BLG311 Biçimsel Diller ve Otomatlar

Otomatlar

A.Emre Harmancı   Tolga Ovatman   Berk Canberk

2017

İçerik

Tanımlar

Otomat

Otomatlar, bir giriş verisi üzerinde çeşitli durumlar ve konfigürasyonlardan geçerek işlemler yapan soyut makine modelleridir. İşlemenin her durumunda mevcut konfigürasyon göz önünde tutularak yeni durum bir durum geçiş fonksiyonu yardımıyla belirlenir. Sonuç olarak belirli bir konfigürasyona erişildiğinde giriş kabul edilmiş sayılır. En genel ve güçlü otomat Turing makinesidir.

Determinist Sonlu Otomat (Deterministic Finite Automata)

Bir DFA girdi olarak sunulan her sembol katarı için ayrı ve eşsiz(unique) bir işlem yaparak kabul/red durumuna karar verir.

Bir DFA'nın biçimsel tanımı

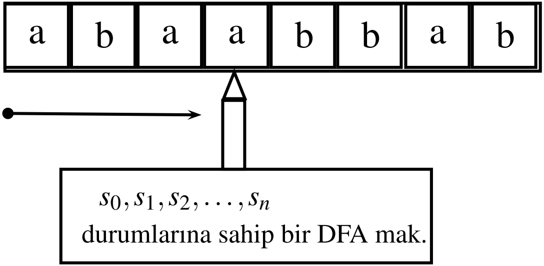
DFA'lar bir beşli vasıtasıyla tanımlanabilir $M = (\Sigma, S, s_0, \delta, F)$

- S: Sonlu ve boş olmayan bir durum kümesi $s \in S$.
- Σ: Giriş alfabesi (sonlu ve boş olmayan bir sembol kümesi)
- s_0: S'e dahil bir başlangıç durumu.
- δ: Durum geçiş fonksiyonu $\delta : S \times \Sigma \to S$
- F: Son durumlar kümesi $F \subseteq S$.

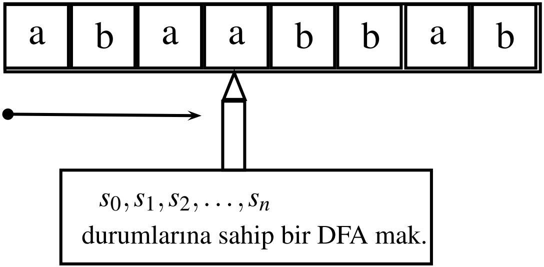
Bu makine her durumda, kabul/red koşullarına uygun olarak $Z = \{0, 1\}$ kümesi içinde çıktılar üreten bir Moore makinesidir.

DFA

Giriş şeridi, durumlar, sonlu denetim, okuma kafası, başlangıç durumu, kabul edilen ya da son durumlar, tanınan ya da kabul edilen dil. Şeridin tüm gözeleri soldan sağa sırayla okunduğunda, son göze okununca otomatın seçeceği durum kabul edilebilen durumlar içinde ise, şeritte yazılı simge katarı otomat tarafından kabul edilen dilin bir katarıdır.



DFA



DFA'nın çalışması durum geçiş fonksiyonunun ardı ardına işletilmesinden ibarettir. $\sigma \in \Sigma$ olarak verilen bir giriş sembolü okunduğunda $\delta(s, \sigma) = s' \in S$ çıktısı üretilir.

Konfigürasyon (Konuşlanım)

Her bir konfigürasyon makinenin çalışması sırasındaki tüm parametrelerin anlık değerlerini içerir.
$s, \omega \in S \times \Sigma^*$

Konfigürasyon üretimi $\vdash_M$ bağıntısı ile gerçekleştirilir. $\vdash_M$'deki çiftler (s, ω) ve (s', ω') olarak alındığında bağıntı aşağıdaki gibi tanımlanabilir:

a $\omega = \sigma\omega' \wedge \sigma \in \Sigma$

b $\delta(s, \sigma) = s'$

Bu şekilde tanımlanan bir geçiş *tek adımda türetme* olarak adlandırılır ve $(s,\omega) \vdash_M (s',\omega')$ şeklinde gösterilir.

- Türetilebilen konfigürasyon: $(s,\omega) \vdash_M^* (s',\omega')$ olarak tanımlanır. $\vdash_M^*$ bağıntısı $\vdash_M$'in yansımalı geçişli kapanışıdır.
- Tanınan kelime: $(s_0,\omega) \vdash_M^* (s_i,\Lambda)$ ve $s_i \in F$.
- Çalışma: $(s_0,\omega_0) \vdash (s_1,\omega_1) \vdash (s_2,\omega) \vdash \ldots \vdash (s_n,\Lambda)$.
- Tanınan dil: $L(M) = \{\omega \in \Sigma^* | (s_0,\omega) \vdash_M^* (s_i,\Lambda) \wedge s_i \in F\}$

Dil Tanıyıcı(Language Recognizer)

$\vdash_M$'in yansımalı geçişli kapanışı ${\vdash_M}^*$ ile ifade edilir. $(q,\omega){\vdash_M}^*(q',\omega')$ ifadesinde (q,ω) ile (q',ω')'ya sonlu sayıda adımda ulaşılır.

$(s,\omega){\vdash_M}^*(q,\Lambda)$ durumunda eğer $q \in F$ ise $\omega \in \Sigma^*$ otomat tarafından tanınmış kabul edilir. Bir başka deyişle
$L(M) = \{\omega \in \Sigma^* | (s,\omega){\vdash_M}^*(q_i,\Lambda) \wedge q_i \in F\}$

Örnek 1

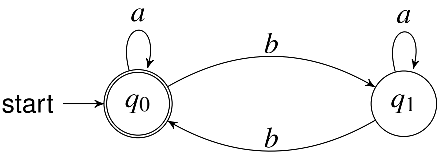
$S = \{q_0, q_1\}$

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

$s_0 = q_0$

$F = \{q_0\}$

q	σ	$\delta(q,\sigma)$
q_0	a	q_0
q_0	b	q_1
q_1	a	q_1
q_1	b	q_0



$$(q_0, aabba) \vdash_M (q_0, abba)$$
$$(q_0, abba) \vdash_M (q_0, bba)$$
$$(q_0, bba) \vdash_M (q_1, ba)$$
$$(q_1, ba) \vdash_M (q_0, a)$$
$$(q_0, a) \vdash_M (q_0, \Lambda)$$

