

# OTOMATA TEORİSİ

## Bölüm 5

### YIĞITLI ÖZDEVİNİRLER

## 5.1. 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 otomatlar, bağlamdan bağımsız dilleri (CFL) tanıyan makine modelidir.
- Bağlamdan-bağımsız diller ise, bağlamdan-bağımsız dilbilgileri (CFG) tarafından türetilen dillerdir. Yığıtlı özdevinirlerin ingilizce 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.

## 5.1.1. Yığıtlı Özdevinirin Temel Model

- Biçimsel olarak, yığıtlı özdevinirler(makineler) PDA bir yedili olarak tanımlanabilir.
- $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$
- $Q$  : Sonlu sayıda durumlar içeren durumlar kümesi
- $\Sigma$  : Sonlu sayıda girişten oluşan Giriş alfabesi
- $\Gamma$  : Sonlu sayıda simge içeren Yığıt Alfabesi (Yığıt ve giriş alfabelerindeki simgelerin tümü ya da bir bölümü ortak olabileceği gibi iki alfabede ortak simge bulanmayabilir.)
- $q_0$  : Başlangıç durumu ( $q_0 \in Q$ ) Başlangıç durumu durumlar kümesinin bir elemanı olduğuna göre  $Q$  boş olmayan bir kümedir.
- $Z_0$  : Yığıt alfabesindeki simgelerden, Yığıt Başlangıç Simgesi olarak adlandırılan özel bir simge ( $Z_0 \in \Gamma$ ).  $Z_0$  giriş alfabesinde yer almayan bir simgedir. Yığıt alfabesinin diğer tüm simgeleri ise giriş alfabesinde de yer alabilir.
- $F$  : Uç durumlar kümesi . Durumlar kümesinin bir alt kümesidir.  $F \subseteq Q$
- $\delta$  : Geçiş ya da hareket işlevi(transition or move function)

Deterministik PDA modelinde geçiş işlevi

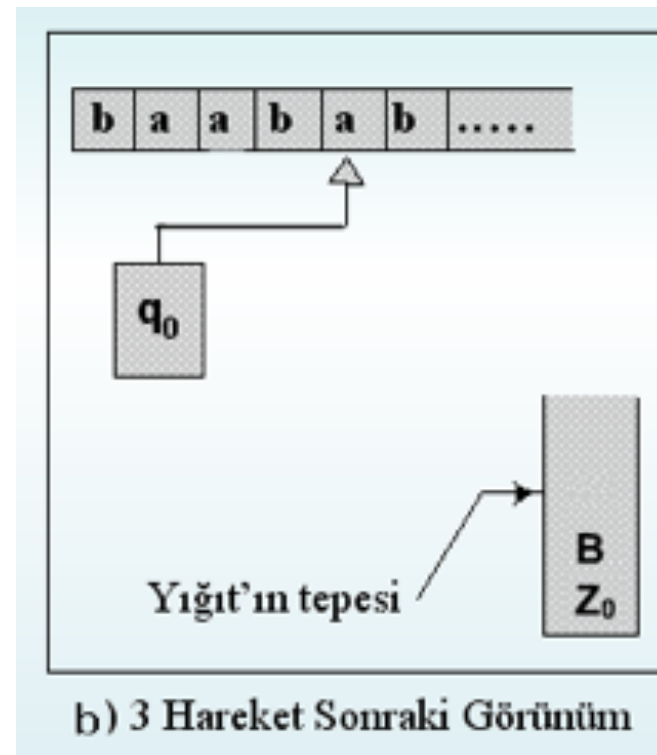
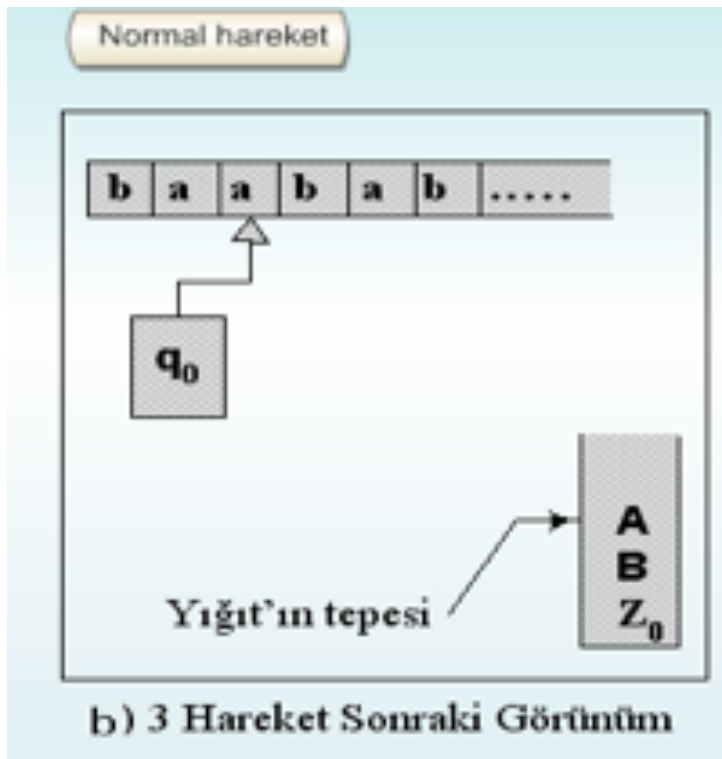
$[Q \times (\Sigma \cup \{\lambda\}) \times \Gamma]^*$  ' dan  $[Q \times \Gamma^*]^*$  ' a bir eşleşme oluşturur.

Deterministik olmayan modelde ise geçiş işlevi,

$[Q \times (\Sigma \cup \{\lambda\}) \times \Gamma]^*$  ' dan  $[Q \times \Gamma^*]^*$  ' ın sonlu alt kümelerine bir eşleşme oluşturur.

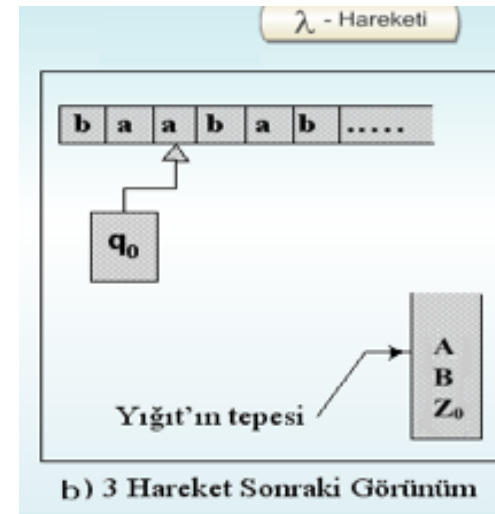
## 5.1.2. PDA'nın Soyut Makine Modeli

- Tanımlandığı biçimiyle yığıtlı özdevinir (PDA) bir matematiksel modeldir. PDA'yı soyut bir makine olarak düşünmek de mümkündür. PDA'nın nasıl çalıştığını daha iyi anlayabilmek için, bilinen bileşenlerden oluşan bir makine modeli de kullanılmaktadır. Bu model aşağıdaki bileşenlerden oluşmaktadır (Çizim 5.1):
- PDA Soyut Makine Modeli:
- Hücrelerden oluşan ve her hücrelerinde bir giriş simgesi bulunan bir mıknatıslı şerit. Yalnız okunabilen şeridin sağ ucu sonsuzdur.



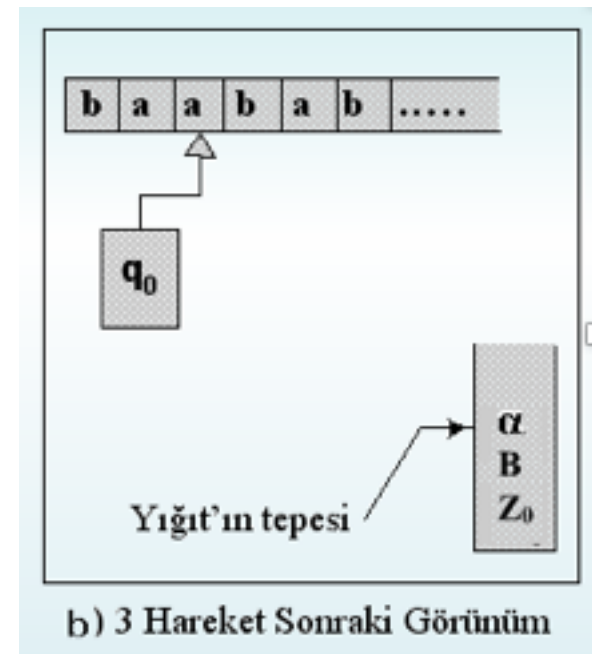
## 5.1.2. PDA'nın Soyut Makine Modeli

$\lambda$ - hareketinde ise aşağıdaki işlemler yapılır: Şeritten bir simge okunmaz ve okuma kafası yer değiştirmez. Dolayısıyla  $\lambda$ - hareketi şeritteki simgeden bağımsız bir harekettir.



$\lambda$ - 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). Yığıtta eklenen dizgi boş ( $\lambda$ ) da olabilir.



### 5.1.3. Anlık Tanımlar (Instantaneous Descriptions)

- PDA'nın belirli bir andan sonraki davranışlarını kestirebilmek için o ana ilişkin üç bilginin bilinmesi gerekir:
  - PDA'nın (sonlu denetim biriminin) bulunduğu durum
  - Giriş dizgisinin henüz işlenmemiş (okuma kafasının üzerinde bulunduğu hücre ile bu hücrenin sağındaki hücrelerden bulunan) kesimi
  - Yığıtın içeriği

Bu üç bilginin belirli bir andaki değerlerinden oluşan üçlüye, PDA'nın anlık tanımı denir:

- **Anlık tanım (ID) =  $(p, v, X)$** 
  - p** : PDA'nın durumu
  - v** : giriş dizgisinin henüz işlenmemiş kesimi
  - X** : yığıtın içeriği

### 5.1.4. PDA'nın Tanıdığı Dil

- Her PDA, giriş alfabesindeki simgelerden oluşan dizgilerin bir kesimini tanır (kabul eder), bir kesimini ise tanımaz. PDA'nın tanıdığı dizgiler kümesi, PDA'nın tanıdığı dildir.
- Dizgileri tanıma açısından PDA'nın iki alt modeli vardır. Bu modeller:
  - Uç durumu tanıyan PDA
  - Boş yığıtla tanıyan PDAOlarak adlandırılır.

