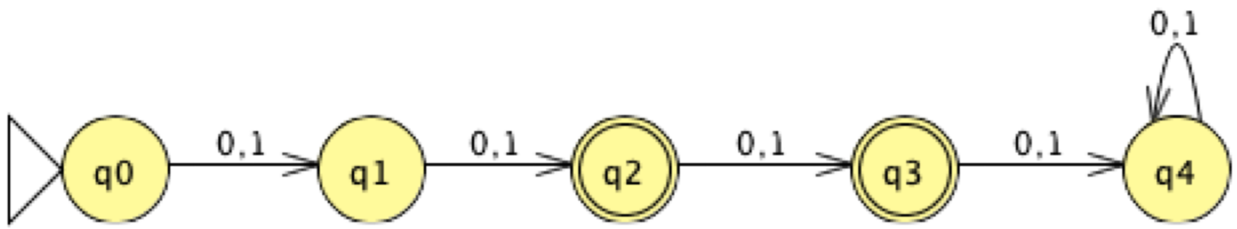
FUNDAMENTOS TEÓRICOS DA COMPUTAÇÃO

--- LINGUAGENS NÃO REGULARES ---

Princípio da Casa dos Pombos

Linguagens Finitas e Infinitas

Toda linguagem é regular se for possível escrever um AF, uma ER ou uma GR para ela (os três formalismos são intercambiáveis)



Note que a linguagem acima consiste cadeias de tamanho 2 ou 3 (finita). Certamente a linguagem acima é **regular** porque é **finita**

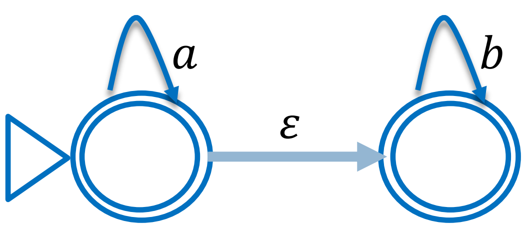
Nestes slides focaremos em linguagens **infinitas**

Limitações das LR

Para entender o poder computacional dos AF's, antes precisamos entender as suas **limitações**

Veremos que certas linguagens não podem ser reconhecidas por nenhum AF

Exemplo: $L_1 = \{a^n b^n : \ n \geq 0\}$, onde precisamos saber **contar** quantas a's foram vistos para saber se é a mesma quantidade de b's. Veja o AF abaixo:



Pode parecer que sim, mas esse AF **não** reconhece L_1 mas sim $L = \{\omega \in \{a^* b^*\}$

Como não há AF que reconheça a linguagem, L_1 não é regular

Geração de AF/ER/GR

A forma de provar que uma linguagem é regular é através da simples geração de um AFDs/AFNs/ERs/GRs

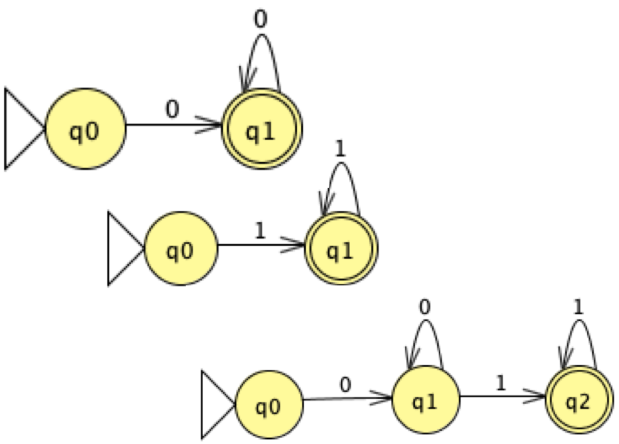
Mas e quando não conseguimos gerar?

