

Hafta 13

Örnek:

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:

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

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

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

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

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 yığtın çıkarır (siler).
- Giriş dizgisinin tüm simgeleri okunduktan sonra, eğer yığtın tepesinde Z_0 varsa, bunu da siler ve yığtı tamamen boşaltır.

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

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

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

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

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

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

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

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

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

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

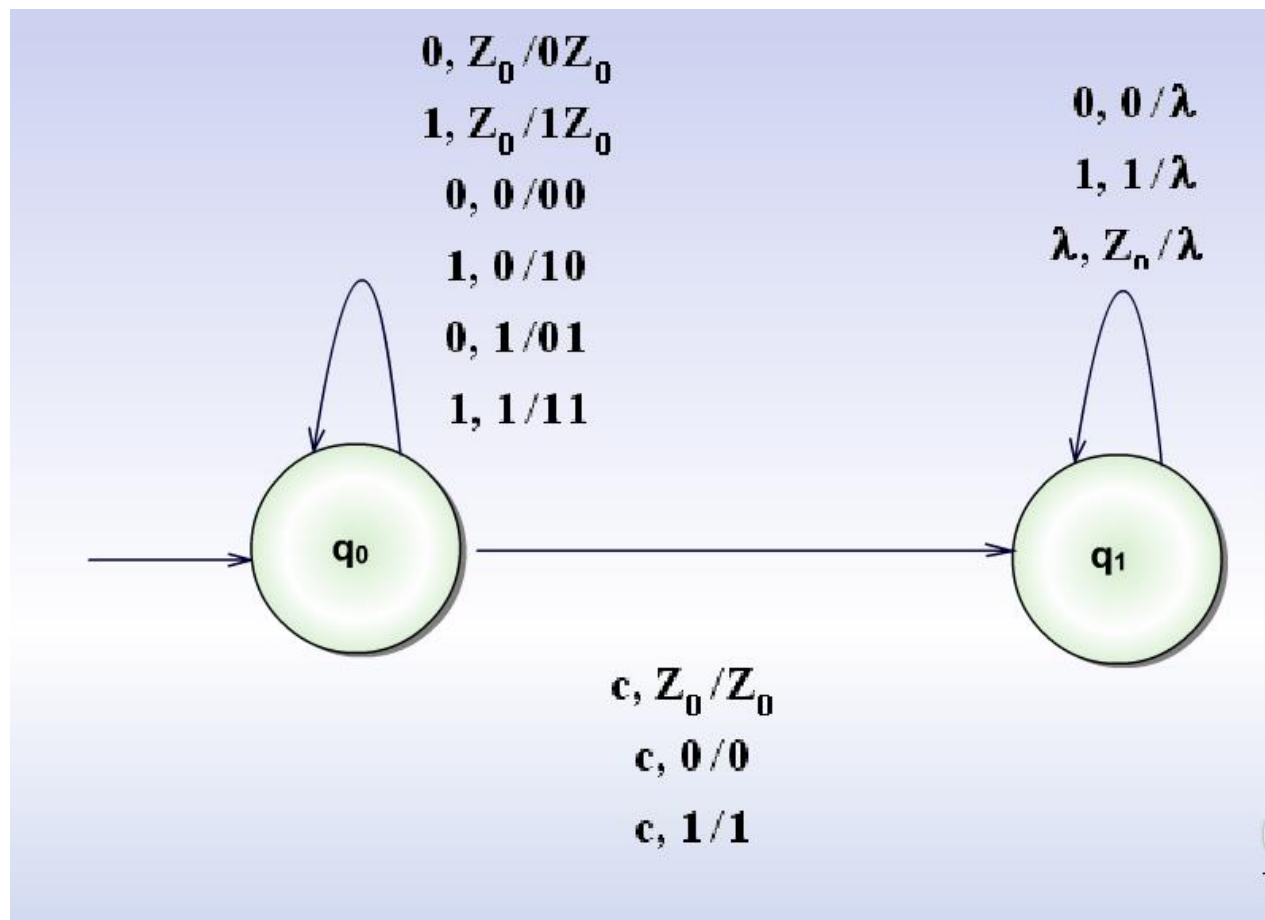
$$\delta(q_1, 1, 1) = (q_1, \lambda)$$

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

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

Örnek olarak verilen **M_{5.1}** deterministik bir PDA'dır.