### 5.1.4. PDA'nın Tanıdığı Dil

- Uç durumla tanıyan PDA modelinde, FA modelinde olduğu gibi, eğer bir dizginin tüm simgeleri okunduktan sonra PDA bir uç durumda bulunuyorsa, bu dizgi PDA tarafından tanınır. Uç durumla tanıyan PDA modelinde, dizginin tüm simgeleri okunduktan sonra, yığıtın içeriğine bakılmaz; dizginin PDA tarafından tanınmasının tek koşulu, son durumun bir uç durum olmasıdır. Anlık tanımları (instantaneous descriptions) kullanarak, uç durumla tanıyan bir PDA'nın tanıdığı dizgiler kümesi aşağıdaki gibi tanımlanır:

$$T(M) = \{ w \mid w \in V_T^* , \quad (q_0, w, Z_0) \vdash^* (p, \lambda, \gamma) , \quad p \in F \}$$

Boş yığıtla tanıyan PDA modelinde ise, dizginin tüm simgeleri okunduktan sonra, eğer yığıt boşalmışsa dizgi PDA tarafından tanınır. Bu modelde, PDA'nın son durumu tanımada etkili olmaz. Anlık tanımları kullanarak, boş yığıtla tanıyan bir PDA'nın tanıdığı dizgiler kümesi aşağıdaki gibi tanımlanır:

$$T(M) = \{ w \mid w \in V_T^* , \quad (q_0, w, Z_0) \vdash^* (p, \lambda, \lambda) , \quad p \in Q \}$$



## Örnek 5.1

$L_{5.1}$  dili aşağıdaki gibi tanımlanıyor:

$$L_{5.1} = \{ w c w^R \mid w \in (0 + 1)^* \}$$

Bu dili türeten bir dilbilgisi aşağıdaki gibi tanımlanabilir:

$$G_{5.1} = \langle V_N, V_T, P, S \rangle$$

$$V_N = \{ S \}$$

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

$$P : S \Rightarrow 0S0 \mid 1S1 \mid c$$

$L_{5.1}$ 'i tanıyan PDA'yı tanımlayalım:

$$M_{5.1} = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, \Phi \rangle$$

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

$$\Sigma = \{ 0, 1, c \}$$

$$\Gamma = \{ 0, 1, Z_0 \}$$

## Örnek 5.1

**PDA'nın çalışma ilkesi:** PDA'nın tanıdığı dizgiler,  $\{0, 1\}$  alfabesindeki herhangi bir  $w$  altdizgisi ile bunun tersinden oluşmaktadır.  $w$  ile  $w^R$  arasında da “c” simgesi yer almaktadır. Bu dizgileri tanıyan PDA:

Önce  $w$ 'nin simgelerini okur ve okuduğu her simgeyi yığta ekler. Ekleme sırasında PDA  $q_0$  durumunda bulunur.

$c$ 'yi okuduğunda,  $w$ 'nin bittiğini anlar, yığta bir değişiklik yapmaz ve durum değiştirerek  $q_1$  durumuna geçer.

Sonra  $w^R$ 'nin simgelerini okur, ve okuduğu her simge için eğer yığtın tepesinde aynı simge varsa bunu yığtadan çıkarır (siler).

Giriş dizgisinin tüm simgeleri okunduktan sonra, eğer yığtın tepesinde  $Z_0$  varsa, bunu da siler ve yığıtı tamamen boşaltır.

Yukarıda sözlü olarak tanımlanan işlemleri yapan PDA'nın hareket işlevi biçimsel olarak aşağıdaki gibi tanımlanır:

## Örnek 5.1

$$\delta : \delta(q_0, 0, Z_0) = (q_0, 0Z_0)$$

$$\delta(q_0, 1, Z_0) = (q_0, 1Z_0)$$

$$\delta(q_0, c, Z_0) = (q_1, Z_0)$$

$$\delta(q_0, 0, 0) = (q_0, 00)$$

$$\delta(q_0, 1, 0) = (q_0, 10)$$

$$\delta(q_0, 0, 1) = (q_0, 01)$$

$$\delta(q_0, 1, 1) = (q_0, 11)$$

$$\delta(q_0, c, 0) = (q_1, 0)$$

$$\delta(q_0, c, 1) = (q_1, 1)$$

$$\delta(q_1, 0, 0) = (q_1, \lambda)$$

$$\delta(q_1, 1, 1) = (q_1, \lambda)$$

$$\delta(q_1, \lambda, Z_0) = (q_1, \lambda)$$

Örnek olarak

verilen  $M_{5.1}$  deterministik bir PDA'dır.

PDA'nın hareket işlevi yandaki gibi tanımlanabileceği gibi, bir çizenek ile de tanımlanabilir (Çizim 5.2). Çizenekte, her hareket iki durum arasındaki bir yay ile gösterilmekte ve yayın üzerine bir üçlü yazılmaktadır.

Üçlünün birinci elemanı şeritten okunan giriş simgesini,

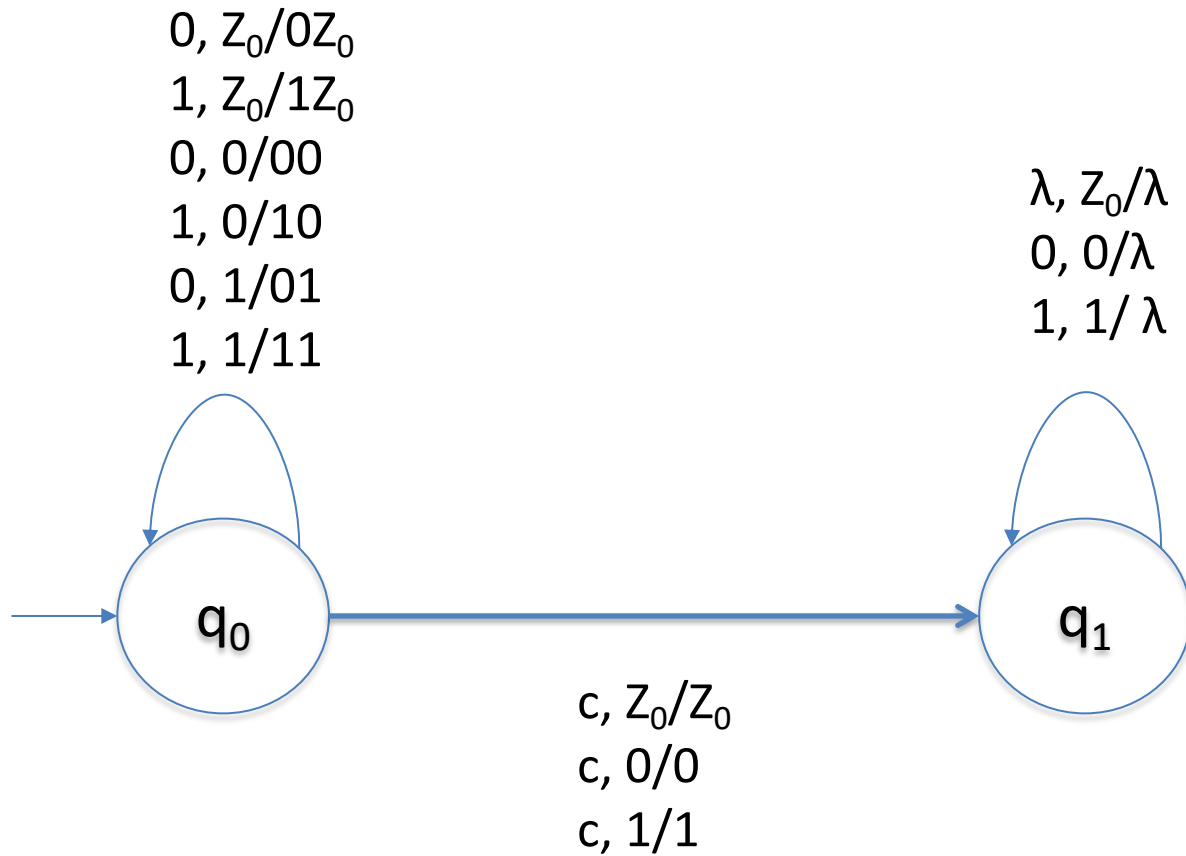
ikinci elemanı yığıtın tepesinde bulunan ve silinen yığıt simgesini,

üçüncü elemanı ise yığıtı eklenen yığıt simgeleri dizgisini göstermektedir.

Bu üçlü ingilizce sözcüklerle kısaca “read, pop, push” üçlüsü olarak gösterilebilir.

Temel PDA modeline göre (Bkz. 5.1) bu üçlünün birinci ve sonuncu elemanı  $\lambda$  olabilir.

## Örnek 5.1 Geçiş Çizeneği



## Örnek 5.2

$$L_{5.2} = \{ w w^R \mid w \in (0 + 1)^* \}$$

Bu dili türeten bir dilbilgisi aşağıdaki gibi tanımlanabilir:

$$G_{5.2} = \langle V_N, V_T, P, S \rangle$$

$$V_N = \{ S \}$$

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

$$P : S \Rightarrow 0S0 \mid 1S1 \mid \lambda$$

$L_{5.2}$ 'yi tanıyan PDA'yı tanımlayalım:

$$M_{5.2} = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, \Phi \rangle$$

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

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

$$\Gamma = \{ A, B, Z_0 \}$$

## Örnek 5.2

**DA'nın çalışma ilkesi:** PDA'nın tanıdığı dizgiler,  $\{0, 1\}$  alfabesindeki herhangi bir  $w$  altdizgisi ile bunun tersinden oluşmaktadır.  $w$  ile  $w^R$  arasında herhangi bir simge yer almamaktadır. Bu dizgileri tanıyan PDA:

Önce  $w$ 'nin simgelerini okur ve yığta okuduğu her  $0$  için bir  $A$ , her  $1$  için de bir  $B$  ekler. PDA'nın yığıt alfabesi, giriş alfabesindeki simgeleri içerebileceği gibi, bu simgeleri içermeyebilir de. Örnek 5.1'de, giriş alfabesindeki tüm simgeler yığıt alfabesinde yer almaktadır. Örnek 5.2'de ise, yığıt alfabesinde, giriş alfabesindeki simgelere yer verilmemiştir.

$w$ 'nin simgeleri okunurken PDA  $q_0$  durumundadır.  $w$  bitip,  $w^R$  başladığında PDA  $q_1$  durumuna geçecektir. Ancak giriş dizgisinde,  $w$ 'nin bitip  $w^R$ 'nin başladığını belirten özel bir simge yoktur. Okunan simge eğer bir önceki simgenin aynısı ise, bu simge  $w$ 'nin bir sonraki simgesi olabileceği gibi,  $w^R$ 'nin ilk simgesi de olabilir. Bu koşullarda iki hareket tanımlamak gerekir: birinci harekette PDA ekleme durumunda ( $q_0$ ) kalır ve yığta ekleme yapar; ikinci harekette ise PDA silme durumuna ( $q_1$ ) geçer ve yığtın tepesindeki simgeyi siler.

Silme durumuna ( $q_1$ ) geçen PDA  $w^R$ 'nin simgelerini okur, ve okuduğu her simge için eğer yığtın tepesinde bu simgenin karşılığı ( $0$  için  $A$ ,  $1$  için  $B$ ) varsa bunu yığıttan çıkarır (siler).

