Material de autoria da Universidade Paulista para a disciplina Linguagens Formais e Autômatos.

A Classe das Linguagens Regulares

Introdução

O estudante desta disciplina já deve ter se debruçado sobre as tarefas de desenvolver um programa em uma Linguagem de Programação e obter o resultado do processamento do mesmo em um computador.

As Linguagens de programação como C#, Java e outras, são notações para se descrever computações por pessoas e para as máquinas. Os sistemas de software que fazem a tradução das linguagens de programação para os computadores são denominados compiladores. (Aho et al., 2008).

O compilador antes de executar a função de tradução propriamente dita, realiza três tarefas, a saber: Análise Léxica, Análise Sintática e Análise Semântica.

Na Análise Léxica, o compilador, entre outros procedimentos, lê um fluxo de caracteres que compõem o programa fonte (normalmente descrito em uma linguagem de programação de alto nível) e os agrupa em sequências significativas denominadas lexemas. Estes lexemas dizem respeito a variáveis, constantes, palavras reservadas, operadores aritméticos e lógicos, etc.

Na Análise Sintática, o compilador verifica se estruturas sintáticas, por exemplo, uma **classe**, os comandos **if** e **while,** etc., estão corretas.

A Análise Semântica realiza tarefas como a verificação de concordância entre a declaração de tipos das variáveis, e uso das mesmas, verificação da concordância entre o número de parâmetros na declaração de um método de uma classe e o número de argumentos no uso do método por um objeto, etc.

Assim, a Análise Léxica, Sintática e Semântica, estão associadas ao processamento das componentes Regular, Livre de Contexto e Dependente de Contexto, respectivamente, da Linguagem de Programação.

Noam Chomksy, estudioso das Linguagens Naturais, propôs uma classificação de linguagens. Tal classificação é conhecida como a hierarquia de Chomsky e distingue quatro classes, a saber:

- Linguagens do tipo 0 ou recursivamente enumeráveis;
- Linguagens do tipo 1 ou sensíveis a contexto;
- Linguagens do tipo 2 ou livres de contexto ;
- Linguagens do tipo 3 ou regulares;

Linguagens Formais e Autômatos é um tópico que há muitos anos faz parte dos currículos de computação e consiste basicamente no estudo da hierarquia de Chomsky. O estudo das componentes regular e livre de contexto são de particular interesse para o projeto e implementação das Linguagens de Programação, ou seja, para o projeto de Compiladores.

Na Unidade II deste curso serão apresentadas as Linguagens Livres de Contexto e também se fará uma breve explanação sobre as Linguagens Sensíveis a Contextos e as Recursivamente Enumeráveis.

UNIDADE I

1 ALFABETOS, CADEIAS, LINGUAGENS E GRAMÁTICAS

Comentado [V1]:

2.1 Definições de Alfabeto, Cadeias, Linguagens

Alfabeto é um conjunto finito de símbolos. Cada símbolo é portanto, uma unidade atômica empregada na construção de cadeias e se trata de um conceito primitivo, sendo a sua representação visual irrelevante.

São exemplos de alfabetos:

Σ_1 = { a, b, o, d}

Σ_2 = {0, 1}

Σ_3 = {$, @, #}

$\Sigma_4 = \{ \ \} = \varnothing$

Uma **cadeia de caracteres**, é uma sequência **finita** de símbolos de um alfabeto justapostos.

Considere o alfabeto Σ_1 = { a, b, o, d}.

São exemplos de cadeias definidas sobre o alfabeto Σ_1, as seguintes sequências de caracteres (ou símbolos) justapostos : abba, oba, dado, boda, abad, boa, boba, bddb, aaaa, a.

Uma **cadeia vazia** é uma cadeia sem símbolos e é representada pelo símbolo ε.

O **comprimento de uma cadeia** ω*,* é o número de símbolos justapostos que se apresentam na mesma. Representa-se como $|\omega|$. Exemplos:

Para ω = aaaa tem-se que $|\omega| = 4$.

Para ω = a, tem-se que $|\omega| = 1$.

Para $\omega = \varepsilon$, tem-se que $|\omega| = 0$.

Duas cadeias definidas sobre o mesmo alfabeto podem ser combinadas e formar uma terceira a partir da operação de **concatenação**. A concatenação das cadeias *v* e ω é representada por $v\omega$ e é formada pela justaposição de *v* e ω.

<u>Exemplo:</u>

Seja o alfabeto Σ = {a, s, m, o, r}

Seja ω_1 = amor e ω_2 = as. Tem-se que $\omega_1\omega_2$ = amoras.

A operação de concatenação é associativa, ou seja (uv)w = u(vw) quaisquer que sejam as cadeias u, v e w.

Uma cadeia v é uma **subcadeia** de uma cadeia ω se e somente se há cadeias x e y tal que ω= xvy.

Um **prefixo** (respectivamente, **sufixo**) de uma cadeia é qualquer seqüência de símbolos inicial (respectivamente, final) da cadeia.

Considere, por exemplo, ω = amoras. São prefixos da cadeia ω: a, am, amo, amor, amora e a própria cadeia amoras. São sufixos de ω: s, as, ras, oras, moras e a própria cadeia amoras.

Para cada cadeia ω e cada número natural i, define-se $\boldsymbol{\omega^i}$ conforme se segue:

$$\omega^0 = \varepsilon \text{ (cadeia vazia)}$$

$$\omega^{i+1} = \omega^i \omega$$

Exemplo: Sejam o alfabeto $\Sigma = \{a, b\}$ e a cadeia ω = ba. Tem-se que:

$\omega^3 = \omega\omega\omega$=bababa

$\omega^1 = \omega$ = ba

$\omega^0 = \varepsilon$

O **reverso de uma cadeia** ω, denotada por ω^R, é definido como:

(1) se $\omega = \varepsilon$, então $\omega^R = \omega = \varepsilon$. Note-se que neste caso, o comprimento da cadeia ω é 0.

(2) Se ω é uma cadeia de comprimento n+1 > 0, então ω = ua para algum $a \in \Sigma$ e w e $w^R = au^R$.

Exemplos:

Sejam o alfabeto $\Sigma = \{a, b\}$ e ω = ba. Tem-se que ω^R = ab.

Sejam o alfabeto $\Sigma = \{a, i, m, r\}$ e ω = rima. Tem-se que ω^R = amir

Linguagem formal é um conjunto finito ou infinito, de cadeias de comprimento finito, sobre um alfabeto finito e não vazio..

Seja o alfabeto $\Sigma = \{0, 1\}$. A seguir, são apresentados exemplos de Linguagens, cujas cadeias são definidas sobre o alfabeto Σ.

a) $L_1 = \emptyset$
b) $L_2 = \{\varepsilon\}$
c) $L_3 = \{1, 01, 10, 11, 111\}$

Além das operações definidas para conjuntos apresentadas no capítulo 1, definem-se a seguir as operações de concatenação e fechamentos;

2.2 Gramática: dispositivo gerador de uma Linguagem.

Gramáticas são dispositivos geradores das cadeias que pertencem a uma Linguagem.

