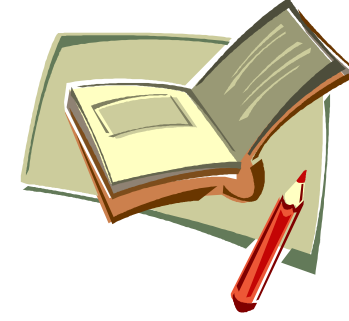




Grammer ve Diller

Prof.Dr. Banu Diri



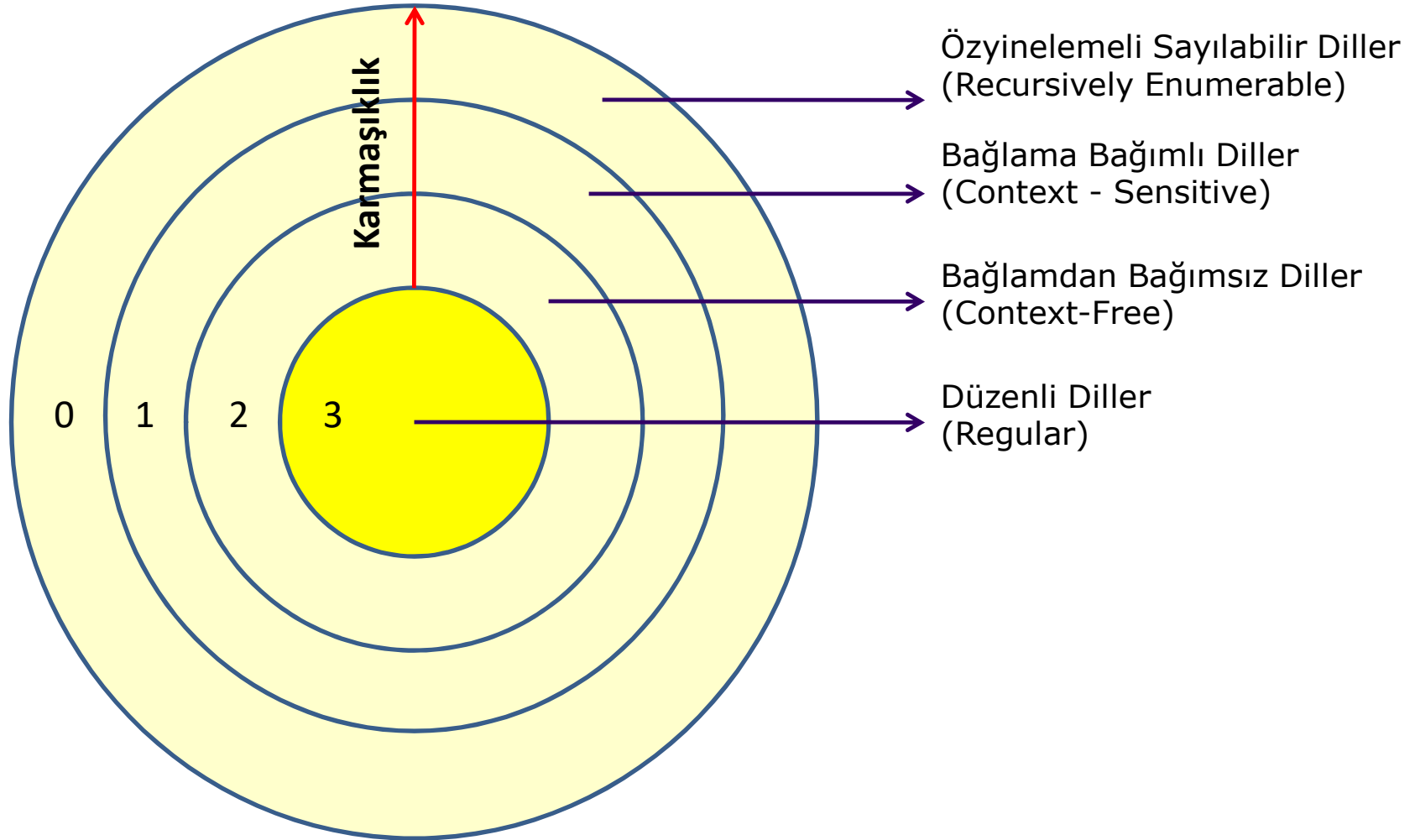
-
1. Her biçimsel dil belirli bir alfabe üzerinde tanımlanır.
 2. Alfabe sonlu sayıda simgelerden oluşan bir kümedir.
 3. Alfabadeki simgelerin arka arkaya getirilmesi ile dizgiler (string) oluşturulur.
 4. Biçimsel dil, bir alfabadeki simgelerden oluşturulan dizgilerin bir kümesidir. Bu kümeyi E ile gösterirsek, bu alfabe üzerinde tanımlanan her dil E 'nin bir alt kümesidir. E deki her dizgi dilin tümcelerini (sentence) oluşturur.
 5. Bir alfabe üzerinde tanımlanan biçimsel bir dil, bu alfabadeki simgelerden oluşan dizgileri *geçerli* ve *geçersiz* diye ikiye ayırır. Dilde yer alan ve tümceleri oluşturan dizgiler *geçerli* tersi ise *geçersiz*'dir.

-
6. Biçimsel dil açısından dizgi, tümce ve sözcük terimleri birbirlerinin yerine kullanılabilir. Tümce dilde yer alan dizgi veya sözcükleri anlatmak için kullanılır. Buna göre, bir alfabe ve bu alfabe üzerinde tanımlı bir dil düşünüldüğünde, alfabedeki simgelerden oluşturulan ve dilde yer alan geçerli dizgiler dilin tümcelerini oluşturmaktadır.
 7. Dilin hangi tümcelerden oluştuğunu gösteren kurallar bütünü ise dilbilgisi (grammar) olarak adlandırılır.

Biçimsel dilbilgisi ve dillerin incelenmesinde, değişik harf grupları değişik alanlarda kullanılır.

Harf Grubu	Örnekler	Kullanım alanları
Latin alfabesinin başındaki büyük harfler	A,B,C,...	Sözdizim değişkenleri
Latin alfabesinin başındaki küçük harfler ve rakamlar	a,b,c,...,0,1,2	Uç simgeler
Latin alfabesinin sonundaki büyük harfler	U,V,W,Y,...	Sözdizim değişkeni ya da uç simgeler
Latin alfabesinin sonundaki küçük harfler	u,v,w,y,...	Uç simge dizgileri (sözcükler)
Yunan alfabesinin başındaki küçük harfler	α , β , γ ,...	Tümcesel yapılar

Chomsky Hiyerarşisi



Sırasıyla;

- Dilbilgisi ve Dilin Biçimsel Tanımı
- Dilbilgisi ve Dillerin Sınıflandırılması
tür-0, tür-1, tür-2, tür-3
- Sağ-doğrusal ve Sol-doğrusal Dilbilgisi
- Türetme/Ayrıştırma Ağacı
Chomsky Normal Biçimi
Yukarıdan-aşağıya Ayrıştırma (Top-down parsing)
Aşağıdan-yukarıya Ayrıştırma (Bottom-up parsing)

tür-0, tür-1, tür-2, tür-3
Örnekler ile anlatılacaktır

Dilbilgisi ve Dilin Tanımı

Bicimsel olarak dilbilgisi bir dördü olarak tanımlanır.

$$G = \langle V_N, V_T, P, S \rangle$$

V_N : Söz dizim değişkenleri kümesi (sonlu bir küme)

V_T : Uç simgeler kümesi (sonlu bir küme)

V_N ve V_T ayrık kümelerdir. $V_N \cap V_T = \emptyset$

S : Başlangıç değişkeni $S \in V_N$

P : Yeniden yazma ve türetme kuralları

$\alpha \Rightarrow \beta$ (α 'nın yerine β konulabilir)

En genel biçimiyle

$$\alpha \in V^+ \quad \beta \in V^*$$

$$V = V_N \cup V_T \quad V^+ = V^* - \{\lambda\}$$

Bir dilbilgisi tarafından tanımlanan dil biçimsel olarak aşağıdaki gibi tanımlanır.

$$L(G) = \{ w \mid w \in V_T^*, S^* \Rightarrow w \}$$

Bu tanıma göre, bir dilin tümceeleri, başlangıç simgesinden (S 'den) başlanarak ve yeniden yazma kuralları yeterli sayıda kullanılarak elde edilen uç simge dizgeleridir.

$$S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \dots \Rightarrow \alpha_n \Rightarrow w$$

$\alpha_1, \alpha_2, \dots, \alpha_n$: tümcesel yapılar

w : tümce

ÖRNEK $G_1 = \langle V_N, V_T, P, S \rangle$

$$V_N = \{ S \}$$

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

$$P: S \Rightarrow 0S1$$

$$S \Rightarrow 01$$

G_1 tarafından türetilen birkaç tümce

$$S \Rightarrow 01$$

$$S \Rightarrow 0S1 \Rightarrow 0011$$

$$S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 000111$$

Bu örneklerden yararlanarak dili

$$L(G_1) = \{ 0^n 1^n \mid n \geq 1 \}$$

Dilbilgisi ve Dillerin Sınıflandırılması

Dilbilgisi ve türettikleri diller, yeniden yazma kurallarının özellikleri-
ne göre:

- tür- $\emptyset$ (kısıtlamasız dilbilgisi ve diller)
- tür-1 (bağlama - bağımlı dilbilgisi ve diller)
- tür-2 (bağlamdan - bağımsız dilbilgisi ve diller) (Context Free Grammars)
- tür-3 (düzgün dilbilgisi ve diller)

Tür- $\emptyset$ Dilbilgisi ve Dil

Tür- $\emptyset$ 'ın yeniden yazma kuralları

$$\alpha \Rightarrow \beta : \alpha \in V^+ \quad \beta \in V^*$$

Bu tip dillere öz yineli & sayılabilir (recursively enumerable) diller de
denir. Dilin tümcelerini ard arda türeten bir yordam vardır. (r.e)

ÖRNEK

$$G_2 = \langle V_N, V_T, P, S \rangle$$

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

$$V_T = \{a\}$$

$$P: S \Rightarrow LAaR$$

$$Aa \Rightarrow aaA$$

$$AR \Rightarrow BRIC$$

$$aB \Rightarrow Ba$$

$$LB \Rightarrow LA$$

$$aC \Rightarrow Ca$$

$$LC \Rightarrow A$$

$$S \Rightarrow L\underline{A}aR \Rightarrow Laa\underline{A}R \Rightarrow Laa\underline{C} \Rightarrow L\underline{a}Ca \Rightarrow \\ \underline{L}Caa \Rightarrow aa$$

$$S \Rightarrow L\underline{A}aR \Rightarrow L\underline{aa}A\underline{R} \Rightarrow L\underline{aa}B\underline{R} \Rightarrow L\underline{a}BaR \Rightarrow \\ \underline{L}Ba\underline{a}R \Rightarrow L\underline{A}aa\underline{R} \Rightarrow L\underline{aa}A\underline{a}R \Rightarrow \\ L\underline{aaaa}A\underline{R} \Rightarrow L\underline{aaaa}C \Rightarrow L\underline{aaaa}Ca \\ L\underline{aa}C\underline{aa} \Rightarrow L\underline{a}C\underline{aaa} \Rightarrow L\underline{C}aaaa \Rightarrow aaaa$$

Dilbilgisinin kuralları dikkatle incelendiğinde $L(G_2)$ dilinin tanımı

$$L(G_2) = \{a^i \mid i = 2^n, n \geq 1\}$$

Tür-1 Dilbilgisi ve Dil

Tür-1'in yeniden yazma kuralları

$$\alpha \Rightarrow \beta : \alpha \in V^+ \quad \beta \in V^* \quad |\alpha| \leq |\beta|$$

Tür-1'de, Tür-0'a göre $|\alpha| \leq |\beta|$ kısıtlaması getirilmiştir.

Tür-1 dilbilgisine bağlama-bağımlı (context sensitive) ^{veya düzeyli (n.)} de denir.

Çünkü tür-1 dilbilgisi, yeniden yazma kurallarının tümü

$\alpha_1 A \alpha_2 \Rightarrow \alpha_1 \beta \alpha_2 \quad A \in V_N, \alpha_1, \alpha_2, \beta \in V^*$ olan normal bir

forma dönüştürülebilir. Bu normal biçimde $\alpha_1 \dots \alpha_2$ bağlamın-

da A'nın yerine β konulabildiği için, dilbilgisi bağlama

bağımlı dilbilgisi olarak adlandırılır.

ÖRNEK

$$G_3 = \langle V_N, V_T, P, S \rangle$$

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

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

$$P: S \Rightarrow aSAB$$

$$S \Rightarrow aAB$$

$$BA \Rightarrow AB$$

$$aA \Rightarrow ab$$

$$bA \Rightarrow bb$$

$$bB \Rightarrow bc$$

$$cB \Rightarrow cc$$

$$S \Rightarrow \underline{a}AB \Rightarrow ab\underline{B} \Rightarrow abc$$

$$S \Rightarrow a\underline{S}AB \Rightarrow aa\underline{A}BAB \Rightarrow aa b \underline{B}AB \Rightarrow$$

$$aab \underline{A}BB \Rightarrow a \underline{a}bb \underline{B}B \Rightarrow a \underline{a}bb \underline{c}B$$

$$\Rightarrow a \underline{a}bb'cc$$

$$L(G_3) = \{a^n b^n c^n \mid n \geq 1\}$$

Tür-2 Dilbilgisi ve Dil

Tür-2 dilbilgisinin yeniden yazma kuralları

$$A \Rightarrow B : A \in V_N \quad B \in V^* \quad \text{biçimindedir.}$$

Yeniden yazma kurallarının sol tarafında tek bir değişken (A) yer almaktadır. Yeniden yazma kuralı, hangi bağlamda olursa, A'nın yerine B konulabileceğini söyler. Bu dilbilgisine Bağlamdan-Bağımsız (context free) dilbilgisi denir.

Programlama dilleri Bağlamdan-Bağımsız dilbilgisidir

ÖRNEK

$$G_1 = \langle V_N, V_T, P, S \rangle$$

$$V_N = \{S\}$$

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

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

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

Tür-3 Dilbilgisi ve Dil

Tür-3 dilbilgisinin yeniden yazma kuralları

$$A \Rightarrow aB$$

$$A \Rightarrow a$$

$$A \Rightarrow \lambda : A, B \in V_N \quad a \in V_T \quad \text{biçimindedir.}$$

Yeniden yazma kuralının sol tarafında tek bir değişken (A). Sağ tarafında ise ya tek bir uç simge ya da bir uç simge ile bir değişken yer almaktadır. Bu dilbilgisine Düzgün (regular) dil denir.

ÖRNEK $G_5 = \langle V_N, V_T, P, S \rangle$

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

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

$$P: S \Rightarrow 0S \mid 0A \mid 0 \mid \lambda$$

$$A \Rightarrow 0B$$

$$B \Rightarrow 1S$$

$$S \Rightarrow 0\underline{S} \Rightarrow 00$$

$$S \Rightarrow 0\underline{A} \Rightarrow 00\underline{B} \Rightarrow 001\underline{S} \Rightarrow 001$$

$$S \Rightarrow 0\underline{S} \Rightarrow 00\underline{A} \Rightarrow 000\underline{B} \Rightarrow 0001\underline{S}$$

$$\Rightarrow 00010\underline{A} \Rightarrow 000100\underline{B} \Rightarrow$$

$$0001001\underline{S} \Rightarrow 00010010$$

$L(G_5) \rightarrow$ içerisindeki her 1'den önce en az iki tane 0 bulunan dizilerdir.

$$(0)^*(001(0)^*)^n(0)^* \quad n \geq 0$$

$$((00)^+1)^*0^*$$

Sağ Doğrusal – Sol Doğrusal Dilbilgisi

Yeniden yazma kuralları

$$A \Rightarrow wB$$

$$A \Rightarrow w : A, B \in V_N, w \in V_T^*$$

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 veya bir uç ^{↑ simgeler} dizgisi ile bir değişken yer alır. Uç simgeler dizgisi sıfır uzunluğunda bir dizgide olabilir.

Yeniden yazma kuralları

$$A \Rightarrow Bw$$

$$A \Rightarrow w : A, B \in V_N, w \in V_T^*$$

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 dizgide 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.

ÖRNEK $G_6 = \langle V_N, V_T, P, S \rangle$

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

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

$$P: S \Rightarrow 0A$$

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

Sağ doğrusal bir dilbilgisidir.

$$S \Rightarrow 0\underline{A} \Rightarrow 0$$

$$S \Rightarrow 0\underline{A} \Rightarrow 010\underline{A} \Rightarrow 010$$

$$S \Rightarrow 0\underline{A} \Rightarrow 010\underline{A} \Rightarrow 01010\underline{A} \\ \Rightarrow 01010$$

$$L(G_6) = 0(10)^* = (01)^*0$$

Türetme Ağacı

Bir dilbilgisi tarafından türetilen tümcesel yapı ve tümceler şu şekilde gösteriliyor olsun:

$$S \Rightarrow \underbrace{P_1 \Rightarrow P_2 \Rightarrow P_3 \dots \Rightarrow P_{n-1} \Rightarrow P_n}_{\text{Tümcesel yapılar}} \Rightarrow \underbrace{w}_{\text{Tümce}}$$

$$P_i \in V^*, \quad w \in V_T^*$$

Bağılamdan -bağımsız dillerin her tümcesel yapısına ve her tümcesine türetme veya ayrıştırma ağacı (derivation or parsing tree) karşı gelir.

ÖRNEK

$$G = \langle V_N, V_T, P, S \rangle$$

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

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

$$P: S \Rightarrow \emptyset S \emptyset \mid 1S1 \mid \emptyset A \emptyset \mid 1A1$$

$$A \Rightarrow \emptyset A 1 \mid 1A \emptyset \mid \emptyset 1 \mid 1 \emptyset$$

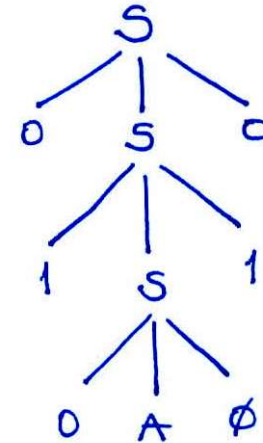
türetilen

$$\mathcal{P} = 010A010 \text{ tümcesel yapısı ile}$$

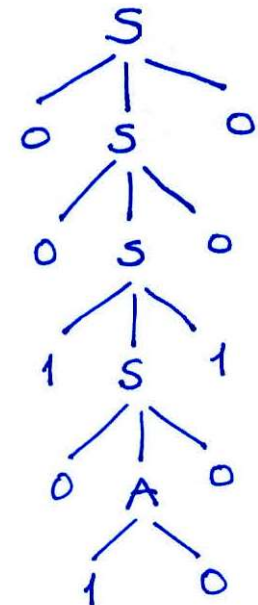
$$w = 0010100100 \text{ tümcesine}$$

karşı gelen türetme ağaçları

$\mathcal{P} \Rightarrow$



$w \Rightarrow$



Türetme ve 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 söz dizim değişkenleri veya uç simgeler olabilir ($X \in V$). Eğer ağaç bir tümeceye karşı geliyorsa yaprakların etiketleri sadece uç simgelerdir ($a \in V_T$) olabilir.
- 4) Eğer bir ara düğümün etiketi A , bu ara düğümün hemen altındaki düğümlerin etiketleri soldan sağa $X_1, X_2, \dots, X_k$ 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 bulunmalıdır.

Soldan ve Sağdan Türetme

Yapraklarının etiketleri uç simgeler olan her türetme ağacına dilin bir tümcesi karşı gelir. Dil belirgin (unambiguous) bir dil ise, dildeki her tümceye bir türetme ağacı gelir, eğer bir tümceye birden fazla türetme ağacı karşı geliyorsa dil belirgin olmayan (ambiguous) bir dildir.

X { Tümceleri birden çok anlam taşıyan belirgin olmayan dillerin uygulamada bir değeri yoktur.

Ex: I saw the man in the park with a telescope

Eğer bir tümce türetilirken, her adımda en soldaki değişkene bir türetme uygulanıyorsa, yapılan türetmeye soldan türetme (leftmost derivation) denir. Soldan türetmenin her ara adımında, eğer tümcesel yapı birden çok değişken içeriyorsa, öncelik en soldaki değişkene verilir.

ÖRNEK

$$G = \langle V_N, V_T, P, S \rangle$$

$$V_N = \{S\}$$

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

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

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

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

Örnek

$$G = \langle V_N, V_T, P, S \rangle$$

$$V_N = \{S\}$$

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

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

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

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

w_1 'in soldan türetilmesi

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

w_2 'in soldan türetilmesi

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

Eğer bir tüme türetilirken, her adımda en sağdaki değişkene bir türetme uygulanıyorsa, yapılan türetmeye **sagdan türetme** (rightmost derivation) denir. Sağdan türetmenin her ara adımında, eğer tümcesel yapı birden çok değişken içeriyorsa, öncelik en sağdaki değişkene verilir.

Örnek

$$G = \langle V_N, V_T, P, S \rangle$$

$$V_N = \{S\}$$

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

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

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

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

w_1 'in sağdan türetilmesi

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

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

w_2 'nin sağdan türetilmesi

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

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

Aşağıdan Yukarıya Doğru Ayırıştırma (Bottom-Up Parsing)

Aşağıdan - yukarıya ayırmada, ayrıştırılacak tümceden (w) başlanarak bir dizi işlem sonunda dilbilgisinin başlangıç simgesi olan S elde edilmeye çalışılır. Bu ayırmada sağdan öncelikli türetme (rightmost derivation) kullanılır.

ÖRNEK

$A \Rightarrow abA|ab$
 $S \Rightarrow aS|AB|B$
 $B \Rightarrow BB|ba$

"aababbaba"

Sırasıyla 9,8,7,6,5,4,3'lü eşlemelere sağdan ilerliyerek bakılır.

$\overset{(5)}{a} \overset{(2)}{a} \overset{(6)}{b} \overset{(4)}{b} \overset{(3)}{b} \overset{(1)}{a} \overset{(1)}{a}$
 ① aababbab $\max=2$
 ② aababbAa $\max=2$
 ③ aababBba $\max=2$
 ④ aabAbaba $\max=3$
 ⑤ aaBbbaba $\max=2$
 ⑥ aAabbaba $\max=2$

$\overset{(2)}{a} \overset{(3)}{A} \overset{(4)}{b} \overset{(1)}{a} \overset{(1)}{b} \overset{(1)}{a}$
 ① aAbab $\max=2$
 ② aAbAa $\max=0$
 ③ aABba $\max=2$

$\overset{(2)}{a} \overset{(3)}{A} \overset{(1)}{B} \overset{(1)}{b} \overset{(1)}{a}$
 ① aABb $\max=2$
 ② aSba $\max=2$

a a b a b b a b a

⑤ ②
⑥ ④ ③ ①

- ① a a b a b b a B max=2
- ② a a b a b b A a max=2
- ③ a a b a b B b a max=2
- ④ a a b A b a b a max=3
- ⑤ a a B b b a b a max=2
- ⑥ a A a b b a b a max=2

a A b a b a

② ③ ①

- ① a A b a B max=2
- ② a A b A a max=0
- ③ a A B b a max=2

a A B b a

② ①

- ① a A B B max=2
- ② a S b a max=2

a S b a

② ①

- ① a S B max=2
- ② S b a max=2

S b a

S B çözüm yok

a S B

S B çözüm yok

a A B B

① ①

- ① a A B max=2
- ③ a S B max=2

S B çözüm yok

a S ⇒ S

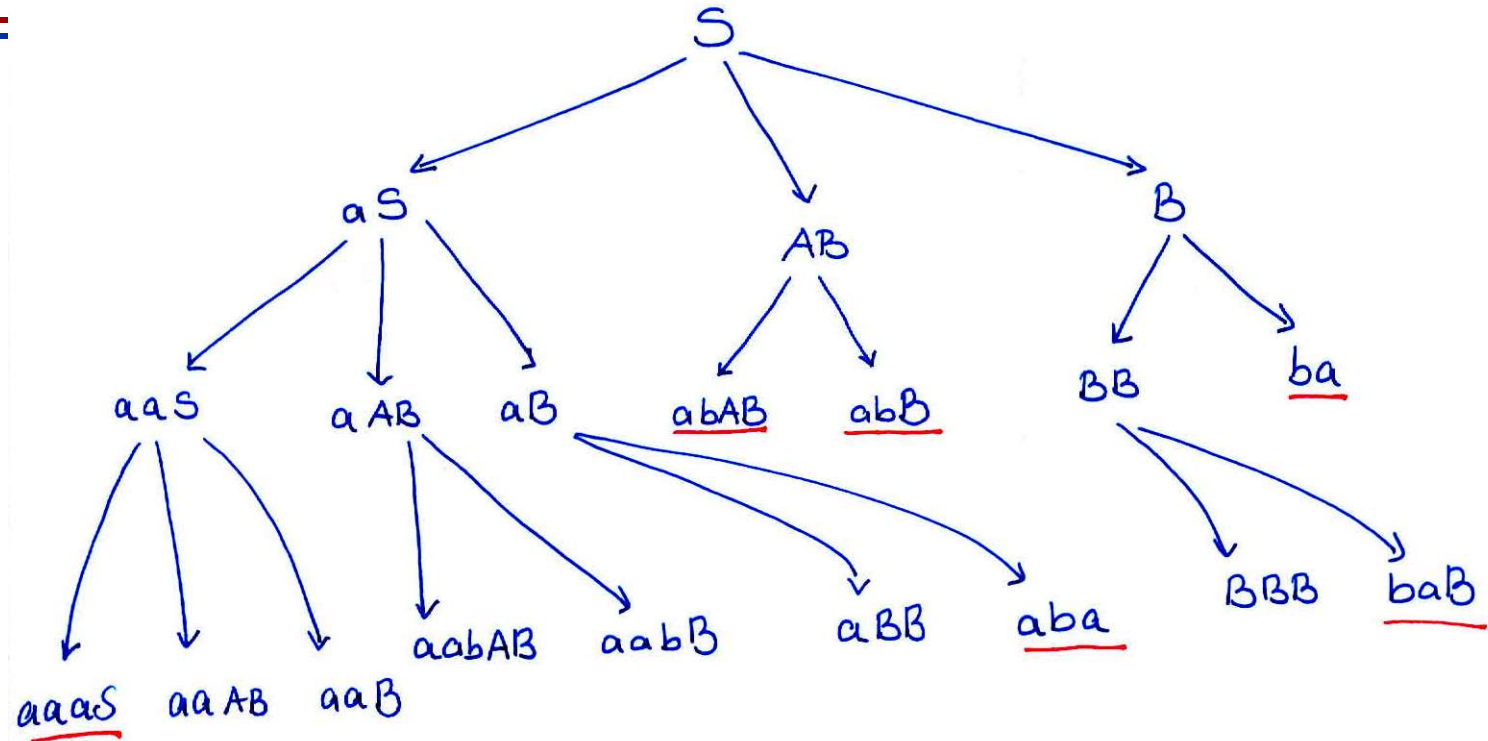
Yukarıdan Aşağıya Ayırıştırma (Top-Down Parsing)

Yukarıdan-aşağıya ayırıştırma, dilbilgisinin başlangıç simgesi olan S 'den başlanarak, ayrıştırılacak tümce (w) türetilmeye çalışılır. w türetilmeye çalışılırken soldan türetme (leftmost derivation) kuralları uygulanır.

