

BM 305 Biçimsel Diller ve Otomatlar (Formal Languages and Automata)

Hazırlayan: M.Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü



Nondeterministic/Deterministic Finite Automata

- Her nondeterministic finite automata için bir deterministic finite automata eşiti vardır.
- $M = (K, \Sigma, \Delta, s, F)$ bir nondeterministic automata ve $M' = (K', \Sigma, \delta', s', F')$ deterministic eşiti olsun.
- M toplam $\{q_0, q_1, q_2, q_3, q_4\}$ olmak üzere 5 duruma sahip olsun.
- M herhangi bir durumdayken girilen 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.
- Nondeterministic automata'da $\{q_0, q_2, q_3\}$ durumlarından bazılarına *e-transition* ile geçilebilir.
- M ve M' automata'larının eşit olabilmesi için,
 $w \in \Sigma^*$ ve $(s, w) \vdash_M^* (f, e)$, $f \in F$ için
 $(E(s), w) \vdash_{M'}^* (Q, e)$, Q' nun en az bir elemanı için $f \in F$ olmak zorundadır.

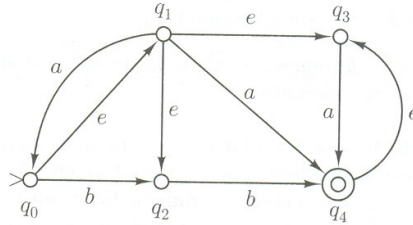
Nondeterministic/Deterministic Finite Automata

- Girilen bir sonraki 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.
- Oluşturulan deterministic automata M' için $K'=2^K$ olacaktır. K kümesinin power kümesinin tüm elemanları kullanılmayabilir.
- M' deterministic automata 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' larıda içine alan kümeyle ifade edilir. Bir q durumu için e -transition aşağıdaki gibi ifade edilir;

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

Nondeterministic/Deterministic Finite Automata

Örnek:

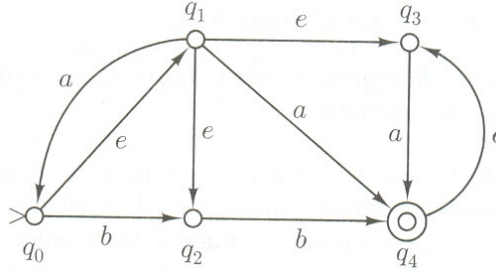


- Yukarıdaki nondeterministic automata için $E(q_0) = \{q_0, q_1, q_2, q_3\}$, $E(q_1) = \{q_1, q_2, q_3\}$ ve $E(q_2) = \{q_2\}$ olarak bulunur.
- $M' = (K', \Sigma, \delta', s', F')$ deterministic automata eşitinin tanımı aşağıdaki şekilde yapılır;
 $K' = 2^K$
 $s' = E(s)$
 $F' = \{Q \subseteq K : Q \cap F \neq \emptyset\}$
ve her $Q \subseteq K$ için ve her $a \in \Sigma$ 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)

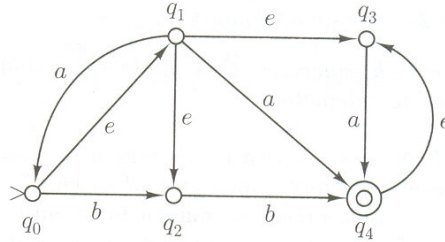


- $\delta'(Q, a)$ geçişi, a girişi için gidilen durumları ve bu durumlarda e -transition'larla gidilen durumlara geçişlerin tümünü ifade eder.
- $s' = \{q_0, q_1, q_2, q_3\}$ olarak elde edilir.
- q_1 durumundayken a girişi için q_0 veya q_4 'e geçilebilir. Böylece $\delta'(q_1, a) = E(q_0) \cup E(q_4)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

Nondeterministic/Deterministic Finite Automata

Örnek: (Devam)

