PROCESSAMENTO de LINGUAGENS I
LESI + LMCC

Exame Especial

Data: 28 de Novembro de 2001
Hora: 17:00

Questão 1 *Considere a seguinte gramática independente do contexto $G = (T, N, S, P)$, com:*

$T = \{a, b, c, d, e, f, g\}$,
$N = \{S, A, B, C\}$,
$S = S$,

$$
\begin{array}{lllll}
P = & \{ & p_0: & S & \rightarrow \; A\ a \\
 & , & p_1: & A & \rightarrow \; B \\
 & , & p_2: & & \mid \; B\ A \\
 & , & p_3: & B & \rightarrow \; C\ d\ A\ b\ e\ c\ f \\
 & , & p_4: & & \mid \; c\ C\ e\ f \\
 & , & p_5: & C & \rightarrow \; g \\
 & , & p_6: & & \mid \; g\ g \\
 & , & p_7: & & \mid \; g\ g\ g \\
 & \} & & &
\end{array}
$$

Responda às seguintes alíneas:

a) *Calcule o $First$ do lado direito da produção p_0.*

b) *Consegue dizer, sem realizar nenhum cálculo se a gramática é $LL(1)$. Justifique a sua resposta.*

c) *Calcule o $Follow(B)$ e o $LA(A \rightarrow BA)$.*

d) *Existe um conflito $LL(1)$ nas derivações do não-terminal C. Qual? Resolva-o apresentando as novas derivações para C.*

Questão 2 *Considere a seguinte gramática independente do contexto abstracta escrita na notação do Lrc:*

```
root Raiz;
Raiz   : ProdRaiz(Arvore)
       ;
Arvore : Folha ()
       | Nodo  (INT Arvore Arvore)
       ;
```

que descreve a estrutura abstracta de árvores binárias em que a informação (neste caso números inteiros) são armezanados nos seus nodos intermédios, i.e., nodos não folhas.

Uma possível notação concreta para estas árvores é apresentada na seguinte frase da linguagem:

```
(3 (15 (3 () ())
       (5 (8 () ()) ())
   )
(4  () (7 () ()))
)
```

a) *Aplique as regras de conversão de gramáticas abstractas para tipos de dados recursivos na linguagem C e respectivas funções construtoras.*

a.1) *Considere que se pretendia utilizar o gestor de memória "Bohem's garbage collector", tal como no 1º trabalho prático. Escreva o código C necessário para re-utilizar esse gestor de memória na implementação das árvores abstractas.*

b) *Considere que o resultado de efectuar o parsing de uma frase concreta da linguagem é uma árvore abstracta do tipo* `Raiz`*. Desenhe essa árvore considerando que o parser recebe como entrada a frase concreta dada como exemplo.*

c) *Defina a gramática independente do contexto concreta que define a estrutura concreta da linguagem. Defina também o parser que reconhece frases concretas da linguagem e constrói a árvore abstracta. (Pode responder esta pergunta usando os sistemas Lex/Yacc ou LRC. Se pretende responder em Lex/Yacc e não respondeu à alinea* `a)`*, então considere que o nome das produções são as funções que constroem os nodos da árvore abstracta).*

d) *Aplique as regras definidas nas aulas teóricas para representar as árvores abstractas como documentos XML. Pode definir esta função quer usando as regras de unparsing do LRC, quer como uma função recursiva (C ou Haskell) sobre o tipo de dados induzido pela gramática abstracta.*

e) *Considere que se pretende calcular a média aritmética dos elementos que ocorrem na árvore. Defina o(s) atributo(s) necessários e as suas regras de cálculo. Isto é, defina a gramática de atributos (graficamente ou numa notação textual).*

f) *Considere que se pretende efectuar uma travessia por níveis da árvore. Isto é, pretende-se calcular (sintetizar) uma lista cujos elementos são a lista de elementos que ocorrem em cada nível da árvore. Isto é, o primeiro elemento da lista contém os elementos que ocorrem no primeiro nível da árvore, o segundo elemento contém os elementos que ocorrem no segundo nível da árvore, etc.*

Por exemplo, uma travessia por níveis da árvore exemplo produziria a seguinte lista:

```
[[3],[15,4],[3,5,7],[8]]
```

Escreva a gramática de atributos de modo a sintetizar esta lista. Utilize os tipos e as funções semânticas sobre listas que achar necessárias.

