

Fractais em máquina de Mealy

Feito por Uriel Pacheco de Souza

Professora: Karina Girard Roggia

O trabalho em questão visa:

- Implementação do simulador de Máquina de Mealy.
 - Entradas: dois arquivos texto, um com a descrição da Máquina de Mealy e outro com a palavra de entrada.
 - Saída: arquivo de imagem ppm.
- Modelagem de três Máquinas de Mealy: Arquivos de entrada e grafo (desenho) recebidas em forma de expressão regular:
 - $(1+2+4)^*3(2+4)^*3(1+2+3+4)^*$
 - $(3+4)^*2(1+2+3+4)^*$
 - $(1+3)(2+4)(2+3+4)(1+3+4)(1+2+4)(1+2+3)(1+2+3+4)^*$

Instruções de compilação

Para a execução do arquivo .py anexado, deve-se possuir python 3.13 instalado na máquina, podendo ser gratuitamente pela microsoft store (no windows) ou pelos comandos (no linux):

- `sudo add-apt-repository ppa:deadsnakes/ppa`
- `sudo apt update`
- `sudo apt install python3.13`

Em seguida, pode-se executar o arquivo programa utilizado para abrir o arquivo .py ou acessar a pasta onde ele se encontra pelo terminal do computador, executando o comando:

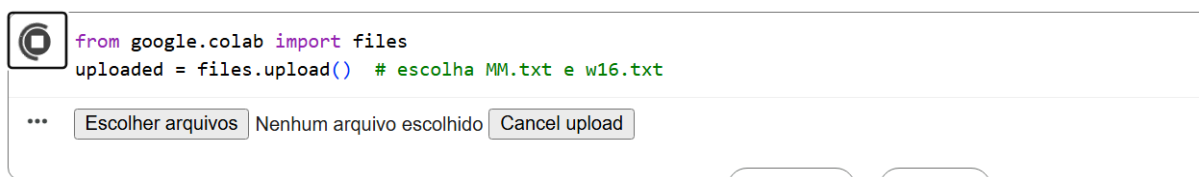
```
python3 mealy.py MM.txt w16.txt saida.ppm
```

- Onde **mealy** é o nome do arquivo código-fonte;
- **MM** é o nome do arquivo contendo a máquina de Mealy no padrão especificado nas especificações do trabalho;
- **w16** é o nome do arquivo contendo a palavra a ser analisada pela máquina de Mealy, também seguindo o padrão estipulado nas especificações;
- E **saida** é o nome do arquivo de saída ppm.

*As extensões .py, .txt e .ppm devem ser mantidas.

ALTERNATIVAMENTE, sem ser necessário baixar python na máquina, pode-se acessar o google colab no link: [LFA.ipynb](#) e execute os blocos de código em sequência:

- Importe os arquivos .txt no primeiro bloco;



O aviso abaixo aparecerá, porém fique tranquilo, os arquivos importados não serão compartilhados nem quaisquer interações realizadas.

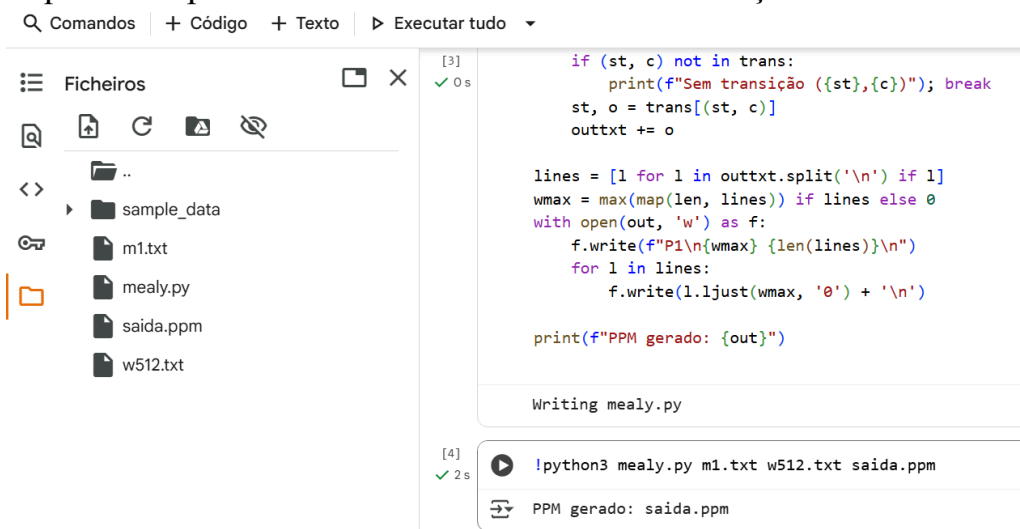
Aviso: este notebook não é de autoria do Google

