

Segunda Prova de Construção de Compiladores.  
 Primeiro Semestre de 2005.  
 Departamento de Computação – UFSCar.  
 José de Oliveira Guimarães.  
 Turma A (Terça).

Lembre-se: justifique tudo a menos de menção em contrário. Qualquer compartilhamento de informações entre os alunos durante a realização desta prova resultará em zero no semestre para todos os envolvidos.

Entregue apenas a folha de respostas. Isto é, não entregue esta folha ou o rascunho.

A gramática  $G$  cujas produções são

1.  $S ::= '( S ', ' A )'$
2.  $S ::= A$
3.  $A ::= N$
4.  $A ::= V A'$
5.  $A' ::= N$
6.  $A' ::= \epsilon$

será utilizada nas questões abaixo.  $N, V, (, )$  e  $,$  são terminais.

1. (2.0) Faça a análise sintática descendente, usando  $G$ , da sentença “ $( N, N )$ ”.
2. (2.5) Calcule as funções *first* e *follow* de  $S, A$  e  $A'$  de  $G$ . Utilize quadros do tipo

first	S	A	A'

follow	S	A	A'

3. (2.5) Faça a tabela de análise sintática descendente não recursiva da gramática  $G$ . Algo do tipo

	( , )	N	V	\$
S				
A				
A'				

Naturalmente, os erros devem ser apontados o mais cedo possível. Coloque um – nas entradas vazias da tabela. Não é necessário justificar.

4. (2.5) Explique porque a regra

Se  $A ::= \alpha | \beta$ , então ou  $b \in \text{first}(\alpha)$  ou  $b \in \text{first}(\beta)$  (não as duas condições ao mesmo tempo)

é necessária para que possamos construir um analisador recursivo descendente para uma gramática. Isto é, se a regra não é obedecida, a gramática não é LL(1).

**5. (2.5) Dada a expressão regular**

**a+b[0-9]\*(A|B)**

**faça o autômato finito e o código em Java que a reconhecem. O código em Java deve ter o formato**

```
void analize( char in[] ) {  
    int s = 1, // estado inicial  
        k = 0; // posicao no vetor in  
    while ( s != estado final )  
        switch( s ) {  
            ... // um case para cada estado  
        }  
}
```

**Não deve existir outro comando de repetição além do while do código acima. Utilize a função “error()” para sinalizar um erro. Despreze os caracteres restantes em in após chegar ao estado final.**