Biçimsel Diller ve Soyut Makineler

Hafta 4

Regüler Dillerin kapalılık özelliği

- Regüler diller aşağıdaki işlemlerde kapalılık özelliğine sahiptir.
  - Birleşim
    - Gösterim: $\cup$
  - Kesişim
    - Gösterim: $\cap$
- **Eğer L_1 ve L_2 regüler ise $L_1 \cup L_2$ ve $L_1 \cap L_2$ regulerdir.**

Örnek

$\Sigma = \{a,b\}$.

$L_1 = \{ w \in \{a,b\}^* \mid w$ çift sayıda a içerir.$\}$

– **L_1** regular midir?

L_2 $= L_2 = \{ w \in \{a,b\}^* \mid w$ tek sayıda b içerir.$\}$

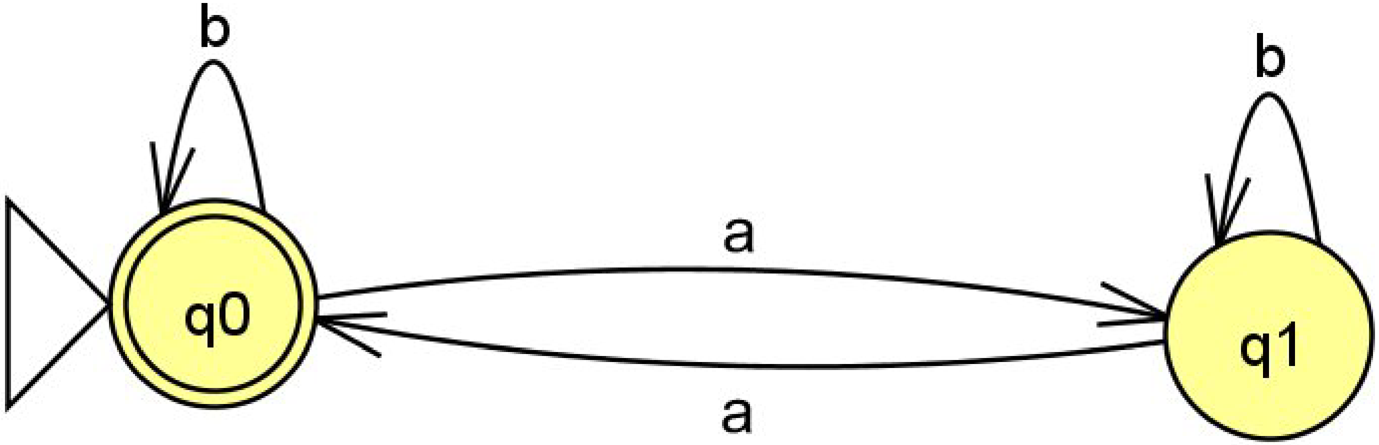
– **L_2** regular midir?

$L_1 \cup L_2$ = ?

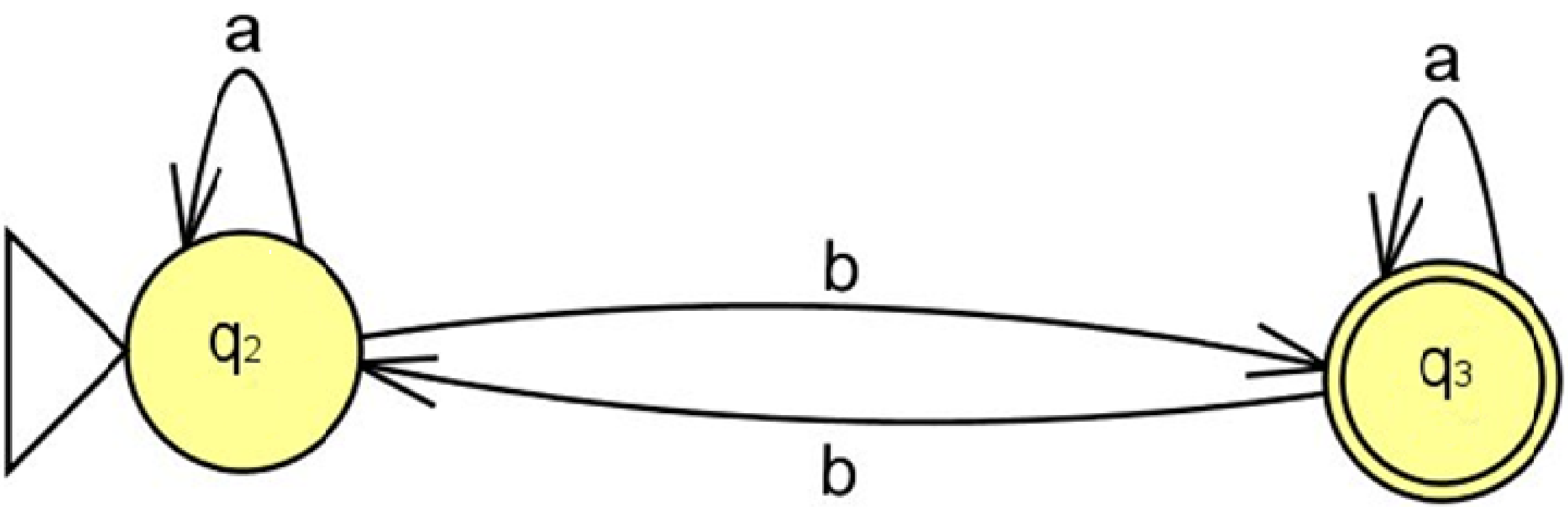
– $L_1 \cup L_2 =$ {w ∈ {a,b}* | w, çift sayıda a VEYA tek sayıda b içerir.} **$L_1 \cap L_2$ = ?**

– $L_1 \cap L_2 =$ {w ∈ {a,b}* | w, çift sayıda a VE tek sayıda b içerir.}

L_1 = { w Є{a,b}* | w çift sayıda a içerir.} kümesi için DFA



L_2 = { w Є{a,b}* | w tek sayıda b içerir.} kümesi için DFA



$\cup$ ve $\cap$ için DFA gerçekleştirme

$M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ ve
$M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ makineleri verilmiş olsun.
Yeni bir makine $\cup$ ve $\cap$ için tasarlamak istiyoruz.
$M = (Q, \Sigma, \delta, s, F)$ bu makine olsun. Burada

