بسم الله الرحمن الرحیم

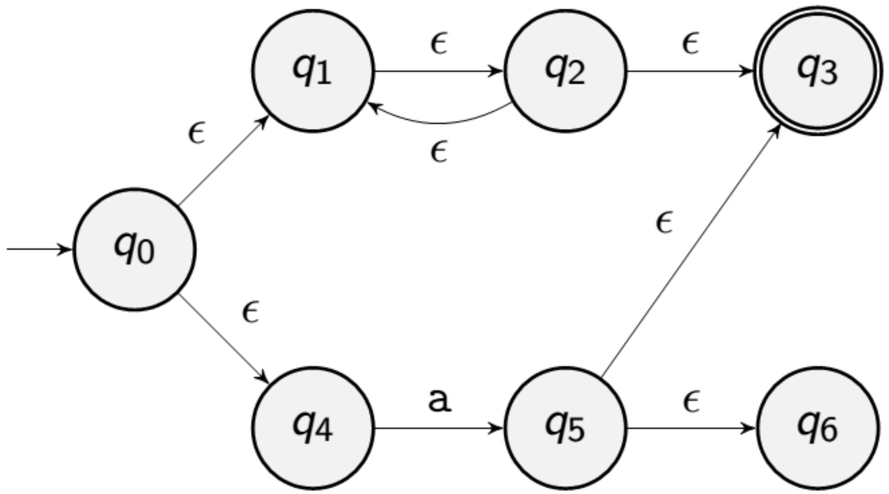
نظریه زبان‌ها و ماشین‌ها

جلسه ۶

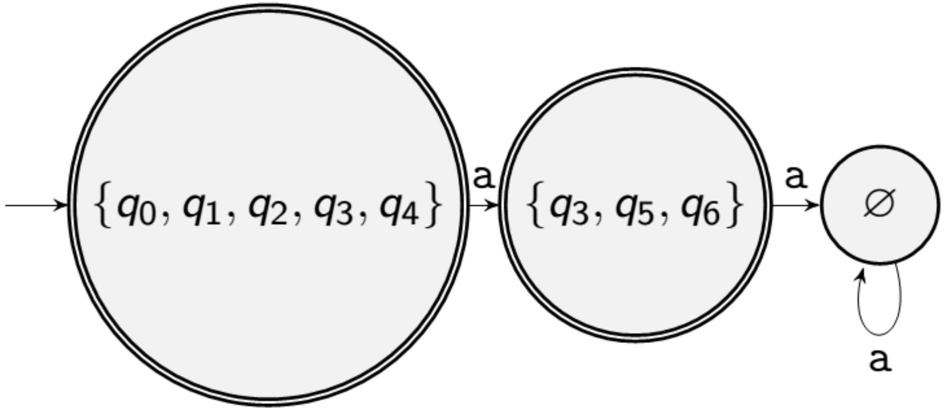
مجتبی خلیلی
دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان

مثال

- تبدیل NFA به DFA (زبان NFA؟): (الفبای {a})



