

## Hafta 12

Bağlamdan Bağımsız Dil Örnek 2:

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

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

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

$$P: S \Rightarrow aAdd$$

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

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

$G_{4.2}$  tarafından türetilen tümcelerden birkaçını bulalım.

$$S \Rightarrow aAdd \Rightarrow abBccdd \Rightarrow abccdd$$

$$S \Rightarrow aAdd \Rightarrow aaAdddd \Rightarrow aabBccddd \Rightarrow aabccddd$$

$$S \Rightarrow aAdd \Rightarrow aaAdddd \Rightarrow aaAddddd \Rightarrow aaaAddddd$$

$$\Rightarrow aaabBccddddd \Rightarrow aaabBccddddd$$

$$\Rightarrow aaabbBcccdtdddd \Rightarrow aaabbcccdtdddd$$

Yukarıdaki örneklerden, bu bağlamdan-bağımsız dilbilgisi tarafından türetilen  $L(G_{4.2})$  dilinin aşağıdaki gibi tanımlanabileceği görülmektedir:

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

**Bağlamdan Bağımsız Dil Örnek 3:**

$$G_{4A} = \langle V_N, V_T, P, S \rangle$$

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

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

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

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

$G_{4A}$  tarafından türetilen tümcelerden birkaçını bulalım:

$$S \Rightarrow 0S0 \Rightarrow 01A10 \Rightarrow 010A110 \Rightarrow 01010110$$

$$S \Rightarrow 0S0 \Rightarrow 01S10 \Rightarrow 011S110 \Rightarrow 0110A1110 \\ \Rightarrow 01100A11110 \Rightarrow 011001011110$$

$$S \Rightarrow 0S0 \Rightarrow 00S00 \Rightarrow 000S000 \Rightarrow 0001A1000 \\ \Rightarrow 00010A11000 \Rightarrow 000100A111000 \\ \Rightarrow 0001000A1111000 \Rightarrow 0001000101111000$$

Yukarıdaki örneklerden, bu bağlamdan-bağımsız dilbilgisi tarafından türetilen  $L(G_{4A})$  dilinin aşağıdaki gibi tanımlanabileceği görülmektedir:

$$L(G_{4A}) = \{u v (v')^R u^R \mid u, v \in (0+1)^*, |u| \geq 1, |v| \geq 1\}$$

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

## TÜRETME AĞACI

Bir dilbilgisi tarafından türetilen tümcesel yapı ve tümceler aşağıdaki gibi gösterilebilir:

$$S \Rightarrow \beta_1 \Rightarrow \beta_2 \Rightarrow \beta_3 \dots \dots \dots \Rightarrow \beta_{n-1} \Rightarrow \beta_n \Rightarrow w \quad \beta_i \in V^*, w \in V_T^*$$