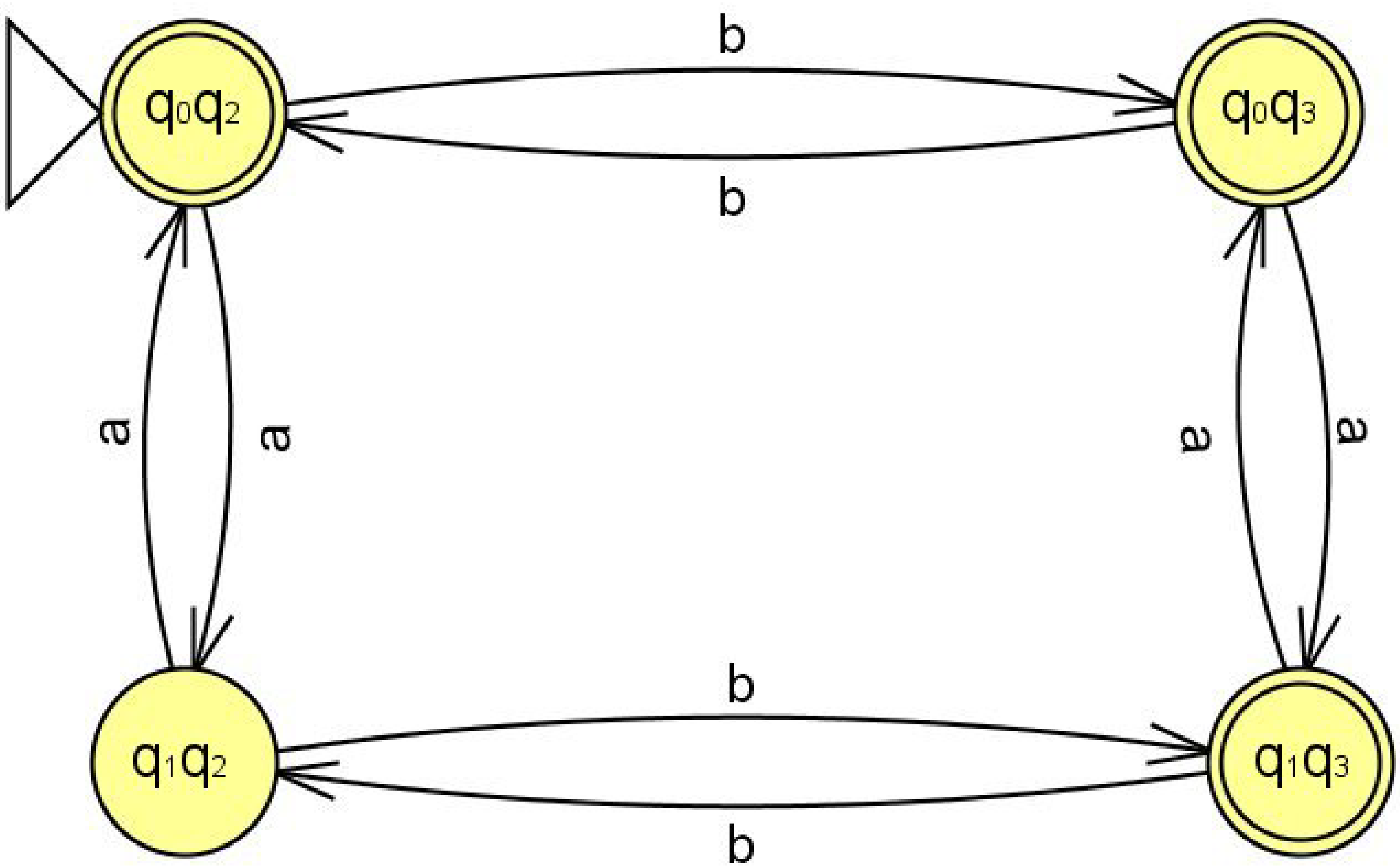
$$Q = Q_1 \text{ X } Q_2$$

$$s = (s_1, s_2)$$

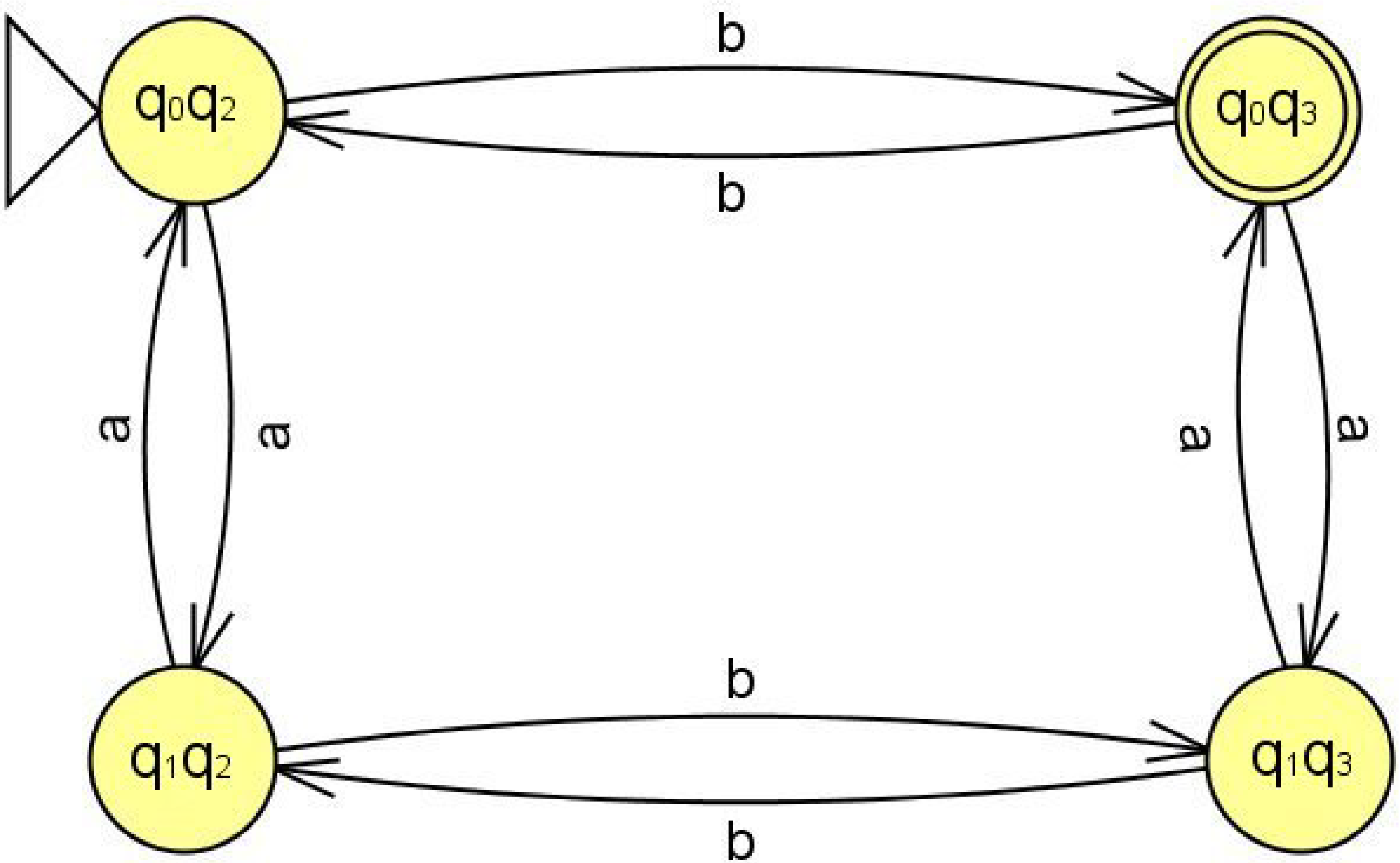
$$\delta((q_1, q_2), \sigma) = (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma))$$

- Birleşim kümesi için, F = ?
  - Cevap: $(Q_1 \text{ X } F_2) \text{ U } (F_1 \text{ X } Q_2)$
- Kesişim Kümesiiçin, F = ?
  - Cevap: $F_1 \text{ X } F_2$

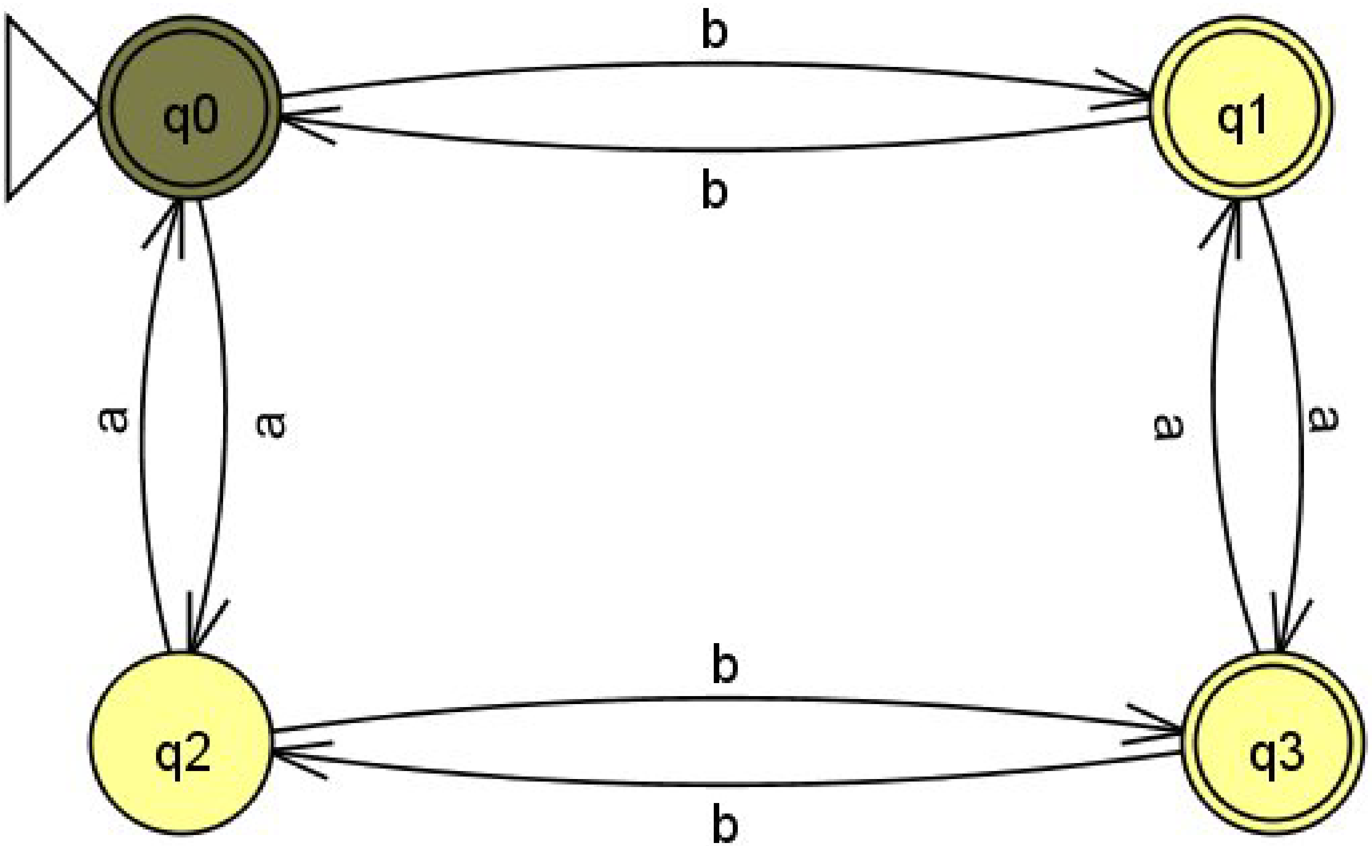
$L_1 \cup L_2 =$ {w Є {a,b}* | w, çift sayıda a VEYA tek sayıda b içerir.}kümesi için DFA



$L_1 \cap L_2$ = {w Є {a,b}* | w, çift sayıda a VE tek sayıda b içerir.}kümesi için DFA

