

Özdevinirler Kuramı ve Biçimsel Diller

Bölüm 4 : Bağlamdan-Bağımsız Dilbilgisi ve Diller

4.1. Bağlamdan-Bağımsız Dilbilgisi

➤ Bağlamdan-bağımsız dilbilgisi ve diller ile bu dilleri tanıyan makine modeli programlama dilleri, derleyiciler, yorumlayıcılar, sözdizim çözümleyiciler, aritmetik deyim çözümleyiciler, ..vb. birçok yazılım bileşeninin bünyesinde yer alır.

➤ Bağlamdan-bağımsız dilbilgisi örnekleri:

➤ Örnek 4.1.

$$G_{4.1} = \langle V_N, V_T, P, S \rangle$$

$$V_N = \{ S \}$$

$$V_T = \{ +, -, *, /, (,), v, c \}$$

$$P : S \Rightarrow S + S \mid S - S \mid S * S \mid S / S \mid (S) \mid v \mid c$$

➤ $L(G_{4.1})$ dili, 4 temel aritmetik işlem (+, -, *, /), değişkenler (v) , değişmezler (c) ve parantezlerden oluşan geçerli aritmetik ifadelerin oluşturduğu kümedir.

Bölüm 4 : Bağlamdan-Bağımsız Dilbilgisi ve Diller

➤ Örnek 4.2.

$$G_{4.2} = \langle V_N, V_T, P, S \rangle$$

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

$$V_T = \{ a, b, c, d \}$$

$$P : S \Rightarrow aAdd$$

$$A \Rightarrow aAd \mid Ad \mid bBcc$$

$$B \Rightarrow bBc \mid Bc \mid \lambda$$

➤ $L(G_{4.2})$ dili aşağıdaki gibi tanımlanabilir:

$$L(G_{4.2}) = \{ a^n b^m c^k d^p \mid n, m, p, k \geq 1, p > n, k > m \}$$

➤ Örnek 4.3.

$$G_{4.3} = \langle V_N, V_T, P, S \rangle$$

$$V_N = \{ S, X, Y, Z \}$$

$$V_T = \{ a, b, c \}$$

$$P : S \Rightarrow XY$$

$$X \Rightarrow aXbb \mid aZbb \mid abb$$

$$Y \Rightarrow cY \mid c$$

$$Z \Rightarrow Zb \mid Xb$$

➤ $L(G_{4.3})$ dili aşağıdaki gibi tanımlanabilir:

$$L(G_{4.3}) = \{ a^n b^m c^k \mid n \geq 1, k \geq 1, m \geq 2n \}$$

Bölüm 4 : Bağlamdan-Bağımsız Dilbilgisi ve Diller

➤ Örnek 4.4.

$$G_{4.4} = \langle V_N, V_T, P, S \rangle$$

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

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

$$P: \quad S \Rightarrow 0S0 \mid 1S1 \mid 0A0 \mid 1A1$$

$$A \Rightarrow 0A1 \mid 1A0 \mid 01 \mid 10$$

➤ $L(G_{4.4})$ dili aşağıdaki gibi tanımlanabilir:

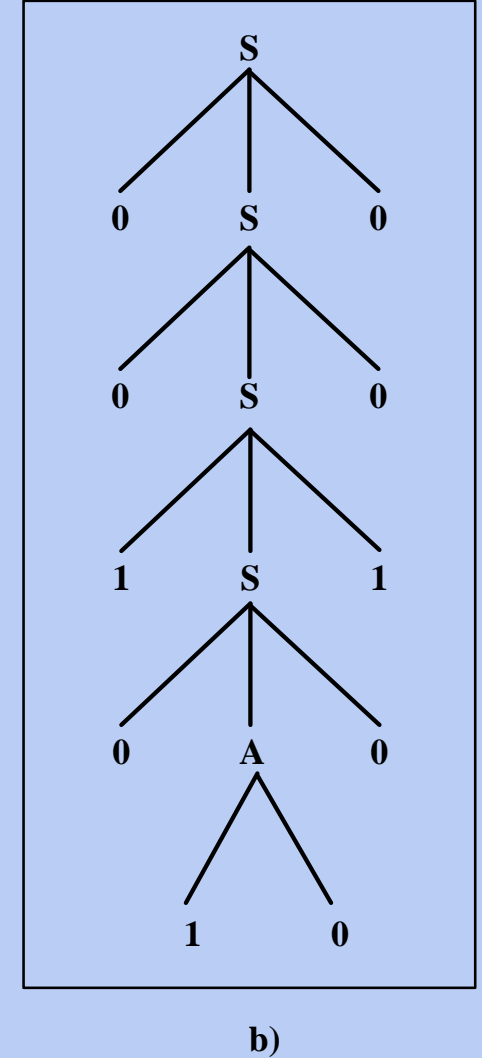
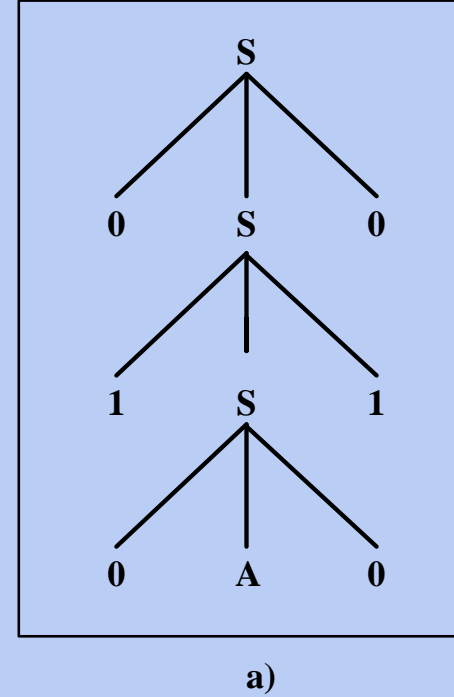
$$L(G_{4.4}) = \{ u v (v')^R u^R \mid u, v \in (0 + 1)^*, |u| \geq 1, |v| \geq 1 \}$$

Yukarıdaki gösterimde u^R u 'nun tersine, $(v')^R$ ise v nin tümlerinin tersine eşittir.

Bölüm 4 : Bağlamdan-Bağımsız Dilbilgisi ve Diller

4.2. Türetme Ağacı

- Bağlamdan-bağımsız dillerin her tümcesel yapısına ve her tümcesine bir türetme ya da ayrıştırma ağacı (*derivation or parsing tree*) karşı gelir.
- Örneğin $G_{4,4}$ tarafından türetilen:
 $\beta = 010A010$ tümcesel yapısı ile
 $w = 0010100100$ tümcesine karşı gelen türetme ağaçları yanda görülmektedir.



Bölüm 4 : Bağlamdan-Bağımsız Dilbilgisi ve Diller

➤ Türetme ya da Ayırıştırma Ağacının Tanımı

1. Ağacın kökünün etiketi S dir.
2. Kök dışındaki ara düğümlerin etiketleri sözdizim değişkenleridir ($A \in V_N$)
3. Eğer ağaç bir tümcesel yapıya karşı geliyorsa, yaprakların etiketleri sözdizim değişkenleri ya da uç simgeler olabilir ($X \in V$). Eğer ağaç bir tümceye karşı geliyorsa yaprakların etiketleri yalnız uç simgeler ($a \in V_T$) olabilir.
4. Eğer bir ara düğümün etiketi A , bu ara düğümün hemen altındaki düğümlerin etiketleri ise soldan sağa $X_1, X_2, X_3, \dots, X_k$ ise, dilbilgisinin yeniden yazma kuralları arasında
$$A \Rightarrow X_1 X_2 X_3 \dots X_k \quad X_1, X_2, X_3, \dots, X_k \in V$$
kuralı yer almalıdır.
5. Eğer bir düğümün etiketi λ ise, bu düğüm bir uç düğüm (yaprak) olmalı ve bu düğümün kardeşi bulunmamalıdır.

