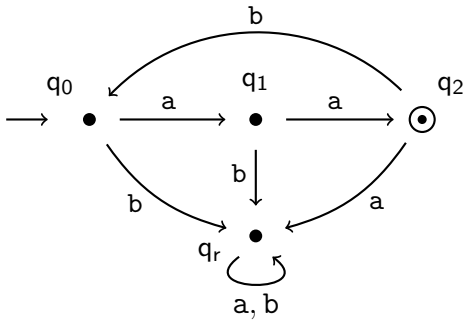
Teoria dos Autômatos e Linguagens Formais

aula 08: Autômatos e expressões regulares

1 Introdução

Considere o autômato abaixo



Qual é a expressão regular que descreve as palavras que esse autômato aceita?

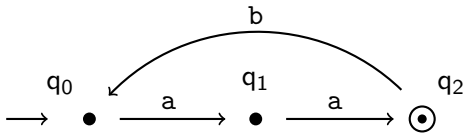
Bom, a gente já sabe analisar os caminhos e laços de um autômato — (*com a ajuda de um esquema*)

E a partir do esquema, a gente obtém a expressão regular.

Mas, hoje a gente vai fazer as coisas de maneira diferente.

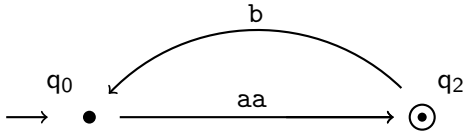
A primeira observação é que se o autômato cai no estado de rejeição, então a palavra não é mais aceita.

Daí, como a gente só está interessado nas palavras que o autômato aceita, o estado de rejeição pode ser removido do diagrama



A seguir a gente nota que se o autômato cai em $\mathtt{q}_1$ então ele precisa passar para $\mathtt{q}_2$ — (*porque se a computação termina em* $\mathtt{q}_1$ *a palavra não é aceita*)

E isso significa que agora a gente pode ver a coisa assim



Aqui a gente já está bem perto.

A última observação é a seguinte.

O autômato aceita as palavras que terminam em $\mathtt{q}_2$.

Em $\mathtt{q}_2$ o autômato pode voltar para $\mathtt{q}_1$.

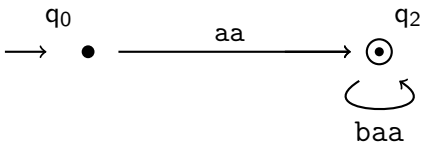
Mas se a computação termina em q_1 a palavra não é aceita.

Então, o autômato precisa voltar para q_2.

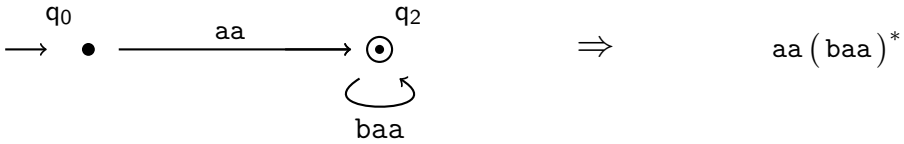
Em outras palavras, se o autômato sai de q_2 então ele precisa voltar para q_2.

Isso é um loop!

E agora, a gente pode ver a coisa assim

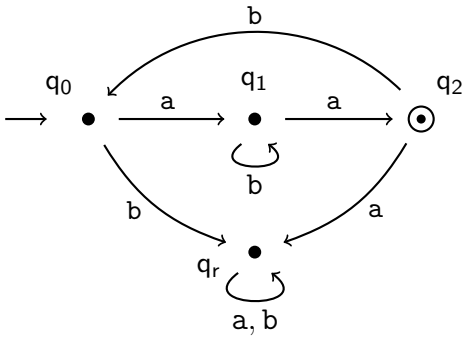


Só que isso já é o mesmo que uma expressão regular



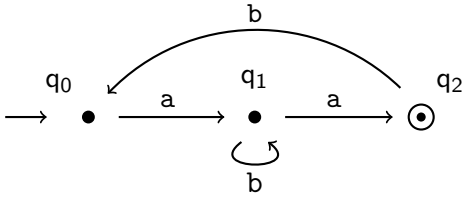
2 Autômatos e expressões regulares

Considere o autômato abaixo



Qual é a expressão regular que descreve as palavras que esse autômato aceita?