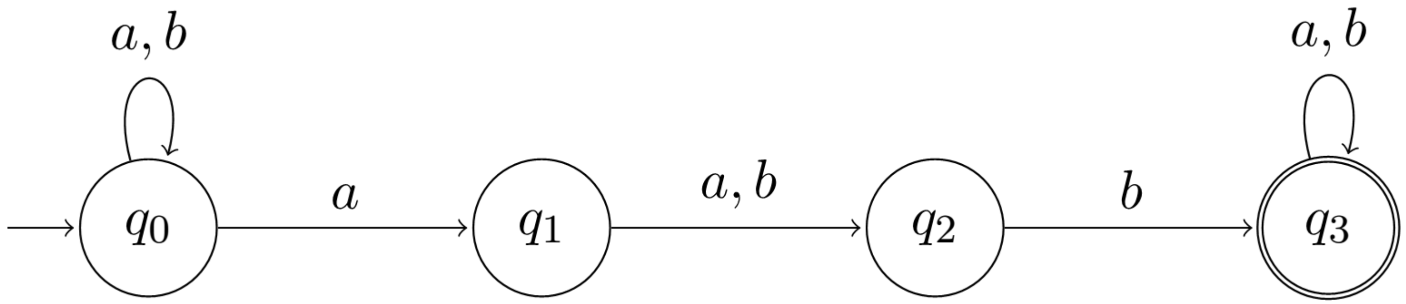
$\{\epsilon, \mathtt{a}\}$



مثال

- زبان روبرو را در نظر بگیرید: همه رشته‌های روی {a,b} به شرط اینکه اگر شامل زیررشته ای باشد که اگر a در موقعیت i باشد، آنگاه b نیز در موقعیت i+2 باشد.

یک NFA برای تشخیص این زبان طراحی کنید.