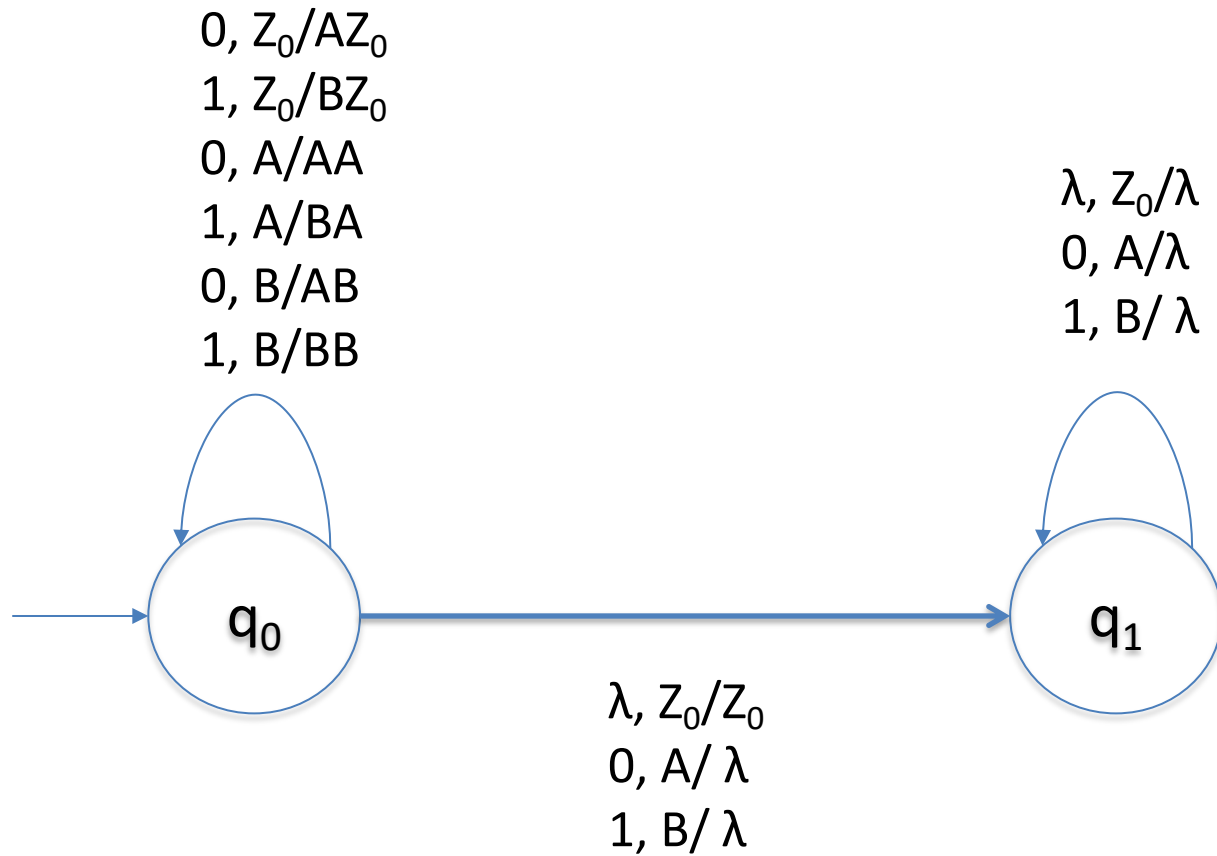
Giriş dizgisinin tüm simgeleri okunduktan sonra, eğer yığtın tepesinde  $Z_0$  varsa, PDA bunu da siler ve yığıtı tamamen boşaltır.

## Örnek 5.2

Yukarıda sözlü olarak tanımlanan işlemleri yapan PDA'nın hareket işlevi biçimsel olarak aşağıdaki gibi tanımlanır:

$$\begin{aligned}\delta : \quad & \delta(q_0, 0, Z_0) = (q_0, AZ_0) \\ & \delta(q_0, 1, Z_0) = (q_0, BZ_0) \\ & \delta(q_0, \lambda, Z_0) = (q_1, \lambda) \\ & \delta(q_0, 0, A) = \{(q_0, AA), (q_1, \lambda)\} \\ & \delta(q_0, 1, A) = (q_0, BA) \\ & \delta(q_0, 0, B) = (q_0, AB) \\ & \delta(q_0, 1, B) = \{(q_0, BB), (q_1, \lambda)\} \\ & \delta(q_1, 0, A) = (q_1, \lambda) \\ & \delta(q_1, 1, B) = (q_1, \lambda) \\ & \delta(q_1, \lambda, Z_0) = (q_1, \lambda)\end{aligned}$$

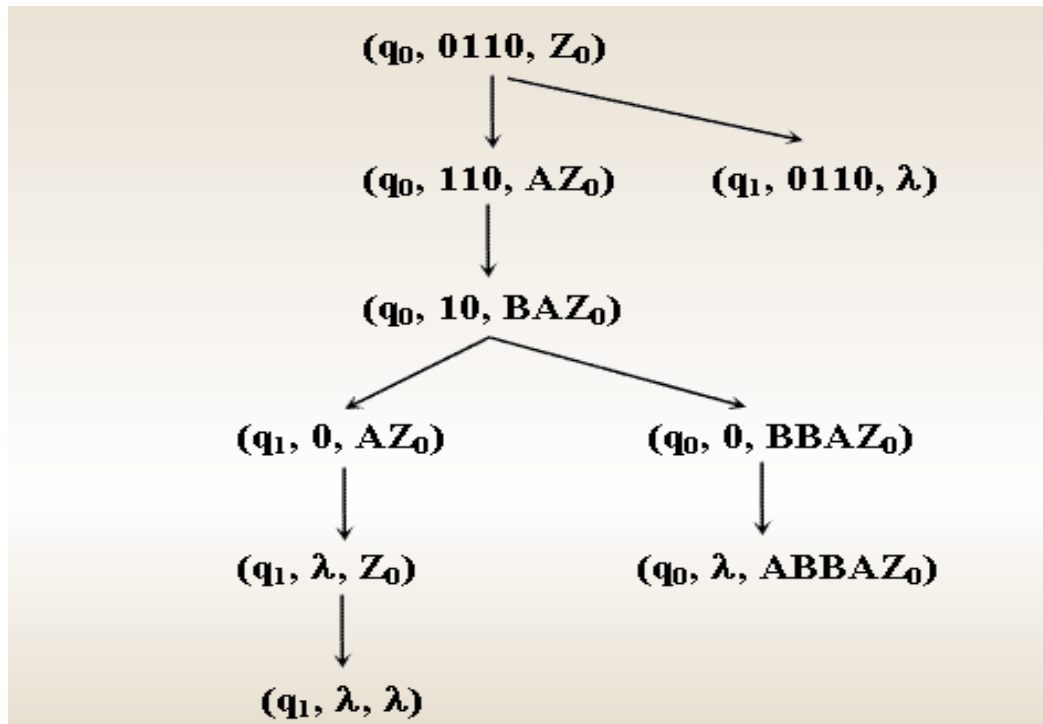
## Örnek 5.2 Geçiş Çizeneği





## Örnek 5.2

- $M_{5.2}$  deterministik olmayan (non deterministic) bir PDA'dır.  $w = 0110$  dizgisinin bu PDA tarafından tanındığını, anlık tanımları izleyerek aşağıdaki gibi bulabiliriz:



$(q_0, 0110, Z_0)$  başlangıç ID'sinden  $(q_1, \lambda, \lambda)$  ID'sine ulaşılabilirdiği için bu makine **0110** dizgisini tanır.

## Örnek 5.3

- $L_{5.3}$  dili,  $\{0, 1\}$  alfabesindeki palindram'ları içeren dil olarak tanımlanıyor. Palindram, her iki yönden okunuşu aynı olan dizgiler kümesidir. Türkçedeki palindram'lara örnek olarak ebe, aba, kok, talat, kepek, teğet, kelek,.. gibi sözcükler verilebilir.  $\{0, 1\}$  alfabesindeki palindramlara örnek olarak ise  $\lambda$ ,  $0, 1, 00, 11, 000, 010, 100001, 101101,..$  gibi dizgiler sayılabilir.  $\{0, 1\}$  alfabesindeki palindramlar, biçimsel olarak aşağıdaki gibi tanımlanabilir:

$$L_{5.3} = \{w w^R + w0w^R + w1w^R \mid w \in (0+1)^*\}$$

Bu dili türeten bir dilbilgisi aşağıdaki gibi tanımlanabilir:

$$G_{5.3} = \langle V_N, V_T, P, S \rangle$$

$$V_N = \{S\}$$

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

$$P: S \Rightarrow 0S0 \mid 1S1 \mid \lambda \mid 0 \mid 1$$

## Örnek 5.3

$L_{5.3}$ 'ü tanıyan PDA'yı tanımlayalım:

$$M_{5.3} = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, \Phi \rangle$$

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

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

$$\Gamma = \{ 0, 1, Z_0 \}$$

**PDA'nın çalışma ilkesi:** PDA'nın tanıdığı dizgiler,  $\{0, 1\}$  alfabesindeki herhangi bir  $w$  altdizgisi ile bunun tersinden oluşmaktadır.  $w$  ile  $w^R$  arasında herhangi bir simge yer almayabileceği gibi,  $0$  ya da  $1$  simgesi de yer alabilir. Bu dizgileri tanıyan PDA:

Önce  $w$ 'nin simgelerini okur ve yığta ekler.  $w$ 'nin simgeleri okunurken PDA  $q_0$  durumundadır.

$w$  bitip,  $w^R$  başladığında PDA  $q_1$  durumuna geçecektir. Ancak giriş dizgisinde,  $w$ 'nin bitip  $w^R$ 'nin başladığını belirten özel bir simge yoktur. Okunan simge  $w$ 'nin bir sonraki simgesi olabileceği gibi,  $w$  ve  $w^R$  arasındaki orta simge ya da  $w^R$ 'nin ilk simgesi de olabilir. Hareketlerin buna göre tanımlanması gerekir. PDA deterministik olmayacaktır.

Silme durumuna ( $q_1$ ) geçen PDA  $w^R$ 'nin simgelerini okur, ve okuduğu her simge için eğer yığtın tepesinde aynı simge varsa bunu yığtattan çıkarır (siler).

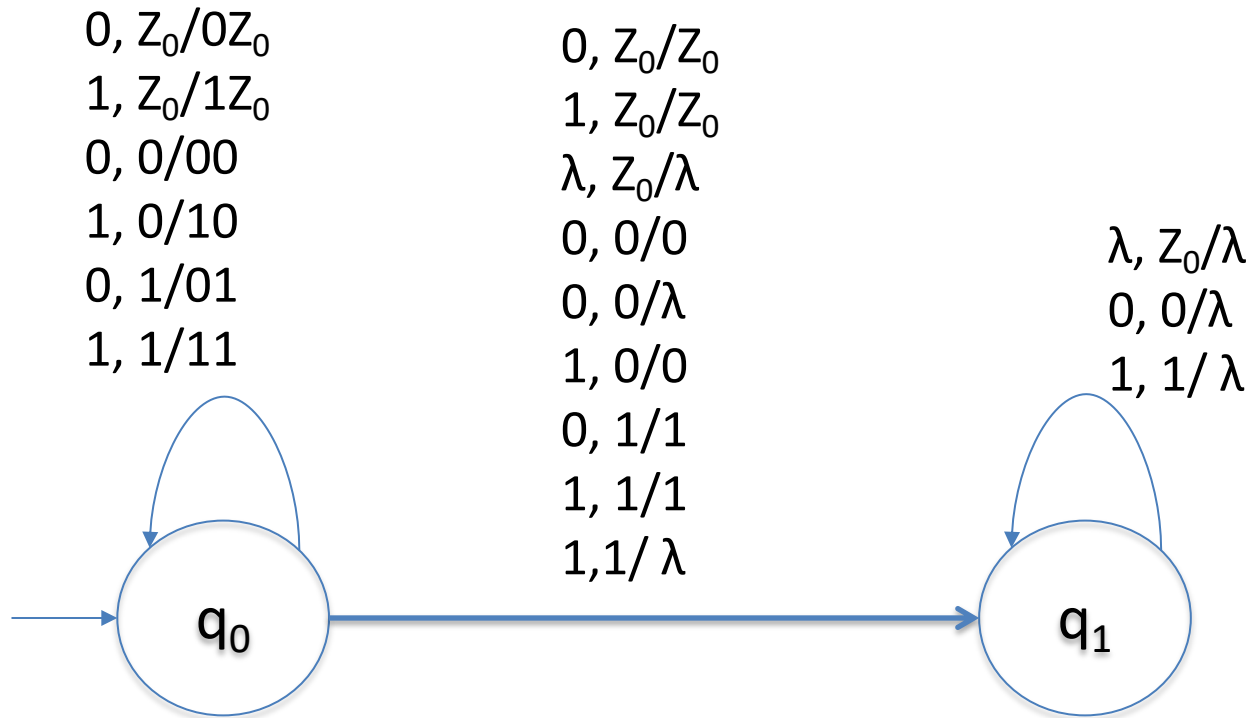
## Örnek 5.3

- Giriş dizgisinin tüm simgeleri okunduktan sonra, eğer yığıtın tepesinde  $Z_0$  varsa, PDA bunu da siler ve yığıtı tamamen boşaltır.

