Biçimsel Diller ve Otomata Teorisi

Hafta 12: İçerikten Bağımsız Diller (II. Bölüm)

Plan

1. Pushdown Otomata (PDA) Giriş
2. PDA Geçişler
3. PDA Ornekler
4. PDA'nun Formal Gösterimi
5. JFLAP ile PDA
6. PDA ile Icerikten Bagimsiz Dillerin Denkligi

Pushdown Otomata (PDA) Giriş

Pushdown kelimesinin karsiligi olarak asagi itimli kelimesini dusunebiliriz.

Pushdown otomata aslinda bir *nondetermisintik* sonlu otoma (NFA) dır. Konvensiyonel NFA'dan farki bir ***yığına*** (stack) sahip olmasidir.

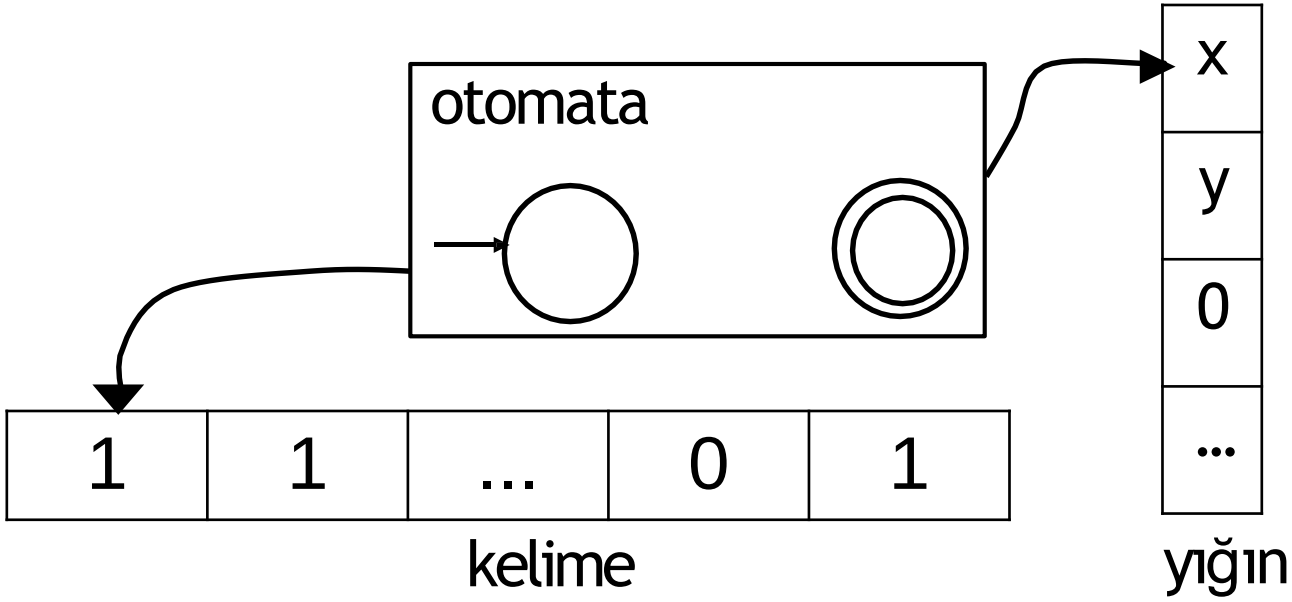
Bu yığını restoranlarda tabaklarin ust uste konuldugu ve altta bu tabakalari yukariya dogru iten bir mekanizmanin oldugu alete benzetebiliriz.

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

Pushdown Otomata (PDA) 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 (PDA) Giriş

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

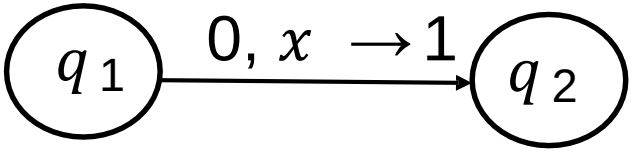
PDA yığından harfler okuyabilir $\Longrightarrow$ Okunan harf yığının en üsteki harfidir, bu harf okunduğunda yığından çıkartılmış olur, böylece yığındaki diğer harfler bir basamak üste çıkar.