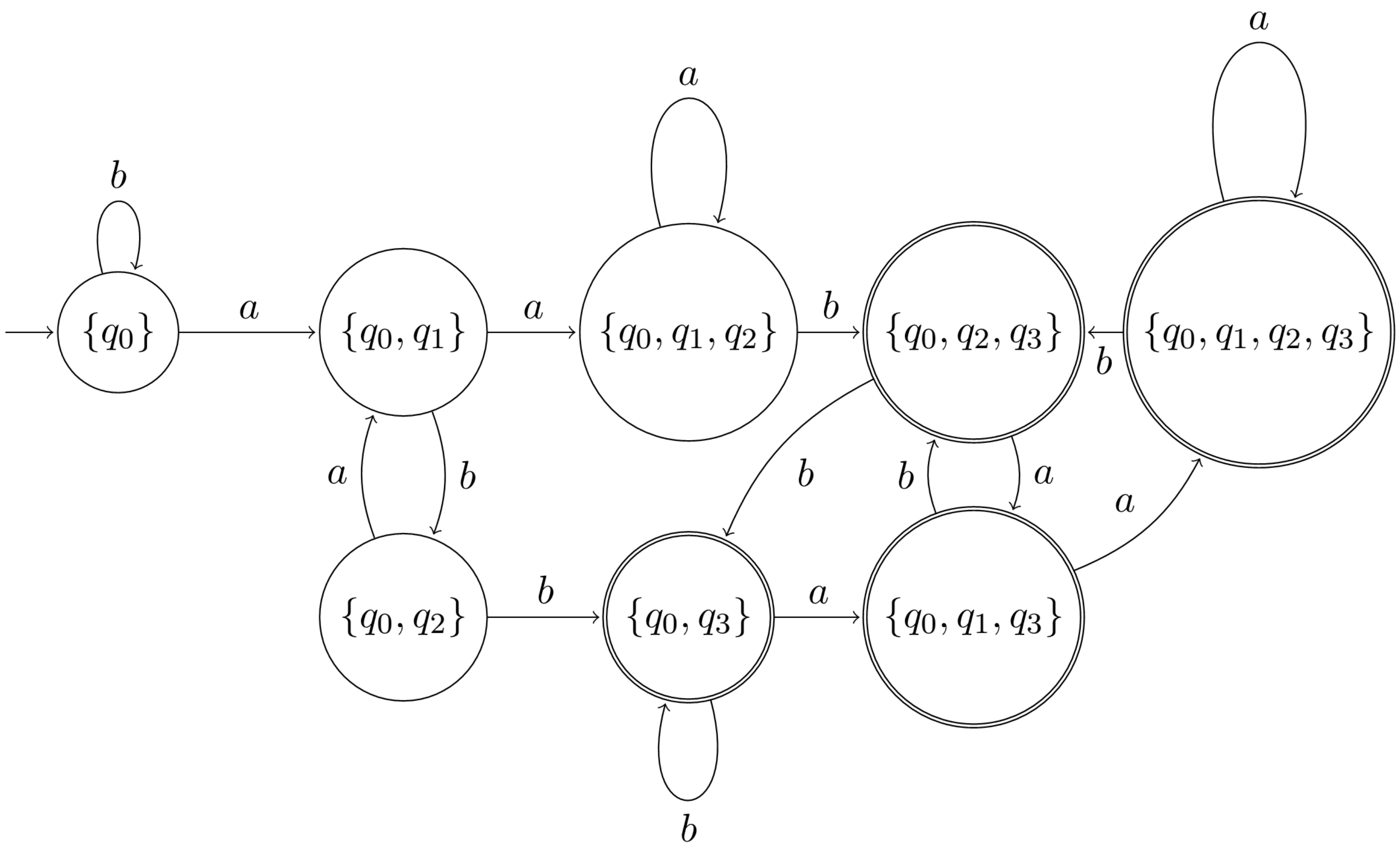


مثال

- زبان روبرو را در نظر بگیرید: همه رشته‌های روی {a,b} به شرط اینکه اگر شامل زیررشته ای باشد که اگر a در موقعیت i باشد، آنگاه b نیز در موقعیت i+2 باشد.

NFA طراحی شده را طبق روال بیان شده به DFA تبدیل کنید (سعی کنید از حالتهای غیرقابل دسترس صرفنظر کنید).

