	Regra 1384	Regra 1385
Regra 1261	Regra 1262	Regra 1263	Regra 1264	Regra 1265	Regra 1386	Regra 1387	Regra 1388	Regra 1389	Regra 1390
Regra 1266	Regra 1267	Regra 1268	Regra 1269	Regra 1270	Regra 1391	Regra 1392	Regra 1393	Regra 1394	Regra 1395
Regra 1271	Regra 1272	Regra 1273	Regra 1274	Regra 1275	Regra 1396	Regra 1397	Regra 1398	Regra 1399	Regra 1400
Regra 1276	Regra 1277	Regra 1278	Regra 1279	Regra 1280	Regra 1401	Regra 1402	Regra 1403	Regra 1404	Regra 1405
Regra 1281	Regra 1282	Regra 1283	Regra 1284	Regra 1285	Regra 1406	Regra 1407	Regra 1408	Regra 1409	Regra 1410
Regra 1286	Regra 1287	Regra 1288	Regra 1289	Regra 1290	Regra 1411	Regra 1412	Regra 1413	Regra 1414	Regra 1415
Regra 1291	Regra 1292	Regra 1293	Regra 1294	Regra 1295	Regra 1416	Regra 1417	Regra 1418	Regra 1419	Regra 1420
Regra 1296	Regra 1297	Regra 1298	Regra 1299	Regra 1300	Regra 1421	Regra 1422	Regra 1423	Regra 1424	Regra 1425
Regra 1301	Regra 1302	Regra 1303	Regra 1304	Regra 1305	Regra 1426	Regra 1427	Regra 1428	Regra 1429	Regra 1430
Regra 1306	Regra 1307	Regra 1308	Regra 1309	Regra 1310	Regra 1431	Regra 1432	Regra 1433	Regra 1434	Regra 1435
Regra 1311	Regra 1312	Regra 1313	Regra 1314	Regra 1315	Regra 1436	Regra 1437	Regra 1438	Regra 1439	Regra 1440
Regra 1316	Regra 1317	Regra 1318	Regra 1319	Regra 1320	Regra 1441	Regra 1442	Regra 1443	Regra 1444	Regra 1445
Regra 1321	Regra 1322	Regra 1323	Regra 1324	Regra 1325	Regra 1446	Regra 1447	Regra 1448	Regra 1449	Regra 1450
Regra 1326	Regra 1327	Regra 1328	Regra 1329	Regra 1330	Regra 1451	Regra 1452	Regra 1453	Regra 1454	Regra 1455
Regra 1331	Regra 1332	Regra 1333	Regra 1334	Regra 1335	Regra 1456	Regra 1457	Regra 1458	Regra 1459	Regra 1460
Regra 1336	Regra 1337	Regra 1338	Regra 1339	Regra 1340	Regra 1461	Regra 1462	Regra 1463	Regra 1464	Regra 1465
Regra 1341	Regra 1342	Regra 1343	Regra 1344	Regra 1345	Regra 1466	Regra 1467	Regra 1468	Regra 1469	Regra 1470
Regra 1346	Regra 1347	Regra 1348	Regra 1349	Regra 1350	Regra 1471	Regra 1472	Regra 1473	Regra 1474	Regra 1475
Regra 1351	Regra 1352	Regra 1353	Regra 1354	Regra 1355	Regra 1476	Regra 1477	Regra 1478	Regra 1479	Regra 1480
Regra 1356	Regra 1357	Regra 1358	Regra 1359	Regra 1360	Regra 1481	Regra 1482	Regra 1483	Regra 1484	Regra 1485
Regra 1361	Regra 1362	Regra 1363	Regra 1364	Regra 1365	Regra 1486	Regra 1487	Regra 1488	Regra 1489	Regra 1490
Regra 1366	Regra 1367	Regra 1368	Regra 1369	Regra 1370	Regra 1491	Regra 1492	Regra 1493	Regra 1494	Regra 1495
Regra 1371	Regra 1372	Regra 1373	Regra 1374	Regra 1375	Regra 1496	Regra 1497	Regra 1498	Regra 1499	Regra 1500

## APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação)

Regra 1501 	Regra 1502 	Regra 1503 	Regra 1504 	Regra 1505 	Regra 1626 	Regra 1627 	Regra 1628 	Regra 1629 	Regra 1630 
Regra 1506 	Regra 1507 	Regra 1508 	Regra 1509 	Regra 1510 	Regra 1631 	Regra 1632 	Regra 1633 	Regra 1634 	Regra 1635 
Regra 1511 	Regra 1512 	Regra 1513 	Regra 1514 	Regra 1515 	Regra 1636 	Regra 1637 	Regra 1638 	Regra 1639 	Regra 1640 
Regra 1516 	Regra 1517 	Regra 1518 	Regra 1519 	Regra 1520 	Regra 1641 	Regra 1642 	Regra 1643 	Regra 1644 	Regra 1645 
Regra 1521 	Regra 1522 	Regra 1523 	Regra 1524 	Regra 1525 	Regra 1646 	Regra 1647 	Regra 1648 	Regra 1649 	Regra 1650 
Regra 1526 	Regra 1527 	Regra 1528 	Regra 1529 	Regra 1530 	Regra 1651 	Regra 1652 	Regra 1653 	Regra 1654 	Regra 1655 
Regra 1531 	Regra 1532 	Regra 1533 	Regra 1534 	Regra 1535 	Regra 1656 	Regra 1657 	Regra 1658 	Regra 1659 	Regra 1660 
Regra 1536 	Regra 1537 	Regra 1538 	Regra 1539 	Regra 1540 	Regra 1661 	Regra 1662 	Regra 1663 	Regra 1664 	Regra 1665 
Regra 1541 	Regra 1542 	Regra 1543 	Regra 1544 	Regra 1545 	Regra 1666 	Regra 1667 	Regra 1668 	Regra 1669 	Regra 1670 
Regra 1546 	Regra 1547 	Regra 1548 	Regra 1549 	Regra 1550 	Regra 1671 	Regra 1672 	Regra 1673 	Regra 1674 	Regra 1675 
Regra 1551 	Regra 1552 	Regra 1553 	Regra 1554 	Regra 1555 	Regra 1676 	Regra 1677 	Regra 1678 	Regra 1679 	Regra 1680 
Regra 1556 	Regra 1557 	Regra 1558 	Regra 1559 	Regra 1560 	Regra 1681 	Regra 1682 	Regra 1683 	Regra 1684 	Regra 1685 
Regra 1561 	Regra 1562 	Regra 1563 	Regra 1564 	Regra 1565 	Regra 1686 	Regra 1687 	Regra 1688 	Regra 1689 	Regra 1690 
Regra 1566 	Regra 1567 	Regra 1568 	Regra 1569 	Regra 1570 	Regra 1691 	Regra 1692 	Regra 1693 	Regra 1694 	Regra 1695 
Regra 1571 	Regra 1572 	Regra 1573 	Regra 1574 	Regra 1575 	Regra 1696 	Regra 1697 	Regra 1698 	Regra 1699 	Regra 1700 
Regra 1576 	Regra 1577 	Regra 1578 	Regra 1579 	Regra 1580 	Regra 1701 	Regra 1702 	Regra 1703 	Regra 1704 	Regra 1705 
Regra 1581 	Regra 1582 	Regra 1583 	Regra 1584 	Regra 1585 	Regra 1706 	Regra 1707 	Regra 1708 	Regra 1709 	Regra 1710 
Regra 1586 	Regra 1587 	Regra 1588 	Regra 1589 	Regra 1590 	Regra 1711 	Regra 1712 	Regra 1713 	Regra 1714 	Regra 1715 
Regra 1591 	Regra 1592 	Regra 1593 	Regra 1594 	Regra 1595 	Regra 1716 	Regra 1717 	Regra 1718 	Regra 1719 	Regra 1720 
Regra 1596 	Regra 1597 	Regra 1598 	Regra 1599 	Regra 1600 	Regra 1721 	Regra 1722 	Regra 1723 	Regra 1724 	Regra 1725 
Regra 1601 	Regra 1602 	Regra 1603 	Regra 1604 	Regra 1605 	Regra 1726 	Regra 1727 	Regra 1728 	Regra 1729 	Regra 1730 
Regra 1606 	Regra 1607 	Regra 1608 	Regra 1609 	Regra 1610 	Regra 1731 	Regra 1732 	Regra 1733 	Regra 1734 	Regra 1735 
Regra 1611 	Regra 1612 	Regra 1613 	Regra 1614 	Regra 1615 	Regra 1736 				

## APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação)

Regra 1751 	Regra 1752 	Regra 1753 	Regra 1754 	Regra 1755 	Regra 1876 	Regra 1877 	Regra 1878 	Regra 1879 	Regra 1880 
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

## APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação)

Regra 2001	Regra 2002	Regra 2003	Regra 2004	Regra 2005	Regra 2126	Regra 2127	Regra 2128	Regra 2129	Regra 2130
Regra 2006	Regra 2007	Regra 2008	Regra 2009	Regra 2010	Regra 2131	Regra 2132	Regra 2133	Regra 2134	Regra 2135
Regra 2011	Regra 2012	Regra 2013	Regra 2014	Regra 2015	Regra 2136	Regra 2137	Regra 2138	Regra 2139	Regra 2140
Regra 2016	Regra 2017	Regra 2018	Regra 2019	Regra 2020	Regra 2141	Regra 2142	Regra 2143	Regra 2144	Regra 2145
Regra 2021	Regra 2022	Regra 2023	Regra 2024	Regra 2025	Regra 2146	Regra 2147	Regra 2148	Regra 2149	Regra 2150
Regra 2026	Regra 2027	Regra 2028	Regra 2029	Regra 2030	Regra 2151	Regra 2152	Regra 2153	Regra 2154	Regra 2155
Regra 2031	Regra 2032	Regra 2033	Regra 2034	Regra 2035	Regra 2156	Regra 2157	Regra 2158	Regra 2159	Regra 2160
Regra 2036	Regra 2037	Regra 2038	Regra 2039	Regra 2040	Regra 2161	Regra 2162	Regra 2163	Regra 2164	Regra 2165
Regra 2041	Regra 2042	Regra 2043	Regra 2044	Regra 2045	Regra 2166	Regra 2167	Regra 2168	Regra 2169	Regra 2170
Regra 2046	Regra 2047	Regra 2048	Regra 2049	Regra 2050	Regra 2171	Regra 2172	Regra 2173	Regra 2174	Regra 2175
Regra 2051	Regra 2052	Regra 2053	Regra 2054	Regra 2055	Regra 2176	Regra 2177	Regra 2178	Regra 2179	Regra 2180
Regra 2056	Regra 2057	Regra 2058	Regra 2059	Regra 2060	Regra 2181	Regra 2182	Regra 2183	Regra 2184	Regra 2185
Regra 2061	Regra 2062	Regra 2063	Regra 2064	Regra 2065	Regra 2186	Regra 2187	Regra 2188	Regra 2189	Regra 2190
Regra 2066	Regra 2067	Regra 2068	Regra 2069	Regra 2070	Regra 2191	Regra 2192	Regra 2193	Regra 2194	Regra 2195
Regra 2071	Regra 2072	Regra 2073	Regra 2074	Regra 2075	Regra 2196	Regra 2197	Regra 2198	Regra 2199	Regra 2200
Regra 2076	Regra 2077	Regra 2078	Regra 2079	Regra 2080	Regra 2201	Regra 2202	Regra 2203	Regra 2204	Regra 2205
Regra 2081	Regra 2082	Regra 2083	Regra 2084	Regra 2085	Regra 2206	Regra 2207	Regra 2208	Regra 2209	Regra 2210
Regra 2086	Regra 2087	Regra 2088	Regra 2089	Regra 2090	Regra 2211	Regra 2212	Regra 2213	Regra 2214	Regra 2215
Regra 2091	Regra 2092	Regra 2093	Regra 2094	Regra 2095	Regra 2216	Regra 2217	Regra 2218	Regra 2219	Regra 2220
Regra 2096	Regra 2097	Regra 2098	Regra 2099	Regra 2100	Regra 2221	Regra 2222	Regra 2223	Regra 2224	Regra 2225
Regra 2101	Regra 2102	Regra 2103	Regra 2104	Regra 2105	Regra 2226	Regra 2227	Regra 2228	Regra 2229	Regra 2230
Regra 2106	Regra 2107	Regra 2108	Regra 2109	Regra 2110	Regra 2231	Regra 2232	Regra 2233	Regra 2234	Regra 2235
Regra 2111	Regra 2112	Regra 2113	Regra 2114	Regra 2115	Regra 2236	Regra 2237	Regra 2238	Regra 2239	Regra 2240
Regra 2116	Regra 2117	Regra 2118	Regra 2119	Regra 2120	Regra 2241	Regra 2242	Regra 2243	Regra 2244	Regra 2245
Regra 2121	Regra 2122	Regra 2123	Regra 2124	Regra 2125	Regra 2246	Regra 2247	Regra 2248	Regra 2249	Regra 2250

# APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação)

Regra 2251	Regra 2252	Regra 2253	Regra 2254	Regra 2255	Regra 2376	Regra 2377	Regra 2378	Regra 2379	Regra 2380
Regra 2256	Regra 2257	Regra 2258	Regra 2259	Regra 2260	Regra 2381	Regra 2382	Regra 2383	Regra 2384	Regra 2385
Regra 2261	Regra 2262	Regra 2263	Regra 2264	Regra 2265	Regra 2386	Regra 2387	Regra 2388	Regra 2389	Regra 2390
Regra 2266	Regra 2267	Regra 2268	Regra 2269	Regra 2270	Regra 2391	Regra 2392	Regra 2393	Regra 2394	Regra 2395
Regra 2271	Regra 2272	Regra 2273	Regra 2274	Regra 2275	Regra 2396	Regra 2397	Regra 2398	Regra 2399	Regra 2400
Regra 2276	Regra 2277	Regra 2278	Regra 2279	Regra 2280	Regra 2401	Regra 2402	Regra 2403	Regra 2404	Regra 2405
Regra 2281	Regra 2282	Regra 2283	Regra 2284	Regra 2285	Regra 2406	Regra 2407	Regra 2408	Regra 2409	Regra 2410
Regra 2286	Regra 2287	Regra 2288	Regra 2289	Regra 2290	Regra 2411	Regra 2412	Regra 2413	Regra 2414	Regra 2415
Regra 2291	Regra 2292	Regra 2293	Regra 2294	Regra 2295	Regra 2416	Regra 2417	Regra 2418	Regra 2419	Regra 2420
Regra 2296	Regra 2297	Regra 2298	Regra 2299	Regra 2300	Regra 2421	Regra 2422	Regra 2423	Regra 2424	Regra 2425
Regra 2301	Regra 2302	Regra 2303	Regra 2304	Regra 2305	Regra 2426	Regra 2427	Regra 2428	Regra 2429	Regra 2430
Regra 2306	Regra 2307	Regra 2308	Regra 2309	Regra 2310	Regra 2431	Regra 2432	Regra 2433	Regra 2434	Regra 2435
Regra 2311	Regra 2312	Regra 2313	Regra 2314	Regra 2315	Regra 2436	Regra 2437	Regra 2438	Regra 2439	Regra 2440
Regra 2316	Regra 2317	Regra 2318	Regra 2319	Regra 2320	Regra 2441	Regra 2442	Regra 2443	Regra 2444	Regra 2445
Regra 2321	Regra 2322	Regra 2323	Regra 2324	Regra 2325	Regra 2446	Regra 2447	Regra 2448	Regra 2449	Regra 2450
Regra 2326	Regra 2327	Regra 2328	Regra 2329	Regra 2330	Regra 2451	Regra 2452	Regra 2453	Regra 2454	Regra 2455
Regra 2331	Regra 2332	Regra 2333	Regra 2334	Regra 2335	Regra 2456	Regra 2457	Regra 2458	Regra 2459	Regra 2460
Regra 2336	Regra 2337	Regra 2338	Regra 2339	Regra 2340	Regra 2461	Regra 2462	Regra 2463	Regra 2464	Regra 2465
Regra 2341	Regra 2342	Regra 2343	Regra 2344	Regra 2345	Regra 2466	Regra 2467	Regra 2468	Regra 2469	Regra 2470
Regra 2346	Regra 2347	Regra 2348	Regra 2349	Regra 2350	Regra 2471	Regra 2472	Regra 2473	Regra 2474	Regra 2475
Regra 2351	Regra 2352	Regra 2353	Regra 2354	Regra 2355	Regra 2476	Regra 2477	Regra 2478	Regra 2479	Regra 2480
Regra 2356	Regra 2357	Regra 2358	Regra 2359	Regra 2360	Regra 2481	Regra 2482	Regra 2483	Regra 2484	Regra 2485
Regra 2361	Regra 2362	Regra 2363	Regra 2364	Regra 2365	Regra 2486	Regra 2487	Regra 2488	Regra 2489	Regra 2490
Regra 2366	Regra 2367	Regra 2368	Regra 2369	Regra 2370	Regra 2491	Regra 2492	Regra 2493	Regra 2494	Regra 2495
Regra 2371	Regra 2372	Regra 2373	Regra 2374	Regra 2375	Regra 2496	Regra 2497	Regra 2498	Regra 2499	Regra 2500

# APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação)

Regra 2501	Regra 2502	Regra 2503	Regra 2504	Regra 2505	Regra 2626	Regra 2627	Regra 2628	Regra 2629	Regra 2630
Regra 2506	Regra 2507	Regra 2508	Regra 2509	Regra 2510	Regra 2631	Regra 2632	Regra 2633	Regra 2634	Regra 2635
Regra 2511	Regra 2512	Regra 2513	Regra 2514	Regra 2515	Regra 2636	Regra 2637	Regra 2638	Regra 2639	Regra 2640
Regra 2516	Regra 2517	Regra 2518	Regra 2519	Regra 2520	Regra 2641	Regra 2642	Regra 2643	Regra 2644	Regra 2645
Regra 2521	Regra 2522	Regra 2523	Regra 2524	Regra 2525	Regra 2646	Regra 2647	Regra 2648	Regra 2649	Regra 2650
Regra 2526	Regra 2527	Regra 2528	Regra 2529	Regra 2530	Regra 2651	Regra 2652	Regra 2653	Regra 2654	Regra 2655
Regra 2531	Regra 2532	Regra 2533	Regra 2534	Regra 2535	Regra 2656	Regra 2657	Regra 2658	Regra 2659	Regra 2660
Regra 2536	Regra 2537	Regra 2538	Regra 2539	Regra 2540	Regra 2661	Regra 2662	Regra 2663	Regra 2664	Regra 2665
Regra 2541	Regra 2542	Regra 2543	Regra 2544	Regra 2545	Regra 2666	Regra 2667	Regra 2668	Regra 2669	Regra 2670
Regra 2546	Regra 2547	Regra 2548	Regra 2549	Regra 2550	Regra 2671	Regra 2672	Regra 2673	Regra 2674	Regra 2675
Regra 2551	Regra 2552	Regra 2553	Regra 2554	Regra 2555	Regra 2676	Regra 2677	Regra 2678	Regra 2679	Regra 2680
Regra 2556	Regra 2557	Regra 2558	Regra 2559	Regra 2560	Regra 2681	Regra 2682	Regra 2683	Regra 2684	Regra 2685
Regra 2561	Regra 2562	Regra 2563	Regra 2564	Regra 2565	Regra 2686	Regra 2687	Regra 2688	Regra 2689	Regra 2690
Regra 2566	Regra 2567	Regra 2568	Regra 2569	Regra 2570	Regra 2691	Regra 2692	Regra 2693	Regra 2694	Regra 2695
Regra 2571	Regra 2572	Regra 2573	Regra 2574	Regra 2575	Regra 2696	Regra 2697	Regra 2698	Regra 2699	Regra 2700
Regra 2576	Regra 2577	Regra 2578	Regra 2579	Regra 2580	Regra 2701	Regra 2702	Regra 2703	Regra 2704	Regra 2705
Regra 2581	Regra 2582	Regra 2583	Regra 2584	Regra 2585	Regra 2706	Regra 2707	Regra 2708	Regra 2709	Regra 2710
Regra 2586	Regra 2587	Regra 2588	Regra 2589	Regra 2590	Regra 2711	Regra 2712	Regra 2713	Regra 2714	Regra 2715
Regra 2591	Regra 2592	Regra 2593	Regra 2594	Regra 2595	Regra 2716	Regra 2717	Regra 2718	Regra 2719	Regra 2720
Regra 2596	Regra 2597	Regra 2598	Regra 2599	Regra 2600	Regra 2721	Regra 2722	Regra 2723	Regra 2724	Regra 2725
Regra 2601	Regra 2602	Regra 2603	Regra 2604	Regra 2605	Regra 2726	Regra 2727	Regra 2728	Regra 2729	Regra 2730
Regra 2606	Regra 2607	Regra 2608	Regra 2609	Regra 2610	Regra 2731	Regra 2732	Regra 2733	Regra 2734	Regra 2735
Regra 2611	Regra 2612	Regra 2613	Regra 2614	Regra 2615	Regra 2736	Regra 2737	Regra 2738	Regra 2739	Regra 2740
Regra 2616	Regra 2617	Regra 2618	Regra 2619	Regra 2620	Regra 2741	Regra 2742	Regra 2743	Regra 2744	Regra 2745
Regra 2621	Regra 2622	Regra 2623	Regra 2624	Regra 2625	Regra 2746	Regra 2747	Regra 2748	Regra 2749	Regra 2750

## APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação)

Regra 2751 	Regra 2752 	Regra 2753 	Regra 2754 	Regra 2755 	Regra 2876 	Regra 2877 	Regra 2878 	Regra 2879 	Regra 2880 
Regra 2756 	Regra 2757 	Regra 2758 	Regra 2759 	Regra 2760 	Regra 2881 	Regra 2882 	Regra 2883 	Regra 2884 	Regra 2885 
Regra 2761 	Regra 2762 	Regra 2763 	Regra 2764 	Regra 2765 	Regra 2886 	Regra 2887 	Regra 2888 	Regra 2889 	Regra 2890 
Regra 2766 	Regra 2767 	Regra 2768 	Regra 2769 	Regra 2770 	Regra 2891 	Regra 2892 	Regra 2893 	Regra 2894 	Regra 2895 
Regra 2771 	Regra 2772 	Regra 2773 	Regra 2774 	Regra 2775 	Regra 2896 	Regra 2897 	Regra 2898 	Regra 2899 	Regra 2900 
Regra 2776 	Regra 2777 	Regra 2778 	Regra 2779 	Regra 2780 	Regra 2901 	Regra 2902 	Regra 2903 	Regra 2904 	Regra 2905 
Regra 2781 	Regra 2782 	Regra 2783 	Regra 2784 	Regra 2785 	Regra 2906 	Regra 2907 	Regra 2908 	Regra 2909 	Regra 2910 
Regra 2786 	Regra 2787 	Regra 2788 	Regra 2789 	Regra 2790 	Regra 2911 	Regra 2912 	Regra 2913 	Regra 2914 	Regra 2915 
Regra 2791 	Regra 2792 	Regra 2793 	Regra 2794 	Regra 2795 	Regra 2916 	Regra 2917 	Regra 2918 	Regra 2919 	Regra 2920 
Regra 2796 	Regra 2797 	Regra 2798 	Regra 2799 	Regra 2800 	Regra 2921 	Regra 2922 	Regra 2923 	Regra 2924 	Regra 2925 
Regra 2801 	Regra 2802 	Regra 2803 	Regra 2804 	Regra 2805 	Regra 2926 	Regra 2927 	Regra 2928 	Regra 2929 	Regra 2930 
Regra 2806 	Regra 2807 	Regra 2808 	Regra 2809 	Regra 2810 	Regra 2931 	Regra 2932 	Regra 2933 	Regra 2934 	Regra 2935 
Regra 2811 	Regra 2812 	Regra 2813 	Regra 2814 	Regra 2815 	Regra 2936 	Regra 2937 	Regra 2938 	Regra 2939 	Regra 2940 
Regra 2816 	Regra 2817 	Regra 2818 	Regra 2819 	Regra 2820 	Regra 2941 	Regra 2942 	Regra 2943 	Regra 2944 	Regra 2945 
Regra 2821 	Regra 2822 	Regra 2823 	Regra 2824 	Regra 2825 	Regra 2946 	Regra 2947 	Regra 2948 	Regra 2949 	Regra 2950 
Regra 2826 	Regra 2827 	Regra 2828 	Regra 2829 	Regra 2830 	Regra 2951 	Regra 2952 	Regra 2953 	Regra 2954 	Regra 2955 
Regra 2831 	Regra 2832 	Regra 2833 	Regra 2834 	Regra 2835 	Regra 2956 	Regra 2957 	Regra 2958 	Regra 2959 	Regra 2960 
Regra 2836 	Regra 2837 	Regra 2838 	Regra 2839 	Regra 2840 	Regra 2961 	Regra 2962 	Regra 2963 	Regra 2964 	Regra 2965 
Regra 2841 	Regra 2842 	Regra 2843 	Regra 2844 	Regra 2845 	Regra 2966 	Regra 2967 	Regra 2968 	Regra 2969 	Regra 2970 
Regra 2846 	Regra 2847 	Regra 2848 	Regra 2849 	Regra 2850 	Regra 2971 	Regra 2972 	Regra 2973 	Regra 2974 	Regra 2975 
Regra 2851 	Regra 2852 	Regra 2853 	Regra 2854 	Regra 2855 	Regra 2976 	Regra 2977 	Regra 2978 	Regra 2979 	Regra 2980 
Regra 2856 	Regra 2857 	Regra 2858 	Regra 2859 	Regra 2860 	Regra 2981 	Regra 2982 	Regra 2983 	Regra 2984 	Regra 2985 
Regra 2861 	Regra 2862 	Regra 2863 	Regra 2864 	Regra 2865 	Regra 2986 	Regra 2987 	Regra 2988 	Regra 2989 	Regra 2990 
Regra 2866 	Regra 2867 	Regra 2868 	Regra 2869 	Regra 2870 	Regra 2991 	Regra 2992 	Regra 2993 	Regra 2994 	Regra 2995 
Regra 2871 	Regra 2872 	Regra 2873 	Regra 2874 	Regra 2875 	Regra 2996 	Regra 2997 	Regra 2998 	Regra 2999 	Regra 3000 

