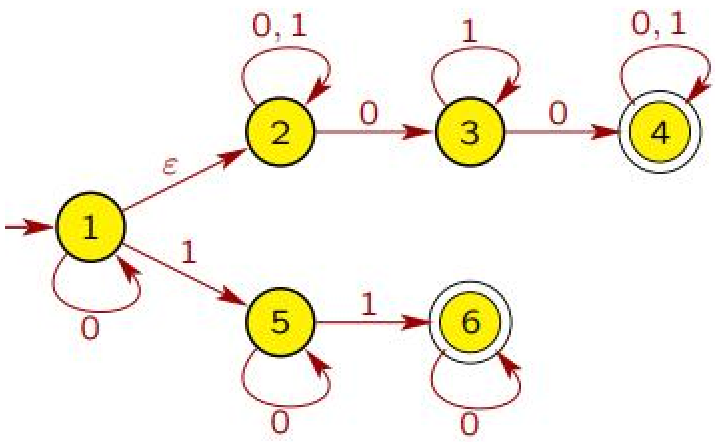
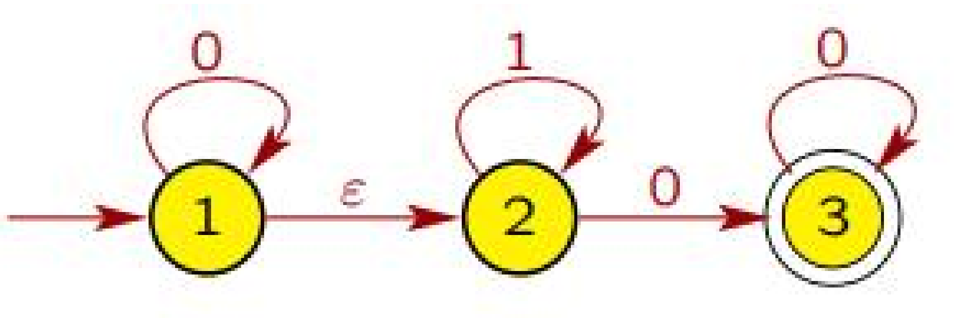
Exercise: L= { w $\in \Sigma^*$ | w en az iki sıfır içerir veya tam olarak iki bir içerir }

Exercise: L= { w ∈ Σ* | w en az iki sıfır içerir veya tam olarak iki bir içerir }



Exercise: $L = \{w \in \Sigma^* \mid w = 0^* 1^* 0^* 0\}$
üç durumla elde ediniz.

Exercise: $L = \{w \in \Sigma^* \mid w = 0^* 1^* 0^* 0\}$
üç durumla elde ediniz.



PAMUKKALE ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
2021 BAHAR

Biçimsel Diller ve Otomata Teorisi
Formal languages and automata theory

NFA to DFA Conversion

Nondeterministic/Deterministic Finite Automata

- Her NFA için bir DFA eşiti olduğu ispatlanmıştır.
- $M = (K, \Sigma, \Delta, s, F)$ bir NFA ve

  $M' = (K', \Sigma, \delta', s', F')$ *DFA* eşiti olsun.
- M toplam $\{q_0, q_1, q_2, q_3, q_4\}$ olmak üzere 5 duruma sahip olsun.
- M bir durumdayken, okunan bir string için $\{q_0, q_2, q_3\}$ durumlarında olabiliyorsa,

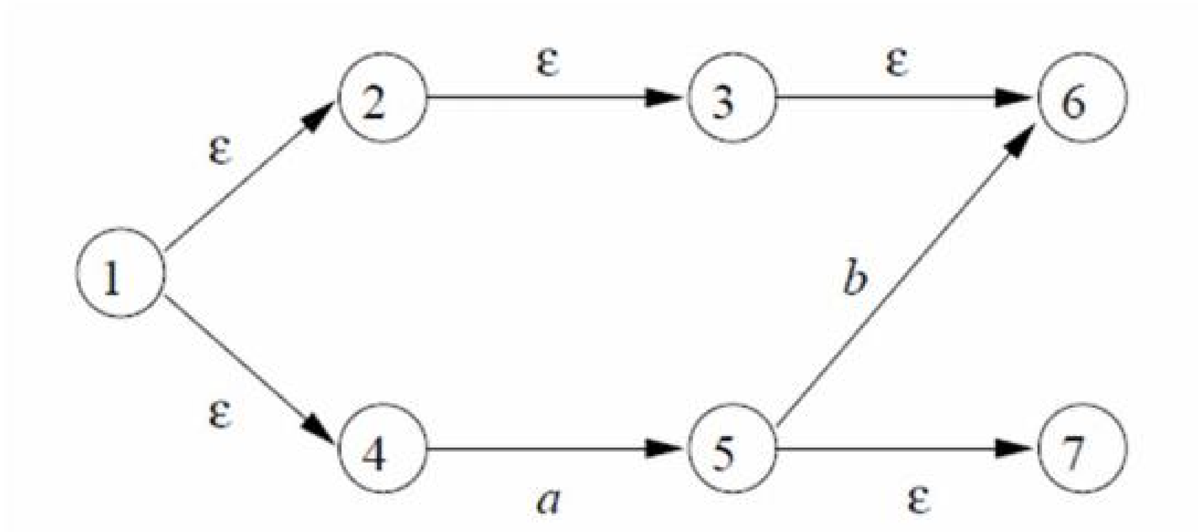
  M' için tek bir durum olarak $\{q_0, q_2, q_3\}$ kümesi alınır.
- NFA'da $\{q_0, q_2, q_3\}$ durumlarından bazılarına ***e-transition*** ile geçilebilir.
- M ve M' automat'larının eşit olabilmesi icin,

$w \in \Sigma^*$ ve $(s, w) \vdash_M^* (f, e)$, $f \in F$ icin

$$(E(s), w) \vdash_M^* (Q, e) , \quad E(s)\text{: epsilon closure}$$

öyleki Q kümesinin en az bir elemanı icin $f \in F$ olmak zorundadır.

