

CYK Algoritması

Dilin ayrıştırılmasında kullanılan dinamik bir ayrıştırma algoritmasıdır. Kurallarda kullanılan değişkenlerin belirli bir karakter kriterine denk gelip gelmediğini bir tablo üzerinde arama yaparak bulmaktadır.

Temel yakalanan bir karakter için bulunan kural değişkenlerini hafızada tutarak sonraki adımlar için hızlı bir çözüm sunar.

CYK algoritması doğal dilin ayrıştırılmasında kullanılan en temel algoritmadır.

~~Örnek~~ Aşağıda verilen Chomsky N.F. ait gramer ve "baaba" karakter katmanı için CYK algoritmasını kullanarak ağaç yapısını elde ediniz.

$$S \rightarrow AB \mid BC$$

$$A \rightarrow BA \mid a$$

$$B \rightarrow CC \mid b$$

$$C \rightarrow AB \mid a$$

① Önce bir tablo oluşturarak köşegenlere her bir karaktere karşılık gelen kural adını yazalım.

Burada b karakteri için B kuralı kullanılırken a için $\{A, C\}$ kuralları kullanılmıştır.

	b	a	a	b	a
b	B				
a		A, C			
a			A, C		
b				B	
a					A, C

Önce bir tablo oluşturarak köşegenlere her bir karaktere karşılık gelen kural adını yazalım.

Soldan sağa ve aşağıdan yukarı doğru bir yol izlersek A ve C'yi birleştiren 2. ve 3. satırlar için bir üste B yazabiliriz.

Benzer şekilde A ve B için 3. satır ve 4. satırda S kuralı ve ayrıca C kuralı ile birleşir.

	b	a	a	b	a
b	B	S, A			
a		A, C	B		
a			A, C	S, C	
b				B	S, A
a					A, C

Soldan sağa ve yukarı

Pushdown Otomat

Kararsiz sonlu yigit otomatini tanimlarsak:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$$

$Q \rightarrow$ durumlar

$\Sigma \rightarrow$ alfabe

$\Gamma \rightarrow$ yigit alfabeti

$q_0 \rightarrow$ baslangic durumu

$z \rightarrow$ yigit baslangici

$F \rightarrow$ son durumlar

Yigit otomati icin tanimlamaya gore

$$Q = \{q_0, q_1, q_2, q_3\}$$

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

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

$$z = 0$$

$$F = \{q_3\}$$

• Durumlar : q_0, q_1, q_2, q_3

Girdi alfabeti : a, b

Yigit alfabeti : $0, 1$

Yigit baslangici 0 barındırıyor,

Son durum q_3

Baslangic belirtilmemiş.

$$\delta(q_0, a, 0) = \{(q_1, 10), (q_3, \lambda)\},$$

$$\delta(q_0, \lambda, 0) = \{(q_3, \lambda)\},$$

$$\delta(q_1, a, 1) = \{(q_1, 11)\},$$

$$\delta(q_1, b, 1) = \{(q_2, \lambda)\},$$

$$\delta(q_2, b, 1) = \{(q_2, \lambda)\},$$

$$\delta(q_2, \lambda, 0) = \{(q_3, \lambda)\}.$$

- 1. kurala göre q_0 durumunda iken girdi alfabesi a ve yığıtın en üstteki elemanı 0 ise
 - q_1 durumuna geçilir ve stack içinde 1 (eklenen sola) eklenir
 - q_3 durumuna geçilir ve yığıtın 0 çıkarılır Burada λ görüldüğümüzde yığıtın çıkarma işlemi yapılmıştır
- 3. kurala göre q_1 durumunda iken girdi alfabesi a ve yığıtın en üstteki elemanı 1 ise ;
 - q_1 durumuna geçilir ve yığıtın 1 eklenir.

Örnek

Geriye kuralları verilen pushdown otomati çizil. hangi dili yatabilabilir gösterin.

$$\delta(q_0, a, 0) = \{(q_1, 10), (q_3, \lambda)\}, \rightarrow a \text{ gördüğünde sonlu durum}$$

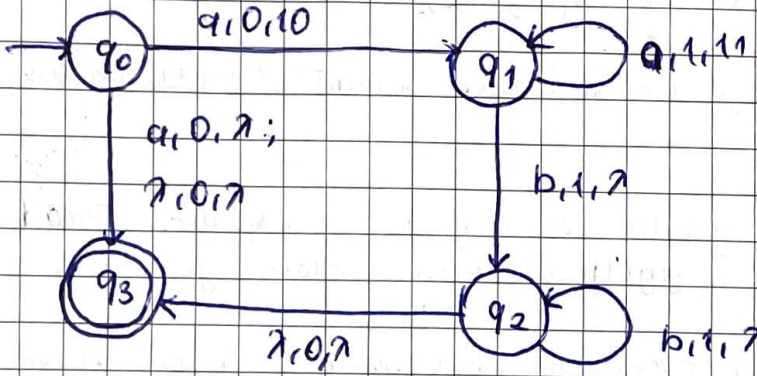
$$\delta(q_0, \lambda, 0) = \{(q_3, \lambda)\},$$

$$\delta(q_1, a, 1) = \{(q_1, 11)\}, \rightarrow a \text{ gördükçe 1 ekle}$$

$$\delta(q_1, b, 1) = \{(q_2, \lambda)\},$$

$$\delta(q_2, b, 1) = \{(q_2, \lambda)\}, \rightarrow b \text{ gördükçe 1 çıkar}$$

$$\delta(q_2, \lambda, 0) = \{(q_3, \lambda)\}.$$

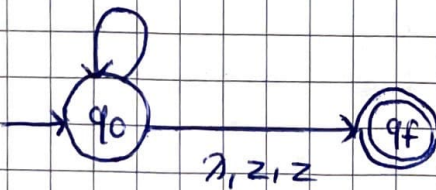


$$L = \{a^n b^n : n \geq 0\} \cup \{a\}$$

Örnek

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

Bu otomat ile a gördükçe 1 çıkartıp 0 ekleriz, b gördükçe 0 çıkartıp 1 ekleriz.



$$a, 0, 00 ; b, 1, 11$$

$$a, z, 0z ; b, 0, \lambda$$

$$b, z, 1z ; a, 1, \lambda$$