

Otomata Teorisi (BİL 2114)

Fırat İsmailoğlu

Hafta 7: İçerikten Bağımsız Diller (II. Bölüm)



ör. google aramasi: " bir * olsam "

arama sonuclari: " bir marti olsam, " " bir kahve fincan olsam ", " bir bulut olsam ", " bir zengin olsam ", " bir fidan olsam " ...

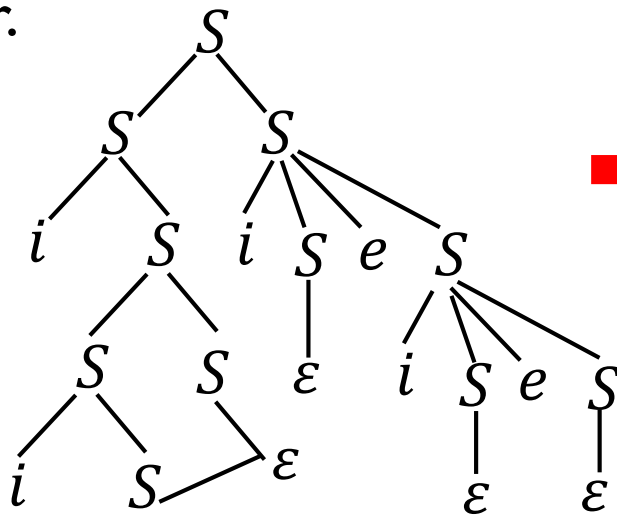
ör. google aramasi: "(nukleer|kirmizi)(baslikli|enerji)"

aramasi " nukleer baslikli ", " nukleer enerji ", " kirmizi baslikli " ve " kirmizi enerji " aramalarini bir defada arar.

ör. $G = (\{S\}, \{i, e\}, R, S)$ ve R kuralları

$$S \rightarrow \varepsilon | SS | iS | iSeS$$

grammeri bir programlama dilinin sintaksina uygun if-else'ler uretir.

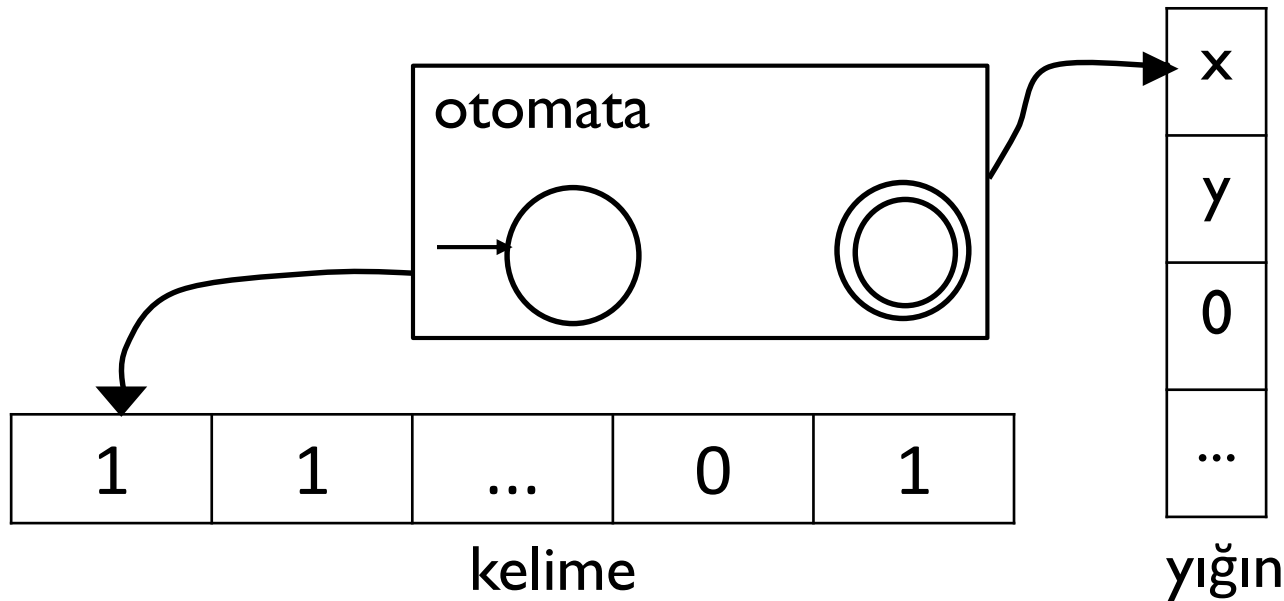


iiieie kelimesini uretir ki bu uygun bir if-else blogudur.



