



CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

Recredenciado pela Portaria Ministerial nº 1.162, de 13/10/16, D.O.U nº 198, de 14/10/2016
ASSOCIAÇÃO EDUCACIONAL LUTERANA DO BRASIL

Adrianne Castro Pereira

Jairo Painkow Junior

Lucas da Silva Vieira

Nalberthy Sousa da Silva

Automatos Celulares: Dimensão, Formato Celular, Estado de uma Célula, Tamanho da
Vizinhança, Regras de Transição.

Palmas – TO
Março 15, 2019

SUMÁRIO

Automatos Celulares: Dimensão, Formato Celular, Estado de uma Célula, Tamanho da Vizinhança, Regras de Transição.	0
INTRODUÇÃO	2
REFERENCIAL TEÓRICO	3
Definição Informal	3
Dimensão	3
Formato da Célula	4
Estados de uma célula	4
Tamanho da Vizinhança	4
Regras de Transição Totalística	5
DESENVOLVIMENTO	6
Definição de Configurações globais:	6
Representações de células e do tabuleiro:	7
Funções de Iteratividade	7
Definição de progressão das gerações de células	9
CONCLUSÃO OU CONSIDERAÇÕES FINAIS	12
REFERÊNCIAS	13

INTRODUÇÃO

Os autômatos celulares são ferramentas simples e poderosas para representar sistemas físicos compostos por elementos discretos com interações locais. Vários sistemas se enquadram nessa definição, tais como o processo de surgimento de um cristal de gelo, em que cada molécula traz suas próprias informações locais e agrupam-se baseadas nas construções das moléculas vizinhas (REITE, 2005). A representação de sistemas evolutivos vem a partir de uma configuração inicial aleatória, cada componente do sistema tem sua evolução baseada na situação atual de seus vizinhos e num conjunto de regras que são iguais para todos os componentes. Embora as regras sejam as mesmas para todos os componentes do sistema, a situação dos componentes vizinhos pode variar indefinida e complexamente durante o tempo, podendo originar novos sistemas e chegando até a sua auto- reprodução.

De acordo com a história reescrita por Burks, Von Neumann estava interessado em nas conexões entre biologia e a ciência dos dispositivos computacionais chamada de Teoria dos Autômatos. Nos seus estudos, predominava a ideia do fenômeno biológico de auto-reprodução, que consiste ne um tipo de organização lógica suficiente para um autônomo controlar a si próprio de tal maneira que pudesse também se reproduzir.

REFERENCIAL TEÓRICO

DEFINIÇÃO INFORMAL

Segundo (weimar, 1996) um autômato celular é caracterizado pelas propriedades fundamentais:

- Consistem em um matriz ou grade de células;
- A evolução é caracterizada por um estado pertencente a um conjunto finito de estados;
- Cada célula evolui de acordo com as mesmas regras que dependem somente do estado em que a célula se encontra e de um número finito de vizinhos;
- A relação com a vizinhança é local e uniforme.

Um exemplo simples a ser considerado é o modelo Greenber-Hasting, um autômato celular que modela uma excitação média. Uma situação exemplo de excitação média são os tecidos nervosos ou musculares (coração), que podem estar três diferentes estados: descansado, excitado ou em recuperação.

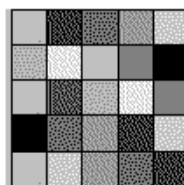
DIMENSÃO

Um Autômato Celular pode ser de uma dimensão: a rede de células é formada por uma única linha, as células possuem vizinhas à esquerda e a direita (Figura 1: 1D). Duas dimensões: as células ficam uma do lado da outra, possuem vizinhas a direita, esquerda, superior, inferior e nas diagonais (Figura 1: 2D). Três dimensões: mesmas características da de suas dimensões, porém, são distribuídas tridimensionalmente e podem possui vizinhos na frente e atrás (Figura 1: 3D) (GREMONINI; VICENTINI, 2008).



