PAMUKKALE ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
2021 BAHAR

Biçimsel Diller ve Otomata Teorisi

Formal languages and automata theory

ALFABE ve KATAR-2

Alphabets and Languages

- Regular expression'lar bir dili doğrudan ∪ , o , * ile tanımlar.

Alphabets and Languages

- $\sum^*$ alfabesi üzerinde tanımlı regular expression'lar,

  $\sum \cup \{(,), \emptyset, \cup, *\}$ alfabesinde tanımlı string'lerdir.

- Bir regular expression aşağıdaki şekillerde elde edilir;

1. $\emptyset$ ve $\sum$ 'nın her elemanı regular expression'dır
2. Eger α ve β regular expression ise, $(\alpha\ \beta)$ regular expression'dır
3. Eger α ve β regular expression ise, $(\alpha \cup \beta)$ regular expression'dır
4. Eger α regular expression ise, α^* regular expression'dır
5. 1 ve 4 dışındaki hiçbir şey regular expression değildir.

Alphabets and Languages

- Eger α bir regular expression ise $L(\alpha)$, α tarafından tanımlanan dili ifade eder. L stringlerden dillere bir fonksiyondur.

- L fonksiyonu aşağıdaki şekillerde tanımlanabilir;

1. $L(\emptyset) = \emptyset$, ve $L(\alpha) = \alpha$, her $\alpha \in \sum$ için
2. α ve β regular expression ise, $L(\alpha\beta) = L(\alpha)L(\beta)$
3. α ve β regular expression ise, $L(\alpha \cup \beta) = L(\alpha) \cup L(\beta)$
4. Eger α regular expression ise, $L(\alpha^*) = L(\alpha)^*$

Alphabets and Languages

$L(((a\cup b)^{*}a)) \ = L((a\cup b)^{*})L(a)$

$= L((a\cup b)^{*})\{a\}$

$= L((a\cup b))^{*}\{a\}$

$= (L(a)\cup L(b))^{*}\{a\}$

$= (\{a\}\cup\{b\})^{*}\{a\}$

$= (a\cup b)^{*}a$

$= \{w\in \{a, b\}^{*} \mid w$ stringleri a ile biter$\}$

- $L(c^{*}(a\cup(bc^{*}))^{*})$ dili $\sum = \{a, b, c\}$ üzerinde tanımlı olsun. Özellikleri nelerdir?
  - ca sıralanışı olabilirmi ?
  - ac sıralanışı olabilirmi ?
  - iki tane a yanyana olabilirmi ?
  - cb sıralanışı olabilirmi ?
  - bc sıralanışı olabilirmi ?

Alphabets and Languages

- $L(0^* \cup (((0^*(1 \cup (11))) ((00^*)(1 \cup (11)))^*)0^*))$ dili

  $\sum = \{0, 1\}$ üzerinde tanımlı olsun. Bu dil bazı parantezleri kaldırarak

  $L(0^* \cup 0^*(1 \cup 11)(00^*(1 \cup 11))^* 0^*)$

  şeklinde kısaca gösterilebilir. Özellikleri nelerdir ?

- α regular expression tarafından $\sum$ alfabesi üzerinde tanımlanan L = L(a) dilleri

  regular languages (düzenli diller) olarak adlandırılır.

- $L = \{0^n 1^n : n \geq 0\}$ dili düzenli dil değildir !!!

Alphabets and Languages

- Bir w string'inin bir L diline ait olup olmadığını bulan algoritmaya

  language recognition device(dil tanıyan cihaz) denilmektedir.

Örnek: Aşağıdaki dili tanıyan bir cihaz nasıl işlem yapabilir?

$L = \{w \in \{0, 1\}^*$: *w string'i 111 substring'ine sahip olamaz*$\}$

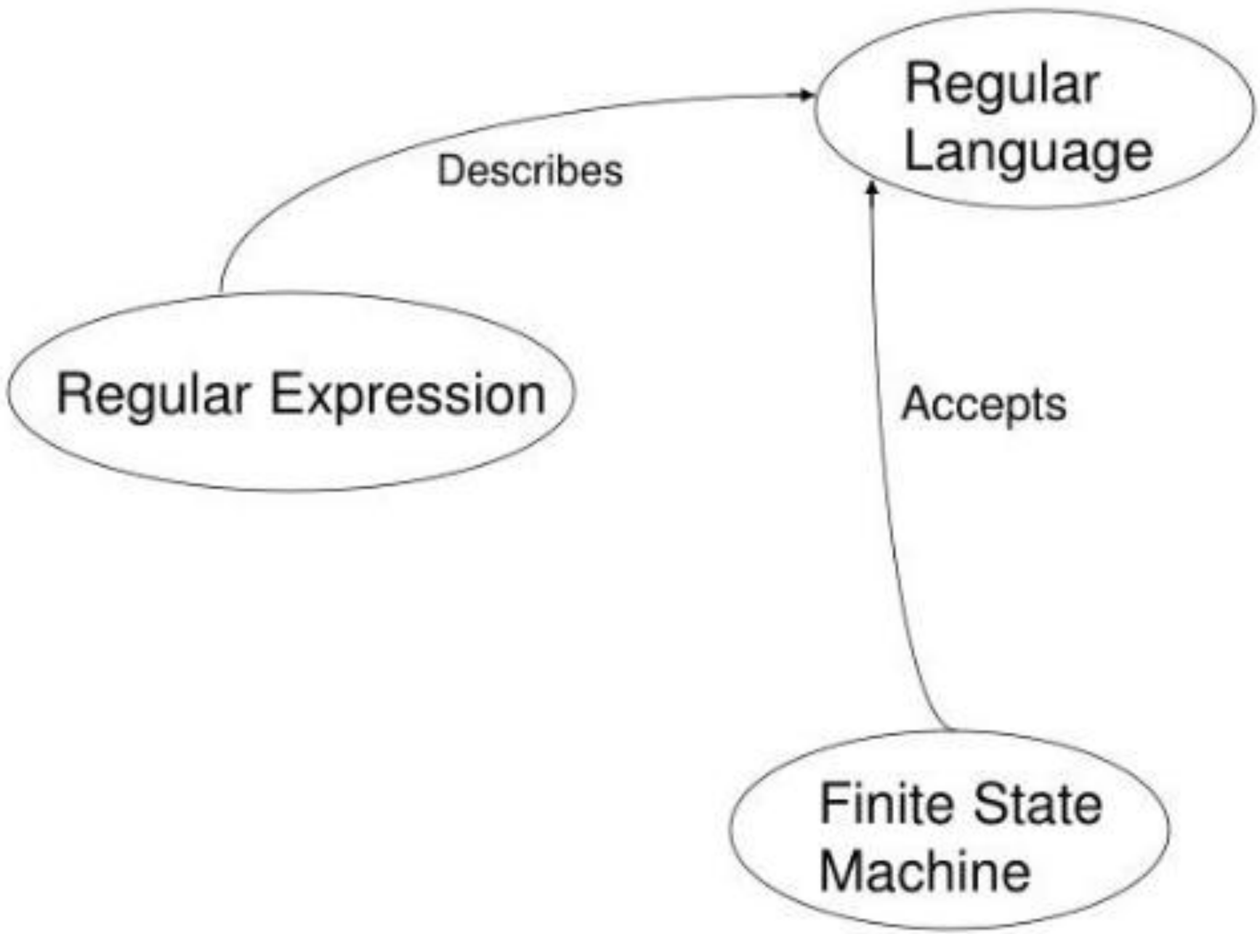
- Başlangıçta sayaç 0 yapılır ve string soldan saga dogru okunur
- Her 0 gelişinde sayaç sıfırlanır
- Her 1 gelişinde sayaç bir artırılır
- Sayaç degeri 3 oldugunda **Hayır** cevabıyla durur
- String tümüyle okundugunda sayaç üçten küçükse **Evet** cevabıyla durur.

- Bir dilin elemanları language generators(dil üreteci) tarafından oluşturulabilir.

  Dil üreteçleri algoritma degildir !

  $L((e \cup b \cup bb)(a \cup ab \cup abb)^*)$ dili nasıl oluşturulur ?

  $\cup$ işlemlerinin seçimi ne şekilde yapılır ?


Regular Language
Describes
Regular Expression
Accepts
Finite State Machine

Exercises

$\Sigma = \{a, b\}$, $L_1 = \{x \in \Sigma^*: |x| < 4\}$ ve $L_2 = \{aa, aaa, aaaa\}$ olsun. Aşağıdaki her bir L_i (i=3,4,5,6) dilinin elemanlarını listeleyiniz:

(a) $L_3 = L_1 \cup L_2$

(b) $L_4 = L_1 \cap L_2$

(c) $L_5 = L_1 L_4$

(d) $L_6 = L_1 - L_2$

•

Exercises

L ={ $a^n b^n c^m$ | n,m$\geq$ 0} dili verilsin. Aşağıdaki hangi katarlar bu dile aittir?

(a) ε

(b) ab

(c) c

(d) aabc

(e) aabbcc

(f) abbcc

Exercises

x bir katar ve α tek bir karakter olmak üzere

$(\alpha\ x)^R = x^R\ \alpha \quad \forall$ **x,**α olduğunu gösteriniz.

Proof: x'in uzunluğu üzerinden tümevarım uygulayarak bulabiliriz..

Eğer $|x| = 0$ (yani $x = \varepsilon$), ise $(a\varepsilon)^R = a = \varepsilon^R a$ olur.

Daha sonra n uzunlukta tüm x katarları için doğru kabul edip n+1 uzunluk için doğru olduğunu gösterelim:

n+ 1 uzunluğunda herhangi bir x katarını ele alalım. $|x| > 0$, x'i tek bir karakter b için yb olarak yeniden yazabiliriz.

$(ax)^R$	$= (ayb)^R$	x yerine yb yazdık
	$= b(ay)^R$	Tersten yazmanın tanımı
	$= b(y^R a)$	$\|y\| = n$ için doğru kabul etmiştik.
	$= (b\, y^R)\, a$	Kaynaştırmanın yer değiştirme özelliği
	$= x^R a$	Tersten yazmanın tanımı

Öyle ise $x = yb$ ise $x^R = by^R$ olur.

Exercise

- L={w∈ {a,b}*| |w| çifttir} dilini sağlayan düzenli ifadeyi yazınız:

L={w∈ {a,b}*| |w| çifttir} dilini sağlayan düzenli ifadeyi yazınız:

((aUb)(aUb))*

Veya

(aa U ab U ba U bb)*

Exercise

- L={w ϵ {a,b}*| w’da tek sayıda a olsun}

Kleene Star ile hep tek sayıda a içeren elde edebilir miyim? a* ={e,a,aa,aaa,aaaa,...}

Kleene Star ile hep çift sayıda a içeren elde edebilir miyim? (aa)* ={e,aa,aaaa,aaaaaa,...}

katar a ile de başlayabilir b ile de,

a’lar yan yana olmak zorunda değil!

- L={w ϵ {a,b}*| w'da tek sayıda a olsun}

b*(ab*ab*)*ab*

b*ab*(ab*ab*)*

Exercise

- L={w $\in$ {a,b}* | w’da birden fazla b olamaz}

- Yani ya bir ya da sıfır adet b olabilir
- Bunlar katarın herhangi bir yerinde olabilir

- L={w ϵ {a,b}* | w'da birden fazla b olamaz}

- Yani ya bir ya da sıfır adet b olabilir
- Bunlar katarın herhangi bir yerinde olabilir
- a*(eUb)a*
- a*Ua*ba*

Exercise

- $L=\{w \in \{a,b\}^* \mid w=a^{2n}b^{2m+1}, n \geq 0, m \geq 0\}$

- a ile başlıyor b ile bitiyor
- a'lar çift b'ler tek sayıda olmalı
- a ve/veya b olmaya da bilir.