Örnek 1

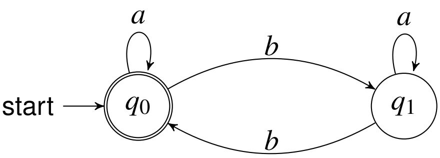
$S = \{q_0, q_1\}$

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

$s_0 = q_0$

$F = \{q_0\}$

q	σ	$\delta(q,\sigma)$
q_0	a	q_0
q_0	b	q_1
q_1	a	q_1
q_1	b	q_0



$L(M) = (a \vee ba^*b)^*$. İlgili gramer:

$V = S \cup \Sigma$

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

$s_0 = q_0 = n_0$

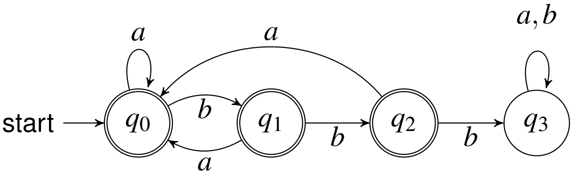
$< q_0 > ::= \Lambda | a < q_0 > | b < q_1 > | a$

$< q_1 > ::= b < q_0 > | a < q_1 > | b$

Örnek 2

$L(M) = \{\omega | \omega \in \{a,b\}^* \wedge \omega$ ardı ardına üç b içermemelidir$\}$
$S = \{q_0, q_1, q_2, q_3\}, \Sigma = \{a,b\}, s_0 = q_0, F = \{q_0, q_1, q_2\}$

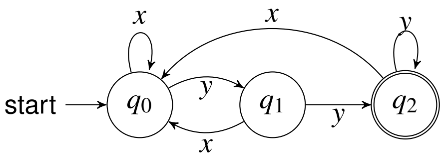
q	σ	$\delta(q,\sigma)$
q_0	a	q_0
q_0	b	q_1
q_1	a	q_0
q_1	b	q_2
q_2	a	q_0
q_2	b	q_3
q_3	a	q_3
q_3	b	q_3



q_3 isimli bir kuyu durumu bulunmaktadır.
$L(M) = [(\Lambda \vee b \vee bb)a]^*(\Lambda \vee b \vee bb)$

Örnek 3

$$S = \{q_0, q_1, q_2\}, \Sigma = \{x, y\}, s_0 = q_0 = n_0, F = \{q_2\}$$



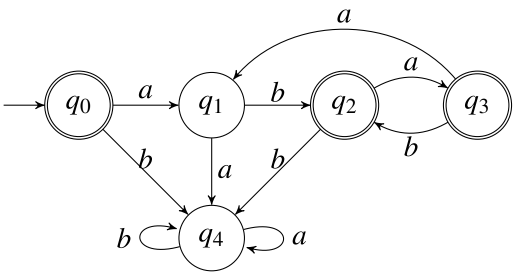
$$< q_0 >::= x < q_0 > | y < q_1 >$$
$$< q_1 >::= y < q_2 > | \ y \ | x < q_0 >$$
$$< q_2 >::= y \ | y < q_2 > | x < q_0 >$$

$$L(M) = ((x \vee yx)^* yy^+ x)^* (x \vee yx)^* yy^+$$
$$L(M) = ((\Lambda \vee y \vee yy^+)x)^* yy^+ = (y^* x)^* yy^+$$
$$L(M) = (x \vee yx \vee yy^+ x)^* yyy^*$$

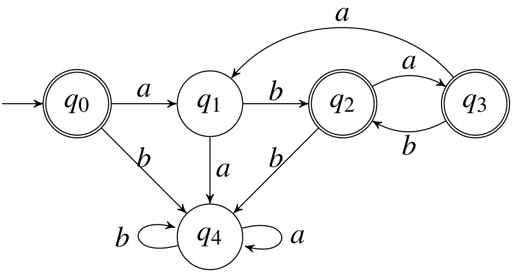
Determinist Olmayan Sonlu Otomatlar (Non-deterministic Finite Automata)

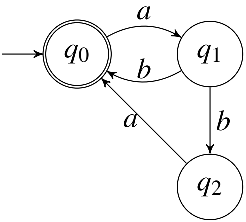
Bir NFA'da bir giriş sembolüne karşılık birçok olası sonraki duruma geçiş mümkündür.

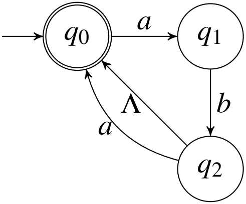
$L = (ab \vee aba)^*$ tanıyan bir DFA:

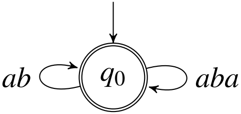


Aynı dili tanıyan üç farklı NFA oluşturulabilir.









NFA'ların biçimsel tanımı

DFA'lar bir beşli vasıtasıyla tanımlanabilir $M = (\Sigma, S, s_0, \Delta, F)$:

- S: Sonlu ve boş olmayan bir durum kümesi $s \in S$.
- Σ: Giriş alfabesi (sonlu ve boş olmayan bir sembol kümesi)
- s_0: S'e dahil bir başlangıç durumu.
- Δ: Durum geçiş bağıntısı$\Delta \subseteq S \times \Sigma^* \times S$
- F: Son durumlar kümesi $F \subseteq S$.

Bu tip otomatlarda konfigurasyon $S \times \Sigma^*$ kümesinden çiftler kullanılarak tanımlanır. Tek adımda türetme tanımı göz önüne alındığında:
$(q, \omega) \vdash_M (q', \omega') \Rightarrow \exists u \in \Sigma^* (\omega = u\omega' \wedge (q, u, q') \in \Delta)$
Deterministik otomatlar için $\Delta \subseteq S \times \Sigma^* \times S$ bağıntısı bir fonksiyona dönüşür $S \times \Sigma \to S$. (q, u, q') üçlüleri için
$|u| = 1 \wedge (\forall q \in S \wedge \forall u \in \Sigma) \exists! q' \in S$
NFA'nın tanıdığı dil $L(M) = \{\omega | (s, \omega) \vdash_m{}^* (q, \Lambda) \wedge q \in F\}$

Örnek bir NFA

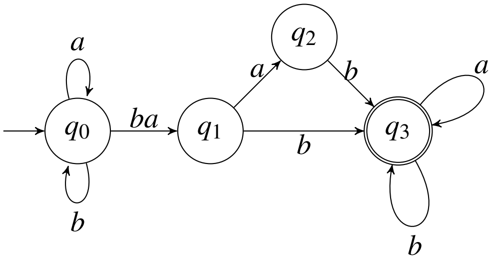
İçinde bab ve $baab$ katarları geçen katarları tanıyan bir NFA tanımlayın.

