Processamento de Linguagens I
LESI + LMCC (3º ano)

Exame de 1ª Época – 1ª Chamada

Data: 20 de Junho de 2005
Hora: 09:30

Dispõe de 2:30 horas para realizar este exame

Em determinada Linguagem de Programação, chamada nLP, não existem tipos pré-definidos; o programador tem de declarar o nome de cada tipo que quiser usar, indicando o seu tamanho (número de bytes que ocupa em memória). Todas as variáveis que a seguir sejam declaradas terão de ser de um dos tipos definidos. Essas variáveis serão alocadas, pelo compilador, em memória sequencialmente, a partir do endereço 0.
A gramática independente de contexto G, abaixo apresentada, define a sub-linguagem de nLP para declaração de tipos e variáveis. O Símbolo Inicial é nLPD, 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:  nLPD     -->  Tipos  Vars
p2:  Tipos    -->  TIPOS  Ts
p3:  Ts       -->  Tipo
p4:             |  Ts  Tipo
p5:  Tipo     -->  IdT  Tam
p6:  Vars     -->  &
p7:             |  VARIAVEIS  Vs
p8:  Vs       -->  Ids  ':'  IdT
p9:             |  Vs   Ids  ':'  IdT
p10: Ids      -->  IdV  RIds
p11: RIds     -->  &
p12:            |  ','  Ids
p13: IdT      -->  id
p14: Tam      -->  num
p15: IdV      -->  id
```

Questão 1: parsing

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

a) Observando G, é evidente que a GIC dada **não é LL(1)**!
Diga o que nos permite tirar de imediato esta conclusão, explicando a sua resposta.
Apresente 2 linhas da Tabela de Decisão LL(1) onde esses conflitos estejam patentes.

b) Explique o raciocínio que se segue para transformar um parser Recursivo-Descendente Puro em um parser Recursivo-Descendente Genérico guiado por uma Tabela de Decisão (se o ajudar a responder, escreva o algoritmo do RD genérico).

c) Prove, construindo as respectivas tabelas de parsing `ACTION` e `GOTO`, que gramática apresentada G **é SLR(1)**.

d) Supondo que vai analisar a frase

```
TIPO  int  4
VARIAVEIS  a , b : int
```

com o parser SLR(1) acima, desenhe a respectiva *Árvore de Parsing* indicando a ordem de redução dos nodos.

e) Diga qual a diferença entre a *stack de parsing* de um *parser LL* e de um *parser LR*. Complete a resposta dizendo como inicializa cada uma.

Questão 2: TDS e gramática tradutora

Transforme G numa **gramática tradutora**, GT, reconhecível pelo yacc, para traduzir o bloco de declarações num conjunto de factos em Prolog que sejam instâncias dos dois seguintes predicados

```
tipo( idtipo, tam ).
variavel( idvar, idtipo, endr ).
```

Note que os endereços a atribuir a cada variável terão de ser gerados pelo seu tradutor de acordo com o critério acima.

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 calcular o **espaço de memória total** ocupado por todas as variáveis declaradas num determinado texto fonte.
A sua GA, além das *regras para o cálculo dos atributos necessários* e para apresentar o resultado desejado (ditas *regras de tradução*), deve incluir *condições de contexto* para garantir que: (a) não haja re-declaração de identificadores (note que os identificadores das variáveis também não podem ser iguais aos identificadores dos tipos); (b) o tipo de todas as variáveis declaradas exista (isto é, tenha sido previamente declarado).
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), diga com clareza qual o resultado produzido pelo Processador de Linguagens que desenvolveu. Para gerar esse resultado, usou uma Árvore de Derivação explicitamente criada em memória, ou foi gerando à medida que fazia o reconhecimento?

Questão 6: sobre o Trabalho Prático 2

Após explicar o que é uma Máquina Virtual e as vantagens do seu uso no contexto da geração de código, justifique a decisão do seu grupo, a nível do 2º TP, escolhendo usar uma MV já existente (o que fez nesse caso), ou optando por desenvolver uma nova MV de raiz.