Epsilon Closure



E (1) = {1,2,3,4,6}

E (2) = {2,3,6}

E (3) = {3,6}

E (4) = {4}

E (5) = {5,7}

E (6) = {6}

Nondeterministic/Deterministic Finite Automata

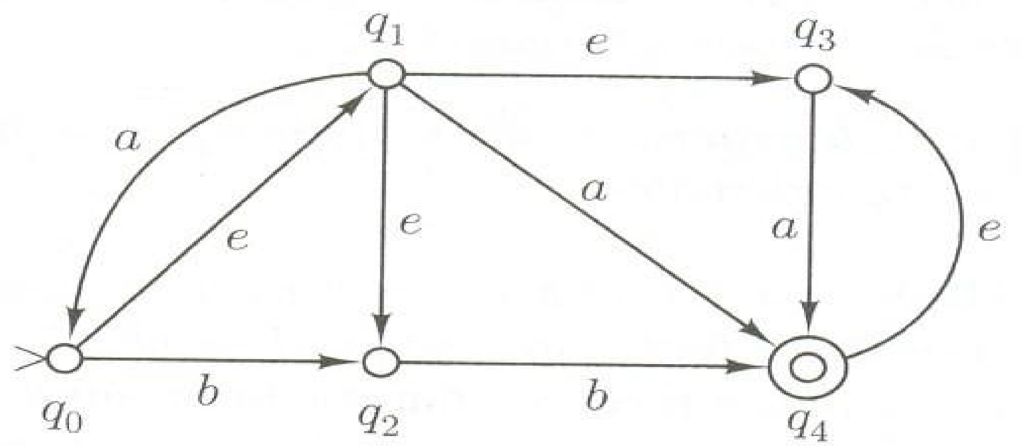
- M $\{q_0, q_2, q_3\}$ durumlarında iken girilen bir sembol q_0'ı q_1 veya q_2'ye, q_2'yi q_0'a ve q_3'ü q_2'ye götürüyorsa bir sonraki durum $\{q_0, q_1, q_2\}$ kümesi olarak alınır.
- Bu şekilde oluşturulabilecek DFA *M'* için en fazla $K'=2^K$ olacaktır. *K* kümesinin power kümesinin tüm elemanları kullanılmayabilir.
- *M' DFA'sı* için final states kümesi *F'*, *M* için tanımlanmış *K* kümesinin altkümelerinden, içerisinde en az bir tane final state bulunanlardan oluşur.
- *M'* için transition function *e-transition'* ları da içine alan kümeyle ifade edilir.

- Bir *q* durumu icin *e-transition* aşağıdaki gibi ifade edilir;

$E(q) = \{p \in K : (q, e) \vdash_M^* (p, e)\}$ $\forall q \in K$ olmak üzere

Nondeterministic/Deterministic Finite Automata

Örnek:



- Yukarıdaki NFA için $E(q_0)= \{q_0, q_1, q_2, q_3\}$, $E(q_1)= \{q_1, q_2, q_3\}$ ve $E(q_2)=\{q_2\}$ olarak bulunur.

- $M' = (K', \sum, \delta', s', F')$ *DFA* eşitinin tanımı aşağıdaki şekilde yapılır;

  $K' \subseteq 2^K \quad s' = E(s)$

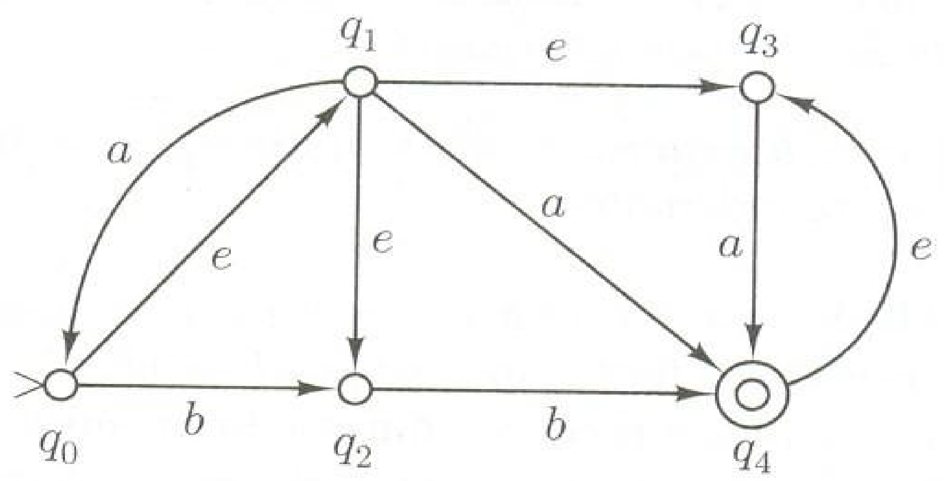
  $F'=\{Q \subseteq K : Q \cap F \neq \emptyset \}$

  ve her $Q \subseteq K$ *için ve her* $a \in \sum$ *için*

  $\delta'(Q, a)= \bigcup \{E(p) : p \in K \text{ ve } (q, a, p) \in \Delta \text{ bazı } q \in Q \text{ için}\}$

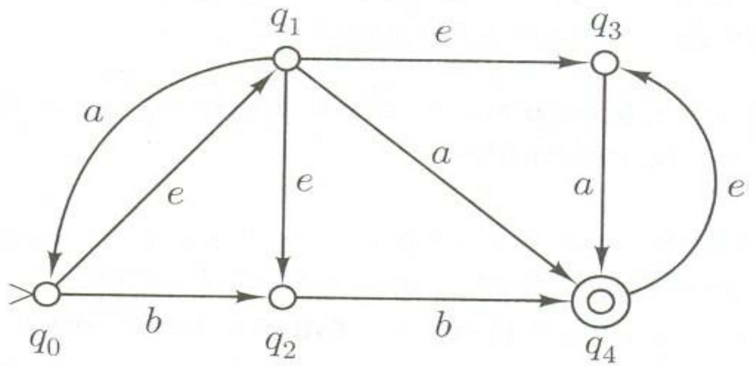
Nondeterministic/Deterministic Finite Automata

Örnek: (Devam)



- $\delta'(Q, a)$ geçişi, *a* girişi için gidilen durumlarn ve bu durumlarda *e- transition* 'larla gidilen durumlara gecişlerin tümünü ifade eder.
- $s' = \{q_0, q_1, q_2, q_3\}$ olarak elde edilir.
- q_1 durumundayken *a* girişi icin q_0 veya q_4'e geçilebilir. Böylece $\delta'(q_1, a) = E(q_0) \cup E(q_4) = \{q_0, q_1, q_2, q_3, q_4\}$ olur.