$S = \{q_0, q_1, q_2, q_3\}$
$\Sigma = \{a, b\}$
$s_0 = q_0$
$F = \{q_3\}$
$\Delta = \{(q_0, a, q_0), (q_0, b, q_0), (q_0, ba, q_1), (q_1, b, q_3), (q_1, a, q_2),$
$(q_2, b, q_3), (q_3, a, q_3), (q_3, b, q_3)\}$
$M = (S, \Sigma, \Delta, s_0, F)$

$< q_0 >::= a < q_0 > |b < q_0 > |ba < q_1 >$
$< q_1 >::= b < q_3 > | \ b \ |a < q_2 >$
$< q_2 >::= b < q_3 > |b$
$< q_3 >::= a \ | \ b \ | \ a < q_3 > \ | \ b < q_3 >$

Örnek bir NFA

$$< q_0 > ::= a < q_0 > | b < q_0 > | ba < q_1 >$$
$$< q_1 > ::= b < q_3 > | b | a < q_2 >$$
$$< q_2 > ::= b < q_3 > | b$$
$$< q_3 > ::= a | b | a < q_3 > | b < q_3 >$$



Olası bir çalışma:

$(q_0, aaabbbaabab) \vdash$
$(q_0, aabbbaabab) \vdash$
$(q_0, abbbaabab) \vdash$
$(q_0, bbbaabab) \vdash$
$(q_0, bbaabab) \vdash$
$(q_0, baabab) \vdash$
$(q_1, abab) \vdash (q_2, bab) \vdash$
$(q_3, ab) \vdash (q_3, b) \vdash$
(q_3, Λ)

Lemma

$M = (S, \Sigma, \Delta, s_0, F) \wedge q, r \in S \wedge x, y \in \Sigma^*$
$\exists p \in S \wedge (q, x) \vdash_M{}^* (p, \Lambda) \wedge (p, y) \vdash_M{}^* (r, \Lambda) \Rightarrow (q, xy) \vdash_M{}^* (r, \Lambda)$

Tanım

Düzenli Gramer: Tüm üretim kuralları Tip-3'tür.
Düzenli dil: Düzenli gramerlerle tanımlanan diller.
Düzenli ifade: $\varnothing, \{\Lambda\}, \{a | a \in \Sigma\}, A \vee B, A.B, A^*$
Düzenli küme: Bir düzenli ifade ile tanımlanabilen küme.

Düzenli gramerler NFA'larla temsil edilebilir.

Tanım

Düzenli Gramer: Tüm üretim kuralları Tip-3'tür.
Düzenli dil: Düzenli gramerlerle tanımlanan diller.
Düzenli ifade: $\varnothing, \{\Lambda\}, \{a|a \in \Sigma\}, A \vee B, A.B, A^*$
Düzenli küme: Bir düzenli ifade ile tanımlanabilen küme.

Düzenli gramerler NFA'larla temsil edilebilir.

a) Uç simge olmayan simgeler durumlara atanır
b) Başlangıç durumu başlangıç simgesine karşılık gelir
c) Son durumlar uç simgelerle biten kurallara karşılık gelir.
d) Λ tanınmalı ise başlangıç durumu son durum olarak tanımlanır.

Sonlu otomatlar tarafından tanınan diller(düzenli diller) birleşme, bitiştirme ve Kleene yıldızı işlemleri altında kapalıdır.

Kleene Teoremi

Her düzenli dil sonlu otomatlarla tanınabilir ve her sonlu otomat düzenli bir dil tanımlar.

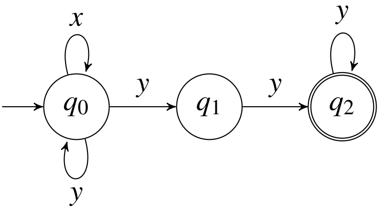
$M = (S, \Sigma, \Delta, s_0, F) \Leftrightarrow G = (N, \Sigma, n_0, \mapsto), L = L(G)$ Tip-3 gramer.

$S = N \wedge F \subseteq N$

$s_0 = n_0$

$\Delta = \{(A, \omega, B) : (A \mapsto \omega B) \in \mapsto \wedge (A, B \in N) \wedge \omega \in \Sigma^*\} \cup \{(A, \omega, f_i) : (A \mapsto \omega) \in \mapsto \wedge A \in N \wedge f_i \in F \wedge \omega \in \Sigma^*\}$

Örnek



$$< q_0 >::= x < q_0 > \mid y < q_0 > \mid y < q_1 >$$
$$< q_1 >::= y \mid y < q_2 >$$
$$< q_2 >::= y < q_2 > \mid y$$

Kleene Teoremi

Her NFA için eşdeğer bir DFA bulunabilir.

NFA $M = (S, \Sigma, \Delta, s_0, F)$ için amacımız...

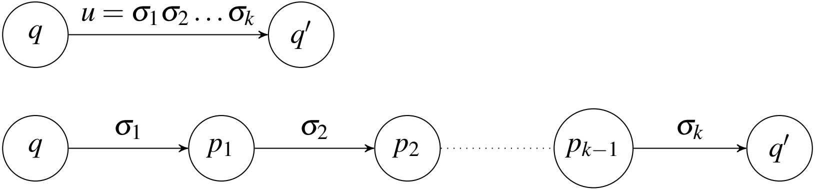
(a) $(q, u, q') \in \Delta$ içerisinde $u = \Lambda$ ve $|u| > 1$ bulunmamalı

(b) Her durumda her simge için bir geçiş olmalı

(c) Her konfigürasyon için birden fazla geçiş olmamalı.

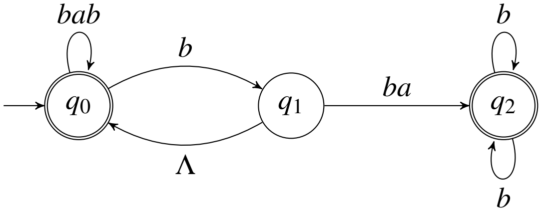
NFA/DFA eşdeğerliği-Faz 1

Δ'nın (q, u, q')'larındaki $|u| > 1$'i elemek için ara adımlar türetilir.