Pushdown Otomata (PDO) Giriş

Yandaki resimdeki alete tabaklar en usten birer birer konur, ve yeni tabaklar geldikce eski tabaklar asagi dogru itilir. Buradan bir tabak almak istedigimizde en usttekini aliriz. Bunun sonucunda bir alttaki tabak en uste cikar. Bu kuralin adi son gelen ilk giderdir (Last In First Out).



Pushdown Otomata (PDO) Giriş

PDA harfleri yığına yazabilir \Rightarrow Yazılan harf yığının en üstüne konur, yığındaki diğer harfler aşağı itilir.

PDA yığından harfler okuyabilir \Rightarrow Okunan harf yığının en üsteki harfidir, bu harf okunduğunda yığından çıkartılmış olur, böylece yığındaki diğer harfler bir basamak üste çıkar.

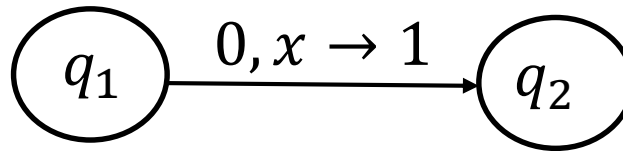
Yığının sonuz miktarda bilgi tutabildiği varsayılır. Yani yığına istediğimiz kadar harf koyabiliriz.

PDA'nin ikinci en önemli özelliği nondeterministik olmasıdır. Böylece aynı anda birden fazla durum aktif olabilir.



Pushdown Otomata Geçişler

En geniş anlamda, bir PDA'da bir durumdan diğer bir duruma, o anki durum, okunan kelimenin harfi ve yığının en üstteki harfi dikkate alınarak geçilir.



Yukarıdaki örnekte, q_1 durumunda iken eğer kelimenin sıradaki harfi 0 ve yığının en üstteki harfi x ise q_2 'ye geçeriz.

Burada x harfini yığından okuruz, yani böylece bu harfi yığından çıkarmış oluruz.

Daha sonra yığının en üstüne 1 harfini koyarız. (Yığından x' i alır, yerine 1 koyarız).



Pushdown Otomata Geçişler

Epsilondan Kaynaklanan Bazı Özel Haller:

1. $\varepsilon, x \rightarrow 1$ ise okunan kelimenin harfi ne olursa olsun yığının en üstünde x harfi varsa q_1 'den q_2 'ye geçiş sağlanır. Yığından x çıkartılır. 1 yığına eklenir (en üstüne konur).
2. $0, \varepsilon \rightarrow 1$ ise yığına bakmaksızın 0 harfini okuduğumuzda geçiş yaparız ve 1 harfini yığının en üstüne koyarız (yığından bir şey çıkartılmaz).
3. $0, x \rightarrow \varepsilon$ ise okunan kelimedeki sıradaki harf 0 ise ve yığının en üstündeki harf x ise geçiş sağlanır. Bu geçiş sonucunda yığından (en üstünden) x harfi çıkartılır. Yığına yeni bir harf konmaz.
4. $\varepsilon, \varepsilon \rightarrow \varepsilon$ ise okunan kelimeye ve yığına bakılmaksızın geçiş olur. Bu geçiş sonucunda yığından bir harf okunmaz (yığından bir harf çıkmaz) ve yığına bir harf konmaz.



Pushdown Otomata Örnek

$L = \{0^n 1^n | n \geq 1\}$ dilini tanıyan PDO tasarlıyalım.

Bu dilin düzenli olmadığını Pumping lemma yardımıyla görmüştük.

Bundan başka, 1 harflerini okumaya başlamadan önce daha önce kaç tane 0 harfi okuduğunu hatırlaması gerektiği içinde bu dili tanıyan bir NSO tanımlayamamıştık.

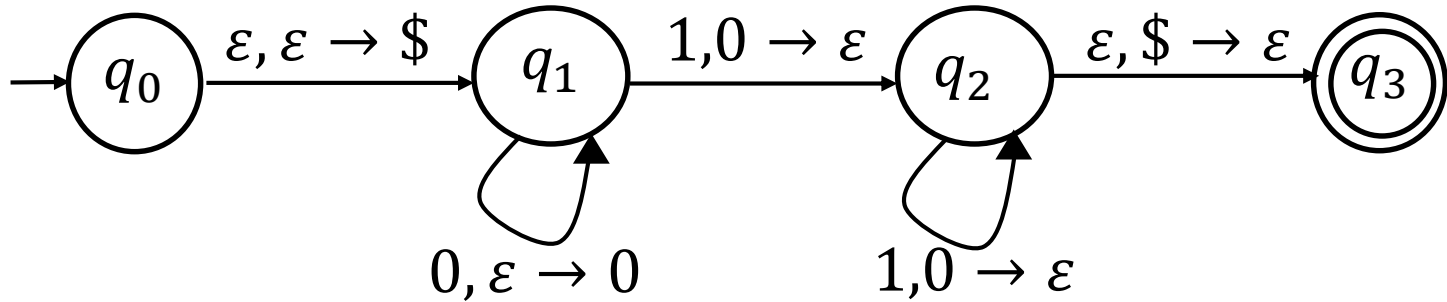
Not. PDO'larda \$ özel bir harftir (simgedir). Yığının en sonundaki (en dibindeki) harfi temsil eder. Bu harf ürettiğimiz yığınla birlikte gelmez. Bunu bilhassa yığına eklemek gerekir.

