

# Otomata Teorisi (BİL 2114)

Fırat İsmailođlu

Hafta 11:  
Turing Makinesi (Bölüm 2)



# Hafta 11

## Plan

1. Turing Makinesi Dizaynı
2. Turing Sava
3. Turing Makinesi Varyasyonları
  - i. Kal opsiyonlu TM
  - ii. Çok Kayıtlı TM
  - iii. Çok Bantlı TM
  - iv. Nondeterministik TM

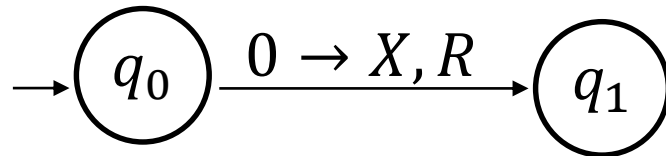


# Turing Makinesi Dizaynı

9. haftada  $L = \{0^n 1^n \mid n \geq 1\}$  dilini tanıyan bir TM göstermistik. Simdi bu TM'yi dizayn etmeye çalışalım.

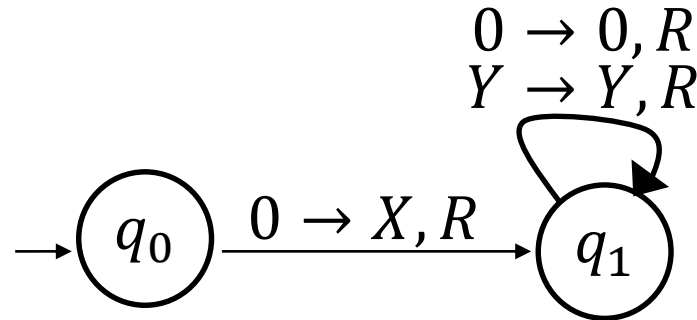
Bu TM'yi dizayn ederken ana çıkış noktamız her bir adımda en soldaki 0 harfi üzerine  $X$  yazmak, daha sonra sağa doğru ilerleyip en soldaki 1 harfini bulmak, bunun üzerine  $Y$  yazmak ve daha sonra yine 0 bulmak umuduyla sola doğru ilerlemektir.

**Aşama I:** 0 gördüğümüz yere  $X$  yazarak sağa doğru ilerleyelim:

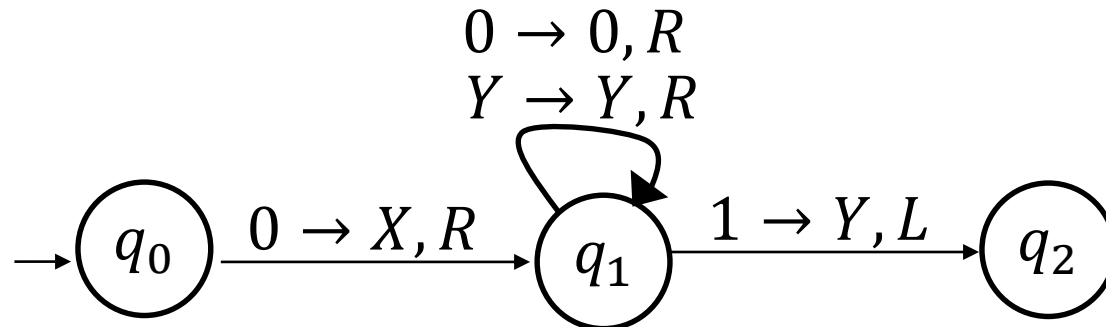


# Turing Makinesi Dizaynı

**Aşama 2:** (En soldaki) 1 harfini bulana kadar sağa doğru ilerleyelim. İlerlerken önümüze 0 harfi ve  $Y$  çıkabilir, bunlara dokunmayalım.

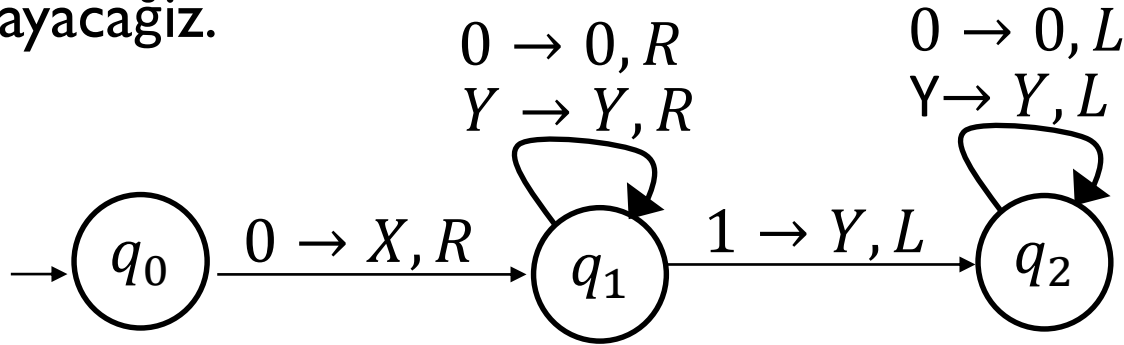


**Aşama 3:** 1 harfini bulduğumuzda üzerine  $Y$  yazalım (bu, bir önce yazdığımız  $X$  ile  $Y$ 'nin eşleştiğini gösterir). Daha sonra tekrar en soldaki 0 harfini bulmak için sola ilerleyelim.

