Program : **B.Tech**

Subject Name: **Discrete Structure**

Subject Code: **CS-302**

Semester: **3rd**

UNIT-3

Propositional Logic

The rules of mathematical logic specify methods of reasoning mathematical statements. Greek philosopher, Aristotle, was the pioneer of logical reasoning. Logical reasoning provides the theoretical base for many areas of mathematics and consequently computer science. It has many practical applications in computer science like design of computing machines, artificial intelligence, definition of data structures for programming languages etc.

Propositional Logic is concerned with statements to which the truth values, "true" and "false", can be assigned. The purpose is to analyze these statements either individually or in a composite manner.

Prepositional Logic – Definition

A proposition is a collection of declarative statements that has either a truth value "true" or a truth value "false". A propositional consists of propositional variables and connectives. We denote the propositional variables by capital letters (A, B, etc). The connectives connect the propositional variables.

Some examples of Propositions are given below −

i. "Man is Mortal", it returns truth value "TRUE"
ii. "12 + 9 = 3 – 2", it returns truth value "FALSE"

The following is not a Proposition −

"A is less than 2". It is because unless we give a specific value of A, we cannot say whether the statement is true or false.

First Order Logic

First-order logic (FOL) models the world in terms of

i. **Objects,** which are things with individual identities
ii. **Properties** of objects that distinguish them from other objects
iii. **Relations** that hold among sets of objects
iv. **Functions,** which are a subset of relations where there is only one "value" for any given "input"

Examples:

a. Objects: Students, lectures, companies, cars .
b. Relations: Brother-of, bigger-than, outside, part-of, has-color, occurs-after, owns, visits.
c. Properties: blue, oval, even, large.
d. Functions: father-of, best-friend, second-half, one-more-than .
e. Variable symbols E.g., x, y, foo
f. Connectives :Same as in PL: not ($\neg$), and ($\wedge$), or ($\vee$), implies ($\rightarrow$), if and only if (biconditional $\leftrightarrow$)

g. Quantifiers: Universal ∀**x** or **(Ax)** , Existential ∃**x** or **(Ex)**

Basic Logic Operation

In propositional logic generally we use five connectives which are –

i. OR (∨)
ii. AND (∧)
iii. Negation/ NOT (¬)
iv. Implication / if-then (→)
v. If and only if (⇔).

OR (∨) – The OR operation of two propositions A and B (written as $A \vee B$) is true if at least any of the propositional variable A or B is true. The truth table is as follows –

A	**B**	**A ∨ B**
True	True	True
True	False	True
False	True	True
False	False	False

Table 3.1 Operation of OR

AND (∧) – The AND operation of two propositions A and B (written as $A \wedge B$) is true if both the propositional variable A and B is true. The truth table is as follows –

A	**B**	**A ∧ B**
True	True	True
True	False	False
False	True	False
False	False	False

3.2 Operation of AND

Negation (¬) – The negation of a proposition A (written as ¬*A*) is false when A is true and is true when A is false. The truth table is as follows –

A	**¬ A**
True	False
False	True

Table 3.3 Operation of Negation

If and only if (⇔) – *A*⇔*B*- is bi-conditional logical connective which is true when p and q are same, i.e. both are false or both are true. The truth table is as follows –

A	**B**	**A ⇔ B**
True	True	True
True	False	False
False	True	False
False	False	True

Table 3.4 Operation of bi conditional

Truth Tables

A truth table lists all possible combinations of truth values. In a two-valued logic system, a single statement *p* has two possible truth values: truth (T) and falsehood (F). Given two statements *p* and *q*, there are four possible truth value combinations, that is, TT, TF, FT, FF. As a result, there are four rows in the truth table. With three statements, there are eight truth value combinations, ranging from TTT to FFF. In general, given *n* statements, there are 2*n* rows (or cases) in the truth table.

Example

A	**B**	**A ⇔ B**
True	True	True
True	False	False

False	True	False
False	False	True

Table 3.5 Truth Table

Tautologies - A Tautology is a formula which is always true for every value of its propositional variables.