Algumas linguagens são facilmente identificadas como **regulares**, como por exemplo:

$A = \{0^n : n \geq 1\}$

$B = \{1^n : n \geq 1\}$

$AB = \{0^n 1^m : n, m \geq 1\}$



AFs não sabem "contar"

As linguagens abaixo não são **regulares**. Em princípio, a razão é que os AF **não sabem contar** diferentes valores de n. Isto é, não dá para "contar" com uma quantidade **finita** de estados

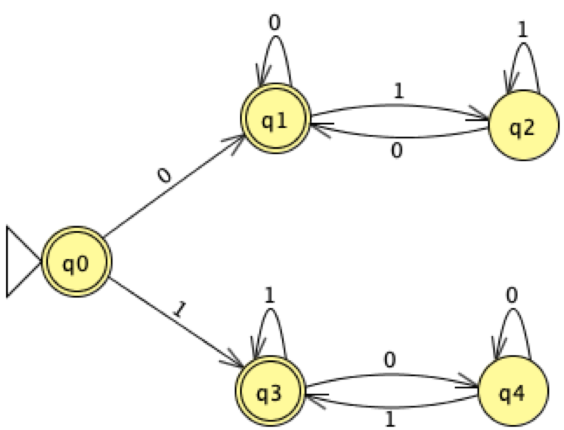
$C = \{0^n 1^n : n \geq 1\}$

$D = \{\omega \in \{0,1\}^* \;:\; |\omega|_0 = |\omega|_1\}$

Não é exatamente “contagem”

Entretanto, a linguagem E é **regular** e existe uma “contagem”.

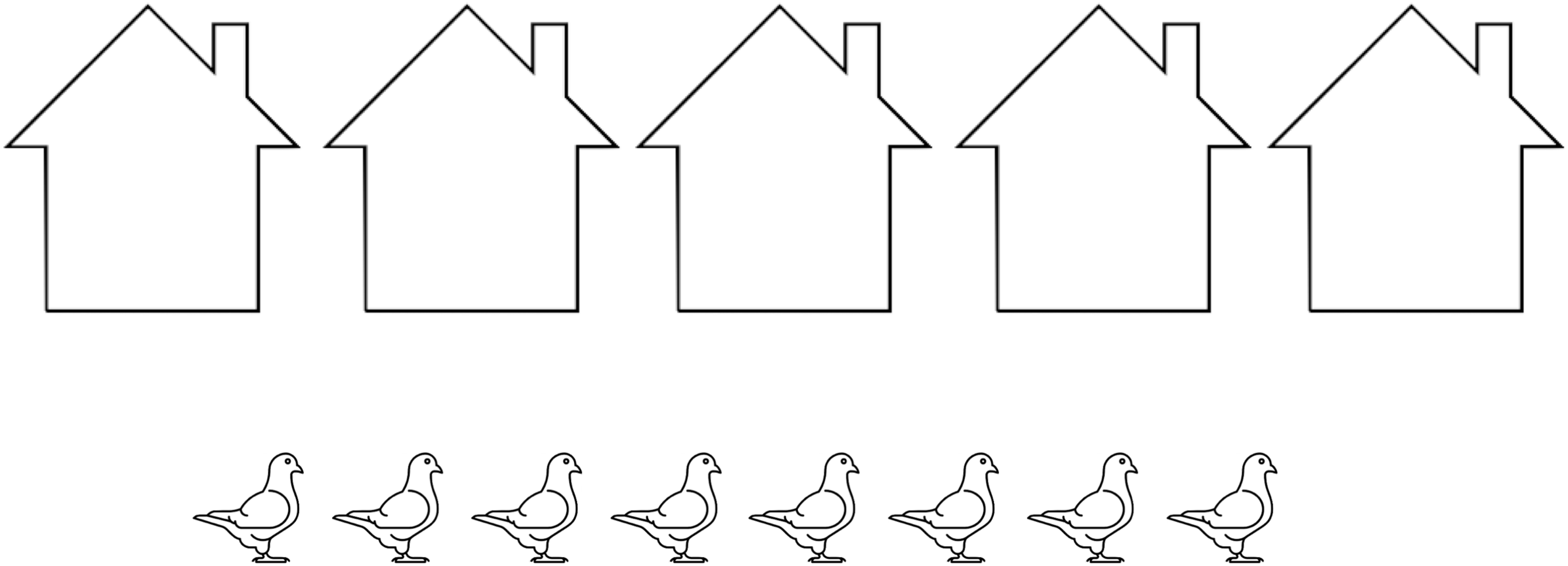
$E = \{\omega \in \{0,1\}^* : \omega$ tem número igual de 01 e 10$\}$