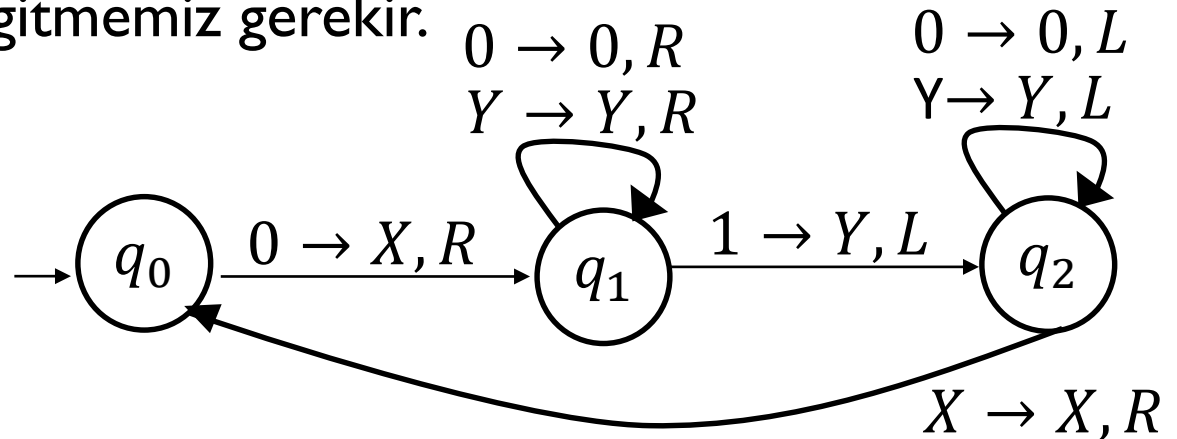


## Turing Makinesi Dizaynı

**Aşama 4:** Buarada amacımız en soldaki (varsa) 0 harfini bulmak. Bu harf en sağdaki  $X$ 'in bir sağında olacaktır. O yuzden  $X$ 'i bulana kadar sola dogru ilerleyelim. Önümüze çıkan 0 ve  $Y$  harfine dokunmayacağız.

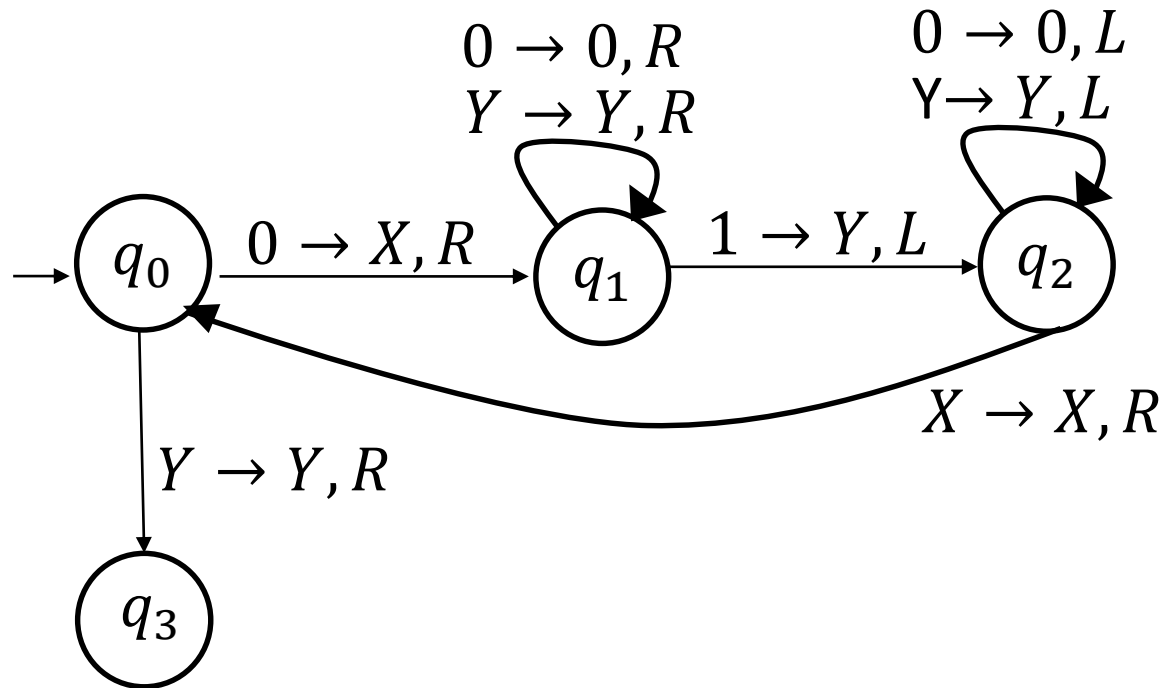


**Aşama 5:**  $X$ 'i bulduktan sonra bir sağındaki 0'a geçip bunun üzerine  $X$  yazacağız. 0'ın üzerine  $X$  yazılmasından  $q_0$  sorumlu idi. O halde  $q_2$ 'den  $q_0$ 'a gitmemiz gerekir.



