FORMAL DİLLER VE OTOMATLAR

DÜZGÜN DİLLERİN ÖZELLİKLERİ

Teorem: L_1 ve L_2 düzgün diller ise;

$L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 L_2$, $\overline{L_1}$, L_1^R , L_1^* da düzgün dillerdir.

Tanım: $\sum_1$ ve $\sum_2$ iki alfabe olsun.

h: $\sum_1 \rightarrow \sum_2^*$ fonksiyonu homomorfizm olarak adlandırılır.

Homomorfizm tek bir sembolün yerine bir dizgi yerleştirme işlemidir.

$w = a_1 a_2 \ldots . a_n$ ise

$h(w) = h(a_1) h(a_2) \ldots \ldots h(a_n)$

$\sum_1$ alfabesinde L bir dil olmak üzere, bu dilin homomorfik görüntüsü şu şekilde tanımlanır:

$$h(L) = \{ h(w) : w \in L \}$$

- Örnek: $\sum_1$ = {a,b} ve $\sum_2$ = {a,b,c} ve h aşağıdaki gibi tanımlanmıştır:

h(a) = ab

h(b) = bbc

Bu durumda h(aba)=abbbcab'dir.

L={aa, aba} dilinin homomorfik görüntüsü:

h(L) = {abab, abbbcab}'dir.

- Bir L dili için r düzgün ifade ise, r’nin her sembolüne homomorfizm uygulayarak h(L) için düzgün ifade elde edilebilir.

Örnek: $\sum_1$ = {a, b} ve $\sum_2$ = {b, c, d} alfabelerini ele alalım.

h(a) = dbcc

h(b) = bdc olarak tanımlansın.

L; r = (a+b*)(aa)* ile gösterilen düzgün bir dil ise, h(L) düzgün dilini gösteren düzgün ifade;

r = (dbcc + (bdc)*)(dbccdbcc)* dır.

Teorem: h bir homomorfizm olsun. L bir düzgün dil ise, bu dilin homomorfik görüntüsü olan h(L) de düzgün bir dildir.

Düzgün Diller ile İlgili Temel Sorular

Membership Question

Verilen bir L dili ve w dizgisi için w ϵ L?

- L dilinin gösterimi düzgün ifade ya da düzgün gramer formunda ise buna eşdeğer DFA oluşturulur.
- w dizgisinin bu DFA tarafından kabul edilip edilmediği kontrol edilir.

The finiteness question

∑ üzerinde tanımlı herhangi bir L düzgün dili için, bu dilin boş, sonlu ya da sonsuz olduğunu belirlemek için bir algoritma?

- L dilinin gösterimi düzgün ifade ya da düzgün gramer formunda ise, buna eşdeğer DFA oluşturulur.
- Başlangıç durumundan herhangi bir kabul durumuna basit bir yol varsa, bu dil boş değildir. Aksi halde boş dildir.
- Bir döngüye ait olan tüm durumlar bulunur. Eğer bu durumlardan bazıları, başlangıç durumundan kabul durumuna giden bir yol üzerinde ise o halde bu dil sonlu değildir (infinite), aksi durumda sonludur.

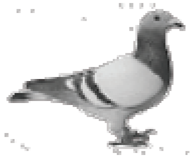
L_1 ve L_2 düzgün dilleri için, $L_1 = L_2$ olup olmadığını belirleyen algoritma

- L_1 , L_2'yi kullanarak yeni bir L_3 dili tanımlanır:

$$L_3 = (L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2)$$

- Tanımlanan bu L_3 dili de düzgün bir dildir.
- Bu nedenle L_3 dilini kabul eden bir DFA bulunabilir.
- Bir önceki teoremi kullanarak L_3 dilinin boş olup olmadığı belirlenir.
- $L_3 = 0$ ise $L_1 = L_2$'dir. Aksi durumda $L_1 \neq L_2$'dir.

Düzgün Olmayan (Nonregular) Dillerin Belirlenmesi

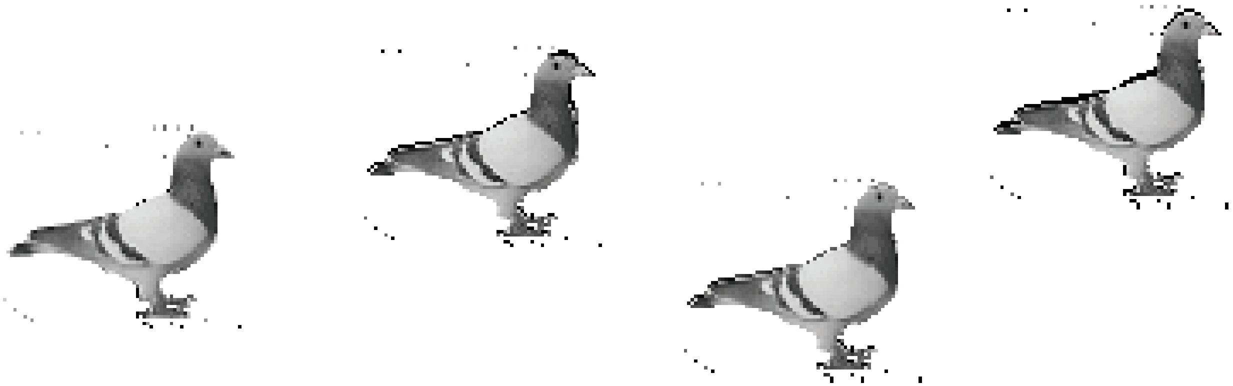


The Pigeonhole Principle

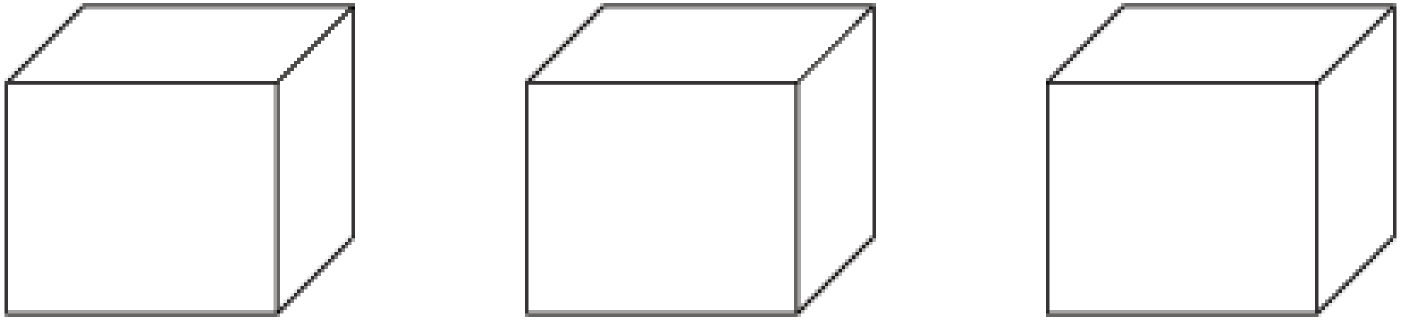
n+1 adet nesneyi n adet kutuya koymak istersek, bir kutuda birden fazla nesne olacaktır.