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

PDA'nin ikinci en onemli ozelligi nondetermisnistik olmasıdır. Böylece aynı anda birden fazla durum aktif olabilir.

Pushdown Otomata Geçişler

En geniş anlamda, bir PDA'da bir durumdan diger 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 koyariz. (Yiğindan x' i alir, yerine 1 koyariz).

Pushdown Otomata Geçişler

<u>Epsilondan Kaynaklanan Bazı Özel Haller</u>:

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 bır 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

$L = \{0^n 1^n \mid n \geq 1\}$ dilini tanıyan PDA tasarlıyalım.

Bu dilin düzenli olmadığını Pumping lemma yardımıyla görmüştük.

Bundan başka, 1 harflerini okumaya başlamadan önce daha önce kaç tane 0 harfi okuduğunu hatirlaması gerektiği içinde bu dili taniyan bir NFA tanımlayamamıştik.

Not. PDA'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 çikarttığımızda) yığının içinde hiçbir harf kalmaz. Yığın boşalmış olur.

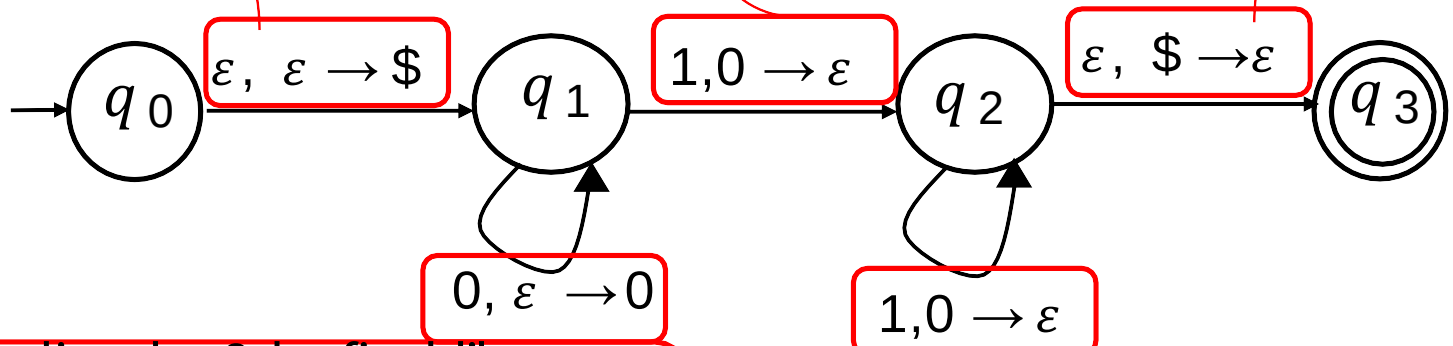
Not. Bazı kaynaklarda $ yerine Z yada Z_0 da kullanılır.

Pushdown Otomata Örnek 1

q_0 'da harf okumaksizin ve yiğina bakilmaksizin 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.

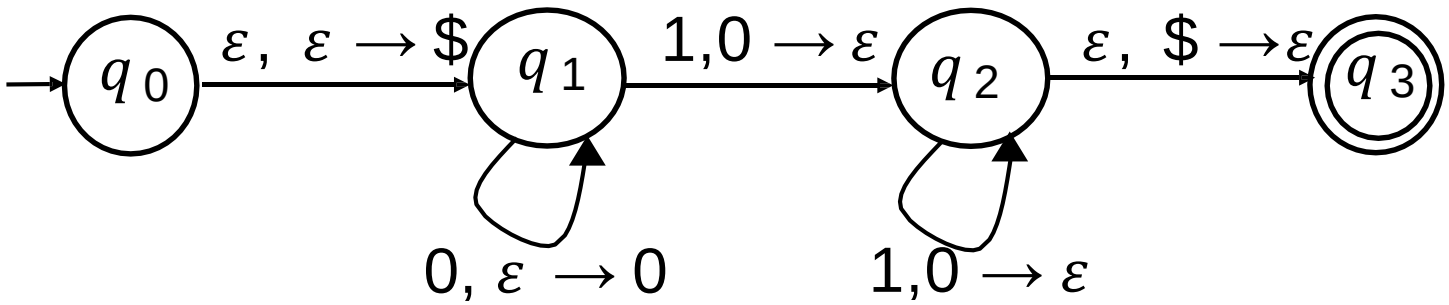
q_1 'de iken kelimeden 1 harfi gelirse yığından bir 0 okuyarak q_2 ' ye geceriz



