UCE30-EL, Engenharia de Linguagens

1ª Teste de EG

Data: 27 de Fevereiro de 2012
Hora: 08:00

Dispõe de 2 hora para realizar este exame

1 Desenho/especificação de uma Linguagem (5v)

Pretende-se definir uma nova Linguagem que permita descrever o programa de uma conferência de 1 ou mais dias. A conferência tem um *nome*, um *local de realização*, uma *data de início* e uma *data de fim*. Cada dia (identificado pelo *nome* e *data*) comporta 1 ou mais sessões, cada qual com um *título*, um *moderador*, um *tipo*—`normal` (artigos longos), `posicional` (artigos curtos), `demo` (demonstrações de ferramentas) ou `painel`—um *período* (hora de início e de fim), uma *sala* e uma *lista de comunicações*. Cada comunição é descrita por um *título* e uma *lista de autores*.
Escreva então uma Gramática Independente de Contexto, GIC, que especifique a linguagem pretendida—note que o estilo da linguagem (mais ou menos verbosa) e o seu desenho são da sua responsabilidade.

2 Especificação Gramatical de Linguagens (5v)

Como sabe, dado um alfabeto ou conjunto de símbolos terminais (vocábulos) a Gramática Independente de Contexto, GIC, especifica rigorosamente (formalmente) a sintaxe de qualquer Linguagem de Programação. Para além de definir a sintaxe da linguagem, a GIC também serve para dizer como validar uma frase (sequência de símbolos) e, caso seja válida, fazer o seu reconhecimento estrutural.
A Gramática Tradutora, GT, é um acréscimo que se faz à GIC, por associação de ações às produções, para descrever a semântica da linguagem, mas sobretudo para determinar como as frases válidas devem ser processadas. Porém a GT está longe de ser uma especificação rigorosa da semântica.
A Gramática de Atributos, GA, embora também assente na GIC, apareceu para formalizar a semântica das Linguagens de Programação.
Explique com clareza o que é que a GA traz relativamente à GT para cumprir essa função.

(vire sff...)

3 Gramáticas de Atributos (10v)

Considere a gramática independente de contexto (*GIC*), abaixo apresentada, que define uma linguagem específica para descrever a lista de compras a fazer numa manhã de sábado de sol em Braga com vista a repor o stock do frigorifico e da despensa.
O Símbolo Inicial é `Compras`, os Símbolos Terminais são escritos só em minúsculas (terminais-variáveis) ou só em maiúsculas (palavras-reservadas) ou entre apostrofes (sinais de pontuação), e a string nula é denotada por `&`; os restantes (sempre começados por maiúsculas) serão os Símbolos Não-Terminais.
Admita ainda que todos os símbolos terminais-variáveis (`data`, `id`, `num` e `string`) tem sempre um atributo intrínseco designado por `valor` (que corresponde ao seu valor léxico).

```
p0:  Compras      -->  LISTA PARA data LCs
p1:  LCs          -->  Setor ´.´
p2:                 |  LCs  Setor '.'
p3:  Setor        -->  IdSetor Cs
p4:  IdSetor      -->  MERCEARIA | FRUTARIA | TALHO | PEIXARIA | PADARIA
P5:  Cs           -->  Item
p6:                 |  Cs ';' Item
p7:  Item         -->  Prod Marca Qt Unid
p8:  Prod         -->  id
p9:  Qt           -->  num
p10: Marca        -->  &
p11:                |  string
p12: Unid         -->  KG | L | M | DZ | UNI
```

Neste contexto e após analisar a *GIC* dada, responda às alíneas seguintes.

a) Construa a árvore de derivação, ou árvore de parsing para a seguinte frase:
`LISTA PARA 2012-02-25 MERCEARIA ovos 1DZ. FRUTARIA laranjas "de Amares"2KG; peras "William"4UNI.`

b) Defina os atributos, regras de cálculo e regras de tradução necessários para: *contar e imprimir o total de items distintos a comprar em cada setor* do mercado municipal.

c) Defina os atributos, regras de cálculo e regras de tradução necessários para: *determinar e imprimir o total de quilos (explicitados)* a carregar para casa.

d) Defina os atributos, regras de cálculo e condições de contexto necessários para garantir que: *não se repetem setores* na mesma lista de compras. Caso essa regra seja violada, deve ser emitida uma mensagem de erro