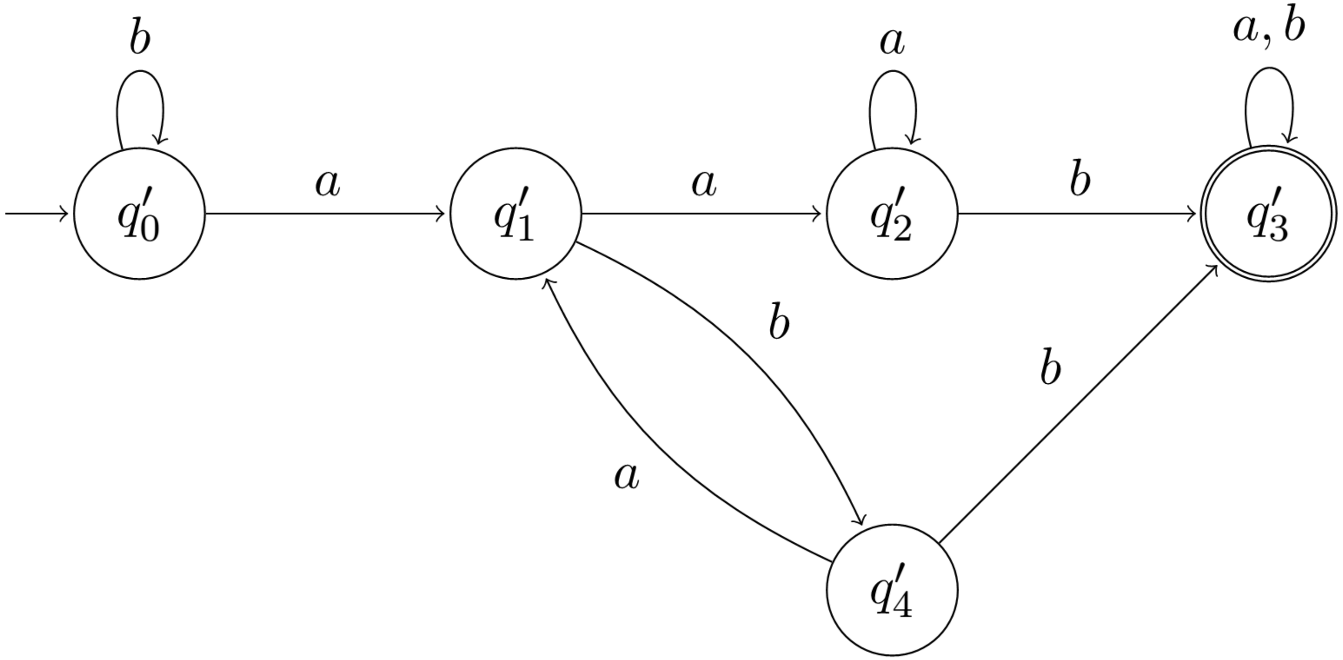
مثال



مثال

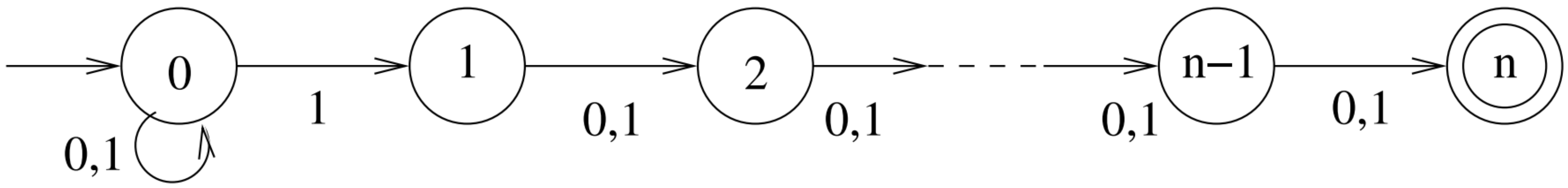
- زبان روبرو را در نظر بگیرید: همه رشته‌های روی {a,b} به شرط اینکه اگر شامل زیررشته ای باشد که اگر a در موقعیت i باشد، آنگاه b نیز در موقعیت i+2 باشد.

اینبار بدون در نظر گرفتن NFA، یک DFA برای تشخیص این زبان طراحی کنید.



مثال

- DFA معادل برای NFA زیر چند حالت دارد؟



زبان منظم و NFA

- زبان L توسط یک DFA تشخیص‌پذیر است اگر و تنها اگر توسط یک NFA تشخیص‌پذیر باشد.
- زبانی که یک NFA تشخیص می‌دهد، زبان منظم است.
- کاربرد؟

بسته بودن

- مثلا اعداد طبیعی تحت عملگر جمع بسته هستند. بنابراین اعمال عملگر جمع بر روی اعضای مجموعه اعداد طبیعی، حاصلی دارد متعلق به مجموعه اعداد طبیعی.

خواص بستاری زبان منظم
(Closure Properties)

IUT-ECE

- چگونه زبان‌های منظم را با هم ترکیب کنیم به طوری که زبان حاصل شده نیز منظم باشد؟

- در کل میخواهیم ببینیم مجموعه زبانهای منظم نسبت به چه عملگرهایی بسته است.

عملگرهای روی زبان منظم

- A و B دو عضو دلخواه از خانواده زبان‌های منظم.

- Complement: $\overline{A} = \{w \mid w \notin A\}$
- Union: $A \cup B = \{ w \mid w \in A \text{ or } w \in B \}$
- Intersection: $A \cap B = \{ w \mid w \in A \text{ and } w \in B \}$
- Reverse: $A^R = \{ w_1 \ldots w_k \mid w_k \ldots w_1 \in A \}$
- Concatenation: $A \circ B = \{ vw \mid v \in A \text{ and } w \in B \}$
- Star: $A^* = \{ w_1 \ldots w_k \mid k \geq 0 \text{ and each } w_i \in A \}$
  $= \{\varepsilon\} \cup A \cup AA \cup AAA \cup AAAA \cup \ldots$

اجتماع

- Union: $A \cup B = \{ w \mid w \in A \text{ or } w \in B \}$

THEOREM 1.25

The class of regular languages is closed under the union operation.

اجتماع

- اثبات ۱ با ساخت یک DFA:

$$A_1 = L(M_1)$$

$$A_2 = L(M_2)$$

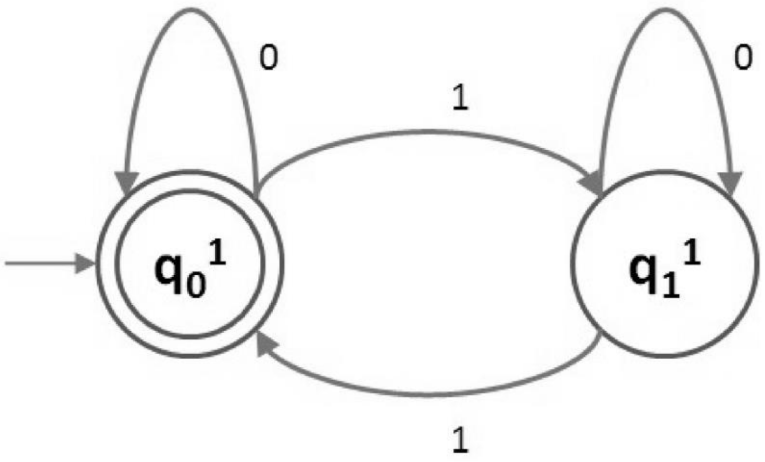
$$A_1 \cup A_2 = L(M)$$

DFA

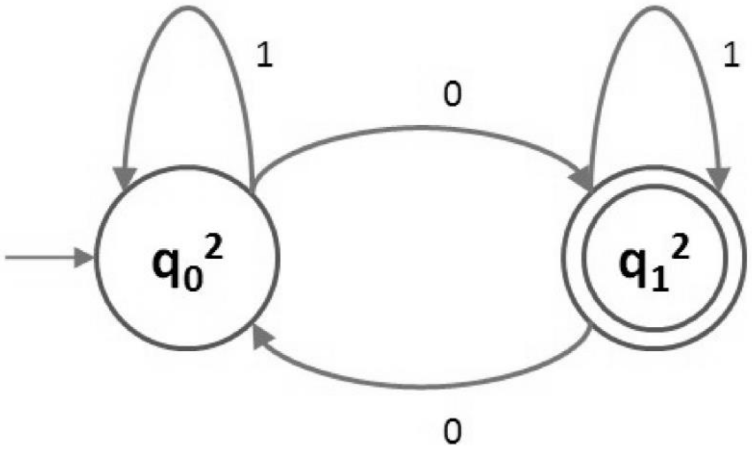
- هدف: ساخت M

اجتماع

- اثبات ۱ با ساخت یک DFA: DFA برای $L_1 \; U \; L_2$



M_1



M_2

اجتماع

- ایده ۱: ساخت یک DFA که به صورت سری، M1 و M2 را اجرا کند.

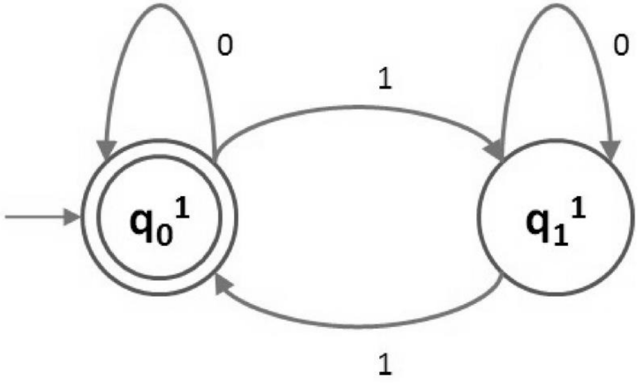
How can we make machine M simulate M_1 and M_2? Perhaps it first simulates M_1 on the input and then simulates M_2 on the input. But we must be careful here! Once the symbols of the input have been read and used to simulate M_1, we can't "rewind the input tape" to try the simulation on M_2. We need another approach.

اجتماع

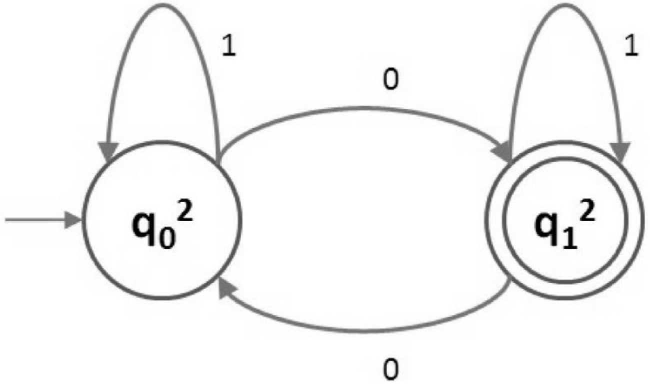
- ایده ۲: ساخت یک DFA که همزمان M1 و M2 را اجرا کند.

