

Teoria dos Autômatos e Linguagens Formais

lista de exercícios 11

1. Raciocinando sobre linguagens

Descreva procedimentos para construir autômatos para as seguintes linguagens

- a) palavras que não podem ser formadas pela concatenação dos blocos **ab**, **baa**, **bab**,
mas pode ser formadas pela concatenação dos blocos **ba**, **abb**, **aba**
- b) palavras onde: se o padrão **aba** aparece então o padrão **bab** também deve aparecer;
mas se o padrão **aba** não aparece então o padrão **bab** pode aparecer ou não
- c) palavras que nem podem ser formadas pela concatenação dos blocos **ab**, **aa**, **ba**
e nem possuem blocos de **a**'s de tamanho ímpar
- d) palavras onde todo **a** é precedido ou sucedido por um bloco de **b**'s de tamanho ímpar

2. Prefixos e sufixos próprios

Em geral, nós assumimos que a própria palavra **p** é um prefixo e um sufixo de si mesma.

E, quando nós temos um prefixo ou sufixo de **p** que é diferente de **p**, nós dizemos que ele é um prefixo ou sufixo *próprio*.

Agora, dada uma linguagem qualquer L (reconhecida por um autômato M), nós podemos considerar as linguagens

$\text{Pref-p}(L)$: conjunto dos prefixos próprios das palavras de L

$\text{Suf-p}(L)$: conjunto dos sufixos próprios das palavras de L

Você consegue construir um autômato que reconhece a linguagem $\text{Pref-p}(L)$?

A princípio, isso parece uma tarefa fácil:

- basta construir um autômato M_{Pref} que reconhece a linguagem $\text{Pref}(L)$
- e a seguir construir um autômato que reconhece a diferença $\text{Pref}(L) - L$;
por meio do produto de M e M_{Pref}

Mas, essa ideia não funciona ...

Quer dizer, uma palavra de L pode estar na linguagem $\text{Pref-p}(L)$ se ela for um prefixo próprio de outra palavra de L .

Por exemplo, se $L = (\text{aba} \cup \text{ab})^*$, então **aba** é uma palavra de L e também é um prefixo de **abaaba** que é outra palavra de L . Logo, **aba** está em $\text{Pref-p}(L)$.

E então, você consegue construir um autômato que reconhece a linguagem $\text{Pref-p}(L)$?

E você consegue construir um autômato que reconhece a linguagem $\text{Suf-p}(L)$?

3. Alternação bagunçada

Considere duas linguagens L_1 e L_2 reconhecidas por autômatos M_1 e M_2 , respectivamente.

A linguagem $\text{Alt-b}(L_1, L_2)$ é definida da seguinte maneira

- para cada par de palavras

$$x = x_1 x_2 \dots x_m \in L_1 \qquad y = y_1 y_2 \dots y_n \in L_2$$

nós podemos formar palavras p alternando os símbolos de x e y de maneira arbitrária;
por exemplo

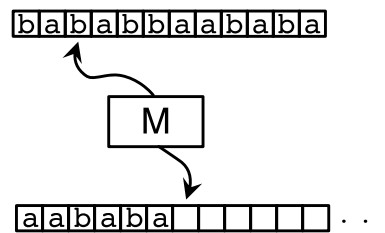
$$p = x_1 x_2 y_1 x_3 \dots x_m y_{n-1} y_n$$

e essas são as palavras que pertencem a $\text{Alt-b}(L_1, L_2)$.

Você consegue construir um autômato que reconhece a linguagem $\text{Alt-b}(L_1, L_2)$?

4. O autômato tradutor

Considere um autômato que ao mesmo tempo em que vai lendo os símbolos de uma palavra de entrada, vai também escrevendo símbolos em uma fita de saída



As transições do autômato tradutor possuem o seguinte formato

$$q_i \xrightarrow{a;b} q_j \qquad q_i \xrightarrow{a;bab} q_j \qquad q_i \xrightarrow{a;e} q_j$$

Quer dizer, ao ler um símbolo na palavra de entrada, o autômato tradutor pode escrever na fita de saída

- um outro símbolo
- uma palavra
- ou nada (i.e., a palavra vazia)

Da mesma maneira que nos autômatos comuns, nós definimos a linguagem reconhecida pelo autômato tradutor como o conjunto de palavras que fazem com que a sua computação termine em um estado final.

E nós definimos a *linguagem produzida* pelo autômato tradutor como o conjunto de palavras que ele escreve na fita de saída ao ler uma palavra que ele aceita na fita de entrada.

Argumente que nós sempre podemos construir um autômato que reconhece a linguagem produzida por um autômato tradutor.