

Fundamentos Teóricos da Computação

CIÊNCIA DA COMPUTAÇÃO

Prof. Dr. João Paulo Aramuni

Sumário

- * **Grafos**
- * **Máquinas de Estados Finitos**

Grafos

- * Para que possamos trabalhar com máquinas de estados finitos, precisamos primeiro entender o conceito de **Grafo**.
- * Utilizaremos esta estrutura para representar graficamente as máquinas de estados finitos.

Grafos

- * A teoria dos grafos é um ramo da matemática que estuda as relações entre os objetos de um determinado conjunto.
- * Grafos são estruturas matemáticas utilizadas para modelar relações entre pares de objetos.
- * Um grafo é uma estrutura matemática que contém dois tipos de entidades: vértices e arestas.

Grafos

- * Existem dois tipos básicos de grafos: os dirigidos e os não dirigidos.
- * Nos grafos dirigidos as arestas (dirigidas) são pares ordenados de vértices, e nos grafos não dirigidos as arestas (**não** dirigidas) são pares não ordenados de vértices.

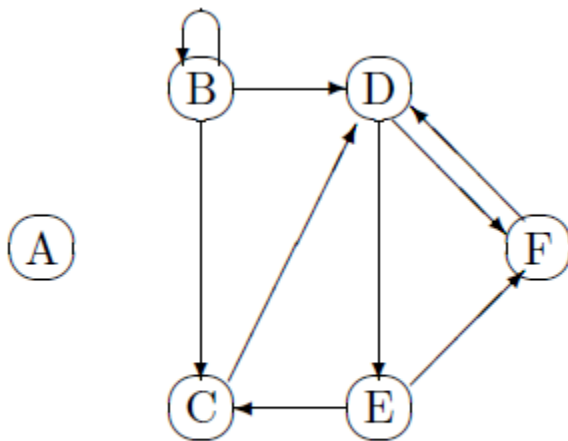
Grafos

- * Na representação gráfica de um grafo, um vértice é, em geral, representado por meio de uma curva fechada, como um círculo, uma oval, etc.
- * Em um grafo dirigido, uma aresta é representada por meio de uma seta ligando as representações dos dois vértices da aresta no sentido do primeiro para o segundo vértice.
- * Em um grafo **não** dirigido, uma aresta é representada por meio de uma linha (reta ou curva) ligando as representações dos dois vértices da aresta.

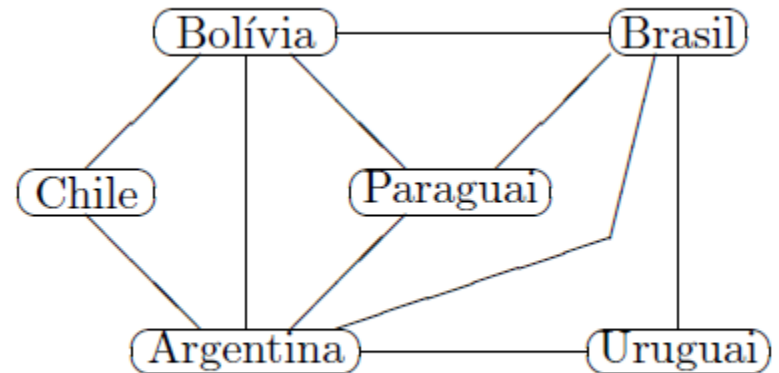
Grafos

- * As figuras a seguir mostram exemplos de representações gráficas de um grafo dirigido (a) e de um grafo não dirigido (b).
- * No grafo não dirigido (b) existe uma aresta conectando dois vértices v e v' se, e somente se, o país v tem fronteira com o país v' .

Grafos



(a) Um grafo dirigido.



(b) Um grafo não dirigido.