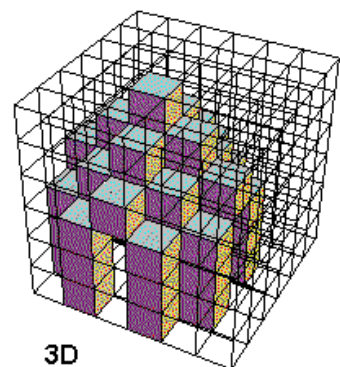
1D

Figura 1



2D

Figura 2

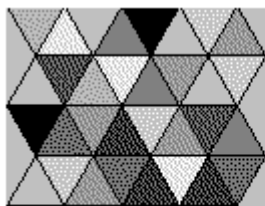


3D

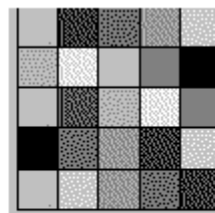
Figura 3

FORMATO DA CÉLULA

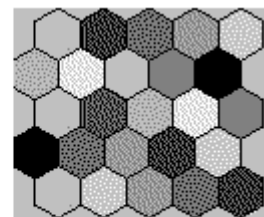
As células de um Autômato Celular podem possuir o formato triangular, quadrangular ou hexagonal. Embora possa possuir diversas formas, em um sistema Autômato Celular, a rede de células de cada sistema são todas da mesma forma, não tendo dois tipos de formato em um mesmo sistema (GREMONINI; VICENTINI, 2008).



Triangular



Quadrangular



Hexagonal

ESTADOS DE UMA CÉLULA

As células de um Autômato Celular podem possuir uma quantidade finita de estados, sendo seu estado alterado de acordo com as regras estabelecidas. Caso todas as células estejam em estado inicial, uma regra pode definir um estado especial para uma célula desencadear todo o processo de evolução, esse estado é chamado de estado inativo (PÁDUA; VIEIRA, 2004). Uma célula pode possuir qualquer valor, normalmente é atribuído um valor entre zero e um (GREMONINI; VICENTINI, 2008).

TAMANHO DA VIZINHANÇA

A definição de um autômato celular depende assim da fixação de uma vizinhança, e duas opções clássicas são as chamadas vizinhanças de von Neumann e de Moore. A vizinhança de von Neumann consiste em considerar, para cada célula, as células imediatamente acima, abaixo, à esquerda e à direita, podendo-se estender o diâmetro da vizinhança para duas células imediatas ou mais; na vizinhança de Moore, as diagonais também são consideradas. Ambas as vizinhanças são ilustradas na Figura 1.

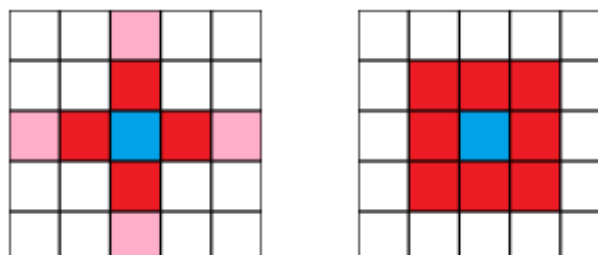


Figura 1 – Vizinhança de von Neumann e vizinhança de Moore para a definição da função de transição de um autômato celular. Em azul, a célula sendo atualizada. Em vermelho, as células adjacentes cujos valores são considerados para a atualização da célula. Em rosa, um exemplo de vizinhança estendida, na qual considera-se as células azul, vermelho e rosa para a atualização. Fonte: https://en.wikipedia.org/wiki/Cellular_automaton (figuras liberadas para uso irrestrito sob licença Creative Commons CC0 1.0 Universal).

REGRAS DE TRANSIÇÃO TOTALÍSTICA

Algumas vezes não importa exatamente qual o estado de cada vizinho de uma determinada célula, mas a quantidade total de células vizinhas que estão num determinado estado. Por exemplo, no *applet* do jogo da vida, como foi dito, para uma célula manter-se viva é necessário que a quantidade de células vivas ao redor seja dois ou três. Esse tipo de representação das regras é chamado de totalístico. No caso de ser considerado também ser considerado o estado atual da célula a ser modificada, essa representação é chamada de totalística exterior.

DESENVOLVIMENTO

Código com configurações pré definidas para lógica de execução do autômato.

Definição de configuração para o código:

Dimensão: 2D

Formato da célula: Quadrangular

Tamanho: Moore R:1

Regra de Transição: Totalística

Para o desenvolvimento do código utilizamos, Linguagem html5, CSS e JavaScript. Para o Desenvolvimento da parte visual, utilizamos o Canvas ferramenta utilizada via javascript para desenhar gráficos, caminhos, caixas, círculos, texto e adicionar imagens.