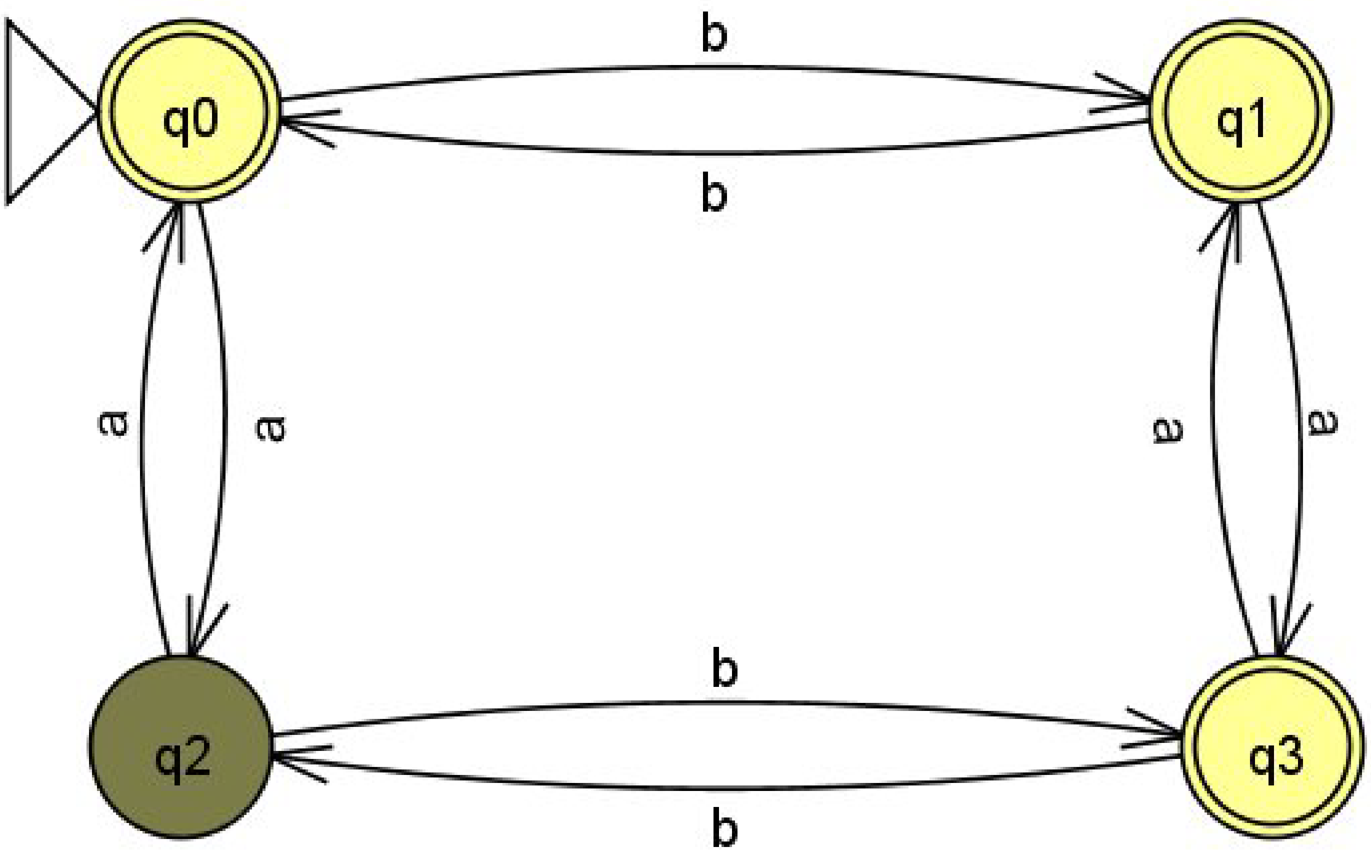


$L_1 \cup L_2$ = {w Є {a,b}* | w, çift sayıda a VEYA tek sayıda b içerir.}kümesi için DFA



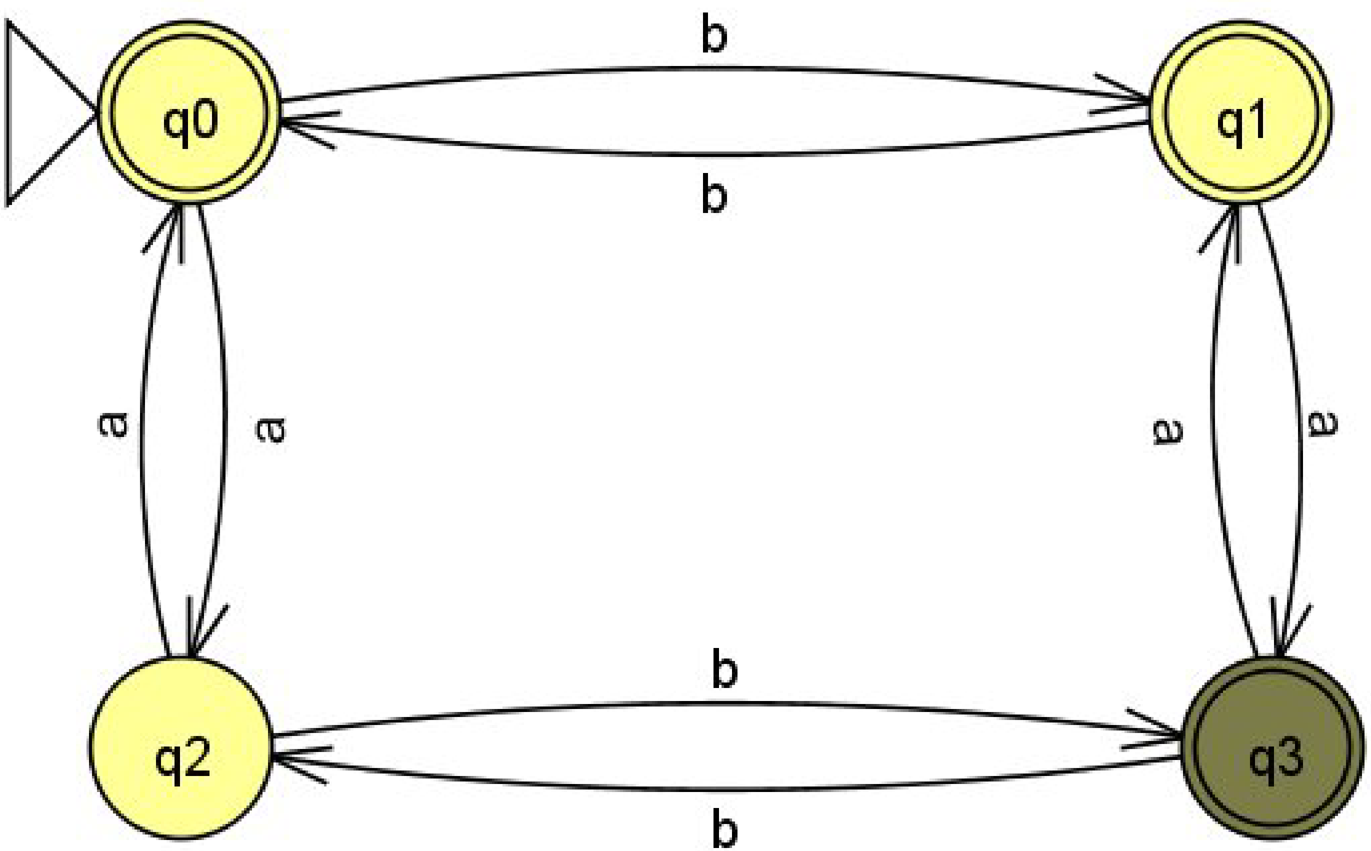
aba

$L_1 \cup L_2$ = {w Є {a,b}* | w, çift sayıda a VEYA tek sayıda b içerir.}kümesi için DFA



