Compiler Construction

Humanly Project Documentation

BS-CS-5B

Course Code: CS-3102

Instructor: Ma'am Raazia Sosan

Student Names:

Aheed Uddin Basri (cs182010)

Muhammad Babar Siddiqi (cs182006)

Aaisha Motan (cs182009)

Table of Contents

Humanly

Introduction

Humanly as its name is a language that is designed to be as human-like as possible. It is inspired by the natural English keywords that people use in their daily life for explaining things very often. The purpose of this language is to reduce ambiguity related to different keywords and maximize understandability so that even a novice can find relations easily between computational logic and human daily tasks.

Lexical Aspects

The lexical aspects of the Humanly are as follows:

Rules for Identifiers

The rules for identifiers that will be used are as follows:

Rule 1. Humanly identifiers will be case sensitive. For example, *'name'* and *'Name'* will be considered as two different identifiers in Humanly.

Rule 2. Humanly identifiers can contain only alphanumeric characters **([a-z], [A-Z], [0-9])** and underscore **(_)**.

Rule 3. The first character of an identifier can only contain an alphabet **([a-z], [A-Z])**.

Rule 4. Identifiers should not start with digits **([0-9])**. For example, *"123humanly"* or *"123_humanly"* is an invalid identifier.

Rule 5. There should not be any whitespace in an identifier. For example, *"num ber"* or *"num 1"* is an invalid identifier.

Rule 6. Any kind of special characters, such as a *semicolon (;)*, *period(.)*, *whitespaces ( )*, *slash (\)*, or *comma(,)* are not permitted to be used in or as an Identifier. For example, *"human-ly"* is an invalid Humanly identifier rather you can use *"human_ly"*.

Rule 7. Reserved Words can't be used as an identifier. For example, *"NUM REPEAT= 20;"* is an invalid statement as *'REPEAT'* is a reserved word.

Rule 8. Humanly identifiers cannot contain only digits **([0-9])**. For example, *"NUM 20 = 10"* is an invalid statement.

Rule 9. There will be no limit on the length of the identifier but it is advisable to use an optimum length of **4 – 15** letters only (for further clarification on Naming Convention refer to Humanly Best Practices/Naming Conventions).

Constants

Integer & Decimal Constants

An integer constant is a sequence of one or more decimal digits ([0-9]). There are no negative integer constants; negative numbers may be obtained by negating an integer constant using the unary - operator. If a variable of type *NUM* is assigned to a constant, it can accept an integer constant. Whereas If a variable of type *DEC* is assigned to a constant, it can accept an integer constant with floating/decimal point.

String Constant

A string constant is a sequence of zero or more printable characters, spaces, or escape sequences surrounded by double quotes "". Each escape sequence starts with a backslash \ and stands for some sequence of characters. The escape sequences are as follows:

Escape Sequence	Meaning
\n	Newline
\t	Tab
\"	Double quote
\\	Backslash

Operators and Punctuators

- The binary operators are + - * / ~= = <- <> < > <= >= AND OR.
- (), {}, [], “,”, “.” and “;” are used as punctuators.
- A delimiter ";" is used to indicate end of line.
- A dot “.” is used to represent a floating-point number. For e.g. (12.5).
- Parentheses groups the expressions in the usual way.
- A leading minus sign negates an integer expression.
- The binary operators +, -, *, and / require integer operands and return an integer result.
- The binary operators >, <, >= and <= compare their operands, which may be either both integer or both string and produce the integer 1 if the comparison is true and 0 otherwise.
- The binary operators =, ~= and <> can compare any two operands of the same type and return either integer 0 or 1. Integers are the same if they have the same value. Strings are the same if they contain the same characters.

Comments

Anything that may appear between the comments will be ignored by the compiler. A comment in Humanly will begin with **/#** and ends with **#/**.

For Example:

```
NUM counter = 0; /# This is a counter to count the loop iterations #/
```

In the above example Humanly compiler will ignore the sentence written between /# #/.

Reserved Keywords

The reserved keywords that will be used for Humanly are as follows along with the comparison with a popular programming language C++.

Note: All the keywords will be case sensitive and will be written in Upper case ([A-Z]) letters only

Humanly	C++
IF	if
ELSE	else
BUTIF	else if
CHOOSE	switch
LISTEN	cin

SHOW	cout
METHOD	functions
REPEAT	while
COLLECTION	array

Data Types

Few of the data types that will be allowed in Humanly are as follows along with the comparison with a popular programming language C++.

Note: All the data types will be case sensitive and will be written in Upper case ([A-Z]) letters only.