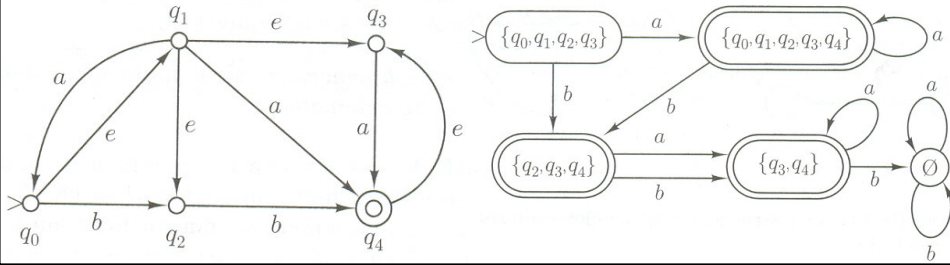
- M , 5 duruma sahiptir böylece $M' = 2^5 = 32$ duruma sahip olur.
- 32 durumdan sadece herhangi bir girişle s' durumundan ulaşılabilenler (reachable) alınır.
- $s' = E(q_0) = \{q_0, q_1, q_2, q_3\}$
 $q \in s'$ için (q, a, p) şeklinde (q_1, a, q_0) , (q_1, a, q_4) ve (q_3, a, q_4) geçişleri tanımlanır. 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 \emptyset elde edilinceye kadar yeni elde edilen durumlar için tekrar edilir.



Nondeterministic/Deterministic Finite Automata

Örnek: (Devam)

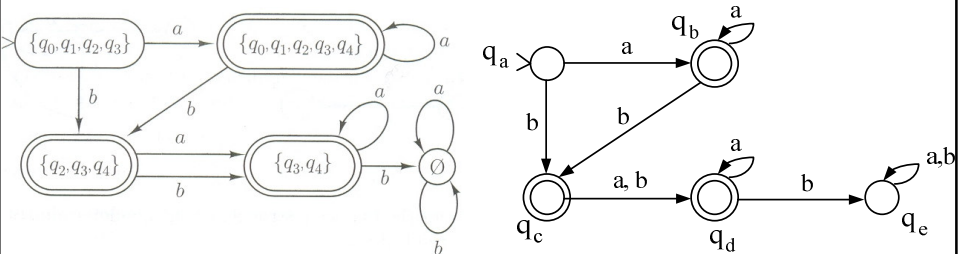
- $\delta'(\{q_0, q_1, q_2, q_3, q_4\}, a) = \{q_0, q_1, q_2, q_3, q_4\}$ kendisi
- $\delta'(\{q_0, q_1, q_2, q_3, q_4\}, b) = \{q_2, q_3, q_4\}$ sonraki durum
- $\delta'(\{q_2, q_3, q_4\}, a) = E(q_4) = \{q_3, q_4\}$ sonraki durum
- $\delta'(\{q_2, q_3, q_4\}, b) = E(q_4) = \{q_3, q_4\}$ sonraki durum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



Nondeterministic/Deterministic Finite Automata

Örnek: (Devam)

- $K' = \{q_a, q_b, q_c, q_d, q_e\}$
- $q_a = \{q_0, q_1, q_2, q_3\}, \quad q_b = \{q_0, q_1, q_2, q_3, q_4\},$
- $q_c = \{q_2, q_3, q_4\}, \quad q_d = \{q_3, q_4\}, \quad q_e = \emptyset$
- $s' = q_a, \quad F' = \{q_b, q_c, q_d\}$
- $\delta' = \{(q_a, a, q_b), (q_a, b, q_c), (q_b, a, q_b), (q_b, b, q_c), (q_c, a, q_d), (q_c, b, q_d),$
 $(q_d, a, q_d), (q_d, b, q_e), (q_e, a, q_e), (q_e, b, q_e)\}$





