1.1.2 Deterministik Olmayan Sonlu Otomata
(Non-Deterministic Finite Automata)

Deterministik Olmayan Sonlu Otomata

- 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= <$Q, \Sigma, \delta, q_0, F$>
- Bu tanımda Q , Σ , $\mathbf{q_0}$ ve F 'nin anlamı deterministik model tanımındakilerle aynıdır.
- Deterministik modelde olduğu gibi deterministik olmayan modelde de Q durumlar kümesini, Σ giriş alfabesini, $\mathbf{q_0}$ başlangıç durumunu, F ise uç durumlar kümesini göstermektedir.

- İki model arasındaki tek fark geçiş işlevinin tanımındadır.

Deterministik modelde geçiş işlevi:

- (Q x , Σ) ‘dan Q’ ya bir eşleme olarak tanımlanırken,

Deterministik olmayan modelde geçiş işlevi:

- (Q x , Σ) ‘dan Q’ nun **altkümelerine** bir eşleme 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.**

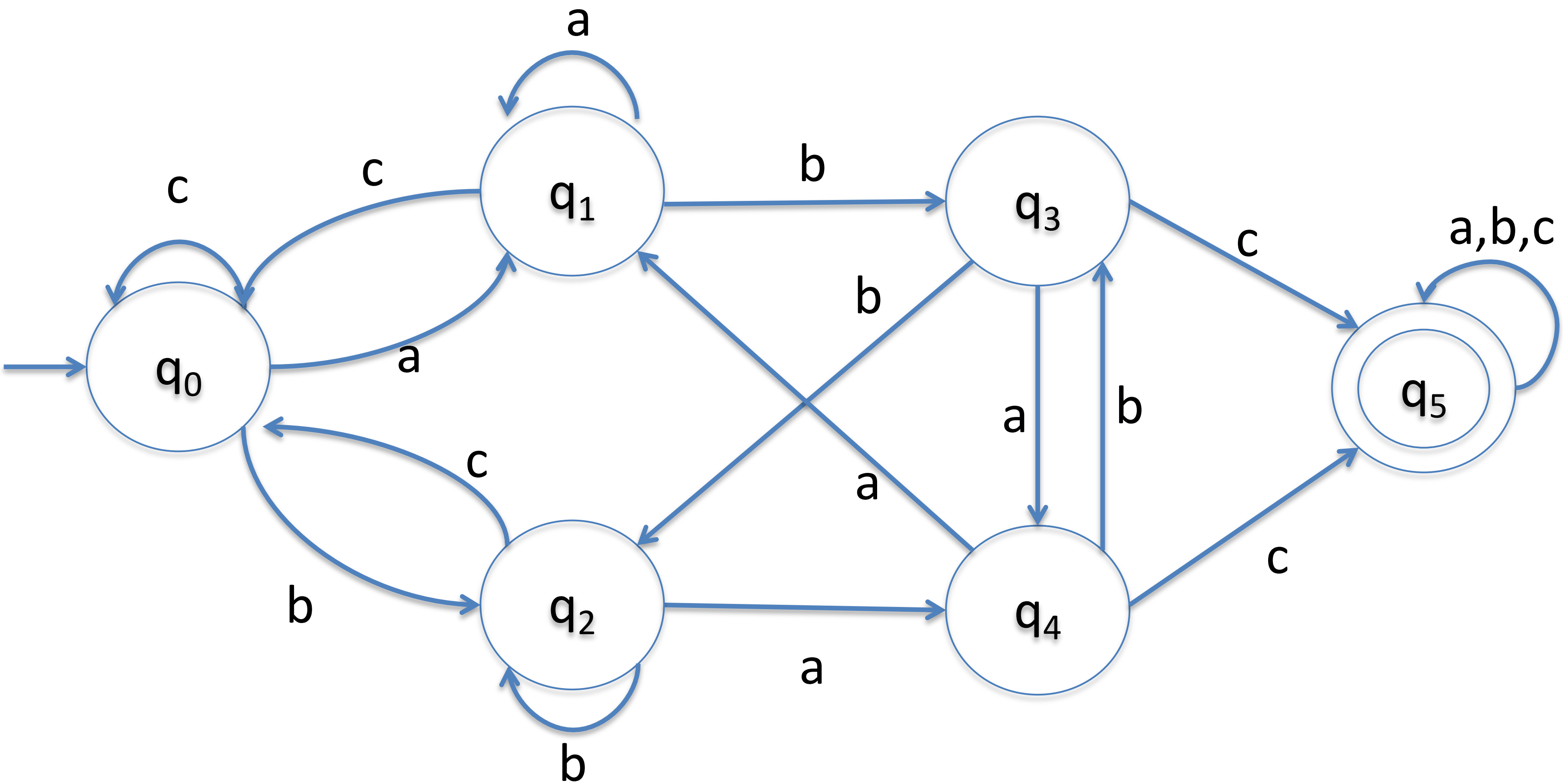
Örnek 1.2

- Sözlü olarak ya da bir düzgün deyimle tanımlanan bir dizgiler kümesini tanıyan DFA'nın deterministik geçiş çizeneğinin oluşturulması oldukça zor olabilir.

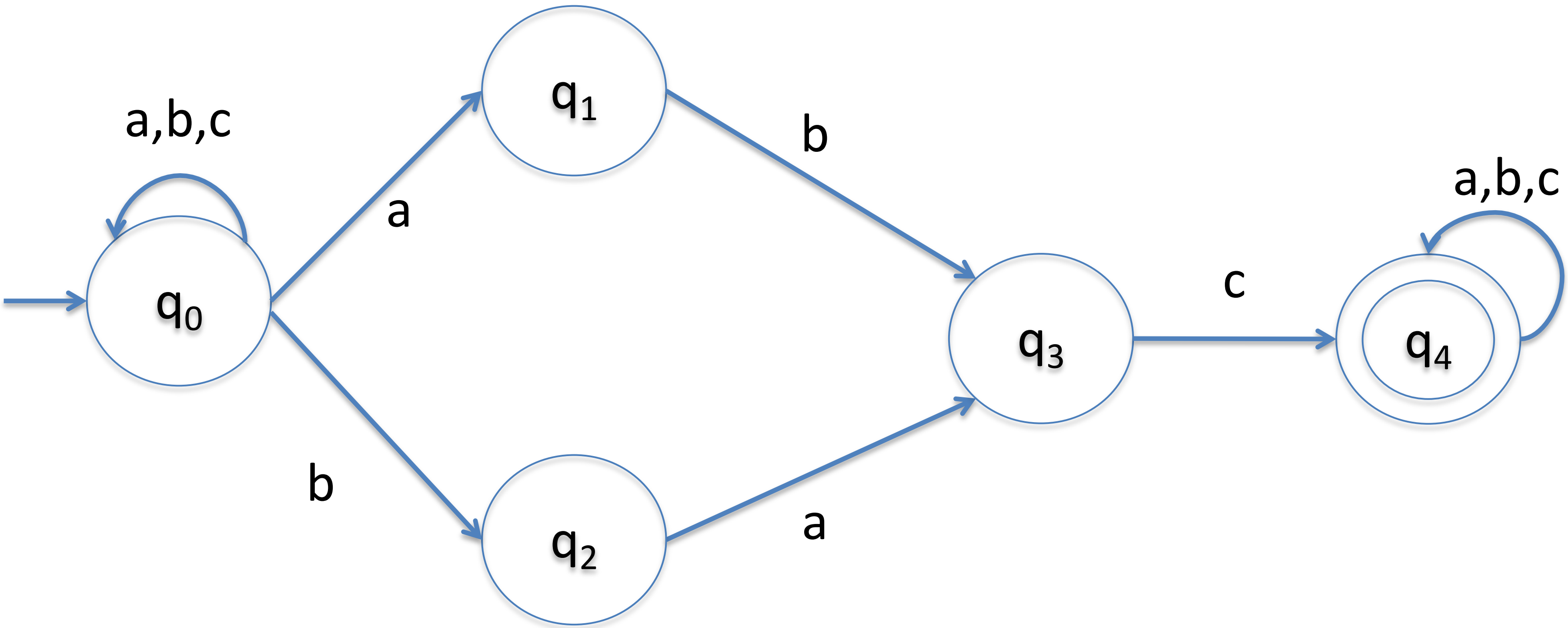
{ a, b, c } alfabesinde, içinde **abc** ya da **bac** altdizgisi bulunan dizgiler kümesi olsun ve bu kümeyi tanıyan DFA'yı $\mathbf{M_{1.2}}$ olarak adlandıralım.

- $\mathbf{M_{1.2}}$'nin tanıdığı dizgilerden birkaç örnek aşağıda gösterilmiştir.
- **T($\mathbf{M_{1.2}}$) = {abc, bac,** cb**abc**b**,** bc**bac**c**, …}**
- $\mathbf{M_{1.2}}$'nin tanıdığı kümenin sözlü tanımı oldukça kolay olmasına karşın, geçiş çizeneğinin oldukça karmaşık olabilir. Üstelik, sözlü tanımdan hareketle bu **çizeneğin elde edilmesini sağlayan sistematik bir yöntem de yoktur.**

Örnek 1.2 DFA



Örnek 1.2 NFA



Örnek 1.3

- Örnek 1.1

M.1.3= <Q, Σ, δ, q0, F>

- Q={q_0, q_1, q_2, q_3 }
- Σ={0,1}
- F={q_3}
- Geçiş fonksiyonları:

δ:

