

UNIVERSIDADE PRESBITERIANA MACKENZIE

- Faculdade de Computação e Informática -



Ciência da Computação

Compiladores – 06N

Prova 1 – 09 de outubro de 2020

Professor: Fabio Lubacheski

Orientações para elaboração da prova:

Esta prova pode ser feita em **dupla** ou **individualmente**, basta que somente um dos integrantes entregue um **arquivo pdf** com as respostas das questões e com o seguinte cabeçalho no início do arquivo.

Nós,

Nome completo e TIA (1º integrante)

Nome completo e TIA (2º integrante)

declaramos que

todas as respostas são fruto de nosso próprio trabalho,
não copiamos respostas de colegas externos a dupla,
não disponibilizamos nossas respostas para colegas externos a dupla e
não realizamos quaisquer outras atividades desonestas para nos beneficiar ou prejudicar outros.

1) (1,0 ponto) O Que são palavras reservadas (=palavras chaves) ? Qual sua função nas linguagens de alto nível? Como é que o analisador léxico efetua o reconhecimento de palavras reservadas ?

2) (1,0 ponto) O que são compiladores *just-in-time* ? De um exemplo de linguagem de programação com essa característica.

3) (2,5 pontos) Considere a expressão regular para **constantes numéricas** de uma linguagem algorítmica com a seguinte notação abaixo:

$(+|-)?n, n((E|e)(+|-)?n)?$

onde

n é uma sequência de um ou mais dígitos, ou seja, $(0|1|2|3|4|5|6|7|8|9)^+$;

A notação **x?** significa que **x** é opcional, ou seja $(x|\epsilon)$.

a) Construa um autômato para a expressão regular acima, para esse item você deve desenhar o diagrama de estados do autômato usando o site <http://madebyevan.com/fsm/>.

b) Baseado no autômato, implemente uma **função** que reconheça as constantes numéricas representadas pela expressão regular.

4) (2,0 pontos) Considere a gramática na notação BNF, com a símbolo inicial S.

$S ::= iEtS \mid iEtSeS \mid a$

$E ::= b$

Esta gramática é ambígua? explique porque a gramática é ambígua.

Caso seja retire a ambiguidade da mesma de forma que ela continue gerando a mesma linguagem.

5) (2,5 pontos) Considere a gramática na notação BNF, com a símbolo inicial A.

A ::= abB
B ::= cC | eD
D ::= da
C ::= bC | λ

- a) Apresente a árvore de derivação para cadeia abeda.
b) Construa um analisador sintático descendente recursivo para essa gramática, considere que a cadeia que será testada já está armazenada na variável `*buffer` e chamada do analisador seria feita na função `main()` conforme abaixo.

```
char *buffer="abeda";

int main(void){
    printf("analizando %s\n",buffer);
    A();
    if( *buffer == '\x0')// fim de buffer
        printf("palavra aceita.");
    else
        printf("\nerro sintatico.");

    printf("\n fim de programa.\n");
    return 0;
}
```

6) (1,0 ponto) Considere a gramática

S ::= ABCDd
A ::= aA | λ
B ::= bC | λ
C ::= cD | λ
D ::= e

Apresente o FIRST de cada um dos não-terminais da gramática

Boa Prova !