Soldan türetmede, türetmenin herhangi bir adımındaki tümcesel yapı α ise ve α içindeki söz dizim değişkenlerinden en soldaki A ise, A kurallarından biri uygulanarak yeni bir tümcesel yapı elde edilir.

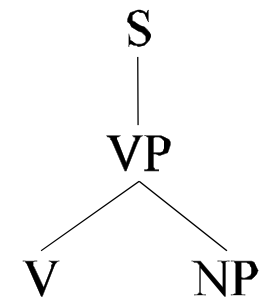
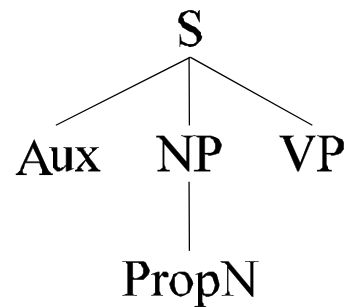
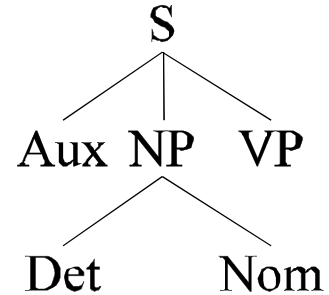
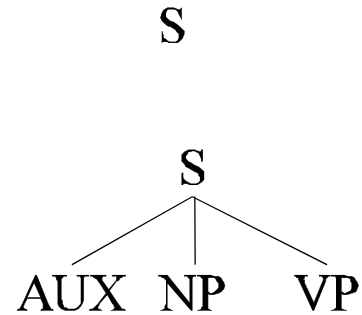
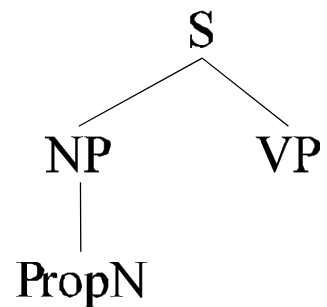
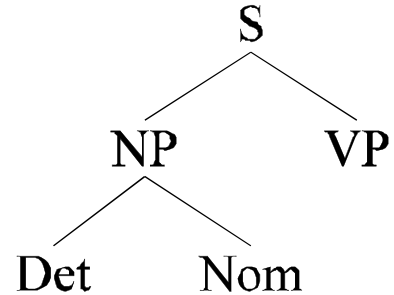
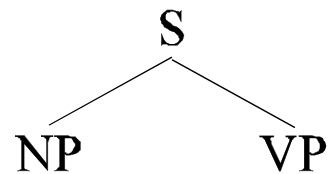
* Ağacın herhangi bir düğümünün etiketi, w 'nin ön eki değilse, bu düğüm ölü bir düğümdür.

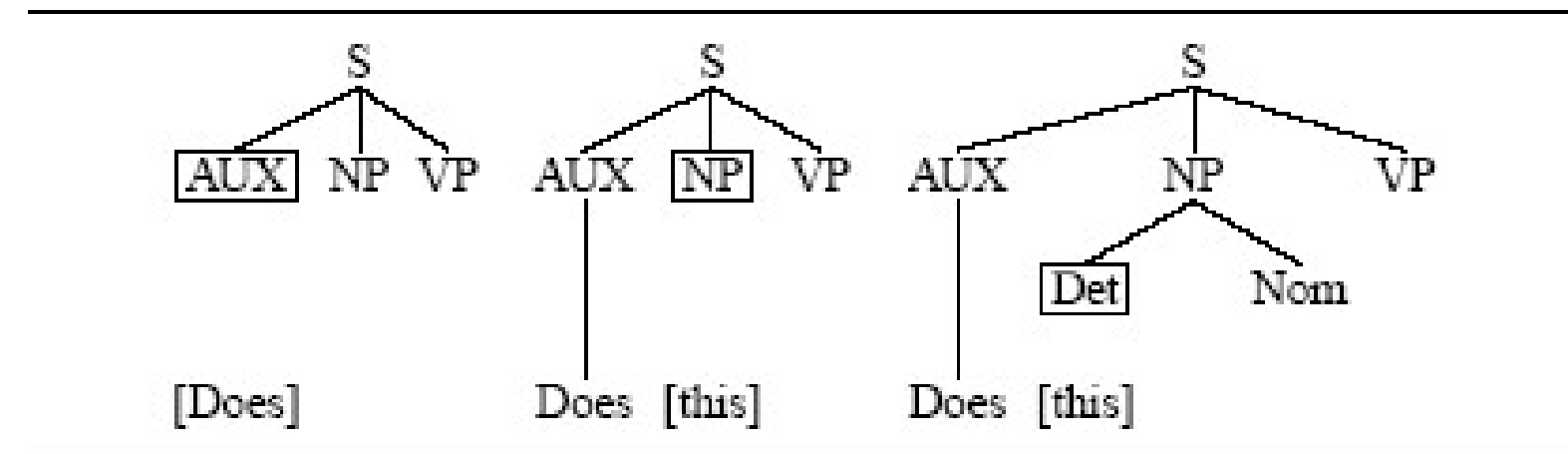
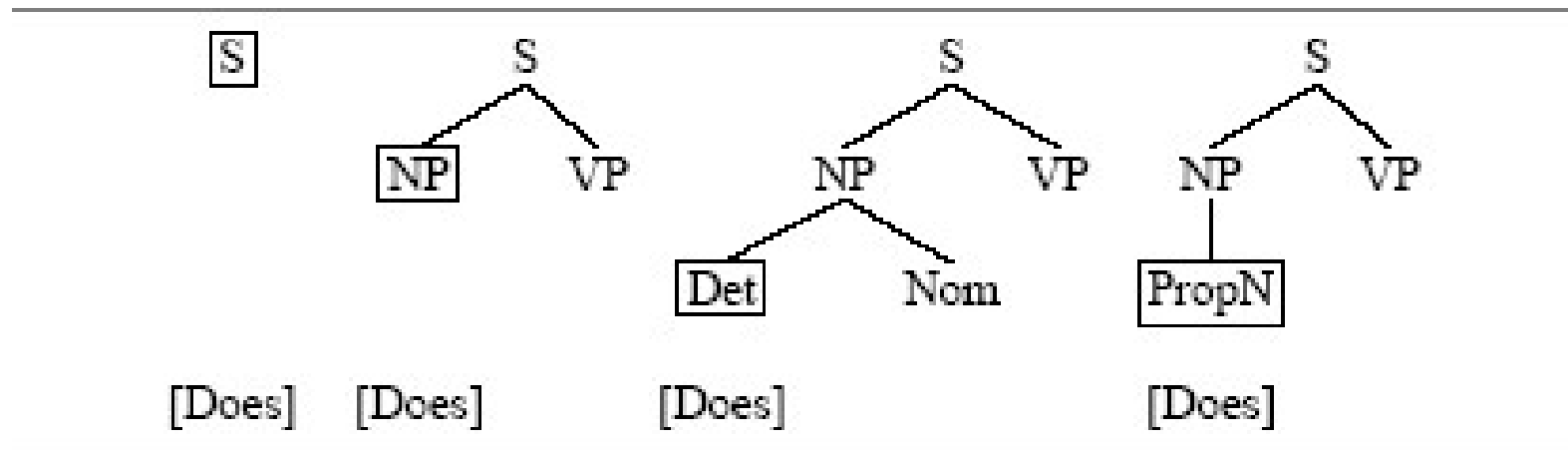
* Türetme işlemi ağacın bir düğümünün etiketi w oluncaya, ya da ağacın tüm yaprakları birer ölü düğüm oluncaya kadar sürdürülür. Etiketi w olan bir düğüm elde edilirse ayrıştırmaumlu sonuçlanmış dur. Etiketi w olan bir düğüm elde edilemez ve ağacın tüm yapraklarının birer ölü düğüm olduğu anlaşılırsa, ayrıştırma olumsuz sonuçlanmış olur.

