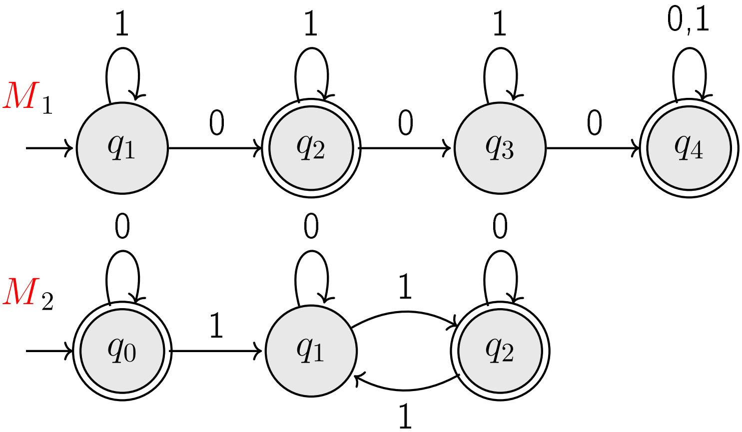
Linguagens regulares

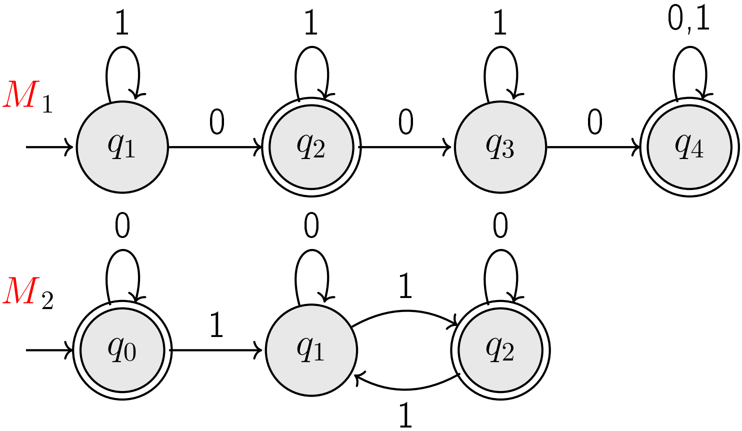
Uma linguagem é **regular** se existe algum autômato finito determinístico que a reconhece.

O nosso objetivo será investigar as linguagens regulares.