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

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

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

$\delta(q_1,1)= \{\}=e$

$\delta(q_2,0)=\{\}=e$

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

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

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

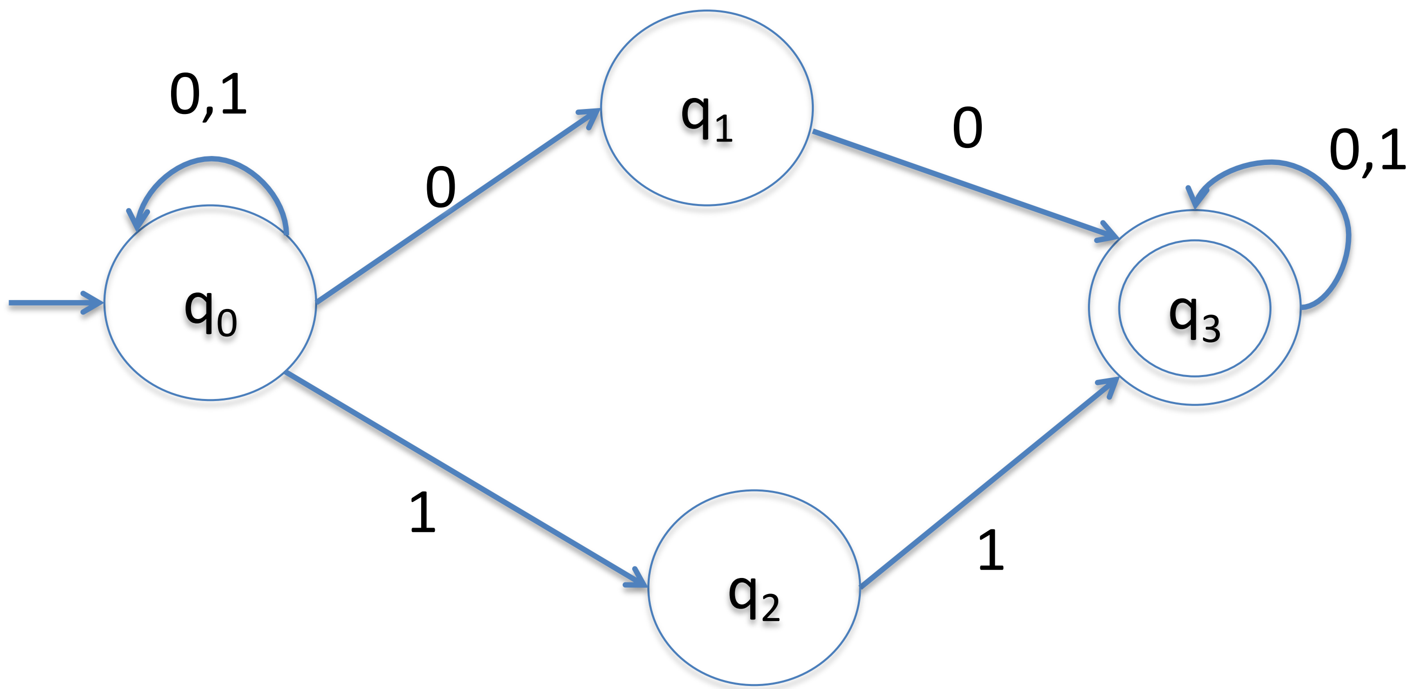
$M_{1.3}$'ün tanımında görüldüğü gibi, deterministik olmayan modelde bazı durumlardan bazı giriş simgeleri ile birden çok duruma geçilebilmektedir.

Bazı durumlardan bazı giriş simgeleri ile de hiçbir duruma geçilemememektedir.

Bu nedenle, deterministik modelde, başlangıç durumu ve uygulanan giriş dizgisi bilindiğinde makinenin hangi durumda bulunacağı kesin olarak bellidir.

Buna karşılık, deterministik olmayan modelde başlangıç durumu ve uygulanan giriş dizgisi bilinen makinenin son durumu belirsiz olabilir.

Örnek 1.3 NFA

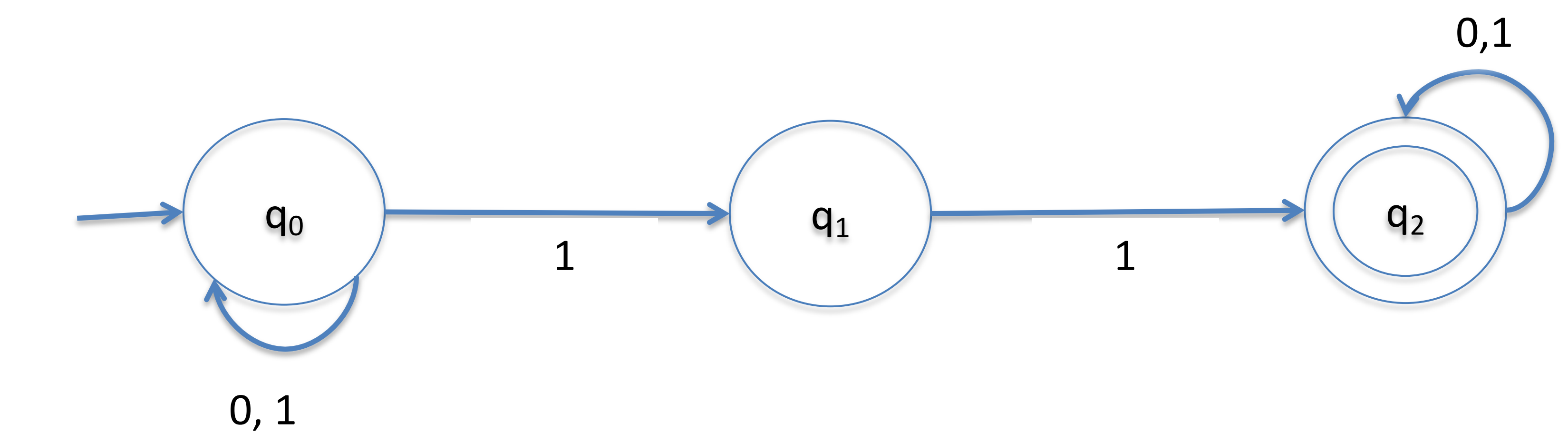


Örneğin başlangı durumu $\mathbf{q_0}$ olan $\mathbf{M_{1.3}}$'e **w = 000** giriş dizgisi uygulandığında, makinenin son
durumu $\mathbf{q_0}$, $\mathbf{q_1}$ ve $\mathbf{q_3}$ durumlarından herhangi biri olabilir.

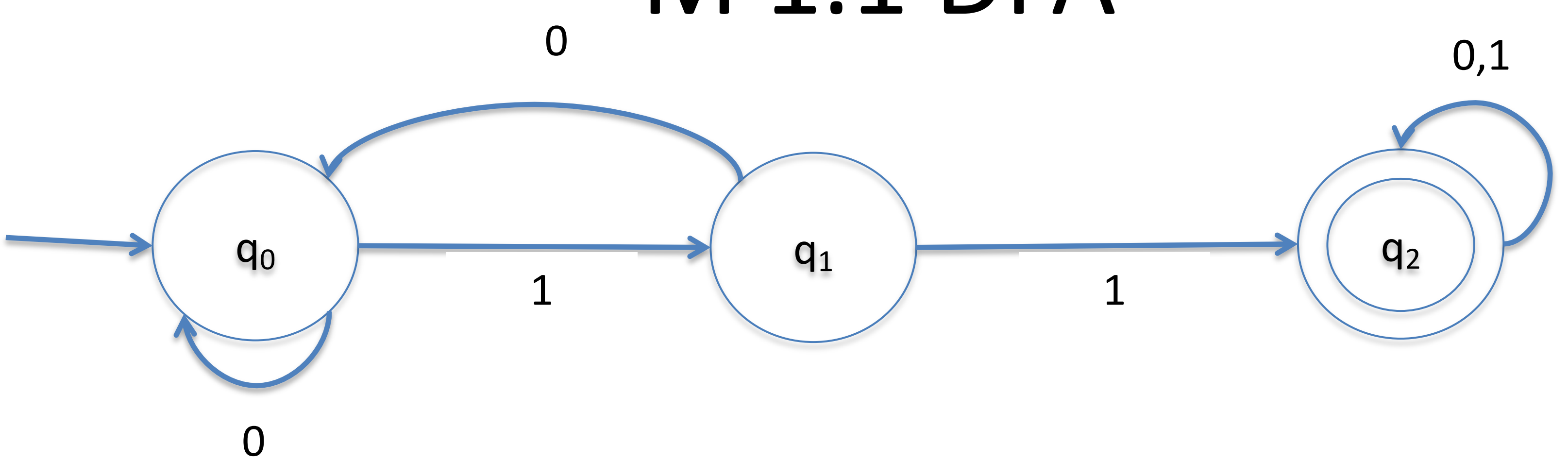
Örnek 1.3

- **Deterministik olmayan bir sonlu özdevinirin (NFA) bir giriş dizgisini tanıması için, geçiş çizeneğinde, bu giriş dizgisi için, başlangıç durumundan başlayarak bir uç duruma ulaşan bir yol bulunabilmesi gerekir.**
  $\mathbf{M_{1.3}}$, **w** = **10001** giriş dizgisini tanır. Çünkü bu giriş dizgisine karşı gelen ve $\mathbf{q_0}$'dan başlayarak tek uç durum olan $\mathbf{q_3}$'te son **bulan birden fazla yollar** bulunabilir.
- Buna karşılık $\mathbf{M_{1.3}}$, **w** = **010** giriş dizgisini tanımaz. Çünkü $\mathbf{q_0}$ ve $\mathbf{q_3}$ durumları arasında bu giriş dizgisine karşı gelen bir yol bulmak mümkün değildir.

M 1.1 NFA



M 1.1 DFA

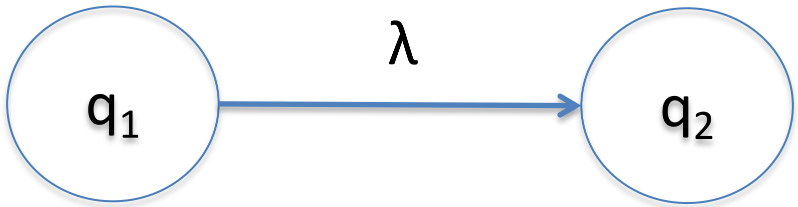


1.1.3 Lambda (λ) Geçişi

- Sonlu özdevinirlerde durum geçişleri, giriş simgelerinin işlenmesi ile gerçekleşir.
- Makine belli bir durumdayken, bir giriş simgesinin uygulanması makinenin bir sonraki duruma geçmesine neden olur.
- Geçiş işlevi ile 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 (λ) Geçişi

- Lambda (λ) geçişi ile, deterministik olmayan modelle genişletilen ve kullanımı kolaylaştırılan sonlu özdevinir tanımı daha da genişletilir.
- Lambda ve lambda-geçişi **soyut** kavramlardır. Lambda boş simge olarak düşünülebilir. Lambda-geçişi ise, hiçbir **giriş** simgesi uygulanmadan (ya da işlenmeden) gerçekleşen durum geçişine karşı gelir.



