Módulo 2 - Autômatos Finitos

Ricardo Vargas Dorneles

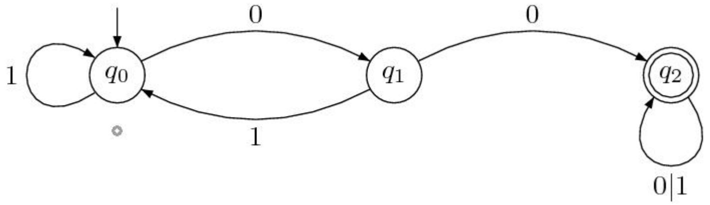
2 de março de 2023

Autômatos Finitos

- Descrição de linguagens : por geração
  por reconhecimento

- Um **Autômato Finito** (A.F.) é um reconhecedor de sentenças regulares. Dada uma sentença ele responde sim ou não se a sentença pertence à linguagem. Um autômato finito é formado por:
  - Uma fita com uma sentença a ser reconhecida
  - Um cabeçote de leitura que aponta para o símbolo corrente
  - Uma memória que armazena o estado atual
- O símbolo corrente e o estado atual determinam qual o próximo estado (transição). Se ao final o autômato está em um estado aceitável, a sentença foi reconhecida.

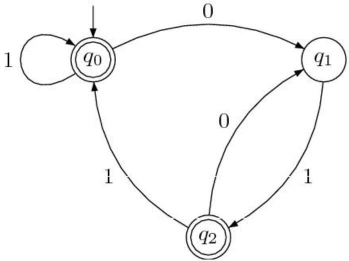
Voltar para o índice



- O autômato acima reconhecerá as sentenças que tiverem dois zeros seguidos em alguma posição, ou seja, reconhece as sentenças geradas pela expressão regular $(0|1)^*00(0|1)^*$.
- O autômato mostrado não reconhece a sentença 0110 porque ao terminar a sentença o autômato está no estado q_1 que não é um estado final.

Voltar para o índice

1. Considere o seguinte automato finito:



- Apresente a seqüência de transições e indique se o autômato reconhece ou não cada uma das sentenças:
  - a) 111010111
  - b) 10101
  - c) 01010
  - d) 1100

Definição Formal de um autômato finito

- É definido por:
  - (V, S, q_0, F, d) onde
  - V = Vocabulário
  - S = Conjunto de estados
  - q_0 = Estado inicial
  - F = Conjunto de estados finais
  - d = Função de transição no formato : d(q,a) = p

Voltar para o índice

Ex: O autômato anterior é definido pela quíntupla (V,S,q_0,F,d)

V={0,1}
S={q_0, q_1, q_2}
F={q_0, q_2}
d(q_0,0)=q_1
d(q_0,1)=q_0
d(q_1,1)=q_2
d(q_2,0)=q_1
d(q_2,1)=q_0

Voltar para o índice

- Matriz de transição : É uma forma reduzida de representar a função de transição

	Símbolos	
	0	1
q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_2	q_2

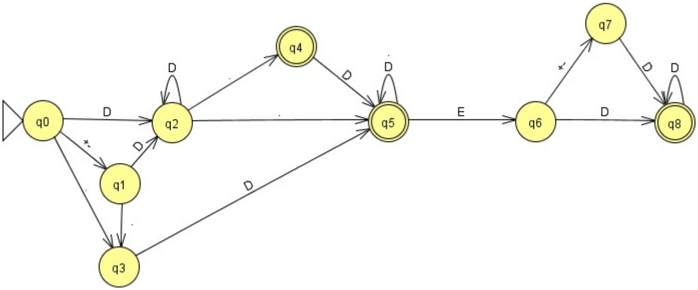
Voltar para o índice

- Faça um AF sobre o vocabulário V={0,1} que reconheça sentenças com um número par de símbolos e que contenha pelo menos dois símbolos.
- (V,S,q_0,F,d)
- V={0,1}
- S=
- F=

Voltar para o índice

- Faça um AF sobre o vocabulário V={0,1} que reconheça sentenças com um número par de zeros e um número par de uns.
- Faça AF's para reconhecer:
  - a)Números inteiros sem sinal
  - b)Números inteiros com sinal
  - c)Identificadores em pascal (iniciam por letra e só pode conter letras, dígitos e underlines
  - d)Números reais

Voltar para o índice



Voltar para o índice

- Faça um AF que reconheça sentenças sobre o alfabeto $\{0,1\}$ que não possuam prefixos onde a diferença entre os números de 0's e 1's é superior a 1.