Finite Automata and Regular Expressions

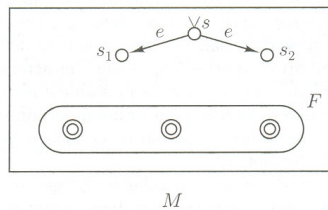
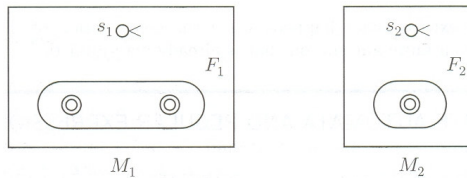
- Finite automata tarafından kabul edilen diller sınıfı aşağıdaki özelliklere sahiptir;
 - Union
 - Concatenation
 - Kleene star
 - Complementation
 - Intersection



Finite Automata and Regular Expressions

■ Union

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



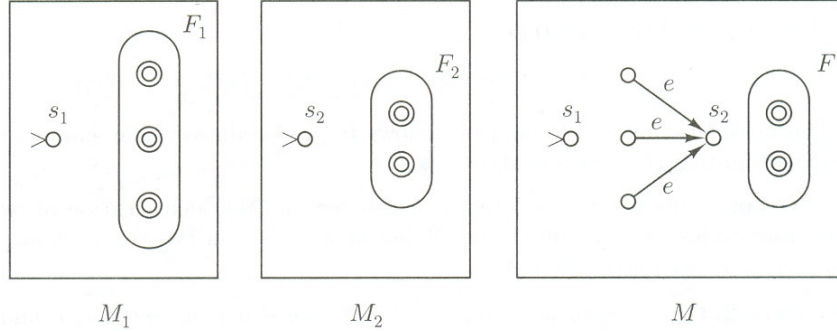
M, M₁ ve M₂ arasında başlangıçta nondeterministic (e-transition) geçiş yapar.



Finite Automata and Regular Expressions

Concatenation

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



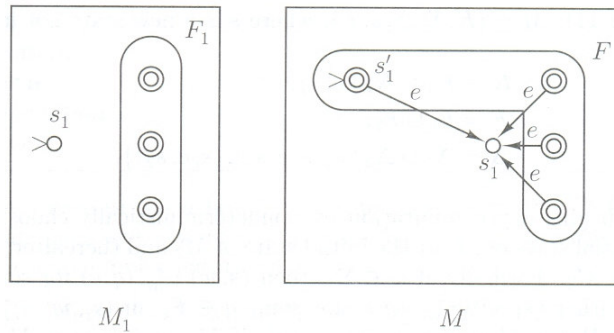
M_1 sonlanınca nondeterministic olarak (e -transition) M_2 'ye geçiş yapar.



Finite Automata and Regular Expressions

Kleene star

- $M_1 = (K_1, \Sigma, \Delta_1, s_1, F_1)$ nondeterministic automata 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

- $M = (K, \Sigma, \delta, s, F)$ deterministic automaton 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 M tanımlanabilir.

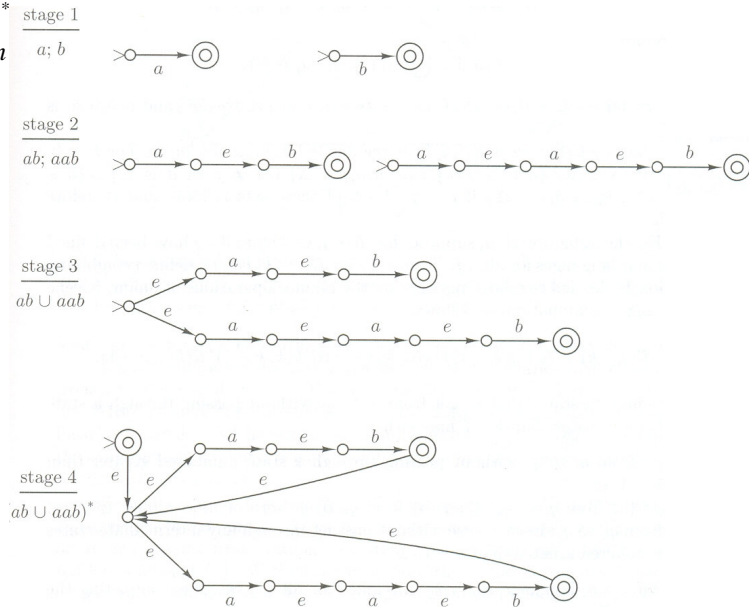
■ Intersection

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



Finite Automata and Regular Expressions

Örnek: $(ab \cup aab)^*$
regular expression
tarafından
tanımlanan dili
kabul eden
nondeterministic
automata'yı
oluşturunuz.