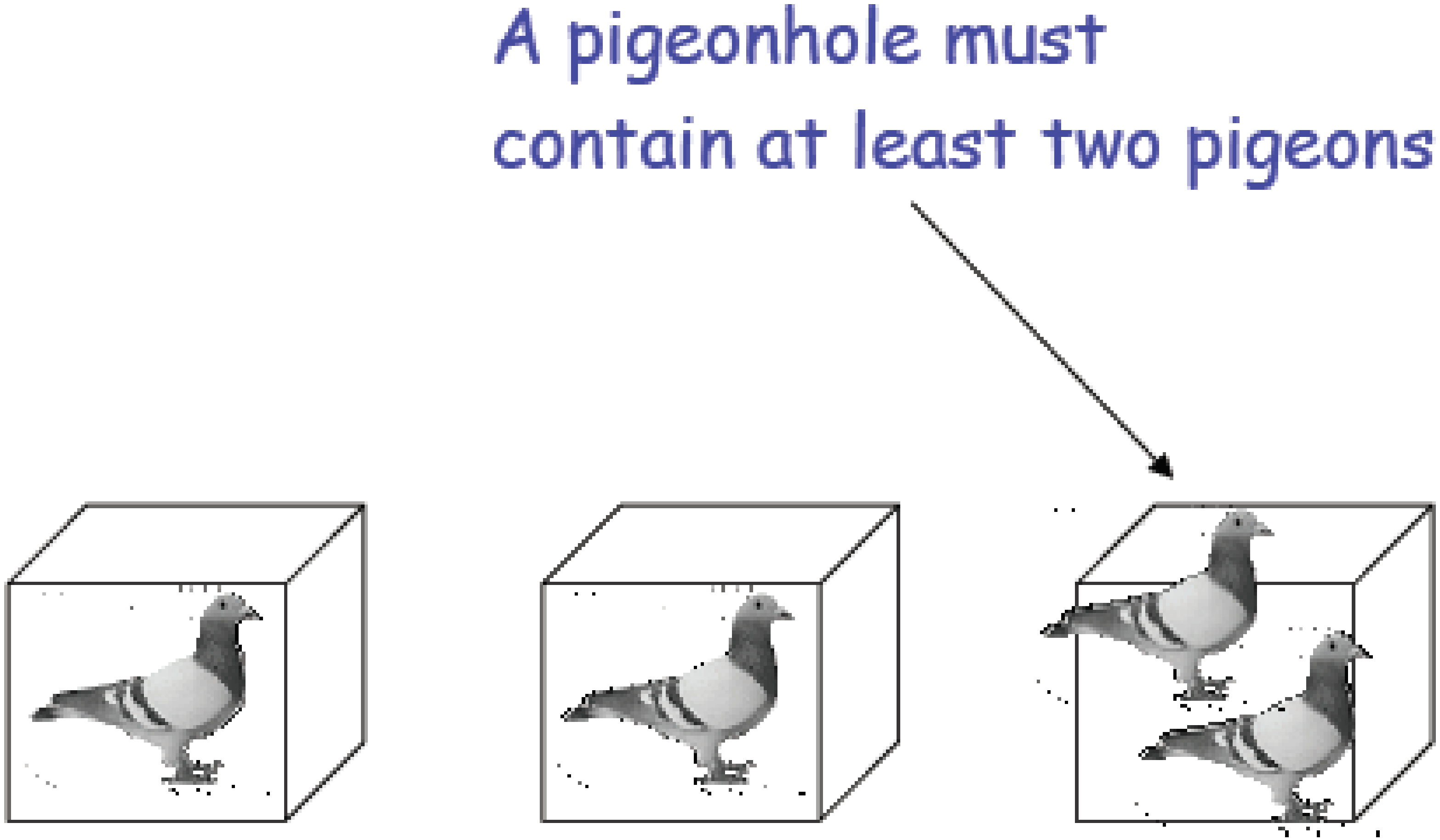
$$\left.\begin{array}{l} f : A \rightarrow B \\ |A| = n+1 \\ |B| = n \end{array}\right\} \text{f bire-bir fonksiyon olamaz}$$