Ele foi criado por **uriel.psrpg@gmail.com**. Ele pode solicitar acesso aos seus dados armazenados pelo Google ou ler dados e credenciais de outras sessões. Revise o código-fonte antes de executar este notebook. Se ainda tiver dúvidas, entre em contato com o criador deste notebook pelo e-mail uriel.psrpg@gmail.com.

Cancelar Executar assim mesmo

- Execute o segundo bloco;
- Altere o terceiro bloco seguindo o exemplo de execução acima (selecionando o nome dos arquivos importados), mantenha o “mealy.py”, pois esse nome é fixo.

Assim, o google colab irá gerar o arquivo .ppm no menu contido na esquerda, o qual você poderá fazer o download e a visualização.



```
[3]
✓ 0 s

if (st, c) not in trans:
    print(f"Sem transição ({st},{c})"); break
st, o = trans[(st, c)]
outtxt += o

lines = [l for l in outtxt.split('\n') if l]
wmax = max(map(len, lines)) if lines else 0
with open(out, 'w') as f:
    f.write(f"P1\n{wmax} {len(lines)}\n")
    for l in lines:
        f.write(l.ljust(wmax, '0') + '\n')

print(f"PPM gerado: {out}")

Writing mealy.py

[4]
✓ 2 s
!python3 mealy.py m1.txt w512.txt saida.ppm

PPM gerado: saida.ppm
```

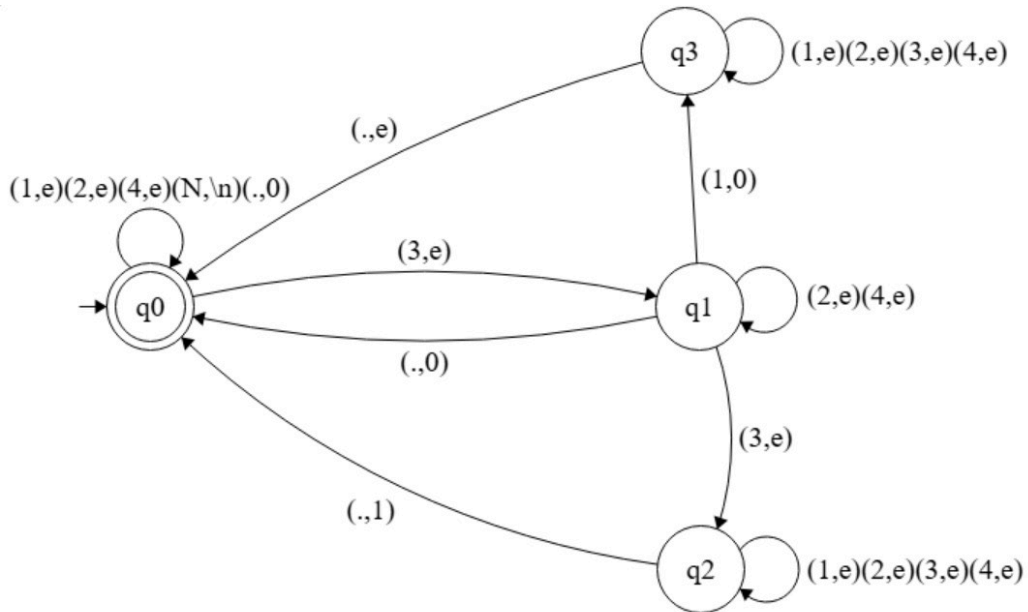
*Há também as últimas duas células de código que você poderá executar para visualizar o ppm e receber um png pelo próprio colab, basta alterar a “saida.ppm” para o nome do arquivo gerado anteriormente.

*A segunda e a quarta células não geram nada, porém, deve-se executá-las para haver resultado na terceira e quinta célula respectivamente.