PDA'nın hareket işlevi yukarıdaki 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 yığıtta eklenen yığıt simgeleri dizgisini göstermektedir. Bu üçlü İngilizce sözcüklerle kısaca “read, pop, push” üçlüsü olarak gösterilebilir. Temel PDA modeline göre (Bkz. 5.1) bu üçlünün birinci ve sonuncu elemanı **λ** olabilir.



Örnek 2:

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:

$$G_{5,4} = \langle V_N, V_T, P, S \rangle$$

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

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

$$P : S \Rightarrow ABA$$

$$A \Rightarrow aA \mid a$$

$$B \Rightarrow bBb \mid bAb$$

L_{5,4}'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, 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 yığıtta eklenir; 4. öbekteki b'ler okunurken de yığıt boşaltılır.

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

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

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

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

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

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

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

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

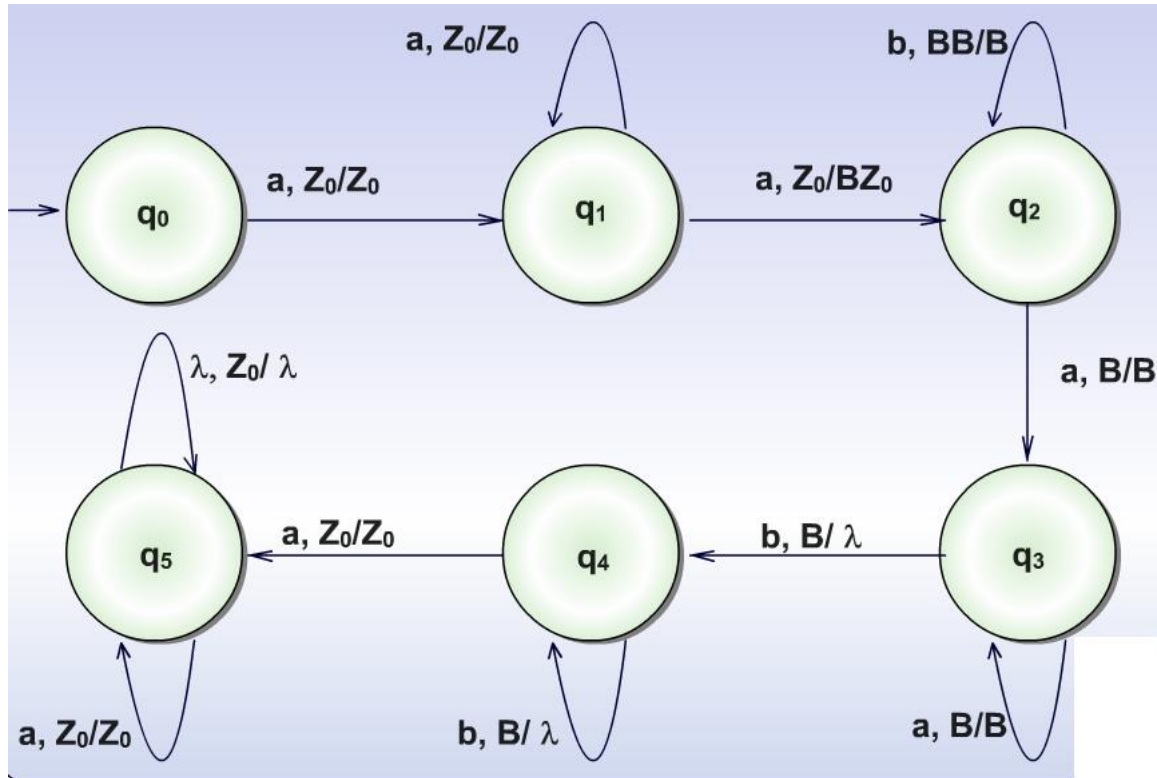
$$\delta(q_4, b, B) = (q_4, \lambda)$$

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

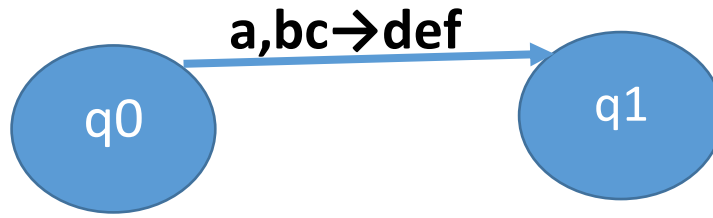
$$\delta(q_5, a, Z_0) = (q_5, Z_0)$$

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

PDA'nın hareket işlevi yukarıdaki gibi tanımlanabileceği gibi, Çizim 5.3'deki çizim ile de tanımlanabilir.