Vamos dividir o código em etapas:

1. definir configurações globais
2. criar representações de células e do tabuleiro
3. adicionar interatividade ao tabuleiro
4. definir progressão das gerações de células

DEFINIÇÃO DE CONFIGURAÇÕES GLOBAIS:

Definindo canvas :

```
var canvas = document.getElementById('canvas');  
var ctx = canvas.getContext('2d');
```

Vetor vazio para armazenamento de células:

```
var celula = [];
```

Definindo quantidade de células por fileira e coluna

(window.innerWidth) retorna dimensão da tela: **altura e largura**

```
var tx = Math.floor(window.innerWidth/20);  
var ty = Math.floor(window.innerHeight/20);
```

Criando Número de pixels | células:

```
var cx = canvas.width = tx * 20;  
var cy = canvas.height = ty * 20;
```

Estado de função de chamada (Botão):

```
var estado = null;
```

Contador de células:

```
var QtCelula = 0;
```

REPRESENTAÇÕES DE CÉLULAS E DO TABULEIRO:

Utiliza orientação a objetos para atribuir vida ao tabuleiro

```
for(var i = 0; i <= ty; i++){percorrendo eixo y e criando uma lista
    celula[i] = new Array();
    for(var j = 0; j <= tx; j++){percorrendo eixo x e criando uma lista
        var vida = {
            viva: false,
            vivaNaProximaGeracao: false
        };
        celula[i][j] = vida; Atribuindo vida a celula do tabuleiro
    }
}
```

FUNÇÕES DE ITERATIVIDADE

Função permite que defina célula morta ou viva na parte de execução do código com ação click.

```
function MouseClick(e){
    var ax = Math.floor(e.clientX/20);
    var ay = Math.floor(e.clientY/20);
    if(celula[ay][ax].viva === false){
        celula[ay][ax].viva = true;
        ctx.fillRect(ax*20,ay*20,14,14);
    }
    else{
        celula[ay][ax].viva = false;
    }
}
```



```
        ctx.clearRect(ax*20,ay*20,14,14);
    }
}
```

```
function Desenhar(){
    ctx.clearRect(0,0,cx,cy);
    for(var i = 0; i < tx; i++){
        for(var j = 0; j < ty; j++){
            if(celula[j][i].viva === true){
                QtCelula++;
                ctx.fillRect(i*20,j*20,14,14);
            }
            celula[j][i].viva = celula[j][i].vivaNaProximaGeracao;
            celula[j][i].vivaNaProximaGeracao = false;
        }
    }
}
```

```
function Iniciar(){
    estado = setInterval(function(){
        Atualizar();
        Desenhar();
    },1000); Velocidade de atualização em milisegundos
}
```

```
function Parar(){
    clearInterval(estado);
}
```

```
function Limpar(){
    Parar()
    ctx.clearRect(0,0,cx,cy);
    for(var i = 0; i <= tx; i++){ percorre até o final do eixo X
```

```

        for(var j = 0; j<= ty; j++){percorre até o final do eixo Y
            celula[j][i].viva = false; Atribui status de celula morta
        }
    }

```

DEFINIÇÃO DE PROGRESSÃO DAS GERAÇÕES DE CÉLULAS

```

function ProximaGeracao(y,x){
    var count = 0;

    Abaixo verifica- se a quantidade de vizinhos

    if(celula[y-1][x-1].viva === true){count++;}
    if(celula[y-1][x].viva === true){count++;}
    if(celula[y-1][x+1].viva === true){count++;}
    if(celula[y][x-1].viva === true){count++;}
    if(celula[y][x+1].viva === true){count++;}
    if(celula[y+1][x-1].viva === true){count++;}
    if(celula[y+1][x].viva === true){count++;}
    if(celula[y+1][x+1].viva === true){count++;}
    if(celula[y][x].viva === true){

        Define regra de visinhos

        if(count == 1 || count == 2){
            return true;
        }
    }
    if(celula[y][x].viva === false)
    {
        if(count === 2){

```

```
        return true;
    }
}
return false;
}
```

Atualiza para proxima vida

```
function Atualizar(){
    for(var i = 1; i <= tx-1; i++){
        for(var j = 1; j <= ty-1; j++){
            celula[j][i].vivaNaProximaGeracao = ProximaGeracao(j,i);
        }
    }
}
```

Após execução do código as Definições globais são definidas, e é feito a etapa de Representações de células e do tabuleiro. Na parte visível da execução temos uma grade de células mortas esperando ação do usuário para dar vida.

A atribuição de vida a uma célula é dada pela função de “clique” definida na etapa de função de iteratividade. Após a marcação do padrão escolhido, precisamos iniciar a auto reprodução através da função “iniciar”, onde iniciar possui uma sub-chamada de função dentro de um método. Na sub-chamada é feito a chamada das funções “desenhar” e “atualizar”. O resultado dessa ação é a auto-reprodução de gerações consecutivas. A função de parada encerra o evento do método. Na função de limpar é feito uma mudança de status de células (vivo para morto). Para atualizar para próxima geração, na função próxima geração é feita uma sub-chamada das funções atualizar e desenhar.

A função atualizar consiste na criação de uma nova geração, após criada ela substitui a geração anterior (uma nova geração é atribuída a Matriz). Já após a chamada da função desenhar é feito a atualização da geração anterior na tela para uma nova geração.