Só porque a linguagem parece requerer memória ilimitada não significa que ela não seja regular. Mas o argumento que o número de cadeias é **infinito** já não prova a não-regularidade? **Não, não prova!**

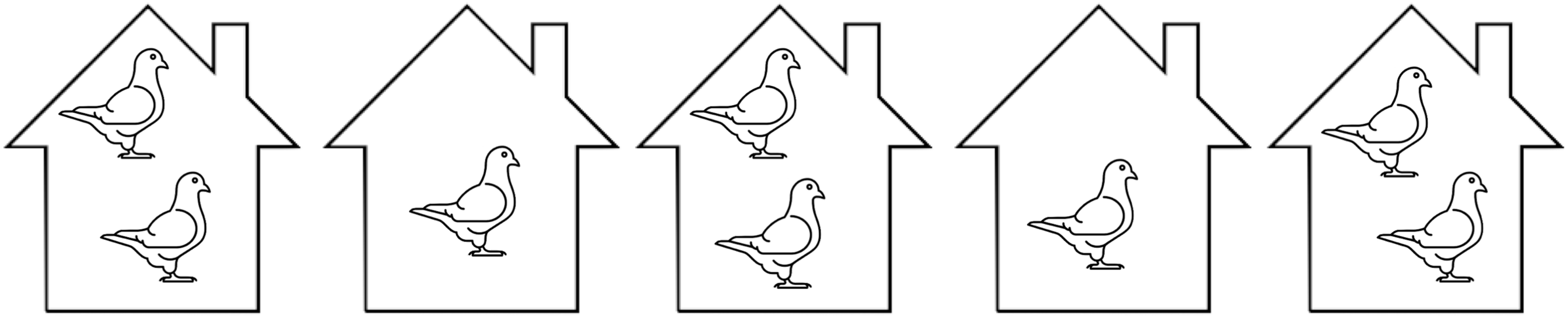
Princípio da Casa dos Pombos

Se n pombos devem ser postos em m casas e $n > m$, então **pelo menos uma casa** irá conter mais de um pombo



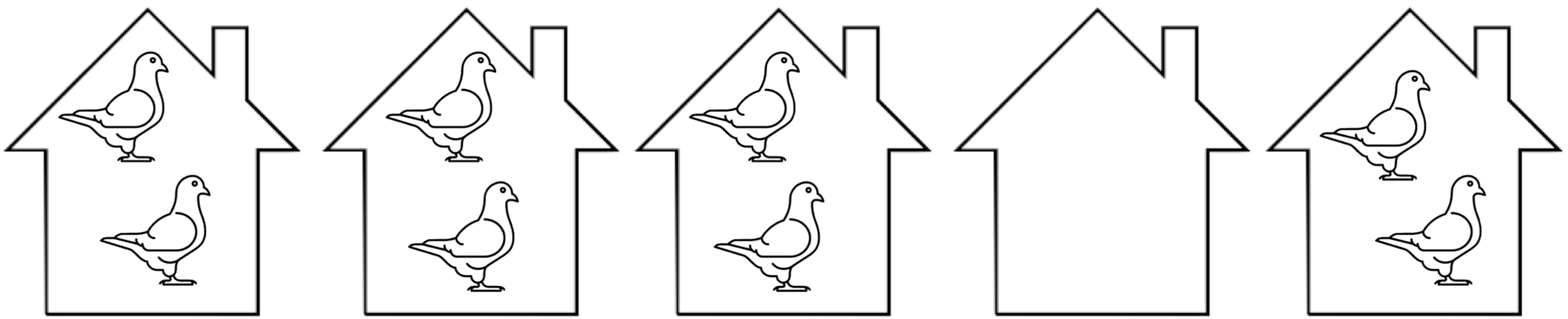
Princípio da Casa dos Pombos

Se n pombos devem ser postos em m casas e $n > m$, então **pelo menos uma casa** irá conter mais de um pombo



