FUNDAMENTOS TEÓRICOS DA COMPUTAÇÃO -- AUTÔMATO FINITO DETERMINÍSTICO --

Dicas para construção de AFDs

Exemplo 1

Gere um AFD para $L_1 = \{\omega \in \{0,1\}^* : \ |\omega|_0$ é par$\}$

Entender a linguagem: Quais as cadeias devem ser reconhecidas? Quais não? A cadeia vazia é aceita ou rejeitada?

Lembrar: Preciso ter um estado que "lembre" que estou em um estado par de zeros? Preciso lembrar quantos 1s eu já li?

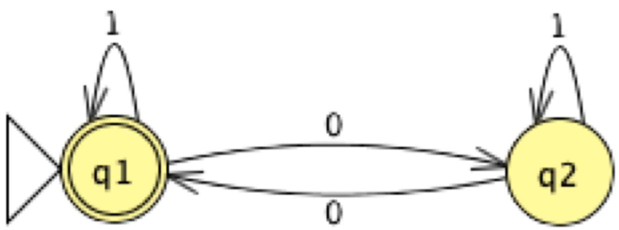
Eventos: Leitura de 0s é um evento importante? E de 1s? É importante avaliar se está em um estado com número par e estados com número impar de zeros?

Estados: Quantos estados criar?

Decisão: Se a cadeia γ já foi lida e você sabe algo sobre ela, qual decisão tomar ao ler um 0? E se for um 1? Ou seja, se a cadeia acabar agora, qual a decisão (aceita/rejeita)?

Exemplo 1

Gere um AFD para $L_1 = \{\omega \in \{0,1\}^* : \ |\omega|_0$ é par$\}$



$\hat{\delta}(q_1, \omega) = \ q_1 \ \leftrightarrow \ |\omega|_0$ é par

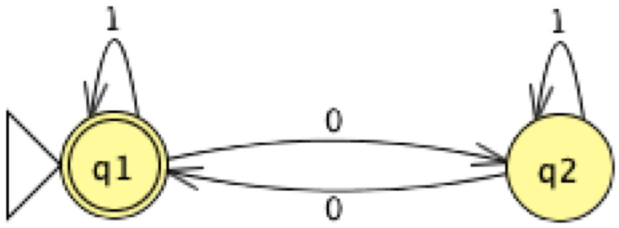
$\hat{\delta}(q_1, \omega) = \ q_2 \ \leftrightarrow \ |\omega|_0$ é ímpar

A cadeia vazia é aceita? Sim, por isso o estado inicial é também o estado de aceitação (única forma)

O estado q1 é um estado que temos um número par de 0s. O estado q2 é um estado que temos um número impar de 0s

Exemplo 1

Gere um AFD para $L_1 = \{\omega \in \{0,1\}^* : \ |\omega|_0$ é par$\}$



Em q1, a autotransição que lê 1s não muda a quantidade de de 0s

Em q2, a autotransição que lê 1s não muda a quantidade de de 0s

O que muda é só quando você lê um zero, que alterna em número par ou impar de 0s

Este autômato é finito e determinístico porque em todos os estados tem sempre uma transição para cada símbolo do alfabeto

Exemplo 2

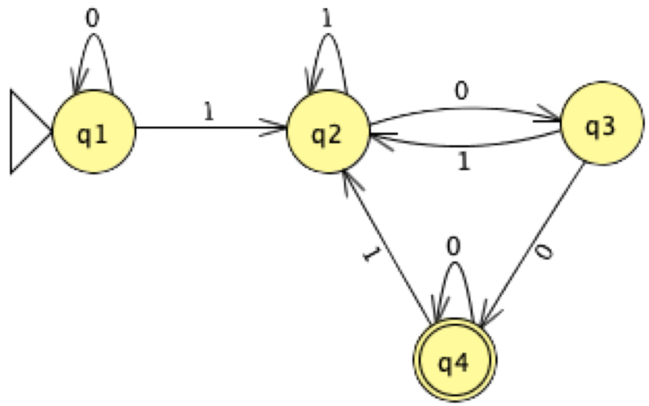
Gere um AFD para $L_2 = \{\omega \in \{0,1\}^* : \ \omega$ termina em 00 e contém pelo menos um $1\}$

Entender a linguagem: Quais as cadeias devem ser reconhecidas? A cadeia vazia não faz parte da linguagem

Ex: $L_2 = \{000100, 111100, 100, \dots\}$

Eventos importantes:
* Já ter lido um 1 em algum momento
* Ler 0 (começo de um padrão importante)
* Ler 0 depois de ter lido um 0 (pode ser estado de aceitação)