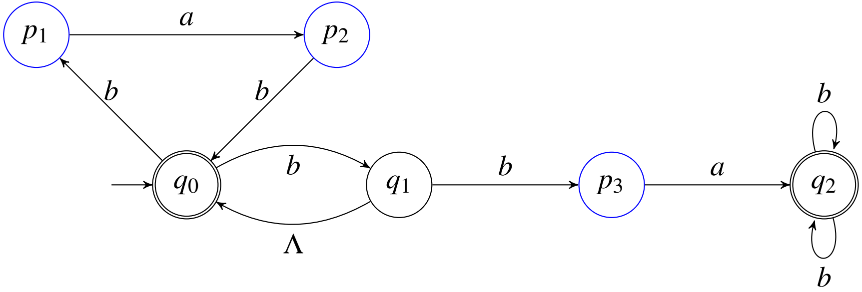


Bu açılım Δ'yı Δ' haline dönüştürür. Bu dönüşüm (q, u, q') üçlüleri $(q, \sigma_1, p_1), (p_1, \sigma_2, p_2), \dots, (p_{k-1}, \sigma_k, q')$ ile değiştirerek yapılır. Yeni makinede $M' = (S', \Sigma, \Delta', s'_0, F')$ $F' \equiv F$ ve $s'_0 \equiv s_0$

NFA/DFA eşdeğerliği-Faz 1



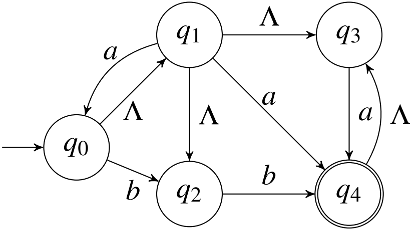
otomatının eşdeğeri:



NFA/DFA eşdeğerliği-Faz 2

Bir durumun erişilebilirlik kümesi

$R(q) = \{p \in S' | (q, \Lambda) \vdash_{M'}{}^{*} (p, \Lambda)\}$ or
$R(q) = \{p \in S' | (q, \omega) \vdash_{M'}{}^{*} (p, \omega)\}$

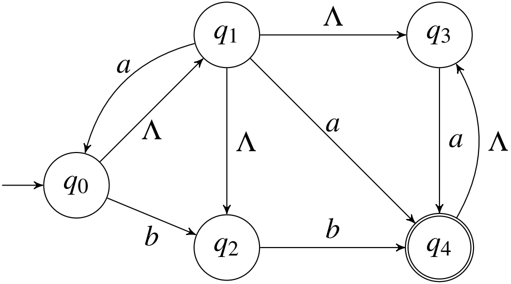


$R(q_0) = \{q_0, q_1, q_2, q_3\}$
$R(q_1) = \{q_1, q_2, q_3\}$
$R(q_2) = \{q_2\}$
$R(q_3) = \{q_3\}$
$R(q_4) = \{q_3, q_4\}$

NFA/DFA eşdeğerliği-Faz 2

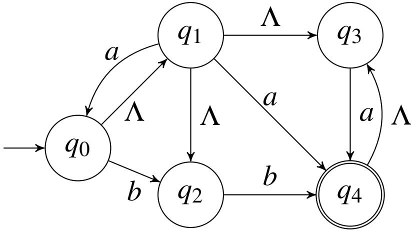
Eşdeğer bir determinist makine oluşturmak için: $M'' = (S'', \Sigma, \delta'', F'')$
$S'' \subseteq \mathscr{P}(S') = 2^{S'}$
$s_0'' = R(s_0')$ başlangıç durumundan Λ geçişleriyle ulaşılabilen durumlar
$F'' = \{Q \subseteq S' | Q \cap F' \neq \varnothing\}$

NFA/DFA eşdeğerliği-Faz 2



Eşdeğer bir determinist makine oluşturmak için δ'': $\forall Q \subseteq S' \wedge \forall \sigma \in \Sigma$
$\delta''(Q,\sigma) = \bigcup_P \{R(p) | \forall q \in Q \wedge \forall p \in S' \wedge \forall (q,\sigma,p) \in \Delta'\}$
Boş katar harici tüm üçlüleri yazalım:
a geçişleri: $(q_1,a,q_0),(q_1,a,q_4),(q_3,a,q_4)$,
b geçişleri: $(q_0,b,q_2),(q_2,b,q_4)$

NFA/DFA eşdeğerliği-Faz 2



Boş katar harici tüm üçlüleri yazalım:

a geçişleri: $(q_1, a, q_0), (q_1, a, q_4), (q_3, a, q_4),$

b geçişleri: $(q_0, b, q_2), (q_2, b, q_4)$

δ'' bu geçişlerle genişletilirse:

$s_0'' = R(q_0) = \{q_0, q_1, q_2, q_3\}(d_0)$

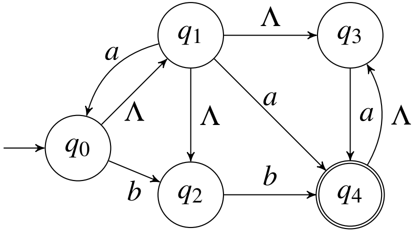
$\delta''(d_0, a) = R(q_0) \cup R(q_4) = \{q_0, q_1, q_2, q_3, q_4\}(d_1)$

$\delta''(d_0, b) = R(q_2) \cup R(q_4) = \{q_2, q_3, q_4\}(d_2)$

$\delta''(d_1, a) = \{q_0, q_1, q_2, q_3, q_4\}(d_1)$

$\delta''(d_1, b) = \{q_2, q_3, q_4\}(d_2)$

NFA/DFA eşdeğerliği-Faz 2



Boş katar harici tüm üçlüleri yazalım:

a geçişleri : $(q_1, a, q_0), (q_1, a, q_4), (q_3, a, q_4),$

b geçişleri : $(q_0, b, q_2), (q_2, b, q_4)$

δ'' bu geçişlerle genişletilirse:

$\delta''(d_2, a) = R(q_4) = \{q_3, q_4\}(d_3)$

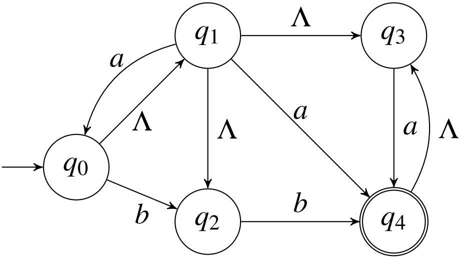
$\delta''(d_2, b) = R(q_4) = \{q_3, q_4\}(d_3)$

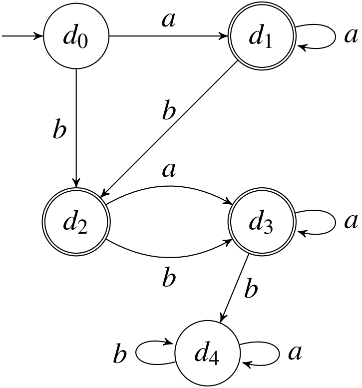
$\delta''(d_3, a) = R(q_4) = \{q_3, q_4\}(d_3)$

$\delta''(d_3, b) = \varnothing(d_4)$

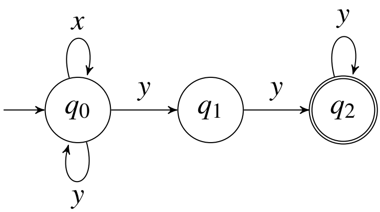
$\delta''(d_4, a) = \delta''(d_4, b) = \varnothing(d_4)$

NFA/DFA eşdeğerliği-Faz 2





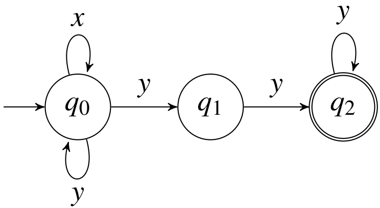
NFA/DFA eşdeğerliğine örnek



$L(M) = (x \vee y)^* yy^+$ dilini tanıyan bir NFA. Eşdeğer bir DFA tanımlayalım.