Nondeterministic/Deterministic Finite Automata



Örnek: (Devam)

- *M,* 5 duruma sahiptir böylece

  M' en fazla $2^5 = 32$ duruma sahip olur.

- *32* durumdan sadece herhangi bir girişle *s'* durumundan

  ulaşılabilenler (reachable states) alınır.

- $s' = E(q_0) = \{q_0, q_1, q_2, q_3\}$

  $q \in s'$ için (q, a, p) şeklinde (q_1, a, q_0), (q_1, a, q_4) ve (q_3, a, q_4) geçişleri tanımlanır.

  Böylece $\delta'(s', a) = E(q_0) \cup E(q_4) = \{q_0, q_1, q_2, q_3, q_4\}$ *olur.*

- *$q \in s'$ için (q, b, p) şeklinde (q_0, b, q_2) ve (q_2, b, q_4) geçişleri tanımlanır.*

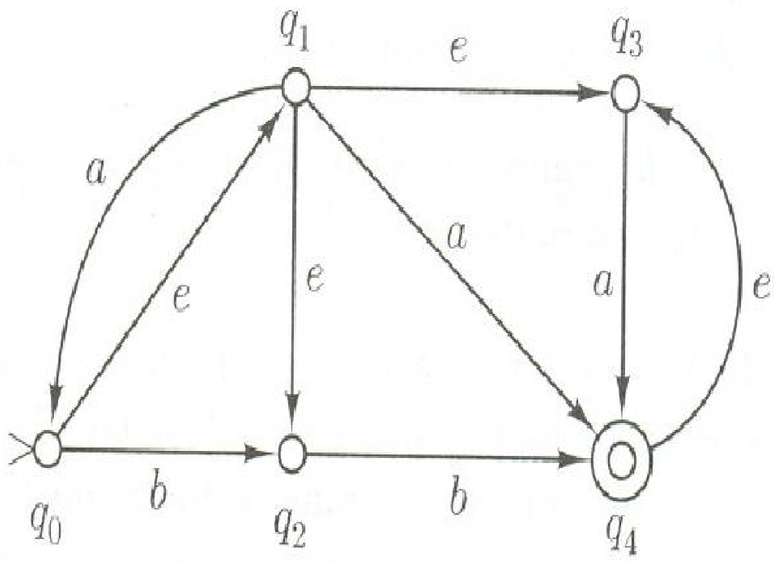
  Böylece $\delta'(s', b) = E(q_2) \cup E(q_4) = \{q_2, q_3, q_4\}$ *olur.*

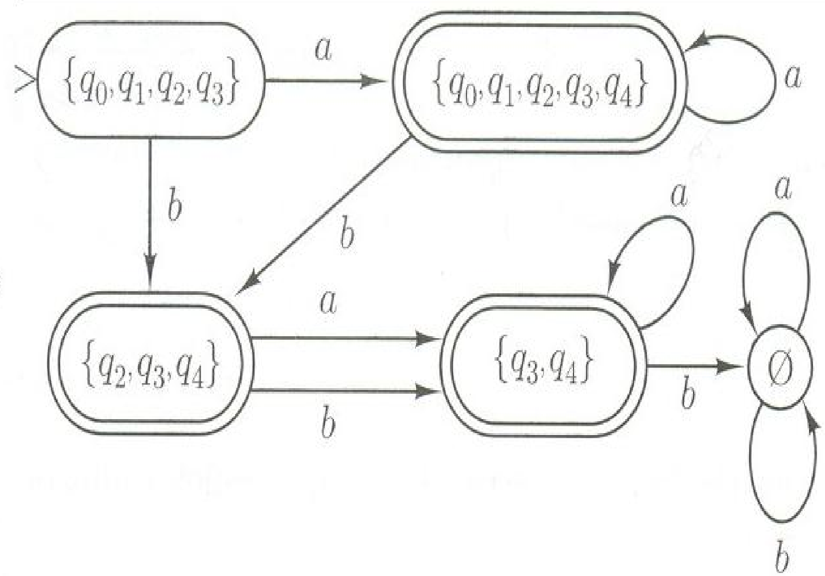
- *Aynı işlemler Ø elde edilinceye kadar yeni elde edilen durumlar için tekrar edilir.*

Nondeterministic/Deterministic Finite Automata

Örnek: (Devam)

- $\delta'(\{q_0, q_1, q_2, q_3, q_4\}, a) = \{q_0, q_1, q_2, q_3, q_4\}$ *kendisi*
- $\delta'(\{q_0, q_1, q_2, q_3, q_4\}, b) = \{q_2, q_3, q_4\}$ *sonraki durum*
- $\delta'(\{q_2, q_3, q_4\}, a) = E(q_4) = \{q_3, q_4\}$ *sonraki durum*
- $\delta'(\{q_2, q_3, q_4\}, b) = E(q_4) = \{q_3, q_4\}$ *sonraki durum*
- $\delta'(\{q_3, q_4\}, a) = E(q_4) = \{q_3, q_4\}$ *kendisi*
- $\delta'(\{q_3, q_4\}, b) = 0$ *sonraki durum*
- $\delta'(0, a) = \delta'(0, b) = 0$ *kendisi*





Nondeterministic/Deterministic Finite Automata

Örnek: (Devam)

