Automata Formal Languages & Logic

Unit 1 - Finite State Machines

Preet Kanwal
Department of Computer Science & Engineering

Automata Formal Languages & Logic

Mathematical Preliminaries

1. Sets

2. Functions & Relations

Automata Formal Languages & Logic

Why are Sets important?

Sets

A *set* is an unordered collection of distinct objects, which may be anything (including other sets).

Example :

Set Notation :

Curly braces with commas separating out the elements

Sets

A *set* is an unordered collection of distinct objects, which may be anything (including other sets).

These are same sets

Set Representation

A set can be represented in any one of the following three ways or forms.

(i) Descriptive form
(ii) Set-builder form or Rule form
(iii) Roster form or Tabular form

Set Representation

(i) Descriptive form - verbal/english description of its elements.

Example :

"The set of all even numbers."
"The set of all real numbers less than 150."
"The set of all negative integers."

Set Representation

(ii) Set-builder form or Rule form

Even Natural Numbers

$\{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \}$

Set Representation

(iii) Roster form or Tabular form

Listing the elements of a set inside a pair of braces { } is called the roster form.

For example,
Let A = {x : x is an integer and $-1 \leq x < 5$ }

In roster form we write A = { -1, 0,1, 2, 3, 4 }

Automata Formal Languages & Logic

Set Membership

$\in$

$\{$

,

,

,

$\}$

Is

in the set?

Automata Formal Languages & Logic

Set Membership

$\notin$

Is in the set?

Order of a Set / Cardinality

The *cardinality* of a set S is the number of elements it contains, denoted by |S|.

Examples:

1) if Set A = {*a*, *b*, *c*, *d*, *e*}
then |A| = 5

2) if Set B = {{*a*, *b*}, {*c*, *d*, *e*, *f*, *g*}, {*h*}}
then |B| = 3

3) if Set C = { $n \in$ N | n < 137}
then |C| = 137

Automata Formal Languages & Logic

Order of a Set / Cardinality

Question :

1) What is $|N|$?

2) Consider the set
$S = \{ n \mid n \in N \text{ and } n \text{ is even} \}$

What is $|S|$?

Automata Formal Languages & Logic

Order of a Set / Cardinality

Which infinity is larger?

$\mathbb{N}$	0	1	2	3	4	5	6	7	8	...
	↕	↕	↕	↕	↕	↕	↕	↕	↕	
S	0	2	4	6	8	10	12	14	16	...

$$n \leftrightarrow 2n$$

$$S = \{\, n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \,\}$$

$$|S| = |\mathbb{N}| = \aleph_0$$

Automata Formal Languages & Logic

Types of Sets

1. Empty Set or Null Set
2. Singleton Set
3. Finite Set
4. Infinite Set
5. Equivalent Set
6. Equal Sets
7. Disjoint Sets
8. Subsets
9. Proper Subset
10. Super Set
11. Universal Set
12. Power Set

Types of Sets

1. Empty Set or Null Set
2. Singleton Set
3. Finite Set
4. Infinite Set
5. Equivalent Set
6. Equal Sets
7. Disjoint Sets
8. Subsets
9. Proper Subset
10. Super Set
11. Universal Set
12. Power Set

A null set or an empty set is a valid set with no member.

$\{\ \}$ $\emptyset$ We use this symbol to denote empty set

Types of Sets - Empty or Null Set

Question : Are these equal ?

$$\emptyset \; ! = \; \{ \emptyset \} \; ?$$

This set contains nothing at all.

This set has one element, which happens to be the empty set.

They are not equal

Types of Sets - Empty or Null Set

Question : Are these equal ?

$$2 \; != \; \{2\} \quad ?$$

This is a number.

This is a set. It contains a number.

They are not equal

Types of Sets

1. Empty Set or Null Set
2. Singleton Set
3. Finite Set
4. Infinite Set
5. Equivalent Set
6. Equal Sets
7. Disjoint Sets
8. Subsets
9. Proper Subset
10. Super Set
11. Universal Set
12. Power Set

If a set contains only one element it is called to be a singleton set.

Example :

Set A = {1}
Set B ={a}
Set C ={4}
Set D ={car}

Types of Sets

1. Empty Set or Null Set
2. Singleton Set
3. Finite Set
4. Infinite Set
5. Equivalent Set
6. Equal Sets
7. Disjoint Sets
8. Subsets
9. Proper Subset
10. Super Set
11. Universal Set
12. Power Set

A set in which number of elements are finite (can be counted) are called Finite Sets.