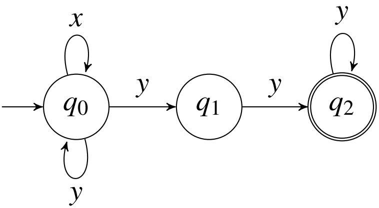
$R(q_0) = \{q_0\}$
$R(q_1) = \{q_1\}$
$R(q_2) = \{q_2\}$

NFA/DFA eşdeğerliğine örnek

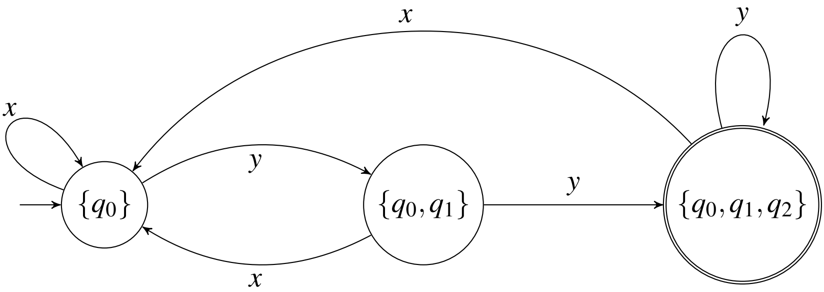


$$\Delta' = \{(q_0,x,q_0),(q_0,y,q_0),(q_0,y,q_1),(q_1,y,q_2),(q_2,y,q_2)\}$$
$$s_0'' = R(q_0) = \{q_0\}$$
$$\delta(s_0'',x) = R(q_0) = \{q_0\}$$
$$\delta(s_0'',y) = R(q_0) \cup R(q_1) = \{q_0,q_1\}$$
$$\delta(\{q_0,q_1\},x) = R(q_0) = \{q_0\}$$
$$\delta(\{q_0,q_1\},y) = R(q_0) \cup R(q_1) \cup R(q_2) = \{q_0,q_1,q_2\}$$
$$\delta(\{q_0,q_1,q_2\},x) = R(q_0) = \{q_0\}$$
$$\delta(\{q_0,q_1,q_2\},y) = R(q_0) \cup R(q_1) \cup R(q_2) = \{q_0,q_1,q_2\}$$

NFA/DFA eşdeğerliğine örnek



$$L(M) = (x \vee y)^* yy^+$$



$$L(M) = ((\Lambda \vee y \vee yy^+)x)^* yy^+$$

Kleene Teoremi (dev.)

Bir L dilinin düzenli olduğunu ispatlamak için; bu dilin, düzenli olduğu bilinen dillerin üzerinde tümleme, birleşim, kesişim, bitiştirme ve Kleen Yıldızı işlemlerinin uygulanmasıyla elde edilebileceğini göstermek yeter.

Kleene Teoremi (cont.)

Bir sonlu otomat ile tanınabilen düzenli diller aşağıdaki işlemler üzerine kapalıdır.

(a) Birleşme
(b) Bitişme
(c) Kleene Yıldızı
(d) Tümleme
(e) Kesişme

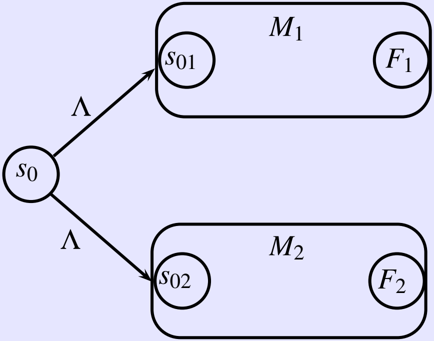
Birleşme

$M_1 = (S_1, \Sigma, \Delta_1, s_{01}, F_1) \leftarrow L(M_1)$ Non-deterministik
$M_2 = (S_2, \Sigma, \Delta_2, s_{02}, F_2) \leftarrow L(M_2)$ Non-deterministik

$M_= (S, \Sigma, \Delta, s_0, F) \leftarrow L(M_1) \cup L(M_2)$ Non-deterministik

$S = S_1 \cup S_2 \cup \{s_0\}$ $F = F_1 \cup F_2$ $\Delta = \Delta_1 \cup \Delta_2 \cup \{(s_0, \Lambda, s_{01}), (s_0, \Lambda, s_{02})\}$

Birleşme

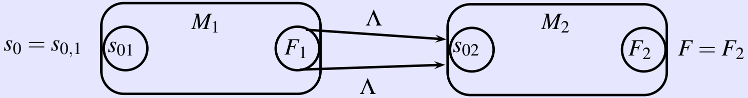


Bitişme (Non-deterministik)

$L(M_1).L(M_2) = L(M)$

$S = S_1 \cup S_2$
$s_0 = s_{01}$
$F = F_2$
$\Delta = \Delta_1 \cup \Delta_2 \cup (F_1 \times \{\Lambda\} \times \{s_{02}\})$

Bitişme

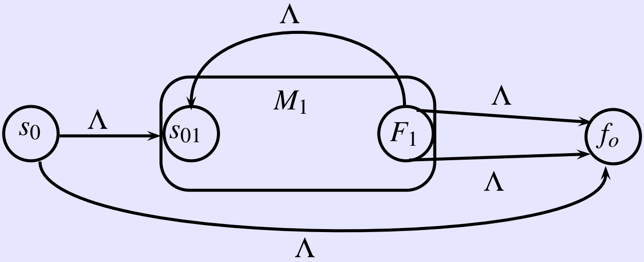


Kleene Yıldızı

$L(M_1)^* = L(M)$

$S = S_1 \cup \{s_0\}$
$F = \{f_o\}$
$\Delta = \Delta_1 \cup (F_1 \times \{\Lambda\} \times \{s_{01}\}) \cup (s_0, \Lambda, s_{01}) \cup (F_1 \times \{\Lambda\} \times F) \cup (s_0, \Lambda, f_o)$

Kleene Yıldızı



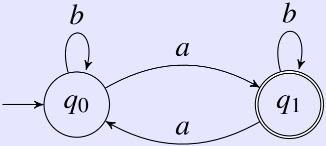
Tümleme (deterministik)

$$\overline{L(M_1)} = L(M)$$

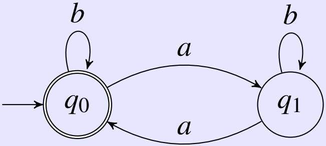
$$F = \{S_1 - F_1\}$$

Tek sayıda a içeren, $\Sigma = \{a, b\}$ üzerine tanımlanmış diller.

