Processamento de Linguagens (e Compiladores)
LEI + LCC

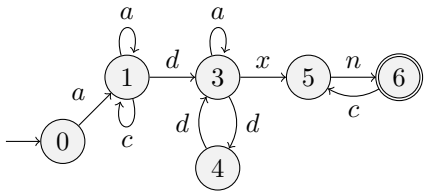
11 de Julho de 2012, 2 horas Exame

Questão 1 (Autómatos) Apresente um autómato determinístico correspondente a:

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

b). $(d b)^+ a^*$

c). Qual a expressão regular correspondente ao seguinte autómato:



Questão 2 (Analisador léxico flex) Pretendemos construir um compilador de C. Para tal, precisamos de um analisador léxico que retorne um código (inteiro) para cada símbolo terminal e que preencha o respectivo valor.

Construa esse analisador léxico flex mas considerando apenas:

- algumas palavras reservadas (nomeadamente while, for, if)
- alguns símbolos especiais (`* + ; { } ++ == +=`);
- identificadores
- *floats*
- constantes char (ex. 'a')

Os comentários C e os espaços brancos, devem ser ignorados.

Questão 3 (filtro flex) Dado um texto HTML, calcula o seu índice (extraindo apenas h1,h2,h3), e escreva na saída os títulos de seções, numerando-as como aparece nos livros e manuais.

Questão 4 (Gramáticas e reconhecedores) Considere a seguinte gramática

```
1  registoParoquial →  registoParoquial registo
2                   |

3  registo     → "#" data batismo
4              | "#" data nascimento

5  batismo     → "batismo-de:" nome

6  nascimento → "nascimento:" nome "mãe:" nome "pai:" nome

7  nome → ID
```

:

a). Apresente uma frase *típica* desta linguagem e apresente também a frase mais curta.

b). Desenhe o autómato LR0 (apenas o estado inicial e mais 4) e indique se há conflitos LR0.

c). Diga porque é que esta gramática não é LL1 e escreva uma que cobrindo a mesma linguagem, seja LL1.

Questão 5 (Flex,Yacc) Pretendemos construir um preprocessador para htmlcdc (html cheio de chavetas).

```
htmlcdc                              html
-------------------------------------------------------------------------
                                 → <html><body>  ...adicionado no início

texto                            → texto
h1{Prato do dia}                 → <h1>Prato do dia</h1>
ul{                              → <ul>
 li{sopa | frango | b{pudim}} →   <li>sopa</li>
                                     <li>frango</li>
                                     <li><b>pudim</b></li>
  }                              → </ul>

                                 → </html></body> ...adicionado no fim
```

Esse tradutor irá usar flex e **yacc**, e deverá aceitar um texto htmlcc e gerar html de acordo com o exemplo apresentado.

Apresente apenas o Yacc considerando que:

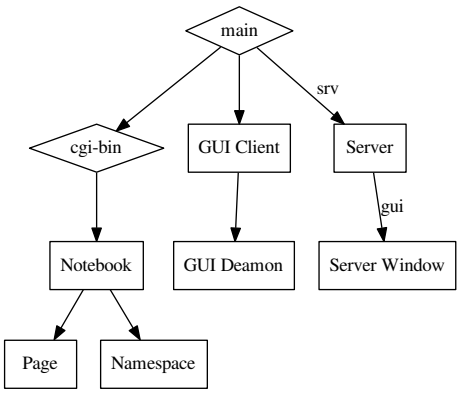
- o conjunto dos terminais será `T = TXT ID '{' '|' '}'`, sento TXT e ID strings.
- um htmlcdc é constituído por texto e anotações
- as anotações podem actuar sobre um elemento htmlcdc ou
- as anotações podem actuar sobre listas (anotam cada um dos seus elementos)

Questão 6 (gramáticas) Escreva a gramática da linguagem **minidot** (subconjunto do dot/graph-viz correspondente ao exemplo abaixo):

```
digraph g {                                  // comentários tipo c++
  node  [shape = box] ;                      // nós são rectangulares

  main  [shape = diamond] ;                  // nós com atributos específicos
  cgi   [label = "cgi-bin", shape = diamond];

  main -> "GUI Client" ;                     // ramos
  main -> cgi -> Notebook ;
  "GUI Client" -> "GUI Deamon" ;
  Notebook -> Page ;
  Notebook -> Namespace ;
  main   -> Server          [label ="srv"] ;
  Server -> "Server Window" [label ="gui"] ;
}
```



A gramática apresentada deverá:

- identificar os terminais, não terminais, axioma, e produções (escolha bons identificadores)
- ser capaz de reconhecer o exemplo apresentado,