q_1 'de kelimeden 0 harfi geldikçe q_1 'de kalınır, ve herbir 0 harfi icin bir 0 harfi yığına eklenir.

q_2 'de iken eğer kelimeden 1 harfi gelirse, bu 1'lere karsilik, yiğindan bir tane 0 okunur (yani cikartilir) ve yigina bir sey eklenmez.

Pushdown Otomata Örnek 1

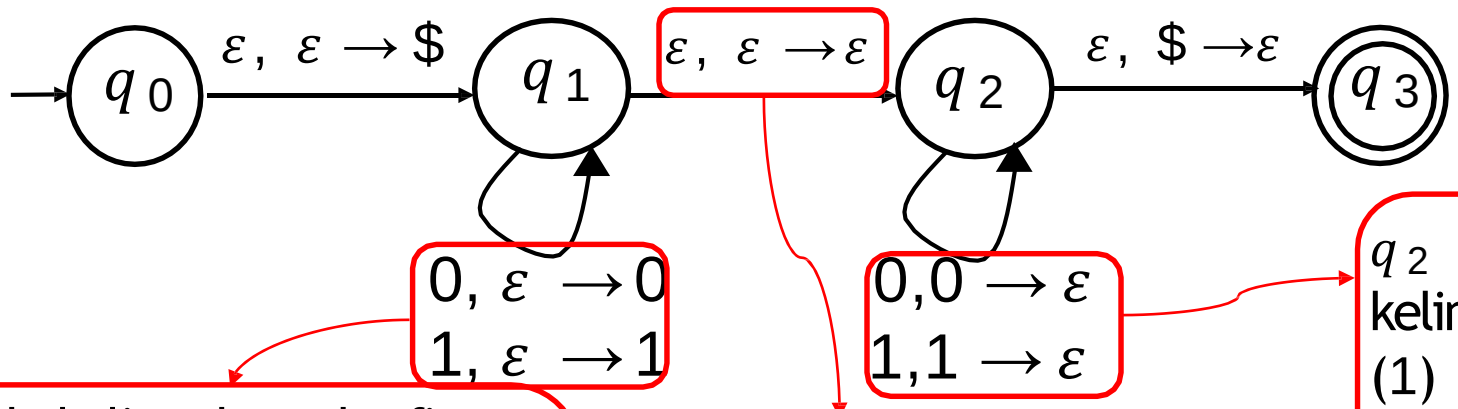


Yukaridaki PDA elimizde varken $w = 0^2 1^2$ kelimesini okuyalim.

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

$L = \{ w\, w^R \mid w \in \{0,1\}^* \}$ dilini tanıyan PDA tasarlıyalım. Burada w^R w kelimesinin tersten yazilmiş halidir. Örnegin $w = 01$ iken $w^R = 10$ olur. Su halde L dili çift uzunluktaki palindromlarin 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 kelimden 0 (1) gelirse ve yığının en üstunde 0 (1) varsa, yığından 0 (1) çikartilarak 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. $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \longrightarrow \mathcal{P}(Q \times \Gamma_\varepsilon)$, 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, Σ_ε, Γ_ε. Bu demektirki bir PDA'da, bir durumdan digerine gecebilmek icin, icinde bulunulan durum, kelimenin o anki harfi ve yiginin o andaki