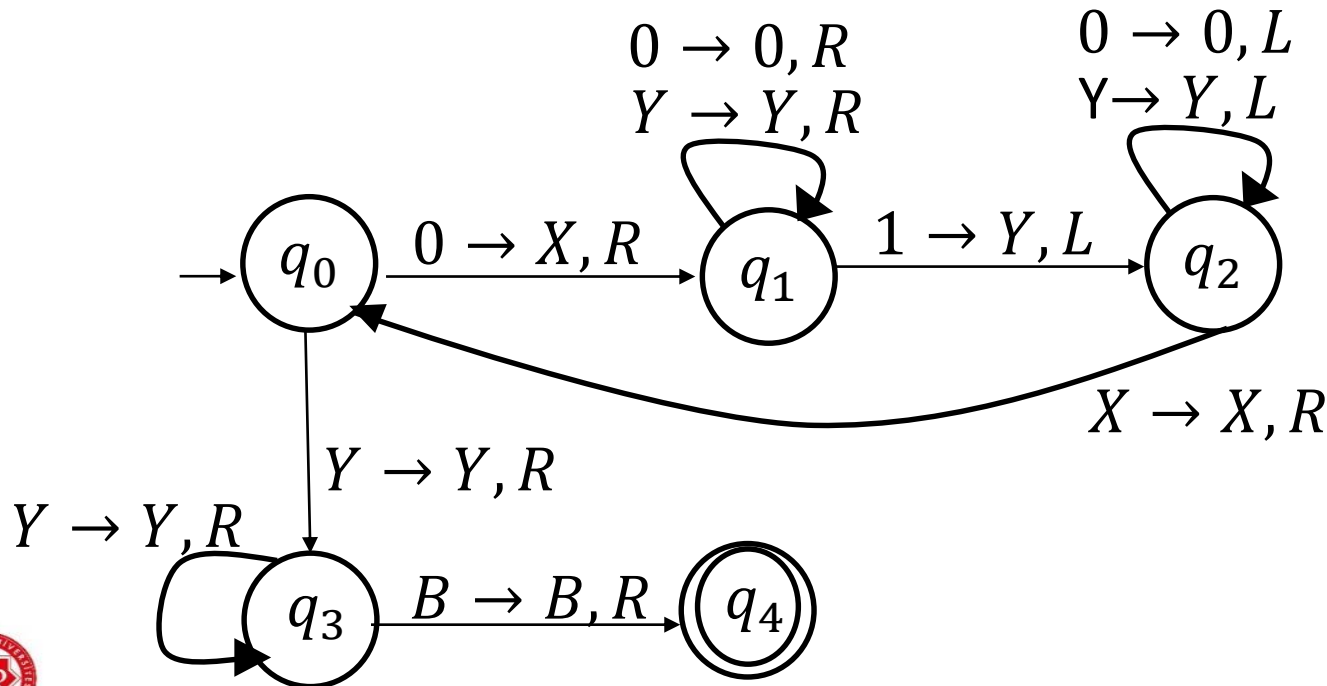
## Turing Makinesi Dizaynı

**Aşama 6:** Her bir  $q_0 - q_2 - q_0$  döngüsünde bir 0 harfi  $X$  yapılır, buna karşılık bir 1 harfi bulunarak  $Y$  yapılır. Böylece her bir döngüde bir  $(0 - 1)$  çifti eşleştirilmiş olur.  $q_0$ 'da iken  $Y$  gelmesi durumunda 0'lar bitmiş, tamamı  $X$  olmuştur. O yüzden bu durumda  $Y$ 'leri geçerek sağa doğru ilerleyip nentin sonuna gelmeye çalışalım.



## Turing Makinesi Dizaynı

**Aşama 7:**  $Y$ 'ler bitene kadar sağa ilerlemeye devam edelim. Bu  $Y$ 'lerin sonunda boşluk ( $B$ ) görürsek tüm  $X$ 'ler ile  $Y$ 'ler birbiriyle eşleşmiş demektir. Bu durumda final durumuna gidebiliriz. Eğer  $Y$ 'lerden sonra  $B$  değil  $1$  görürsek, bu durumda ilk başta verilen kelimedeki  $1$  sayısı  $0$  sayısından fazla olduğu anlamına gelir.  $q_3$ 'de kalırız. Bu kelimeyi kabul etmeyiz. Not eğer  $0$  sayısı  $1$  sayısından fazla ise  $q_1$  de takili kalırız, final durumuna gidemeyiz.



## Turing Savı (Turing Tezi)

**Turing Savı:** Herhangi bir bilgisayarla yapılan her şey bir Turing Makinesi ile de yapılabilir.

Bu savın doğruluğunu ispat edemeyiz, ama yanlış olduğunu da gösteremeyiz. Günümüze kadar bu sav çürütülememiştir. Yani bir bilgisayarın yaptığı bir hesaplama için bir Turing makinesinin bulunamadığı hiç olmamıştır.