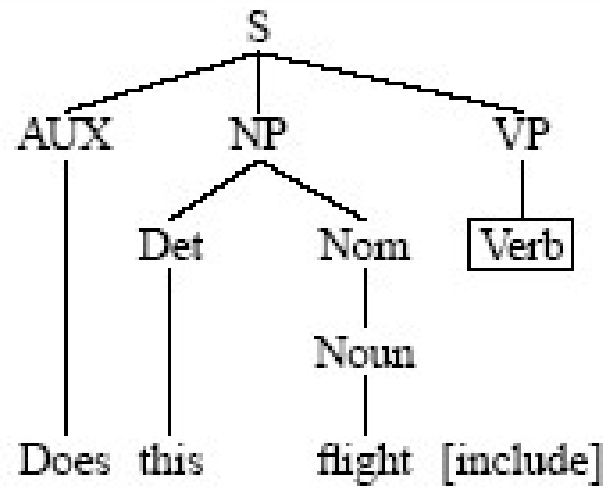
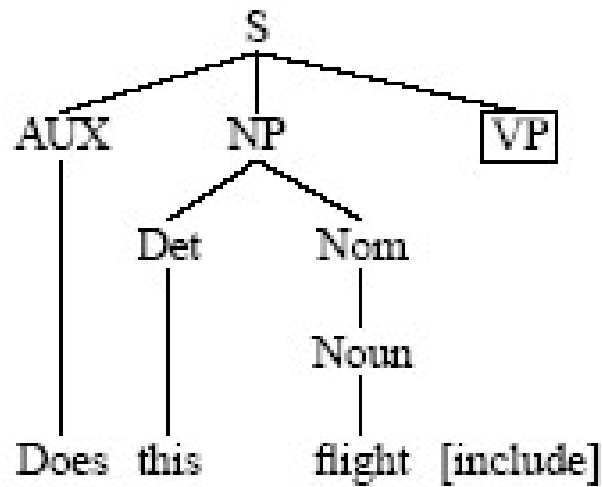
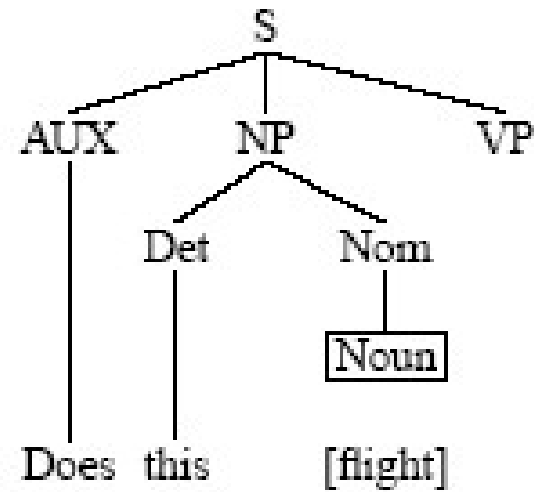
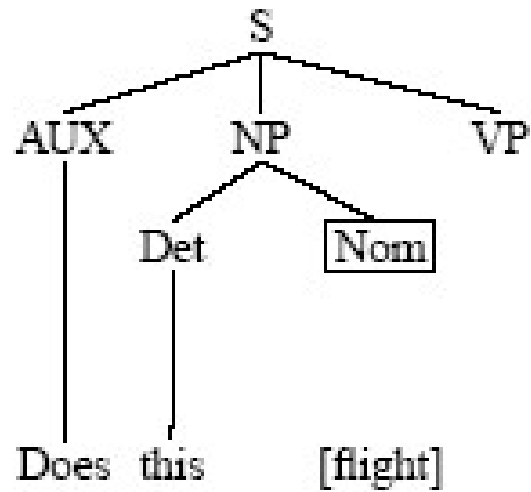


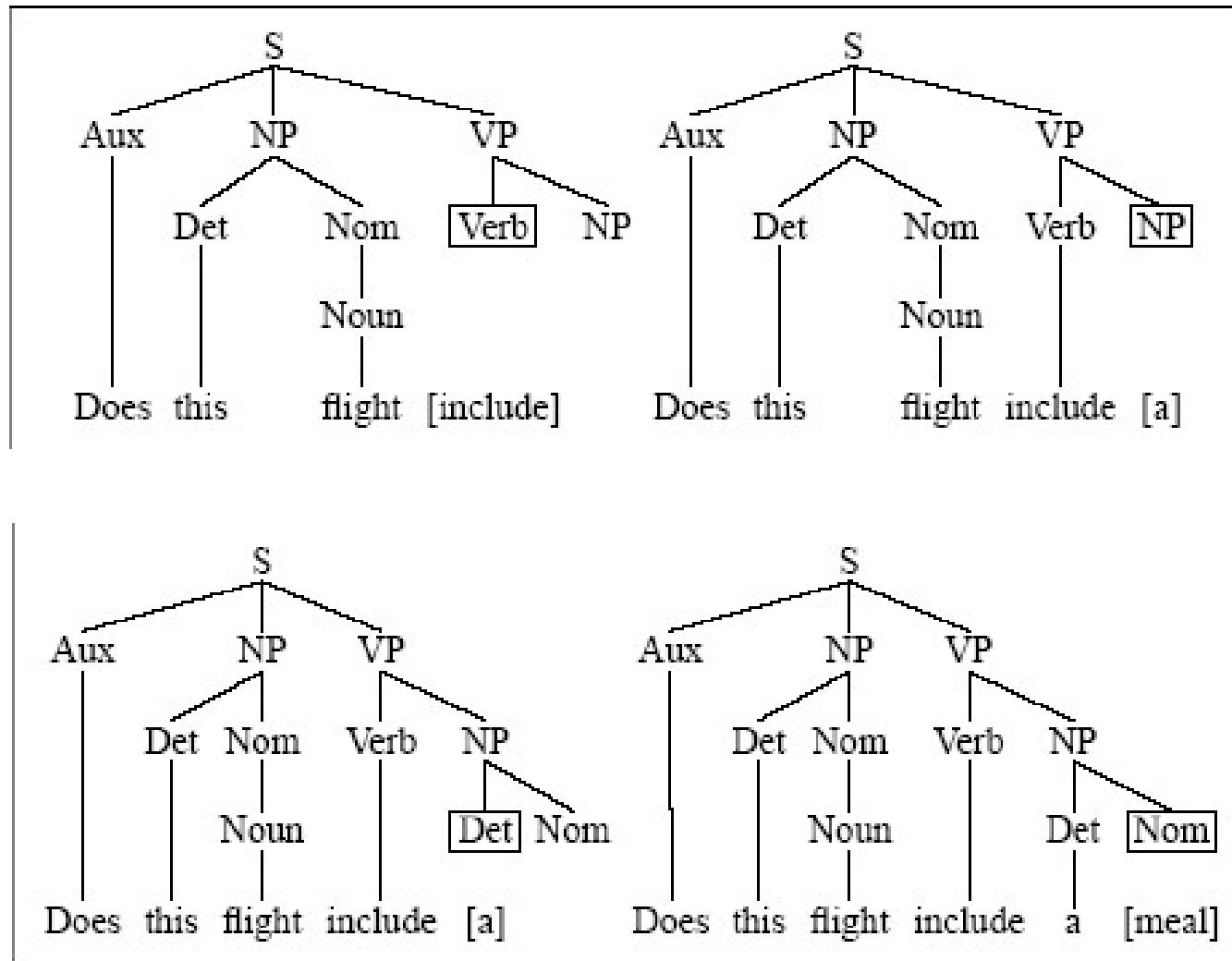
Ayrıştırma işleminin karmaşıklığı, dilbilgisinin yeniden yazma kurallarının sayısı ile ayrıştırılan tümcedeki simge sayısına paralel olarak üssel biçimde artar.

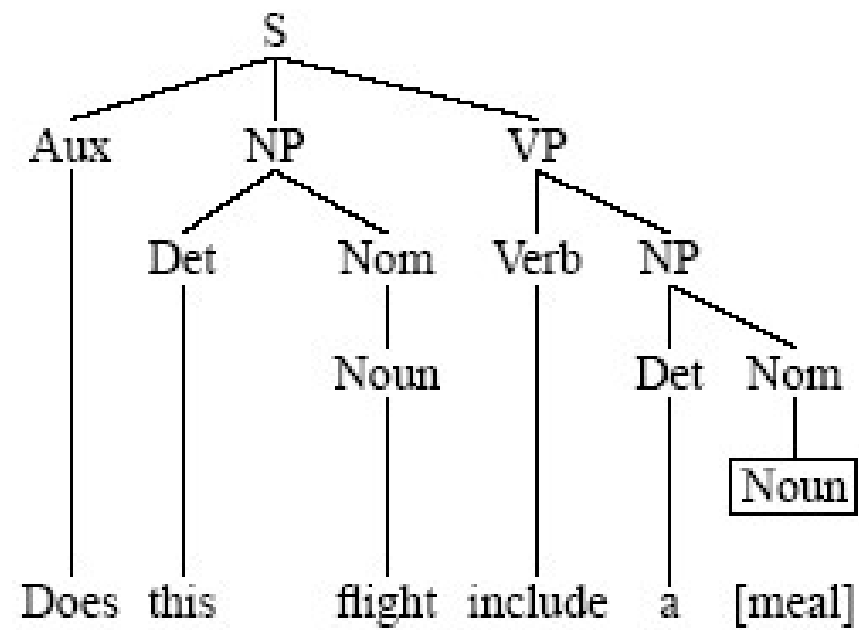
Yukarıdan-Aşağıya Ayırıştırma (Top-Down Parsing, Left-to-Right, Depth-First)

