Biçimsel Diller ve Soyut Makineler

$5 + (a*b+c) + (a | b*2 + 3*2)$

yığıt, bellek gerekli

Derleme işlemi

desen dilleri

→ Regüler ifadeler : lex, regex

Kaynak Program
↓
Lexical Analysis
↓ token dizisi
Syntax analysis
↓ syntax tree
Semantik Analiz
↓
Ara kod dönüşümü
↓
Kod optimizasyonu
↓
Kod üretimi
↓
Hedef kod

0101

(isteğe bağlı)

→ reg. yolla move falan.

mak. koduna çevrilir

- • Deterministik finite automata : Bir karakter katarının o dile ait olup olmadığını belirler.
- • Nondeterministik " " (Bazı sırıları şimdi göremeyebilirim, birden fazla cevap olabilir)

* NFA → DFA ya dönüştürülmektedir
* NFA - Λ → NFA → DFA dönüşüm
  (Λ: Boşluk)

* En hızlı DFA

→ Gramerler
- tür-0
- tür-1
- $\boxed{\text{tür-2}}$ → CFG (Context Free Grammar)
- tür-3

(Pushdown Automata) (PDA) → yapısında stack (yığıt) barındıran soyut makine

Dil ile ilgilendiren kısım öbeği bu üçü kadar.

→ Assembly kodu, makine kodu olabilir.

Kod optimizasyonu daha kolaydır.

→ PDA ile çözülen problem
→ Turing ile çözülen PDA ile çözülemeyen

$L = \{a^n b^n \mid n \geq 0\}$

2'li PDA ile çift yığıt kullanılarak çözülebilir. 3'lü çözülemez.

$L = \{a^n b^n c^n \mid n \geq 0\}$

←

| X | X | X | Y | Y | Y | Z | Z | Z | EoS

Okuma / Yazma

Sola hareket / Sağa hareket

Turing Makinesi

Parsing Problemi → (yetersizse)

Deterministik: Alfabedeki her simgeye karşılık bir cevap olmalı ve yalnızca 1 tane olmalı.

Finite: Sonlu, belirli olması

automata: özdevir

Regüler ifadeler

boşluk

boşluğun kümesi

→ boş katar

$x\Lambda = x$

$\Lambda x = x$

1) Λ bir Ri'dir. $L(\Lambda) = \{\Lambda\}$
2) $\emptyset$ bir Ri'dir. $L(\emptyset) = \{\ \}$
3) x bir Ri'dir. $L(x) = \{x\}$ x örüntüsünün tanımına uyan ifadelerden oluşur.
4) a) $L(x) \cup L(Y) = L(X+Y)$ → birleşim operatörü (veya anlamında) • X bir Ri ve Y bir Ri ise X+Y Ri'dir.
   b) $L(x) \cdot L(Y) = L(X.Y)$ → X Ri, Y Ri ise X.Y Ri'dir. (. birleştirme anlamı)
   c) $L(R^*) = (L(R))^*$ → Kleen star operatörü → 0'dan veya dilediğim kadar yan yana yazdırabilirim

Örnek: $(0 + 10^*) = \{0, 1, 10, 100, 100 \ldots\}$

1 olsa bu olmalı! $(0^* 1 0^*) = \{1, 01, 001, 10, 100, 010, 00100, \ldots\}$

$(0 + \Lambda)(1 + \Lambda) = \{\Lambda, 0, 01, 1\}$

$(a+b)^* = \{a, b, aa, bb, ab, \Lambda, aab, aba, \ldots\}$

() ()
()()()(),

yazdırılan; 2 defa, 0 defa
; 4 defa gibi

$(a+b)^* abb$ = son 3 simgesi aab ile biten herşey) $= \{abb, aabb, babb, ababb, baabb, \ldots\}$

*, +, ., de önceliktir

$(11)^* = \{\Lambda, 11, 1111, 111111, \ldots\}$ → çift sayıda 1 içeren bütün karakter katarlarını içeren Ri'dir.

$(aa + ab + ba + bb)^* = \{\Lambda, aa, ab, ba, bb, aaaa, aaab, \ldots\}$

Örnek: a ile biten tek sayıda karakter içeren kelimeleri kabul eden dili yazınız.