Nasil ki Newton'un hareket kanunları Fizik biliminin temelini oluşturuyorsa, Turing Savı da bilgisayar bilimlerinin temelini oluşturur. Bu temele dayanarak algoritma tanımı verebiliriz.

**Algoritma Tanımı:** Bir  $f: T \rightarrow D$  fonksiyonun algoritması bir Turing makinesidir, öyleki her  $t \in T$  için bu  $t$  elamanini ilgili TM'nin bantına yazdığımızda, TM'nin final durumuna vardığımızda bantta  $f(t) \in D$  yazılmış olur.



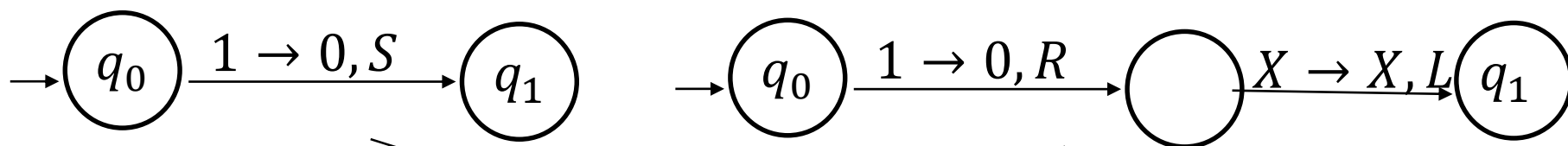
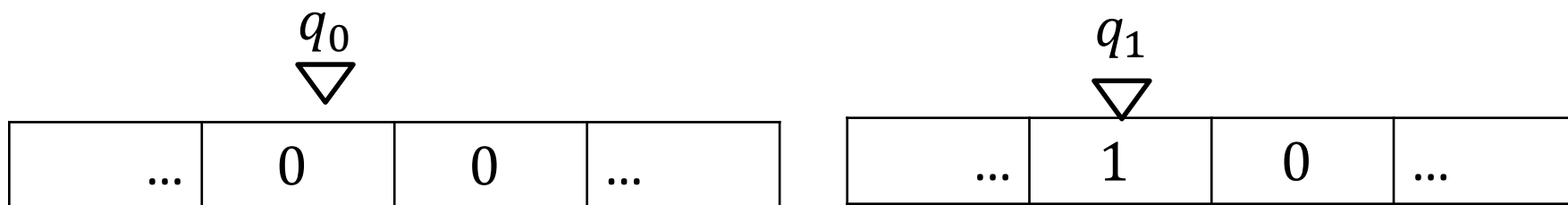


# Turing Makinesi Varyasyonları

## I. Kal (Stay) Opsiyonlu TM

Kal opsiyonlu TM'de okuma-yazma basi, banttın bir harfi okuduktan sonra, saga yada sola hareket etmeden okudugu harfin hucresinde kalabilir. Boylece gecis fonksiyonu soyle olur:

$$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, S\}$$



birbirlerine denktirler.



## 2. Çok Kayitli (Multipletrack) TM

Cok kayitli TM'de birden fazla bant bulunur. Okuma-yazma kafasi bir tanedir ve bu bantlari ayni anda okur ve yazar. Daha sonra kafa  $b \rightarrow B, R$  saga yada sola gider. Gecis fonksiyonu soyle olur:

$$\delta: Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R\}$$

$k$  tane banttan ayni anda okuyoruz.

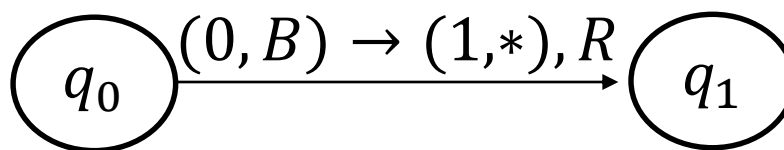
$k$  tane banta ayni anda yaziyoruz.

$q_0$   
▽

$B$	0	1	$B$
$B$	$B$	$B$	$B$

$q_1$   
▽

$B$	1	1	$B$
$B$	*	$B$	$B$



### 3. Çok Bantlı (Multitape) Turing Makinesi

Cokbantli TM, birden fazla banta ve her bir bant icin ayri bir okuma-yazma kafasina sahip TM'lere denir.

Baslangicta yalnızca 1. banta verilen kelime yazilir. Diger tum bantların tum hucrelerine bosluk  $B$  yazilir.

Bu tur TM'ler gecis fonksiyonu asagidaki gibidir:

$$\delta: Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R\}^k$$

$k$  tane banttan ayni anda okuyoruz.

$k$  tane banta ayni anda yaziyoruz.