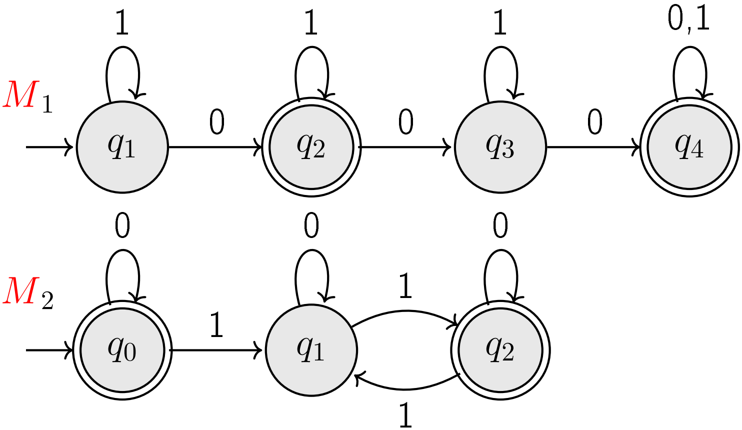


M_1 **não** aceita ϵ: estado inicial **não é** de aceitação

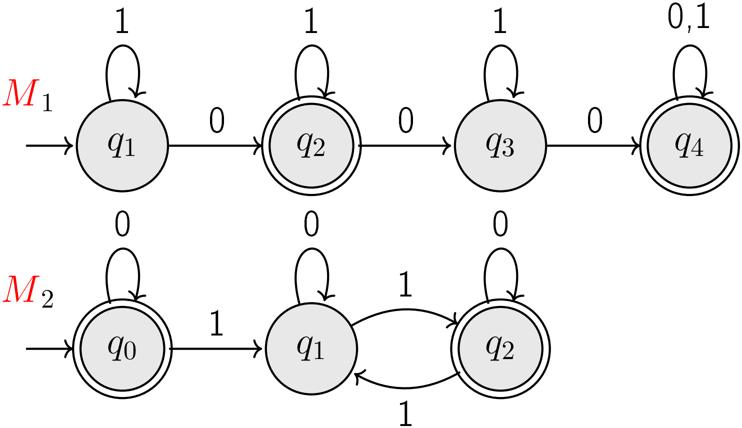
M_2 aceita ϵ: estado inicial **é** de aceitação



Strings com 6 símbolos **aceitas** por M_1 e **rejeitadas** por M_2: 000001, 000010, 000100, 001000, 000111, 011111, 101111, . . .

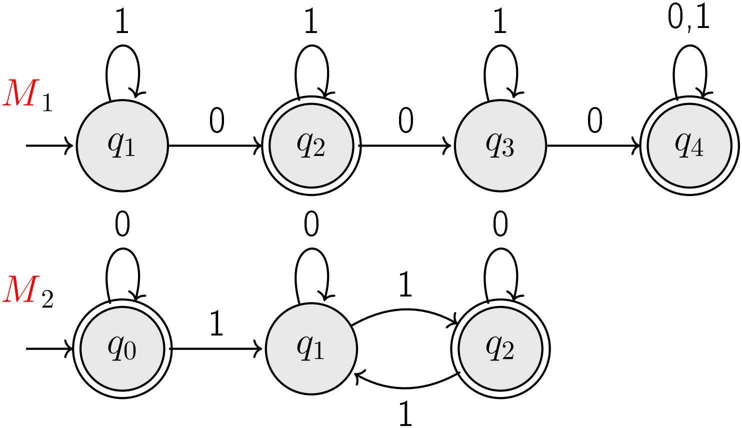


Strings com 6 símbolos **rejeitas** por M_1 e **aceitas** por M_2: 001111, 010111, 101101, 110110, 111111, ...

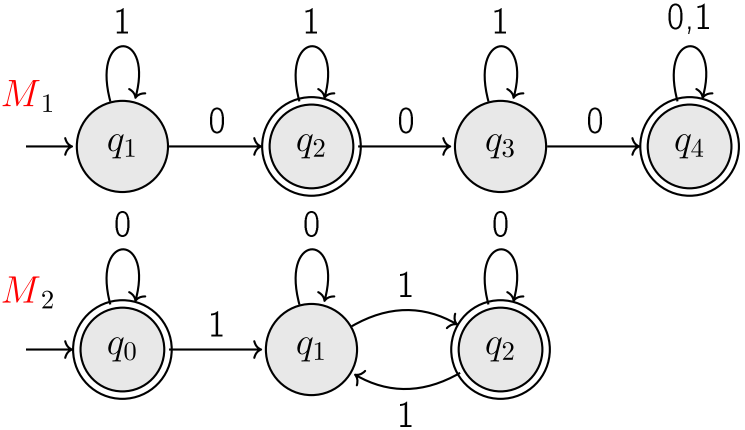


Strings com 6 símbolos **aceitas** por M_1 e por M_2:

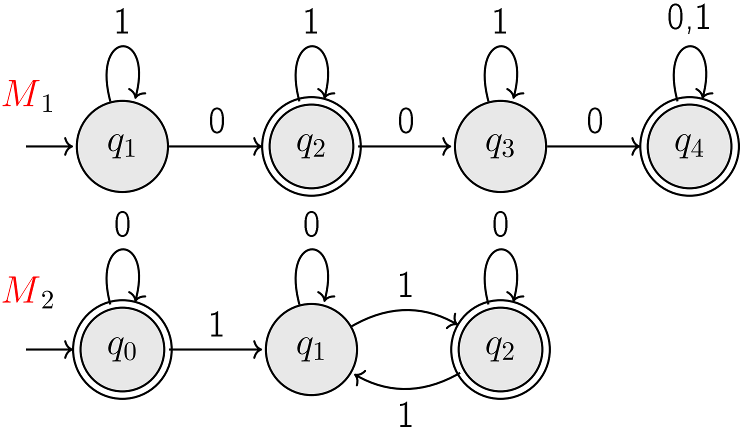
000000, 110000, 101000, 100100, 100010, . . .



