

	Primeira Avaliação	Nota: 9,0
Curso:	Ciência da Computação	
Disciplina:	Compiladores	
Aluno(a):		Data:

- 1) Sendo "L" uma letra e "D" um dígito (número natural), marque a expressão regular que especifica formalmente os identificadores da linguagem Tiny. (1 pt)

- a) D*
- b) L L* ✓
- c) D (L + D)*
- d) L (L + D)*
- e) L (L + D)*
- f) Nenhuma das opções anteriores

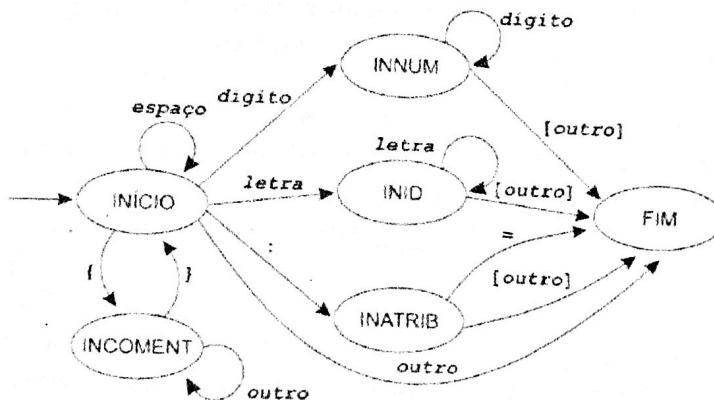
- 2) Sobre os compiladores, marque a opção que corresponde a uma afirmativa verdadeira. (1 pt)

- a) A análise léxica é a primeira fase do processo de compilação e tem como objetivo principal identificar e corrigir erros de natureza léxica, o que é necessário para que a análise sintática possa ser executada posteriormente.
- b) A análise sintática é a segunda fase do processo de compilação e tem como principal objetivo informar ao programador quais correções devem ser feitas a cada não conformidade do código fonte em relação a sintaxe da linguagem, normalmente especificada com uma gramática livre do contexto.
- c) A análise semântica é a segunda fase do processo de compilação e tem como objetivo verificar se a árvore sintática atende às regras semânticas da linguagem que podem ser definidas por um AFe.
- d) As fases de análise sintática e geração de código tem sua implementação dependente da máquina alvo.
- e) As análises léxica, sintática, semântica e geração de código fazem parte do front-end de um compilador e tem em conjunto a função de gerar código objeto para uma máquina alvo que é o back-end.
- f) Nenhuma das anteriores é verdadeira. ✓

- 3) Sobre AFD, AFN e a implementação do analisador léxico de um compilador marque a opção verdadeira. (1 pt)

- a) Os AFNs são preferíveis aos AFDs para a implementação de analisadores léxicos pois pode-se aplicar sobre eles o algoritmo de minimização de autômatos. Este algoritmo gera um autômato com o menor número de estados para uma linguagem, o que é traduzido em melhor legibilidade do código.

- b) Os AFDs são preferíveis na implementação, mas podem não ser preferíveis no projeto por uma questão de facilidade de especificação do reconhecedor (por parte do desenvolvedor) para certas linguagens.
- c) Os AFNs e AFDs são formalismos equivalentes, desta forma a escolha de qualquer um deles é indiferente, tanto a nível de projeto quanto de implementação do analisador léxico.
- d) O principal motivo para utilização de um AFD na implementação é que sempre é possível construir a partir de um AFN um AFD que possua apenas um estado final.
- x e) Nenhuma das opções anteriores é verdadeira
- 4) Considere o seguinte autômato, que aceita marcas da linguagem Tiny. Ele tem "INÍCIO" como estado inicial e "DONE" como estado final e em suas transições, "d" corresponde a um número (de zero a nove) e "l" corresponde a uma letra (de "a" a "z"). Marque a opção que apresenta as afirmativas verdadeiras. (0,5 pt)



- I. Apresenta não-determinismo.
 II. Apresenta o princípio da verificação a frente.
 III. Aceita a seguinte string: {{coment}}
 IV. Aceita a seguinte string: :

- a) I
 b) II
 c) III
 d) IV
 e) I, II
 f) II, III
 g) II, IV
 h) I, II, III
 i) Nenhuma das opções anteriores

5) Marque a opção que apresenta a variável que armazena o lexema de uma marca no analisador léxico da linguagem Tiny (0,5 pt)

- a) () state
- b) () save
- x c) () TokenType
- d) (X) tokenString
- e) () currentToken
- f) () Nenhuma das anteriores

e

6) 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. (0,5 pt)

- a) () Movimenta a cabeça de leitura do autômato à esquerda quantas posições forem necessárias até o primeiro caractere do token corrente.
- b) (X) Realiza o movimento da cabeça de leitura do autômato para a esquerda até o início da fita.
- x 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) () Nenhuma das opções está correta.

A

E

7) Com relação ao analisador léxico da linguagem Tiny pergunta-se: Qual das linhas em um arquivo de código fonte Tiny não geraria qualquer erro léxico? Marque uma das opções. (0,5 pt)

- a) () 1-5+;3
- b) (X) media+;5
- x c) () media::5;
- x d) () media=:5
- e) () Nenhuma das opções anteriores

C