4 pigeons



3 pigeonholes



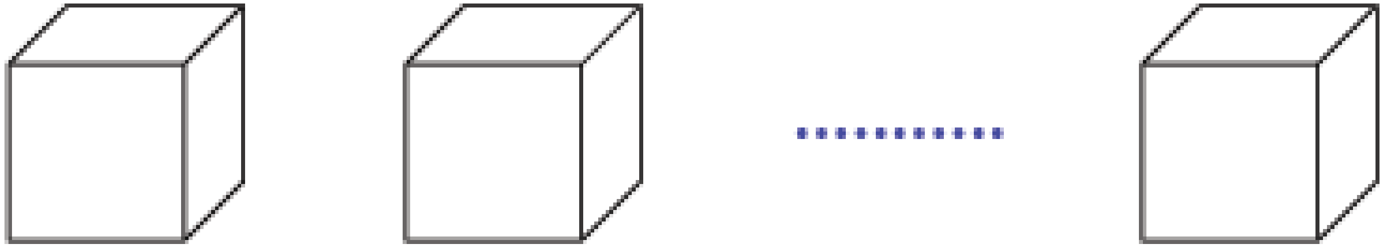


A pigeonhole must contain at least two pigeons

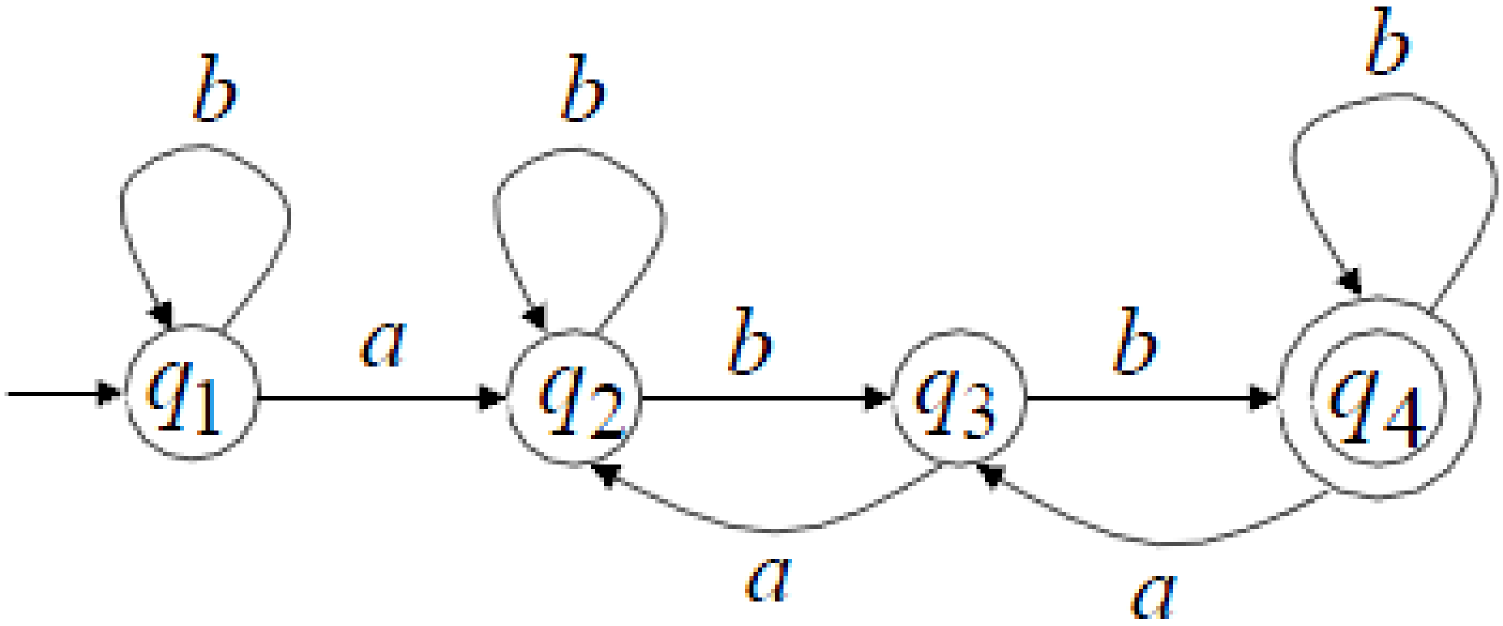
n pigeons

m pigeonholes

$n > m$



DFA with 4 states

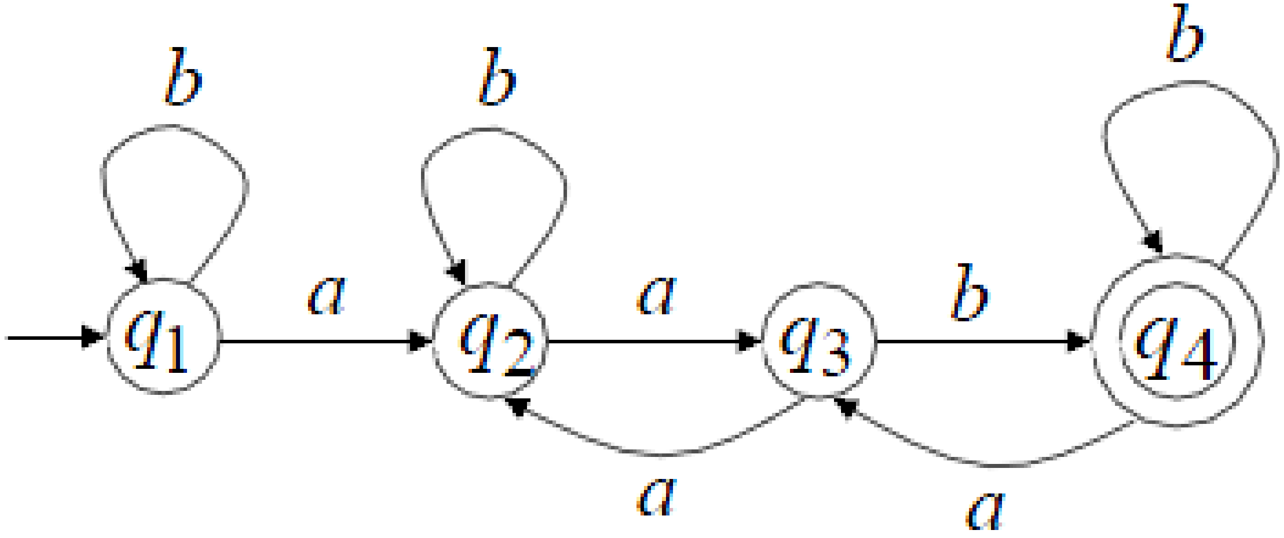


In walks of strings: a

aa

aab

no state
is repeated



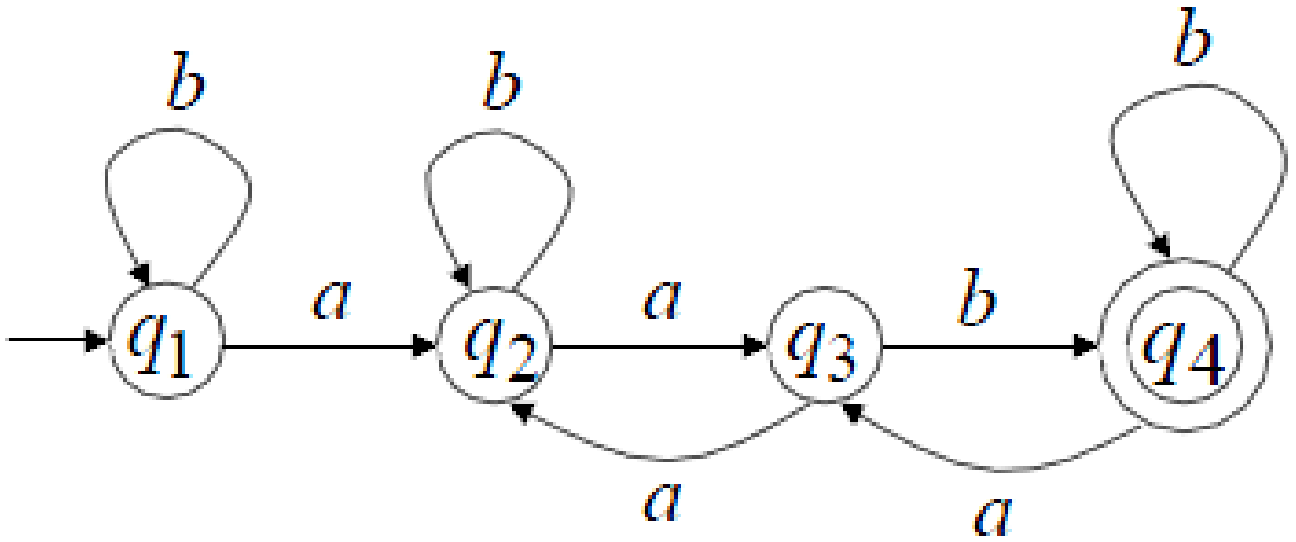
In walks of strings: $aabb$

$bbaa$

$abbabb$

$abbbabbabb...$

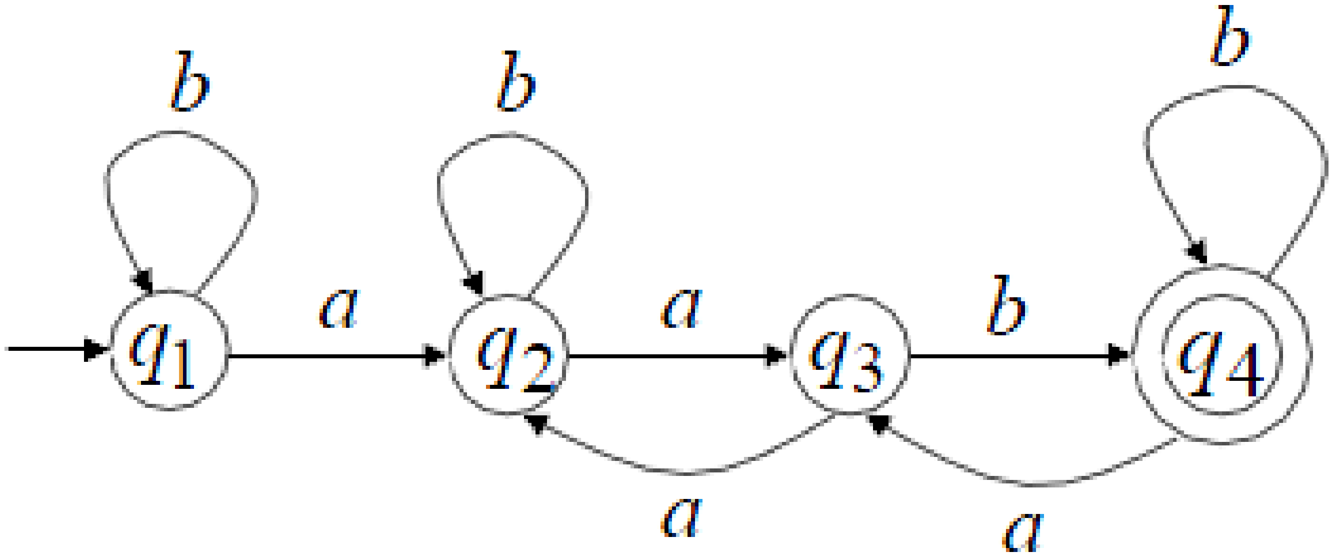
a state is repeated



If string w has length $|w| \geq 4$:

Then the transitions of string w are more than the states of the DFA

Thus, a state must be repeated

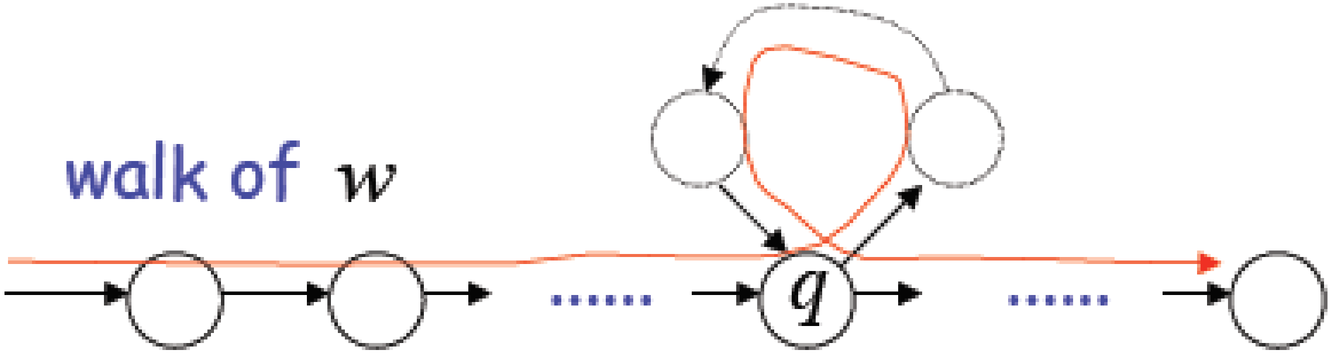


In general, for any DFA:

String w has length $\geq$ number of states

⇩

A state q must be repeated in the walk of w

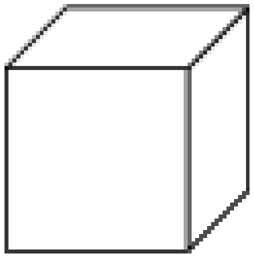


Repeated state

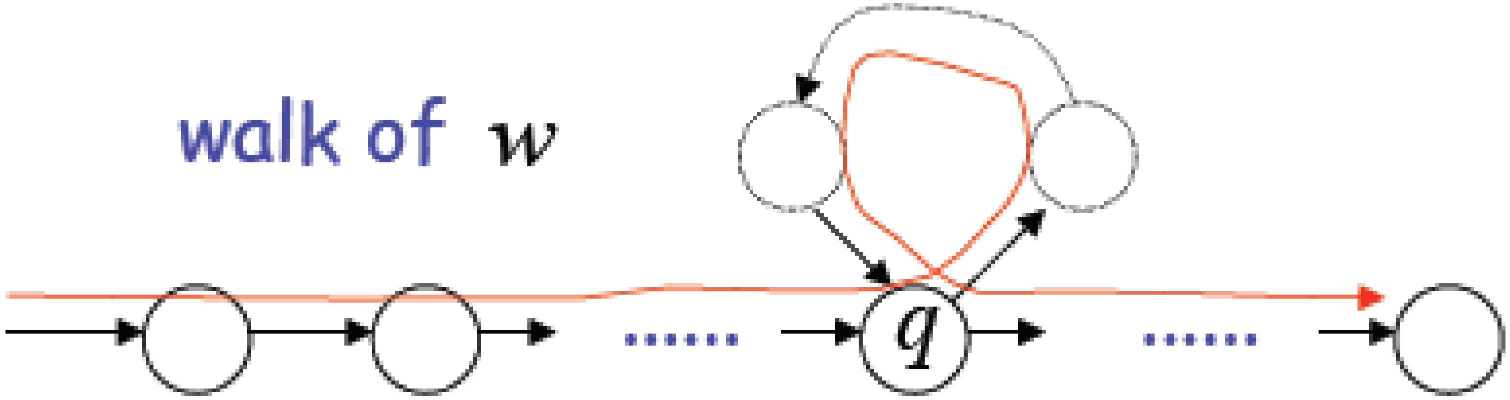
In other words for a string w:

$\xrightarrow{a}$ transitions are pigeons

$\textcircled{q}$ states are pigeonholes



walk of w



Repeated state

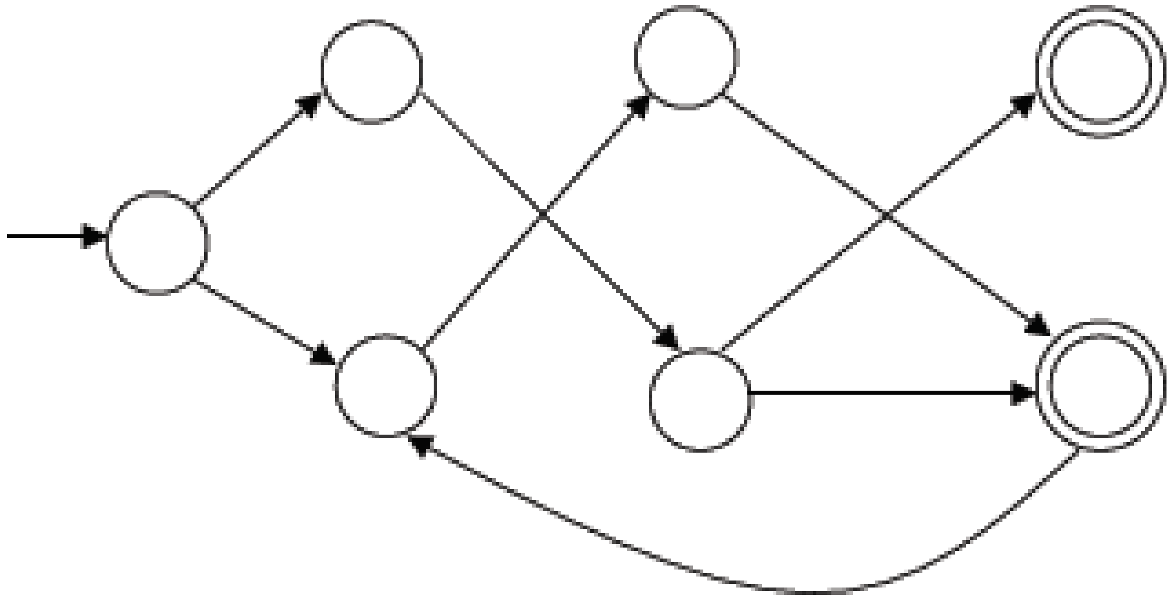
The Pumping Lemma (Pompalama Önsavı)

Bir dilin düzgün olmadığını ispatlamak için kullanılan bir kuraldır. Pigeonhole prensibinin başka bir formunu kullanır. Bu kural şu gözleme dayanmaktadır: n durumlu bir geçiş grafiğinde, n ya da daha fazla uzunluktaki bir yol, herhangi bir durumu tekrar etmelidir. Yani bir döngü(cycle) içermelidir.

Bir L dilinin düzgün olduğunu nasıl ispatlarız?
-Bu dil için düzgün ifade yazın
-Bir FA çizin
-Düzgün grameri oluşturun

Take an infinite regular language L

There exists a DFA that accepts L

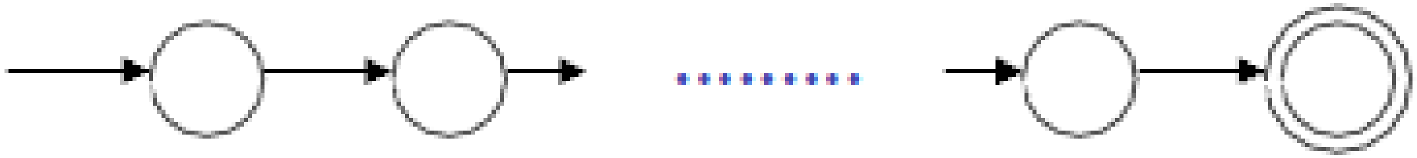


m

states

Take string w with $w \in L$

There is a walk with label w :

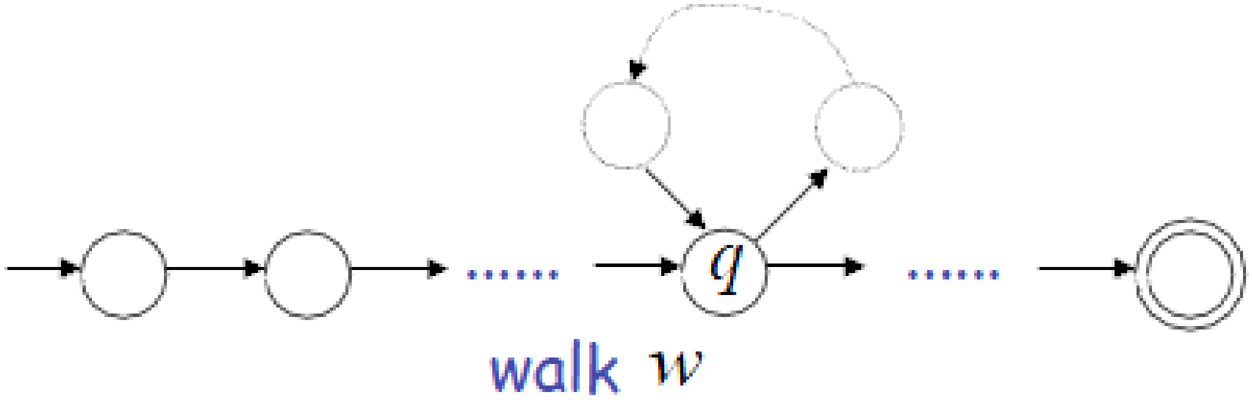


walk w

If string w has length $|w| \geq m$ (number of states of DFA)

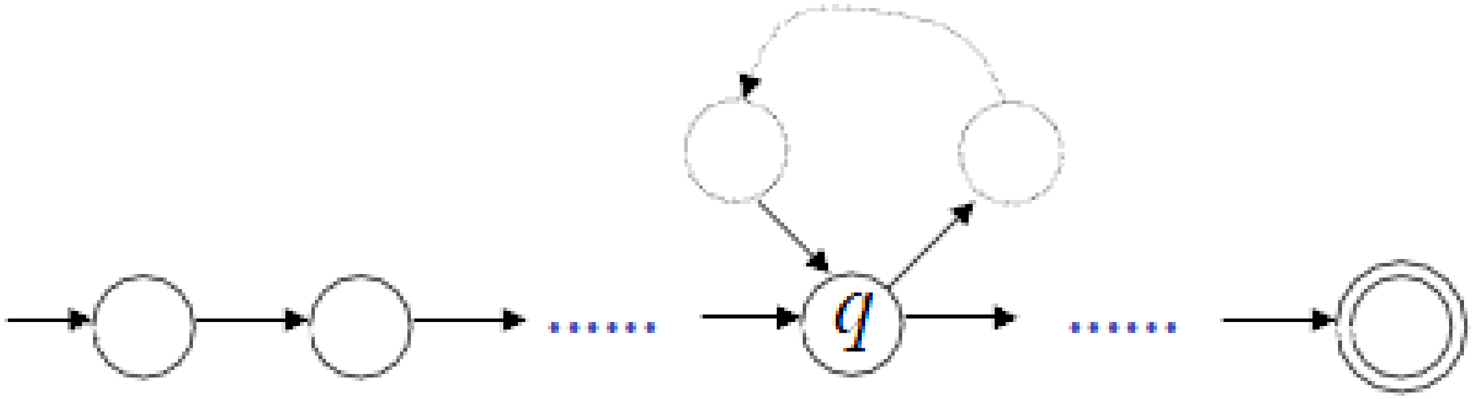
then, from the pigeonhole principle:

a state is repeated in the walk w



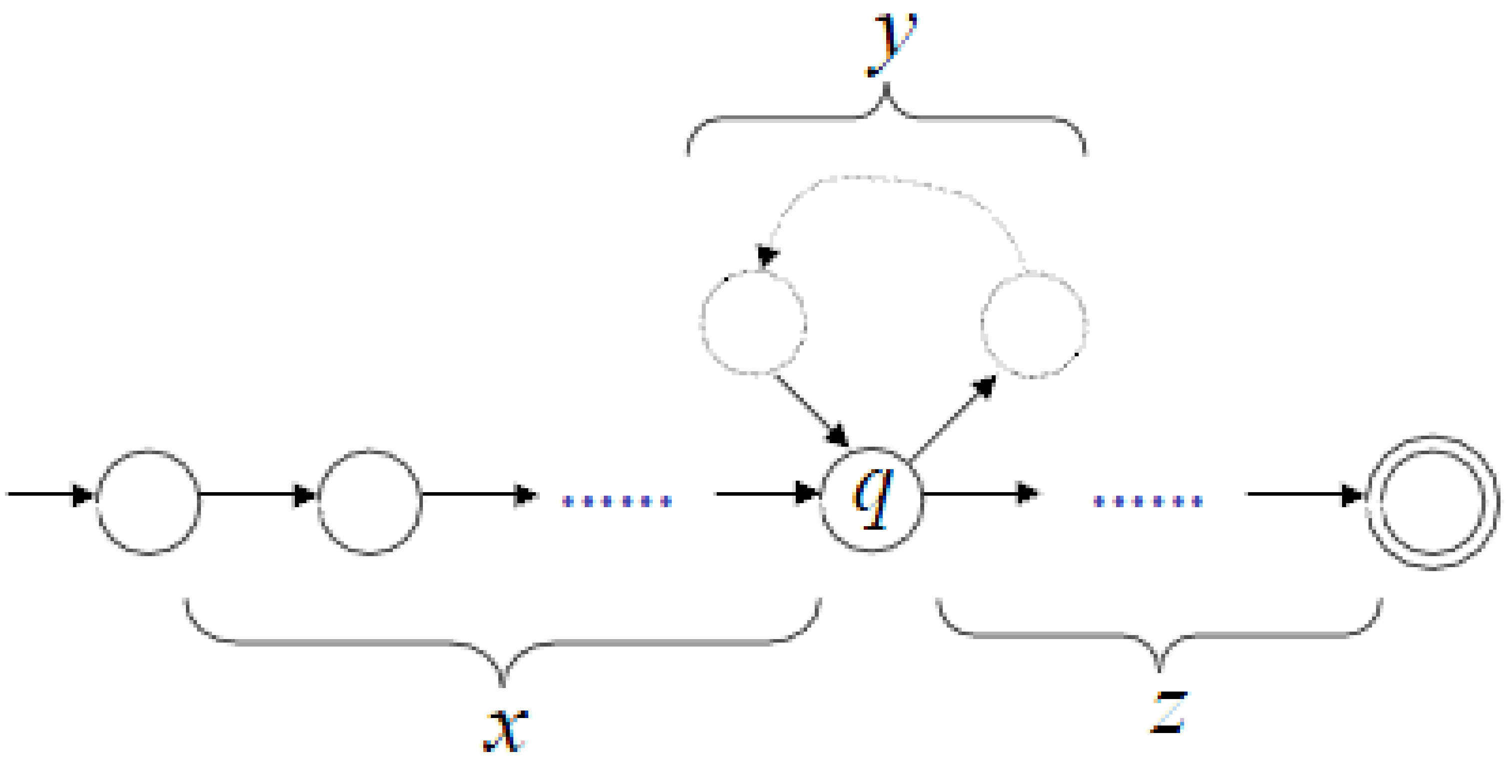
Pompalama bir dil gramerinin bir kısmının şişirilmesi veya artırılması olarak algılanabilir. Döngü kısmı pompalanarak dil içerisinde başka dizgiler türetilebilir. Bir dil ne kadar pompalanırsa pompalansın yine aynı dil grubuna ait olmalıdır.

