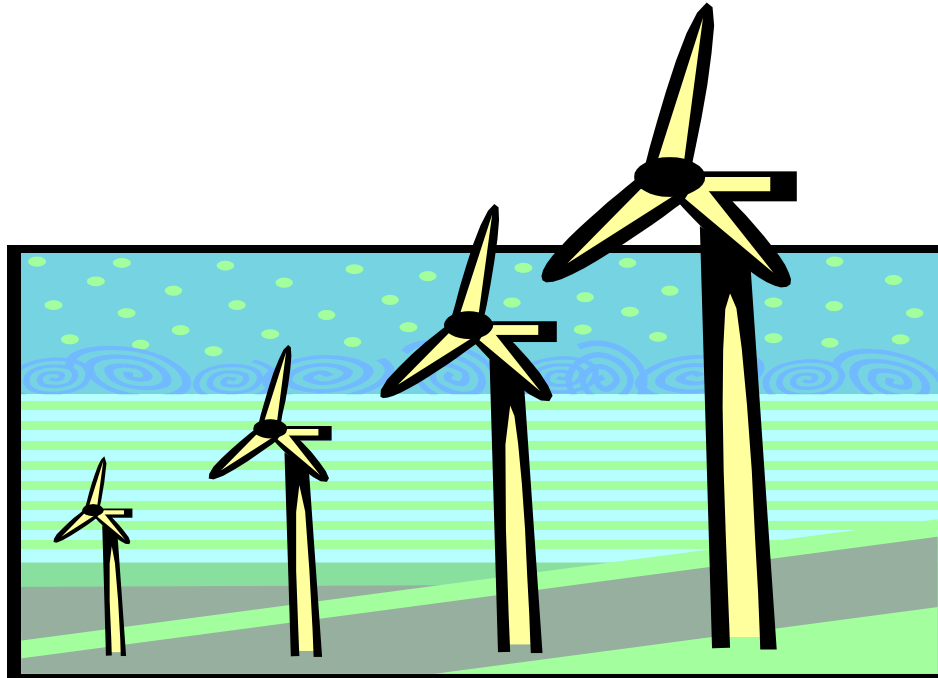


# **SONLU DURUM MAKİNELERİ (FINITE STATE MACHINES) (SDM-FSM)**



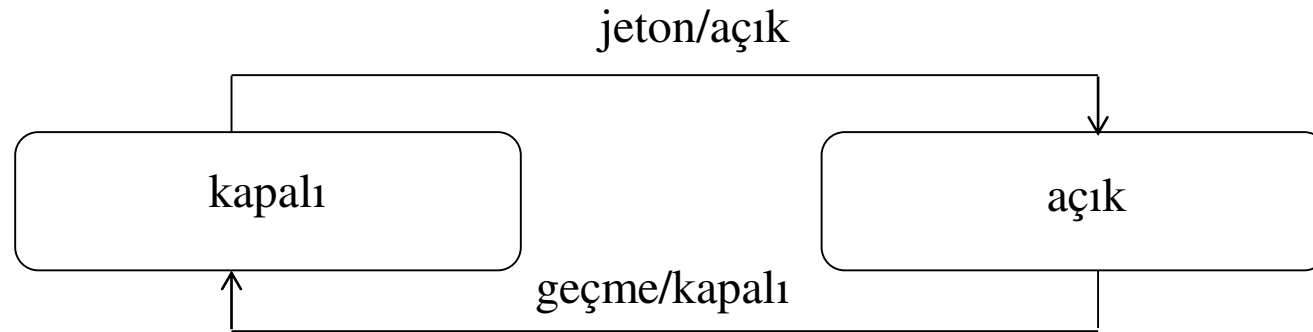
# Sonlu-durum Makineleri

---

Sonlu Durum Makinelerini anlatmak için verilecek en basit örnek, Metro İstasyonlarındaki turnikelerdir.

- Basit bir cihaz olan bu turnikeler aslında sonlu durum makinesi tarafından yönetilmektedir.
- Turnikenin açık ve kapalı olmak üzere sadece iki durumu mevcuttur.
- Kapalı olduğu durumda yolcu, turnike üzerindeki deliğe jetonunu atar ve mevcut durumun değişmesine neden olur.
- Böylece turnike artık açık durumuna geçmiştir.

- Turnikenin kapalıdan açık duruma geçmesi ve geçme işlemi tamamlandıktan sonra tekrar kapalı duruma geçmesi şekil’de gösterilmiştir.



- Geçiş yolu üzerindeki etiketler, iki parçadan oluşmaktadır. İlk parça, değişime neden olacak olayın kendisi, ikinci de olayın sonucudur.

---

Kısaca,

- Sonlu Durum Makinesi sonlu sayıda duruma (*state*) sahip, verilen girişi bir durumdan diğer bir duruma ileten ve çıkış üretebilen bir ağıdır.
- Kullanım alanları olarak konuşma tanıma ve herhangi bir dilin modellemesinde kullanılması verilebilir.

# Sonlu-durum makineleri (Finite-state machines)

Bir sonlu-durum makinesi

$M = (I, O, S, f, g, \sigma)$  6-lısı'ndan oluşur.

a) **I**, giriş sembollerinin sonlu kümesi

b) **O**, çıkış sembollerinin sonlu kümesi

c) **S**, sonlu durumlar kümesi *→ statelerin kümesi*

d) **f**:  $S \times I \rightarrow S$  gelecek durumun fonksiyonu

e) **g**:  $S \times I \rightarrow O$  çıkış fonksiyonu

f)  $\sigma$  başlangıç durumu

*→ I ile giriş sembol-  
kümelerinin*

*→ çıkış  
fonksiyonu*  
*→ durum fonksi-  
yonu*

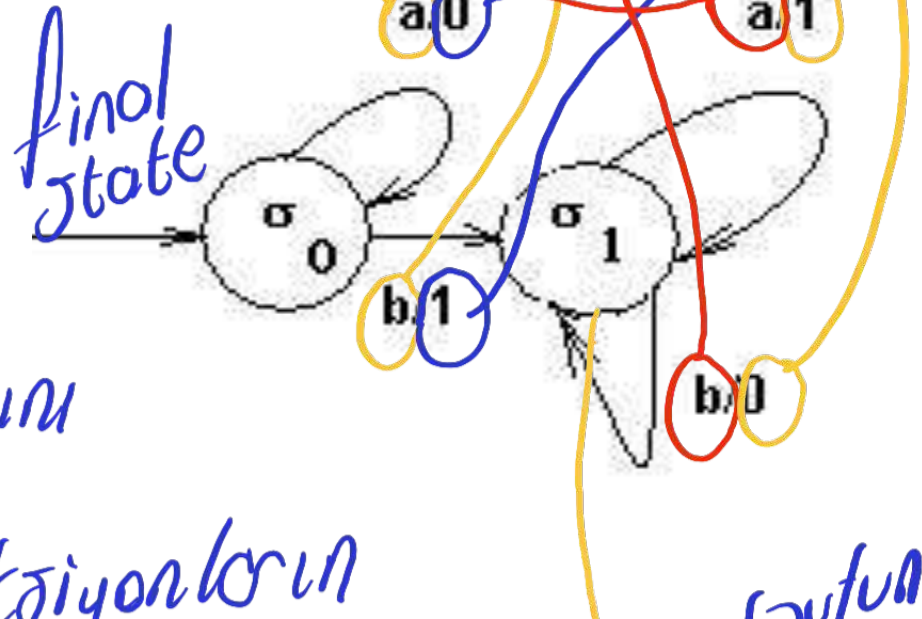
# Sonlu-durum makinelerine örnek

input

- $I = \{a, b\}$
- $O = \{0, 1\}$
- $S = \{\sigma_0, \sigma_1\}$
- Tabloda verilmiş

olan f ve g fonksiyon değerlerinin geçiş diyagramı (transition diagram)

Giriş sembolleri →		f input		g output	
Durumlar	$\sigma_0$	a $\sigma_0$	b $\sigma_1$	a 0	b 1
	$\sigma_1$	$\sigma_1$	$\sigma_1$	1	0

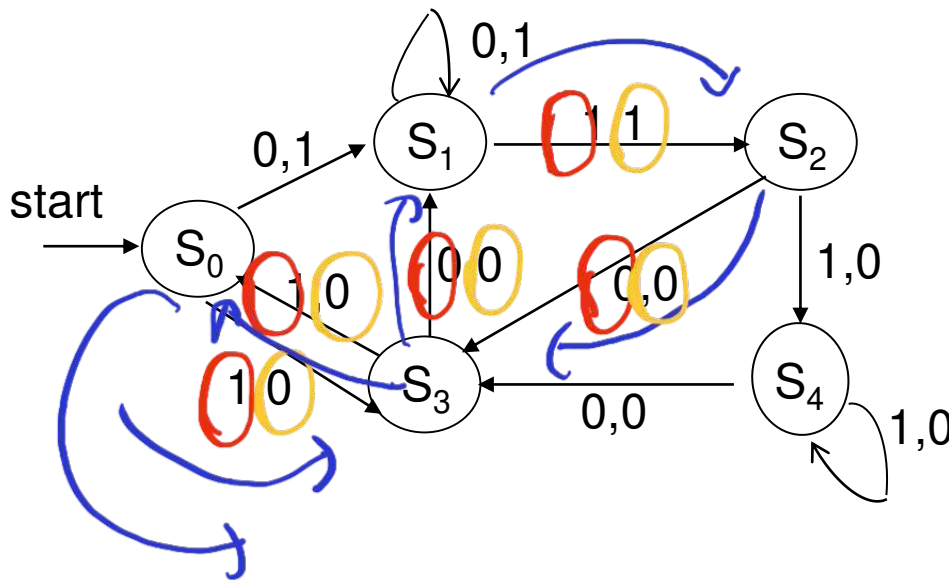


Benim state diyagramını  
çizebilmem için  
fonksiyonların

# Örnek

banana 7 ve 7 fonksiyonları verilmesi lazım  
hangi inputta hangi output oluşturulduğunu çıkarabil-  
mem için

Giriş değeri 101011 iken FSM/SDM makinesi çıkış olarak ne üretir?



Giriş : 101011

Çıkış: 001000

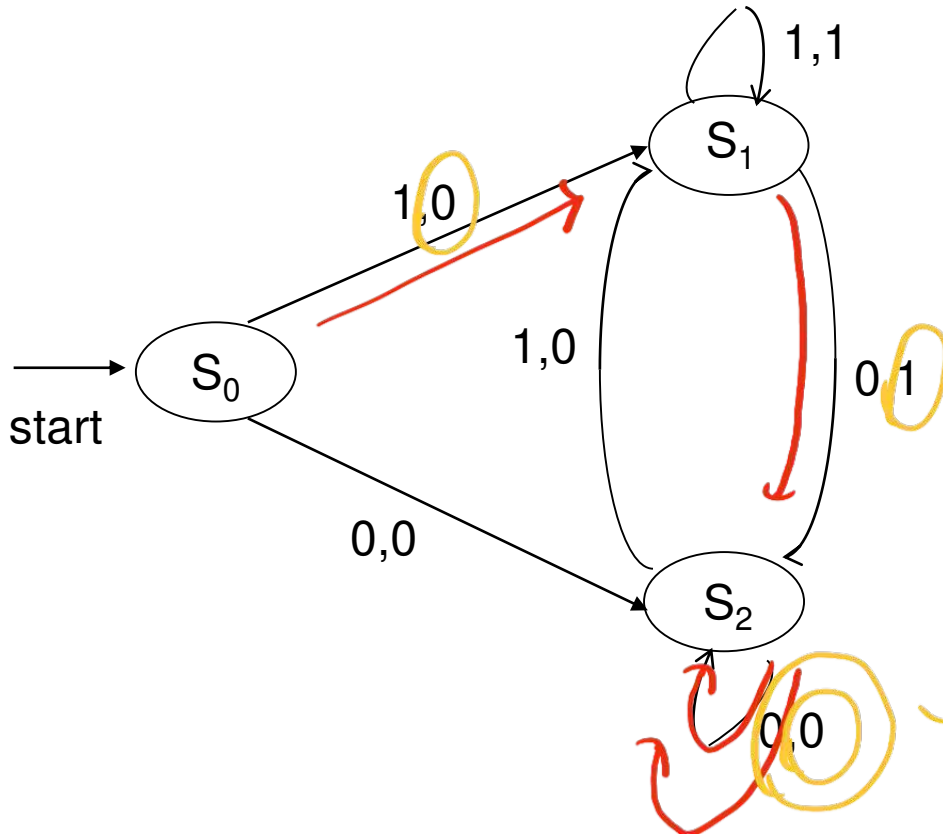
001000 → output

Burdaki  
gift  
halkının

B makinenin içinde hiçbir yerde tanımlama yoksa  
Örnek *input geldikçe makinenin içinde döner*

anlamı  
statemiz  
burda  
tanımlanıyor  
demek-  
bir

State diyagramı aşağıda verilmiş olan işlem neyi  
gerçekleştirmektedir?



eğer bunu  
giriş olarak  
verirsek bu  
çıkar 0010  
yani 2  
olarak çıkar

Sağa bir kez shift  
(öteleme)

(çıkış)

0100

4



Bizim direk burdan anlamamıza zor o yüzden kofadan bir  
giriş vericek mesela

Örnek

1000 → binary(8)

Çıktı içeren sonlu durum makinelerine verilen en genel örnek ikili sayı tabanında, iki sayıyı toplayan SDM/FSM'dir. *ben toplama yapıyordum 2 tane statetimin var bir eldesiz toplama bir de eldenin oluşturu toplama*

- Toplama için kabul edilen çiftlerin giriş kümesi {00,01,10,11}, çıkış kümesi de {0,1} verilmiş olsun.

- Verilen x,y giriş değerlerine göre, iki durumdan biri oluşur

- \* x ile y değerinin toplanması

- \* x,y ve 1 değerlerinin toplanması

- Böylece elimizdeki iki durumdan biri eldesiz ( $S_0$ ) diğeri de eldeli ( $S_1$ ) toplama durumudur.

2 tane ikili tabandan

1. state elde olmadan bir şeyleri topuyorum  
2. state eldeyle bir şeyleri topluyordum

0	0
+ 0	+ 1

$$\begin{array}{r} ( )_2 \\ + \\ ( )_2 \\ \hline ( )_2 \end{array}$$

bir sayıyı toplayıp  
2 li tabanda bir  
sayı elde ediyoruz  
bu durumda

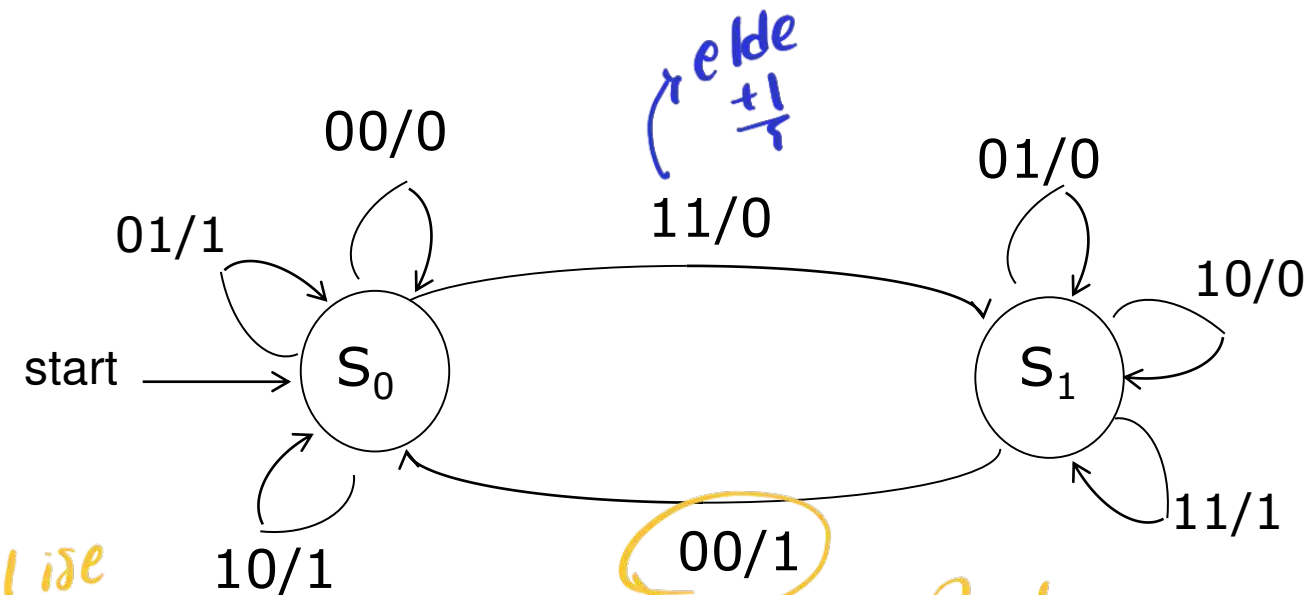
$$\begin{array}{r} 0 \\ + 0 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ + 1 \\ \hline 0 \end{array} \quad \text{carry}$$

4 tane durum  
var toplamada

- Başlangıç durumumuz ( $S_0$ ) ise,  
seri toplama yapan sonlu durum makinemiz nasıl olur?

elde yok

elde var



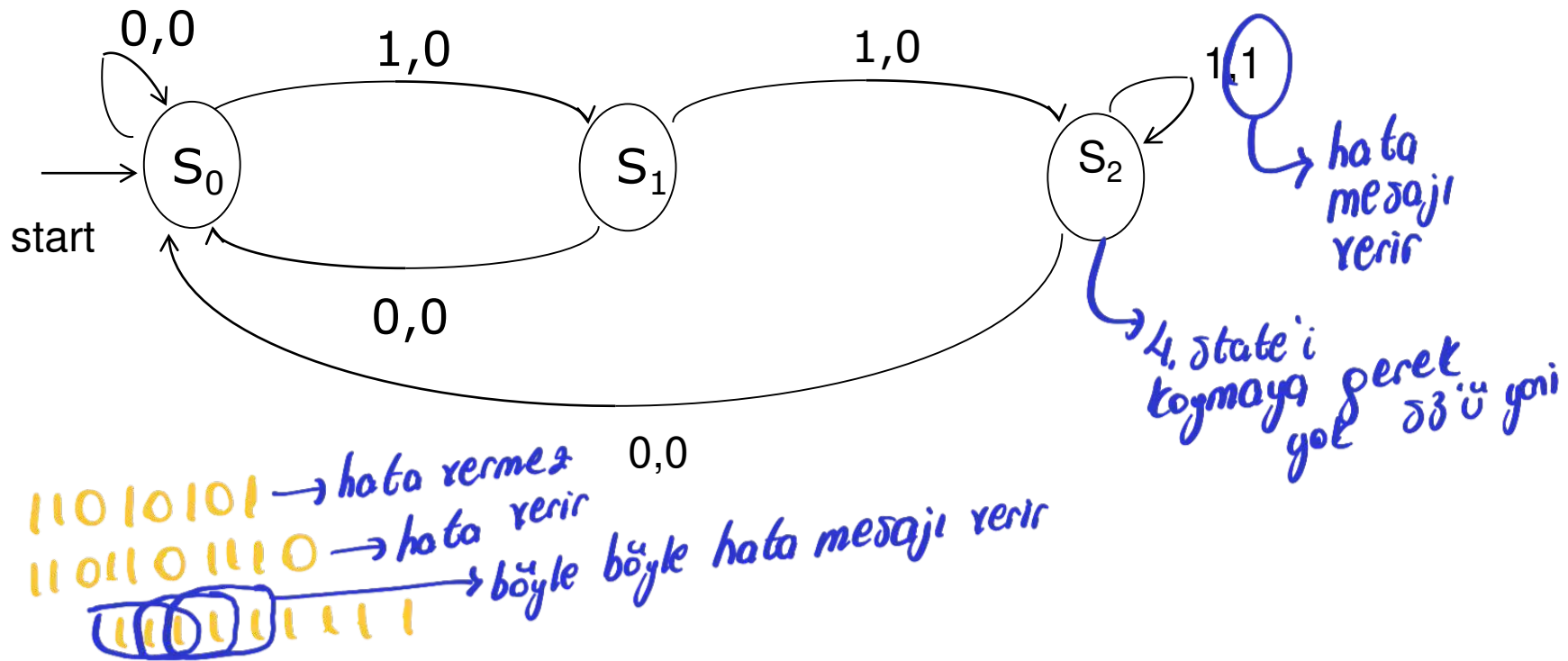
elde 1 ise  
kayıp 29  
4 du-  
rum  
çıkar

$$\begin{array}{r} +1 \\ +1 \\ \hline 0 \end{array} \quad \begin{array}{r} 0 \\ +1 \\ \hline 1 \end{array} \quad \text{elde yine 1}$$

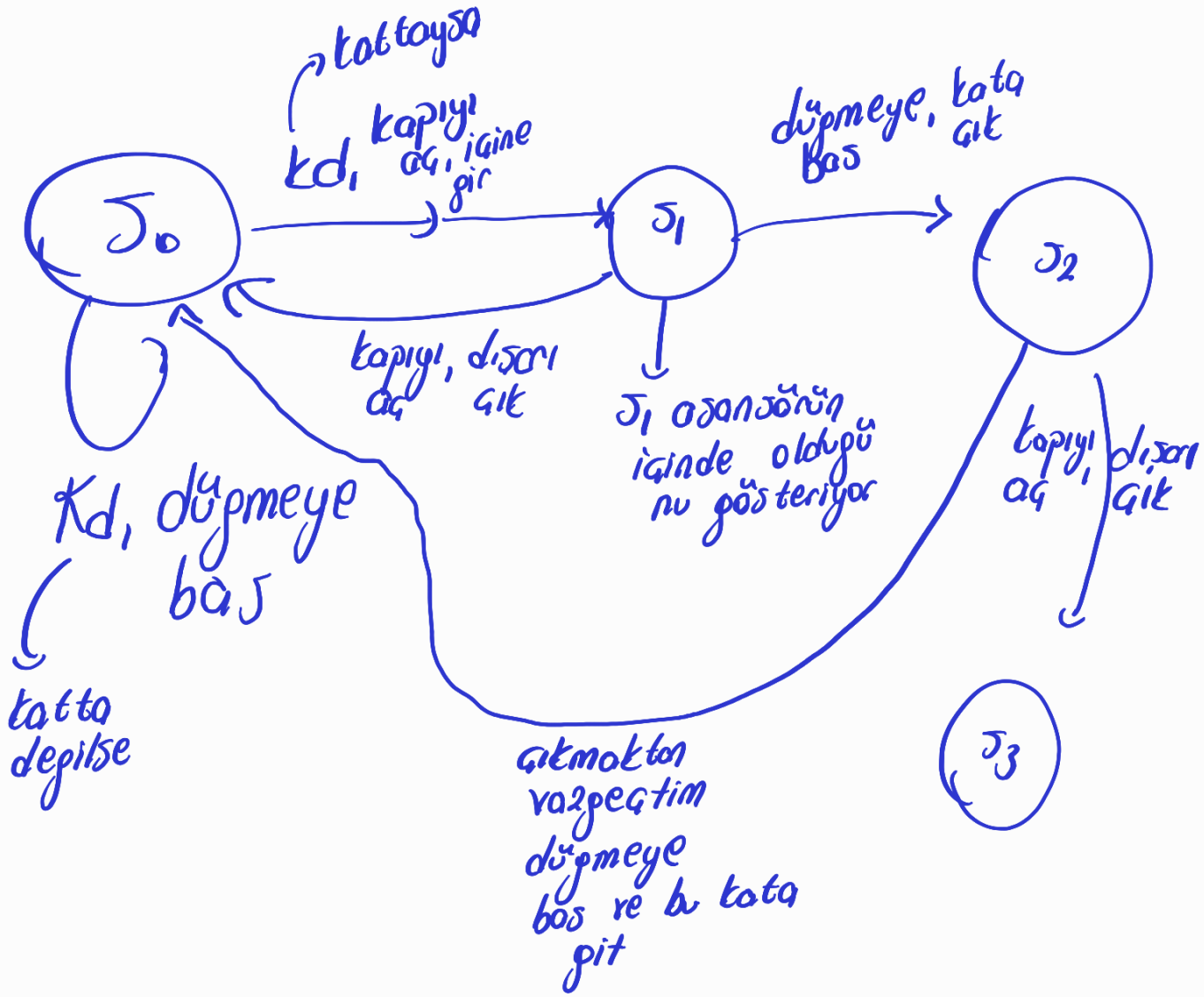
Burda  
artık  
eldemi  
kalmadı

## Örnek

Öyle bir FSM/SDM tasarlayınız ki, gönderilen mesaj içerisinde arka arkaya tekrar eden **3** tane **1** varsa "hata mesajı" olarak dışarıya **1** değerini versin.



## Aşanör Jorusu



Bir değişkenimiz maksimum 4 karakterden oluşacak ilk karakterimiz kesinlikle nümerik olmayacak gelen bir inputun verdiklerimiz kuralların bu makineyi tasarlıyoruz

10