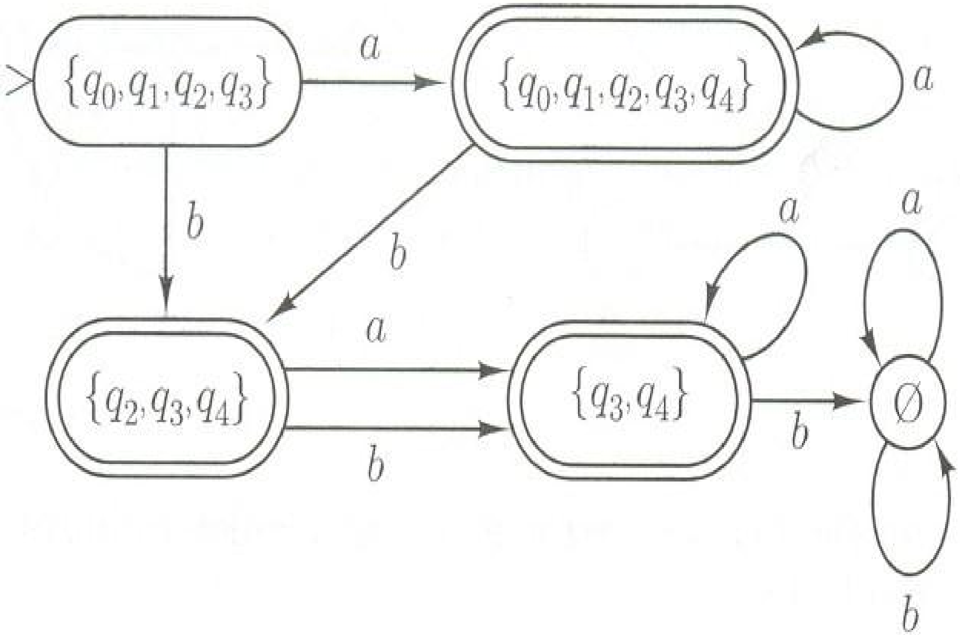
- $K' = \{q_a, q_b, q_c, q_d, q_e\}$

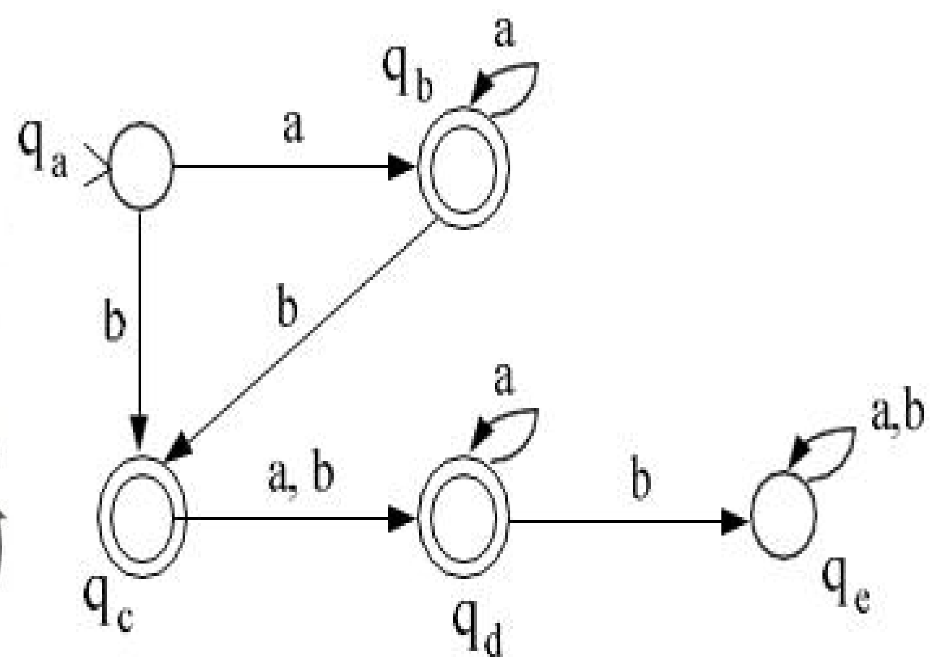
  $q_a = \{q_0, q_1, q_2, q_3\}$, $q_b = \{q_0, q_1, q_2, q_3, q_4\}$,

  $q_c = \{q_2, q_3, q_4\}$, $q_d = \{q_3, q_4\}$, $q_e = 0$

- $s' = q_a$, $F' = \{q_b, q_c, q_d\}$

- $\delta' = \{(q_a, a, q_b), (q_a, b, q_c), (q_b, a, q_b), (q_b, b, q_c), (q_c, a, q_d), (q_c, b, q_d), (q_d, a, q_d), (q_d, b, q_e), (q_e, a, q_e), (q_e, b, q_e)\}$





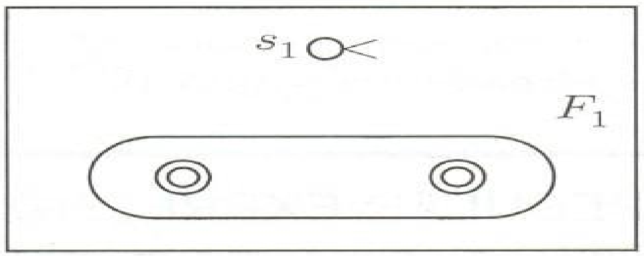
Finite Automata and Regular Expressions

- Finite automata tarafından kabul edilen diller sınıfı aşağıdaki özelliklere sahiptir;

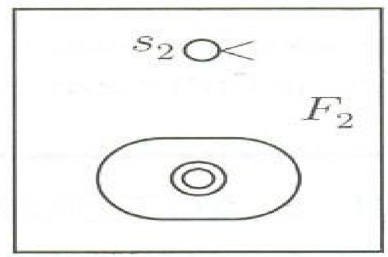
  - Union
  - Concatenation
  - Kleene star
  - Complementation
  - Intersection