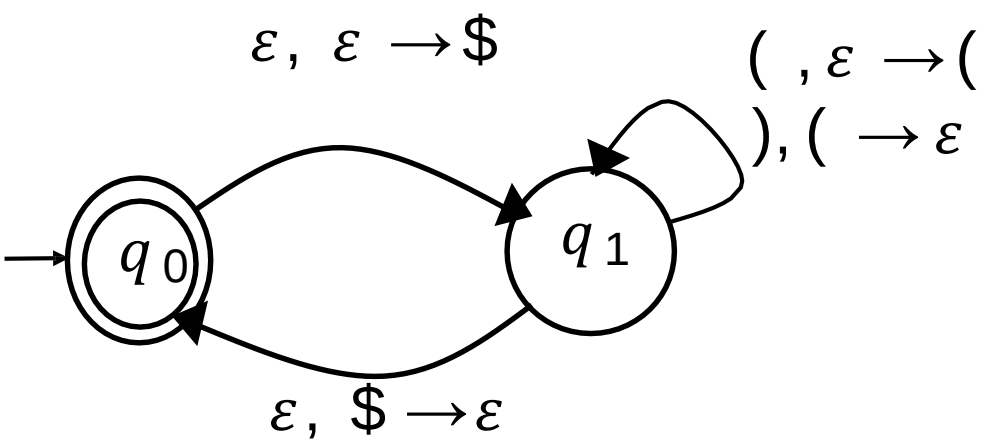
en ustteki harfi dikkate alinir. Gecis icin bunlarinin ucunun de c musait olmasi gerekir.

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 koyabilecigimizi ifade eder.

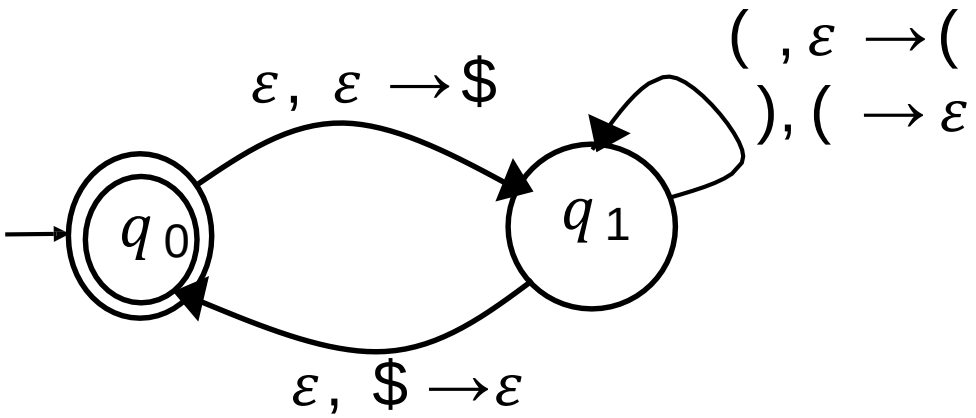
or. $\{(q_1,), (q_2,)\} \in \mathcal{P}(Q \times \Gamma)_\varepsilon$.

or. M = $(\{q_0, q_1\}, \{(,)\}, \{\$, (,)\}, \delta, q_0, \{q_0\}$ PDA'sı şu şekilde olsun:



Bu PDA, düzgün olarak ic ice geçmiş parentezlerin dilini taniyan PDA'dır.

Bu PDA icin $w = (((\ \ ))()$ kelimesini okuyalim:



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	–

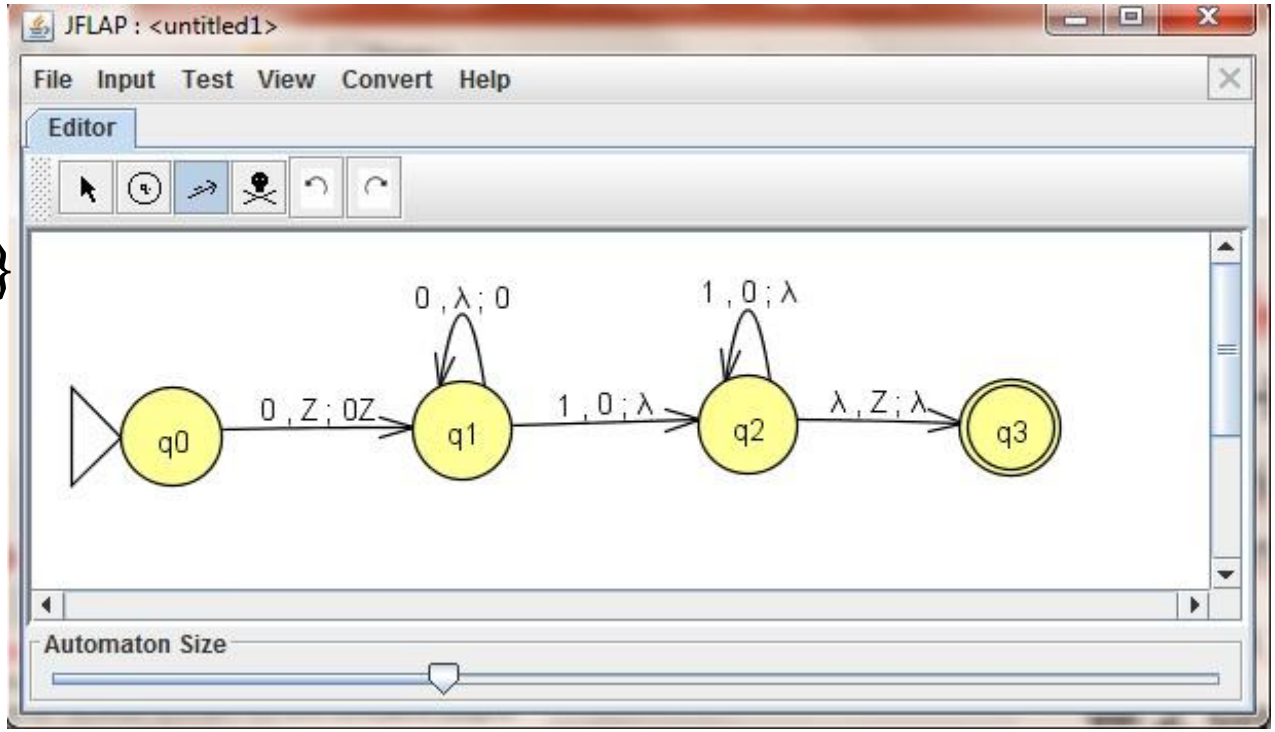
JFLAP ile Pushdown Otomata

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 PDA olusturudugumuzda otomatik olarak gelir.

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



PDA ile Icerikten Bagimsiz Dillerin Birbirine Denkligi

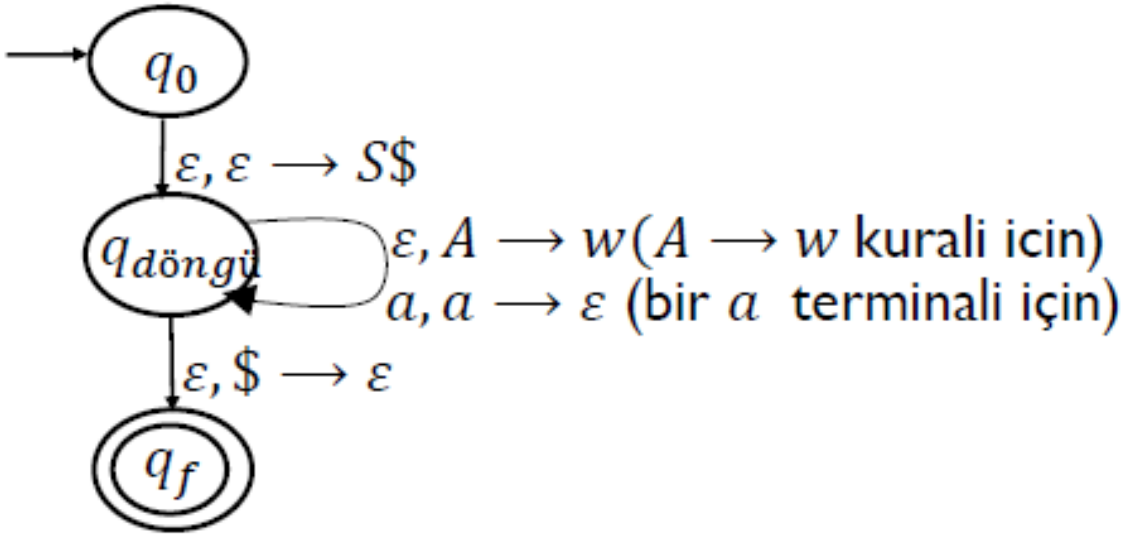
Teorem: Bir dilin içerikten bağimsiz olmasi icin gerek ve yeter şart o dilin bir pushdown otomata tarafindan tanınmasıdır.

