BÖLÜM 1
SONLU ÖZDEVİNİRLER

1.1.Sonlu Özdevinir (FA)

Sonlu özdevinir (finite automata : FA) modeli, kesikli giriş ve çıkışları olan matematiksel bir modeldir. Bu model birçok donanım ve yazılım sisteminin matematiksel bir modelidir. Bu metin düzenleyici (text editör) ve derleyicilerin belirli kesimleri ile zaman uyumlu dizsel devreler (synchronous sequential circuits) sonlu özdevinir modeline uygun yazılım ve donanım sistemlerine örnek olarak verilebilir. Hatta sayısal bilgisayarlar bile bu model e uygun sistemler olarak görülebilir. Dolayısıyla bilgisayar bilimleri ve mühendisliği için sonlu özdevinir modeli, çok kullanılan önemli bir modeldir.

Sonlu özdevinir için bu kitapta, modelin İngilizce adının baş harflerinden oluşan ve yaygın biçimde kullanılan FA kısa adı kullanılacaktır.

Sonlu özdevinir modelinin birçok türü vardır. Sonlu özdevinirleri öncelikle:

A. Sonlu durumlu tanıyıcı (finite state recognizer)
B. Çıkış üreten özdevinir

modelleri olarak sınıflandırmak mümkündür. Bu sınıfların her birinin de kendi içinde altsınıfları vardır. Ancak kısaca “sonlu özdevinir” ya da FA denilince, temel model olan “deterministik sonlu durum tanıyıcı” modeli anlaşılır.

1.1.1.Deterministik Sonlu Özdevinir (DFA) Modeli

Temel modelde, deterministik sonlu özdevinir bir beşli olarak tanımlanır.

$DFA=<Q, \sum, \delta, q_0, F>$

Q: Sonlu sayıda durum içeren Durumlar Kümesi

$\sum$: Sonlu sayıda giriş simgesinden oluşan Giriş Alfabesi

$\mathbf{q_0}$: Başlangıç durumu ($q0 \in Q$)

Başlangıç durumu durumlar kümesinin bir elemanı olduğuna göre Q boş olmayan (içinde en az q0 bulunan) bir kümedir.

F: Uç durumlar kümesi

Durumlar kümesinin bir altkümesidir. $F \subseteq \mathbf{Q}$

δ: Durum geçiş işlevi (state transition function)

Durum geçiş işlevi ya da kısaca geçiş işlevi, $(Q \times \sum)$'dan Q'ya bir eşleşme oluşturur.

Temel modelde geçiş işlevi ile her durum (durum, giriş simgesi) çiftine bir (durum) eşlenmektedir. Bu durumlardan ilki "şimdiki durum", ikincisi ise "sonraki durum" olarak adlandırılırsa, durum geçiş işlevi ile her (şimdiki durum, giriş simgesi) çiftine bir (sonraki durum) eşlendiği söylenebilir. Matematiksel olarak geçiş işlevi:

$$\delta(q_i, a) = q_j \qquad \sqrt{} q_i \in Q, \; a \in \Sigma \Rightarrow q_j \in Q$$

olarak tanımlanır.

Yukarıdaki tanımdan anlaşıldığı gibi, temel FA modeli deterministik bir modeldir. Bu model kısaca DFA (*deterministik finite automata*) olarak adlandırılır. Kısaca FA (*finite automata*) denildiğinde de DFA anlaşılır. Bu modele göre, FA q0 durumundan başlar ve uygulanan her giriş simgesi yeni bir duruma geçer. Her an FA'nın bulunduğu durum kesin olarak bellidir.

Örnek 1.1.

$M_{1.1}$ = DFA=<Q, ∑, δ, q_0, F>

$Q = \{q_1, q_2, q_3\}$

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

$F = \{q_2\}$

δ: $\delta(q_0, 0) = q_0$

$\delta(q_0, 1) = q_1$

$\delta(q_1, 0) = q_0$

$\delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_2$

$\delta(q_2, 1) = q_2$

Geçiş Çizeneği

Örnek 1.1’de görüldüğü gibi, geçiş işlevinin matematiksel tanımı hem uzun, hem de anlaşılırlığı zor olan bir tanımdır. Bu nedenle geçiş işlevinin tanımı için genellikle “geçiş çizeneği (*transition diagram*)” olarak adlandırılan bir çizenek kullanılır.

Yönlü bir çizge olan geçiş çizeneğinde her durum için bir düğüm bulunur. Durum geçişleri ise yönlü yayar ile gösterilir ve yayların üzerine geçişe neden olan giriş simgesi yazılır. Geçiş çizeneğinde başlangıç durumu “→” ile, uç durumlar ise çift çember ile gösterilir. Örnek 1.1’deki FA için oluşturulan geçiş çizeneği Çizim 1.1’de görülmektedir.

Çizim 1.1. $M_{1.1}$'in Deterministik Geçiş Çizeneği

DFA’nın Tanıdığı Dizgiler Kümesi

Tanımlanan modele göre, DFA’nın girişine her seferinde bir giriş simgesi uygulanır. Her giriş simgesi DFA’nın bir sonraki duruma geçmesine neden olur. DFA, bulunduğu durumu yeni bir giriş simgesi uygulanana kadar korur. Uygulanan giriş simgelerinin sayısı ne kadar çok olursa olsun deterministik olduğu için model çalışır. Başlangıçta q0 durumunda bulunan

DFA, belirli sayıda giriş simgesinden oluşan bir dizinin (string) uygulamasından sonra belli bir duruma ulaşır. Uygulanan giriş dizgisi w, bu dizgi uygulandıktan sonra DFA’nın ulaştığı durumu da q1 ile gösterelim. q0,w ve q1 arasındaki ilişkiyi durum geçiş işlevini kullanarak ve bu işlevin anlamını biraz genişleterek aşağıdaki gibi gösterebiliriz:

$$\delta(q_0, w) = q_i$$

q0 durumundan başlayan ve w giriş dizgisinin uygulanmasından sonra q1 durumuna ulaşan DFA, eğer qi bir uç durum ise w giriş dizgisini tanır. Giriş alfabesi ∑={I1,I2,...,Ik} olan bir DFA'ya uygulanabilecek giriş dizgileri kümesi sonsuz bir kümedir. Giriş alfabesindeki simgelerden oluşan sonlu ya da sonsuz uzunluktaki her dizgi bu kümede yer alır. Giriş alfabesindeki simgelerden oluşan dizgilerin bir kısmı DFA tarafından tanınan, diğer kısmı ise tanınmayan dizgilerdir. Bu aşamada, bir sonlu özdevinir (M) tarafından tanınan dizgiler kümesini aşağıdaki gibi tanımlamak mümkündür.

$$T(M) = \{ w \mid \delta(q_0, w) = q_i \in F \}$$

Buna göre DFA’ya uygulanabilecek dizgiler kümesi, DFA tarafından tanınan ve tanınmayan olmak üzere iki altkümeye ayrılmaktadır. DFA’nın tanıdığı küme, DFA’yı başlangıç durumundan bir uç duruma götüren dizgilerin kümesidir. Örnek 1.1’de tanımlanan DFA’yı (M1.1) incelersek, örneğin 0101 dizgisinin M1.1 tarafından taınmadığını; buna karşılık 0110 dizgisinin M1.1 tarafından tanındığını görürüz. Çünkü q0 durumundan başlaya M1.1, 0101 uygulandıktan sonra q1 durumuna ulaşır ve q1 bir uç durum değildir. Buna karşılık 0110 dizgisi M1.1’i başlangıç durumundan q2 durumuna götürür ve q2 bir uç durumdur. M1.1’in tanıdığı dizgiler kümesi sonsuz bir kümedir. Bu kümedeki dizgilerden birkaç örnek aşağıda gösterilmiştir.

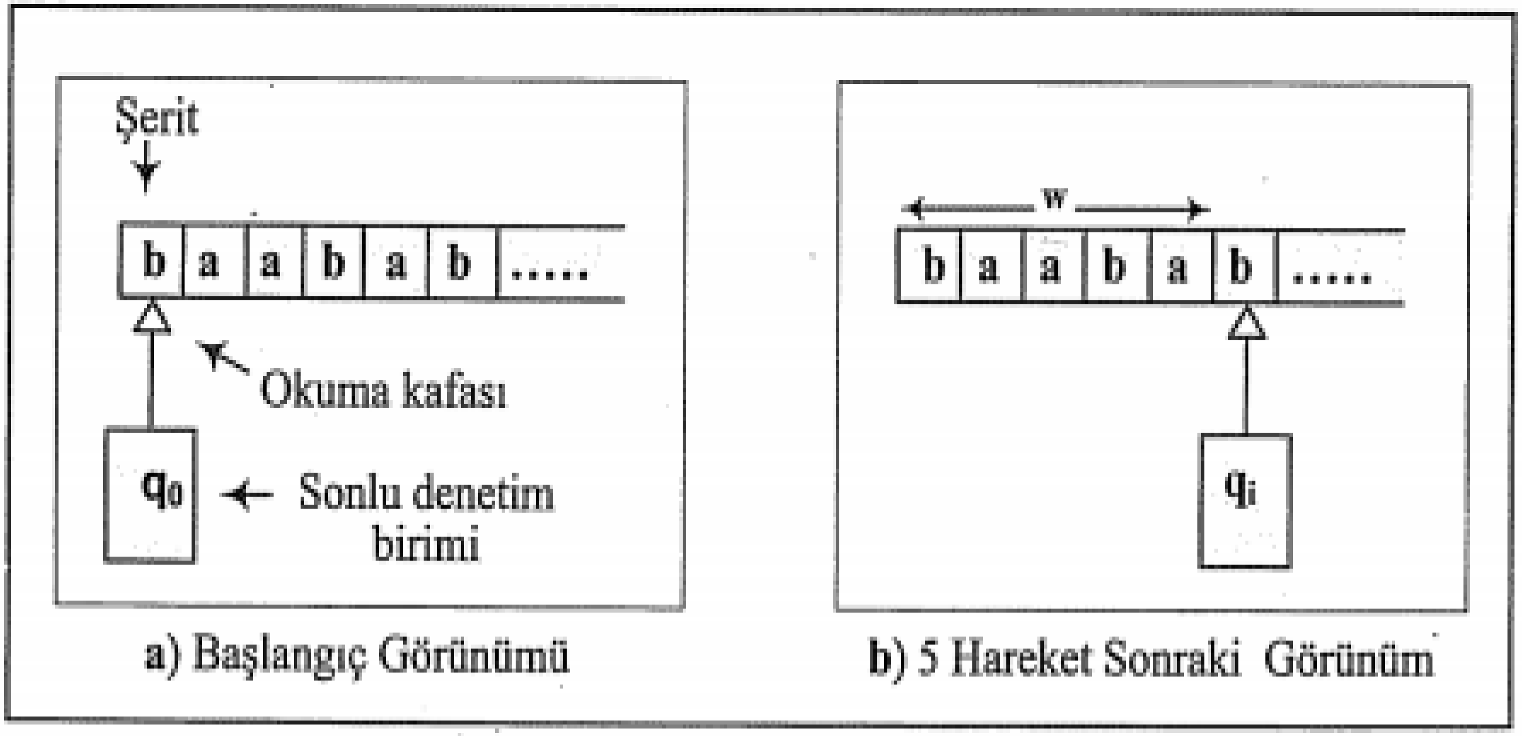
$$T(M_{1.1}) = \{11, 011, 110, 0110, 0110, 01011, \ldots\}$$

Çizim 1.1’deki geçiş çizeneği incelendiğinde, bir dizginin M1.1 tarafından tanınabilmesi için dizgide art arda iki tane 1 bulunması gerekli ve yeterli olduğu görülür. Buna göre T(M1.1) sözlü olarak,{0,1} alfabesinde, içinde 11 alt dizgisi bulunan dizgiler kümesi olarak tanımlanabilir.

DFA’nın Şeritli Makine Modeli

Tanımlandığı biçimiyle deterministik sonlu özdevinir modeli matematiksel bir modeldir. Deterministik bir sonlu özdevinir oluşturmak için bir durumlar kümesi ile bir giriş alfabesi oluşturmak, durumlardan birini başlangıç durumu olarak seçmek, durumların bir kesimini uç (tanıyan) durumlar olarak nitelemek ve bir geçiş işlevi tanımlamak yeterlidir. Oluşturulan DFA soyut bir makine olarak görülebilir ve kısaca “makine” olarak adlandırılır. Ancak “makine” adlandırmasının tamamen simgesel olduğu, oluşturulan sonlu özdevinirin makine ile hiç benzerliği olmayan bir şey de olabileceği unutulmamalıdır.

Soyut bir makine olan deterministik sonlu özdevinirin daha kolay anlaşılması için, somut bir şeritli makine modeli düşünülebilir. Bu amaçla, DFA için kullanılan somut makine modeli, bilinen bileşenlerden oluşan ve Çizim 1.2’de görülen bir modeldir.



Çizim 1.2. DFA'nın Şeritli Makine Modeli

Çizim 1.2. DFA’nın Şeritli Makine Modeli Başlangıç Görünümü

Çizim 1.2. DFA’nın Şeritli makine Modeli 5 Hareket Sonraki Görünüm

Çizim 1.2.a’da görüldüğü gibi DFA’nın makine modeli 3 elemandan oluşmaktadır.

- Hücrelerden oluşan ve her hücresinde bir giriş simgesi bulunan bir mıknatıslı şerit. Yalnız okunabilen şeridin sağ ucu sonsuzdur. Başlangıçta şerit üzerinde giriş simgelerinden oluşan bir dizgi kayıtlıdır.
- Bir sonlu denetim birimi(SDB).SDB’nin sonlu sayıda durumu vardır. Bu durumlardan biri başlangıç durumudur ve SDB başlangıçta bu durumda bulunur.
- Bir okuma kafası. Şeridin okunması soldan sağa doğru tek yönlü gerçekleşir. Belirli bir anda okuma kafası şeridin hücrelerinden biri üzerinde bulunur ve üzerinde bulunduğu hücrede kayıtlı simgeyi okuyabilir.

Çizim 1.2’deki modele göre, DFA (makine) aşağıda açıklanan biçimde çalışır:

- Başlangıçta şerit üzerinde bir giriş dizgisi kayıtlıdır ve okuma kafası şeridin ilk (en soldaki) hücresi üzerindedir.
- Makinenin her hareketinde aşağıdaki işlemler yapılır:
    1. Şeritten bir simge okunur.
    2. Okuma kafası bir sağdaki hücreye geçer.
    3. Sonlu denetim birimi bir sonraki duruma geçer.
- Belirli bir sayıda hareket sonunda okuma kafası belirli bir hücrenin üzerinde, sonlu denetim birimi ise belirli bir durumda (qi) bulunur. Okuma kafasının üzerinde bulunduğu hücrenin solundaki hücrelerde yer alan dizgiyi w olarak adlandıralım. Eğer qi bir uç durumsa makine w’yi tanır(Çizim 1.2.b).

1.1.2.Deterministik Olmayan Sonlu Özdevinir (NFA) Modeli

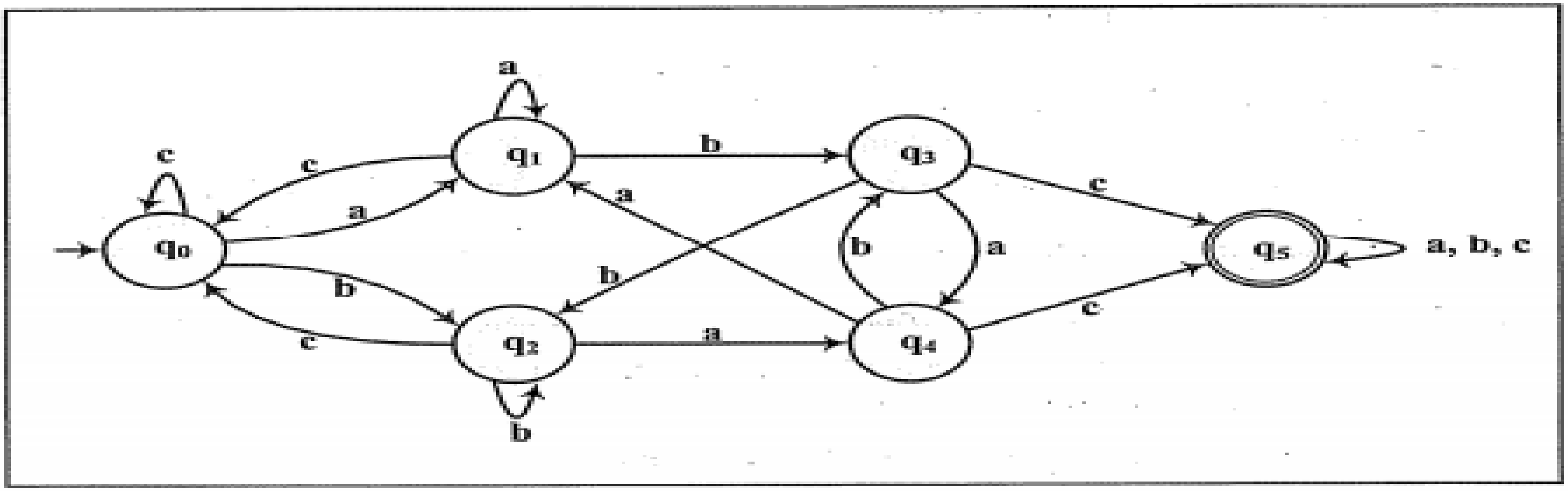
Sözlü olarak ya da bir düzgün deyimle (bkz 2.2.) tanımlanan belirli bir dizgiler kümesini tanıyan DFA'nın deterministik geçiş çizeneğinin oluşturulması oldukça zor olabilir.

Örnek 1.2. "{a,b,c} alfabesinde, içinde abc ve bac alt dizgilerinden en az biri, en az bir kez bulunan dizgiler kümesi'ni düşünelim ve bu kümeyi tanıyan DFA'yı $M_{1.2}$ olarak adlandıralım.

$M_{1.2}$'nin tanıdığı dizgilerden birkaç örnek aşağıda gösterilmiştir.

$T(M_{1.2})$={abc, bac, cbabcb, bccaabcbac,…}

$M_{1.2}$'nin deterministik geçiş çizeneği ise Çizim 1.3'de görülmektedir. $M_{1.2}$ 'nin tanıdığı kümenin sözlü tanımı oldukça kolay olmasına karşın, geçiş çizeneğinin oldukça karmaşık olduğu görülmektedir. Üstelik sözlü tanımdan hareketle bu çizeneğin elde edilmesini sağlayan sistematik bir yöntem de yoktur.



Çizim 1.3. $M_{1.2}$'nin Deterministik Geçiş Çizeneği

Deterministik modelin kullanım güçlüğü nedeniyle, kullanımı daha kolay ve daha esnek bir model olan deterministik olmayan(*non deterministic*) model geliştirilmiştir. Deterministik olmayan sonlu özdevinir (*non deterministic finite automata* : NFA) aşağıdaki gibi tanımlanır:

$$NFA = \langle Q, \sum, \delta, q_0, F \rangle$$

Bu tanımda Q,∑,q0 ve F'nin anlamları deterministik model tanımındakilerle aynıdır. Deterministik modelde olduğu gibi deterministik olmayan model de Q durumlar kümesini, ∑ giriş alfabesini, q0 başlangıç durumunu, F ise uç durumlar kümesini göstermektedir. İki model arasındaki tek fark geçiş işlevinin tanımındadır. Deterministik modelde geçiş işlevi:

- (Q x ∑)'dan Q'ya

bir eşleşme olarak tanımlanırken, deterministik olmayan modelde geçiş işlevi:

- (Q x ∑)'dan Q'nun altkümelerine

bir eşleşme olarak tanımlanır. Buna göre deterministik modelde her durumdan her giriş simgesi ile bir ve yalnız bir duruma geçilirken, deterministik olmayan modelde bir durumdan bir giriş simgesi ile geçilebilecek durum sayısı sıfır, bir ya da birçok olabilmektedir.

Örnek 1.3. $M_{1.3}$ =<Q, $\sum$, δ, q_0, F>

$Q = \{ q_0, q_1, q_2, q_3\}$

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

$F = \{ q_2\}$

$\delta = \delta\{q_0, 0\} = \{ q_0, q_1\}$

$\delta\{q_0, 1\} = \{ q_0, q_2\}$

$\delta\{q_1, 0\} = \{ q_3\}$

$\delta\{q_1, 1\} = \{ \} = \phi$

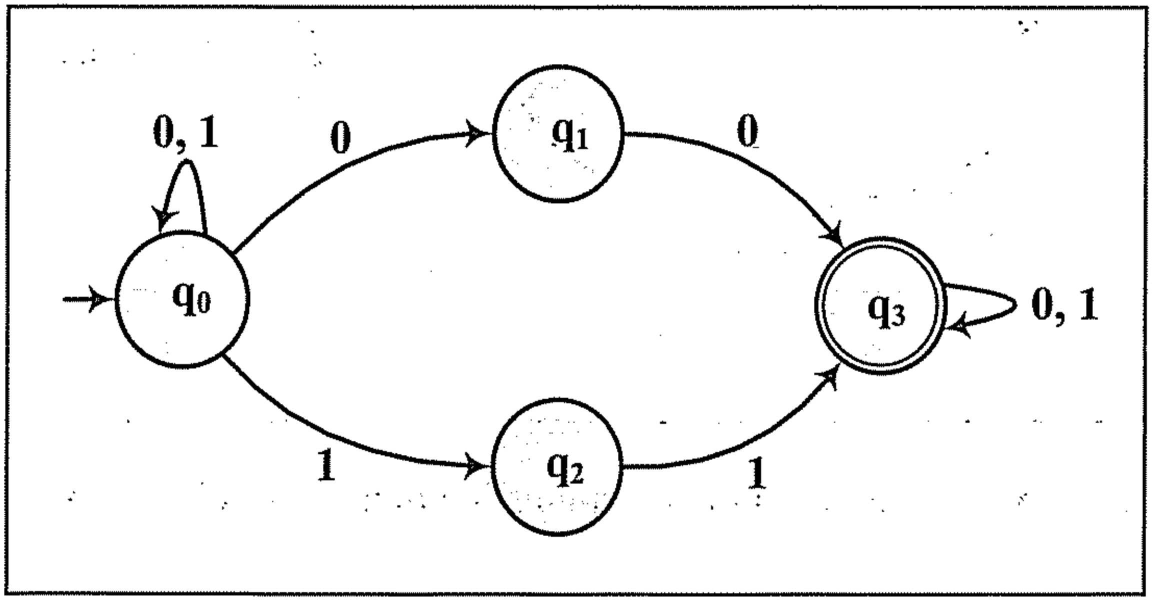
$\delta\{q_2, 0\} = \phi$

$\delta\{q_2, 1\} = \{q_3\}$

$\delta\{q_3, 0\} = \{q_3\}$

$\delta\{q_3, 1\} = \{q_3\}$

$M_{1.3}$'ün tanımında görüldüğü gibi, deterministik olmayan modelde bazı durumlarda bazı giriş simgeleri ile birden çok duruma geçilebilmektedir. Bazı durumlardan bazı giriş simgeleri ile de hiçbir duruma geçilemememektedir. Bu nedenle, deterministik modelde, başlangıç durumu ve uygulanan giriş dizgisi bilindiğinde makinanın hangi durumda bulanacağı kesin olarak bellidir. Buna karşılık, deterministik olmayan modelde başlangıç durumu ve uygulanan giriş dizgisi bilinen makinenin son durumu belirsiz olabilir. Örneğin başlangıç durumu q0 olan ve $M_{1.3}$'e w=000 giriş dizgisi uygulandığında, makinenin son durumu q0, q1 ve q3 durumlarından herhangi biri olabilir.



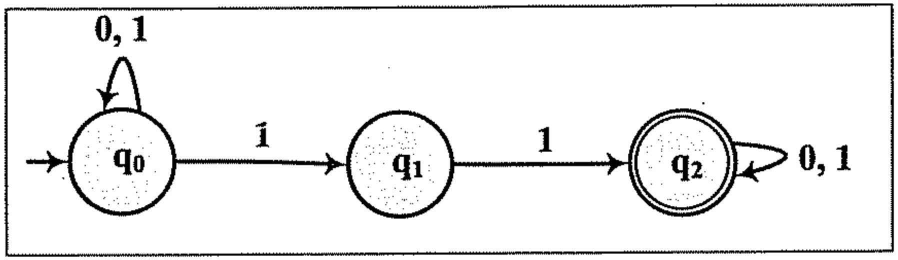
Çizim 1.4. $M_{1.3}$'ün Deterministik Olmayan Geçiş Çizeneği

Çizim 1.4.M1.3’nin Deterministik Olmayan Geçiş Çizeneği

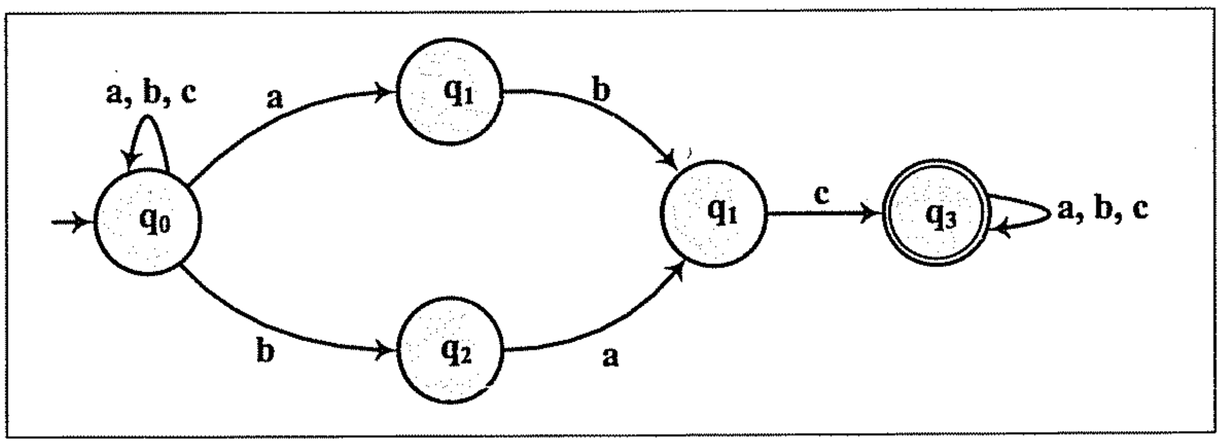
Deterministik olmayan bir sonlu özdevinirin(NFA) bir giriş dizgisini tanıması için, geçiş çizeneğinde, başlangıç durumundan başlayarak bu giriş dizgisi ile bir uç duruma ulaşan bir yol bulunabilmesi gerekir. Örneğin $M_{1.3}$, w=10001 giriş dizgisini tanır. Çünkü bu giriş dizgisine karşı gelen ve q_0'dan başlayarak tek uç durum olan q_3'te son bulan bir yol

$$\left(q_0 \rightarrow q_0 \rightarrow q_1 \rightarrow q_3 \rightarrow q_3 \rightarrow q_3 \text{ ya da } q_0 \rightarrow q_0 \rightarrow q_0 \rightarrow q_1 \rightarrow q_3 \rightarrow q_3\right)$$

bulunabilmektedir. Buna karşılık $M_{1.3}$, w=010 giriş dizgisini tanımaz. Çünkü q_0 ve q_3 durumları arasında bu giriş dizgisine karşı gelen bir yol bulmak mümkün değildir. Deterministik olmayan geçiş çizeneklerinin oluşturulması ve okunması deterministik çizeneklere göre daha kolaydır. Örneğin Çizim 1.4 incelendiğinde, $M_{1.3}$'ün tanıdığı kümenin "{0,1} alfabesinde, içinde 00 ve 11 alt dizgilerinden en az biri, en az bir kez bulunan dizgiler kümesi" olduğu kolaylıkla bulunabilir. Oysa Çizim 1.3 incelenerek, $M_{1.2}$'nin tanıdığı kümeyi bulmak, başka bir deyişle Çizim 1.3'teki deterministik geçiş çizeneğinin oluşturmak hiç de kolay değildir. Bir karşılaştırma olanağı sağlamak için Çizim 1.5'te $M_{1.1}$ ve $M_{1.2}$'nin deterministik olmayan geçiş çizenekleri verilmiştir.



a) $M_{1.1}$



b) $M_{1.2}$

Çizim 1.5. $M_{1.1}$ ve $M_{1.2}$ 'nin Deterministik Olmayan Geçiş Çizenekleri

Çizim 1.5.a.$M_{1.1}$ ve $M_{1.2}$'nin Deterministik Olmayan Geçiş Çizeneği

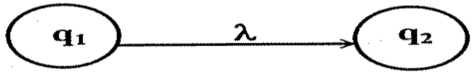
Çizim 1.5.b.$M_{1.1}$ ve $M_{1.2}$'nin Deterministik Olmayan Geçiş Çizeneği

1.1.3.Lambda(λ) Geçişi

Sonlu özdevinirlerde durum geçişleri, giriş simgelerinin işlenmesi ile gerçekleşir. Makine belli bir durumdayken, bir geçiş simgesinin uygulanması makinenin bir sonraki duruma geçmesine neden olur. Geçiş işleviyle tanımlanan “sonraki durum” deterministik modelde tek bir durumdur. Deterministik olmayan modelde ise “sonraki durum” durumlar kümesinin bir altkümesi olup bu altkümede sıfır, bir ya da birçok durum bulunabilir. Şimdiye kadar yapılan tanımlara göre, DFA ya da NFA modeline uygun bir makine, giriş simgesi uygulanmadığı sürece bulunduğu durumu korur. Lambda geçişi ile deterministik olmayan modelle genişletilen ve kullanımı kolaylaştırılan sonlu özdevinir tanımı daha da genişletilir. Lambda ve lambda geçişi soyut kavramlardır. Lambda bir boş simge olarak düşünülebilir. Lambda geçişi ise, hiçbir giriş simgesi uygulanmadan (ya da işlenmeden) gerçekleşen durum geçişine karşı gelir. Bir makinenin q1 ve q2 durumları arasında

$$\boldsymbol{\delta(q_1, \lambda) = q_2}$$

biçiminde tanımlanan λ-geçişi varsa, q1 durumunda bulunan makinenin, hiçbir giriş simgesi işlenmeden, kendiliğinden q2 durumuna geçebileceği anlaşılır.



Lambda geçişi, modelin esnekliğini arttıran; geçiş çizeneklerinin daha kolay oluşturulmasını ve okunmasını sağlayan soyut bir kavramdır. Diğer geçiler gibi, lambda geçişi de tek yönlüdür. q1 ve q2 durumları arasındaki, q1'den q2'ye λ-geçişi, hiçbir giriş simgesi uygulanmadan makinenin q1 durumundan q2 durumuna geçebileceğini gösterir. Buna göre, q1 durumunda bulunan makinenin aynı zamanda q2 durumunda da bulunduğu anlamı çıkar. Ancak bunun tersi doğru değildir.

q1 ve q2 durumları arasında, q1'den q2'ye λ-geçişi varsa işlevsel olarak:

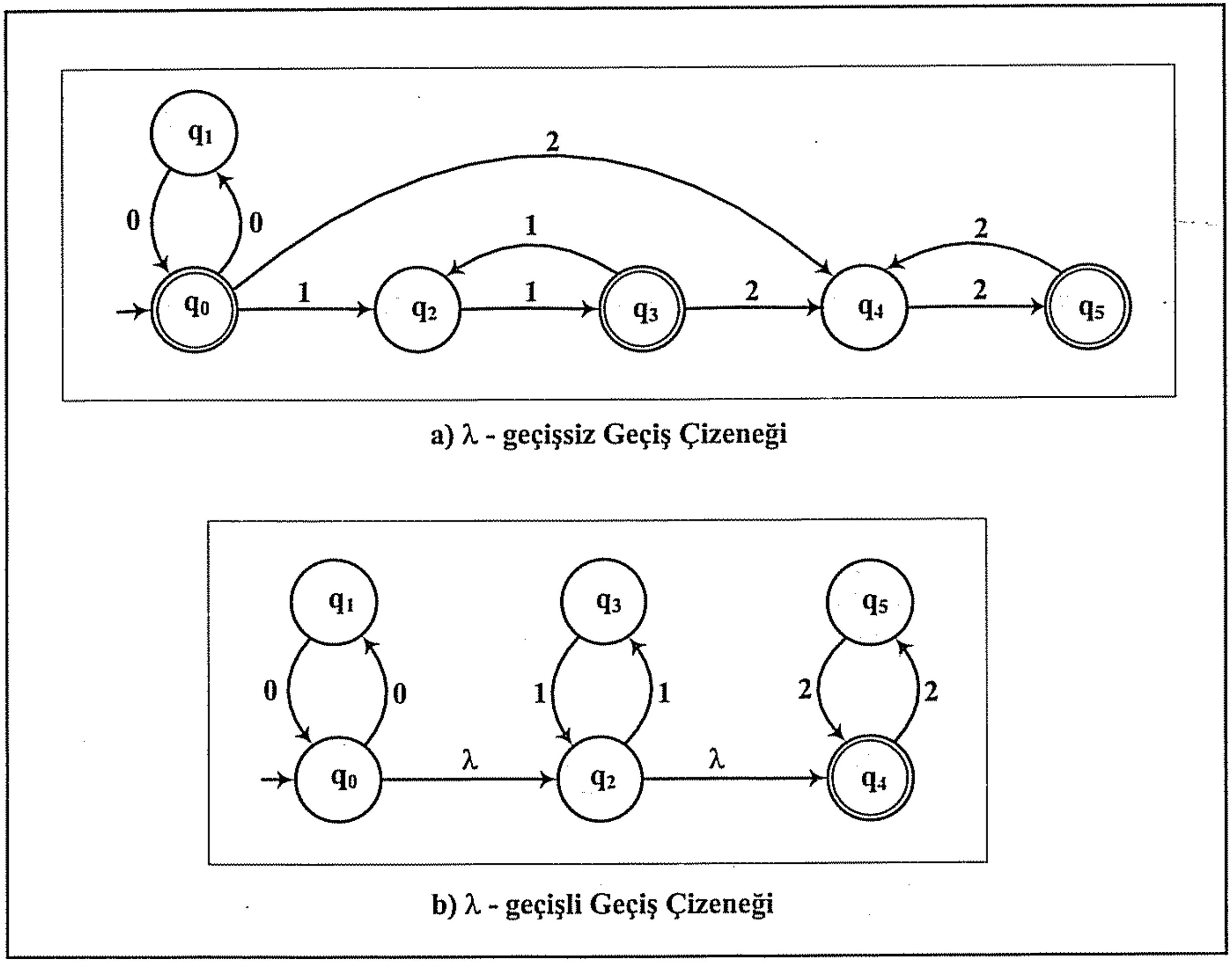
- eğer q1 başlangıç durumu ise q2 de başlangıç durumu,
- eğer q2 bir uç durum ise q1 de bir uç durum

niteliği kazanır.

Örnek 1.4. Lambda geçişlerinin sonlu özdevinir modeline sağladığı esnekliği göstermek için, tanıdığı küme {0,1,2} alfabesinde:

$$\mathbf{T(M_{1.4}) = \{0^{2n}\,1^{2m}\,2^{2k} \quad | \quad n \geq 0,\ m \geq 0,\ k \geq 0\}}$$

biçiminde tanımlanan makineyi düşünelim. M1.4’ün, λ-geçişi kullanmadan oluşturulan geçiş çizeneği Çizim 1.6.a’da, λ-geçişi kullanılarak oluşturulan geçiş çizeneği ise Çizim 1.6.b’de görülmektedir. λ-geçişsiz ve λ-geçişli geçiş çizenekleri incelendiğinde, λ-geçişlerinin sonlu özdevinir modeline kazandırdığı esneklik ve kullanım kolaylığı açık biçimde görülmektedir.



Çizim 1.6. $M_{1.4}$ için Geçiş Çizenekleri : a) λ-Geçişsiz b) λ- Geçişli

Çizim 1.6.a. $M_{1.4}$ için Geçiş Çizenekleri- λ Geçişsiz

Çizim 1.6.b. $M_{1.4}$ için Geçiş Çizenekleri- λ Geçişsiz

λ-Geçişsiz Eşdeğer Geçiş Çizeneğinin Bulunması

λ-geçişlerinin kullanımı, geçiş çizeneklerinin kullanımını kolaylaştırır ancak gücünü arttırmaz. Başka bir deyişle λ-geçişleri kullanılarak oluşturulan her geçiş çizeneğine denk, λ-geçişsiz bir geçiş çizeneği bulunabilir. İki geçiş çizeneğinin denkliği, tanıdıkları kümenin eşitliği anlamını taşımaktadır. Buna göre, λ-geçişli bir geçiş çizeneği tarafından tanınan, ancak λ-geçişsiz hiçbir geçiş çizeneği tarafından tanınmayan hiçbir dizgiler kümesi yoktur. Bu da, geçiş çizeneklerinde λ-geçişi kullanmanın, NFA modelinin ifade gücünde hiçbir değişiklik yapmadığını gösterir.

λ-geçişli bir geçiş çizeneği verildiğinde, λ-geçişlerini tek tek yok ederek eş değer bir geçiş çizeneği elde etmek mümkündür. Bu kapsamda, q1 ve q2 durumları arasında q1'den q2'ye λ-geçişi varsa:

- q2 durumundan başlayan her durum geçişine (√δ(q2,a)=qk) karşılık q1 durumundan başlayan ve aynı giriş simgesi ile aynı duruma ulaşan bir durum geçişi (δ(q1,a)=qk) eklendikten,

- eğer q1 durumu başlangıç durumu ise, q2 durumu da başlangıç durumu yapıldıktan ve

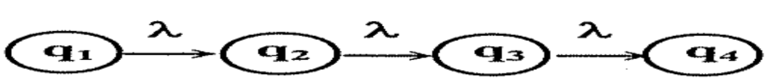
- eğer q2 durumu bir uç durum ise, q1 durumu da uç durum yapıldıktan sonra , λ-geçişi silinebilir.

Örnek 1.5. $M_{1.5}$ makinesi {a.b.c} alfabesinde, sıfır ya da iki tane a ile, ya da çift sayıda b (0,2,4,… tane b) ile başlayıp cc ile biten dizgiler kümesini tanıyan makine olsun. $M_{1.5}$'in tanıdığı kümedeki dizgilerden birkaç örnek aşağıda görülmektedir.

$T(M_{1.5})$={cc,aacc,bbcc,bbbbcc,bbbbbbcc,…}

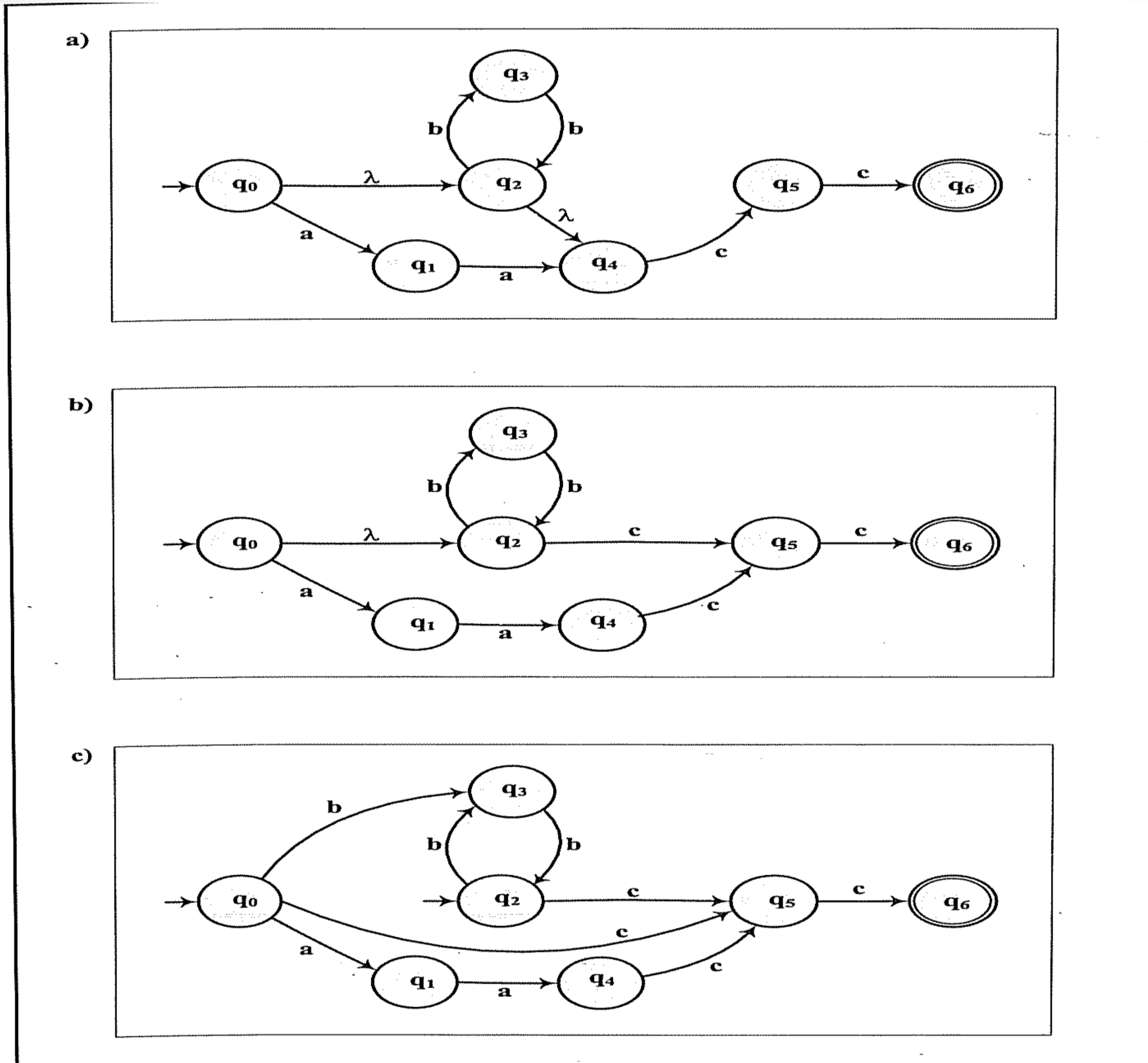
$M_{1.5}$ için oluşturulan λ-geçişli geçiş çizeneği Çizim 1.7.a'da görülmektedir. bu çizeneğe eş değer, λ-geçişsiz geçiş çizeneğini elde edebilmek için, önce q_2 ile q_4 arasındaki, sonra da q_0 ile q_2 arasındaki λ-geçişleri yok edilir. q_2 ile q_4 arasındaki λ-geçişini yok edebilmek için q_2 ile q_5 arasına c-geçişi eklenir ve Çizim 1.7.b'deki geçiş çizeneği elde edilir. q0 ile q_2 arasındaki λ-geçişini yok edebilmek için ise q_0 ile q_3 arasına b-geçişi, q_0 le q_5 arasına c-geçişi eklenip q_2 durumu başlangıç durumu yapılır ve Çizim 1.7.c'deki λ-geçişsiz geçiş çizeneği elde edilir.

Eğer λ-geçişleri yok edilerek geçiş çizeneğinde zincir yapısında birden çok λ-geçişi varsa, yok etme işlemini zincirin ucundan başlatarak geriye doğru sürdürmekte yarar vardır. Örneğin q_1 ile q_2; q_2 ile q_3; q_3 ile de q_4 arasında üç λ-geçişi varsa:



- önce q_3 ile q_4 arasındaki,
- sonra q_2 ile q_3 arasındaki,
- son olarak da q_1 ile q_2 arasındaki

λ-geçişi yok edilir.



Çizim 1.7. Geçiş Çizeneğinde λ-Geçişlerinin Yok Edilmesi

Çizim 1.7.a

Çizim 1.7.b

Çizim 1.7.c

1.1.4.Deterministik ve Deterministik Olmayan Sonlu Özdevinir Modellerinin Denkliği

Acaba deterministik ve deterministik olmayan özdevinir modelleri denk midir? Eğer her iki model ile tanınabilen kümeler sınıfı aynı ise bu iki model denktir. Yok Eğer modellerden biri ile tanınabilen ancak diğeri ile tanınamayan kümeler varsa bu iki model denk değildir. Tanımlarına göre deterministik Olmayan model deterministik modeli de kapsadığına başka bir deyişle her DFA aynı zamanda bir NFA olduğuna göre, deterministik bir FA tarafından tanınan ancak hiçbir NFA tarafından tanınmayan bir küm düşünülemez. Ancak NFA Tarafından tanınan her küme için bu kümeyi tanıyan bir DFA Bulmak mümkün müdür? Eğer bu sorunun yanıtı Evet ise deterministik ve deterministik olmayan modellerin denk olduğu söylenebilir.

Yukarıdaki sorunun yanıtını bir başka soruya yanıt arayarak bulmaya çalışalım. Bir dizgi ve bir DFA verildiğinde geçiş çizeneğini kullanarak bu dizginin bu DFA tarafından tanınıp tanımadığını bulmak kolaydır ancak bir dizgi ve bir NFA verildiğinde geçiş çizeneğini kullanarak bu dizginin bu NFA tarafından tanınıp tanınmadığı bulmak kolay değildir.

Örnek 1.6. $M_{1.6} = \langle Q, \sum, \delta, q_0, F \rangle$

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

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

$q_0 = A$

$F = \{C\}$

δ: $\delta(A, 0) = \{A\}$

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

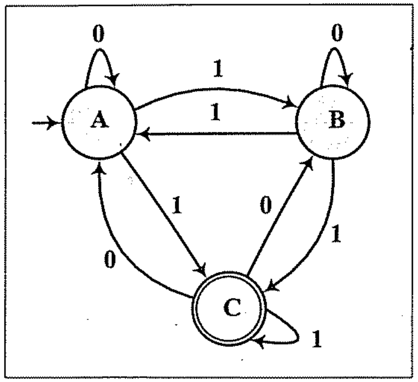
$\delta(B, 0) = \{B\}$

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

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

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

Örnek olarak $M_{1.6}$'yı düşünelim. Deterministik olmayan modele göre tanımlanan bu makinenin geçiş çizeneği Çizim 1.8.a'da görülmektedir. Başlangıç durumu A, tek uç durum da C olduğuna göre, verilen bir dizginin $M_{1.6}$ tarafından tanınabilmesi için A ve C durumları arasından, bu dizgiye karşı gelen bir yolun bulunabilmesi gerekir. Bu işlemin sistemli bir şekilde yapılmasını sağlayacak bir yöntem bulmaya çalışalım. Bu amaçla Çizim 1.8.b'de görülen geçiş çizelgesinden yararlanabiliriz. Geçiş çizelgesi, geçiş çizeneği ile aynı bilgiyi içeren ve makinenin her durumundan, her giriş simgesi ile hangi durumlara gidebileceğini gösteren bir çizelgedir. Başka bir deyişle geçiş çizelgesi her durumun her giriş simgesine karşı gelen ardıllarını kolaylıkla bulmayı



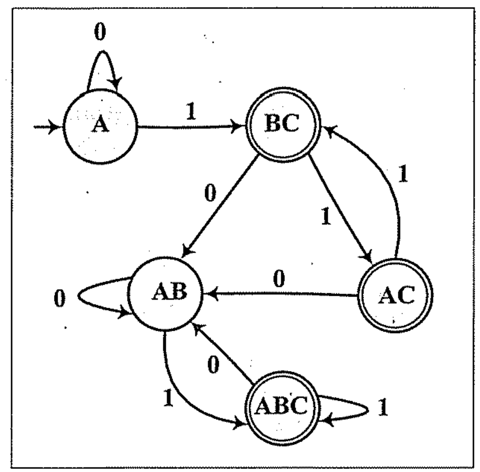
a) Deterministik Olmayan Geçiş Çizeneği

	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

Çizim 1.8. NFA'ya Denk DFA'nın Bulunması

Çizim 1.8.a

Çizim 1.8.c

sağlar. Ancak verilen bir w giriş dizgisini $M_{1.6}$ tarafından tanınabilmesi için başlangıç durumu olan A'nın w-ardılları arasında tek uç durum olan C'nin yer alması gerekir. Bu amaçla, Çizim 1.8.c'de görülen ve başlangıç durumunun tüm ardıllarını içeren çizelgeyi hazırlayabiliriz. Bir w giriş dizgisi verildiğinde bu ardıl çizelgesi yardımıyla, w'nin M1.6 tarafından tanınıp tanınmadığı kolaylıkla bulunabilir. Örneğin w=011 ise A'nın 011-ardılı AC olduğu ve AC bir uç durum içerdiği için de 011 dizgisinin M1.6 tarafından tanındığı; buna karşılık w=110 ise A'nın 110-ardılının AB olduğu ve AB içinde bir uç durum bulunmadığı için de 110 dizgisinin $M_{1.6}$ tarafından tanınmadığı kolaylıkla bulunabilir.

$M_{1.6}$'da başlangıç durumunun ardıllarını gösteren Çizim 1.8.c'deki çizelge ile deterministik bir özdevinir tanımlanmaktadır. Bu özdeviniri $Md_{1.6}$ olarak adlandıralım. $Md_{1.6}$'nın biçimsel tanımı aşağıdaki gibidir.

$M_{D1.6} = \langle Q, \sum, \delta, q_0, F \rangle$

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

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

$q_0 = A$

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

δ: $\delta(A, 0) = A$

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

$\delta(BC, 0) = AB$

$\delta(BC, 1) = AC$

$\delta(AB, 0) = AB$

$\delta(AB, 1) = ABC$

$\delta(AC, 0) = AB$

$\delta(AC, 1) = BC$

$\delta(ABC, 0) = AB$

$\delta(ABC, 1) = ABC$

$M_{D1.6}$'nın geçiş çizeneği Çizim 1.8.d'de görülmektedir. $M_{D1.6}$ verilen bir w giriş dizgisinin $M_{1.6}$ tarafından tanınıp tanınmadığını bulmak üzere geliştirildi. Dolayısıyla $M_{1.6}$ ve $M_{D1.6\text{'nın}}$ tanıdığı dizgiler kümesinin aynı olduğu ve bu iki makinenin denk olduğu açıktır. Başka bir deyişle $M_{D1.6}$ deterministik olmayan M1.6 makinesiyle aynı kümeyi tanıyan dolayısıyla $M_{1.6}$'ya denk deterministik bir makinedir. Bir kez bulduktan sonra deterministik makinenin durumlarına A,B,C,… ya q_0,q_1,q_2 gibi adlar vererek durumları yeniden adlandırmak mümkündür.

1.1.5.İki Yönlü Sonlu Özdevinir Modeli

Sonlu özdevinirlerin temel modeli olan DFA modeli tek yönlü bir modeldri. Çizim 1.2’de soyut bir makine olarak sunulan bu modelde, okuma şeridi soldansağa doğru tek yönlü işlenir. DFA’nın her hareketinde, şeritten bir simge okunur ve okuma kafası bir sağdaki hücreye geçer. Tek yönlü modelde okuma kafası şerit üzerinde sola doğru hareket edemez.

İki yönlü sonlu özdevinir modelinde ise okuma kafasının hem sağa hem de sola doğru hareket etmesi mümkündür. Makinenin her hareketinde okuma kafası, üzerinde bulunduğu hücredeki simgeyi okuduktan sonra ya bir sağdaki ya da bir soldaki hücreye geçer.

İki yönlü sonlu özdevinirlerin yalnız deterministik modeli incelenecek ve bu model 2DFA olarak adlandırılacaktır. 2DFA bir beşli olarak aşağıdaki gibi tanımlanır.

2DFA=<Q,∑,δ,q0,F>

2DFA’nın tanımında yer alan durumlar kümesi (Q), giriş alfabesi (∑), başlangıç durumu (q0) ve uç durumların (F) anlamları, DFA tanımındakilerle aynıdır. (bkz 1.1.1). DFA ve 2DFA arasındaki tek fark geçiş işlevinin tanımındadır. 2DFA modelinde geçiş işlevi:

(Q x ∑)’dan Q x {R,L}’ye bir eşleme

olarak tanımlanır. Bu tanımda R (Right) ve L(Left), okuma kafasının bir sağ mı yoksa bir sola mı geçeceğini gösterir.

Örnek 1.7. $M_{1.7} = <Q, \sum, \delta, q_0, F>$

$Q = \{q_0, q_1, q_2\}$

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

$F = \{q_0, q_1, q_2\}$

δ: $\delta(q_0, 0) = \{q_0, R\}$

$\delta(q_0, 1) = \{q_1, R\}$

$\delta(q_1, 0) = \{q_1, R\}$

$\delta(q_1, 1) = \{q_2, L\}$

$\delta(q_2, 0) = \{q_0, R\}$

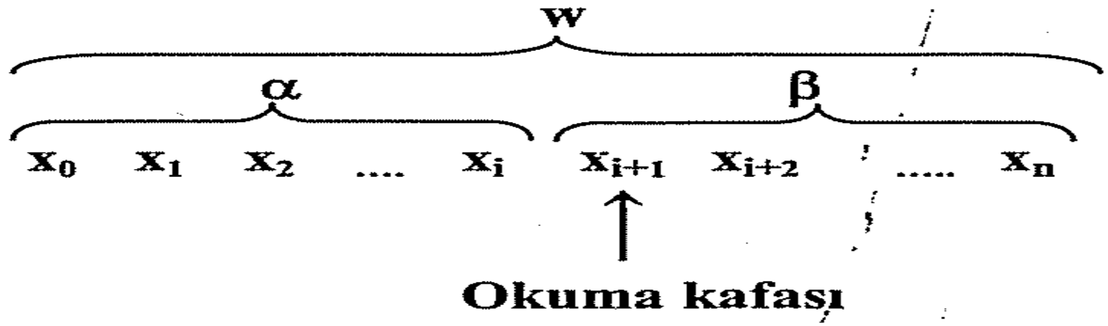
$\delta(q_2, 1) = \{q_2, L\}$

Örnek 1.7'deki $M_{1.7}$ makinesi, {0,1} alfabesinde içinde 11 alt dizgisi bulunmayan dizgileri tanımlayan bir 2DFA'dır. Tüm durumlar uç durum olduğuna göre, bir dizginin $M_{1.7}$ tarafından tanınabilmesi için sonlu sayıda hareket sonunda tüm dizginin işlenmesi ve okuma kafasının dizginin sağına geçmesi yeterlidir. Tek yönlü modelde her zamana sağlanan bu koşul iki yönlü özdevinirlerde her zaman sağlanamaz. İki nedenle okuma kafası sonlu hareket sonunda dizginin sağına ulaşamaz. Bu nedenlerden birincisi okuma kafasının hareketlerinin bir döngü oluşturması, ikinci ise şeridin ilk hücresi üzerinde bulunan okuma kafasının sola ilerlemesini gerektiren bir hareketle karşılaşılmasıdır.

Bir dizginin 2DFA tarafından tanınıp tanınmadığını bulmak pek kolay değildir. Bunun için şeritli makine modeli üzerinde okuma kafasının hareketlerinin izlenmesi gerekir. Ancak hareketler iki yönlü olduğu ve aynı hücre üzerinden okuma kafası defalarca geçebildiği için bu işlem oldukça uzun ve güç olabilir. Diğer bir yöntem ise, şeritli makine modelini kullanmadan, hareketlerin anlık tanımlarla izlenmesidir.

Anlık Tanım (ID: Instantenaous Description)

Makinenin giriş dizgisi w olsun. Başlangıçta okuma kafası w’nin ilk (en soldaki) simgesi üzerindedir. Belirli sayıda hareket sonunda, okuma kafası w’nin belirli bir simgesinin üzerinde bulunur. Bu durumda, w’nin bir kesimi okuma kafasının solunda yer alır. w’nin okuma kafasının solunda kalan kesiminin α ile, geri kalan kesimini ise β ile gösterelim. Buna göre okuma kafası β’nın ilk (en soldaki) simgesi üzerinde bulunmaktadır.



Anlık tanım makinenin bulunduğu durum ile giriş dizgisinin okuma kafasının solunda ve sağında kalan kesimlerini gösteren bir üçlüdür.

ID = (α, q, β)

Eğer ID1'den tek bir hareketle ID2'ye geçiliyorsa, bu iki anlık tanım arasındaki ilişki:

$$ID_1 \vdash ID_2$$

ile gösterilir.

Eğer ID1'den bir dizi hareket sonunda ID2'ye geçilebiliyorsa bu ilişki de:

$$ID_1 \overset{*}{\vdash} ID_2$$

ile gösterilir.

Anlık tanım ve anlık tanımlar arasındaki geçiş ilişkisine dayalı olarak iki yönlü sonlu bir özdevinirin (2DFA) tanıdığı dizgiler kümesi aşağıdaki gibi tanımlanır:

$$T(M) = \{w \mid (q_0, w) \overset{*}{\vdash} (w, q_i),\ q_i \in F\}$$

Yukarıdaki tanımda, (q_0,w) başlangıç anlık tanımıdır (ID0). Başlangıçta okuma kafası w'nin ilk simgesi üzerinde bulunduğu ve okuma kafasının solunda dizginin hiçbir simgesi bulunmadığı için ID0'da α yoktur. (w,q1) ise, makinenin bir uç durumda ($q_1 \in F$) okuma kafasının ise w'nin sağındaki boş hücre üzerinde bulundupu son ID'dir. Son ID'de, w'nin tümü okuma kafasının solunda bulunduğu için bu kez β yoktur.

Anlık tanımları kullanarak, w_1=100101 ve w_2=100110 giriş dizgilerini iki yönlü sonlu özdevinir olan $M_{1.7}$ makinesi tarafından tanınıp tanınmadığını inceleyelim:

a) $w_1 = 1000101$

$(q_0, 1000101) \vdash (1, q_1, 000101) \vdash (10, q_1, 00101) \vdash (100, q_1, 0101) \vdash$
$(1000, q_1, 101) \vdash (100, q_2, 0101) \vdash (1000, q_0, 101) \vdash (10001, q_1, 01) \vdash$
$(100010, q_1, 1) \vdash (10001, q_2, 01) \vdash (100010, q_0, 1) \vdash (1000101, q_1)$

Sonuç : $w_1 = 1000101$ dizgisi $M_{1.7}$ tarafından tanınır.

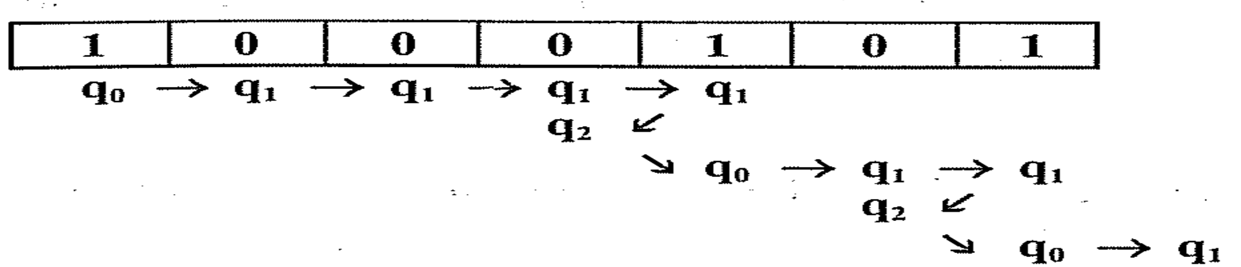
b) $w_2 = 100110$

$(q_0, 100110) \vdash (1, q_1, 00110) \vdash (10, q_1, 0110) \vdash (100, q_1, 110) \vdash$
$(10, q_2, 0110) \vdash (100, q_0, 110) \vdash (1001, q_1, 10) \vdash (100, q_2, 110) \vdash$
$(10, q_2, 0110) \vdash (100, q_0, 110)$ ....

Sonuç : ID geçişlerinde döngü oluştuğu için $w_2 = 100110$ dizgisi $M_{1.7}$ tarafından tanınmaz.

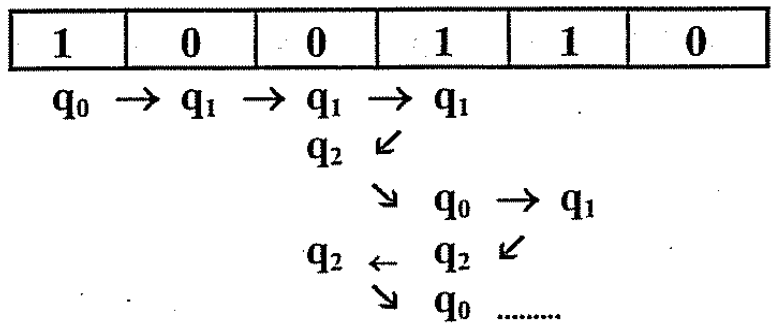
Giriş dizgilerinin 2DFA tarafından tanınıp tanınmadığı ID'ler arası geçiş ilişkileri yerine, şerit üzerindeki hareketleri ve durum değişimlerini inceleyerek de bulabiliriz. Bu yöntemle w_1=100101 ve w_2=100110 giriş dizgilerinin $M_{1.7}$ makinesi tarafından tanınıp tanınmadığı aşağıdaki gibi bulunur:

a) $w_1 = 1000101$



Sonuç: w_1 = 100101 dizgisi $M_{1.7}$ tarafından tanınır.

b) $w_2 = 100110$



Sonuç: Makinenin hareketlerinde döngü oluştuğu ve w_2 = 100110 dizgisinin $M_{1.7}$ tarafından tanınmadığı yukarıda açık biçimde görülmektedir. Şeridin bir hücresi altındaki durumlar dizisinde aynı durumun birden çok kez yer alması döngünün göstergesidir.

Tek ve İki Yönlü Özdevinirlerin Denkliği

Daha önce, deterministik olmayan her sonlu özdevinire denk (aynı kümeyi tanıyan) deterministik bir sonlu özdevinirin bulunabileceği, dolayısıyla NFA ve DFA modellerin denk olduğunu gördük. Benzer biçimde, iki yönlü sonlu her özdevinire denk, tek yönlü bir sonlu özdevinir bulunabilir. Dolayısıyla 2DFA ve DFA modellerinin ifade güçleri denktir. İki modelin denkliği verilen bir 2DFA'ya denk tek yönlü bir DFA'nın bulunmasını sağlayan yöntem ile ispat edilebilir. Böyle bir yöntem vardır. Ancak bu yönteme bu kitapta yer verilmeyecektir

1.2.Çıkış Üreten Özdevinirler

Bölümün başında sonlu özdevinirlerin öncelikle:

a) Sonlu durumlu tanıyıcı (finite state recognizer)
b) Çıkış üreten özdevinir

modelleri olarak sınıflandırıldığı belirtilmişti. Tanıyıcı model, bir giriş alfabesindeki simgelerden üretilebilecek dizgilerin bir kesimini tanıyan bir modeldir. Bu model biçimsel dillerle ilgili söz dizim çözümleme (*syntax analysis*), ayrıştırma (*parsing*) ve benzeri uygulamalar başta olmak üzere birçok yazılım uygulamasında kullanılır. Sonlu durumlu tanıyıcı modeli deterministik ve deterministik olmayan türlerinin bulunduğu ve bu iki türün modelleme gücü açısından denk olduğu da yukarıdaki kesimlerde görüldü.

Çıkış üreten özdevinir modeli, adından da anlaşıldığı gibi girişine uygulanan bir giriş dizgisine yanıt olarak bir çıkış dizgisi üreten bir modeldir. Bu açıdan çıkış üreten özdevinir modelinin, giriş dizgilerini çıkış dizgilerine dönüştüren bir model olarak görmek de mümkündür. Böylece sonlu özdevinirlerin iki ana türü “tanıyıcılar” ve “dönüştürücüler” olarak nitelenebilir. Aslında tanıyıcıların da çıkış ürettikleri düşünülebilir. Mademki bir tanıyıcı girişine uygulanan giriş dizgilerinin bir kısmını tanımakta, diğer bir kısmını ise tanımamaktadır. FA modeli ikili çıkış üreten bir model olarak görülebilir. Modelin ürettiği çıkışlar örneğin uç durumlar için “1: tanıdı”, diğer durumlar için “0:tanımadı” olabilir. Bu açıdan düşünüldüğünde ise çıkış üreten sonlu özdevinir modeli tanıyıcı modeli de kapsayan daha geniş bir model olarak görülebilir.

Çıkış üreten özdevinirleri Moore ve Mealy makinesi olarak adlandırılan iki türü vardır. Moore makinesi durum düzeyinde çıkış üreten bir model iken Mealy makinesi durum geçişi düzeyinde çıkış üreten bir modeldir. Aşağıda Moore ve Mealy makineleri ayrı ayrı incelenmektedir.

Moore Makinesi

- Moore makinesi bir altılı olarak tanımlanır:

  M=<Q, ∑, Δ, δ, λ, **q0>**

  Q: Sonlu durumlar kümesi

  ∑: Giriş alfabesi (sonlu bir küme)

  Δ: Çıkış alfabesi (sonlu bir küme)

  δ: Durum geçiş işlevi: (Q x ∑)’dan Q’ya bir eşleme

  λ: Çıkış işlevi: Q’dan Δ’ya bir eşleme

  q_0: Başlangıç durumu: q0 ⊊ Q

Moore makinesi DFA modelini genellemesi olarak görülebilir. Başka bir deyişle DFA modeli özel bir Moore makinesi olarak düşünülebilir. Gerçekten de eğer Moore modelinde çıkış alfabesi olarak {0,1} alfabesi alınır ve çıkış işlevi ile uç durumlara 1, uç olmayan durumlara da 0 değeri eşlenirse:

$\Delta = \{0, 1\}$ $\quad\quad$ $\sqrt{q_i} \in F \quad \Rightarrow \lambda(q_i) = 1$ $\quad\quad$ $\sqrt{q_i} \notin F \Rightarrow \lambda(q_i) = 0$

DFA modeli elde edilir.

Moore makinesinin durum geçiş işlevi (δ) ile çıkış işlevi (λ) bir çizelge ya da çizenek ile gösteriebilir. Moore makinesini tanımlamak için oluşturulan çizelge çoğunlukla "Durum Çizelgesi (*State Table*)" çizenek ise çoğunlukla "Durum Çizeneği (*State Diagram*)" olarak adlandırılır. Örnek 1.8'de tanımlanan $M_{1.8}$ Moore makinesini durum çizenek ve çizelgesi Çizim 1.9'da görülmektedir.

Örnek 1.8.

$M_{1.8}$ makinesi, girişine uygulanan ikili sayı X ise çıkışında z=Mod(X,5) değerinin üreten Moore makinesi olarak tanımlanıyor.

$$M_{1.8} = \langle Q, \sum, \Delta, \delta, \lambda, q0 \rangle$$

$$Q = \{A, B, C, D, E\}$$

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

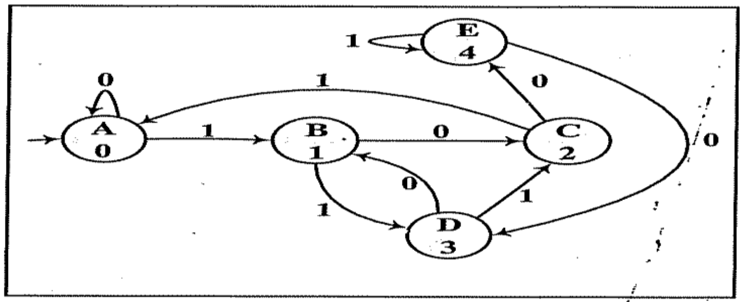
$$\Delta = \{0,1,2,3,4\}$$

$$q_0 = A$$

δ ve λ’nın tanımı, Çizim 1.9.a’daki durum çizeneği ile Çizim 1.9.b’deki durum çizelgesinde yer almaktadır.

$M_{1.8}$ makinesine örneğin w = 1011110 giriş dizgi uygulandığında makinenin durumları ve üreteceği çıkışlar aşağıdaki gibi değişecektir.

Giriş :		1		0		1		1		1		1		0	
Durum:	A	→	B	→	C	→	A	→	B	→	D	→	C	→	E
Çıkış :	0		1		2		0		1		3		2		4



Durum Çizeneği

ŞD	SD x = 0	SD x = 1	z
→ A	A	B	0
B	C	D	1
C	E	A	2
D	B	C	3
E	D	E	4

Durum Çizelgesi

Çizim 1.9. Örnek Moore Makinesi ($M_{1.8}$)

Mealy Makinesi

Moore makinesi gibi Mealy makinesi de bir altılı olarak tanımlanır.

$$\mathbf{M = <Q, \sum, \Delta, \delta, \lambda, q0>}$$

Altılının çıkış işlevi (λ) dışındaki beş elemanının tanımı Moore makinesindekilerle aynıdır. Başka bir deyişle Moore ve Mealy makineleri arasındaki tek fark çıkış işlevindedir. Moore makineleri için Q'dan Δ'ya bir eşleme olarak tanımlanan çıkış işlevi Mealy makineleri için (Q x $\sum$)'dan Δ'ya bir eşleme olarak tanımlanır.

Örnek 1.9.

$M_{1.9}$ makinesi giriş alfabesi {0,1} çıkış alfabesi ise {0,1,2} olan ve ürettiği çıkış ile son iki giriş simgesinden kaç tanesinin değerini bir öncekinden farklı olduğunu gösteren Mealy makinesi olarak tanımlanıyor. Makinenin üreteceği ilk iki çıkış değeri belirlenirken başlangıç durumundan önceki iki giriş değerinin 00 olduğu varsayılacaktır.

$$M = \langle Q, \sum, \Delta, \delta, \lambda, q0 \rangle$$

$$Q = \{A, B, C, D\}$$

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

$$\Delta = \{0, 1, 2\}$$

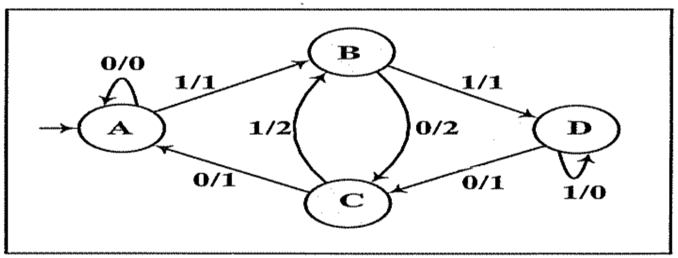
$$q0 = A$$

δ ve λ, Çizim 1.10.a'daki Durum Çizeneği ve Çizim 1.10.b'deki Durum Çizelgesi ile tanımlanmaktadır. Yapılan tasarıma göre makinenin durumları önceki iki giriş simgesi değerini 00, 01, 10 ya da 11 olmasına göre sırasıyla A, B, C ya da D olmaktadır.

Örnek bir giriş dizgisiyle bu giriş dizgisine karşılık makinenin üretmesi gereken çıkış dizgisi aşağıdaki gibidir:

X : 0 1 1 0 1 0 1 0 1 1 1 1 0 1 0 0 1 1 1 0 0 0 0 1

Z : 0 1 1 1 2 2 2 2 2 1 0 0 1 2 2 1 1 1 0 1 1 0 0 1



Durum Çizeneği

ŞD	SD, z x = 0	x = 1
→A	A, 0	B, 1
B	C, 2	D, 1
C	A, 1	B, 2
D	C, 1	D, 0

Durum Çizelgesi

Çizim 1.10. Örnek Mealy Makinesi ($M_{1.9}$)


CREATED USING
POWTOON

Moore ve Mealy Makinelerinin Eşdeğerliği

Moore ve Mealy makinelerinin her ikisi de çıkış üreten sonlu özdevinir modelidir. Birer altılı olarak tanımlanan Moore ve Mealy modellerinde altılının beş elemanı ortaktır. Moore ve Mealy modellerinin ayıran çıkış işlevidir. Moore modelinde, makine her durum için bir çıkış simgesi üretmektedir. Belirli bir durumda bulunan bir Moore makinesine, n giriş simgesinden oluşan bir dizgi uygulandığında, makine(n+1) uzunluğunda bir çıkış dizgisi üretecektir. Başlangıçtaki durumda q0, uygulanan n giriş simgesi sonrasındaki durumlar da q_1, q_2, …,q_n ile gösterilerek, Moore makinesinin giriş dizgisi, durum geçişleri ve çıkış dizgisi ilişkisi aşağıdaki gibi özetlenebilir. Hiçbir giriş simgesi uygulamadan, bulunduğu başlangıç durumuna göre Moore makinesinin bir çıkış simgesi (z_0) ürettiği dikkati çekmektedir.

$$
\begin{array}{ccccccccccccccc}
 & x_1 & & x_2 & & x_3 & & \ldots & & & x_{n-1} & & x_n & \\
q_0 & \to & q_1 & \to & q_2 & \to & q_3 & \to \ldots & \to & q_{n-2} & \to & q_{n-1} & \to & q_n \\
z_0 & & z_1 & & z_2 & & z_3 & \ldots & & z_{n-2} & & z_{n-1} & & z_n
\end{array}
$$

Mealy modelinde ise makine her durum geçişi için bir çıkış simgesi üretmektedir. Belirli bir durumda bulunan bir Mealy makinesine, n giriş simgesinden oluşan bir giriş dizgisi uygulandığında, makine n uzunluğunda bir çıkış dizgisi üretecektir. Çünkü Mealy modelinde, giriş simgesi uygulanmadığı sürece makine çıkış üretmemektedir. Başlangıçtaki durum q0, uygulanan n giriş simgesi sonrasındaki durumlar da $q_1, q_2, \ldots, q_n$ ile gösterilerek, Mealy makinesinin giriş dizgisi, durum geçişleri ve çıkış dizgisi ilişkisi aşağıdaki gibi özetlenebilir.

$$\begin{array}{ccccccccccccccc}
 & x_1 & & x_2 & & x_3 & & & \cdots & & & x_{n-1} & & x_n & \\
q_0 & \rightarrow & q_1 & \rightarrow & q_2 & \rightarrow & q_3 & \rightarrow & \cdots & \rightarrow & q_{n-2} & \rightarrow & q_{n-1} & \rightarrow & q_n \\
 & z_1 & & z_2 & & z_3 & & & \cdots & & & z_{n-1} & & z_n &
\end{array}$$

Buna göre n uzunluğundaki X giriş dizgisine karşılık Mealy makinesi zmealy çıkış dizgisinin üretirken, Moore makinesi z_0 Z_{moore} çıkış dizgisini üretmektedir. Z_{mealy} ve Z_{moore} çıkış dizgiler n uzunluğunda dizgilerdir.

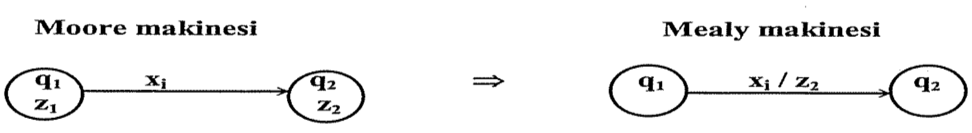
Eğer aynı giriş ve aynı çıkış alfabesine sahip bir Mealy ve bir Moore makinesinin ürettiği Z_{mealy} ve Z_{moore} çıkış dizgileri, tüm X giriş dizgileri için aynı ise bu iki makine birbirine denktir. Başka bir deyişle, Moore makinesinin başlangıçta ürettiği z_0 çıkış simgesi dışarda tutulursa uygulanan giriş dizgisi ne olursa olsun birbirine denk Mealy ve Moore makineleri aynı çıkış dizgisini üretecektir.

Moore Makinesine Eşdeğer Mealy Makinesinin Bulunması

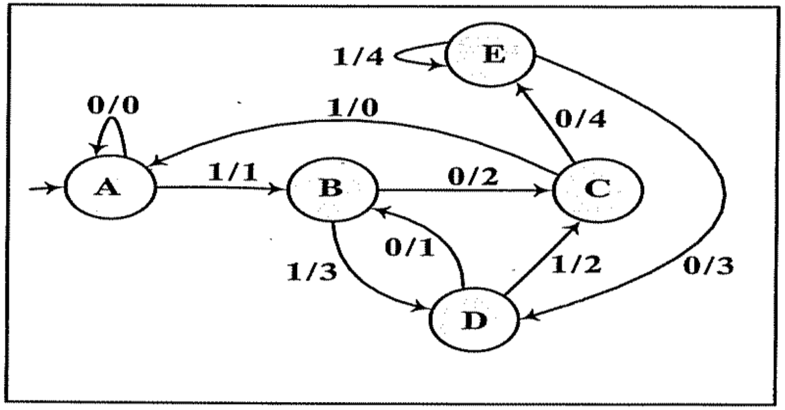
Önerme 1.1. $M_1 = <Q, \sum, \Delta\ \delta, \lambda, q_0>$ bir Moore makinesi olsun. M_1'e eşdeğer Mealy makinesi aşağıdaki gibi bulunur.

$$M_2 = <Q, \sum, \Delta, \delta, \lambda, q_0>, \quad \lambda'(q, a) = \lambda(\delta(q, a))$$

Yukarıdaki tanıma göre bir Moore makinesi ile bu makineye eşdeğer Mealy makinesi arasındaki tek fark çıkış işlevindedir. Bilindiği gibi Moore makinesinde her duruma bir çıkış simgesi eşlenirken, Mealy makinesinde her durum geçişine (durum, giriş simgesi çiftine) bir çıkış simgesi eşlenir. Bu kitapta önermenin ispatına yer verilmeyecektir. Bir Moore makinesine eşdeğer Mealy makinesi bulunurken, her durum geçişine eşlenen çıkış değeri, durum geçişinin ucundaki sonraki duruma Moore makinesinde eşlenen çıkış değerine eşittir. Bu kuralı aşağıdaki çizimle özetlemek mümkündür.



Örnek 1.8’deki Moore makinesine ($M_{1.8}$) eşdeğer Mealy makinesinin, Önerme 1.1’e göre bulunan durum çizenek ve çizelgesi Çizim 1.11’de görülmektedir. bir Moore makinesi ile bu makineye eşdeğer Mealy makinesinin hem durum sayılarının hem de durumlar arası geçiş sayılarının eşit olduğu görülmektedir.



Durum Çizeneği

ŞD	SD, z: x = 0	SD, z: x = 1
→ A	A, 0	B, 1
B	C, 2	D, 3
C	E, 4	A, 0
D	B, 1	C, 2
E	D, 3	E, 4

Durum Çizelgesi

Çizim 1.11. $M_{1.8}$ Moore Makinesine Eşdeğer Mealy Makinesi

Mealy Makinesine Eşdeğer Moore Makinesinin Bulunması

Önerme 1.2. $M_2 <Q, \sum, \Delta\ \delta, \lambda, q_0>$ bir Mealy makinesi olsun. M_2'ye eşdeğer M_1 Moore makinesi aşağıdaki gibi bulunur:

$M_1 = <Q', \sum', \Delta', \delta', \lambda', q_0'>$

$Q' = Q \times \Delta$

$q_0' = [q_0, zj]$ (zj = çıkış simgelerinden rastgele seçilmiş biri)

$\delta'([q_1, z_k], x_j) = [\delta(q_1, x_j)], \lambda(q_1, x_j)]$

$\lambda'([q_1, z_k]) = z_k$

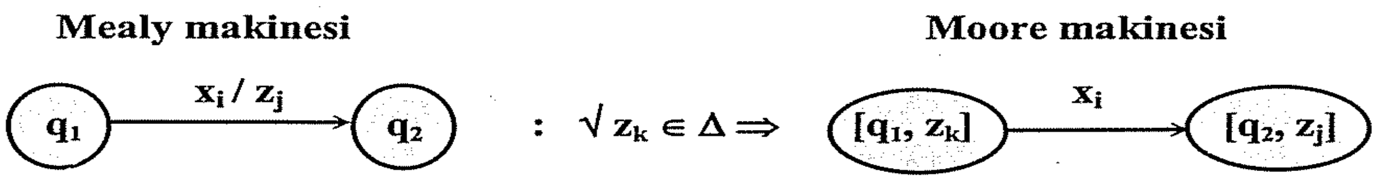
Önerme 1.2'ye göre bir Mealy makinesi (M_2) ile bu makineye eşdeğer Moore makinesinin (M_1) tanımını oluşturan altılıların dördü birbirinden farklıdır.

- M_2'nin her [durum çıkış simgesi] çiftine karşılık M_1 makinesinin bir durumu vardır. Buna göre M_2 makinesinin durum sayısı n, çıkış simgesi sayısı ise m ise M_1 makinesinin n x m durumu bulanacaktır. Örneğin M_9 Mealy makinesinin 4 durumu, 4 de çıkış simgesi vardır. M_9 Mealy makinesinin, Önerme 1.2'ye göre bulunacak eşdeğeri Moore makinesinin 4x4 = 16 durumu olacak ve bu durumlar sembolik olarak [A,0], [A,1], …, [D,4] diye gösterilmektedir.
- M_1'in durumlarından, sembolik gösterimde birinci elemanı q_0 olanlar arasından rastgele seçilen biri başlangıç durumu yapılır. Örneğin çıkış alfabesinde 4 simge bulunan M_9 Mealy makinesinin başlangıç durumu A olduğuna göre bu makineye eşdeğer Moore makinesinin başlangıç durumu [A,0], [A,1], [A,2] ve [A,3] arasından rastgele seçilen biri olacaktır.
- M_1'in geçiş işlevi

$$\delta' ([q_1, z_k], x_j) = [\delta(q_1, x_j)],\lambda(q_1, x_j)]$$

olarak tanımlanmaktadır.

Bu tanımın daha kolay anlaşılmasını sağlamak için aşağıdaki çizim oluşturulabilir.



Bir Mealy makinesine eşdeğer Moore makinesinin, yukarıdaki çizimle açıklamaya çalışılan durum geçiş işlevi, sözlü olarak da şöyle anlatılabilir. Mealy makinesinde q_1 durumundan q_2 durumuna z_j geçişi sırasında z_j çıkış simgesi üretiliyorsa eşdeğer Moore makinesinde birinci elemanı q_1 olan her durumdan x_j giriş simgesi ile birinci elemanı q_2, ikinci elemanı ise z_j olan duruma bir geçiş oluşturulur. Buna göre eğer çıkış alfabesinde m çıkış simgesi varsa, Mealy makinesindeki her durum geçişine karşılık, eşdeğer Moore makinesinde m geçiş bulunacaktır. Böylece bir Mealy makinesinin her durumuna karşılık eşdeğer Moore makinesinde m durum bulunduğu gibi, Mealy makinesindeki her durum geçişine karşılık eşdeğer Moore makinesinde m durum geçişi bulunacaktır.

Örnek 1.10 Mealy makinesi aşağıdaki gibi tanımlanıyor:

$M_{1.10} = \langle Q, \sum, \Delta, \delta, \lambda, q_0 \rangle$

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

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

$\Delta = \{0, 1\}$

$q_0 = A$

$\delta(A, 0) = B$	$\lambda(A, 0) = 0$
$\delta(A, 1) = A$	$\lambda(A, 1) = 1$
$\delta(B, 0) = B$	$\lambda(B, 0) = 1$
$\delta(B, 1) = C$	$\lambda(B, 1) = 1$
$\delta(C, 0) = A$	$\lambda(C, 0) = 0$
$\delta(C, 1) = C$	$\lambda(C, 1) = 0$

$M_{1.10}$ Mealy makinesine eşdeğer Moore makinesini ($M'_{1.10}$) bulalım.

$M'_{1.10} = \langle Q', \sum, \Delta, \delta', \lambda', q'_0 \rangle$

$Q' = \{[A, 0], [A, 1], [B, 0], [B, 1], [C, 0], [C, 1]\}$

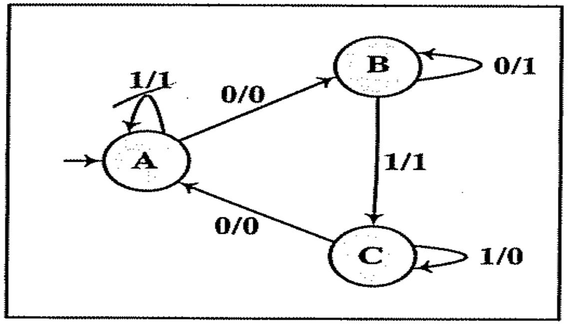
$q'_0 = [A, 0]$

$\delta'([A, 0], 0) = [B, 0]$	$\lambda[A, 0] = 0$
$\delta'([A, 1], 0) = [B, 0]$	$\lambda[A, 1] = 1$
$\delta'([A, 0], 1) = [A, 1]$	$\lambda[B, 0] = 0$
$\delta'([A, 1], 1) = [A, 1]$	$\lambda[B, 1] = 1$
$\delta'([B, 0], 0) = [B, 1]$	$\lambda[C, 0] = 0$
$\delta'([B, 1], 0) = [B, 1]$	$\lambda[C, 1] = 1$
$\delta'([B, 0], 1) = [C, 1]$	
$\delta'([B, 1], 1) = [C, 1]$	
$\delta'([C, 0], 0) = [A, 0]$	
$\delta'([C, 1], 0) = [A, 0]$	
$\delta'([C, 0], 1) = [C, 0]$	
$\delta'([C, 1], 1) = [C, 0]$	

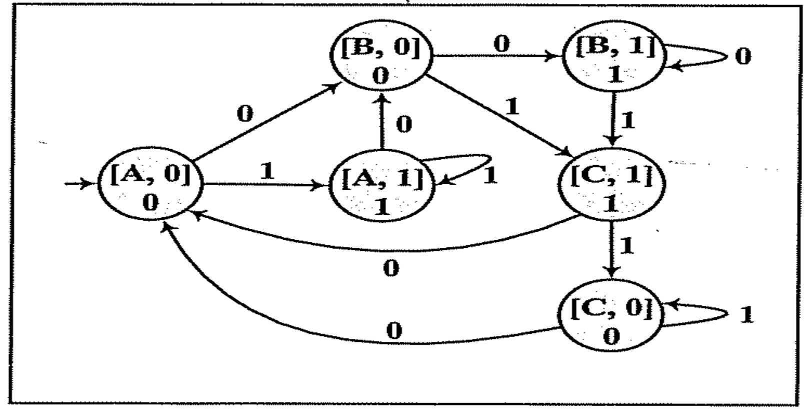
Çizim 1.12’de $M_{1.10}$ Mealy makinesi ile bu makineye eşdeğer Moore makinesinin durum çizeneği görülmektedir. Önerme 1.2’ye göre bir Mealy makinesine eşdeğer Moore makinesini bulurken, Moore makinesinin durumlarının [durum adı, çıkış simgesi] ikilileri ile gösterilmesi gerekir. Ancak eşdeğer Moore makinesi bulunduktan sonra, bu makinenin durumları yeniden adlandırılabilir. Nitekim, $M_{1.10}$ Moore makinesinin Çizim 1.12.c’e görülen durum çizeneği, makinenin durumları:

[A,0] için q_0
[A,1] için q_1
[B,0] için q_2
[B,1] için q_3
[C,0] için q_4
[C,1] için q_5

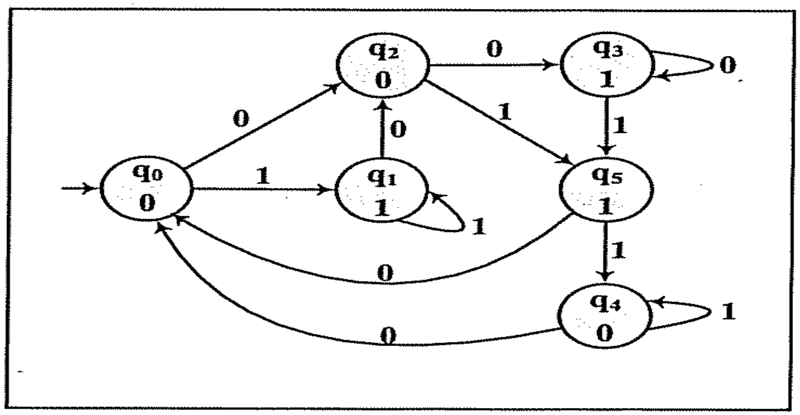
adları kullanılarak oluşturulmuştur.



a) $M_{1.10}$ Mealy Makinesi



b) Eşdeğer Moore Makinesi ($M'_{1.10}$)



c) $M'_{1.10}$ Moore Makinesi - Durumlar Yeniden Adlandırıldıktan Sonra

Çizim 1.12. Mealy Makinesine Eşdeğer Moore Makinesinin Bulunması


CREATED USING
POWTOON

1.3.Sonlu Özdevinirlerin İndirgenmesi

Sonlu özdevinirlerle ilgili olarak şimdiye kadar birden çok model tanımlandı. Tanımlanan modelleri aşağıdaki gibi özetlemek mümkündür:

Tanıyıcı model. Sonlu özdevinirlerin temel modeli olan ve kısaca FA olarak adlandırılan bu modelin deterministik (DFA) ve deterministik olmayan (NFA) türleri vardır.

Çıkış üreten özdevinir modeli. Bu modelin de Mealy ve Moore modeli (ya da makinesi) olarak adlandırılan iki türü bulunmaktadır.

Gerek FA modelinde gerekse çıkış üreten sonlu özdevinir modelinde tanımlanan sonlu özdevinirin (makinenin) karmaşıklığı durum sayısı ile doğru orantılıdır. Bu nedenle, bir sonlu özdevinir verildiğinde bu sonlu özdevinirin indirgenmesi önem taşıyabilir. Bir sonlu özdevinirin indirgenmesi ya da yalınlaştırılması, bu sonlu özdevinire denk, durum sayısı en küçük sonlu özdevinirin bulunmasıdır. Sonlu özdevinirlerin indirgenmesi DFA, Moore ve Mealy modelleri için görülecektir. Bunun için önce belirli tanımların yapılması gerekir.

1.3.1. Ardıl, Öncel, Denk ve Ayırt Edilebilir Durum Tanımları

DFA, Moore ve Mealy modellerinin en genişi Mealy modelidir. Çünkü tanıyıcı (FA) model, Moore modelinin çıkış alfabesi {kabul, red} olan bir alt türü olarak görülebilir. Moore modeli ise, Mealy modelinin, çıkış işlevi giriş alfabesinden bağımsız olan bir alt türü olarak görülebilir. Bu nedenle tanımlar verilirken Mealy modeli esas alınacaktır ve örnek olarak aşağıda tanımı verilen $M_{1.11}$ makinesi kullanılacaktır.

Örnek 1.11

Mealy türü $M_{1.11}$ makinesinin giriş ve çıkış alfabeleri {0, 1} alfabesidir. Başlangıç durumu A olan makinenin geçiş ve çıkış işlevleri aşağıda durum çizelgesi ile tanımlanmaktadır.

ŞD	SD, z	
	x = 0	x = 1
→ A	A, 0	D, 1
B	C, 0	E, 1
C	G, 0	E, 1
D	G, 0	F, 1
E	E, 1	C, 0
F	B, 0	D, 1
G	B, 0	E, 1

Ardıl

M makinesinin S_1 durumundan x giriş simgesi ile S_2 durumuna geçiliyorsa:

$$S_1 \xrightarrow{x} S_2$$

S_1'in x-ardılı S_2'dir. w bir giriş simgeleri dizisi olmak üzere, eğer S_1 durumundan w giriş dizgisi ile S_2 durumuna geçiliyorsa, S_1'in w-ardılı S_2'dir. Örneğim $M_{1.11}$ makinesinde A durumunun 1-ardılı D'dir. Aynı makine B durumunun 011-ardılı ise C'dir.

X bir giriş simgesi, w ise giriş simgelerinin bir dizgisi olmak üzere, deterministik modellerde, bir durumun x ve w ardılı her zaman tek bir durumdur. Deterministik olmayan modellerde ise x ve w ardılları, durumların birer altkümesidir. Geçiş çizeneğinde yer aldığı için bir durumun x-ardılı kolaylıkla bulunabilir. Bir durumun $w=x_1x_2…x_k$ ardılını bulmak için ise, geçiş çizeneği kullanılarak önce x1-ardılı bulunur; daha sonra x_1-ardılın x_2-ardılı, … vb bulunarak w-ardılı elde edilir. Örneğin $M_{1.3}$ makinesinde q_0 durumunun 0-ardılı $\{q_0, q_1\}$'dir. $\{q_0,q_1\}$ yerine q_0q_1 yazılabilir. Buna göre q_0 durumunun 0-ardılı q_0q_1, 1-ardılı q_0q_2, 100-ardılı ise $q_0q_1q_3$'dür. Aynı makinenin q_1 durumunun 1-ardılı ile q_2 durumunun 01-ardılı ise yoktur. (boş kümedir.)

Öncel

M makinesinin S_1,S_2, … ve S_i durumlarından x giriş simgesi ile S_j durumuna geçiliyorsa, S_j durumunun x-önceli, {S_1,S_2,…, S_j}'dir. Kullanılan model deterministik olsa bile, bir durumun x ve w öncelleri, durumların birer altkümesidir. Örneğin $M_{1.11}$ makinesi için B durumunun 0-önceli FG, 001-önceli de AEF'dir. Aynı makine için D durumunun 0 ve 110 öncelleri ise yoktur. Bir makinenin x-öncellerini bulmak için öncel çizelgesi adı verilen bir çizelge oluşturulabilir. $M_{1.11}$ makinesi için oluşturulan öncel çizelgesi aşağıda görülmektedir.

ŞD	Önceki Durum	
	x = 0	x = 1
A	A	-
B	FG	-
C	B	E
D	-	AF
E	E	BCG
F	-	D
G	CD	-

Denk Durumlar

M makinesi S_1 ve S_2 durumlarından herhangi birinde iken, hangi giriş simgesi uygulanırsa uygulansın makine hep aynı çıkış simgesi üretiyorsa, bu durumlara *1-denk durumlar* denir. Örneğin $M_{1.11}$ makinesinin A ve B durumları 1-denktir. C ve F durumları da 1-denktir. Buna karşılık E ve G durumları 1-denk değildir.

M makinesi S_1 ve S_2 durumlarından herhangi birinde iken, uzunluğu n ya da daha kısa olan hangi giriş dizgisi uygulanırsa uygulansın, makine hep aynı çıkış dizgisini üretiyorsa, bu durumlara *n-denk durumlar* denir. Örneğin $M_{1.11}$ makinesinin A ve D durumları 2-denktir. B ve C durumları 3-denktir. B ve C durumları aynı zamanda 4-denktir, 5-denktir,…vb. Buna karşılık A ve D durumları 3-denk değildir.

M makinesi S_1 ve S_2 durumlarından herhangi birinde iken, uzunluğuna bakılmaksızın, hangi giriş dizgisi uygulanırsa uygulansın, makine hep aynı çıkış dizgisini üretiyorsa, bu durumlara *denk durumlar* denir. Örneğin $M_{1.11}$ makinesinin B ve C durumları denk durumlardır. D ve F durumları da denktir. Ancak 2-denk olan A ve D durumları denk değildir.

Uygulamada, tüm n'ler için n-denk olan durumlara *denk durumlar* denir. N ve n'den küçük tüm k değerleri (√k<=n) için k-denk olan iki duruma ise *n-denk* denir. Örneğin $M_{1.11}$ makinesinin A ve F durumları 2-denk durumlardır. Bu durumların 2-denk olarak nitelenmesi, 3-denk olmadıkları anlamına gelir. Ancak 2-denk olan durumlar doğal olarak 1-denktir. Ancak durum denkliği her zaman en büyük n değeri üzerinde belirtilir.

Ayırt Edilebilir Durumlar

M makinesinin S_1 ve S_2 durumlarını ayırt etme için eğer en az n uzunluğunda bir giriş dizgisi gerekli ise bu durumlara *n-ayırt edilebilir durumlar* denir. Eğer S_1 ve S_2 durumları n-ayırt edilebilir ise, bu iki durum (n-1)-denktir. Örneğin $M_{1.11}$ makinesinin A ve E durumları 1-ayırt edilebilir. B ve D durumları ise 2-ayırt edilebilir. Buna karşılık A ve F durumları 3-ayırt edilebilir. D ve F durumlarını ise hiçbir giriş dizgisi ile ayırt etmek mümkün değildir. Denk olan D ve F durumları ayırt edilemez durumlardır.

$A \xrightarrow[Z=0]{X=0} A$	$E \xrightarrow[Z=1]{X=0} E$	**A ve E 1-ayırdedilebilir**
$B \xrightarrow[Z=11]{X=10} E$	$D \xrightarrow[Z=10]{X=10} B$	**B ve D 2-ayırdedilebilir**
$A \xrightarrow[Z=010]{X=010} G$	$F \xrightarrow[Z=011]{X=010} E$	**A ve F 3-ayırdedilebilir**

Makine Denkliği

M_1 ve M_2, giriş ve çıkış alfabeleri aynı olan iki makine olsun. M_1 makinesinin durumlarını S_{11}, S_{12}, S_{13},… olarak; M_2 makinesinin durumları ise S_{21},S_{22},S_{23},… olarak gösterelim. eğer M_1 makinesi S_{11}, M_2 makinesi de S_{21} durumundayken, her iki makineye aynı giriş dizgisi uygulandığında, giriş dizgisi ne olursa olsun her iki makine de aynı çıkış dizgisini üretiyorsa, bu ki durum (M_1 makinesinin S_{11} durumu ile M_2 makinesi de S_{21} durumu) denktir.

Eğer M_1 makinesinin her durumu için M_2 makinesinde bu duruma denk bir durum varsa; M_2 makinesinin her durumu için de M_1 makinesinde bu duruma denk bir durum varsa, M_1 ve M_2 makineleri denktir.

$$\left.\begin{array}{lllll}\surd\, S_{1i} \in M_1 & \rightarrow & \exists\, S_{2j} \in M_2 & : & S_{1i} \equiv S_{2j} \\ \surd\, S_{2i} \in M_2 & \rightarrow & \exists\, S_{1j} \in M_1 & : & S_{2i} \equiv S_{1j}\end{array}\right\} M_1 \equiv M_2$$

Makine İndirgeme

Bir M makinesi verildiğinde, bu makinenin indirgenmesi ya da yalınlaştırılması, bu makineye denk makinelerden durum sayısı en küçük olanının bulunması anlamı taşır. Bu makineyi M_{min} ya da M* ile gösterebiliriz. Bir makineye denk bir ya da birden çok en küçük makine olabilir.

1.3.2. İndirgeme Yöntemi

Sonlu özdevinirlerin indirgenmesi için denklik bölümleri (*equivalence partitions*) kullanılır. Bir M makinesi için, P_k ile gösterilen k-denklik bölümlemesi, k-denk durumların aynı bölümde yer aldığı bir bölümlemedir. Örneğin Mealy makinesi olan $M_{1.11}$ için 0-denklik bölümlemesi

P_0 = (ABCDEFG)

tek bölüm içerir. Çünkü hiçbir giriş simgesi uygulanmadan, bir Mealy makinesinin durumlarını ayırt etmek mümkün değildir. M1.11 için 1-denklik bölümlemesi ise

P_1 = (ABCDFG)(E)

iki bölüm içerir. Çünkü makinenin E dışındaki tüm durumları 1-denktir. Makinenin denklik bölümlemesini bulmak için sırasıyla P_0, P_1, P_2,.. bulunur. Denklik bölümlemelerinin türetilmesi

$P_{k+1} = P_k$

elde edilinceye kadar sürdürülür. $P_{k+1} = P_k$ elde edildiğinde, denklik bölümlenmesinin

$P = P_k$

olduğu anlaşılır.

Pm denklik bölümlemesinden, Pm+1 denklik bölümlemesinin türetilmesi için Önerme1.3’den yararlanılır.

Önerme1.3.

M makinesinin S_1 ve S_2 durumunun (m+1) denk olması için aşağıdaki iki koşulum sağlanması gerekli ve yeterlidir.

a. S_1 ve S_2 m-denk olmalı (P_m denklik bölümlemesinde aynı bölümde bulunmalı.)
b. Tüm x giriş simgeleri için S_1 ve S_2 durumlarının x-ardılları da m*denk olmalı (Pm denklik bölümlemesinde aynı bölümde bulunmalı.)

1.3.2.1. Mealy Makinelerinin İndirgenmesi

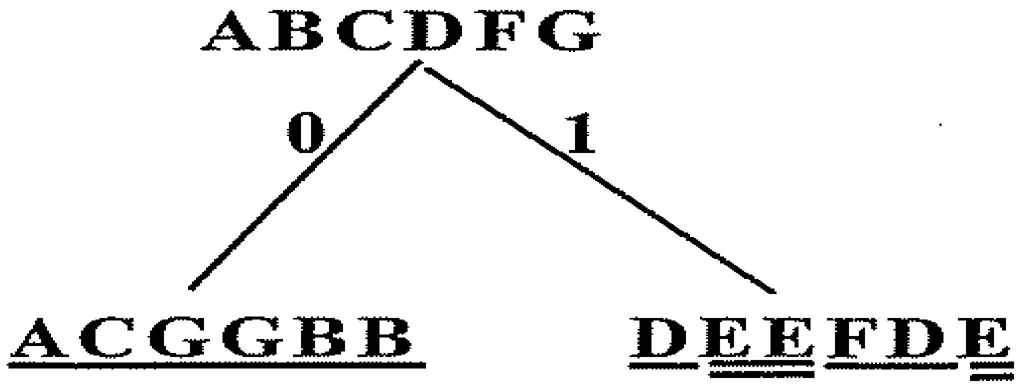
Mealy makinelerinin indirgenmesi M1.11 makinesi örnek alınarak incelenecektir. Daha önce belirtildiği gibi, hiçbir giriş uygulanmadan Mealy makinesinin durumları ayırt edilemez. Başka bir deyişle Mealy makinesinin tüm durumları 0-denktir.

$P_0 = (ABCDEFG)$

P_1 denklik bölümlemesini bulmak için durum çizelgesinin incelenmesi yeterlidir. M1.11'in durum çizelgesinde, E dışındaki tüm durumlar için, x geçişi sırasında üretilen çıkış değerinin birinci kolonda 0, ikinci kolonda ise 1 olduğu görülür. Buna göre E dışındaki durumlar 1-denktir.

$P_1 = (ABCDFG)(E)$

P_2 denklik bölümlemesi bulmak için Önerme 1.3'den yararlanılır. $M_{1.11}$ makinesinin iki durumunun 2-denk olabilmesi için, hem bu iki durumun, hem de bu iki durumun 0 ve 1-ardıllarının 1-denk olması; bunun için de P_1 denklik bölümlemesinde aynı bölümde bulunması gerekir.



P_1 bölümlemesinde aynı bölümde bulunan ABCDFG durumlarının 0-ardılları, P_1 bölümlemesinde aynı bölümde yer aldığı için 1-denktir. Ancak bu durumlarının 1- ardıllarının tümü 1-denk değildir. Buradan, ADF durumlarının kendi aralarında, BCG durumlarının da kendi aralarında 2-denk olduğu anlaşılır ve aşağıdaki P_2 denklik bölümlemesi elde edilir.

P_2 = (ADF)(BCG)(E)

P_3 denklik bölümlemesini bulmak için ADF ve BCG durumlarının 0 ve 1-adrılları incelenir:

Yapılan incelemede ADF durumlarının 1-ardıllarının P_2'de aynı bölümde yer aldığı, ancak 0-ardıllarının P_2'de aynı bölümde yer almadığı görülür. Bu nedenle DF durumlarının 3-denk olduğu, ancak A durumunun D ve F durumlarına 3-denk olmadığı anlaşılır ve P_3'de A durumu DF durumlarından ayırır. BCG durumlarına gelince, bu durumların hem 0 hem de 1-ardılları P_2'de aynı bölümde yer almaktadır. Buna göre BCG durumları 3-denk durumlardır ve P_3 denklik bölümlemesinde aynı bölümde yer alacaklardır. Sonuç olarak P_3 denklik bölümlemesi aşağıdaki gibi oluşur:

$$P_3 = (A)(DF)(BCG)(E)$$

P_4 denklik bölümlemesini bulmak için DF ve BCG durumlarının 0 ve 1-ardıllarını incelemek gerekir. BCG durumlarının, daha önce incelenen ve yukarıda yer alan 0 ve 1-ardılları P_3'de de aynı bölümde yer almaktadır. Bu nedenle BCG durumları P_4'de de aynı bölümde yer alacaktır.

DF durumlarının ardılları incelendiğince ise, bu iki durumun 0 ve 1-ardıllarının P_3'de aynı bölümde yer aldığı görülür. Bu nedenle DF durumları P_4'de aynı bölümde yer alacaktır. Sonuç olarak, P_4 denklik bölümlemesi P_3'e eşittir.

$P_4 = P_3 = (A)(DF)(BCG)(E)$

Böylece M1.11 makinesinin denklik bölümlemesi:

$P = (A)(DF)(BCG)(E)$

olarak elde edilir. Denklik bölümlemesinde 4 bölüm olduğu için, $M_{1.11'e}$ denk en küçük makinenin 4 durumu olacaktır. En küçük makinenin durumları:

A için S_0,

DF için S_1,

BCG için S_2,

E için S_3

Diye adlandırılırsa, $M_{1.11}$ makinesine denk en küçük makinenin durum çizelgesi aşağıdaki gibi bulunur.

ŞD	SD, z	
	x = 0	x = 1
S_0	S_0, 0	S_1, 1
S_1	S_2, 0	S_1, 1
S_2	S_2, 0	S_3, 1
S_3	S_3, 1	S_2, 0

1.3.2.2. Moore Makinelerinin İndirgenmesi

Bilindiği gibi Moore makinelerinde çıkış işlevi durumlar kümesinden çıkış alfabesine bir eşlemedir. Bu eşlemeyle her duruma bir çıkış simgesi eşlenir. Bir Moore makinesi belirli bir durumda iken belirli bir çıkış simgesi üretir. Bu nedenle hiçbir giriş simgesi uygulanmadan belirli durumlar ya da durum altkümeleri ayırt edilebilir. Başka bir deyişle, Moore makinesinin tüm durumları 0-denk değildir. Moore makinesinin P_0 0-denklik bölümlemesinde, çıkış simgesi kadar bölüm bulunur.

Mealy ve Moore makinelerinin indirgenmesindeki tek fark P_0 denklik bölümlemesinin oluşturulmasıdır. İndirgenmenin sonraki adımları benzer biçimde yürütülür. Moore makinelerinin indirgenmesi, örnek bir makine üzerinde görülecektir.

Örnek 1.12.

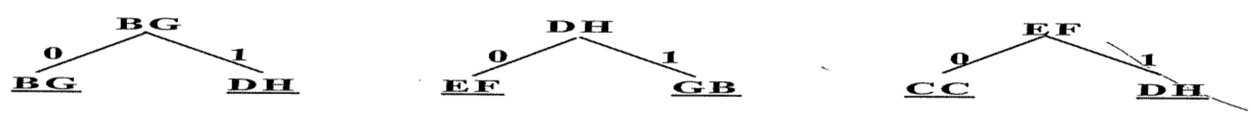
Moore türü $M_{1.12}$ makinesinin giriş alfabesi {0,1}, çıkış alfabesi ise {0,1,2} alfabesidir. Başlangıç durumu A olan makinenin geçiş ve çıkış işlevleri aşağıdaki durum çizelgesi ile tanımlanmaktadır.

	SD		
ŞD	x = 0	x = 1	z
→ A	C	B	0
B	B	D	1
C	A	H	2
D	E	G	1
E	C	D	2
F	C	H	2
G	G	H	1
H	F	B	1

$M_{1.12}$ makinesi aşağıdaki gibi indirgenir:

P_0 = (A)(BDGH)(CEF)

P_1 = (A)(BG)(DH)(C)(EF)



$P_2 = P_1$ = (A)(BG)(DH)(C)(EF)

P = (A)(BG)(DH)(C)(EF)

Denklik bölümlemesinde 5 bölüm olduğu için, M1.12'ye denk en küçük makinenin 5 durumu olacaktır. En küçük makinen durumları:

A için S_0,

BG için S_1,

DH için S_2,

C için S_3,

EF için S_4

diye adlandırılırsa, $M_{1.12}$ makinesine denk en küçük makinenin durum çizelgesi aşağıdaki gibi bulunur.

ŞD	SD, z x = 0	x = 1	z
➔ S_0	S_3	S_1	0
S_1	S_1	S_2	1
S_2	S_4	S_1	1
S_3	S_0	S_2	2
S_4	S_3	S_2	2

1.3.2.3.Deterministik Sonlu Özdevinirlerin (DFA) İndirgenmesi

Bilindiği gibi DFA modeli, Moore makinesinin kısıtlı bir türüdür. Moore modelinde çıkış alfabesi {kabul, red} gibi ikili bir alfabeyle sınırlanırsa, durumlar çıkış işlevi ile “kabul eden” ve “kabul etmeyen” durumlar olmak üzere ikiye ayrılır ve DFA modeli elde edilir. Buna göre DFA’ların indirgenmesi Moore makinelerinin indirgenmesi ile aynı olacaktır.

Önerme 1.13.

Deterministik bir özdevinir (DFA) olan M1.13 aşağıdaki durum çizelgesi ile tanımlanmaktadır.

ŞD	SD x = 0	SD x = 1
→ q_0	q_0	q_1
q_1	q_2	q_4
q_2	q_4	q_7
q_3	q_6	q_5
q_4	q_5	q_3
q_5	q_5	q_7
q_6	q_7	q_2
q_7	q_7	q_5

$M_{1.13}$ makinesi aşağıdaki gibi indirgenir:

$P_0 = (q_0\, q_1\, q_2\, q_3\, q_4\, q_6)(q_5\, q_7)$

$P_1 = (q_0\, q_1)(q_2\, q_3)(q_4\, q_6)(q_5\, q_7)$

$P_2 = (q_0)(q_1)(q_2\, q_3)(q_4\, q_6)(q_5\, q_7)$

$P_3 = P_2$

$P = (q_0)(q_1)(q_2\, q_3)(q_4\, q_6)(q_5\, q_7)$

Denklik bölümlemesinde 5 bölüm olduğu için, $M_{1.13}$ denk en küçük makinenin 5 durumu olacaktır. En küçük makinenin durumları:

q_0 için S_0,

q_1 için S_1,

q_2q_3 için S_2,

q_4q_6 için S_3,

q_5q_7 için S_4

diye adlandırılırsa $M_{1.13}$ makinesine denk en küçük makinenin durum çizelgesi aşağıdaki gibi bulunur.

ŞD	SD, z	
	x = 0	x = 1
→ S_0	S_0	S_1
S_1	S_2	S_3
S_2	S_3	S_4
S_3	S_4	S_2
S_4	S_4	S_4