Strings com 5 símbolos **rejeitas** por M_1 e por M_2:
00111, 01110, 10101, 11010, 11100, . . .



$L(M_1) = \{w : w \text{ tem um ou pelo menos três 0's}\}$



$L(M_2) = \{w : w \text{ tem um número par de 1's}\}$

União de linguagens regulares

Teorema. Se A_1 e A_2 são linguagem regulares, então $A_1 \cup A_2$ é regular.

União: $X \cup Y = \{w : w \in X \text{ ou } w \in Y\}$

$$
\begin{aligned}
\Sigma &= \{0, 1, \ldots, 9, a, b, c, \ldots, z\} \\
A_1 &= \{mac, mae, map, mat\} \\
A_2 &= \{0121, 0323, 0338\} \\
A_1 \cup A_2 &= \{mac, mae, map, mat, 0121, 0323, 0338\}
\end{aligned}
$$

União de linguagens regulares

Teorema. Se A_1 e A_2 são linguagem regulares, então $A_1 \cup A_2$ é regular.

Usar

$$M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1) \text{ e } A_1 = L(M_1)$$
$$M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2) \text{ e } A_2 = L(M_2)$$

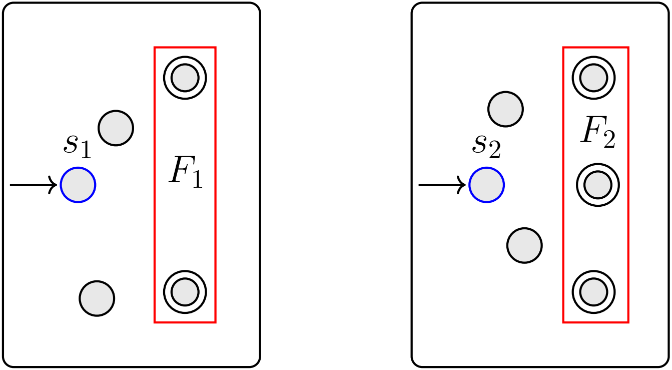
para construir

$$M = (Q, \Sigma, \delta, s, F)$$

tal que $A_1 \cup A_2 = L(M)$.

União de linguagens regulares

Teorema. Se A_1 e A_2 são linguagem regulares, então $A_1 \cup A_2$ é regular.



M_1 M_2

União de linguagens regulares

Teorema. Se A_1 e A_2 são linguagem regulares, então $A_1 \cup A_2$ é regular.

Ideia:Dada um string w, primeiro execute M_1 com w como entrada. Depois execute M_2 com w como entrada. Aceite w se M_1 aceitou w ou de M_2 aceitou w.

Opsss! A regra é clara!

Depois de ler e simular M_1 com um símbolo de w esse símbolo se foi para sempre, ..., não pode ser lido novamente.

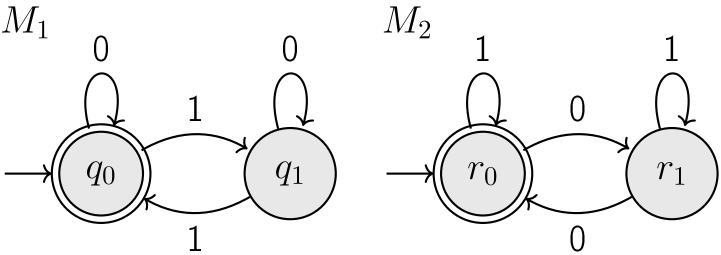
União de linguagens regulares

Teorema. Se A_1 e A_2 são linguagem regulares, então $A_1 \cup A_2$ é regular.