Formalmente, uma gramática é uma quádrupla $G = (V, \Sigma, P, S)$, onde:

V = conjunto finito de símbolos não-terminais.

Σ = conjunto finito de símbolos terminais da gramática. Este conjunto também é denominado **alfabeto.**

$S \in V$ e é denominado símbolo inicial ou **raiz** da gramática;

P = conjunto de produções ou regras de substituição da gramática.

Uma **regra de produção** é representada da seguinte forma:

$$\alpha \rightarrow \beta \text{ onde, } \alpha \in (V \cup \Sigma)^+ \text{ e } \beta \in (V \cup \Sigma)^*.$$

Naturalmente, $\alpha \in (V \cup \Sigma)^+$ informa que no lado esquerdo da produção deve existir um ou mais símbolos, terminais ou não-terminais. Analogamente, a expressão $\beta \in (V \cup T)^*$ indica que no lado direito da produção podem figurar quaisquer cadeias de símbolos terminais, não-terminais e até mesmo isoladamente, a cadeia vazia.

Exemplo: Seja $G_1 = (V, \Sigma, P, S)$, onde:

$V = \{S, X, Y\}$
$\Sigma = \{a, 0, 1\}$
$P = \{ S \to aX$
$X \to a X \mid 0X \mid 1 X \mid \varepsilon \}$
S é o símbolo inicial da gramática.

Observação: O símbolo “|” é traduzido como “ou”. Assim sendo, escrever

$$X \to a X \mid 0X \mid 1 X \mid \varepsilon$$

equivale a representar o seguinte conjunto de regras:
$X \to a X$; $X \to 0 X$; $X \to 1 X$ e $X \to \varepsilon$;

Exemplo:

$G_2 = (V, \Sigma, P, S)$, onde:
V = {Expressao, Termo, Fator}
$\Sigma = \{x, y, +, *\}$

Expressao é o símbolo inicial da gramática.
O conjunto das produções é representado por:

P = { Expressao → Expressão + Termo
Expressao→Termo
Termo → Termo * Fator
Termo → Fator
Fator → x
Fator → y
}

3 LINGUAGENS REGULARES – PARTE I

Teorema 1: Se L é uma linguagem gerada por uma Gramática Regular, então L é uma Linguagem Regular.
Teorema 2: Se L é uma Linguagem Regular, então existe G, Gramática Regular que gera L.
Exemplo: Considere-se A Gramática G_3 Linear à Direita

$G_3 = (V, \Sigma, P, S)$, onde:
$V = \{S, X, Y, F\}$
$\Sigma = \{a, b\}$
S é o símbolo inicial.
$P = \{ S \to aX$
$X \to bF$
$F \to aY \mid \varepsilon$
$Y \to bF\}$

3.3 Expressões Regulares

Exemplo: A expressão regular a* representa a linguagem:

$$L_1 = \{\varepsilon, a, aa, aaa, ...\}$$

Exemplo: A expressão regular $(b|a^*)$ representa a linguagem :

$L_2 = \{\varepsilon, b, a, aa, aaa, ...\}$

Exemplo: A expressão regular (ba*) representa a linguagem:

$L_3 = \{b, ba, baa, baaa, baaaa, ...\}$

Exemplo: A expressão regular (a | ba*) representa a linguagem:

$L_4 = \{a, aa, aaa, aaaa, b, ba, baa, baaa, baaaa ...\}$

Exemplo: A expressão regular a(b|c)* representa a linguagem:

$L_5 = \{a, ab, ac, abb, acc, abc, acb,abbc ...\}$

Exemplo: A expressão regular (a | ba)* representa a linguagem:

$L_6 = \{\varepsilon, ba, a, baa, aba, baba, aa,...\}$

Exemplo: A expressão regular $(ab)^+$ representa a linguagem:

$L_7 = \{ab, abab, ababab, ...\}$

3.4 Autômatos Finitos – Dispositivos Reconhecedores de Linguagens Regulares

Exemplo: Seja M um autômato finito determinístico (Q, Σ, g, q0, F), onde:

Q = {q0, q1, q2}

Σ = {a, b, c}

q0 é o estado inicial

F = {q2}

e a função g pode ser apresentada como a tabela abaixo:

g	a	b	c
q0	q1	-	-
q1	-	q2	-
q2	-	-	q2

Pode-se também denotar a função g, como:

g= {((q0,a), q1), ((q1,b), q2), ((q2, c), q2)

Ainda, a função g, pode ser especificada como:

g(q0, a) = g1; g(q1, b) = q2; g(q2, c) = q2.

M pode ser representado como na figura seguinte.

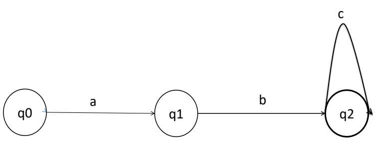


Figura 3.3 – Autômato Finito Reconhecedor da Linguagem

$L_1 = \{\omega \mid \omega = abc^*\}$

Exemplo: Sejam Σ = { x, y} e L, uma linguagem definida sobre o alfabeto Σ, tal que

L= { ω | ω apresenta a sub-cadeia xyxy}

A representação algébrica para o autômato finito é:

Q = {q0, q1, q2, q3, q4}

Σ = {x, y}

g = {((q0,x), q1), {((q0,y), q0), ((q0,x), q1), ((q1,x), q1), ((q1,y), q2), ((q2,x), q3), ((q2,y), q0), ((q3,x), q1), ((q3,y), q4), ((q4,x), q4), ((q4,y), q4)}

q0 é o estado inicial.

F = {q4}

A representação gráfica do autômato é como se ilustra na figura seguinte:

.

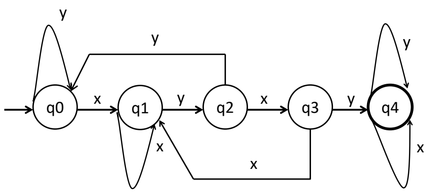


Figura 3.4 – Autômato Finito para reconhecimento de

L= { ω | ω apresenta a sub-cadeia xyxy}

<Exercício de aplicação _ início>

Identifique qual é a Linguagem reconhecida pelo autômato finito ilustrado a seguir. Apresente também a representação algébrica do autômato.

