

Arthur Gomes de Siqueira

1) a) Alfabeto  $\rightarrow$  Conjunto finito de símbolos ou caracteres  
 $\Sigma \rightarrow$  Conjunto de símbolos (um alfabeto)  $\rightarrow \{0, 1, \dots, k\}$

b) Palavra vazia  $\rightarrow$  Não tem codar,  $\epsilon$   
Símbolo  $\rightarrow \epsilon$

c) Sufixo  $\rightarrow$  Subsequência final de símbolos de uma palavra

Prefixo  $\rightarrow$  Subsequência inicial de símbolos de uma palavra

Subpalavra  $\rightarrow$  Sequência de símbolos contíguos de uma palavra

d) Palavra  $\rightarrow$  Sequência finita de símbolos justapostos

e) Linguagem  $\rightarrow$  É um conjunto de sequências de símbolos (palavras) do alfabeto  
Símbolo  $\rightarrow L$

f) Linguagem regular é uma linguagem formal que pode ser expressa usando expressões regulares

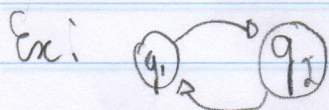
g) Token: É um símbolo que representa um objeto lógico

h) Lexema é uma sequência de caracteres reconhecidos por padrões

2) a) GR  $\rightarrow$  São gramáticas mais restritas e mais próximas das formas de produção  
Ex:  $a(ba)^*$



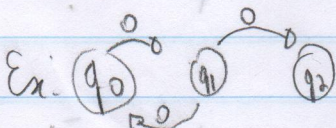
1) b) Autômato finito é o primeiro modelo computacional de descrição da linguagem que não depende por mecanismos de reconhecimento, que pode ser escrito como um teste aplicado a cada ocorrência da palavra.



Tipos: Determinístico → Máquina de estados finitos que aceita ou rejeita cadeia de símbolos, quando um único nome de computação para cada cadeia de entrada.



Não determinístico → Mesma ideia do determinístico, a diferença é que ao ler um símbolo há mais de uma possibilidade de estado destino.

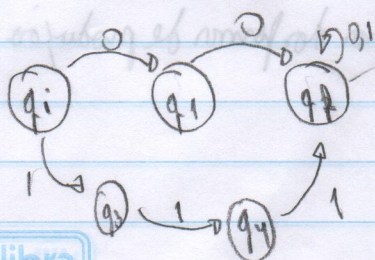


2) c) Expressões regulares: Consistem de constantes e operadores que derivam conjuntos de cadeias de caracteres, operando sobre esses conjuntos.

Ex:  $(0 \cup 1)^* 1 1 (0 \cup 1)^*$

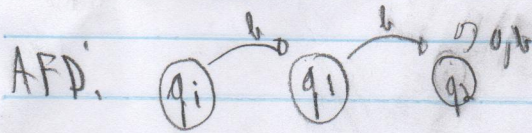
3) a) ER:  $(00 \wedge 111)(0 \cup 1)^*$

APD:

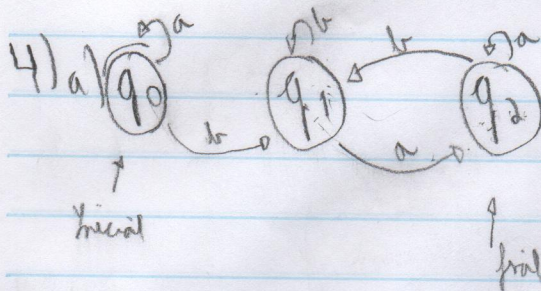
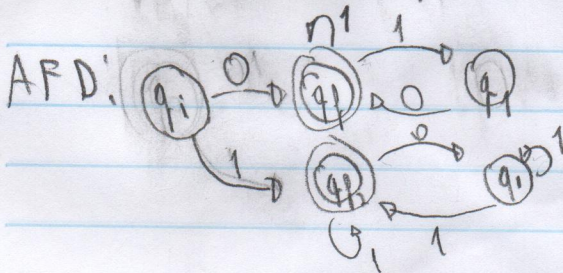




3) b) ER:  $bb(a \cup b)^*$



c) ER:  $0(1+0)^*1^*1(01^+)^*1^*$



b) Recorte 'aba' mas não recorte 'bbbabbb'

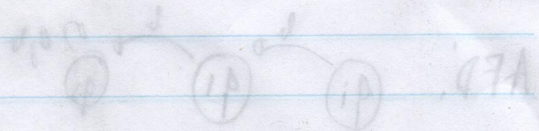
c)  $a^*b(b^*a^+ \cup a^+b^*)^*$

5) a) Ele é o primeiro fase de um processo de compilação, sua função é fazer o leitura do programa fonte, reconhecer o caractere, agrupar os caracteres em lexemas, e produzir um sequência de símbolos léxicos conhecidos como tokens

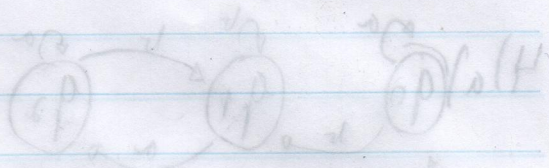
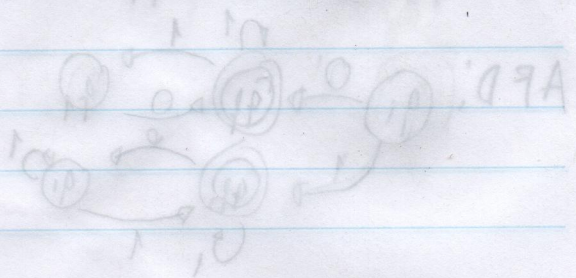
b) É responsável que não reconheça os palavras reservadas, constantes, identificadores e outras palavras que pertencem o léxico de programação



c) Elapso para o analisador sintático e para o leitor com o texto



c)  $ER: HO(1+D) \times 1(0+1) \times 1^*$



d)  $ER: HO(1+D) \times 1(0+1) \times 1^*$

c)  $ER: HO(1+D) \times 1(0+1) \times 1^*$

7) O texto é lido e analisado pelo analisador sintático e pelo leitor com o texto

3) O texto é lido e analisado pelo analisador sintático e pelo leitor com o texto