Bir yığındaki \$ harfini okuduğumuzda (yani yığından bu harfi çıkarttığımızda) yığının içinde hiçbir harf kalmaz. Yığın boşalmış olur.

Not. Bazı kaynaklarda \$ yerine Z yada Z_0 da kullanılır.



Pushdown Otomata Örnek I



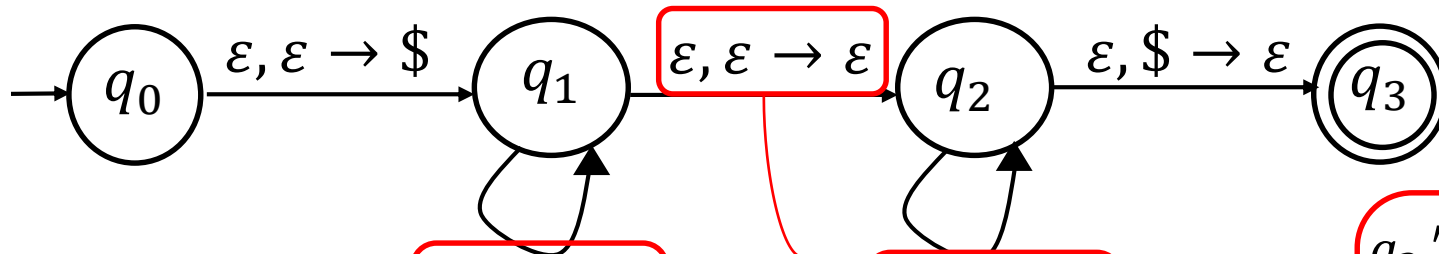
Yukarıdaki PDO elimizde varken $w = 0^21^2$ kelimesini okuyalım.

Okunan Harf	Aktif Durum	Yığın
	q_0, q_1	\$
0	q_1	0, \$
0	q_1	0,0, \$
1	q_2	0, \$
1	q_2	\$
	q_3	—



Pushdown Otomata Örnek II

$L = \{ww^R \mid w \in \{0,1\}^*\}$ dilini tanıyan PDO tasarlıyalım. Burada w^R w kelimesinin tersten yazılmış halidir. Örneğin $w = 01$ iken $w^R = 10$ olur. Su halde L dili çift uzunluktaki palindromların dilidir.



q_1 'de kelimeden 0 harfi geldikçe q_1 'de kalınır, ve yığına 0 harfi eklenir.
 q_1 'de kelimeden 1 harfi geldikçe q_1 'de kalınır, ve yığına 1 harfi eklenir.

q_1 'de iken kelimeden ve yığından hiçbir harf okumadan ve yığına hiçbir şey eklemeden q_2 'ye geçilir.

q_2 'de iken kelimeden 0 (1) gelirse ve yığının en üstünde 0 (1) varsa, yığından 0 (1) çıkartılarak yine q_2 'de kalınır.

Pushdown Otomata Formal Gosterimi

Bir pushdown otomata 6-li siradir $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$. Burada

1. Q durumlarin kumesidir,
2. Σ alfabe (kelime icin),
3. Γ yiginin kullandigi alfabe,
4. $\delta: Q \times \Sigma_{\varepsilon} \times \Gamma_{\varepsilon} \rightarrow \mathcal{P}(Q \times \Gamma_{\varepsilon})$, ge
5. $q_0 \in Q$ baslangic durumu,
6. $F \subseteq Q$ final durumlarinin kumesidir.

Not 1. Yiginin kullandigi alfabe Γ (yigina ekledigimiz harfler) kelimenin alfabesinden (Σ) farkli olabilir. Ornegin $\$ \in \Gamma$.

