

U.C. 21018

Compilação

26 de junho de 2017

-- INSTRUÇÕES --

- O estudante deverá responder à prova na folha de ponto e preencher o cabeçalho e todos os espaços reservados à sua identificação, com letra legível.
- No fim da prova, poderá ficar na posse do enunciado.
- Verifique no momento da entrega da(s) folha(s) de ponto se todas as páginas estão rubricadas pelo vigilante. Caso necessite de mais do que uma folha de ponto, deverá numerá-las no canto superior direito.
- Em hipótese alguma serão aceites folhas de ponto dobradas ou danificadas.
- Exclui-se, para efeitos de classificação, toda e qualquer resposta apresentada em folhas de rascunho.
- Os telemóveis deverão ser desligados durante toda a prova e os objectos pessoais deixados em local próprio da sala de exame.
- Utilize unicamente tinta azul ou preta.
- A prova é constituída por **3** páginas (esta página de rosto e duas com as questões), contém 5 questões, sem consulta, e termina com a palavra **FIM**. Verifique o seu exemplar e, caso encontre alguma anomalia, dirija-se ao professor vigilante nos primeiros 15 minutos da mesma, pois qualquer reclamação sobre defeito(s) de formatação e/ou de impressão que dificultem a leitura não será aceite depois deste período.

Duração: 150 minutos

1ª Questão (4 valores)

Considere a seguinte gramática:

```
P -> D A
D -> type id ;
A -> id = E ;
E -> E + T | E - T | T
T -> T * F | T / F | F
F -> ( E ) | id
```

Construa as tabelas de acções e saltos do analisador sintáctico ascendente LR, pelo método SLR.

2ª Questão (4 valores)

Apresente a especificação *flex* para um analisador léxico que identifique os seguintes tokens e imprima no ecrã o respectivo token ID. No caso dos identificadores, o programa deve imprimir também, entre parênteses, o nome do identificador.

- Identificador (token ID: IDENTIFIER) – uma letra seguida de letras e dígitos
- Número real (token ID: NOT_CIENT) – número real em notação científica.
- Operador condicional (token ID: OP_ARIT) – operador aritmético (+, -, *, /).

3ª Questão (4 valores)

Apresente o código intermédio, em notação TAC (*three address code*) correspondente ao seguinte excerto de código em linguagem C:

```
int a[100];
int i;

a[0] = 1;
a[1] = 1;
for (i = 2; i < 100; i++)
    a[i] = a[i-2] + a[i-1];
```

4ª Questão (4 valores)

Apresente a especificação *bison* para um analisador sintático para comparação entre expressões de valores constantes. As constantes são identificadas pelo analisador léxico, sendo que o analisador sintático deverá considerar as seguintes operações:

- As expressões são compostas por operações aritméticas sobre números (constantes), sendo possíveis as operações +, -, * e /.
- Os operadores de comparação possíveis são <, <=, >, >=, =, !=.

Considere ainda que os operadores * e / tem precedência sobre + e - e que pode usar parênteses para alterar a ordem de avaliação

5ª Questão (4 valores)

Optimize o seguinte código gerado em TAC (*three address code*), explicitando o tipo de otimização que está a fazer:

```
a = 2 * 2
t1 = a / 2
t2 = t1 + a
t3 = t2 - 2
t4 = t3 / 2
t5 = t3
t6 = t4 + t5
c = 4 + t6
```

FIM