PAMUKKALE ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
2021 BAHAR

Biçimsel Diller ve Otomata Teorisi

Formal languages and automata theory

ALFABE ve KATAR

Alphabets and Languages

- Bir alfabe sonlu sayıda sembolden oluşur

Örnek: Roman alfabesi = {a, b,..., z} *Binary alfabe = {0, 1}*

- String bir alfabede tanımlanan sonlu sayıda sembolün sıralanışıyla elde edilir

"bilgisayar" Roman alfabesinde tanımlanmıştır.

"0111011" Binary alfabede tanımlanmıştır.

- Bir alfabedeki her sembol bir string'tir
- Sembol içermeyen string empty string olarak adlandırılır

Alphabets and Languages

- String olarak genellikle u, v, w, x, y, z harfleri kullanılacaktır. $w = abc$
- $\sum$ alfabesi için $\sum^*$ ise bu alfabede oluşturulan boş string'te dahil tüm string'lerin kümesini göstermektedir (*: Kleene Star)
- Bir string'in length (uzunluk) degeri $|w|$ şeklinde gösterilir

  $|101| = 3$, $|e| = 0$,
- w = bilgisayar, $w(3)$ = l, $w(1)$ = b
- concatenation iki string'in ardarda eklenmesidir

  $w = x \circ y$, $w = xy$

  $|w| = |x| + |y|$

  $w(j) = x(j)$, $j=1,\dots,|x|$ $w(|x| + j) = y(j)$, $j = 1,\dots,|y|$

  01 **o** 001 = 01001

  bilgisayar **o** mühendisligi = bilgisayarmühendisligi

  $w \circ e = e \circ w = w$

Concatenation (o) x oy=xy

- Kaynaştırma ilişkisellik özelliğine (associativity) sahiptir: x(yz)=(xy)z
- Kaynaştırma değişme özelliğine (commutativity) sahip değildir: xy ≠ yx
- Kaynaştırmanın birleşme üzerine dağılma özelliği vardır x(yuz)= xy u xz
- Kaynaştırmanın kesişim üzerine dağılma özelliği yoktur x(ynz) ≠xy n xz

Alphabets and Languages

- Bir string *v*, *w* string'i içinde substring olarak belirtilir.

  $w = xvy$, x ve y, *e (empty)* string olabilir

- Eger bazı *x*'ler için $w = xv$ ise string *v*, string *w* içinde suffix olarak adlandırılır.
- Eger bazı *y*'ler için $w = vy$ ise string *v*, string *w* içinde prefix olarak adlandırılır.

  road   *roadrunner* 'da prefix,   *abroad* ' da suffix ve

  broader 'da substring'tir

- w^i bir string'in i kez tekrarını gösterir

  $w^0 = e$ (empty string)

  $w^{i+1} = w^i \circ w, \quad i \geq 0$

  $ab^2 = abab$

  $ab^* = \{a, ab, abb, \ldots\}$

  $(ab)^* = \{e, ab, abab, ababab, \ldots\}$

Alphabets and Languages

- Bir string w için w^R reversal olarak adlandırılır.

  $w = bilgisayar$, $w^R = rayasiglib$

- Bir alfabe $\sum$ üzerinde tanımlı string kümesi, $\sum^*$ kümesinin altkümesidir ve language (dil) olarak adlandırılır.

- $\sum$, $\sum^*$ ve Ø birer dildir.

$\{aba, czr, d, f\}$ bir sonlu dildir ve $\sum = \{a, b, c, \ldots, z\}$ alfabesi üzerinde tanımlıdır

$\{0, 01, 011, 0111, \ldots\}$ bir sonsuz dildir ve $\sum = \{0, 1\}$ alfabesi üzerinde tanımlıdır

Alphabets and Languages

- Bir dil $L = \{w \in \sum^*: w\ string'i\ \ P\ özelligine\ sahiptir\}$ şeklinde tanımlanır

  $L = \{w \in \{0, 1\}^*: w\ eşit\ sayıda\ 1\ ve\ 0'a\ sahiptir\}$

  $L = \{w \in \sum^*: w = w^R\}$

- Bir alfabe'de tanımlı iki dil arasında kaynaştırma (concatenation) tanımlanabilir

