

Otomata Teorisi

Fırat İsmailođlu, PhD

Hafta 8: İçerikten Bađımsız Diller (III. Bölüm)



Hafta 8

Plan

1. Pushdown Otomata (PDO) Giriş
2. PDO Geçişler
3. PDO Örnekler
4. PDO'nun Formal Gösterimi
5. JFLAP ile PDO
6. PDO ile İçerikten Bağımsiz Dillerin Denkliği



Pushdown Otomata (PDO) Giriş

Pushdown kelimesinin karsiligi olarak asagi itimli kelimesini dusunebiliriz.

Pushdown otomata aslinda bir nondeterministik sonlu otoma (NSO) dur. Konvensiyonel NSO'dan farki bir **yığına** (stack) sahip olmasidir.

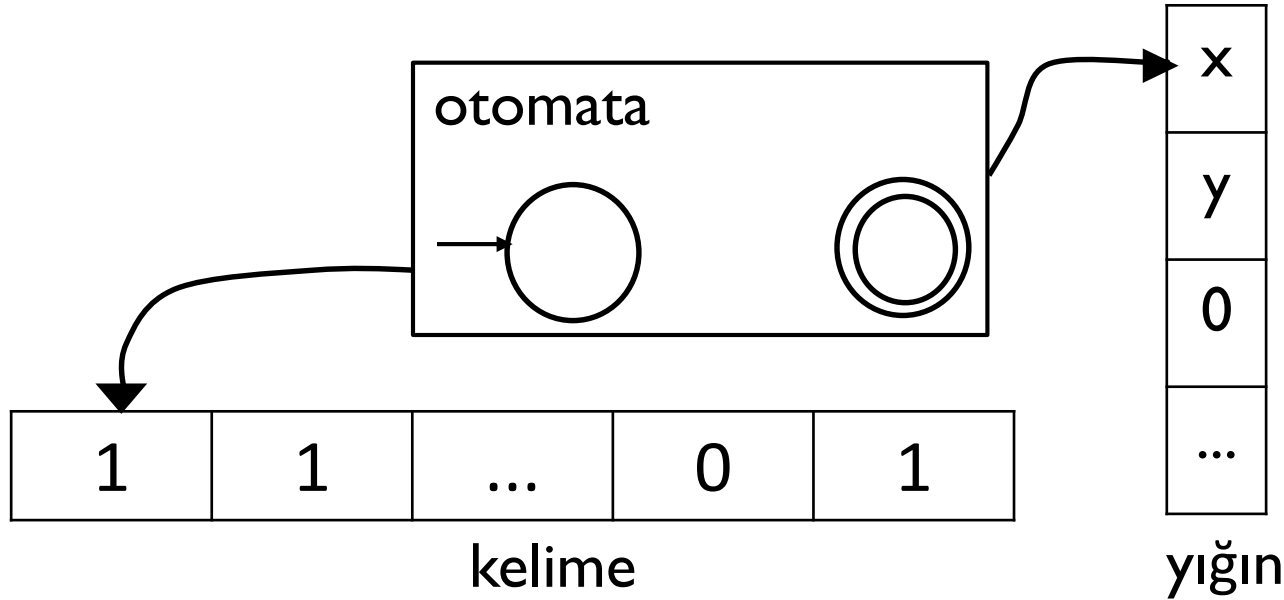
Bu yığıni restoranlarda tabaklarin ust uste konuldugu ve altta bu tabakalari yukariya dogru iten bir mekanizmanin oldugu alete benzetebiliriz.

PDO'lar icerikten bagimsiz grammerlerin gücüne sahiptirler. İcerikten bagimsiz grammerler icerikten bagimsiz dilleri uretirler, PDO'lar ise icerikten bagimsiz dilleri tanırlar.



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.

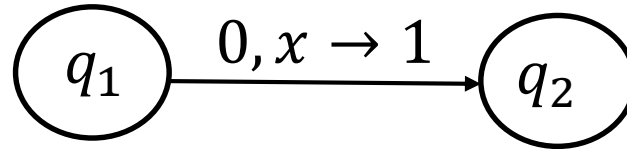
Yığının sonuz miktarda bilgi tutabildiği varsayılır. Yani yığına istediğimiz kadar harf koyabiliriz.