Yukarıda sözlü olarak tanımlanan işlemleri yapan PDA'nın hareket işlevi biçimsel olarak aşağıdaki gibi tanımlanır:

$$\begin{aligned}\delta : \delta(q_0, 0, Z_0) &= \{(q_0, 0Z_0), (q_1, Z_0)\} \\ \delta(q_0, 1, Z_0) &= \{(q_0, 1Z_0), (q_1, Z_0)\} \\ \delta(q_0, \lambda, Z_0) &= (q_1, \lambda) \\ \delta(q_0, 0, 0) &= \{(q_0, 00), (q_1, 0), (q_1, \lambda)\} \\ \delta(q_0, 1, 0) &= \{(q_0, 10), (q_1, 0)\} \\ \delta(q_0, 0, 1) &= \{(q_0, 01), (q_1, 1)\} \\ \delta(q_0, 1, 1) &= \{(q_0, 11), (q_1, 1), (q_1, \lambda)\} \\ \delta(q_1, 0, 0) &= (q_1, \lambda) \\ \delta(q_1, 1, 1) &= (q_1, \lambda) \\ \delta(q_1, \lambda, Z_0) &= (q_1, \lambda)\end{aligned}$$

## Örnek 5.3 Geçiş Çizeneği



## Örnek 5.4

- $L_{5.4}$  dili aşağıdaki gibi tanımlanıyor:

$$L_{5.4} = \{ a^i b^n a^j b^n a^k \mid i, j, k, n \geq 1 \}$$

Bu dili türeten bir dilbilgisi aşağıdaki gibi tanımlanabilir:

$$G_{5.4} = \langle V_N, V_T, P, S \rangle$$

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

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

$$P : S \Rightarrow ABA$$

$$A \Rightarrow aA \mid a$$

$$B \Rightarrow bBb \mid bAb$$

## Örnek 5.4

- $L_{5.4}$ 'ü tanıyan PDA'yı tanımlayalım:

$$M_{5.4} = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, \Phi \rangle$$

$$Q = \{ q_0, q_1, q_2, q_3, q_4, q_5 \}$$

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

$$\Gamma = \{ B, Z_0 \}$$

**PDA'nın çalışma ilkesi:** PDA'nın tanıdığı dizgiler, 5 öbekten oluşmaktadır. Her öbekte en az bir simge yer almaktadır. 1., 3., ve 5. öbeklerdeki simgeler **a**; 2. ve 4. öbeklerdeki simgeler ise **b**'dir. 2. ve 4. öbeklerdeki b'lerin sayıları da eşittir. Buna göre PDA'nın 1., 3. ve 5. öbeklerdeki a'ların sayılarının en az bir olduğunu denetlemesi yeterlidir. Yığıt 2. ve 4. öbeklerdeki b'lerin sayılarının eşit olduğunu denetlemek için kullanılır. Bunun için de 2. öbekteki b'ler okunurken yığıta eklenir; 4. öbekteki b'ler okunurken de yığıt boşaltılır.

## Örnek 5.4

- Yukarıda sözlü olarak tanımlanan işlemleri yapan PDA'nın hareket işlevi biçimsel olarak aşağıdaki gibi tanımlanır:

$$\delta : \quad \delta(q_0, a, Z_0) = (q_1, Z_0)$$

$$\delta(q_1, a, Z_0) = (q_1, Z_0)$$

$$\delta(q_1, b, Z_0) = (q_2, BZ_0)$$

$$\delta(q_2, b, B) = (q_2, BB)$$

$$\delta(q_2, a, B) = (q_3, B)$$

$$\delta(q_3, a, B) = (q_3, B)$$

$$\delta(q_3, b, B) = (q_4, \lambda)$$

$$\delta(q_4, b, B) = (q_4, \lambda)$$

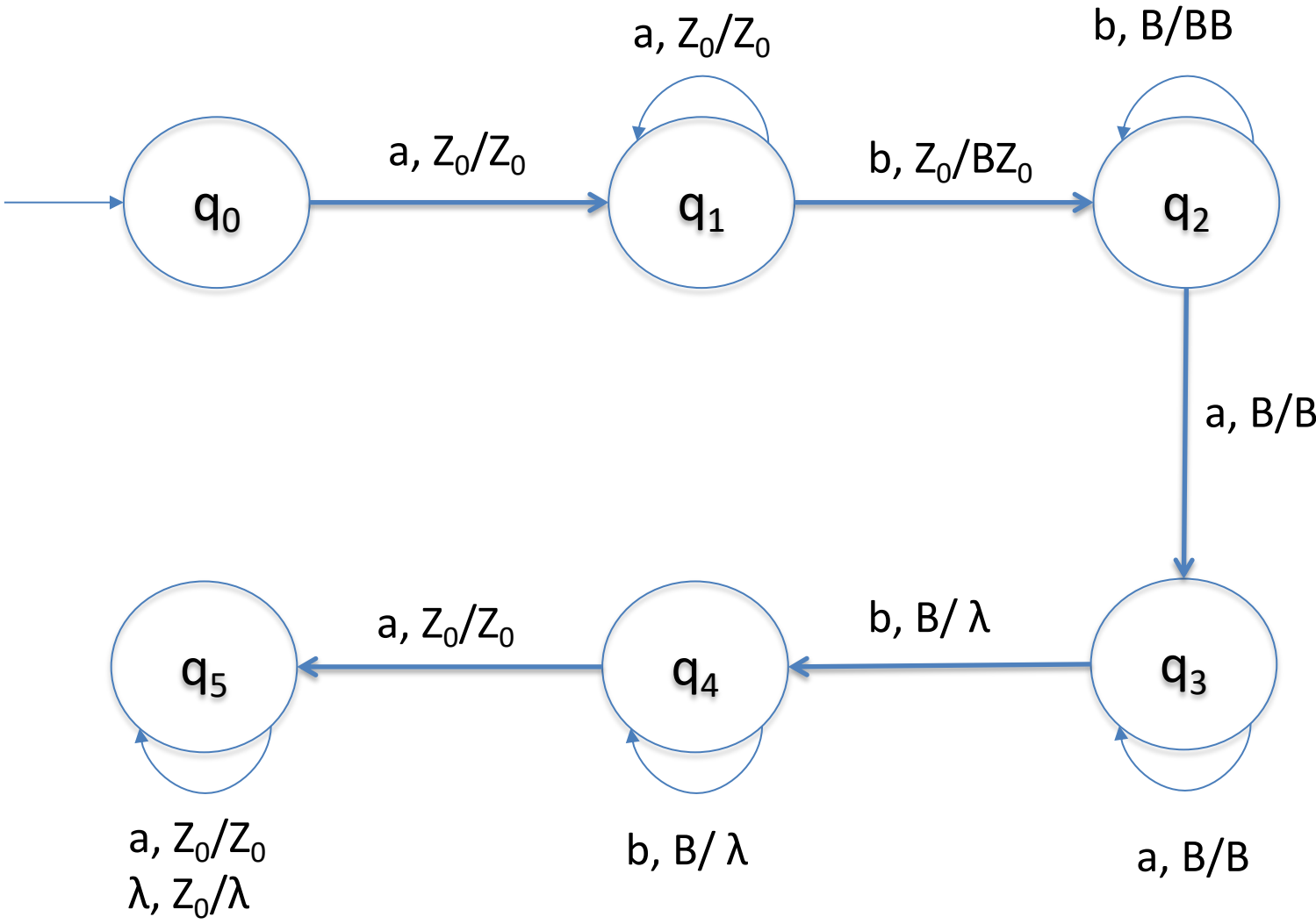
$$\delta(q_4, a, Z_0) = (q_5, Z_0)$$

$$\delta(q_5, a, Z_0) = (q_5, Z_0)$$

$$\delta(q_5, \lambda, Z_0) = (q_5, \lambda)$$



# Örnek 5.4 Geçiş Çizeneği



### 5.1.5. PDA'nın Deterministik Olma Koşulu

- Örneklerin de gösterdiği gibi, bağlamdan-bağımsız her dil (CFL) için, bu dili tanıyan deterministik bir PDA bulmak mümkün değildir. Başka bir deyişle, deterministik PDA'lar tarafından tanınan diller, bağlamdan-bağımsız dillerin bir altkümesidir. Her CFL için, dili tanıyan bir PDA bulunabilir. Ancak CFL'lerin yalnız bir altkümesi için, dili tanıyan deterministik bir PDA bulmak mümkündür. Diğer CFL'ler için ise, dili tanıyan deterministik olmayan PDA'lar bulunabilir; ancak dili tanıyan deterministik PDA bulmak mümkün değildir.
- Bir PDA'nın deterministik olabilmesi için, her anlık tanım için, yapılabilecek hareket sayısının en çok bir olması; başka bir deyişle aynı koşullarda hiçbir zaman birden çok hareket yapılamaması gerekir. Bunun için de:
  - Her  $\delta(q, a, X)$  için tanımlı en çok bir hareket olması
  - Eğer  $\delta(q, a, X)$  için bir hareket tanımlı ise de, hiçbir  $a$  giriş simgesi için  $\delta(q, a, X)$  hareketinin tanımlı olmaması

### 5.1.5. PDA'nın Deterministik Olma Koşulu

- gerekir. Özellikle aynı koşullarda sağlanan bir normal hareket ile bir  $-\lambda$  hareketi tanımlı ise PDA'nın deterministik olmayacağı görülmektedir. Yukarıda verilen örneklerden,  $M_{5.1}$ 'in deterministik olduğu;  $M_{5.2}$  ile  $M_{5.3}$ 'ün de deterministik olmadığı açıktır.  $M_{5.4}$ 'e gelince, bu PDA'nın  $q_0, q_1, q_2, q_3$  ve  $q_4$  durumları için tanımlı hareketler deterministiktir. Ancak  $q_5$  durumu için tanımlı iki hareket nedeniyle  $M_{5.4}$  deterministik değildir.

## 5.2. BAĞLAM DAN-BAĞIMSIZ DİLBİLGİSİ (CFG) – YIĞITLI ÖZDEVİNİR (PDA) EŞDEĞERLİĞİ

- Bağlamdan-bağımsız bir dilbilgisi tarafından türetilen dil bağlamdan-bağımsız bir dildir. Bağlamdan-bağımsız bir dilbilgisi verildiğinde, bu dilbilgisi tarafından türetilen dili tanıyan bir PDA bulunabilir. Diğer taraftan, bir PDA verildiğinde, bu PDA tarafından tanınan bağlamdan-bağımsız dili türeten bağlamdan-bağımsız bir dilbilgisi bulunabilir. Dolayısıyla, PDA'larla CFG'ler arasında bir eşdeğerlik ilişkisi vardır. Bu eşdeğerlik dillerin eşitliği üzerine kuruludur. Eğer bir PDA'nın tanıdığı dil ile bir CFG'nin türettiği dil aynı dil ise, bu PDA ile bu CFG birbirinin eşdeğeridir.

### Örnek 1

$$L_{5.1} = \{ a^n b^m c^k \mid n \geq 1, m \geq 2, k \geq 1, m = n + k \}$$

- a)  $L_{5.1}$  dilini türeten bağlamdan-bağımsız bir dilbilgisi tanımlayınız.
- b)  $L_{5.1}$  dilini boş yığıtla tanıyan bir PDA tanımlamanız isteniyor. Bunun için önce PDA'nın nasıl çalışacağını belirleyip sözel olarak açıklayınız. Sonra da PDA'nın biçimsel tanımını veriniz. Bu bağlamda PDA'nın hareketlerinin önce işlev tanımlarını yazınız, sonra da hareketleri bir geçiş çizeneği ile gösteriniz.