Humanly	C++
NUM	int
WORD	string
DEC	float

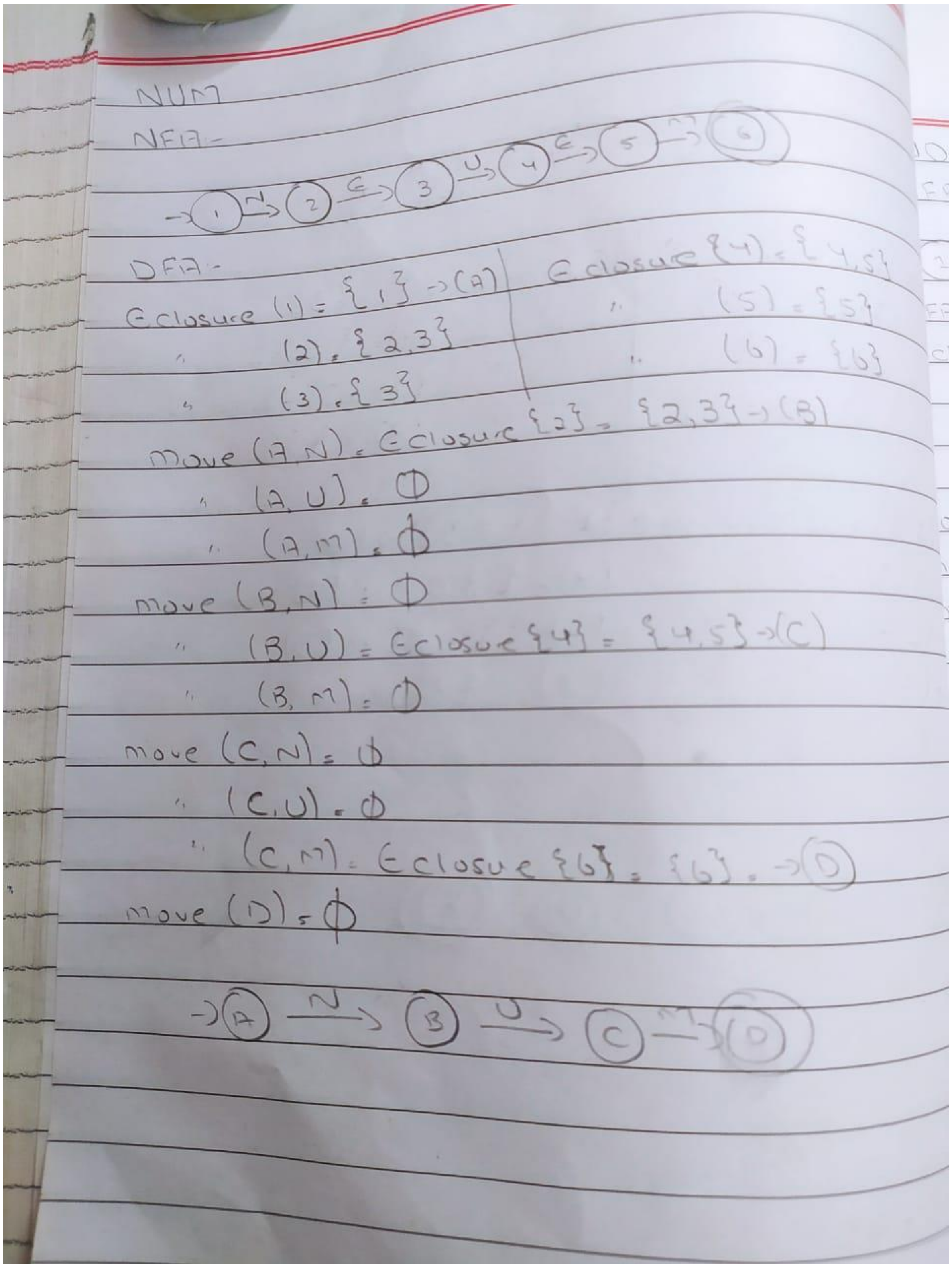
Humanly Best Practices/Naming Conventions

Few of the best practice/naming conventions that can be used are mentioned below:

- Use meaningful names that describe the actual data being stored, or the actual activity or meaning implemented by the function or method. (for example: counter can be used if a person is making a counter to count something)
- Names should begin with letters followed by a series of letters, numbers. (For example: num1 and not 1num)
- Names(identifiers/variables) should not start with numbers/digits/underscore. (for e.g.: words like 1num, _num should not be used)
- Name must be as precise as possible but meaningful. (for e.g.: count for counting)
- Naming convention should also justify rules of programming language such as use either lower casing, Capitalize or only camel casing would do a perfect job. (for e.g.: count, Count or LoopCount)
- Natural language words can also be used to define variable(s) name. (for e.g.: any word that is human understandable such as, count for counting)
- Avoid using special characters in naming conventions. (for e.g.: %, #, * etc. should not be used)
- Name should avoid using Upper casing as all reserved words and data types for humanly uses upper casing but will not be considered invalid if the identifier is not a keyword. (for e.g.: “NUM NUMBER = 20” is a valid identifier.)

