

BM 305 Biçimsel Diller ve Otomatlar (Formal Languages and Automata)

Hazırlayan: M.Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

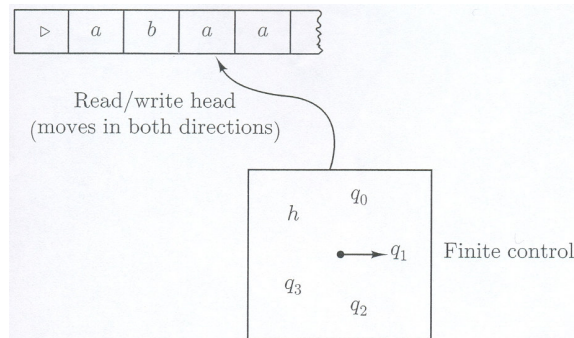
Konular

- Turing Machine
- Computing with Turing Machines
- Recursive and Recursively Enumerable Languages
- Extensions of the Turing Machines

Turing Machines

- Pushdown automata ve finite automata genel bilgisayar modelleridir.
- Ancak PDA ve FA, $L = \{a^n b^n c^n : n \geq 0\}$ gibi basit bir dili tanıma kapasitesine sahip değildirler.
- Turing makinesi buna benzer ve daha karmaşık dilleri tanıyabilir.
- Turing makinesi, Alan Turing tarafından 1936 yılında ortaya atılmıştır.
- Turing makinesi bir finite control, bir tape ve tape üzerinde yazma okuma için kullanılan bir head' ten oluşmaktadır.

Turing Machines



- Temel olarak bir Turing makinesi bir tape ve bir finite state control'dan oluşur.
- Finite state control ve tape arasında iletişim bir okuma/yazma kafası tarafından sağlanır.
- Control unit bulunulan durum ve okunan sembole bağlı olarak her adımda iki farklı işlemi gerçekleştirir;
 1. Control unit yeni bir duruma geçer.
 2. a. Tape üzerinde okunan alana yeni bir sembol yazar.
b. Okuma/yazma kafasını bir sağa veya sola hareket ettirir.

Turing Machines

Özellikler:

- Tape ünitesinin sol kısmı sonlu ve sağ kısmı sonsuz uzunlukta olabilir.
- Makinenin tape ünitesinin en solundan da sola gitmesini önlemek için en sol kısımda $\triangleright$ işaretçisi bulunmaktadır.
- Okuma/yazma kafası $\triangleright$ işaretini okuduğunda otomatik olarak hemen bir sağa geçer.
- Kafanın sola ve sağa hareketini sağlamak için $\leftarrow$ ve $\rightarrow$ sembolleri kullanılır.
- $\leftarrow$ ve $\rightarrow$ sembolleri alfabeye ait değildir.
- Tape üzerindeki input string $\triangleright$ işaretinden hemen sonra başlar ve string'ten sonraki kısmın tamamı başlangıçta boştur ve $\sqcup$ sembolüyle gösterilir.

Turing Machines

Tanım:

Bir Turing Makinesi $M = (K, \Sigma, \delta, s, H)$ şeklinde bir quintuple ile gösterilir.

K sonlu durumlar kümesi

Σ alfabe, boşluk sembolü $\sqcup$ ve ensoldaki sonlandırma sembolü $\triangleright$ elemanıdır, ancak sola ve sağa gidiş sembolleri $\leftarrow$ ve $\rightarrow$ elemanı değildir.

s başlangıç durumu

$H \subseteq K$ halting states (bitiş durumları)

δ geçiş fonksiyonu, $(K - H) \times \Sigma$ dan $K \times (\Sigma \cup \{\leftarrow, \rightarrow\})$ kümesinedir.

a) tüm $q \in K - H$ için, eğer $\delta(q, \triangleright) = (p, b)$ ise $b = \rightarrow$ olur

b) tüm $q \in K - H$ ve $a \in \Sigma$ için, eğer $\delta(q, a) = (p, b)$ ise $b \neq \triangleright$ olur

a. Okuma kafası sol baştaki sembolü ($\triangleright$) okursa otomatik olarak bir sağa geçer.

b. Okuma kafası alfabedeki bir sembolü okuduğunda yeni geçeceği durumda $\triangleright$ sembolünün üzerinde duramaz ve üzerine birşey yazamaz.

Turing Makinesinde halting state'ler için δ geçiş fonksiyonu tanımlanmaz ve halting state'lerden birisine geçer geçmez çalışması sonlanır.



Turing Machines

Örnek: Bir Turing Makinesi $M = (K, \Sigma, \delta, s, \{h\})$ şeklinde tanımlanmıştır.

$K = \{q_0, q_1, h\}, \Sigma = \{a, \sqcup, \triangleright\},$
 $s = q_0$ olsun. Geçiş fonksiyonu δ
 yandaki tabloda verildiği gibidir.

q, σ	$\delta(q, \sigma)$
q_0, a	$(q_1, \sqcup)$
$q_0, \sqcup$	$(h, \sqcup)$
$q_0, \triangleright$	$(q_0, \rightarrow)$
q_1, a	(q_0, a)
$q_1, \sqcup$	$(q_0, \rightarrow)$
$q_1, \triangleright$	$(q_1, \rightarrow)$

- M başlangıç durumu q_0 dan çalışmaya başlar ve sağa doğru tüm a ları $\sqcup$ ile değiştirir.
- İlk $\sqcup$ sembolünü bulduğunda halt durumuna geçerek çalışmasını sonlandırır.
- M makinesi q_0 ve q_1 durumları arasında sürekli gidip gelir.
- Burada $\delta(q_1, a)$ durumunda hiçbir zaman olamaz. Ancak $\delta, (K - H) \times \Sigma$ üzerinde bir fonksiyon olduğu için tanımlanmalıdır.



Turing Machines

Örnek: Bir Turing Makinesi $M = (K, \Sigma, \delta, s, \{h\})$ şeklinde tanımlanmıştır. $K = \{q_0, h\}, \Sigma = \{a, \sqcup, \triangleright\}, s = q_0$ ve $H = \{h\}$ olsun. Geçiş fonksiyonu δ aşağıdaki tabloda verildiği gibidir.

q, σ	$\delta(q, \sigma)$
q_0, a	$(q_0, \leftarrow)$
$q_0, \sqcup$	$(h, \sqcup)$
$q_0, \triangleright$	$(q_0, \rightarrow)$

- M makinesi sola doğru tarama yapar ve ilk $\sqcup$ sembolünü bulduğunda halt durumuna geçerek çalışmasını sonlandırır.
- Eğer $\triangleright$ sembolüne kadar a varsa, en sola geldiğinde hemen sağa geçer ve tekrar sola geçer.
- Diğer deterministik makinelerin tersine Turing Makinesinin çalışması hiç sonlandırılmayabilir.

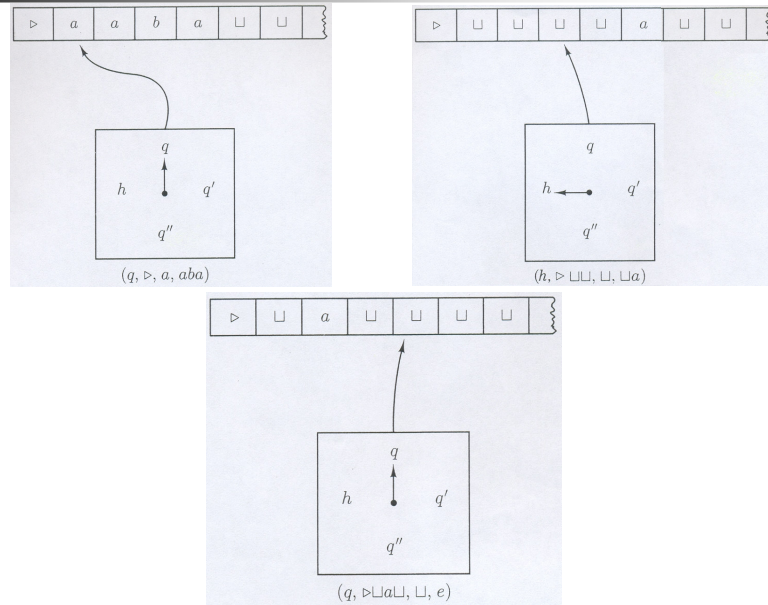
Turing Machines

Tanım: Bir Turing makinesi $M = (K, \Sigma, \delta, s, H)$ için konfigürasyon $K \times \Sigma^* \times (\Sigma^* (\Sigma - \{\sqcup\}) \cup \{e\})$ kümesinin bir elemanıdır.

- Konfigürasyon sol bitiş sembolü ile başlar ve hiçbir zaman boşluk sembolüyle ($\sqcup$) bitmez.
- $(q, \triangleright a, aba)$, $(q, \triangleright \sqcup \sqcup \sqcup, \sqcup a)$ ve $(q, \triangleright \sqcup a \sqcup \sqcup, e)$ konfigürasyonlardır.
- $(q, \triangleright baa, abc \sqcup)$ ve $(q, \sqcup aa, ba)$ konfigürasyon değildirler.
- (q, wa, u) konfigürasyonunda tape içeriği kısaca $w\underline{a}u$ şeklinde gösterilir. Okuma kafası 'a' dadır.
- (q, wa, u) yerine kısa olarak $(q, w\underline{a}u)$ yazılabilir.

Turing Machines

Örnek:





Turing Machines

Tanım: Bir Turing makinesi $M = (K, \Sigma, \delta, s, H)$ için iki konfigürasyon $(q_1, w_1 a_1 u_1)$ ve $(q_2, w_2 a_2 u_2)$ için $a_1, a_2 \in \Sigma$ ise $(q_1, w_1 \underline{a_1} u_1) \vdash_M (q_2, w_2 \underline{a_2} u_2)$ konfigürasyon geçişi için

$b \in \Sigma \cup \{\leftarrow, \rightarrow\}$ için $\delta(q_1, a_1) = (q_2, b)$ geçiş fonksiyonu vardır ve burada,

1. $b \in \Sigma$, $w_1 = w_2$, $u_1 = u_2$ ve $a_2 = b$, veya
2. $b = \leftarrow$, $w_1 = w_2 a_2$ ve
 - (a) $u_2 = a_1 u_1$, eğer $a_1 \neq \sqcup$ ve $u_1 \neq e$, veya
 - (b) $u_2 = e$, eğer $a_1 = \sqcup$ ve $u_1 = e$, veya
3. $b = \rightarrow$, $w_2 = w_1 a_1$ ve
 - (a) $u_1 = a_2 u_2$, veya
 - (b) $u_1 = u_2 = e$, ve $a_2 = \sqcup$ olur.



Turing Machines

Örnek: $w, u \in \Sigma$ olsun. u 'nun sonu $\sqcup$ olmasın ve $a, b \in \Sigma$ olsun.

Durum 1. $\delta(q_1, a) = (q_2, b)$

Örnek: $(q_1, w \underline{a} u) \vdash_M (q_2, w \underline{b} u)$

Durum 2. $\delta(q_1, a) = (q_2, \leftarrow)$

(a) Örnek: $(q_1, w \underline{b} a u) \vdash_M (q_2, w \underline{b} a u)$

(b) Örnek: $(q_1, w \underline{b} \sqcup) \vdash_M (q_2, w \underline{b})$

Durum 3. $\delta(q_1, a) = (q_2, \rightarrow)$

(a) Örnek: $(q_1, w \underline{a} b u) \vdash_M (q_2, w \underline{a} b u)$

(b) Örnek: $(q_1, w \underline{a} \sqcup) \vdash_M (q_2, w \underline{a} \sqcup)$



Turing Machines

Tanım: $\vdash_M^*$ ilişkisi $\vdash_M$ ilişkisinin reflexive, transitive closure'dur. C_2 konfigürasyonu C_1 'den oluşturulmuştur eğer $C_1 \vdash_M^* C_2$ olursa

$C_1 \vdash_M C_2 \vdash_M \dots \vdash_M C_n$ konfigürasyon geçişleri için $length=n$ olur ve kısaca $C_1 \vdash_M^n C_n$ şeklinde gösterilir.

Burada n başlangıçtan sonuca gitmek için gereken adım sayısı olarak ifade edilmektedir.



Turing Machines

Örnek: Bir Turing makinesi $M = (K, \Sigma, \delta, s, \{h\})$, $K = \{q_0, q_1, h\}$, $\Sigma = \{a, \sqcup, \triangleright\}$, $s = q_0$ olsun. Geçiş fonksiyonu δ aşağıdaki gibi tanımlansın.

q, σ	$\delta(q, \sigma)$
q_0, a	$(q_1, \sqcup)$
$q_0, \sqcup$	$(h, \sqcup)$
$q_0, \triangleright$	$(q_0, \rightarrow)$
q_1, a	(q_0, a)
$q_1, \sqcup$	$(q_0, \rightarrow)$
$q_1, \triangleright$	$(q_1, \rightarrow)$

$(q_1, \triangleright \sqcup aaaa)$ başlangıç konfigürasyonundan çalışmaya başlarsa yandaki geçişleri yapar,

$(q_1, \triangleright \sqcup aaaa) \vdash_M (q_0, \triangleright \sqcup \underline{a}aaa)$
 $\vdash_M (q_1, \triangleright \sqcup \sqcup aaaa)$
 $\vdash_M (q_0, \triangleright \sqcup \sqcup \underline{a}aa)$
 $\vdash_M (q_1, \triangleright \sqcup \sqcup \sqcup aa)$
 $\vdash_M (q_0, \triangleright \sqcup \sqcup \sqcup \underline{a}a)$
 $\vdash_M (q_1, \triangleright \sqcup \sqcup \sqcup \sqcup a)$
 $\vdash_M (q_0, \triangleright \sqcup \sqcup \sqcup \sqcup \underline{a})$
 $\vdash_M (q_1, \triangleright \sqcup \sqcup \sqcup \sqcup \sqcup)$
 $\vdash_M (q_0, \triangleright \sqcup \sqcup \sqcup \sqcup \sqcup \sqcup)$
 $\vdash_M (h, \triangleright \sqcup \sqcup \sqcup \sqcup \sqcup \sqcup)$



Turing Machines

Tanım: Karmaşık Turing makineleri basit makinelerin birleşimi şeklinde oluşturulabilir.

Basit Makineler:

Symbol-writing machines:

Her $a \in \Sigma \cup \{\leftarrow, \rightarrow\} - \{\triangleright\}$ için $M_a = (\{s, h\}, \Sigma, \delta, s, \{h\})$ tanımlanabilir ve $\delta(s, b) = (h, a)$ olur, burada $b \in \Sigma \cup -\{\triangleright\}$.

Buradada $\delta(s, \triangleright) = (s, \rightarrow)$ otomatik geçişi geçerlidir.

Bu makine sadece a işlemini yapar. Eğer $a \in \Sigma$ ise a yazılır, $a \in \{\leftarrow, \rightarrow\}$ ise sola veya sağa gidilir ve makine durur.

Yazma makinesi çok sık kullanılacağı için M_a yerine kısaca a kullanılır.

Head-moving machines:

$M_{\leftarrow}$ ve $M_{\rightarrow}$ şeklindedir ve kısaca L (left) ve R (right) olarak gösterilir.



Turing Machines

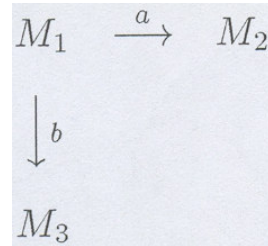
Tanım: Karmaşık Turing makineleri basit makineleri birleştirerek oluşturulabilir.

Makine birleştirme kuralları:

- Makineler finite otomatlardaki durumlar gibidir ve durumların bağlanması şeklinde birleştirilir.
- Bir makineden diğerine yapılan bağlantı ilkinin halt durumuna geçmesiyle çalışır ve ikinciye geçilir.
- İkinci makine başlangıç durumuyla çalışmaya başlar.

Örnek: Yandaki şekilde M_1 , M_2 ve M_3 Turing makinesidir.

M_1 başlangıç durumunda çalışmaya başlar. M_1 halt durumuna geçince okunan sembol a ise M_2 başlangıç durumunda çalışmaya başlar, b ise M_3 başlangıç durumunda çalışmaya başlar.





Turing Machines

Örnek: İki R makinesi aşağıdaki gibi birleştirilsin.



- Bu makine okuma kafasını önce bir sağa geçirir ve okunan sembol a , b , $\triangleright$ veya $\sqcup$ ise bir sağa daha geçirir.
- Eğer bir geçiş alfabedeki tüm sembolleri içerirse etiket yazılmadan

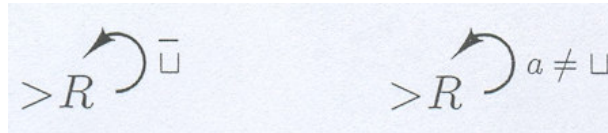
$R \rightarrow R$

şeklinde gösterilir. Daha da basitleştirilerek RR veya R^2 şeklinde gösterilebilir.



Turing Machines

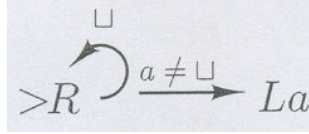
Örnek: Eğer $a \in \Sigma$ ise, birçok sembol kullanılan oklar yerine $\bar{a}$ şeklinde gösterimde kullanılabilir.



- Yukarıdaki şekilde soldaki makine $\sqcup$ bulana kadar sağa gider ve $R_{\sqcup}$ şeklinde gösterilir.
- Yukarıdaki sağdaki şekil aynı işlemi ifade etmektedir. Ancak okunan a sembolünün daha sonra kullanılmasını sağlamaktadır.

Turing Machines

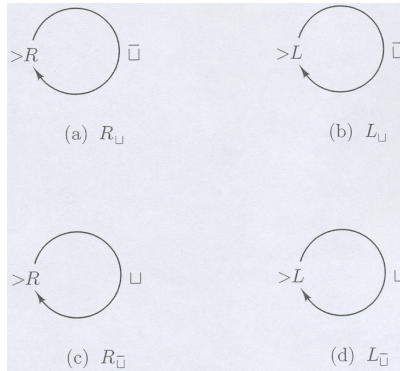
Örnek: Aşağıdaki makine bir sembol bulana kadar sağa gider ve bulduğu sembolü bir soldaki alana kopyalar.



La bir sola gitmeyi ve en son okunan $a \in \Sigma$ sembolünü yazmayı ifade etmektedir.

Turing Machines

Örnek: Aşağıdaki makineler hep sağa veya sola gider ve bir sembol arar. Aradığını bulur bulmaz çalışması sonlanır.

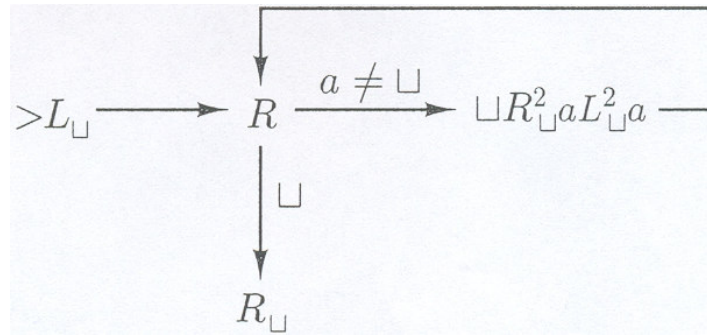


- (a) $R_{\square}$, sağa doğru tarama yapar ve ilk bulduğu boşlukta durur
- (b) $L_{\square}$, sola doğru tarama yapar ve ilk bulduğu boşlukta durur.
- (c) $R_{\square}$, sağa doğru tarama yapar ve ilk bulduğu sembolde durur.
- (d) $L_{\square}$, sola doğru tarama yapar ve ilk bulduğu sembolde durur.



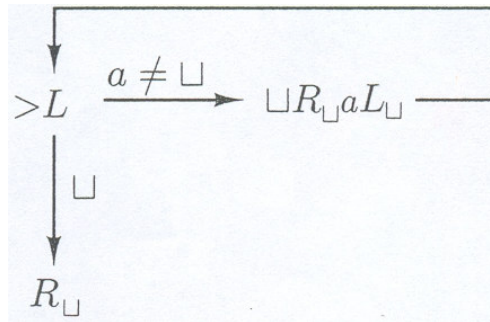
Turing Machines

Örnek: Aşağıdaki kopyalama makinesi bir w stringini sağ kısmına kopyalar. $\sqcup w \sqcup$ string'i için sonuç string'i $\sqcup w \sqcup w \sqcup$ olur.

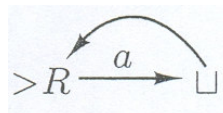


Turing Machines

Örnek: Aşağıdaki sağa kaydırma makinesi bir w stringini bir sağa kaydırır. $\sqcup w \sqcup$ string'i için sonuç string'i $\sqcup \sqcup w \sqcup$ olur.



Örnek: Aşağıdaki makine tape üzerindeki tüm a ları siler.





Computing with Turing Machines

- Turing makinesinde işlemler için gerekli giriş string'i $\triangleright$ sembolünün sağına yazılır ve içinde boşluk sembolü yoktur.
- Giriş string'inin sağındaki kısım tümüyle boşluk sembolüdür.
- Bundan sonraki örneklerde giriş string'i ile $\triangleright$ sembolü arasında bir boşluk sembolü ($\sqcup$) vardır.
- Eğer $M = (K, \Sigma, \delta, s, H)$ bir Turing makinesi ve $w \in (\Sigma - \{ \sqcup, \triangleright \})^*$ ise M makinesinin w girişi için başlangıç konfigürasyonu $(s, \triangleright \sqcup w)$ şeklindedir.



Computing with Turing Machines

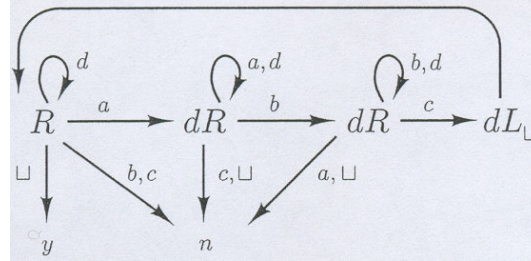
Tanım : $M = (K, \Sigma, \delta, s, H)$ bir Turing makinesi $H = \{y, n\}$ şeklinde iki tane halt state'e sahip olsun. Bunlar y kabul konfigürasyonu (**accepting configuration**) ve n red konfigürasyonu (**rejecting configuration**) olsun.

- Eğer $w \in (\Sigma - \{ \sqcup, \triangleright \})^*$ girişi için $(s, \triangleright \sqcup w)$ konfigürasyonu M makinesini accepting configuration'lardan birisinde sonlandırırsa w bu dile aittir, halting configuration'lardan birisinde sonlandırırsa dile ait değildir.
- M makinesi bir dili belirler (**decide**) ve
 - eğer $w \in L$ ise w string'ini kabul eder
 - eğer $w \notin L$ ise w string'ini red eder
- Bir dili belirleyen bir Turing makinesi varsa bu dil özyineli (**recursive**) olarak adlandırılır.



Computing with Turing Machines

Örnek : $L = \{a^n b^n c^n : n \geq 0\}$ dilini tanıyan Turing makinesi aşağıdadır.



- M makinesi n döngü yapar. Her döngüde makine, girişin en solundan başlar ve ilk bulduğu a yerine d , ikinci bulduğu b yerine d ve üçüncü olarak bulduğu c yerine d yazar.
- Okuma kafası yeniden string'in en soluna gider.
- Makine bir a ararken b veya c ye, b ararken c veya $\sqcup$ ye, c ararken a , veya $\sqcup$ ya rastlarsa n durumuna gider.
- Eğer bir a araken $\sqcup$ gelirse (sağ kısmın tamamı d olmuştur) çalışmasını y durumuna geçerek sonlandırır.



Computing with Turing Machines

Tanım : $M = (K, \Sigma, \delta, s, \{h\})$ bir Turing makinesi ve $\Sigma_0 \subseteq \Sigma - \{\sqcup, \triangleright\}$ ve $w \in \Sigma_0$ olsun.

Eğer M , w giriş için halt state'lere ulaşıyorsa ve bazı y ler için $(s, \triangleright \sqcup w) \vdash_M^* (h, \triangleright \sqcup y)$ ise, y M makinesinin çıkışı olarak adlandırılır ve $M(w)$ şeklinde gösterilir. $M(w)$ sadece makinenin halt durumuna ulaşması halinde tanımlıdır.

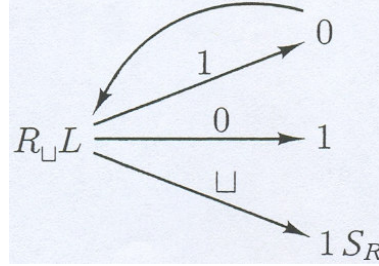
Eğer bir f fonksiyonu Σ_0 dan Σ_0 a tanımlı ve tüm $w \in \Sigma_0$ için $M(w) = f(w)$ ise M makinesi f fonksiyonunu hesaplar.

M makinesinin çalışması bittiğinde tape üzerinde $\triangleright \sqcup f(w)$ vardır ve bu fonksiyona **recursive** denilir.



Computing with Turing Machines

Örnek : Aşağıda binary olarak yazılmış sayının bir fazlasını hesaplayan bir Turing makinesi görülmektedir. ($\text{succ}(n) = n + 1$)



- M makinesi önce girişin en sağıını bulur.
- Sonra 1 gördüğü sürece sola gider ve her 1 değerini 0 olarak değiştirir.
- İlk gördüğü 0 yerine 1 yazarak çalışmasını sonlandırır.
- Eğer sola giderken $\sqcup$ sembolü görürse yerine 1 yazar ve tüm girişi sağa bir pozisyon shift ederek çalışmasını sonlandırır.



Computing with Turing Machines

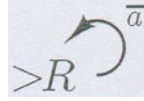
Tanım: $M=(K, \Sigma, \delta, s, H)$ bir Turing makinesi, $\Sigma_0 \subseteq \Sigma - \{\sqcup, \triangleright\}$ alfabe ve $L \subseteq \Sigma_0^*$ olsun.

- Eğer sadece $w \in L$ iken M makinesi halt durumuna geçerse, M makinesi L dilini yarı belirler (**semidecides**) denir.
- Bir dil bir Turing makinesi tarafından semidecide ediliyorsa bu dil özyineli sıralı (**recursively enumerable**) olarak adlandırılır.



Computing with Turing Machines

Örnek: $L = \{w \in \{a, b\}^* : w \text{ içinde en az bir } a \text{ vardır}\}$ şeklinde tanımlı bir dil aşağıdaki Turing makinesi tarafından semidecide edilir.



- $w \in \{a, b\}^*$ girişi için makine $(q_0, \triangleright \sqcup w)$ başlangıç konfigürasyonundan çalışmaya başladığında sağa doğru ilk a okuduğunda çalışmasını sonlandırır.
- Eğer a bulamazsa sonsuza kadar çalışır ve hiçbir zaman halt durumuna ulaşamaz.
- M makinesi L dilini semidecide yapar ve L recursively enumerable dildir.

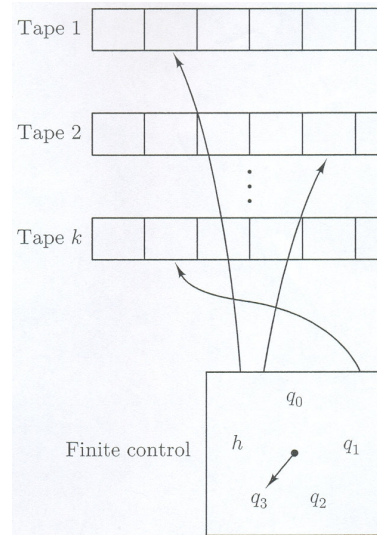


Extensions of the Turing Machines

Turing makinesine ekstra özellikler kazandırılarak özel problemlerin çözümünde kullanılabilir.

Multi Tapes:

- Birden fazla tape birimine sahiptir.
- Her adımda tüm kafalar okuma yapar.
- Bulunulan duruma ve okunan sembollere bağlı olarak birkaç tanesine yazma yapılır veya sağa veya sola hareket edilir ve durum değiştirilir.
- k -tape Turing makinesi k adet tape ünitesine sahiptir.
- Şimdiye kadar görülen Turing makinesi 1-tape makinedir.





Extensions of the Turing Machines

Tanım: $k \geq 1$ olmak üzere $M=(K, \Sigma, \delta, s, H)$ bir k -tape Turing makinesi ise K, Σ, s ve H tanımları 1-tape ile aynıdır. Ancak δ transition fonksiyonu $(K - H) \times \Sigma^k$ dan $K \times (\Sigma \cup \{\leftarrow, \rightarrow\})^k$ ya tanımlıdır.

- Herbir durum q için $(a_1, \dots, a_k)$ k -tuple tape sembolüdür ve $\delta(q, (a_1, \dots, a_k)) = (p, (b_1, \dots, b_k))$ şeklindedir. p yeni durumu ve b_j ise j .tape üzerindeki işlemi gösterir.
- Eğer $a_j = \triangleright$ ise $b_j = \rightarrow$ olur ve otomatik olarak bir sağa geçer.



Extensions of the Turing Machines

Tanım: $M=(K, \Sigma, \delta, s, H)$ bir k -tape Turing makinesi olsun. Bir konfigürasyon $K \times (\triangleright \Sigma^* \times (\Sigma^*(\Sigma - \{\sqcup\}) \cup \{e\}))^k$ kümesinin bir elemanıdır.

- Bir konfigürasyon durumu, tape içeriklerini ve her tape için kafa pozisyonunu belirler.
- Giriş string'i 1.tape üzerindedir ve genellikle diğer tape üniteleri boştur.
- k -tape bir Turing makinesi çalışmasını sonlandırınca, sonuç 1.tape üzerindedir ve diğer tape'ler dikkate alınmaz.



Extensions of the Turing Machines

Örnek: 2-tape bir kopyalama makinesi $M=(K, \Sigma, \delta, s, H)$ şeklinde tanımlansın. $w \in \{a, b\}^*$ için $\triangleright \sqcup w \sqcup$ string'ini $\triangleright \sqcup w \sqcup w \sqcup$ şekline dönüştürür.

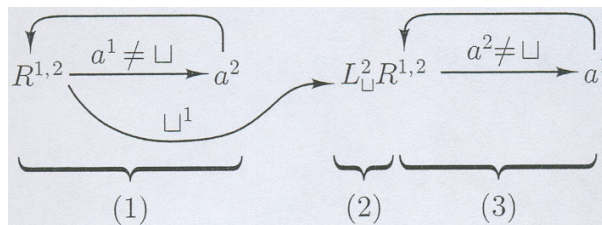
- (1) Her iki tape üzerinde sağa git ve 1.tape'deki her sembolü 2.tape'e kopyala.
- (2) 2.tape üzerinde en soldaki blank sembolüne git.
- (3) Her iki tape üzerinde sağa git ve 2.tape'deki her sembolü 1.tape'e kopyala.



Extensions of the Turing Machines

Örnek: (devam)

Başlangıç: 1.tape $\triangleright \sqcup w$ (2) bitince: 1.tape $\triangleright \sqcup w \sqcup$
 2.tape $\triangleright \sqcup$ 2.tape $\triangleright \sqcup w$
 (1) bitince: 1.tape $\triangleright \sqcup w \sqcup$ (3) bitince: 1.tape $\triangleright \sqcup w \sqcup w \sqcup$
 2.tape $\triangleright \sqcup w \sqcup$ 2.tape $\triangleright \sqcup w \sqcup$



- $L_{\sqcup}^1$ 1.tape üzerinde sola doğru boşluk sembolü arar, $\sqcup^2$ 2.tape üzerine boşluk yazar, $R^{1,2}$ 1. ve 2.tape üzerinde bir sağa geçer. a^1 etiketi 1.tape'den a okunduğunda yapılan işlemi gösterir.



Extensions of the Turing Machines

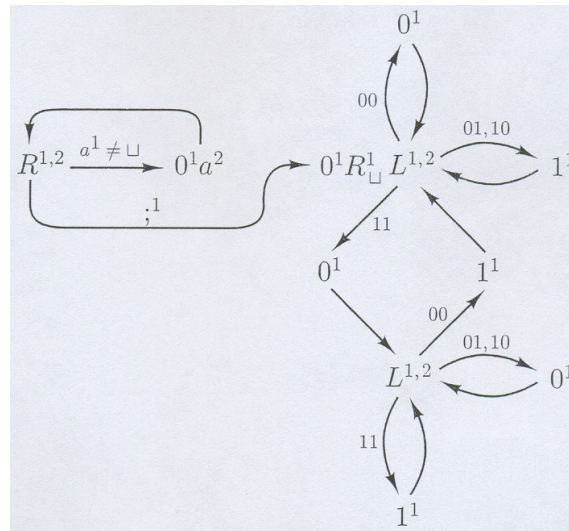
Örnek: 2-tape iki binary sayıyı toplayan Turing makinesi $M=(K, \Sigma, \delta, s, H)$ aşağıdaki şekilde oluşturulabilir. 01 etiketi 1.tape üzerinde 0, 2.tape üzerinde 1 olduğunu ifade eder.

- İki binary sayı 1.tape üzerindedir ve aralarında ; sembolü vardır.
- Makine önce birinci sayıyı 2.tape üzerine kopyalar ve yerine 0 yazar.
- 1.tape üzerinde ikinci sayı vardır ve önünde 0 lar bulunur.
- İki binary sayı sağdan (least significant bit) sola doğru toplanarak sonuç 1.tape üzerine yazılır.
- Sonlandığı durum carry bit bilgisinide gösterir.



Extensions of the Turing Machines

Örnek: (devam)





Extensions of the Turing Machines

Two-way Infinite Tape:

- *Tape birimi iki yönlü sonsuz uzunluğa sahiptir.*
- *Sembolünün bir anlamı olmaz ve kullanılmaz.*
- *Head başlangıçta giriş string'inin başındadır.*

Multiple Heads:

- *Bir tape birimine sahiptir ancak çok sayıda head bulunur.*
- *Bazı durumlarda bu tür Turing makinesi bir işi çok basit şekilde yapar.*

Two-Dimensional Tape:

- *İki boyutlu sonsuz uzunluğa sahip bir tape birimine sahiptir.*



Ödev

- Problemleri çözünüz 4.1.1a (sayfa 191)
- Problemleri çözünüz 4.1.2a (sayfa 191)
- Problemleri çözünüz 4.1.3 (sayfa 192)
- Problemleri çözünüz 4.2.1 (sayfa 200)
- Problemleri çözünüz 4.3.5 (sayfa 209)