Victoria University of Bangladesh

Name: Md. Ziaul Hoque “Sohel”

Student ID: 2221220031

Course Title: Theory Of Computing

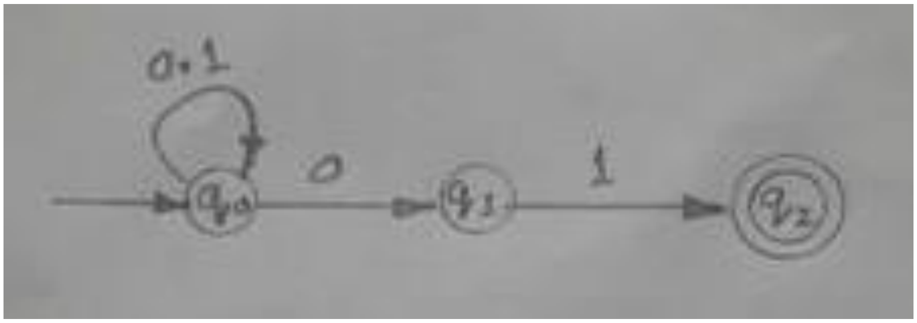
Course Code: CSI 317

Batch: 22nd(evening)

Ans to the Que No 1(A)

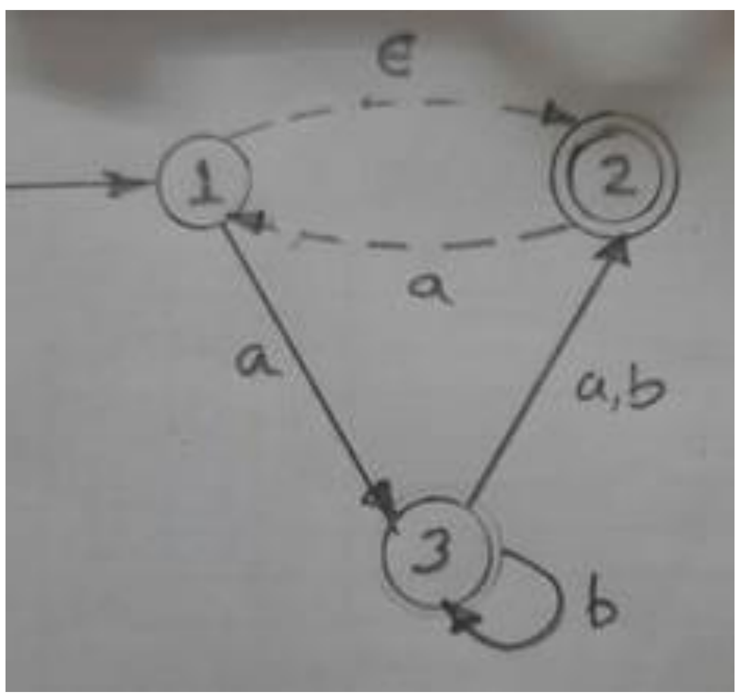
L = {w | w is a binary string ends in 01}

NFA,



δ N	0	1
→ q0	{q0,q1}	{q0}
q1	Ø	{q2}
*q2	Ø	Ø

Ans to the Que No 1B)



Let NFA N = (Q, Σ, δ, 1, F), where Q = {1, 2, 3}, Σ = {a, b}, 1 is the start state, F = {2}, and the transition function δ as in the diagram of N.

To construct a DFA M = (Q′ , Σ, δ′ , q′ 0 , F′) that is equivalent to NFA N, first we compute the ε-closure of every subset of Q = {1, 2, 3}.

Set R ⊆ Q	ε-closure E(R)
∅	∅
{1}	{1, 2}
{2}	{2}
{3}	{3}
{1, 2}	{1, 2}
{1, 3}	{1, 2, 3}
{2, 3}	{2, 3}
{1, 2, 3}	{1, 2, 3}

