

Minimização de Autômatos Finitos

Alisson da Silva Vieira, RA: 2046229

Higor Luiz Farinha Celante, RA.: 1602870

Lucas Henrique Malaquias da Silva Donadi, RA: 1711598

Maria Fernanda Pinguelli Sodré, RA: 2046407

15 de outubro de 2019

1 Introdução

1.1 Sobre o Autômato Finito Determinístico

Autômato Finito(AFD), pode ser entendido como um Máquina de estados finita que tem a possibilidade de se aceitar ou rejeitar sequências de entradas, gerando um único ramo de computação para cada sequência de entrada. O primeiro conceito relacionado ao AFD, foi proposto por McCulloch e Pitts em 1943. Esse modelo, teve como proposta, definir estruturas mais simples para reproduzir máquinas de estado finitas.([MCCULLOCH; PITTS,](#)) ([HOPCROFT JOHN E.; MOTWANI,](#))

1.2 Minimização de Autômatos

No estudo da teoria dos autômatos, a Minimização de AFD é uma transformação de um determinado autômato finito determinístico em outro autômato equivalente, com o menor número possível de estados. Para comprovar a equivalência entre dois autômatos finitos determinísticos, se ambos descreverem a mesma linguagem regular.([BRZOZOWSKI,](#))

2 Instruções para executar o código

Para executar o código é necessário ter instalado a aplicação web Jupyter Notebook, que permite a criação e compartilhamento de código, equações, visualizações gráficas e texto. Segue os comandos necessários para instalar o Jupyter Notebook:

1. `apt-get install python-pip python-dev`
2. `pip install --upgrade pip`
3. `pip install jupyter`

Dentro do repositório do projeto, execute o comando:

4. `jupyter notebook --allow-root`

Após o comando em alguns instantes o diretório irá abrir no navegador padrão do sistema:

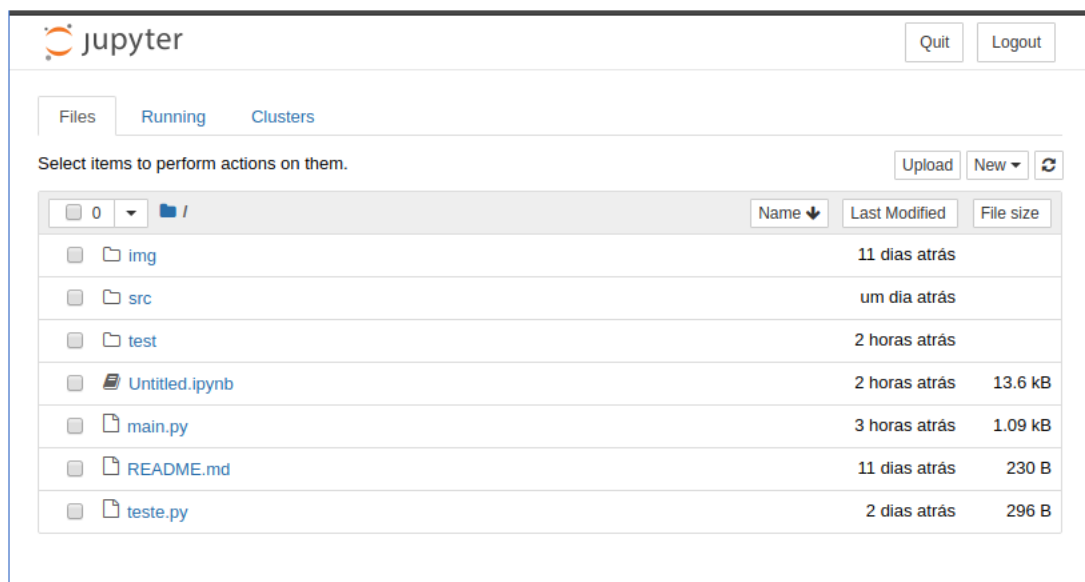


Figura 1 – Jupyter executando no navegador.