Pretend that you are M. As the input symbols arrive one by one, you simulate both M_1 and M_2 simultaneously. That way, only one pass through the input is necessary. But can you keep track of both simulations with finite memory? All you need to remember is the state that each machine would be in if it had read up to this point in the input. Therefore, you need to remember a pair of states. How many possible pairs are there? If M_1 has k_1 states and M_2 has k_2 states, the number of pairs of states, one from M_1 and the other from M_2, is the product $k_1 \times k_2$. This product will be the number of states in M, one for each pair. The transitions of M go from pair to pair, updating the current state for both M_1 and M_2. The accept states of M are those pairs wherein either M_1 or M_2 is in an accept state.

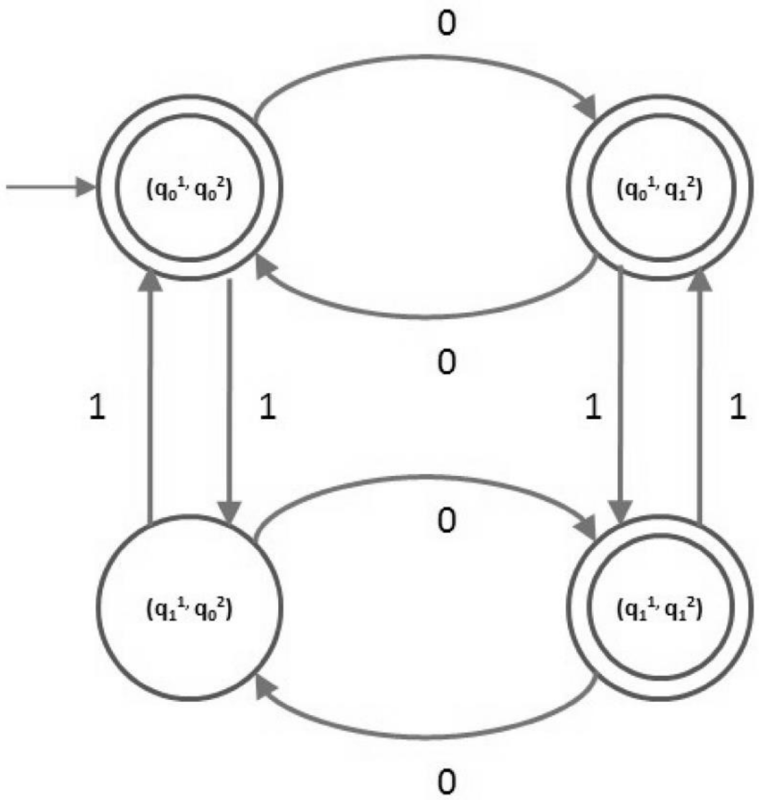
اجتماع


0
1
0
q_0^1
q_1^1
1

M_1


1
0
1
q_0^2
q_1^2
0

M_2


0
(q_0^1, q_0^2)
(q_0^1, q_1^2)
0
1
1
1
1
0
(q_1^1, q_0^2)
(q_1^1, q_1^2)
0

اجتماع (اثبات)

PROOF

Let M_1 recognize A_1, where $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$, and
M_2 recognize A_2, where $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$.

Construct M to recognize $A_1 \cup A_2$, where $M = (Q, \Sigma, \delta, q_0, F)$.

اجتماع

$$M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1),$$
$$M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2).$$

$$\longrightarrow \quad M = (Q, \Sigma, \delta, q_0, F).$$

1. $Q = \{(r_1, r_2) | \ r_1 \in Q_1 \text{ and } r_2 \in Q_2\}$.
This set is the ***Cartesian product*** of sets Q_1 and Q_2 and is written $Q_1 \times Q_2$.
It is the set of all pairs of states, the first from Q_1 and the second from Q_2.

اجتماع

$$M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1),$$
$$M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2).$$

$$\longrightarrow$$

$$M = (Q, \Sigma, \delta, q_0, F).$$

2. Σ, the alphabet, is the same as in M_1 and M_2.

