

Segunda Prova de Construção de Compiladores.
Primeiro Semestre de 2004.
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. (2.0) Faça UM e apenas UM dos items abaixo.

- (a) Cite duas aplicações de técnicas de compilação excluindo-se a aplicação clássica (que é fazer um compilador que toma um programa e produz outro programa).
- (b) Cite duas vantagens de interpretadores sobre compiladores. Explique cada uma delas. Cite uma vantagem de compiladores sobre interpretadores. Assuma que o interpretador tome o código fonte, compile-o para código de uma máquina virtual e depois execute este código. Assuma que o compilador produza código de máquina diretamente. Seja sucinto.

2. (2.0) Faça a expressão regular que reconheça a mesma string que o analisador léxico dado abaixo. Desenhe o autômato finito correspondente. Veja que “return true” significa chegar a um estado final e que “return false” é a sinalização de um erro.

```
boolean analize( char in[] ) {
    k = 0;
    while ( in[k] == '$' || in[k] == '+' || in[k] == '-' )
        k++;
    if ( in[k] == '0' || in[k] == '1' )
        k++;
    else
        return false;
    if ( in[k] == '#' ) k++;
    if ( in[k] >= 'a' && in[k] <= 'z' ) {
        k++;
        return true;
    }
    else
        return false;
}
}
```

3. (2.0) Fatore e tire a recursão à esquerda do trecho de gramática abaixo. Assuma que os não terminais cujas regras não são mostradas derivem terminais não utilizados no trecho abaixo (isto é, não se preocupe com A, B e M). Lembre-se da recursão indireta ...

$T ::= i \mid I \mid P$

$I ::= i M$

$P ::= A B \mid T A$

4. (4.5) Dada a gramática

1. $S ::= A a b$

2. $A ::= a$

3. $A ::= \epsilon$

faça os seguintes itens:

(a) (1.0) encontre o first e follow de cada não terminal. Lembre-se de que o eof sempre se segue ao ...;

(b) (1.0) faça a tabela de análise LL(1). Coloque – nas entradas inválidas;

(c) (2.0) uma das condições para uma gramática ser LL(1) é a seguinte: se $A ::= \alpha \mid \beta$ são duas produções distintas da gramática, se $\beta \xRightarrow{*} \epsilon$, então α não derivará qualquer *string* começando com um terminal que pertence a $\text{follow}(A)$.

Justifique esta regra; isto é, explique porque, se ela não for obedecida, haverá duas escolhas corretas a se fazer em dado ponto da análise LL(1). Utilize um exemplo.

(d) (0.5) a gramática dada acima é LL(1)? Justifique muito sucintamente.

5. (3.5) Usando a gramática

$Z ::= S\$$

$S ::= (L)$

$S ::= x$

$L ::= S$

faça o autômato de análise LR(0) e analise a sentença (x). Em cada passo da análise, mostre qual o token corrente colocando um • em frente a ele e mostre a pilha de estados (veja exemplo no quadro). Faça o autômato com o *layout* mostrado no quadro (para facilitar a correção).