

Projetos de programação:

- Grupos com 1, 2 ou 3 membros;
- Os projetos serão sorteados em sala de aula no dia 25 de maio de 2010.
- Prazo para entrega: 08 de junho de 2010 (duas semanas);
- Todos os projetos deverão ser apresentados pelo grupo em sala de aula no dia da entrega;
- As apresentações terão a duração de 5 minutos cada e deverão demonstrar a funcionalidade do programa como exemplos, além comentar sobre dificuldades encontradas na construção do mesmo e soluções obtidas.

Usar a seguinte codificação para autômatos finitos:

- $Q = \{00,01,\dots,49\}$
- $\Sigma = \{a,b,\dots,z\}$
- q_0 é 00 ou 50
- $\delta = \{q_i\sigma q_j, \dots\}$
- $F = \{50,51,\dots,99\}$
- Transição em vazio é representada por q_iq_j apenas
- Usar apenas 00 (se inicial e não-final) ou 50 (se inicial e final), nunca os dois
- Transições são separadas por "#"
- Exemplo: 00a00#00b01#01b01#01c51 (aceita a^*bb^*c)

Usar a seguinte codificação para gramáticas:

- $N = \{A,B,\dots,Z\}$
- $\Sigma = \{a,b,\dots,z\}$
- Raiz é S
- $P = \{X \rightarrow \dots, Y \rightarrow \dots, \dots\}$ onde $X,Y \in N, \dots \in (N \cup \Sigma)^*$
- Regra vazia é representada por $X \in N$ apenas
- Regras são separadas por "#"
- Exemplo: SaSa#S#Sb#Sc (gera a linguagem $a^i(b|c|\varepsilon)a^i, i \geq 0$)

Observações gerais para os projetos:

- Devem aceitar os dados de entrada na forma de um arquivo texto sem formatação, conforme especificado acima;
- Devem gerar os dados de saída na forma de um arquivo texto sem formatação, conforme especificado acima;
- Os programas deverão validar as entradas conforme as exigências do algoritmo que for aplicado, erros ou divergências deverão ser informados;
- Devem ser entregues (em formato impresso e digital):
 - código executável;
 - código-fonte comentado;
 - fundamentação teórica;
 - documentação sobre a implementação (estruturas de dados e algoritmos utilizados);
 - descrição das ferramentas utilizadas (linguagens, processadores, ambientes e respectivas versões);
 - manual de instalação;

- manual de usuário com descrição da interface;
- exemplos de entrada e respectivas saídas.

Projetos:

1. Entrada: GLD que gera L
Saída: GLUD que gera L
2. Entrada: AF que aceita L
Saída: AF sem transições em vazio que aceita L
3. Entrada: AF que aceita L
Saída: AF determinístico que aceita L
4. Entrada: GLUD que gera L
Saída: AF que aceita L
5. Entrada: AF que aceita L
Saída: GLUD que gera L
6. Entrada: AF que aceita L
Saída: AF sem estados inúteis e sem estados inacessíveis que aceita L
7. Entrada: AF que aceita L
Saída: AF mínimo que aceita L
8. Entrada: AF que aceita L
Saída: AF que aceita L^R
9. Entrada: AF que aceita L
Saída: AF que aceita Σ^*-L
10. Entrada: AF que aceita L
Saída: SIM se L é vazia, NÃO caso contrário
11. Entrada: AF que aceita L
Saída: SIM se L é infinita, NÃO caso contrário
12. Entrada: GLC que gera L
Saída: GLC isenta de símbolo inúteis que gera L
13. Entrada: GLC que gera L
Saída: GLC isenta de símbolo inacessíveis que gera L
14. Entrada: GLC que gera L
Saída: GLC isenta de regras vazias que gera L
15. Entrada: GLC que gera L
Saída: GLC isenta de regras unitárias que gera L