Bom, o primeiro passo é remover o estado de rejeição

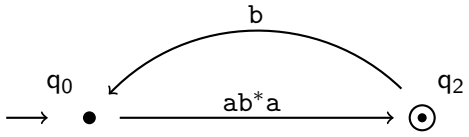


Daí, a gente nota que a computação não pode terminar em q_1 — (*porque ali a palavra não é aceita*)

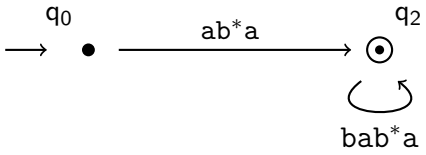
Mas, o autômato pode passar um tempinho ali — (*lendo o símbolo* b)

E depois ele deve ir para q_2.

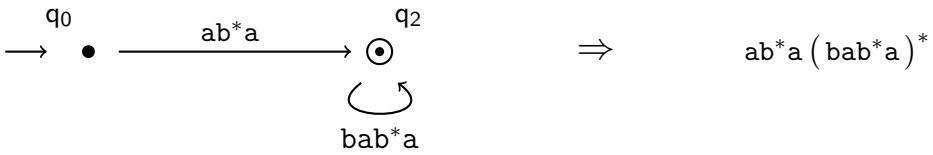
Então dessa vez, a gente pode reorganizar a coisa assim



Daí a gente já sabe que se o autômato sai de q_2 ele precisa voltar para q_2 — (*um loop*)



E isso já é o mesmo que uma expressão regular

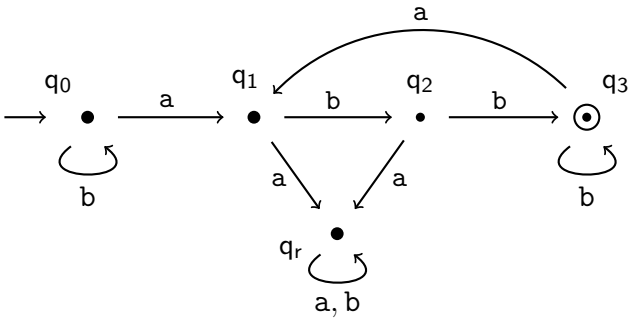


A seguir nós vamos ver mais exemplos.

Exemplos

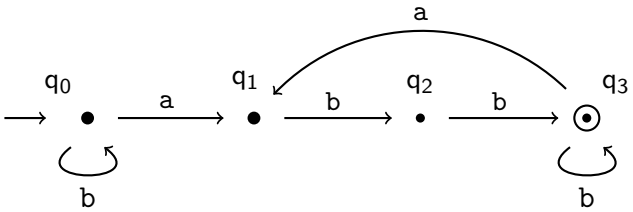
a. Todo a é seguir por ao menos 2 b's

Considere o autômato

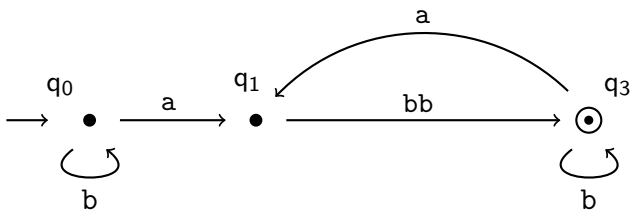


Qual é a expressão regular que descreve as palavras que esse autômato aceita?