Örnek 3:



Bu makinada giriş olarak a okunduğunda stack ten bc yi sil yerine def yaz.

b
c
Z_0

d
e
f
Z_0

Örnek 4:

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

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:

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

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

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

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

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 almamaktadır. Bu dizgileri tanıyan PDA:

- Önce w 'nin simgelerini okur ve yığıtı 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 yığıtı ekleme yapar; ikinci harekette ise PDA silme durumuna (q_1) geçer ve yığıtın tepesindeki simgeyi siler.
- Silme durumuna (q_1) geçen PDA w^R 'nin simgelerini okur, ve okuduğu her simge için eğer yığıtın tepesinde bu simgenin karşılığı (0 için A , 1 için B) varsa bunu yığıttan çıkarır (siler).
- 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:

$$\delta : \delta(q_0, 0, Z_0) = (q_0, AZ_0)$$

$$\delta(q_0, 1, Z_0) = (q_0, BZ_0)$$

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

$$\delta(q_0, 0, A) = \{(q_0, AA), (q_1, \lambda)\}$$

$$\delta(q_0, 1, A) = (q_0, BA)$$

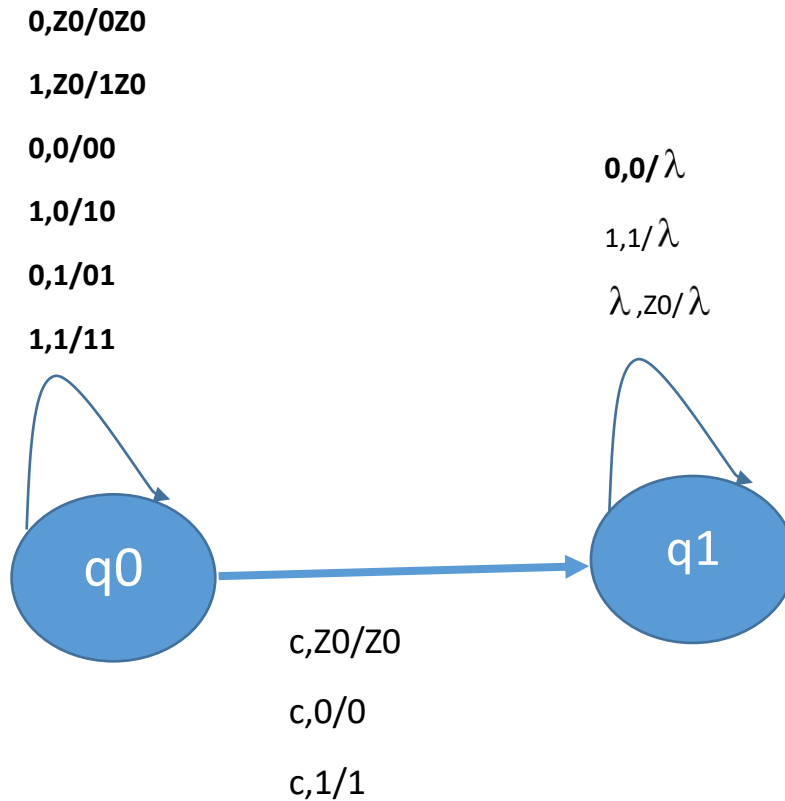
$$\delta(q_0, 0, B) = (q_0, AB)$$

$$\delta(q_0, 1, B) = \{(q_0, BB), (q_1, \lambda)\}$$

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

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

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



Örnek 5:

$$L = \{a^i b^{i+j} a^j \mid i > 0, j \geq 0\}$$

$$L = \{a^i b^i b^j a^j \mid i > 0, j \geq 0\}$$

$\Downarrow \quad \Downarrow$

A B

$$L \rightarrow AB$$

$$L \rightarrow aAb \mid ab$$

$$B \rightarrow bBa \mid \varepsilon$$

Önemli Not :b, a / ε yığıta b geldiğinde a yı sil yerine bir şey yazma.

