MAC-414 – Tomatinhos — Prova 1 – 29/9/2009
Resoluções

1. Para cada uma das afirmativas abaixo, prove, se for verdadeira, e dê um contra-exemplo se for falsa (obs.: os quatro ítens têm soluções curtas). Chutes não serão considerados.

   (a) Um AD reconhece a linguagem vazia se e só se nenhum estado final é acessível.

   Verdadeiro: Isto é caso particular do ítem seguinte.

   Erro comum: confundir linguagem vazia, palavra vazia e $\{\lambda\}$.

   (b) Um AND reconhece a linguagem vazia se e só se nenhum estado final é acessível.

   Verdadeiro:

   - (só se) Se algum estado final é acessível, o rótulo de algum passeio $S \to F$ é reconhecido e a linguagem é não vazia.
   - (se) Se nenhum estado final é acessível, nenhum passeio é vitorioso, e o autômato não reconhece nenhuma palavra.

   (c) Um AD reconhece Σ^* e só se todo estado acessível é final.

   Verdadeiro:

   - (se) Se todo estado acessível é final, $\delta(s, x) \in F$ para toda palavra x. Logo, toda palavra é aceita.
   - (só se) Se algum estado não final q é acessível, então existe uma palavra x tal que $\delta(s, x) = q$; portanto x não é aceita.

   (d) Um AND reconhece Σ^* e só se todo estado acessível é final.

   Tanto *se* quanto *só se* são falsos (bastava provar que um deles é falso para falsificar a afirmativa - e para provar que é falso, basta um contra-exemplo).

   - (se) Contra-exemplos: tome um AND em que todos os estados são acessíveis e finais. Acrescente uma letra nova ao alfabeto, sem introduzir novas arestas. Claro que nenhuma palavra contendo a nova letra será reconhecida. Outro tipo de contra-exemplo, que eu vi numa das provas: o autômato da Q3.
   - (só se) Comece com um estado que é ao mesmo tempo inicial e final, e com um laço para cada letra do alfabeto. Introduza à vontade estados acessíveis, não finais. Apesar dele, Σ^* é reconhecido.

2. Mostre que a linguagem sobre $\Sigma = \{a, b\}$ descrita por

$$\{x : |x|_a = |x|_b\} + (a + b)^*(aa + bb)(a + b)^*$$

é regular; ou seja, encontre uma expressão regular para ela e prove que está correta. Relembrando: $|x|_a$ é o número de ocorrências da letra a na palavra x.

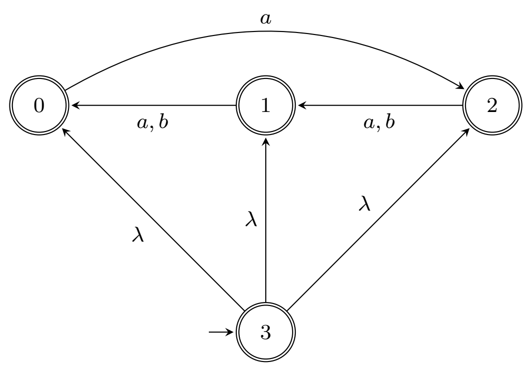
Sejam $L_1 = \{x : |x|_a = |x|_b\}$ e $L_2 = (a + b)^*(aa + bb)(a + b)^*$. Queremos uma ER para $L_1 \cup L_2$. Mas $L_1 \cup L_2 = (L_1 \cap \overline{L_2}) \cup L_2$.