Bom, o primeiro passo é remover o estado de rejeição

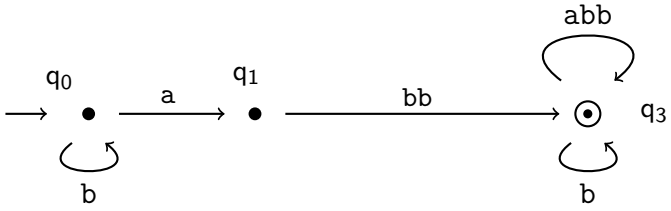


Daí, a gente já sabe que a computação não pode parar em q_2 — (*e remove ele do diagrama também*)

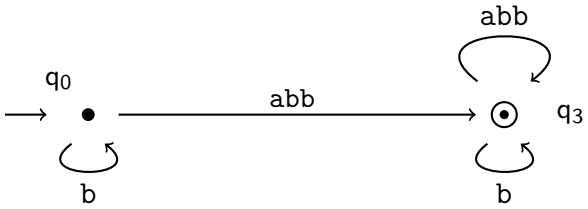


Nesse ponto, a gente vê que se o autômato sai de q_3 ele precisa voltar para q_3 — (*um loop*)

E reorganiza a coisa assim



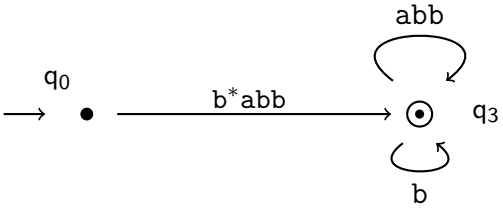
Daí, a gente remove o q_1 do diagrama



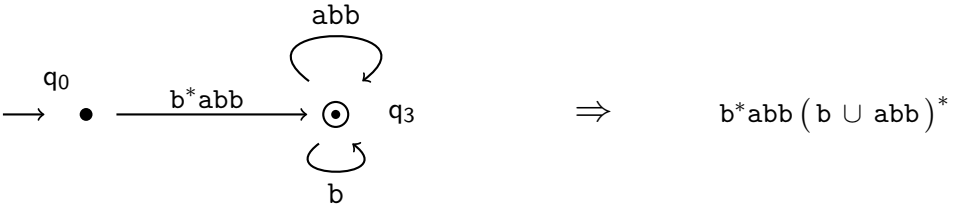
A última observação é que o autômato pode passar algum tempo em q_1 — (*lendo o símbolo* b)

Mas alguma hora ele precisa sair — (*para a palavra ser aceita*)

Então agora a coisa fica assim



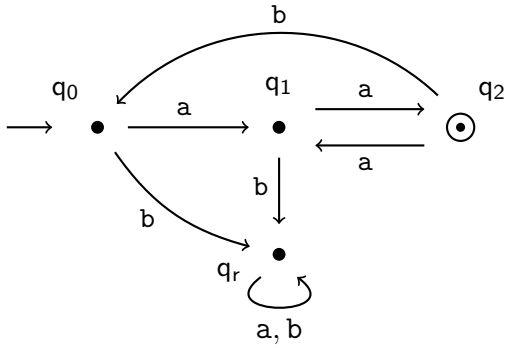
E isso já é a mesma coisa que uma expressão regular



◇

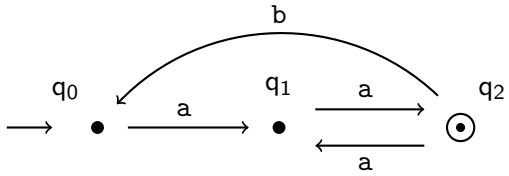
b. Dois loops no estado final

Considere o autômato



Qual é a expressão regular que descreve as palavras que esse autômato aceita?