- $L=\{w \in \{a,b\}^* \mid w=a^{2n}b^{2m+1}, n \geq 0, m \geq 0\}$

- a ile başlıyor b ile bitiyor
- a’lar çift b’ler tek sayıda olmalı
- a ve/veya b olmaya da bilir.
- (aa)*(bb)*b

- $a^*Ub^* \neq (a \cup b)^*$
- $(ab)^* \neq a^*b^*$

$\phi^*=\{e\}$ *ϕ: empty set* *e=empty string*

$e^*=e$

$e\circ r=er=r \quad r\circ e=re=r$

$r^*=(r\cup e)^*$

$r^{**}=r^*=r^*r^*$

$rr^*=r^*r=r^+$

$r^*r^+=r^+$

$(r^*)^+=(r^+)^*=((r^*)^+)^*=r^*$

$(r^*\cup s^*)^*=(r^*s^*)^*=(r\cup s)^*=(r^*s)^*r^*$

$(rs)^*r=r(sr)^*$

$r^+\cup r^*=r^*$

$r^+\cap r^*=r^+$

$(r\cup s^*)^*=(r^*\cup s)^*=(r^*\cup s^*)^*=(r\cup s)^*$

$(r\cup s)^*\neq(rs)^*$

$(r\cup s)^* \neq(r^*s)^*$

Exercise

Aşağıdakilerden her birini olabildiğince kısaca sözel olarak tanımlayın (başka bir deyişle, her bir düzenli ifade tarafından tanımlanan dili tanımlayın):

(a) L($((a^*a)\ b) \cup b$ **)**

(b) L($(((a^*b^*)^*ab) \cup ((a^*b^*)^*ba))(b \cup a)^*$ **)**

(a) Sıfır veya daha fazla a ve ardından tek bir b içeren herhangi bir a katarı.

$((a^*a)\ b) \cup b = \{(a^*a)\ b,\ \ b\} = \{ab, aab, aaab, \ldots, b\}$

(b) En az bir ab veya ba içeren a ve/veya b'lerin herhangi bir katarı.

$(((a^*b^*)^*ab) \cup ((a^*b^*)^*ba))\ (b \cup a)^*$

- $(a^*b^*)^* = (a \cup b)^*$

Exercise

- Bu düzenli ifadelerin her birini, aynı kümeyi temsil eden daha basit bir düzenli ifade olarak yeniden yazınız.

(a) $\emptyset^* \cup a^* \cup b^* \cup (a \cup b)^*$

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

(c) $(a^*b)^* \cup (b^*a)^*$

(a) $\emptyset^* \cup a^* \cup b^* \cup (a \cup b)^*$

$\emptyset^* = \{e\}$, ve $\varepsilon \subseteq (a \cup b)^*$.

$a^* \subseteq (a \cup b)^*$.

$b^* \subseteq (a \cup b)^*$.

Bu nedenle, ilk üç terim sonuncu terimin alt kümelerini tanımladığından, onları sonuncu terimle birleştirmek herhangi bir yeni eleman eklemez.

Bu durumda kısaca

$(a \cup b)^*$ yazabiliriz.

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

Bunu çözmek için, regular ifadeler için bazı denklikleri kullanacağız.

$((a^*b^*)^* (b^*a^*)^*)^* = ((a \cup b)^*(b \cup a)^*)^*$

$(A^*B^*)^* = (A \cup B)^*$ kullanırsak

$((a \cup b)^*(b \cup a)^*)^* = ((a \cup b)^*(a \cup b)^*)^*$

$A^*A^* = A^*$ kullanırsak

$((a \cup b)^*(a \cup b)^*)^*=((a \cup b)^*)^* =$

$(A^*)^* = A^*$ kullanırsak

$(a \cup b)^*$

(c) $(a^*b)^* \cup (b^*a)^*$

$(a^*b)^* \cup (b^*a)^* = (a \cup b)^*$ ({a, b} kümesindeki tüm katarlar.) ,

$(a^*b)^*$ katarı e'nin ve b ile biten tüm dizelerin birleşimidir.

$(b^*a)^*$ katarı e'nin ve a ile biten tüm dizelerin birleşimidir.

{a, b} üzerindeki herhangi bir katar ya boş olur ya da a veya b ile biter.

Öyleyse $(aUb)^*$ elde edilir.

Ödev

- Problemleri çözünüz 1.8.1, 1.8.2, 1.8.3, 1.8.5 (sayfa 51-52)