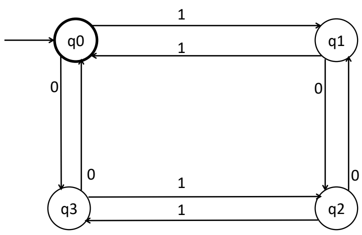


Figura 3.5 – Autômato Finito

<Exercício de aplicação _ fim>

Exemplo: Considere-se o autômato M, representado em sua forma algébrica como:

M = (Q, Σ, g, q0, F)

Onde Q = {q0, q1, q2, q3, q4, qf}

Σ = {a, b}

q0 é o estado inicial e F = {qf}

Tem-se que:

g(q0, a) = q1	g(q0,b) = q3	g(q1,a) = q1
g(q1, b) = q2	g(q2,a) = q3	g(q2, b) = q4
g(q3, a) = qf	g(q3, b) = q4	g(q4, b) = q3

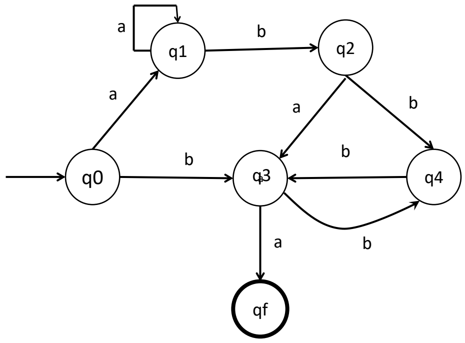


Figura 3.6 – Exemplo de autômato finito

<u>Exemplo:</u> Considere a seguinte gramática regular:

G = (V, Σ, P, S) onde:

V = {A, B, C, D};

Σ = {a,b};

A é o símbolo inicial da gramática;

P = { A $\rightarrow$ bA | a B;

B $\rightarrow$a C |bA;

C $\rightarrow$a D | b A

D $\rightarrow$ bA | aD | ε }

Para se obter o autômato correspondente, tem-se que:

- A é o estado inicial
- Q = {A, B, C, D}
- Σ = {a,b}
- F = {D}

A função g é dada pode ser obtida a partir das produções da gramática G. Considere-se a seguinte tabela.

Produção	Transição
A $\rightarrow$ bA	g(A, b) = A
A $\rightarrow$ aB	g(A, a) = B
B $\rightarrow$ aC	g(B, a) = C
B $\rightarrow$ bA	g(B, b) = A
C $\rightarrow$ aD	g(C, a) = D
C $\rightarrow$ bA	g(C, b) = A

D → bA	g(D, b) = A
D → aD	g(D, b) = D

g = {(g(A,b),A), (g(A,a),B), (g(B,a),C), (g(B,b),A), (g(C,a),D), (g(C,b),A), (g(D,b),A), (g(D,a),D), (g(D,ε),D)}
O autômato é apresentado na figura abaixo:

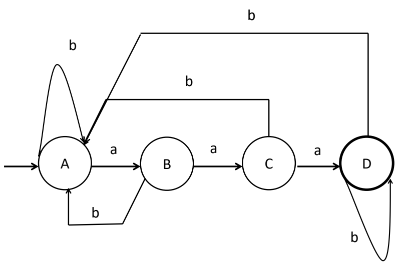


Figura 3.10 – Exemplo de autômato finito

3.6 Obtenção da Gramática Regular a partir de Autômatos Finitos.

Transição	Produção
-	$Q_f \rightarrow \varepsilon$
$g(q_i, a) = q_k$	$Q_i \rightarrow aQ_k$

Exemplo: Considere-se autômato, representado na figura abaixo.

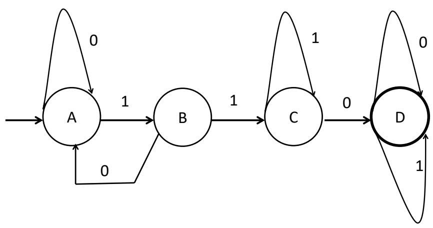


Figura 3.11 – Autômato finito reconhecedor da Linguagem
$L(M) = \{\omega \mid \omega = (0|1)^*11^+0\ (0|1)^*\}$

P = { A → 0A | 1 B ; B → 1C | 0A; C → 1C | 0D; D → 0D | 1D | ε }

Note-se a correspondência entre os estados do autômato e os símbolos não-terminais da gramática. Em particular, A é o símbolo inicial da gramática.

3.7 Equivalência entre autômatos finitos não-determinísticos e determinísticos;

Dois autômatos finitos M1 e M2 (determinísticos ou não-determinísticos) são equivalentes se e somente se L(M1) e L(M2).

A Classe dos Autômatos Finitos Determinísticos é equivalente à Classe dos Autômatos Finitos Não-Determinísticos.

Exemplo: Seja a linguagem L = { ω | ω = a (b|a)* bc} definida sobre o alfabeto Σ = {a,b,c}. O autômato finito não-determinístico M_1 reconhecedor desta linguagem é apresentado na figura abaixo:

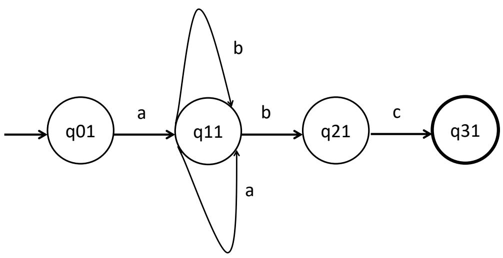


Figura 3.13 – Exemplo de autômato finito não-determinístico

A representação algébrica de M_1 é dada por:

Q = {q01, q11, q21, q31}

F = {q31}

Σ = {a, b, c}

O estado inicial é q01 e a função estendida g_1 é dada por:

g_1((q10,a)) = {q11}

g_1((q11,a)) = {q11}

g_1((q11,b)) = {q11, q12}

g_1((q12,c) = {q13}

Tem-se que:

q02 = E(q01)= q01

g_2({q01},a) = {q11}

g_2({q11},b) = {q11, q21}

g_2({q11,q21},a) = {q11}

g_2({q11,q21},b) = {q11,q21}

g_2({q11,q21},c) = {q31}

Os estados do autômato M_2 podem ser nomeados como:

q_{02}, q_{12} = {q11}, q_{22} = {q11, q21} e q_{32} = {q_{31}}.

A representação gráfica de M_2 é apresentada a seguir.

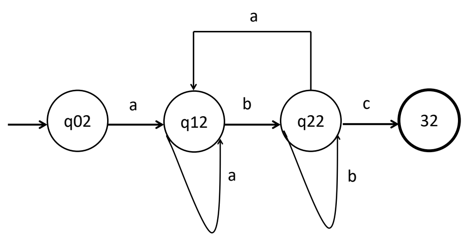


Figura 3.14 – Exemplo de autômato finito determinístico

4. LINGUAGENS REGULARES - 2

4.1 Minimização de Estados.

Os autômatos finitos determinísticos são dispositivos que recohecem linguagens regulares. Os algoritmos associados aos mesmos apresentam desempenho em consumo de espaço de memória dependente do número de estados. Neste sentido, é importante que o autômato apresente o menor número de estados possíveis. Existe teorema que garante a minimização de estados de um autômato finito M qualquer. Sejam considerados os teoremas enunciados seguidamente.

Teorema: Seja M = (Q, Σ, g, q0, F) um autômato finito. Então existe um autômato finito M' = (Q', Σ, g', q0, F') que possui o menor número possível de estados, tal que L(M) = L(M').

Teorema: Unicidade do Autômato Mínimo: O autômato finito mínimo determinístico de uma linguagem é único, a menos de ismorfismo.

Para se aplicar o algoritmo de minimização, o autômato finito:

a) deve ser determinístico;
b) não pode apresentar estados inacessíveis;
c) sua função de transição g deve ser total, ou seja, a partir de qualquer estado são previstas transições para todos os símbolos do alfabeto.