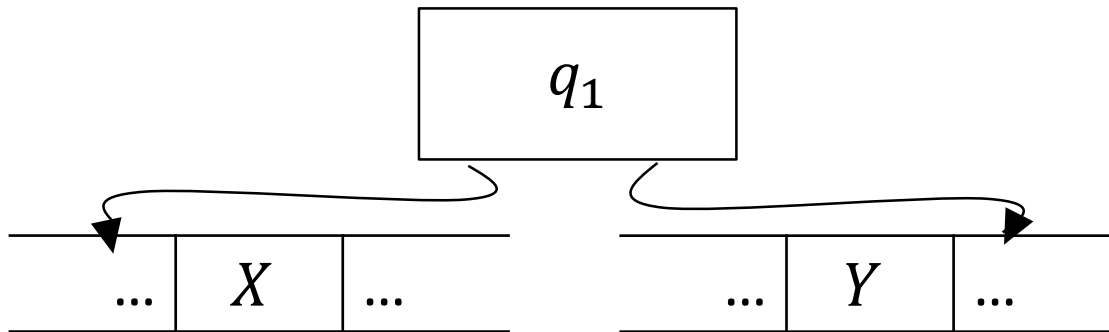
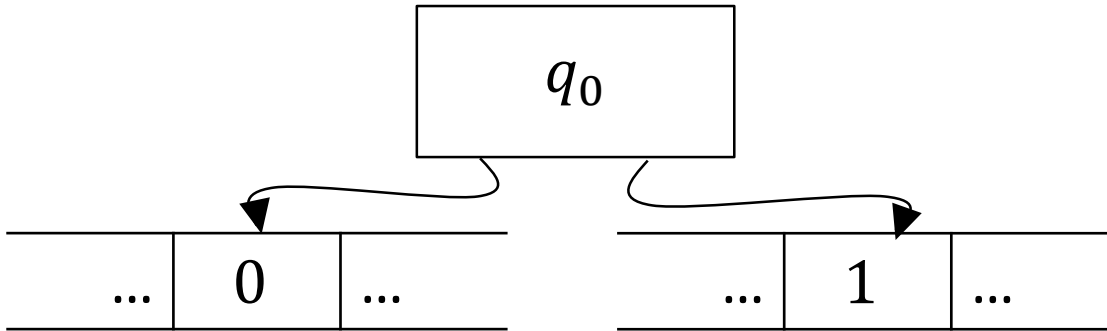
$k$  tane bant ayri ayri saga yada sola gidiyor



## Çok Bantlı Turing Makinesi

ör.  $\delta(q_0, 0, 1) = (q_1, X, Y, L, R)$

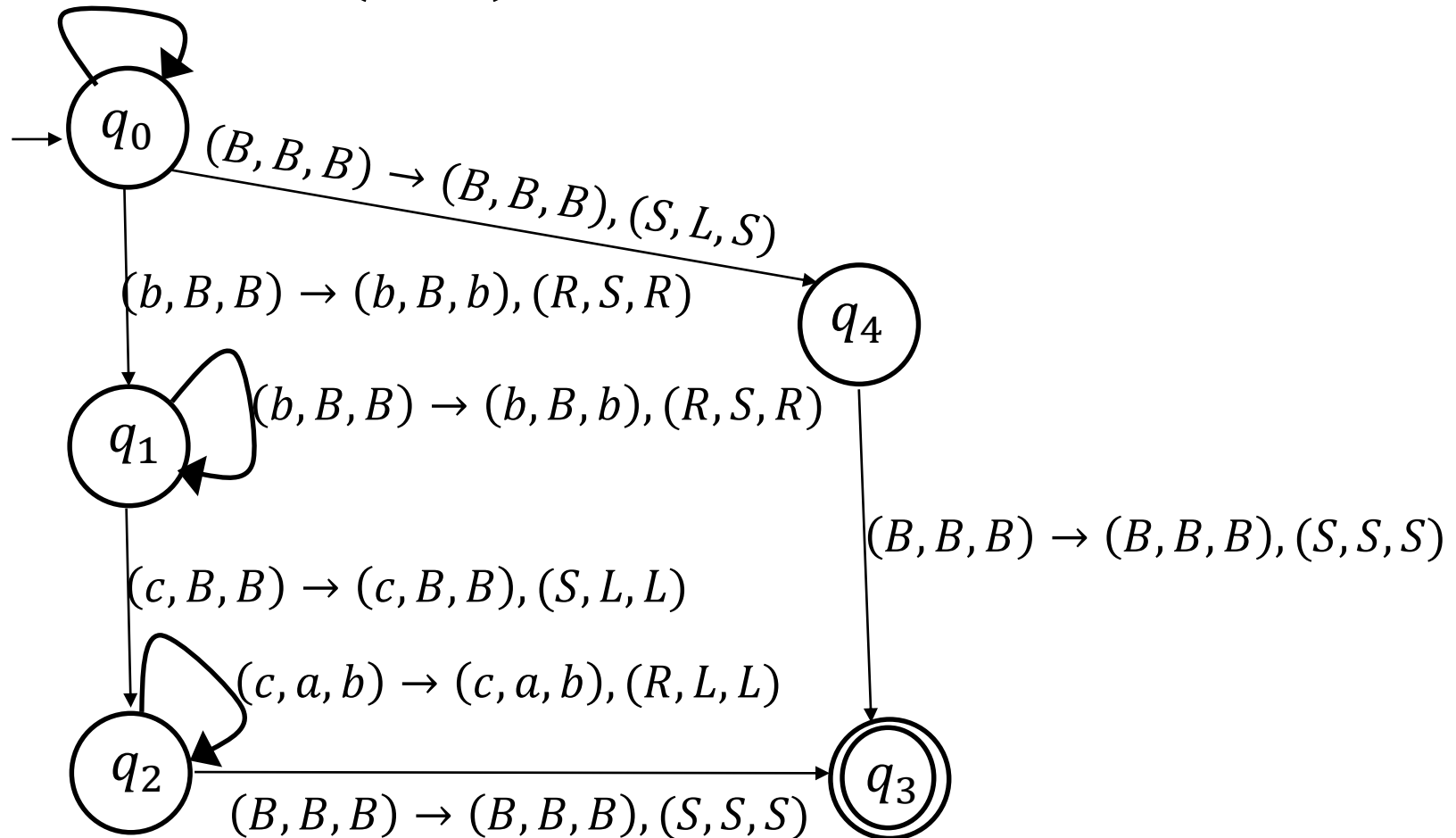
Bu geciste, TM  $q_0$ 'da iken birinci banttın 0, ikinci banttın 1 okundugunda  $q_1$  durumuna gidilir. Birinci banta  $X$ , ikinci banta  $Y$  yazilir. Birinci bantta sola, ikinci bantta saga gidilir.