Finite Automata and Regular Expressions

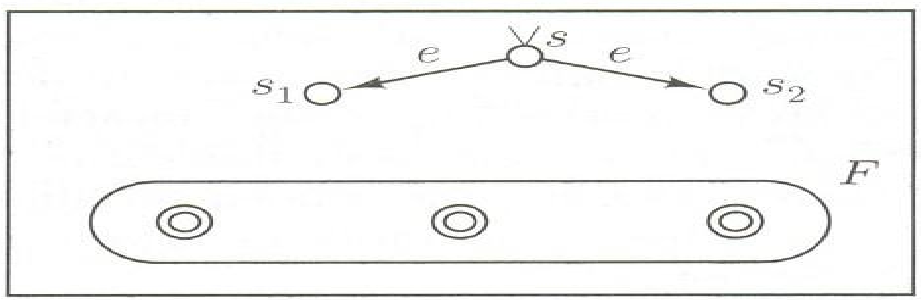
- Union
  - $M_1 = (K_1, \textstyle\sum, \Delta_1, s_1, F_1)$ ve $M_2 = (K_2, \textstyle\sum, \Delta_2, s_2, F_2)$ ***NFA olsun***.
  - $L(M) = L(M_1) \cup L(M_2)$ *olacak şekilde yeni bir automata M tanımlanabilir.*



M_1



M_2

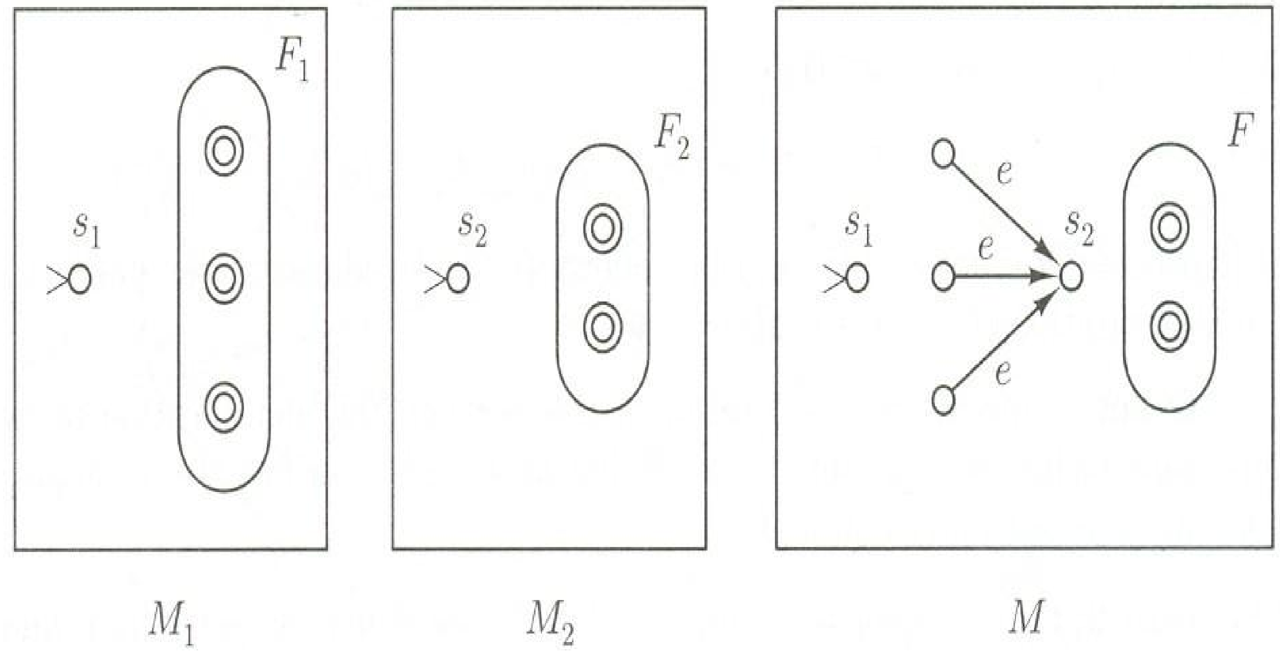


M

M otomatı, M_1 ve M_2 arasında başlangıçta nondeterministic (e-transition) geçiş yapar.

Finite Automata and Regular Expressions

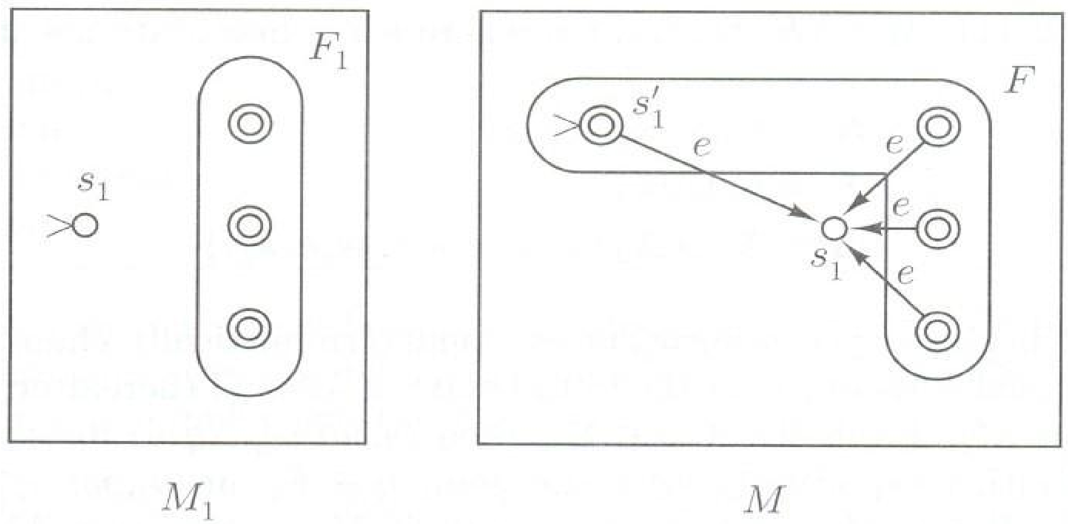
- Concatenation
  - $M_1 = (K_1, \Sigma, \Delta_1, s_1, F_1)$ ve $M_2 = (K_2, \Sigma, \Delta_2, s_2, F_2)$ **NFA olsun**.
  - $L(M) = L(M_1) \circ L(M_2)$ olacak şekilde yeni bir automata M tanımlanabilir.