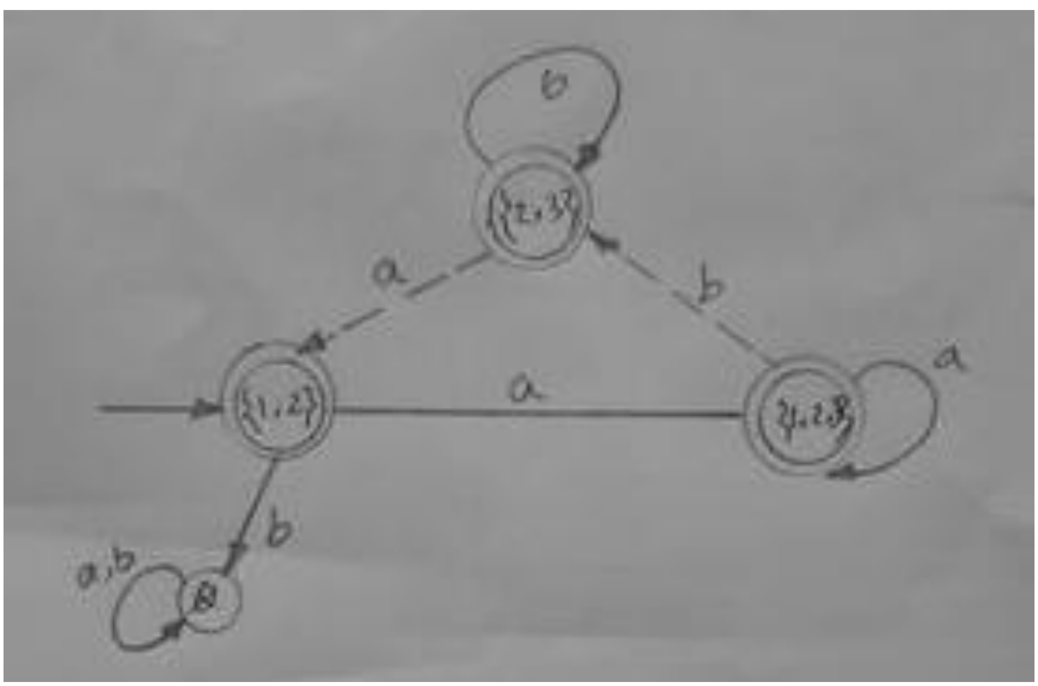
Then define Q′ = P(Q), so

Q ′ = { ∅, {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3} }.

The start state of M is then E({1}) = {1, 2}. The set of accept states of M is

F ′ = { {2}, {1, 2}, {2, 3}, {1, 2, 3} }

We define the transitions in the DFA M as in the following diagram:



Ans to the Que No 1(C)

Non-Terminal:

A set of non-terminals (V). Non-terminals are syntactic variables that denote sets of strings. The non-terminals define sets of strings that help define the language generated by the grammar. A set of tokens, known as terminal symbols (Σ). Terminals are the basic symbols from which strings are formed.

Ans to the Que No 2(A)

Ambiguous:

A context free grammar is called ambiguous if there exists more than one LMD or more than one RMD for a string which is generated by grammar. There will also be more than one derivation tree for a string in ambiguous grammar.

Context Free Grammars (CFG):

CFG stands for context-free grammar. It is is a formal grammar which is used to generate all possible patterns of strings in a given formal language. Context-free grammar G can be defined by four tuples as: G = (V, T, P, S)

G is the grammar, which consists of a set of the production rule. It is used to generate the string of a language.

T is the final set of a terminal symbol. It is denoted by lower case letters.

V is the final set of a non-terminal symbol. It is denoted by capital letters.

P is a set of production rules, which is used for replacing non-terminals symbols(on the left side of the production) in a string with other terminal or non-terminal symbols(on the right side of the production).

S is the start symbol which is used to derive the string. We can derive the string by repeatedly replacing a non-terminal by the right-hand side of the production until all non-terminal have been replaced by terminal symbols.

Ans to the Que No 3(A)

Chomsky Hierarchy represents the class of languages that are accepted by the different machine. The category of language in Chomsky's Hierarchy is as given below:

1. Type 0 known as Unrestricted Grammar.
2. Type 1 known as Context Sensitive Grammar.
3. Type 2 known as Context Free Grammar.
4. Type 3 Regular Grammar.

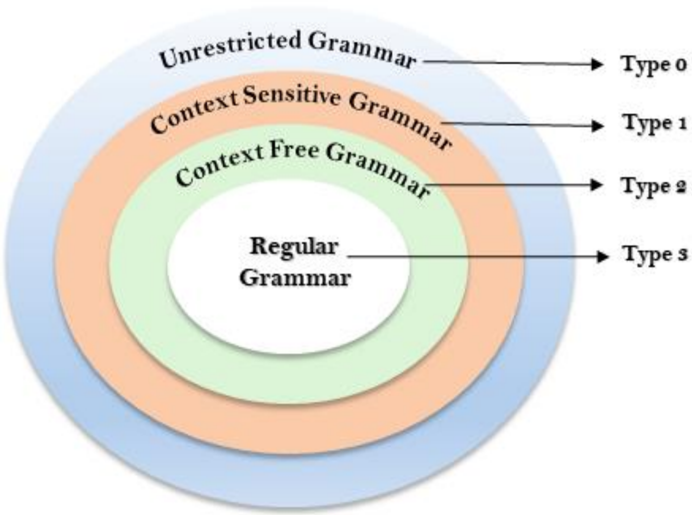


Fig: Chomsky Hierarchy

Type 0 known as Unrestricted Grammar : Type 0 grammar is known as Unrestricted grammar. There is no restriction on the grammar rules of these types of languages. These languages can be efficiently modeled by Turing machines.

Type 1 known as Context Sensitive Grammar: Type 1 grammar is known as Context Sensitive Grammar. The context sensitive grammar is used to represent context sensitive language.

