

**U.C. 21018**

**Compilação**

**16 de Junho de 2016**

## **-- 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 -> V S
V -> t i ;
S -> i <- E ;
E -> E + T | T
T -> T * F | F
F -> { E } | i
```

Construa as tabelas de ações e saltos do analisador sintático ascendente LR, pelo método LR canónico.

### 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: NUMREAL) – número real na notação com vírgula, sem zeros não significativos (tanto à esquerda como à direita).
- Operador condicional (token ID: OPCOND) – operador condicional, para comparação entre dois valores (< , > , <= , >= , != , ==).

### 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[10];
int i;

a[0] = 0;
for (i = 1; i < 10; i++)
    a[i] = a[i-1] * 2;
```

#### 4ª Questão (4 valores)

Apresente a especificação *bison* para um analisador sintático de expressões sobre strings, que devolva o resultado da avaliação da expressão analisada. As constantes (numéricas e strings) são identificadas pelo analisador léxico, sendo que o analisador sintático deverá considerar as seguintes operações:

- Concatenação de strings: o resultado de  $s1 + s2$  é uma string com a sequência de caracteres de  $s1$  seguida da sequência de caracteres de  $s2$ . **Ex.** “abc”+”123” é “abc123”
- Repetição: o resultado de  $s * n$ , em que  $s$  é uma string e  $n$  é um número inteiro, e é a concatenação de  $s$  com ela própria  $n$  vezes. Ex. “ab”\*3 é “ababab”

Considere ainda que o operador  $*$  tem precedência sobre  $+$  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 = 3 - 1
t1 = a * 2
t2 = t1 + a
t3 = t2 - 2
t4 = t3 / 2
t5 = t3
t6 = t5 + b
c = 4 + t6
```

**FIM**