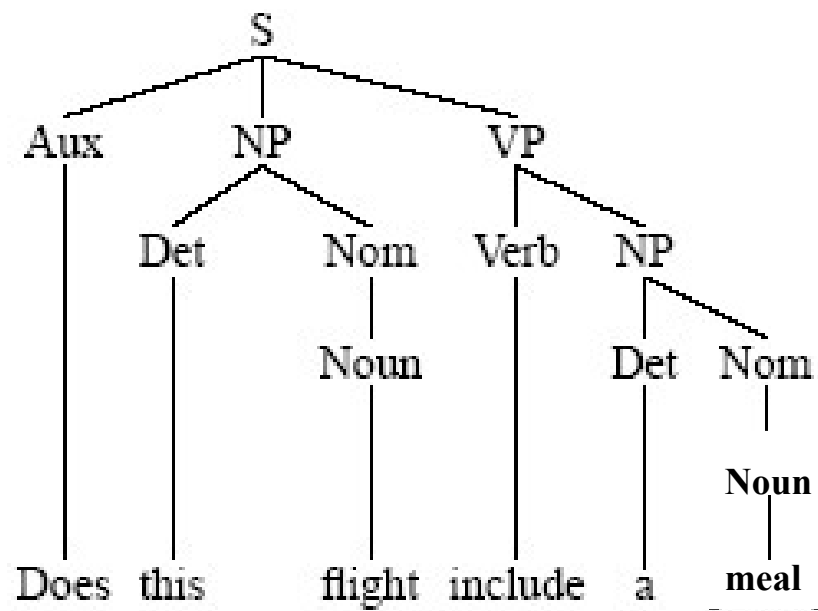












Aşağıdan-Yukarı Ayırıştırma (Bottom-Up Parsing)

$S \rightarrow NP VP$

$S \rightarrow Aux NP VP$

$S \rightarrow VP$

$NP \rightarrow Det Nominal$

$Nominal \rightarrow Noun$

$Nominal \rightarrow Noun Nominal$

$NP \rightarrow Proper-Noun$

$VP \rightarrow Verb$

$VP \rightarrow Verb NP$

$Det \rightarrow that | this | a$

$Noun \rightarrow book | flight | meal | money$

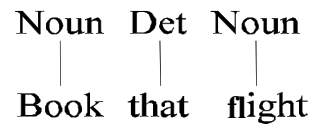
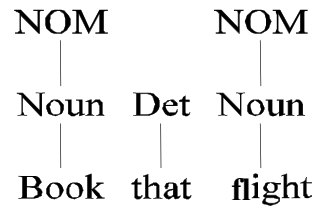
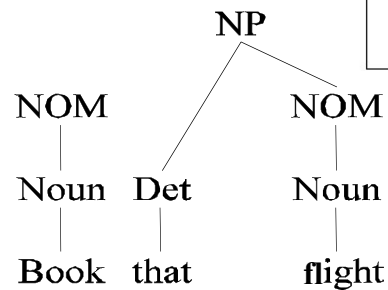
$Verb \rightarrow book | include | prefer$

$Aux \rightarrow does$

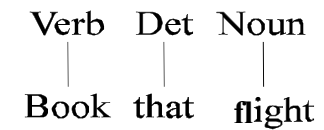
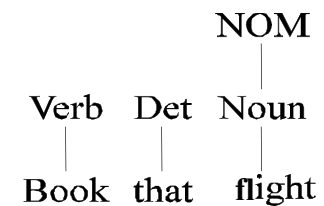
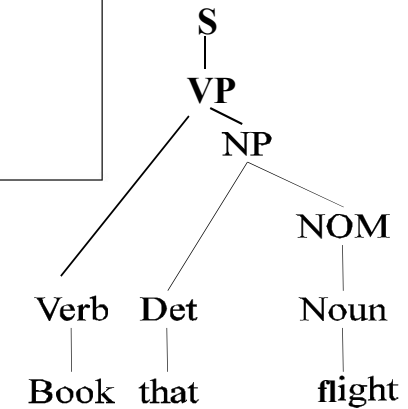
$Prep \rightarrow from | to | on$

$Proper-Noun \rightarrow Houston | TWA$

$Nominal \rightarrow Nominal PP$



Book that flight



S -> Aux NP VP

S -> NP VP

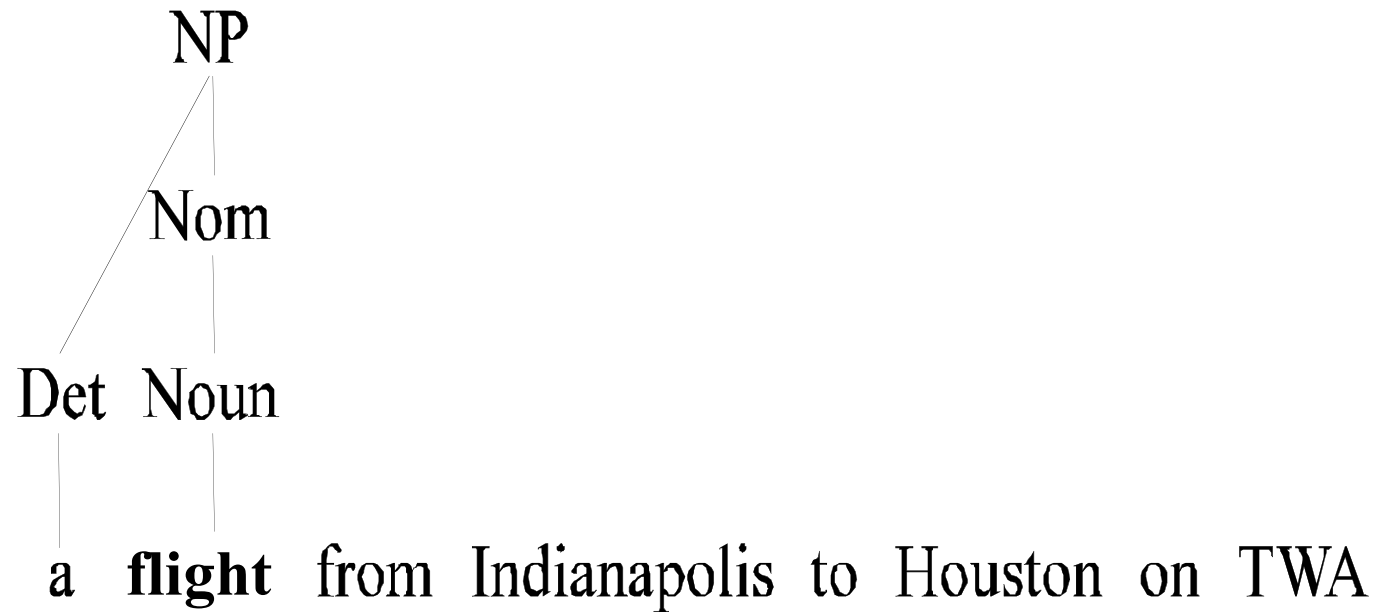
NP -> Det Nominal

NP -> NP PP

A flight from Indianapolis to Houston on TWA

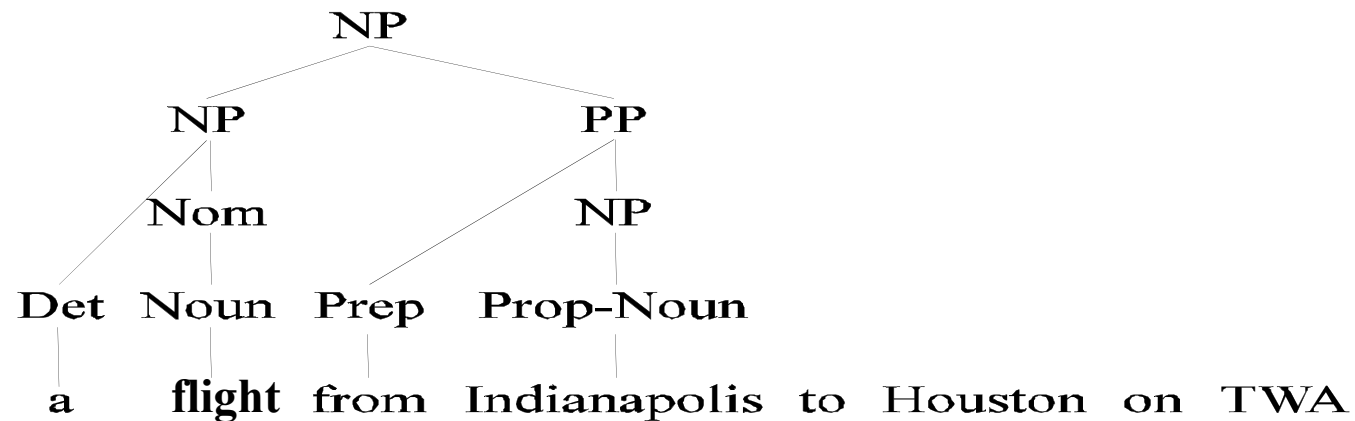
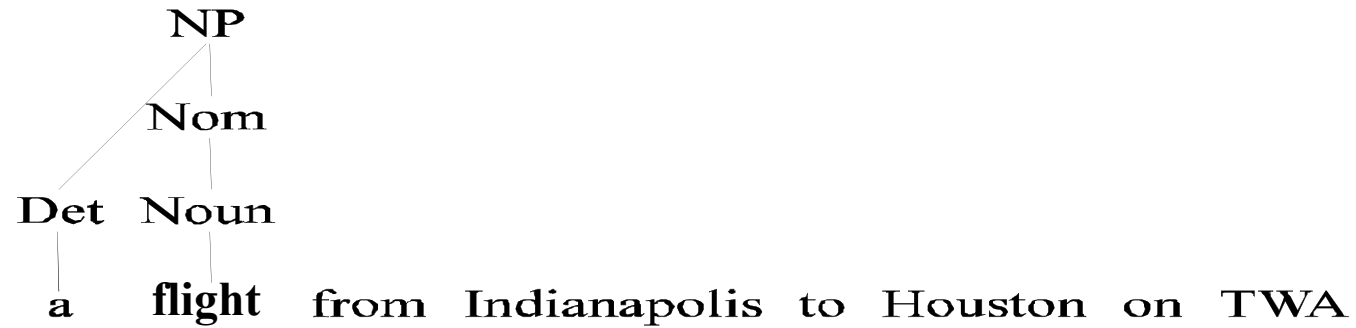
S -> Aux NP VP
S -> NP VP
NP -> Det Nominal
NP -> NP PP