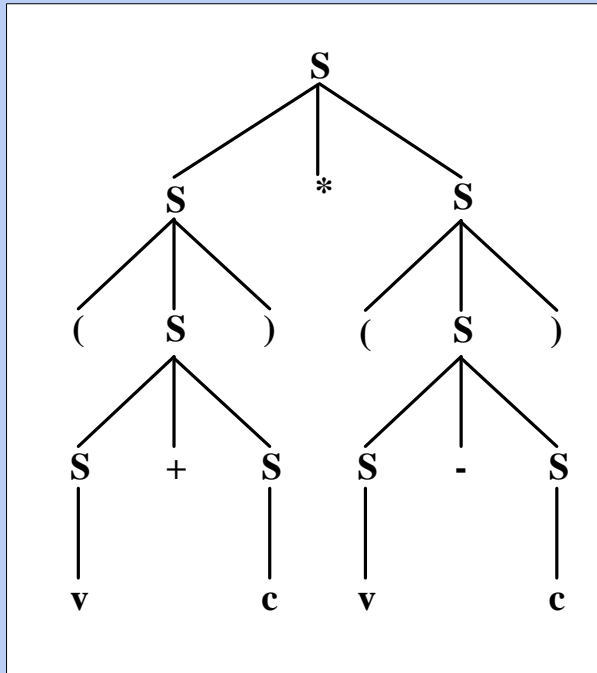
Bölüm 4 : Bağlamdan-Bağımsız Dilbilgisi ve Diller

➤ Soldan ve Sağdan Türetme

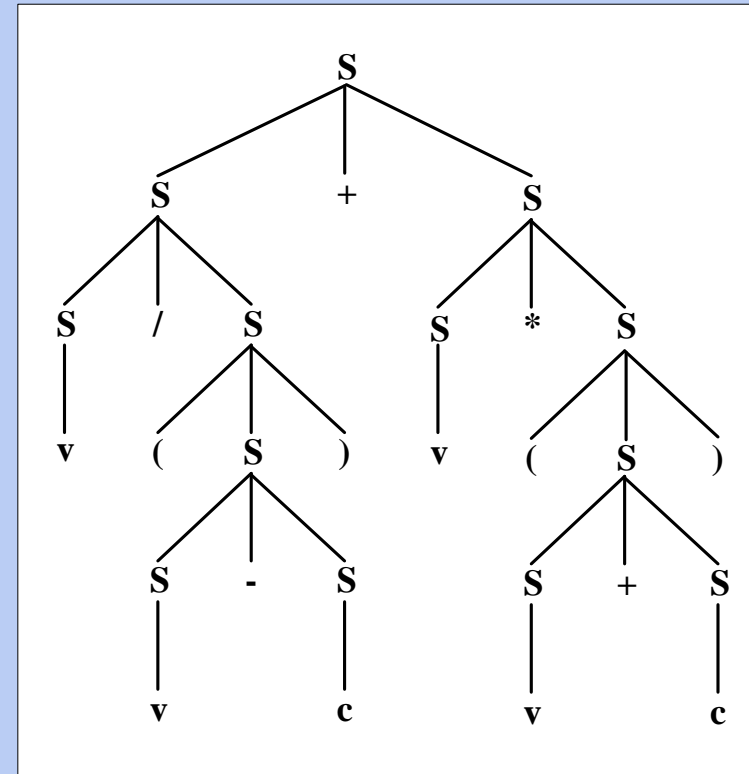
➤ w_1 ve w_2 $G_{4.1}$ ile türetilmiş iki tümcedir.

$$w_1 = (v + c)^*(v - c)$$

$$w_2 = v/(v - c) + v^*(v + c)$$



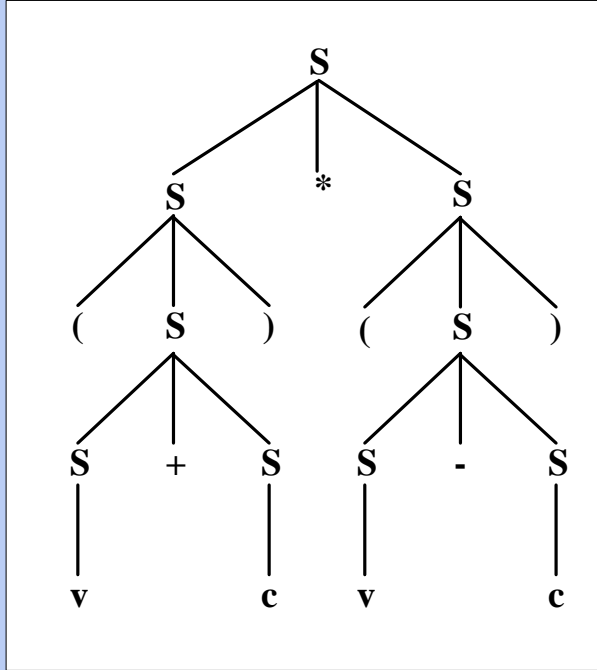
a)



b)

