

Processamento de Linguagens – LEI

Exame - versão (b)

12 de Julho de 2013 — 14h

Dispõe de 2:00 horas para realizar este teste.

Questão 1: Linguagens Regulares ($4v = 1+1+1+1$)

Considere as seguintes Expressões Regulares $e1$ e $e2$:

$$e1 = a^+ (c^* d + (b c)^*)$$

$$e2 = a^+ c^* d + (a^+ b c)^*$$

E responda às seguintes alíneas:

- a) Diga se $e1$ e $e2$ são equivalentes;
- b) Escreva uma gramática regular equivalente a $e1$;
- c) Construa o autômato finito não-determinista correspondente a $e2$;
- d) Calcule o respetivo autômato finito determinista.

Questão 2: Linguagens Regulares ($4v = 1+1.5+1.5$)

Um documento de texto específico descreve todos os grupos inscritos a uma dada UC conforme se exemplifica abaixo. Cada descrição de grupo começa pela palavra "GRUPO" seguido por dois pontos (":") seguido pelo número do grupo (espaços são permitidos entre qualquer um destes elementos). Depois vêm blocos onde se descreve cada aluno do grupo (número, nome, estatuto). O nome do aluno aparece em um linha começada pela palavra "NOME" seguida de zero ou mais espaços e do carácter "=" e inclui todos os caracteres a partir daí até terminar com o carácter ".". O mesmo acontece para os restantes campos de aluno.

Exemplo

```
GRUPO: 1
NOME=Antonio Carlos Mendes.
NUM=a57543.
STATUS=Ordinário.
NOME =Ana Gabriela Dias Moura.
NUM=E4544.
STATUS=TE.
...
GRUPO : 2
NOME=Rui Ramos Costa.
.....
```

Desenvolva um Filtro de Texto com o Flex que receba o texto acima descrito e:

- a) conte o número total de grupos;
- b) verifique se todos os grupos têm no máximo 3 alunos (escreva uma mensagem de erro se tiver mais de 3 membros, e uma mensagem de aviso se tiver menos de 3);
- c) produza uma listagem ordenada alfabeticamente por nome de todos os alunos composta por número, nome e identificador do grupo a que está associado.

Questão 3: Linguagens Independentes de Contexto ($7v = 2 + 1.5 + 1.5 + 1.5 + 1.5$)

Para descrever famílias completas, definiu-se uma linguagem específica que permite identificar os pais, com datas de nascimento, e seus filhos. Mostra-se abaixo 1 exemplo de frase válida dessa linguagem:

FAMILIA

Pai: "Luis", "Sousa Alves" 1955-06-30

Mae: "Rosa Maria", "Silva" 1955-08-19

Casamento: "Braga" 1980

Filhos: "João", "Nuno", "Miguel Maria".

FAMILIA

Pai: "Manuel Paulo", "Alves" 1956-07-20

Mae: "Matilde", "Sampaio Marques" 1958-12-19

Casamento: "Foz-do-Douro, Porto" 1978

Filhos: .

O Objetivo é desenvolver um processador para essa linguagem que reconheça uma frase válida e produza algumas informações sobre a família, como por exemplo uma lista de filhos com os apelidos herdados com o formato que se exemplifica abaixo:

Família 1:

João Silva Sousa Alves

Nuno Silva Sousa Alves

Miguel Maria Silva Sousa Alves

Família 2:

sem filhos

Neste contexto responda às seguintes alíneas:

- Especifique uma gramática para a linguagem indicando o conjunto de símbolos terminais (T), o conjunto de símbolos não-terminais (N), o axioma (S) e o conjunto de produções (P);
- Desenhe a árvore de derivação para a frase exemplo, indicando as produções usadas;
- Especifique em flex o analisador léxico para o processador pretendido;
- Especifique em yacc o parser pretendido;
- Acrescente-lhe as ações semânticas necessárias para gerar a saída pretendida como acima ilustrado e para garantir que ambos os progenitores tenham no mínimo 18 anos à data do casamento.

Questão 4: Linguagens Independentes de Contexto ($5v = 1.5 + 1 + 1.5 + 1$)

Considere a seguinte gramática:

Percursos	→	Pontos Caminhos	0	PONTOS Pontos Caminhos Caminhos 0
Pontos	→	Ponto Ps	1	
Ps	→	Ponto Ps	2	
		ε	3	
Ponto	→	id '(' X ',' Y ')'	4	
X	→	num	5	
Y	→	num	6	
Caminhos	→	Caminho '!' Cs	7	
Cs	→	Caminho '!' Cs	8	
		ε	9	
Caminho	→	id MAIORES num DIFICULD Grau PASSOS Ids id		
Ids	→	id Resto	11	
Resto	→	'(' id Resto	12	
		ε	13	
Grau	→	FACIL MEDIO DIFICIL		
		14 15 16		

E responda às seguintes alíneas:

- a) Calcule o *lookahead* de cada uma das produções;
- b) Prove que a gramática é LL(1) construindo a respetiva Tabela LL(1);
- c) Escreva em C as funções de reconhecimento dos símbolos não-terminais "Pontos", "Ps" e "Ponto" que fariam parte do parser recursivo descendente desta gramática;
- d) Reescreva a gramática usando recursividade à esquerda de modo a obter uma gramática equivalente à primeira mas mais pequena.