Example − Prove $[(A \rightarrow B) \land A] \rightarrow B$ is a tautology

The truth table is as follows –

A	**B**	**A → B**	**(A → B) ∧ A**	**[(A → B) ∧ A] → B**
True	True	True	True	True
True	False	False	False	True
False	True	True	False	True
False	False	True	False	True

Table 3.6 Tautology

As we can see every value of $[(A \rightarrow B) \land A] \rightarrow B$

is "True", it is a tautology.

Contradictions - A Contradiction is a formula which is always false for every value of its propositional variables.

Example − Prove $(A \lor B) \land [(\neg A) \land (\neg B)]$ is a contradiction

The truth table is as follows –

A	**B**	**A ∨ B**	**¬ A**	**¬ B**	**(¬ A) ∧ (¬ B)**	**(A ∨ B) ∧ [(¬ A) ∧ (¬ B)]**
True	True	True	False	False	False	False
True	False	True	False	True	False	False

False	True	True	True	False	False	False
False	False	False	True	True	True	False

Table 3.7 Contradiction

As we can see every value of $(A \vee B) \wedge [(\neg A) \wedge (\neg B)]$ is "False", it is a contradiction.

<u>**Contingency**</u> **-** A Contingency is a formula which has both some true and some false values for every value of its propositional variables.

Example – Prove $(A \vee B) \wedge (\neg A)$ a contingency

The truth table is as follows –

A	**B**	**A ∨ B**	**¬ A**	**(A ∨ B) ∧ (¬ A)**
True	True	True	False	False
True	False	True	False	False
False	True	True	True	True
False	False	False	True	False

Table 3.8 Contingency

As we can see every value of $(A \vee B) \wedge (\neg A)$ has both "True" and "False", it is a contingency.

Algebra of Proposition

1. Identity:

$p \vee p \equiv p$ $\quad p \wedge p \equiv p$ $\quad p \rightarrow p \equiv T$ $\quad p \leftrightarrow p \equiv T$

$p \vee T \equiv T$ $\quad p \wedge T \equiv p$ $\quad p \rightarrow T \equiv T$ $\quad p \leftrightarrow T \equiv p$

$p \vee F \equiv p$ $\quad p \wedge F \equiv F$ $\quad p \rightarrow F \equiv \sim p$ $\quad p \leftrightarrow F \equiv \sim p$

$T \rightarrow p \equiv p$

$F \rightarrow p \equiv T$

2. Commutative:

$p \vee q \equiv q \vee p$ $\quad p \wedge q \equiv q \wedge p$ $\quad p \rightarrow q \neq q \rightarrow p$ $\quad p \leftrightarrow q \equiv q \leftrightarrow p$

3. Complement:

$p \vee \sim p \equiv T$ $\quad p \wedge \sim p \equiv F$ $\quad p \rightarrow \sim p \equiv \sim p$ $\quad p \leftrightarrow \sim p \equiv F$

$\sim p \rightarrow p \equiv p$

4. Double Negation:

$\sim(\sim p) \equiv p$

5. Associative:

$p \lor (q \lor r) \equiv (p \lor q) \lor r$

$p \land (q \land r) \equiv (p \land q) \land r$

6. Distributive:

$p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$

$p \land (q \lor r) \equiv (p \land q) \lor (p \land r)$

7. Absorbtion:

$p \lor (p \land q) \equiv p$

$p \land (p \lor q) \equiv p$

8. De Morgan's:

$\sim(p \lor q) \equiv \sim p \land \sim q$

$\sim(p \land q) \equiv \sim p \lor \sim q$

9. Equivalence of Contrapositive:

$p \rightarrow q \equiv \sim q \rightarrow \sim p$

10. Others:

$p \rightarrow q \equiv \sim p \lor q$

$p \leftrightarrow q \equiv (p \rightarrow q) \land (q \rightarrow p)$

<u>**Logical Implication**</u>

An implication $A \rightarrow B$ is the proposition "if A, then B". It is false if A is true and B is false. The rest cases are true. The truth table is as follows –

A	B	A → B
True	True	True
True	False	False
False	True	True
False	False	True

Table 3.9 Logical Implication

<u>**Logical Equivalences**</u>

Two statements X and Y are logically equivalent if any of the following two conditions hold –

i. The truth tables of each statement have the same truth values.
ii. The bi-conditional statement $X \Leftrightarrow Y$
iii. is a tautology.