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 Geçiřler

Epsilon dan 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 geçiř saęlanır. Yığından x çıkartılır. 1 yığına eklenir (en üstüne konur).
2. $0, \varepsilon \rightarrow 1$ ise yığına bakmaksızın 0 harfini okuduęumuzda geçiř yaparız ve 1 harfini yığının en üstüne koyarız (yıęından bir řey çıkartılmaz).
3. $0, x \rightarrow \varepsilon$ ise okunan kelimedeki sıradaki harf 0 ise ve yığının en üstündeki harf x ise geçiř saęlanır. Bu geçiř sonucunda yığından (en üstünden) x harfi çıkartılır. Yığına yeni bir harf konmaz.
4. $\varepsilon, \varepsilon \rightarrow \varepsilon$ ise okunan kelimeye ve yığına bakılmaksızın geçiř olur. Bu geçiř sonucunda yığından bir harf okunmaz (yıęından bir harf çıkmaz) ve yığına bir harf konmaz.



Pushdown Otomata Örnek

$L = \{0^n 1^n \mid n \geq 1\}$ 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.

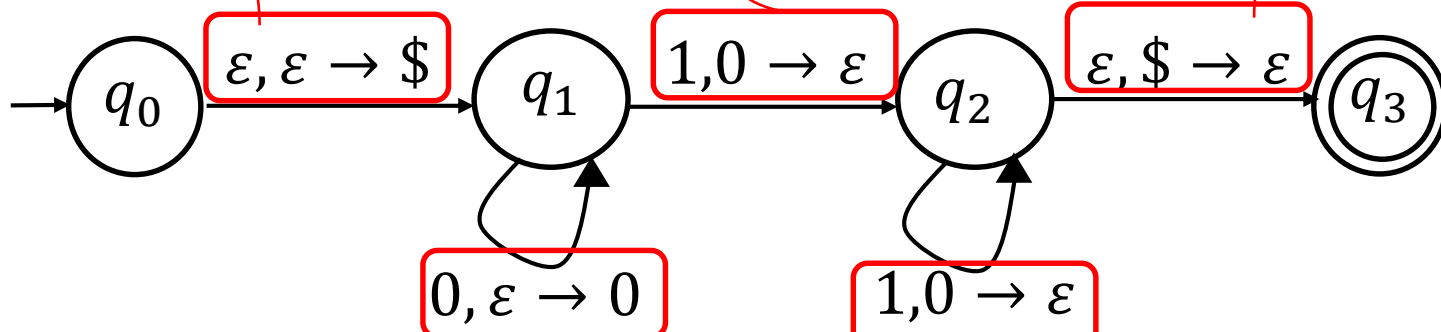


Pushdown Otomata Örnek I

q_0 'da harf okumaksizin ve yığına bakılmaksizin q_1 'e gidilir. \$ yığına eklenir.

q_2 'de iken eğer yığının en üstünde \$ varsa yani yığının en alt harfi varsa q_3 'e geçilir.

q_1 'de iken kelimededen 1 harfi gelirse yığından bir 0 okuyarak q_2 'ye geceriz

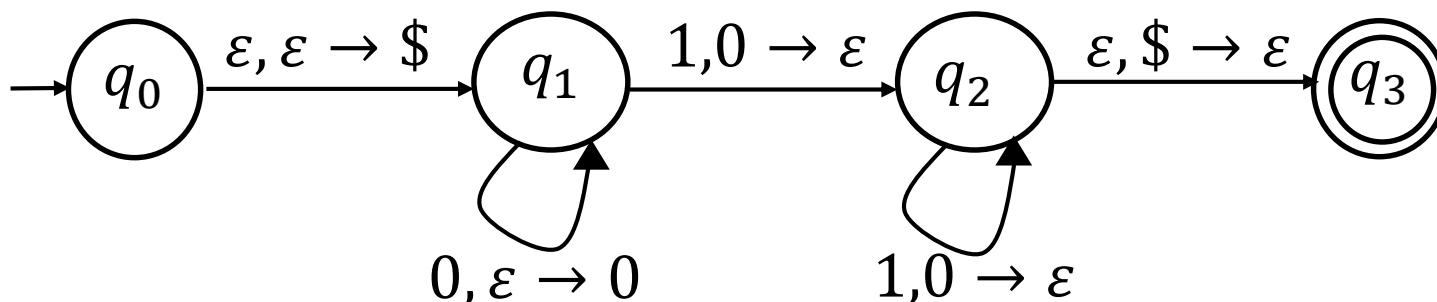


q_1 'de kelimededen 0 harfi geldikçe q_1 'de kalınır, ve her bir 0 harfi için bir 0 harfi yığına eklenir.

q_2 'de iken eğer kelimededen 1 harfi gelirse, bu 1'lere karşılık, yığından bir tane 0 okunur (yani çıkartılır) ve yığına bir şey eklenmez.