## Çok Bantlı Turing Makinesi

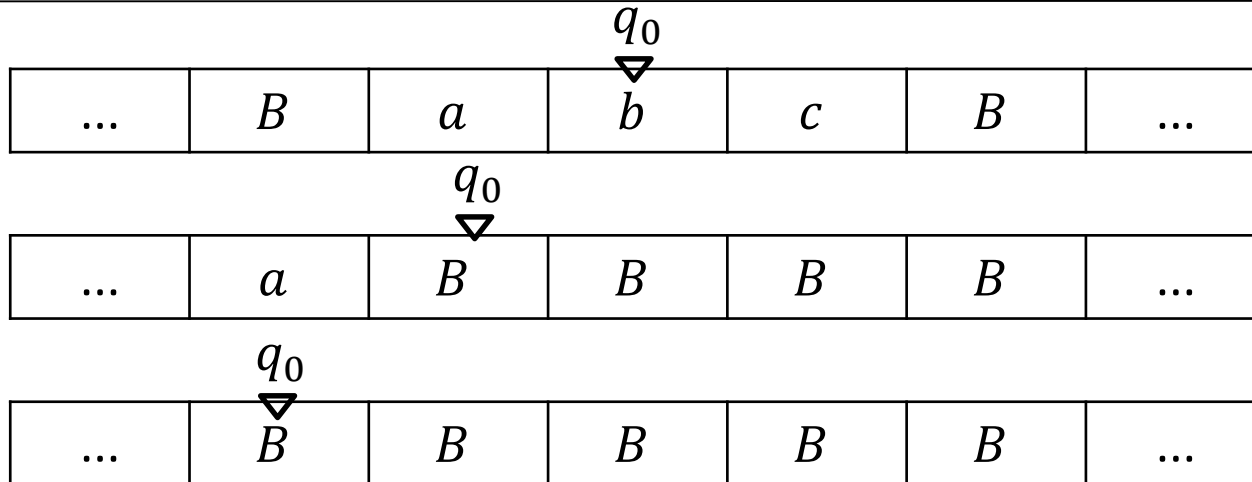
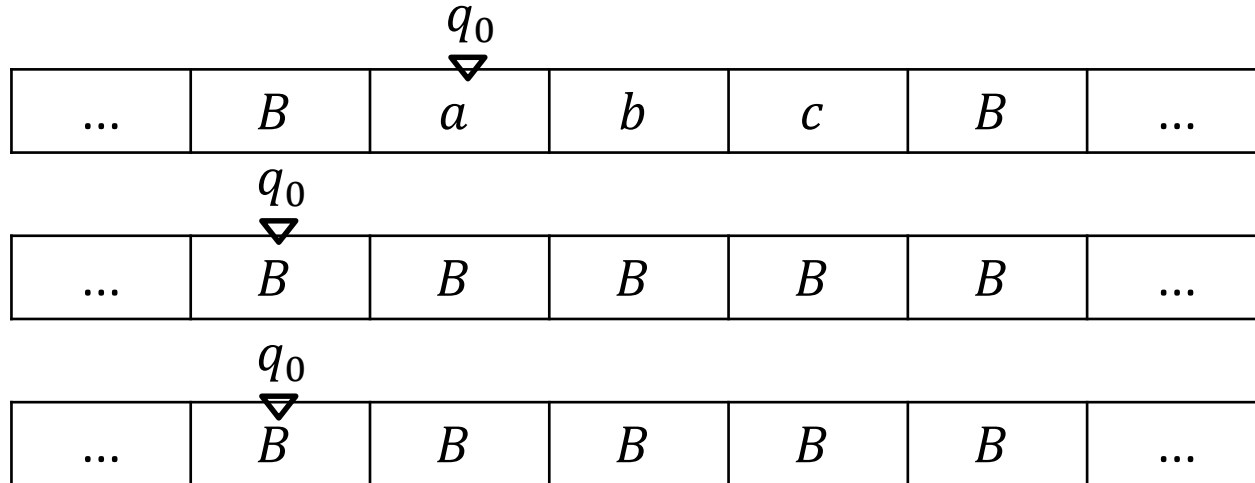
ör. Aşağıdaki çok bantlı TM  $L = \{a^n b^n c^n \mid n \geq 1\}$  dilini tanır.