Let q be the first state repeated in the walk of w



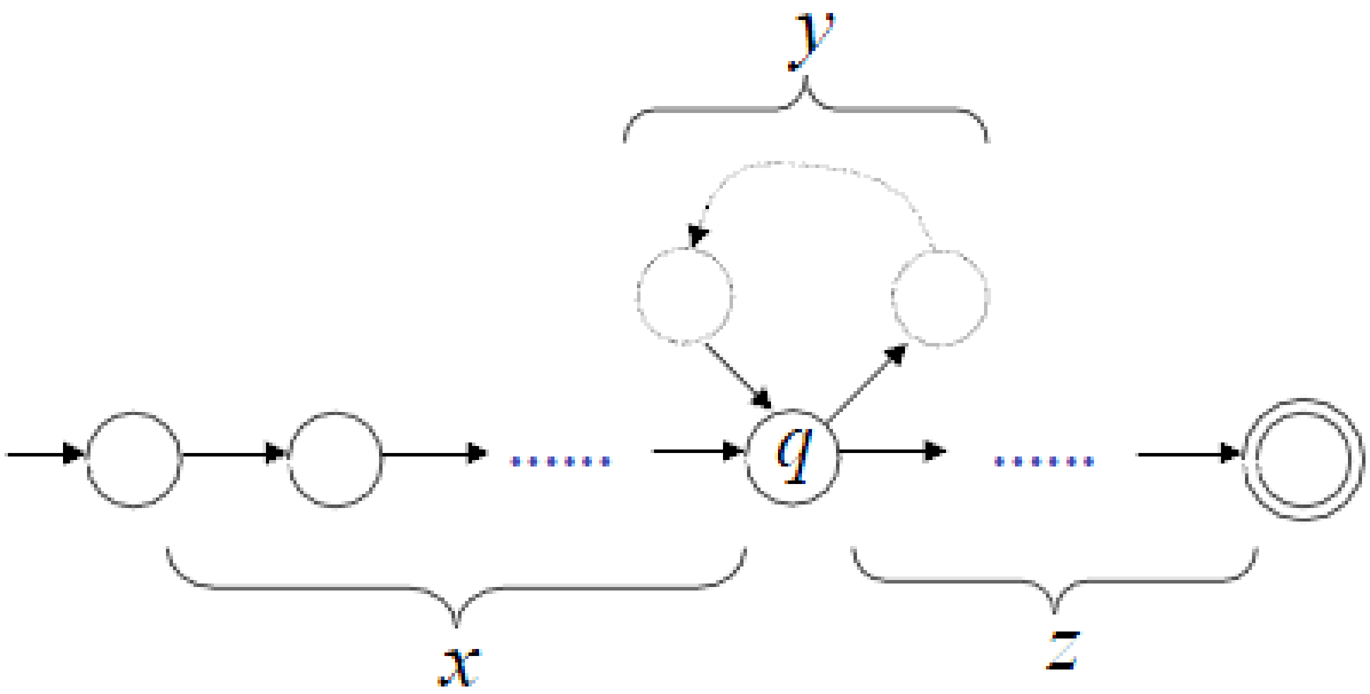
walk w

Write $w = x\ y\ z$

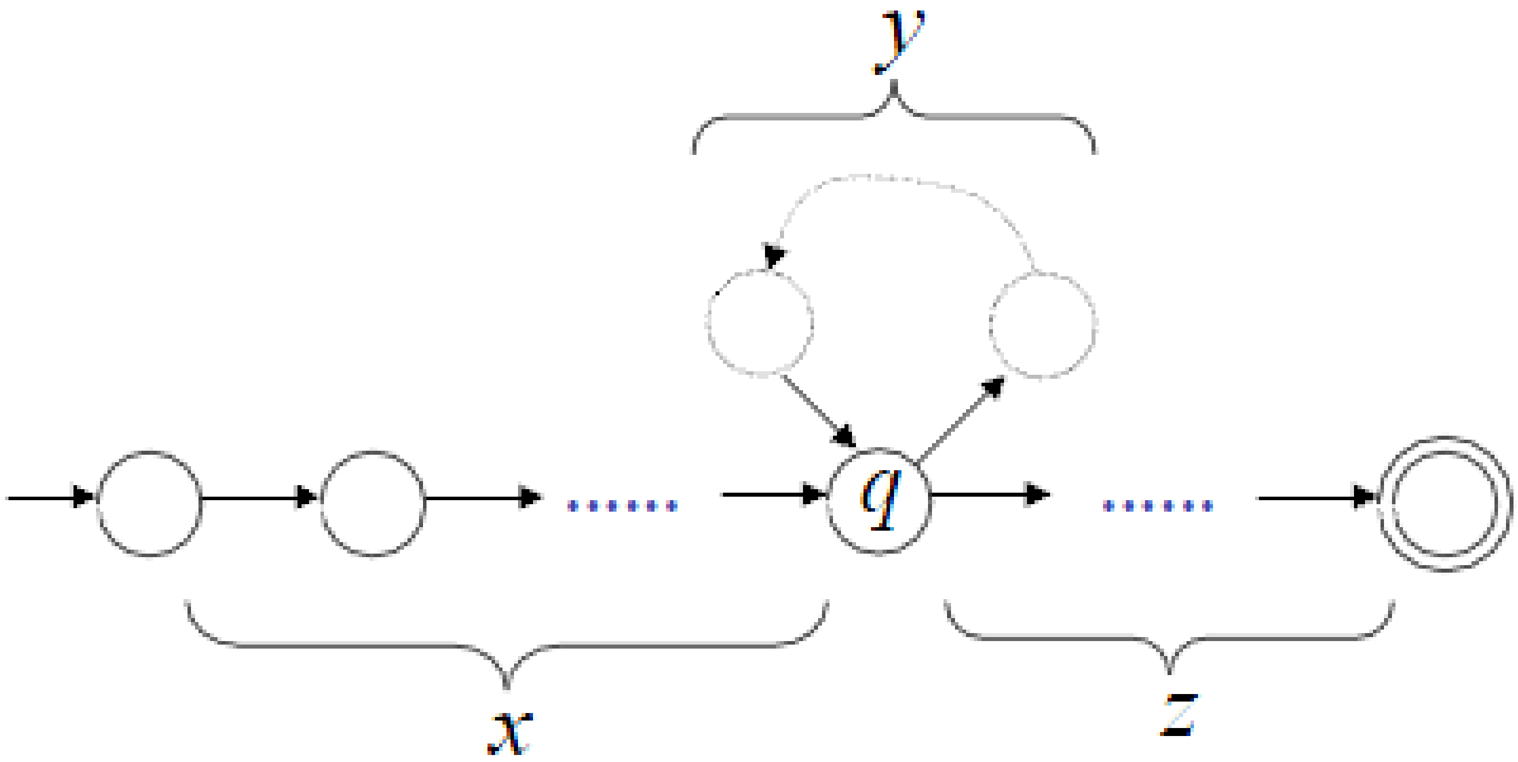


Observations: length $|x\,y| \leq m$ number of states of DFA

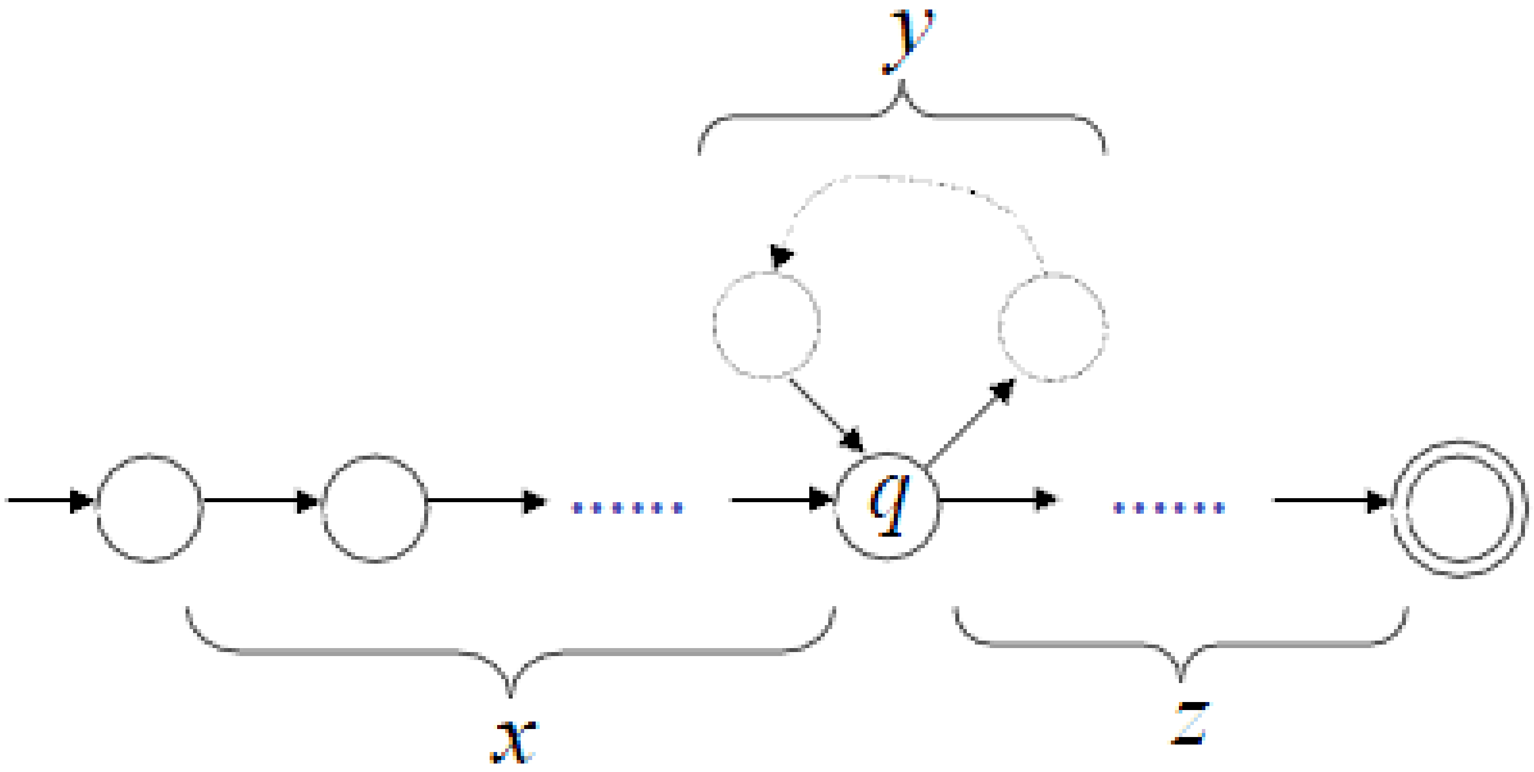
length $|y| \geq 1$



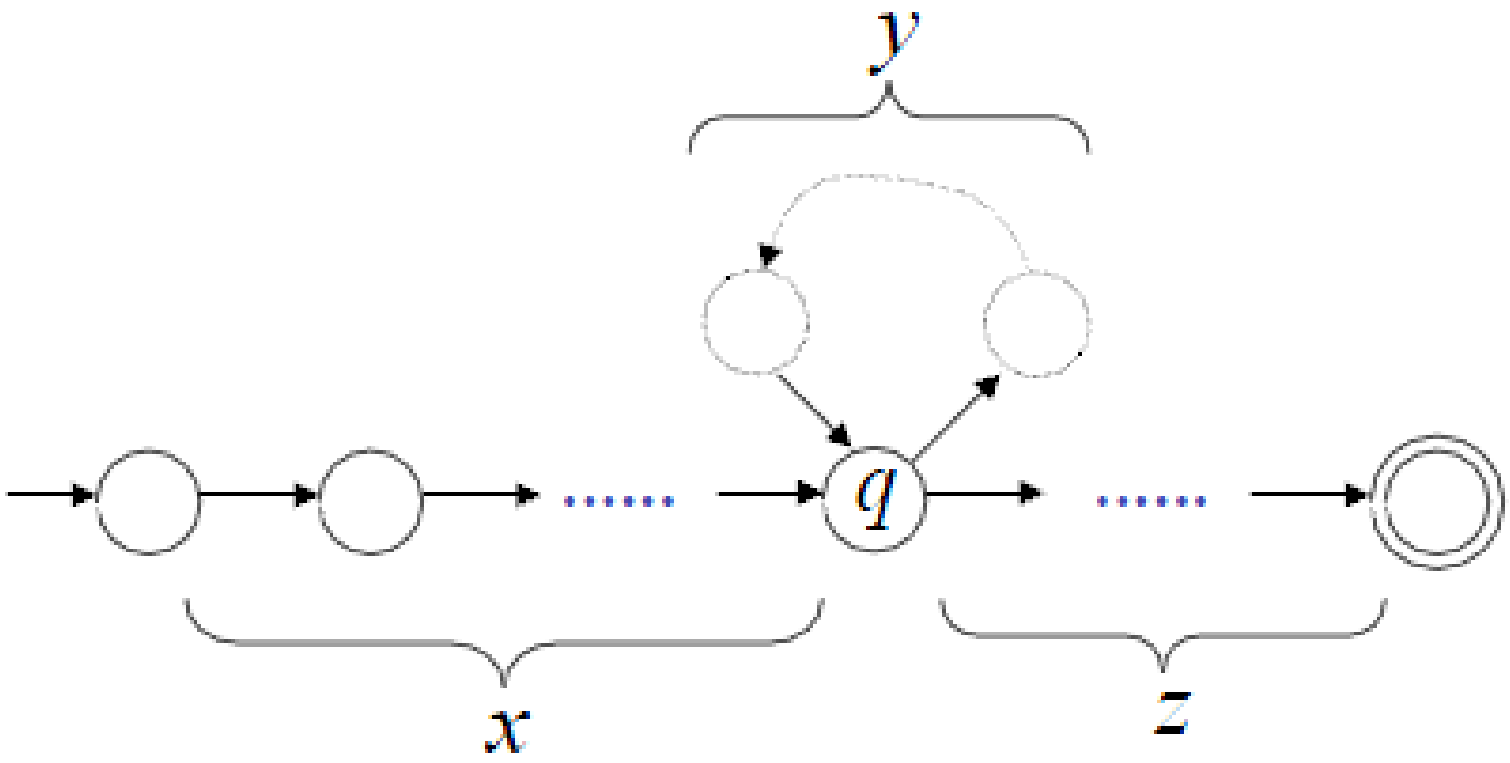
Observation: The string $x\ z$ is accepted



