

Processamento de Linguagens I

LESI + LMCC (3º ano)

Exame de 2ª Época

Data: 22 de Julho de 2005
Hora: 14:00

Dispõe de 2:30 horas para realizar este exame

OrgGeo é uma Linguagem de Domínio Específico (DSL) que se destina a descrever a organização geo-política de um país, ou sua região. Para o efeito, pretende-se identificar as Localidades de cada Concelho de cada Distrito da região em causa.

A gramática independente de contexto G , abaixo apresentada, define a linguagem **OrgGeo** pretendida. O Símbolo Inicial é **OrgGeo**, os Símbolos Terminais são escritos em minúsculas (pseudo-terminais), ou em maiúscula (palavras-reservadas), ou entre apostrofes (sinais de pontuação) e a string nula é denotada por $\&$; os restantes serão os Símbolos Não-Terminais.

```
p1: OrgGeo  --> Distritos
p2: Distritos--> IdD DescD ','
p3:         | Distritos IdD DescD ','
p4: DescD   --> Concelhos
p5: Concelhos--> Concelho
p6:         | Concelhos ';' Concelho
p7: Concelho --> Locais ':' IdC
p8: Locais   --> IdL LstIds
p9: LstIds   --> &
p10:        | ',' Locais
p11: IdD     --> id
p12: IdC     --> id
p13: IdL     --> id
```

Questão 1: parsing

Recordando os seus conhecimentos sobre análise sintáctica *Top-Down* e *Bottom-Up*, responda às alíneas seguintes:

- Mostre que G não é **LL(1)**, apresentando as linhas da Tabela de Decisão LL(1) onde esses conflitos estejam patentes.
Diga, justificando, se acha que poderia concluir que a condição LL(1) não era verificada, mesmo sem construir a tabela.
- Diga porque razão o parser Recursivo-Descendente Puro foi durante muitos anos o mais usado e recorde as principais razões que levaram ao seu abandono.
Explique, então, as vantagens do parser LL(1), referindo o conceito de parser guiado por uma Tabela de Decisão.
- Escreva uma função, pertencente a um parser Recursivo-Descende Puro, para reconhecer o símbolo Não-Terminal **LstIds** e outra para reconhecer o terminal **id**.

d) Supondo que vai analisar a frase

```
BRAGA
Braga, Sequeira : Braga;
Feira-Nova : Amares .
PORTO
Porto : Porto.
```

com um parser *Top-Down LL*, desenhe a respectiva *Árvore de Parsing* indicando a ordem de redução dos nodos.

- e) Verifique se G é **LR(0)**, construindo o respectivo autómato. Caso não seja, mostre se a estratégia **SLR(1)** resolve os conflitos identificados.
- f) Diga qual a diferença entre a *stack de parsing* de um *parser LR* e a *stack valor* do respectivo *tradutor dirigido pela sintaxe*.

Questão 2: TDS e gramática tradutora

Transforme G numa **gramática tradutora**, GT , reconhecível pelo yacc, para gerar um sítio HTML com uma página por cada Distrito, contendo uma lista de itens com os Concelhos e para cada um uma outra lista com suas localidades. Completa a GT anterior de modo a gerar também uma página inicial (a *homepage* do seu sítio) com um título (fixo) do projecto e links para as páginas dos Distritos.

Questão 3: TDSem e gramática de atributos

Depois de transformar G numa gramática independente de contexto abstracta (pode reduzir algumas produções que lhe pareçam supérfluas), escreva uma **Gramática de Atributos**, GA , para construir a **Tabela de Localidades** que associe a cada identificador de Localidade os respectivos Concelho e Distrito. Note que podem existir Localidades com nomes repetidos, desde que tal não aconteça no mesmo Concelho. Nomes de Concelhos e de Distritos é que não podem ser duplicados.

Para facilitar a leitura da sua resposta, reúna numa tabela (no início ou no fim) os **atributos herdados e sintetizados, ou intrínsecos** de cada símbolo (NT ou T) de GA .

Questão 5: sobre o Trabalho Prático 1

Recordando o exercício que o seu grupo resolveu no 1º TP proposto (se fez mais do que um, escolha o que mais lhe agradou), esquematize, desenhando um diagrama de blocos, a arquitectura da sua solução incluindo as fases de geração e de processamento, deixando clara a filosofia e funcionamento do seu sistema.

Escreva a respectiva *makefile* para completar a descrição anterior.

Questão 6: sobre o Trabalho Prático 2

Para resolver o 2º TP, que envolvia o desenvolvimento de um Compilador com base numa Gramática de Atributos, teve de tomar algumas decisões: considerar ma *Máquina Real* ou uma *Máquina Virtual*; e nesse caso decidir se seria uma *Máquina de Stack* ou de *Registos*; se iria gera *Assembly* ou directamente *Código Binário (Byte code)*. Explique essas alternativas e indique as escolhas do seu grupo.