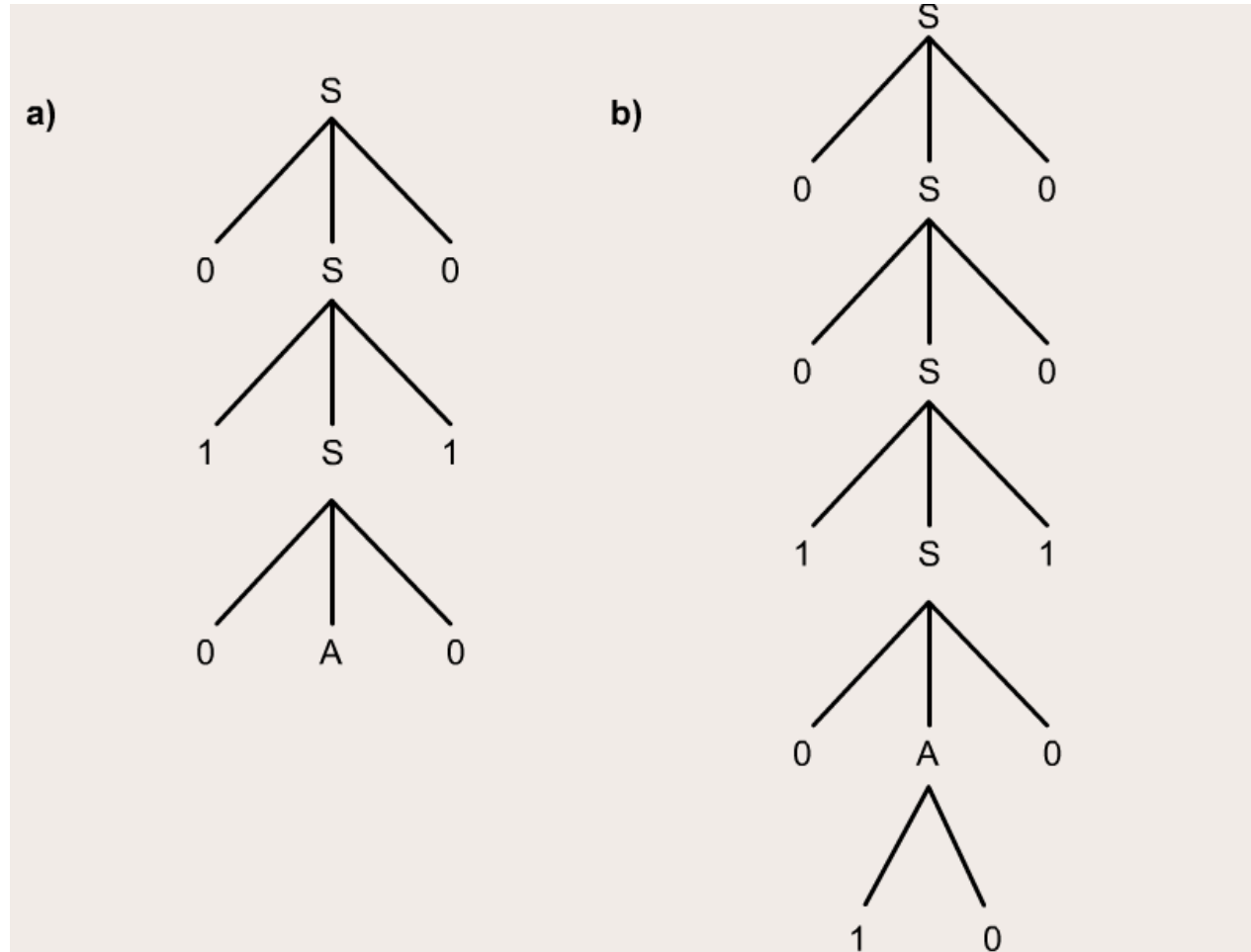
Tümcesel yapılarTümce

Bağlamdan-bağımsız dillerin her tümcesel yapısına ve her tümcesine bir **türetme ya da ayrıştırma ağacı (derivation or parsing tree)** karşı gelir. Örneğin **G4.4** tarafından türetilen:

$\beta = 010A010$  tümcesel yapısı ile

$w = 0010100100$  tümcesine

karşı gelen türetme ağaçları Çizim 4.1'de görülmektedir.



## YIĞITLI ÖZDEVİNİRLER (PUSHDOWN AUTOMATA)

Düzgün diller, düzgün ya da tür-3 dilbilgileri tarafından türetilen ve sonlu özdevinirler tarafından tanınan dillerdir. Düzgün dil, düzgün dilbilgisi ve sonlu özdevinirler birbiriyle ilişkili bir dil-dilbilgisi-makine üçlüsü oluşturur. Benzer biçimde, bu bölümün konusu olan yığıtlı özdevinirler ile bağlamdan-bağımsız dilbilgisi ve diller birbiriyle ilişkili bir dil-dilbilgisi-makine üçlüsü oluşturur.

**Yığıtlı özdevinirler, bağlamdan-bağımsız dilleri (CFL) tanıyan makine modelidir.**

Bağlamdan-bağımsız diller ise, bağlamdan-bağımsız dilbilgileri (CFG) tarafından türetilen dillerdir. Yığıtlı özdevinirlerin İngilizce karşılığı "**Pushdown Automata (PDA)**"dır. Modelin Türkçe adlandırılmasında, modelin yığıt içeren bir model olması ve pushdown automata'nın sözcük sözcük karşılığının çok anlamlı olmaması nedeniyle, "yığıtlı özdevinir" terimi tercih edilmiştir. Kısa ad olarak ise **PDA** kullanılacaktır.