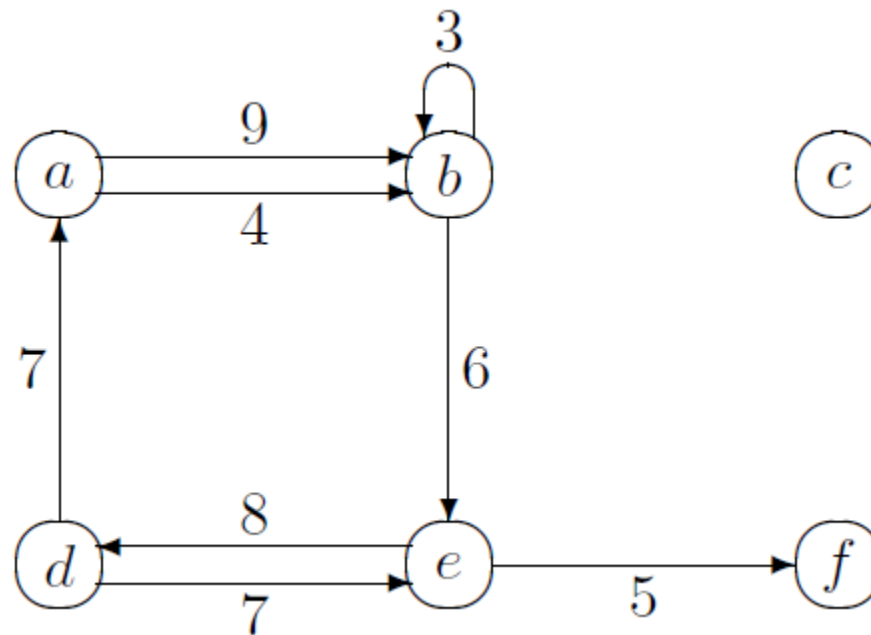
Grafos

- * Além dos dois tipos básicos de grafos exemplificados anteriormente, existem variações.
- * Por exemplo, existem os grafos mistos, que têm ambos os tipos de arestas, dirigidas e não dirigidas.
- * Em alguns contextos, os grafos contêm, não um conjunto de arestas, mas um multi-conjunto. Neste último caso, podem existir várias arestas para um único par de vértices.

Grafos

- * Neste curso, iremos dar ênfase aos *Grafos Dirigidos Rotulados*.
- * A representação gráfica é similar à de um grafo não rotulado. A diferença é que, para uma aresta (a, r, b) , coloca-se, além da seta de a para b , o rótulo r adjacente à seta, como mostra o exemplo a seguir.
- * Observe que agora podem existir várias arestas que saem do mesmo vértice e entram em um vértice comum; basta que seus rótulos sejam diferentes.
- * Um grafo não dirigido rotulado pode ser definido de forma análoga.

- * Exemplo de *Grafo Dirigido Rotulado*:



Grafos

- * Um grafo que não tem ciclos é dito ser um grafo acíclico.
- * Um grafo não dirigido é dito conexo se existe caminho de qualquer vértice a qualquer outro. E um grafo dirigido em que existe caminho de qualquer vértice a qualquer outro é dito ser fortemente conexo.

Máquinas de Estados Finitos

- * As máquinas de estados finitos são máquinas abstratas que capturam as partes essenciais de máquinas concretas.
- * Ou seja, capturam o funcionamento básico de máquinas concretas.
 - * Máquinas de Refrigerante
 - * Relógios Digitais
 - * Programas de computador
 - * Compiladores

Máquinas de Estados Finitos

- * Não são as abstrações mais poderosas
 - * Mas modelam um grande espectro de máquinas
 - * O próprio computador digital, se considerarmos que sua memória é limitada, pode ser modelado por meio de uma máquina de estados finitos.
- * Existem basicamente , dois tipos de máquinas de estados finitos: os transdutores e os reconhecedores (ou aceitadores) de linguagens. Abordaremos inicialmente os reconhecedores de linguagem.

Máquinas de Estados Finitos

- * Os transdutores são máquinas com entrada e saída, e os reconhecedores são máquinas com apenas duas saídas possíveis; geralmente uma delas significa “aceitação” da entrada e, a outra, “rejeição” da entrada.

Máquinas de Estados Finitos

- * Reconhecedores (ou aceitadores)
 - * Possuem apenas duas saídas possíveis
 - * Aceitação ou rejeição da entrada
 - * Reconhecedores de linguagem são chamados ***autômatos finitos***
 - * Linguagens reconhecíveis por autômatos finitos são chamadas ***linguagens regulares***
- * Transdutores
 - * Possuem entrada e saída

Um olhar para o futuro

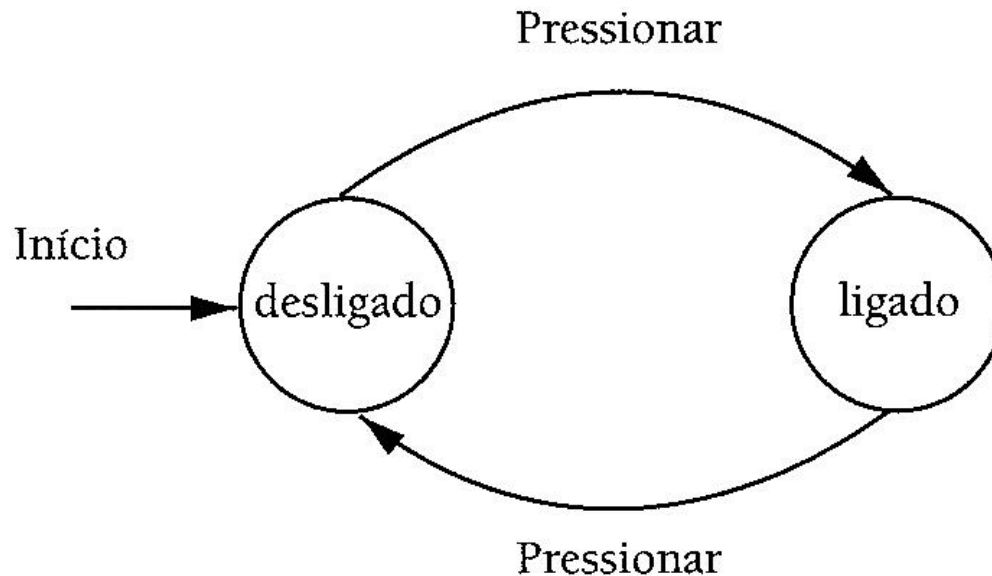
- * As linguagens reconhecidas por máquinas de estados finitos são denominadas *linguagens regulares*.
- * Existem duas notações bastante úteis para a especificação de linguagens regulares:
 - * Expressões Regulares
 - * Gramáticas Regulares

Exemplo 1

- * Seja uma máquina elétrica qualquer
 - * Possui 3 entradas
 - * Entrada de Liga (pulso)
 - * Entrada de Desliga (pulso)
 - * Entrada de Falha (retentiva)
 - * Possui uma saída
 - * Liga
 - * Funcionamento
 - * Um pulso em liga, liga a máquina
 - * Um pulso em desliga, desliga a máquina
 - * Sinal de falha, desliga a máquina e coloca em defeito

Modelo de um autômato finito

- * Um autômato finito que modela um interruptor liga/desliga
 - * Observe que utilizaremos aqui os conceitos de **Grafo**.



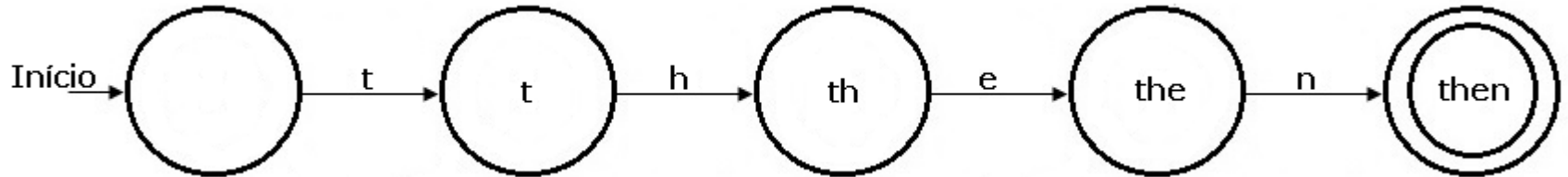
Exemplo 2

- * Às vezes, o que é memorizado por um estado pode ser muito mais complexo que uma escolha entre ligado/desligado.
- * Este exemplo mostra um autômato finito que poderia fazer parte de um **analisador léxico**.

Exemplo 2

- * O trabalho desse autômato é reconhecer a palavra *then*
- * Desse modo, ele precisa de cinco estados, cada um dos quais representa uma posição diferente na palavra *then*, conforme o que foi atingido até o momento
- * Essas posições correspondem aos prefixos da palavra, variando desde o string vazio até a palavra completa

Autômato finito que modela o reconhecimento de *then*



Entendendo o diagrama de estados

- * Cada estado é representado por uma **oval** e cada transição possível é representada por uma **seta** ligando o estado ao qual ela se aplica ao estado que resulta de sua aplicação.
- * O estado inicial é ressaltado por meio de uma **seta que o aponta** e o estado final é ressaltado por meio de uma **oval dupla**.
- * Se existe uma aresta de um estado e_1 para um estado e_2 com rótulo a no diagrama de estados, será dito que há uma *transição de e_1 para e_2 sob a* .

Exemplo 3

- * **Um homem, um leão, um coelho e um repolho** devem atravessar um rio (da margem esquerda para a direita) usando uma canoa. Somente o homem consegue guiar a canoa, e ele só pode transportar um dos outros três de cada vez. Sempre que chegam a uma das margens, o homem desce da canoa primeiro, retira o outro ocupante (se tiver algum), coloca o próximo ocupante na canoa, caso este seja o caso, e depois sobe na canoa. O leão não pode ficar na mesma margem do rio que o coelho sem a presença do homem, e o coelho não pode ficar na mesma margem do rio que o repolho sem a presença do homem.

Como resolver o problema?

- * Quais informações são realmente relevantes para a solução do problema?
- * Como estruturar estas informações (qual representação) para facilitar a solução?

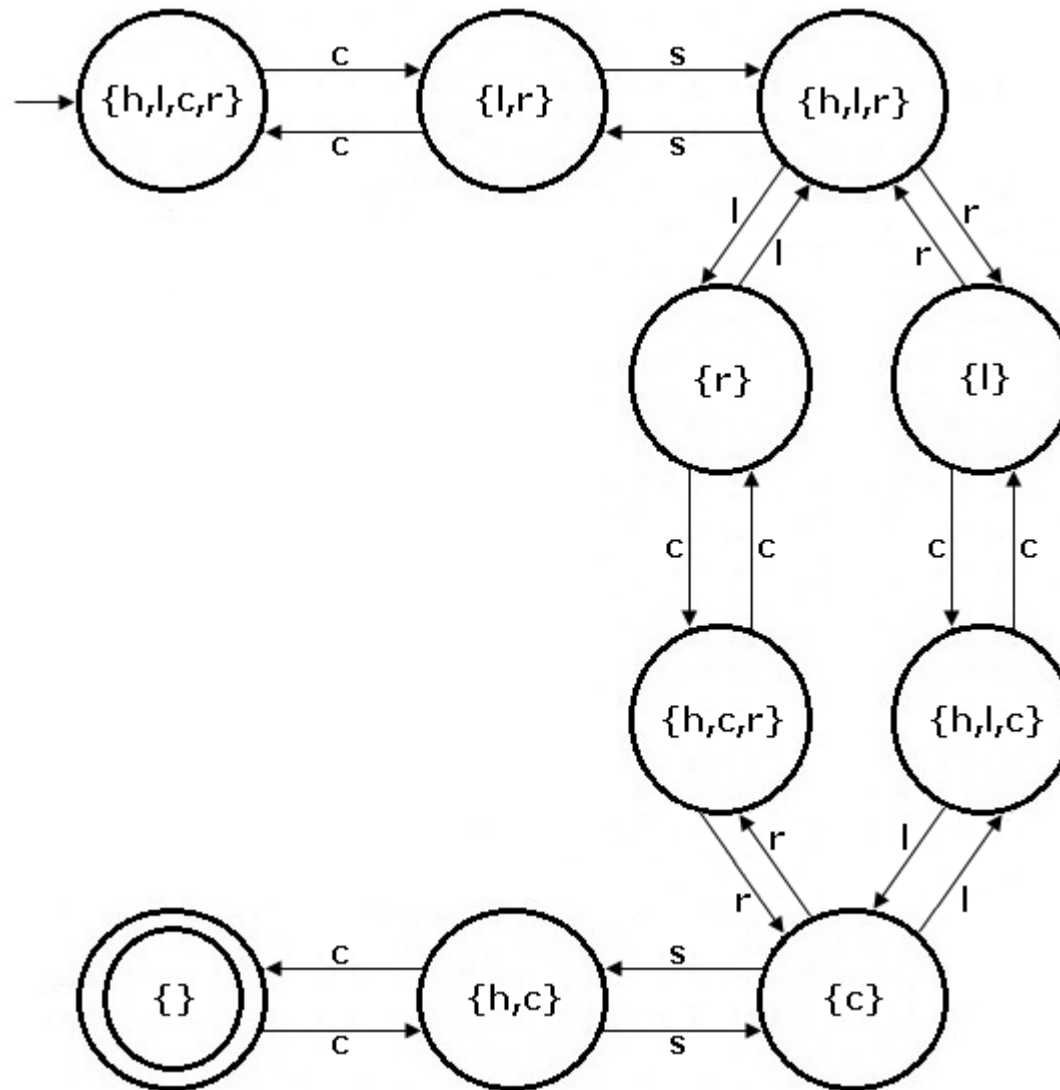
Informações Relevantes

- * Em que margem do rio estão o homem, o leão, o coelho e o repolho?
- * Qual a sequência de movimentações que levou à situação atual do sistema?

Representações

- * Estado (quem está na margem esquerda?)
 - * $h \rightarrow$ homem
 - * $l \rightarrow$ leão
 - * $c \rightarrow$ coelho
 - * $r \rightarrow$ repolho
- * Sequência de movimentações
 - * Palavra
 - * Exemplos de palavras reconhecidas:
 - * $w = cslcrsc$
 - * $w = csrclsc$
 - * $s \rightarrow$ homem sozinho na canoa

Diagrama de estados para o homem, o leão, o coelho e o repolho



Aceitação

- * Uma palavra w é reconhecida, ou aceita, se, partindo do estado inicial, ela leva a um estado final
 - * Se w não é reconhecida, diz-se que é rejeitada
- * Se x é um prefixo de w , ou seja, $w = xy$, para verificar se w é reconhecida depois do processamento x basta
 - * Saber o estado atual e;
 - * y

Dúvida

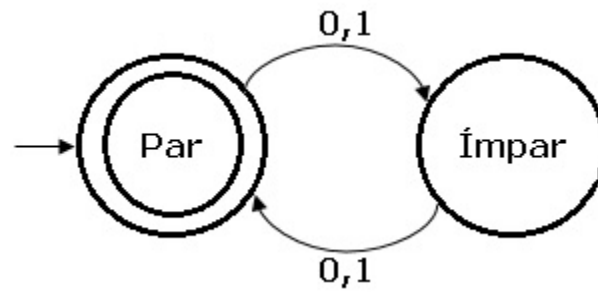
- * Dada uma palavra w de $\{s, l, c, r\}^*$, como saber se w é uma solução?
- * Deve-se verificar se há um “caminho correspondente” para a w , partindo do estado inicial até chegar ao estado final.
- * Se chegar, w é uma solução. Caso contrário, não.
- * No primeiro caso, diz-se que w é reconhecida, ou aceita, e no segundo diz-se que não é reconhecida, e sim rejeitada.

Exercícios

- 1) Projete um autômato finito que reconheça palavras binárias que tenham um número par de símbolos.
- 2) Projete um autômato finito que reconheça palavras binárias com número ímpar de 1s.
- 3) Projete um autômato finito que reconheça palavras binárias que comecem com 0 e terminem com 0.
 - * *Definição informal:* $\{0\}\{0,1\}^*\{0\}$
 - * *Definição formal:* $L = \{ 0w0 \mid w \text{ e } \{0,1\}^* \}$
- 4) Projete um autômato finito que reconheça palavras sobre o alfabeto $\{a,b\}$ que não tenham dois a's consecutivos.
 - * $L = \{ w \in \{a,b\}^* \mid w \text{ nunca tem dois a's consecutivos} \}$

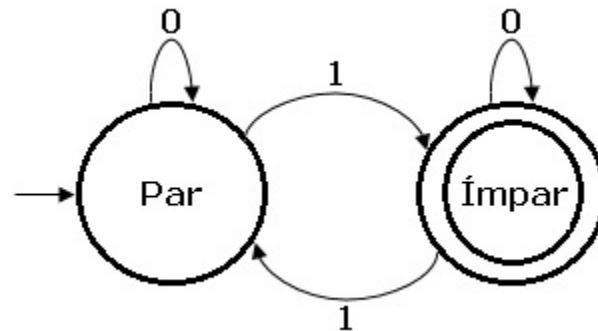
Respostas

1)