Lambda (λ) Geçişi

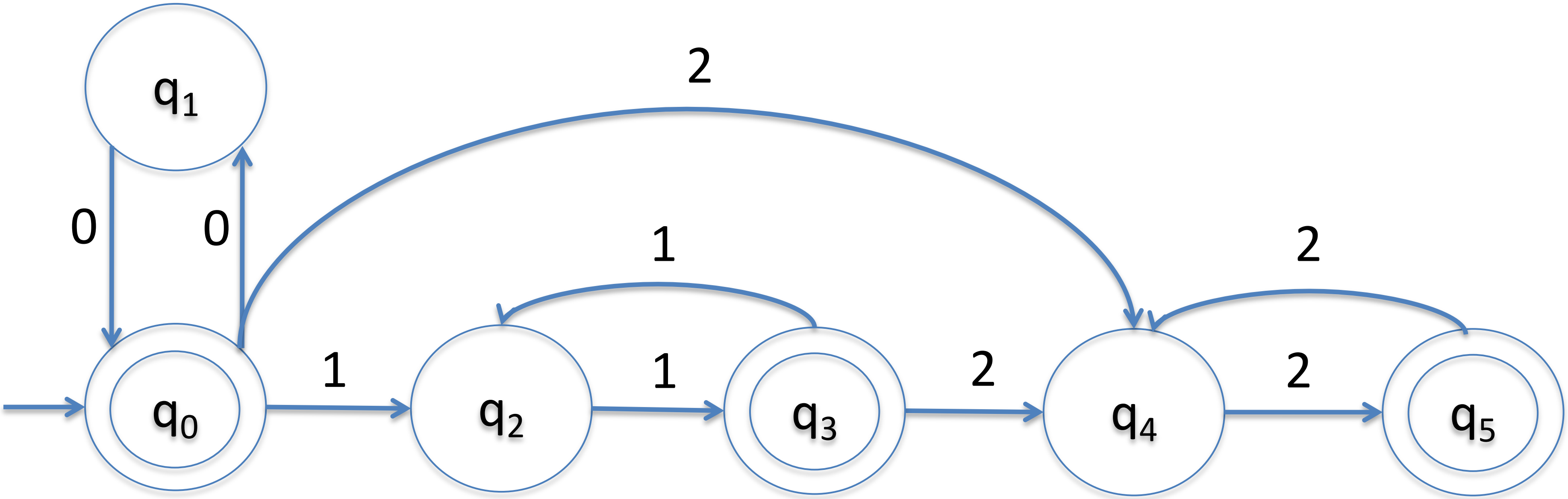
- Lambda geçişi, modelin esnekliğini arttıran; geçiş çizeneklerinin daha kolay oluşturulmasını ve okunmasını sağlayan soyut bir kavramdır. Diğer geçişler gibi, lambda geçişi de **tek yönlüdür.**
- $\mathbf{q_1}$ ve $\mathbf{q_2}$ durumları arasındaki, $\mathbf{q_1}$'den $\mathbf{q_2}$'ye λ geçişi, hiçbir giriş simgesi uygulanmadan makinenin $\mathbf{q_1}$ durumundan $\mathbf{q_2}$ durumuna geçebileceği gösterir.
- Buna göre, $\mathbf{q_1}$ durumunda bulunan makinenin aynı zamanda $\mathbf{q_2}$ durumunda da bulunduğu anlamı çıkar.
- Ancak bunun tersi doğru değildir.
- $\mathbf{q_1}$ ve $\mathbf{q_2}$ durumları arasında, $\mathbf{q_1}$'den $\mathbf{q_2}$'ye λ geçişi varsa, işlevsel olarak:
  - ***eğer q_1 başlangıç durumu ise, q_2 de başlangıç durumu,***
  - ***eğer q_2 bir uç durum ise, q_1 de bir uç durum niteliği kazanır.***

Örnek 1.4

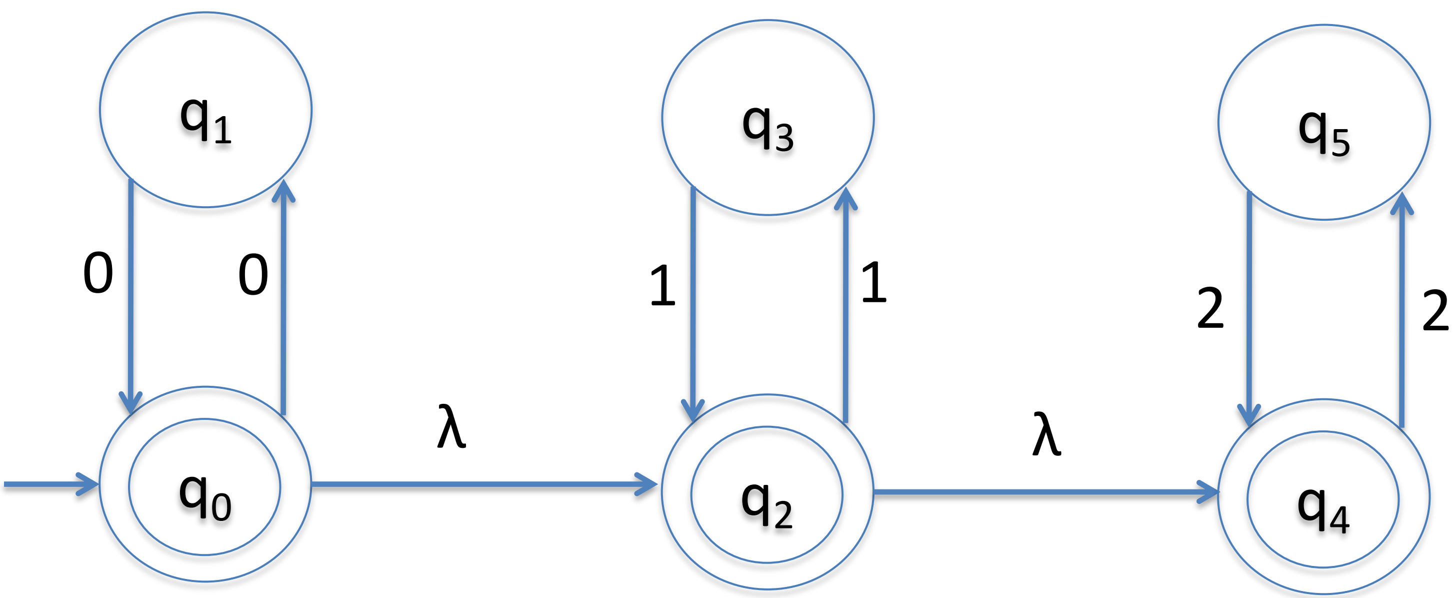
- Lambda geçişlerinin sonlu özdevinir modeline sağladığı esnekliği göstermek için, tanıdığı küme **{ 0, 1, 2 }** alfabesinde:
- $T(M_{1.4})=\{0^{2n}1^{2m}2^{2k} \mid n \geq 0, m \geq 0, k \geq 0 \}$

- $T(M_{1.4})$={e, 00, 11, 22, 0011, 0022, 1122, 001122, 0000, 1111, 2222,0000112222,… }

Örnek 1.4 NFA(λ geçişsiz)



Örnek 1.4 NFA(λ geçişli)