1) $(ab + ba + bb + aa)^* a$

$((a+b)(a+b))^* a$

2) a ile biten çift sayıda karakter içeren " " " " "

$(a+b)(aa + ab + bb + ba)^* a$

3) Çiftinci sayıdaki harfleri a olan çift sayıda karakter içeren kelimeleri kabul eden dili yazınız.

$((a+b)\ a\ (a+b)\ a)^*$

4)

$L = \{ x^{2n+1} \mid n \geq 0 \} \rightarrow \boxed{x(xx)^*} + (xx)^* x$

$x^{2n} . x$

olabilir

$x . x^{2n}$

Örnek: $\Sigma = \{a, b\}$ alfabesinde tanımlı en az iki tane a içeren bütün katarları üreten R.İ 'yi yazınız.

$L = (a+b)^* a (a+b)^* a (a+b)^*$

Sadece 2 a olsaydı: $L = (b^* a b^* a b^*)$

Örnek: En az 1 a ve en az 1 b içeren " " " " " "

$(a+b)^* a (a+b)^* b (a+b)^* + (a+b)^* b (a+b)^* a (a+b)^*$

Örnek:

PASCAL

$, (0+1+2+3+4+ \ldots +9)^*$

(*) | → veya olarak da + 'lar karışmasın diye.

digit → 0|1|2|3...|9

dizide olabilir

unsigned_integer → digit . digit*

içinde

unsigned number → unsigned_integer ((.unsigned_integer | Λ) E (+|-|Λ) unsigned_integer) | Λ)

digits → digit digit*

digit+

opt_fraction → (. digits)?

opt_exp → (E (+|-)? digits)?

unsigned_num → digits opt_fraction opt_exp

Biçimsel Diller

Regüler ifadelerle ilgili tanımlamalar

* + , ifadeleri

1. $\emptyset^* = \Lambda$
2. $\Lambda^* = \Lambda$
3. $R^+ = RR^* = R^*R$
4. $R^*R^* = R^*$
5. $(PQ)^*P = P(QP)^*$
6. $RR^* = R^*R$
7. $(a+b)^* = (a^*b^*)^* = (a+b^*)^* = a^*(ba^*)^*$
8. $R+R = R$
9. $L(M+N) = LM+LN$

$$R = Q + RP$$

$$\Lambda \notin P$$

$$\begin{aligned}
R &= Q + (Q+RP)P \\
&= Q + QP + RPP \\
&= Q + QP + (Q+RP)PP \\
&= Q + QP + QPP + RPPP \\
&= Q + QP + QPP + QPPP + RPPPP \\
&\ \vdots
\end{aligned}$$

$$\boxed{R = QP^*}$$

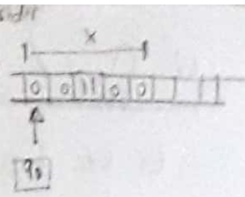
Deterministic Finite Automata (DFA)

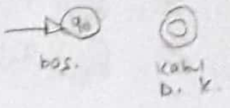
$M = \langle \Sigma , Q , q_0 , \delta , F \rangle$

Σ: Sonlu sayıda elemandan oluşan alfabe
Q: Durumlar kümesi
q_0: başlan. durumu $q \in Q$
δ: geçiş fonk.
F: kabul durumları kümesi

$\delta : \Sigma \times Q \to Q$ — Sadece bir durum.

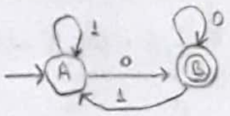
$L(M) = \{ x \in \Sigma^* \mid \delta(q_0, x) \subseteq F \}$ — alt kümesidir





Örnek :

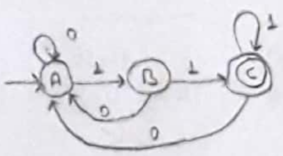
$L = \{ x \in (0,1)^* \mid x, 0\text{'la biter} \}$ bu dili tanıyan bir DFA çiziniz.



$x = 0101010010$ [EoS]

Örnek :

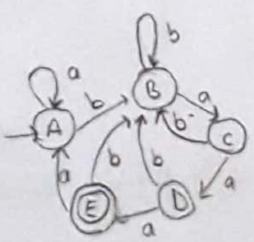
$L = \{ x \in (0,1)^* \mid x, 11 \text{ ile biter} \}$ bu dili tanıyan DFA'yı çiziniz.



