

BM 305 Biçimsel Diller ve Otomatlar (Formal Languages and Automata)

Hazırlayan: M.Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Alphabets and Languages

- Bir **alfabe** sonlu sayıda sembolden oluşur
Roman alfabeti = $\{a, b, \dots, z\}$
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.
 $w = abc$
- Σ alfabe için Σ^* ise bu alfabede oluşturulan boş string'te dahil tüm string'lerin kümesini göstermektedir
- Bir string'in **length (uzunluk)** değeri $|w|$ şeklinde gösterilir
 $|101| = 3, \quad |e| = 0,$
- $w = \text{bilgisayar}, \quad w(3) = 1, \quad w(1) = b$
- **concatenation** iki string'in ardarda eklenmesidir
 $w = x \circ y, \quad w = xy$
 $|w| = |x| + |y|$
 $w(j) = x(j), \quad j=1, \dots, |x| \quad w(|x| + j) = y(j), \quad j = 1, \dots, |y|$
 $01 \circ 001 = 01001$
 $\text{bilgisayar} \circ \text{mühendisliği} = \text{bilgisayarmühendisliği}$
 $w \circ e = e \circ w = w$

Alphabets and Languages

- Bir string v , w string'i içinde **substring** olarak belirtilir.
 $w = xvy, \quad x \text{ ve } y, e \text{ string olabilir}$
- Eğer bazı x 'ler için $w = xv$ ise string v , string w içinde **suffix** olarak adlandırılır.
- Eğer bazı y 'ler için $w = vy$ ise string v , string w içinde **prefix** olarak adlandırılır.
 $road \quad roadrunner$ 'da prefix, $abroad$ ' da suffix ve $broader$ 'da substring'tir
- w^i bir string'in i kez tekrarını gösterir
 $w^0 = e$ empty string
 $w^{i+1} = w^i \circ w, \quad i \geq 0$
 $do^2 = dodo$

Alphabets and Languages

- Bir string w için w^R **reversal** olarak adlandırılır.
 $w = bilgisayar, w^R = rayasiglib$
- Bir alfabe Σ üzerinde tanımlı string kümesi, Σ^* kümesinin altkümesidir ve **language(dil)** olarak adlandırılır.
- Σ, Σ^* ve \emptyset birer dildir.
 $\{aba, czt, d, f\}$ bir sonlu dildir ve $\Sigma = \{a, b, c, \dots, z\}$ alfabesi üzerinde tanımlıdır
 $\{0, 01, 011, 0111, \dots\}$ bir sonsuz dildir ve $\Sigma = \{0, 1\}$ alfabesi üzerinde tanımlıdır

Alphabets and Languages

- Bir dil $L = \{w \in \Sigma^* : w \text{ string'i } P \text{ özelliğine sahiptir}\}$ şeklinde tanımlanır
 $L = \{w \in \{0, 1\}^* : w \text{ eşit sayıda 1 ve 0'a sahiptir}\}$
 $L = \{w \in \Sigma^* : w = w^R\}$
- Bir alfabe'de tanımlı iki dil arasında concatenation tanımlanabilir
 $L = L_1 \circ L_2 \quad L = L_1 L_2$
 $L = \{w \in \Sigma^* : w = x \circ y \text{ ve } x \in L_1 \text{ ve } y \in L_2\}$

Örnek:

$$\begin{aligned}\Sigma &= \{0, 1\}, & L_1 &= \{w \in \Sigma^* : w \text{ çift sayıda 0'a sahip}\} \\ L_2 &= \{w \in \Sigma^* : w \text{ 0'la başlar ve 1'le devam eder}\} \\ L_1 \circ L_2 &= \{w \in \Sigma^* : w \text{ tek sayıda 0'a sahiptir}\}\end{aligned}$$

Alphabets and Languages

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

$$L^* = \{w \in \Sigma^*: w = w_1 \circ \dots \circ w_k \quad k \geq 0 \quad \text{ve} \quad w_1 \circ \dots \circ w_k \in L\}$$

Örnek:

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

$$110001110011 \in L^* \qquad 1 \circ 100 \circ 01 \circ 1 \circ 100 \circ 1 \circ 1$$

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

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

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

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

Bu dil sadece singleton kümelerle ve $\cup, \circ, *$ ile ifade edilebilir.

$$L = \{0\}^* \circ \{1\} \circ \{0\}^* \circ \{0\} \circ \{1\} \circ \{0\}^* \circ ((\{1\} \circ \{0\}^*) \cup \emptyset^*)$$

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

- Yukarıdaki ifade **regular expression(düzenli ifade)** olarak adlandırılır. Regular expression'lar bir dili doğrudan \emptyset, \cup ve $*$ ile tanımlar.

Alphabets and Languages

- Σ^* alfabeti üzerinde tanımlı regular expression'lar, $\Sigma \cup \{ (,), \emptyset, \cup, * \}$ alfabetinde tanımlı string'lerdir.
- Bir regular expression aşağıdaki şekillerde elde edilir;
 1. \emptyset ve Σ 'nin her elemanı regular expression'dır
 2. Eğer α ve β regular expression ise, $(\alpha\beta)^*$ de regular expression'dır
 3. Eğer α ve β regular expression ise, $(\alpha \cup \beta)^*$ de regular expression'dır
 4. Eğer α regular expression ise, α^* da regular expression'dır
 5. 1 ve 4 dışındaki hiçbir şey regular expression değildir.

Alphabets and Languages

- Eğer α bir regular expression ise $\mathcal{L}(\alpha)$, α tarafından tanımlanan dili ifade eder. \mathcal{L} stringlerden dillere bir fonksiyondur.
- \mathcal{L} fonksiyonu aşağıdaki şekillerde tanımlanabilir;
 1. $\mathcal{L}(\emptyset) = \emptyset$, ve $\mathcal{L}(a) = a$, her $a \in \Sigma$ için
 2. α ve β regular expression ise, $\mathcal{L}(\alpha\beta) = \mathcal{L}(\alpha)\mathcal{L}(\beta)$
 3. α ve β regular expression ise, $\mathcal{L}(\alpha \cup \beta) = \mathcal{L}(\alpha) \cup \mathcal{L}(\beta)$
 4. Eğer α regular expression ise, $\mathcal{L}(\alpha^*) = \mathcal{L}(\alpha)^*$

Alphabets and Languages

$$\begin{aligned}
 \mathcal{L}(((a \cup b)^* a)) &= \mathcal{L}((a \cup b)^*) \mathcal{L}(a) \\
 &= \mathcal{L}((a \cup b)^*) \{a\} \\
 &= \mathcal{L}((a \cup b))^* \{a\} \\
 &= (\mathcal{L}(a) \cup \mathcal{L}(b))^* \{a\} \\
 &= (\{a\} \cup \{b\})^* \{a\} \\
 &= (a \cup b)^* a \\
 &= \{w \in \{a, b\}^* : w \text{ stringleri } a \text{ ile biter}\}
 \end{aligned}$$

- $\mathcal{L}(c^*(a \cup (bc^*))^*)$ dili $\Sigma = \{a, b, c\}$ üzerinde tanımlı olsun. Özellikleri nelerdir ?
 - ca sıralanışı olabilir mi ?
 - ac sıralanışı olabilir mi ?
 - İki tane a yanyana olabilir mi ?
 - cb sıralanışı olabilir mi ?
 - bc sıralanışı olabilir mi ?

Alphabets and Languages

- $\mathcal{L}(0^* \cup ((0^*(1 \cup (11))) ((00^*)(1 \cup (11)))^* 0^*))$ dili $\Sigma = \{0, 1\}$ üzerinde tanımlı olsun. Bu dil bazı parantezleri kaldırarak $\mathcal{L}(0^* \cup 0^*(1 \cup 11)(00^*(1 \cup 11))^* 0^*)$ şeklinde kısaca gösterilebilir. Özellikleri nelerdir ?
- α regular expression tarafından Σ alfabesi üzerinde tanımlanan $L = \mathcal{L}(\alpha)$ 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?

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

- Başlangıçta sayaç 0 yapılır ve string soldan sağa doğru okunur
 - Her 0 gelişinde sayaç sıfırlanır
 - Her 1 gelişinde sayaç bir artırılır
 - Sayaç değeri 3 olduğunda **Hayır** cevabıyla durur
 - String tümüyle okunduğunda sayaç üçten küçükse **Evet** cevabıyla durur.
- Bir dilin elemanları language generators(dil üretici) tarafından oluşturulabilir. Dil üreticileri algoritma değildir !

$\mathcal{L}((e \cup b \cup bb)(a \cup ab \cup abb)^*)$ dili nasıl oluşturulur ? \cup işlemlerinin seçimi ne şekilde yapılır ?

Ödev

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