Caso o autômato não satisfaça a estas condições, faz-se necessário gerar um autômato equivalente. Para tanto, deve-se:

a) gerar um autômato finito determinístico a partir de um autômato finito não-determinístico;
b) eliminar os estados inacessíveis e suas correspondentes transições;
c) transformar a função de transição g em total, inserindo um estado auxiliar q_{aux} e incluindo as transições não previstas para este estado.

Eliminação dos estados inacessíveis:

Estados inacessíveis são aqueles para os quais não existe no autômato qualquer caminho formado por transições válidas, que permita atingi-los a partir do estado inicial do autômato.

O conjunto de estados acessíveis pode ser definido como o fechamento de {q0}, segundo a relação:

{(p, q) | g(p, a) =q para a ∈ Σ}.

Em outras palavras, o conjunto de estados **acessíveis**, pode ser obtido, a partir deste algoritmo:

1 – Inicalizar o conjunto EA, como EA = {q0}

2 - Para **todo estado** p, p ∈ EA e para **todo símbolo** a ∈ Σ, tal que q = g(p,a) ∉ EA, inserir q em EA.

Exemplo: Considere-se o autômato M = (Q, Σ, g, q0, F), onde:

Q = {q1, q2, q3, q4, q5, q6, q7, q8} (conjunto de estados).

F = {q1, q3, q7} (conjunto de estados finais);

Σ = {x, y} (alfabeto de entrada);

q1 é o estado inicial;

A função programa g é dada por:

g = {((q1,x), q2), {((q1,y), q4), {((q2,x), q5), {((q2,y), q3), {((q3,x), q2), ((q3,y), q6), ((q4,x), q1), ((q4,y), q5), ((q5,x), q5), ((q5,y), q5), ((q6,x), q3), ((q6,y), q5, ((q7, x), q6), ((q7,y),q8), ((q8,x), q7), (q8, y), q3)}.

Pede-se obter o conjunto **EA**, ou seja o conjunto de estados acessíveis do autômato.

O algoritmo apresentado, inicializa o conjunto EA, com o estado inicial. Tem-se, portanto que:

EA = {q1}

A próxima etapa no levantamento dos estados acessíveis consiste em identificar para **todo estado** p, p ∈ EA e para **todo símbolo** a ∈ Σ, o estado q tal que q = g(p,a).

Assim, para prosseguir o exemplo, representa-se a função g, de forma tabular de forma que mais facilmente sejam identificados os pares de estados possíveis (p,q).

	x	y
q1	q2	q4
q2	q5	q3
q3	q2	q6
q4	q1	q5
q5	q5	q5
q6	q3	q5
q7	q6	q8
q8	q7	q3

O único estado que pertence a EA é q1. Como o alfabeto Σ, é binário, há que se considerar apenas os pares (q1,x) e (q1, y). Tem-se que g(q1,x) = q2 e g(q1, y) = q4. Os estados q2 e q4, passam a pertencer ao conjunto EA. Tem-se que :

EA = {q1, q2, q4}

As próximas iterações, portanto dizem respeito à consideração dos pares: (q2,x), (q2, y), (q4,x), (q4, y). Tem-se que:

g(q2, x) = q5, g(q2, y) = q3, g(q4,x) = **q1**, g(q4, y) = **q5**

Desta forma, os estados q5, q3 devem ser acrescentados aos elementos de EA. Tem-se que:

EA = {q1, q2, q4, q3, q5}

A iteração seguinte consiste em considerar os pares (q3,x), (q3, y) e (q5,x) e (q5, y), recém inseridos em EA. Tem-se que:

g(q3, x) =**q2**, g(q3, y) = q6, g(q5,x) = **q5**, g(q5, y) = **q5**.

Observe-se que o único estado a inserir é q6, visto que q2 e q5 pertencem a EA. Tem-se que:

EA = {q1, q2, q4, q3, q5, q6}

Analogamente, sejam considerados os pares (q6,x) e (q6, y).

Tem-se que: g(q6, x) = **q3** e g(q6, y) = **q5**

Constata-se que não há mais estados a inserir em EA.

Portanto, o conjunto de estados acessíveis coincide com EA obtido na última iteração do algoritmo. Naturalmente, o conjunto dos estados inacessívies é:

EA'= Q – EA = {q7, q8}.

Como se pode constatar em autômato finito, podem ocorrer estados inacessíveis e estes devem ser eliminados para se minimizar os estados.

A eliminação dos estados não-alcançáveis, resulta no seguinte autômato:

M = (Q, Σ, g, q0, F), onde:

Q = {q1, q2, q3, q4, q5 } (conjunto de estados).

F = {q1, q3 } (conjunto de estados finais);

Σ = {x, y} (alfabeto de entrada);

q1 é o estado inicial;

A função programa g é dada por:

g = {((q1,x), q2), {((q1,y), q4), {((q2,x), q5), {((q2,y), q3), {((q3,x), q2), ((q3,y), q6), ((q4,x), q1), ((q4,y), q5), ((q5,x), q5), ((q5,y), q5), ((q6,x), q3), ((q6,y), q5)}.

Para se prosseguir com o algoritmo de minimização, constrói-se uma tabela relacionando os estados distintos, onde cada par de estados ocorre somente uma vez.

q2					
q3					
q4					
q5					
q6					
	q1	q2	q3	q4	q5

Em seguida, marcam-se estados trivialmente não-equivalentes.

q2	x				
q3					
q4	x		x		
q5	x		x		
q6	x		x		
	q1	q2	q3	q4	q5

Finalmente, marcam-se os estados não-equivalentes:
análise do par (q1, q3)
g(q1, x) = q2
g(q1, y) = q4
g(q3, x) = q2
g(q3, y) = q6.
Nada se pode afirmar a respeito do par (q1, q3). Pendência do par (q4, q6).
análise do par (q2, q4)
g(q2, x) = q5
g(q4, x) = q1
g(q2, y) = q3
g(q4, y) = q5
Os pares q1 e q5, bem como os pares q3 e 5 não são equivalentes, portanto o par (q2, q4) deve ser marcado.

q2	x				
q3					
q4	x	x	x		
q5	x		x		
q6	x		x		
	q1	q2	q3	q4	q5

análise do par (q2, q5)
g(q2, x) = q5
g(q2, y) = q3
g(q5, x) = q5
g(q5, y) = q5.
O par (q3, q5) está marcado, portanto o par (q2, q5) deve ser marcado.
Análise do par (q3, q6)
g(q3, x) = q5
g(q3, y) = q3
g(q6, x) = q3
g(q6, y) = q5.
O par (q3, q5) está marcado, portanto o par (q2, q6) deve ser marcado.
análise do par (q4, q5)
G(q4, x) = q1

G(q4, y) = q5
G(q5, x) = q5
G(q5, y) = q5.
O par (q1, q5) está marcado, portanto o par (q4, q5) deve ser marcado.
análise do par (q4, q6)
G(q4, x) = q1
G(q4, y) = q5
G(q6, x) = q3
G(q6, y) = q5.

Nada se pode afirmar sobre o par (q4, q6) pois o par (q1, q3) não estão marcados.

análise do par (q5, q6)
G(q5, x) = q5
G(q5, y) = q5
G(q6, x) = q3
G(q6, y) = q5.
O par (q3, q5) está marcado, portanto o par (q5, q6) deve ser marcado.