Voltar para o índice

- Construa autômatos finitos que aceitem as seguintes linguagens sobre o alfabeto {0,1}

  a) O conjunto de todos strings que terminam em 00.
  b) O conjunto de todos os strings com 3 0's consecutivos.
  c) O conjunto de todos strings tal que a cada 3 caracteres consecutivos pelo menos dois sao 0.
  d) O conjunto de todos os strings tal que o penúltimo simbolo e 0.

Voltar para o índice

- Desenvolva autômatos finitos determinísticos que reconheçam as seguintes linguagens sobre $\Sigma = \{a, b\}$

  a)$\{$w|w possui aaa como subpalavra$\}$
  b)$\{$w|o sufixo de w é aa$\}$
  c)$\{$w|w possui número ímpar de a e b$\}$
  d)$\{$w| w possui número par de a's e ímpar de b's ou w possui número par de b's e ímpar de a's$\}$.
  e)$\{$w | o quinto símbolo da direita para a esquerda de w é a$\}$

Voltar para o índice

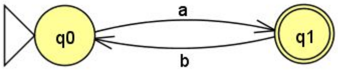
Conversão de AF para ER

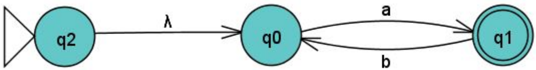
- A conversão de um autômato finito para uma expressão regular é útil porque a ER é um formalismo mais sucinto e usado em geradores de analisadores léxicos e outros programas.
- Dois algoritmos usados para fazer essa conversão são:
  - Método de eliminação de estados
  - Método de Arden
- Começaremos pelo método de eliminação de estados

Método de eliminação de estados

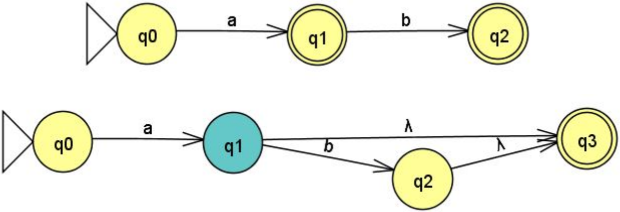
- Se baseia em ir simplificando o autômato até chegar em um autômato com um estado inicial e um final, ligados por uma transição equivalente à expressão regular.
- As simplificações no autômato são mostradas a seguir:

- O estado inicial não pode ter nenhuma transição chegando nele. Se houver, é criado um novo estado inicial e acrescentada uma transição vazia do novo estado inicial ao estado inicial original. O mesmo deve ser feito se o estado inicial for um estado final.

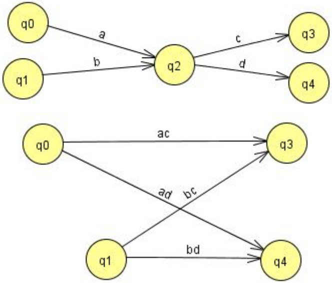




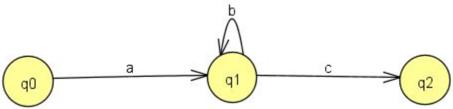
- O estado final deve ser único e não pode haver transições saindo dele. Se houver mais de um, é criado um novo estado final e puxadas transições ϵ (transições associadas a uma expressão regular ϵ) de cada estado final para o novo estado final. O mesmo ocorre se houver transições saindo do estado final.

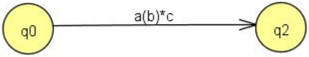


- Elimina-se cada um dos estados do autômato, exceto o inicial e o final.
- Suponha que o estado X a ser eliminado tenha N transições chegando nele e M transições saindo. Combina-se cada transição chegando com cada transição saindo, gerando NxM transições.