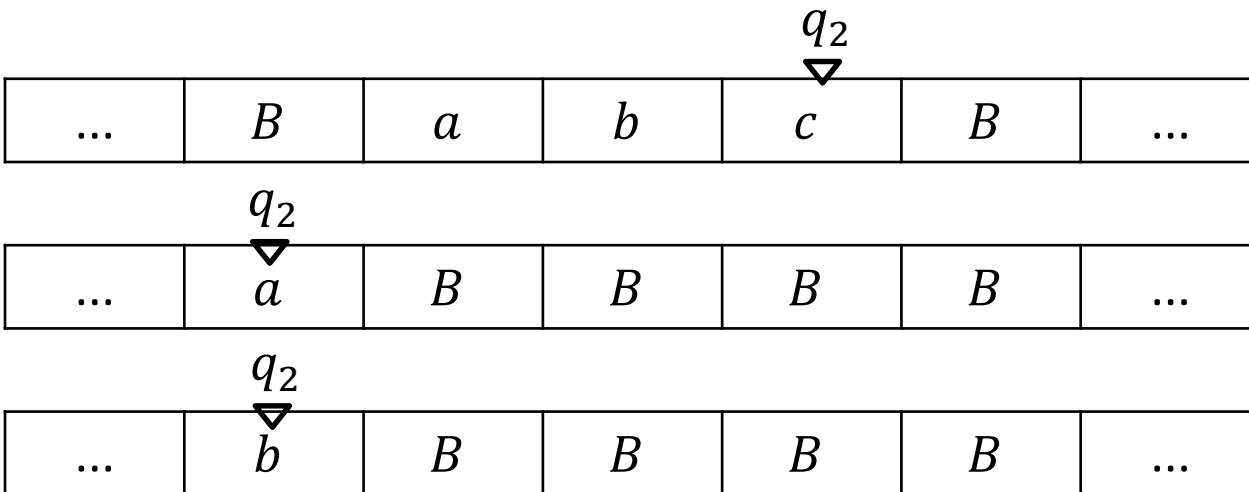
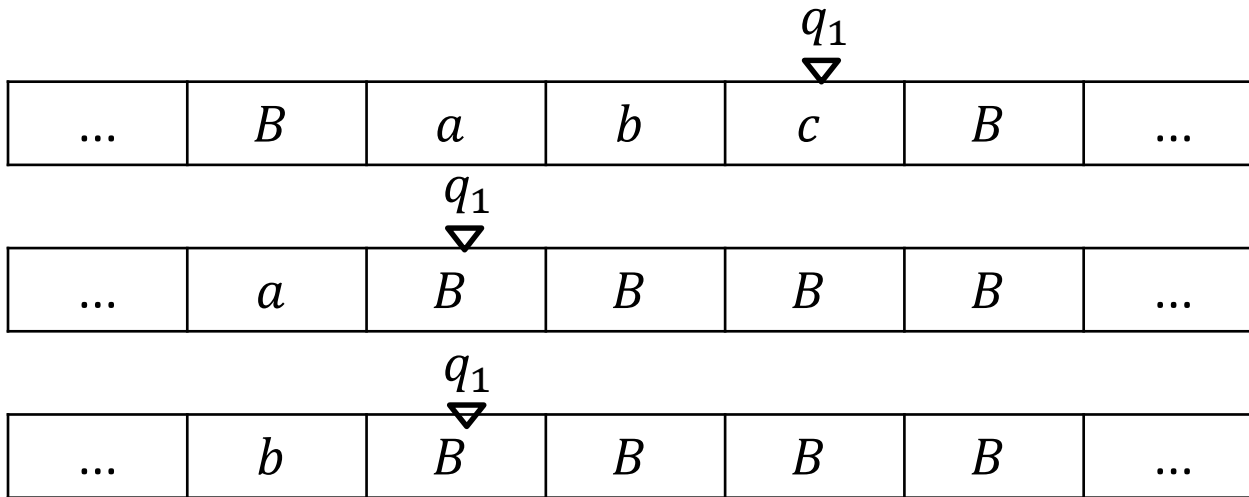
$(a, B, B) \rightarrow (a, a, B), (R, R, S)$

