------ LFA 1

Prof. Leandro I. Pinto

Nome: _____

1) Construa AFD para as seguintes Linguagens:
   a) L = (a+b)(a+b)*
   b) L = (ba*b+a)*
   c) $L = \{a^n(ba)^*b^n \mid n = 2\}$
   d) $L = \{w|w \in \{a,b,c\}*, |w|_a mod\ 2 = 0\ e\ |w|_b mod\ 2 = 1\}$

2) Um programador fará uma ferramenta de software capaz de reconhecer uma linguagem de programação simples. Para ganhar tempo, ele utilizará um programa chamado Bison, o qual gera um executável capaz de reconhecer linguagens a partir de uma gramática. Abaixo é mostrada a linguagem a ser reconhecida, especifique uma gramática regular para tal linguagem.

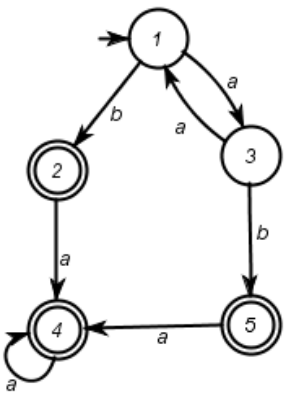
   <...>: representa um símbolo;
   *: concatenação múltipla
   +: ou

   Um arquivo nessa linguagem pode conter o seguinte:

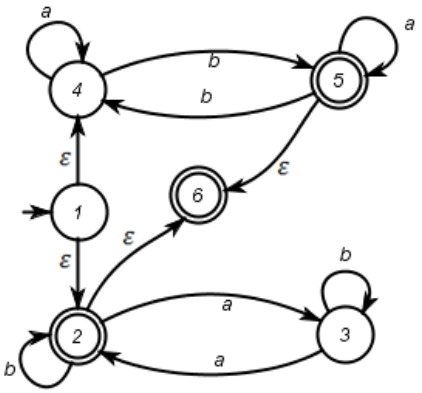
```
(<função:>
(
  (<se> <expr> <então> <atrib>* <fim>) +
  (<enquanto> <expr> <entao> <aatrib>* <fim>)
)*
<fim>)*
```

3) Prove que a seguinte linguagem é regular: $L = \{w \in \{a,b\}*|w = \sigma_1\sigma_2 \dots \sigma_3\ e\ \sigma_i \neq a\ se\ i\ mod\ 2 = 0\}$.

4) Através do algoritmo de minimização, encontre o autômato mínimo para o autômato a seguir:



5) Encontre um AFN e um AFD equivalentes para o seguinte AFNe:



Minimização de Autômatos

1. Deve ser AFD, não ter estados inacessíveis e a função programa deve ser total;
2. Construir uma tabela relacionando os estados;
3. Marcar estados trivialmente não equivalentes;
4. Marcar estados não equivalentes para cada par $\{q_u, q_v\}$ não marcado, para cada símbolo $a \in \Sigma$, suponha que $\delta(q_u,\ a) = p_u$ e $\delta(q_v, a) = p_v$;
   - $p_u = p_v$, então q_u e q_v são equivalentes para 'a';
   - $p_u \neq p_v$ e $\{p_u, p_v\}$ não está marcado, então $\{q_u, q_v\}$ é incluído numa lista a partir de $\{p_u, p_v\}$;
   - $p_u \neq p_v$ e o par $\{p_u, p_v\}$ está marcado, então:
     - $\{q_u, q_v\}$ deve ser marcado;
     - Marcar todos da lista encabeçada por $\{q_u, q_v\}$, se houver;
5. Unificar os estados equivalentes

Conversão AFNe para AFN

Seja $M = \{\Sigma, Q, \delta, q0, F\}$ um AFNe qualquer. Seja $M' = \{\Sigma, Q, \delta', q0, F'\}$ um AFN construído a partir de M como segue:

δ': tal que $\delta' \colon Q \times \Sigma \to 2^Q$ onde $\delta'(q, a) = \delta(\{q\}, a)$
F': Conjunto de todos os estados q pertencentes a Q tal que algum elemento do FECHO-e(q) pertence a F.

Conversão AFN para AFD

Seja $M = \{\Sigma, Q, \delta, q0, F\}$ um AFNe qualquer. Seja $M' = \{\Sigma, Q', \delta', \langle q0\rangle, F'\}$ um AFN construído a partir de M como segue:

Q': Conjunto de combinações de estados de Q, denotado por $\langle q1q2 \dots qn\rangle$;
δ': tal que $\delta(\langle q1 \dots qn\rangle, a) = \langle p1 \dots pm\rangle$ se, e somente se $\delta(\{q1, \dots, qn\}, a) = \{p1, \dots, pm\}$;
$\langle q0\rangle$: Estado Inicial;
F':Conjunto de todos os estados $\langle q1q2 \dots qn\rangle$ pertencentes a Q' tal que alguma componente qi pertence a F;

AFNe para Gramática

Teorema 2.19 Linguagem Regular → Gramática Regular.

Se L é Linguagem Regular, então existe G, Gramática Regular que gera L.

Prova: Se L é uma Linguagem Regular, então existe um AFD M = (Σ, Q, δ, q_0, F) tal que ACEITA(M) = L. A idéia central da demonstração é construir uma Gramática Linear à Direita G a partir de M tal que GERA(G) = ACEITA(M), ou seja, cuja derivação simula a função programa estendida. Seja G = (V, T, P, S) tal que:

V = Q ∪ {S}
T = Σ

e P é construído como segue (suponha q_i, q_k elementos de Q, q_f elemento de F e a elemento de Σ):

Transição	Produção
-	$S \to q_0$
-	$q_f \to \varepsilon$
$\delta(q_i, a) = q_k$	$q_i \to aq_k$