# Cevap 1

b) PDA'nın çalışma ilkesi:

a'lar yığığa atılır.

a'lar bitinceğe kadar her b için yığıttan bir a silinir.

a'lar bittikten sonra kalan b'ler yığığa atılır.

her c için yığıttan bir b silinir.

$G_{5.1} = \langle V_N, V_T, P, S \rangle$

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

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

P:  $S \Rightarrow AB$

$A \Rightarrow aAb \mid ab$

$B \Rightarrow bBc \mid bc$

PDA'nın biçimsel tanımı:

$M_{5.1} = \langle Q, \Sigma, \Gamma, q_0, \delta, Z_0, \Phi \rangle$

$Q = \{ q_0, q_1, q_2, q_3, q_4 \}$

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

$\Gamma = \{ Z_0, a, b \}$

$F = \Phi$

$\delta$ :  $\delta(q_0, a, Z_0) = (q_1, aZ_0)$

$\delta(q_1, a, a) = (q_1, aa)$

$\delta(q_1, b, a) = (q_2, \lambda)$

$\delta(q_2, b, a) = (q_2, \lambda)$

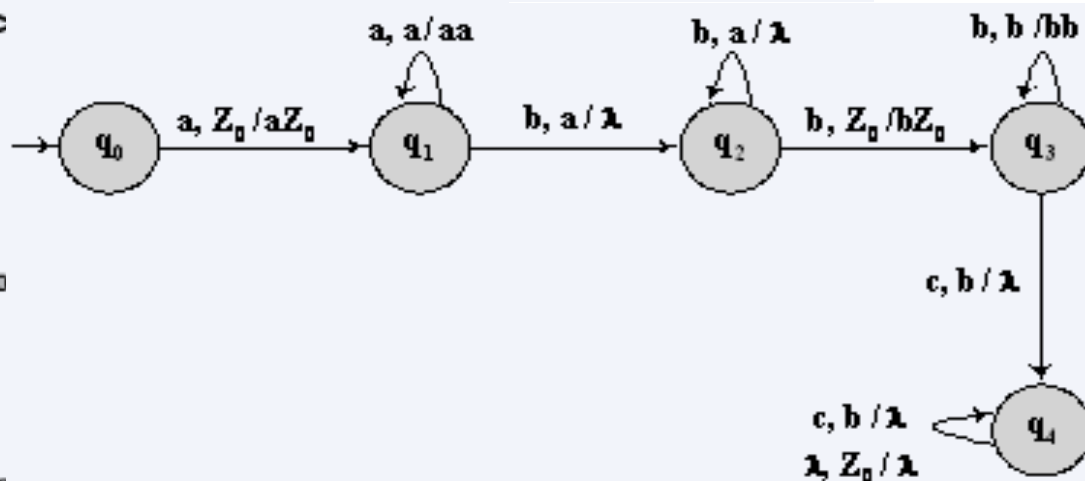
$\delta(q_2, b, Z_0) = (q_3, bZ_0)$

$\delta(q_3, b, b) = (q_3, bb)$

$\delta(q_3, c, b) = (q_4, \lambda)$

$\delta(q_4, c, b) = (q_4, \lambda)$

$\delta(q_4, \lambda, Z_0) = (q_4, \lambda)$



## Örnek 2

$$L_{5.2} = \{ a^n b^{n-k} c^k \mid n \geq 1, k \geq 1, n \geq k \}$$

dilini boş yığıtla tanıyan bir PDA tasarlamamız isteniyor. Bunun için:

- PDA'nın nasıl çalışacağını belirleyip sözel olarak açıklayınız.
- PDA'nın biçimsel tanımını veriniz. Bu bağlamda PDA'nın hareketlerinin önce işlev tanımlarını yazınız sonra da hareketleri bir geçiş çizeneği ile gösteriniz.

## Cevap 2

b)  $M_{S.5.1} = \langle Q, \Sigma, \Gamma, q_0, \delta, Z_0, \Phi \rangle$

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

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

$\Gamma = \{ Z_0, a, b, c \}$

$F = \Phi$

$\delta : \delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, \lambda)$

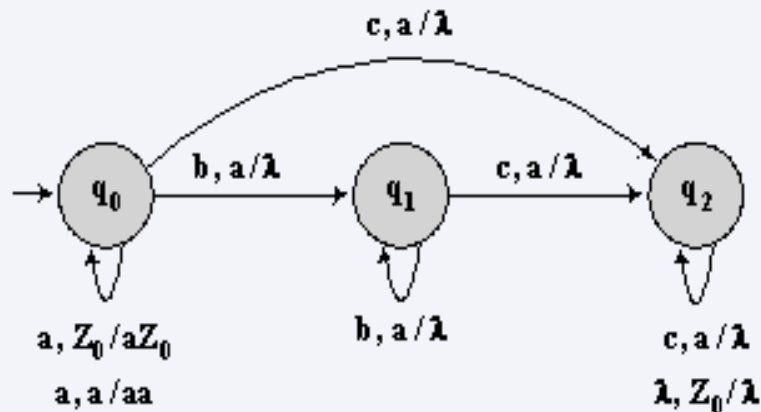
$\delta(q_0, c, a) = (q_2, \lambda)$

$\delta(q_1, b, a) = (q_1, \lambda)$

$\delta(q_1, c, a) = (q_2, \lambda)$

$\delta(q_2, c, a) = (q_2, \lambda)$

$\delta(q_2, \lambda, Z_0) = (q_2, \lambda)$



a)

PDA'nın çalışma ilkesi:

**a**'lar yığığa atılır.

her **b** için yığıttan bir **a** silinir.

her **c** için yığıttan bir **a** silinir.



### Örnek 3

$\{a, b, c\}$  alfabesindeki bağlamdan-bağımsız (CF)  $L_{5.3}$  dili "içindeki **c**'lerin sayısı **b**'lerin sayısının iki katı olan dizgiler kümesi" olarak tanımlanıyor.  $L_{5.3}$  dili  $\lambda$ 'yı içermiyor. Dilin tümcelerinden bazı örnekler aşağıda görülmektedir.

$L_{5.3} = \{a, aaa, bcac, cbc, baacccacb, ccaaacbcabbcaaaac, ....\}$

- $L_{5.3}$  dilini boş yığıtla tanıyan bir PDA oluşturmanız isteniyor. Bunun için önce PDA'nın nasıl çalışacağını kısaca açıklayınız. Daha sonra PDA'nın biçimsel tanımını veriniz ve durumların anlamlarını belirtiniz.
- PDA'nın hareketlerini bir çizeneğe gösteriniz. Oluşturduğunuz PDA deterministik midir? Niçin?
- $L_{5.3}$  dilini türeten bağlamdan-bağımsız bir dilbilgisi (CFG) tanımlayınız ve  $w = abcaabccac$  tümcesi için türetme ağacını oluşturunuz.

**a) PDA'nın çalışma ilkesi:**

a'lar okunduğunda yığıtta hiçbir işlem yapılmaz.

her b okunduğunda, yığıtta:

birden çok c varsa, bunlardan ikisi silinir,

tek bir c varsa, bu c silinir ve yığıtta bir b eklenir,

hiç c yoksa, yığıtta iki b eklenir.

her c okunduğunda, yığıtta:

en az bir b varsa, bunlardan biri silinir,

hiç b yoksa, yığıtta bir c eklenir.

**PDA'nın biçimsel tanımı:**

**$M_{5.3} = \langle Q, \Sigma, \Gamma, q_0, \delta, Z_0, \Phi \rangle$**

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

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

**$\Gamma = \{ Z_0, b, c \}$**

**$F = \Phi$**

**$\delta : \delta (q_0, a, Z_0) = (q_1, Z_0)$**

**$\delta (q_0, b, Z_0) = (q_1, bbZ_0)$**

**$\delta (q_0, c, Z_0) = (q_1, cZ_0)$**

**$\delta (q_1, a, Z_0) = (q_1, Z_0)$**

**$\delta (q_1, a, b) = (q_1, b)$**

**$\delta (q_1, a, c) = (q_1, c)$**

**$\delta (q_1, b, Z_0) = (q_1, bbZ_0)$**

**$\delta (q_1, b, b) = (q_1, bbb)$**

**$\delta (q_1, b, c) = (q_2, \lambda )$**

**$\delta (q_1, c, Z_0) = (q_1, cZ_0)$**

**$\delta (q_1, c, b) = (q_1, \lambda )$**

**$\delta (q_1, c, c) = (q_1, cc)$**

**$\delta (q_2, \lambda, Z_0) = (q_1, b)$**

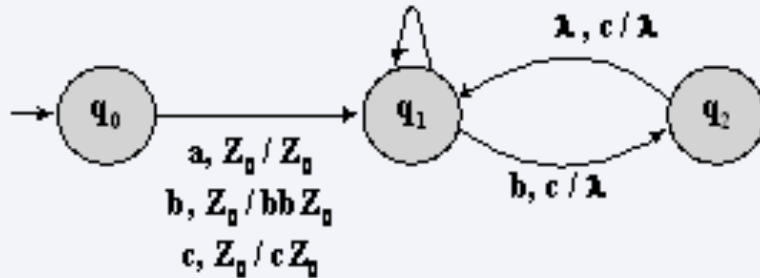
**$\delta (q_2, \lambda, c) = (q_1, \lambda )$**

**$\delta (q_1, \lambda, Z_0) = (q_1, \lambda )$**

## Cevap 3

b)

$a, Z_0 / Z_0$   
 $a, b / b$   
 $a, c / c$   
 $b, Z_0 / bbZ_0$   
 $b, b / bbb$   
 $c, Z_0 / cZ_0$   
 $c, b / \lambda$   
 $c, c / cc$   
 $\lambda, Z_0 / \lambda$   
 $\lambda, Z_0 / bZ_0$   
 $\lambda, c / \lambda$



$M_{5.3}$  deterministik değildir.

c)  $G_{5.3} = \langle V_N, V_T, P, S \rangle$

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

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

$P: S \Rightarrow aA \mid bCC \mid cD$

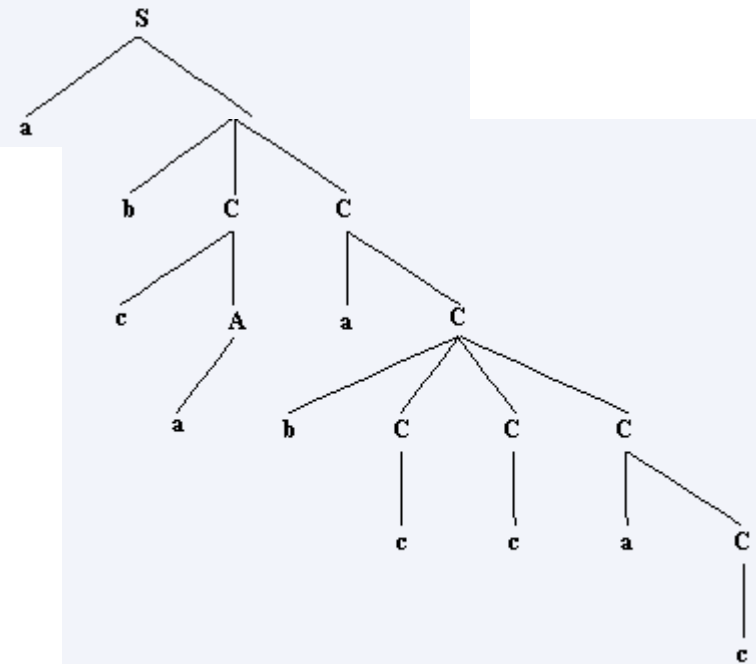
$A \Rightarrow aA \mid bCC \mid cD \mid a \mid \lambda$

$B \Rightarrow aB \mid bA \mid cBD \mid b$

$C \Rightarrow aC \mid bCCC \mid cA \mid c$

$D \Rightarrow aD \mid bC \mid cB$

$w = abcaabccac$  için türetme ağacı:



#### Örnek 4

Bağlamdan-bağımsız (CF)  $L_{5.4}$  dili  $\{a, b, c\}$  alfabesinde, içindeki **b** ve **c**'lerin sayılarının toplamı, **a**'ların sayısına eşit olan dizgiler (strings) kümesi olarak tanımlanıyor.  $L_{5.4}$  dili  $\lambda$  içermiyor.