Q2	x				
Q3		x			
Q4	x	x	x		
Q5	x	x	x	x	
Q6	x	x	x		x
	Q1	Q2	Q3	Q4	Q5

Q4 e q6 são equivalentes.
Q1 e q3 são equivalentes.
A próxima etapa consiste na unificação dos estados equivalentes;

g	x	y
Q13	Q2	Q46
Q2	Q5	Q13
Q46	Q13	Q5
Q5	Q5	Q5

Ao final, eliminam-se os estados inúteis: q5 é inútil

g	x	y
Q13	Q2	Q46
Q2	-	Q13
Q46	Q13	-

<Resumo_Início>

Alfabeto é um conjunto finito de símbolos.

Linguagem formal é um conjunto de cadeias definidas sobre um alfabeto.

A Gramática é um dispositivo gerador da Linguagem. Define-se formalmente a Gramática como uma quádrupla G = (V, Σ, P, S), onde V é um conjunto de símbolos não-terminais, T é um conjunto de símbolos terminais e coincide com o alfabeto da Linguagem e S é o símbolo inicial da gramática. P é o conjunto de regras ou produções.

A **aplicação** de uma **regra de produção** $\alpha \rightarrow \beta$ é denominada **derivação**

Noam Chomsky classificou as Linguagem em quatro tipos, a saber: Linguagens Regulares, Linguagens Livres de contexto, Linguagens Recursivas e Linguagens Recursivamente Enumeráveis.

As Linguagens Regulares são geradas pelas Gramáticas Regulares.

As Gramáticas Regulares podem ser de dois tipos: Linear à Direita e Linear à Esquerda.

Uma Gramática é Linear à Direita, se todas as produções são da forma:

$A \rightarrow \omega B$ ou $A \rightarrow \omega$., onde A, B $\in$ V e $\omega \in \Sigma^*$

Uma Gramática Linear à Esquerda, se todas as produções são da forma:

$A \rightarrow B\omega$ ou $A \rightarrow \omega$. A, B $\in$ V e $\omega \in \Sigma^*$

São enunciados dois teoremas que relacionam Linguagens Regulares e gramáticas regulares:

- Se L é uma linguagem gerada por uma Gramática Regular, então L é uma Linguagem Regular.
- Se L é uma Linguagem Regular, então existe G, Gramática Regular que gera L.

Autômatos Finitos são dispositivos reconhecedores das Linguagens Regulares.

Os Autômatos Finitos podem ser determinísticos ou não-determinísticos, podendo estes últimos apresentarem-se com transições em vazio.

A partir de uma Gramática Regular é possível obter um autômato finito. Também é possível obter o autômato finito que reconhece uma linguagem, a partir de sua gramática.

É provado que a classe dos autômatos finitos determinísticos é equivalente à classe dos autômatos finitos não-determinísticos. Em outras palavras, se por uma lado o projeto de autômatos finitos não-determinísticos é mais intuitivo, por outro lado, o não-determinismo não aumenta o poder de reconhecimento de linguagens regulares.

Existe um algoritmo para construir o autômato finito determinístico mínimo, ou seja, o autômato com menor número de estados. O autômato finito determinístico de uma linguagem é único, a menos de isomorfismo.

Há que se salientar que o lema conhecido como Lema do Bombeamento permite estudar as propriedades de uma linguagem regular.

As Linguagens regulares, apresentam, além do formalismo gerador e do formalismo reconhecedor, o formalismo denotacional conhecido como expressões regulares.

A definição de Autômato Finito pode ser estendida, de forma que, além da informação fornecida por estes dispositivos, seja aquela de aceitar ou não uma palavra cadeia de entrada, o dispositivo gera cadeias de saída.

<Exercícios início>

Questão 1: Seja o alfabeto Σ = {*, 0, @, !}. Assinale a alternativa que não apresenta uma palavra definida sobre o alfabeto Σ

a)0*0*!

b)0@@

c)!!!!!!!

d)!

e)exclamação.

Resposta correta: alternativa e)

Resolução do exercício

A) Alternativa incorreta.

Justificativa: A palavra é formada a partir dos símbolos 0, * e ! que pertencem ao alfabeto

B) Alternativa incorreta

Justificativa: A palavra é formada a partir dos símbolos 0 e @, que pertencem ao alfabeto

C) Alternativa incorreta

Justificativa: A palavra é formada a partir do símbolo ! $\in \sum$

D) Alternativa incorreta

Justificativa: A palavra é formada a partir do símbolo ! $\in \sum$

E) Alternativa correta

Justificativa: A palavra exclamação não apresenta símbolos do alfabeto $\sum$.

Questão 2: Considere a gramática G = (V, $\sum$, P, S}, onde V = {S, X, Y}; $\sum$ = {a, b}, S é o símbolo inicial e P = { S $\rightarrow$ aX | b Y; X $\rightarrow$ bX | aY; Y$\rightarrow$bY | a X| ε}

Assinale a alternativa que representa uma derivação correta.

a)S$\Rightarrow$bY$\Rightarrow$bbY$\Rightarrow$bbaX$\Rightarrow$bbaaY$\Rightarrow$bbaa$\varepsilon$$\Rightarrow$bbaa

b) S$\Rightarrow$bX$\Rightarrow$bbX$\Rightarrow$bbbX$\Rightarrow$bbbaY$\Rightarrow$bbba$\varepsilon$$\Rightarrow$bbba

c) S$\Rightarrow$bX$\Rightarrow$bbY$\Rightarrow$bbaX$\Rightarrow$bbaaY$\Rightarrow$bbaa$\varepsilon$$\Rightarrow$bbaa

d) S$\Rightarrow$aX$\Rightarrow$ aε = a

e) S$\Rightarrow$bY$\Rightarrow$bYY$\Rightarrow$b.

Resposta correta: alternativa a)

Resolução do exercício

a) alternativa correta

Justificativa: São aplicadas sucessivamente as regras: S$\rightarrow$bY, Y$\rightarrow$bY, Y$\rightarrow$aX, X$\rightarrow$aY e finalmente, Y$\rightarrow\varepsilon$

b) alternativa incorreta

Justificativa: Esta alternativa supõe a aplicação da regra: S$\rightarrow$bX, que não pertence ao conjunto das produções P.

c) alternativa incorreta

Justificativa: Esta alternativa supõe a aplicação da regra: S$\rightarrow$bX, bem como da regra X$\rightarrow$bY, que não pertencem ao conjunto das produções P.

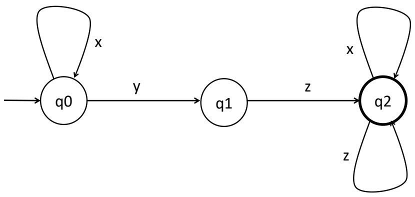
d) alternativa incorreta

Justificativa: Esta alternativa supõe a aplicação da regra: X$\rightarrow\varepsilon$, que não pertence ao conjunto das produções P.

e) alternativa incorreta:

Justificativa: Esta alternativa supõe a aplicação da regra: $Y \rightarrow YY$, que não pertence ao conjunto das produções P.

