

Segunda Prova de Construção de Compiladores.  
Primeiro Semestre de 2002.  
Departamento de Computação – UFSCar.  
José de Oliveira Guimarães.  
Turma B (Quinta).

Lembre-se: justifique tudo a menos de menção em contrário.  
Entregue apenas a folha de respostas. Isto é, não entregue esta folha ou o rascunho.

1. (3.0) Dada a gramática
- |               |                     |
|---------------|---------------------|
| 1. $S ::= Ac$ | 2. $S ::= Bd$       |
| 3. $A ::= a$  | 4. $A ::= \epsilon$ |
| 5. $B ::= b$  | 6. $B ::= Cd$       |
| 7. $C ::= c$  |                     |

faça:

- a) calcule first e follow para cada não terminal. A gramática é suficientemente simples para que o cálculo de first e follow seja feito sem o auxílio da definição destas funções.  
b) monte a tabela de análise descendente não recursiva. Mostraremos apenas uma das regras para construir a tabela: se  $\epsilon \in \text{first}(\alpha)$ , adicione

$A ::= \alpha$

em  $M[A, b]$  para cada  $b \in \text{follow}(A)$ ;

2. (2.5) Prove que a gramática

$S ::= \text{"if"} E \text{"then"} S EL \mid id$   
 $EL ::= \text{"else"} S \mid \epsilon$   
 $E ::= id \mid \text{Numero}$

é ambígua. id e Numero são terminais.

3. (1.0) Desenhe o autômato finito que representa a seguinte expressão regular:  
 $(0|1)+A*B?[0-9]$

O ? quer dizer opcional.

4. (1.5) Cite dois motivos pelos quais interpretadores são mais fáceis de fazer do que compiladores.

5. (2.5) Faça uma gramática com operadores binários + e \* e operador unário prefixado ^ tal que:

- + seja associado à direita;
- \* seja associado à esquerda;
- ^ tenha maior precedência do que +
- + tenha maior precedência do que \*.

Não é necessário justificar.