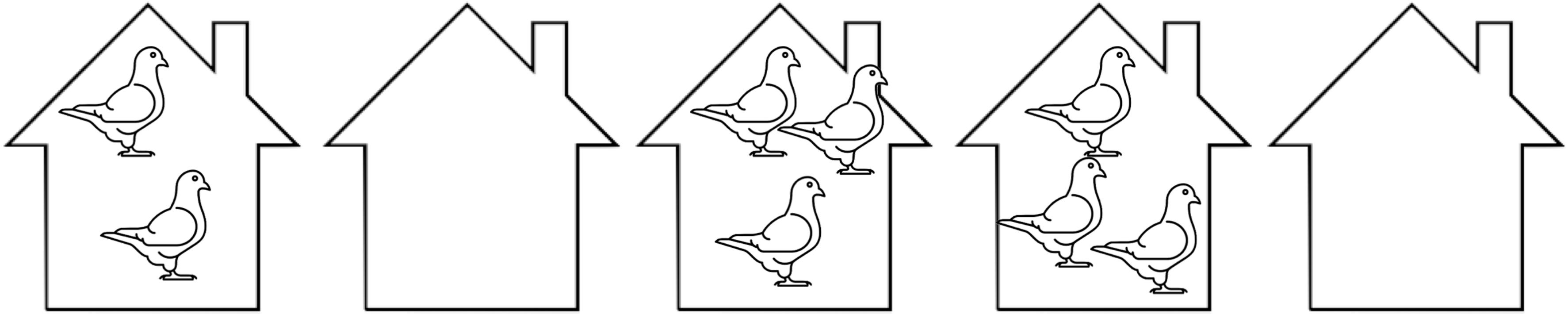
Princípio da Casa dos Pombos

Se n pombos devem ser postos em m casas e $n > m$, então **pelo menos uma casa** irá conter mais de um pombo



Princípio da Casa dos Pombos

Se n pombos devem ser postos em m casas e $n > m$, então **pelo menos uma casa** irá conter mais de um pombo



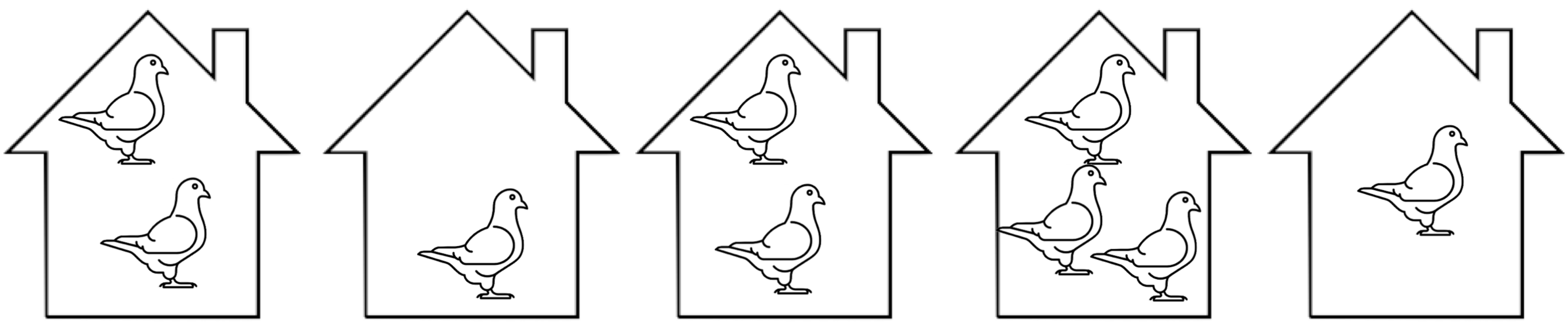
Os pombos correspondem à **cadeia de entrada** e as casinhas correspondem **aos estados** do autômato

Mas o que isso tem a ver com provar se uma linguagem é **regular**?