Bom, o primeiro passo é remover o estado de rejeição

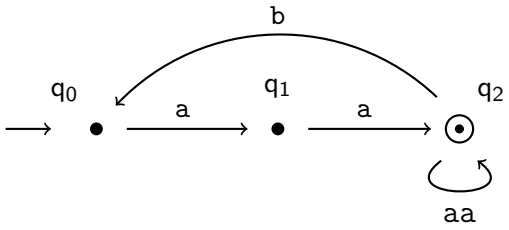


Daí, a gente nota que existem duas maneira de sair do estado q_2

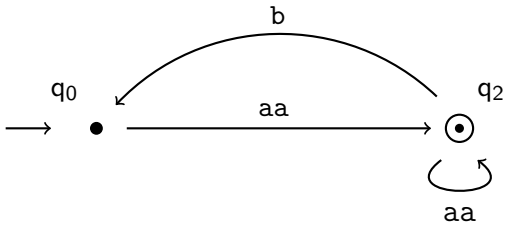
E nós vamos examiná-las uma de cada vez.

Se o autômato sai de q_2 para o estado q_1, então ele precisa voltar logo em seguida — (*um loop*)

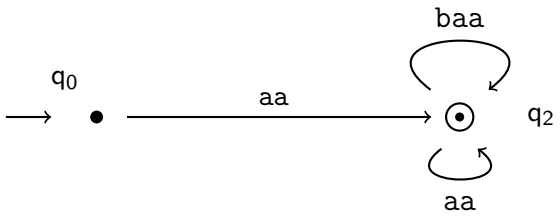
Daí, a gente reorganiza o esquema assim



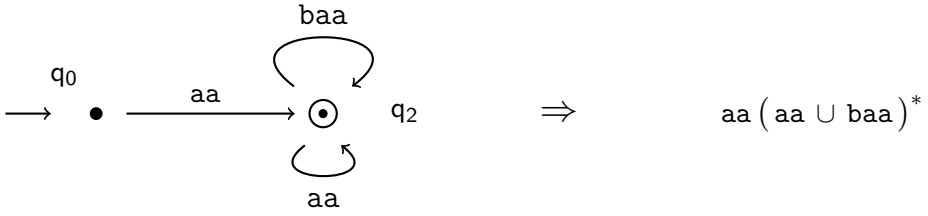
Agora, a gente pode remover o estado q_1



E daí, a gente vê outro loop em torno de q_2



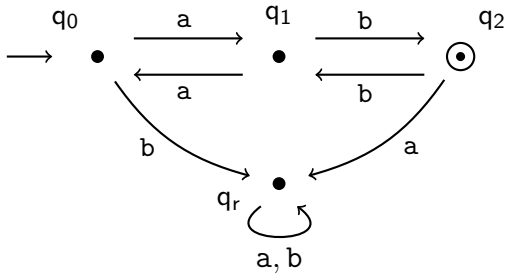
E isso já é a mesma coisa que a expressão regular



◇

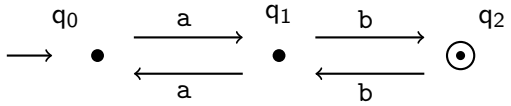
c. Dois loops no estado intermediário

Considere o autômato



Qual é a expressão regular que descreve as palavras que esse autômato aceita?

Bom, o primeiro passo é remover o estado de rejeição

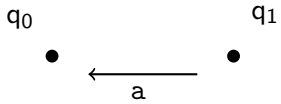


E agora?

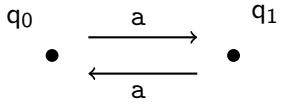
Bom, agora existem duas maneiras de raciocinar.

A primeira delas consiste em focar a atenção em uma transição do diagrama.