Questão 3**:** - **Considere a figura abaixo. Assinale a alternativa que *não* apresenta uma transição do autômato ilustrado.**



a)g(q0,x) = q0
b)g(q0,y)=q1
c)g(q1,z) = q2
d)g(q2,x)=q2
e)g(q2,z)=q1

Resposta correta: alternativa e)

Resolução do exercício

a) alternativa incorreta

Justificativa: existe a transição g(q0, x) = q0.

b) alternativa incorreta

Justificativa: existe a transição g(q0, y) = q1.

c) alternativa incorreta.

Justificativa: existe a transição g(q1, z) = q2.

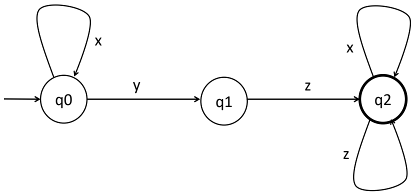
d) alternativa incorreta.

Justificativa: existe a transição g(q2, x) = q2

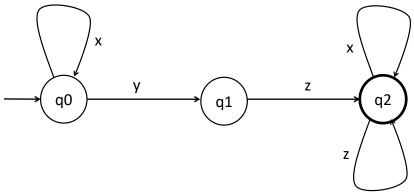
e) alternativa correta.

Justificativa: Não existe a transição g(q2, z) = q1, mas existe a transição g(q2, z) = q2.

Questão 4: **Considere o autômato representado na figura abaixo e assinale a alternativa que apresenta a palavra aceita pelo mesmo.**



a)xxy
b)xzx
c)yzy
d)yzx
e)xz
Resposta correta: alternativa d)
Resolução do exercício
a) alternativa incorreta
Justificativa: o autômato reconhece a linguagem L(w) = {w | w = x*(yz)(x|z)* e portanto xxy $\notin$ L(w).
b) alternativa incorreta
Justificativa: o autômato reconhece a linguagem L(w) = {w | w = x*(yz)(x|z)* e portanto xzx $\notin$ L(w).
c) alternativa incorreta
Justificativa: o autômato reconhece a linguagem L(w) = {w | w = x*(yz)(x|z)* e portanto yzy $\notin$ L(w).
d) alternativa correta
Justificativa: o autômato reconhece a linguagem L(w) = {w | w = x*(yz)(x|z)* e portanto yzx $\in$ L(w).
e) alternativa incorreta
Justificativa: o autômato reconhece a linguagem L(w) = {w | w = x*(yz)(x|z)* e portanto xz $\notin$ L(w).
Questão 5: Para o autômato apresentado, assinale a alternativa que contemple o conjunto de produções P da gramática G = {V, Σ, P, S} que gera a Linguagem reconhecida por esta máquina de estados. Considerar: V = {Q_0, Q_1, Q_2}, Σ = {x, y,z} e Q_0, o símbolo inicial da gramática.



a) P = { Q0 $\rightarrow$ x Q_0; $Q_1 \rightarrow$z Q_2; $Q_2 \rightarrow$ x Q_2 | z Q_2 | ε }
b) P = { Q0 $\rightarrow$ x Q_0 | y Q_1; $Q_1 \rightarrow$z Q_2; $Q_2 \rightarrow$ x Q_2 | z Q_2 | ε }
c) P = { Q0 $\rightarrow$ y Q_1; $Q_1 \rightarrow$z Q_2; $Q_2 \rightarrow$ x Q_2 | z Q_2 | ε }
d) P = { Q0 $\rightarrow$ x Q_0 | y Q_1; $Q_2 \rightarrow$ x Q_2 | z Q_2 | ε }
e) P = { Q0 $\rightarrow$ x Q_0 | y Q_1; $Q_1 \rightarrow$z Q_2; $Q_2 \rightarrow$ z Q_2 | ε }
Resposta correta: alternativa b)

Resolução do exercício

a) alternativa incorreta.
Justificativa: Falta a regra Q0 $\rightarrow$| y Q_1 no conjunto das produções.
b) alternativa correta.
Justificativa: todas as transições são representadas pelas regras correspondentes na gramática.
c) alternativa incorreta.
Justificativa: Falta a regra: Q0 $\rightarrow$ x Q_0.
d) alternativa incorreta.
Justificativa: Falta a regra $Q_1 \rightarrow$z Q_2 no conjunto das produções.
e) alternativa incorreta.
Justificativa: Falta a regra $Q_2 \rightarrow$ x Q_2 no conjunto das produções.

<u>Questão 6</u>: Considere o alfabeto Σ = {a, b}. A Linguagem L = {ω | $\omega = a^+$ (b a)$^+$}.Assinale a alternativa que apresenta uma palavra que pertença a L
a)bab
b)aabab
c)abab
d)aaba
e)bbab
Resposta correta: alternativa d)

Resolução do exercício

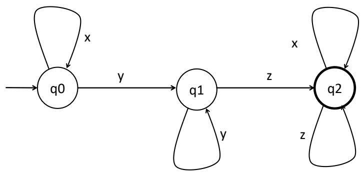
a) alternativa incorreta.
Justificativa: a expressão regular $a^+(ba)^+$ indica que no início de uma palavra pertencente à Linguagem deve ocorrer obrigatoriamente um símbolo a, o que não se sucede com a palavra “bab”.
b) alternativa incorreta.

Justificativa: a expressão regular $a^+(ba)^+$ indica que após o símbolo "b" deve se suceder um símbolo "a", o que não ocorre com a palavra "aabab".
c) alternativa incorreta.
Justificativa: a expressão regular $a^+(ba)^+$ indica que após o símbolo "b" deve se suceder, obrigatoriamente, um símbolo "a", o que não ocorre com a palavra "abab".
d) alternativa correta.
Justificativa: A palavra aaba, pertence à linguagem gerada pela expressão regular $a^+(ba)^+$, ou seja apresenta duas ocorrências do símbolo "a" e uma ocorrência da subcadeia "ba".
e) alternativa incorreta.
Justificativa: a expressão regular $a^+(ba)^+$ indica que no início de uma palavra pertencente à Linguagem deve ocorrer obrigatoriamente um símbolo "a", o que não se sucede com a palavra "bbab".
Questão 7:Considere o alfabeto $\Sigma = \{a, b\}$ e $L = \{w \mid w = a^+b^*a^*\}$. *Não* se trata de uma palavra pertencente a esta Linguagem:
a)abb
b)abba
c)abab
d)abba
e) ab
Resposta correta: alternativa c)
Resolução do exercício
a) alternativa incorreta.
Justificativa: A expressão regular gera a subcadeia abb.
b) alternativa incorreta
Justificativa: A expressão regular gera a subcadeia abba.
c) alternativa correta.
Justificativa: A expressão regular não gera a subcadeia abab, pois após a ocorrência obrigatória do símbolo a e da sequência opcional dos símbolos b, só podem figurar na cadeia símbolos "a". .
d) alternativa incorreta.
Justificativa: A expressão regular gera a subcadeia abba.
e) alternativa incorreta.
Justificativa: A expressão regular gera a subcadeia ab.
Questão 8: Considere o alfabeto $\Sigma = \{a, b\}$ e L = {w | w apresente a sub-cadeia bbb}. Assinalar a alternativa que especifique esta Linguagem, empregando o formalismo denotacional expressão regular.
a)$L(w) = \{w \mid w = (a \mid b)^* bbb\}$
b) $L(w) = \{w \mid w = (a \mid b)^* bbb\ a^+\}$
c) $L(w) = \{w \mid w = a^* bbb\}$
d) $L(w) = \{w \mid w = (a \mid b)^* bbb\ (a \mid b)^*\}$
e) $L(w) = \{w \mid w = (a \mid b)^+ bbb\ (a \mid b)^+\}$
Resposta correta: alternative d)