Princípio da Casa dos Pombos

A ideia é que se eu saí de um estado q e sigo outros estados e chego no mesmo estado q, isto indica que o trecho pode ser repetido inúmeras vezes

$$q \underbrace{\cdots\cdots\cdots}\, q \underbrace{\cdots\cdots\cdots}\, q \cdots$$



Os pombos correspondem à **cadeia de entrada** e as casinhas correspondem **aos estados** do autômato

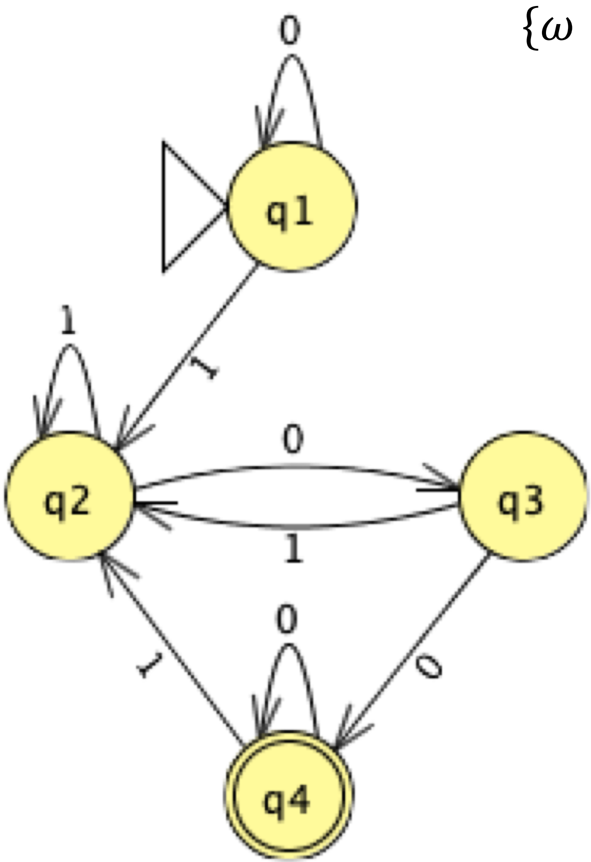
Princípio da Casa dos Pombos

É importante ressaltar que o princípio da casa de pombos em si **não prova** a não regularidade de uma linguagem

Motivação

Considere o seguinte AFD. Quantos estados? As cadeias são infinitas?

$\{\omega \in \{0,1\}^* : \omega$ ter pelo menos um 1 e termina em 00$\}$



Algumas cadeias (não todas)

$\omega_1 = 100$

$\omega_2 = 0011001000$

$\omega_3 = 0011001001001000$

$\omega_4 = 0011000$

Motivação

$\omega_1 = 100$

$\omega_1 = q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} q_4$

Este é o número mínimo de estados até o estado de aceitação

Motivação

$$\omega_1 = 100$$

$$\omega_1 = q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} q_4$$

$$\omega_2 = 0011001000$$

$$\omega_2 = q_1 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} \boldsymbol{q_4} \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} \boldsymbol{q_4} \xrightarrow{0} \boldsymbol{q_4}$$

Motivação

$\omega_1 = 100$

$\omega_1 = q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} q_4$

$\omega_2 = 0011001000$

$\omega_2 = q_1 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} q_4 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} q_4 \xrightarrow{0} q_4$

$\omega_3 = 0011001001001000$

$\omega_3 = q_1 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} \boldsymbol{q_4} \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} \boldsymbol{q_4} \xrightarrow{1} q_2$
$\xrightarrow{0} q_3 \xrightarrow{0} \boldsymbol{q_4} \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} \boldsymbol{q_4} \xrightarrow{0} \boldsymbol{q_4}$