NFA’s, DFA’s and Regular Expressions

Data Types



NUM

NFA:-

$$\begin{tikzcd} \arrow[r] & 1 \arrow[r, "N"] & 2 \arrow[r, "\epsilon"] & 3 \arrow[r, "U"] & 4 \arrow[r, "\epsilon"] & 5 \arrow[r, "M"] & \boxed{6} \end{tikzcd}$$

DFA:-

ϵclosure $(1) = \{1\} \to (A)$

" $(2) = \{2,3\}$

" $(3) = \{3\}$

ϵclosure $\{4\} = \{4,5\}$

" $(5) = \{5\}$

" $(6) = \{6\}$

move $(A, N) = \epsilon$closure $\{2\} = \{2,3\} \to (B)$

" $(A, U) = \phi$

" $(A, M) = \phi$

move $(B, N) = \phi$

" $(B, U) = \epsilon$closure $\{4\} = \{4,5\} \to (C)$

" $(B, M) = \phi$

move $(C, N) = \phi$

" $(C, U) = \phi$

" $(C, M) = \epsilon$closure $\{6\} = \{6\} \to (D)$

move $(D) = \phi$

$$\begin{tikzcd} \arrow[r] & A \arrow[r, "N"] & B \arrow[r, "U"] & C \arrow[r, "M"] & \boxed{D} \end{tikzcd}$$

WORD

NFA

$$\begin{tikzcd} \arrow[r] & 1 \arrow[r, "W"] & 2 \arrow[r, "\epsilon"] & 3 \arrow[r, "O"] & 4 \arrow[r, "\epsilon"] & 5 \arrow[r, "R"] & 6 \arrow[r, "\epsilon"] & 7 \arrow[r, "D"] & 8 \end{tikzcd}$$

DFA

ϵ closure $\{1\} = \{1\} \rightarrow (A)$
" $\{2\} = \{2,3\}$
" $\{3\} = \{3\}$
" $\{4\} = \{4,5\}$

ϵ closure $\{5\} = \{5\}$
" $\{6\} = \{6,7\}$
" $\{7\} = \{7\}$
" $\{8\} = \{8\}$

move $(A, W\} = \epsilon$ closure $(2) = \{2,3\} \rightarrow (B)$
move $(B, O) =$ " $(4) = \{4,5\} \rightarrow (C)$
move $(C, R) =$ " $(6) = \{6,7\} \rightarrow (D)$
move $(D, R\} =$ " $(8) = \{8\} \rightarrow (E)$
move $(E) = \phi$

$$\begin{tikzcd} \arrow[r] & 1 \arrow[r, "W"] & 2 \arrow[r, "O"] & 3 \arrow[r, "R"] & 4 \arrow[r, "D"] & 5 \end{tikzcd}$$

DEC

NFA:-

$$\begin{tikzcd} \arrow[r] & 1 \arrow[r, "D"] & 2 \arrow[r, "\epsilon"] & 3 \arrow[r, "E"] & 4 \arrow[r, "\epsilon"] & 5 \arrow[r, "C"] & 6 \end{tikzcd}$$