Resolução do exercício
a) alternativa incorreta.
Justificativa: A expressão regular produz palavras (a | b)* bbb, cujo sufixo é necessariamente bbb, ou seja a palavra termina necessariamente com esta sub-palavra.
b) alternativa incorreta:
Justificativa: A expressão regular (a | b)* bbb a^+ produz palavras com a sub-cadeia bbb, mas devem apresentar um sufixo constituído de um ou mais símbolos a.
c) alternativa incorreta
Justificativa: A expressão regular a * bbb produz a palavra bbb e as palavras com o sufixo bbb, antecedido necessariamente por um prefixo constituído de um ou mais símbolos a.
d) alternativa correta
Justificativa. A expressão regular (a | b)* bbb (a |b)*produz todas as palavras que apresentam a subcadeia bbb.
e) alternativa incorreta
Justificativa: a expressão regular $(a \mid b)^+$ bbb $(a|b)^+$, exclui a palavra bbb.
Questão 9: Considere a gramática G = (V, Σ , P, S}
V = {S, A}
Σ = {1, 0}
P = { S $\rightarrow$ 1A; A $\rightarrow$ 0A | 1A | ε}
S é o símbolo inicial.
Assinale a alternativa que representa uma derivação *incorreta.*
a)S $\Rightarrow$ 1A $\Rightarrow$10A $\Rightarrow$10ε = 10
b) S $\Rightarrow$ 1A $\Rightarrow$11A $\Rightarrow$11ε = 11
c) S $\Rightarrow$ 0A $\Rightarrow$01A $\Rightarrow$01ε = 01
d) S $\Rightarrow$ 1A $\Rightarrow$11A $\Rightarrow$111Aε = 111ε = 111
e) S $\Rightarrow$ 1A $\Rightarrow$10A $\Rightarrow$100Aε = 100ε = 100
Resposta correta: alternativa c)
Resolução do exercício
a) alternativa incorreta
Justificativa: Esta derivação supõe a aplicação das regras S$\rightarrow$ 1A, A $\rightarrow$0A e A$\rightarrow\varepsilon$.
.
b) alternativa incorreta.
Justificativa: Esta derivação supõe a aplicação das regras S$\rightarrow$ 1A, A $\rightarrow$1A e A$\rightarrow\varepsilon$.
c) alternativa correta
Justificativa: A derivação é incorreta. Supõe a aplicação da regra S $\rightarrow$ 0A, que não existe no conjunto das produções.
d) alternativa incorreta
Justificativa: : Esta derivação supõe a aplicação das regras S$\rightarrow$ 1A, duas aplicações sucessivas da regra A $\rightarrow$1A e ao final, a produção A$\rightarrow\varepsilon$.
e) alternativa incorreta.

Justificativa: Esta derivação supõe a aplicação das regras $S \rightarrow 1A$, duas aplicações sucessivas da regra $A \rightarrow 0A$ e ao final, a produção $A \rightarrow \varepsilon$.

Questão 10: Assinale a alternativa que contempla o conjunto de produções P da gramática G = {V, Σ, P, S} que gera a Linguagem reconhecida pelo autômato abaixo apresentado. Considerar: V = {Q_0, Q_1, Q_2}, Σ = {x, y, z} e Q_0, símbolo inicial.



a)P = { $Q_0 \rightarrow xQ_0 \mid y\,Q_1 \mid z\,Q_0$;$Q_1 \rightarrow yQ_1 \mid zQ_2$; $Q_2 \rightarrow xQ_2 \mid z\,Q2 \mid \varepsilon$ }

b) P = { $Q_0 \rightarrow xQ_0$;$Q_1 \rightarrow yQ_1 \mid zQ_2$; $Q_2 \rightarrow xQ_2 \mid z\,Q2 \mid \varepsilon$ }

c) P = { $Q_0 \rightarrow xQ_0 \mid y\,Q_1$;$Q_1 \rightarrow yQ_1$; $Q_2 \rightarrow xQ_2 \mid z\,Q2 \mid \varepsilon$ }

d) P = { $Q_0 \rightarrow xQ_0 \mid y\,Q_1$;$Q_1 \rightarrow yQ_1 \mid z\,Q_2$; $Q_2 \rightarrow z\,Q2 \mid \varepsilon$ }

e) P = { $Q_0 \rightarrow xQ_0 \mid y\,Q_1$;$Q_1 \rightarrow yQ_1 \mid zQ_2$; $Q_2 \rightarrow xQ_2 \mid z\,Q2 \mid \varepsilon$ }

Resposta correta: alternativa e)

Resolução do exercício

a) alternativa incorreta

Justificativa: Não existe a produção $Q_0 \rightarrow z\,Q_0$.

b) alternativa incorreta.

Justificativa: Falta a produção $Q_0 \rightarrow y\,Q_1$

c) alternativa incorreta.

Justificativa: falta a produção $Q_1 \rightarrow zQ_2$

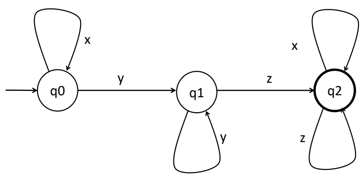
d) alternativa incorreta.

Justificativa: falta a produção $Q_2 \rightarrow xQ_2$

e) alternativa correta.

Justificativa: todas as transições presentes no autômato e somente estas, estão representadas no conjunto das produções.

