



NECMETTİN ERBAKAN
ÜNİVERSİTESİ
KONYA



SONLU OTOMATA (FINITE AUTOTAMA)

Otomata Teorisi
Necmettin Erbakan Üniversitesi
Bilgisayar Mühendisliği Bölümü

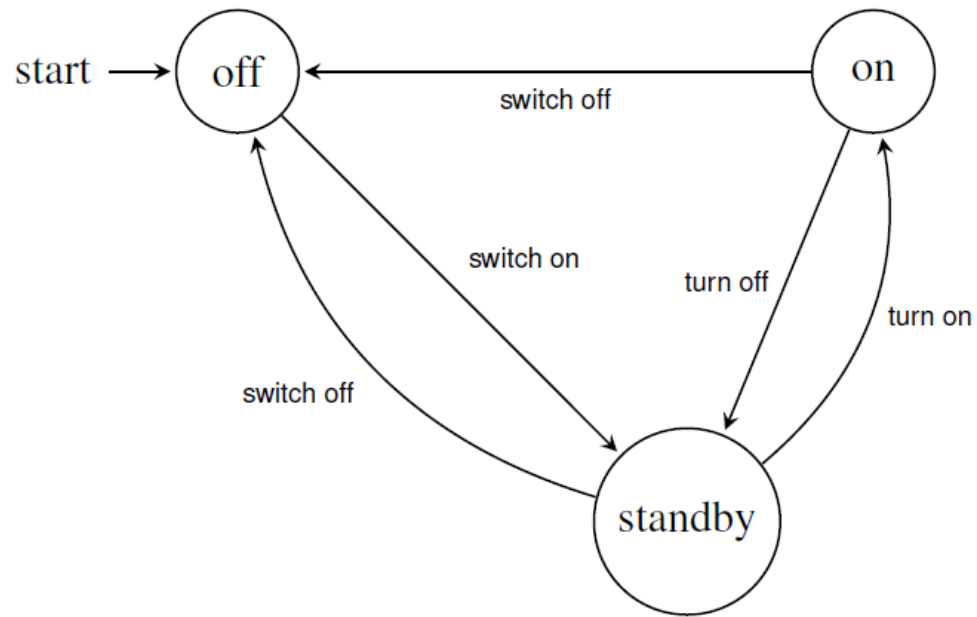


Sonlu Otomata

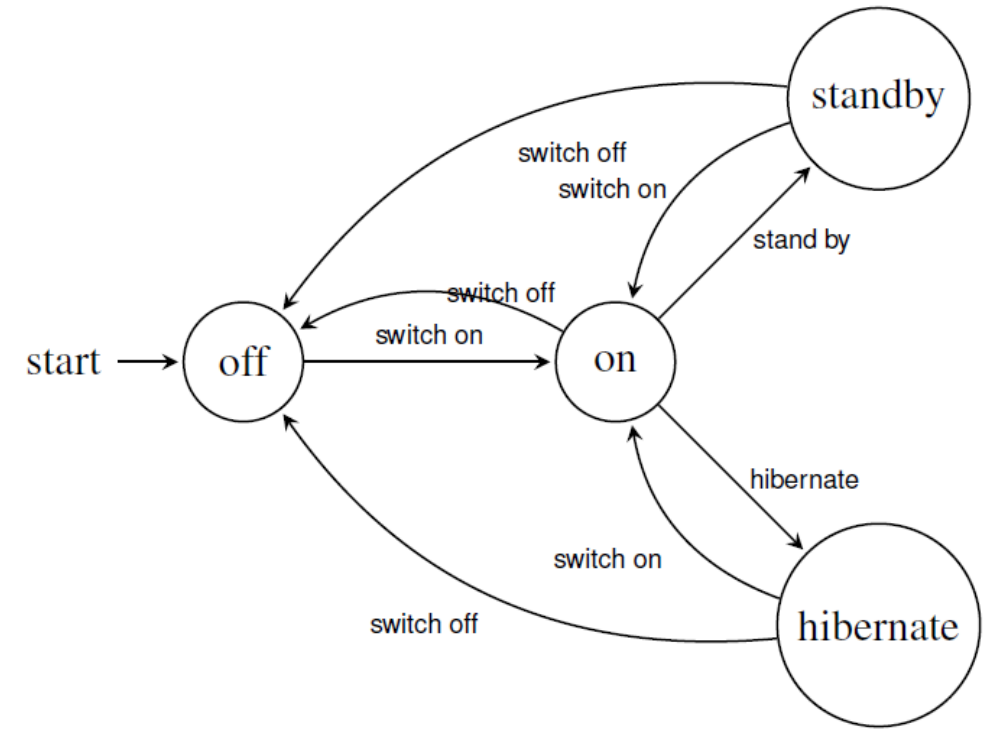
- Sonlu durum kümesinden
- Sonlu giriş kümesinden
- Geçiş Fonksiyonundan
- Özel bir başlangıç durumundan oluşur.