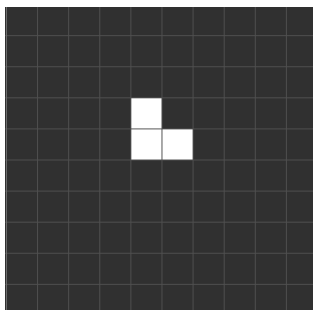
Regra totalística usada no código:

- Se uma célula está OFF, ela fica ON se exatamente 2 dos seus vizinhos estão ON;
- Se uma célula está ON, ela continua ON se exatamente 2 ou 1 de seus vizinhos estão ON;
- Caso contrário, ela fica OFF.

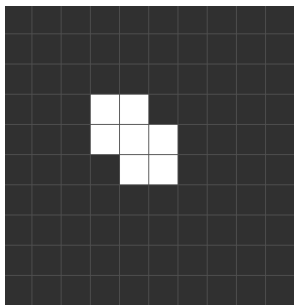
Exemplo:

Dada regra acima e a definição de entrada abaixo: Obteremos nas 5 próximas gerações os seguintes resultados.

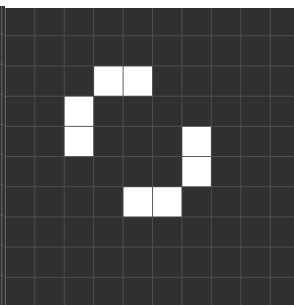
Entrada:



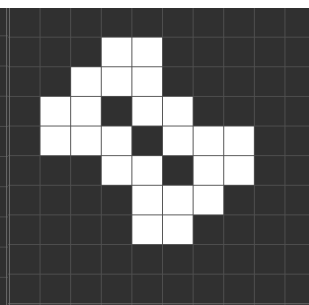
Geração 1:



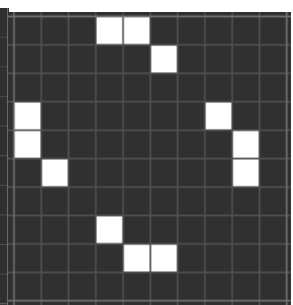
Geração 2:



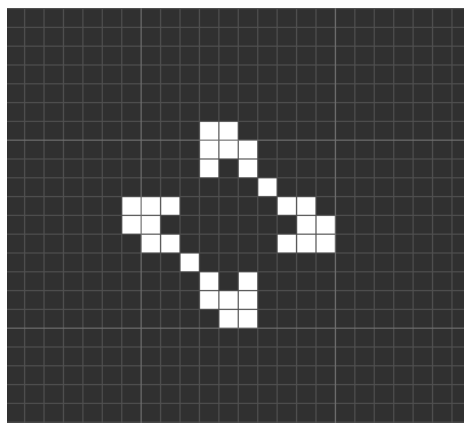
Geração 3:



Geração 4:



Geração 5:



CONCLUSÃO OU CONSIDERAÇÕES FINAIS

Autômatos celulares são formados por uma matriz (rede de células) que possuem estados alterados de acordo com seu estado interior e o estado das células vizinhas em um tempo discreto.

O breve histórico da evolução de autômato celular tem ênfase nas principais configurações para implementação do código. O código realiza todas as operações pré-definidas. Possui uma dimensão 2D com formato quadrangular, seu tamanho de vizinhança corresponde corretamente com a regra totalística. A progressão de suas gerações é de forma expansiva. Após cada geração, suas células se multiplicam exponencialmente.

REFERÊNCIAS

AGUIAR, M. S.; DIMURO, G. P.; COSTA, F. A. Modelos matemáticos baseados em autômatos celulares para geoprocessamento. SBMAC. São Carlos, SP, Brasil, 2004, 58 p.

USP, Oliveira, R. F Disponível em:

<http://www.esalq.usp.br/lepse/imgs/conteudo_thumb/Aut-matos-Celulares> Acesso em: 20 de março de 2019

USP, Oliveira, R. F Disponível em:

<http://www.esalq.usp.br/lepse/imgs/conteudo_thumb/Aut-matos-Celulares-Implementa--es-de-Von-Neumann--Conway-e-Wolfram.pdf> Acesso em: 20 de março de 2019

WIKIPEDIA. Máquina de Moore Disponível em: Acesso em: 22 de março de 2019

UFPE. Máquina de Moore Disponível em:

<<http://www.cin.ufpe.br/~if114/Monografias/Automatos%20Celulares/index.html>> Acesso em: 22 de março de 2019

WIKIPEDIA. Autômato celular. Disponível em: Acesso em: 22 de março de 2019