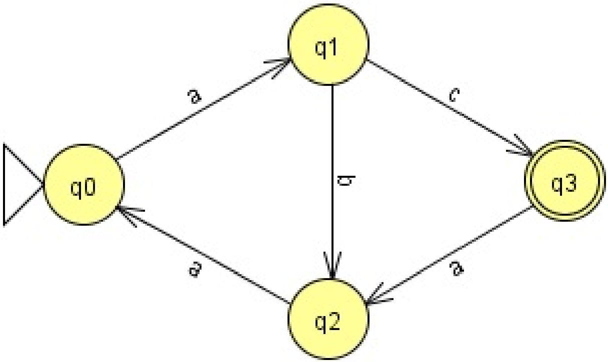
- Se o estado X tem uma transição para ele próprio, coloca-se o fecho dessa transição dentro de cada transição resultante das NxM combinações.





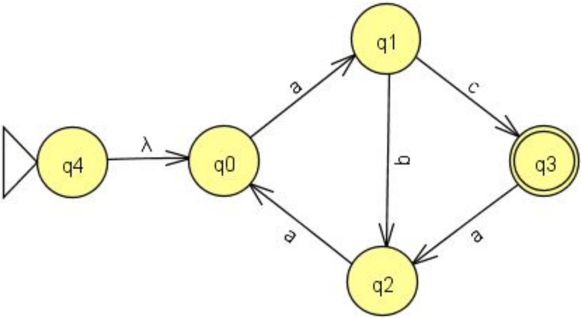
- Transições paralelas são substituídas pela transição com a união das transições paralela.

- Construa, para o autômato abaixo, a expressão regular equivalente:

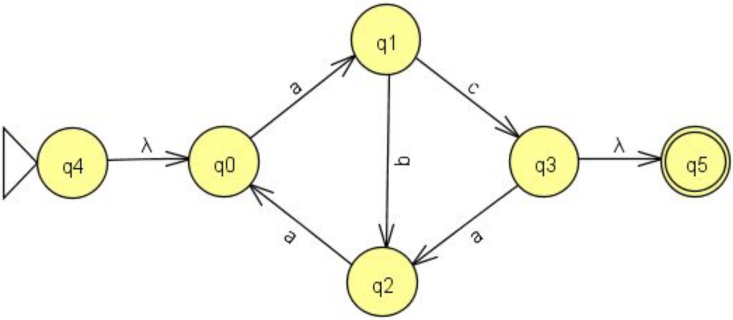


Voltar para o índice

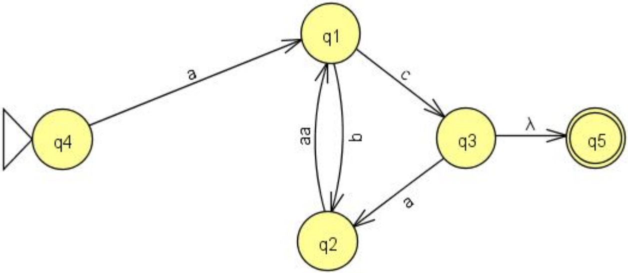
- Primeiro passo: Como o estado inicial tem uma transição chegando, gera-se um outro estado inicial:



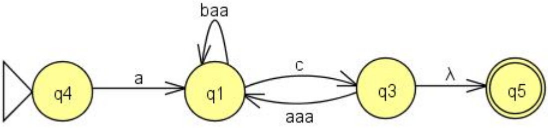
- Segundo passo: Como o estado final tem transições saindo, gera-se um outro estado final:



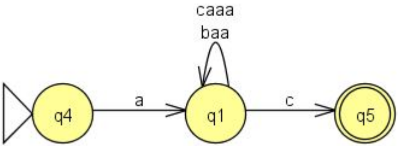
- Passo 3: Elimina-se q0 (podia ser qualquer estado). A sequência de transições λ-**a** entre q4 e q1 torna-se a transição **a** entre q4 e q1, a sequência de transições **a**-**a** entre q2 e q1 torna-se a transição **aa** entre q2 e q1, e o estado q0 é removido.