$L_{5.4} = \{ab, ca, baab, acab, caaababc, \dots\}$

$L_{5.4}$  dilini boş yığıtla tanıyan deterministik ( $\lambda$  -hareketleri dışında) bir PDA tasarlamamız isteniyor. Bunun için:

- PDA'nın çalışma ilkesini (nasıl çalışacağını) belirleyip sözel olarak açıklayınız.
- PDA'nın durumlarını belirleyip tanımlayınız (durum sayısının gerektiği kadar ve olabildiğince az olması isteniyor).
- PDA'nın biçimsel tanımını veriniz.
- $w = aabcca$  tümcesinin PDA tarafından nasıl tanındığını ID'ler dizisi ile gösteriniz.

## Cevap 4

a) PDA'nın çalışma ilkesi:

Yığıt boşsa:

**a** okunursa yığita **A** eklenir,

**b** ya da **c** okunursa yığita **B** eklenir.

Yığıtın üstünde A varsa:

**a** okunursa yığita **A** eklenir,

**b** ya da **c** okunursa yığıttan **A** silinir.

Yığıtın üstünde B varsa:

**a** okunursa yığıttan **B** silinir,

**b** ya da **c** okunursa yığita **B** eklenir.

b) PDA'nın  $\lambda$ 'yı tanımaması için 2 durumlu olması gereklidir.

$q_0$  : ilk simgenin okunduğu durum.

$q_1$  : sonraki simgelerin okunduğu durum.

c) PDA'nın biçimsel tanımı:

$M_{5.4} = \langle Q, \Sigma, \Gamma, q_0, \delta, Z_0, \Phi \rangle$

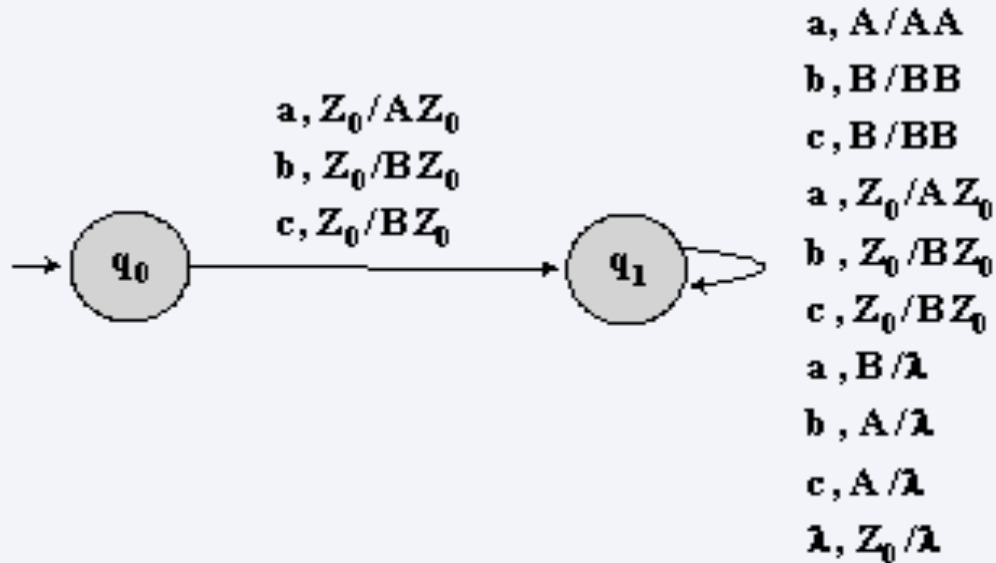
$Q = \{ q_0, q_1 \}$

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

$\Gamma = \{ Z_0, A, B \}$

$F = \Phi$

$\delta$  :



**G**<sub>5.5</sub> dilbilgisi aşağıdaki gibi tanımlanıyor.

**G**<sub>5.5</sub> =  $\langle V_N, V_T, P, S \rangle$

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

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

$P : S \Rightarrow aAb \mid cBd \mid ab \mid cd$

$A \Rightarrow aAb \mid ab$

$B \Rightarrow cBd \mid cd$

a) Bu dilbilgisinin türettiği dili tanımlayan matematiksel bir gösterim oluşturunuz.

b)  $L(G_{5.5})$  dilini boş yığıtla tanıyan deterministik bir PDA tasarlamamız isteniyor. PDA'nın çalışma ilkesini (nasıl çalışacağını) belirleyip sözel olarak açıklayınız.

c) PDA'nın durumlarını belirleyip tanımlayınız (durum sayısının gerektiği kadar ve olabildiğince az olması isteniyor).

d) PDA'nın biçimsel tanımını veriniz.

## Cevap 5

a)  $L_{5.5} = \{ a^n b^n \cup c^n d^n \mid n \geq 1 \}$

b) PDA'nın çalışma ilkesi:

ilk simge **a** ise: her **a** için yığta bir **A** ekler, her **b** için yığıttan bir **A** siler.

ilk simge **c** ise: her **c** için yığta bir **C** ekler, her **d** için yığıttan bir **C** siler.

c) Beş durumlu bir PDA tasarlanabilir

d)  $M_{5.5} = \langle Q, \Sigma, \Gamma, q_0, \delta, Z_0, \Phi \rangle$

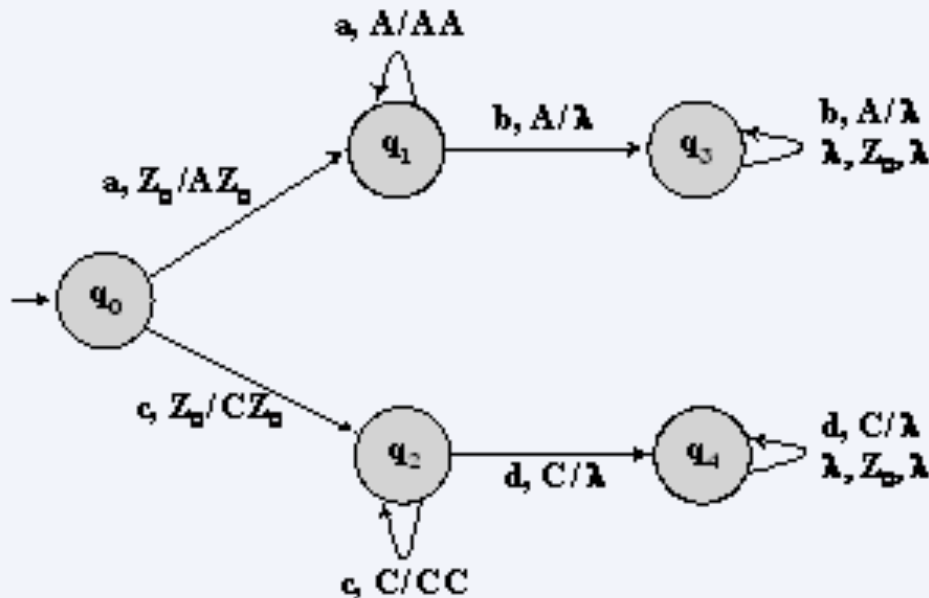
$Q = \{ q_0, q_1, q_2, q_3, q_4 \}$

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

$\Gamma = \{ Z_0, A, C \}$

$F = \Phi$

$\delta :$



Bağılamdan-bağımsız (CF)  $L_{5.6}$  dili  $\{a, b\}$  alfabesinde, eşit sayıda **a** ve **b** içeren dizgiler (strings) kümesi olarak tanımlanıyor ( $\lambda \in L_{5.6}$ ).  $L_{5.6}$  dilini boş yığıtla tanıyan deterministik bir PDA tasarlamamız isteniyor. Bunun için:

- a) PDA'nın çalışma ilkesini (nasıl çalışacağını) belirleyip sözel olarak açıklayınız.
- b) PDA'nın durumlarını belirleyip tanımlayınız (durum sayısının gerektiği kadar ve olabildiğince az olması isteniyor).
- c) PDA'nın biçimsel tanımını veriniz.
- d)  $w1 = aabbba$  ve  $w2 = baabba$  tümcelerinin PDA tarafından nasıl tanındığını ID dizileri ile gösteriniz.



## Cevap 6

a) PDA'nın çalışma ilkesi:

Okunan ilk simge yığığa eklenir.

Sonraki simgelerin her biri için:

Eğer okunan simge yığının üstündeki simge ile aynı ise,  
okunan simge yığığa eklenir.

Eğer okunan simge yığının üstündeki simgeden farklı ise,  
yığının üstündeki simge silinir.

Yığın boş ise, okunan simge yığığa eklenir.

b) PDA'da tek bir durum ( $q_0$ ) bulunması yeterlidir.

d)  $w_1 = aabbba$  :

$(q_0, aabbba, Z_0) \vdash (q_0, abbba, aZ_0) \vdash (q_0, bbba, aaZ_0)$   
 $\vdash (q_0, bba, aZ_0) \vdash (q_0, ba, Z_0)$   
 $\vdash (q_0, a, bZ_0) \vdash (q_0, \lambda, Z_0) \vdash (q_0, \lambda, \lambda)$

$w_2 = baabba$  :

$(q_0, baabba, Z_0) \vdash (q_0, aabba, bZ_0) \vdash (q_0, abba, Z_0)$   
 $\vdash (q_0, bba, aZ_0) \vdash (q_0, ba, Z_0)$   
 $\vdash (q_0, a, bZ_0) \vdash (q_0, \lambda, Z_0) \vdash (q_0, \lambda, \lambda)$

c)  $M_{5.6} = \langle Q, \Sigma, \Gamma, q_0, \delta, Z_0, \Phi \rangle$

$Q = \{ q_0 \}$

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

$\Gamma = \{ Z_0, a, b \}$

$F = \Phi$

$\delta$



$a, Z_0 / aZ_0$

$a, a / aa$

$b, a / \lambda$

$b, Z_0 / bZ_0$

$b, b / bb$

$a, b / \lambda$

$\lambda, Z_0 / \lambda$

Bağlamdan-bağımsız  $L_{5.7}$  dili aşağıdaki gibi tanımlanıyor:

$$L_{5.7} = \{ a^n (bc)^n \mid n \geq 1 \}$$

$L_{5.7}$  dilini boş yığıtla tanıyan deterministik bir PDA tasarlamamız isteniyor. Bunun için:

- PDA'nın çalışma ilkesini belirleyip sözel olarak açıklayınız.
- Durum sayısının gerektiği kadar ve olabildiğince az olmasını gözeterek PDA'nın durumlarını belirleyip tanımlayınız.
- PDA'nın biçimsel tanımını veriniz.
- $w = aabc bc$  tümcesinin PDA tarafından nasıl tanındığını ID dizisi ile gösteriniz.

## Cevap 7

a) PDA'nın çalışma ilkesi:

Her **a** için yığita bir **A** eklenir.