Pushdown Otomata Örnek I

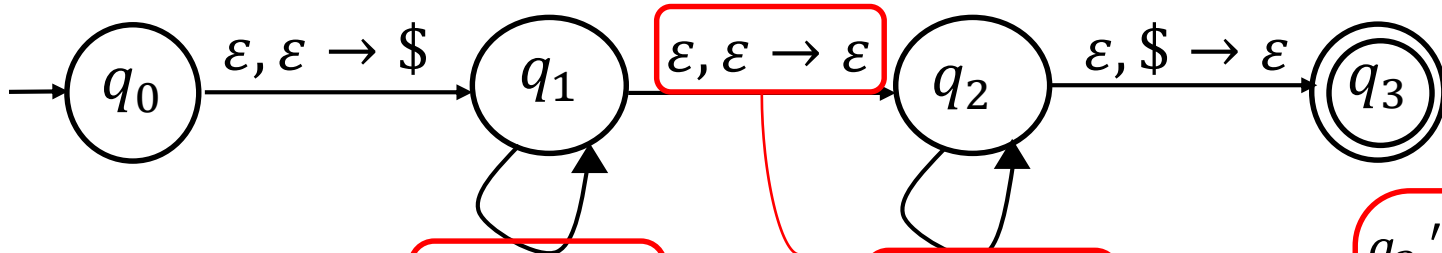


Yukarıdaki PDO elimizde varken $w = 0^21^2$ kelimesini okuyalım.

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

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

q_2 'de iken kelimedenden 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 c musait olmasi gerekir.

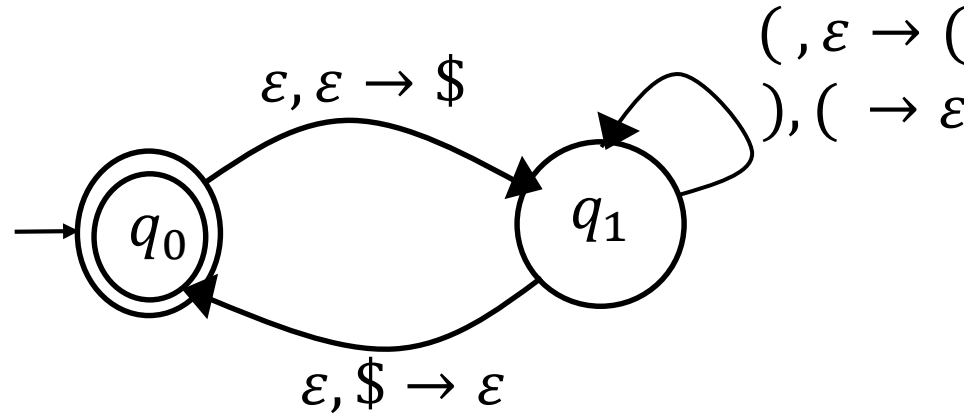


Pushdown Otomata Formal Gosterimi

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

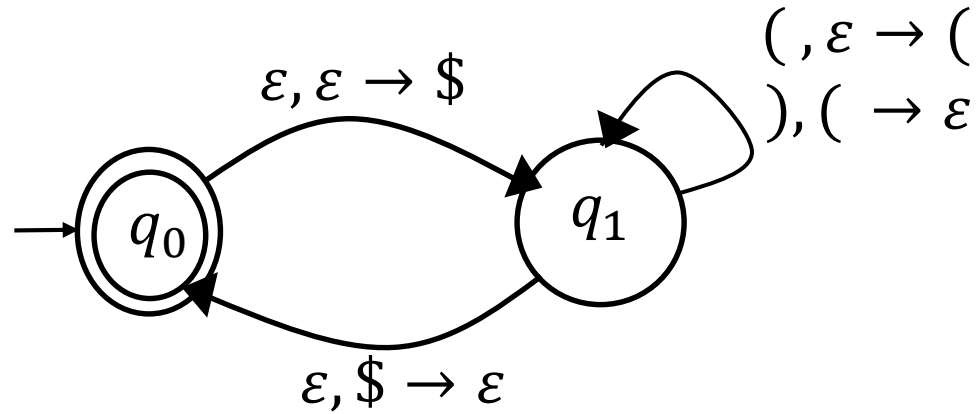
or. $M = (\{q_0, q_1\}, \{(\cdot, \cdot)\}, \{\$, (\cdot)\}, \delta, q_0, \{q_0\})$ PDO'su su sekilde olsun:



Bu PDO, düzgün olarak ic ice geçmiş parentezlerin dilini tanıyan 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	—

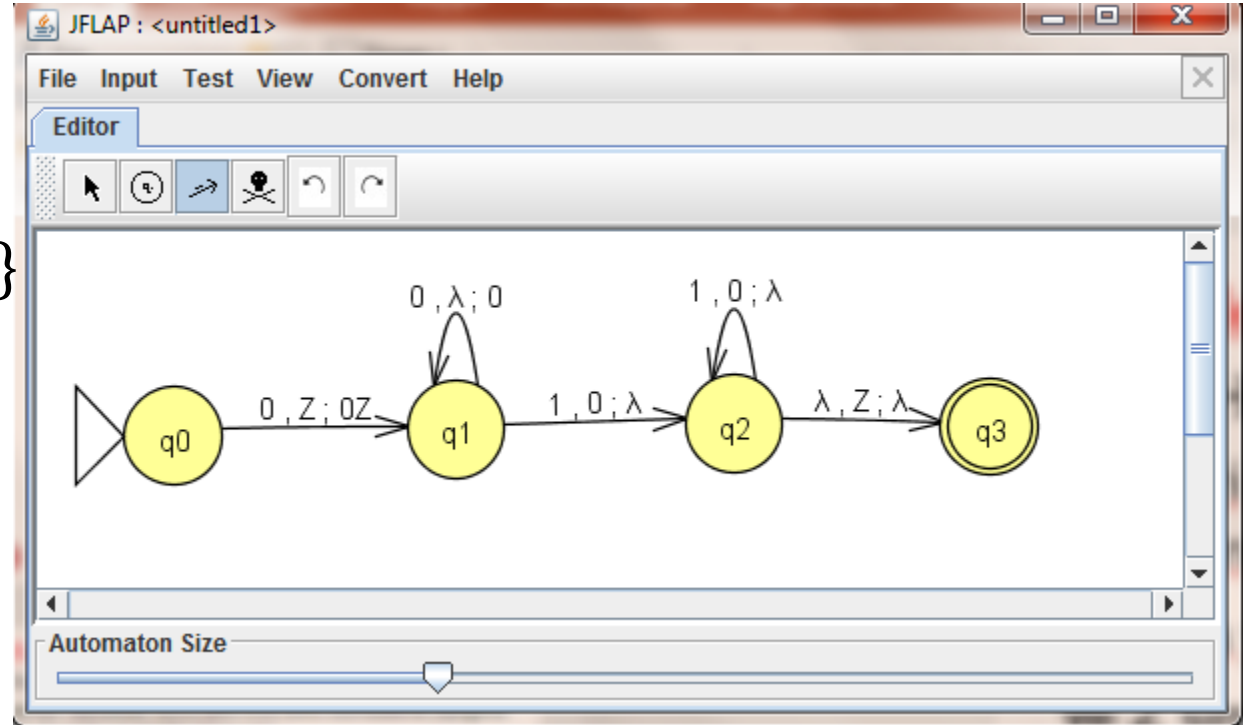


JFLAP ile Pushdown Otomata

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.)



PDO ile İcerikten Bağımsiz Dillerin Birbirine Denkliği

Teorem: Bir dilin içerikten bağımsiz olması için gerek ve yeter şart o dilin bir pushdown otomata tarafından tanınmasıdır.