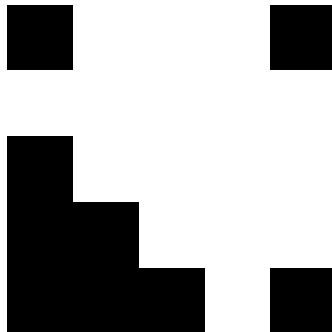
Modelagens das expressões regulares

Primeira expressão regular

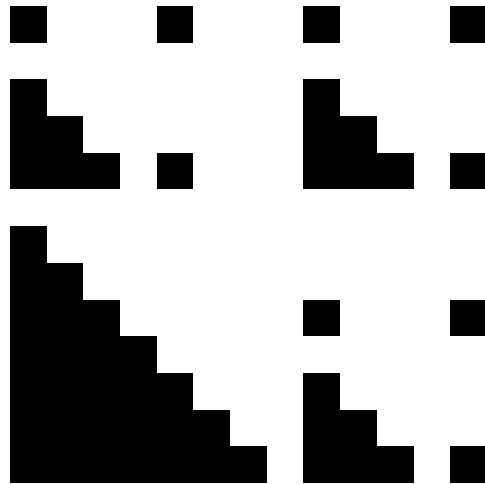
$(1+2+4)^*3(2+4)^*3(1+2+3+4)^*$



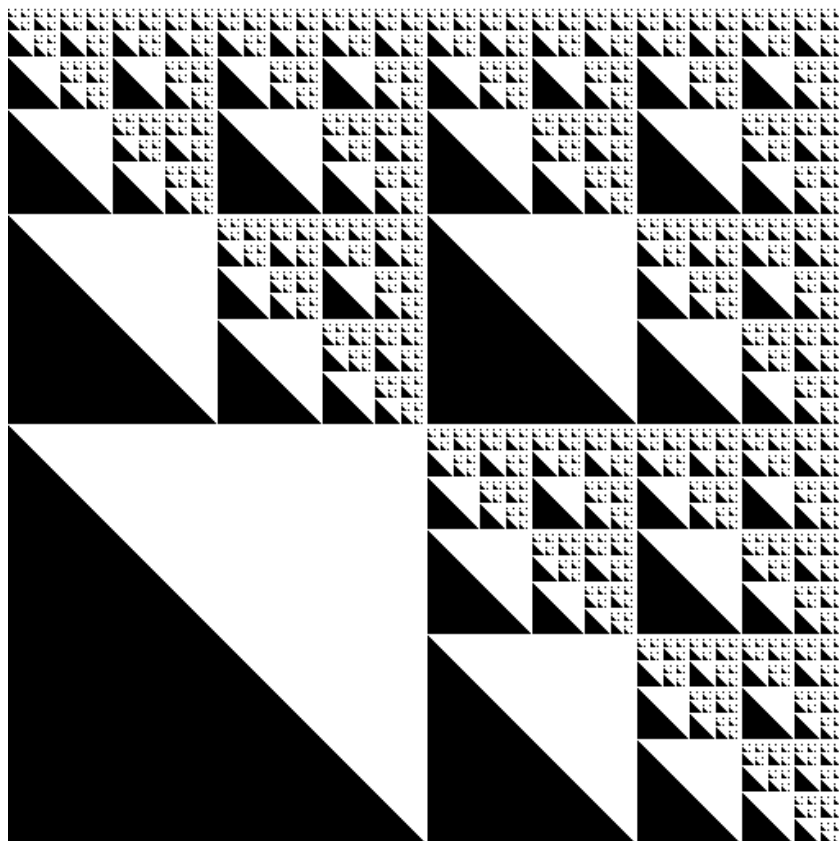
- W8



- **W16**



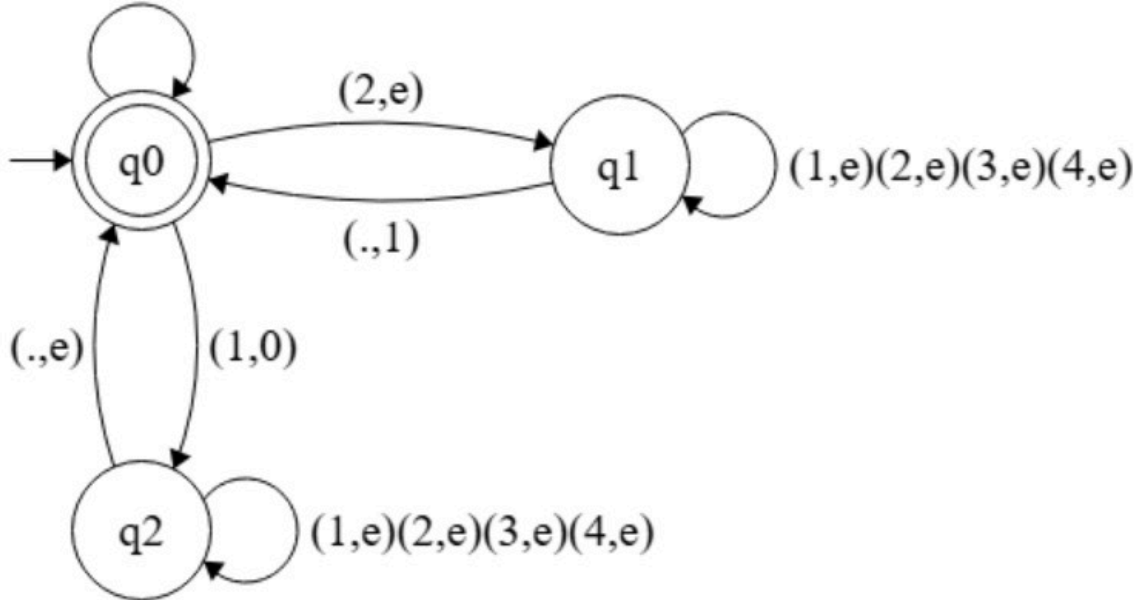
- **W512**



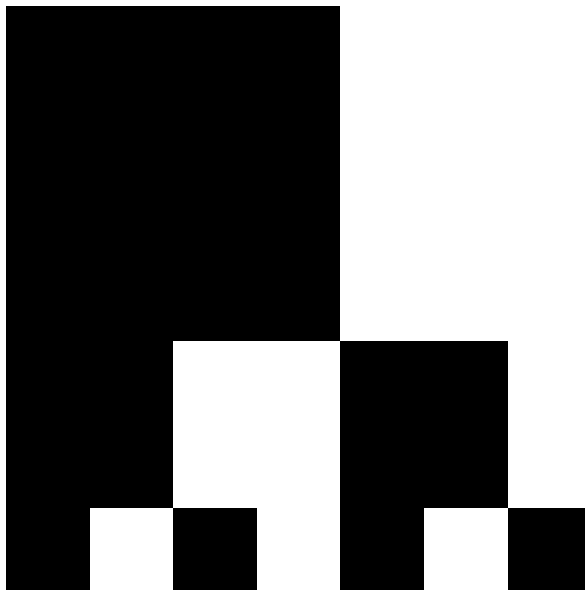
Segunda expressão regular

$(3+4)^*2(1+2+3+4)^*$

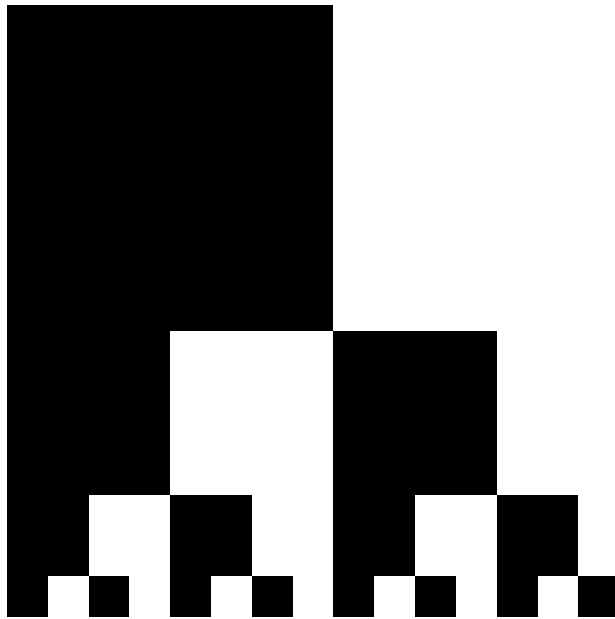
$(3,e)(4,e)(N,\backslash n)(.,0)$



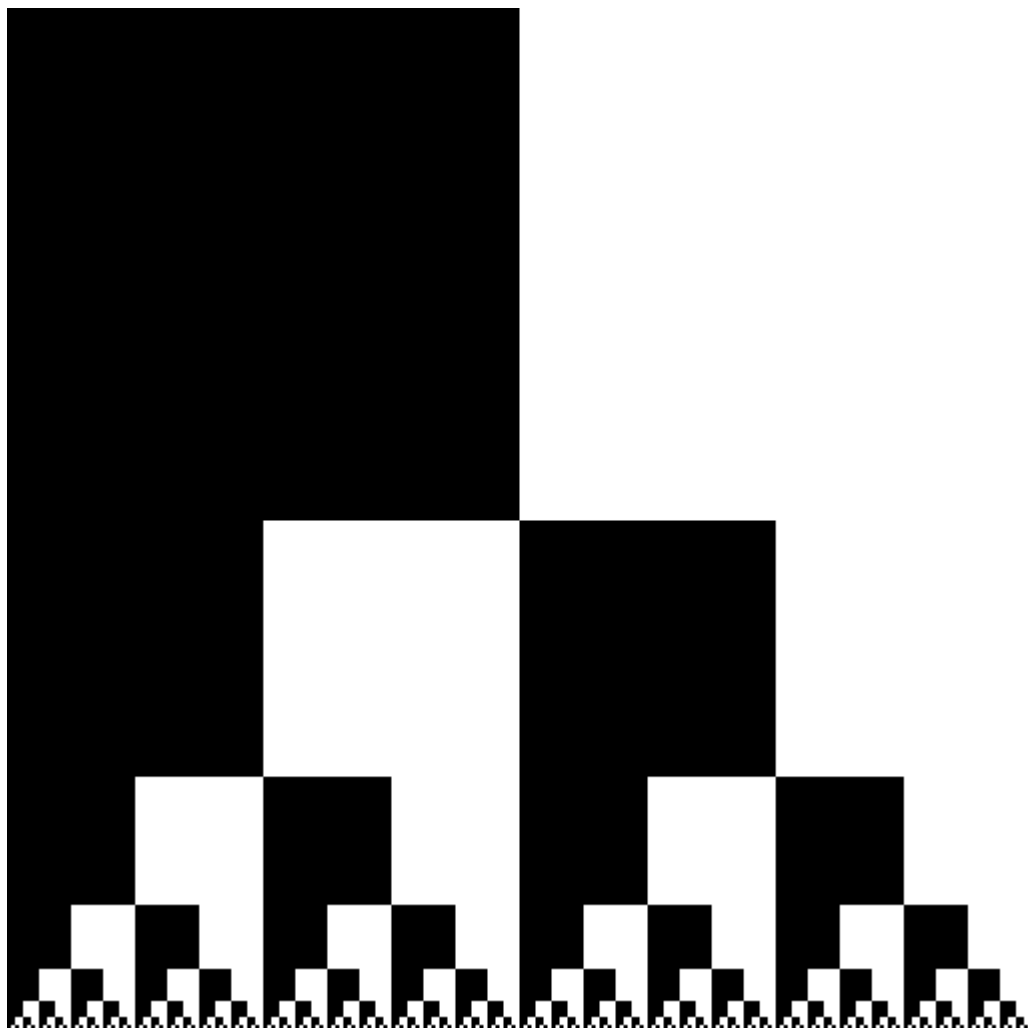
- W8



- **W16**

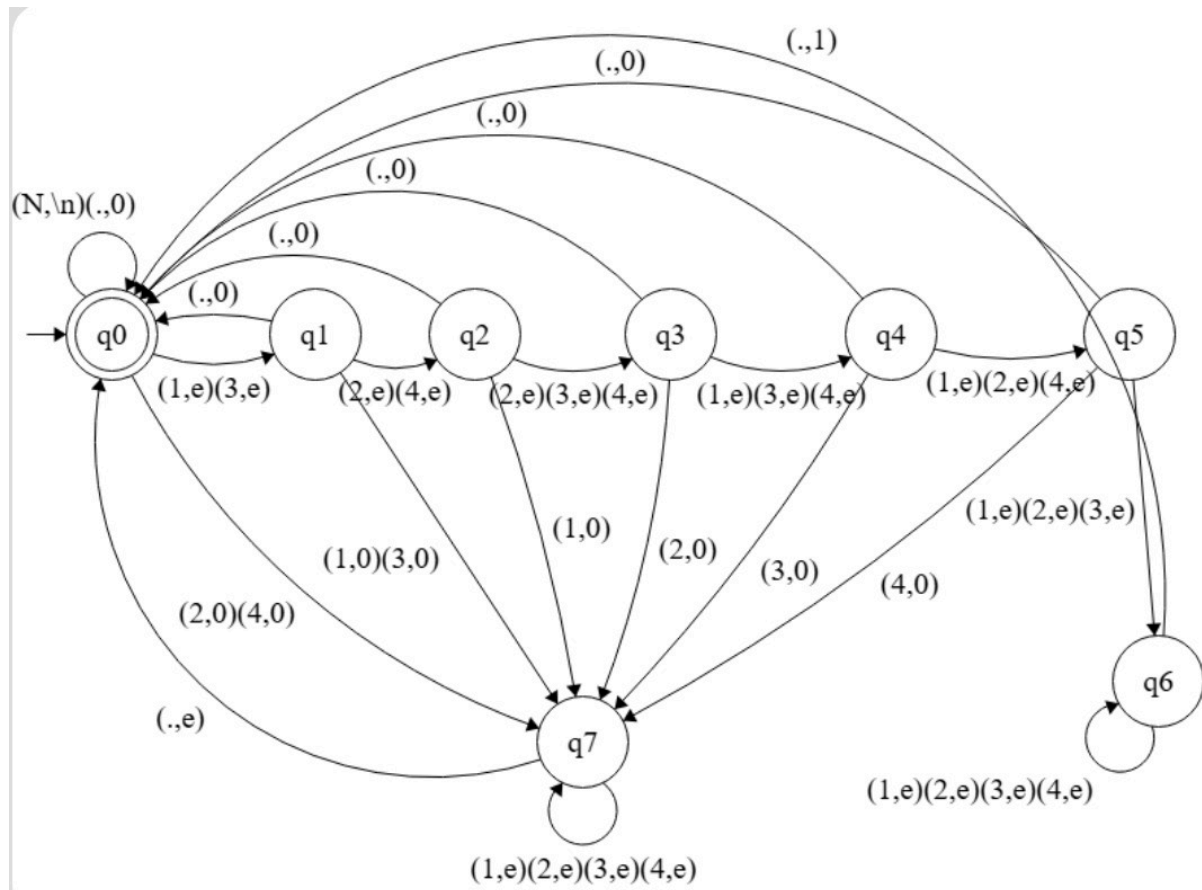


- **W512**



Terceira expressão regular

$(1+3)(2+4)(2+3+4)(1+3+4)(1+2+4)(1+2+3)(1+2+3+4)^*$



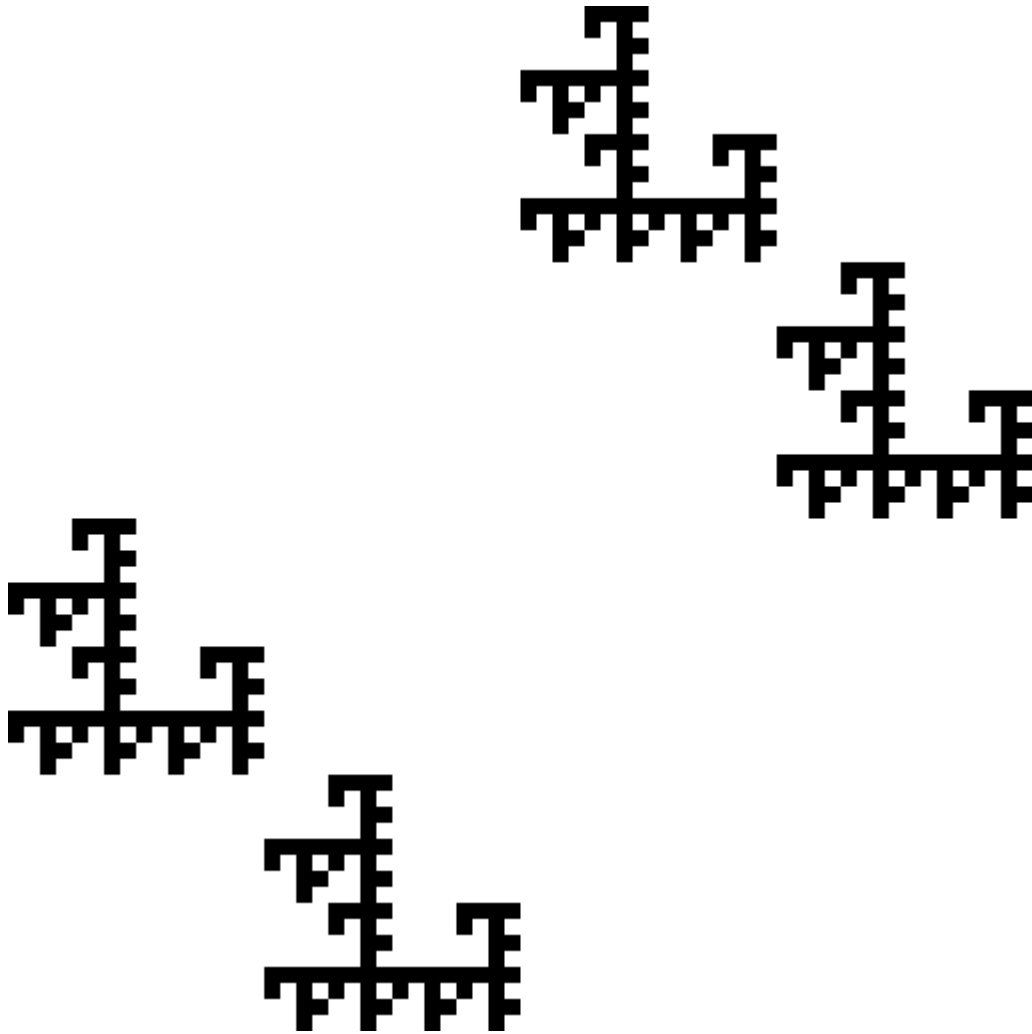
- **W8**

*Imagem branca, pois as palavras que retornam 1 pelo autômato são muito complexas para ter como saída tal valor, visto que a palavra contida em W8 só possui 3 símbolos por célula (subpalavra entre os pontos), e o autômato precisa de 6 símbolos no mínimo para retornar 1 e pintar o pixel.

- **W16**

*Imagem branca pelo mesmo motivo apresentado acima, todas as células com apenas 4 símbolos retornarão 0.