Sonlu Otomata Örnekler

TV

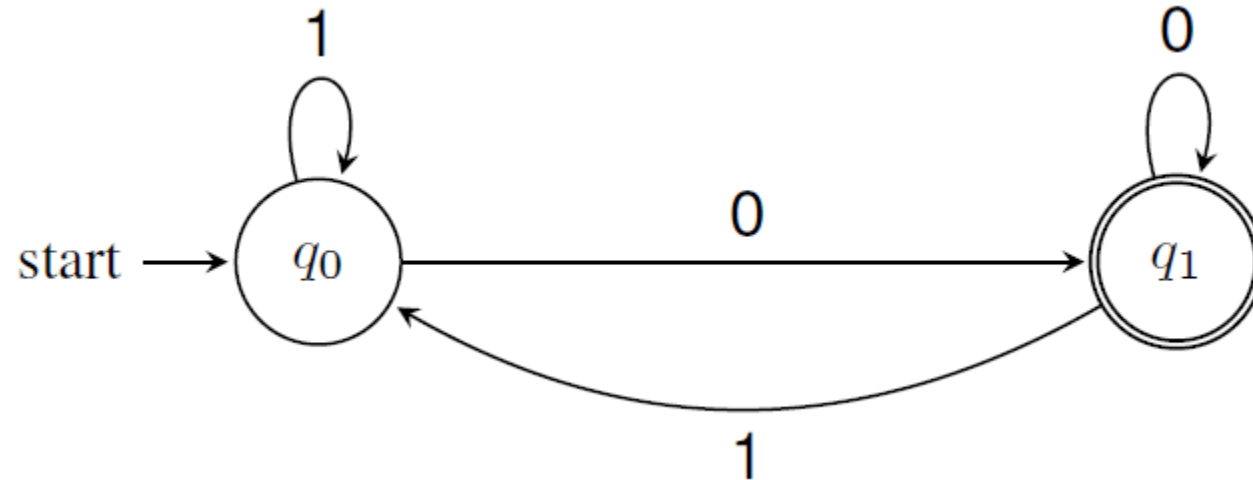


PC



Sonlu Otomat

- Sonlu otomat kabul durumu olan bir sonlu durum makinesidir.
- Örnek: M sonlu otomatı



- Giriş metni $w=0010$ M tarafından kabul edilir. Ayrıca 0, 01, 110, 010, ... gibi sonu 0 olan bütün metinler.

Belirli Sonlu Otomat (BSO) (Deterministic Finite Automat, DFA)

- Sabit ve sonlu kapasiteye sahip merkezi işlem ünitesidir.
- Giriş bilgisini string olarak bir input tape üzerinden alır.
- Bir çıkış üretmez, giriş bilgisinin kabul edilip edilmediğini gösterir.
- Dil tanıyıcı cihaz (Language Recognition Device) olarak işlem yapar.
- Derleyicide lexical analysis (begin, for ... vb.) aşamasında kullanılır.
- Protokol tanımlarında kullanılır.

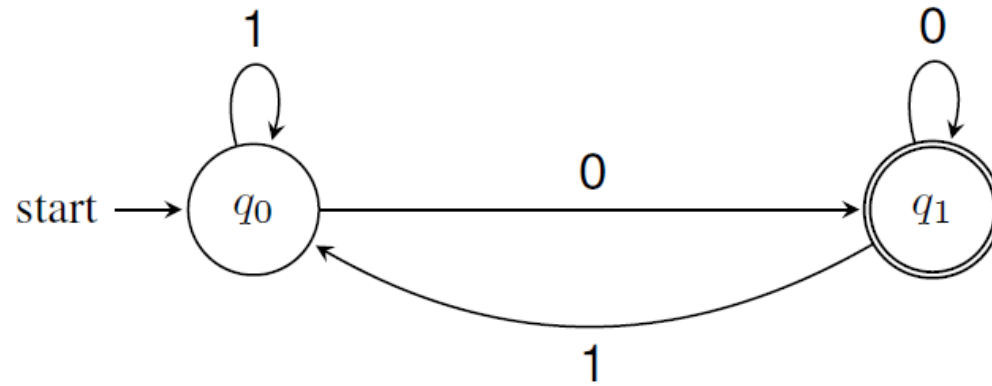


Belirli Sonlu Otomat (BSO) (Deterministic Finite Automat, DFA)

- Belirli sonlu otomat $M=(Q; \Sigma; \delta; q_0; F)$ şeklinde 5 ifade ile gösterilir.
- Q sonlu durumlar kümesi.
- Σ sonlu alfabe.
- $\delta : Q \times \Sigma \rightarrow Q$ geçiş fonksiyonu.
- $q_0 \in Q$ başlangıç durumu.
- $F \subseteq Q$ kabul edilen durumlar kümesidir.

