Hafta 4

Otomata Çeşitleri DFA NFA modellerinin karşılaştırılması

Biçimsel Diller de kullanılan bazı yapılar

Otomata çeşitleri:

Finite Automata;

Pushdown Automata: Yığıtlı Tek Yönlü Hafıza

Turing Makinesi: Random Accses Memory (Çift Yönlü Hafıza)

Problemin hesaplanabilir bir model olabilmesi için Turing makinesi ile hesaplanabilir olması gerekir.

Diller: Karakter katarları kümesidir.

Katar (String): Bir alfabe üzerinde tanımlı karakter (letter) dizisidir.

Örneğin Alfabe $\Sigma = \{a, b\}$ şeklinde olsun. Bu alfabede tanımlanabilecek katar(string) aşağıdaki gibi olabilir:

$$\begin{aligned} \text{strings}: & \ a \\ & \ ab \\ & \ abba \\ & \ baba \end{aligned}$$

λ sıfır katardır. (Boş String)

$$\lambda w = w\lambda = w$$

$$\lambda abba = abba\lambda = abba$$

Üs işlemi:

$w^n = wwww....w$

$\quad n \text{ adet } w \text{ olmalıdır}$

$(abba)^2 = abbaabba$

$w^0 = \lambda$

$\{a,b\}^3 = \{a,b\}\{a,b\}\{a,b\}$

Bu yapıda oluşturulabilecek stringler aşağıdaki gibidir;

$\{aaa, aab, aba, abb, baa, bab, bba, bbb\}$

Örneğin

$L = \{a, bb\}^*$

*dan dolayı üs 0 olabilir bundan dolayı λ dır.

üssün 1 olması durumunda $\{a, bb\}$

üssün 2 olması durumunda $\{a, bb\}\{a, bb\}$

sistem çıkışı $\{aa, abb, bba, bbbb\}$

üssün 3 olması durumunda $\{a, bb\}\{a, bb\}\{a, bb\}$

sistem çıkışı $\left\{\begin{matrix} aaa, aabb, abba, abbbb, bbaa \\ bbabb, bbbba, bbbbbb \end{matrix}\right\}$

Kleene (Yıldızı) İşlemi;

$\Sigma^* = \Sigma$ alfabesi üzerinde tanımlı olası bütün katarlar(stringler) kümesidir.

$\Sigma = \{a, b\}$ ise

$\Sigma^* = \{\lambda, a, b, aa, bb, ba, ab, bb, aaa, bbb, ababab, ...\}$

Dilin veya ifadenin + üs ile kapanması

$\sum^{*} = \sum$ alfabesi üzerinde tanımlı olası
λ dışındaki tüm katarlar kümesidir.

$\sum = \{a, b\}$ ise
$\sum^{+} = \{a, b, aa, bb, ba, ab, bb, aaa, bbb, abababa, ...\}$
$\sum^{+} = \sum^{*} - \lambda$

Bazı Örnekler ve Tanımalar

Örnek 1: $L = \{a^n b^n : n \geq 0\}$ bu dil nasıl tanımlanır.

Dilde tanımlanacak olan sitringlerde eşit sayıda a ve b ler olmalıdır.

$\lambda, ab, aabb, aaabbb, aaaabbb......$

Bir string (katarın) tersinin alınması

Tanım : $L^R = \{w^R : w \in L\}$

Örnek : $\{ab, aab, baba\}^R = \{ba, baa, abab\}$

$L = \{a^n b^n ; n \geq 0\}$
$L^R = \{b^n a^n ; n \geq 0\}$

Bitiştirme veya birleştirme

$L_{1,2} = \{xy : x \in L_1, y \in L_2\}$
Örneğin $L_1 = \{a, ab, ba\}$ $L_2 = \{b, aa\}$
$\{ab, aaa, abb, abaa, bab, baaa\}$

Örnek 2=

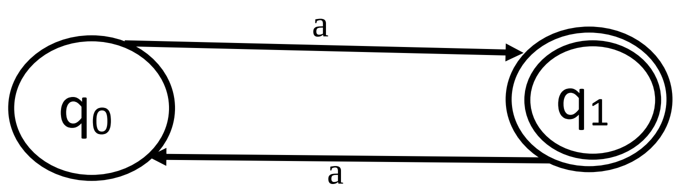
$L = \{ a^{2n+1} \mid n \geq 0 \}$ için DFA tasarlayalım.

