

Processamento de Linguagens I

LESI + LMCC (3º ano)

7ª Ficha Prática

Ano Lectivo de 01/02

1 Objectivos

Esta ficha prática contém exercícios para serem resolvidos nas aulas teórico-práticas com vista a sedimentar os conhecimentos relativos a:

- Implementação de parsers com a ferramenta yacc;
- Com cálculo directo;
- Com construção da árvore de sintaxe abstracta.

2 Enunciados

2.1 Agenda de Contactos

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="ep1">
```

```

<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 concreta para representar a informação numa agenda deste tipo. Utilize as ferramentas Lex/Yacc para gerar um reconhecedor para essa linguagem.
- b) Acrescente as acções semânticas necessárias 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 as acções semânticas necessárias 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, o seu parser sinalizará a primeira ou a segunda ocorrência do identificador como sendo o identificador repetido?
- d) Utilize as acções semânticas necessárias para especificar o invariante das referências: o valor do atributo **ref** numa REFERENCIA deve conter o **identificador** numa entrada previamente definida.

2.2 Árvores binárias

Considere a seguinte gramática independente do contexto abstracta escrita na notação do yacc:

```

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 armazenados 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.
- 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. Responda a esta pergunta usando as ferramentas Lex/Yacc.
- d) Considere que se pretende calcular a média aritmética dos elementos que ocorrem na árvore. Acrescente as acções que fazem o cálculo directo dessa média.
- e) 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]]
```

Acrescente as acções semânticas necessárias para realizar esta tarefa.