Belirli Sonlu Otomatların Gösterimleri

Geçiş Diyagramı (Transition Diagram)

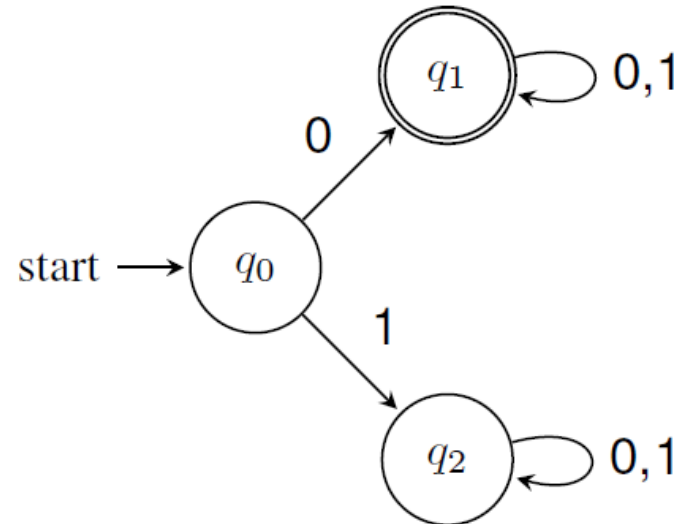


Geçiş Tablosu (Transition Table)

Q/Σ	0	1
$\rightarrow q_0$	q_1	q_0
$*q_1$	q_1	q_0

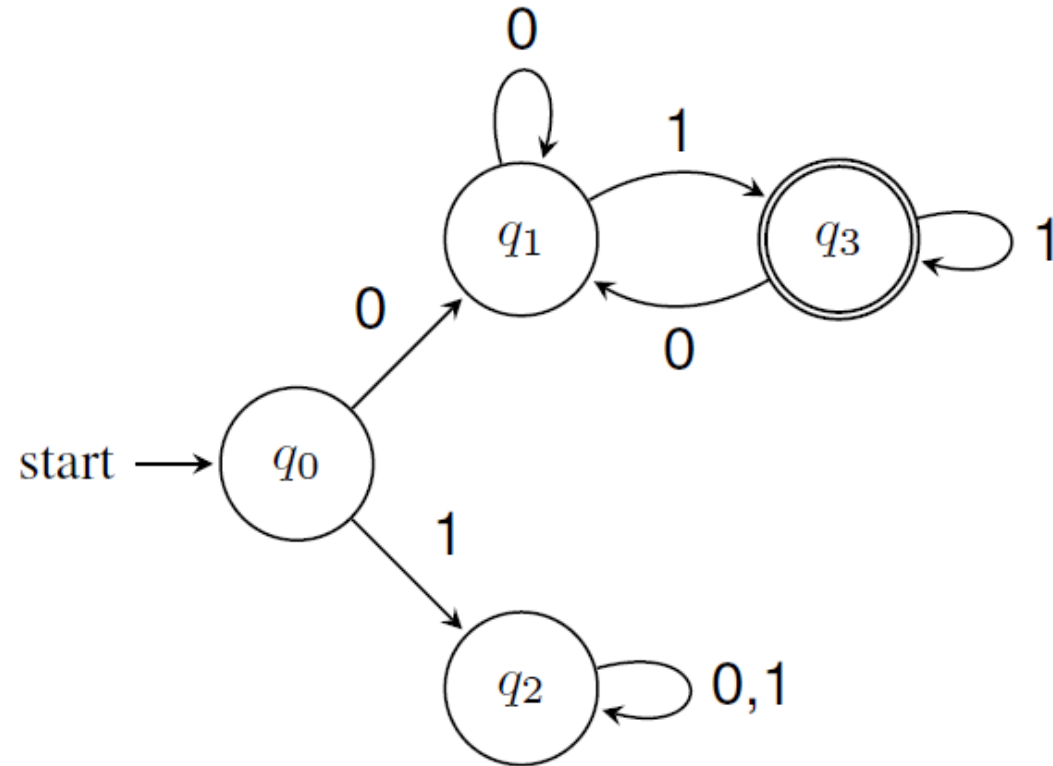
Belirli Sonlu Otomat

- Tanım: Eğer bir giriş stringinin hesaplaması M belirli sonlu otomatın kabul edilen herhangi durumunda biterse o string kabul edilir.
- Tanım: Bir M BSO'nin dili, $L(M)$ şeklinde gösterilir ve M tarafından kabul edilen bütün stringleri ifade eder.
- Örnek: $L : \{w \mid w \in \{0,1\}^* \text{ ve } w \text{ 0 ile başlayacak}\}$



