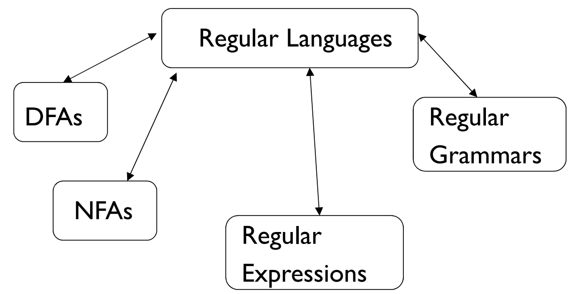
Department of Computer & Information Sciences
Pakistan Institute of Engineering and Applied Sciences

Properties of Regular Languages

Umar Faiz
http://www.pieas.edu.pk/umarfaiz/cis317

Standard Representations of Regular Languages



Properties of Regular Languages

- Decision Properties of Regular Languages
- Closure Properties of Regular Languages
- Equivalence and Minimization of Automata

Decision Properties of Regular Languages

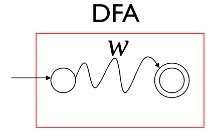
Decision Properties of RLs

Question: Given regular language L
and string w
how can we check if $w \in L$?

Answer: Take the DFA that accepts L
and check if w is accepted

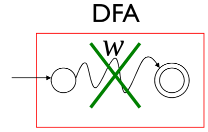
Decision Properties of RLs

DFA



$w \in L$

DFA



$w \notin L$

Decision Properties of RLs

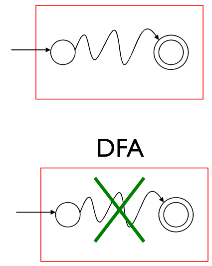
Question: Given regular language L how can we check if L is empty: $(L = \emptyset)$?

Answer: Take the DFA that accepts L. Check if there is any path from the initial state to a final state

Decision Properties of RLs

DFA

DFA



$L \neq \emptyset$

$L = \emptyset$

Decision Properties of RLs

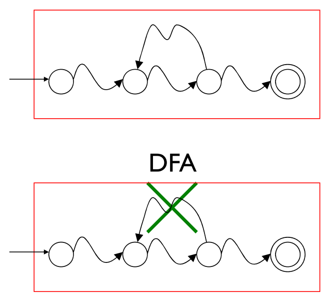
Question: Given regular language L how can we check if L is finite?

Answer: Take the DFA that accepts L. Check if there is a walk with cycle from the initial state to a final state

Decision Properties of RLs

DFA

DFA



L is infinite

L is finite

Decision Properties of RLs

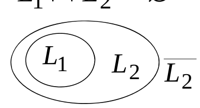
Question: Given regular languages L_1 and L_2 how can we check if $L_1 = L_2$?

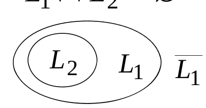
Answer: Find if $(L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2) = \emptyset$

Decision Properties of RLs

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

$L_1 \cap \overline{L_2} = \emptyset$ and $\overline{L_1} \cap L_2 = \emptyset$





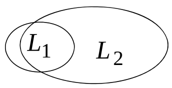
$L_1 \subseteq L_2$ $\qquad$ $L_2 \subseteq L_1$

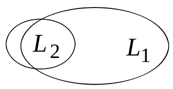
$$L_1 = L_2$$

Decision Properties of RLs

$$(L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2) \neq \emptyset$$

$L_1 \cap \overline{L_2} \neq \emptyset$ or $\overline{L_1} \cap L_2 \neq \emptyset$





$L_1 \not\subset L_2$ $L_2 \not\subset L_1$

$L_1 \neq L_2$

Decision Properties of RLs

Non-regular languages $\{a^n b^n : n \geq 0\}$

$\{vv^R : v \in \{a,b\}^*\}$

Regular languages

a^*b $b^*c + a$

$b + c(a+b)^*$

etc...

Decision Properties of RLs

How can we prove that a language L is not regular?

Prove that there is no DFA that accepts L

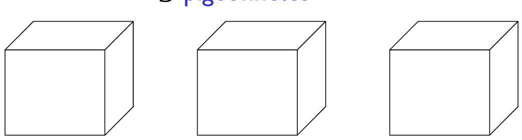
Problem: this is not easy to prove

Solution: the Pumping Lemma !!!

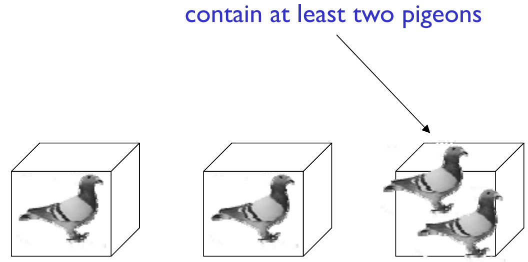
The Pigeonhole Principle

4 pigeons

3 pigeonholes

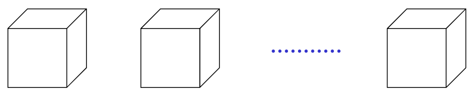


A pigeonhole must contain at least two pigeons



n pigeons

m pigeonholes $n > m$



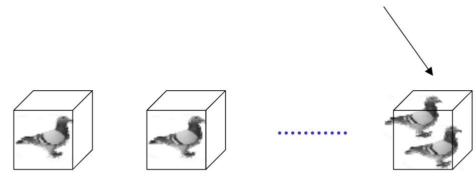
The Pigeonhole Principle

n pigeons

m pigeonholes

$n > m$

There is a pigeonhole with at least 2 pigeons



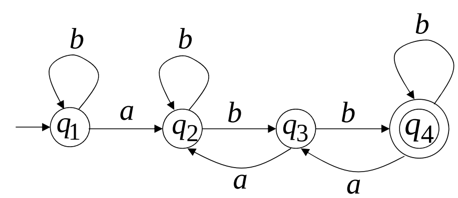
The Pumping Lemma Motivation

Consider the language

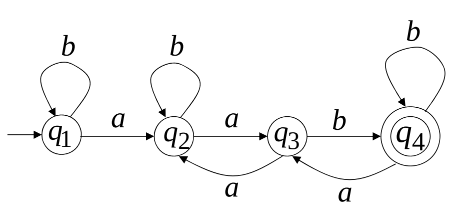
L1 = 01* = {0, 01, 011, 0111, … }

The string 011 is said to be pumpable in L1 because can take the underlined portion, and pump it up (i.e. repeat) as much as desired while always getting elements in L1.

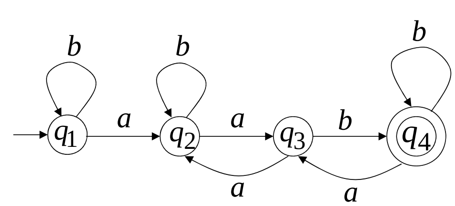
DFA with 4 states



In walks of strings: a, aa, aab — no state is repeated



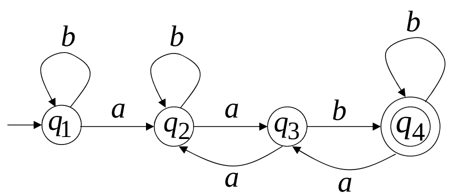
In walks of strings: $aabb$, $bbaa$, $abbabb$, $abbbabbabb...$ — a state is repeated



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

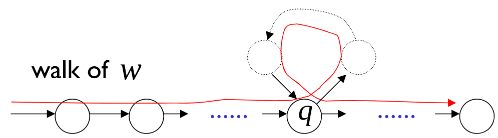


In general, for any DFA:

String w has length $\geq$ number of states

A state q must be repeated in the walk of w

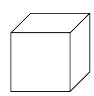
walk of w



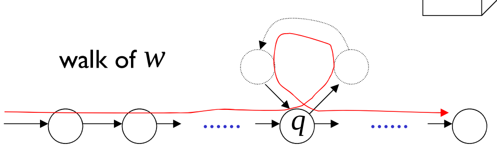
Repeated state

In other words for a string w :

$\xrightarrow{a}$ transitions are pigeons 

(q) states are pigeonholes 

walk of w



Repeated state

Closure Properties
of Regular Languages

Closure Properties of RL's

Closure means "being closed" in the same type of language domain, such as RL's.

"if certain languages are regular, and a language *L* is formed from them by certain operations, then *L* is also regular."

Closure Properties of RL's

Language operations for the above statement to be true include:

- Union
- Closure (star)
- Intersection
- Complement
- Difference
- Reversal
- Concatenation
- Homomorphism
- Inverse homomorphism

Operations on Languages

- The usual set operations

$$\{a,ab,aaaa\} \cup \{bb,ab\} = \{a,ab,bb,aaaa\}$$
$$\{a,ab,aaaa\} \cap \{bb,ab\} = \{ab\}$$
$$\{a,ab,aaaa\} - \{bb,ab\} = \{a,aaaa\}$$

- Complement:

$$\overline{L} = \Sigma^* - L$$

$$\overline{\{a,ba\}} = \{\lambda, b, aa, ab, bb, aaa, \ldots\}$$

31

Closure Properties of RL's

Closure under Union:

Let L and M two languages, then their union is defined by:

L ∪ M = { w : w ∈ L or w ∈ M}

Let A_L and A_M two finite automata accepting L and M respectively:

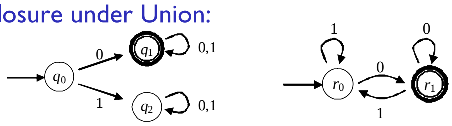
Example:

- $L_1 = \{0x \mid x \in \{0,1\}^*\}$ = strings that start with 0
- $L_2 = \{x0 \mid x \in \{0,1\}^*\}$ = strings that end with 0
- $L_1 \cup L_2 = \{x \in \{0,1\}^* \mid x$ starts with 0 or ends with 0 (or both)$\}$

32

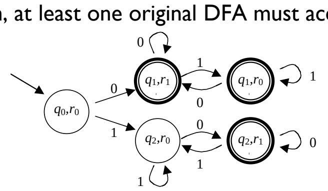
Closure Properties of RL's

Closure under Union:



$\{0x \mid x \in \{0,1\}^*\}$ $\{x0 \mid x \in \{0,1\}^*\}$

For union, at least one original DFA must accept:



33

Closure Properties of RL's

Closure under Union:

If L_1 and L_2 are any regular languages, $L_1 \cup L_2$ is also a regular language.

Proof 1: Using DeMorgan's laws

- Because the regular languages are closed for intersection and complement, we know they must also be closed for union

$$L_1 \cup L_2 = \overline{\overline{L_1} \cap \overline{L_2}}$$

34

Closure Properties of RL's

Closure under Union:

If L_1 and L_2 are any regular languages, $L_1 \cup L_2$ is also a regular language.

Proof 2: Product construction

- Same as for intersection, but with different accepting states
- Accept where either (or both) of the original DFAs accept
- Accepting state set is $(F_1 \times R) \cup (Q \times F_2)$

35

Closure Properties of RL's

Closure under Intersection:

Let L and M two languages, then their intersection is defined by:

L1 ∩ L2 = {x | x ∈ L1 and x ∈ L2}

Let A_L and A_M two finite automata accepting L and M respectively:

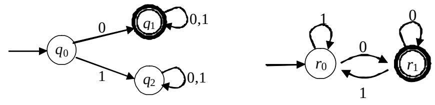
Example:

- $L_1 = \{0x \mid x \in \{0,1\}^*\}$ = strings that start with 0
- $L_2 = \{x0 \mid x \in \{0,1\}^*\}$ = strings that end with 0
- $L_1 \cap L_2 = \{x \in \{0,1\}^* \mid x$ starts and ends with 0$\}$

36

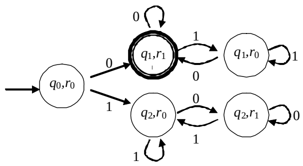
Closure Properties of RL's

Closure under Intersection:



$\{0x \mid x \in \{0,1\}^*\}$ $\{x0 \mid x \in \{0,1\}^*\}$

For intersection, both original DFAs must accept:



Closure Properties of RL's

Closure under Intersection:

If L_1 and L_2 are any regular languages, $L_1 \cap L_2$ is also a regular language.

Let L_1 and L_2 be any regular languages. By definition:

$M_1 = (Q, \Sigma, \delta_1, q_0, F_1)$ with $L(M_1) = L_1$
$M_2 = (R, \Sigma, \delta_2, r_0, F_2)$ with $L(M_2) = L_2$

Define a new DFA $M_3 = (Q \times R, \Sigma, \delta, (q_0,r_0), F_1 \times F_2)$
For δ, define it so that for all $q \in Q$, $r \in R$, and $a \in \Sigma$, we have $\delta((q,r),a) = (\delta_1(q,a), \delta_2(r,a))$

M_3 accepts if and only if both M_1 and M_2 accept
So $L(M_3) = L_1 \cap L_2$, so that intersection is regular

Closure Properties of RL's

Closure under Complement:

For any language L over an alphabet Σ, the *complement* of L is

$$\overline{L} = \{x \in \Sigma^* \mid x \notin L\}$$

Example: $L = \{0x \mid x \in \{0,1\}^*\}$ Strings that start with zero

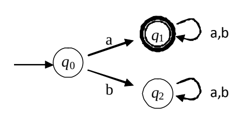
$\overline{L} = \{1x \mid x \in \{0,1\}^*\} \cup \{\varepsilon\}$ Strings that do not start with zero

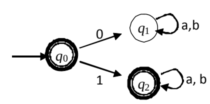
Closure Properties of RL's

Closure under Complement:

Given a DFA for any language, it is easy to construct a DFA for its complement

- Make the accepting states be non-accepting, and make the non-accepting states be accepting





$L = \{ax \mid x \in \{a,b\}^*\}$ $\overline{L} = \{bx \mid x \in \{0,1\}^*\} \cup \{\varepsilon\}$

Closure Properties of RL's

Closure under Reversal:

The **reversal** L^R of a language L is the language consisting of the reversals of all its strings.

The **reversal** of a string $w = a_1a_2, \ldots a_n$ is $w^R = a_na_{n-1}\ldots a_2a_1$.

Closure Properties of RL's

Closure under Reversal:

Definition: $L^R = \{w^R : w \in L\}$
Examples:

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

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

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

Closure Properties of RL's

Closure under Concatenation:

Definition:

Example: $L_1 L_2 = \{xy : x \in L_1, y \in L_2\}$

$\{a, ab, ba\}\{b, aa\}$

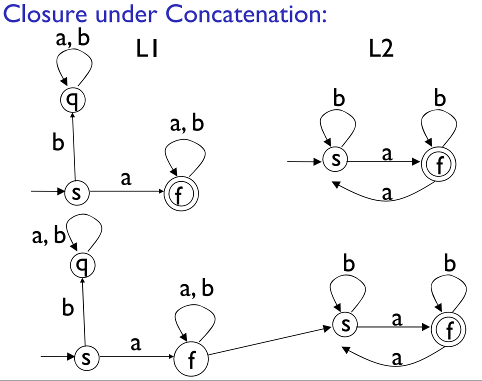
$= \{ab, aaa, abb, abaa, bab, baaa\}$

Steps to obtain a finite automation accepting L1L2:

1. Make an e-transition from all accept states in L1to the initial state in L2
2. Unmark all accepts states in L1
3. Remove the mark of the initial state in L2

Closure Properties of RL's

Closure under Concatenation:



Closure Properties of RL's

Closure under Concatenation:

$$L^n = \underbrace{LL \cdots L}_{n}$$

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

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

$$L^0 = \{\lambda\}$$

$$\{a, bba, aaa\}^0 = \{\lambda\}$$

Closure Properties of RL's

Closure under Concatenation:

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

$$L^2 = \{a^n b^n a^m b^m : n, m \geq 0\}$$

$$aabbaaabbb \in L^2$$

Closure Properties of RL's

Closure under Star-Closure (Kleene*):

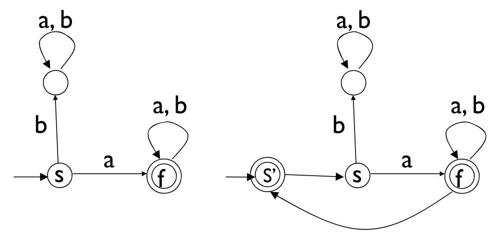
Definition: $L^* = L^0 \cup L^1 \cup L^2 \cdots$

Example:

$$\{a, bb\}^* = \left\{\begin{array}{l} \lambda, \\ a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \ldots \end{array}\right\}$$

Closure Properties of RL's

Closure under Star-Closure (Kleene*):



Closure Properties of RL's

Closure under Positive Closure:

Definition: $L^{+} = L^{1} \cup L^{2} \cup \cdots$

$= L^{*} - \{\lambda\}$

Example:

$$\{a, bb\}^{+} = \left\{ \begin{array}{l} a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \ldots \end{array} \right\}$$

49

Closure Properties of RL's

Closure under Difference:

Theorem : If L1 and L2 are regular languages, then so is L1– L2.

Example:

$L1=\{a, a^3, a^5, a^7,\text{-----}\}$

$L2=\{a^2, a^4, a^6,\text{-----}\}$

$L1\text{-}L2 = \{a,a^3, a^5, a^7\text{----}\}$

$RE=a(a)^*$

50

Closure Properties of RL's

Closure under Difference:

Theorem: If L & M are regular languages then L – M is also regular.

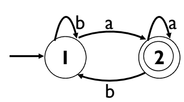
- L – M = L ∩ M
- L = { set of all strings ending in a}
- M= { set of all strings ending in a}
- L – M = ϕ

51

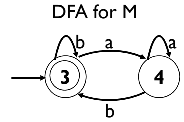
Closure Properties of RL's

Closure under Difference:

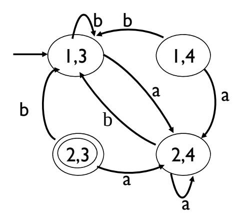
DFA for L



DFA for M



DFA for L-M



52

Closure Properties of RL's

Closure under Homomorphism:

A homomorphism is a function *h* which substitutes a particular string for each symbol.

That is, $h(a) = x$, where a is a symbol and x is a string.

Given $w = a_1a_2\ldots a_n$, define

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

Given a language, define

$h(L) = \{h(w) \mid w \in L\}$.

53

Closure Properties of RL's

Closure under Homomorphism:

Example:

Let function *h* be defined as

$h(0)= ab$ and $h(1) = \varepsilon$,

then *h* is a string homomorphism.

For examples,

1. $h(0011) = h(0)h(0)h(1)h(1)$

   $= abab\varepsilon\varepsilon = abab$.

2. If RE $r = 10^*1$, then $h(L(r)) = (ab)^*$.

54

Closure Properties of RL's

Closure under Homomorphism:

Theorem - If L is an RL, then $h(L)$ is also an RL where h is a homomorphism.

55

Closure Properties of RL's

Closure under Inverse Homomorphism:

Let h be a homomorphism from some alphabet Σ to strings in another alphabet T. Let L be an RL over T. Then $h^{-1}(L)$ is the set of strings w such that $h(w)$ is in L. $h^{-1}(L)$ is read "h inverse of L."

56

Closure Properties of RL's

Closure under Inverse Homomorphism:

Example:
Let $L = L((00 + 1)^*)$
Let string homomorphism h be defined as

$$h(a) = 01,\ h(b) = 10.$$

It can be proved that

$$h^{-1}(L) = L((ba)^*)$$

57

Closure Properties of RL's

Closure under Inverse Homomorphism:

Theorem - If h is a homomorphism from alphabet Σ to alphabet T, and L is an RL, then $h^{-1}(L)$ is also an RL.

58

Theorems for Closure Properties of RL's

Let L and M be two RL's over alphabet

- Theorem – The union $L \cup M$ is an RL.
- Theorem – The complement $= \Sigma^* - L$ is an RL (Σ^* is the universal language)
- Theorem – The intersection $L \cap M$ is an RL.
- Theorem – The difference $L - M$ is an RL.
- Theorem – The concatenation LM and the closure L^* are RL's (
- Theorem - Reversal L^R of an RL L is also an RL.

59