Mais uma Ideia:

Apelar para paralelismo!
Como?

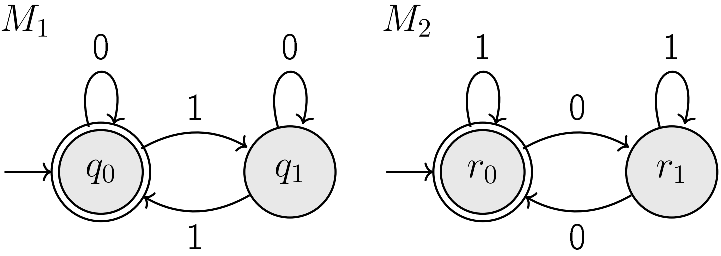
Exemplo $A_1 \cup A_2$?



$A_1 = L(M_1) = \{w : w \text{ tem um número par de } 1\text{'s}\}$

$A_2 = L(M_2) = \{w : w \text{ tem um número par de } 0\text{'s}\}$

Exemplo $A_1 \cup A_2$?



$w =$	1	0	0	1	1	0	1	0	1	0	1	...

w está em $A_1 \cup A_2$?

Rabiscos . . . 

w =

1	0	0	1	1	0	1	0	1	0	1	. . .

Ideia:  anda em M_1 enquanto  cuida de M_2.

	ϵ	1	0	0	1	1	0	1	0	1	. . .
	q_0	q_1	q_1	q_1	q_0	q_1	q_1	q_0	q_0	q_1	. . .
	r_0	r_0	r_1	r_0	r_0	r_0	r_1	r_1	r_0	r_0	. . .

Ao final,  e  se juntam, compartilham seus resultados e decidem **aceitar** w se **uma das duas aceitar**.

União de linguagens regulares

Teorema. Se A_1 e A_2 são linguagem regulares, então $A_1 \cup A_2$ é regular.

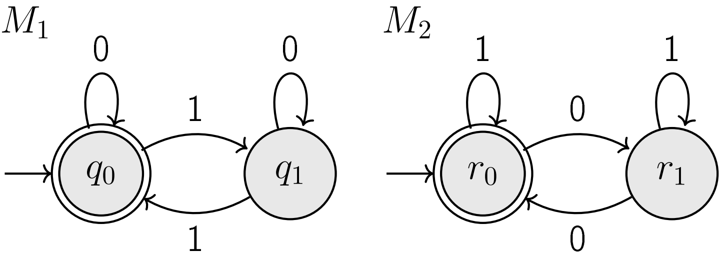
Rascunho de Demonstração:Considere o autômato $M = (Q, \Sigma, \delta, s, F)$ em que

$$
\begin{aligned}
Q &= Q_1 \times Q_2 = \{(q_1, q_2) : q_1 \in Q_1 \text{ e } q_2 \in Q_2\} \\
s &= (s_1, s_2) \\
F &= (F_1 \times Q_2) \cup (Q_1 \times F_2) \\
&= \{(q_1, q_2) : q_1 \in F_1 \textbf{ ou } q_2 \in F_2\}
\end{aligned}
$$