### Yığıtlı Özdevinirin Temel Modeli

Biçimsel olarak, yığıtlı özdevinir (PDA) bir yedili olarak tanımlanabilir.

$PDA = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$

$Q$  : Sonlu sayıda durum içeren Durumlar Kümesi

$\Sigma$  : Sonlu sayıda giriş simgesinden oluşan Giriş Alfabeti

$\Gamma$  : Sonlu sayıda simge içeren Yığıt Alfabeti. Yığıt ve giriş alfabelerindeki simgelerin tümü ya da bir bölümü ortak olabileceği gibi iki alfabede hiç ortak simge bulunmayabilir de.

$q_0$  : Başlangıç durumu. ( $q_0 \in Q$ )

Başlangıç durumu durumlar kümesinin bir elemanı olduğuna göre  $Q$  boş olmayan bir kümedir.

$Z_0$  : Yığıt alfabetindeki simgelerden, Yığıt Başlangıç Simgesi olarak adlandırılan özel bir simge ( $Z_0 \in \Gamma$ ).  $Z_0$ , giriş alfabetinde yer almayan bir simgedir. Yığıt alfabetinin diğer tüm simgeleri ise giriş alfabetinde de yer alabilir.

$F$  : Uç durumlar kümesi

Durumlar kümesinin bir altkümesidir :  $F \subseteq Q$

$\delta$  : Geçiş ya da hareket işlevi (transition or move function)

Deterministik PDA modelinde geçiş işlevi,

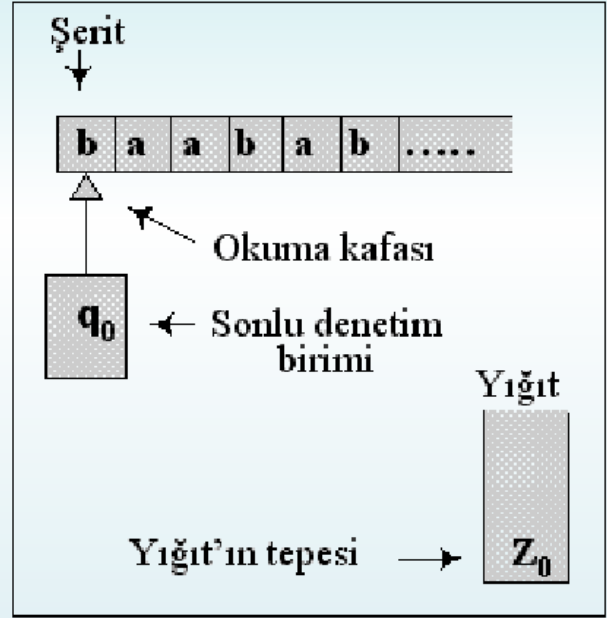
$[Q \times (\Sigma \cup \{\lambda\}) \times \Gamma]^*$ 'dan  $[Q \times \Gamma^*]$ 'a

bir eşleme oluşturur. Deterministik olmayan modelde ise geçiş işlevi,

$[Q \times (\Sigma \cup \{\lambda\}) \times \Gamma]^*$ 'dan  $[Q \times \Gamma^*]$ 'in sonlu altkümelerine bir eşleme oluşturur.

## PDA'nın Soyut Makine Modeli

- Hücrelerden oluşan ve her hücresinde bir giriş simgesi bulunan bir mıknatıslı şerit. Yalnız okunabilen şeridin sağ ucu sonsuzdur.
- Bir sonlu denetim birimi (SDB)
- Bir okuma kafası
- Bir yığıt