- **W512**



Observações

- Cada expressão induz alguma restrição para a máquina de Mealy correspondente, ou seja, a primeira expressão por exemplo aceita as palavras que contenham dois símbolos 3, as quais não pode haver o símbolo 1 entre eles (aceita no caso do AFD correspondente à expressão, no caso da máquina de Mealy retorna 1 como saída). Já a segunda expressão aceita as palavras que contenham ao menos um símbolo 2, sendo que não pode haver um símbolo 1 antes do primeiro símbolo 2.
- A maior dificuldade foi o primeiro passo de modelar a máquina de Mealy, várias dúvidas surgiram, como qual estágio deveria ser o final por exemplo, mas o exemplo fornecido no documento de especificações auxiliou bastante nesse quesito.

Referências

HOPCROFT, John E.; MOTWANI, Rajeev; ULLMAN, Jeffrey D. **Introdução à Teoria de Autômatos, Linguagens e Computação**. 3. ed. São Paulo: Pearson Addison Wesley, 2008.

MENEZES, Paulo Blauth. **Linguagens Formais, Autômatos e Computabilidade**. 3. ed. Porto Alegre: Bookman, 2011.

SIPSER, Michael. **Introduction to the Theory of Computation**. 3rd ed. Boston: Cengage Learning, 2012.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023: Informação e documentação — Referências — Elaboração**. Rio de Janeiro: ABNT, 2018.

PYTHON SOFTWARE FOUNDATION. **Python 3 Documentation**. Disponível em: <https://docs.python.org/3/>. Acesso em: 22 out. 2025.

NETPBM. **The PPM (Portable Pixmap) Format**. Disponível em: <https://netpbm.sourceforge.net/doc/ppm.html>. Acesso em: 22 out. 2025.

GOOGLE. **Google Colaboratory**. Disponível em: <https://colab.research.google.com/>. Acesso em: 22 out. 2025.