Exemplo 2



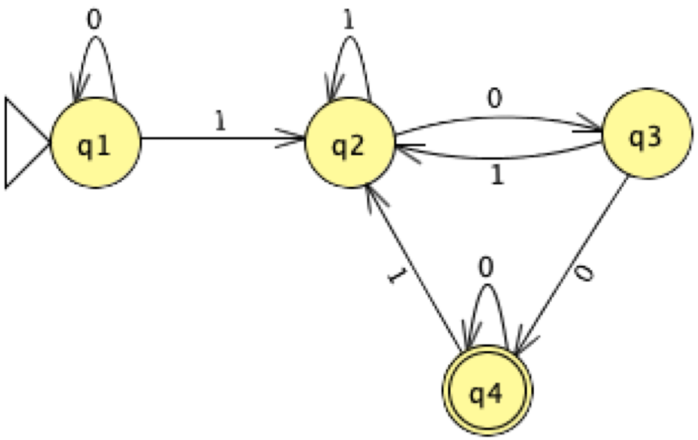
No estado q1 não importa a quantidade de 0s lida, porque ainda não foi lido nenhum um 1

O estado q2 é importante porque já foi lido um 1 e, nesse estado, já não importa muito a quantidade de 1s

O estado q3 é importante porque pode ser o início do padrão de terminar com dois 0s

O estado q4 é importante porque é um estado de aceitação, mas se leu um 1 volta tudo de novo para o estado q2

Exemplo 2



$\hat{\delta}(q_1, \omega) = \ q_1 \ \leftrightarrow \ \ \omega = 0^k$ com $k \geq 1$

$\hat{\delta}(q_1, \omega) = \ q_2 \ \leftrightarrow \ \ \omega = \alpha 1$ com $\alpha \in \Sigma^*$

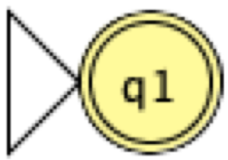
$\hat{\delta}(q_1, \omega) = \ q_3 \ \leftrightarrow \ \ \omega = \alpha 1 \beta 0$ com $\alpha, \beta \in \Sigma^*$

$\hat{\delta}(q_1, \omega) = \ q_4 \ \leftrightarrow \ \ \omega = \alpha 1 \beta 00$ com $\alpha, \beta \in \Sigma^*$

Exemplo 3

Gere um AFD para $L_3 = \{\omega \in \{0,1\}^* : \ \omega = \varepsilon\}$

Entender a linguagem: O estado inicial é o único estado que podemos não ter lido nenhum símbolo. A única forma de aceitar a cadeia vazia é fazendo com que o estado inicial seja também o estado de aceitação

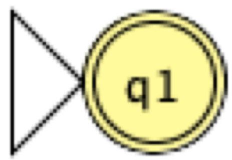


Este autômato está correto?

Exemplo 3

Gere um AFD para $L_3 = \{\omega \in \{0,1\}^* : \ \omega = \varepsilon\}$

Entender a linguagem: O estado inicial é o único estado que podemos não ter lido nenhum símbolo. A única forma de aceitar a cadeia vazia é fazendo com que o estado inicial seja também o estado de aceitação

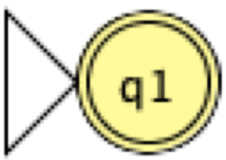


Não. Para ser um autômato finito determinístico tem que ter transições para outros estados consumindo os símbolos do alfabeto

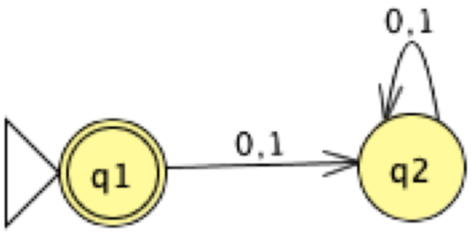
Exemplo 3

Gere um AFD para $L_3 = \{\omega \in \{0,1\}^* : \ \omega = \varepsilon\}$

Entender a linguagem: O estado inicial é o único estado que podemos não ter lido nenhum símbolo. A única forma de aceitar a cadeia vazia é fazendo com que o estado inicial seja também o estado de aceitação



Para ser um autômato finito determinístico tem que ter transições para outros estados consumindo os símbolos do alfabeto



Exemplo 4

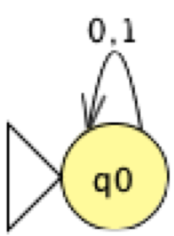
Gere um AFD para $\Sigma = \{0,1\}\ e\ L_4 = \emptyset$

Entender a linguagem: A única forma de reconhecer “nada” é se não houver estado de aceitação. Formalmente isso é possível? Se sim, é possível gerar um AFD?

Exemplo 4

Gere um AFD para $\Sigma = \{0,1\}\ e\ L_4 = \emptyset$

Entender a linguagem: A única forma de reconhecer "nada" é se não houver estado de aceitação. Formalmente isso é possível? Se sim, é possível gerar um AFD?



Exemplo 5

Gere um AFD para $L_5 = \{\omega \in \{a, b\}^* : \ |\omega| \geq 3$ e o terceiro símbolo de ω é a$\}$

Entender a linguagem: Quais as cadeias devem ser reconhecidas?