Nesta página temos o arquivo `Untitled.ipynb`, que é onde está o código do projeto, selecione o documento para visualiza-lo, e a página a seguir irá ser carregada em seu navegador:

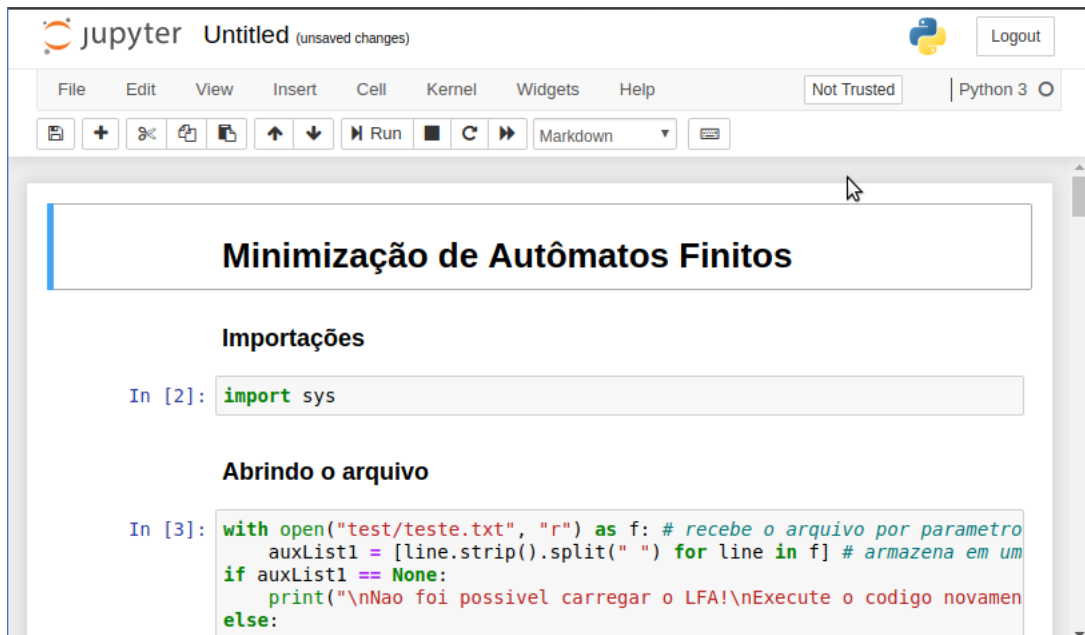


Figura 2 – Documento Untitled.ipynb executando no navegador.

Na seção "Abrindo o arquivo", em sua primeira linha temos a importação do arquivo que está na pasta "test", além do arquivo "teste.txt" temos também o "teste2.txt" e o "teste3.txt", para trocar o arquivo é só mudar o texto que leva ao arquivo. E para executar todo o código é selecionar "Cell -> Run Cells" as saídas do código serão mostrada bloco por bloco.

3 Desenvolvimento

3.1 Método: Busca e remoção dos estados inalcançáveis

Procura os estados inalcançáveis, de forma que os estados subsequentes também sejam considerados inalcançáveis, caso nenhum outro estado alcançável aponte para ele. Então é feita a exclusão das transições para estados inalcançáveis juntamente com os estados em si.

```

lista_mantidos = [] + estados_iniciais
lista_excluidos = []
# busca por inalcançáveis
for estado in estados:
    for tr in transicoes:
        if((tr[0] != estado) and (tr[2] == estado)):
            lista_mantidos.append(estado)

retirados = list(set(estados) - set(lista_mantidos))
print("Estados inalcançáveis",retirados)
alc_por_nalc = []

for tra in transicoes:
    if((tra[0] in retirados) and (tra[2] != tra[0])):
        #print(tra[0])
        alc_por_nalc.append(estado)

print("Possíveis estados alcançáveis por inalcançáveis",alc_por_nalc)

for tra in transicoes:
    for st in alc_por_nalc:
        if((tra[2] == st) and (tra[0] not in retirados)):
            del(st)
print("Estados à serem retirados",alc_por_nalc + retirados)
retirados = retirados + alc_por_nalc

```

Figura 3 – Busca de estados inalcançáveis

```

newTrs = []
index_excluded = []
#for ex in sort_out:
#    index_excluded.append(estados.index(ex))
for ttrs in estado_transicoes:
    for state in ttrs:
        #print(state[0])
        #print(retirados)
        if((str(state[0]) not in retirados) and (str(state[2]) not in retirados)):
            print(ttrs)
            newTrs.append(ttrs)
            break
estados = list(set(estados) - set(retirados))
print(estados)