A flight from Indianapolis to Houston on TWA



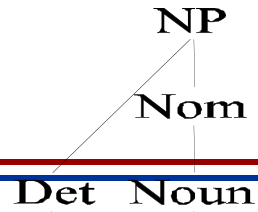
S -> Aux NP VP
S -> NP VP
NP -> Det Nominal
NP -> NP PP

A flight from Indianapolis to Houston on TWA

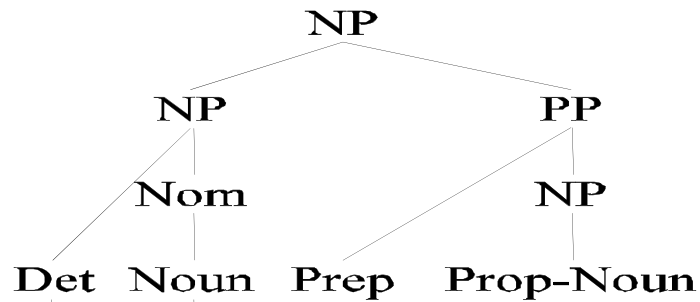


S -> Aux NP VP
 S -> NP VP
 NP -> Det Nominal
 NP -> NP PP

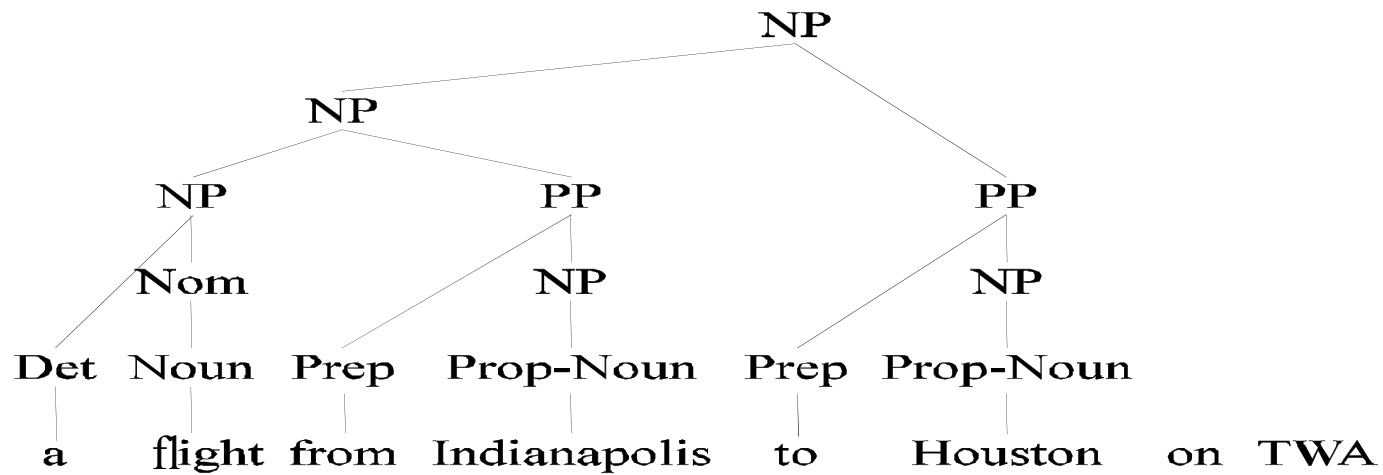
A flight from Indianapolis to Houston on TWA



a flight from Indianapolis to Houston on TWA

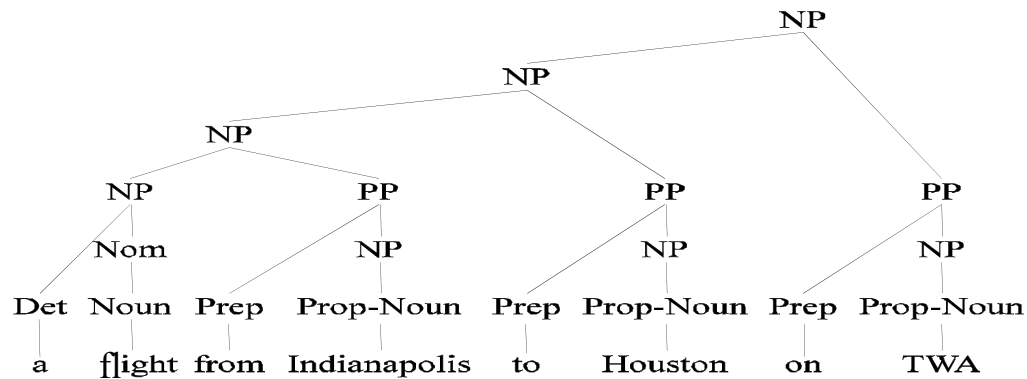
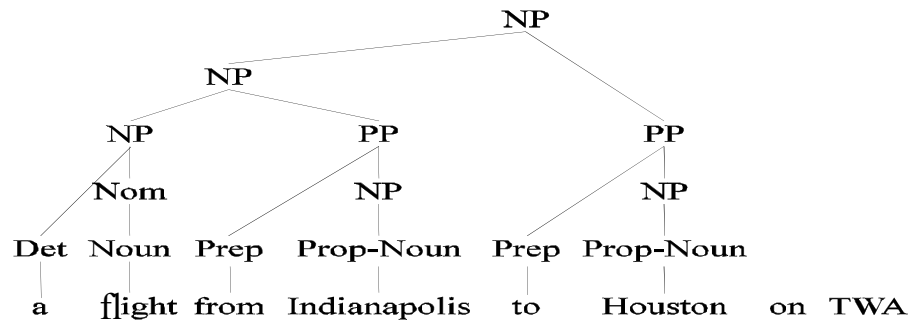
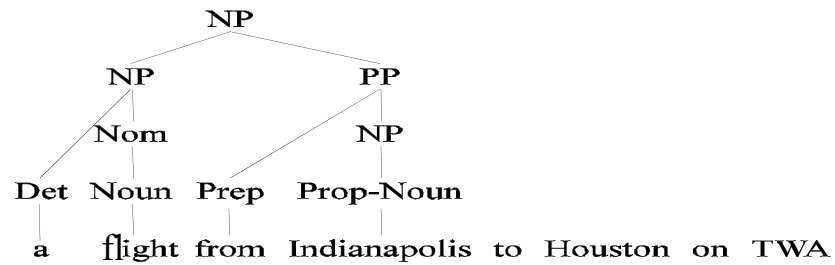
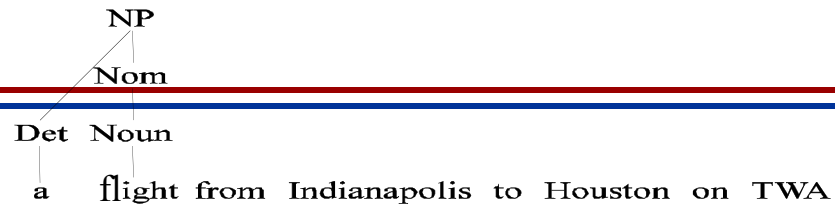


a flight from Indianapolis to Houston on TWA



S -> Aux NP VP
 S -> NP VP
 NP -> Det Nominal
 NP -> NP PP

A flight from Indianapolis to Houston on TWA



Örnek “Bir cümle isim veya fiil grubundan oluşur.”, “isim grubu isim ve/veya sıfat, fiil grubu da isim grubu ve/veya fiil den oluşur”.