DFA:-

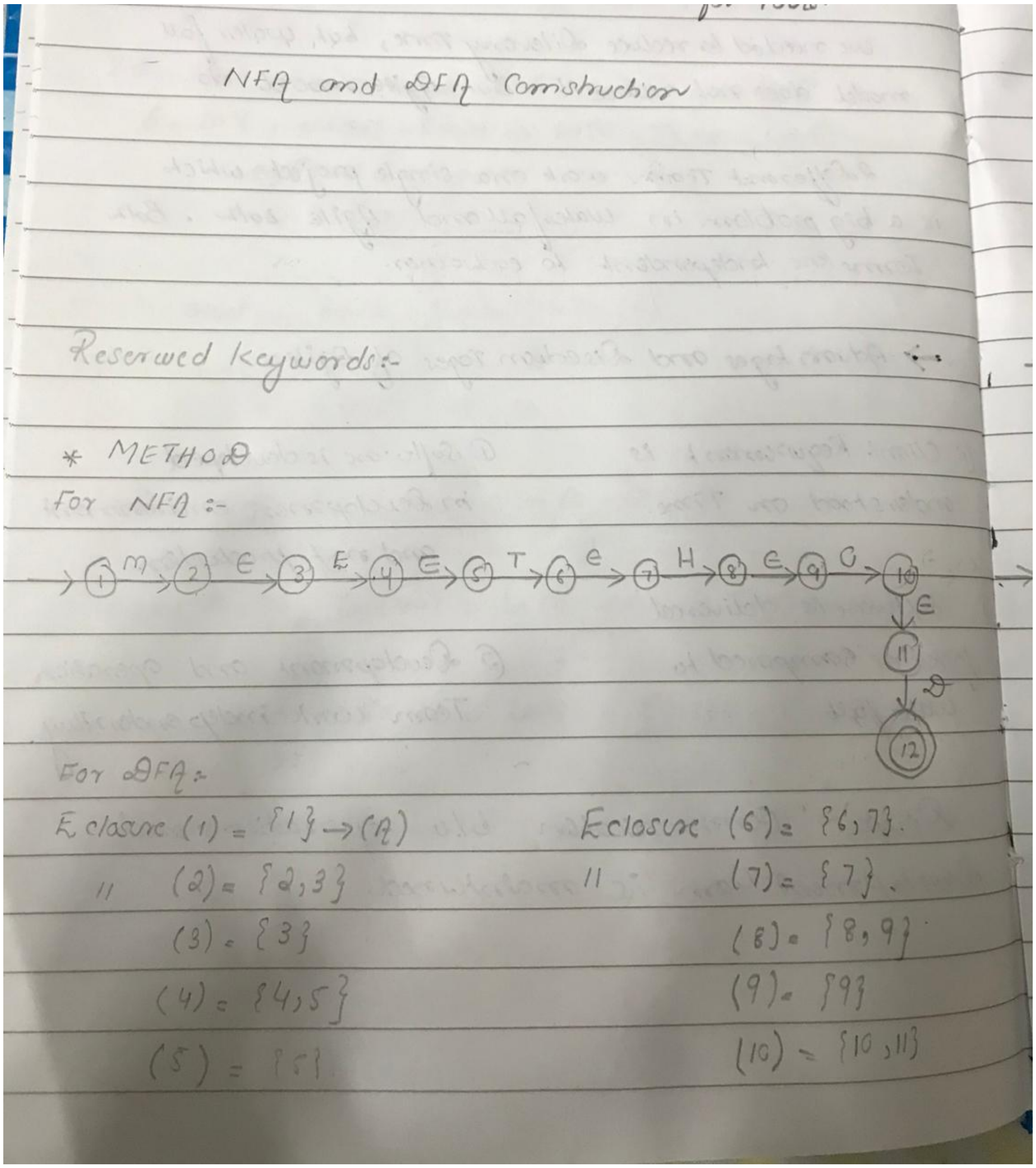
ϵ closure $\{1\} = \{1\} \rightarrow (A)$
" $\{2\} = \{2,3\}$
" $\{3\} = \{3\}$

ϵ closure $\{4\} = \{4,5\}$
" $\{5\} = \{5\}$
" $\{6\} = \{6\}$

move $(A, D) = \{2\} = \{2,3\} \rightarrow (B)$
" $(B, E) = \{4\} = \{4,5\} \rightarrow (C)$
" $(C, C) = \{6\} = \{6\} \rightarrow (D)$
" $(D) = \phi$

$$\begin{tikzcd} \arrow[r] & A \arrow[r, "D"] & B \arrow[r, "E"] & C \arrow[r, "C"] & D \end{tikzcd}$$

Reserevd Keywords



NFA and DFA Construction

Reserved Keywords :-

* METHOD

For NFA :-

$$\begin{tikzcd}
\arrow[r] & 1 \arrow[r, "m"] & 2 \arrow[r, "\epsilon"] & 3 \arrow[r, "E"] & 4 \arrow[r, "\epsilon"] & 5 \arrow[r, "T"] & 6 \arrow[r, "\epsilon"] & 7 \arrow[r, "H"] & 8 \arrow[r, "\epsilon"] & 9 \arrow[r, "O"] & 10 \arrow[d, "\epsilon"] \\
 & & & & & & & & & & 11 \arrow[d, "D"] \\
 & & & & & & & & & & 12
\end{tikzcd}$$