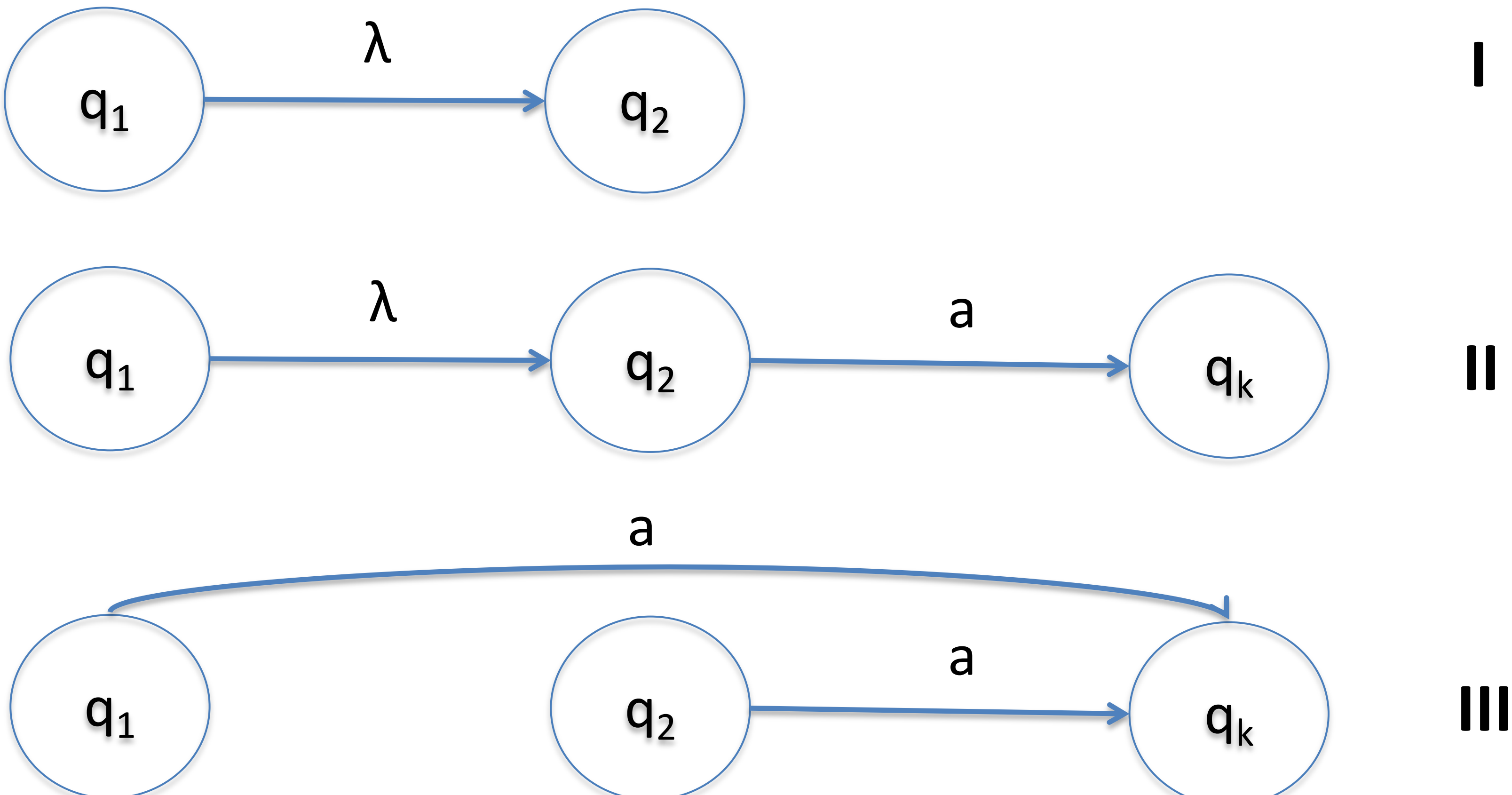


λ Geçişsiz Eşdeğer Geçiş Çizeneğinin Bulunması

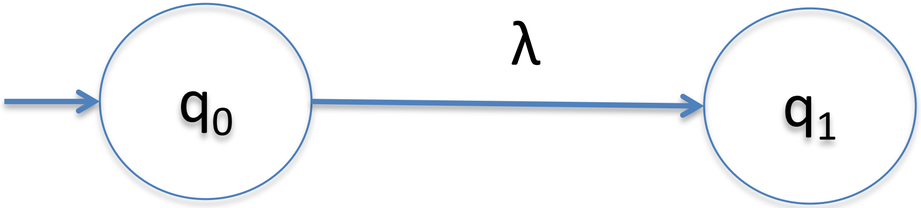
- λ geçişlerinin kullanımı, geçiş çizeneklerinin kullanımını kolaylaştırır ancak gücünü arttırmaz.
- Başka bir deyişle, λ geçişleri kullanılarak oluşturulan her geçiş çizeneğine denk, λ-geçişsiz bir geçiş çizeneği bulunabilir.
- İki geçiş çizeneğinin denkliği, tanıdıkları kümenin eşitliği anlamını taşımaktadır.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

λ Geçişsiz Eşdeğer Geçiş Çizeneğinin Bulunması Kural 1

q_1 q_2 gibi davranır: q_1 durumundayken aynı zamanda q_2 durumunda olabilir. Tersi geçerli değildir.

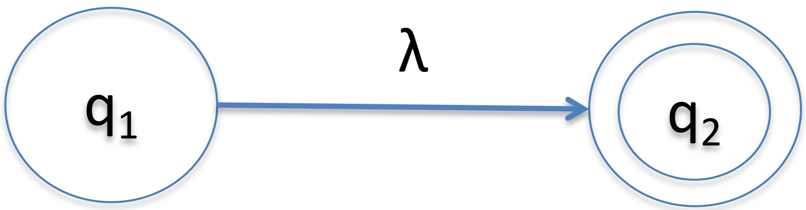


λ Geçişsiz Eşdeğer Geçiş Çizeneğinin Bulunması
Kural 2:



q_0 başlangıç durumu ise q_1 başlangıç durum olur.

λ Geçişsiz Eşdeğer Geçiş Çizeneğinin Bulunması Kural 3

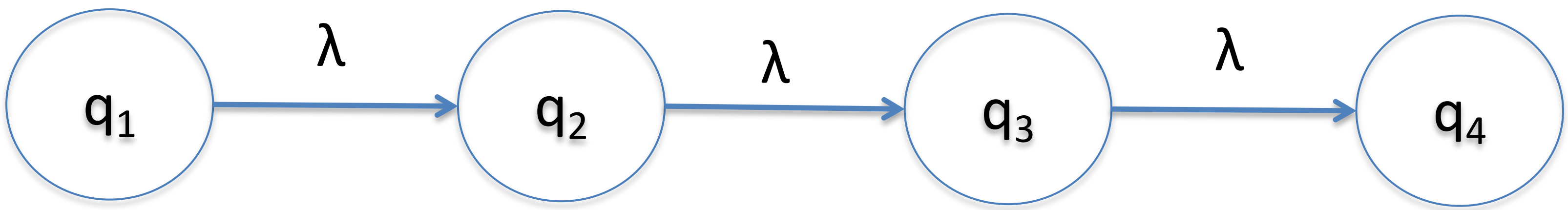


q_2 uç durum ise q_1 de uç durum olur.

λ Geçişsiz Eşdeğer Geçiş Çizeneğinin Bulunması Kural 4

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

- önce $\mathbf{q_3}$ ile $\mathbf{q_4}$ arasındaki,
- sonra $\mathbf{q_2}$ ile $\mathbf{q_3}$ arasındaki,
- son olarak da $\mathbf{q_1}$ ile $\mathbf{q_2}$ arasındaki λ-geçişi yok edilir.