Example

Set A = {4, 8, 12, 16}

Set B = {90, 180}

Types of Sets

1. Empty Set or Null Set
2. Singleton Set
3. Finite Set
4. Infinite Set
5. Equivalent Set
6. Equal Sets
7. Disjoint Sets
8. Subsets
9. Proper Subset
10. Super Set
11. Universal Set
12. Power Set

A set in which number of elements are infinite are called Infinite Sets.

Example

N = Set of all natural no's
= {1, 2, 3, … }

Z = Set of all integers
= {..,-3,-2,-1,0,1,2,3..}

Types of Sets

1. Empty Set or Null Set
2. Singleton Set
3. Finite Set
4. Infinite Set
5. Equivalent Set
6. Equal Sets
7. Disjoint Sets
8. Subsets
9. Proper Subset
10. Super Set
11. Universal Set
12. Power Set

If the number of elements are same for two different sets, then they are called an equivalent sets.

Example:
If A = {1,2,3,4} and
B = {Red, Blue, Green, Black}

Here, Set A and B are equivalent,
since |A| = |B|= 4

Types of Sets

1. Empty Set or Null Set
2. Singleton Set
3. Finite Set
4. Infinite Set
5. Equivalent Set
6. Equal Sets
7. Disjoint Sets
8. Subsets
9. Proper Subset
10. Super Set
11. Universal Set
12. Power Set

The two sets A and B are said to be equal if they have exactly the same elements, order of elements do not matter.

Example:
if A = {1,2,3,4} and
B = {4,3,2,1}

Then, Set A and B are equal
& can be denoted as :
A = B

Types of Sets

1. Empty Set or Null Set
2. Singleton Set
3. Finite Set
4. Infinite Set
5. Equivalent Set
6. Equal Sets
7. Disjoint Sets
8. Subsets
9. Proper Subset
10. Super Set
11. Universal Set
12. Power Set

The two sets A and B are said to be disjoint if the set does not contain any common element.

Example:

Set A = {1,2,3,4} and
Set B = {5,6,7,8} are disjoint sets, because there is no common element between them.

Types of Sets

1. Empty Set or Null Set
2. Singleton Set
3. Finite Set
4. Infinite Set
5. Equivalent Set
6. Equal Sets
7. Disjoint Sets
8. Subsets
9. Proper Subset
10. Super Set
11. Universal Set
12. Power Set

A set S is a subset of a set T (denoted $S \subseteq T$) if all elements of S are also elements of T.

Examples:

1. $\{ 1, 2, 3 \} \subseteq \{ 1, 2, 3, 4 \}$
2. $N \subseteq Z$ (every natural number is an integer)
3. $Z \subseteq R$ (every integer is a real number)

Automata Formal Languages & Logic

Types of Sets - Subsets

Question :

Are there any sets S where $\emptyset \subseteq S$?

Types of Sets

1. Empty Set or Null Set
2. Singleton Set
3. Finite Set
4. Infinite Set
5. Equivalent Set
6. Equal Sets
7. Disjoint Sets
8. Subsets
9. Proper Subset
10. Super Set
11. Universal Set
12. Power Set

if B is a proper subset of A (denoted as $B \subset A$), then all elements of B are in A but A contains atleast one element that is not in B.

Example :

$A = \{1, 3, 5\}$

$B = \{ 1, 5\}$

$C = \{ 1, 3, 5\}$

then,

$B \subset A$

But, $C \subseteq A$

Types of Sets

1. Empty Set or Null Set
2. Singleton Set
3. Finite Set
4. Infinite Set
5. Equivalent Set
6. Equal Sets
7. Disjoint Sets
8. Subsets
9. Proper Subset
10. Super Set
11. Universal Set
12. Power Set

if $B \subseteq A$ (B is a subset of A) then,
$A \supseteq B$ (A is a superset of B)

Here,
if $B \subset A$ (B is a proper subset of A) then,
$A \supset B$ (A is a proper superset of B)

Consider Example :
$A = \{1, 3, 5\}$
$B = \{ 1, 5\}$
$C = \{ 1, 3, 5\}$

Here, $A \supseteq C$ or $C \supseteq A$
and $A \supset B$

Automata Formal Languages & Logic

Types of Sets - Subsets

Question :

Is every set a subset and superset of itself??