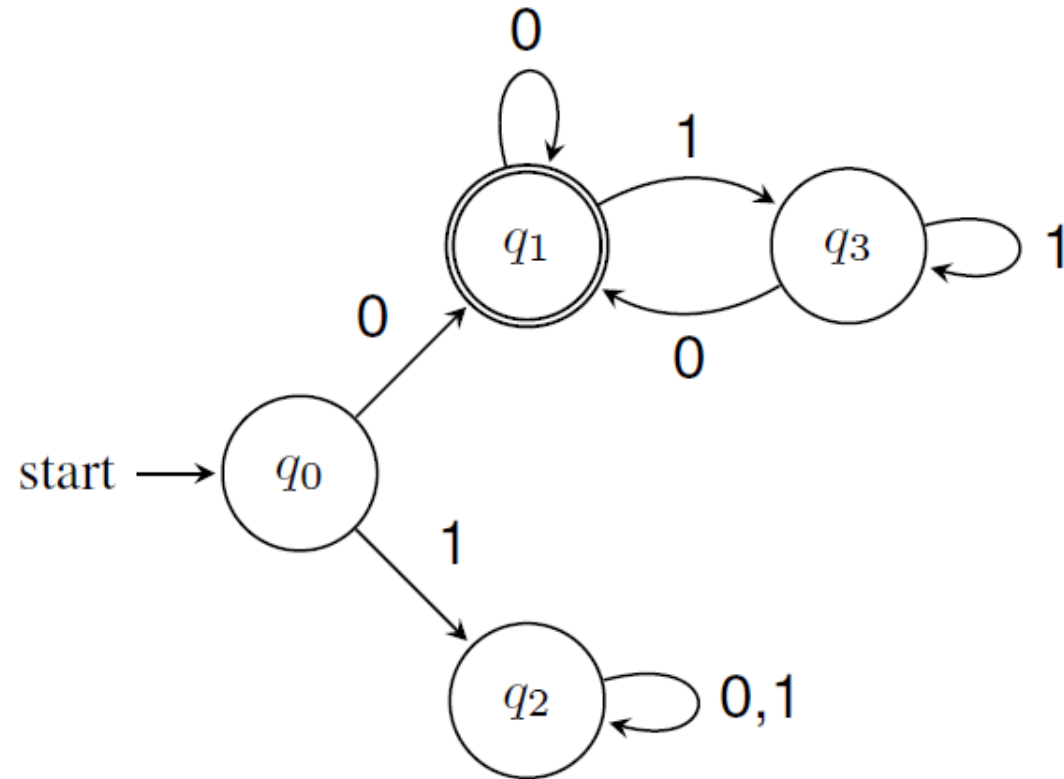
Belirli Sonlu Otomat

- Örnek: $L : \{w \mid w \in \{0,1\}^* \text{ ve } 0 \text{ ile başlayan ve } 1 \text{ ile biten}\}$



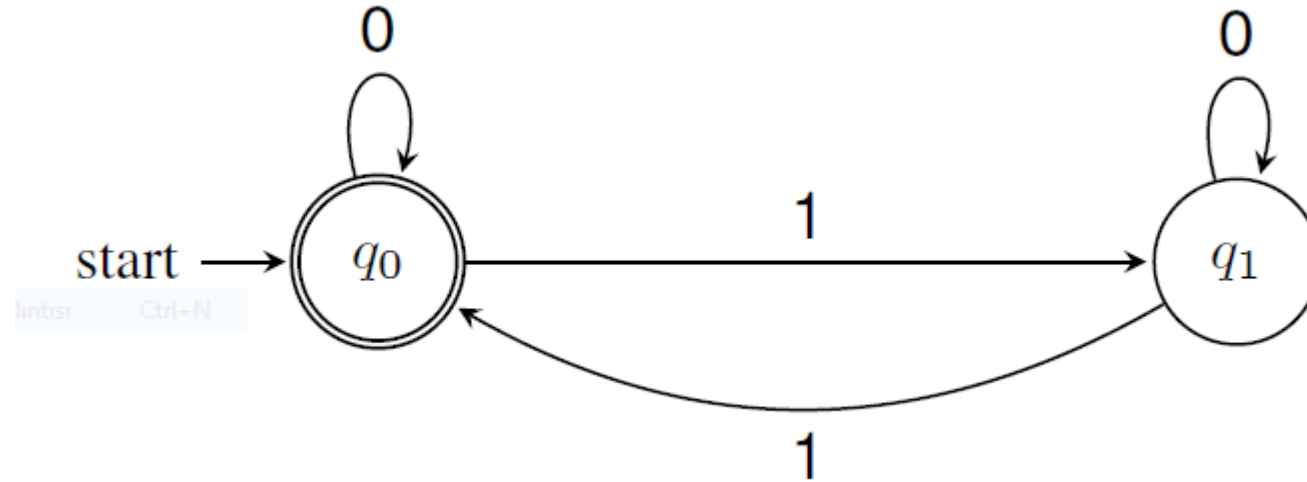
Belirli Sonlu Otomat

- Örnek: $L : \{w \mid w \in \{0,1\}^* \text{ ve } 0 \text{ ile başlayan ve } 0 \text{ ile biten}\}$



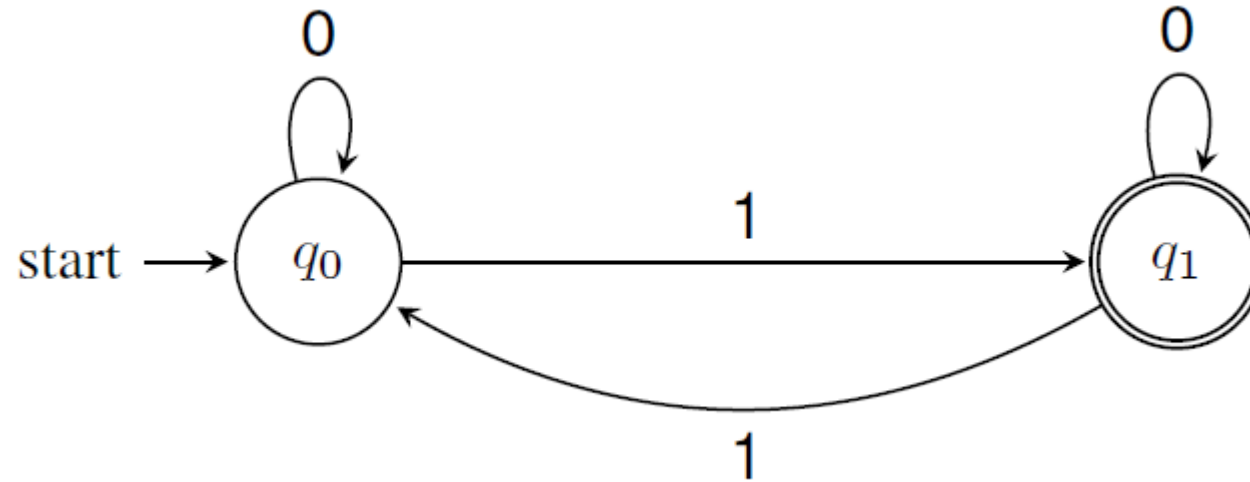
Belirli Sonlu Otomat

- Örnek: $L : \{w \mid w \in \{0,1\}^* \text{ ve çift sayıda } 1 \text{'den oluşacak}\}$



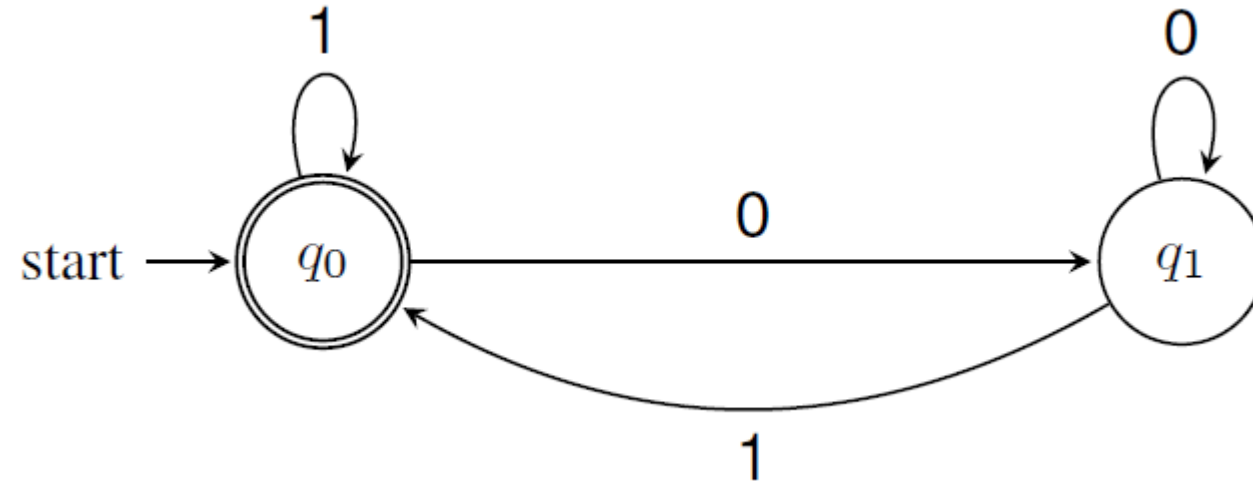
Belirli Sonlu Otomat

- Örnek: $L : \{w \mid w \in \{0,1\}^* \text{ ve tek sayıda } 1 \text{ 'den oluşacak}\}$



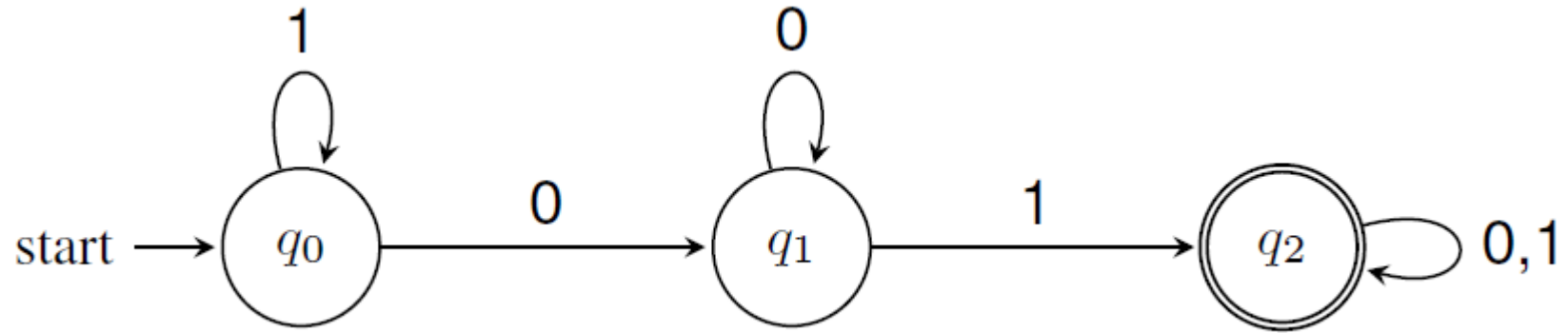
Belirli Sonlu Otomat

- Örnek: $L : \{w \mid w \in \{0,1\}^* \text{ ve sonu } 1 \text{ ile bitecek}\}$



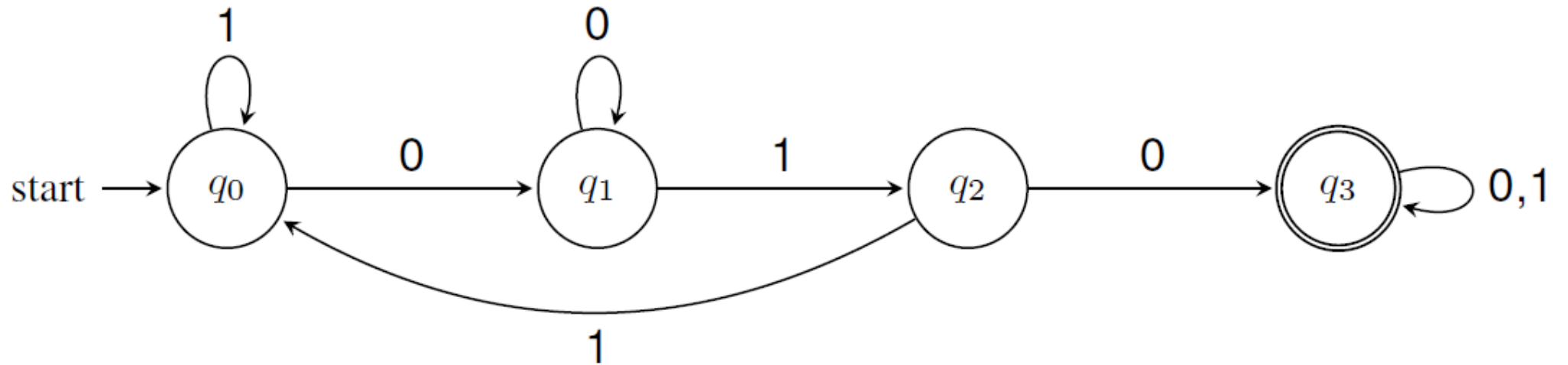
Belirli Sonlu Otomat

- Örnek: $L : \{w \mid w \in \{0,1\}^* \text{ ve içerisinde } 01 \text{ alt metnini içerecek}\}$



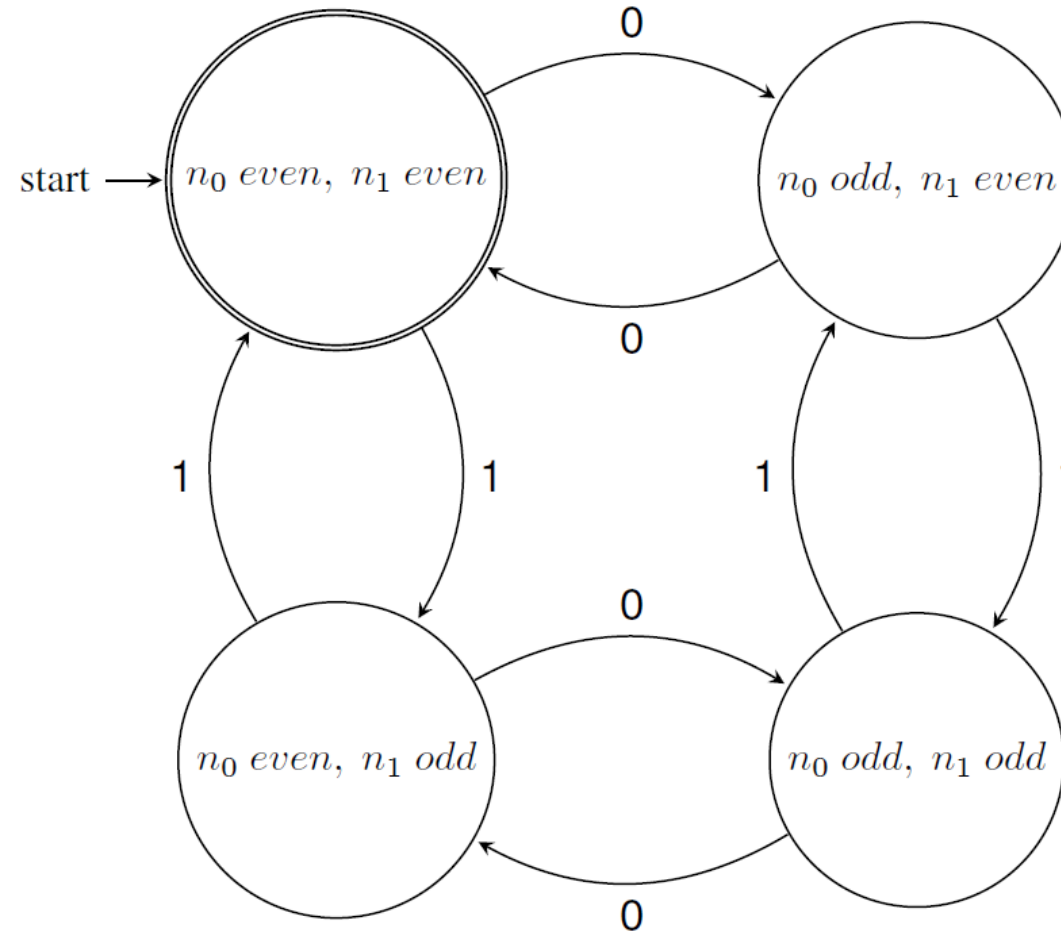
Belirli Sonlu Otomat

- Örnek: $L : \{w \mid w \in \{0,1\}^* \text{ ve içerisinde } 010 \text{ alt metnini içerecek}\}$



Belirli Sonlu Otomat

- Örnek: $L : \{w \mid w \in \{0,1\}^* \text{ ve } 0\text{'ların ve } 1\text{'lerin sayısı çift olacak}\}$

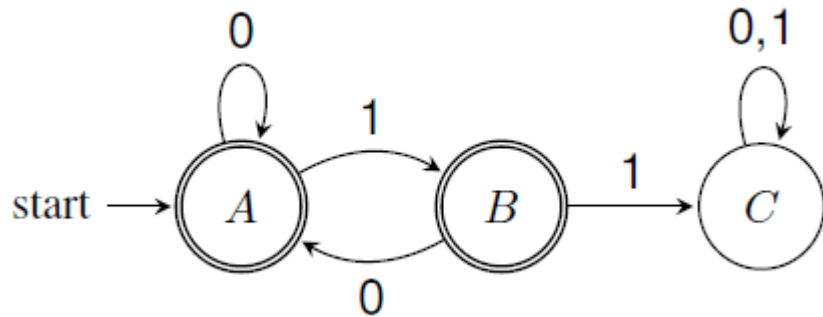




Ödev

- $L : \{w \mid w \in \{0,1\}^* \text{ ve } 0\text{'ların sayısı} = 1\text{'lerin sayısı mod } 2\}$ olan BSO tasarlayınız.

BSO'un Uygulanması

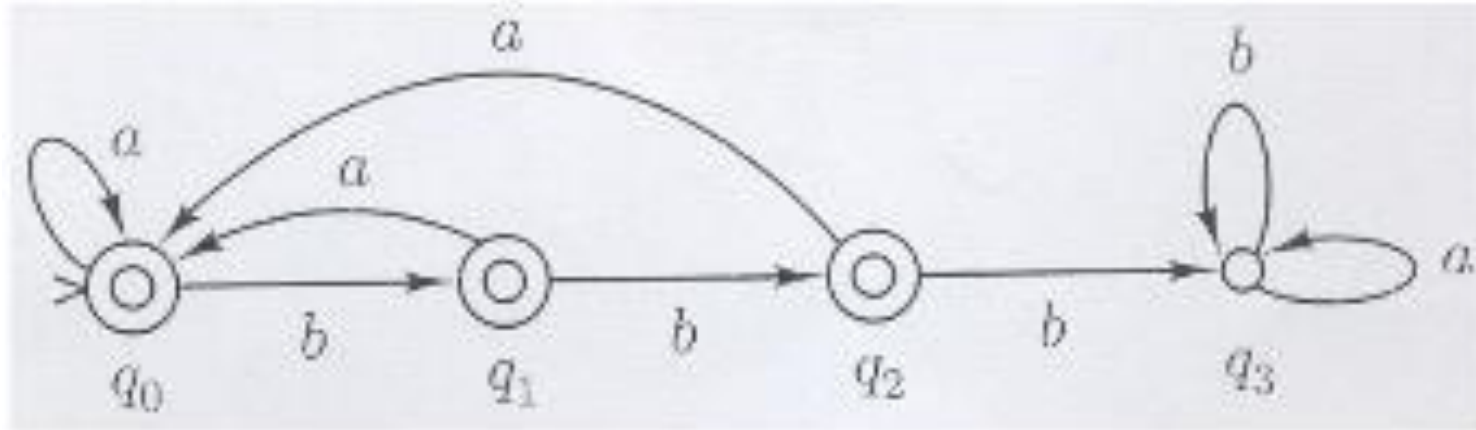


```

enum STATES {A, B, C} state;
int i;
char w[]; // from the user
state = A; // A is start state
for (i=0; i<strlen(w); i++){
    switch (state){
        case A:
            if (w[i]=='0') state=A;
            else state=B;
            break;
        case B:
            if (w[i]=='0') state=A;
            else state=C;
            break;
        case C:
            state=C;
            break;
    }
}
if (state == A || state == B){
    printf("accepted.\n");
}
else{
    printf("rejected.\n");
}
  
```

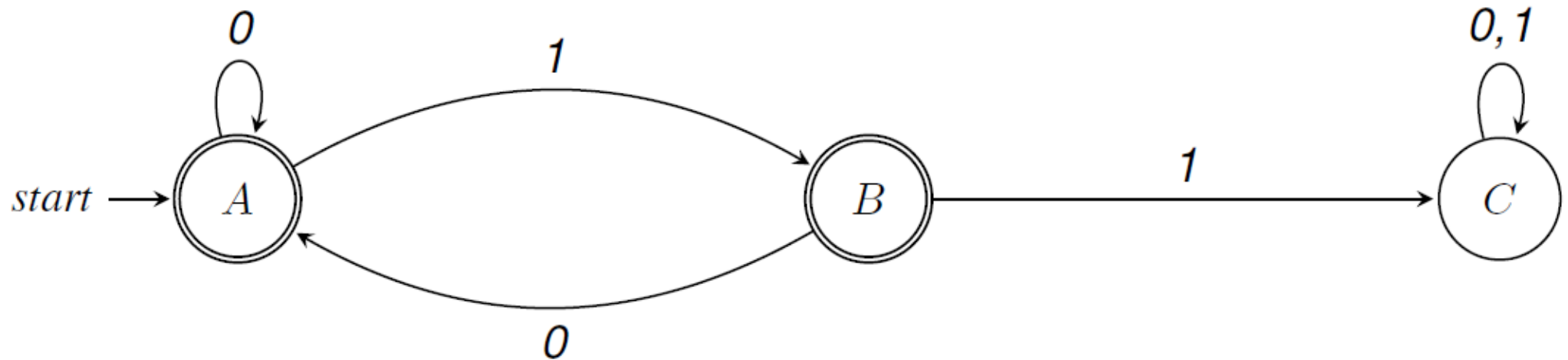

Ödev

- $L : \{w \mid w \in \{0,1\}^* \text{ ve } w \text{ 23'e tam bölünebilen binary tam sayıdır}\}$
- Yukarıda tanımlanan düzenli dil için BSO tanımlayınız ve bir programlama dilinde kodlayınız.
- Aşağıdaki otomatın tanıdığı dili regular expression ile ifade ediniz.



Düzenli Dil

- Tanım: Eğer bir dil bir BSO tarafından kabul ediliyorsa **düzenli dildir** (Regular Language).
- Örnek: $L_1 : \{w \mid w \in \{0,1\}^* \text{ ve arka arkaya 2 tane 1 içermeyecek} \}$



Düzenli Dil İspat

- Bu ispat tümevarım yöntemi ile yapılacaktır. Bunun için tekrar eden geçişler $\delta^*(w)$ şeklinde yani w okunduktan sonra M 'in yeni durumu şeklinde tanımlanacaktır.
- Böylelikle bütün $w \in \Sigma^*$ ve $x \in \Sigma$ için $\delta^*(wx) = \delta(\delta^*(w), x)$ şeklinde tanımlanabilir. Ayrıca $L(M) = \{w \mid \delta^*(w) \in F\}$ aşağıdaki kurallara göre ispat edilebilir.
- Kural A: If w 1 içermez veya iki tane 0 sıfır arasında 1 içerir.
- Kural B: If w arka arkaya olmayan 1'ler içerir.
- Kural C: If w arka arkaya 2 tane 1 içerir.

Düzenli Dil İspat

- Basis: If $w=\varepsilon$ then $\delta^*(w):=A$
- $w=ux$ olduğu varsayılırsa varsayım u ve x alt stringleri içinde geçerli olacaktır. Bunun için 3 duruma dikkat edilecektir:
 - $\delta^*(u):=A$
 - $\delta^*(w)=\delta(A,x)=\{\text{if } x=0 \rightarrow A, \text{ if } x=1 \rightarrow B$
 - İddia doğrudur: $u0 \in A$ ve $u1 \in B$
 - $\delta^*(u):=B$
 - $\delta^*(w)=\delta(B,x)=\{\text{if } x=0 \rightarrow A, \text{ if } x=1 \rightarrow C$
 - İddia doğrudur: $u0 \in A$ ve $u1 \in C$
 - $\delta^*(u):=C$
 - $\delta^*(w)=\delta(C,x)=\{\text{if } x=0 \rightarrow C, \text{ if } x=1 \rightarrow C$
 - İddia doğrudur: $u0 \in C$ ve $u1 \in C$