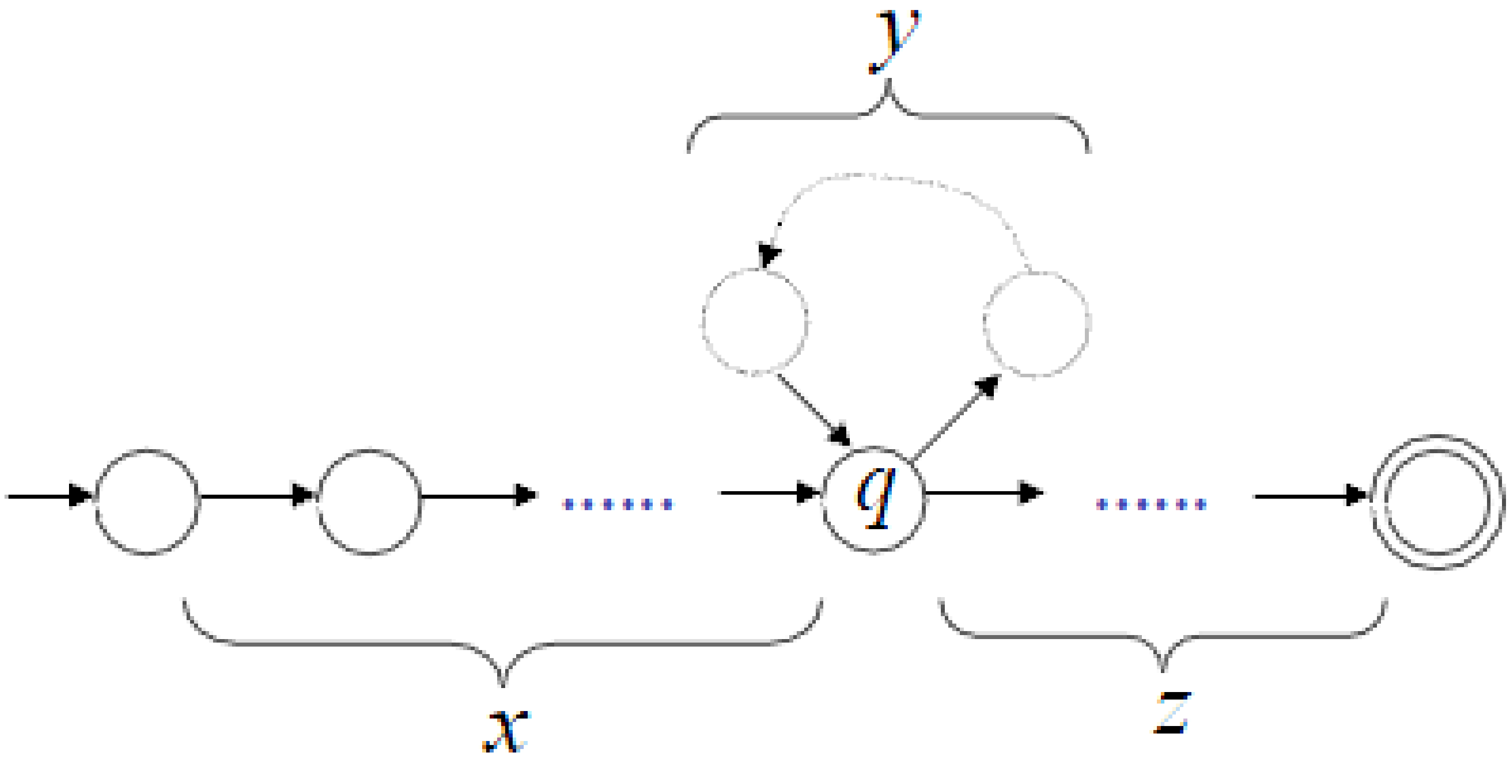
Observation: The string $x\ y\ y\ z$ is accepted



Observation: The string $x\ y\ y\ y\ z$ is accepted

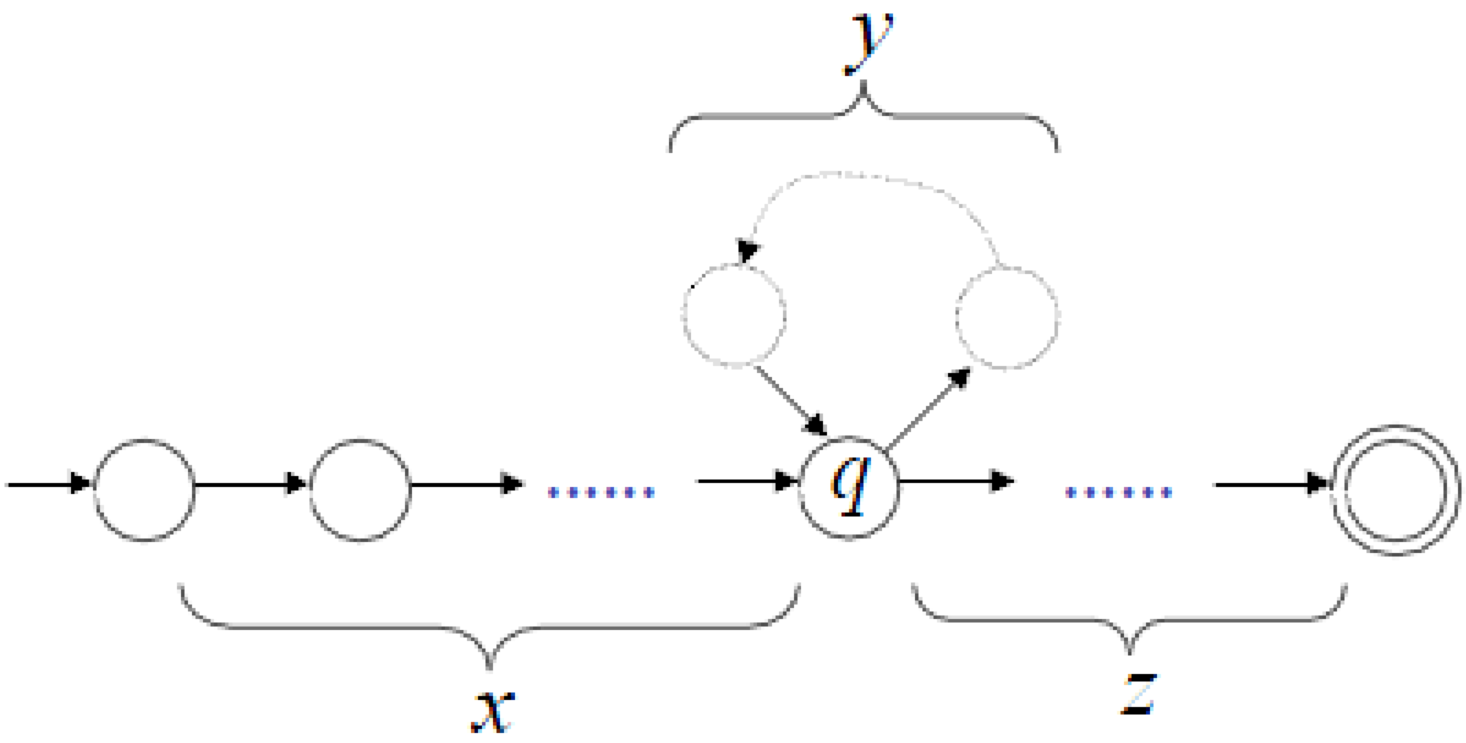


In General: The string $x\ y^i\ z$ is accepted $i = 0, 1, 2, \ldots$



In General: $xy^{i}z \in L$ $i = 0, 1, 2, \ldots$

Language accepted by the DFA



Teorem: L, sonlu olmayan düzgün bir dil olsun. Herhangi bir w ϵ L dizgisi için $|w| \geq m$ eşitsizliğini sağlayan bir m pozitif tamsayısı vardır. Aşağıdaki koşulları sağlayan herhangi x, y ve z için w = xyz yazılabilir:

$|xy| \leq m$

$|y| \geq 1$ ve her $i \geq 0$ için $w_i = xy^iz \;\epsilon\; L$

Başka bir deyişle L'deki yeterince uzun her dizgi 3 kısma bölünebilir. Ortadaki kısmın herhangi sayıdaki tekrarları L'deki başka dizgilere karşılık gelmektedir.

