

Mais autômatos finitos

Agora a gente já entendeu como os autômatos funcionam.

E daí, é fácil construir um autômato que realiza a seguinte tarefa

→ *Verificar se a quantidade de a's na palavra de entrada é par ou ímpar*

Bom, a gente já sabe que a resposta vai ser dada na forma de q_0 e q_1 .

Mas dessa vez, a leitura de um b não muda o jeito como as coisas estão.

Daí que, as regras que a gente vai usar são as seguintes

$$\begin{array}{ll} (q_0, a) \longrightarrow q_1 & (q_1, a) \longrightarrow q_0 \\ (q_0, b) \longrightarrow q_0 & (q_1, b) \longrightarrow q_1 \end{array}$$

Se a computação começa em q_0 , então

- a resposta q_0 significa que o número de a 's é par
- a resposta q_1 significa que o número de a 's é ímpar

E aqui nós já temos mais um autômato.

0.1 Diagrama de estados e transições

Nós dissemos há pouco que q_0 e q_1 são como luzinhas que ficam acendendo e apagando durante a computação.

Mas a terminologia correta é que q_0, q_1 são os *estados da computação* — (*porque eles indicam como as coisas estão*)

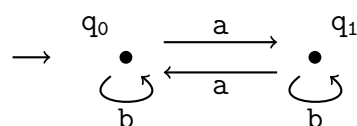
Daí a gente nota que as regras indicam como o estado da computação muda a medida que os símbolos vão sendo lidos

$$(q_i, a) \longrightarrow q_j \quad \text{e} \quad (q_i, b) \longrightarrow q_k$$

E por isso, elas são chamadas de *regras de transição* de estados — (*ou regras de mudança de estado*)

A computação do autômato é determinada completamente pelos estados e as regras de transição.

E a maneira mais conveniente de apresentar essa informação é por meio de um *diagrama de estados e transições*



onde a setinha ao lado de q_0 indica que ele é o *estado inicial* — (*i.e., o estado onde a computação começa*)