$Q \times \Sigma$	0	1
A	A	B
B	A	C
C	A	C

Örnek :

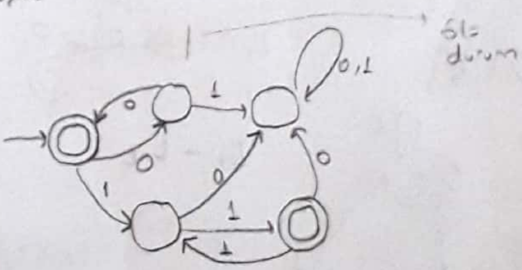
$L = (a+b)^* baaa$



Örnek :

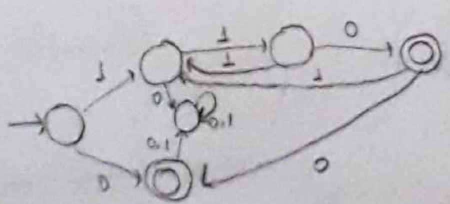
$(00)^* (11)^*$ bu dili tanıyan DFA ?

* Boşluk bu dile ait oldu. dolayı baş. durumu kabul durumudur.



Örnek :

$L = (11 + 110)^* 0$ DFA ?



$\cup$, $\cap$, / Kümelerinin DFA'sı

L_1 ve L_2 dilleri için

$M_1 = \langle Q_1, \Sigma, \delta_1, q_1, F_1 \rangle$

$M_2 = \langle Q_2, \Sigma, \delta_2, q_2, F_2 \rangle$

$\rightarrow$ $M = \langle Q, \Sigma, \delta, q_0, F \rangle$

olası durumlar veya $\leftarrow$ $Q = Q_1 \times Q_2$

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

$L_1 \cup L_2 : Q_1 \times F_2 \cup F_1 \times Q_2$

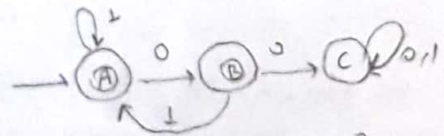
$L_1 \cap L_2 : F_1 \times F_2$

Örnek: $L_1 = \{ x \in (0,1)^* \mid 00,\ x\text{'in altkatarı değil} \}$

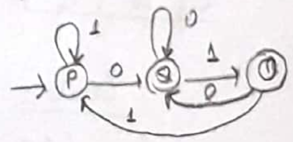
$L_2 = \{ x \in (0,1)^* \mid x,\ 01 \text{ ile biter} \}$

$L_1 \cup L_2$, $L_1 \cap L_2$ için DFA?

M_1:



M_2:



$Q_2 = \{P, Q, R\}$

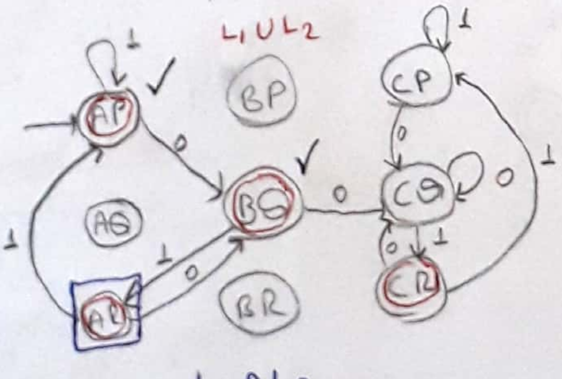
$F_2 = \{R\}$

$Q_1 = \{A, B, C\}$

$F_1 = \{A, B\}$

$L_1 \cap L_2 \Rightarrow F_1 \times F_2 = ((A,R),(B,R))$

$L_1 - L_2$



aynı makinenin kabul durumu

Biçimsel D. Hafta 3

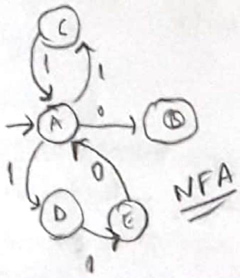
Nondeterministik Finite Automata (NFA)

$R = (11 + 110)^* 0$

$M_{NFA} = \langle Q, \Sigma, q_0, F, \delta \rangle$

$\delta : Q \times \Sigma \to 2^Q$

$\delta^*(q_0, x) \cap F \neq \emptyset$ ise $x \in \mathcal{L}(M_{NFA})$



NFA

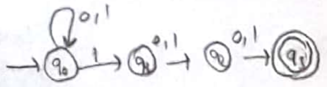
1) $\forall q \in Q$ için $\delta^*(q, \Lambda) = \{q\}$

2) $\forall q \in Q$ için, $y \in \Sigma^*$, $a \in \Sigma$

$$\delta^*(q, ya) = \bigcup_{p \in \delta^*(q,y)}$$

Örnek:

$L = (0+1)^*(1)(0+1)(0+1)$ bu dili tanıyan NFA?