Bu gramerde “Küçük çocuk kırmızı top aldı” cümlesinin çözümü

$C \rightarrow \dot{I}G \ FG$: (cümle isim ve fiil grubundan oluşur)

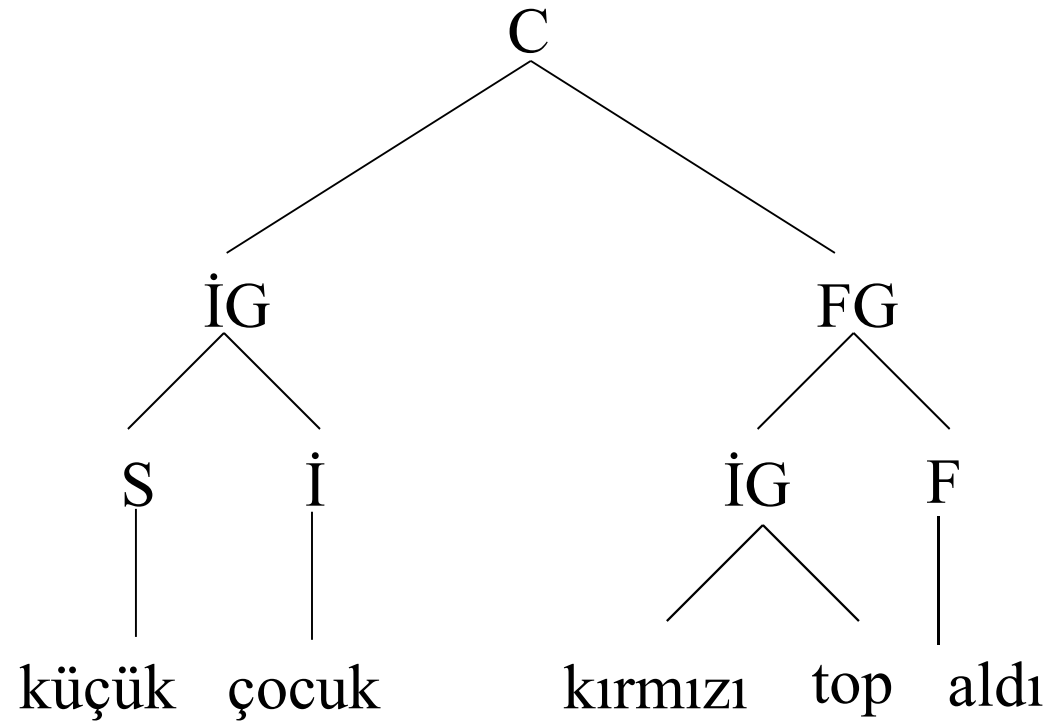
$\dot{I}G \rightarrow S \ \dot{I}$: (isim grubu $\rightarrow$ sıfat+isim)

$FG \rightarrow \dot{I}G \ F$: (fiil grubu $\rightarrow$ isim grubu + fiil)

$S \rightarrow \text{küçük} | \text{kırmızı}$

$\dot{I} \rightarrow \text{çocuk} | \text{top}$

$F \rightarrow \text{aldı}$



Genişletilmiş Geçiş Ağları – GGA

Augmented Network Transition Grammer- ATN

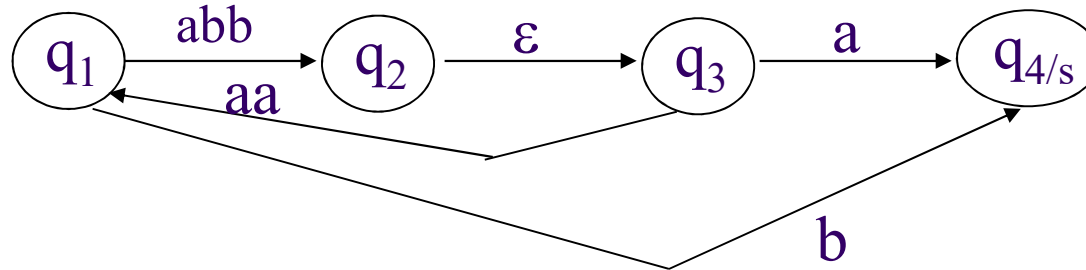
- Biçimsel dillerin, doğal dilleri tam olarak ifade edemediği durumlarda Genişletilmiş Geçiş Ağları tercih edilir.
- GGA, cümlelerin sözdizimsel analizinde kullanılır.
- GGA sonlu makinelere benzeyen durum ve bu durumlar arası geçiş kurallarından oluşmaktadır.

Üç bileşenden oluşur:

- En az bir başlangıç ve son durumu olan sonlu sayıdaki durumlar kümesi
- Belli bir metindeki mümkün olan harflerden oluşan küme
- Sonlu sayıdaki bir durumdan diğer bir duruma geçişi sağlayan geçişler kümesi

-
- ❖ Geçiş ağlarında bir durumdan diğer bir duruma geçmek için gerekli harf okunur ve bu harf geçilecek olan duruma geçmek için gereken harfle karşılaştırılır. Uygun olması durumunda geçilir.
 - ❖ Geçiş ağlarında doğru bir yol, bir başlangıç durumundan başlayıp, son duruma ulaşan geçişler sağlandığında tamamlanır.
 - ❖ Geçişlerdeki harflerin birbirine eklenmesiyle oluşan metin, ağın kabul etmesi için verilen metinle aynı ise, bu metin ağ tarafından kabul edilmiş demektir.

Genişletilmiş Geçiş Ağı örneği



Bu ağda tanınabilecek birkaç metin örneği:

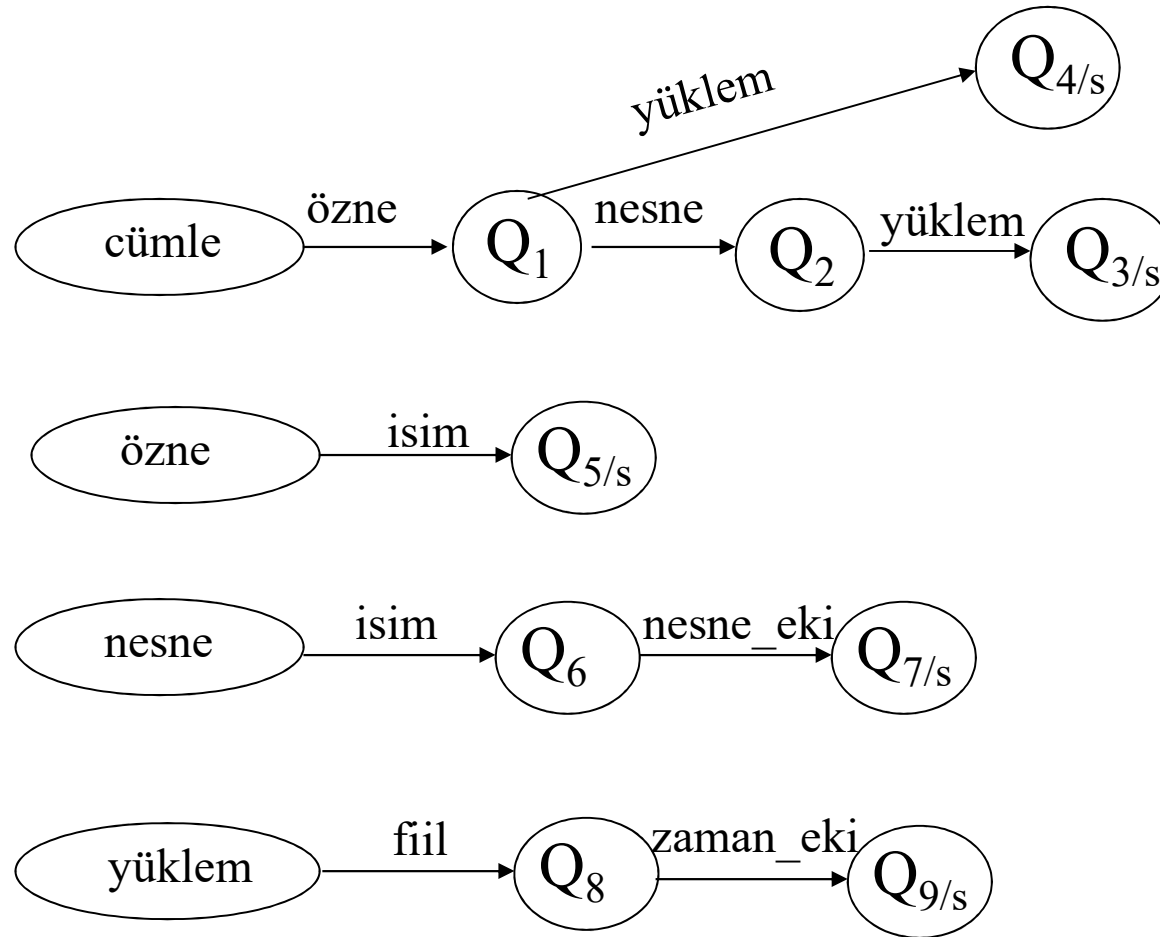
abba, abbaaabba, abbaab, b

Fakat *abbab, baab* yi tanıyamaz.

-
- ❖ Dilin grameri, durumlara karşı düşen yönlü graflar şeklinde ifade edilir.
 - ❖ Bağlarda ise, çoğalmakta olan durumlar arasındaki geçişleri tanımlayan etiket sınıfları bulunmaktadır.
 - ❖ Bu etiketler özel kelimeler, kelime kategorileri veya cümlelerin önemli parçalarını tanıyan diğer ağlara yönlendirme gibi değerler almaktadır.
 - ❖ GGA, dil tanımında belirtilen cümle birimlerini, hiyerarşik yapı içerisinde yukarıdan aşağıya doğru çözümlemeye çalışır.

Basit bir Türkçe sözdizimi kural kümesinin GGA yapısı

“Ali camı kırdı” cümlesi morfolojik analiz sonucunda “Ali cam + ı kır + dı”



-
- ❖ Bir sözdizimi kural kümesinin GGA biçiminde ifade edilmesin yararı, cümle birimlerinin bir kez tanımlandıktan sonra sonsuz kez kullanılıyor olmasıdır.
 - ❖ Bir sıfat tamlaması bir kez tanımlanıp birçok yerde kullanılabilir.
 - ❖ GGA'da döngüler bulunabilir. Bu özellik ile dilin kabul ettiği cümle yapıları sayısı sonsuza kadar artırabilir.

“yırtık kırmızı büyük top”