Questão 3 *Pretende-se criar uma agenda para gerir contactos de pessoas e empresas. Uma agenda deste tipo é essencialmente constituída por uma lista de entradas. Uma entrada pode ser simples, correspondendo a uma pessoa ou empresa, ou pode ser composta, neste caso corresponde a um grupo de pessoas ou empresas ou grupos aninhados. De notar que num grupo podem, também, aparecer referências a entradas ou grupos previamente definidos na agenda. A agenda poderá conter um número livre de grupos aninhados.*

Cada entrada é constituída pelos seguintes items de informação:

id	*uma string que corresponderá ao identificador único da entrada.*
tipo	*campo com dois valores possíveis: pessoa ou empresa.*
nome	*nome da pessoa ou da empresa.*
email	*email da pessoa ou empresa.*
telefone	*telefone da empresa ou pessoa.*
gid	*no caso da entrada se tratar dum grupo este campo deverá conter o identificador do grupo.*
ref	*uma string contendo o identificador duma entrada ou dum grupo previamente definido.*

Um exemplo possível de uma agenda na notação concreta de XML apresenta-se de seguida:

```
<AGENDA>
  <ENTRADA id="e1" tipo="pessoa">
    <NOME>José Carlos Ramalho</NOME>
    <EMAIL>jcr@di.uminho.pt</EMAIL>
    <TELEFONE>253 604479</TELEFONE>
  </ENTRADA>
  <GRUPO gid="epl">
    <ENTRADA id="e2" tipo="pessoa">
      <NOME>Pedro Henriques</NOME>
      <EMAIL>prh@di.uminho.pt</EMAIL>
      <TELEFONE>253 604469</TELEFONE>
    </ENTRADA>
    <ENTRADA id="e3" tipo="pessoa">
      <NOME>João Saraiva</NOME>
      <EMAIL>jas@di.uminho.pt</EMAIL>
      <TELEFONE>253 604479</TELEFONE>
    </ENTRADA>
    <REFERENCIA ref="e1"/>
  </GRUPO>
  <ENTRADA id="e4" tipo="pessoa">
    <NOME>José João Almeida</NOME>
    <EMAIL>jj@di.uminho.pt</EMAIL>
    <TELEFONE>253 604433</TELEFONE>
  </ENTRADA>
</AGENDA>
```

Desenvolva então, cada uma das seguintes alíneas:

a) *Defina uma gramática independente do contexto abstracta para representar a informação numa agenda deste tipo. Utilize a notação (formal) utilizada nas aulas teóricas.*

b) *Acrescente os atributos necessários e respectivas equações de cálculo para calcular o número de entradas* ***de cada tipo*** *na agenda (um grupo deverá ser contabilizado com um número igual ao das suas subentradas e assim sucessivamente). As referências deverão ser contabilizadas como entradas ou como grupos caso se refiram a um ou a outro respectivamente.*

c) *Acrescente os atributos necessários e respectivas equações de cálculo para garantir a unicidade dos identificadores de entrada e de grupo (trate os dois tipos de identificador dentro da mesma restrição e não como dois problemas separados). No caso de existir um identificador repetido, a sua gramática de atributos sinalizará a primeira ou a segunda ocorrência do identificador como sendo o identificador repetido?*

d) *Utilize atributos para especificar o invariante das referências: o valor do atributo* ***ref*** *duma REFERENCIA deve conter o* ***identificador*** *duma entrada previamente definida.*