KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

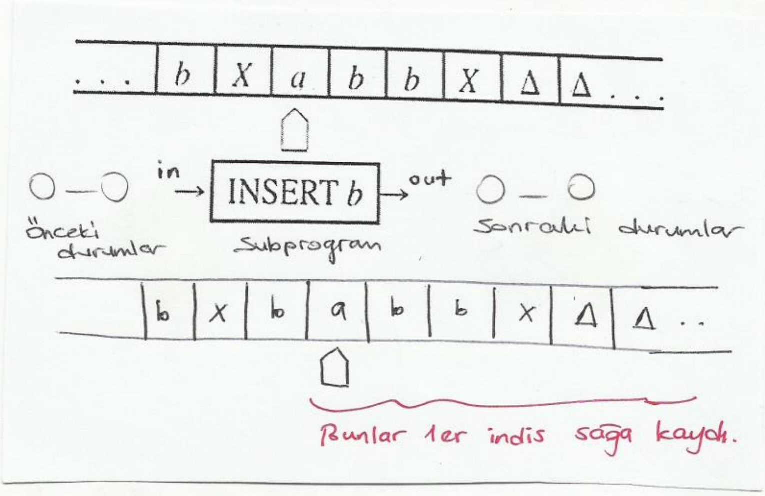
BIL 348 OTOMATA TEORİSİ MINSKY TEOREMİ ve ARDEN TEOREMİ DERS NOTLARI

MELEK ÇİFTÇİBAŞI

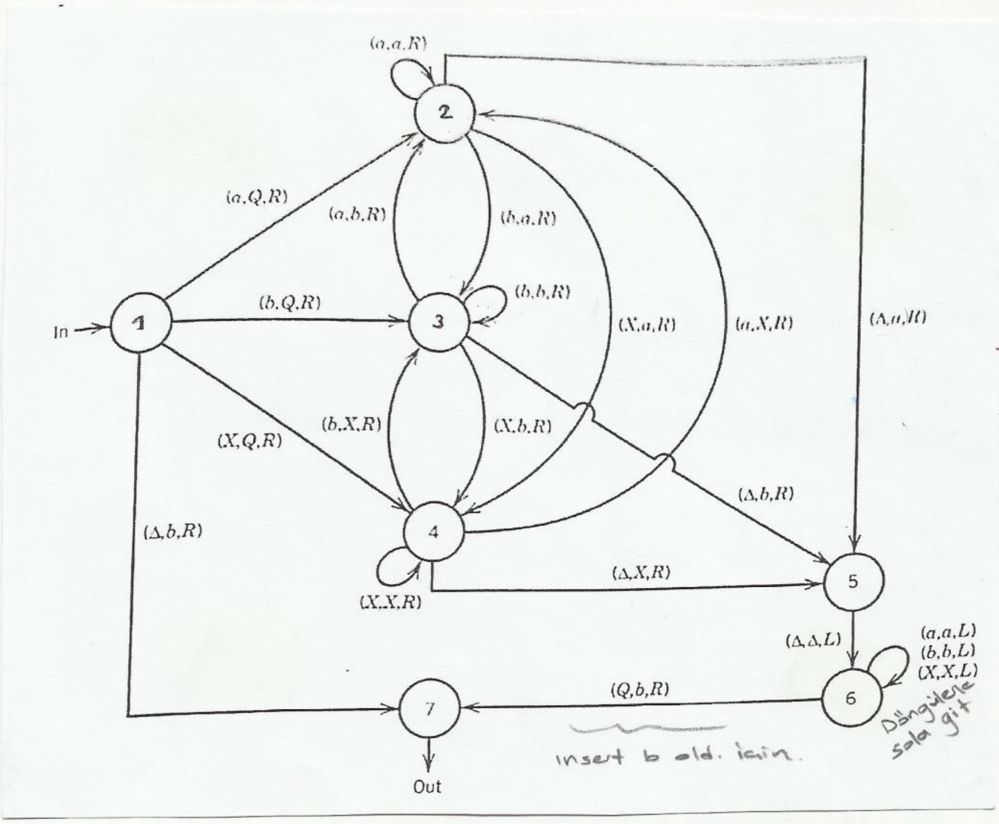
2014-2015 Bahar Dönemi

THE SUBPROGRAM INSERT :

Tapede o anda okuma yazma hafasının işaret ettiği index



Bir karakteri sağa doğru shift yaparken sağındakini de shift yapmalıyız ki onun yerisi kaybolmasın.



- kafanın işaret ettiği karakteri hatırlayabilmek için Q ya çevirmiş.

Q : bnin insert yapacağı indis

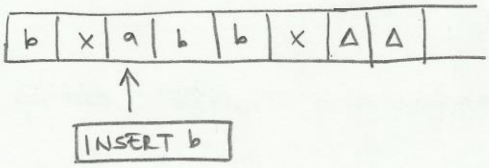
- o indisi boşalt indisin sağındaki karakterleri shift sağa doğru.
- 4-5 arası geçiş tümü shift Δ rastlayınca bnin insert yapılacağı yere Q yazılmıştı.
- Q → b 6-7 arası geçiş

★ a → Q   b → a   b → b

X → b   Δ → X   Δ → Δ

Bir önceki karakter neyse bir sonraki karaktere onu yazıyor. (Bu şekilde shift yapar) Bilgileri kaybetmemek için.