Finite Automata and Regular Expressions

- $M = (K, \Sigma, \Delta, s, F)$ bir automata olsun (deterministic veya nondeterministic olabilir).
- Bu automata için $L(R) = L(M)$ olacak şekilde bir regular expression R oluşturulabilir.
- $L(M)$ sonlu sayıda basit dillerin birleşimi olsun.
- $K = \{q_1, \dots, q_n\}$ ve $s = q_1$ olsun.
- $i, j = 1, \dots, n$ ve $k = 0, \dots, n$ için Σ^* üzerinde bir $R(i, j, k)$ regular expression tanımlanabilir. $R(i, j, k)$ string'leri M otomat'ını q_i durumundan q_j durumuna götürür.
- $R(i, j, n) = \{w \in \Sigma^* : (q_i, w) \vdash_M^* (q_j, e)\}$

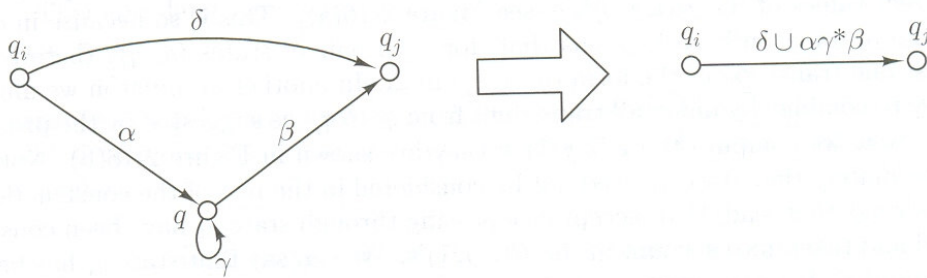
Kabul edilen dil ise aşağıdaki gibi tanımlanır;

$$L(M) = \bigcup \{R(i, j, n) : q_j \in F\}$$



Finite Automata and Regular Expressions

Örnek: İki durum arasındaki geçişin regular expression ile ifade edilmesi

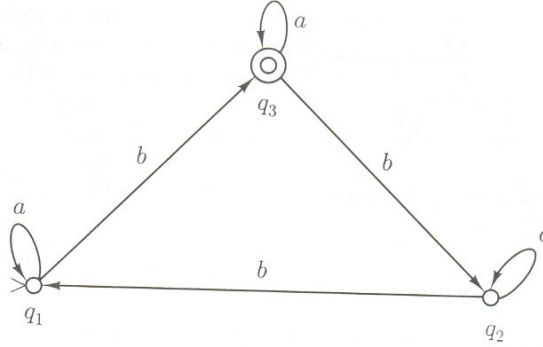


- İki durum arasındaki alternatif yollar \cup ile birleştirilir.
- Kendi kendisine dönem geçişler $*$ ile ifade edilir.
- Ardarda geçişler concatenation ile ifade edilir.

Finite Automata and Regular Expressions

Örnek: $L = \{w \in \{a, b\}^* : w \text{ içindeki } b \text{ sayısı } 3k+1 \text{ şeklinde olan tüm stringler}\}$

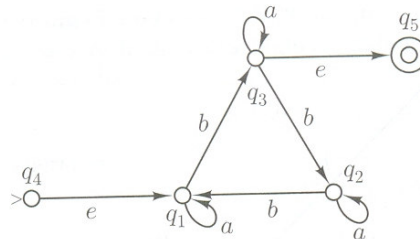
şeklinde tanımlanan dili kabul eden otomat için regular expression oluşturunuz.