L_2 consiste das palavras em que ocorrem duas letras sucessivas iguais. Assim, $\overline{L_2}$ consiste da palavras em que as letras se alternam. Para essas palavras estarem em L_1, precisam terminar pela letra diferente da com que começam. Logo, $L_1 \cap \overline{L_2} = (ab)^* + (ba)^*$, e a ER completa é

$$(ab)^* + (ba)^* + (a + b)^*(aa + bb)(a + b)^*.$$

3. Desenhe o grafo do seguinte autômato não determinístico:

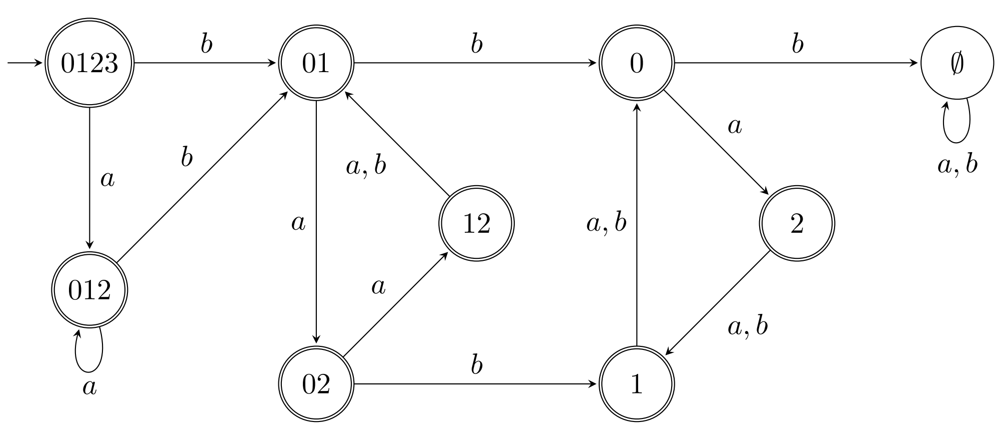
$$\begin{array}{rcl} \mathcal{A} & = & (K, \Sigma, \Delta, S, F) \\ K = F & = & \{0,1,2,3\} \\ S & = & \{3\} \\ \Sigma & = & \{a,b\} \\ \Delta & = & \{(i,\sigma,i-1) \mid \sigma = a,b, \quad i = 1,2\} \cup \\ & & \{(3,\lambda,i) \mid i = 0,1,2\} \cup \{(0,a,2)\}. \end{array}$$



Para cada uma das palavras abaixo, decida se está em $L(\mathcal{A})$, justificando sua decisão:

Existem pelo menos três jeitos naturais de resolver esta questão. Vou omitir os detalhes, que são chatos.

- Na raça: como o único não determinismo desse autômato está no estado 3, ao qual nunca se volta, basta seguir cada palavra a partir dos estados $0, 1, 2$ e ver se dá para terminar. A única que termina um passeio vitorioso é a do ítem (b), começando do estado 1.
- Aplicando o simulador: é só seguir o algoritmo.
- Construindo a parte acessível do AD correspondente, e processando as palavras nesse autômato. Aí está o AD:



(a) *baaaababa*
Rejeita.

(b) *ababaabaab*
Aceita.

(c) *ababaaaabab*
Rejeita.

4. Para cada uma das linguagens abaixo sobre $\Sigma = \{a, b\}$, escreva uma expressão regular que a descreva. Explique sua construção.

(a) $L1 = \{\ x \in \Sigma^* :$ os segmentos aa e bb não ocorrem em $x\ \}$

As letras devem se alternar, ou seja, as palavras têm forma $ababa\ldots$, podendo começar e acabar com qualquer letra. Isto dá

$$(ab)^* + b(ab)^* + (ab)^*a + b(ab)^*a = (\lambda + b)(ab)^*(\lambda + a)$$

(b) $L2 = \{\ x \in \Sigma^* : x$ tem um número ímpar de a's ou um número par de b's $\}$

Para número par de b's, eles devem ocorrer aos pares, com qualquer quantidade de a's entre eles. O mais simples é $(a^*ba^*b)^*a^*$ (ou seu reverso).

Para número ímpar de a's, podemos começar pela expressão para número par, que vem da anterior: $(b^*ab^*a)^*b^*$. A ela acrescentamos mais uma ocorrência de a: $(b^*ab^*a)^*b^*ab^*$.

Como o que foi pedido foi uma coisa **ou** outra, o que se quer é a união:

$$(b^*ab^*a)^*b^*ab^* + (a^*ba^*b)^*a^*.$$