Questão 11: Considere o autômato apresentado na figura abaixo e assinale a alternativa que apresenta uma palavra aceita por este dispositivo.



a)xyy
b)yyy
c)zyy
d)yz
e)xz
Resposta correta: alternativa d)
Resolução do exercício
a) alternativa incorreta.
Justificativa: O autômato reconhece as cadeias produzidas pela expressão regular $x^*(y^+z)(x|z)^*$, ou seja deve apresentarobrigatoriamente sub-cadeias, tais como yz, yyz, yyyz, etc., o que não se sucede com a palavra xyy.
b) alternativa incorreta.
Justificativa: O autômato reconhece as cadeias produzidas pela expressão regular $x^*(y^+z)(x|z)^*$, ou seja deve apresentarobrigatoriamente sub-cadeias, tais como yz, yyz, yyyz, etc., o que não se sucede com a palavra yyy.
c) alternativa incorreta.
Justificativa: O autômato reconhece as cadeias produzidas pela expressão regular $x^*(y^+z)(x|z)^*$, ou seja deve apresentarobrigatoriamente sub-cadeias, tais como yz, yyz, yyyz, etc., o que não se sucede com a palavra xyy.
d) alternativa correta.
Justificativa: yz é a palavra de menor comprimento, produzida pela expressão regular $x^*(y^+z)(x|z)^*$.
e) alternativa incorreta.
Justificativa: O autômato reconhece as cadeias produzidas pela expressão regular $x^*(y^+z)(x|z)^*$, ou seja deve apresentarobrigatoriamente sub-cadeias, tais como yz, yyz, yyyz, etc., o que não se sucede com a palavra xz.
<u>Questão 12:</u> **Considere o alfabeto Σ = {a, b} e L = {w | w = (a|b)*(aa|bb). A Linguagem L, assim representada, pode ser descrita como o conjunto de:**
a) todas as palavras contendo aa como subpalavra;
b)todas as palavras contendo exatamente dois símbolos b
c)todas as palavras que terminam com aa ou bb
d)todas as palavras que não possuem dois a consecutivos
e)todas as palavras que começam com aa.
Resposta correta: alternativa c)

Resolução do exercício

a) Alternativa incorreta.

Justificativa: A expressão regular **(a|b)*(aa|bb)** impõe que uma das subcadeias, aa ou bb sejam o sufixo da palavra, ou seja, se encontrem ao final da palavra.

b) Alternativa incorreta:

Justificativa: A expressão regular **(a|b)*(aa|bb)** impõe que uma das subcadeias, aa ou bb seja o sufixo da palavra, ou seja, se encontre ao final da palavra. Cumpre observar, que a restrição é mais forte que a ocorrência de dois símbolos "b". Estes devem ser consecutivos e se apresentarem ao final da palavra.

c) Alternativa correta.

Justificativa: A expressão regular **(a|b)*(aa|bb)** impõe que uma das subcadeias, aa ou bb sejam o sufixo da palavra, ou seja, se encontrem ao final da palavra.

d) alternativa incorreta.

A expressão regular **(a|b)*(aa|bb)** impõe que uma das subcadeias, aa ou bb seja o sufixo da palavra, ou seja, se encontre ao final da palavra. Cumpre observar, que os dois símbolos "a" devem ser consecutivos e se apresentarem ao final da palavra.

e) alternativa incorreta.

Justificativa: A restrição da ocorrência de aa, ou bb diz respeito ao final da palavra e não ao início da mesma.

<u>Questão 13:</u> **Assinale a alternativa que apresente uma aplicação para o estudo das Linguagens Regulares**

a) Implementação de um Analisador Semântico;

b)Representação da estrutura sintática da Língua Portuguesa;

c)Implementação de um Analisador Léxico de uma Linguagem de Programação;

d)Representação da componente sintática da Linguagem Natural;

e)Representação da componente semântica das Linguagens de Programação

Resposta correta: alternativa c)

Resolução do exercício

a) alternativa a) incorreta.

Justificativa: A implementação de uma Analisador Semântico faz uso de um formalismo denominado Gramática de Atributos.

b) alternativa incorreta.

Justificativa: A representação da componente sintática de uma Linguagem faz uso, em geral, da Gramática Livre de Contexto.

c) alternativa correta

Justificativa: O Analisador Léxico faz uso do estudo da componente Regular de uma Linguagem de Programação. O Analisador Léxico varre o código fonte e classifica cada palavra em tokens. São exemplos de tokens: palavras reservadas, identificadores, operadores relacionais, etc.

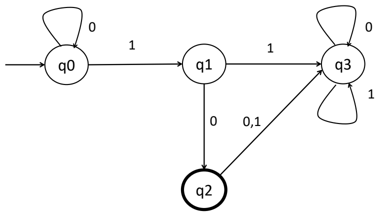
d) alternativa incorreta.

Justificativa: A componente sintática de uma Linguagem faz uso do estudo dos formalismos da componente Livre de Contexto.

e) Alternativa incorreta.

Justificativa: A representação da componente semântica da Linguagem de Programação faz uso do formalismo conhecido como Gramática de Atributos.

Questão 14: **Considere o seguinte autômato. Assinale a cadeia reconhecida pelo autômato.**



a)11100
b)11110
c)011001
d)0101001
e)0010
Resposta correta: e)

Resolução do exercício

a) alternativa incorreta.
Justificativa: Esta cadeia alcança o estado q3, após o consumo da subcadeia 111, que é um estado inútil, ou seja a partir do mesmo não é possível alcançar o estado final q2. Uma cadeia que não alcance o estado final é rejeitada pelo autômato.
b) alternativa incorreta.
Justificativa: Esta cadeia alcança o estado q3, após o consumo da subcadeia 111, que é um estado inútil, ou seja a partir do mesmo não é possível alcançar o estado final q2. Uma cadeia que não alcance o estado final é rejeitada pelo autômato.
c) alternativa incorreta.
Justificativa: Esta cadeia alcança o estado q3, após o consumo da subcadeia 011, que é um estado inútil, ou seja a partir do mesmo não é possível alcançar o estado final q2. Uma cadeia que não alcance o estado final é rejeitada pelo autômato.
d) alternativa incorreta.
Justificativa: Esta cadeia alcança o estado q2, após o consumo da subcadeia 010, mas em seguida após o consumo do símbolo 1, que se sucede, alcança o estado q3, que é um estado inútil. Assim sendo, não é possível alcançar novamente o estado final q2. Uma cadeia que não alcance o estado final é rejeitada pelo autômato.
e) alternativa correta.

Após o consumo da cadeia 0010, o autômato alcança o estado final q2. A cadeia é esgotada, ou seja totalmente lida. Nesta configuração, diz-se que o autômato aceita, ou reconhece, a cadeia.

<u>Questão 15:</u> **Considere a seguinte gramática G = ({X, Y, Z}, {a,b},P,X}, com P = {X→ a Y ; Y → bZ | ε; Z → aY}. Assinale a palavra que é gerada por esta gramática:**

a)abba;
b)abbb;
c)abab;
d)ababab;
e)ababa;

Resposta correta: alternativa e)

Resolução do exercício

a) alternativa incorreta.

Justificativa: não há como derivar a palavra abba, pois não há como prossegur a seguinte derivação: X ⇒ a Y ⇒ ab Z.

b) alternativa incorreta.

Justificativa: não há como derivar a palavra abbb, pois não há como prosseguir a seguinte derivação: X ⇒ a Y ⇒ ab Z

c) alternativa incorreta.

Justificativa: não há como derivar a palavra abab, pois não há como prosseguir a seguinte derivação: X ⇒ a Y ⇒ abZ⇒ abaY⇒ ababZ

d) alternativa incorreta.

Justificativa: não há como derivar a palavra ababab, pois não há como prosseguir a seguinte derivação: X ⇒ a Y ⇒ abZ⇒ abaY⇒ ababZ⇒ ababaY⇒ abababZ

e) alternativa correta.

Considere-se a seguinte derivação:

X ⇒ a Y ⇒ abZ⇒ abaY⇒ ababZ⇒ ababaY⇒ ababaε = ababa