Bölüm 4 : Bağlamdan-Bağımsız Dilbilgisi ve Diller

➤ Soldan ve Sağdan Türetme



a)

➤ w_1 in soldan türetilmesi:

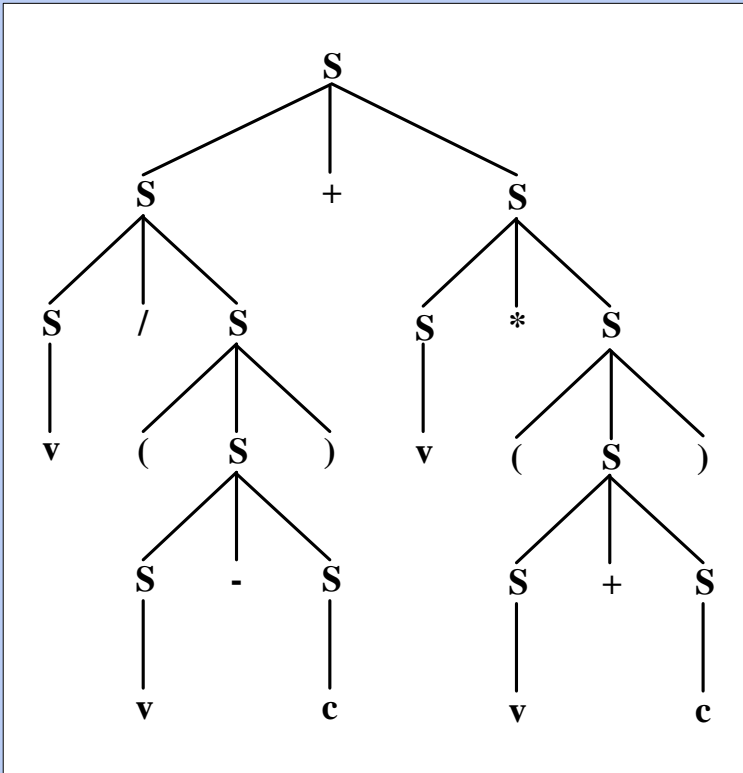
$$\begin{aligned} S &\Rightarrow \mathbf{S} * S \\ &\Rightarrow (\mathbf{S}) * S \\ &\Rightarrow (\mathbf{S} + S) * S \\ &\Rightarrow (v + \mathbf{S}) * S \\ &\Rightarrow (v + c) * \mathbf{S} \\ &\Rightarrow (v + c) * (\mathbf{S}) \\ &\Rightarrow (v + c) * (\mathbf{S} - S) \\ &\Rightarrow (v + c) * (v - \mathbf{S}) \\ &\Rightarrow (v + c) * (v - c) \end{aligned}$$

➤ w_1 in sağdan türetilmesi:

$$\begin{aligned} S &\Rightarrow S * \mathbf{S} \\ &\Rightarrow S * (\mathbf{S}) \\ &\Rightarrow S * (S - \mathbf{S}) \\ &\Rightarrow S * (\mathbf{S} - c) \\ &\Rightarrow \mathbf{S} * (v - c) \\ &\Rightarrow (\mathbf{S}) * (v - c) \\ &\Rightarrow (S + \mathbf{S}) * (v - c) \\ &\Rightarrow (\mathbf{S} + c) * (v - c) \\ &\Rightarrow (v + c) * (v - c) \end{aligned}$$

Bölüm 4 : Bağlamdan-Bağımsız Dilbilgisi ve Diller

➤ Soldan ve Sağdan Türetme



b)