$n = 0 \Rightarrow a$

$n = 1 \Rightarrow aaa$

$n = 2 \Rightarrow aaaaa$

Tek sayıda a içeren stringleri tanıyan makinedir.



Şekil 1 a da görüldüğü gibi;

Hücrelerden oluşan ve her hücresinde bir giriş simgesi bulunan bir mıknatıslı şerit. Yalnız okunabilen şeridin sağ ucu sonsuzdur. Başlangıçta şerit üzerinde giriş simgelerinde oluşabilen bir dizgi kayıtlıdır.

Bir sonlu denetim birimi (SDB). Sonlu sayıda durumu vardır. Bu durumlardan biri başlangıç durumdur ve SDB başlangıçta bu durumda bulunur.

Bir okuma kafası şeridinin okunması soldan sağa doğru tek yönlü gerçekleşir. Belirli bir anda okuma kafası şeridin hücrelerinden biri üzerinde bulunur ve üzerinde bulunduğu kayıtlı simgeyi okuyabilir.

Şekil1 de verilen DFA makinesi aşağıdaki gibi çalışır:

Başlangıçta şerit üzerinde bir giriş dizgisi kayıtlıdır ve okuma kafası şeridin ilk (en soldaki) hücresi üzerindedir.

Makinenin her hareketi aşağıdaki gibi yapılır:

- Şerit bir simge okunur
- Okuma kafası bir sağdaki hücreye geçer
- Sonlu denetim birimi bir sonraki duruma geçer

Belirli sayıda hareket okuma kafası belirli bir hücrenin üzerinde sonlu denetim birimi ise belirli bir durumda (q_i) bulunur. Okuma kafasının üzerinde bulunduğu hücrenin solundaki hücrelerde yer alan dizgiyi w olarak adlandıralım.

Eğer q_i bir uç durum ise makine w'yi tanır.

Deterministik ve Deterministik Olmayan Sonlu Özdevinirlerin Denkliği

Eğer her iki model ile tanımlanabilen kümelr sınıfı aynı ise bu iki model denktir. Yok eğer modellerden biri ile tanınabilen ancak diğeri ile tanımlanamayan kümeler var ise bu iki model denk değildir.

Tanımlarına göre deterministik olmayan model deterministik modelide kapsadığına başka bir deyişle her DFA aynı zamanda bir NFA olduğuna göre deterministik bid FA tarafından tanınan ancak hiçbir NFA tarafından tanınmayan bir küme

Örnek:

$M_{1,6} =< Q, \sum, \delta, q_0, F >$

$Q : \{A, B, C\}$

$\sum : \{0,1\}$

$q_0 : A$

$F : \{C\}$

$\delta : \delta(A,0) = \{A\}$

$\delta(A,1) = \{B,C\}$

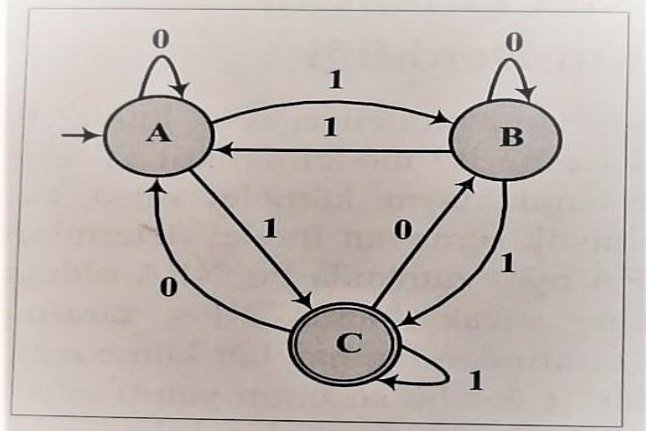
$\delta(B,0) = \{B\}$

$\delta(B,1) = \{A,C\}$

$\delta(C,0) = \{A,B\}$

$\delta(C,1) = \{C\}$

Deterministik modele göre tanımlanmış bu makinenin geçiş çizeneği aşağıdaki gibidir.



a) Deterministik Olmayan Geçiş Çizeneği

Başlangıç durumu A tek uç durum da C olduğuna göre verilen bir dizginin M makinesi tarafından tanınması için A ve C uç durumları arasında bu dizgiye karşı gelen bir yolun bulunabilmesi gerekir. Bu işlemin sistemli bir şekilde yapılmasını sağlayacak bir yöntem buluncaka olursa.

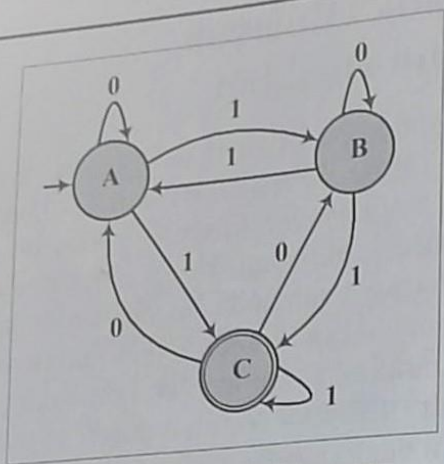
Aşağıdaki tablo elde edilir:

	0	1
→A	A	BC
B	B	AC
C	AB	C

b) Geçiş Çizelgesi

Geçiş çizelgesi geçişçizeneği ile aynı bilgiyi içeren ve makinin her durumundan her giriş simgesi ile hangi durumlara gidilebileceğini gösteren bir çizelgedir. Diğer bir ifade ile geçiş çizelgesi her durumun her girş simgesine karşı gelen ardıllarını kolaylıkla bulmayı sağlar.

Ancak verile bir w giriş dizgisinin M makinesi tarafından tanınabilmesi için başlangıç durumu olan A’nın w ardılları arasında tek uç durum C nin yer alması gerekir.



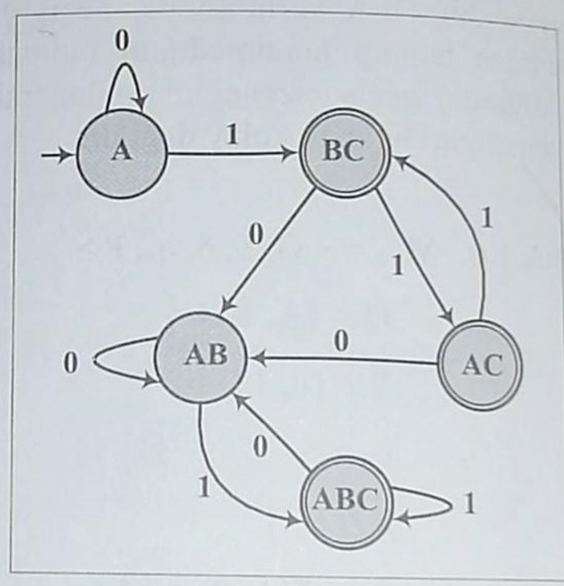
a) Deterministik Olmayan Geçiş Çizeneği

	0	1
→A	A	BC
B	B	AC
C	AB	C

b) Geçiş Çizelgesi

	0	1
→A	A	BC
BC	AB	AC
AB	AB	ABC
AC	AB	BC
ABC	AB	ABC

c) Başlangıç Durumunun Ardılları Çizelgesi



d) Deterministik Geçiş Çizeneği

Oluşturulan yeni makinenin biçimsel tanımı aşağıdaki gibidir:

$$M_{1,6} = < Q, \sum, \delta, q_0, F >$$

$$Q : \{A, BC, AB, AC, ABC\}$$

$$\sum : \{0,1\}$$

$$q_0 : A$$

$$F : \{BC, AC, ABC\}$$

$$\delta : \delta(A,0) = A$$

$$\delta(A,1) = BC$$

$$\delta(BC,0) = AB$$

$$\delta(BC,1) = AC$$

$$\delta(AB,0) = AB$$

$$\delta(AB,1) = ABC$$

$$\delta(AC,0) = AB$$

$$\delta(AC,1) = BC$$

$$\delta(ABC,0) = AB$$

$$\delta(ABC,1) = ABC$$

Görüldüğü gibi iki makinede birbirine denk makinelerdir. Deterministik makinede A,B,C gibi durumlar yeniden adlandırılarak q0,q1,q2… gibi adlandırılabilir.