Motivação

$\omega_1 = 100$

$\omega_1 = q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} q_4$

$\omega_2 = 0011001000$

$\omega_2 = q_1 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} q_4 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} q_4 \xrightarrow{0} q_4$

$\omega_3 = 0011001001001000$

$\omega_3 = q_1 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} q_4 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} q_4 \xrightarrow{1} q_2$
$\xrightarrow{0} q_3 \xrightarrow{0} q_4 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} q_4 \xrightarrow{0} q_4$

$\omega_4 = 0011000$

$\omega_4 = q_1 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} \boldsymbol{q_4} \xrightarrow{0} \boldsymbol{q_4}$

Motivação

É possível dividir as cadeias em $\alpha\beta\gamma$, sendo que α e γ podem ocorrer 0 ou 1 vez

$$\omega_1 = \boxed{100} = \alpha^0\beta^1\gamma^0$$

$$\omega_2 = \boxed{001100}\boxed{100}\boxed{0} = \alpha\beta^1\gamma$$

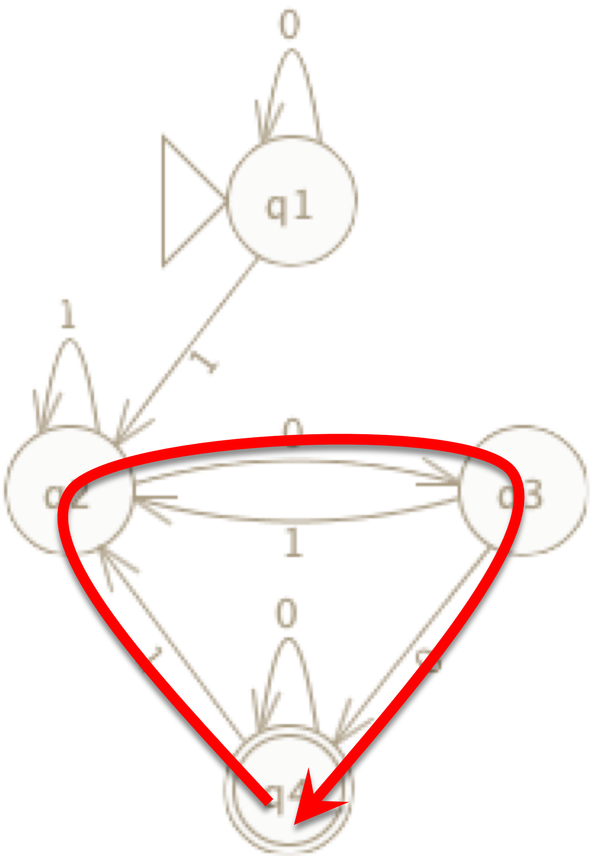
$$\omega_3 = \boxed{001100}\boxed{100}\boxed{100}\boxed{100}\boxed{0} = \alpha\beta^3\gamma$$

$$\omega_4 = \boxed{001100}\boxed{0} = \alpha\beta^0\gamma$$

E se tiver uma cadeia $\alpha\beta^{200}\gamma$ ela vai ou não pertencer a linguagem?