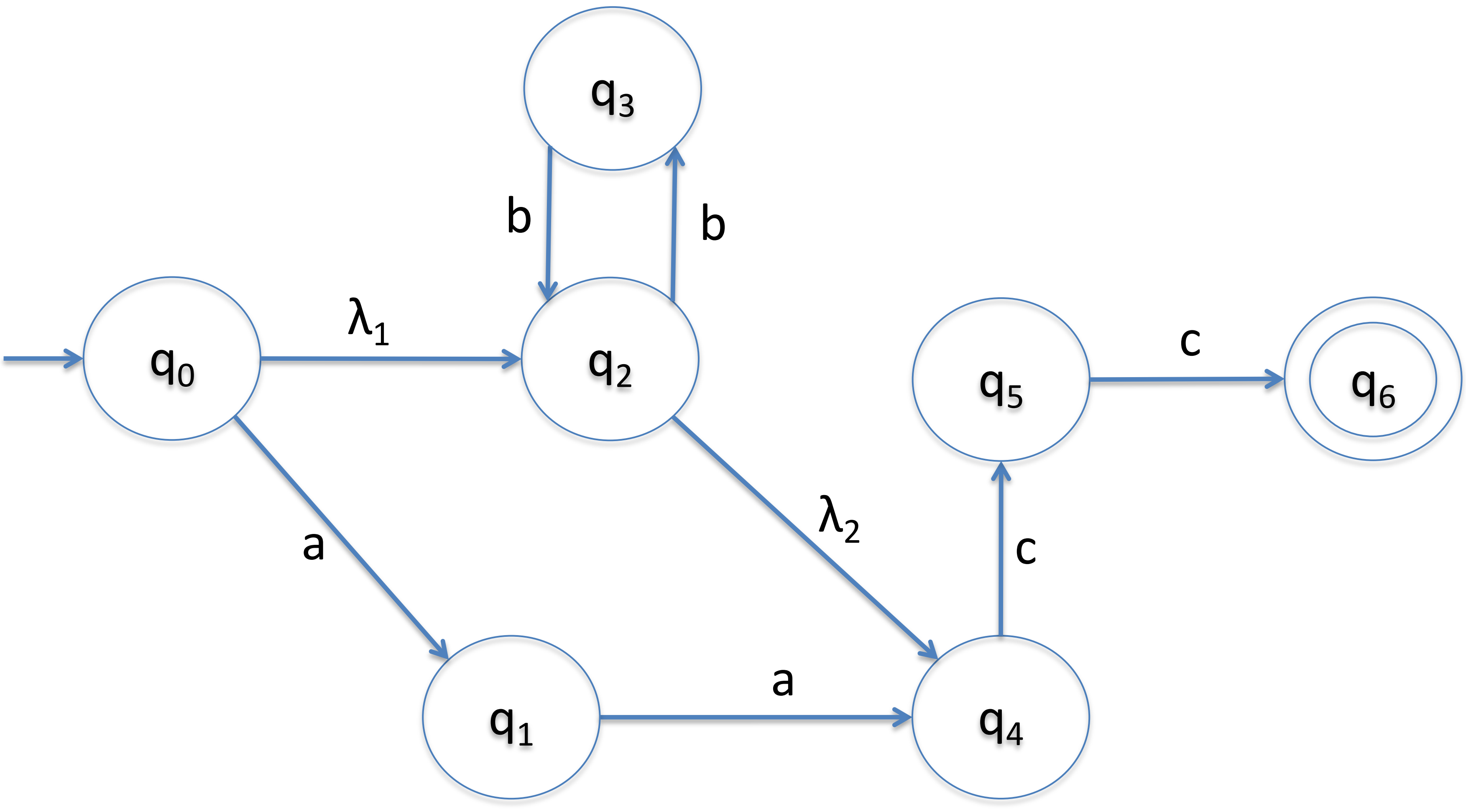


Örnek 1.5

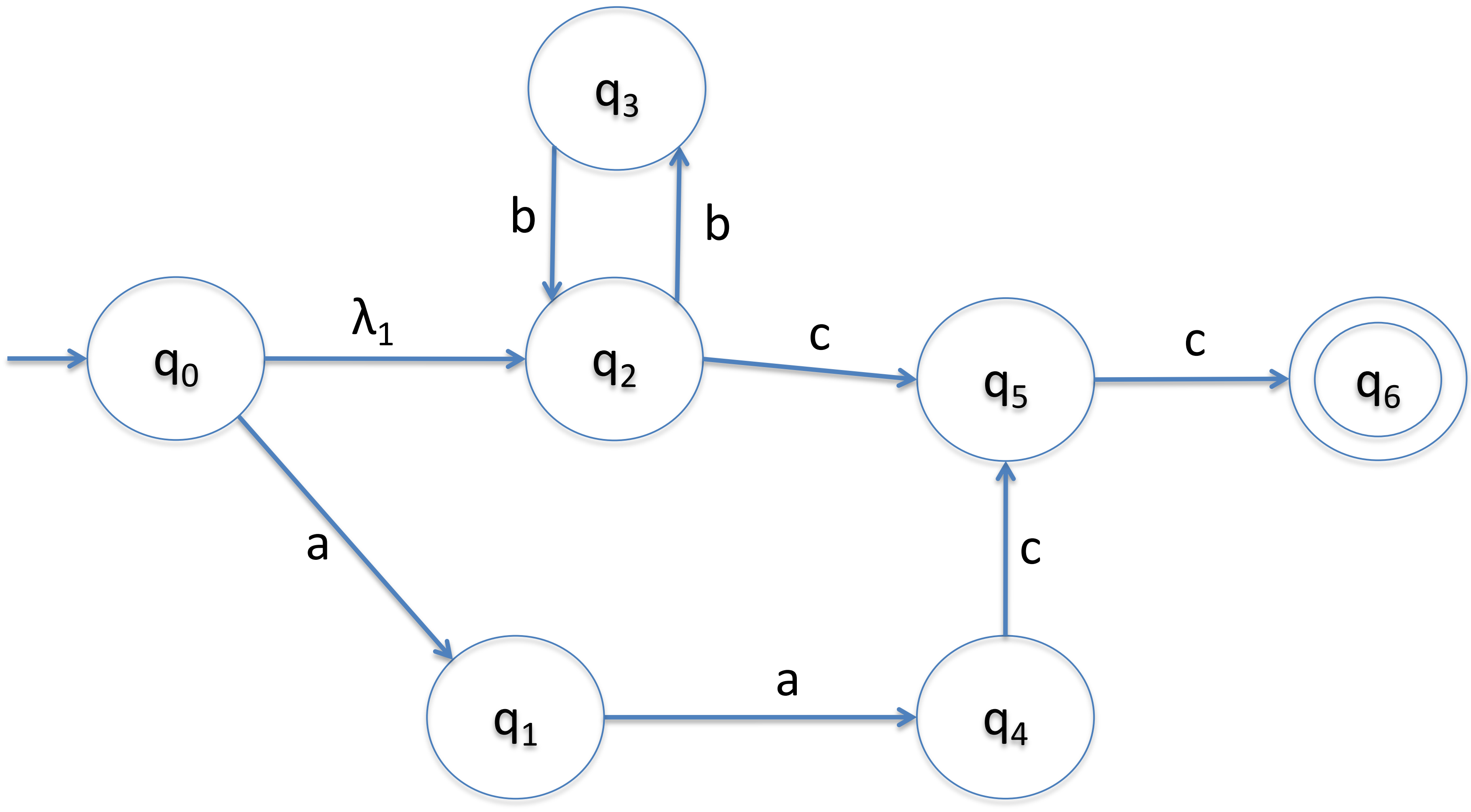
- **$M_{1.5}$** makinesi, **{ a, b, c }** alfabesinde, **aa** ile ya da çift sayıda **b** (0, 2, 4, … tane **b**) ile başlayıp, ardından **cc** ile biten dizgiler kümesini tanıyan makine olsun. **$M_{1.5}$**'in tanıdığı dizgiler kümesi aşağıdaki gibi tanımlanabilir:
- T(**$M_{1.5}$**) = {cc, aacc, bbcc, bbbbcc, bbbbbbcc, …}

Buna göre λ-geçişlerini yok ediniz.

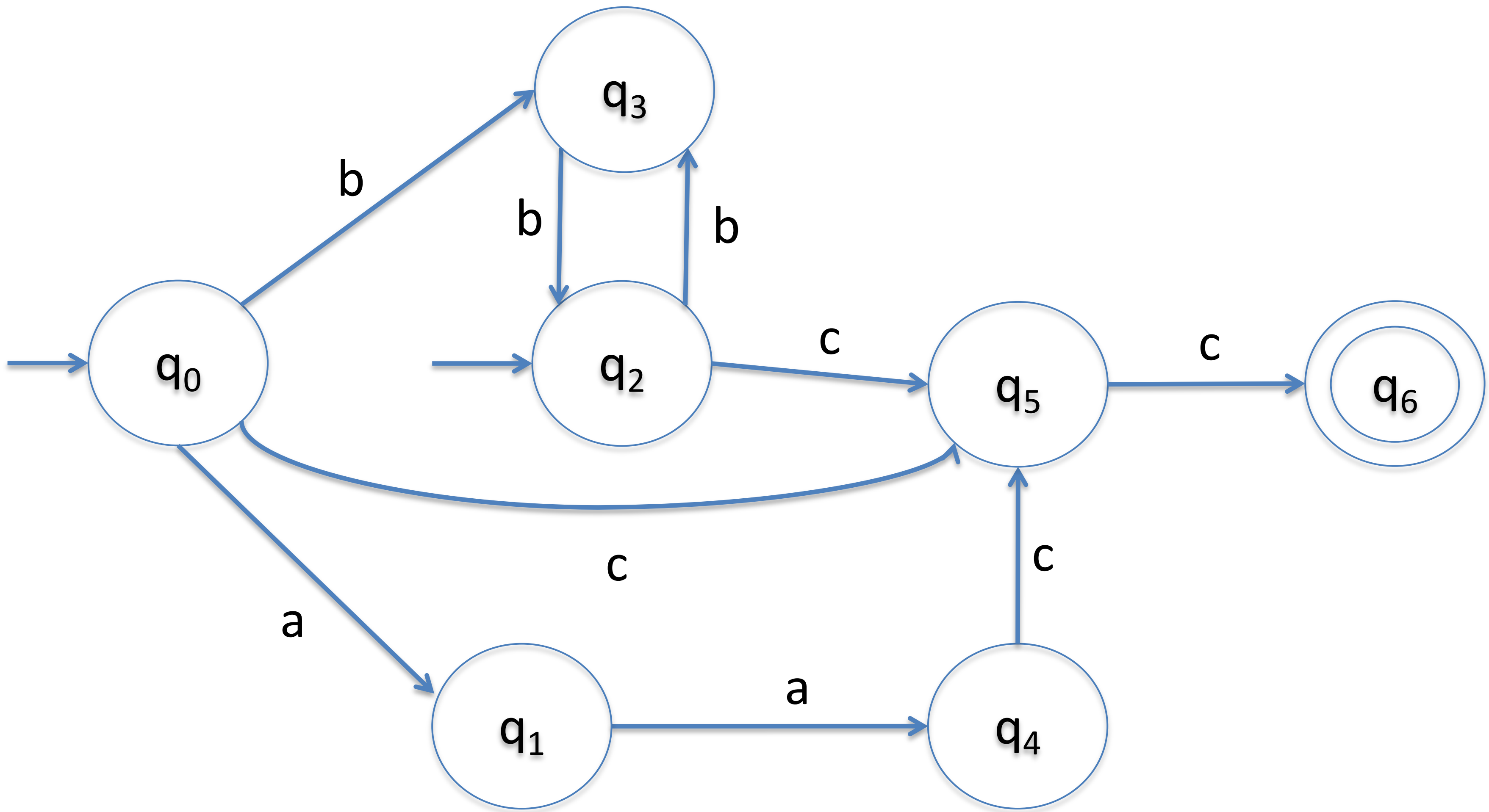
λ-Geçişli M 1.5



λ-Geçişli M 1.5: λ_2 kaldırma



λ-Geçişli M 1.5: λ_1 kaldırma



Deterministik ve Deterministik Olmayan Sonlu Otamata Modellerinin Denkliği

- 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.
- Deterministik olmayan model deterministik modeli kapsar.DFA aynı zamanda NFA olur.
- NFA tarafından tanınan bir kümeyi kapsayan bir DFA bulunabilir mi?

Örnek 1.6 NFA

M.1.6= $<Q, \Sigma, \delta, q0, F>$

- Q={A,B,C }
- Σ={0,1}
- q_0=A
- F={C}
- Geçiş fonksiyonları:

δ:

δ(A, 0)={A}

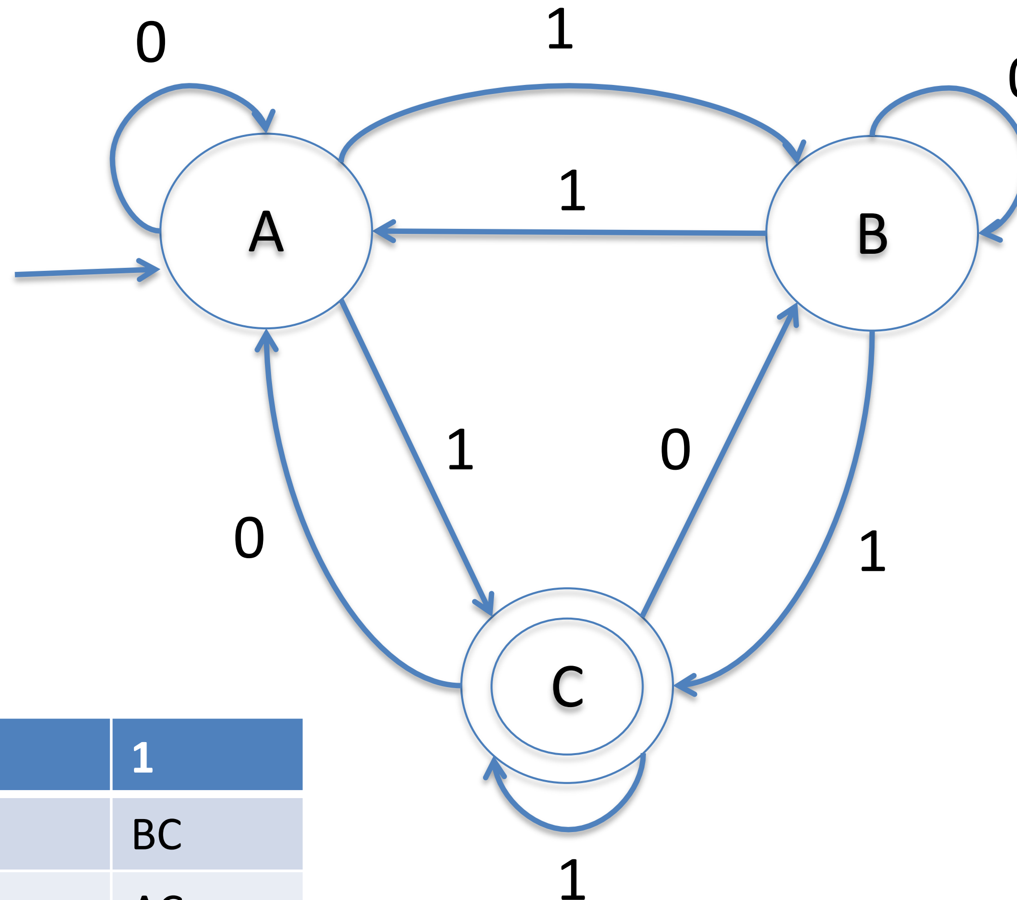
δ(A,1)={B,C}

δ(B,0)={B}

δ(B,1)= {A,C}

δ(C,0)={A,B}

δ(C,1)={C}



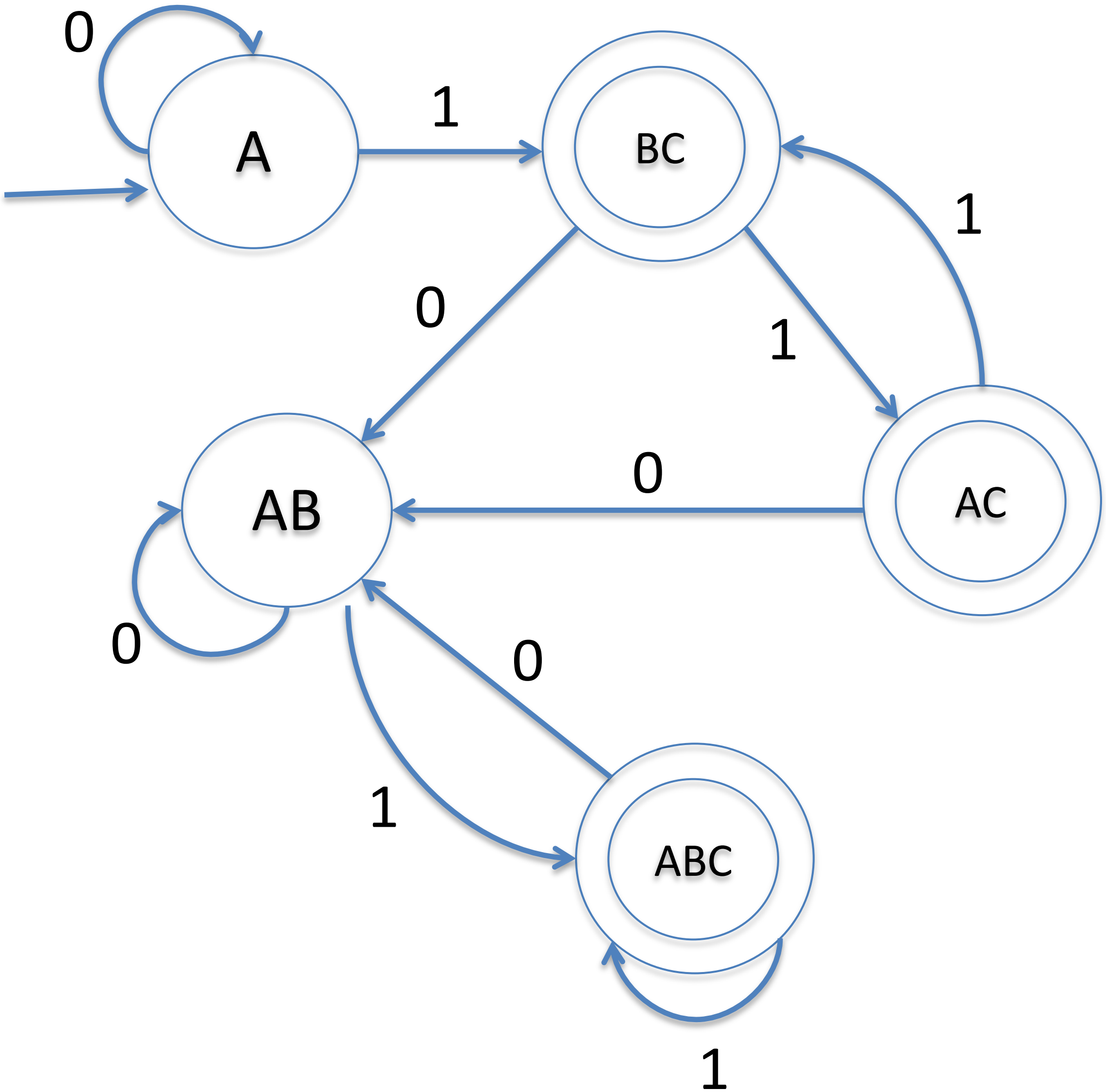
	0	1
→ A	A	BC
B	B	AC
C	AB	C

	0	1
A	A	BC
B	B	AC
C	AB	C

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

M 1.6 DFA

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



M 1.6

M.1.6= <Q, Σ, δ, q0, F>

- Q={A, BC,AB, AC, ABC }
- Σ={0,1}
- q_0=A
- F={BC,AC, ABC}
- Geçiş fonksiyonları:

δ:

δ(A, 0)=A δ(A,1)=BC

δ(BC,0)=AB δ(BC,1)= AC

δ(AB,0)=AB δ(AB,1)= ABC

δ(AC,0)=AB δ(AC,1)= BC

δ(ABC,0)=AB δ(ABC,1)= ABC

Özdevinirler (Otamatlar) Kuramı Ve Biçimsel Diller

Sayfa 40 , 41 Soruları

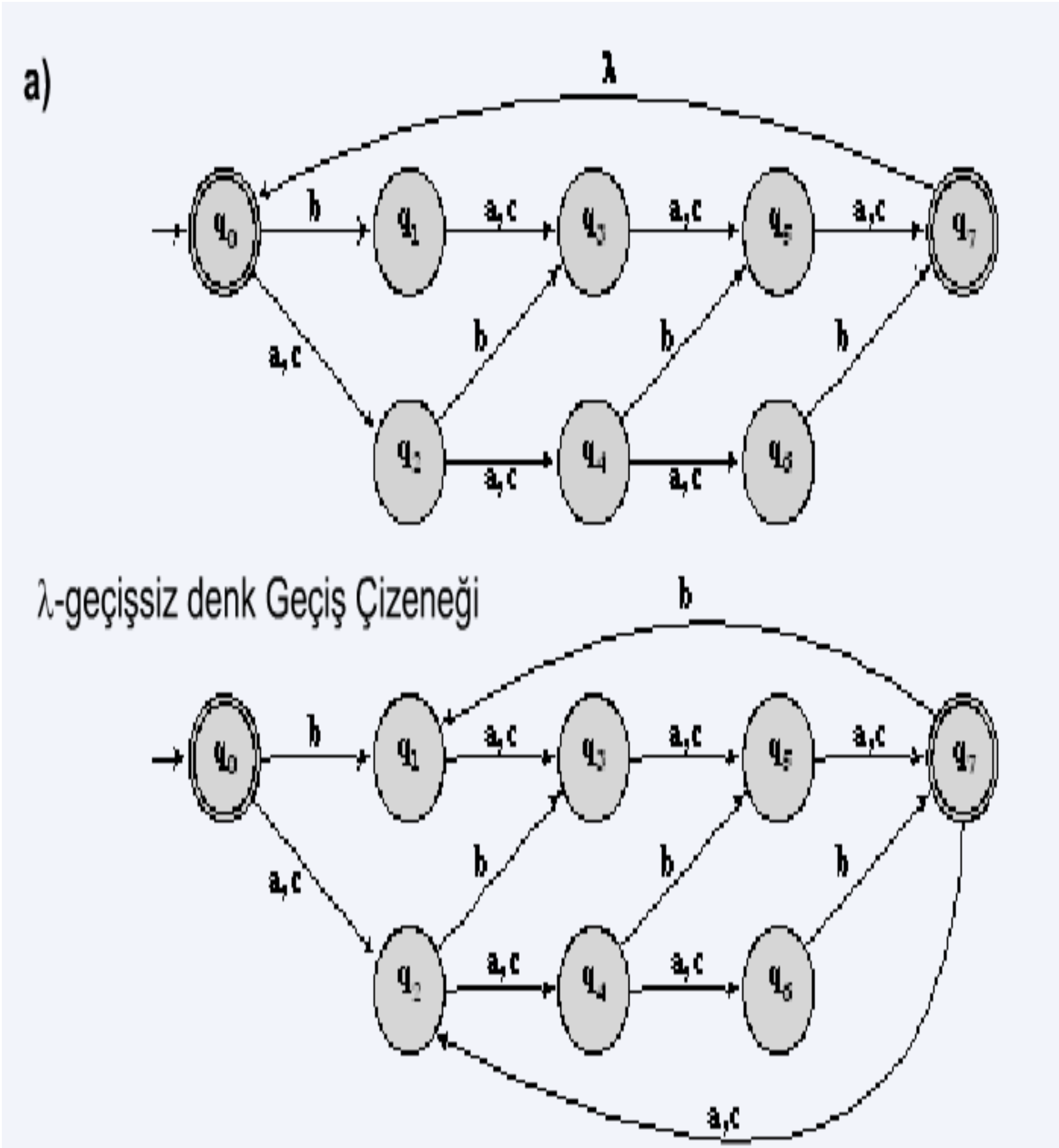
Örnek 1

Aşağıda sözel olarak tanımlanan her küme için, kümeyi tanıyan bir sonlu özdevinirin (NFA) geçiş çizeneğini oluşturunuz. Oluşturduğunuz geçiş çizeneği λ–geçişleri içeriyorsa, bu geçişleri yok ederek λ-geçişsiz denk geçiş çizeneğini bulunuz.