Theorem: The language $L = \{a^n b^n : n \geq 0\}$

is not regular

Proof: Use the Pumping Lemma

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

Assume for contradiction **(çelişme)**
that L is a regular language

Since L is infinite
we can apply the Pumping Lemma

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

Let m be the integer in the Pumping Lemma

Pick a string w such that: $w \in L$

length $|w| \geq m$

We pick $w = a^m b^m$

Write: $a^m b^m = x\ y\ z$

From the Pumping Lemma

it must be that length $|x\ y| \leq m, \quad |y| \geq 1$

$$xyz = a^m b^m = \overbrace{\underbrace{a\ldots a}_{x}\underbrace{a\ldots a}_{y}\mathrlap{\underbrace{}_{z}}a\ldots a}^{m}\overbrace{b\ldots b}^{m}$$

Thus: $y = a^k, \quad k \geq 1$

$$x\ y\ z = a^m b^m \qquad y = a^k, \quad k \ge 1$$

From the Pumping Lemma: $x\ y^i\ z \in L$

$$i = 0, 1, 2, \ldots$$

Thus: $x\ y^2\ z \in L$

$$x\,y\,z = a^m b^m \qquad y = a^k, \quad k \geq 1$$

From the Pumping Lemma: $x\,y^2\,z \in L$

$$xy^2z = \overbrace{\underbrace{a...a}_{x}\underbrace{a...a}_{y}\underbrace{a...a}_{y}\rlap{\underbrace{}_{z}}a...a}^{m+k}\overbrace{b...b}^{m} \in L$$

Thus: $a^{m+k} b^m \in L$

$$a^{m+k}b^{m} \in L$$

BUT: $L = \{a^{n}b^{n} : n \geq 0\}$

$$\Downarrow$$

$$a^{m+k}b^{m} \notin L$$

CONTRADICTION!!!

Therefore: Our assumption that L is a regular language is not true

Conclusion: L is not a regular language

$L = \{a^n b^n : n \geq 0\}$ dili düzgün bir dil midir?

Hayır değildir! Bunu çelişme ile ispat yöntemini ve Pigeonhole prensibini kullanarak gösterelim.

L dilinin düzgün bir dil olduğunu kabul edelim. O halde, bu dili kabul eden bir $M=(Q, \{a,b\}, \delta, q_0, F)$ DFA'sı olmalıdır.

$\delta^*(q_0, a^i)$ $i = 1,2,3,..$ Bu şekilde sınırsız sayıda i vardır. Ancak M otomatında sadece sınırlı sayıda durum bulunmaktadır. O halde pigeonhole prensibine göre, $n \neq m$ olmak üzere;

$\delta^*(q_0, a^n) = q$ ve

$\delta^*(q_0, a^m) = q$ 'yu sağlayan herhangi bir q durumu olmalıdır.

M otomatı $a^n b^n$ dizgisini kabul ettiği için;

$\delta^*(q, b^n) = q_f \in F$

Sonuç olarak;

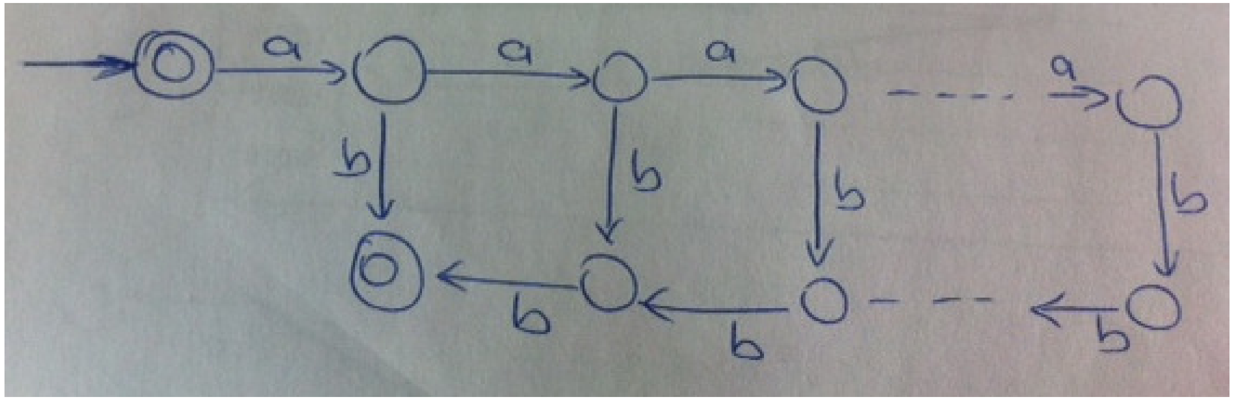
$\delta^*(q_0, a^m b^n) = \delta^*(\delta^*(q_0, a^m), b^n) = \delta^*(q, b^n) = q_f$

Bu durum esas varsayım ile çelişmektedir.

Esas varsayım :

M sadece n = m olması durumunda $a^m b^n$ dizgisini kabul eder.

Sonuç olarak L dili düzgün bir dil olamaz.



$M=(Q, \{a,b\}, \delta, q_0, F)$ $|a| = n$

$|x| \geq n$ olsun. x dizgisi n+1 durumdan geçmelidir.

$x=a_1a_2a_3\ldots a_ny$ olsun.

$q_0 = \delta^*(q_0, \lambda)$

$q_1 = \delta^*(q_0, a_1)$

$q_2 = \delta^*(q_0, a_1a_2)$

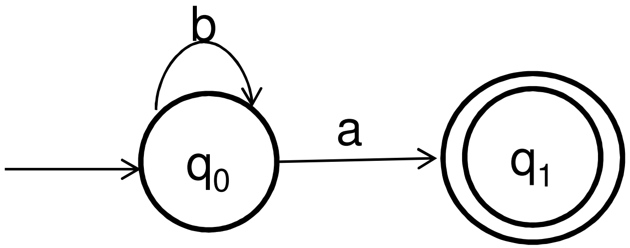
.

.

$q_n = \delta^*(q_0, a_1a_2\ldots a_n)$

Pigeonhole prensibine göre, herhangi bir durumu en azından iki kere içermelidir.

Örneğin;



x = a

|x| = 1

Durumlar = q_0 q_1

n= Geçilen farklı durumların sayısı = 2

x = bba |x|= 3

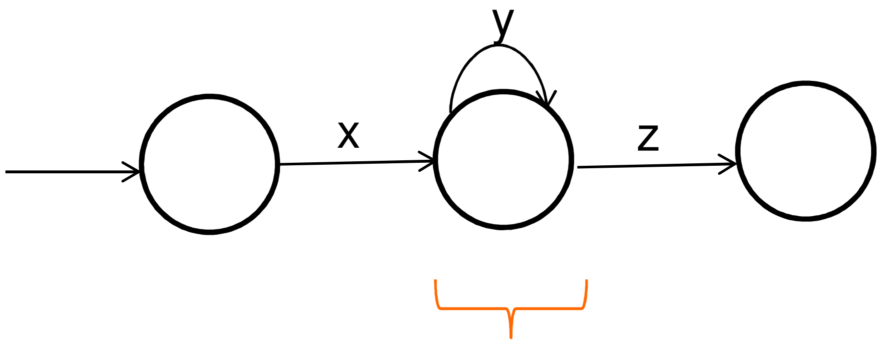
Durumlar = q_0 q_0 q_1

n = 2

|x|≥ n olan herhangi bir dizgide, bir durum mutlaka tekrarlanmıştır. Eğer bir durum bir ya da daha fazla kere tekrar edilmişse, geçiş diyagramında bir döngü olduğu anlamına gelmektedir. Eğer bir döngü varsa, bu dile ait çeşitli dizgiler üretilebilir.

a, bba, bbba, bbbba,….

Eğer bir DFA'da n durum varsa, n uzunluğundaki herhangi bir yol n+1 durumdan geçmelidir ve bir döngü içermelidir.



Dizginin bu kısmı pompalanarak (beslenerek) dil içerisinde başka dizgiler türetilebilir. Pumping kuralı burdan ileri gelmektedir.

Çelişme ile ispat:

İlk başta, ele alınan dilin pumping kuralında verilen özelliğe sahip olduğu varsayılır ve daha sonra bunun bir çelişkiye yol açtığı gösterilir ve buradan da dilin düzgün bir dil olmadığı sonucuna varılır.

$L=\{a^n b^n \mid n \geq 0\}$ Düzgün değil

- L düzgün bir dildir. O halde bir DFA tarafından kabul edilmelidir.
- DFA'daki durumların sayısı m olsun ve $|w| \geq m$ olmak üzere $w \in L$ dizgisini ele alalım.
- Pumping Lemma kullanarak w'yu üç kısma ayırabiliriz. w=xyz. Burada $n \geq 0$ için $xy^n z$ de L'deki bir dizgidir.
- $w = a^m b^m$ olsun.
- $|xy| \leq m$ olduğu için, y hep a'lardan oluşmalıdır.
- Ancak bu durumda da xy^2z dizgisinde b'lerden daha fazla sayıda a vardır.
- Bu da çelişmedir.

EXAMPLE.

The language of palindromes

$$L_2 = \{w \in \{0,1\} \mid w^R = w\}$$

is not regular.

Proof: Let $k > 0$ be arbitrary. Let $z = 0^k 1 0^k$.
Then $z \in L_2$ and $|z| \geq k$.

Let $z = uvw$ be any decomposition of z satisfying $|v| \geq 1$ and $|uv| \leq k$. Then $u = 0^r$ and $v = 0^s$, where $0 \leq r$ and $1 \leq s \leq k$. Then $uv^2w = 0^{k+s}10^k \not\in L_2$.

As L_2 does not satisfy the conclusion of the Pumping Lemma, it cannot be regular.