Sonlu Durum ve Turing Makineleri

Ders 12

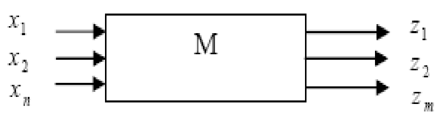
Yrd.Doç.Dr. İbrahim TÜRKYILMAZ

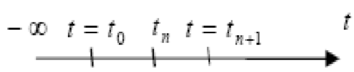
Sonlu Durum Makinesi

- Sonlu durum makinesi aşağıdakilerden oluşur:
  a) Bir σ başlangıç durumu,
  b) Sonlu sayıda duruma sahip olan sonlu durum kümesi $S=\{s_0, s_1, \ldots, s_n\}$,
  c) Giriş sembolleri, $I=\{i_1, i_2, \ldots, i_n\}$,
  d) Çıkış sembolleri, $O=\{o_1, o_2, \ldots, o_m\}$,
  e) Bu parametreler ile bir geçiş fonksiyonu tanımlanır $f : S \times I \rightarrow S$,
  f) Bir de çıktı fonksiyonu $g : S \times I \rightarrow O$,

  Böylece sonlu durum makinesi $M=(S,I,O,f,g, \sigma)$ ile ifade edilir.

- $O(t+1)$ çıkışı temelde $I(t)$'ye bağlıdır.
- $S(t)$'ye, o andaki durum veya makinelerin başından geçen olaylar (History) denir.

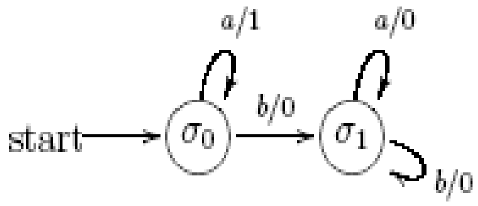




Örnek

- I={a,b} gibi iki giriş sembollü ve O={0,1} gibi iki çıkış sembollü bir sonlu durum makinesi tanımlayalım.
- Bu makine I* dan herhangi bir stringi kabul eder ve stringin başındaki a'lar kadar 1 çıktısı ve diğer durumlarda sadece 0 çıktısı verir.
- Ara durumlarda S={σ_0, σ_1} dır ve σ_0 başlangıç durumdur. Herhangi bir 'b' görülmediği durumlarda σ_0 durumundadır, 'b' görüldüğü anda σ_1 durumuna geçilir. Geçiş ve çıkış fonksiyonu tabloda verilmiştir. Yanda da geçiş diyagramı sunulmuştur.

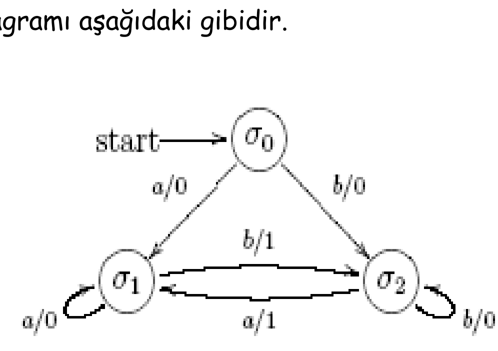
	f		g	
$\mathcal{S}$ \ $\mathcal{I}$	a	b	a	b
σ_0	σ_0	σ_1	1	0
σ_1	σ_1	σ_1	0	0



12-3

Örnek

- I={a,b} gibi iki giriş sembollü ve O={0,1} gibi iki çıkış sembollü bir sonlu durum makinesi tanımlayalım.
- Makine giriş sembolü değiştiğinde 1 diğer durumlarda 0 çıktısı vermesi istenmektedir.
- Geçiş diyagramı aşağıdaki gibidir.

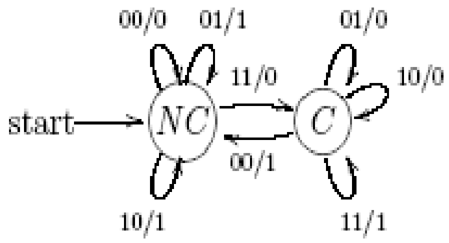


12-4

Örnek

Seri Toplayıcı:

- İki basamaklı ikilik düzende sayıyı giriş kabul eder ve çıkış olarak toplamlarını verir.
- Giriş kümesi I={00, 01, 10, 11}.
- Çıktı kümesi ise O={0, 1}.
- Durum kümesi S={NC, C} dir.
  - NC no-carry (taşıma yok), C carry (taşıma var) anlamındadır.
- Geçiş diyagramı aşağıdaki gibidir.

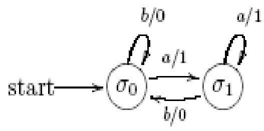


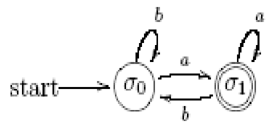
Sonlu durum otomatı

- Bir sonlu durum otomatı sonlu durum makinesi gibidir fakat çıktısı yoktur.
- Sadece kabullenme veya son durumlar olarak adlandırılan durumları vardır.
- Sonlu durum otomatı aşağıdakilerden oluşur:
  a) Bir σ başlangıç durumu,
  b) Sonlu sayıda duruma sahip olan sonlu durum kümesi
  $S=\{s_0, s_1, \ldots, s_n\}$,
  a) Giriş sembolleri, $I=\{i_1, i_2, \ldots, i_n\}$,
  b) Bir geçiş fonksiyonu $f : S \times I \rightarrow S$,
  c) Kabullenme veya son durumlar için $F(\subseteq S)$ alt kümesi
- Böylece sonlu durum otomatı $A=(S, I, f, \sigma, F)$ ile ifade edilir.
- Eğer verilen string başlangıç durumdan son duruma ulaşıyorsa otomat verilen stringi kabul eder denir.

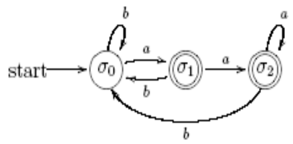
Örnek

- Aşağıdaki geçiş diyagramı a ve b lerden oluşan ve a ile biten stringleri kabul eden otomata aittir.
- İlk diyagram sonlu durum makinesiyle aynı şemayı kullanmakta ve 1 kabul veya tanımayı; 0 ise kabul etmemeyi ifade etmektedir. Diğer diyagram ise kabul durumunu çift çember ile göstermektedir.





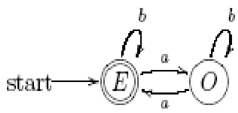
- Eğer iki sonlu durum otomatı tamamıyla aynı string kümesini kabul ediyorsa denktir denir. Aşağıdaki diyagramdaki sonlu durum otomatı yukarıdaki otomata denktir.



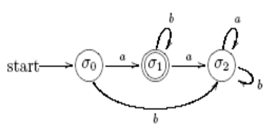
12-7

Örnek

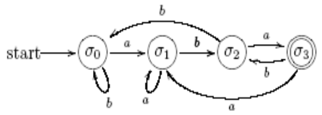
- Aşağıdaki otomat a, b'lerden oluşan ve çift sayıda a olan stringleri girdi kabul etmektedir.



- Aşağıdaki otomat a ile başlayan ve herhangi bir sayıda b ile devam eden stringleri girdi kabul etmektedir.



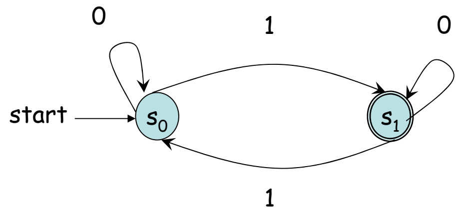
- Aşağıdaki otomat aba ile sonlanan stringleri girdileri kabul etmektedir.



12-8

Örnek

	f	
I / S	0	1
s_0	s_0	s_1
s_1	s_1	s_0



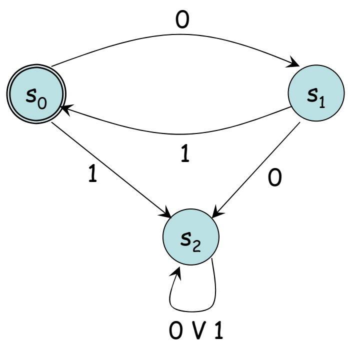
S={s0,s1}, I={0,1}

- 0 ve 1 sayılarından oluşan bir katarı girdi olarak kabul eden aşağıdaki sonlu durum otomatı, tek sayıda 1 girdilerini kabul eder. Son durum s_1 dir.
- Çıkışta bir işaret yok. Tek sayıda 1 vererek bunu yine s_1'e getirmek mümkün. Bu akseptör tek sayıda bir bulunan bir katarı kabul eder (10101110001'i kabul etmez, 6 adet 1 var).
- Sonuç : Sonlu akseptörü (durum otomatı) verilen bir katarın verilen bir gramere uygun olup olmadığını kontrol eder. Uygunluk son duruma erişip erişmeme ile anlaşılıyor.

12-9

Örnek

- I={0,1} , S={ s_0, s_1, s_2}, s_2 dipsiz kuyu giren çıkamaz.



- Bu akseptör boş katar, 01,0101, 0101001 gibi katarları kabul eder.

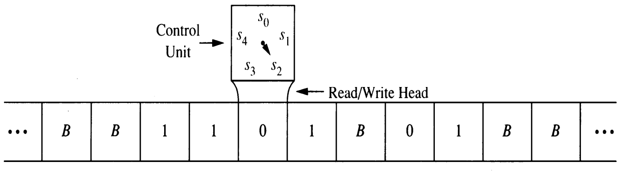
12-10

Turing Makineleri

- **Turing makinesi**, Karmaşık matematiksel hesapların belirli bir düzenek tarafından yapılmasını sağlayan hesaplama makinesi.
- Karmaşık hesapların belirli bir düzenek tarafından yapılıp yapılanamayacağı 20.yy'ın başlarında büyük bir tartışma konusu olmuştu.
- Öteden beri el ile veya zihinden yapılan hesaplamalar çok zaman almakla birlikte, birçok hatayı da beraberinde getiriyordu.
- Tüm bu tartışmalar sürerken, 1936 yılında, ünlü matematikçi **Alan M. Turing** "Saptama Problemi Hakkında Bir Uygulamayla Birlikte Hesaplanabilir Sayılar" (*On computable numbers, with an application to the Entscheidungs problem*) isimli bir makalesini yayınladı.
- Makalesinde teorik ve matematiksel temellere dayalı sanal bir makineden bahseden Turing, her türlü matematiksel hesabın bu sanal makineyle yapılabileceğini iddia ediyordu.
- Turing'in 1950 yılında yayınlanan "Hesaplama Mekanizması ve Zeka" (*Computing Machinery and Intelligence*) isimli ikinci makalesi ise, makineler ve zekayla ilgili birçok tartışmalı konuya cevap niteliğindeydi. İşte bu makalelerde sözü geçen sanal makine daha sonraları **Turing Makinesi** (*The Turing Machine*) olarak isimlendirildi.

Turing Makineleri

- Bilgisayar bilimlerinin önemli bir kısmını oluşturan otomatlar ve Algoritma Analizi çalıştırmalarının altındaki dil bilimin (language science) en temel taşlarından birisidir.
- 1936 yılında Alan Turing tarafından ortaya atılan makine günümüzde pek çok teori ve standardın belirlenmesinde önemli rol oynamaktadır.
- Turing makinesi aşağıdakilerden oluşur;
  - Ardışık hücrelerden oluşan sonsuz teyp.
  - Her bir hücre boş veya verilen alfabeden bir sembol içerir.
  - Kontrol ünitesi sonlu talimatlar kümesini içerir.
  - Teyp kafası teypten (band) semboller okur veya yazar (siler).



Turing Makineleri

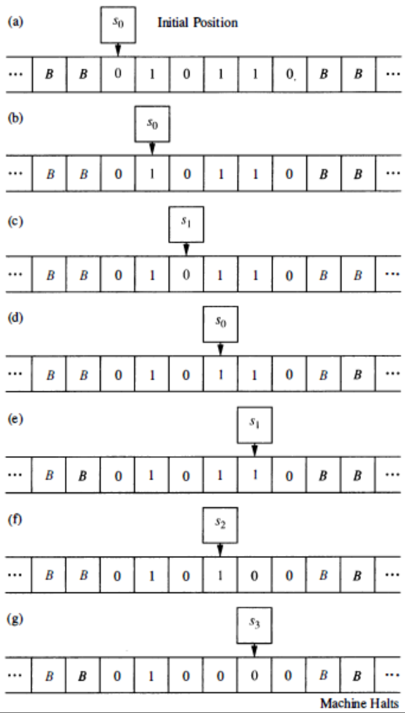
- Bu makinelerde şerit şeklinde bellek vardır. Her bir bellek gözünde alfabenin sembollerinden biri olacaktır. Yine,
  - Bir Q kümesi (başı q_0)
  - Bir A alfabesi (b boşluk dahil)
- g : Q × A→ Q × A {R, L} kümesi bu şeridi okuyup kafanın sağ tarafa mı, yoksa sol tarafa mı hareket ettiğini belirtiyor.
- (q_i, a_j, q_k, a_l, Y) ile tanımlanır
  - q_i : Makinanın durumu
  - a_j : kafanın şeritten okuduğu sembol
  - q_k : Makinanın yeni konumu
  - a_l : Kafanın şerite yazdığı yeni karakter
  - Y : R veya L olarak sağa yada sola doğru kafanın hareketi (bir göz hareket edecek şekilde).
- Böyle bir bellek özelliği olan soyut makine Turing Makinesi olarak bilinir.
- Turing Makinesi programı belirli bir işlemi yapan sıralı beşlilerden (q_i, a_j, q_k, a_l, Y) oluşur.

12-13

Örnek

- Yedi tane
  - (s_0,0,s_0,0,R),
  - (s_0,1,s_1,1,R),
  - (s_0,B,s_3,B,R),
  - (s_1,0,s_0,0,R),
  - (s_1,1,s_2,0,L),
  - (s_1,B,s_3,B,R)
  - (s_2,1,s_3,0,R),

  sıralı beşli ile tanımlanmış ve şekil (a) da görülen band üzerinde çalıştırılmaya başlanan T Turing Makinesinde bandın son durumu ne olur?
- Bu sanal makine ne yapar?



12-14

Kümeleri Tanımlama için Turing Makinesini Kullanma

- Kümeleri tanımlama için Turing makinelerini kullanabiliriz. Bunu yapabilmek için son durum kavramının tanımlanması gerekmektedir. T Turing makinesinin son durumu, T yi tanımlayan sıralı beşlilerin ilk durumu olmayan durum olarak tanımlanabilir (bir önceki örnekte karşımıza çıkan s_3 durumu gibi).

Tanım: V, I alfabesinin bir altkümesi olsun. $T=(A,I,f,s_0)$ Turing makinesi, ancak ve ancak x banda yazıldığında başlangıç yerinden başlar ve son durumda durursa, V* içerisindeki x stringini tanır.

- V* ın bir A altkümesini tanımak için V içerinde olmayan semboller kullanabiliriz. Bunun anlamı, I giriş alfabesi V içerinde olmayan semboller içerebilir. Bu ekstra semboller genellikle taglar (markers) olarak adlandırılır.
- T Turing makinesinin V* içerisindeki x stringini ne zaman tanımaz?

  T durmadığında veya son durumda durmadığında, x stringi tanınmaz.

Örnek

Soru: İkinci basamağında 1 olan stringleri tanıyan bir Turing makinesi tasarlayınız. $(0 \cup 1)1(0 \cup 1)^*$ şeklinde bir düzenli küme oluşturmalıdır.

- En soldaki boş bank hücresinden işlem başlayan ve sağa doğru hareket eden ve ikinci sembolü 1 olup olmadığını belirleyen. Eğer ikinci sembol 1 ise makine son duruma hareket etmelidir. Makine son duruma ulaşmadığında durmamalıdır.
- Böyle bir makineyi tasarlamak için $(s_0,0,s_1,0,R)$, $(s_0,1,s_1,1,R)$ sıralı beşlileriyle ilk sembol okunmalı ve Turing makinesi s_1 durumuna geçmelidir.
- Daha sonra, $(s_1,0,s_2,0,R)$, $(s_1,1,s_3,1,R)$ sıralı beşlileriyle ikinci sembol okunmalı ve 0 ise s_2 durumuna geçilmeli veya sembol 1 ise s_3 durumuna geçilmelidir. İkinci basamağında 0 olan stringi tanımamak için s_2 son durum olmamalıdır.
- İkinci basamağında 1 olan stringin tanınması için s_3 son durum olmalıdır. Böylece $(s_2,0,s_2,0,R)$ sıralı beşlisini ekleyebiliriz. Boş string ve bir basamaklı bir stringi tanımak istemediğimizden $(s_0,B,s_2,0,R)$ ve $(s_1,B,s_2,0,R)$

Turing Makinesiyle Fonksiyon Hesaplama

- Turing Makinesi fonksiyonların değerlerini bulan bilgisayarlar olarak ta düşünülebilir.
- Bu durum, x string girdisi verildiğinde band üzerindeki y stringinde duran bir Turing makinesi olarak düşünülebilir.
- T(x)=y tanımlanabilir ve T nin tanım kümesi T yi durduran stringler kümesidir, verilen x girdisi için T durmadığında T(x) tanımlı değildir.

- Turing makinesini fonksiyon hesaplayıcı olarak düşünmek için tamsayıların sıralı k-lılarının band üzerinde ifade etmenin bir yolunu bulmalıyız.
- Bir n tamsayısını ifade etmek için n+1 tane 1 kullanabiliriz.
- Böylece 0'ı ifade etmek için 1 stringini, 5'i ifade etmek için 111111 strigini kullanabiliriz.
- $(n_1,n_2,\ldots, n_k)$ sıralı k-lısını ifade etmek için n_1+1 tane 1, ardından *, n_2+1 tane 1, ardından *, bu şeklide devam ederek n_k+1 tane 1 kullanabiliriz.
- Örneğin (2,0,1,3) sıralı dörtlüsü için 111*1*11*1111 kullanabiliriz.

Örnek

Soru: Negetif olmayan iki tamsayıyı toplayan Turing makinesini tasarlayınız.

- $f(n_1,n_2)=n_1+n_2$ fonksiyonunu hesaplayan Turing makinesine ihtiyacımız var. (n_1,n_2) çifti n_1+1 tane 1, ardından * ve bunları takiben n_2+1 tane 1 den oluşan bir string ile ifade edilir.
- T makinesi bu girdiyi alıp n_1+n_2+1 tane 1 olan bir çıktı vermelidir.
- Makine girdi stringinin en soldaki 1 den başlamalıdır ve bu 1 silinmelidir. $n_1=0$ olduğunda, silme işleminden sora * dan önce hiç 1 olmadığından, * en soldaki 1 ile yer değiştirmeli ve makine durmalıdır.
  - $(s_0,1,s_1,B,R)$,
  - $(s_1,*,s_3,B,R)$,
  - $(s_1,1,s_2,B,R)$,
  - $(s_2,1,s_2,1,R)$,
  - $(s_2,*,s_3,1,R)$,

Örnek

- Bu sistemde 1 ve 4 sayılarını temsil etmek için aşağıdaki gösterim kullanılır.

b	1	1	b	1	1	1	1	1	b		

- Yazacağımız program m=2 ile n=5 sayılarını toplayacaktır.

b	1	1	1	b	1	1	1	1	1	1	b		

- Burada kafanın konumunun nerede olduğu önemlidir. Kafa en soldaki 1'in üzerindedir.

q_i	s_j	q_k	s_l	Y
0	1	0	1	R
0	b	1	1	L
1	1	1	1	L
1	b	2	b	R
2	1	3	b	R
3	1	4	b	R

12-19