2.Formal Specifications Techniques

This techniques used for **mathematical** equation. In this techniques we have two notations.

2.1. Relational Notations

It is based on the concerts of **entities and attributes.**

Entities are **named elements** and **attributes** are **relations of entities**.

In this we have

2.1.1 Implicit Equation

2.1.2 Recurrence Equation

2.1.3 Algebraic Axioms

2.1.4 Regular Expressions

2.1.1 Implicit Equation

Implicit equations specify the properties of a solution **without stating a solution method**. Matrix inversion is specified as follows

$$M \times M' = I + E$$

Matrix inversion has the property that the original matrix (M) multiplied by its inverse (M1) yield an identify matrix, I denotes the identify matrix and E specifies allowable computational errors

2.1.2 Recurrence Relations

Its consists of an **initial part** called basis and one or more recursive parts

Example: Fibonacci Series

$$F(0) = 0$$

$$F(l)=l$$

$$F(N) = F(N-1) + F(N-2) \text{ for } N>1$$

2.1.3 Algebraic Axioms

It is used to specify the **properties** of abstract data types

Example: Stack

(Stk is of type STACK, itm is of type ITEM)

1. EMPTY(NEW) = true
2. EMPTY(PUSH(Stk,itm)) = false .
3. POP(NEW) = error
4. TOP(NEW) = error
5. POP(PUSH(stk,itm)) = stk
6. TOP(PUSH(stk,itm)) = item

2.1.4 Regular Expressions

The rules for forming regular expressions are as follows.

Axioms: The basis symbols in the alphabet of interest form regular expressions.

Alternation: If Rl and R2 are regular expressions, then (R1/R2) is a regular expression.

Composition: If Rl and R2 are regular expressions, then (R1.R2) is a RE

Closure: If Rl is a regular expressions, then (Rl)* is a regular expression.

Completeness: Nothing else is a regular expression.

Example: (a(b/c)) denotes {ab,ac}

2.2 STATE ORIENTED NOTATION

2.2.1 Decision tables

Decision tables are widely used in **data processing application**. A decision table is segmented into **four quadrants**, condition state, condition entry, action states and action entry

Example:

Conditions	**R1**	**R2**	**R3**
Withdrawal Amount <= Balance	T	F	F
Credit granted	-	T	F
Actions			
Withdrawal granted	T	T	F

2.2.2 Event tables

specify actions to be taken when **events occur under different set of conditions**. Event tables are viewed as two- dimensional tables or of higher dimension.

Example:

Staff	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
Designer	4	4	3	3	2	2	1
Developer	0	0	1	2	4	4	3
Tester	0	0	0	0	2	2	2
Total	4	4	4	5	8	8	6

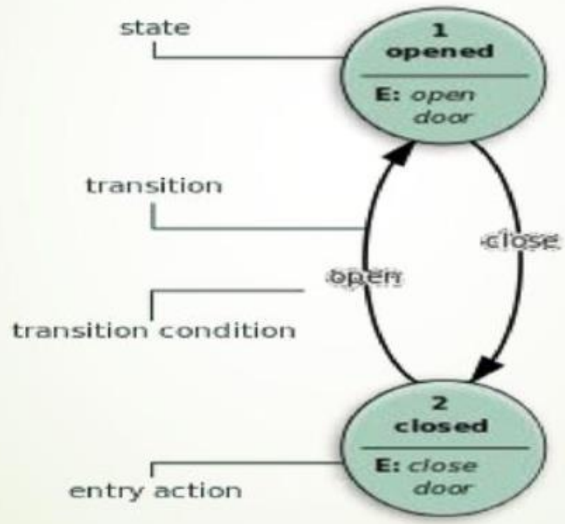
2.2.3 Transition Table

Transition Tables are used to specify **changes in the state** of a system as a function of next state

Current State	Current Input	
INPUT	A	B
S0	S0	S1
S1	S1 (next state)	S0

2 .2.4 Finite State Mechanisms:

utilize **data flow diagrams** in which the data streams are specified using **regular expressions** and the actions in the **processing nodes** are specified using transition labels.



2.2.5 Petri Net

It represent the **technique and systematic** methods have been developed for synthesizing and **analysing** petri nets. Petri nets were invented to **overcome the limitations of finite state mechanisms in specifying parallelism.**