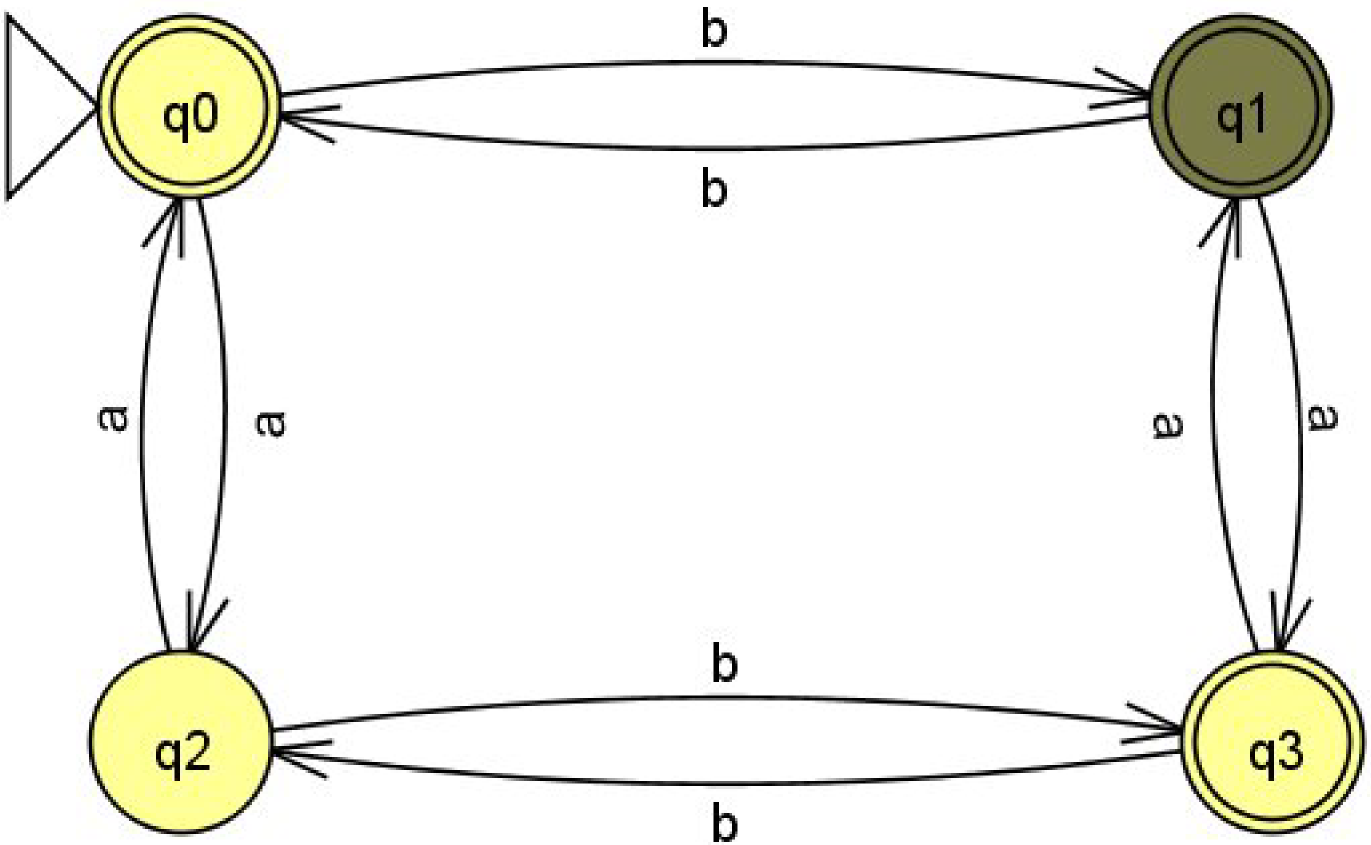
aba

$L_1 \cup L_2$ = {w Є {a,b}* | w, çift sayıda a VEYA tek sayıda b içerir.}kümesi için DFA



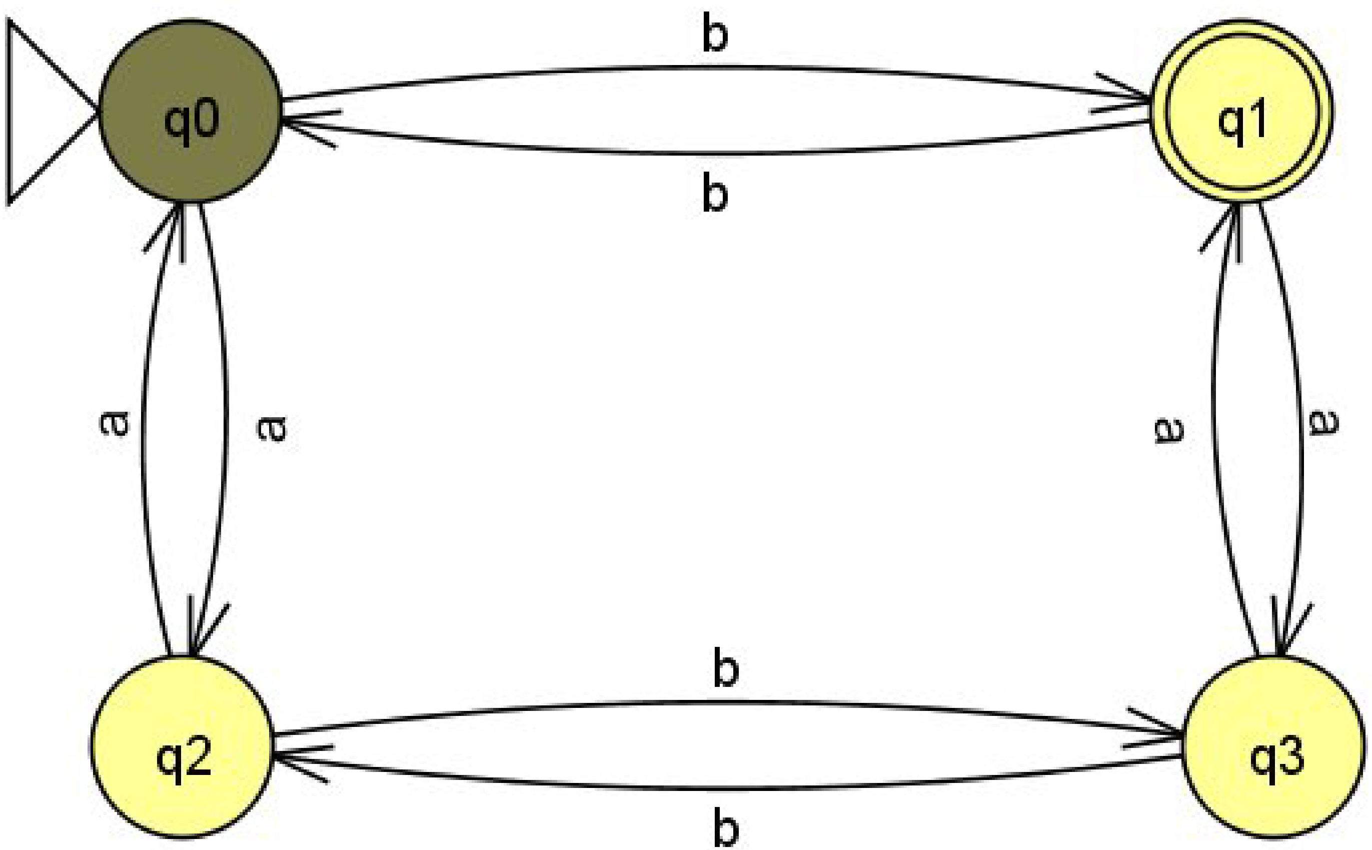
aba

$L_1 \cup L_2$ = {w Є {a,b}* | w, çift sayıda a VEYA tek sayıda b içerir.}kümesi için DFA



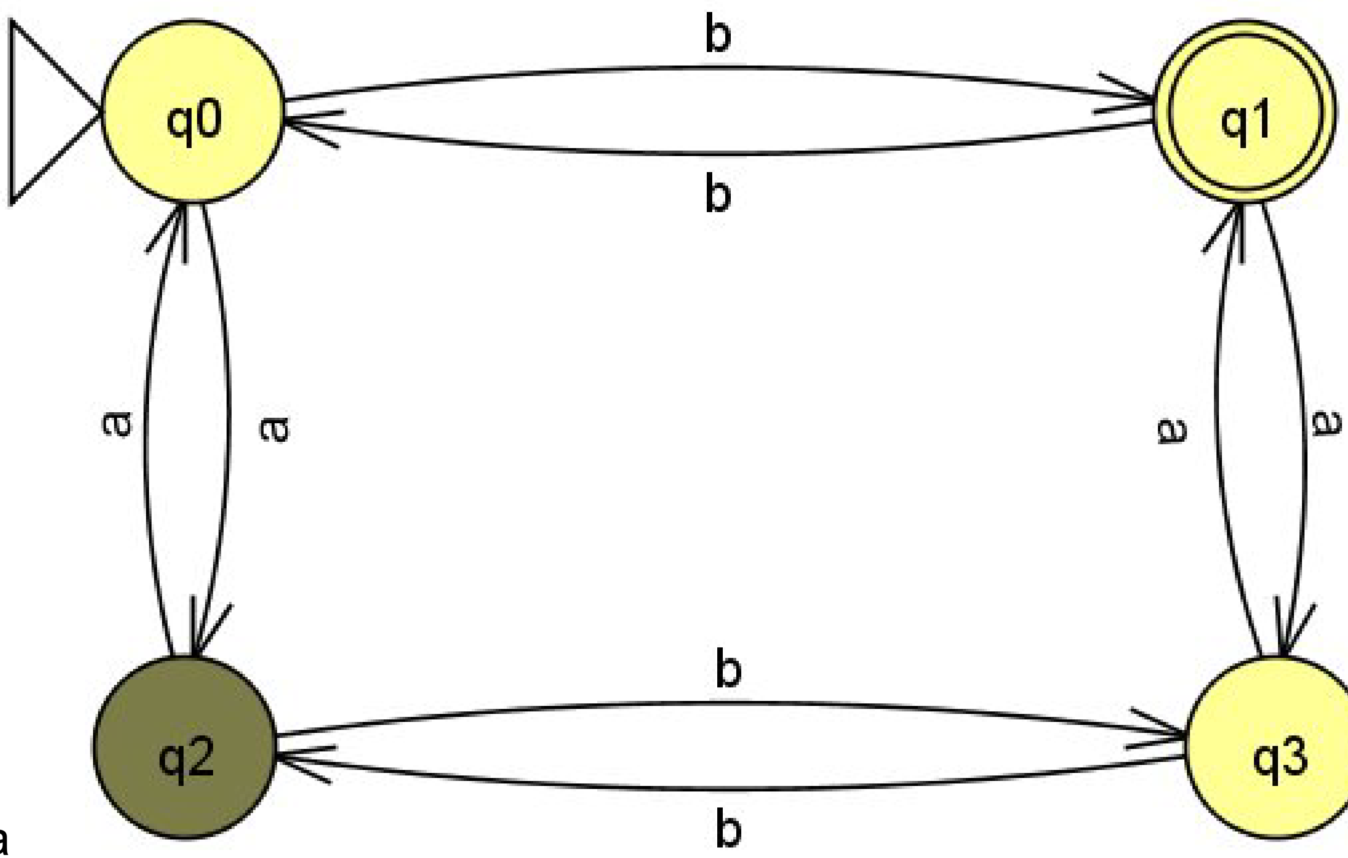
aba

$L_1 \cap L_2$ = {w Є {a,b}* | w, çift sayıda a VE tek sayıda b içerir.}kümesi için DFA