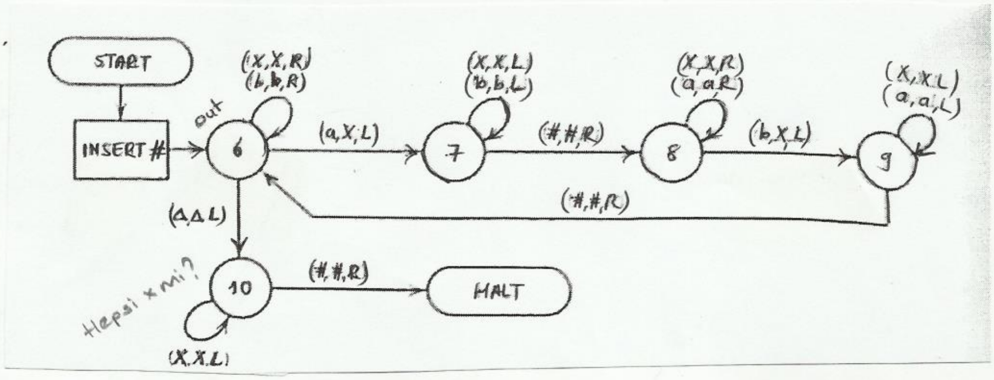
★ Δ , a , R   2 durumunda mesela en son okuduğu a olduğu için a yazar.



a → Q , R
b → a , R
b → b , R
X → b , R
Δ → X , R
Δ → Δ , L

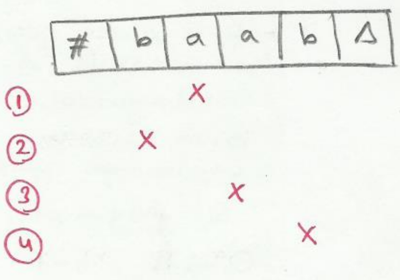
Başa kadar git ← a, b, Δ, =, L
Insert b ← Q → b , R

EQUAL: Herhangi bir TM'de a sayısı b sayısına eşitse HALT yapıyor.



a X b X
sıralı döngüde
a X b X
En son hepsi X ise HALT

#: sola doğru gelirken tape'in başını algılamak için.



- INSERT # yapıyor.
1 - ilk rastladığımız a → X sola
- # görünce sağa
2 - ilk rastladığımız b → X sola
3 a → X tape başına git
4 b → X
5 Δ → Δ bitti hepsi X mi? (X, X, L)

Delete:

INSERT'e çok benziyor. Bu sefer sola kaydıracağız.