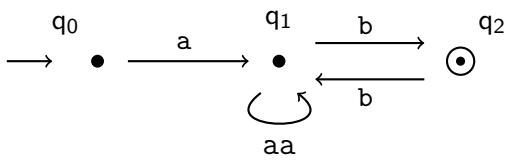
Por exemplo,



Daí, a gente nota que depois dessa transição, o autômato sempre volta para $\mathtt{q_1}$ — (*um loop*)



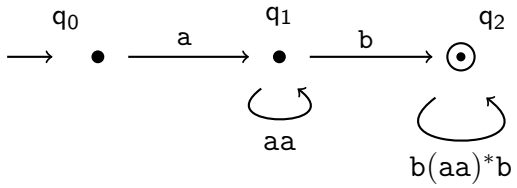
Logo, a gente pode remover a transição do diagrama e colocar o loop em seu lugar



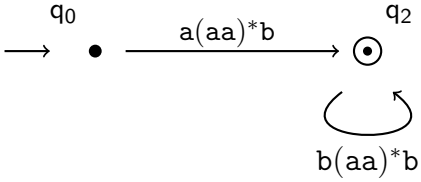
A seguir a gente nota que se o autômato sai de q_2 para q_1, então ele precisa voltar para q_2.

Mas antes disso, ele pode ficar um tempinho em q_1 — (*lendo blocos* aa)

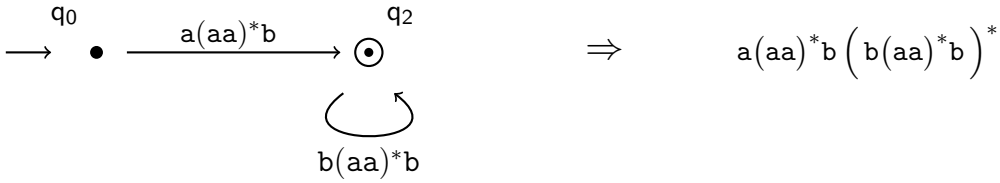
Então a gente pode reorganizar o diagrama assim



Agora, a gente pode remover o estado q_1

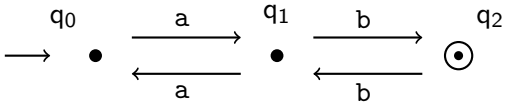


E isso já é a mesma coisa que a expressão regular



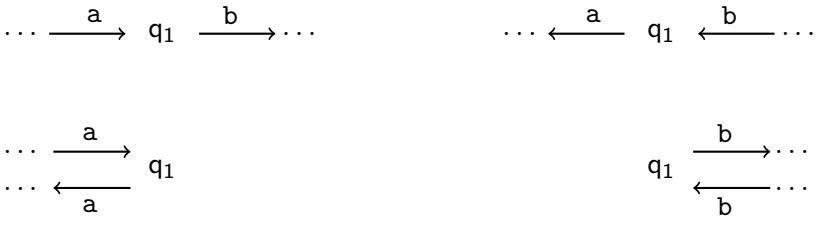
Mas, essa não é a única maneira de fazer as coisas.

Quer dizer, vamos olhar outra vez para o diagrama — (*sem o estado de rejeição*)



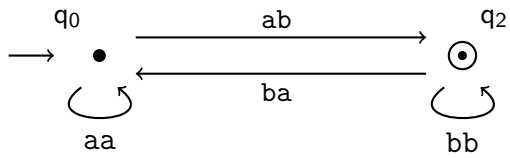
Daí, ao invés de focar na transição, a gente foca no estado q_1.

E vê que existem 4 trajetórias que passam por q_1

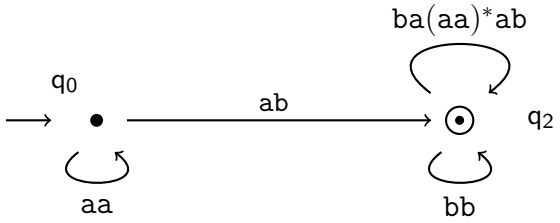


Então, a gente pode remover o estado q_1 do diagrama, e colocar essas trajetórias no seu lugar.