M_1 sonlanınca nondeterministic olarak (e-transition) M_2 'ye geçiş yapar.

Finite Automata and Regular Expressions

- Kleene star
  - $M_1 = (K_1, \sum, \Delta_1, s_1, F_1)$ ***NFA olsun.***
  - $L(M) = L(M_1)^*$ *olacak şekilde yeni bir automata M tanımlanabilir.*



M_1 sonlanınca nondeterministic (e- transition) olarak başlangıç durumuna geçiş yapar. Yeni başlangıç durumu aynı zamanda bitiş durumudur.

Finite Automata and Regular Expressions

- Complementation

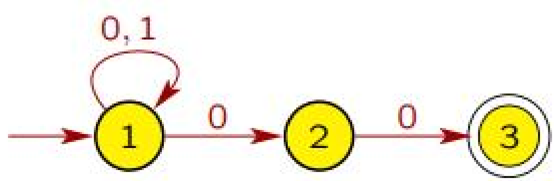
  - $M = (K, \Sigma, \delta, s, F)$ bir **DFA olsun.**

  - $\overline{L} = \Sigma^* - L(M)$ olacak şekilde yeni bir automata tanımlanabilir.

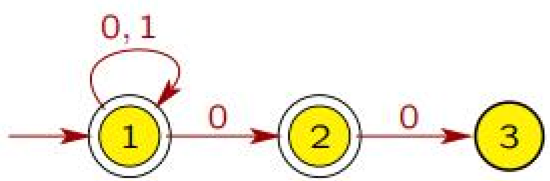
  - $\overline{M} = (K, \Sigma, \Delta, s, K - F)$ olacak şekilde yeni bir automata $\overline{M}$ tanımlanabilir.

  - ***(NFA için geçerli değil!) : NFA için de complementation elde edilebilir. Fakat bu şekilde bir yöntem geçerli değildir.***

- Aşağıdaki M1 NFA'sı $C = \{ w \in \Sigma^* \mid w$ 00 ile biter$\}$, dilini tanır.



M1'in kabul edilen ve kabul edilmeyen durumlarının değiştirilmesi, aşağıdaki M2 NFA'sını verir.



M2 için $100 \notin \overline{C}$ = { w | w 00 ile bitmez}, bu nedenle M2 , $\overline{C}$ dilini tanımıyor.

Finite Automata and Regular Expressions

- Intersection
  - $L_1 \cap L_2 = \Sigma^* - ((\Sigma^* - L_1) \cup (\Sigma^* - L_2))$

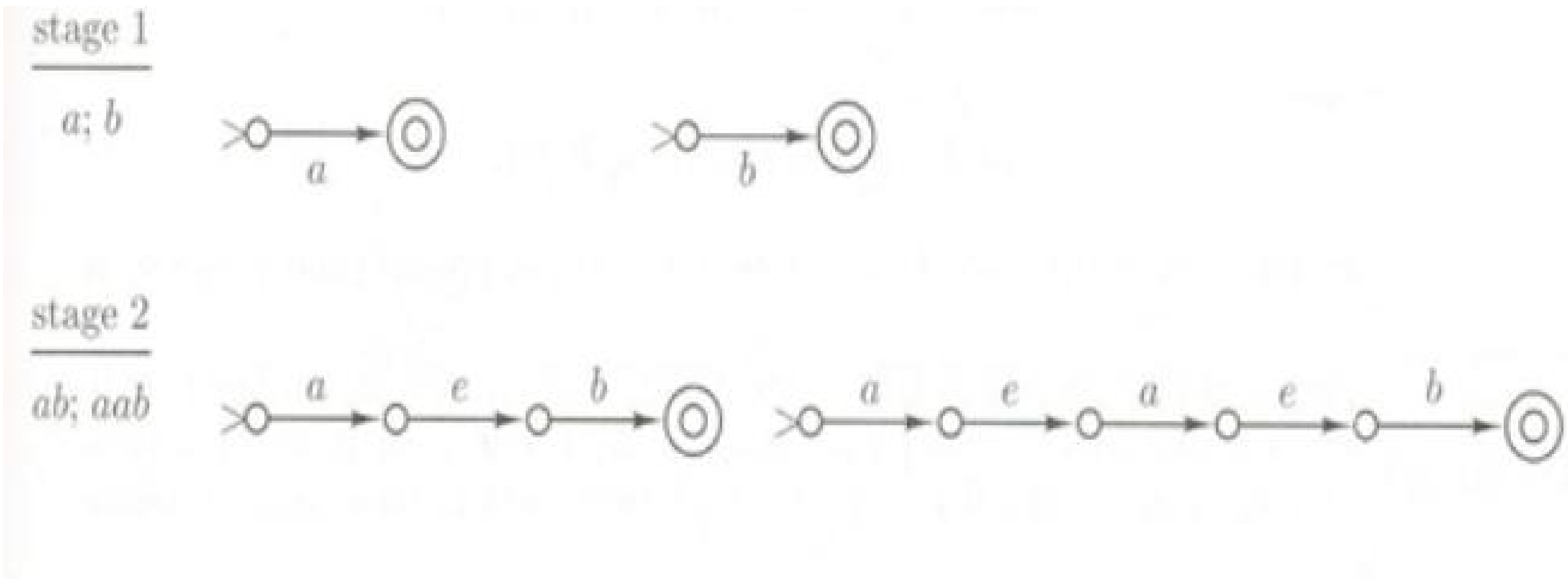
Finite Automata and Regular Expressions

Örnek:(ab ∪ aab) regular expression tarafından tanımlanan dili kabul eden NFA'yı (e-NFA) oluşturunuz (pp.79).*