-- | F | R | I | E | N | D | Δ --

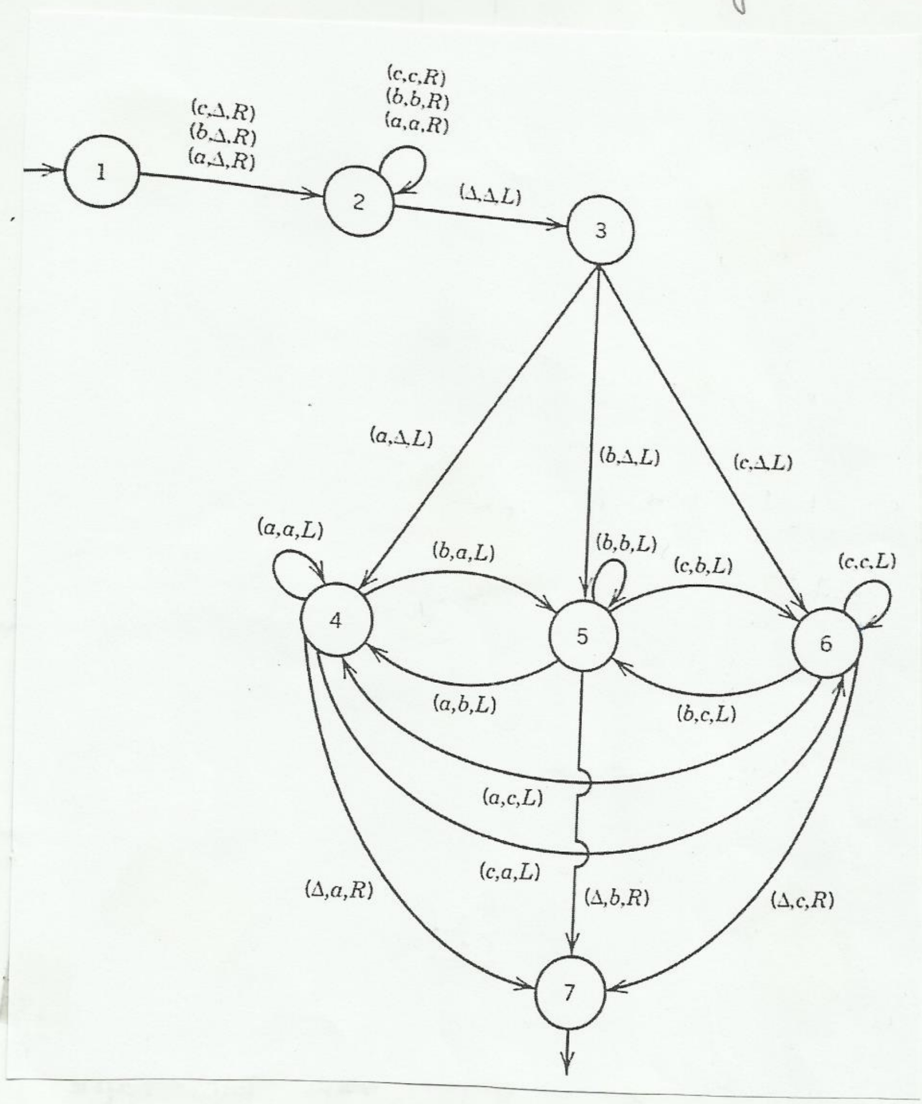
↑
Delete

| F | I | E | N | D | Δ ---

↑ 1er karakter sola kaydıracağız

- INSERT'te hatırlamak için Q'ya çeviriyorduk. Burada silinecek bu yüzden Δ (boşluk) çevrilir.

- DELETE öncesi hangi indisi işaret ediyorsa sonrası yine aynı indis.



c → Δ sağa
Δ → Δ sola
a → Δ sola
b → a sola
a → b sola (4-5)
Δ → a sağa

(Yukarıda a → Δ olduğundan dolayı)

- Önceki geçişler a

.. | a | b | a | Δ | ..

↑

Δ → b → a → Δ

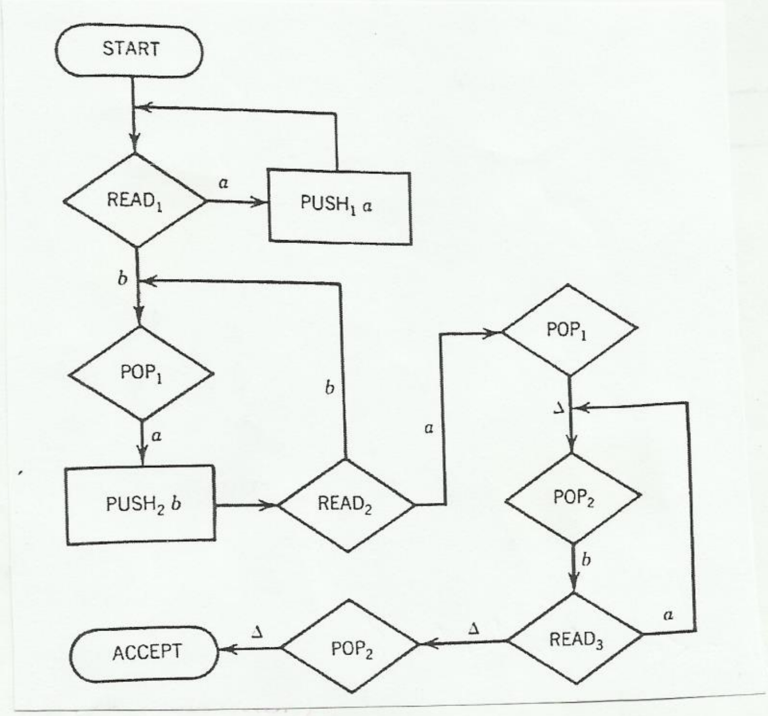
b ← a ← Δ ←

(Sonra tekrar sola) yine aynı indisi işaret etsin.

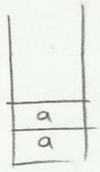
MINSKY TEOREMİ: $\boxed{2PDA = TM}$

2 yığınlı PDA'nın TM eşdeğeri nedir bunu araştırıyoruz.

ÖR $L = \{ a^n b^n a^n \ , \ n>0 \}$

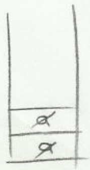


a	a	b	b	a	a	Δ

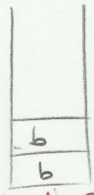


Stack 1

READ1 $\xrightarrow{a}$ PUSH$_1$ a



Stack 1

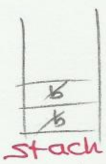


Stack 2

READ1 $\xrightarrow{b}$ POP$_1$ a $\longrightarrow$ PUSH$_2$ b

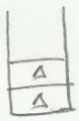
READ2 $\xrightarrow{a}$ POP$_1$ Δ (yığın boş mu)

POP$_2$ $\xrightarrow{b}$ READ$_3$ a

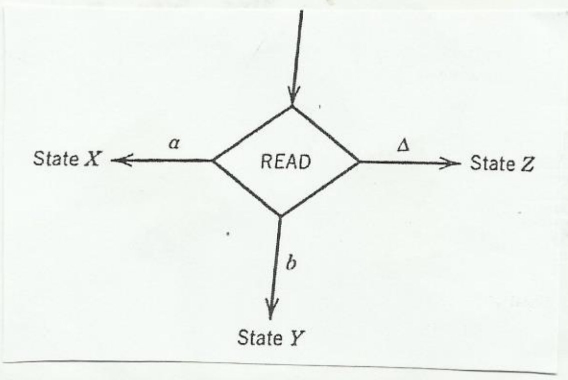


Stack 2

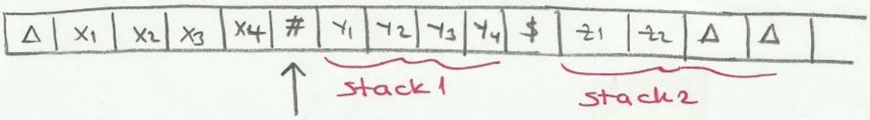
READ$_3$ $\xrightarrow{\Delta}$ POP$_2$ (yığın boş mu)

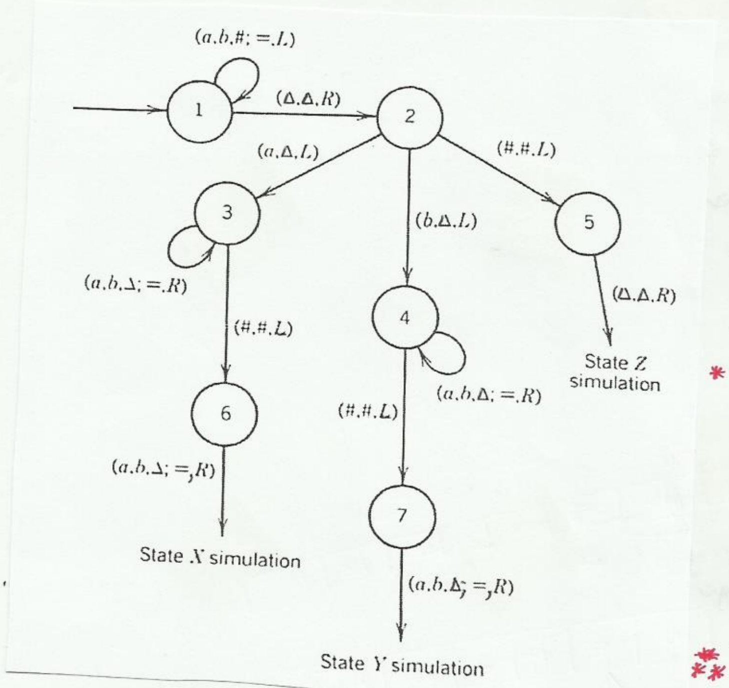


READ



PM → TM geçişine benzer iki yığın tek bir tape ile simüle edilmeli.





Eşdeğerleri çıkarırken en son durumda kafa yine # üzerine gelecek. (Hepsinin buluşma noktası)

* # konumundan tape başına gideriz (1 durumu)
a, b, # ; = . L (Okuduğunu aynen yaz sola git)

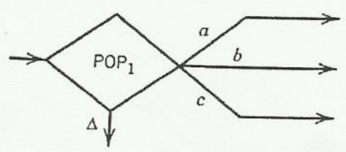
En başında old. anlamak için Δ karakteri var.

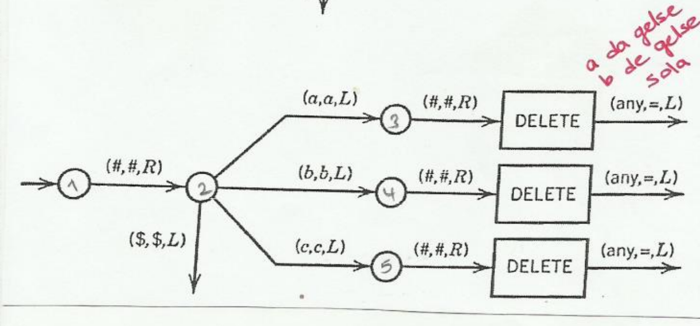
** Tapeden okuduğu karakteri Δ yapıyoruz (2 durumu)

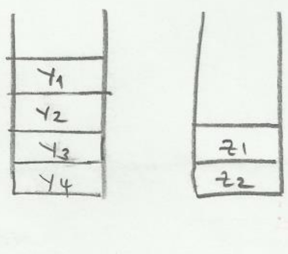
* (3, 4) Döngülerle sağa gidiyoruz.

* # 'i görünce sola gidiyoruz sonra tekrar sağa (Diyez üzerinde kalmak için) (3 → 6, 4 → 7)

POP1 : iki farklı yığın olduğu için POP1, POP2 diyoruz. Eşdeğerleri farklı, tapede yerleri farklı old. için.



