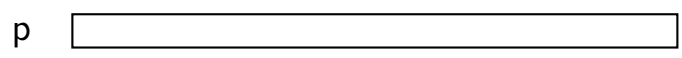
Considere a seguinte propriedade

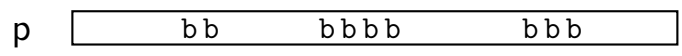
→ *Não existem* b*'s isolados na palavra*

Quer dizer, todo b deve ter um outro b ao seu lado.

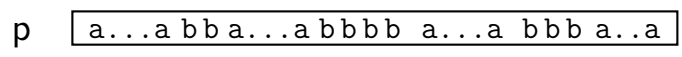
Agora imagine uma palavra p bem grande que tem essa propriedade



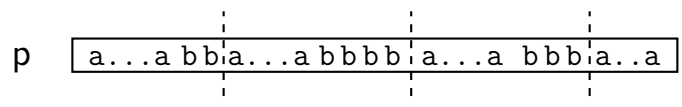
E a seguir imagine que nós estamos vendo todos os símbolos b da palavra p



Quer dizer, pela definição da propriedade, os b's sempre aparecem em blocos de tamanho ≥ 2.
Finalmente, como esses são todos os b's, todo o resto devem ser a's



Nesse ponto, a gente imagina que a palavra é quebrada logo após o último b de cada bloco



E assim a gente vê que essa palavra é uma

→ concatenação de blocos da forma $\underbrace{a..a}_{>0}\,\underbrace{b..b}_{\geq 2}$

seguida (opcionalmente) por um bloco de a's

Mas isso ainda não está completamente certo.
Quer dizer, a nossa palavra pode começar com b's.
E daí ela não corresponderia à descrição acima.
Só que, uma pequena esperteza resolve o problema.
Quer dizer, a gente pode reescrever a coisa assim

→ concatenação de blocos da forma $\underbrace{a..a}_{\geq 0}\,\underbrace{b..b}_{\geq 2}$

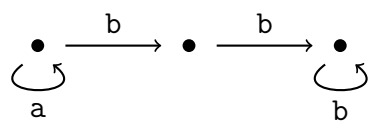
seguida (opcionalmente) por um bloco de a's

e agora a descrição funciona em todos os casos.

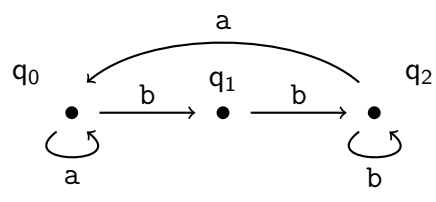
Certo.

O próximo passo é construir o autômato com base nessa descrição.
A gente começa construindo um caminho que lê blocos da forma

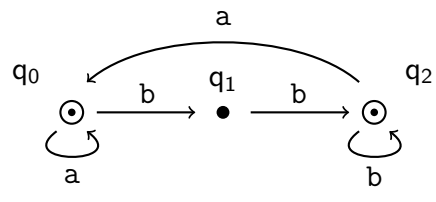
$\underbrace{a..a}_{\geq 0}\,\underbrace{b..b}_{\geq 2}$



Daí a gente fecha o laço para ler uma concatenação de blocos desse tipo

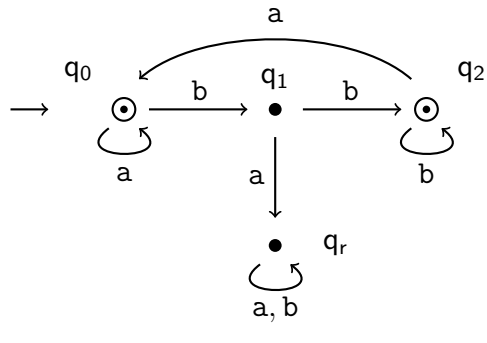


E marca os estados q_0 e q_2 como finais



Pronto.

Agora basta acrescentar o estado de rejeição para concluir a construção



◇