Finite Automata and Regular Expressions

Örnek:(ab ∪ aab) regular expression tarafından*

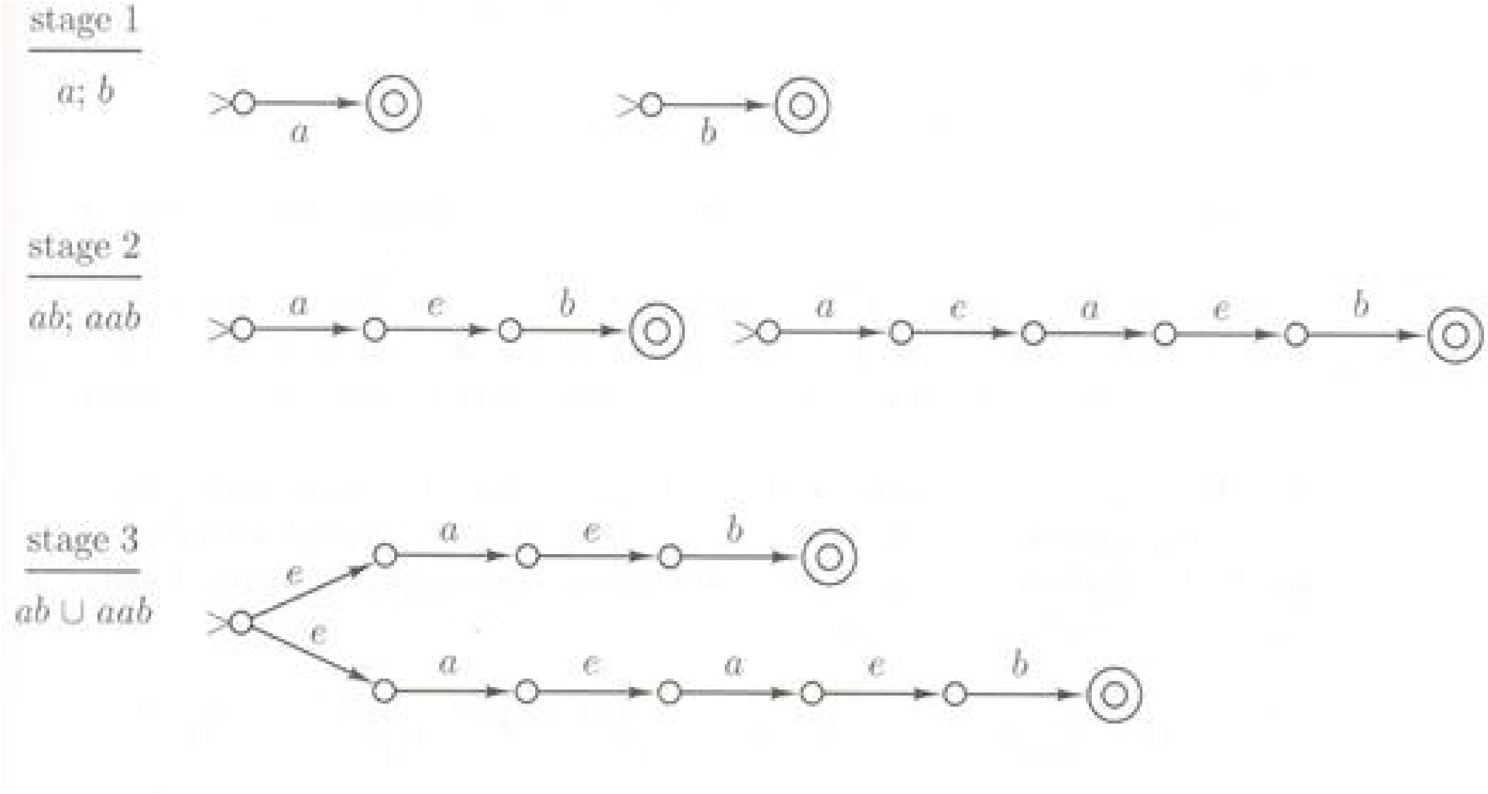
tanımlanan dili kabul eden NFA'yı oluşturunuz.



Finite Automata and Regular Expressions

Örnek:(ab ∪ aab) regular expression tarafından*

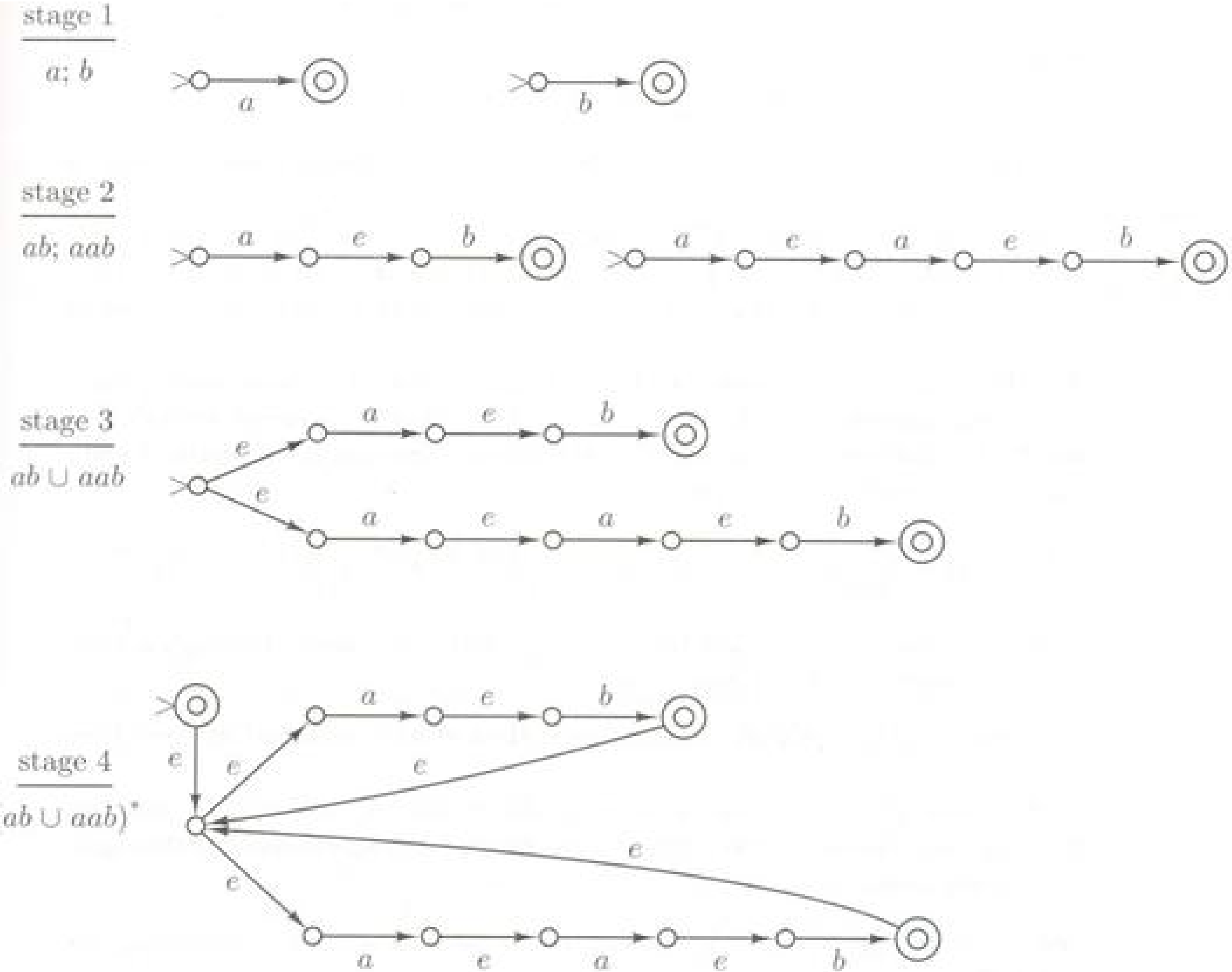
tanımlanan dili kabul eden NFA'yı oluşturunuz.