For DFA :-

ϵ closure (1) = {1} → (A)	ϵclosure (6) = {6, 7}.
〃 (2) = {2, 3}	〃 (7) = {7}.
(3) = {3}	(8) = {8, 9}
(4) = {4, 5}	(9) = {9}
(5) = {5}	(10) = {10, 11}

Move (A, m) = Eclosure {2} = {2, 3} → (B).
" (B, E) = Eclosure {4} = {4, 5} → (C)
" (C, T) = Eclosure {6} = {6, 7} → (D)
" (D, H) = Eclosure {8} = {8, 9} → (E)
" (E, O) = Eclosure {10} = {10, 11} → (F).
" (F, D) = Eclosure {12} = {12} → (G).
(G) = ∅

$$\begin{tikzcd} {} \arrow[r] & A \arrow[r, "m"] & B \arrow[r, "E"] & C \arrow[r, "T"] & D \arrow[r, "H"] & E \arrow[r, "O"] & F \arrow[r, "D"] & \boxed{G} \end{tikzcd}$$

* · REPEAT

FOR NFA :-

$$\begin{tikzcd} {} \arrow[r] & 1 \arrow[r, "R"] & 2 \arrow[r, "\epsilon"] & 3 \arrow[r, "E"] & 4 \arrow[r, "\epsilon"] & 5 \arrow[r, "P"] & 6 \arrow[r, "\epsilon"] & 7 \arrow[r, "E"] & 8 \arrow[r, "\epsilon"] & 9 \arrow[r, "A"] & 10 \arrow[r, "\epsilon"] & 11 \arrow[d, "T"] \\ &&&&&&&&&&& \boxed{12} \end{tikzcd}$$

FOR DFA :-

Eclosure (1) = {1} → (A)
" (2) = {2, 3}
" (3) = {3}
" (4) = {4, 5}.
" (5) = {5}.
" (6) = {6, 7}.
" (7) = {7}

Eclosure (8) = {8, 9}.
L " (9) = {9}.
" (10) = {10, 11}.
" (11) = {11}.
" (12) = {12}.

Date

Move (A, R) = Eclosure {2} = {2, 3} $\longrightarrow$ B.
" (B, E) = Eclosure {4} = {4, 5} $\longrightarrow$ C
" (C, P) = Eclosure {6} = {6, 7} $\longrightarrow$ D
" (D, E) = Eclosure {8} = {8, 9} $\longrightarrow$ E
" (E, A) = Eclosure {10} = {10, 11} $\longrightarrow$ F
" (F, T) = Eclosure {12} = {12} $\longrightarrow$ G.
" (G) = ϕ

$$\begin{tikzcd}
{} \arrow[r] & \circled{A} \arrow[r, "R"] & \circled{B} \arrow[r, "E"] & \circled{C} \arrow[r, "P"] & \circled{D} \arrow[r, "E"] & \circled{E} \arrow[r, "A"] & \circled{F} \arrow[r, "T"] & \circledcirc{G}
\end{tikzcd}$$

* COLLECTION.

For NFA:-

$$\begin{tikzcd}
{} \arrow[r] & 1 \arrow[r, "C"] & 2 \arrow[r, "\epsilon"] & 3 \arrow[r, "O"] & 4 \arrow[r, "\epsilon"] & 5 \arrow[r, "L"] & 6 \arrow[r, "\epsilon"] & 7 \arrow[r, "L"] & 8 \arrow[r, "\epsilon"] & 9 \arrow[d, "E"] \\
 & & & & & & & & & 10 \arrow[d, "\epsilon"] \\
 & & & & & & & & & 11 \arrow[d, "C"] \\
 & & & & & & & & & 12 \arrow[d, "\epsilon"] \\
 & & & 20 & 19 \arrow[l, "N"'] & 18 \arrow[l, "\epsilon"'] & 17 \arrow[l, "O"'] & 16 \arrow[l, "\epsilon"'] & 15 \arrow[l, "I"'] & 14 \arrow[l, "\epsilon"'] & 13 \arrow[l, "T"']
\end{tikzcd}$$

For DFA:-

Eclosure {1} = {1} → (A)
" {2} = {2,3}
" {3} = {3}
" {4} = {4,5}
" {5} = {5}
" {6} = {6,7}.
" {7} = {7}
" {8} = {8,9}
" {9} = {9}
" {10} = {10,11}