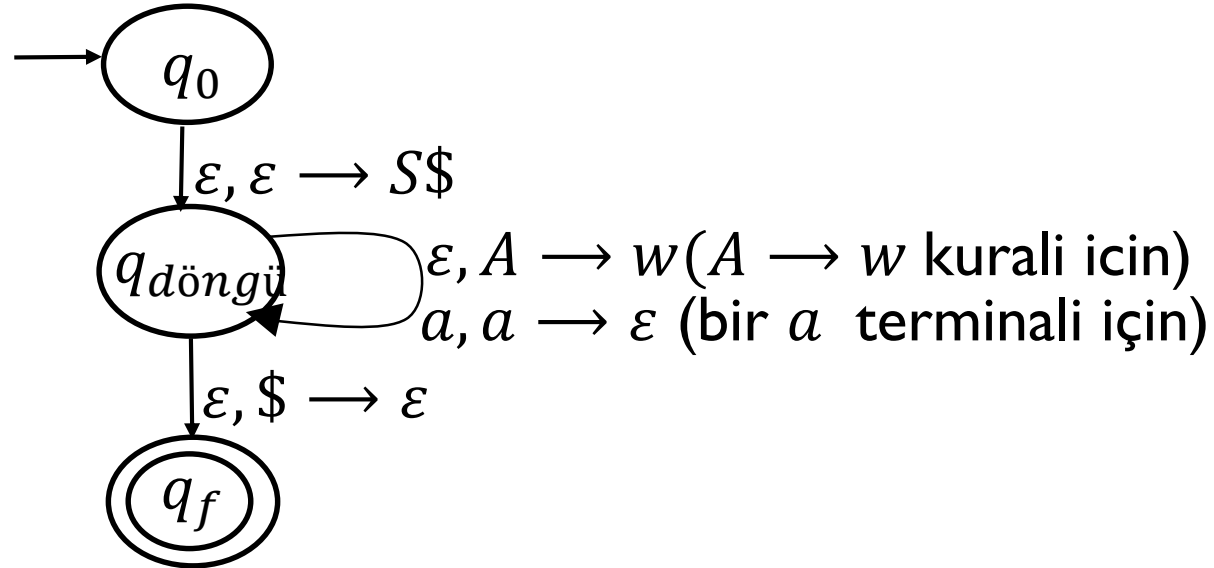
Lemma 1: Bir dil içerikten bağımsiz ise onu tanıyan bir pushdown otomata vardır.

Bir içerikten bağımsiz grammerden bir PDO elde ederek, verilen bir kelimenin bu grammerden üretilmediğini kolayca görebiliriz. Böylece bu iş için CYK algoritmasına alternatif bir yöntemimiz olur.

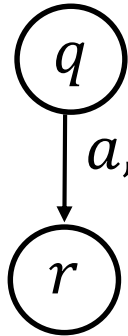


İçerikten Bağımsız Grammerlerden Pushdown Otomataya

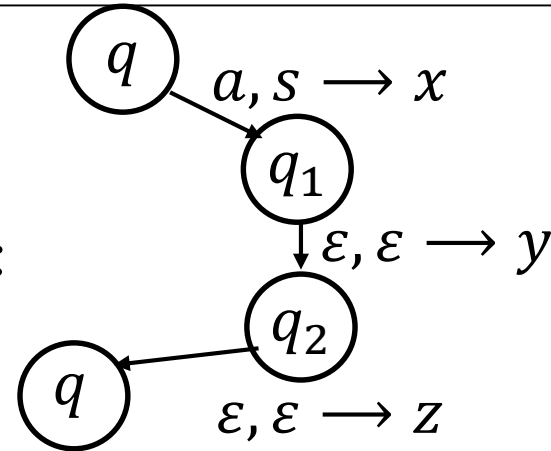
Bir içerikten bağımsız grammerden türetilen PDO'nun üç durumu vardır: q_0 , $q_{dönü}$ ve q_f . Bu PDO şu formdadır:



Not:



şu şekilde de gösterilebilir:



İçerikten Bağımsız Grammerlerden Pushdown Otomataya

Bir içerikten bağımsız grammerden PDO türetmek için aşağıdaki prosedür izlenir.

1. Bir önceki şekildeki gibi q_0 , $q_{dönğü}$ ve q_f durumları çizilerek birbirine bağlanır.
2. Yiğina başlangıç değişkeni S ve $\$$ eklenir. ($\epsilon, \epsilon \rightarrow S\$$)
3. Değişkenler (nonterminaller) yigindan çıkarılarak yerlerine bu değişkene karşılık gelen kelime eklenir. Örneğin bir $A \rightarrow w$ kuralı için yigindan A çıkarılıp w kelimesi en sağ sembolden başlayarak eklenir. Ortaya çıkan geçişler $q_{dönğü}$ 'ye eklenir.
4. Yigindan terminaller çıkarılır. Örneğin bir a terminali için, $a, a \rightarrow \epsilon$ geçisi $q_{dönğü}$ 'ye eklenir.
5. $\$$ yigindan çıkarılarak ($\epsilon, \$ \rightarrow \epsilon$) final durumu q_f' ye gidilir.