**a**'lar bittikten sonra her **(bc)** çifti için yığıttan bir **A** silinir.

b) PDA'nın 3 durumunun ( $q_0$ ,  $q_1$ ,  $q_2$ ) bulunması yeterlidir.

d)  $w_1 = aabcabc$  :

$(q_0, aabcabc, Z_0) \vdash (q_0, abcbc, AZ_0) \vdash (q_0, bcabc, AAZ_0)$

$\vdash (q_1, cbc, AAZ_0) \vdash (q_1, bc, AZ_0)$

$\vdash (q_1, c, AZ_0) \vdash (q_2, \lambda, Z_0) \vdash (q_2, \lambda, \lambda)$

c)  $M_{5.7} = \langle Q, \Sigma, \Gamma, q_0, \delta, Z_0, \Phi \rangle$

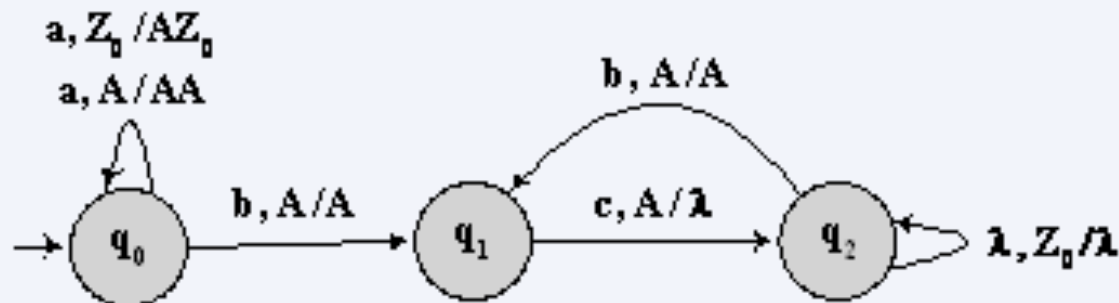
$Q = \{ q_0, q_1, q_2 \}$

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

$\Gamma = \{ Z_0, A \}$

$F = \Phi$

$\delta$  :



## Örnek 8

Bağlamdan-bağımsız  $L_{5.8}$  dili aşağıdaki gibi tanımlanıyor:

$$L_{5.8} = \{ (ab)^n (bc)^n \mid n \geq 1 \}$$

$L_{5.8}$  dilini boş yığıtla tanıyan deterministik bir PDA tasarlamamız isteniyor. Bunun için:

- a) PDA'nın çalışma ilkesini belirleyip sözel olarak açıklayınız.
- b) Durum sayısının gerektiği kadar ve olabildiğince az olmasını gözeterek PDA'nın durumlarını belirleyip tanımlayınız.
- c) PDA'nın biçimsel tanımını veriniz.

## Cevap 8

a) PDA'nın çalışma ilkesi:

**ab**'lere karşılık yığta **XY**'ler eklenir.

**ab**'ler bittikten sonra **bc**'ler okunur. Okunan her **b** için yığtadan bir **Y**, okunan her **c** için de yığtadan bir **X** silinir.

b) PDA'nın 5 durumunun bulunması yeterlidir.

c) PDA'nın biçimsel tanımı:

$M_{5.8} = \langle Q, \Sigma, \Gamma, q_0, \delta, Z_0, \Phi \rangle$

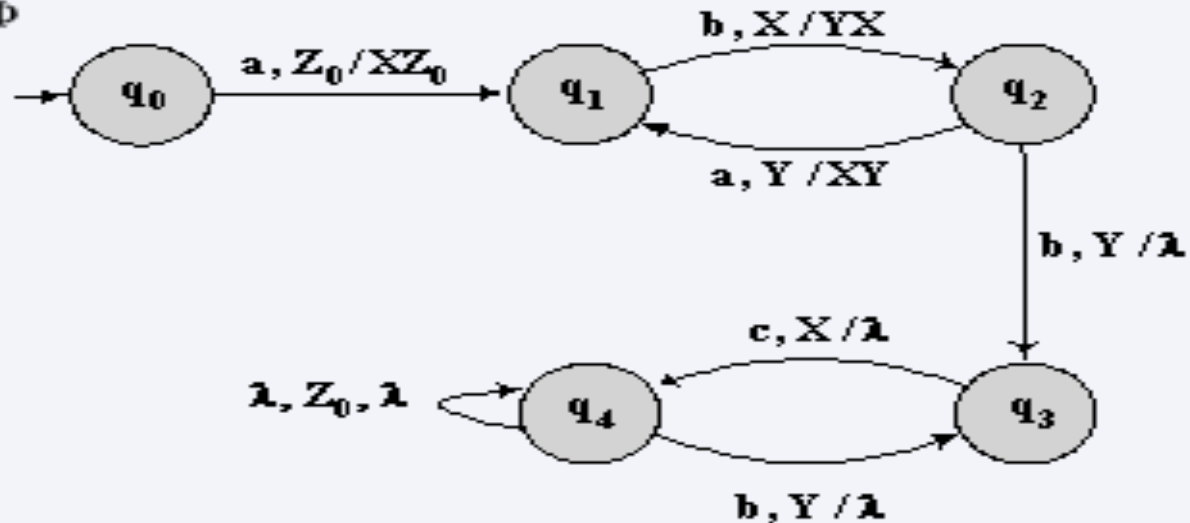
$Q = \{ q_0, q_1, q_2, q_3, q_4 \}$

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

$\Gamma = \{ Z_0, X, Y \}$

$F = \Phi$

$\delta :$



### Örnek 9

$$L_{5.9} = \{ a^n b^m c^k d^s \mid 1 \leq n \leq m \leq 2n, 1 \leq s \leq k \leq 2s \}$$

- a) Yukarıda küme tanımı verilen bağlamdan-bağımsız  $L_{5.9}$  dilini türeten bir dilbilgisi tanımlayınız.
- b)  $L_{5.9}$  dilini boş yığıtla tanıyan bir PDA oluşturunuz.

## Cevap 9

a)  $G_{5.9} = \langle V_N, V_T, P, S \rangle$

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

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

$P : S \Rightarrow AB$

$A \Rightarrow aAb \mid aAbb \mid ab \mid abb$

$B \Rightarrow cBd \mid ccBd \mid cd \mid ccd$

b) PDA'nın çalışma ilkesi (PDA deterministik değildir):

- \* Okunan her **a** için yığıtta bir ya da iki **X** eklenir.
- \* Okunan her **b** için yığıttan bir **X** çıkarılır.
- \* Son **b** okunduğunda yığıt boşalmış olmalıdır.
- \* Okunan her **c** için yığıtta bir **Y** eklenir.
- \* Okunan her **d** için yığıttan bir ya da iki **Y** çıkarılır.

PDA'nın biçimsel tanımı:

$M_{5.9} = \langle Q, \Sigma, \Gamma, q_0, \delta, Z_0, \Phi \rangle$

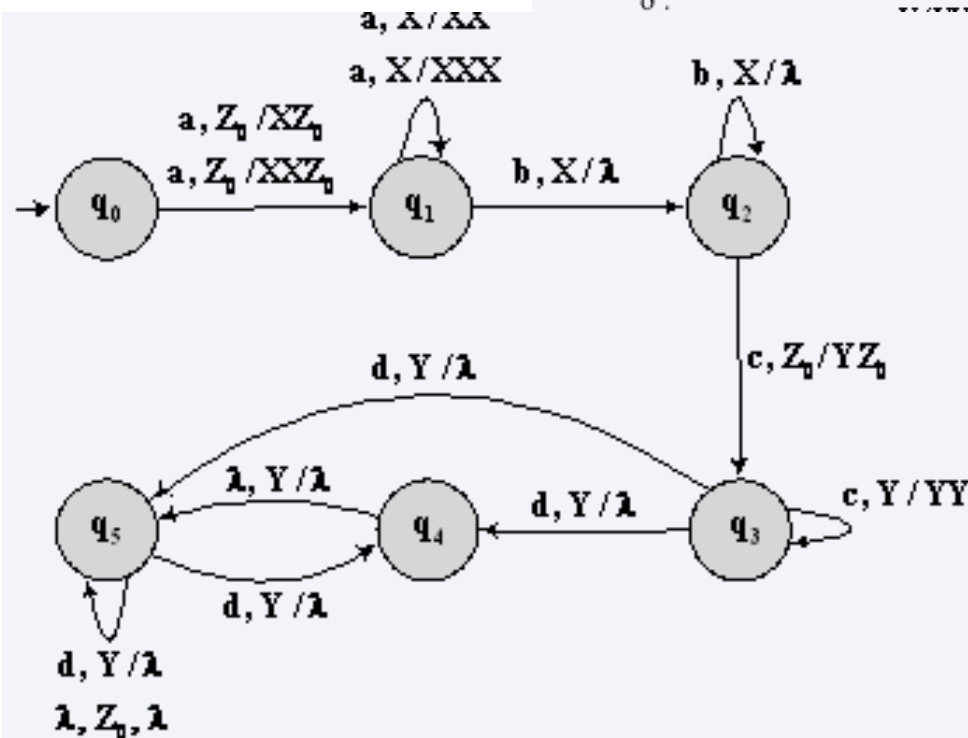
$Q = \{ q_0, q_1, q_2, q_3, q_4, q_5 \}$

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

$\Gamma = \{ Z_0, X, Y \}$

$F = \Phi$

$\delta :$



Bağılamdan-bağımsız  $L_{5.10}$  dili aşağıdaki gibi tanımlanıyor:

$$L_{5.10} = \{ a^n b^m c^k d^n \mid n > 0, m > 0, k \geq m \}$$

- a)  $L_{5.10}$  dilini türeten bir dilbilgisi oluşturunuz.
- b)  $L_{5.10}$  dilini boş yığıtla tanıyan deterministik bir PDA tasarlamamız isteniyor. Bunun için PDA'nın çalışma ilkesini (nasıl çalışacağını) belirleyip sözel olarak açıklayınız, durumları belirleyip tanımlayınız ve PDA'nın biçimsel tanımını oluşturunuz.



## Cevap 10

a)  $G_{5.10} = \langle V_N, V_T, P, S \rangle$

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

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

$$P : S \Rightarrow aSd \mid aAd$$

$$A \Rightarrow bAc \mid Ac \mid bc$$

$M_{5.10} = \langle Q, \Sigma, \Gamma, q_0, \delta, Z_0, \Phi \rangle$

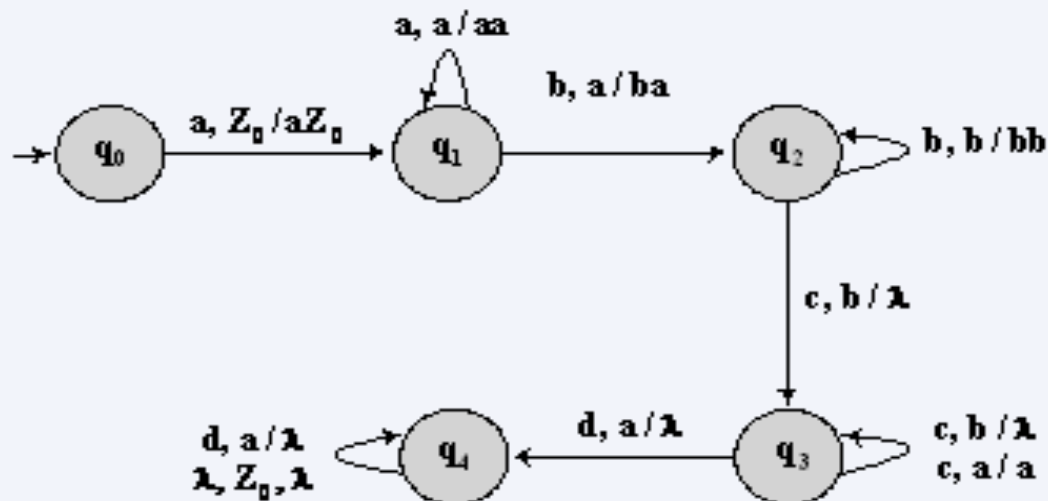
$$Q = \{ q_0, q_1, q_2, q_3, q_4 \}$$

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

$$\Gamma = \{ Z_0, a, b \}$$

$$F = \Phi$$

$\delta :$



b) PDA'nın çalışma ilkesi:

**a**'lar okunarak yığığa eklenir.

**b**'ler okunarak yığığa eklenir.

yığığındaki **b**'ler bitinceğe kadar okunan her **c** için

yığığından bir **b** çıkarılır.

yığığındaki **b**'ler bitikten sonra, okunan **c**'ler için yığığına bir işlem yapılmaz.

okunan her **d** için yığığından bir **a** çıkarılır.

$$L_{5.11} = \{ 1^n 0^k 1^m 0^n \mid n \geq 2, 1 \leq k < n \}$$

Yukarıda küme tanımı verilen dili tanıyan bir PDA tasarlamamız isteniyor.