  $L = L_1 \circ L_2 \qquad L = L_1L_2$

  $L = \{w \in \sum^*: w = x \circ y\ \ ve\ x \in L_1\ \ ve\ \ y \in L_2\}$ *Örnek:*

  $\sum = \{0, 1\}, \qquad L_1 = \{w \in \sum^*: w\ çift\ sayıda\ 0'a\ sahip\}$

  $L_2 = \{w \in \sum^*: w\ 0'la\ başlar\ ve\ 1'le\ devam\ eder\}$ $L_1 \circ L_2 = \{w \in \sum^*: w\ tek\ sayıda\ 0'a\ sahiptir\}$

Alphabets and Languages

- Kleen star işlemi L^* ile gösterilir ve bir dilde 0 veya daha fazla string'in concatenate edilmesiyle elde edilir.

  $L^* = \{w \in \sum{}^*: w = w_1 \circ ... \circ w_k \quad k \geq 0 \ \ ve \ \ w_1 \circ ... \circ w_k \in L\}$ *Örnek:*

  $L = \{01, 1, 100\}$

  $110001110011 \in L^*$ 1o100o01o1o100o1o1

- L^+ işlemi LL^* dilini ifade etmek için kullanılır.

  $L^+ = \{w \in \sum{}^*: w = w_1 \circ ... \circ w_k \quad k \geq 1 \ \ ve \ w_1 \circ ... \circ w_k \in L\}$

  L^+ dili, L dilinin closure'u olup L dilini ve L dilindeki string'lerin eklenmesiyle elde edilen tüm string'leri içeren bir dildir.

Alphabets and Languages

- $L = \{w \in \{0, 1\}^*$: *w iki veya üç tane 1 tekrarını içerir ve birinci ile ikinci ardarda gelemez*$\}$ şeklinde tanımlı bir dil olsun.

  Bu dil sadece singleton kümelerle ve ∪ , o , * ile ifade edilebilir.

  $L = \{0\}^* \text{o} \{1\} \text{o} \{0\}^* \text{o} \{0\} \text{o} \{1\} \text{o} \{0\}^* \text{o} ((\{1\} \text{o} \{0\}^*) \cup 0^*)$

  $L = 0^*10^*010^*(10^* \cup 0^*)$ şeklinde {, } ve o kaldırılarak kısaca gösterilebilir

- Yukarıdaki ifade regular expression (düzenli ifade) olarak adlandırılır. 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. Ø ve ∑ ' nın her elemanı regular expression'dır
  2. Eger α ve β regular expression ise, (α β) regular expression'dır
  3. Eger α ve β regular expression ise, (α ∪ β) regular expression'dır
  4. Eger α regular expression ise, α* regular expression'dır
  5. 1 ve 4 dışındaki hiçbir şey regular expression degildir.

Alphabets and Languages

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

- L fonksiyonu aşagıdaki şekillerde tanımlanabilir;

1. $L(\emptyset) = \emptyset$, ve $L(\alpha) = \alpha$, her $\alpha \in \Sigma$ 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

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

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

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

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

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

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

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

- $L(c^{*}(a\cup(bc^{*}))^{*})$ dili $\sum = \{a, b, c\}$ üzerinde tanımlı olsun. Özelikleri 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 tarafindan $\sum$ alfabesi üzerinde tanımlanan L = L(a) dilleri

  regular languages (düzenli diller) olarak adlandırılır.