Type 2 known as Context Free Grammar: Type 2 Grammar is known as Context Free Grammar. Context free languages are the languages which can be represented by the context free grammar (CFG). Type 2 should be type 1.

Type 3 Regular Grammar: Type 3 Grammar is known as Regular Grammar. Regular languages are those languages which can be described using regular expressions. These languages can be modeled by NFA or DFA.

Ans to the Que No 3(B)

Difference between Regular Language and Context free Language:

Context Free Grammar :

- Language generated by Context Free Grammar is accepted by Pushdown Automata
- It is a subset of Type 0 and Type 1 grammar and a superset of Type 3 grammar.
- Also called phase structured grammar.
- Different context-free grammars can generate the same context-free language.
- Classification of Context Free Grammar is done on the basis of the number of parse trees.
- Only one parse tree->Unambiguous.
- More than one parse tree->Ambiguous.

Regular Grammar:

- It is accepted by Finite State Automata.
- It is a subset of Type 0 ,Type 1 and Type 2 grammar.
- The language it generates is called Regular Language.
- Regular languages are closed under operations like Union, Intersection, Complement etc.
- They are the most restricted form of grammar.

Ans to the Que No 3(C)

To convert the regular expression to Finite Automata (FA) we can use the Subset method.

We will divide the given expression into three parts as follows –

“1” ,”(0+1)*, and “0”

NFA with Epsilon transition is as follows –

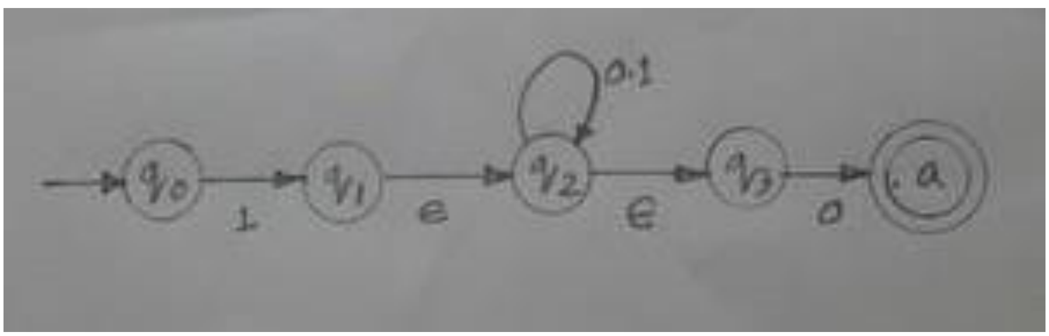


Fig: NDFA with NULL transition for RA: 1(0+1)*0

Now, we will remove the epsilon transition.

After removing, the **transition diagram** is given below –

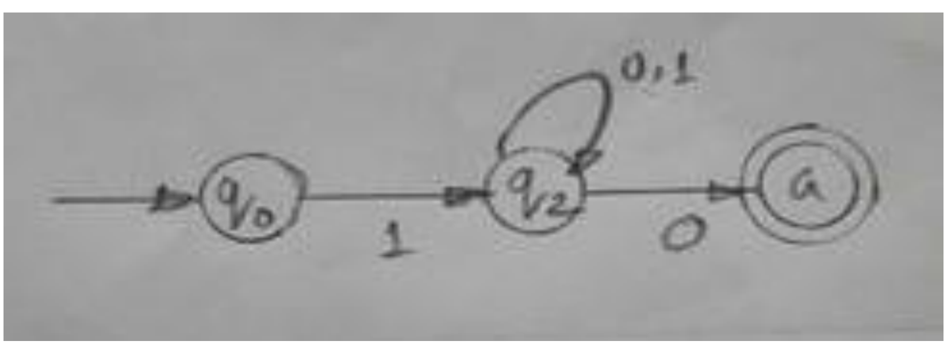


Fig: NDFA without NULL transition for RA: 1(0+1)*0

Ans to the Que No 5(A)

Turing machines:

Turing introduced Turing machines in the context of research into the foundations of mathematics. More particularly, he used these abstract devices to prove that there is no effective general method or procedure to solve.

Ans to the Que No 5(B)

Empty Stack:

On reading the input string from the initial configuration for some PDA, the stack of PDA gets empty.

Let P =(Q, ∑, Γ, δ, q0, Z, F) be a PDA. The language acceptable by empty stack can be defined as:

N(PDA) = {w | (q0, w, Z) ⊢* (p, ε, ε), q ∈ Q}

Equivalence of Acceptance by Final State and Empty Stack

- If L = N(P1) for some PDA P1, then there is a PDA P2 such that L = L(P2). That means the language accepted by empty stack PDA will also be accepted by final state PDA.
- If there is a language L = L (P1) for some PDA P1 then there is a PDA P2 such that L = N(P2). That means language accepted by final state PDA is also acceptable by empty stack PDA.

END