Example – Prove $\neg(A \vee B) and [(\neg A) \wedge (\neg B)]$ are equivalent

Testing by 1st method (Matching truth table)

A	B	A V B	¬ (A V B)	¬ A	¬ B	[(¬ A) Λ (¬ B)]
True	True	True	False	False	False	False
True	False	True	False	False	True	False
False	True	True	False	True	False	False
False	False	False	True	True	True	True

Table 3.10 Truth Table

Here, we can see the truth values of $\neg(A \vee B) and [(\neg A) \wedge (\neg B)]$

are same, hence the statements are equivalent.

Testing by 2nd method (Bi-conditionality)

A	B	¬ (A V B)	[(¬ A) Λ (¬ B)]	[¬ (A V B)] ⇔ [(¬ A) Λ (¬ B)]
True	True	False	False	True
True	False	False	False	True
False	True	False	False	True
False	False	True	True	True

Table 3.11 Truth Table

As $[\neg(A \vee B)] \Leftrightarrow [(\neg A) \wedge (\neg B)]$

is a tautology, the statements are equivalent.

<u>Inverse, Converse, and Contra-positive</u>

1. Implication / if-then ($\rightarrow$) is also called a conditional statement. It has two parts –
   i. Hypothesis, p
   ii. Conclusion, q

As mentioned earlier, it is denoted as $p \rightarrow q$

Example of Conditional Statement – "If you do your homework, you will not be punished." Here, "you do your homework" is the hypothesis, p, and "you will not be punished" is the conclusion, q.

1. **Inverse** – An inverse of the conditional statement is the negation of both the hypothesis and the conclusion. If the statement is "If p, then q", the inverse will be "If not p, then not q". Thus the inverse of $p \rightarrow q$ is $\neg p \rightarrow \neg q$

Example – The inverse of "If you do your homework, you will not be punished" is "If you do not do your homework, you will be punished."

2. **Converse** – The converse of the conditional statement is computed by interchanging the hypothesis and the conclusion. If the statement is "If p, then q", the converse will be "If q, then p". The converse of $p \rightarrow q$ is $q \rightarrow p$

Example – The converse of "If you do your homework, you will not be punished" is "If you will not be punished, you do not do your homework".

3. **Contra-positive** – The contra-positive of the conditional is computed by interchanging the hypothesis and the conclusion of the inverse statement. If the statement is "If p, then q", the contra-positive will be "If not q, then not p". The contra-positive of $p \rightarrow q$ is $\neg q \rightarrow \neg p$

Example – The Contra-positive of " If you do your homework, you will not be punished" is "If you are not punished, then you do not do your homework".

Duality Principle

Duality principle states that for any true statement, the dual statement obtained by interchanging unions into intersections (and vice versa) and interchanging Universal set into Null set (and vice versa) is also true. If dual of any statement is the statement itself, it is said **self-dual** statement.

Example – The dual of $(A \cap B) \cup C$ is $(A \cup B) \cap C$

Predicate Logic - It deals with predicates, which are propositions containing variables.

Definition - A predicate is an expression of one or more variables defined on some specific domain. A predicate with variables can be made a proposition by either assigning a value to the variable or by quantifying the variable.

The following are some examples of predicates –

i. Let E(x, y) denote "x = y"
ii. Let X(a, b, c) denote "a + b + c = 0"
iii. Let M(x, y) denote "x is married to y"

Well Formed Formula - Well Formed Formula (wff) is a predicate holding any of the following –

i. All propositional constants and propositional variables are wffs
ii. If x is a variable and Y is a wff, $\forall xY$and $\exists xY$ are also wff
iii. Truth value and false values are wffs Each atomic formula is a wff
iv. All connectives connecting wffs are wffs

Normal Forms

We can convert any proposition in two normal forms –

i. Conjunctive normal form
ii. Disjunctive normal form

i. Conjunctive Normal Form - A compound statement is in conjunctive normal form if it is obtained by operating AND among variables (negation of variables included) connected with ORs. In terms of set operations, it is a compound statement obtained by Intersection among variables connected with Unions.