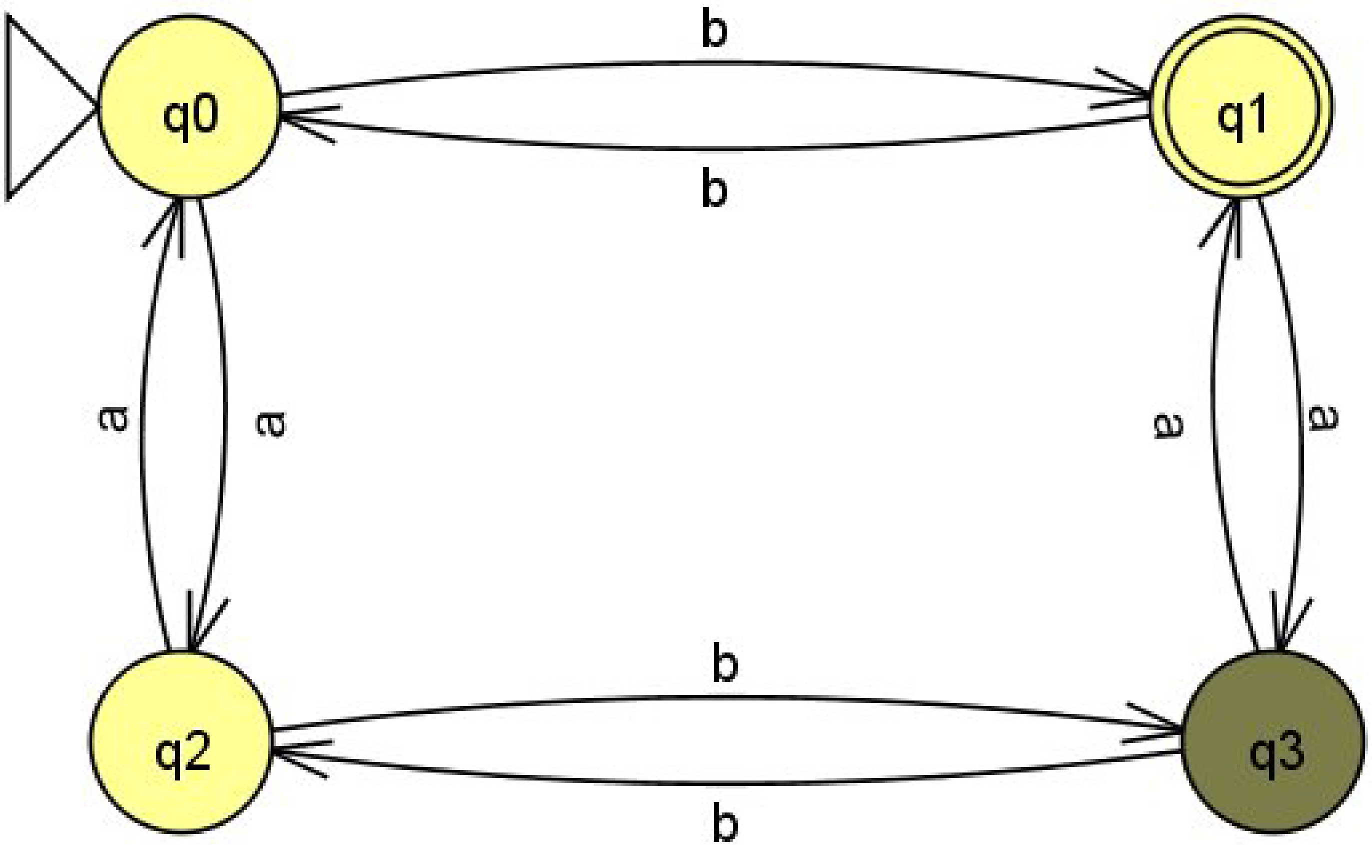
aba

$L_1 \cap L_2$ = {w Є {a,b}* | w, çift sayıda a VE tek sayıda b içerir.}kümesi için DFA



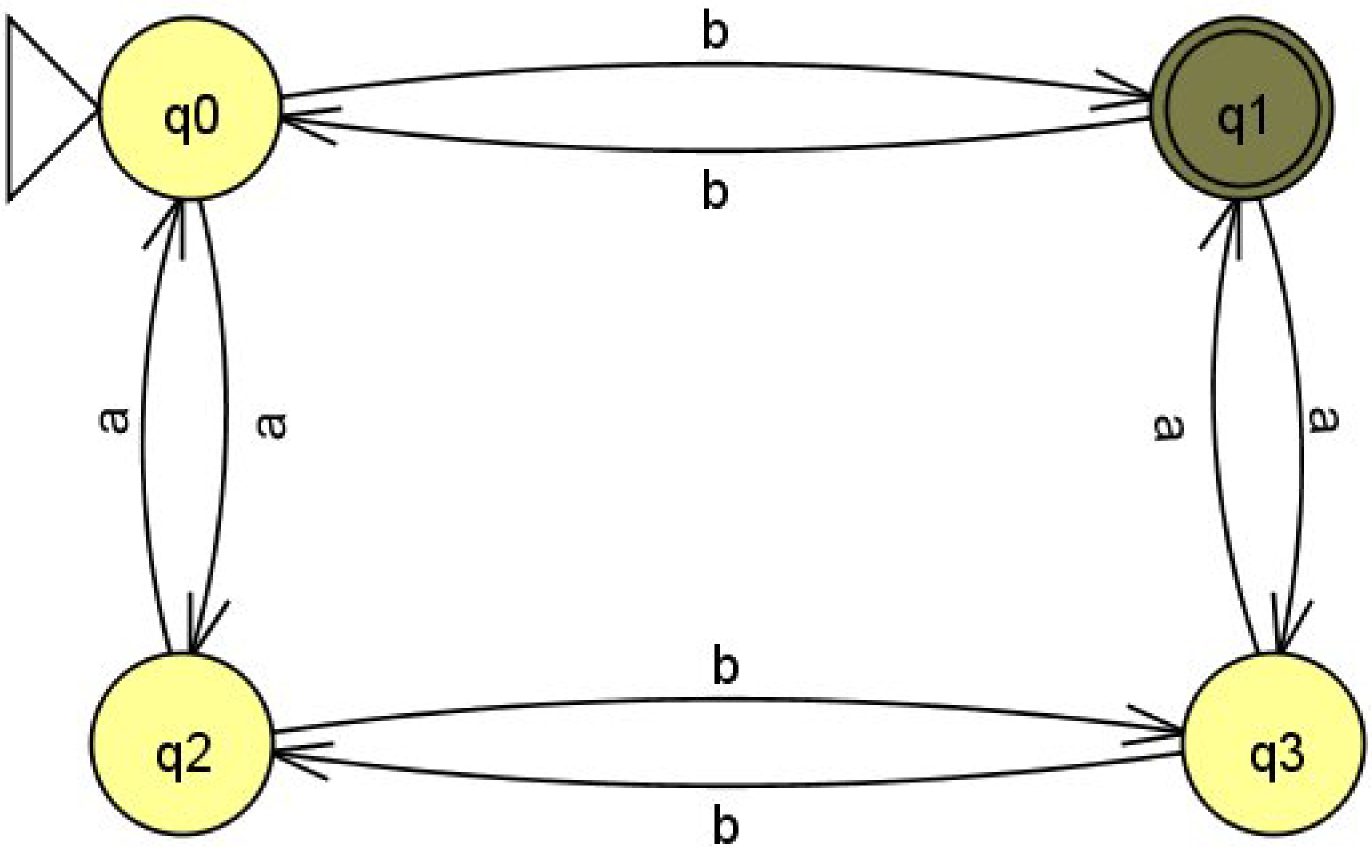
aba

$L_1 \cap L_2$ = {w Є {a,b}* | w, çift sayıda a VE tek sayıda b içerir.}kümesi için DFA