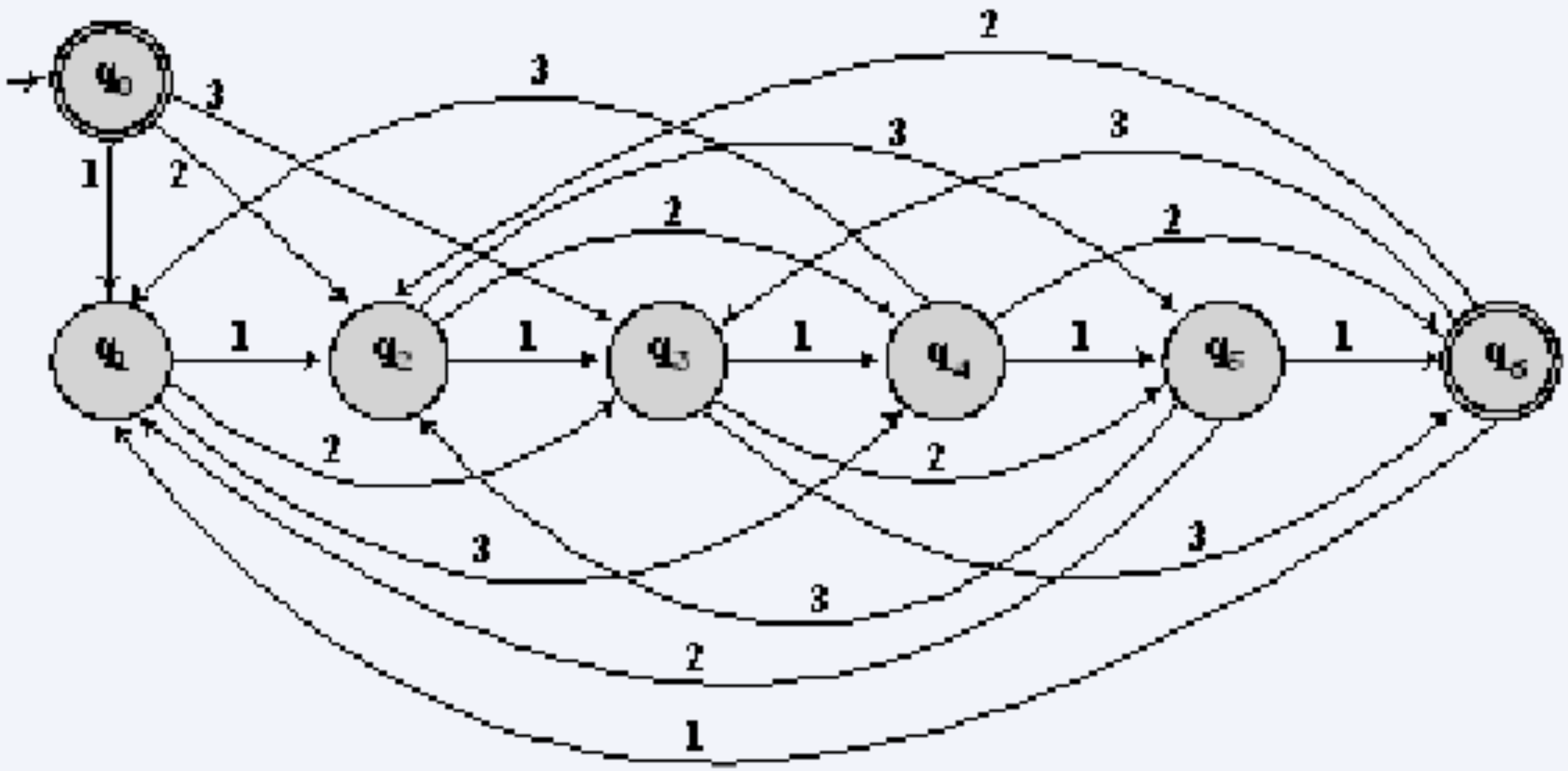
a) **{ a, b, c }** alfabesinde, uzunluğu dördün katı (0, 4, 8, 12, …) olan ve ilk dört, ikinci dört, üçüncü dört, .. simgeden biri ve yalnız biri **b** olan dizgiler kümesi.
$L_{S.1.1.1}$ = { λ, acbc, abac, caab, ..., abccccab, baaacbcaabca, ... }

b) **{ 1, 2, 3 }** alfabesinde, rakamların toplamı altının katları (6, 12, 18, ..) olan dizgiler kümesi.
$L_{S.1.1.2}$ = { λ, 33, 222, 11121, 123, 3123, 233121, 1122321, 32321331, ... }

Çözüm 1.a

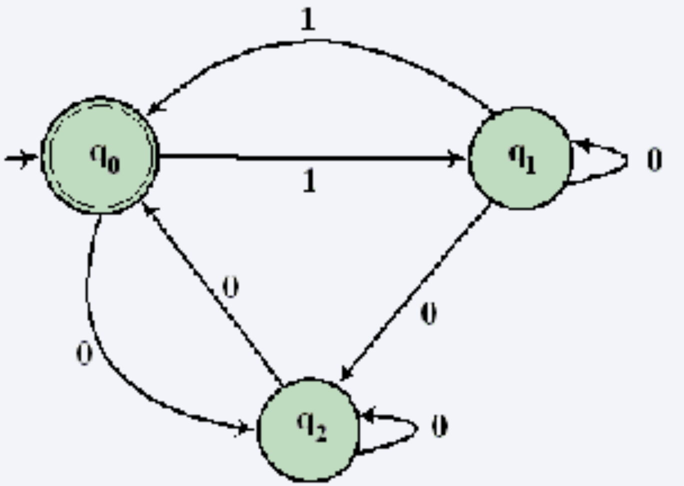

a)
λ
b
a,c
a,c
a,c
a,c
b
b
b
a,c
a,c
λ-geçişsiz denk Geçiş Çizeneği
b
b
a,c
a,c
a,c
a,c
b
b
b
a,c
a,c
a,c

Çözüm 1.b



Örnek 2

Yanda geçiş çizeneği verilen NFA'ya denk bir DFA bulunuz. DFA'nın durumlarını S_0, S_1, S_2, … diye adlandırarak geçiş çizelgesi ve geçiş çizeneğini oluşturunuz.

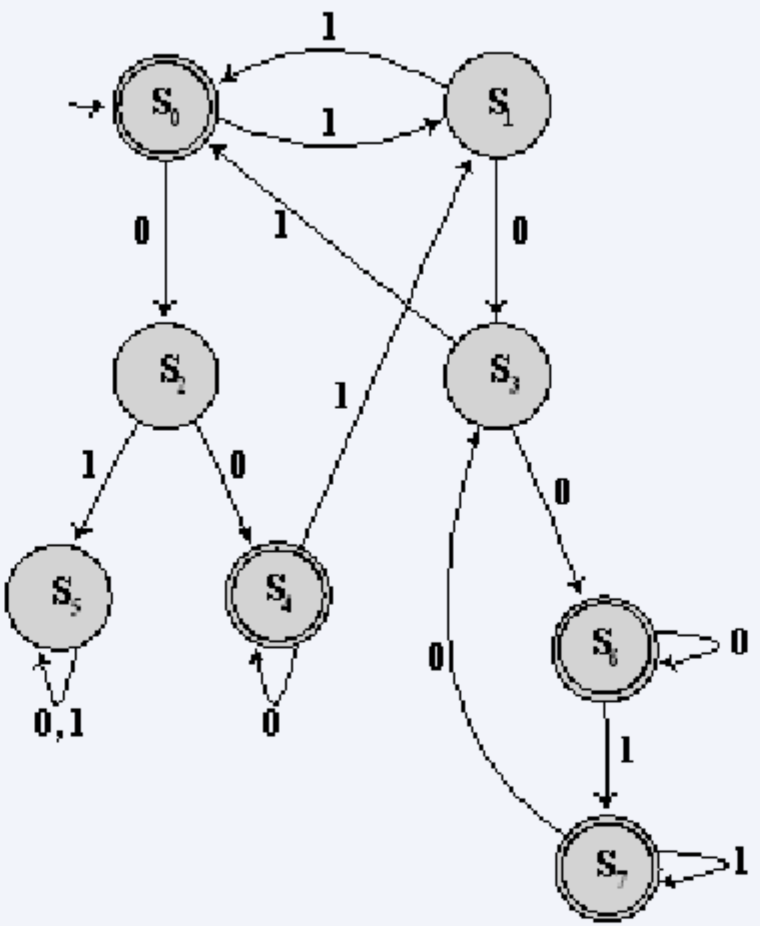


Çözüm 2

Eşdeğer DFA'nın Geçiş Çizelgesi:

	0	1
→ S_0	S_2	S_1
S_1	S_3	S_0
S_2	S_4	S_5
S_3	S_6	S_0
S_4	S_4	S_1
S_5	S_5	S_5
S_6	S_6	S_7
S_7	S_3	S_7

Eşdeğer DFA'nın Geçiş Çizelneği:



Örnek 3

L$_{S.1.3}$ dili **{ a, b, c }** alfabesinde, içindeki her **cd** altdizgisinden sonra en az bir **a**, içindeki her **dc** altdizgisinden sonra da en az bir **b** bulunan dizgiler kümesi olarak tanımlanıyor. **L**$_{S.1.3}$'de yer alan dizgilerden birkaç örnek aşağıda görülmektedir.

L$_{S.1.3}$ **= {** λ**, a, bc, abc, ccdab, dadcbbadcba, bcab, abcdaddcbaddcbd, …. }**

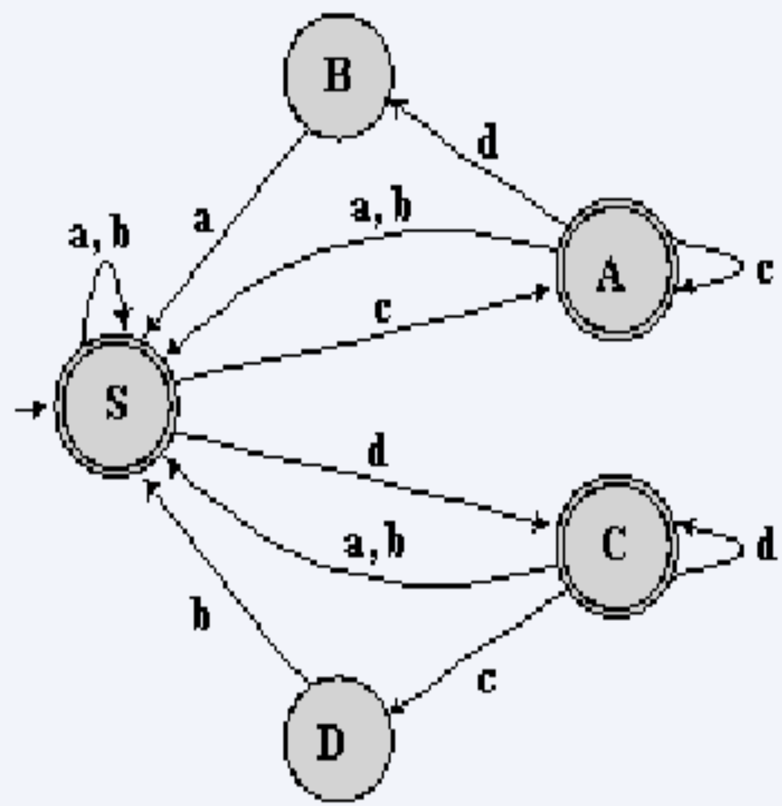
a) **L**$_{S.1.3}$'ü tanıyan NFA'nın geçiş çizeneğini oluşturunuz. Oluşturduğunuz geçiş çizeneği olabildiğince az durum içersin ve λ-geçişleri içermesin.

b) **L**$_{S.1.3}$'ü tanıyan DFA'nın geçiş çizeneğini bulunuz.