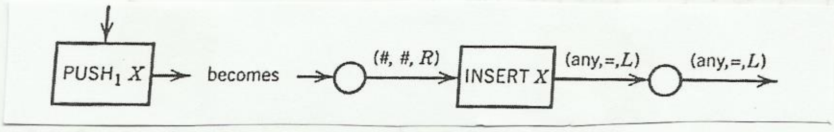




- 2. durum yığın boşsa $, $, L yine # üzerine.
- karakter varsa onu tekrar yazıyor sola.
- sildikten sonra sola tekrar #'a gelmek için (any, =, L)

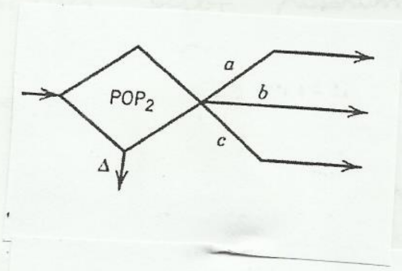
DELETE gibi. y1 sil y2 y3 y4 sola kaydır.

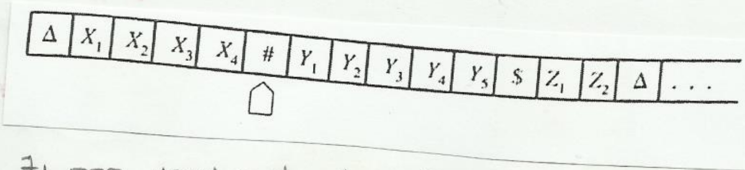
PUSH1 :



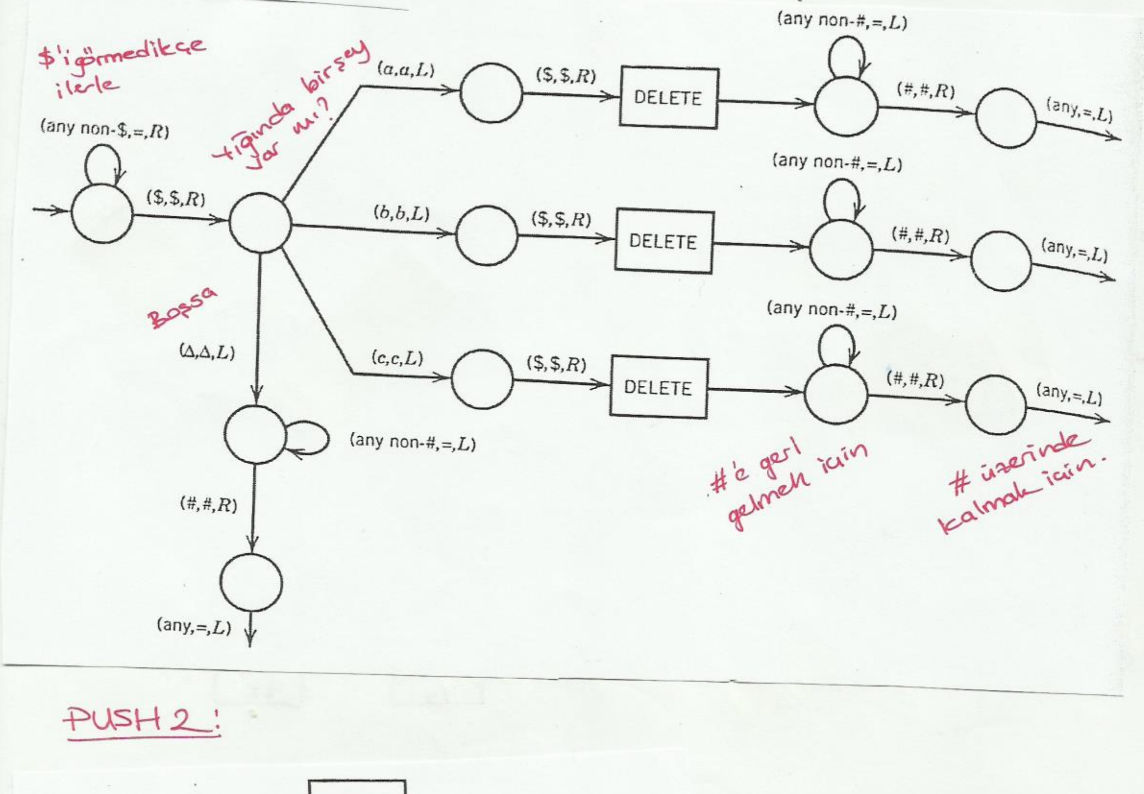
sonsuz büyüklükte yığın var boş da olsa dolu da olsa INSERT yapıyoruz. Y1 in önüne Y0 INSERT yaptık. Sonra yine Y1'e kafa işaret eder.

POP2 :

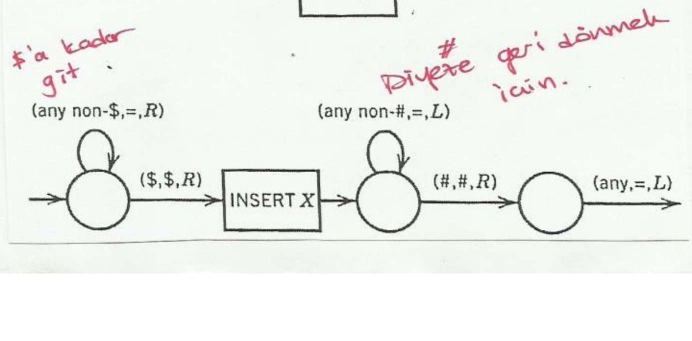




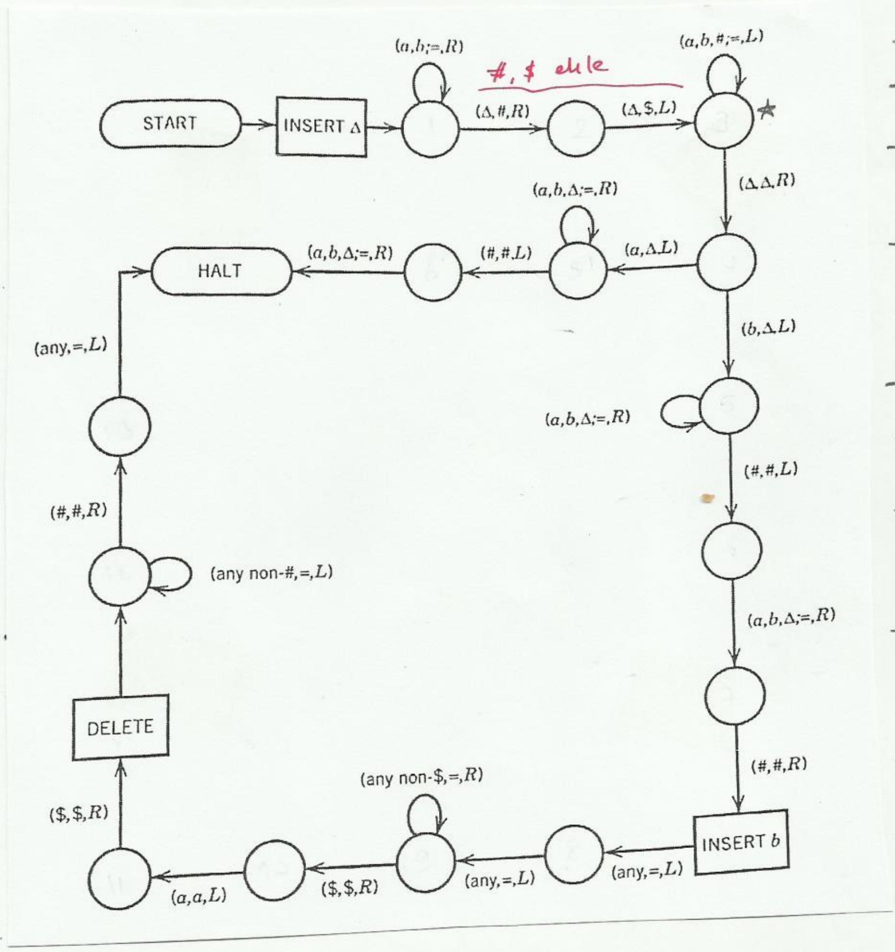
Z1 pop yapılacak 1. yığını geçeceğiz sonra #'e geri geleceğiz.



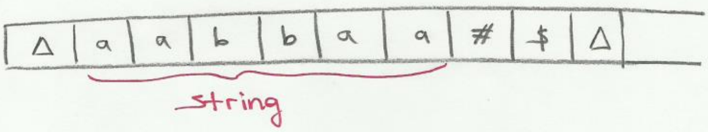
PUSH2 :

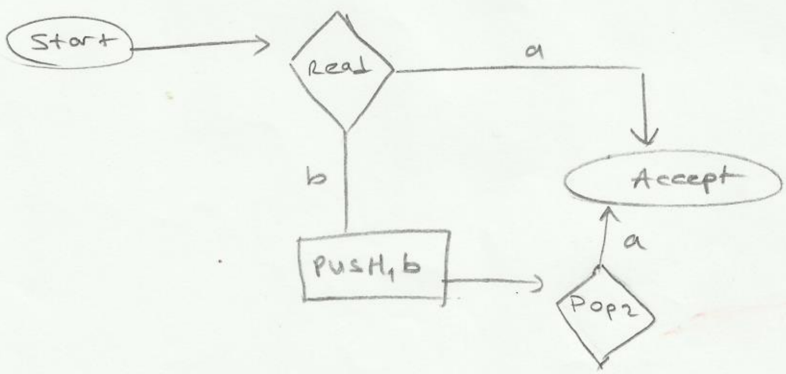


PUSH2 (Devamı)



- Yığınlar boş
- # peşine $ gelecek
- string başına gitti ★
- a, b hangisini okursa okusun #'e kadar gidecek.
- sonra tekrar sağa
- #, Δ ve HALT
- #'tan sonra b'yi ekleyeceğiz (INSERT)
- DELETE bir karakter var mı diye bakıyoruz
- a'yı siliyoruz.
- #'i görünce sağa sonra tekrar sola HALT





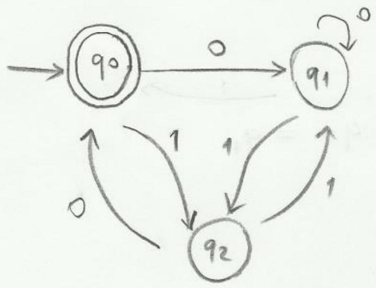
14

ARDEN TEOREMİ :

$\boxed{X = AX + B}$ → $\boxed{X = A^{*}B}$

A: 0,1'lerden oluşmuş R.E

B: Hem durum hem de RE ikisi de olabilir.



$q_0 = 0.q_1 + 1.q_2 + \varepsilon$ (Bitiş durumu old. için ε vardır)

$q_1 = 0.q_1 + 1.q_2$ (Hepsi kendinden çıkan oklar)

$q_2 = 0.q_0 + 1.q_1$

$\overset{X}{q_1} = \overset{A.X}{0q_1} + \overbrace{1.q_2}^{B} = \boxed{0^{*}.1.q_2 = q_1}$

$q_2 = 0q_0 + 1.q_1$

$q_2 = 0.q_0 + 1.0^{*}.1q_2$

$\overset{X}{q_2} = \overbrace{1.0^{*}.1}^{A}.\overset{X}{q_2} + \overbrace{0.q_0}^{B}$ $\boxed{q_2 = (1.0^{*}.1)^{*}\ 0.q_0}$

$q_0 = 0.q_1 + 1.q_2 + \varepsilon$

$= 0.0^{*}.1.q_2 + 1.q_2 + \varepsilon$

$= (0.0^{*}.1 + 1)q_2 + \varepsilon$

$\underbrace{q_0}_{X} = \underbrace{(0.0^{*}.1+1)(1.0^{*}.1)^{*}\ 0}_{A}\ \underbrace{q_0}_{X} + \underbrace{\varepsilon}_{B}$

$q_0 = \left[(0.0^{*}.1+1).(1.0^{*}.1)^{*}\ 0\right]^{*}.\varepsilon$

Çarpan olarak geldiğinde direkt silebiliriz.

q_1 in içerisine q_2 yi gömelim;

$$q_0 = 0.q_1 + 1.q_2 + \varepsilon$$
$$q_1 = 0.q_1 + 1.q_2 = 0^*.1.q_2$$
$$q_2 = 0.q_0 + 1.q_1$$

$$q_1 = 0^*.1.(0.q_0 + 1.q_1)$$
$$0^*.1\,0.q_0 + 0^*.1.1.q_1$$

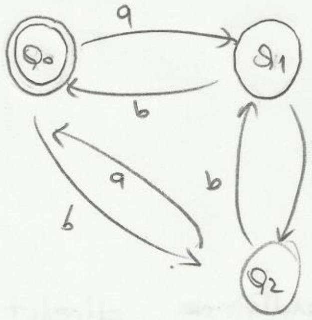
$$\underbrace{q_1}_{X} = \underbrace{0^*.1.1}_{A}.\underbrace{q_1}_{X} + \underbrace{0^*.1.0.q_0}_{B} = (0^*.1.1)^*.0^*.1.0.q_0 = q_1$$

$$q_0 = 0.q_1 + 1.q_2 + \varepsilon$$
$$= 0.(0^*.1.1)^*.0^*.1\,0\,q_0 + 1.(0.q_0 + 1.q_1) + \varepsilon$$
$$= 0(0^*.1.1)^*\,0^*.1\,0.q_0 + 1.(0.q_0 + 1\underbrace{(0^*.1.1)^*.0^*.1.0.q_0}_{q_1} + \varepsilon$$
$$= 0.(0^*.1.1)^*\,0^*.1.0.q_0 + 1.0.q_0 + 1(0^*.1.1)^*.0^*.1.0.q_0 + \varepsilon$$

$$\underbrace{q_0}_{X} = \underbrace{\left[0.(0^*.1.1)^*.0^*.1.0 + 1.0 + 1(0^*.1.1)^*0^*.1.0\right]}_{A}\underbrace{q_0}_{X} + \varepsilon$$

$$RE = \left[\underbrace{0.(0^*.1.1)^*0^*1.0}_{I} + \underbrace{1.0}_{II} + \underbrace{1(0^*.1.1)^*.0^*.1.0}_{III}\right]^*.\varepsilon$$

ÖR:



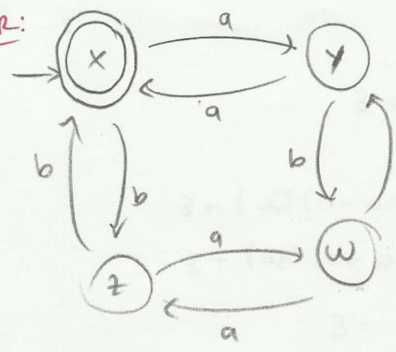
$$Q_0 = a.Q_1 + b.Q_2 + \varepsilon$$
$$Q_1 = a.Q_2 + b.Q_0$$
$$Q_2 = a.Q_0 + b.Q_1$$

Bunların hepsi kendinden çıkan okları gösteriyor.

$$Q_1 = a(a.Q_0 + b.Q_1) + bQ_0$$
$$= aa.Q_0 + abQ_1 + bQ_0$$

$$\underbrace{Q_1}_{X} = \underbrace{ab}_{A}.\underbrace{Q_1}_{X} + \underbrace{(aa+b).Q_0}_{B} = (ab)^*(aa+b)Q_0$$

ÖR:



$$X = aY + bZ + \overset{son}{\varepsilon}$$
$$Y = aX + bW$$
$$Z = aW + bX$$
$$W = aZ + bY$$

Y'yi elimine:

$$X = aaX + abW + bZ + \varepsilon$$
$$Z = aW + bX$$
$$W = aZ + baX + bbW$$

Z'yi elimine:

$$X = aaX + abW + baW + bbX + \varepsilon$$
$$(aa+bb)X + (ab+ba)W + \varepsilon$$

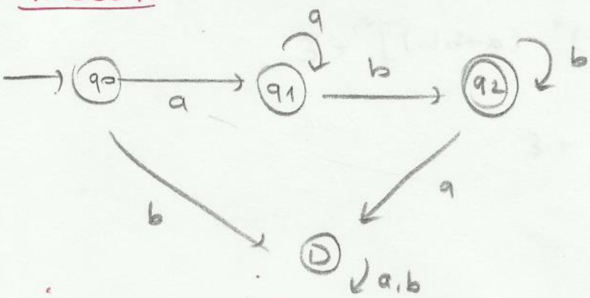
$$W = aaW + abX + baX + bbW$$
$$W = \underbrace{(aa+bb)}_{A}W + \underbrace{(ab+ba)X}_{B}$$
$$W = (aa+bb)^*(ab+ba)X$$

W'yi elimine:

$$X = (aa+bb)X + (ab+ba)(aa+bb)^*.(ab+ba)X + \varepsilon$$
$$= [aa+bb+(ab+ba)(aa+bb)^*(ab+ba)]X + \varepsilon$$

$$RE = [aa+bb+(ab+ba)(aa+bb)^*(ab+ba))^*$$

Videoda:



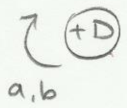
$$X = A^* B$$

$$q_0 = aq_1 + b.D$$
$$q_1 = a.q_1 + bq_2$$
$$q_2 = a.D + bq_2 + \lambda$$

kendi üzerine geliyorsa
RE dahil edilmiyor
Bitiş hariç

$$D = aD + bD$$
$$= (a+b)D + \phi$$
$$= (a+b)^*\phi = \phi$$

Bitiş:



$$D = aD + bD + \varepsilon$$
$$= (a+b)D + \varepsilon$$
$$D = (a+b)^*\varepsilon$$

$$Q_0 = a.Q_1 + bQ_2 + \varepsilon$$

$$= a.(ab)^*.(aa+b).Q_0 + b.\underbrace{(aQ_0 + b.Q_1)}_{Q_2} + \varepsilon$$

$$= a(ab)^*(aa+b)Q_0 + b(aQ_0 + b.(ab)^*(aa+b)Q_0) + \varepsilon$$

$$= a(ab)^*(aa+b).Q_0 + baQ_0 + bb(ab)^*(aa+b)Q_0) + \varepsilon$$

$$= [a(ab)^*(aa+b) + ba + bb(ab)^*(aa+b)]Q_0 + \varepsilon$$

$$Q_0 = [(ab)^*(aa+b)((a+bb)+ba]Q_0 + \varepsilon$$

$$RE = [(ab)^*(aa+b).(a+bb) + ba]^*$$

$$Q_2 = aQ_0 + b(a.Q_2 + bQ_0)$$

$$= aQ_0 + baQ_2 + bbQ_0$$

$$\underbrace{Q_2}_{X} = \underbrace{ba}_{A}\underbrace{Q_2}_{X} + \underbrace{(a+bb)Q_0}_{B} = (ba)^*.(a+bb)Q_0$$

$$Q_0 = a\underbrace{(a.Q_2 + b.Q_0)}_{Q_1} + b\underbrace{(ba^*)(a+bb)Q_0}_{Q_2} + \varepsilon$$

$$Q_0 = Q_0[aa\underbrace{(ba^*).(a+bb)}_{Q_2} + ab + b(ba)^*(a+bb)] + \varepsilon$$

$$RE = [aa(ba)^*(a+bb) + ab + b(ba)^*(a+bb)]^* + \varepsilon$$

$$RE = [(ba)^*(a+bb)(aa+b) + ab]^* + \varepsilon$$