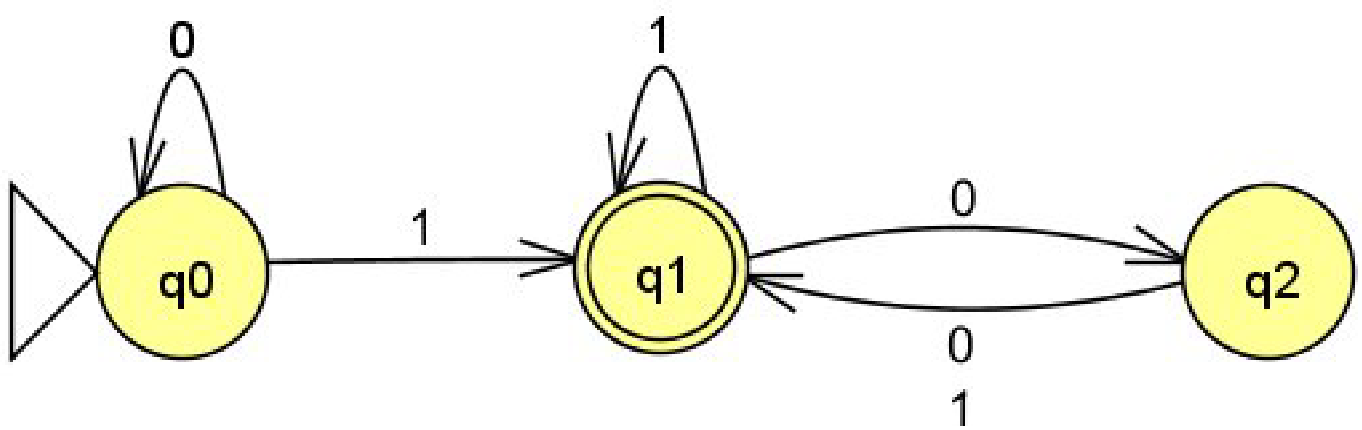
aba

$L_1 \cap L_2$ = {w Є {a,b}* | w, çift sayıda a VE tek sayıda b içerir.}kümesi için DFA



aba

A = {w | w, en az bir tane 1 içerir ve son 1'i çift sayıda 0 izler} kümesi için DFA



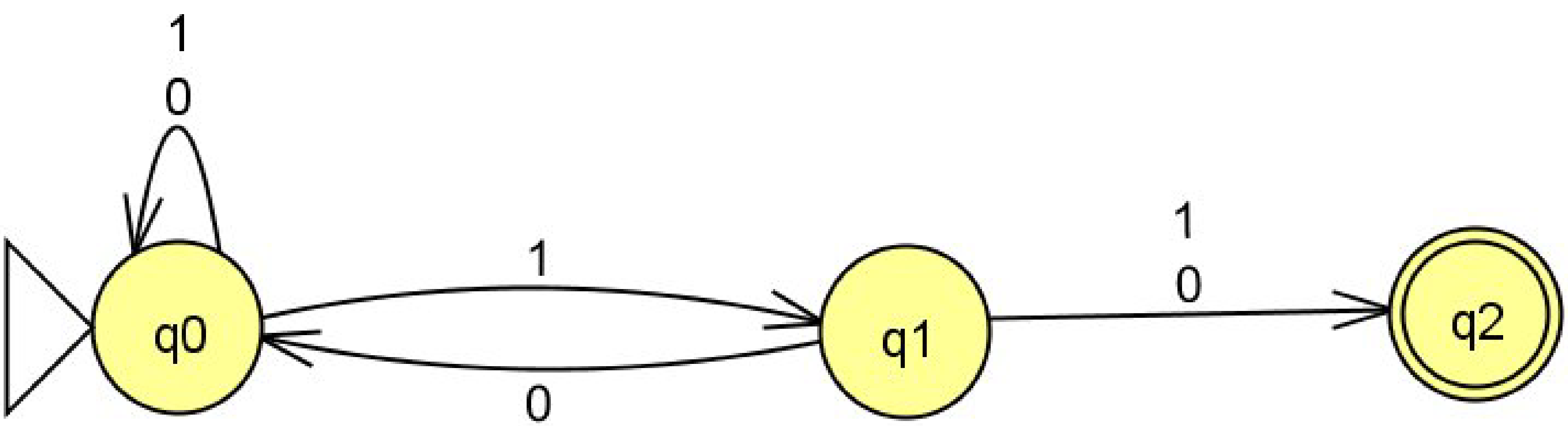
Nondeterministic Finite Automaton (NFA)

- DFA’nın daha genelleştirilmiş biçimidir.
  - Herhangi bir durumda iken bu durumdan bazı geçişler olmayabilir.
  - Bir geçişten birden fazla olabilir.
- Avantaj: Esneklik
  - Tasarım daha kolay hale gelmektedir.

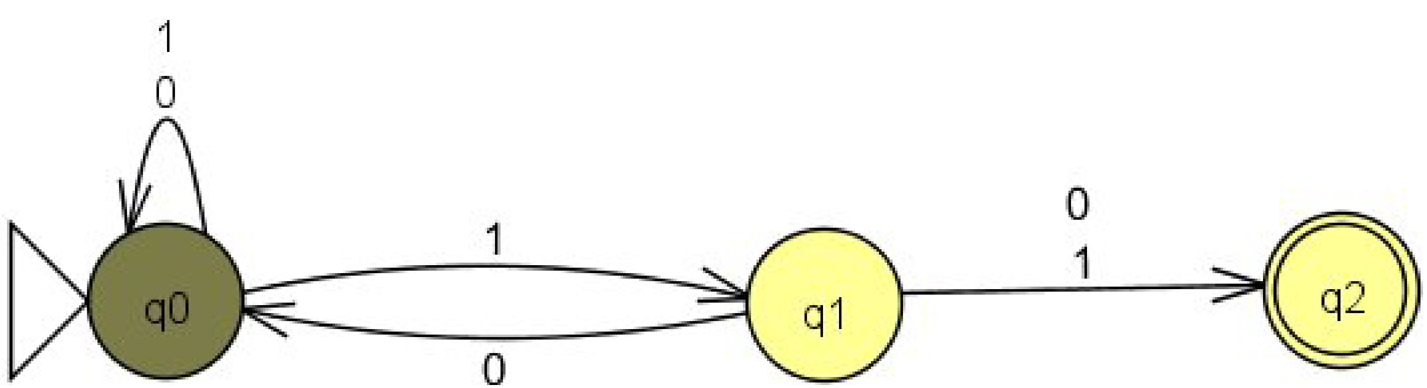
NFA nasıl çaışır?

- NFA’nın başlangıç durumundan başlanarak, ilgili katar izlenip bir kabul durumunda biterse w NFA tarafından kabul edilir.
- NFA tarafından kabul edilen dil, bu NFA tarafından kabul edilen karakter katarlarının kümesidir.