Finalmente, para $(q_1, q_2) \in Q$ e $a \in \Sigma$

$$\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a)).$$

Exemplo $A_1 \cup A_2$

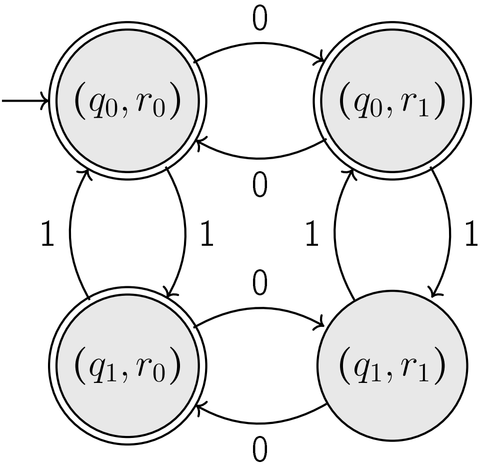


$A_1 = L(M_1) = \{w : w \text{ tem um número par de } 1\text{'s}\}$

$A_2 = L(M_2) = \{w : w \text{ tem um número par de } 0\text{'s}\}$

Exemplo $A_1 \cup A_2$

M



$A = L(M) = A_1 \cup A_2$

Exemplo $A_1 \cup A_2$

Se o autômato está no estado:

(q_0, r_0): até agora, número de 0's e 1's é par

(q_0, r_1): até agora, número de 0's é par e 1's é ímpar

(q_1, r_0): até agora, número de 0's é ímpar e 1's é par

(q_1, r_1): até agora, número de 0's e 1's é ímpar

> Se M_1 tem k_1 estados e M_2 tem k_2 estados, então M tem $k_1 \times k_2$ estados.

Interseção de linguagens regulares

Teorema. Se A_1 e A_2 são linguagem regulares, então $A_1 \cap A_2$ é regular.

Usar

$$M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1) \text{ e } A_1 = L(M_1)$$
$$M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2) \text{ e } A_2 = L(M_1)$$

para construir

$$M = (Q, \Sigma, \delta, s, F)$$

tal que $A_1 \cap A_2 = L(M)$.

Interseção de linguagens regulares

Teorema. Se A_1 e A_2 são linguagem regulares, então $A_1 \cap A_2$ é regular.

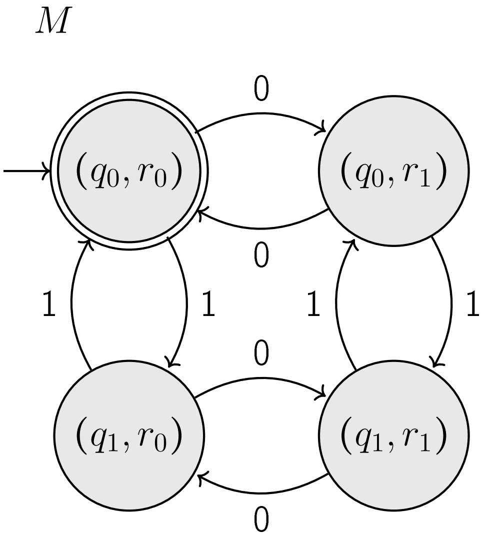
Rascunho de Demonstração:Considere o autômato $M = (Q, \Sigma, \delta, s, F)$ em que

$$
\begin{aligned}
Q &= Q_1 \times Q_2 = \{(q_1, q_2) : q_1 \in Q_1 \text{ e } q_2 \in Q_2\} \\
s &= (s_1, s_2) \\
F &= (F_1 \cap F_2) \\
&= \{(q_1, q_2) : q_1 \in F_1 \textbf{ e } q_2 \in F_2\}
\end{aligned}
$$

Finalmente, para $(q_1, q_2) \in Q$ e $a \in \Sigma$

$$\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a)).$$

$$L(M) = A_1 \cap A_2$$



Concatenação de linguagens regulares

Teorema. Se A_1 e A_2 são linguagem regulares, então $A_1 \circ A_2$ é regular.

Concatenação:

$$XY = \{w : \text{existem } x \in X \text{ e } y \in Y \text{ tq } w = xy\}$$

$$\begin{aligned}
\Sigma &= \{0, 1, \ldots, 9, a, b, c, \ldots, z\} \\
A_1 &= \{mac, mae\} \\
A_2 &= \{0121, 0323, 0338\} \\
A_1A_2 &= \{mac0121, mac0323, mac0338, mae0121, \\
&\qquad mae0323, mae0338\}
\end{aligned}$$

Concatenação de linguagens regulares

Teorema. Se A_1 e A_2 são linguagem regulares, então $A_1 \circ A_2$ é regular.