Certamente que a cadeia vazia não faz parte da linguagem

Ex: $L_5 = \{aaa, aaaa, abab, bbab, bbabb, \ldots\}$

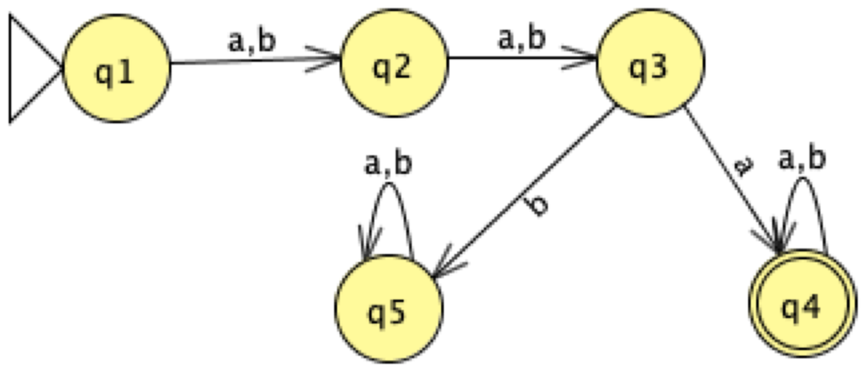
Em outras palavras, $\alpha a \beta$, com $|\alpha| = 2$

Eventos importantes:

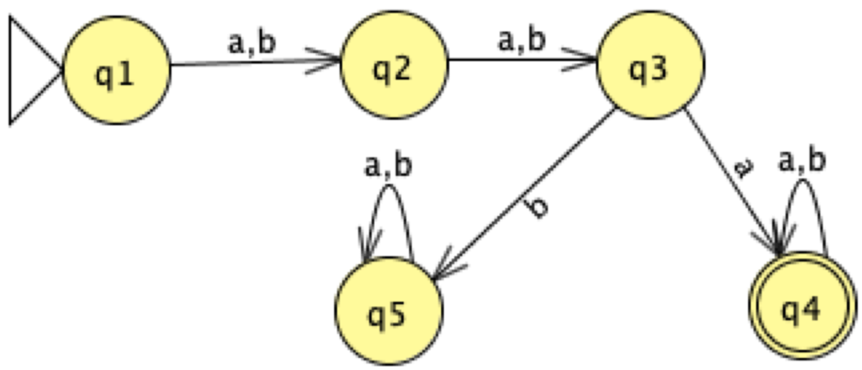
* Saber quantos símbolos já foram lidos
* Saber se o terceiro símbolo é 'a'
* Depois da análise do terceiro símbolo, os outros símbolos não importam

Exemplo 5

Gere um AFD para $L_5 = \{\omega \in \{a, b\}^* : \ |\omega| \geq 3$ e o terceiro símbolo de ω é a$\}$



Exemplo 5



$$\hat{\delta}(q_1, \omega) = q_1 \leftrightarrow \omega = \varepsilon$$

$$\hat{\delta}(q_1, \omega) = q_2 \leftrightarrow \omega \in \Sigma^1$$

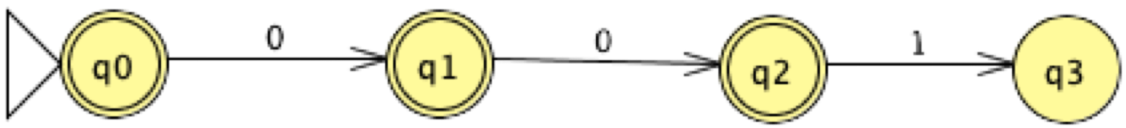
$$\hat{\delta}(q_1, \omega) = q_3 \leftrightarrow \omega \in \Sigma^2$$

$$\hat{\delta}(q_1, \omega) = q_4 \leftrightarrow \omega = \alpha a \beta, \alpha \in \Sigma^2 \; e \; \beta \in \Sigma^*$$

$$\hat{\delta}(q_1, \omega) = q_5 \leftrightarrow \omega = \alpha b \beta, \alpha \in \Sigma^2 \; e \; \beta \in \Sigma^*$$

Exemplo 6

Gere um AFD para $L_6 = \{\omega \in \{0,1\}^* : \ \omega \text{ não contém } 001\}$

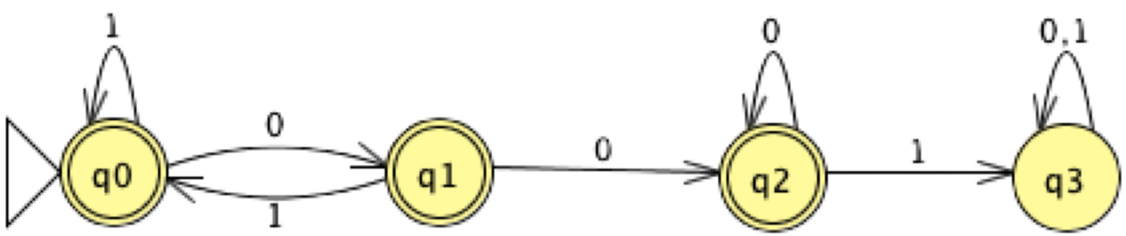


Fica claro que a sequência **001** não deve ser estado de aceitação

Para ser um **autômato finito determinístico** faz-se necessário ter transições **para todos** os símbolos do alfabeto

Exemplo 6

Gere um AFD para $L_6 = \{\omega \in \{0,1\}^* : \ \omega$ não contém 001$\}$



$\hat{\delta}(q_0, \omega) = \ q_3 \ \leftrightarrow \ \ \omega = \alpha 001 \beta$, com $\alpha, \beta \in \{0,1\}^*$

q_3 é um estado de rejeição

Exemplo 7

Gere um AFD para $L_7 = \{\omega \in \{0,1\}^* :$ o terceiro símbolo a partir do fim em ω é 1$\}$

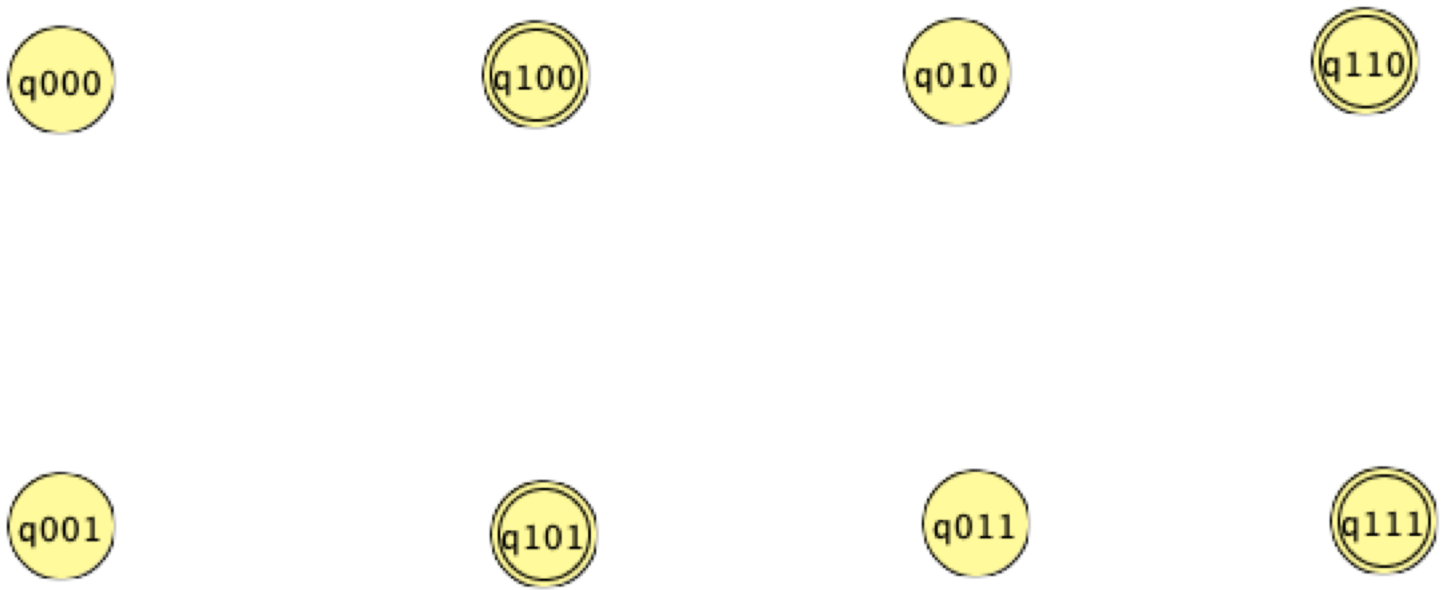
Reescrevendo: $L_7 = \{\alpha 1 \beta \; : \; \alpha \in \{0,1\}^*, \; \beta \in \{0,1\}^2\}$

Estados de aceitação: $\{\alpha 100, \alpha 101, \alpha 110, \alpha 111\}$

Estados de rejeição: $\{\alpha 000, \alpha 001, \alpha 010, \alpha 011\}$

A ideia é "lembrar" dos três últimos símbolos lidos

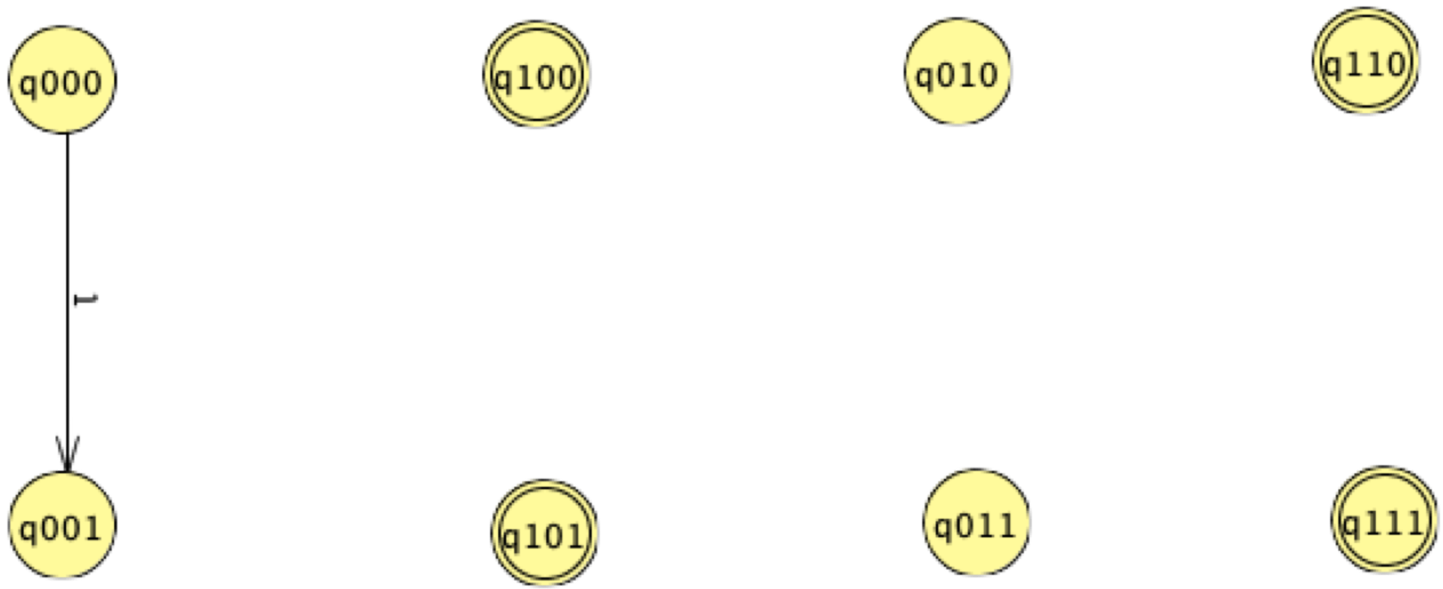
Exemplo 7



Criamos um estado para cada possibilidade dos 3 últimos símbolos

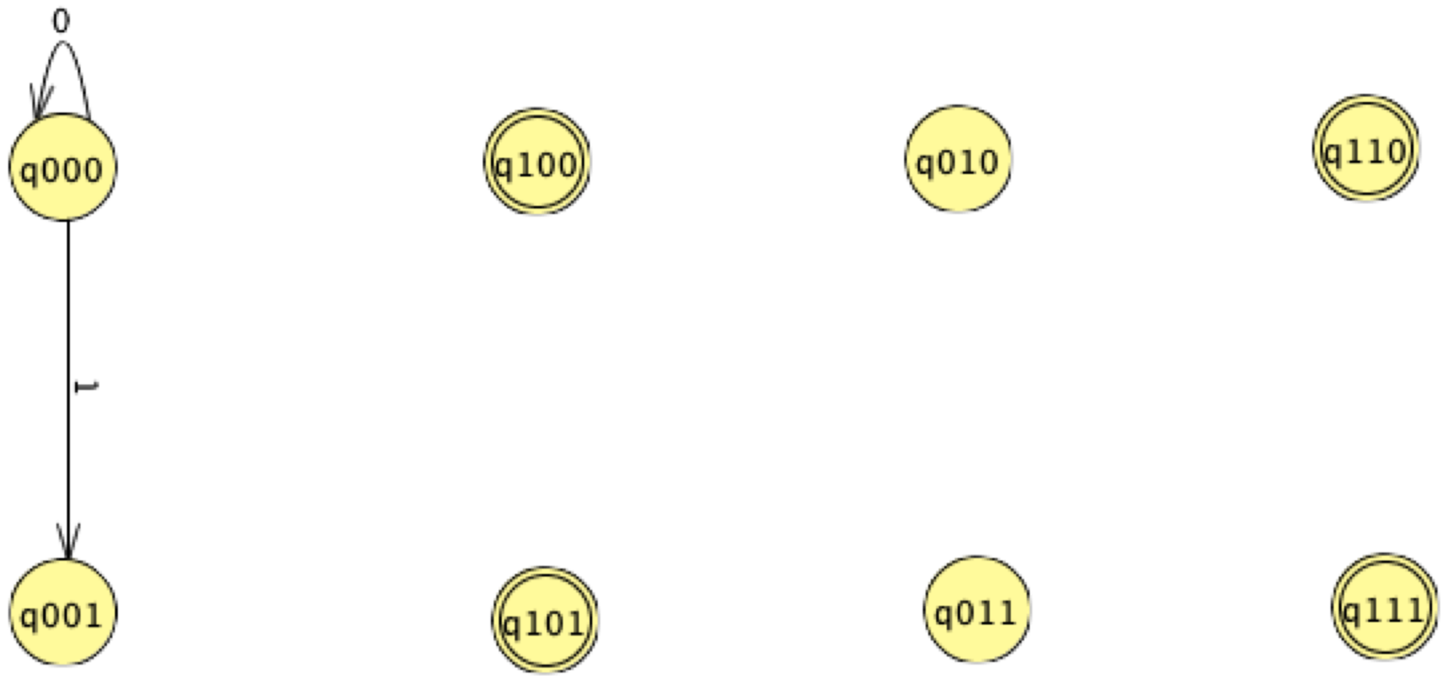
No caso, se estou no estado q_{101}, significa que os três últimos símbolos foram 101

Exemplo 7



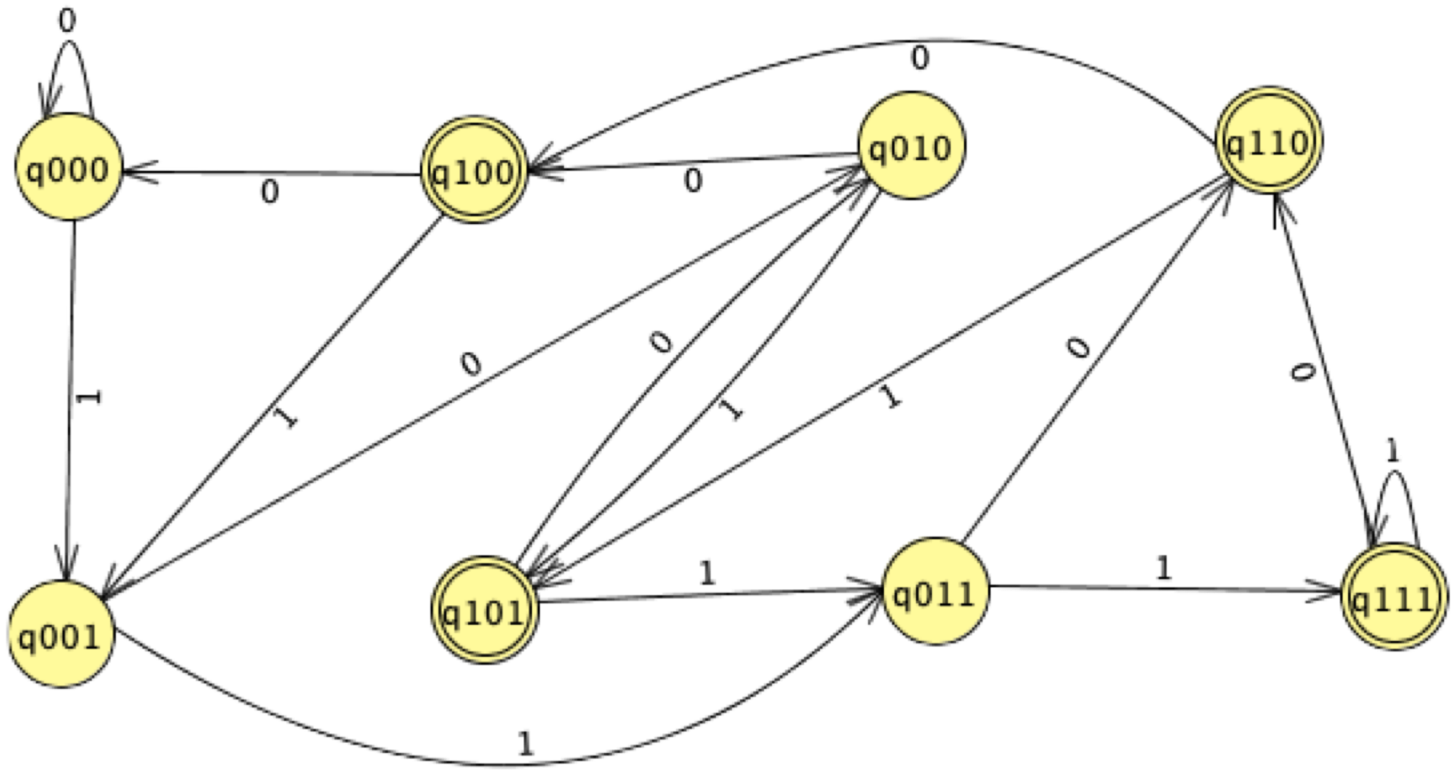
Se estou no estado q_{000} e o próximo símbolo foi o **1**, eu transito para o estado q_{001}, porque se tinha lido $\alpha 000$ e agora li $\alpha 000\textcolor{red}{1}$, portanto, o estado dos últimos três símbolos é $\textcolor{red}{q_{001}}$

Exemplo 7



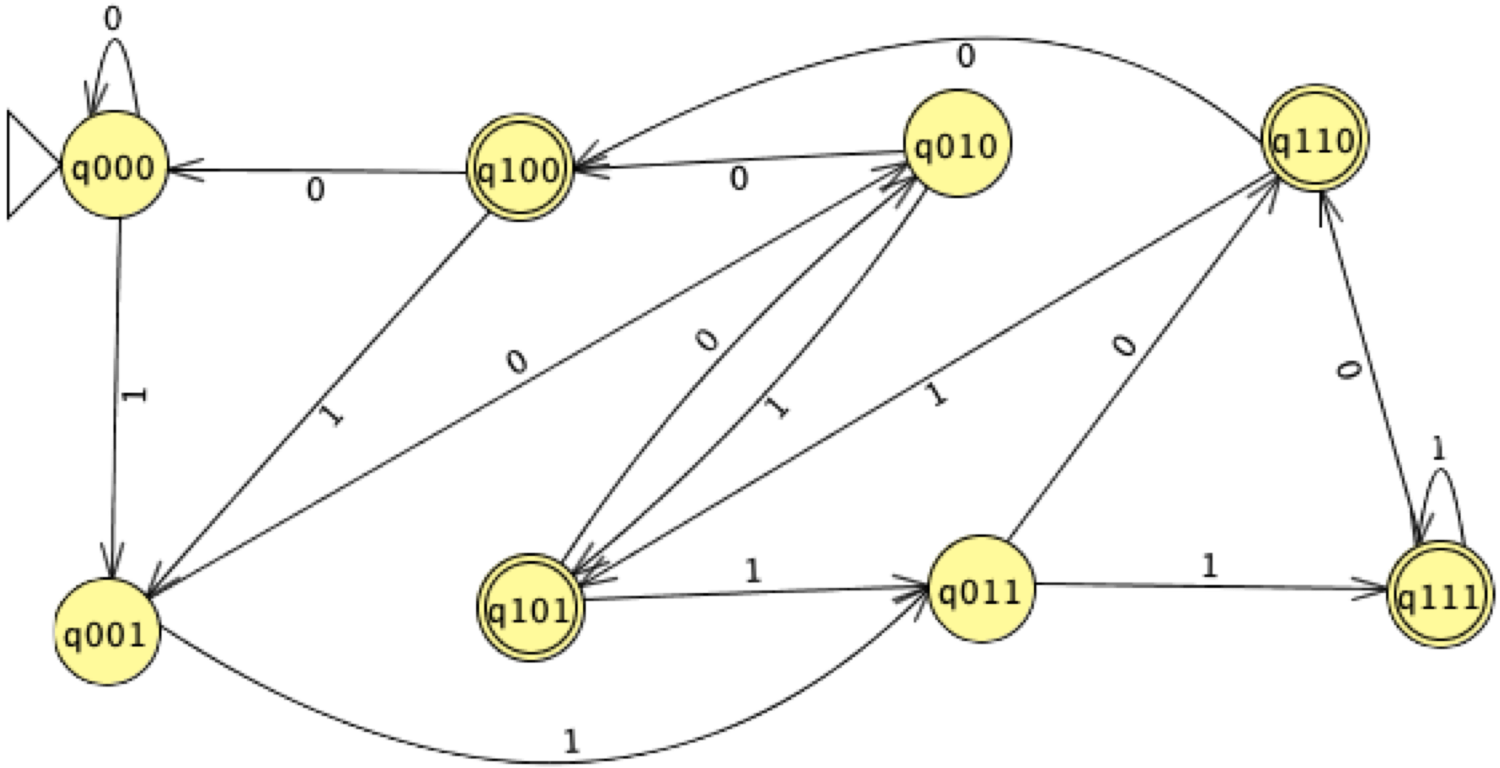
Se estou no estado q_{000} e o próximo símbolo foi o **0**, eu permaneço no mesmo estado q_{000}, porque se tinha lido $\alpha 000$ e agora li $\alpha 000\color{red}{0}$, o estado dos últimos três símbolos não muda e permanece $\color{red}{q_{000}}$