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

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}*

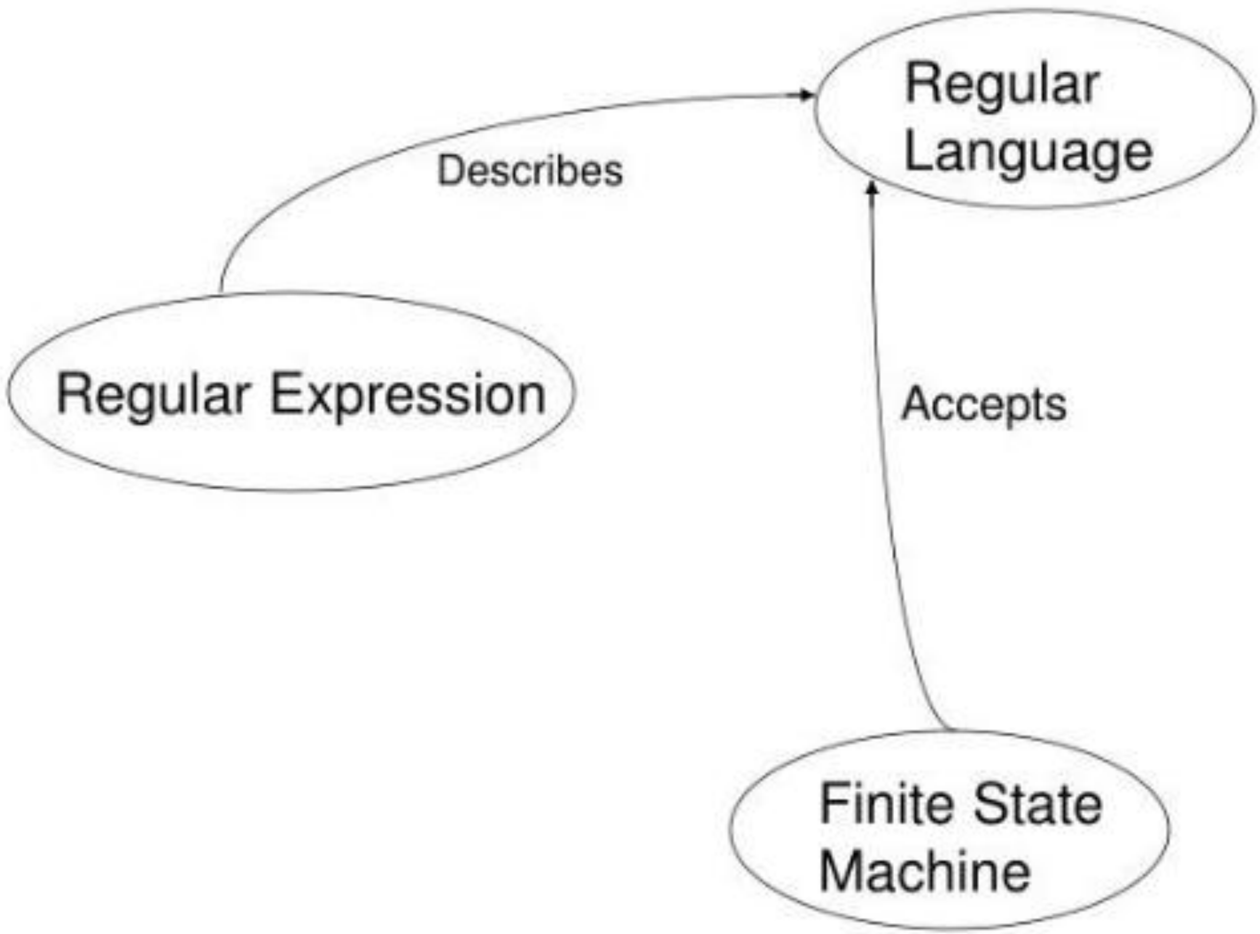
- Başlangıçta sayaç 0 yapılır ve string soldan saga dogru okunur
- Her 0 gelişinde sayaç sıfırlanır
- Her 1 gelişinde sayaç bir artırılır
- Sayaç degeri 3 oldugunda **Hayır** cevabıyla durur
- String tümüyle okundugunda sayaç üçten küçükse **Evet** cevabıyla durur.

- 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 Li (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

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

$(ax)^R = x^R a$ ∀ **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

- $L=\{w \in \{a,b\}^* \mid |w|$ çifttir$\}$ dilini sağlayan düzenli ifadeyi yazınız:

L={wϵ {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*Ua*ba*

Exercise

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

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

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

Ödev

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