Not 2. Gecis fonksiyonun 3 tane girisi vardir: $Q, \Sigma_{\varepsilon}, \Gamma_{\varepsilon}$. Bu demektirki bir PDO'da, bir durumdan digerine gecebilmek icin, icinde bulunan durum, kelimenin o anki harfi ve yiginin o andaki en ustteki harfi dikkate alinir. Gecis icin bunlarinin ucunun de musait olmasi gerekir.

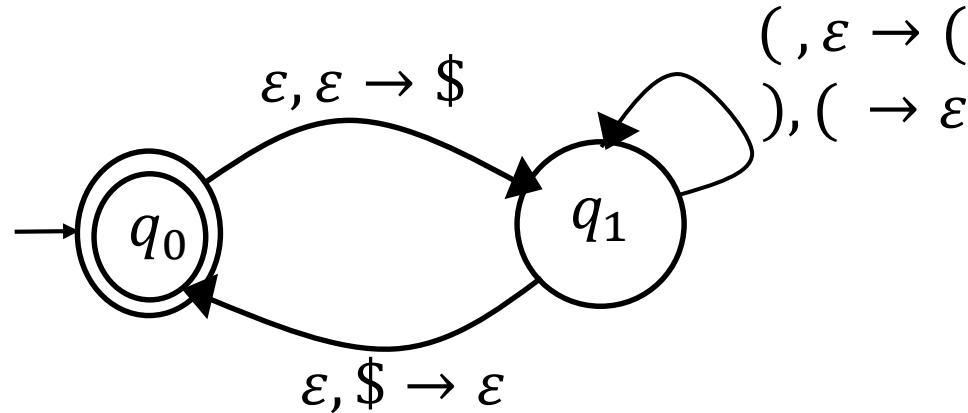


Pushdown Otomata Formal Gosterimi

Not 3. Gecis fonksiyonun deger kumesi $\mathcal{P}(Q \times \Gamma_\varepsilon)$ dir. Gecis sonunda birden fazla duruma gecebilecegimizi ve birden fazla harfi yigina koyabilecegimizi ifade eder.

or. $\{(q_1, 0), (q_2, \varepsilon)\} \in \mathcal{P}(Q \times \Gamma_\varepsilon)$.

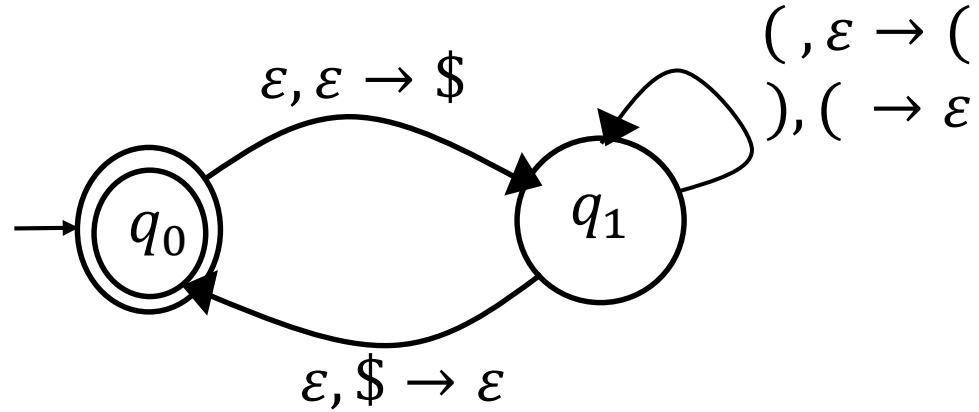
or. $M = (\{q_0, q_1\}, \{ (,) \}, \{ \$, (,) \}, \delta, q_0, \{q_0\})$ PDO'su su sekilde olsun:



Bu PDO, düzgün olarak ic ice geçmiş parentezlerin dilini taniyan PDO'dur.



Bu PDO için $w = (())()$ kelimesini okuyalım:



Okunan Harf	Aktif Durum	Yığın
	q_0, q_1	\$
(q_1	(, \$
(q_1	(, (, \$
)	q_1	(, \$
)	q_0	—
	q_0, q_1	\$
(q_1	(, \$
)	q_0	—



JFLAP ile Pushdown Otomata

JFLAP → Pushdown Automata → Multiple Character Input
adimlari izlendikten sonra daha once sonlu otomatalarda yaptigimiz
gibi, durumlar ve gecisler olusturulur.

JLAP'da yiginin son elemanini simgelemek icin \$ yerine Z kullanilir.
Ve Z simgesi, JFLAP'da bir PDO olusturudugumuzda otomatik
olarak gelir.

$L = \{0^n 1^n | n \geq 1\}$
(ϵ yerine
 λ kullaniliyor.)

