Repeated Application of the Rules

$*$ → istediğin kadar tekrar et

$$(a+b^{*})^{*} = (a+b)^{*}$$

$$(a^{*})^{*} = a^{*}$$

ANCAK

$$(aa+ab^{*})^{*} \neq (aa+ab)^{*}$$

→ Check the "abbabb"

⇒ Bir RE cebirsel kuralını henüz bulurmad,

ÖRN $b^{*}(abb^{*})^{*}(\Lambda+a)$

→ duble a içermeyen kelimelerin ifadesi

→ kelime b'yle başlıyor

$(a+b^{*})^{*} = (a+b)^{*}$ 01/11/24

Uç durum } $\Lambda = \Lambda$ ①

sonsuz kere tekrar. ②

$$(a+b^{*})(a+b^{*})^{*} = (a+b)(a+b)^{*}$$

$$bbb\,(a+b^{*})^{*} = (a+b)(a+b)(a+b)^{*}$$

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

$$bbb\,(a+b^{*})^{*} = b\quad b\quad b\quad (a+b)^{*}$$

→ Türetme aşamaları kullanılarak birbirine benzer şekilde türetilebilir.

→ Double) ard arda bitişik 2 harf) aa

→ Çift sayı ve double aynı şey değil.

(some)

→ Bazı ifadesini görürsen a, b, 1 olabilir.

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

→ Türetilen tüm kelimeler double a içermeyecek

→ (abb^*) ⇒ ortadaki b en az b içermesini sağlayan

→ some b : herhangi bir sayı veya harf olabilir (0, 1, b)

→ (abb^*) ⇒ a'dan sonra b gelmeli (double a olmayacak)

EVEN - EVEN

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

Anlayabilmek için bileşenlere ayıralım:

tip 1 = aa
tip 2 = bb
tip 3 = $(ab + ba)(aa + bb)^* (ab + ba)$

$E = [\text{tip } 1 + \text{tip } 2 + \text{tip } 3]^*$

Tip3 | $(ab + ba)(aa + bb)^* (ab + ba)$

1. ihtimal ab Λ → 1.1 ab
→ 1.2 ba

2. ihtimal ba → 2.1 ab
→ 2.2 ba

$(aa + bb)(aa + bb)^*$ $(2n+1)$

$2x$
$2y$

$(2n+1)(2x+2y)$) Her türlü çift

Finite) sonlu

α

Yet Another Method for Defining Languges

State → Taşların tahtadaki muhtemel pozisyonları

→ Her sayı geldiğinde durum değişmeyebilir (same state)

Final State → Oyunu kazandıran durum

Oyunda bir başlangıç state'i var (unique)

Finite Automaton

Finite → Alfabeden oluşturulan olası durumlar veya harflerin sonlu olm.

Automaton → Inputa göre durumun değişmesi

Automata → Automaton'un çoğul hali

Final Otom

maybe one

① Sonlu sayıda durum. Bir start state, some final state

② Bir alfabe olmalı

③ Sonlu sayıda geçiş (transition)

→ Bu tanım FA nasıl çalıştığını anlatmıyor

→ Input string son harfe kadar harf harf soldan başlayarak okunur

→ Start state'ten başlar.

→ Son harfe gelince okuma biter

ÖRN: $\Sigma = \{a,b\}$

States	Input
− x	a → y b → z
y	a → x b → z
+ z	a → z b → z

① kuralları yaz

② $\underbrace{\text{Start}}_{-}$ ve $\underbrace{\text{final}}_{+}$ state'i işaretle yanına −, + at

soru aaa, abba okunursa ne olur?

Kural soldan başla teker teker sağa doğru oku.

① $\cancel{a}\cancel{a}\cancel{a}$ $x\ a \rightarrow y\ a \rightarrow x\ a \rightarrow y$

aaa bu dilin parçası değil

② $\cancel{a}\cancel{b}\cancel{b}\cancel{a}$ $x\ a \rightarrow y\ b \rightarrow z\ b \rightarrow z\ a \rightarrow z$

abba bu dilin bir parçasıdır.//

Dilin Tanımı → içerisinde b bulunan tüm stringler

$(a+b)^* b\ (a+b)^*$) Bu dilin reguler expression'ı

→ Finite otomata aslında bir dili tanımlama yöntemi.

Transition Table

Her bir satır → state
sütun → geçiş yapılacak state

	a	b
start x	y	z
y	x	z
final z	z	z

→ satır

sütun

Abstract Definition of an FA

① Finite states $Q = \{q_0, q_1, q_2 \ldots\}$ q_0 → start state

② Q'nun alt kümesi → final states

③ Bir alfabe $\Sigma = \{x_1, x_2, x_3 \ldots\}$

④ $\delta(q_i, x_j) = x_k$) Transition func. her durum.

Transition Diagram

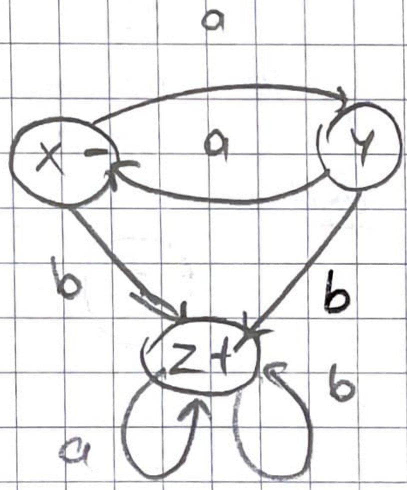
Circle → O anki durum

Arrow → Bir durumdan diğerine geçiş

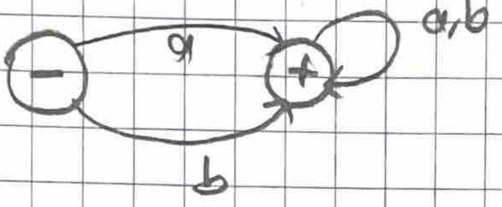
Label arrows → Tekrar eden kelime

Minus (−) → Start state

Plus (+) → Final state



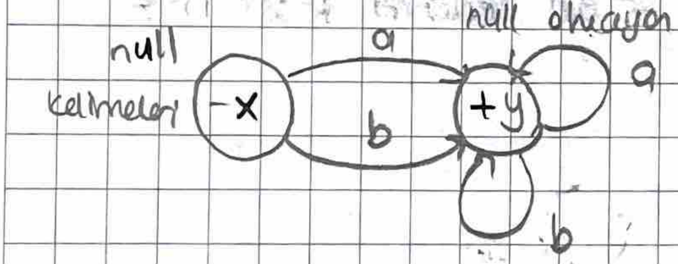
ÖRN $\overbrace{(a+b)(a+b)^*}^{\text{düzenli ifade}} = (a+b)^+$ ↳ düzenli ifade değil



Bu makine Λ hariç her stringi kapsar

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

→ Λ karakteri içermediği için: $(a+b)$

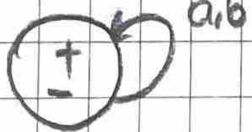


→ İçinde null içermeyen tüm kelimeleri kabul eden finite automata çiz.

→ Ünite sonundaki soruları çöz.

→ b ile biten dilin düzenli ifadesi, finite automatası

ÖRN



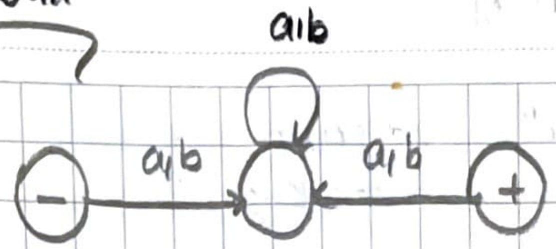
Bu dil;
Null da dahil a ü. a ve b
içeren tüm kelimeleri kabul eden

Düzenli ifade → $(a+b)^*$

Önemli

ÖRN

①



sonsuz döngüye girer, çıkamaz.

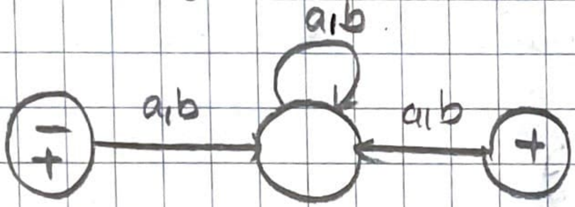
→ Böyle bir Final Automata vardır, herhangi bir dili kabul etmez.

Bunlar 2 tiptir:

① No final state

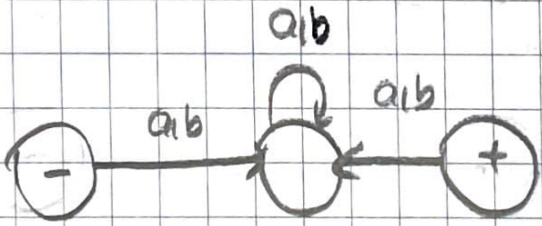
② Final states can't be reached from the start state

*



①

Λ ✓



②

Λ ✗

Aynı değil birini kabul eder diğeri etmez.

→ Sunu 2' de L_1, L_2 işlemlere bak,,

FAs and Their Languages

Bir kelimenin bir dile ait olup olm. belirlemek FA'da, RE'e göre daha kolay.

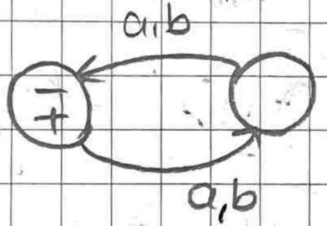
Sınav sorusu tipi

Bir dil verilir, FA çiz.
Bir otomata, dil formula.

ÖRN $\Sigma = \{a,b\}$ çift sayıda harf içeren tüm kelimeleri kapsayan otomata

abba ...

① Başarılı State
② Başarısız State
③ Son durum Başarısız State



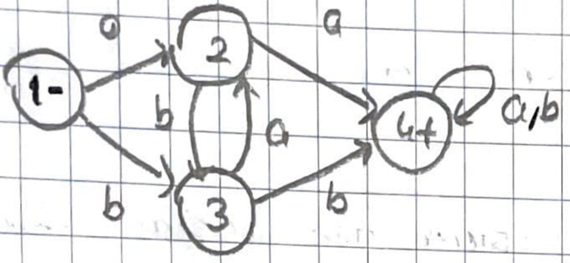
ÖRN a ile başlayan kelimeleri kabul eder

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

→ verilen dil için unique bir machine yoktur. Birden fazla olabilir.

⇒ Her dil için en az bir makine var mıdır? Yoktur

ÖRN Kuralı bulm. için makinede geçerli kelimeler önemli, Bu yüzden +'dan başlıyoruz ve -'ye doğru gidiyoruz.



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

ÖRN 3. harfi b olan kelimelerin çizmesi

* 4. soru, çöz ve çözüm yöntemi bul.

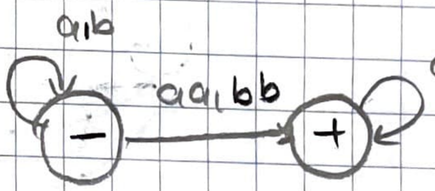
8/11/24

Decisions to make

Crash → kararsızlık

→ Aynı input crash edilebilir, except de edilebilir.

→ Eğer bir input crash ederse reject edilir.

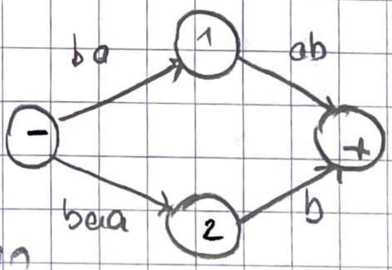


Input → baa
Accepted → (b)(aa)
Reject → (b)(a)(a)
crashed → (ba)(a) ?

→ Kişiden kişiye değişmemesi gerekiyor, net olmalı (well defined)

Change the Definition of The Abstract Machine

Bu makinenin ekstra yeteneği FA kıyasla : birden fazla kelime okuma



FA farklı old. için Transition Graphs