$x = 11 \overset{?}{\in} \mathcal{L}(M_{NFA})$

son biti kopart

$$\delta^*(q_0, 11) = \bigcup_{\boxed{p \in \delta^*(q_0,1)}} \delta(p,1) = \delta(q_0,1) \cup \delta(q_1,1) = \{q_0, q_1, q_2\}$$

$\{q_0, q_1\}$

* q_3 içerilmediğinden, Red

recursiyon (zaman kaybı) (yığıt)

$$\delta^*(q_0, 111) = \bigcup_{\boxed{p \in \delta^*(q_0,11)}} \delta(p,1) = \delta(q_0,1) \cup \delta(q_1,1) \cup \delta(q_2,1)$$

$\{q_0, q_1, q_2\}$

$= \{q_0, q_1\} \cup \{q_2\} \cup \{q_3\}$

$= \{q_0, q_1, q_2, q_3\} \supset \{F\}$

NFA → DFA

$\delta(q_0, 0) = \{q_0\}$

$\delta(q_0, 1) = \{q_0, q_1\}$

$\delta(q_0q_1, 0) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_2\}$

$\delta(q_0q_1, 1) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0, q_1, q_2\}$

$\delta(q_0q_2, 0) = \{q_0, q_3\}$

$\delta(q_0q_2, 1) = \{q_0, q_1, q_3\}$

$\delta(q_0 q_1 q_2, 0) = \{q_0 q_2 q_3\}$

$\delta(q_0 q_1 q_2, 1) = \{q_0, q_1, q_2, q_3\}$

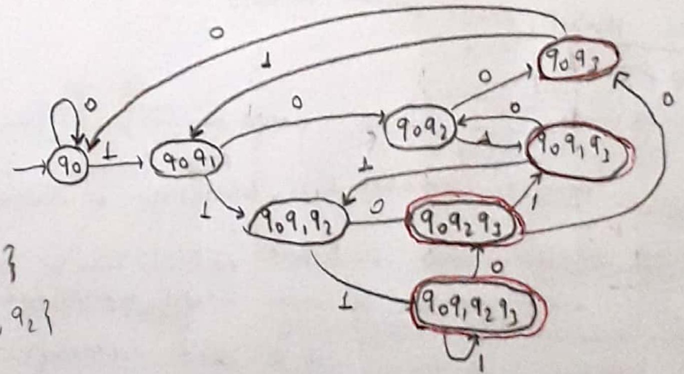
$\delta(q_0 q_3, 0) = \{q_0\}$

$\delta(q_0 q_3, 1) = \{q_0 q_1\}$

$\delta(q_0 q_1 q_3, 0) = \{q_0 q_2\}$

$\delta(q_0 q_1 q_3, 1) = \{q_0 q_1 q_2\}$

⋮



* içinde "q_3" olanlar kabul durumu kabul edilir.

Scanned by CamScanner

NFA-Λ

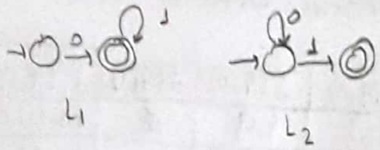
$X\Lambda = \Lambda X = X$

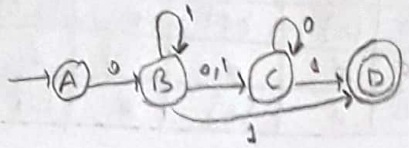
$L_1 = 01^*$

$L_2 = 0^*1$

$L_3 = L_1L_2 = 01^*0^*1$

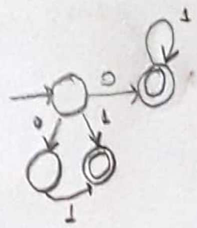
$L_4 = L_1 + L_2 = 01^* + 0^*1$





$L = 01^*0^*1$

$L = 01^* + 0^*1$



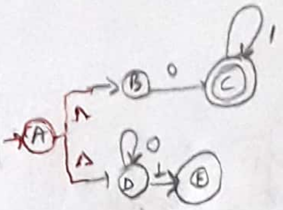
NFA çözümü

1) $\delta^*(q,\Lambda) = \{q\}$

2) $\boxed{\delta^*(q,xa) = \bigcup_{p\in\delta^*(q,x)} \delta(p,a)}$

$L = 01^*0^*1$

NFA-Λ çözüm.