Tümleme



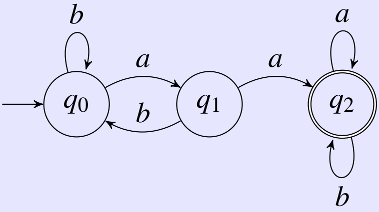
$$q_1 = b^*a(b \vee ab^*a)^*$$



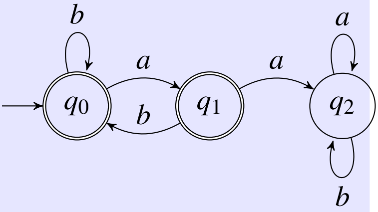
$$q_0 = (b \vee ab^*a)^*$$

En az bir aa çifti içeren, $\Sigma = \{a, b\}$ üzerine tanımlanmış diller.

Tümleme



$q_2 = (b \vee ab)^* aa(a \vee b)^*$



$q_0 \vee q_1 = (b \vee ab)^* (a \vee \Lambda)$

Kesişme

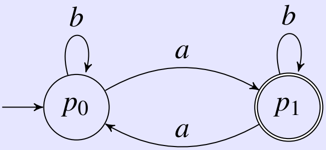
$$L(M_1) \wedge L(M_2) = L(M)$$

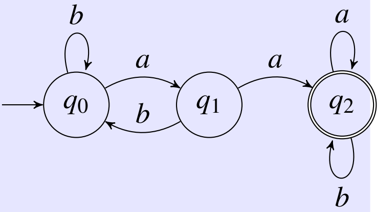
$$S \subseteq S_1 \times S_2$$
$$s_0 = (s_{01}, s_{02})$$

$$F \subseteq F_1 \times F_2$$
$$[(p_1, p_2) \in F] \iff [[p_1 \in F_1] \wedge [p_2 \in F_2]]$$

$$\Delta \subseteq \Delta_1 \times \Delta_2$$
$$[((p_1, p_2), \sigma, (q_1, q_2)) \in \Delta] \iff [[(p_1, \sigma, q_1) \in \Delta_1] \wedge [(p_2, \sigma, q_2) \in \Delta_2]]$$

Kesişme





$$\begin{aligned} S &= \{p_0, p_1\} \\ s_0 &= \{p_0\} \\ F &= \{p_1\} \\ \Delta &= \{(p_0, b, p_0), (p_0, a, p_1), \\ &\quad (p_1, b, p_1), (p_1, a, p_0)\} \end{aligned}$$

$$\begin{aligned} S &= \{q_0, q_1, q_2\} \\ s_0 &= \{q_0\} \\ F &= \{q_2\} \\ \Delta &= \{(q_0, b, q_0), (q_0, a, q_1), \\ &\quad (q_1, b, q_0), (q_1, a, q_2), \\ &\quad (q_2, a, q_2), (q_2, b, q_2)\} \end{aligned}$$

Kesişme

M_1 :

$$S = \{p_0, p_1\}$$
$$s_0 = \{p_0\}$$
$$F = \{p_1\}$$
$$\Delta = \{(p_0, b, p_0), (p_0, a, p_1), (p_1, b, p_1), (p_1, a, p_0)\}$$

M_2 :

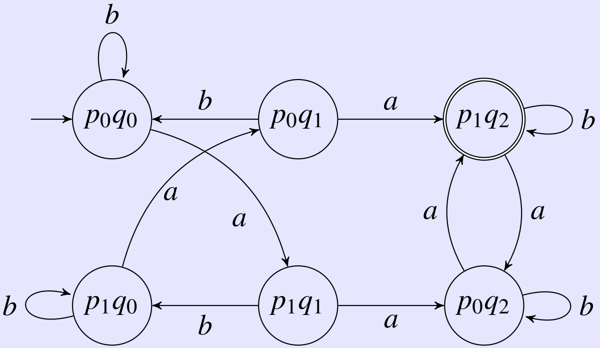
$$S = \{q_0, q_1, q_2\}$$
$$s_0 = \{q_0\}$$
$$F = \{q_2\}$$
$$\Delta = \{(q_0, b, q_0), (q_0, a, q_1), (q_1, b, q_0), (q_1, a, q_2), (q_2, a, q_2), (q_2, b, q_2)\}$$

$M_1 \cap M_2$:

$$S = \{(p_0, q_0), (p_0, q_1), (p_0, q_2), (p_1, q_0), (p_1, q_1), (p_1, q_2)\}$$
$$s_0 = \{(p_0, q_0)\}$$
$$F = \{(p_1, q_2)\}$$
$$\begin{aligned}\Delta = \{&[(p_0, q_0), b, (p_0, q_0)], [(p_0, q_1), b, (p_0, q_0)], [(p_0, q_2), b, (p_0, q_2)] \\ &[(p_1, q_0), b, (p_1, q_0)], [(p_1, q_1), b, (p_1, q_0)], [(p_1, q_2), b, (p_1, q_2)] \\ &[(p_0, q_0), a, (p_1, q_1)], [(p_0, q_1), a, (p_1, q_2)], [(p_0, q_2), a, (p_1, q_2)] \\ &[(p_1, q_0), a, (p_0, q_1)], [(p_1, q_1), a, (p_0, q_2)], [(p_1, q_2), a, (p_0, q_2)]\}\end{aligned}$$