APÊNDICE B – REGRAS DE TRANSIÇÃO (continuação)

Regra 3001 	Regra 3002 	Regra 3003 	Regra 3004 	Regra 3005 
Regra 3006 	Regra 3007 	Regra 3008 	Regra 3009 	Regra 3010 
Regra 3011 	Regra 3012 	Regra 3013 	Regra 3014 	Regra 3015 
Regra 3016 	Regra 3017 	Regra 3018 	Regra 3019 	Regra 3020 
Regra 3021 	Regra 3022 	Regra 3023 	Regra 3024 	Regra 3025 
Regra 3026 	Regra 3027 	Regra 3028 	Regra 3029 	Regra 3030 
Regra 3031 	Regra 3032 	Regra 3033 	Regra 3034 	Regra 3035 
Regra 3036 	Regra 3037 	Regra 3038 	Regra 3039 	Regra 3040 
Regra 3041 	Regra 3042 	Regra 3043 	Regra 3044 	Regra 3045 
Regra 3046 	Regra 3047 	Regra 3048 	Regra 3049 	Regra 3050 
Regra 3051 	Regra 3052 	Regra 3053 	Regra 3054 	Regra 3055 
Regra 3056 	Regra 3057 	Regra 3058 	Regra 3059 	Regra 3060 
Regra 3061 	Regra 3062 	Regra 3063 	Regra 3064 	Regra 3065 
Regra 3066 	Regra 3067 	Regra 3068 	Regra 3069 	Regra 3070 
Regra 3071 	Regra 3072 	Regra 3073 	Regra 3074 	Regra 3075 
Regra 3076 	Regra 3077 	Regra 3078 	Regra 3079 	Regra 3080 
Regra 3081 	Regra 3082 	Regra 3083 	Regra 3084 	Regra 3085 
Regra 3086 	Regra 3087 	Regra 3088 	Regra 3089 	Regra 3090 
Regra 3091 	Regra 3092 	Regra 3093 	Regra 3094 	Regra 3095 
Regra 3096 	Regra 3097 	Regra 3098 	Regra 3099 	Regra 3100 
Regra 3101 	Regra 3102 	Regra 3103 	Regra 3104 	Regra 3105 
Regra 3106 	Regra 3107 	Regra 3108 	Regra 3109 	Regra 3110 
Regra 3111 	Regra 3112 	Regra 3113 	Regra 3114 	Regra 3115 
Regra 3116 	Regra 3117 	Regra 3118 	Regra 3119 	Regra 3120 
Regra 3121 	Regra 3122 	Regra 3123 	Regra 3124 	Regra 3125 