Eclosure {11} = {11}
" {12} = {12,13}
" {13} = {13}.
" {14} = {14,15}.
" {15} = {15}.
" {16} = {16,17}.
" {17} = {17}.
" {18} = {18,19}.
" {19} = {19}.
" {20} = {20}

move (A,C) = Eclosure {2} = {2,3} → B
" (B,O) = Eclosure {4} = {4,5} → C
" (C,L) = Eclosure {6} = {6,7} → D
" (D,L) = Eclosure {8} = {8,9} → E
" (E,E) = Eclosure {10} = {10,11} → F
" (F,C) = Eclosure {12} = {12,13} → G
" (G,T) = Eclosure {14} = {14,15} → H
" (H,I) = Eclosure {16} = {16,17} → I
" (I,O) = Eclosure (18) = {18,19} → J
" (J,N) = Eclosure (20) = {20} → K
" (K) = ∅

$$\begin{tikzcd}
\arrow[r] & \boxed{A} \arrow[r, "C"] & \boxed{B} \arrow[r, "O"] & \boxed{C} \arrow[r, "L"] & \boxed{D} \arrow[r, "L"] & \boxed{E} \arrow[r, "E"] & \boxed{F} \arrow[r, "C"] & \boxed{G} \arrow[r, "T"] & \boxed{H} \arrow[r, "I"] & \boxed{I} \arrow[r, "O"] & J \arrow[r, "N"] & \boxed{\boxed{K}}
\end{tikzcd}$$

LISTEN

$$\begin{tikzcd}[column sep=small]
{} \arrow[r] & 1 \arrow[r, "L"] & 2 \arrow[r, "\varepsilon"] & 3 \arrow[r, "I"] & 4 \arrow[r, "\varepsilon"] & 5 \arrow[r, "S"] & 6 \arrow[r, "\varepsilon"] & 7 \arrow[r, "T"] & 8 \arrow[r, "\varepsilon"] & 9 \arrow[r, "E"] & 10 \arrow[r, "\varepsilon"] & 11 \arrow[r, "N"] & \boxed{12}
\end{tikzcd}$$

Epsilon Closure:-

ε-closure (1) = {1} = ⓐ

" (2) = {2, 3}

" (3) = {3}

" (4) = {4, 5}

" (5) = {5}

" (6) = {6, 7}

" (7) = {7}

" (8) = {8, 9}

" (9) = {9}

" (10) = {10, 11}

" (11) = {11}

" (12) = {12}

ε-closure (move (a, L)) = ε-closure (2) = {2, 3} = ⓑ

" (move (b, I)) = " (4) = {4, 5} = ⓒ

" (move (c, S)) = " (6) = {6, 7} = ⓓ

" (move (d, T)) = " (8) = {8, 9} = ⓔ

" (move (e, E)) = " (10) = {10, 11} = ⓕ

" (move (f, N)) = " (12) = {12} = ⓖ

$$\begin{tikzcd}
{} \arrow[r] & a \arrow[r, "L"] & b \arrow[r, "I"] & c \arrow[r, "S"] & d \arrow[r, "T"] & e \arrow[r, "E"] & f \arrow[r, "N"] & \boxed{g}
\end{tikzcd}$$

SHOW
~~LISTEN~~

$$\begin{tikzcd} {} \arrow[r] & 1 \arrow[r, "S"] & 2 \arrow[r, "\varepsilon"] & 3 \arrow[r, "H"] & 4 \arrow[r, "\varepsilon"] & 5 \arrow[r, "O"] & 6 \arrow[r, "\varepsilon"] & 7 \arrow[r, "W"] & \circledcirc 8 \end{tikzcd}$$

Epsilon Closure:-

ε-closure $(1) = \{1\} =$ ⓐ
" $(2) = \{2, 3\}$
" $(3) = \{3\}$
" $(4) = \{4, 5\}$
" $(5) = \{5\}$
" $(6) = \{6, 7\}$
" $(7) = \{7\}$
" $(8) = \{8\}$

ε-closure $(\text{move}(a, S)) = \varepsilon$-closure $(2) = \{2, 3\} =$ ⓑ
" $(\text{move}(b, H)) =$ " $(4) = \{4, 5\} =$ ⓒ
" $(\text{move}(c, O)) =$ " $(6) = \{6, 7\} =$ ⓓ
" $(\text{move}(d, W)) =$ " $(8) = \{8\} =$ ⓔ

$$\begin{tikzcd} {} \arrow[r] & a \arrow[r, "S"] & b \arrow[r, "H"] & c \arrow[r, "O"] & d \arrow[r, "W"] & \circledcirc e \end{tikzcd}$$

