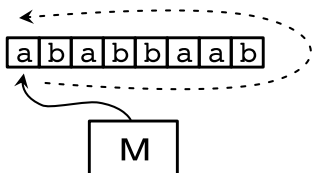


O autômato vai-e-volta

A nossa próxima observação é a seguinte

→ andar para a frente e depois para trás também é a mesma coisa que andar só uma vez para frente

Quer dizer, agora nós vamos examinar os autômatos *vai-e-volta*, que lêem a palavra de entrada uma vez andando para frente, e uma segunda vez andando para trás



Por exemplo, a gente pode reconhecer a seguinte linguagem dessa maneira

→ palavras que possuem uma quantidade par de a's e uma quantidade de b's múltipla de 3

Quer dizer, basta verificar a condição associada aos a's na computação da ida, e verificar a condição associada aos b's na computação da volta.

Mas isso não faz a menor diferença.

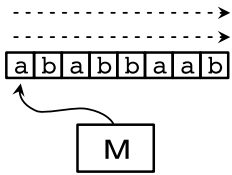
Quer dizer,

→ Tudo o que os autômatos *vai-e-volta* fazem também pode ser feito por um autômato comum

A intuição é a seguinte.

Nós acabamos de ver que andar para trás é a mesma coisa que andar para frente.

Logo, nós podemos pensar no autômato *vai-e-volta* como um autômato *vai-vai*



Quer dizer, esse autômato anda duas vezes para frente: uma vez realizando a computação da ida, e a outra vez simulando a computação da volta.

Ora, mas andar duas vezes para frente é a mesma coisa que realizar duas computações sobre a mesma palavra de entrada.

E na aula passada, nós vimos que um autômato comum pode realizar duas computações simultâneas sobre a mesma palavra de entrada — (*por meio da técnica do autômato produto*)

Logo, nós sempre podemos construir um autômato comum que faz a mesma coisa que um autômato *vai-e-volta*.

Mas, essa ideia ainda não funciona ...

Quer dizer, no autômato *vai-e-volta* a computação da volta acontece depois da computação da ida.

Daí que, a computação da volta pode fazer coisas diferentes, dependendo do resultado da computação da ida.

E é por isso que a gente não pode fazer as duas computações ao mesmo tempo.

Ou será que pode?

Vamos raciocinar um pouquinho.

Quer dizer, de que maneira a computação da ida pode afetar a computação da volta?

Bom, o nosso autômato não escreve.

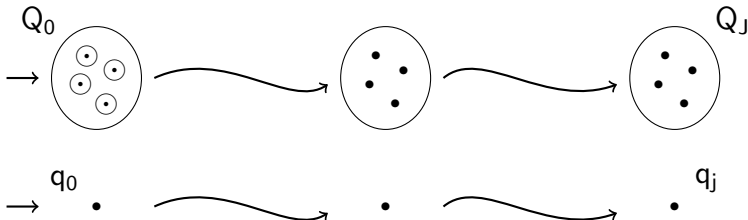
Logo, a palavra de entrada é sempre a mesma.

Mas, o estado em que a computação da ida termina é o estado em que a computação da volta começa.

E é assim que as duas se comunicam!

Certo.

Agora está na hora de visualizar como é que o autômato *vai-vai* funciona



Quer dizer, são duas linhas de execução

- a computação (determinística) da ida
- e o cálculo (não-determinístico) dos caminhos da volta que terminam em algum estado final

Quando nós chegamos ao final da palavra de entrada, nós descobrimos o estado q_j em que a computação da ida termina.

E esse também é o estado em que a computação da volta começa.

Por outro lado, Q_J é a coleção de estados onde a computação da volta deve começar para terminar em algum estado final.

Logo, o autômato *vai-e-volta* aceita a palavra de entrada quando $q_j \in Q_J$.

Essa é a verificação que o autômato *vai-vai* deve fazer.

Mas se o autômato *vai-vai* pode fazer isso, então um autômato comum também pode fazer

— (*por meio da técnica do autômato produto*)