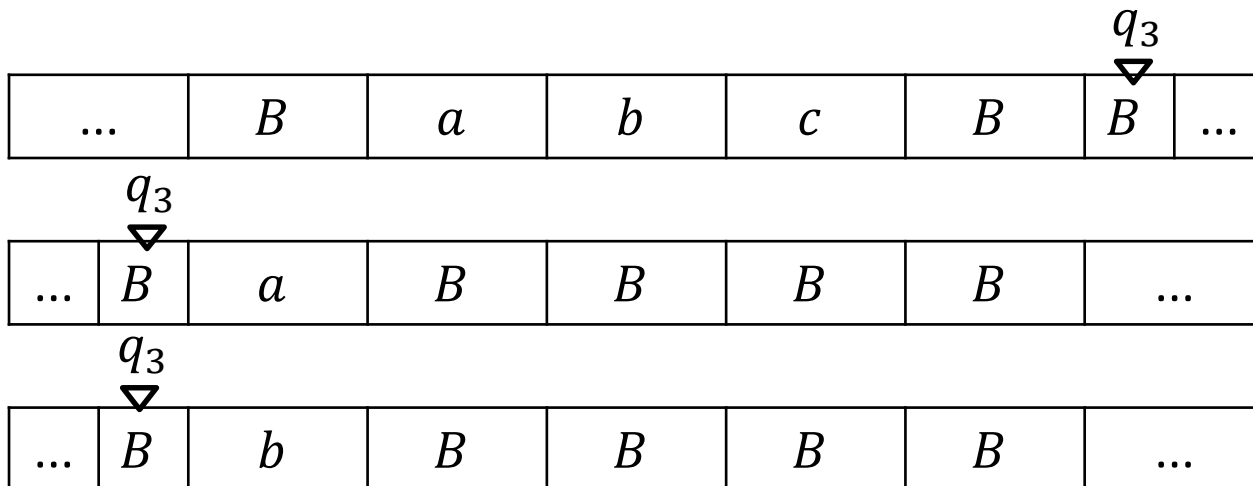
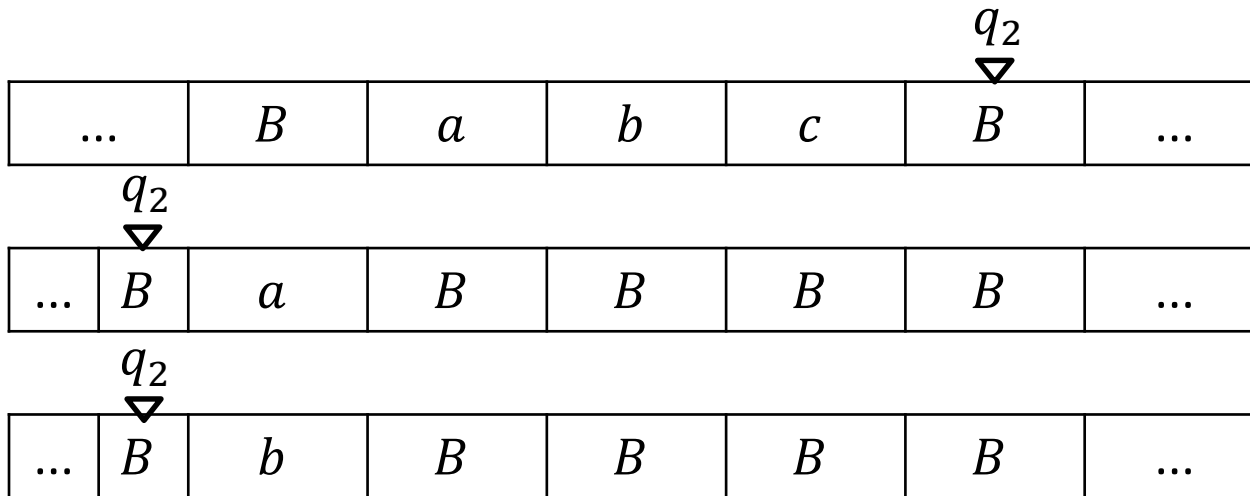


# Çok Bantlı Turing Makinesi

ör.  $w = abc$  kelimesini TM ye yazarak okuyalım.







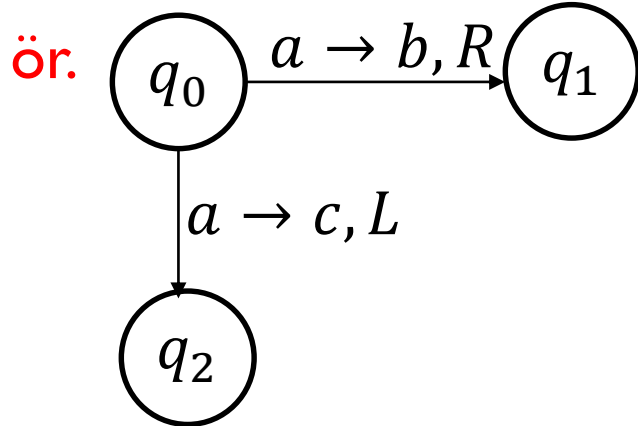


## 4. Nondeterministik Turing Makinesi (NTM)

NTM'de , standart bir TM'den farklı olarak bir durumda iken banttın bir harf okuduğunda, hangi duruma gideceğimiz, hangi harfi yazacağız ve hangi yöne gideceğimiz ile ilgili birden çok seçenek vardır. Bu seçeneklerden biri seçilir.

Geçiş fonksiyonu aşağıdaki gibidir:

$$\delta: Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L, R\})$$



$$\delta(q_0, a) \rightarrow \{(q_1, b, R), (q_2, c, L)\}$$

$q_0$ 'da iken banttın  $a$  okunduğunda bu iki durumdan biri olur: Yani ya  $b$  yazılıp  $q_1$  'e gidilir yada  $c$  yazılıp  $q_2$  'e gidilir.