a) Başlangıç Görünümü

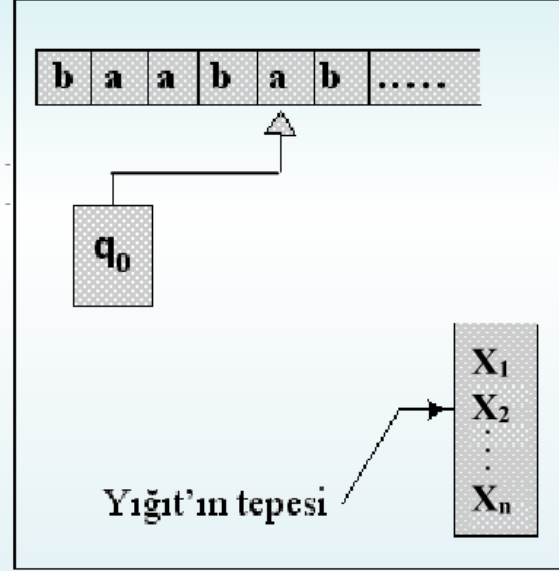
### Çalışma Şekli:

- Başlangıçta şerit üzerinde giriş simgelerinden oluşan bir dizgi (**w**) kayıtlıdır ve okuma kafası ilk (en soldaki) hücre üzerindedir.
- Yığıtın her elemanına bir yığıt simgesi yazılabilmektedir ve başlangıçta yığıtta **Z<sub>0</sub>** başlangıç simgesi kayıtlıdır.
- Makinenin 2 türlü hareketi vardır. Bu hareketler normal hareket ve **λ**-hareketi (**λ**-move) olarak adlandırılır.

## Normal Hareket

PDA'nın yandaki çizimde görülen soyut makine modeli aşağıda açıklanan biçimde çalışır.

- Şeritten bir simge okunur.
- Okuma kafası bir sağdaki hücreye geçer.
- Yığıtın en üstündeki simge silinir (pop).
- Yığıtın üstüne, yığıt simgelerinden oluşan bir dizgi ( $\alpha \in \Gamma$ ) eklenir (push).
- Yığıtı eklenen dizgi boş ( $\lambda$ ) da olabilir. Eğer yığıtı eklenen dizgi birden çok yığıt simgesinden oluşan bir dizgi ise ( $\alpha = X_1X_2...X_n \mid X_1, X_2, ..., X_n \in \Gamma$ ) bu simgelerin en soldaki ( $X_1$ ) simge en üste gelecek biçimde yığıtı ekleneceği varsayılacaktır.

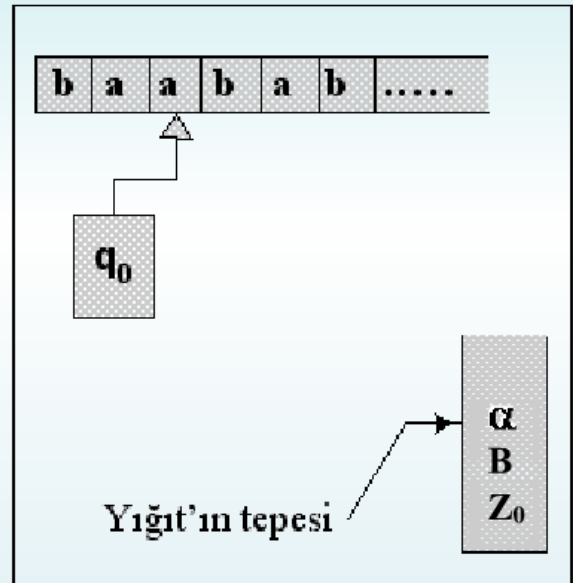


b) 3 Hareket Sonraki Görünüm

## Lambda Hareketi

$\lambda$ -hareketinde ise aşağıdaki işlemler yapılır:

- Şeritten bir simge okunmaz ve okuma kafası yer değiştirmez. Dolayısıyla  $\lambda$ -hareketi şeritteki simgeden bağımsız bir harekettir.
- Yığıtın en üstündeki simge silinir (pop).
- Yığıtın üstüne, yığıt simgelerinden oluşan bir dizgi ( $\alpha \in \Gamma^*$ ) eklenir (push). Yığıtı eklenen dizgi boş ( $\lambda$ ) da olabilir.



b) 3 Hareket Sonraki Görünüm