A coisa fica assim

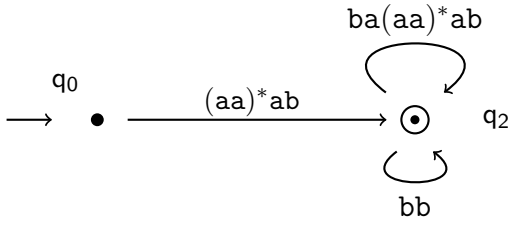


Daí, a gente vê o loop em torno do estado q_2

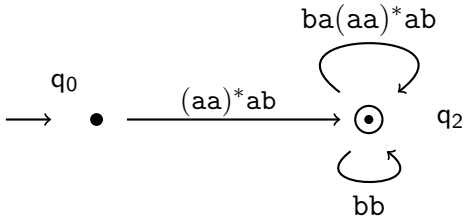


A seguir, a gente nota que o autômato pode passar algum tempinho em q_0 antes de ir para q_2 — (*lendo blocos* aa)

Mas alguma hora ele tem que ir para q_2



E isso já é a mesma coisa que uma expressão regular



$$\Rightarrow \qquad (aa)^*ab\Big(bb \cup ba(aa)^*ab\Big)^*$$

É engraçado que a gente obteve expressões regulares diferentes nos dois raciocínios

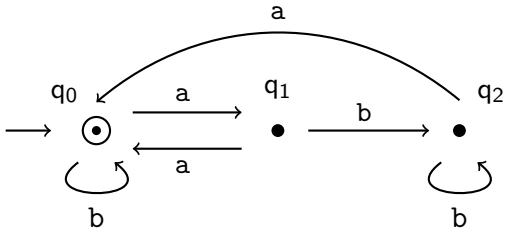
$$a(aa)^*b\Big(b(aa)^*b\Big)^* \qquad\qquad (aa)^*ab\Big(bb \cup ba(aa)^*ab\Big)^*$$

Mas, essas duas expressões regulares são a mesma coisa — (*porque?*)

◇

d. Loops no estado inicial-final

Considere o autômato



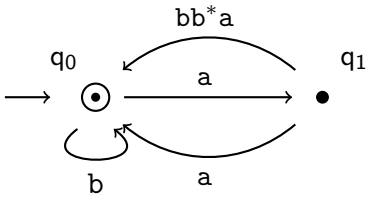
Qual é a expressão regular que descreve as palavras que esse autômato aceita?

Bom, esse autômato não tem um estado de rejeição.

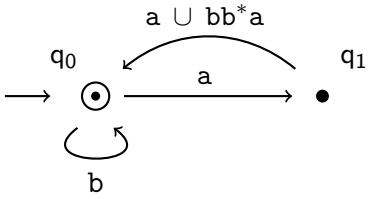
Então, a gente começa focando a atenção no estado $\mathtt{q}_2$.

Se o autômato chega em $\mathtt{q}_2$ (vindo de $\mathtt{q}_1$) ele precisa voltar para $\mathtt{q}_0$.

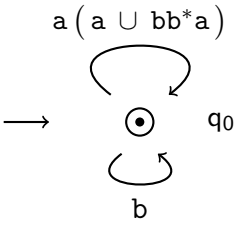
Então, a gente pode reorganizar o diagrama assim



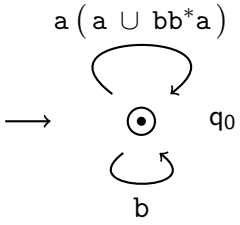
o que também pode ser apresentado assim



Agora, já é fácil ver o laço em torno de $\mathtt{q}_0$ — (*passando por* $\mathtt{q}_1$)



E isso já é a mesma coisa que uma expressão regular

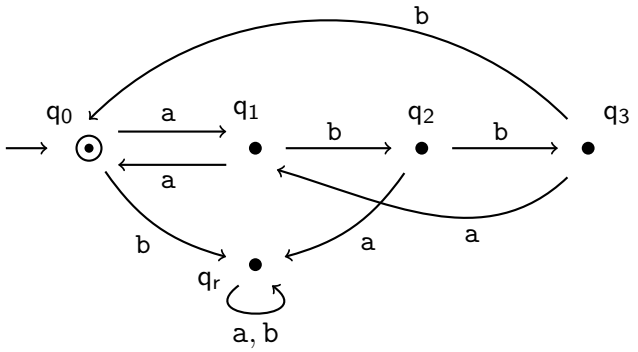


$$\Rightarrow \qquad \Big(\mathtt{b} \cup \mathtt{a}\big(\mathtt{a} \cup \mathtt{bb}^*\mathtt{a}\big) \Big)^*$$

◇

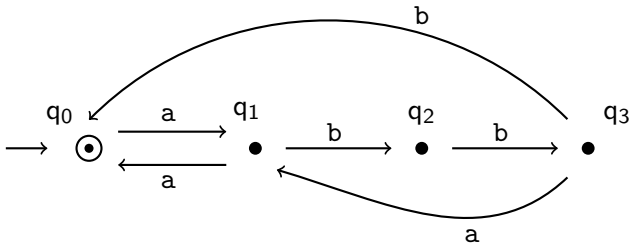
e. Enrolada

Considere o autômato

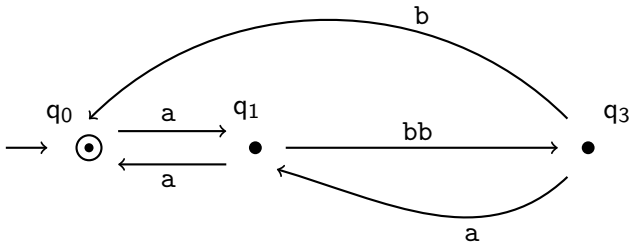


Qual é a expressão regular que descreve as palavras que esse autômato aceita?

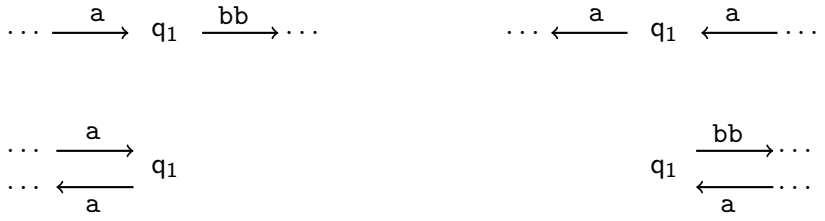
Bom, o primeiro passo é remover o estado de rejeição



Daí, é fácil remover o estado q_2

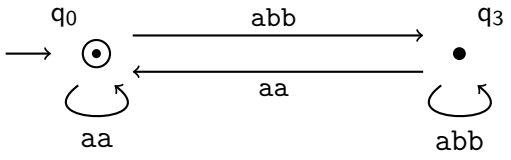


Nesse ponto, a gente visualiza as trajetórias que passam por q_1

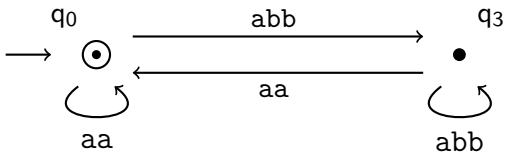


Então, a gente remove o estado q_1 do diagrama, e coloca as trajetórias no seu lugar

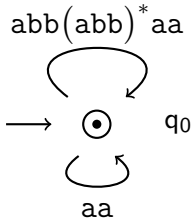
A coisa fica assim



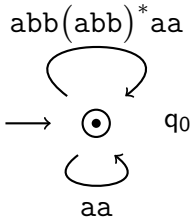
Aqui, a gente já vê o loop em torno de $\mathtt{q}_1$



Aqui, a gente já vê o laço em torno de $\mathtt{q}_0$ — (*passando por* $\mathtt{q}_3$)



E isso já é a mesma coisa que uma expressão regular



$$\Rightarrow \qquad \Big(\mathtt{aa} \cup \mathtt{abb} \big(\mathtt{abb} \big)^* \mathtt{aa} \Big)^*$$

$\diamond$