Example - $(A \vee B) \wedge (A \vee C) \wedge (B \vee C \vee D)$

ii. Disjunctive Normal Form- A compound statement is in conjunctive normal form if it is obtained by operating OR among variables (negation of variables included) connected with ANDs. In terms of set operations, it is a compound statement obtained by Union among variables connected with Intersections.

Example - $(A \wedge B) \vee (A \wedge C) \vee (B \wedge C \wedge D)$

Quantifiers - The variable of predicates is quantified by quantifiers. There are two types of quantifier in predicate logic – Universal Quantifier and Existential Quantifier.

1. **Universal Quantifier -** Universal quantifier states that the statements within its scope are true for every value of the specific variable. It is denoted by the symbol ∀.

$\forall x P(x)$ is read as for every value of x, P(x) is true.

Example – "Man is mortal" can be transformed into the propositional form $\forall x P(x)$

where P(x) is the predicate which denotes x is mortal and the universe of discourse is all men.

2. **Existential Quantifier -** Existential quantifier states that the statements within its scope are true for some values of the specific variable. It is denoted by the symbol ∃.

$\exists xP(x)$ is read as for some values of x, P(x) is true.

Example – "Some people are dishonest" can be transformed into the propositional form $\exists xP(x)$

where P(x) is the predicate which denotes x is dishonest and the universe of discourse is some people.

3. **Nested Quantifiers -** If we use a quantifier that appears within the scope of another quantifier, it is called nested quantifier.

Example

i. $\forall a \exists b P(x,y)$ where $P(a,b)$ denotes $a+b=0$
ii. $\forall a \forall b \forall c P(a,b,c)$ where $P(a,b)$ denotes $a+(b+c)=(a+b)+c$

Note – $\forall a \exists b P(x,y) \neq \exists a \forall b P(x,y)$

Rules of Inference - To deduce new statements from the statements whose truth that we already know, **Rules of Inference** are used.

- Mathematical logic is often used for logical proofs. Proofs are valid arguments that determine the truth values of mathematical statements.
- An argument is a sequence of statements. The last statement is the conclusion and all its preceding statements are called premises (or hypothesis). The symbol "∴ ", (read therefore) is placed before the conclusion. A valid argument is one where the conclusion follows from the truth values of the premises.
- Rules of Inference provide the templates or guidelines for constructing valid arguments from the statements that we already have.

Table of Rules of Inference

Rule of Inference Name Rule of Inference Name

1. **Addition**

If P is a premise, we can use Addiction rule to derive $P \vee Q$

$P \therefore P \vee Q$

Example

Let P be the proposition, "He studies very hard" is true

Therefore – "Either he studies very hard Or he is a very bad student." Here Q is the proposition "he is a very bad student".

2. **Conjunction**

If P and Q are two premises, we can use Conjunction rule to derive $P \wedge Q$

$PQ \therefore P \wedge Q$

Example

Let P – "He studies very hard"

Let Q – "He is the best boy in the class"

Therefore − "He studies very hard and he is the best boy in the class"

3. Simplification

If $P \wedge Q$ is a premise, we can use Simplification rule to derive P.

$P \wedge Q \therefore P$

Example

"He studies very hard and he is the best boy in the class", $P \wedge Q$

Therefore − "He studies very hard"

4. Modus Ponens

If P and $P \rightarrow Q$ are two premises, we can use Modus Ponens to derive Q.

$P \rightarrow Q P \therefore Q$

Example

"If you have a password, then you can log on to facebook", $P \rightarrow Q$

"You have a password", P Therefore − "You can log on to facebook"

5. Modus Tollens

If $P \rightarrow Q$and $\neg Q$ are two premises, we can use Modus Tollens to derive $\neg P$

$P \rightarrow Q \neg Q \therefore \neg P$

Example

"If you have a password, then you can log on to facebook", $P \rightarrow Q$

"You cannot log on to facebook", $\neg Q$

Therefore − "You do not have a password "

6. Disjunctive Syllogism

If $\neg P$ and $P \vee Q$ are two premises, we can use Disjunctive Syllogism to derive Q.