Tasarlayacağınız PDA'nın mutlaka deterministik olması gerekmiyor. Boş yığıtla ya da uç durumla tanıyan bir PDA tasarlayabilirsiniz.

a) PDA'nın çalışma ilkesini sözel olarak açıklayınız.

b) PDA'nın biçimsel tanımını veriniz.

c)  $w = 110100$  için, anlık tanımlar dizisini oluşturarak, PDA'nın bu dizgiyi tanıyıp tanımadığını gösteriniz.

# Cevap 11

a) PDA'nın çalışma ilkesi:

ilk gruptaki 1'ler (en az 2 tane) okunarak yığta eklenir.

ikinci gruptaki 0'lar (en az 1 tane) okunur ve her 0 için yığıttan bir 1 silinir.

üçüncü gruptaki 1'ler okunarak yığta eklenir.

dördüncü gruptaki 0'lar okunur ve her 0 için yığıttan bir 1 sililir.

b) PDA'nın biçimsel tanımı:

$$M_{5.10} = \langle Q, \Sigma, \Gamma, q_0, \delta, Z_0, \Phi \rangle$$

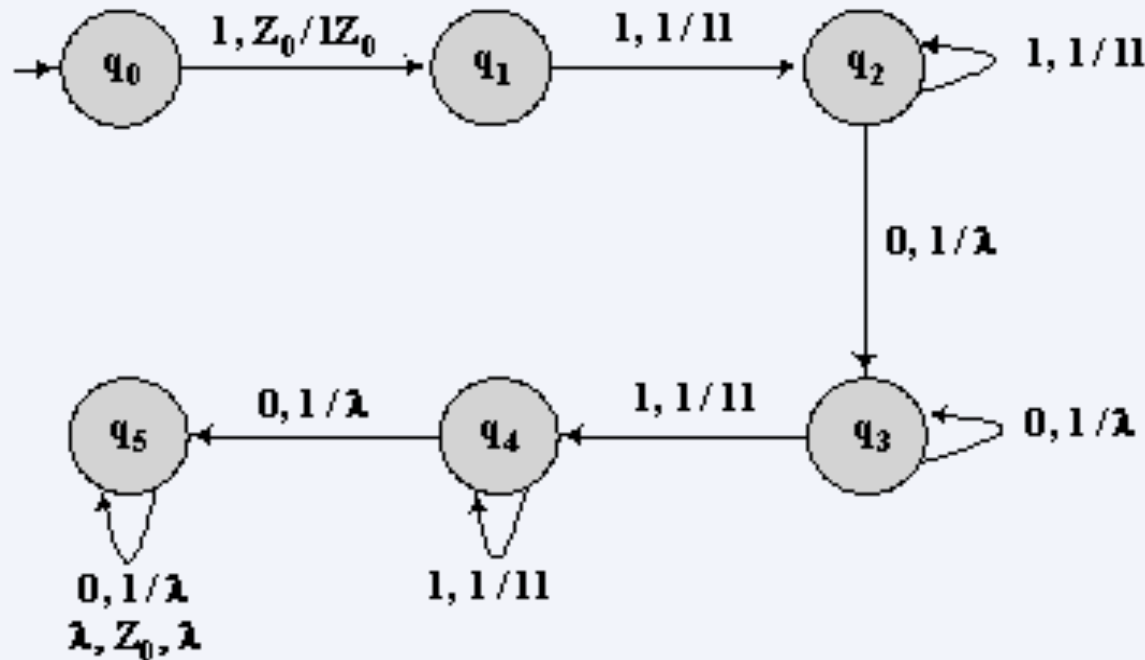
$$Q = \{ q_0, q_1, q_2, q_3, q_4, q_5 \}$$

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

$$\Gamma = \{ Z_0, 0, 1 \}$$

$$F = \Phi$$

$\delta :$



$$L_{5.12} = \{ 1^n 0^k \mid n \geq 1, n \leq k < 2n \}$$

Yukarıda küme tanımı verilen dili tanıyan bir PDA tasarlamamız isteniyor.

Tasarlayacağınız PDA'nın mutlaka deterministik olması gerekmiyor. Boş yığıtla ya da uç durumla tanıyan bir PDA tasarlayabilirsiniz.

- a) PDA'nın çalışma ilkesini sözel olarak açıklayınız.
- b) PDA'nın biçimsel tanımını veriniz.
- c)  $w = 1110000$  için, anlık tanımlar dizisini oluşturarak, PDA'nın bu dizgiyi tanıyıp tanımadığını gösteriniz.

# Cevap 12

a) PDA'nın çalışma ilkesi (PDA deterministik değildir):

⇒ Okunan her 1 için yığta bir ya da iki X eklenir.

⇒ Okunan her 0 için yığıttan bir X çıkarılır.

b) PDA'nın biçimsel tanımı:

$$M_{5.12} = \langle Q, \Sigma, \Gamma, q_0, \delta, Z_0, \Phi \rangle$$

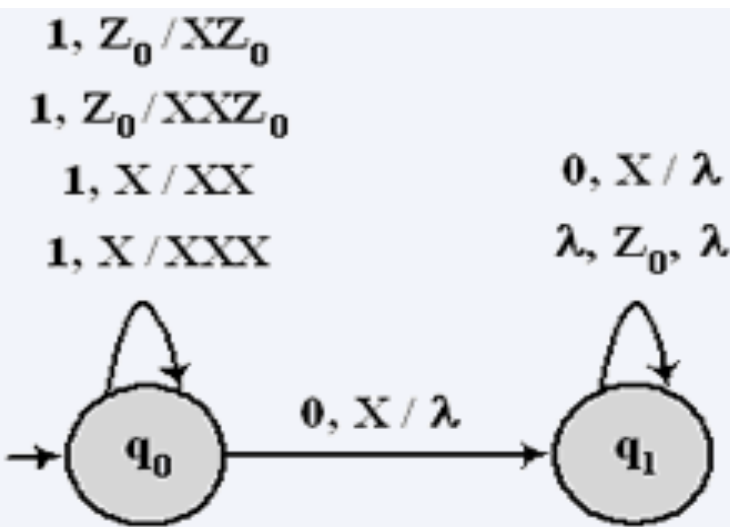
$$Q = \{ q_0, q_1 \}$$

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

$$\Gamma = \{ Z_0, X \}$$

$$F = \Phi$$

$\delta :$



Bağlamdan-bağımsız  $L_{5.13}$  dili aşağıdaki gibi tanımlanıyor.

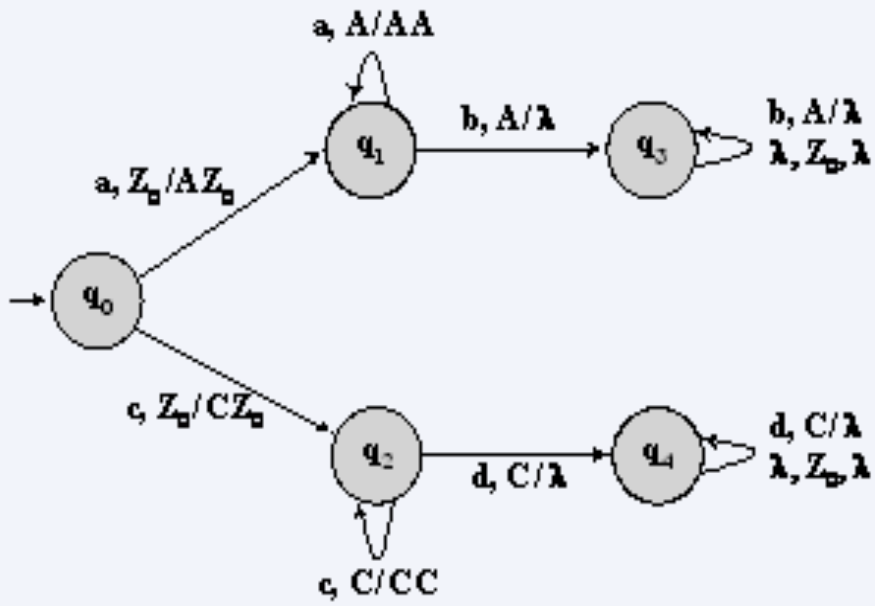
$$L_{5.13} = \{ a^n b c^n \cup c^n b a^n \mid n > 0 \}$$

- a)  $L_{5.13}$  dilini türeten bir dilbilgisi oluşturunuz.
- b)  $L_{5.13}$  dilini boş yığıtla tanıyan deterministik bir PDA tasarlamamız isteniyor. Bunun için PDA'nın çalışma ilkesini (nasıl çalışacağını) belirleyip sözel olarak açıklayınız, durumları belirleyip tanımlayınız ve PDA'nın biçimsel tanımını oluşturunuz.

# Cevap 13

- a)  $G_{5.13} = \langle V_N, V_T, P, S \rangle$   
 $V_N = \{ S, B_1, B_2 \}$   
 $V_T = \{ a, b, c \}$   
 $P : S \Rightarrow aB_1c \mid cB_2a$   
 $B_1 \Rightarrow aB_1c \mid b$   
 $B_2 \Rightarrow cB_2a \mid b$

- b) PDA'nın biçimsel tanımı:  
 $M_{5.13} = \langle Q, \Sigma, \Gamma, q_0, \delta, Z_0, \Phi \rangle$   
 $Q = \{ q_0, q_1, q_2, q_3, q_4 \}$   
 $\Sigma = \{ a, b, c \}$   
 $\Gamma = \{ Z_0, A, C \}$   
 $F = \Phi$   
 $\delta :$



### Örnek 14

$G_{5.14}$  dilbilgisi aşağıdaki gibi tanımlanıyor.

$$G_{5.14} = \langle V_N, V_T, P, S \rangle$$

$$V_N = \{ S \}$$

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

$$P : S \Rightarrow aSb \mid \lambda$$

a)  $G_{5.14}$  dilbilgisini GNF biçimine dönüştürünüz. Dönüştürme işlemini sezgisel olarak kolaylıkla yapabilirsiniz.

b)  $L(G_{5.14})$  dilini tanıyan PDA'nın tanımını, 5.2.1'deki 1. yöntemi kullanarak, sistematik biçimde oluşturunuz.



## Cevap 14

a)  $G_{5.14} = \langle V_N, V_T, P, S \rangle$

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

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

$$P : S \Rightarrow aSB \mid \lambda$$

$$B \Rightarrow b$$

b)  $L(G_{5.14})$  dilini tanıyan PDA:

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

$$Q = \{ q_0 \}$$

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

$$\Gamma = \{ S, B \}$$

$$Z_0 = S$$

$$F = \emptyset$$

$\delta :$



$a, S/SB$

$b, B/\lambda$

$\lambda, S/\lambda$