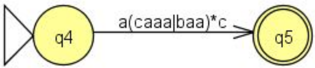
- Passo 4: Elimina-se q2 (podia ser qualquer estado. Escolhi o q2 porque tem menos transições). A sequência de transições b-aa a partir de q1 torna-se a transição **baa** de q1 para q1. A sequência de transições **a-aa** entre q3 e q1 torna-se a transição **aaa** entre q3 e q1, e o estado q2 é removido.



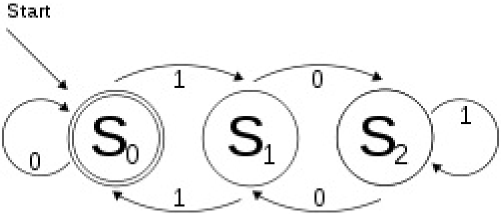
- Passo 5: Elimina-se q3 (podia ser o q1. Escolhi o q3 porque não tem transição para ele mesmo). A sequência de transições c-λ a partir de q1 torna-se a transição **c** de q1 para q5. A sequência de transições **c-aaa** de q1 para q1 torna-se a transição **caaa** de q1 para q1, e o estado q3 é removido.



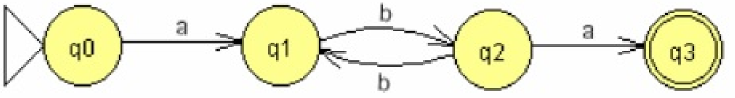
- Passo 6: Elimina-se q1. A sequência de transições a-c torna-se a transição **a(caaa|baa)*c** de q4 para q5 e essa transição é a expressão regular equivalente.



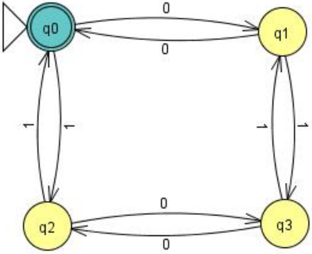
O autômato abaixo reconhece números binários múltiplos de 3. Encontre a expressão regular equivalente:



Encontre a expressão regular equivalente ao autômato a seguir:



- Encontre uma expressão regular equivalente ao autômato a seguir:

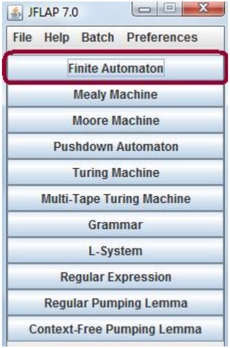


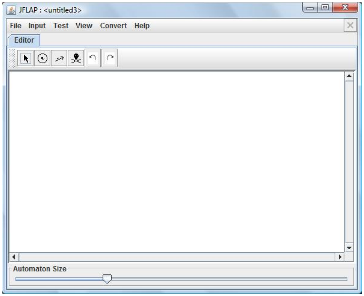
JFLAP

- O JFLAP é um pacote de ferramentas gráficas que podem ser utilizadas como auxílio no aprendizado dos conceitos básicos de Linguagens Formais e Teoria de Autômatos.
- Suas principais funcionalidades são:
  - Criação de autômatos finitos, expressões regulares e gramáticas.
  - Conversão entre os formalismos citados acima.
  - Verificação se sentenças dadas pertencem à linguagem definida.
- Disponível em:
  - http://www.jflap.com-about.com/download.html
  - http://www.jflap.org
  - Acervo da turma no UCS Virtual

Voltar para o índice

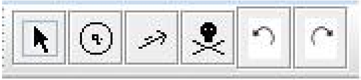
Criação de Autômatos





Voltar para o índice

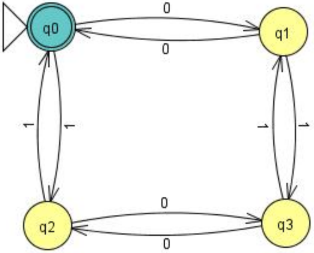
Menu do editor de autômatos



1. Definição de atributo de estado
2. Criação de estado
3. Inclusão de transição
4. Remoção de elemento
5. Desfazer
6. Refazer

Voltar para o índice

Crie o autômato a seguir no editor do JFLAP:



Voltar para o índice