

Otomata Teorisi (BİL 2114)

Fırat İsmailoğlu

Hafta 10: Turing Makinesi (Bölüm 3)

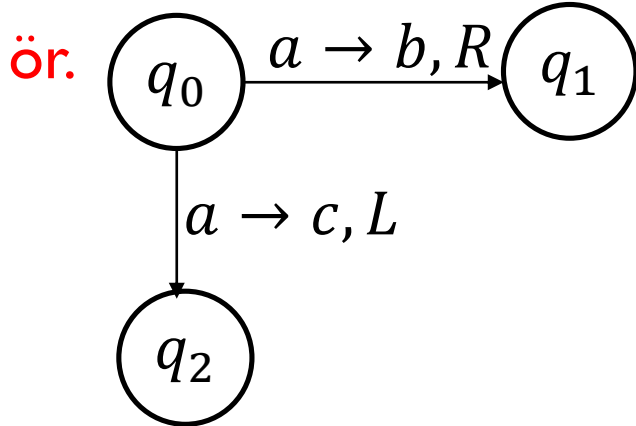


Nondeterministik Turing Makinesi (NTM)

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

Geçiş fonksiyonu aşağıdaki gibidir:

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

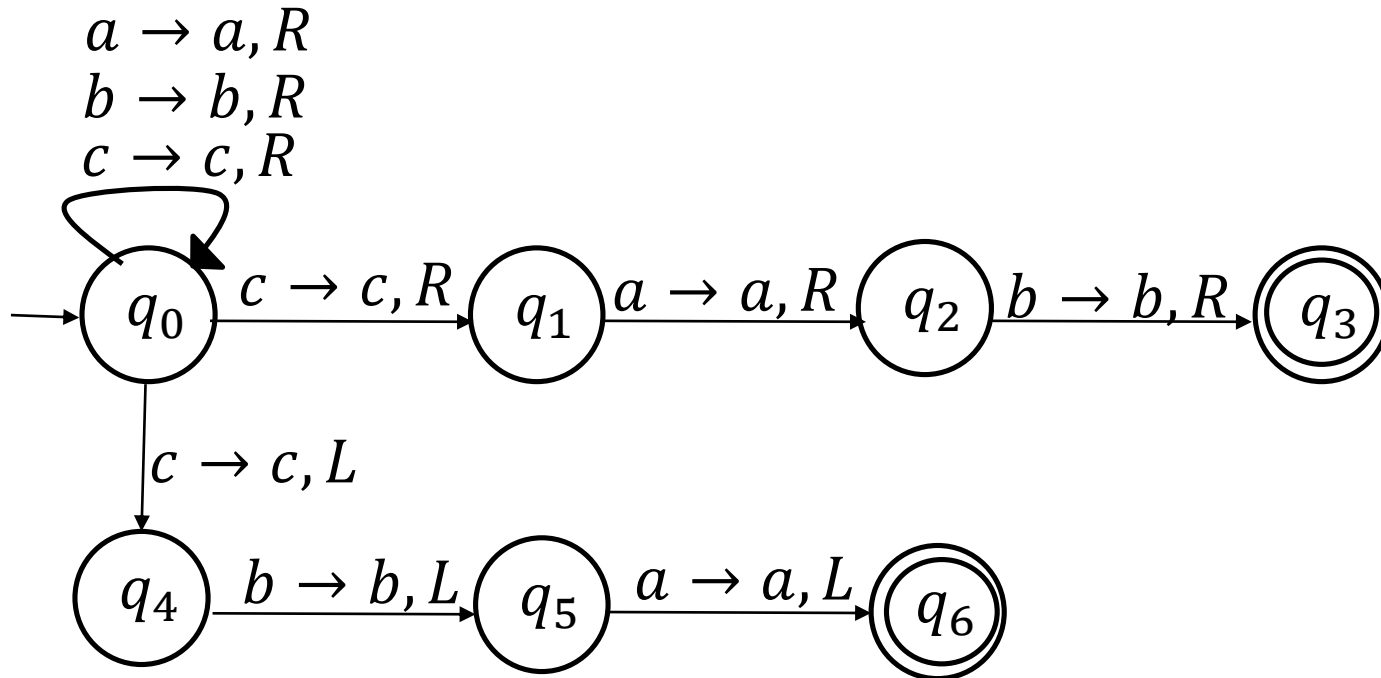


$$\delta(q_0, a) \rightarrow \{(q_1, b, R), (q_2, c, L)\}$$

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.