➤ w_2 nin soldan türetilmesi:

$$\begin{aligned} S &\Rightarrow S + S \\ &\Rightarrow S / S + S \\ &\Rightarrow v / S + S \\ &\Rightarrow v / (S) + S \\ &\Rightarrow v / (S - S) + S \\ &\Rightarrow v / (v - S) + S \\ &\Rightarrow v / (v - c) + S \\ &\Rightarrow v / (v - c) + S * S \\ &\Rightarrow v / (v - c) + v * S \\ &\Rightarrow v / (v - c) + v * (S + S) \\ &\Rightarrow v / (v - c) + v * (v + S) \\ &\Rightarrow v / (v - c) + v * (v + c) \end{aligned}$$

➤ w_2 nin sağdan türetilmesi:

$$\begin{aligned} S &\Rightarrow S + S \\ &\Rightarrow S + S * S \\ &\Rightarrow S + S * (S) \\ &\Rightarrow S + S * (S + S) \\ &\Rightarrow S + S * (S + c) \\ &\Rightarrow S + S * (v + c) \\ &\Rightarrow S + v * (v + c) \\ &\Rightarrow S / S + v * (v + c) \\ &\Rightarrow S / (S) + v * (v + c) \\ &\Rightarrow S / (S - S) + v * (v + c) \\ &\Rightarrow S / (S - c) + v * (v + c) \\ &\Rightarrow v / (v - c) + v * (v + c) \end{aligned}$$

Bölüm 4 : Bağlamdan-Bağımsız Dilbilgisi ve Diller

4.3. Dilbilgisinin Yalınlaştırılması

Bağlamdan bağımsız bir dilbilgisi verildiğinde, bu dilbilgisinin daha az değişken ve kural içeren, ve belirli biçimdeki kuralları içermeyen eşdeğer bir dilbilgisine dönüştürülmesine dilbilgisinin yalınlaştırılması denir.

4.3.1. Özyineli Kural, Özyineli Değişken

- $A \Rightarrow \alpha_1 A \alpha_2$ Özyineli değişken
- $A \Rightarrow A \alpha_2$ Doğrudan özyineli değişken

4.3.2. Yok Edilebilir Değişken

- $A \Rightarrow \lambda$ ya da
- $A \xRightarrow{*} \lambda$ ise A yok edilebilir bir değişkendir.

Bölüm 4 : Bağlamdan-Bağımsız Dilbilgisi ve Diller

➤ Algoritma 4.1. Yok edilebilir değişkenlerin bulunması

1. $T_L = \{ A \mid (A \Rightarrow \lambda) \in P \} ;$

2. $T_{Eski} = \Phi ;$

3. $(T_L = T_{Eski})$ oluncaya kadar aşağıdaki işlemleri tekrarla:

3.1. $T_{Eski} = T_L ;$

3.2. (V_N) 'deki her değişken (A) için aşağıdaki işlemleri tekrarla:

eğer en az bir $(A \Rightarrow \alpha)$ kuralı için $\alpha \in (T_L)^*$ ise

A 'yı (T_L) 'ye ekle;

son 3.2;

son 3;

Bölüm 4 : Bağlamdan-Bağımsız Dilbilgisi ve Diller

➤ **Algoritma 4.2.** Başlangıç değişkeni dışında yok edilebilir değişkeni bulunmayan eşdeğer dilbilgisinin bulunması

1. P'deki tüm $(A \Rightarrow \lambda)$ kurallarını çıkar;
 2. Eğer $\lambda \in L(G)$ ise P'ye $(S \Rightarrow \lambda)$ kuralını ekle;
 3. P'deki her $(A \Rightarrow \alpha)$ kuralı için aşağıdaki işlemleri tekrarla:
 - 3.1. Eğer α yok edilebilir değişkenler içeriyorsa, bu değişkenlerin bir ya da birkaçını α 'dan çıkararak oluşturulabilen her α_i için P'ye $(A \Rightarrow \alpha_i)$ kuralını ekle;son 3.1;
- son 3;

Bölüm 4 : Bağlamdan-Bağımsız Dilbilgisi ve Diller

➤ Örnek 4. 5.

$$G_{4.5} = \langle V_N, V_T, P, S \rangle$$

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

$$V_T = \{ a, b, c \}$$

$$P: S \Rightarrow ACA$$

$$A \Rightarrow B \mid C \mid aAa$$

$$B \Rightarrow bB \mid b$$

$$C \Rightarrow cC \mid \lambda$$

➤ Önce dilbilgisindeki yok edilebilir değişkenler bulunur.

$$T_L = \{ S, A, C \}$$

➤ $G'_{4.5} = \langle V_N, V_T, P, S \rangle$

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

$$V_T = \{ a, b, c \}$$

$$P: S \Rightarrow ACA \mid AC \mid CA \mid AA \mid A \mid C \mid \lambda$$

$$A \Rightarrow B \mid C \mid aAa \mid aa$$

$$B \Rightarrow bB \mid b$$

$$C \Rightarrow cC \mid c$$

Bölüm 4 : Bağlamdan-Bağımsız Dilbilgisi ve Diller

4.3.3. Birim Türetme Kuralları

- $A \Rightarrow B$ biçimindeki kurallara birim türetme kuralı denir.
- Algoritma 4.3. Bir değişkenden (A) türetilen değişkenler kümesinin bulunması
 1. $T_A = \{ A \}$;
 2. $T_{Eski} = \Phi$;
 3. ($T_A = T_{Eski}$) oluncaya kadar aşağıdaki işlemleri tekrarla:
 - 3.1. $T_{Yeni} = T_A - T_{Eski}$;
 - 3.2. $T_{Eski} = T_A$;
 - 3.3. (T_{Yeni})’deki her değişken (B) için aşağıdaki işlemleri tekrarla:
 - 3.3.1. her ($B \Rightarrow C$) kuralı için aşağıdaki işlemleri tekrarla:
$$T_A = T_A \cup \{ C \} ;$$
son 3.3.1;
 - son 3.3;
 - son 3;

Bölüm 4 : Bağlamdan-Bağımsız Dilbilgisi ve Diller

➤ **Algoritma 4.4.** Birim türetme kuralı içermeyen dilbilgisinin bulunması

1. $P = P - \{ A \Rightarrow B \mid A, B \in V_N \}$

2. (V_N) 'deki her değişken (A) için, eğer (T_A) 'da A 'dan başka en az bir değişken varsa aşağıdaki işlemleri tekrarla:

2.1. $T_A = T_A - \{A\}$;

2.2. (T_A) 'daki her değişken (B) için aşağıdaki işlemleri tekrarla:

B kurallarının tümü $(B \Rightarrow \beta_1 \mid \beta_2 \mid \beta_3 \mid \dots \beta_B)$ olmak üzere,

$$P = P \cup \{ A \Rightarrow \beta_1 \mid \beta_2 \mid \beta_3 \mid \dots \beta_B \} ;$$

Son 2.2 ;

Son 2 ;

Bölüm 4 : Bağlamdan-Bağımsız Dilbilgisi ve Diller

➤ Örnek 4.6.

$$G_{4.6} = \langle V_N, V_T, P, S \rangle$$

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

$$V_T = \{ a, b \}$$

$$P : S \Rightarrow A \mid AA \mid AC \mid CA \mid ACA \mid \lambda$$

$$A \Rightarrow B \mid aAa \mid aa$$

$$B \Rightarrow C \mid bB \mid b$$

$$C \Rightarrow cC \mid c$$

➤ Önce her değişkenden türetilebilecek değişkenler bulunur:

$$T_S = \{ S, A, B, C \} \quad T_A = \{ A, B, C \} \quad T_B = \{ B, C \}$$

➤ $G'_{4.6} = \langle V_N, V_T, P, S \rangle$

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

$$V_T = \{ a, b \}$$

$$P : S \Rightarrow AA \mid AC \mid CA \mid ACA \mid \\ aAa \mid aa \mid bB \mid b \mid cC \mid c \mid \lambda$$

$$A \Rightarrow aAa \mid aa \mid bB \mid b \mid cC \mid c$$

$$B \Rightarrow bB \mid b \mid cC \mid c$$

$$C \Rightarrow cC \mid c$$