Types of Sets

1. Empty Set or Null Set
2. Singleton Set
3. Finite Set
4. Infinite Set
5. Equivalent Set
6. Equal Sets
7. Disjoint Sets
8. Subsets
9. Proper Subset
10. Super Set
11. Universal Set
12. Power Set

A universal set, denoted by capital U is all the elements, or members, of any group under consideration, including its own elements.

Types of Sets

1. Empty Set or Null Set
2. Singleton Set
3. Finite Set
4. Infinite Set
5. Equivalent Set
6. Equal Sets
7. Disjoint Sets
8. Subsets
9. Proper Subset
10. Super Set
11. Universal Set
12. Power Set

Power set of any set S is the set of all subsets of S, including the empty set and S itself

$S = \{a, b, c, d\}$

$P(S) = \{$
$\emptyset,$
$\{a\}, \{b\}, \{c\}, \{d\},$
$\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\},$
$\{b, d\}, \{c, d\},$
$\{a, b, c\}, \{a, b, d\},$
$\{a, c, d\}, \{b, c, d\},$
$\{a, b, c, d\}$
$\}$

$|S| < |P(S)|$

Automata Formal Languages & Logic

Combining Sets - Operations on Set

Union

$A \cup B$

{ 1, 2, 3, 4, 5 }

$A = \{ 1, 2, 3 \}$
$B = \{ 3, 4, 5 \}$

Intersection

$A \cap B$

{ 3 }

$A = \{ 1, 2, 3 \}$
$B = \{ 3, 4, 5 \}$

$A \cup B = \{ x : x \in A \text{ or } x \in B \}$

$A \cap B = \{ x : x \in A \text{ and } x \in B \}$

Automata Formal Languages & Logic

Combining Sets

Difference

$A - B$

$\{ 1, 2 \}$

$A = \{ 1, 2, 3 \}$
$B = \{ 3, 4, 5 \}$

Compliment

$A' = \overline{A} = A^C$

A - B = { x : x ∈ A and x ∉ B}

A’ = { x : x ∈ U and x ∉ A}

Automata Formal Languages & Logic

Set Identities

Identity	*Name*
$A \cap U = A$ $A \cup \emptyset = A$	Identity laws
$A \cup U = U$ $A \cap \emptyset = \emptyset$	Domination laws
$A \cup A = A$ $A \cap A = A$	Idempotent laws
$\overline{(\overline{A})} = A$	Complementation law
$A \cup B = B \cup A$ $A \cap B = B \cap A$	Commutative laws
$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$	Associative laws
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	Distributive laws
$\overline{A \cap B} = \overline{A} \cup \overline{B}$ $\overline{A \cup B} = \overline{A} \cap \overline{B}$	De Morgan's laws
$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$	Absorption laws
$A \cup \overline{A} = U$ $A \cap \overline{A} = \emptyset$	Complement laws

$\emptyset' = U$

Cartesian Product of Sets

The Cartesian product of two sets A and B, denoted A × B, is the set of all possible ordered pairs where ,

the elements of A are first and the elements of B are second.

$$A \times B = \{ (a, b) : a \in A \text{ and } b \in B\}$$

Cartesian Product of Sets

A = {1, 3, 5} and B ={2, 4} then,
A X B = { (a, b) : a ∈ A and b ∈ B}

A X B = { (1,2), (1,4), (3,2), (3, 4), (5,2), (5,4) }

Arrow Diagram of A ×B :

1 → 2
1 → 4
3 → 2
3 → 4
5 → 2
5 → 4

Partition of a Set

A partition of a set is a grouping of its elements into non-empty subsets, in such a way that every element is included in exactly one subset.

For example the regions A1, A2, A3, A4, A5 form partitions of Set A as shown below

A1, A2, A3, A4, A5 are disjoint subsets of A.

A = A1 U A2 U A3 U A4 U A5

Automata Formal Languages & Logic

Mathematical Preliminaries

1. Sets

2. Functions & Relations

Functions

Function Notation

$$y = f(x)$$

Output

Name of Function

Input

Example :

$f(x) = \{x + 4,\ 0 < x < 5\}$

$f(2) = 6 = y$

Functions

Domain of a Function :

Set of possible inputs to a function is called its domain.

Range of a Function :

Set of possible outputs from a function is called its range.

Notation :

f : D -> R

Example :

$f(x) = \{x + 4, 0< x <5\}$

Domain = {1,2,3,4}
Range = {5,6,7,8}

f : {1,2,3,4} -> {5,6,7,8}

Functions

- A function *f* may not necessarily use all the elements of the Range

f : Student -> Grades

Domain of f : Set of Students

Range of f : {A, B, C, F}

Automata Formal Languages & Logic

Types of Functions

1)One-to-One (Injective) function
2) Onto (Surjective) function
3) One-to-One Correspondence (Bijective) function

Types of Functions

1)One-to-One (Injective) function - A function f: A → B is One to One if for each element of A there is a distinct element of B.

Image(a) = 4
Pre-image(4) = a

Types of Functions

2)Onto (Surjective) function - all elements in the Range must be used(mapped)

Types of Functions

3) One-to-One Correspondence (Bijective) function - Both one-to-one and onto.

Relations

A relation between two sets is a collection of ordered pairs (x,y), containing one object from each set. Here x is from first set and y is from second set.

The set of the first components of each ordered pair is called the domain and the set of the second components of each ordered pair is called the range.

A function is a special relation in which each possible input maps to exactly one output value.

Example :

R = {(1,2), (2,4), (3,6)}

Domain is {1,2,3}
Range is {2,4,6}

Automata Formal Languages & Logic

Is this Relation a function?

Automata Formal Languages & Logic

Binary Relations

A binary relation is a relationship between two sets. Formally, binary relation over sets X and Y is a subset of the Cartesian product X × Y;

R	a	b
0	×	×
1	×	
2		×

Element a is related to b by R is denoted by aRb.

aRb denotes (a, b) ∈ R

Eg:

A = {0, 1, 2}

B = {a, b}

A x B = { (0, a), (1, a), (0, b), (1, b), (2, a), (2, b) }

R = { (0, a), (1, a), (0, b), (2, b) } ⊆ A x B

Representing a Relation

Representing a Relation :

1) Matrix form

2) Directed Graph (Digraph)

	a	b	c	d
a	0	1	0	1
b	0	1	0	1
c	1	1	0	0
d	0	1	0	0

Example :
Set A = {a, b, c, d}

Relation R on Set A is a relation from A to A ($\subseteq$ A x A)

R = {(a, d),(a, b),(b, b),(b, d), (c, b), (c, a), (d, b)}

Properties of a Relation

A relation R on a set A (A = {1, 2, 3, 4}) is

Reflexive : iff $(a,a) \in R$ for every element $a \in A$.
R = {(1, 1), (2, 2), (3, 3), (4, 4), (1, 2)} is reflexive

Symmetric : iff $(b,a) \in R$ whenever $(a,b) \in R$, for all $a,b \in A$.
R = {(1, 2), (2, 1), (1, 4), (4, 1), (3, 3)} symmetric

Transitive : iff $(a,c) \in R$ whenever $(a,b) \in R$ and $(b,c) \in R$, for all $a,b,c \in R$.
R = {(1,2), (2,1), (1,1)}

Automata Formal Languages & Logic

Properties of a Relation

Practice

Relation	Reflexive	Symmetric	Transitive

Automata Formal Languages & Logic

Properties of a Relation

Practice

Relation	Reflexive	Symmetric	Transitive
	Y	N	Y
	Y	Y	Y
	N	N	Y
	N	Y	N
	Y	Y	Y
	N	N	Y
	N	N	N
	N	N	N

Equivalence Relations

Denoted by $x \equiv y$

A relation R on a set A is called an equivalence relation if it is reflexive, symmetric, and transitive.

Example: Let S = {a,b,c,d,e,f}.

Relation R on set S be
{(a,a),(b,b), (b,c),(c,b),(c,c),(d,d), (d,e),(d,f),(e,d),(e,e),(e,f),(f,d), (f,e),(f,f)}

	a	b	c	d	e	f
a	1					
b		1	1			
c		1	1			
d				1	1	1
e				1	1	1
f				1	1	1

Equivalence Classes

Let R be an equivalence relation on a set A.

The set of all the elements that are related to an element 'a' of A is called equivalence class of 'a'.

Denoted by $[a]_R$ or [a] when the relation is implicit.

Example :

[b] = {b, c}

[d] = {d, e, f}

	a	b	c	d	e	f
a	1					
b		1	1			
c		1	1			
d				1	1	1
e				1	1	1
f				1	1	1

THANK YOU

Preet Kanwal

Department of Computer Science & Engineering

preetkanwal@pes.edu

+91 80 6666 3333 Extn 724