NFA A = {w in {0,1}* | w’nin sondan ikinci sembolü 1’dir}

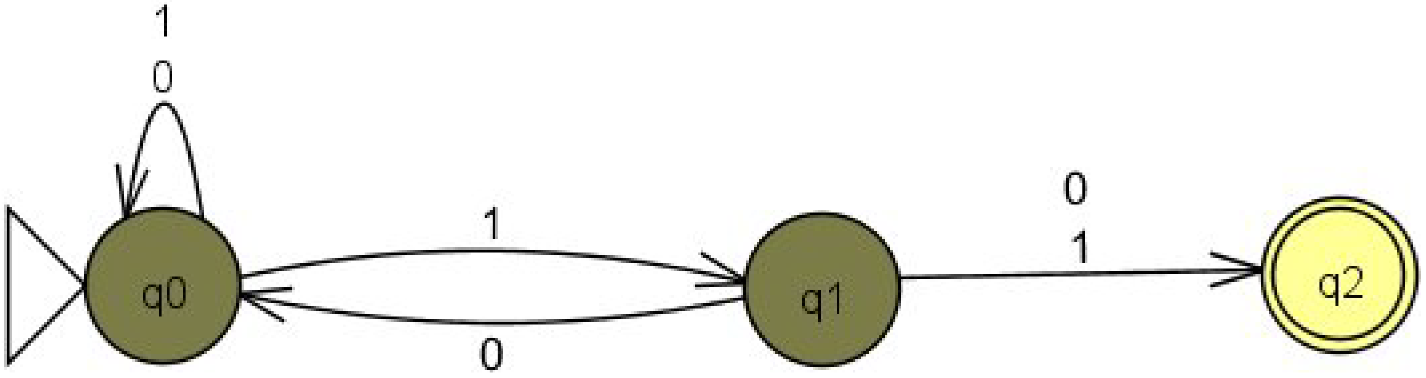


NFA A = {w in {0,1}* | w'nin sondan ikinci sembolü 1'dir}



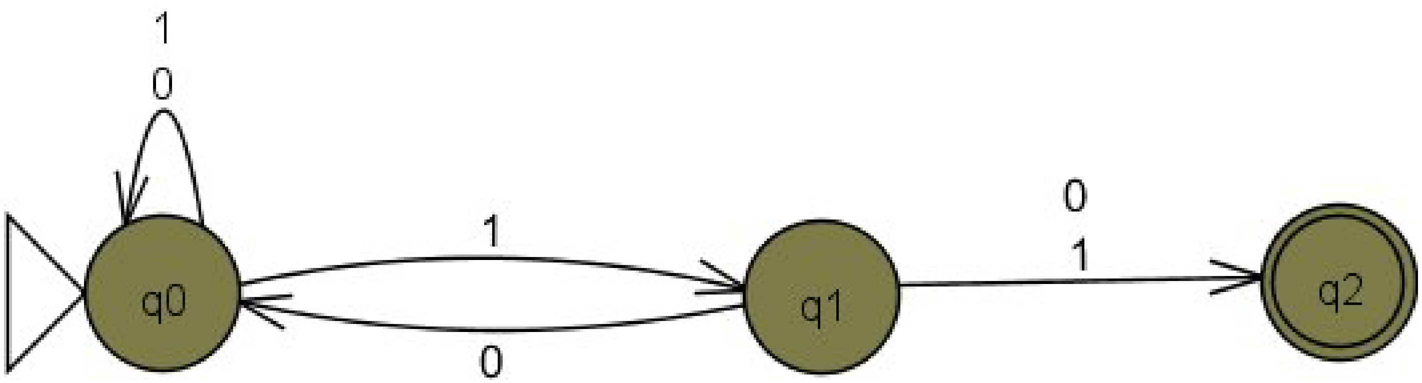
110

NFA A = {w in {0,1}* | w’nin sondan ikinci sembolü 1’dir}



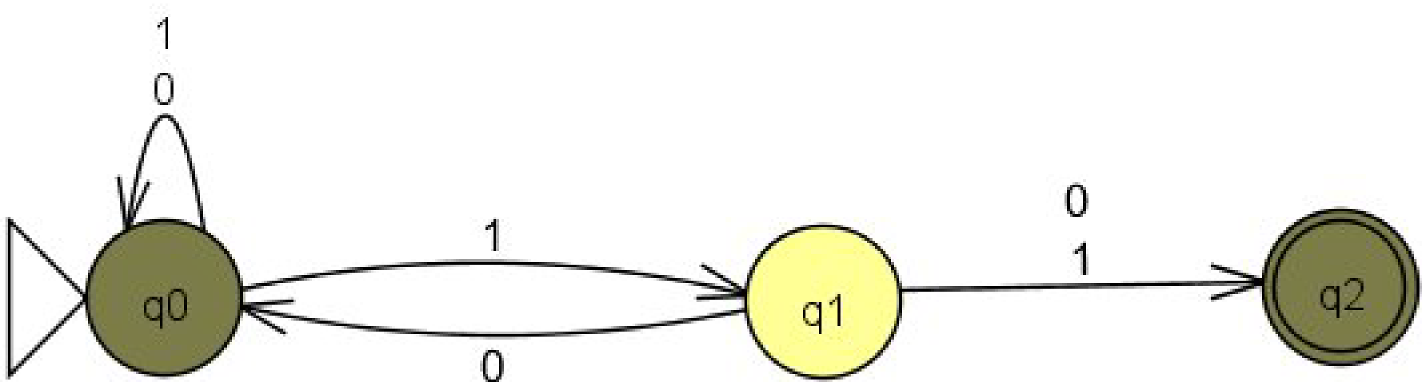
110

NFA A = {w in {0,1}* | w'nin sondan ikinci sembolü 1'dir}



110

NFA A = {w ∈{0,1}* |w'nin sondan ikinci sembolü 1'dir}



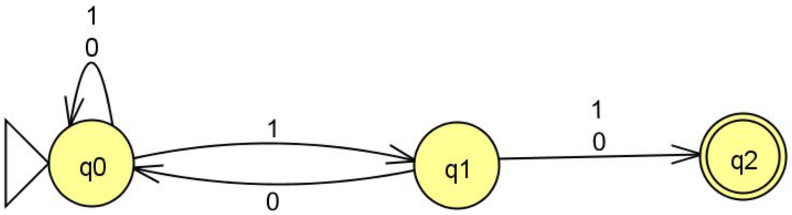
110

NFA'nın biçimsel tanımı

- NFA $M = (Q, \Sigma, \delta, s, F)$ Burada;
  - Q – Durumların sonlu kümesi
  - Σ - Giriş alfabesi
  - s – Başlangıç durumu
  - $F \subseteq Q$ – Kabul durumları kümesi
  - δ bir durum geçiş fonksiyonudur ve $Q \times \Sigma_e \times Q$'nin alt kümesidir.
- (p, u, q) δ'de ise , NFA p durumunda u okuyabilir ve q 'ya gider.

NFA'nın biçimsel tanımı (devam)

- $\delta^*(q, w)$ bir durumlar kümesidir ve
- $p \in \delta^*(q, w)$ ise q'dan p'ye w etiketli bir yol vardır.
  - Örnek:
- $\delta^*(q_0, 1) = ?$
  - Cevap: $\{q_0, q_1\}$
- $\delta^*(q_0, 11) = ?$
  - Cevap: $\{q_0, q_1, q_2\}$



NFA kabulü

- $\delta^*(q_0, w) \cap F$ kümesi bir boş küme değilse w karakter katarı M makinesi tarafından tanınır.

NFA'nın tanıdığı dil:

- $L(M) = \{w \text{ in } \Sigma^* \mid w, M \text{ tarafından tanınır}\}$.

NFA ve DFA'nın karşılaştırılması

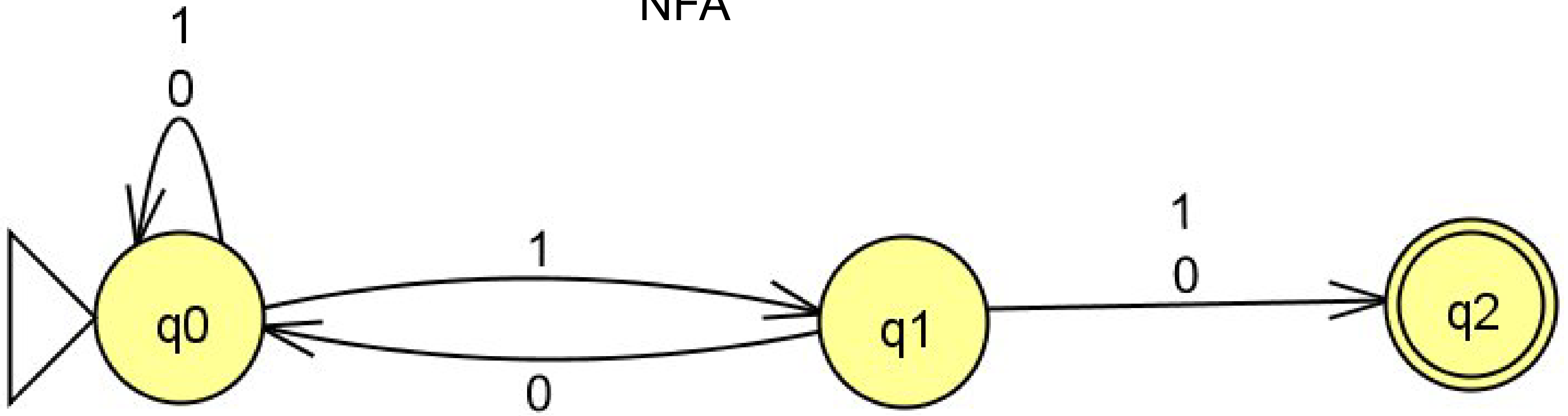
- NFA , DFA'dan daha mı güçlüdür?
  - Cevap: Hayır
- Theorem:
  - Her NFA makinesi için eşdeğer bir DFA vardır.