## APÊNDICE C – LIMITE ADIABÁTICO

```

public void percorrerAutomato() {
    for (int lin=0; lin<alturaAc; lin++) {
        for (int col=0; col<larguraAc; col++) {
            // se estados inertes, continua inerte (rocha, água ou cinza)
            if ((acAtual[lin][col]==1)|| (acAtual[lin][col]==2)|| (acAtual[lin][col]==4)) {
                acCentral = acAtual[lin][col];
            }
            else { // canto superior esquerdo
                if ((lin==0)&&(col==0)) {
                    acCentral = acAtual[lin][col] ;
                    vizsup = acCentral ;
                    vizinf = acAtual[lin+1][col];
                    vizesq = acCentral;
                    vizdir = acAtual[lin][col+1] ;
                }
                else { // canto superior direito
                    if ((lin==0)&&(col==larguraAc-1)) {
                        acCentral = acAtual[lin][col] ;
                        vizsup = acCentral ;
                        vizinf = acAtual[lin+1][col];
                        vizesq = acAtual[lin][col-1];
                        vizdir = acCentral;
                    }
                }
                else { // canto inferior esquerdo
                    if ((lin==alturaAc-1)&&(col==0)) {
                        acCentral = acAtual[lin][col];
                        vizsup = acAtual[lin-1][col];
                        vizinf = acCentral;
                        vizesq = acCentral;
                        vizdir = acAtual[lin][col+1];
                    }
                }
                else { //canto inferior direito
                    if ((lin==alturaAc-1)&&(col==larguraAc-1)) {
                        acCentral = acAtual[lin][col];
                        vizsup = acAtual[lin-1][col] ;
                        vizinf = acCentral;
                        vizesq = acAtual[lin][col-1];
                        vizdir = acCentral;
                    }
                }
                else { // lado superior
                    if ((lin==0)&&(col>0)&&(col<(larguraAc-1))) {
                        acCentral = acAtual[lin][col] ;
                        vizsup = acCentral;
                        vizinf = acAtual[lin+1][col];
                        vizesq = acAtual[lin][col-1];
                        vizdir = acAtual[lin][col+1];
                    }
                }
                else { //lado inferior
                    if ((lin==alturaAc-1)&&(col>0)&&(col<(larguraAc-1))) {
                        acCentral = acAtual[lin][col] ;
                        vizsup = acAtual[lin-1][col];
                        vizinf = acCentral;
                        vizesq = acAtual[lin][col-1];
                        vizdir = acAtual[lin][col+1];
                    }
                }
                else { //lado esquerdo
                    if ((lin>0)&&(lin<alturaAc-1)&&(col==0)) {
                        acCentral = acAtual[lin][col] ;
                        vizsup = acAtual[lin-1][col];
                        vizinf = acAtual[lin+1][col];
                        vizesq = acCentral;
                        vizdir = acAtual[lin][col+1];
                    }
                }
                else { //lado direito
                    if ((lin>0)&&(lin<(alturaAc-1)&&(col==larguraAc-1))) {
                        acCentral = acAtual[lin][col] ;
                        vizsup = acAtual[lin-1][col];
                        vizinf = acAtual[lin+1][col];
                        vizesq = acAtual[lin][col-1];
                        vizdir = acCentral;
                    }
                }
                else {
                    acCentral = acAtual[lin][col] ;
                    vizsup = acAtual[lin-1][col];
                    vizinf = acAtual[lin+1][col];
                    vizesq = acAtual[lin][col-1];
                    vizdir = acAtual[lin][col+1];
                }
            }
        }
    }
}

```