Kesişme

$M_1 \cap M_2:$



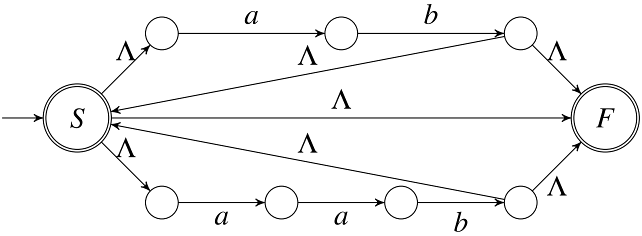
$$\begin{aligned}\Delta = \{ & [(p_0,q_0),b,(p_0,q_0)],[(p_0,q_1),b,(p_0,q_0)],[(p_0,q_2),b,(p_0,q_2)] \\ & [(p_1,q_0),b,(p_1,q_0)],[(p_1,q_1),b,(p_1,q_0)],[(p_1,q_2),b,(p_1,q_2)] \\ & [(p_0,q_0),a,(p_1,q_1)],[(p_0,q_1),a,(p_1,q_2)],[(p_0,q_2),a,(p_1,q_2)] \\ & [(p_1,q_0),a,(p_0,q_1)],[(p_1,q_1),a,(p_0,q_2)],[(p_1,q_2),a,(p_0,q_2)]\}\end{aligned}$$

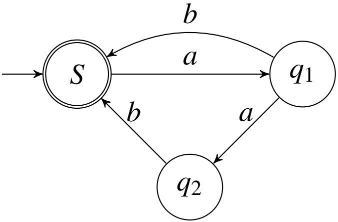
Kleene Teoremi (dev.)

Düzenli diller tümleme ve kesişim işlemlerine kapalıdır.

$$
\begin{aligned}
L_1 \cap L_2 &= \overline{\overline{L_1} \cup \overline{L_2}} \\
&= \Sigma^* - (\Sigma^* - L_1) \cup (\Sigma^* - L_2)
\end{aligned}
$$

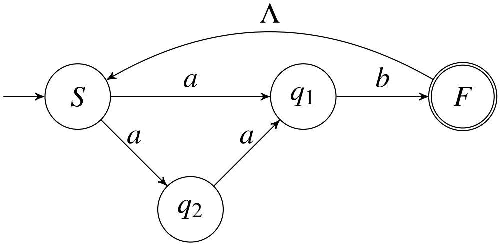
$(ab \vee aab)^*$

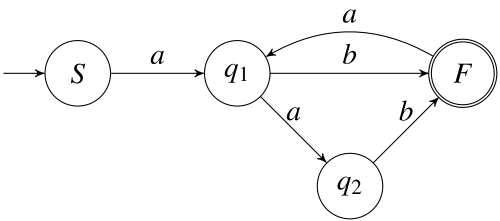




$$< S >::= \Lambda | a < q_1 >$$
$$< q_1 >::= b < S > | a < q_2 >$$
$$< q_2 >::= b < S >$$

$$(ab \vee aab)^+ = (ab \vee aab)(ab \vee aab)^*$$





$< S >::= a < q_1 >$

$< q_1 >::= b|b < F > |a < q_2 >$

$< q_2 >::= b|b < F >$

$< F >::= a < q_1 >$

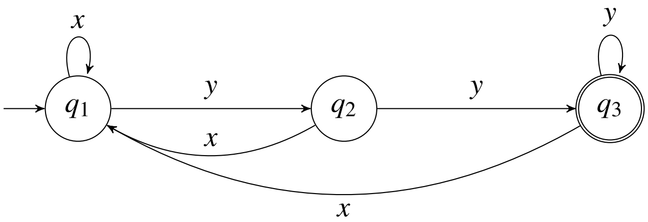
Bir DFA Tarafından Tanınan Düzenli Dili Oluşturmanın Sistematik Yöntemi

$X = XA \cup B \;\wedge\; \Lambda \notin A$ is $X = BA^*$.

Bu ifadeyi düzenli ifadelerle yazalım:

$x = xa \vee b \;\wedge\; \Lambda \neq a \Rightarrow x = ba^*$

Örnek



$q_1 = q_1x \vee q_2x \vee q_3x \vee \Lambda$
$q_2 = q_1y$
$q_3 = q_2y \vee q_3y$
q_3 için teoremi kullanabiliriz
$(q_3) = (q_3)y \vee q_2y$ böylece $q_3 = q_2yy^* \Rightarrow q_3 = q_1yy^+$

Örnek

$(q_3) = (q_3)y \vee q_2 y$ böylece $q_3 = q_2 yy^* \Rightarrow q_3 = q_1 yy^+$

Bu eşitlik kullanılarak:

$q_1 = q_1 x \vee q_1 yx \vee q_1 yy^+ x \vee \Lambda$

$q_1 = q_1(x \vee yx \vee yy^+ x) \vee \Lambda$

$q_1 = (x \vee yx \vee yy^+ x)^* = (y^* x)^*$

$q_3 = (y^* x)^* yy^+$

Örnek otomat önceki yansılarda sunulan bir NFA'nın DFA eşdeğridir.

Sezgisel olarak $(x \vee y)^* yy^+$ ifadesinin NFA'yı temsil ettiğini söylemiştik.

Bu iki ifadenin eşdeğerliğini göstermek gerekirse.

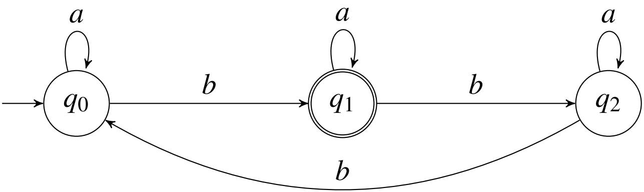
Örneğin tanıtı

İspat edilecek $(y^*x)^*yy^+ = (x \vee y)^*yy^+$
$(y^*x)^* = (x \vee y)^*$

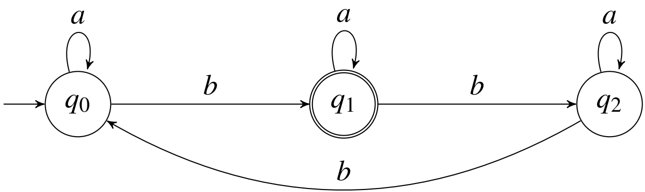
a) $(x \vee y)^*$ ifadesine ait DFA'yı ve $(y^*x)^*$ ifadesine ait NFA'yı çizin
b) (a)'daki NFA'yı DFA'ya çevirin
c) (b)'de elde ettiğiniz DFA'yı indirgeyin

Örnek

İçinde $3k+1$ adet b simgesi bulunan otomatı oluşturunuz ve ilgili düzenli ifadeyi bulunuz.$\Sigma = \{a, b\}$