$\neg P P \vee Q \therefore Q$

Example

"The ice cream is not vanilla flavored", $\neg P$

"The ice cream is either vanilla flavored or chocolate flavored", $P \lor Q$

Therefore − "The ice cream is chocolate flavored”

7. Hypothetical Syllogism

If $P \rightarrow Q$ and $Q \rightarrow R$ are two premises, we can use Hypothetical Syllogism to derive $P \rightarrow R$

$$P \rightarrow Q \quad Q \rightarrow R \quad \therefore P \rightarrow R$$

Example

"If it rains, I shall not go to school”, $P \rightarrow Q$

"If I don't go to school, I won't need to do homework", $Q \rightarrow R$

Therefore − "If it rains, I won't need to do homework"

8. Constructive Dilemma

If $(P \rightarrow Q) \land (R \rightarrow S)$ and $P \lor R$ are two premises, we can use constructive dilemma to derive $Q \lor S$

$$(P \rightarrow Q) \land (R \rightarrow S) \quad P \lor R \quad \therefore Q \lor S$$

Example

“If it rains, I will take a leave”, $(P \rightarrow Q)$

“If it is hot outside, I will go for a shower”, $(R \rightarrow S)$

“Either it will rain or it is hot outside”, $P \lor R$

Therefore − "I will take a leave or I will go for a shower"

9. Destructive Dilemma

If $(P \rightarrow Q) \land (R \rightarrow S)$ and $\neg Q \lor \neg S$ are two premises, we can use destructive dilemma to derive $P \lor R$

$$(P \rightarrow Q) \land (R \rightarrow S) \quad \neg Q \lor \neg S \quad \therefore P \lor R$$

Numerical

1. Prove by truth table that the following is tautology.

$$(p \leftrightarrow q \land r) \Rightarrow (\sim r \rightarrow \sim p)$$

Solution: Given statement can be written as

$$[p \leftrightarrow (q \wedge r)\ ] \Rightarrow [(\sim r) \rightarrow (\sim p)]$$

Suppose $A \equiv p \leftrightarrow (q \wedge r)$ and $B \equiv$ **($\sim r \rightarrow$ $\sim p$)**then A $\Rightarrow B$ is a tautology.

Truth table:

p	q	r	$q \wedge r$	$A \equiv p \leftrightarrow (q \wedge r)$	$\sim r$	$\sim p$	$B \equiv (\sim r) \rightarrow (\sim p)$	$A \Rightarrow B$
T	T	T	T	**T**	F	F	**T**	**T**
T	T	F	F	**F**	T	F	**F**	**T**
T	F	T	F	**F**	F	F	**T**	**T**
T	F	F	F	**F**	T	F	**F**	**T**
F	T	T	T	**F**	F	T	**T**	**T**
F	T	F	F	**T**	T	T	**T**	**T**
F	F	T	F	**T**	F	T	**T**	**T**
F	F	F	F	**T**	T	T	**T**	**T**

Table 3.12 Truth Table

Thus the given statement is a tautology.

2. Obtain the principal disjunctive normal form of the following formula;-

$$\sim (p \vee q) \leftrightarrow (p \wedge q)$$

Solution: Given: $\sim(p \vee q) \Leftrightarrow (p \wedge q)$

$$\Leftrightarrow \quad [\sim(p \vee q) \Rightarrow (p \wedge q)] \wedge [(p \wedge q) \Rightarrow \sim (p \vee q)]$$

$$\Leftrightarrow \quad [\sim(p \vee q) \wedge (p \wedge q)] \vee [(p \vee q) \wedge \sim (p \wedge q)] \wedge [(p \wedge q) \wedge \sim (p \vee q)] \vee [(p \vee q) \wedge \sim (p \wedge q)]$$

$$\Leftrightarrow [\sim p \wedge \sim q \wedge p \wedge q] \vee [(p \vee q) \wedge (\sim p \vee \sim q)] \wedge [(p \wedge q) \wedge (\sim p \wedge \sim q)] \vee [(p \vee q) \wedge (\sim p \vee \sim q)]$$