CHOOSE

$$\begin{tikzcd}
{} \arrow[r] & 1 \arrow[r, "C"] & 2 \arrow[r, "\varepsilon"] & 3 \arrow[r, "H"] & 4 \arrow[r, "\varepsilon"] & 5 \arrow[r, "O"] & 6 \arrow[r, "\varepsilon"] & 7 \arrow[r, "O"] & 8 \arrow[r, "\varepsilon"] & 9 \arrow[r, "S"] & 10 \arrow[r, "\varepsilon"] & 11 \arrow[r, "E"] & 12
\end{tikzcd}$$

Epsilon Closure :-

ε-closure $(1) = \{1\} = \textcircled{a}$

" $(2) = \{2,3\}$

" $(3) = \{3\}$

" $(4) = \{4,5\}$

" $(5) = \{5\}$

" $(6) = \{6,7\}$

" $(7) = \{7\}$

" $(8) = \{8,9\}$

" $(9) = \{9\}$

" $(10) = \{10,11\}$

" $(11) = \{11\}$

" $(12) = \{12\}$

ε-closure $(\text{move}(a, C)) = \varepsilon$-closure $(2) = \{2,3\} = \textcircled{b}$

" $(\text{move}(b, H)) =$ " $(4) = \{4,5\} = \textcircled{c}$

" $(\text{move}(c, O)) =$ " $(6) = \{6,7\} = \textcircled{d}$

" $(\text{move}(d, O)) =$ " $(8) = \{8,9\} = \textcircled{e}$

" $(\text{move}(e, S)) =$ " $(10) = \{10,11\} = \textcircled{f}$

" $(\text{move}(f, E)) =$ " $(12) = \{12\} = \textcircled{g}$

$$\begin{tikzcd}
{} \arrow[r] & a \arrow[r, "C"] & b \arrow[r, "H"] & c \arrow[r, "O"] & d \arrow[r, "O"] & e \arrow[r, "S"] & f \arrow[r, "E"] & g
\end{tikzcd}$$

Name: Aheed   Reg: CS182010   Date:

Reserved Keywords:-

IF

NFA

$$\begin{tikzcd} {} \arrow[r] & \boxed{1} \arrow[r, "I"] & \boxed{2} \arrow[r, "\epsilon"] & \boxed{3} \arrow[r, "F"] & \boxed{\boxed{4}} \end{tikzcd}$$

DFA:

ϵclosure (1) = {1} → (A)

" (2) = {2, 3}

" (3) = {3}

(4) = {4}

move (A, I) = {1} . ϵclosure {2} = {2,3} → (B)

move (A, F) = {1} = ϕ

" (B, I) = {3, 4} = ϕ

" (B, F) = {3 4} . ϵ closure{4} = {4} → (C)

$$\begin{tikzcd} {} \arrow[r] & \boxed{A} \arrow[r, "I"] & \boxed{B} \arrow[r, "F"] & \boxed{\boxed{C}} \end{tikzcd}$$

Date:

ELSE

NFA

$$\begin{tikzcd} \arrow[r] & 1 \arrow[r, "E"] & 2 \arrow[r, "\epsilon"] & 3 \arrow[r, "L"] & 4 \arrow[r, "\epsilon"] & 5 \arrow[r, "S"] & 6 \arrow[r, "\epsilon"] & 7 \arrow[r, "E"] & 8 \end{tikzcd}$$

DFA:-

Eclosure (1) = {1} → (A)	∈closure (5) = {5}
〃 (2) = {2,3}	〃 (6) = {6,7}
〃 (3) = {3}	〃 (7) = {7}
〃 (4) = {4,5}	〃 (8) = {8}

move (A, E) = ∈ closure {2} = {2,3} → (B)

move (B, L) = 〃 (4) = {4,5} → (C)

move (C, S) = 〃 (6) = {6,7} → (D)

move (D, E) = 〃 (8) = {8} → (E)

move (E) = ϕ

$$\begin{tikzcd} \arrow[r] & A \arrow[r, "E"] & B \arrow[r, "L"] & C \arrow[r, "S"] & D \arrow[r, "E"] & E \end{tikzcd}$$

BUT9F

$$\begin{tikzcd} \arrow[r] & 1 \arrow[r, "B"] & 2 \arrow[r, "\varepsilon"] & 3 \arrow[r, "U"] & 4 \arrow[r, "\varepsilon"] & 5 \arrow[r, "T"] & 6 \arrow[r, "\varepsilon"] & 7 \arrow[r, "9"] & 8 \arrow[r, "\varepsilon"] & 9 \arrow[r, "F"] & 10 \end{tikzcd}$$