Örnek



Olası bir çözüm $a^*ba^*[(ba^*)^3]^*$

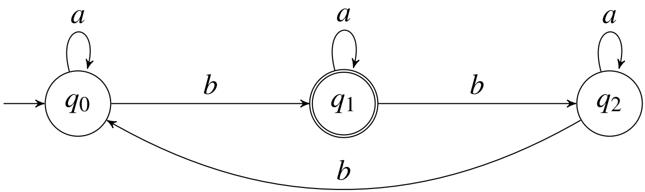
$q_0 = q_0a \vee q_2b \vee \Lambda$

$q_1 = q_0b \vee q_1a$

$q_2 = q_1b \vee q_2a$

$$\begin{array}{l} q_2 = q_2a \vee q_1b \Rightarrow q_2 = q_1ba^* \\ q_1 = q_1a \vee q_0b \Rightarrow q_1 = q_0ba^* \\ \hline q_2 = q_0(ba^*)(ba^*) \end{array}$$

Örnek



Olası bir çözüm $a^*ba^*[(ba^*)^3]^*$

$q_0 = q_0a \vee q_2b \vee \Lambda$

$q_1 = q_0b \vee q_1a$

$q_2 = q_1b \vee q_2a$

$q_0 = q_0a \vee q_0(ba^*)^2b \vee \Lambda$

$q_0 = q_0(a \vee (ba^*)^2b) \vee \Lambda$

$q_0 = (a \vee (ba^*)^2b)^*$

$q_1 = (a \vee (ba^*)^2b)^*b \vee q_1a$

$q_1 = (a \vee (ba^*)^2b)^*ba^*$

$q_1 = (a \vee (ba^*)^2b)^*(ba^*)$

Pumping Lemma

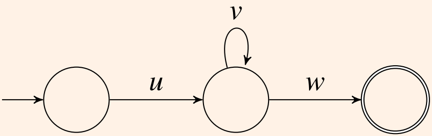
Teorem

L, sözcük uzunluğu kısıtlı olmayan düzenli bir dil ise bu dilin verilen bir n sayısından uzun sözcükleri u, v, w katarlarından oluşacak şekilde uv^iw olarak yazılabilir. Ancak

i $|uv| \leq n$

ii $v \neq \Lambda$

iii $\forall x \in L \quad x = uv^iw \land i \geq 0$



Tanıt

Her düzenli dil L, sonlu determinist bir otomat tarafından tanınır.

- $M = \{S, \Sigma, \delta, s_0, F\}$
- $|S| = n$

$X = \sigma_1 \sigma_2 \ldots \sigma_\ell$, L diline ait bir katar ve $|X| = \ell > n$

$(s_0, \sigma_1 \sigma_2 \ldots \sigma_\ell) \vdash_M (s_1, \sigma_2 \ldots \sigma_\ell) \vdash_M \ldots \vdash_M (s_{\ell-1}, \sigma_\ell) \vdash_M (s_\ell, \Lambda)$ ve $s_\ell \in F$

$\ell > n \wedge |S| = n$ ise güvercin deliği ilkesine göre
$\exists(i,j) : s_i = s_j \wedge 0 \leq i < j \leq n^\dagger \wedge i \neq j$

† Birden fazla çift varsa $|i - j|$ en küçük olan çift seçilir.

Tanıt

Bu durumda s_i, s_j çiftimiz için
$\sigma_1 \sigma_2 \ldots \sigma_i = u$ $\sigma_i \ldots \sigma_j = v$ ve $\sigma_j \ldots \sigma_\ell = w$ olacaktır.

$X = uvw$ şeklinde yazılabilir; bu durumda $X_0 = uw$ ve
$X_m = uv^m w \;\; (m \in \mathbb{N}^+)$ katarları M otomatı tarafından tanınabilir.

Eğer $|X_0| = |uw| \geq n$ ise tanıt X yerine uw sözcüğü ile en baştan tekrarlanabilir. M makinesinin diyagramı bağlı bir graf olarak kabul edilebilir; bu durumda iki durumun arasındaki en uzun yolun uzunluğu durum sayısı olan n'i aşamaz.

$$(s_0, uv^m w) \vdash_M^* (s_i, v^m w) \vdash_M^* (s_i, v^{m-1} w) \vdash_M^* (s_i, w) \vdash_M^* (s_\ell, \Lambda)$$

Pumping Lemma ile Tanıtlama

Pumping lemma kullanılarak bir L dilinin sonlu otomat tarafından tanınamayacağı çelişki yöntemiyle tanıtlanabilir.

1. L dili M otomatı tarafından tanınabiliyor olsun ve M'nin n durumu olsun.
2. L içerisinden seçeceğimiz en az n uzunluğunda bir katar ile Teoremdeki koşullarla çelişki oluşturacağız.

"Pumping lemma" 'yı sağlamayan herhangi bir dil düzenli dil olamaz.‡

‡Gerektirme tek yönlüdür

Örnek 1

$$L(M) = a^n b^n | n \in \mathbb{N}^+$$

Varsayımlar:

- n durumlu bir M otomatı L'yi tanısın
- $x = a^n b^n$ seçelim. Böylece $x \in L$ ve $|x| \geq n$.

Pumping lemma'ya göre

- $x = uvw$ ve $|uv| \leq n$
- 1 harici kullanacağımız herhangi bir i sayısı $uv^i w$ ve $v = a^m$ için çelişki oluşur

Örneğin $uv^2 w$ katarında $a^{n+m} b^n$ ifadesi m adet a'nın x'in ilk kısmına eklenmesiyle elde edilebilir. Bu bir çelişkidir çünkü pumping lemma'ya göre $uv^2 w \in L$, ama $n + m \neq n$.

Örnek 2

$$L(M) = a^{i^2} | i \geq 0$$

Varsayımlar:

- n durumlu bir M otomatı L'yi tanısın
- $x = a^{n^2}$ seçelim

Pumping lemma'ya göre

- $x = uvw$ ve $0 < |v| \leq n$
- $n^2 = |uvw| < |uv^2w| = n^2 + |v| \leq n^2 + n < n^2 + 2n + 1 = (n+1)^2$
- $|uv^2w|$ ifadesini sağlayan bir i^2 olmalı ama karesi n^2 ve $(n+1)^2$'si arasında olan bir i sayısı bulunamaz

Örnek 3

$$L(M) = \{a^s \mid s \text{ bir asal sayı } \}$$

Varsayımlar:

- n durumlu bir M otomatı L'yi tanısın
- $x = a^p a^q a^r$ seçelim ve $p, q \geq 0 \;\; r > 0$
- $p + q + r$ asal olsun

Pumping lemma'ya göre

- $x = uvw$ ve $0 < |v| \leq n$
- $uv^2w = a^p a^q a^r$
- $x = uv^{(p+q+r+1)}w$ için
  $|x| = p + (p+q+r+1)q + r = (q+1)(p+q+r)$ asal olmayan bir sayıdır.