R: **Claro que sim!**

Motivação



Há um ciclo que se repete

$\omega_3 = 0011001001001000$

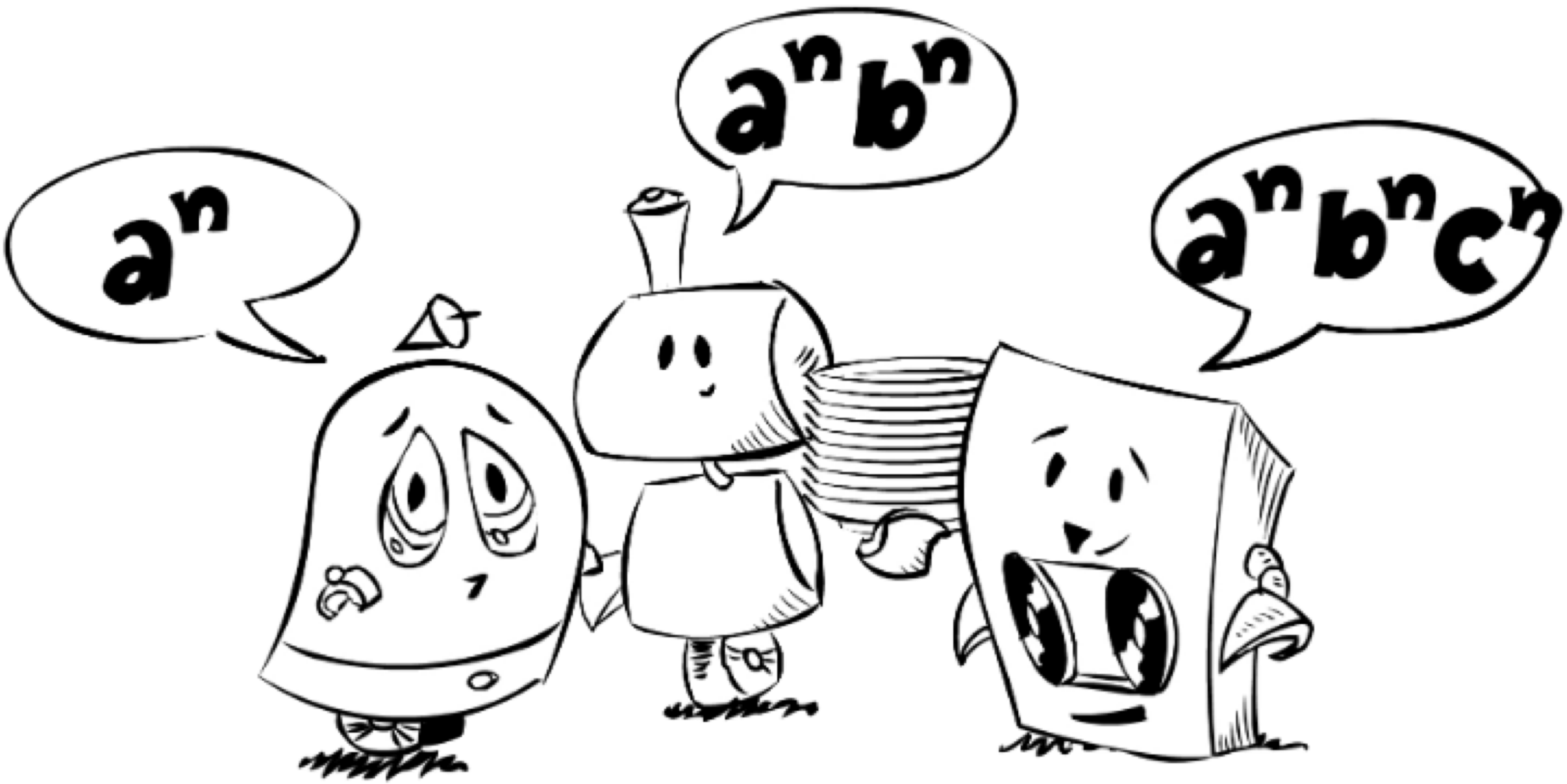
$\omega_3 = q_1 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} \boldsymbol{q_4}$
$\xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} \boldsymbol{q_4}$
$\xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} \boldsymbol{q_4}$
$\xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} \boldsymbol{q_4}$
$\xrightarrow{0} q_4$

Vai de q_4 e retorna ao q_4

Motivação

É possível representar uma linguagem com **infinitas cadeias** usando um autômato de estados finitos **sem ciclos**?

R: NÃO! Se a linguagem é infinita, **certamente tem ciclos**, e existe **uma** cadeia que é repetida ciclicamente


a^n
$a^n b^n$
$a^n b^n c^n$