$$\Leftrightarrow \quad [\sim p \wedge \sim q \wedge p \wedge q] \vee [(p \vee q) \wedge (\sim p \vee \sim q)] \qquad \text{[By De-Margon]}$$

$$\Leftrightarrow \quad [\sim p \wedge \sim q \wedge p \wedge q] \vee [\{(p \vee q) \wedge \sim p\} \vee \{(p \vee q) \wedge \sim q\}] \qquad \text{[By Distributive Law]}$$

$$\Leftrightarrow \quad [\sim p \wedge \sim q \wedge p \wedge q] \vee [(p \wedge \sim p) \vee (q \wedge \sim p) \vee (p \wedge \sim q) \vee (q \wedge \sim q)] \text{[By Distributive Law]}$$

This is required principle disjunctive normal form.

3. Investigate the validity of the following argument

$p \rightarrow r$

$\sim p \rightarrow q$

$q \rightarrow s$

$\therefore$ **$\sim r \rightarrow s$**

Solution- "If it rains, I will take a leave", $(P \rightarrow Q)$

"If it is hot outside, I will go for a shower", $(R \rightarrow S)$

"Either I will not take a leave or I will not go for a shower", $\neg Q \lor \neg S$

Therefore − "Either it rains or it is hot outside"

p	q	r	s	$p \rightarrow r$	$\sim p$	$\sim p \rightarrow q$	$q \rightarrow s$	$(p \rightarrow r) \wedge (\sim p \rightarrow q)$	A	$\sim r$	$B = \sim r \rightarrow s$	$A \Rightarrow B$
T	T	T	T	T	F	T	T	T	T	F	T	T
T	T	T	F	T	F	T	F	T	F	F	T	T
T	T	F	T	F	F	T	T	F	F	T	T	T
T	T	F	F	F	F	T	F	F	F	T	F	T
T	F	T	T	T	F	T	T	T	T	F	T	T
T	F	T	F	T	F	T	T	T	T	F	T	T
T	F	F	T	F	F	T	T	F	F	T	T	T
T	F	F	F	F	F	T	T	F	F	T	F	T
F	T	T	T	T	T	T	T	T	T	F	T	T
F	T	T	F	T	T	T	F	T	F	F	T	T
F	T	F	T	T	T	T	T	T	T	T	T	T
F	T	F	F	T	T	T	F	T	F	T	F	T
F	F	T	T	T	T	F	T	F	F	F	T	T
F	F	T	F	T	T	F	T	F	F	F	T	T
F	F	F	T	T	T	F	T	F	F	T	T	T
F	F	F	F	T	T	F	T	F	F	T	F	T

Table 3.13 Truth Table

Since all entries in the last column are of "*T*" only, therefore $A \Rightarrow B$ is a tautology. Hence the given argument.

4. Prove that the validity of the following argument:

"If Ram is selected in IAS examination, then he will not be able to go to London. Since Ram is going to London, he will not be selected in IAS examination. **Solution**: Suppose $p \equiv$ Ram is selected in IAS examination

$q \equiv$ Ram is going to London

The given argument, in symbolic form, may be written as

$p \to {\sim}q$ (Premises)

q (Premises)

$\therefore$ ${\sim}p$ (Conclusion)

Here two premises are $p \to {\sim}q$, q and conclusion is ${\sim}p$. The given argument will be valid if

$[(p \to {\sim}q) \wedge (q)] \Rightarrow ({\sim}p)$ is a tautology.

Suppose A $\equiv (p \to {\sim}q) \wedge (q)$ and B $\equiv {\sim}p$, then A $\Rightarrow$ B is a tautology. Truth table:

p	q	${\sim}q$	$p \to {\sim}q$	$A \equiv (p \to {\sim} q) \wedge q$	$B \equiv {\sim}p$	$A \Rightarrow B$
T	T	F	F	F	F	T
T	F	T	T	F	F	T
F	T	F	T	T	T	T
F	T	T	T	F	T	T

Table 3.14 Truth Table

Since all entries in the last column are of "*T*" only, therefore *A* ->*B* is a tautology. Hence the given argument is Valid.

