UNIVERSIDADE ABERTA
www.uab.pt

Ministério da Educação e Ciência

U.C. 21018

Compilação

23 de junho de 2014

-- 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:

```
List -> ( Expr , List ) | ( Expr )
Expr -> Term ? Expr | Term
Term -> Factor % Term | Factor
Factor -> id | [ Expr ]
```

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

2ª Questão (4 valores)

Apresente a especificação *flex* para um analisador léxico que identifique os tokens de expressões e os transforme num documento XML.
O programa deverá converter as expressões para a notação XML, em que as tags correspondem <expr> e </expr> delimitam a expressão. Além disso, as entidades correspondentes a caracteres especiais devem ser convertidas no correspondente símbolo. Por exemplo, a expressão:

```
3 > 1
```

deverá ser convertido em:

```
<expr> 3 &gt; 1 </expr>
```

Os símbolos a ser alterados para entidades XML são os seguintes:

- `<` que representa o símbolo <
- `>` que representa o símbolo >
- `&` que representa o símbolo &

Os outros símbolos possíveis são +,-,*,/,= e |, para além de números ou identificadores.
NOTA: não é necessário verificar a validade das expressões. Uma expressão 5+*-2 é aceitável.

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[64];
int i, j;

for (i = 0; i < 8; i++)
  for (j = 0; j < 8; j++)
    a[i*10+j] = i+j;
```

4ª Questão (4 valores)

Apresente a especificação *bison* para um analisador sintáctico de expressões lógicas, que devolva o resultado da avaliação da expressão analisada. As constantes lógicas são identificadas pelo analisador léxico, sendo que o analisador sintáctico deverá considerar as seguintes operações:

- Conjunção: o resultado de `x & y` é **1** se `x` e `y` forem **verdadeiros**. Caso contrário é **0**.
- Disjunção: o resultado de `x | y` é **1** se `x` ou `y` for **verdadeiro**. Caso contrário é **0**.
- Disjunção exclusiva: o resultado de `x % y` é **0** se `x` e `y` tiverem o mesmo valor lógico. Caso contrário é **1**.

Considere ainda que o operador `&` tem precedência sobre `|` e este sobre `%`, e que pode usar parêntesis para alterar a ordem de avaliação.

5ª Questão (4 valores)

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

```
a = 3 - 1
t1 = a / 2
t2 = t1 + a
t3 = t2 - 2
t4 = t3 - 1
t5 = t4 + b
c = 4 + t5
```

FIM