NFA-Λ

$\delta : Q\times(\Sigma\cup\{\Lambda\}) \rightarrow 2^Q$

boşluk kapanması

yani o durumda boşlukla ulaşılabilen küme

① kendisi bu kümeye ait.
③ yeni elemanları Λ ile gittiği yerler
② kendisinden Λ ile gidilen

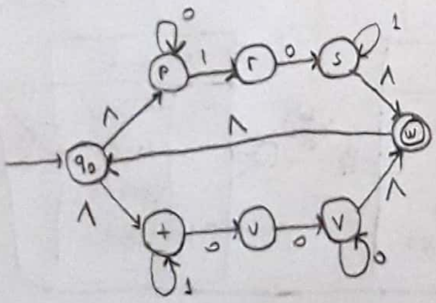
1) $\forall q\in Q,\ \delta^*(q,\Lambda) = \boxed{\Lambda(\{q\})}$

2) $\forall q\in Q$ için, $y\in\Sigma^*,\ a\in\Sigma$

$$\delta^*(q,\underbrace{ya}_{\text{bir katar}}) = \Lambda\Big(\bigcup_{p\in\delta^*(q,y)} \delta(p,a)\Big)$$

$\delta^*(q_0,x)\cap F \neq \emptyset$ ise x katarı kabul edilir.

Örnek



$x = 010 \overset{?}{\in} L(M_{NFA-\Lambda})$

$$\delta^*(q_0,010) = \Lambda\Big(\bigcup_{\boxed{I\in\delta^*(q_0,01)}} \delta(I,0)\Big) = \Lambda(\delta(r,0)) = \Lambda\{s\}$$

$= \{s,w,q_0,p,t\} \supset \{w\}$ old. x katarı kabul edilir.

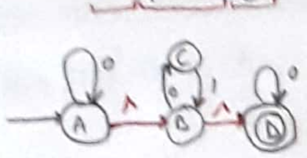
$$\delta^*(q_0,01) = \Lambda\Big(\bigcup_{\boxed{I\in\delta^*(q_0,0)}} \delta(I,1)\Big) = \Lambda\big(\delta(p,1)\cup\delta(u,1)\big) = \Lambda(\{r\}\cup\emptyset) = \{r\}$$

$$\delta^*(q_0,0) = \Lambda\Big(\bigcup_{\boxed{I\in\delta^*(q_0,\Lambda)}} \delta(I,0)\Big) = \Lambda\big(\overset{\emptyset}{\delta(q_0,0)}\cup\overset{p}{\delta(p,0)}\cup\overset{u}{\delta(t,0)}\big)$$

$\{q_0,p,t\} \qquad = \{p,u\}$

Scanned by CamScanner

* Örnek: $L = 0^*(01)^*0^*$ DFA, NFA ve NFA-Λ çizin.



NFA-Λ

$q \backslash \delta$	$\delta(q,\Lambda)$	$\delta(q,0)$	$\delta(q,1)$	$\delta^*(q,0)$	$\delta^*(q,1)$
A	{B}	{A}	∅	{A, B, C, D}	∅
B	{D}	{C}	∅	{C, D}	∅
C	∅	∅	{B}	∅	{B, D}
D	∅	{D}	∅	{D}	∅

* Tek geçiş

* ulaşılacak tüm durumlar

$$\delta^*(A,0) = \Lambda\Big(\bigcup_{\boxed{p \in \delta^*(A,\Lambda)}} \delta(p,0)\Big) =$$