<u>Introduction to Finite State Machine</u>

Finite state machine : a finite state machine (sometimes called a finite state automaton) is a computation model that can be implemented with hardware or software and can be used to simulate sequential logic and some computer programs. Finite state automata generate regular languages. Finite state machines can be used to model problems in many fields including mathematics, artificial intelligence, games, and linguistics.

A Finite state machine has 6 tuples (Q, Σ , $\delta : Q \times \Sigma \rightarrow Q$, g, q_0, q_f)

1. Finite set of states Q
2. Finite input alphabet Σ
3. Transition function $\delta : Q \times \Sigma \rightarrow Q$
4. Various possibilities for output g
5. Initial state q_0
6. Final state q_f

Finite state machines as models of physical system equivalence machines:

Deterministic Finite-State Automata

A DFSA can be formally defined as A = (Q, Σ, ∂, q_0, F):

- Q, a finite set of states
- Σ, a finite alphabet of input symbols
- q0 ∈ Q, an initial start state
- F ⊆ Q, a set of final states
- ∂ (delta): Q x Σ → Q, a transition function

We can define ∂ on words, ∂_w, by using a recursive definition:

- ∂_w : Q x Σ* → Q a function of (state, word) to a state
- $\partial_w(q,\varepsilon)$ = q in state q, output state q if word is ε
- $\partial_w(q,xa) = \partial(\partial_w(q,x),a)$ otherwise, use ∂ for one step and recurse

For an automaton A, we can define the language of A:

- $L(A) = \{w \in \Sigma^* : \partial_w(q_0,w) \in F\}$
- L (A) is a subset of all words w of finite length over Σ, such that the transition function $\partial_w(q_0,w)$ produces a state in the set of final states (F).
- Intuitively, if we think of the automaton as a graph structure, then the words in L(A) represent the “paths” which end in a final state. If we concatenate the labels from the edges in each such path, we derive a string w ∈L(A).

1. States are shown as circles;
2. the start state is indicated by the bold incoming arrow.
3. The next state function and output functions are shown using directed arrows from one state to

another. Each arrow is labeled with one element of I and one element of O.

4. If the machine is in some state s and the current input symbol is x, then we follow the arc labeled x/y from s to a new state and produce output y.

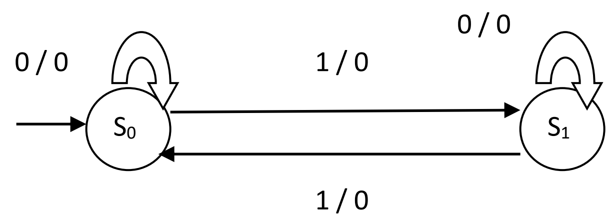

Figure 3.1

Finite State Machines as Language Recognizers

Input = {0,1}

acceptingstate : A state is said to be an accepting state if its output is 1.

rejectingstate : A state is said to be an rejecting state if its output is 0.

An input sequence is said to be accepted by the finite state machine if it leads the machine from the initial state to an accepting state. On the other hand, an input sequence is said to be rejected by the finite state machine if it leads the machine from the initial state to an rejecting state.

finite state language

A language is said to be a finite state language if there is a finite state machine that accepts exactly all sentences in the language.

Theorem Let L be a finite state language accepted by a finite state machine with N states. For any sequence α whose length is N or larger in the language, α can be written as uvw such that v is nonempty and uv^iw is also in the language for i≧0, where vi denotes the concatenation of i copies of the sequence v. (In other words, uw, uvw, uvvw, uvvvw, … are all in the language.)

Proof :

Letα= a1a2a3…aN, without loss of generality.

Let s_{j0}, s_{j1}, s_{j2}, …,s_{jN} denote the states the machine visits, where s_{j0} is the initial state and s_{jN} is an accepting state.

Among the N+1 states s_{j0}, s_{j1}, s_{j2}, …,s_{jN} there are two of them that are the same. Suppose that is state s_k, we realize that the sequences uw, uvw, uvvw, uvvvw, …, uv^iw, … will all lead the machine from the initial state s_{j0} to the accepting state s_{jN}.