Usar

$$M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1) \text{ e } A_1 = L(M_1)$$
$$M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2) \text{ e } A_2 = L(M_1)$$

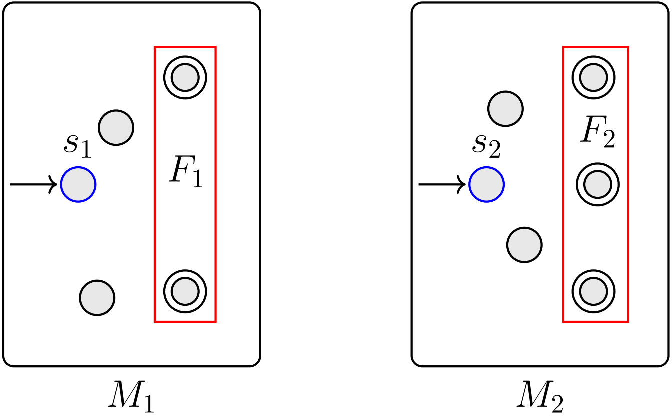
para construir

$$M = (Q, \Sigma, \delta, s, F)$$

tal que $A_1 \circ A_2 = L(M)$.

Concatenação de linguagens regulares

Teorema. Se A_1 e A_2 são linguagem regulares, então $A_1 \circ A_2$ é regular.



Concatenação de linguagens regulares

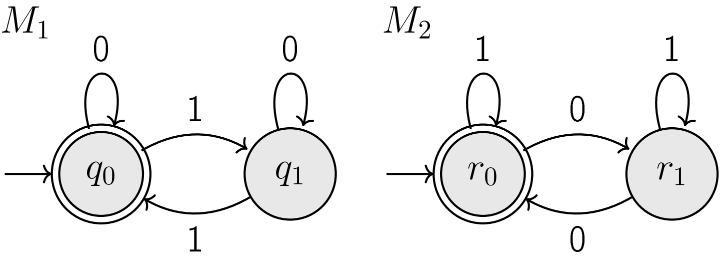
Teorema. Se A_1 e A_2 são linguagem regulares, então $A_1 \circ A_2$ é regular.

M deve aceitar uma string w se existem strings x e y tais que

- $w = xy$;
- M_1 aceita x; e
- M_2 aceita y.

Baita problema: não sabemos quando mudar da máquina M_1 para a M_2.
Não sabemos onde *quebrar* w.
Para começo de conversa nem temos w ... M recebe w em prestações, um símbolo por vez ...

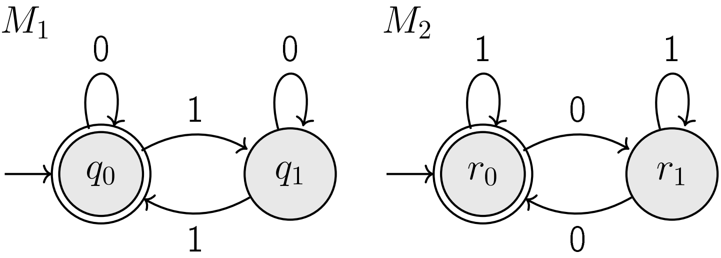
Exemplo $A_1 \circ A_2$



$A_1 = L(M_1) = \{w : w \text{ tem um número par de } 1\text{'s}\}$

$A_2 = L(M_2) = \{w : w \text{ tem um número par de } 0\text{'s}\}$

Exemplo $A_1 \circ A_2$



w =

1	0	0	1	1	0	1	0	1	0	1	...

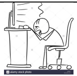
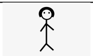
w está em A_1A_2?

Rabiscos . . .

w =

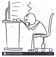
1	0	0	1	1	0	1	0	1	0	1	. . .

Ideia: Usar *threads*. É esse o nome?

	ϵ	1	0	0	1	1	0	1	0	1	. . .
	q_0	q_1	q_1	q_1	q_0	q_1	q_1	q_0	q_0	q_1	. . .
	r_0	r_0	r_1	r_0	r_0	r_0	r_1	r_1	r_0	r_0	. . .
					r_0	r_0	r_1	r_1	r_0	r_0	. . .
								r_0	r_1	r_1	. . .
									r_0	r_0	. . .

Rabiscos . . . 

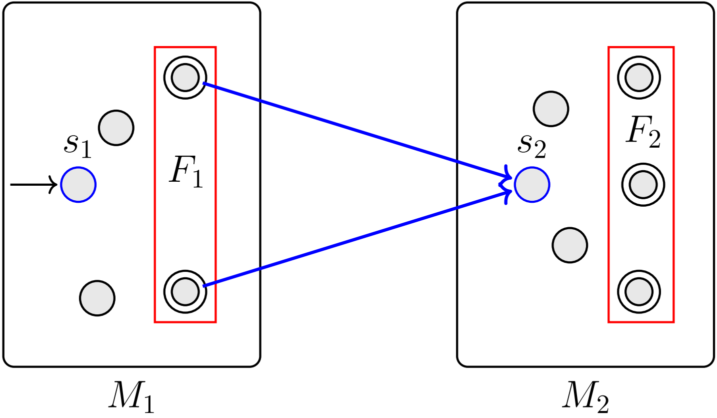
Ao final, **se**

 decide **aceitar** **ou**  decide **aceitar** **ou**
 decide **aceitar** **ou**  decide **aceitar**,
então w é **aceito** como uma string em $A_1 \circ A_2$.

Notemos que além de  andando nos estados de M_1 precisamos de no máximo duas pessoas andando nos estados de M_2, já que M_2 tem dois estados.

Concatenação de linguagens regulares

Teorema. Se A_1 e A_2 são linguagem regulares, então $A_1 \circ A_2$ é regular.



Concatenação de linguagens regulares

Teorema. Se A_1 e A_2 são linguagem regulares, então $A_1 \circ A_2$ é regular.

Rascunho de Demonstração:Considere o autômato $M = (Q, \Sigma, \delta, s, F)$ em que

$$Q = Q_1 \times \mathcal{P}(Q_2) = \{(q, R) : q \in Q_1 \text{ e } R \subseteq Q_2\}$$

$$s = \begin{cases} (s_1, \{s_2\}), & \text{se } s_1 \in F_1 \\ (s_1, \varnothing) & \text{se } s_1 \notin F_1 \end{cases}$$

Concatenação de linguagens regulares

Teorema. Se A_1 e A_2 são linguagem regulares, então $A_1 \circ A_2$ é regular.

$$F = \{(q, R) : q \in Q_1 \textbf{ e } R \cap F_2 \neq \varnothing\}$$

Em palavras, M aceita uma string se e somente se houver um estado na segunda coordenada R que seja um estado de aceitação de M_2.
Então M aceita se e somente se uma das possíveis *threads* de concatenação terminar em um estado de aceitação de M_2.

Concatenação de linguagens regulares

Teorema. Se A_1 e A_2 são linguagem regulares, então $A_1 \circ A_2$ é regular.

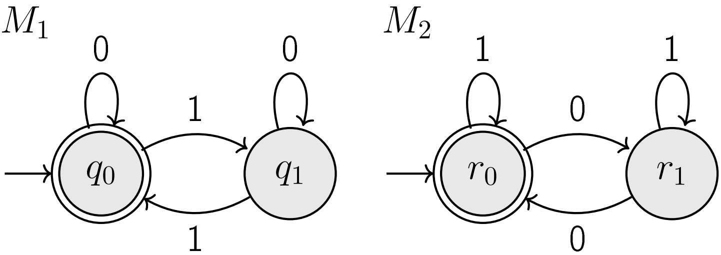
Finalmente, para $(q, R) \in Q$ e $a \in \Sigma$ temos que $\delta((q, R), a)$ é

