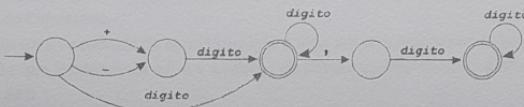


	Primeira Avaliação	Nota: 8,5
Curso:	Ciência da Computação	
Disciplina:	Compiladores	
Aluno(a):	Gilvânia Bonita de Oliveira	Data: 15/05/25

- 1) Considere o seguinte AFD que aceita constantes numéricas no padrão especificado. Pede-se que seja marcada a opção que apresenta uma expressão regular que denote constantes numéricas neste formato. Nota: Nas expressões regulares, "d" corresponde a "dígito" do AFD. (1 pt)



- a) $(+ \mid -)^* d^+ (, d)^*$
b) $(+ \mid - \mid \epsilon) d^+ (, d)^*$
c) $(+ \mid -) (d^* \mid d^*, d^*)$
d) $(+ \mid - \mid \epsilon) (d^* \mid d^*, d^*)$

$(^t/^-) ^* D^+ (^d / D^*) ^*$

E

Nenhuma das opções anteriores é uma resposta para a questão.

- 2) Com relação a Análise Léxica e a Linguagem Tiny marque V (Verdadeiro) ou F (Falso). (1 pt)

- a) (V) O Analisador Léxico de Tiny implementa o princípio da subcadeia mais longa.
 b) (V) O Analisador Léxico de Tiny implementa o conceito de verificação a frente.

- 3) Sobre o código do Analisador Léxico da Linguagem Tiny marque V (Verdadeiro) ou F (Falso). (2 pts)

- a) (F) O procedimento `reservedLookup` recebe como argumento um arquivo de código fonte e realiza uma busca neste arquivo por palavras reservadas. X
 b) (F) A solução para implementação utilizada no analisador léxico de Tiny é a que representa a função programa do autômato como uma matriz indexada pelos estados e símbolos do alfabeto. X
 c) (F) A função `UngetNextChar()` decrementa de um o campo `linePos` de uma variável do tipo `TokenType`.
 d) (F) As variáveis `stringVal` e `numVal` tem a função de armazenar o lexema do token corrente.

Williane Bonia Marcineta

- 4) Considerando a função *UngetNextChar* do analisador léxico da linguagem Tiny e a definição formal de autômato marque a opção que apresenta uma afirmativa correta sobre esta função. (1 pt)

- a) () Movimenta a cabeça de leitura do autômato à esquerda quantas posições forem necessárias até o primeiro caractere do token correto.
- b) () Realiza o movimento da cabeça de leitura do autômato para a esquerda até o início da fita.
- c) () Descarta o último caractere lido e avança a cabeça de leitura uma célula para a direita.
- d) () Descarta o último caractere lido e avança a cabeça de leitura para a direita até encontrar o primeiro caractere que case com o de algum tipo de marca da linguagem.
- e) (X) Nenhuma das opções está correta.

- C
5) Sobre a função *getToken*, do analisador léxico da linguagem Tiny, marque a opção correta: (1 pt)

- a) () Chama a função *getnextchar* que por sua vez reconhece e retorna a primeira sequência de caracteres que casa com o padrão de algum tipo de token da linguagem.
- b) () Recebe como argumento uma árvore sintática e retorna um conjunto de tokens.
- c) () Retorna 1 ou 0, representando respectivamente se uma sequência de caracteres lida é λ ou não um token da linguagem.
- d) () Verifica se uma sequência de caracteres lido é uma palavra reservada da linguagem. X
- e) () É a função que implementa a fita e a cabeça de leitura da fita, mas não a função programa, de um AFD X
- f) (X) Nenhuma das opções está correta.

- C
6) Sobre as Expressões Regulares, qual a sua relevância no contexto da análise léxica? Marque uma das opções. (1 pt)

- a) () Permitem especificar todos os tipos de marcas da linguagem exceto as palavras reservadas. X
- b) () Permitem especificar todos os tipos de marcas da linguagem exceto os operadores aritméticos e relacionais. X
- c) () Sua relevância está no fato de que permitem especificar formalmente todas as marcas de uma linguagem, o que não é possível com um autômato. X
- d) () Correspondem a uma especificação em mais alto nível de abstração de uma implementação de um AFD segundo a solução na qual a função programa é implementada como uma matriz indexada por estados e símbolos do alfabeto. X
- e) () Correspondem a uma representação gráfica dos caminhos de processamento para a aceitação dos tokens de uma linguagem. X
- f) (X) Nenhuma das opções está correta.

7) Sobre o compilador e as fases do processo de compilação marque V (verdadeiro) ou F (falso): (2 pts)

- a) (F) A análise léxica tem como função principal reconhecer os tokens de uma linguagem e construir uma árvore que tenha os tokens como folhas.
 - b) (F) A análise léxica recebe como entrada uma sequência de tokens e verifica se estes estão concatenados de forma correta.
 - c) (F) A análise léxica é independente da máquina alvo, mas para construir o analisador sintático é necessário conhecer a arquitetura da máquina que irá executar o código compilado.
 - d) (F) As fases podem ser divididas em frente e fundo, sendo que a análise léxica e a sintática estão na frente e a análise semântica e geração de código estão no fundo.
- 8) Com relação ao analisador léxico da linguagem Tiny pergunta-se: Qual das linhas em um arquivo de código fonte Tiny geraria exatamente 1 erro léxico? Marque uma das opções. (1 pt)
- a) () media1 - 5 +
 - b) () media++5
 - c) () media::5
 - d) (X) media=:5
 - e) () Nenhuma das opções anteriores

media E/S EOF St

media E/S EOF St

media 5 EOF St - 2 ms
Ima Ima

media = 5 EOF St - 1 ms
Ima