اجتماع

IUT-ECE

$$M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1),$$
$$M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2).$$

$$\longrightarrow \quad M = (Q, \Sigma, \delta, q_0, F).$$

3. δ, the transition function, is defined as follows. For each $(r_1, r_2) \in Q$ and each $a \in \Sigma$, let

$$\delta\big((r_1, r_2), a\big) = \big(\delta_1(r_1, a), \delta_2(r_2, a)\big).$$

اجتماع

$$M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1),$$
$$M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2).$$

$\longrightarrow$

$$M = (Q, \Sigma, \delta, q_0, F).$$

4. q_0 is the pair (q_1, q_2).

5. F is the set of pairs in which either member is an accept state of M_1 or M_2. We can write it as

$$F = \{(r_1, r_2) |\ r_1 \in F_1 \text{ or } r_2 \in F_2\}.$$

This expression is the same as $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$.

اشتراک

- Intersection: L1 $\cap$ L2 = { w | w $\in$ L1 and w $\in$ L2 }

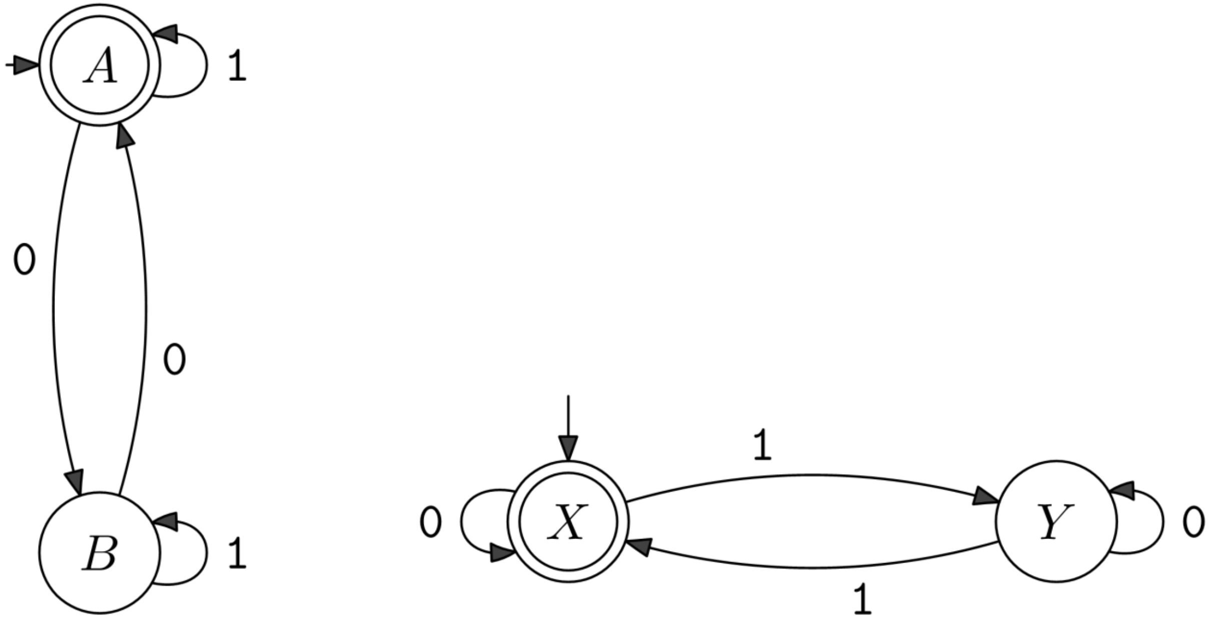
- اثبات مانند اثبات اجتماع، تنها تفاوت در گام ۵ است که داریم:

$$F = F_1 \times F_2.$$

مثال

IUT-ECE

- یک DFA که زبان متناظر با آن برابر است با اشتراک دو زبان متناظر با دو DFA زیر؟



مثال

- یک DFA که زبان متناظر با آن برابر است با اشتراک دو زبان متناظر با دو DFA زیر؟