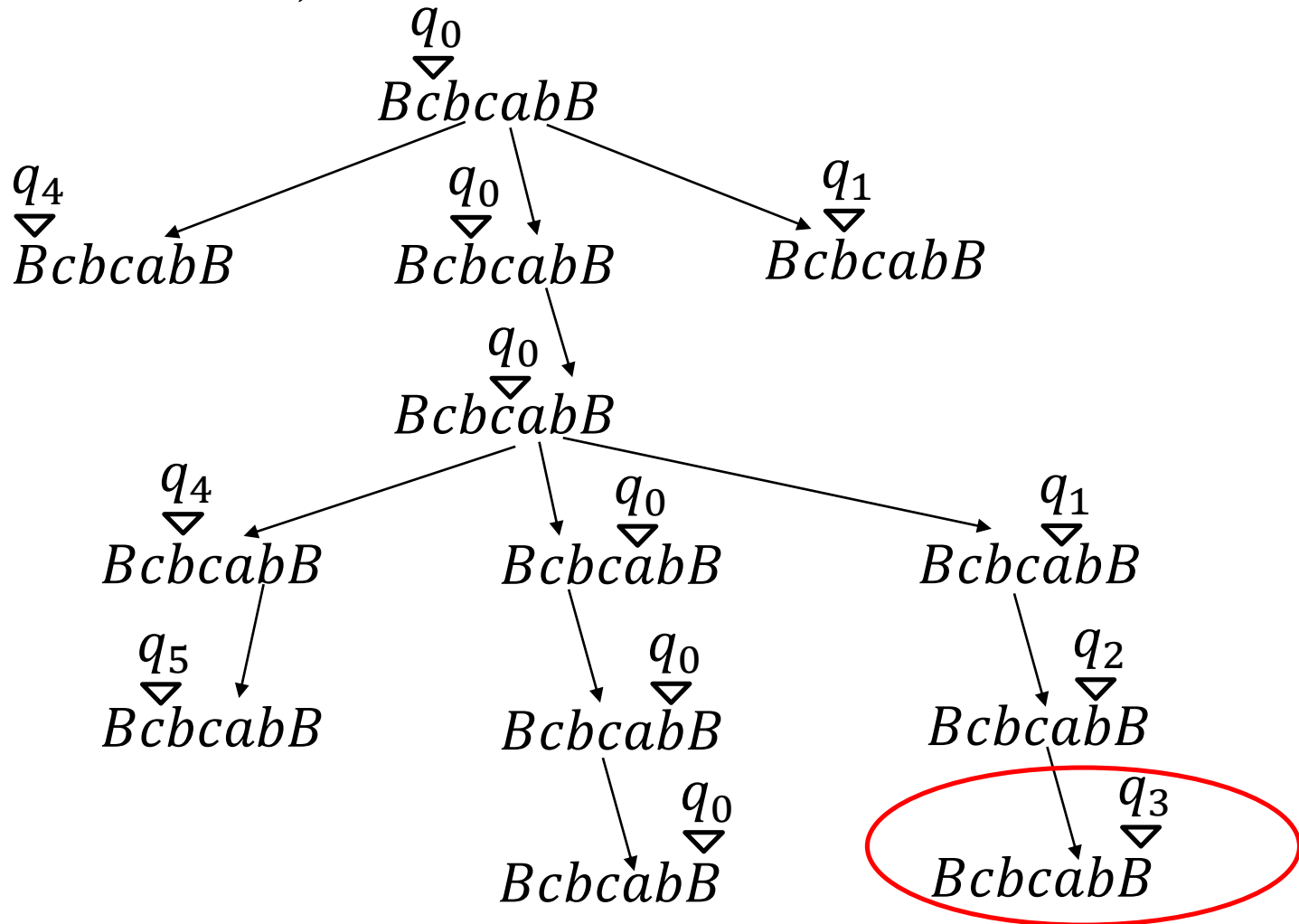
Nondeterministik Turing Makinesi (NTM)

ör. Aşağıdaki NTM $\Sigma = \{a, b, c\}$ harfleri kullanılarak üretilen kelimelerden sonu 'cab' yada 'abc' ile biten kelimeleri tanır.



Nondeterministik Turing Makinesi (NTM)

Örnek olarak $w = cbcab$ kelimesini banta yazıp okuyalım. Bu kelimenin NTM de işlenmesini ağaç diyagramı ile göstereceğiz.



Halting (Durma) Problemi

Verilen bir programın (C , Java, yada bir Turing Makinesi) her zaman sonlanıp sonlanmayacağını, yani sonsuz donguye girip girmeyeceğinin (önceden) bilinmesi problemine “Halting Problemi” denir.

Halting Problemi karar verilemez bir problemdir. Yani öyle bir program yazamayızki (öyle bir TM oluşturmamızki) girdi olarak verilen her program için bunun her zaman sonlanıp sonlanmayacağını bize soylesin.

Öyle bir program yazamayızki verilen bir programın içindeki tüm hataları bulsun. Hata yoksa program sonlanır desin.



