Hafta 9

Sağ-Doğrusal ve Sol-Doğrusal Dilbilgisi

Yeniden yazma kuralları

$$\mathbf{A \Rightarrow wB}$$

$$\mathbf{A \Rightarrow w} \quad : \quad \mathbf{A, B \in V_N} \quad , \quad \mathbf{w \in V_T^*}$$

biçiminde olan dilbilgisine **sağ-doğrusal (right-linear)** dilbilgisi denir. Sağ-doğrusal dilbilgisinin yeniden yazma kurallarının sol tarafında bir değişken, sağ tarafında ise bir uç simgeler dizgisi ya da bir uç simgeler dizgisi ile bir değişken yer alır. Uç simgeler dizgisi sıfır uzunluğunda bir dizgi de olabilir.

A→ Bw

A→ w :A,B $\in V_N$, w$\in V_T^*$

biçiminde olan dilbilgisine **sol-doğrusal (left-linear)** dilbilgisi denir. Sol-doğrusal dilbilgisinin yeniden yazma kurallarının sol tarafında bir değişken, sağ tarafında ise bir uç simgeler dizgisi ya da bir değişken ile bir uç simgeler dizgisi yer alır. Uç simgeler dizgisi sıfır uzunluğunda bir dizgi de olabilir.

Sağ-doğrusal ve sol-doğrusal dilbilgileri tarafından türetilen diller düzgün dillerdir. Tür-3 dilbilgileri tarafından türetilen diller de düzgün diller olduğuna göre tür-3, sağ-doğrusal ve sol-doğrusal dilbilgileri denk dilbilgileridir. Tür-3, sağ-doğrusal ve sol-doğrusal dilbilgilerinin tümü düzgün dilbilgileri; türettikleri diller ise düzgün diller olarak adlandırılır.

Düzgün dilbilgileri için verilen ve birbirine denk olduğu belirtilen üç biçimden, tür-3 sağ-doğrusalın özel bir biçimidir. Başka bir deyişle her tür-3 dilbilgisi aynı zamanda sağ-doğrusaldır. Sağ-doğrusal her dilbilgisi de, ek değişkenler kullanılarak, kolaylıkla tür-3 dilbilgisine dönüştürülebilir. Sağ-doğrusal dilbilgisini tür-3 dilbilgisine dönüştürmek için aşağıdaki kurallar yazılır.

$$A \Rightarrow a_1\, a_2 \; \ldots .\, a_n \quad \text{yerine} \quad \left\{ \begin{array}{l} A \Rightarrow a_1\, A_1 \\ A_1 \Rightarrow a_2\, A_2 \\ \ldots\ldots\ldots\ldots \\ A_{n\text{-}1} \Rightarrow a_n \end{array} \right.$$

$$A \Rightarrow a_1\, a_2 \; \ldots .\, a_n\, B \quad \text{yerine de} \quad \left\{ \begin{array}{l} A \Rightarrow a_1\, A_1 \\ A_1 \Rightarrow a_2\, A_2 \\ \ldots\ldots\ldots\ldots \\ A_{n\text{-}1} \Rightarrow a_n\, B \end{array} \right.$$

Bu nedenle tür-3 ve sağ-doğrusal dilbilgilerinin denk olduğu açıktır. Sol-doğrusal ve sağ-doğrusal dilbilgilerinin denkliği ise bu kadar açık değildir. Ancak düzgün dilbilgisinin üç biçimi (tür-3, sağ-doğrusal ve sol-doğrusal) tarafından türetilen dillerin sonlu özdevinirler (FA) tarafından tanınan düzgün diller olduğu gösterildiğinden dolaylı olarak bu üç biçimin eşdeğerliği de ispatlanmış olmaktadır.

Örnek:

$G_{3.6} = < V_N, V_T, P, S >$

$V_N = \{ S, A, B \}$

$V_T = \{0, 1\}$

$P: \quad S \Rightarrow 0A$

$\qquad A \Rightarrow 10A \mid \lambda$

$G_{3.6}$ sağ-doğrusal bir dilbilgisidir. Bu dilbilgisitarafından türetilen tümcelerden birkaçını bulalım:

$S \Rightarrow 0A \Rightarrow 0$

$S \Rightarrow 0A \Rightarrow 010A \Rightarrow 010$

$S \Rightarrow 0A \Rightarrow 010A \Rightarrow 01010A \Rightarrow 01010$

Yukarıdaki örneklerin incelenmesinden, **L($G_{3.6}$)** dilinin aşağıdaki gibi tanımlanabileceği görülür:

L($G_{3.6}$) = **$0(10)^* = (01)^*0$**

$G_{3.6}$ tarafından türetilen ve yukardaki düzgün deyimlerle gösterilen dil, aşağıdaki sol-doğrusal (**$G'_{3.6}$**) ve tür-3 **($G''_{3.6}$)** dilbilgileri tarafından da türetilir. Dolayısıyla **$G_{3.6}$, $G'_{3.6}$** ve **$G''_{3.6}$** aynı dili türeten, denk dilbilgileridir.

$$G_{3.6} = < V_N, V_T, P, S >$$

$$V_N = \{ S, A, B \}$$

$$V_T = \{0, 1\}$$

$$P: \quad S \Rightarrow 0A$$

$$A \Rightarrow 10A \mid \lambda$$

$$G'_{3.6} = < V_N, V_T, P, S >$$

$$V_N = \{ S \}$$

$$V_T = \{0, 1\}$$

$$P: \quad S \Rightarrow S10 \mid 0$$

$$G''_{3.6} = < V_N, V_T, P, S >$$

$$V_N = \{ S, A, B \}$$

$$V_T = \{0, 1\}$$

$$P: \quad S \Rightarrow 0A$$

$$A \Rightarrow 1B \mid \lambda$$

$$B \Rightarrow 0A$$

Düzgün Dil Sonlu Özdevinir İlişkisi

Düzgün Dilbilgisinin Türettiği Dili Tanıyan Sonlu Özdevinirin Bulunması

Teorem 3.1. Bir düzgün dilbilgisi verildiğinde, bu dilbilgisinin türettiği dili tanıyan bir sonlu özdevinir bulunabilir.

Sonlu özdevinirler tarafından tanınan kümelerin düzgün kümeler olduğu tanımlanmıştı. Teorem 3.1'de ise düzgün dilbilgileri tarafından türetilen dillerin sonlu özdevinirler tarafından tanındığı belirtilmektedir. Burdan, düzgün dilbilgileri tarafından türetilen dillerin (düzgün dillerin) düzgün kümeler olduğu sonucu çıkmaktadır.

Teorem 3.1, bir düzgün dilbilgisi verildiğinde, bu dilbilgisinin türettiği dili tanıyan sonlu özdevinirin nasıl bulunacağı gösterilerek ispat edilecektir. Düzgün dilbilgileri için üç farklı biçim olduğu için de, dilbilgisine eşdeğer sonlu özdevinirin nasıl bulunacağı tür-3, sağ-doğrusal ve sol-doğrusal dilbilgileri için ayrı ayrı gösterilecektir.

Tür-3 Dilbilgisinin Türettiği Dili Tanıyan Sonlu Özdevinirin Bulunması

Bir tür-3 dilbilgisi **(G)** verildiğinde, **T(M) = L(G)** eşitliğini sağlayan sonlu özdevinir **(M)** aşağıdaki gibi bulunur.

$G = < V_N, V_T, P, S >$

$M = < Q, \Sigma, q_0, \delta, F >$

$Q = V_N \cup \{C\}$ C, V_N'de bulunmayan yeni bir değişken

$\Sigma = V_T$

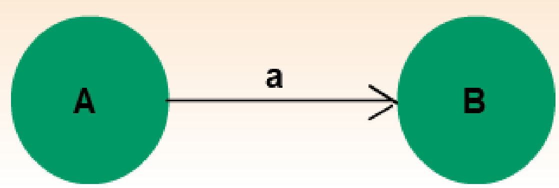
$q_0 = S$

$F = \{C\}$ eğer L(G) λ'yı içermiyorsa

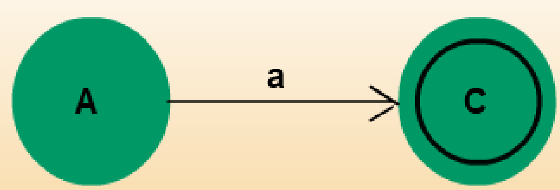
$= \{C, S\}$ eğer L(G) λ'yı içeriyorsa

δ: Sonlu özdevinirin durum geçişleri, dilbilgisinin yeniden yazma kurallarından aşağıdaki gibi türetilir.

$A \Rightarrow aB$ için $\delta(A, a) = B$



$A \Rightarrow a$ için $\delta(A, a) = C$



Örnek:

Dilbilgisi:

$G_{3.7} = \langle V_N, V_T, P, S \rangle$

$V_N = \{S, A, B\}$

$V_T = \{0, 1\}$

$P: \quad S \Rightarrow 0S \mid 1A$

$A \Rightarrow 0B$

$B \Rightarrow 0B \mid 1S \mid 1$

FA:

$G_{3.7}$ tür-3 bir dilbilgisidir. Bu dilbilgisinin türettiği dili tanıyan **FA** aşağıdaki gibi bulunur:

$M_{3.7} = \langle Q, \Sigma, q_0, \delta, F \rangle$

$Q = \{S, A, B, C\}$

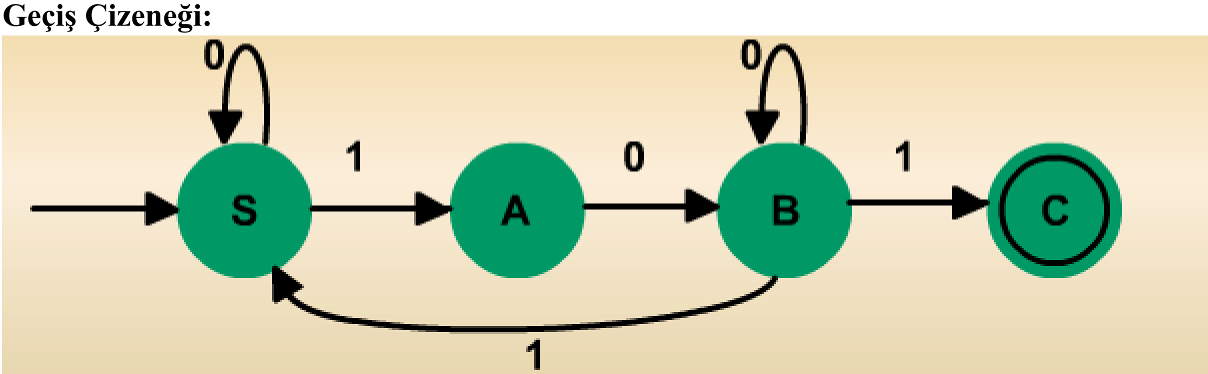
$\Sigma = \{0, 1\}$

$q_0 = S$

$F = \{C\}$

δ : Makinenin durum geçişleri Çizim 3.1'de görülmektedir.

Geçiş Çizeneği:



Çizim 3.1. $G_{3.7}$'nin türettiği Dili Tanıyan $M_{3.7}$'nin Geçiş Çizeneği

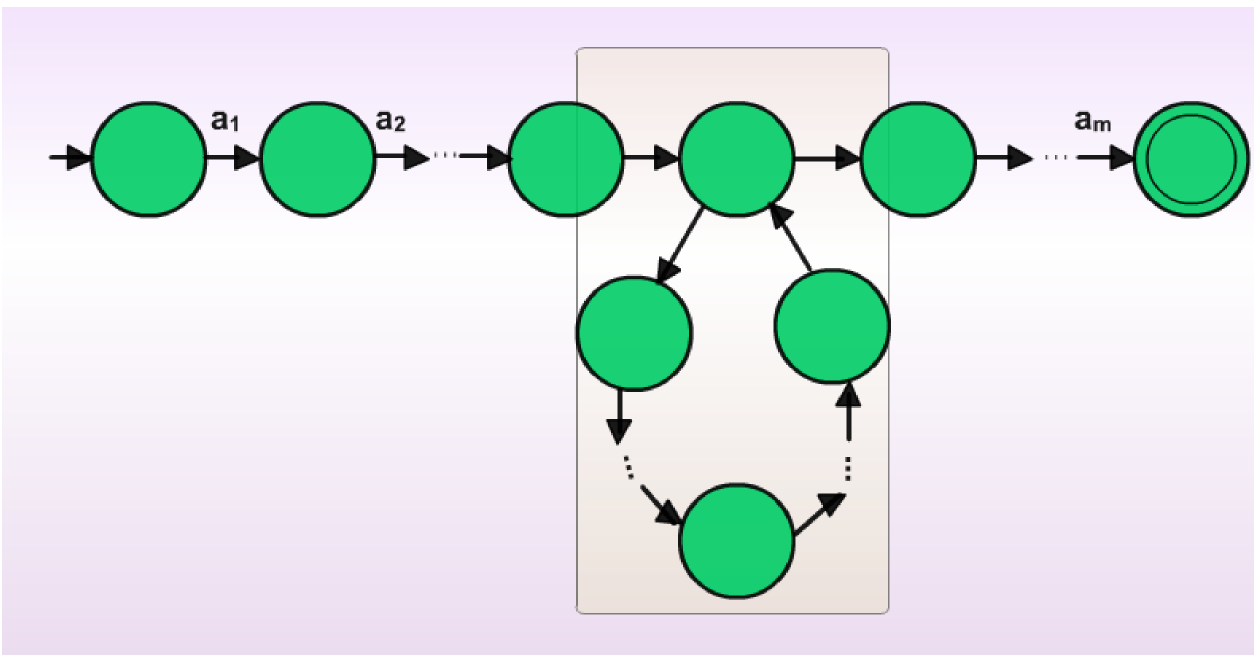
DÜZGÜN DİLLERİN KİMİ ÖZELLİKLERİ

Pumping Lemma

L düzgün bir dil olsun. L'yi tanıyan bir ya da, kendi aralarında denk birçok sonlu özdevinir vardır. L'yi tanıyan özdevinirlerden durum sayısı en küçük olanın (M) durum sayısının **n** olduğunu varsayalım. Eğer, L'nin sözcüklerinden birinin uzunluğu **n**'den büyük ise:

$$\mathbf{w} = \mathbf{a_1 a_2 a_3 \dots a_m} \qquad |\mathbf{w}| = \mathbf{m} > \mathbf{n}$$

M **w**'yi tanırken en az bir durumdan en az iki kere geçer. Başka bir deyişle, M'de **w**'ye karşı gelen yol mutlaka döngülüdür.



Bu durumda M

$$\mathbf{w} = \mathbf{u\,v^i\,t} \qquad \mathbf{i} \geq \mathbf{0}$$

biçimindeki tüm tümceleri tanır ve bu tümcelerin tümü L'de yer alır. Dolayısıyla L sonsuz sayıda tümce içeren bir dildir.

Birçok dilin düzgün olmadığını ispatlamak için kullanılabilen ve pumping lemma olarak bilinen bu özellik şu şekilde özetlenebilir:

“Her düzgün dil için öyle bir tamsayı **(n)** vardır ki, eğer dilde uzunluğu bu değerden büyük bir tümce **(w)** varsa, bu tümce

$$\mathbf{w} = \mathbf{u\,v\,t}$$

biçiminde yazılabilir ve dil

$$\mathbf{w} = \mathbf{u\,v^{i}\,t} \qquad \mathbf{i} \geq \mathbf{0}$$

biçimindeki tüm tümceleri içerir."

Bir dilin düzenli ifade (Regular expression) olup olmadığının belirlenmesi için kullanılan pumping lemma). Basitçe düzenli ifadede olup olmadığı sınanacak bir w dili için (yani L = w için) **w= xyz** şeklinde bir açılım kullanılarak test edilir. Test sırasında aşağıdaki koşulların sağlanması beklenir:

1. **$|y| \geq 1$**
2. **$|xy| \leq p$**
3. **bütün $i \geq 0$ için, $xy^iz \in L$**

Yukarıdaki üç şartı da sağlayan bir dil için düzenli ifadedir denilebilir. Ayrıca bu dilin düzenli ifade olabilmesi için yukarıdaki y teriminin üstü olan i değeri istenildiği kadar arttırılabilmeli ve elde edilen sonuç yine düzenli ifade (Regular expression) olmalıdır.

Daha basit bir ifadeyle, bir dilden üretilen bir terim üç parçaya bölündüğünde ortasındaki parça boş olmamalı, istenildiği kadar tekrarlanabilmeli ve çıkan bütün sonuçlar yine aynı dilden olmalıdır.

Örnek:

$L = \{a^n b^n : n \geq 0\}$ dilinin bir düzenli ifade olmadığının ispatıdır.

Dilimizi pumping lemma kalıbına olan $w = xyz$ şekline benzetmeye çalışalım. Bu durumda dilimizdeki üretilebilen en kısa terim için $w = a^p b^p$ olduğu söylenebilir ve bu terimi kalıba benzetirken xy için a ve z için b olduğunu söyleyebiliriz. Çünkü y'nin sıfır boyutunda olamayacağını biliyoruz, bu durumda y değeri ya a ya da b olmalıdır. İki durumda birbirinin aynısıdır. Bu durumda

y=a

z=b

olarak kabul edelim.

Şimdi sonucumuzu pumping lemma ile geliştirelim ve bir sonraki beklentimiz olan xy^2z terimini üretelim. Bu durumda çıktımız $\mathbf{w = a^2b}$ olacaktır. Dikkat edilirse bu yeni terim, dil tanımımız olan $\boldsymbol{L = \{a^n b^n : n \geq 0\}}$ tanımı ile çelişmektedir ve bu dilin bir üyesi değildir.

Dolayısıyla $L = \{a^n b^n : n \geq 0\}$ dilinin bir düzenli ifade olmadığı veya düzenli ifade olarak yazılamayacağı ispatlanmış olur.