

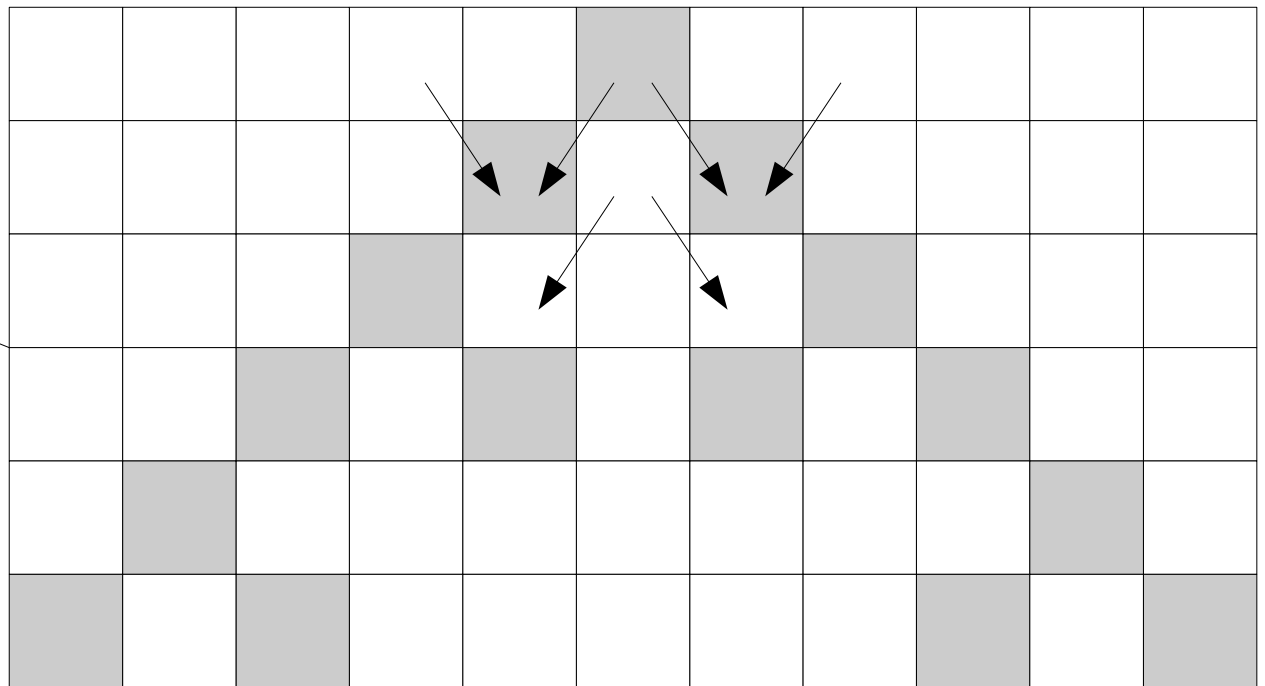
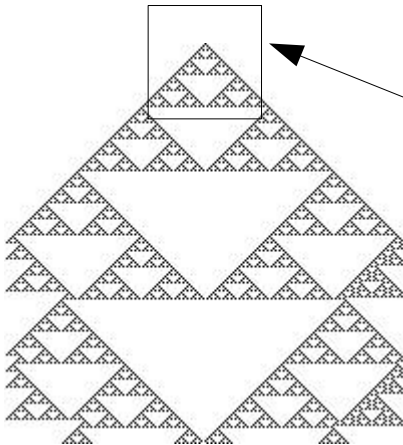
AUTOMATOS CELULARES – REGRA 90

Considere um autômato celular unidimensional:

Um autômato que segue a regra 90, tem suas células preenchidas da seguinte forma:

- 1) Se tanto a célula da esquerda quanto a da direita estiverem preenchidas, ou ambas vazias, a célula atual NÃO SERÁ PREENCHIDA;
- 2) Se somente uma das células, esquerda ou direita estiver preenchida, a célula atual SERÁ PREENCHIDA.

Veja o exemplo:



AUTOMATOS CELULARES – REGRA 90

Uma implementação simples na linguagem GuaráScript:

```
# Cria um bitmap com dimensões (dim_i x dim_j)
pixmap = matrix(0, dim_i, dim_j)

# O pixel central da primeira linha do bitmap deve estar
# preenchido.
pixmap[0, dim_j / 2] = 1

# A operação lógica XOR(^) resultará em um pixel preenchido(1),
# somente se o conteúdo das células esquerda e direita ao pixel
# atual, consideradas na linha anterior, for diferente.
for (i = 1; i < dim_i; i = i + 1) {
    for (j = 1; j < dim_j - 1; j = j + 1) {
        pixmap[i, j] = pixmap[i - 1, j - 1] ^ pixmap[i - 1, j + 1]
    }
}
```

AUTOMATOS CELULARES – REGRA 90

A imagem produzida pelo autômato da regra 90 se parece muito com o triângulo de Sierpinski, cujo algoritmo recursivo, em GuaráScript, é apresentado a seguir:

```
function Sierpinski_Triangle(n, pixmap, x1, y1, x2, y2, x3, y3, color = 1) {  
    dim_pixmap = dim(@pixmap)  
    dim_i = dim_pixmap[0]  
    dim_j = dim_pixmap[1]  
  
    # Divide o triângulo n vezes.  
    if (n >= 1) {  
        # Desenha o triângulo principal.  
        Pixmap_Triangle(pixmap, x1, y1, x2, y2, x3, y3, color)  
  
        # Desenha os triângulos interiores.  
        Sierpinski_Triangle(n - 1, pixmap, x1, y1, (x1 + x2) / 2, (y1 + y2) / 2, (x1 + x3) / 2,  
(y1 + y3) / 2, color)  
        Sierpinski_Triangle(n - 1, pixmap, (x1 + x2) / 2, (y1 + y2) / 2, x2, y2, (x3 + x2) / 2,  
(y3 + y2) / 2, color)  
        Sierpinski_Triangle(n - 1, pixmap, (x3 + x1) / 2, (y3 + y1) / 2, (x3 + x2) / 2, (y3 +  
y2) / 2, x3, y3, color)  
    }  
}
```