$$\begin{cases} (\delta_1(q, a), \delta_2(R, a) \cup \{s_2\}), & \text{se } q \in F_1 \text{ (fork!)} \\ (\delta_1(q, a), \delta_2(R, a)), & \text{se } q \notin F_1 \end{cases}$$

em que

$$\delta_2(R, a) = \{\delta_2(r, a) : r \in R\}$$

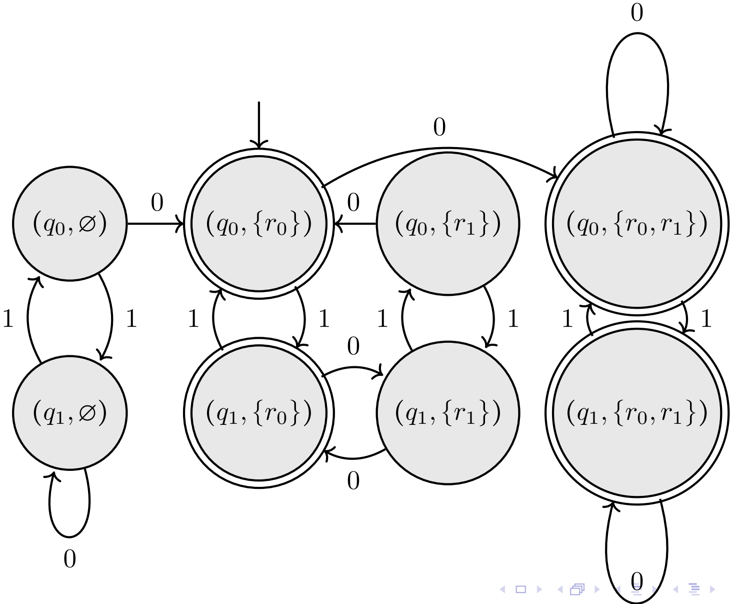
Exemplo $A_1 \circ A_2$



$A_1 = L(M_1) = \{w : w \text{ tem um número par de } 1\text{'s}\}$

$A_2 = L(M_2) = \{w : w \text{ tem um número par de } 0\text{'s}\}$

Exemplo $A_1 \circ A_2$



Exemplo $A_1 \circ A_2$

Se M_1 tem k_1 estados e M_2 tem k_2 estados, então M tem $k_1 \times 2^{k_2}$ estados.

Conclusão

Teorema. Se A_1 e A_2 são linguagem regulares, então $A_1 \circ A_2$ é regular.

Pausa . . .

Observe que M com entrada w acompanha todos os *threads* possíveis e aceita w se e somente se um desses *threads* aceita w.

Agora é uma boa hora para fazermos uma pausa e pedirmos ajuda ao **não-determinismo**.

Apêndice: conjunto das partes

O **conjunto das partes** ou **conjunto potência** de A, denotado por $\mathcal{P}(A)$, é o conjunto

$$\{B : B \subseteq A\}.$$

Por exemplo, o conjunto das partes de $\{1, 2, 3\}$ é

$$\{\varnothing, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$$

e $\mathcal{P}(\varnothing) = \{\varnothing\}$.

Quantos elementos tem $\mathcal{P}(\{1, 2, 3, 4\})$?

Quantos elementos tem $\mathcal{P}(\{1, \ldots, n\})$?

O que é . . . 

- **linguagem formal**: linguagem em que as strings são formadas de acordo com um conjunto específico de regras.
- **linguagem regular**: linguagem que é reconhecida por um autômato finito determinístico.
- **operador estrela**: aplicado a um alfabeto resulta na linguagem de todas as strings sobre o alfabeto. Se Σ é um alfabeto escrevemos $\Sigma^\star$
- **operador estrela**: aplicado a uma linguagem resulta na linguagem formada pela concatenação de zero ou mais strings da linguagem. Se L é uma linguagem escrevemos $L^\star$