L={a,b,c,d} alfabesinde…..

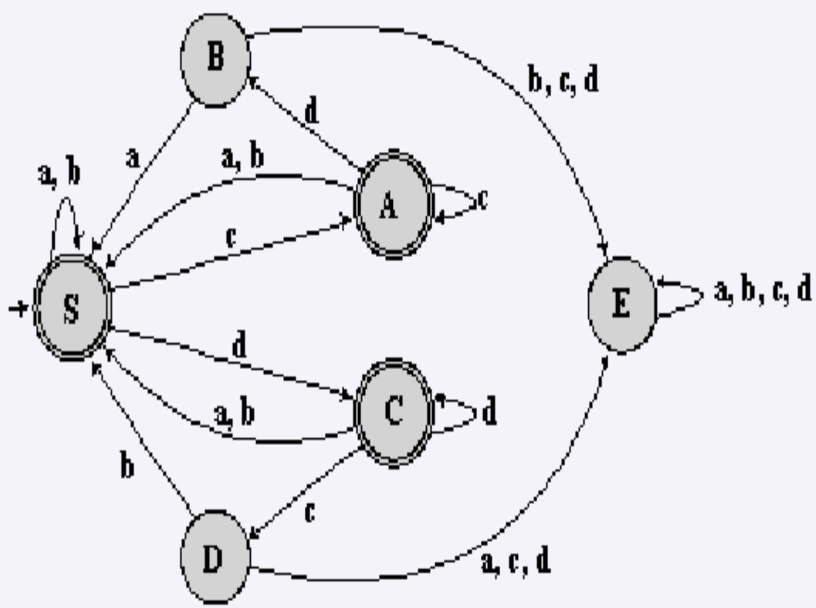
Çözüm 3.a

a) $L_{S.1.3}$'ü tanıyan NFA'nın Geçiş Çizeneği:



Çözüm 3.b

b) L$_{S.1.3}$'ü tanıyan DFA'nın Geçiş Çizeneği:



Örnek 4

$\mathbf{L}_{S.1.4}$ dili, **{ a, b, c, d }** alfabesinde, içindeki her **abc** altdizgisinden sonra en az 2 tane **d** bulunan dizgiler kümesi olarak tanımlanıyor. $\mathbf{L}_{S.1.4}$ 'de yer alan dizgilerden birkaç örnek aşağıda görülmektedir.

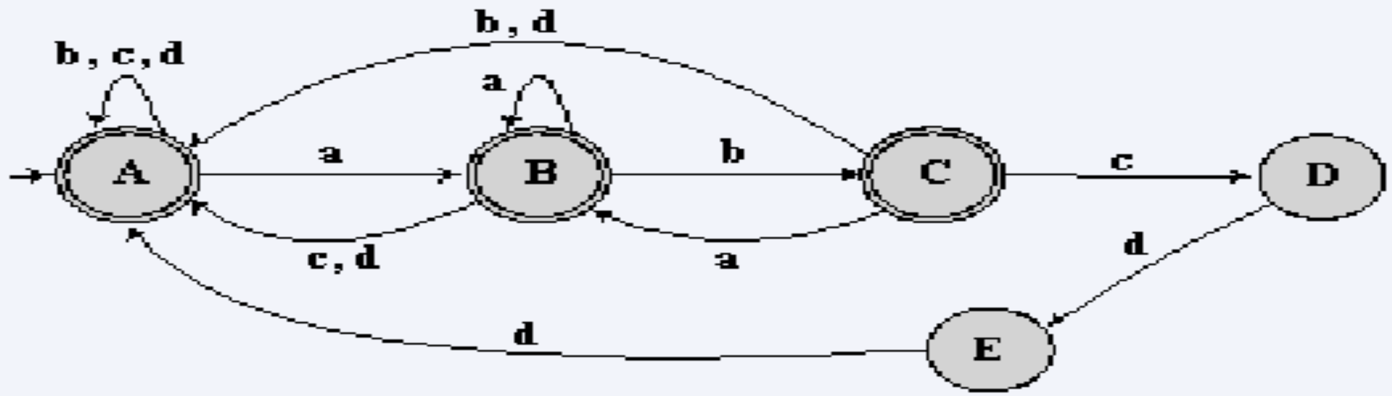
$\mathbf{L}_{S.1.4}$ **= {** λ**, a, bc, abbd, cda, dabcdda, bcabb, ababcddddcadabcdd, .. }**

a) $\mathbf{L}_{S.1.4}$'ü tanıyan NFA'nın geçiş çizeneğini oluşturunuz. Oluşturduğunuz geçiş çizeneği olabildiğince az durum içersin ve λ–geçişleri içeremesin.

b) $\mathbf{L}_{S.1.4}$'ü tanıyan DFA'nın geçiş çizeneğini bulunuz.

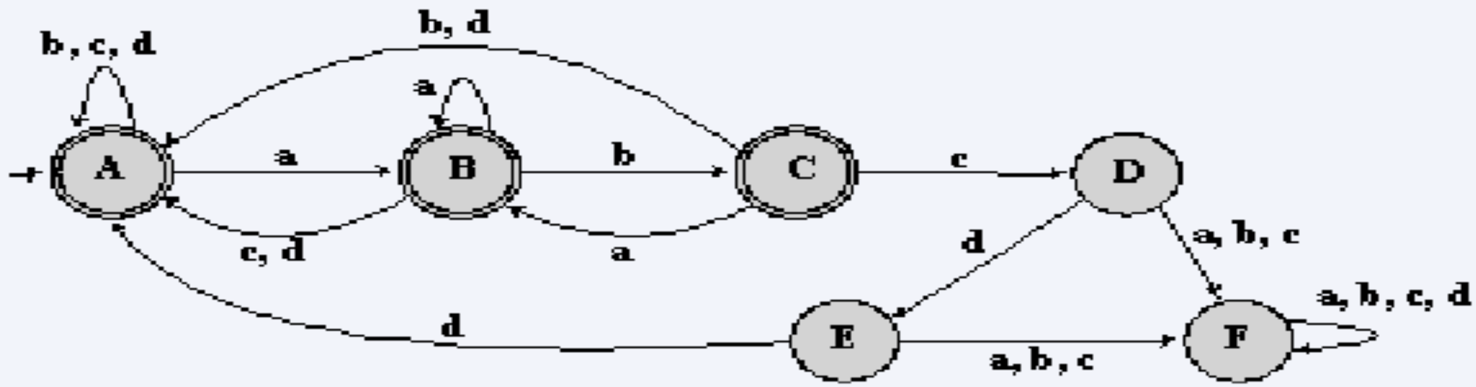
Çözüm1.4.a

a) L$_{s.1.4}$'ü tanıyan NFA'nın Geçiş Çizeneği:

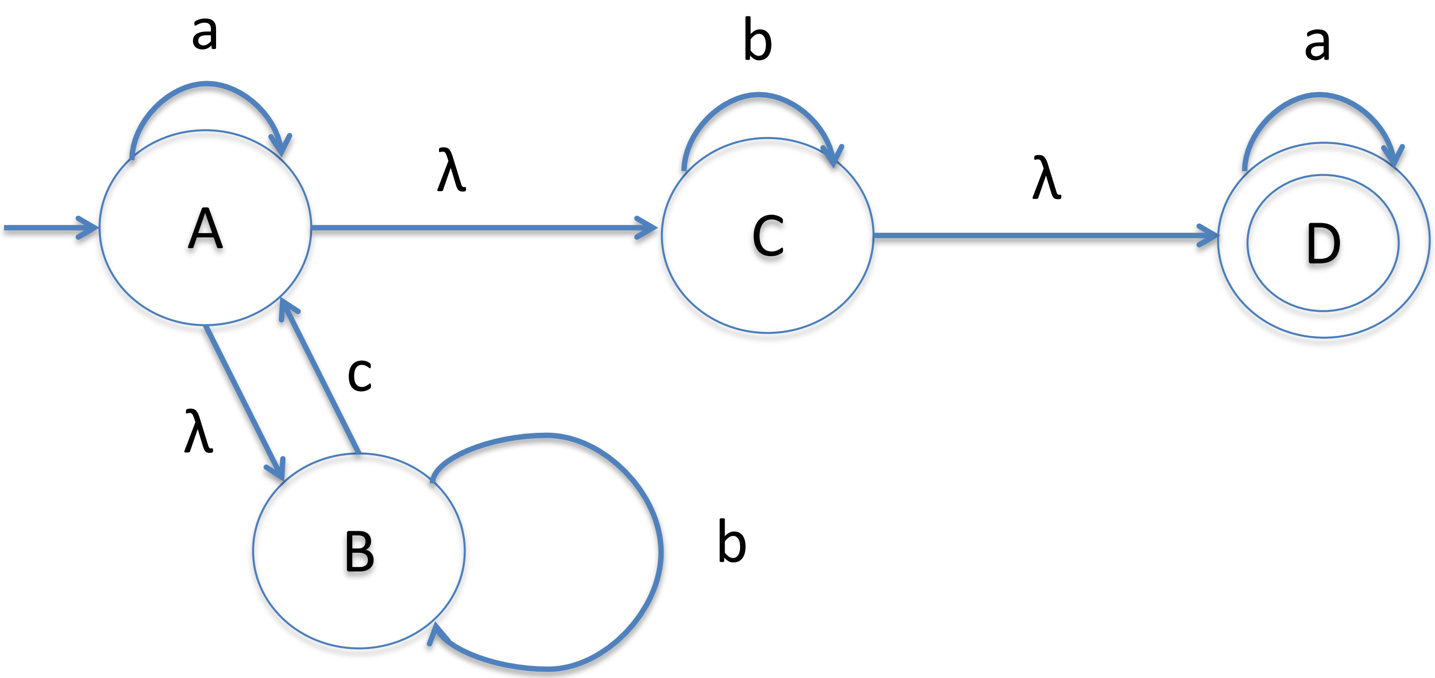


Çözüm 4.b

b) L$_{S.1.4}$'ü tanıyan DFA'nın Geçiş Çizeneği:



S1.6(a)



Yukarıdaki geçiş çizeneği göre λ-geçişlerini yok ederek λ-geçişsiz denk geçiş çizeneğini bulunuz.