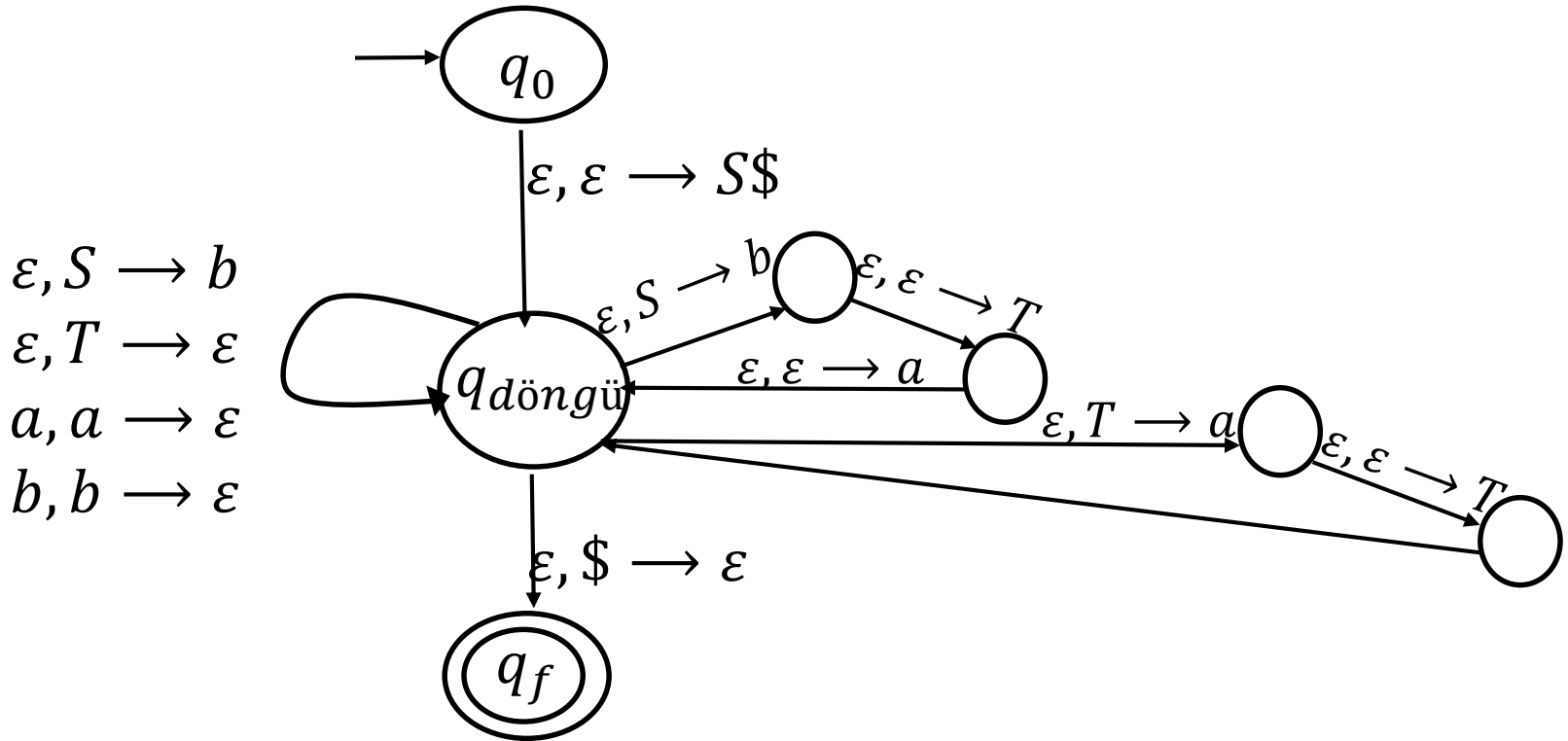
İçerikten Bağımsız Grammerlerden Pushdown Otomataya

ör. $G = (\{S, T\}, \{a, b\}, R, S)$ ve R kuralları

$$S \rightarrow aTb|b$$

$$T \rightarrow Ta|\varepsilon$$

İle verilen grammerden PDO elde edelim:



İçerikten Bağımsız Grammerlerden Pushdown Otomataya

ör. $G = (\{E\}, \{a, *, +, (,)\}, R, E)$ ve R kuralları

$$S \rightarrow (E) \mid E + E \mid E * E \mid a$$

İle verilen grammerden PDO elde edelim:

