

Processamento de Linguagens

LEI (3º ano)

2º Teste

Data: 25 de Junho de 2008
Hora: 14:30

Este Teste tem 3 questões distribuídas em 3 páginas Dispõe de 2:00 horas para realizar o teste.
--

Questão 1: Desenho/especificação de uma Linguagem (6v)

Uma Universidade (caracterizada pelo *nome*, *código nacional* e *nome do responsável*) está organizada hierarquicamente em Escolas que se compõem por Departamentos que ministram Cursos. Todas estas componentes da estrutura hierárquica são caracterizadas pelos mesmos 3 elementos indicados para a universidade.

Pretende-se definir uma nova Linguagem que permita descrever a estrutura hierárquica de uma Universidade, de acordo com a explicação acima. O estilo da linguagem (mais ou menos verbosa) e o seu desenho são da sua responsabilidade. Escreva para isso uma **Gramática Independente de Contexto**, *GIC*, que especifique a Linguagem pretendida.

Questão 2: Gramáticas, Linguagens e Parsing (6v)

A gramática independente de contexto *GIC*, abaixo apresentada, define uma linguagem específica para apoio na gestão de produção de uma empresa fabricante de automóveis.

O Símbolo Inicial é *Autos*, 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 (sempre começados por maiúsculas) serão os Símbolos Não-Terminais.

```
p1: Autos      --> Auto '.'
p2:           | Autos Auto '.'
p3: Auto       --> Identific '-' Caracteriz '{' Componentes '}'
p4: Identific  --> Modelo Tipo Ano Criador
p5: Caracteriz --> Cilind
p6: Componentes --> Componente
p7:           | Componentes ';' Componente
p8: Componente --> IdComp Desc PECAS : LstPecas
p9: Modelo     --> id
p10: Criador   --> str
p11: Cilind    --> num
p12: IdComp    --> id
p13: Tipo      --> CLASSICO
p14:           | DESPORTIVO
p15: Ano       --> num
p16: Desc      --> str
p17: LstPecas  --> &
p18:           | CodPeca LstPecas
p19: CodPeca   --> id
```

Neste contexto e após analisar a *GIC* dada, responda às próximas questões.

- Descreva por palavras suas a linguagem gerada pela *GIC* acima.
- Escreva uma Frase válida da linguagem gerada pela *GIC* dada, mostrando a respectiva Árvore de Derivação.
- Reduza a gramática *GIC* eliminando as produções inúteis, isto é, aquelas que derivam apenas em um símbolo.
- Altere a *GIC* de modo a obrigar a separar os códigos de cada peça (*CodPeca*) por uma vírgula.
- Embora ambos os pares de produções p6/p7 e p17/18 tenham uma função idêntica, há diferenças entre eles! explique essas diferenças com clareza.
- Se fosse usar um parser Top-Down do tipo LL(1) para reconhecer as frases da linguagem gerada pela *GIC*, o que deveria carregar inicialmente para a *stack de parsing* ?
- Embora siga também a estratégia Top-Down o parser Recursivo-Descende Puro é muito distinto do LL(1); esclareça essa diferença.

Questão 3: Processamento, Gramática Tradutora (8v)

A gramática independente de contexto *GIC*, abaixo apresentada, define uma linguagem específica para apoio ao responsável pelo economato do Departamento de Informática (DI). O Economato tem de distribuir material consumível (de escritório em geral, e de informática em particular) aos vários membros dos vários grupos de investigação do DI; em cada entrega, pode enviar uma ou mais coisas e a uma ou mais pessoas.

O Símbolo Inicial é *Econom*, 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 (sempre começados por maiúsculas) serão os Símbolos Não-Terminais.

```
p1: Econom --> ECONOMATO DO MES Mes Membros MOVIMENTOS Moves
p2: Mes     --> id
p3: Membros --> Grupo
p4:         | Membros Grupo
p5: Grupo   --> GRUPO id ':' Invests
p6: Invests --> Inv
p7:         | Invests Inv
p8: Inv     --> '(' SglInv NomeInv ')'
p9: SglInv  --> id
p10: NomeInv --> str
p11: Moves  --> Move
p12:        | Moves Move
p13: Move   --> ENTREGA ':' Maters PARA Pessoas '.'
p14: Maters --> CodMat
p15:        --> Maters ';' CodMat
p16: CodMat --> id
p17: Pessoas --> SglInv
p18:        | Pessoas ',' SglInv
```

Neste contexto e após analisar a *GIC* dada, transforme-me a numa **gramática tradutora**, *GT*, reconhecível pelo *yacc*, para:

- calcular e imprimir: o número de grupos; o número de investigadores por grupo; o número de entregas mensais; o número máximo de materiais enviados numa mesma entrega;
- identificar e listar por ordem alfabética todos os materiais entregues ao longo do mês, indicando a quantidade de cada (considera-se que em cada entrega é enviado 1 unidade de cada material da lista a cada pessoa);
- detectar e sinalizar as seguintes situações de erro: a Sigla de um Investigador, a quem é feita uma entrega, não fazer parte da lista de Investigadores declarados nos Grupos inicialmente; a mesma Sigla aparecer mais de uma vez na mesma lista de Pessoas de uma entrega.

Inclua também a especificação do Analisador Léxico, em notação do *flex*, a integrar no tradutor gerado pelo *yacc* para devolver os Símbolos Terminais com o respectivo valor semântico (quando necessário).

Para isso note que *id* é uma palavra formada letras, dígitos e sinais de pontuação, desde que comece por uma letra; e que *str* é uma sequência de caracteres entre aspas.