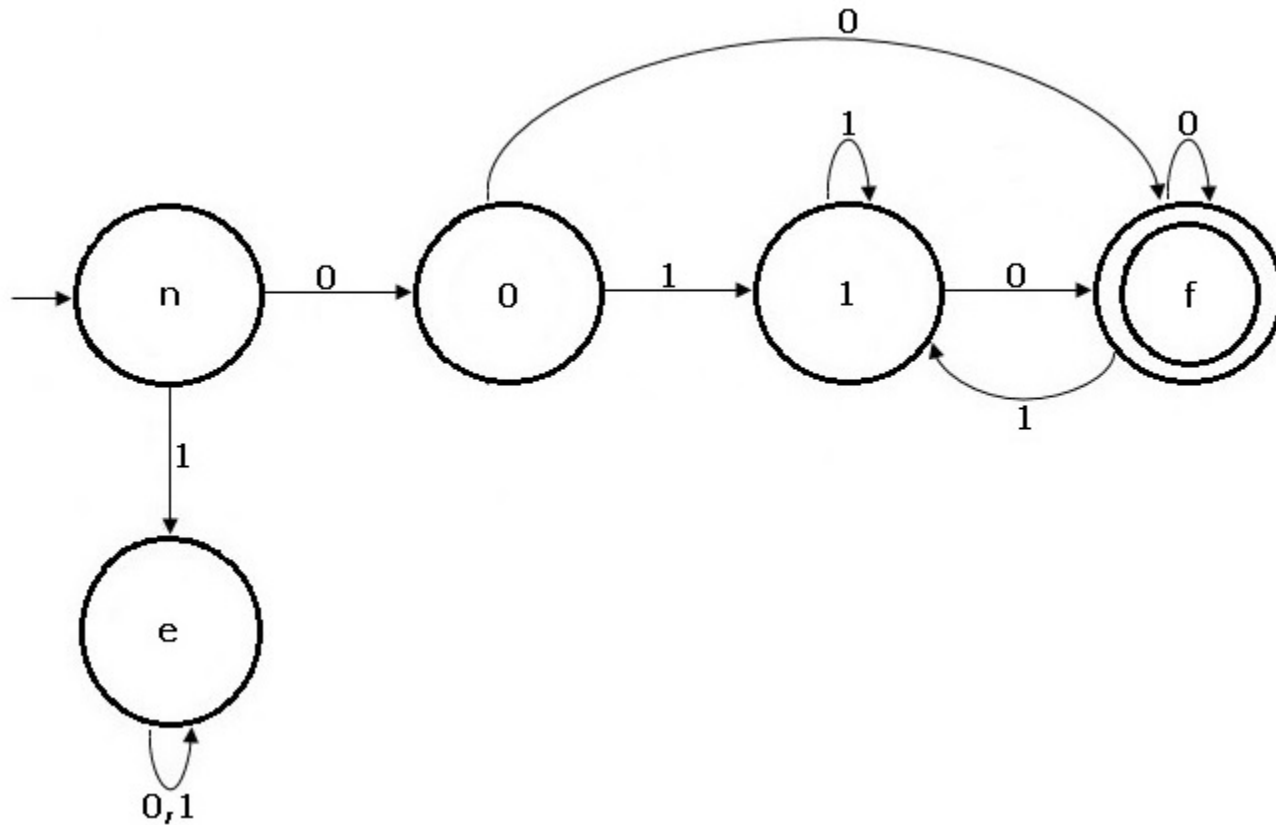
Respostas

2)



Respostas

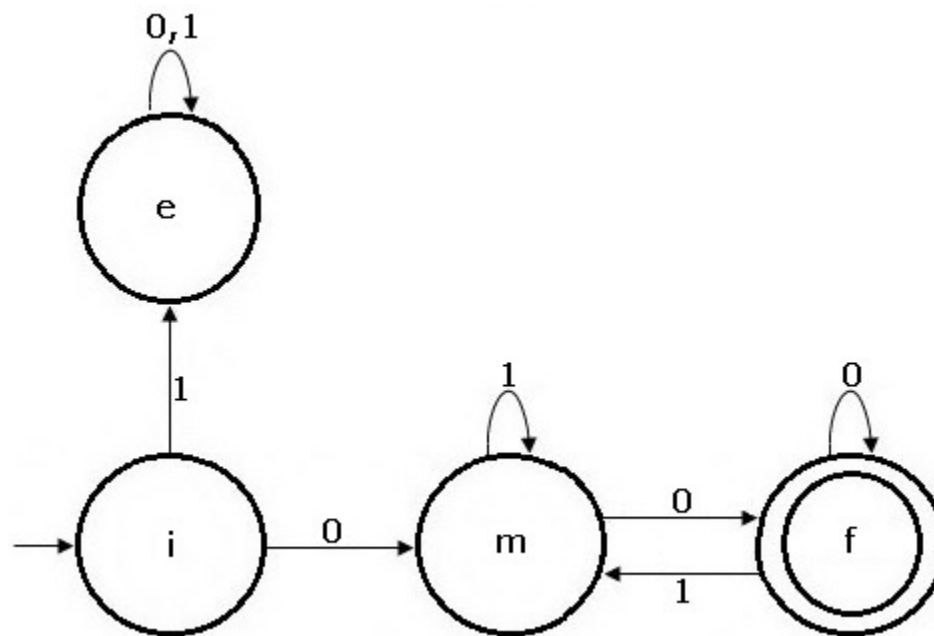
3)



Um olhar para o futuro

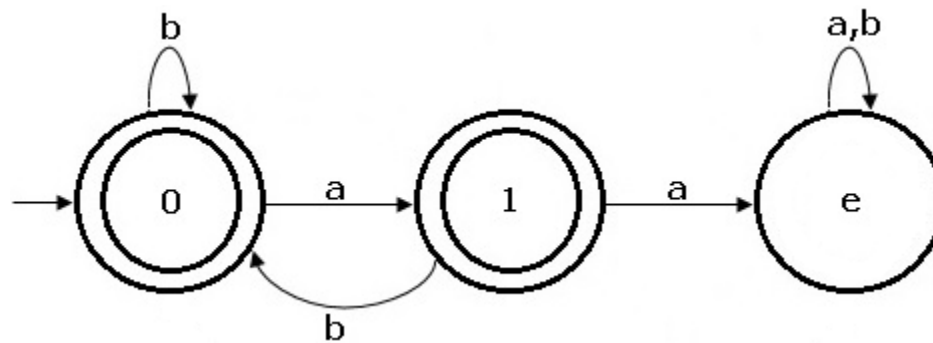
AF MÍNIMO

3)



Respostas

4)



Obrigado.

joapauloaramuni@gmail.com
joapauloaramuni@fumec.br