Finite Automata and Regular Expressions

Örnek: (Devam)

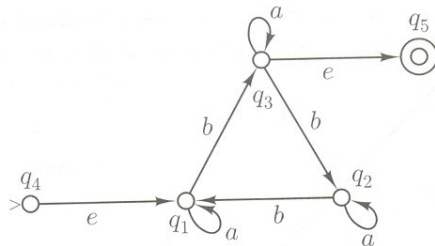
- Başlangıç ve bitiş durumlarının önüne sonuna ϵ -transition'larla geçişe sahip olan yeni başlangıç ve bitiş durumları eklenir.
- $s = q_{n-1}$ ve $f = q_n$ olarak belirlenir. Sonuçta elde edilecek regular expression $R(n-1, n, n)$ şeklinde ifade edilecektir.
- İlkönce $R(i, j, 0)$, sonra $R(i, j, 1)$ olacak şekilde tüm basit regular expression'lar belirlenir.
- Her aşamada bir state kaldırılır. ($R(i, j, 1)$ için q_1 , $R(i, j, 2)$ için q_2 , ..., $R(n-1, n, n-2)$)



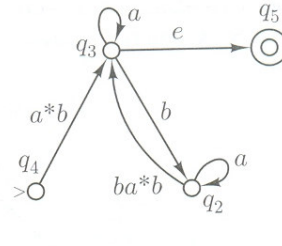
$R(i, j, 0)$

Finite Automata and Regular Expressions

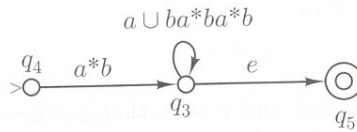
Örnek: (Devam)



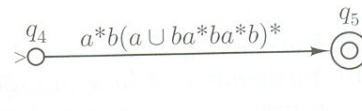
$R(i, j, 0)$



$R(i, j, 1)$



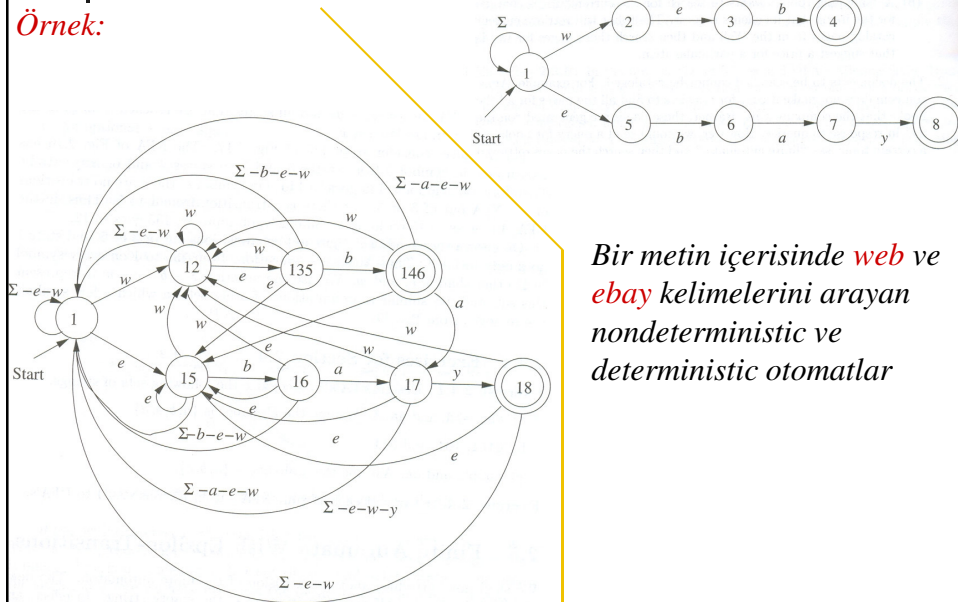
$R(i, j, 2)$



$R(i, j, 3)$

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 nondeterministic automata'lara eşit deterministic automata'ları bulunuz (sayfa 74-75)
- Problemleri çözünüz 2.3.4, 2.3.7 (sayfa 83-84)