Finite Automata and Regular Expressions

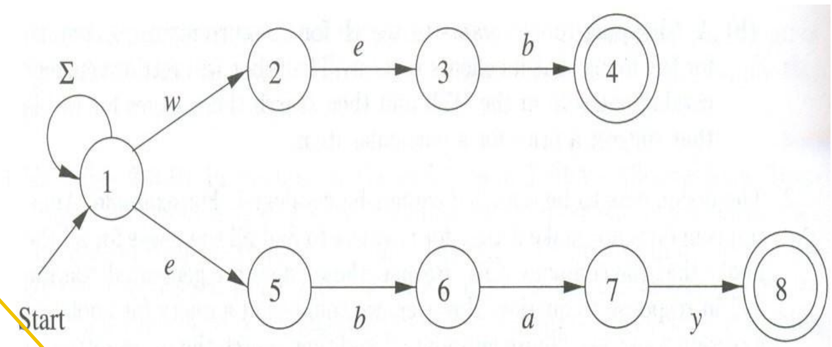
Örnek:(ab ∪ aab) regular expression tarafından*

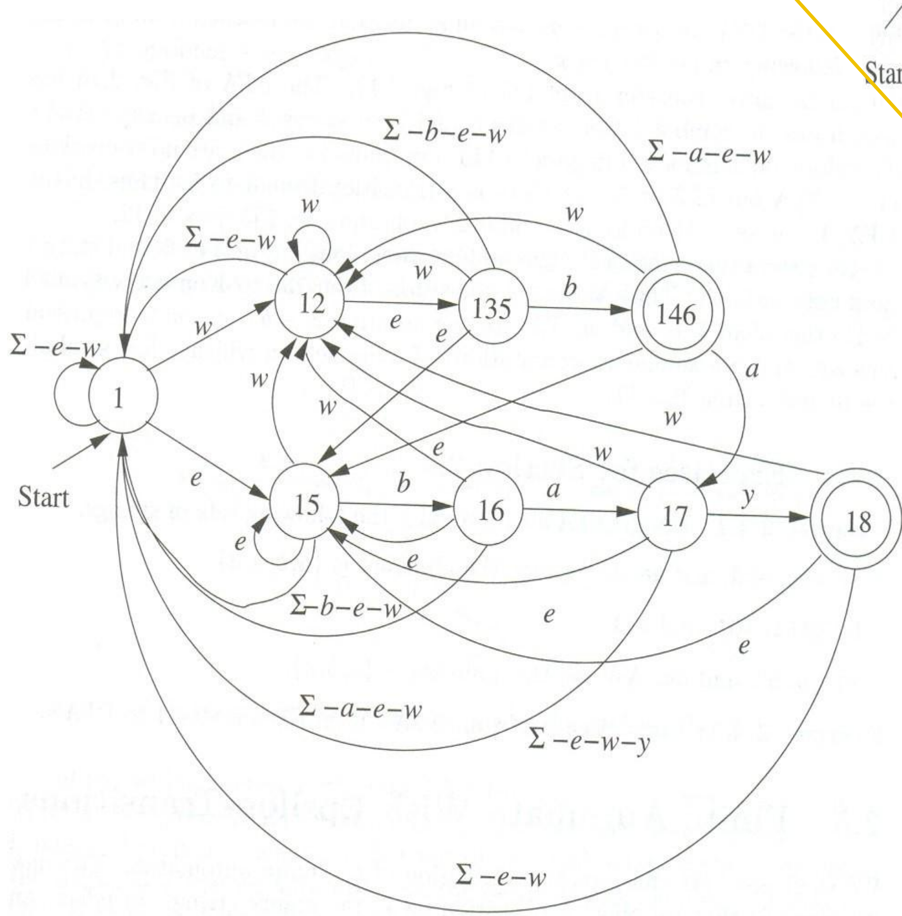
tanımlanan dili kabul eden NFA'yı oluşturunuz.



Nondeterministic/Deterministic Finite Automata

Örnek:





Bir metin içerisinde *web* *ve* *ebay* *kelimelerini arayan nondeterministic ve deterministic otomatlar*

Ödev

- Problemleri çözünüz 2.2.9 (sayfa 75)
- Problem 2.2.6, 2.2.7 ve 2.2.8' de bulunan NFA'lara eşit DFA'ları bulunuz (sayfa 74-75)
- Problemleri çözünüz 2.3.4, 2.3.7 (sayfa 83-84)