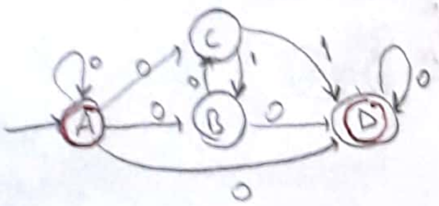
{A, B, D}

$$\Lambda\big(\delta(A,0) \cup \delta(B,0) \cup \delta(D,0)\big)$$

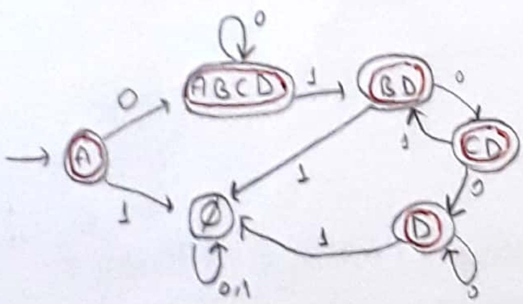
{A, C, D}

= {A, C, D, B}

NFA

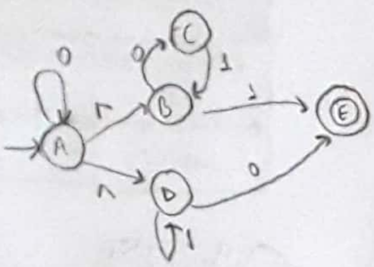


DFA



Biçimsel hafta 4

Extra Örnek $0^*((01)^*1 + 1^*0)$ boşluk geçişli NFA çiz NFA'ya dönüştür oradan DFA'ya ..



NFA-Λ

$Q \backslash \delta$	$\delta(q,\lambda)$	$\delta(q,0)$	$\delta(q,1)$	$\delta^*(q,0)$	$\delta^*(q,1)$
A	$\{B,D\}$	$\{A\}$	$\emptyset$	$\{A,B,C,D,E\}$	$\{E,D\}$
B	$\emptyset$	$\{C\}$	$\{E\}$	$\{C\}$	$\{E\}$
C	$\emptyset$	$\emptyset$	$\{B\}$	$\emptyset$	$\{B\}$
D	$\emptyset$	$\{E\}$	$\{D\}$	$\{E\}$	$\{D\}$
E	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

$$\delta^*(A,0) = \Lambda\left(\bigcup_{r \in \delta^*(A,\lambda)} \delta(r,0)\right) = \Lambda\begin{pmatrix}\delta(A,0)\\ \delta(B,0)\\ \delta(D,0)\end{pmatrix}$$

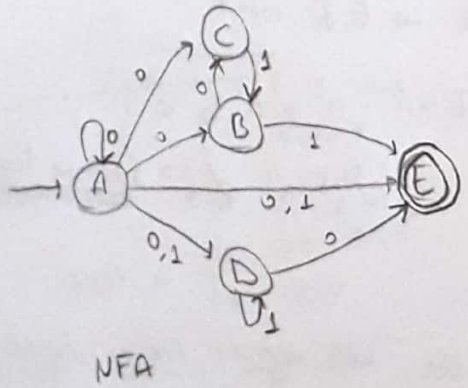
$\{A,B,D\}$ kendisi boşlukla gidilen yerler → sonra bunlardan her birinden 0'la gidilen yerler → ve her birinden 0'la gittiğim yerler

$$= \Lambda(\{A,C,E\})$$ → her birinden Λ ile gittiğim yerler.

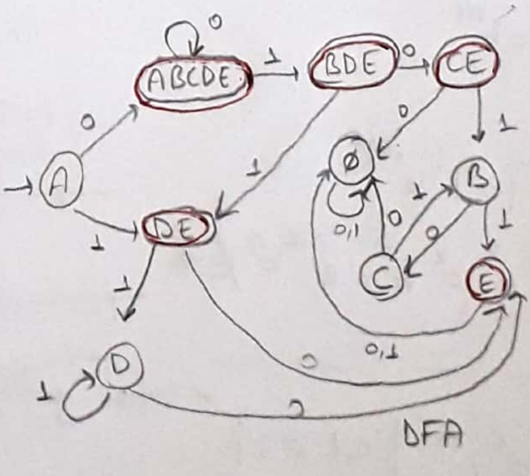
$$= \{A,C,E,B,D\}$$ → bunlardan da Λ ile gidilen yerlere bakılır.

$$\delta^*(B,0) = \Lambda\left[\bigcup_{r \in \delta^*(B,\lambda)} \delta(r,0)\right] = \{C\}$$

$$\delta^*(B,1) = \Lambda\left[\bigcup_{r \in \delta^*(B,\lambda)} \delta(r,1)\right] = \{E\}$$



NFA



DFA

0'la ikisi de boş küme ile durum ($\emptyset$) gelir

→ içinde E bulunan tüm durumlar kabul kümesini oluşturur.

Örnek $w = 011$ birden çok yol doğru sonuca ulaşılır.

$w = 011$ hemen ulaşılır. tek yol

Regüler Dillerin Özellikleri!
(Pumping Lemma)

* Dilin sonsuz sayıda elemanı olda için tekrar gerek yani döngü (v)

Haftaya Quiz
* dönüşüm Vizede
- regüler ifadeler
- NFA, NFA-Λ, DFA

* Durum sayısı indirgeme vize
* Quiz konuları vizede de çıkabilir

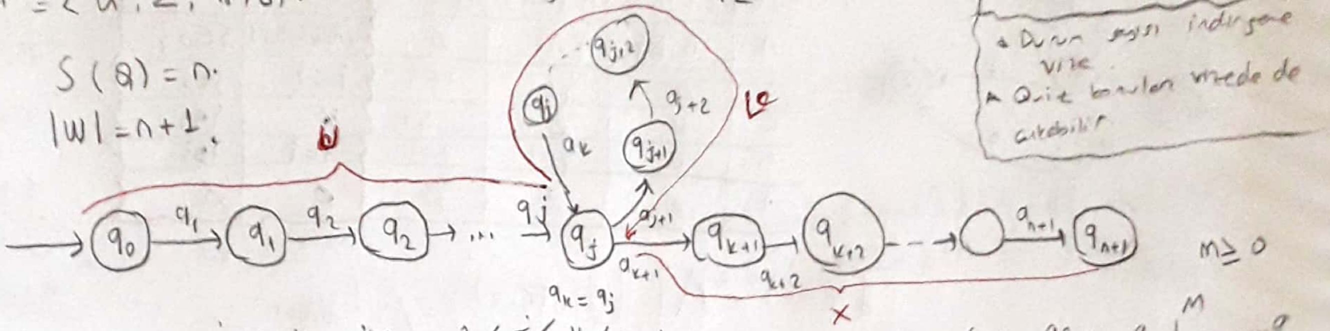
Bir DFA $= \langle Q, \Sigma, q_0, \delta, F \rangle$

$S(Q) = n$

$|w| = n+1$

$|a_j a_{j+1} \dots a_k| \leq 1$

* 3 parçaya ayrılmalı.



j ve k gibi $1 \leq j < k \leq n+1$

$w = \underbrace{a_1 a_2 \dots a_j}_{u} \underbrace{(a_{j+1} a_{j+2} \dots a_k)}_{v} \underbrace{a_{k+1} \dots a_{n+1}}_{x}$

* $\boxed{\text{belli değil}}$ X

$\boxed{\text{Sınavda Sorulabilir}}$

$i \geq 0$

$|v| \geq 1$

$|uv| \leq n$

$u v^i x \overset{?}{\in} \mathcal{L}(DFA)$ ise $\mathcal{L}$ dili regülerdir.

DFA, NFA tasarlanabilir.

Örnek: $\mathcal{L} = \{ a^m b^m \mid m \geq 0 \}$

* Bir DFA çizilemez
Nedeni regüler değil
Pumping lemma ile açıklayın.

$m > 0$ olsun

$a^m b^m$

$\underbrace{aa \dots}_{x} \underbrace{\dots a}_{y} \underbrace{bb \dots b}_{z}$

$y = a^k$ olsun $k > 0$

$x = a^{n-k}$, $z = b^m$

$x y^i z \overset{?}{\in} \mathcal{L}$

$i=2$ için $a^{m-k} (a^k)^2 b^m$

$a^{m-k} a^k a^k b^m = a^{n+k} b^n \notin \mathcal{L}$

$\underbrace{aa \dots}_{x = a^{n-k}} \underbrace{\dots a bb}_{y = a^k b^l} \underbrace{\dots b}_{z = b^{n-l}}$

$i=1$ $x y^i z \rightarrow \in \mathcal{L}$ evet

$i=2$ $x y^2 z = a^{n-k} a^k b^l a^k b^l b^{n-l}$

$= a^n b^l a^k b^n \notin \mathcal{L}$ sıralama hatalı

* hafıza gerektirdiği için DFA değildir regüler değildir,

Örnek: $\mathcal{L} = \{ a^n b^l c^{n+l} \mid n, l \geq 0 \}$

Haftaya: Durum sayısı indirgeme Optimizasyon!

Regüler bir dildir.

Örnek: $\mathcal{L} = \{ a^k b^n a^m \mid k, n, m \geq 0 \}$

- Biçimsel Soru Çözümü -