Lemma 1: Bir dil içerikten bağimsiz ise onu taniyan bir pushdown otomata vardır.

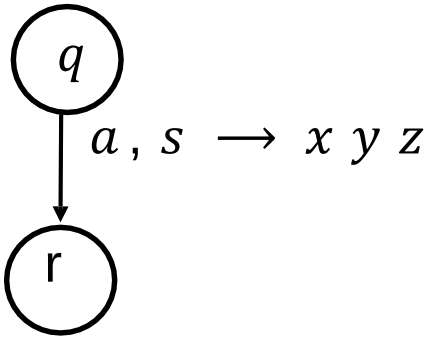
Bir içerikten bağisimsiz grammerden bir PDA elde ederek, verilen bir kelimenin bu grammerden üretıp üretilmedigini kolayca görebiliriz. Böylece bu iş için CYK algoritmasina alternatif bir yontemimiz olur.

İcerikten Bağimsız Grammerlerden Pushdown Otomataya

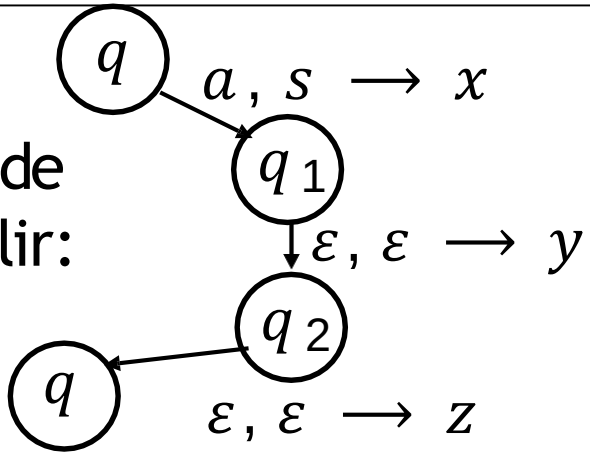
Bir içerikten bagimsiz grammerden türetilen PDA'nın üç durumu vardır: q_0, $q_{döngü}$ ve q_f. Bu PDA şu formdadir:



Not:



şu şekilde de gösterilebilir:



İcerikten Bağimsız Grammerlerden Pushdown Otomataya

Bir içerikten bagimsiz grammerden PDA türetmek icin asagidaki prosedür izlenir.