```

Figura 4 – Remoção de estados inalcançáveis

3.1.1 Exemplo:

O estado "s8", só pode ser alcançado pelo estado "s7", mas "s7" é inalcançável, portanto, "s8" é inalcançável.

```
Estados inalcançáveis ['s7']  
Possíveis estados alcançáveis por inalcançáveis ['s8']  
Estados à serem retirados ['s8', 's7']
```

Figura 5 – Representação de estados não alcançáveis

3.2 Método: Busca de estados equivalentes

Com base nos parâmetros, esse método tem o objetivo de encontrar os estados equivalentes.

```
def equivalent_match(sx,sy,transicoes,alfabeto,estados_finais): # retorna se equivalent (compara as transicoes )  
    if sx == sy: #Se forem a mesma transicao já nem compara  
        return None  
  
    if not(is_estado_final(sx,estados_finais) == is_estado_final(sy, estados_finais)): #Se ambas não forem estados de  
        aceitação ou o contrário, significa que não tem a mesma decisão para a palavra vazia, sendo não equivalentes  
        return None  
  
    for simbolo in alfabeto:#Verifica a equivalencia para cada letra do alfabeto, até achar a primeira a não ser equivalente  
        linhaX = find_row(sx, simbolo, transicoes)# acha a linha com o estado, e o simbolo do alfabeto desta iteração  
        linhaY = find_row(sy, simbolo, transicoes)  
  
        if linhaX[2] != linhaY[2]:#se ambas forem para estados diferentes verifica se são estados finais/aceitação  
            if not(is_estado_final(linhaX[2],estados_finais) and is_estado_final(linhaY[2], estados_finais)):#se ambas forem  
                para estados diferentes de não aceitação devolvemos nada  
                return None  
  
    return [sx, sy]  
alfabeto: list  
equiv = []  
for n in estados[:len(estados)-1]: # Ex: 0, 1, 2, 3, 4 menos o ultimo  
    for i in reversed(estados[1:]): # Ex: 1, 2, 3, 4, 5 menos o primeiro  
        equiv.append(equivalent_match(n,i,transicoes,alfabeto,estados_finais)) # n= estado1 , i= estado 2,  
equiv = list(filter(None, equiv))  
print(equiv)
```

Figura 6 – Busca de estados equivalentes

3.2.1 Método: Verificação de transição correspondente aos parâmetros

Esse método tem como objetivo, retornar a transição correspondente, com base no estado, símbolo e lista de transições, parâmetros do mesmo método.

3.2.2 Método: Verificação de estado final

A verificação de estado final ocorre com a comparação dos estados finais com o estado passado por parâmetro no método. Se o estado for igual a algum dos estados presentes na lista de estados finais, retorna "True".

3.3 Método: Construção da versão do Autômato Minimizada

Através desse método, a versão minimizada do autômato é criada, após todos os processos descritos anteriormente.

3.3.1 Método: Renomeação dos estados e transições

Este método dá um novo nome e renomeia os estados e transições, através dos estados equivalentes, também tirando as transições repetidas.

```
for transisp in newTrs:
    for transi in transisp:
        for compativeis in pcompativeis:
            for frac in compativeis:
                nome = str()
                for part in compativeis:
                    nome = str(nome.strip('') + part.strip(''))
                if(str(transi[0]) == frac):
                    transi[0] = str(nome)
                if(str(transi[2]) == frac):
                    transi[2] = str(nome)

            # print(transi)

# print(newTrs)
setTrs = []
for estado in newTrs:
    for trsc in estado:
        if(trsc not in setTrs):
            setTrs.append(trsc)
print(setTrs)
```

Figura 7 – Renomeia estados equivalentes

3.4 Método: Montar máquina

Cria nova máquina com os novos estados e transições em um novo arquivo .txt.

Referências

BRZOZOWSKI, J. A. . *Canonical regular expressions and minimal state graphs for definite events*. Polytechnic Press of Polytechnic Inst. of Brooklyn, Brooklyn, N.Y., pp. 529–561, MR 0175719. Citado na página [1](#).

HOPCROFT JOHN E.; MOTWANI, R. U. J. D. *Introduction to Automata Theory, Languages, and Computation 2 ed.* Polytechnic Press of Polytechnic Inst. of Brooklyn, Brooklyn, N.Y., pp. 529–561, MR 0175719: Addison Wesley. Citado na página [1](#).

MCCULLOCH, W. S.; PITTS, E. . *A logical calculus of the ideas imminent in nervous activity*. University of Illinois, College of Medicine. Citado na página [1](#).