Exemplo 7



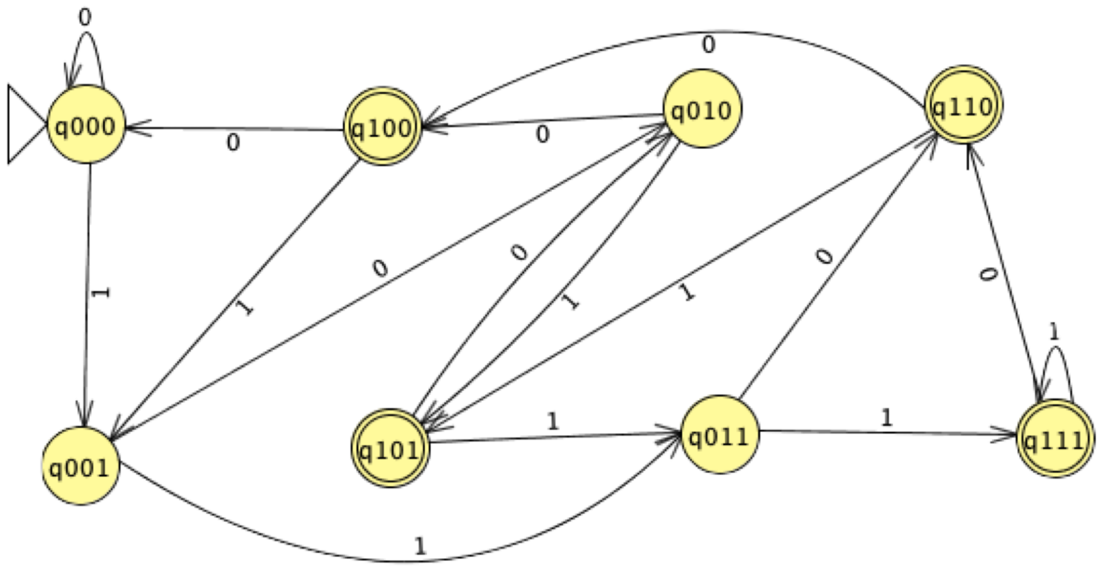
Este autômato só considera os três últimos símbolos. Falta identificar qual seria o estado inicial…

Exemplo 7



Como só importa os três últimos símbolos, qualquer um pode ser o estado inicial

Exemplo 7



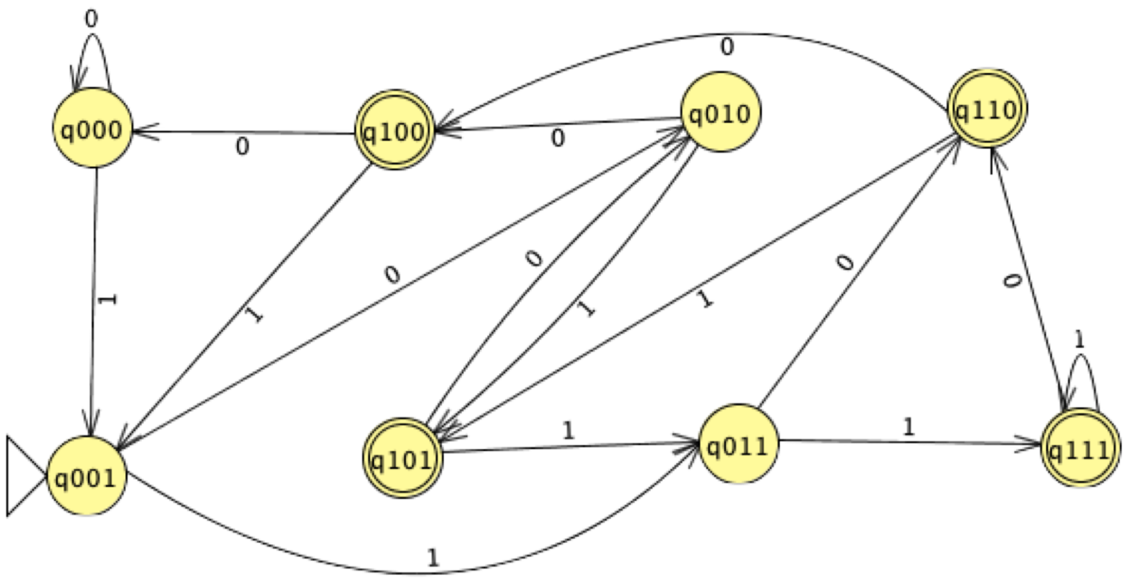
$$\hat{\delta}(q_{000}, \omega) = q_{100} \leftrightarrow \omega = 100$$

$$\hat{\delta}(q_{000}, \omega) = q_{101} \leftrightarrow \omega = 101$$

$$\hat{\delta}(q_{000}, \omega) = q_{110} \leftrightarrow \omega = 110$$

$$\hat{\delta}(q_{000}, \omega) = q_{111} \leftrightarrow \omega = 111$$

Exemplo 7



$$\hat{\delta}(q_{001}, \omega) = q_{100} \leftrightarrow \omega = 100$$

$$\hat{\delta}(q_{001}, \omega) = q_{101} \leftrightarrow \omega = 101$$

$$\hat{\delta}(q_{001}, \omega) = q_{110} \leftrightarrow \omega = 110$$

$$\hat{\delta}(q_{001}, \omega) = q_{111} \leftrightarrow \omega = 111$$


Como está o projeto que você precisa entregar amanhã?
A espera de um milagre!