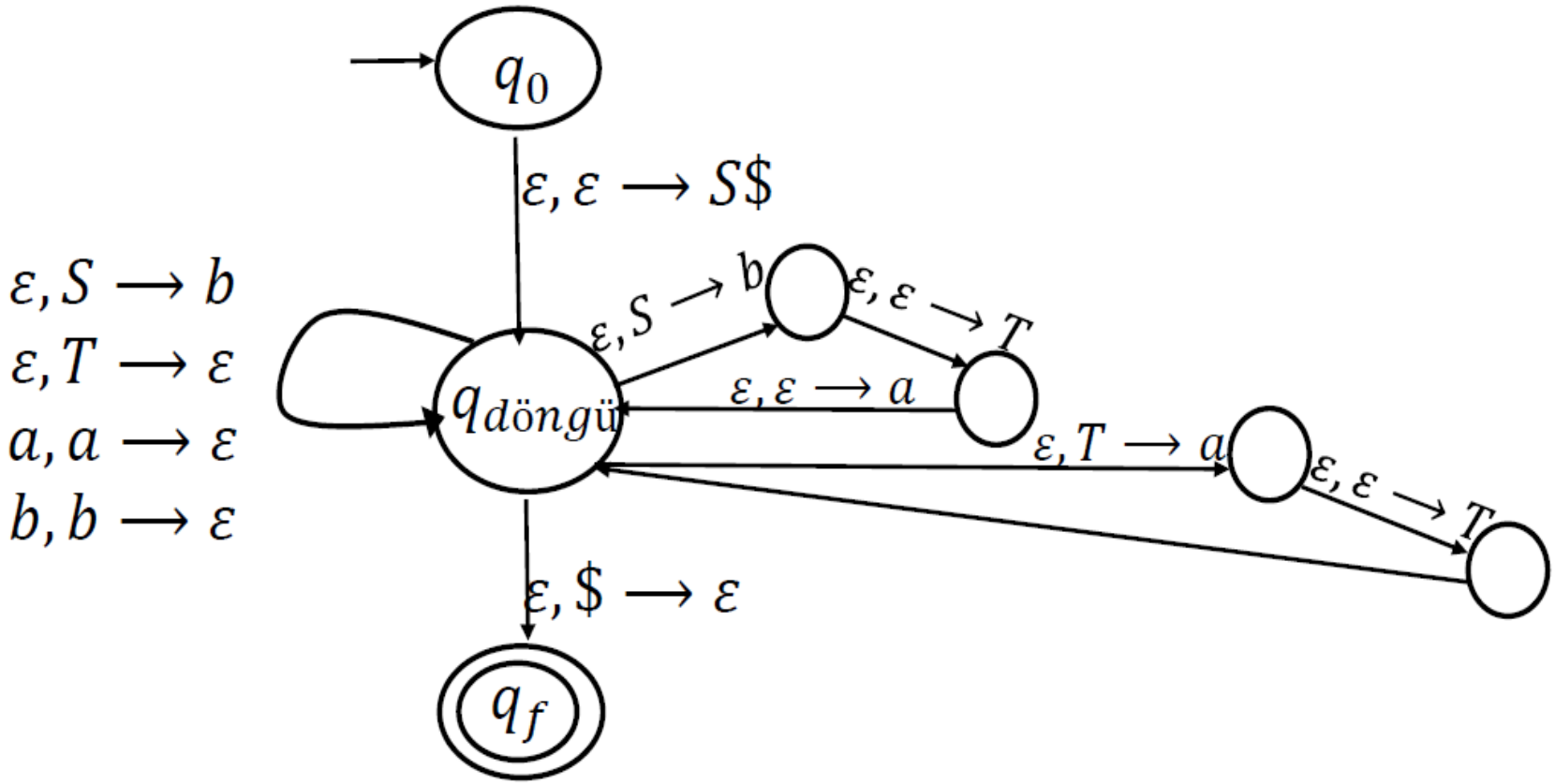
1. Bir onceki sekildeki gibi q_0, $q_{döngü}$ ve q_f durumlari çizilerek birbirine baglanir.
2. Yiğina başlangıç degiskeni S ve \$ eklenir. ($\varepsilon, \varepsilon \longrightarrow S\$$)
3. Degiskenler (nonterminaller) yigindan cikartilarak yerlerine bu degiskene karsilik gelen kelime eklenir. Ornegin bir $A \longrightarrow w$ kurali icin yigindan A cikartilip w kelimesi en sag sembolden baslayarak eklenir. Ortaya çikan geçişler $q_{döngü}$'ye eklenir.
4. Yigindan terminaller cikartilir. Ornegin bir a terminali için, $a, a \longrightarrow \varepsilon$ gecisi $q_{döngü}$'ye eklenir.
5. \$ yigindan cikartilirak (ε, \$ $\longrightarrow \varepsilon$) final durumu q_f' ye gidirlir.

İcerikten Bağimsız Grammerlerden Pushdown Otomataya

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

$$S \longrightarrow aTb|b$$
$$T \longrightarrow Ta|\varepsilon$$

İle verilen grammerder PDA elde edelim:



İcerikten Bağimsız Grammerlerden Pushdown Otomataya

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

$$S \longrightarrow (E) \,|\, E + E \,|\, E * E \,|\, a$$

İle verilen grammerden PDA elde edelim: