

Bil 2114 Otomata Teorisi Çalışma Soruları ve Cevapları–III (Hafta 7,8,9)

1. Formal olarak $G = (\{S\}, \{a, b\}, R, S)$ ve R türetim kuralları

$$\begin{aligned}S &\rightarrow aSa \\S &\rightarrow bSb \\S &\rightarrow \varepsilon\end{aligned}$$

olarak verilen grammerinin türettiği dili bulunuz.

Cevap: R kurallarını kullanarak rastgele kelimeler türetip, türetilen kelimelerin ortak özelliğini tespit edip bu kelimelerin oluşturduğu dili bulmaya çalışalım.

$$S \Rightarrow aSa \Rightarrow aaSaa \Rightarrow aabSbaa \Rightarrow aabbbaa$$

$$S \Rightarrow bSb \Rightarrow baSab \Rightarrow baab$$

$$S \Rightarrow aSa \Rightarrow abSba \Rightarrow abaSaba \Rightarrow ababSbaba \Rightarrow ababbaba$$

Dikkat edilirse yukarıda üretilen kelimeler iki yarımдан oluşuyor, öyleki ikinci yarımdeki kelime ilk yarımdeki kelimenin tersten yazılmış hali.

Su halde bu grammer tarafından üretilen dil:

$$L(G) = \{ww^R \mid w \in \{a, b\}^*\}$$

Bu dil aynı zamanda çift uzunluktaki palindromların dilidir.

Not 1. Burada w^R , w kelimesinin tersten (reverse) yazılmış halidir.

Not 2. Yukarıdaki kelimeleri türetim ağacı oluşturarak da üretebiliriz (bakın 6. Hafta notları 17 ve 18 nolu slidelar).

2. Formal olarak $G = (\{S\}, \{a, b\}, R, S)$ ve R türetim kuralları

$$\begin{aligned}S &\rightarrow aSb \\S &\rightarrow SS \\S &\rightarrow \varepsilon\end{aligned}$$

olarak verilen grammerinin türettiği dili bulunuz.

Cevap: Bir önceki sorudaki gibi R kurallarını kullanarak rastgele kelimeler türetelim ve bu kelimelerin oluşturduğu dili bulmaya çalışalım.

$$S \Rightarrow aSb \Rightarrow aSSb \Rightarrow aaSbb \Rightarrow aaSSbb \Rightarrow aaaSbbb \Rightarrow aaabbb$$

$$S \Rightarrow aSb \Rightarrow ab$$

$$\begin{aligned}S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaSSbb \Rightarrow aaSSaSbbb \Rightarrow aaaSbaSSbbb \Rightarrow aaabaaSbbbb \\ \Rightarrow aaabaabbbb\end{aligned}$$

Turetilen kelimelerin ortak ozelligi ayni sayida a ve b harfi icermesidir.

Su halde bu grammer tarafından uretilen dil:

$$L(G) = \{w \in \{a, b\}^* | n_a(w) = n_b(w)\}$$

Burada $n_a(w)$ w kelimesindeki toplam a harfi sayisini gostermektedir.

3. $L = \{a^n b^m | 0 \leq n \leq m \leq 2n\}$ dilini tureten grammeri bulunuz ve formal olarak gosteriniz.

Cevap: Bu dilin kelimelerinde a harflerini b harfleri takip eder.

b harflerinin sayisi a harflerinin sayisina esit olabilir ($n \leq m$). Bunun icin asagidaki kurali yazabiliriz:

$$S \rightarrow aSb$$

Ayrıca b harflerinin sayisi a harflerinin sayisinin iki katina esit olabilir ($m \leq 2n$). Yani her bir a harfi icin iki tane b harfi yazabiliriz. Bunun icin asagidaki kurali yazabiliriz:

$$S \rightarrow aSbb$$

Son olarak, non-terminal terim olan S sonlandirmak (terminate etmek) icin asagidaki kurala ihtiyacimiz var:

$$S \rightarrow \varepsilon$$

Sonuc olarak soruda verilen L dilini tureten G grammeri

$$G = (\{S\}, \{a, b\}, R, S) \text{ ve } R \text{ kurallari:}$$

$$S \rightarrow aSb$$

$$S \rightarrow aSbb$$

$$S \rightarrow \varepsilon$$

sekinde verilir.

4. $\Sigma = \{a, b\}$ alfabesi kullanilarak olusturulan ve icinde her zaman iki tane b harfi olan kelimelerin dili icin bir duzenli ifade bulunuz ve bu dili tureten grammerin kurallarini veriniz.

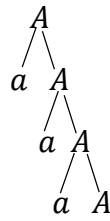
Cevap: İcinde iki tane b harfi olan kelimelerde bu iki b harfinin arasina, soluna yada sagina istedigimiz sayida a gelebilir. Su halde duzenli ifade soyle olur:

$$R = a^* b a^* b a^*$$

Bu dilin grammer kurallarini uretelim. Istedigimiz kadar a harfi uretmek icin asagidaki kurali kullanabiliriz:

$$A \rightarrow aA$$

Bu kural rekürsif olarak sonsuz sayıda a uretebilir:



A non-terminalini, uretilen kelimelerin genel formunda (duzenli ifade ile gosterilen) yazarak asagidaki kurali olusturabiliriz:

$$S \rightarrow AbAbA$$

Son olarak A non-terminalinin sonlanmasi icin asagidaki kurali ekleyelim:

$$A \rightarrow \varepsilon$$

Bu duzenli ifadeye karsilik gelen dili ureten grammerin kurallari:

$$S \rightarrow AbAbA$$

$$A \rightarrow aA$$

$$A \rightarrow \varepsilon$$

5. Formal olarak $G = (\{S, O\}, \{a, b\}, R, S)$ ve R türetim kuralları

$$S \rightarrow aO|bO|\varepsilon$$

$$O \rightarrow aS|bS$$

olarak verilen grammerinin türettiği dili bulunuz.

Cevap: S başlangıç nonterminalinden başlayarak ve yukarıda verilen kuralları kullanarak rastgele kelimeler türetelim.

$$S \Rightarrow aO \Rightarrow aaS \Rightarrow aabO \Rightarrow aabaS \Rightarrow aaba$$

$$S \Rightarrow bO \Rightarrow baS \Rightarrow babO \Rightarrow babbS \Rightarrow babbbo \Rightarrow babbbaS \Rightarrow babbba$$

$$S \Rightarrow aO \Rightarrow abS \Rightarrow ab$$

Yukarıda türetilen kelimelerin ortak özelliği çift uzunlukta olmasıdır. Su halde verilen G grammeri ile türetilen dil:

$$L(G) = \{w \in \{a, b\}^* \mid |w| = 2n, n \geq 0\}$$

Ekstra Soru. $\Sigma = \{a, b\}$ alfabeti kullanılarak oluşturulan çift uzunluktaki kelimelerin dilini üreten grammerin kurallarını veriniz.

Cevap: İki tane non-terminal tanımlayalım: Kelimede tek sayıda terminal olduğu durumda T non-terminali, çift sayıda terminal olduğu durumda S non-terminali kelimeye olsun. Çift sayıda terminal olduğu durumda, kelimenin uzunluğu çift olacağından bu kelimeleri kabul

edebiliriz (örneğin $w = aabbabS$). O halde S nonterminalini sonlandırabiliriz. Bunun için aşağıdaki kuralı yazabiliriz:

$$S \rightarrow \varepsilon$$

Kelimede tek sayıda terminal varken, kelimeye bir tane a yada bir tane b terminali eklediğimizde kelime çift uzunlukta olur. Tek sayıda terminal olduğu durumda kelime T non-terminali bulunuyordu. O halde T varken a yada b eklendiğinde kelimeye, kelimeye çift sayıda terminal olur bu ise S nonterminalinin olmasını gerektirir. Bu durumlar için aşağıdaki kuralları ekleyebiliriz:

$$S \rightarrow aO$$

$$S \rightarrow bO$$

Yukarıdakine benzer şekilde, çift sayıda terminal varken (S varken) bir tane a yada bir tane b terminali eklediğimizde kelime tek uzunlukta olur bu ise O nonterminalinin olmasını gerektirir. Bu durumlar için aşağıdaki kuralları ekleyebiliriz:

$$O \rightarrow aS$$

$$O \rightarrow bS$$

Sonuç olarak elde ettiğimiz kuralları şu şekilde özetleyebiliriz:

$$S \rightarrow aO|bO|\varepsilon$$

$$O \rightarrow aS|bS$$

6. Formal olarak $G = (\{A, B, C, S\}, \{a, b\}, R, S)$ ve R türetim kuralları

$$S \rightarrow AB|BB$$

$$A \rightarrow CC|AB|a$$

$$B \rightarrow BB|CA|b$$

$$C \rightarrow BA|AA|b$$

olarak verilen G grammerinin $w = aabb$ kelimesini türetebileceğini CKY algoritmasını kullanarak gösteriniz.

Cevap:

CKY algoritması kullanarak verilen bir kelimenin grammer tarafından türetildiğine karar verilirken aşağıdaki gibi bir piramit oluştururuz ve bu piramiti tabandan başlayarak yukarı doğru doldürürüz. Eğer başlangıç nonterminali S piramitin en tepesine ulaşırsa verilen kelimenin grammer tarafından türetildiği sonucuna ulaşırız.

4	aabb / S,A,B,C			
3	aab / A,C	abb / S,A,C		
2	aa / C	ab / S,A	bb / S,A,B	
1	a / A	a / A	b / B,C	a / B,C
	a	a	b	b

1. $a \leftarrow A, b \leftarrow B, C$

2. $aa \leftarrow AA \leftarrow C$

$ab \leftarrow (A)(B, C) = AB, AC \leftarrow S, A$

$bb \leftarrow (B, C)(B, C) = BB, BC, CB, CC \leftarrow S, A, B$

3. $aab; (aa, b)$ ve (a, ab) şeklinde ayrıştırılır.

$aa, b \leftarrow (C)(B, C) = CB, CC \leftarrow A$ $a, ab \leftarrow (A)(S, A) = AS, AA \leftarrow C$

$abb; (ab, b)$ ve (a, bb) şeklinde ayrıştırılır.

$ab, b \leftarrow (S, A)(B, C) = SB, SC, AB, AC \leftarrow A$ $a, bb \leftarrow (A)(S, A, B) = AS, AA, AB \leftarrow S, C$

4. $aabb; (aab, b), (aa, bb)$ ve (a, abb) şeklinde ayrıştırılır.

$aab, b \leftarrow (A, C)(B, C) = AB, AC, CB, CC \leftarrow S, A$

$aa, bb \leftarrow (C)(S, A, B) = CS, CA, CB \leftarrow B$

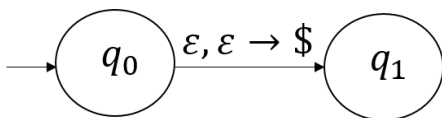
$a, abb \leftarrow (A)(S, A, C) \leftarrow AS, AA, AC \leftarrow C$

7. $L = \{a^n b^m \mid n > m, m \geq 1\}$ dilinin kelimelerini kabul eden bir pushdown otomata dizayn ediniz.

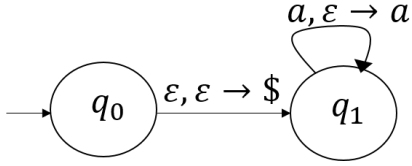
Cevap:

PDO'yu aşamalar halinde dizayn edelim.

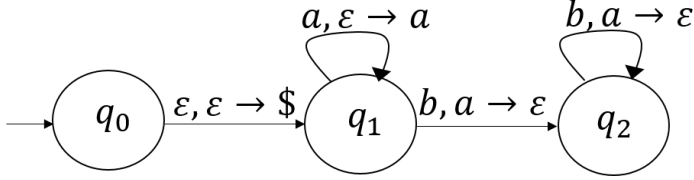
Aşama 1: Yığına \$ sembolü ekleyelim, böylece yigindan \$ sembolünü çıkarattığımızda yiginin boş kaldığını anlarız.



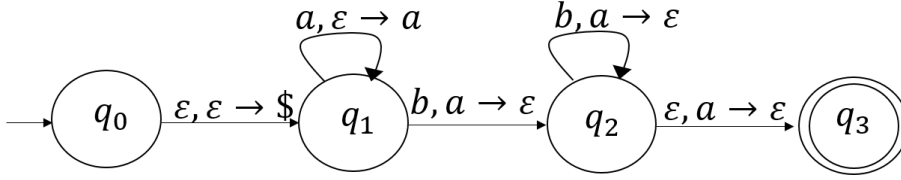
Aşama 2: Kelimeden a harfi okudukça (yigindan bir şey çıkarmadan) bunu yığına ekleyelim.



Asama 3: Kelimeden b harfi okumaya basladigimizda; okunan her b harfi için yigindan bir a harfi cikartalim.

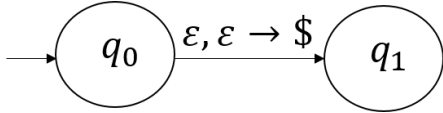


Asama 4: Bu dilin kelimelerindeki toplam a harfi sayisi, toplam b harfi sayisindan fazla olmalidir. Yani okunan her bir b için yigindan bir a cikarttigimizda yiginda hala a harfinin kalmis olmasi gerekir. Boyle bir durumda final durumuna gidebiliriz.

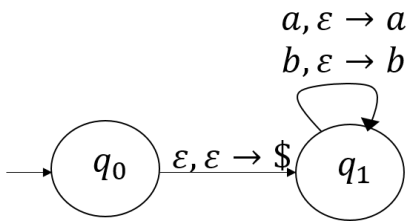


8. $L = \{wcw^R \mid w \in \{a, b\}^*\}$ dilinin kelimelerini kabul eden bir pushdown otomata dizayn ediniz.

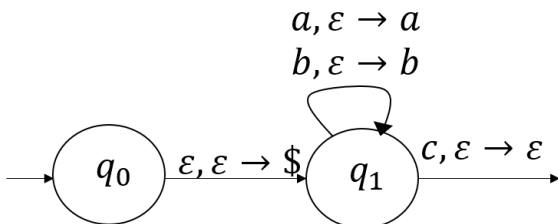
Cevap: Asama 1: Yığına $\$$ sembolü ekleyelim.



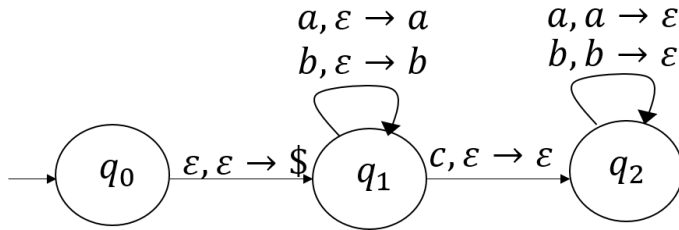
Asama 2: Kelimeden a ve b harfleri okudukça yigina ekleyelim.



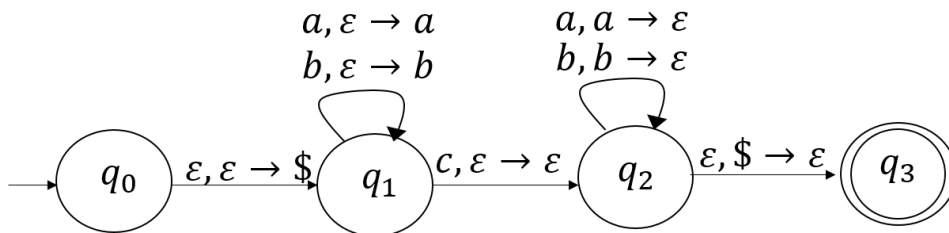
Asama 3: Kelimeden c harfini okuduğumuzda kelimenin ortasına gelmişiz demektir. Su halde bu harfi okuduğumuzda yigina bir sey ekleyip-cikarmadan bir sonraki duruma gecelim.



Asama 4: Kelimeden okunan a harfi için, eger o anda yiginin en ustunde a harfi varsa bu iki a birbirne eslenir ve yigindan cikarilir. Ayni sey b harfi için de geçerlidir.



Asama 5: Eger kelimenin ikinci yarısındaki a lar ile ilk yarısındaki a lar ile birebir eslesmisse ve ayni sekilde kelimenin ikinci yarısındaki b ler ile ilk yarısındaki b ler ile birebir eslesmisse yiginda yalnız $\$$ sembolü kalir. Bu durumda kelimeyi kabul ederiz, final durumuna gideriz.



9. $M = (\{q_0, q_1, q_2, q_3\}, \{a, b, c\}, \{\$, c\}, \delta, q_0, \{q_3\})$ ve geçişleri:

$$\delta(q_0, c, \epsilon) = (q_1, \$)$$

$$\delta(q_1, a, \epsilon) = (q_1, c)$$

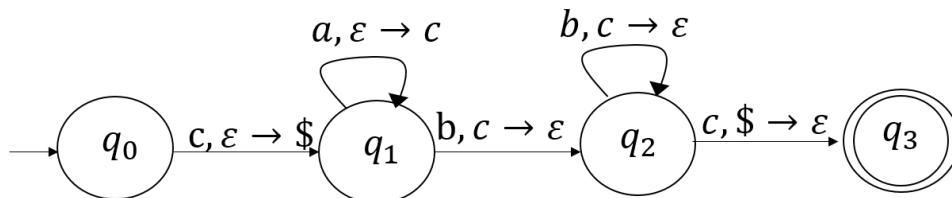
$$\delta(q_1, b, c) = (q_2, \epsilon)$$

$$\delta(q_2, b, c) = (q_2, \epsilon)$$

$$\delta(q_2, c, \$) = (q_3, \epsilon)$$

sekinde verilen pushdown otomatayı (PDO) çiziniz ve bu PDO'nun tanıdığı dili formal olarak gösteriniz.

Cevap:



Bu PDO kelimeden bir c harfi okuyarak final durumuna doğru hareket eder. Bu durumda dilin kelimeleri c harfi ile başlamalıdır. $L = \{c\}$

PDO q_1 'de iken a harfi okur ve okuduğu her bir a için bir c harfi yigina koyar. Demekki c harfi ile başlayan dilin kelimeleri belirli bir sayıda a harfi ile devam eder. $L = \{ca^n | n \geq 1\}$

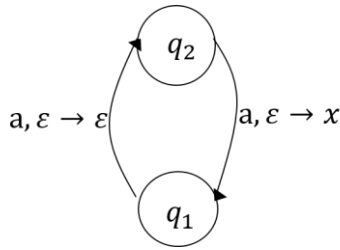
a harflerinden sonra b harfleri gelmesi durumunda, okunan her bir b için yiginden bir c cikartilir. Bu işlemler sonucunda yiginda yalnızca $\$$ kalmisa tum a ile tum b ler birbiryile eslesmistir demektir. Yani toplam a harfi kadar, toplam b harfi vardir. $L = \{ca^n b^n | n \geq 1\}$

Son olarak kelimden bir c harfi okumamiz durumunda yiginden $\$$ cikartip kabul durumuna gidiyoruz. O halde kabul edilecek kelimeler c harfi ile bitmelidir. $L = \{ca^n b^n c | n \geq 1\}$

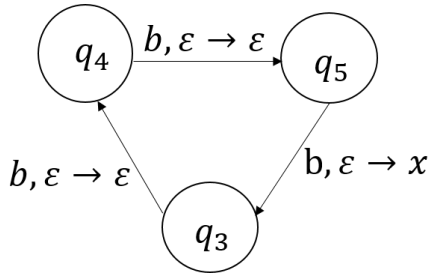
10. $L = \{a^{2n} b^{3n} | n \geq 0\}$ dilinin kelimelerini kabul eden bir pushdown otomata dizayn ediniz.

Cevap: Okudugumuz her iki tane a harfine karsilik uc tane b harfi okumaliyiz.

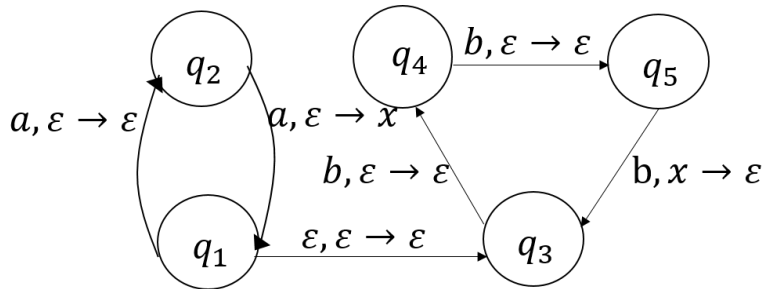
Okudugumuz her iki a harfi için yigina bir tane x sembolü koyalım:



Okudugumuz her uc b harfi için yigina bir tane x sembolü koyalım:

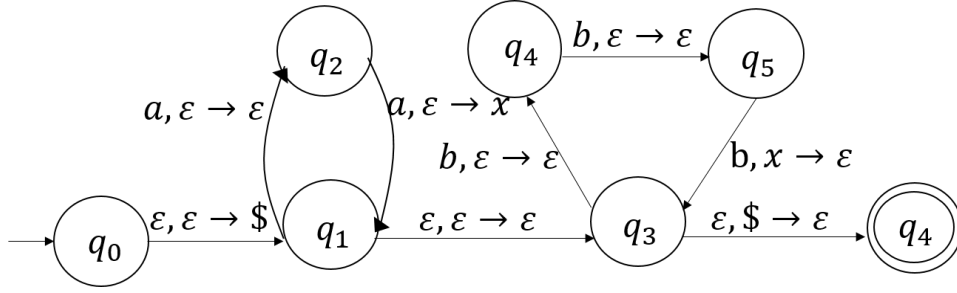


Bu iki yapıyı birbirine bağlayalım.



Burada $\epsilon, \epsilon \rightarrow \epsilon$ gecisiyle nondeterminizmden faydalanıyoruz. Aslında q_1 durumuna her vardigimizda $\epsilon, \epsilon \rightarrow \epsilon$ geçişi sayesinde q_3 'e de varmis oluruz. Fakat q_3 'de a oku olmadigi için a okunmasi durumunda q_3 'e varan kol ilerlemez. Ancak b okumaya basladigimizda bu kol da calisir. Nondeterminizm sayesinde bir anlamda istediğimiz zaman birinci yapıdan ikici yapıya gecebiliyormusuz gibi düşünebiliriz.

Son olarak baslangic durumunda yigina $\$$ sembolunu ekleyelim ve q_3 durumunda iken yigindaki tum x 'lerin cikarilmasi halinde (bu her iki tane a harfine karsilik uc tane b harfi okunduğunu gösterir) final durumuna gidelim.

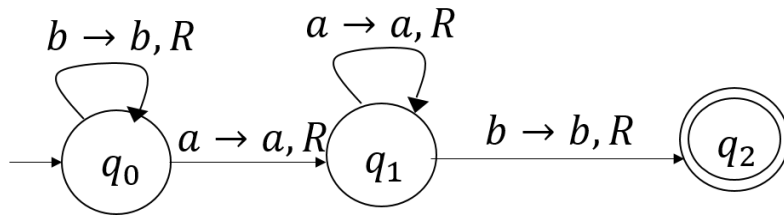


11. $\Sigma = \{a, b\}$ alfabeti kullanılarak oluşturulan ve içinde 'ab' altkelimesi geçen kelimelerin dilini düzenli ifade ile gösteriniz ve bu dili tanıyan bir Turing makinesi dizayn ediniz.

Cevap:

Düzenli ifade: $R = (a \cup b)^* ab (a \cup b)^*$

Turing Makinesi:



12. İkili sistemde (binary) ifade edilen sayılardan tek sayıları tanıyan Turing makinesi dizayn ediniz.

Cevap: Tek sayılar olan 1, 3, 5, 7 ... nin ikili sistemde değerleri sırasıyla 1, 11, 101, 111, ... dir. Dikkat edilirse ikili sistemde gösterilen tek sayıların son basamağı 1'dir. Demekki bu sayıları Turing makinesinin bandına yazdığımızda sayı bittikten sonraki ilk B sembolünün hemen solunda 1 harfi olacaktır. Bu halde stratejimiz kelimeyi okumaya başladıktan sonra B'yi bulana kadar sağa doğru ilerlemek, B'yi bulduğumuzda sola geçip bu hücredeki harfin 1 olup olmadığını kontrol etmek, eğer 1 ise verilen kelimeyi kabul etmektir. Bu uhalde Turing makinesi aşağıdaki gibi olur.