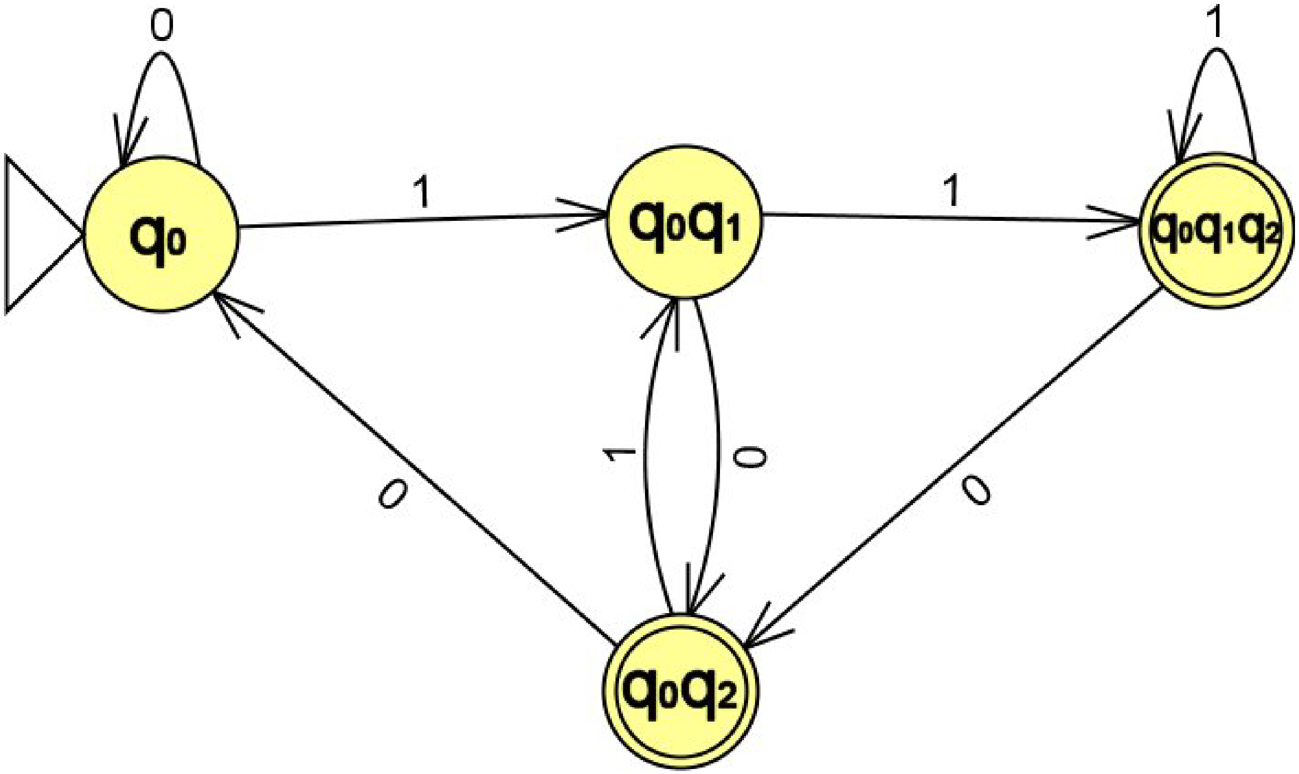
Eşdeğer DFA'nın bulunması

- NFA $M = (Q, \Sigma, \delta, s, F)$
- DFA $M' = (Q', \Sigma, \delta, s', F')$ Burada:
  - $Q' = 2^Q$
  - $s' = \{s\}$
  - $F' = \{P \mid P \cap F \neq \Phi\}$
  - $\delta(\{p_1, p_2, p_m\}, \sigma) = \delta^*(p_1, \sigma) \cup \delta^*(p_2, \sigma) \cup \ldots \cup \delta^*(p_m, \sigma)$

Örnek:Eşdeğer DFA’nın bulunması

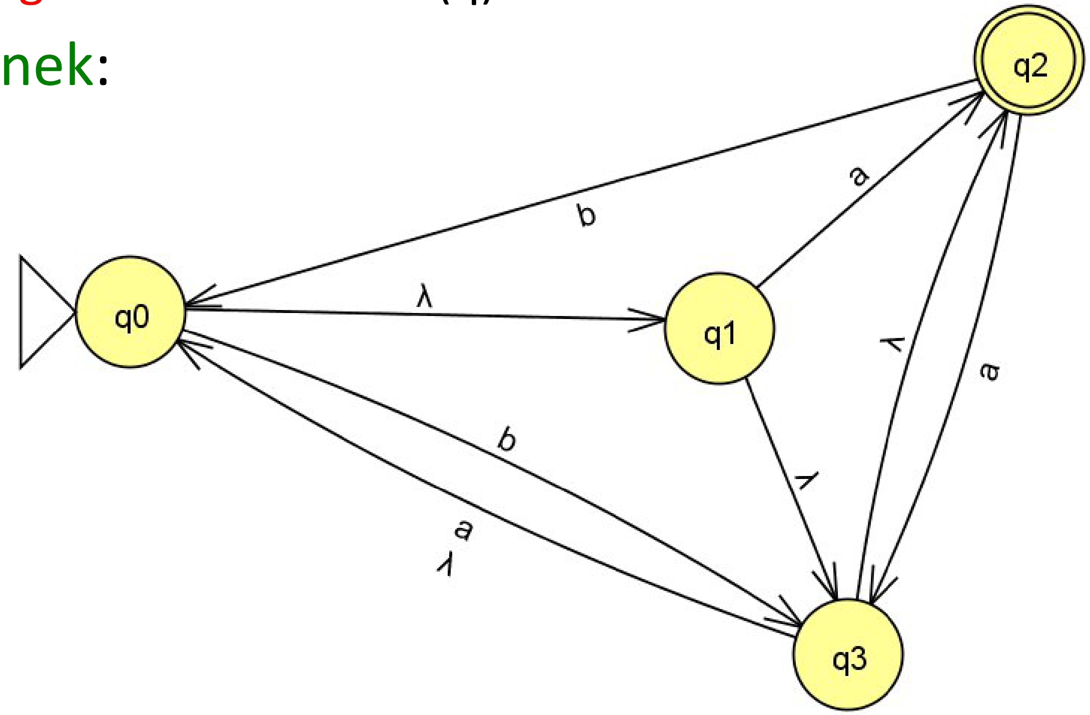
NFA




0
1
1
1
q0
q0q1
q0q1q2
1
0
0
0
q0q2

Boşluk geçişli NFA

- Durumların boşluk kapanması: $\delta^*(q, \Lambda)$.
  - gösterim: e-closure(q).
- Örnek:



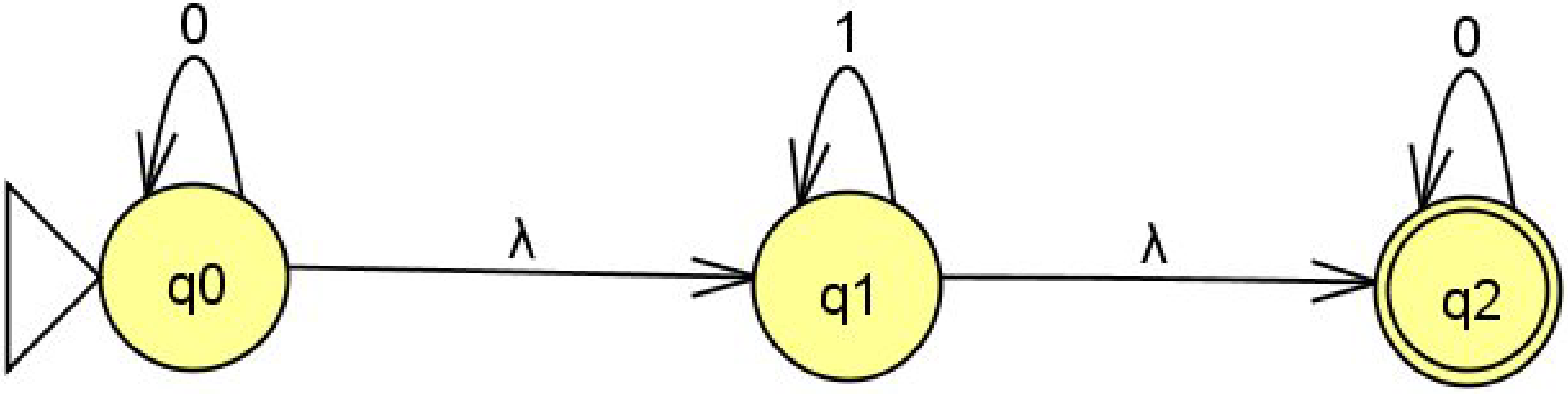
Handling epsilon transitions (contd.)

- Durumun boşluk kapanmasının bulunması:
  - e-closure($\{s_1, \ldots , s_m\}$) = e-closure(s_1) $\cup \ldots \cup$ e-closure(s_m)

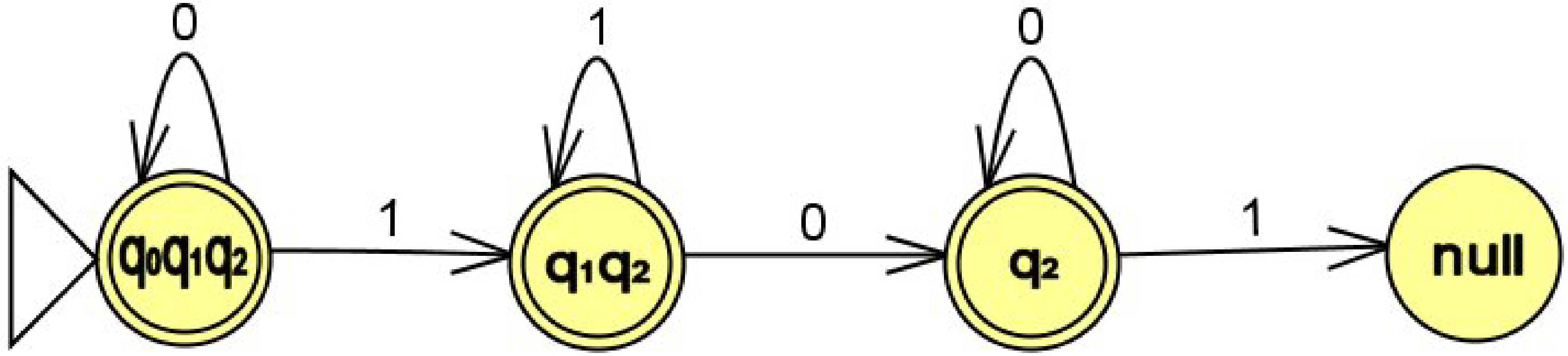
  s' = e-closure({s}) olsun ve

  $\delta(\{p_1,\ldots, p_m\}, \sigma)$ = e-closure($\delta^*(p_1, \sigma)$) $\cup \ldots \cup$ e-closure($\delta^*(p_m, \sigma)$)

Örnek



DFA = ?



- Theorem:
  - (a) Her regüler ifade için eşdeğer bir NFA vardır.
  - (b) Her DFA için eşdeğer bir regüler ifade vardır.