Evrensel Turing Makinesi (Universal Turing Machine)

TM'nin bir cesit bilgisayar oldugunu soylemistik. Fakat su ana kadar gordugumuz TM'ler yalnızca bir gorev yapıyordu (ornegin, sadece cikarma yada toplama..) .

Gunumuz bilgisayarları ise cok amaclidir, birden fazla gorev yapabiliirler. Bu bilgisayarlar genel olarak bir program ve bu programin bir girdisi verildiginde verilen programi verilen girdi icin calistirirlar.

Evrensel Turing Makinesi (ETM) gunumuz cok amacli bilgisayarları gibidir. Bir TM ve bu TM'nin girdilerini girdi olarak alır ve TM'yi verilen girdiler uzerinde calistirarak sonuc uretir.

Yani ETM nin girdisi yine bir TM ve bu TM'nin kendi girdileridir.



Evrensel Turing Makinesi (Universal Turing Machine)

Bu düşünce günümüzde de her zaman kullanılır. Örneğin derleyici (compiler) in kendisi de bir programdır, yazılmış programları girdi olarak alır.

ETM'yi inşa etmek için girdi olarak verilen TM'yi banta yazabilmemiz gerekir. Bunun için tek yapmamız gereken TM'nin geçişlerini ikili (binary) olarak yazmaktır. Bunun genel formulu şöyledir:

$$\delta(q_i, X_j) = (q_k, X_l, D_m) \equiv 0^i 10^j 10^k 10^l 10^m$$

Burada $q_1, q_2, q_3 \dots$ durumlardır. Özel olarak q_1 her zaman başlangıç ve q_2 final durumudur.

$X_1, X_2, X_3, X_4 \dots$ bantın harfleridir. Özel olarak X_1 her zaman 0, X_2 her zaman 1 ve X_3 her zaman B dir. Böylece 0'ı 0 ile, 1'i 00 ile B 'yi 000 ile göstereceğiz.



Evrensel Turing Makinesi (Universal Turing Machine)

D_m , solu yada sağı gösterir. Sol için 0, sağ (R) için 00 a esittir.

Ayrıca geçişler birbirinden 11 kullanarak ayrılır:

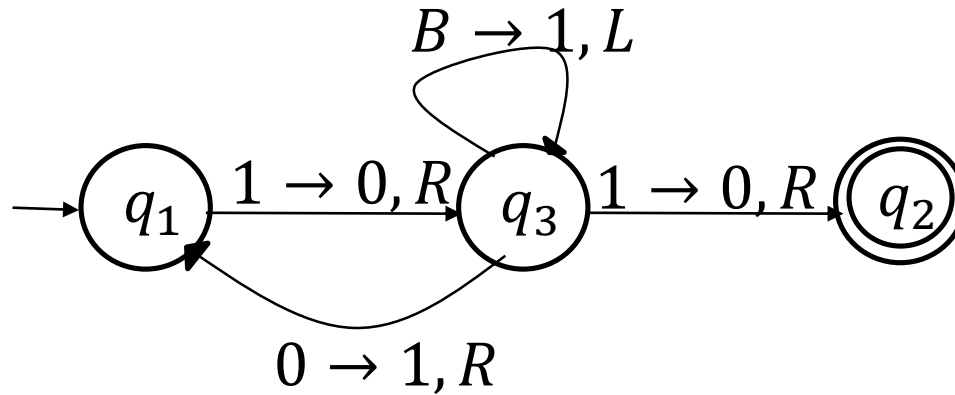
C_i 'ler bir TM'nin geçişleri olmak üzere, n tane geçişe sahip bir TM, ETM'nin bantına şu şekilde yazılır:

$$C_1 11 C_2 11 \dots C_{n-1} 11 C_n$$

ETM'nin bantına TM ile beraber bu TM'nin okuyacağı kelime yazılır. (Bu kelime eğer ikili (0-1) olarak verilmemişse ikili hale dönüştürülür.) ETM'nin bantında TM ve bunun okuyacağı kelime 111 kullanılarak ayrılır.



ör.



Yukarıdaki gibi bir TM ve $w = 1011$ kelimesi verilsin. Bu TM'yi w üzerinde çalıştıracak bir ETM oluşturalım.

Bu TM'nin gecisleri

$$\delta(q_1, 1) = (q_3, 0, R) \equiv 0100100010100$$

$$\delta(q_3, 0) = (q_1, 1, R) \equiv 0001010100100$$

$$\delta(q_3, 1) = (q_2, 0, R) \equiv 00010010010100$$

$$\delta(q_1, B) = (q_3, 1, L) \equiv 01000100010010$$

ETM'nin bantı:

B	B	0	1	0	0	1	0	0	0	1	0	0	0	1	1	.	.	.	0	0	0	1	0	0	0	1	0	0	1	0	1	1	1	1	0	1	1	B	B
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

