

## Propriedades de Fechamento

Se  $A$  e  $B$  são linguagens regulares para que operações

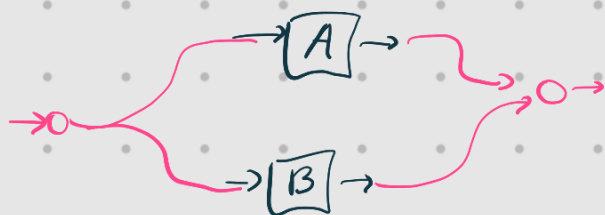
$A \otimes B$  é regular?

qualquer operação

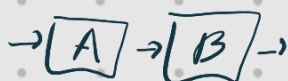
Obviamente, a classe das linguagens regulares é fechada por:

- união
- concatenação
- estrela de Kleene

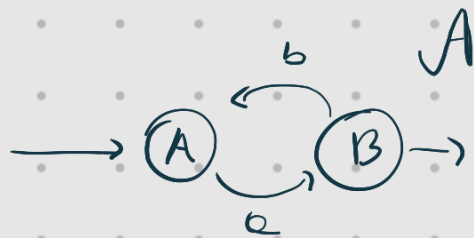
União



Concatenação



Estrela



$$L(A) = a(ba)^*$$

agora como faríamos  $L(A) = (a(ba)^*)^*$

Vamos tentar generalizar:

no caso  $L(A) = a$   $\rightarrow \text{O} \xrightarrow{a} \text{O} \rightarrow$

e tentamos colocar uma volta  $\epsilon$



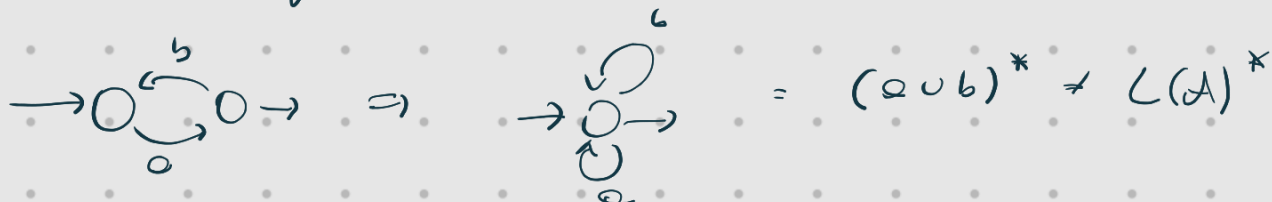
resulta em  $a^+$

se tentarmos juntar o inicial com o final, obtemos

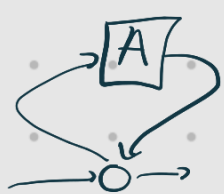


Porém no caso  $L(A) = a(ba)^*$

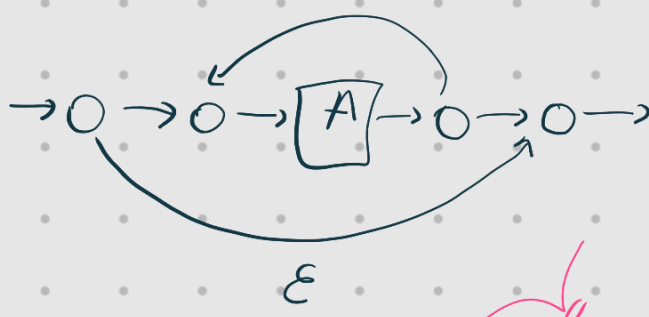
se tentarmos juntar o inicial com o final, obtemos:



→ Uma melhor solução seria criar um estado a parte:



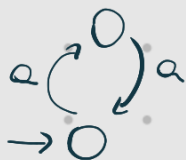
ou criar dois novos



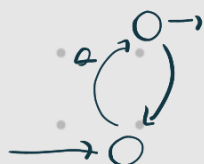
## Complemento

Caso de  $L_{PAR}$ :

$$L_{PAR} = \{a^n \mid n \text{ é par}\}$$



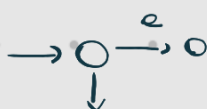
$$L_{IMPAR} = L_{PAR}^c = \{a^n \mid n \text{ não é par}\}$$



Todos que eram saída, não são mais quem não era saída agora é.



$$\Rightarrow L = a$$

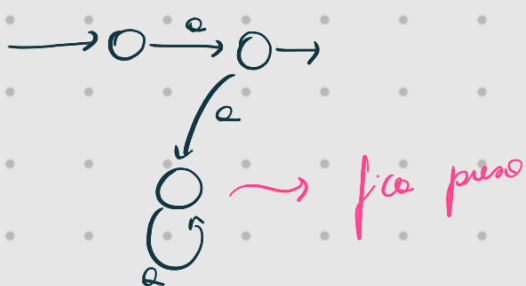


→ Nesse houve a seguir a, então  $L = \epsilon$

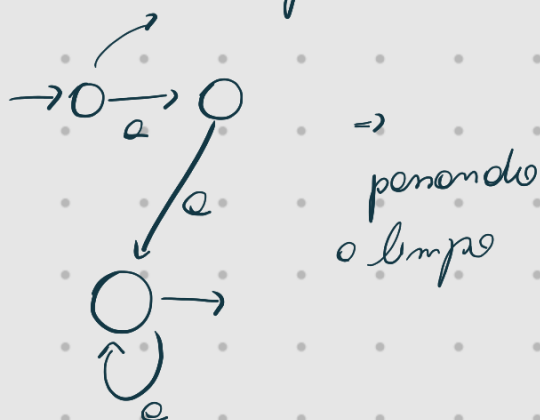
Mas então falta a, aa, etc.

Queremos  $\Sigma^* \setminus \{a\}$   $\leadsto$  queremos todos os caracteres menos  
 $a$  !, ou seja,  $a$  ou  $2$  ou  $3$  ou  $4$ , etc  
 $a$

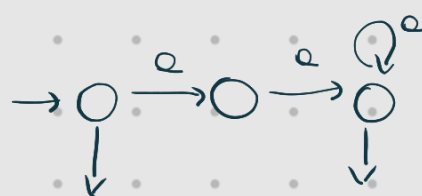
tentativo



para melhorar, faremos:



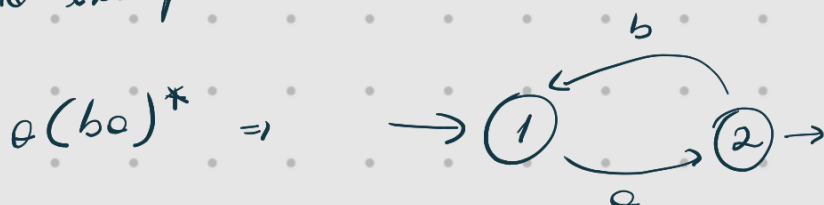
$\rightarrow$   
 pensando  
 o tempo



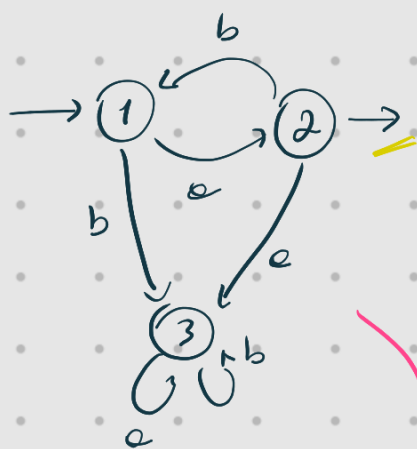
! Cuidado!  
 todos os estados  
 tem que ter todas  
 as letras do alfabeto  
 dele

~~~~~

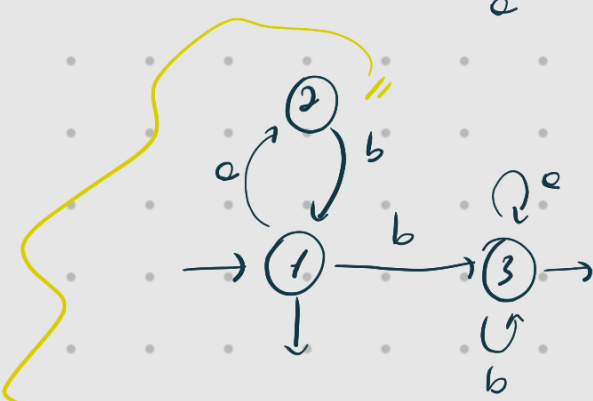
No exemplo



podemos adicionar o estado em que ficam presos:



Assim  
 encontramos  
 o complemento

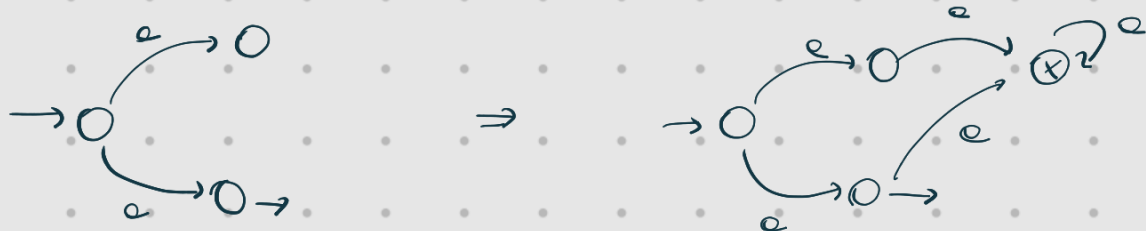


Ele aceita todos  
 do alfabeto menos  
 $a(ba)^*$

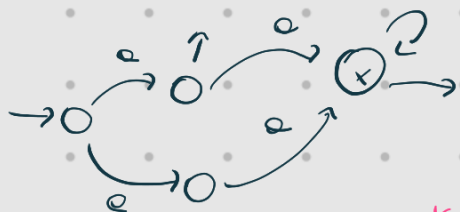
todos que tinham nada, agora não têm e todos que não  
 tinham, agora têm.

mais uma pegadinha:

o automato deve ser determinístico



⇓ complemento



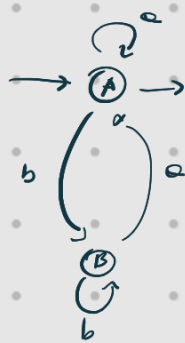
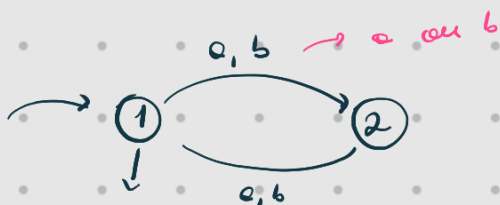
Porém continue  
exatando o que  
não poderia

Regra para encontrar o complemento

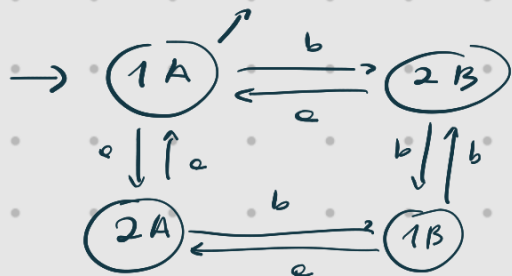
- 1 - Converte para determinístico caso não seja
- 2 - Crie o estado de "fim da bondição" se o automato não estiver completo
- 3 - Troque quem é final / não-final

## Interseção

Ex: - alfabeto  $\Sigma = \{a, b\}$   
- comprimento par  
- não termina em "b"



Vamos fazer uma combinação:



Assim sabemos qual  
pertencem à interseção