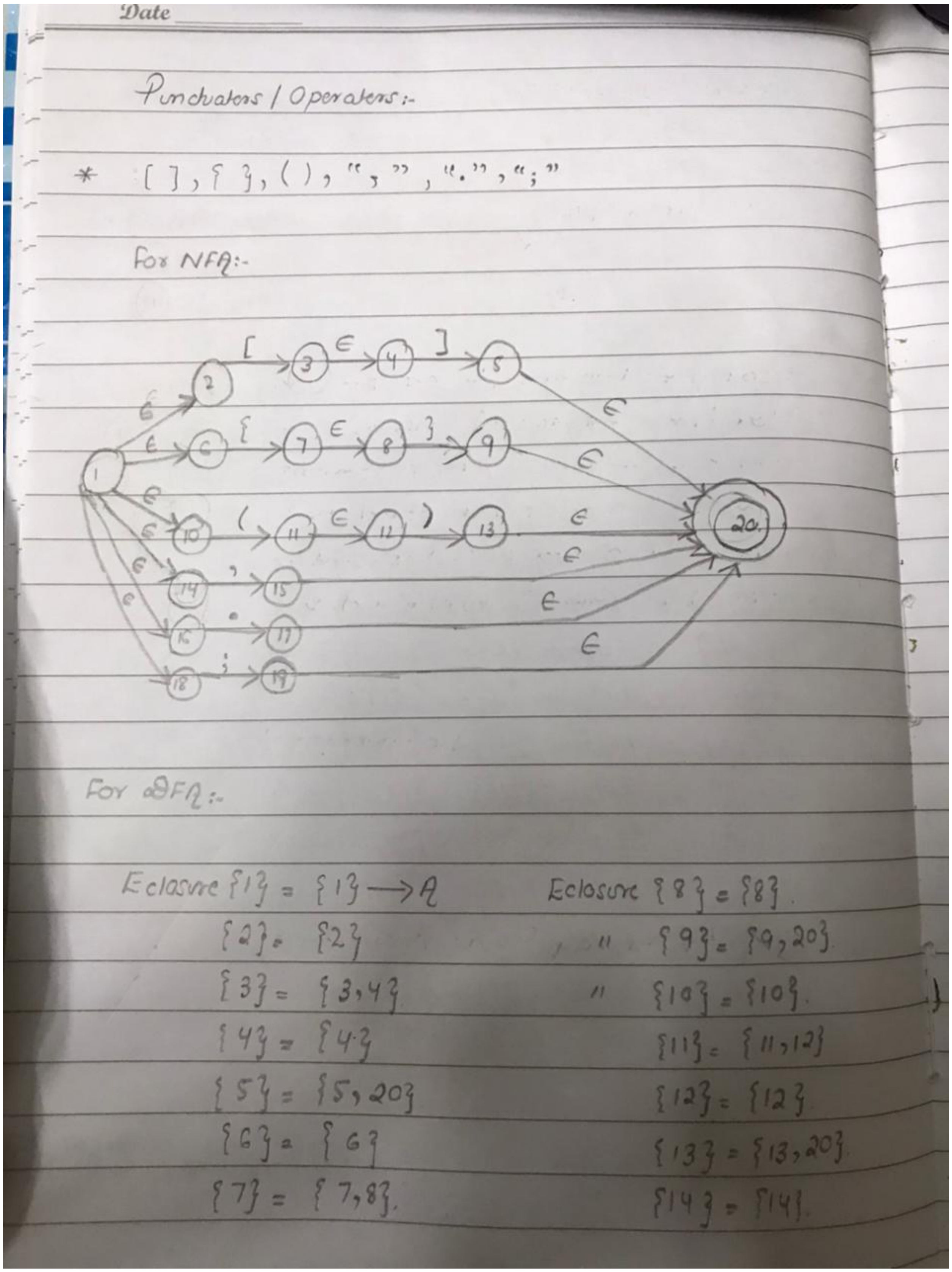
Epsilon Closure:-

ε - closure $(1) = \{1\} =$ a
" $(2) = \{2,3\}$
" $(3) = \{3\}$
" $(4) = \{4,5\}$
" $(5) = \{5\}$
" $(6) = \{6,7\}$
" $(7) = \{7\}$
" $(8) = \{8,9\}$
" $(9) = \{9\}$
" $(10) = \{10\}$

ε-closure (move (a, B)) = ε-closure $(2) = \{2,3\} =$ b
" (move (b, U)) = " $(4) = \{4,5\} =$ c
" (move (c, T)) = " $(6) = \{6,7\} =$ d
" (move (d, 9)) = " $(8) = \{8,9\} =$ e
" (move (e, F)) = " $(10) = \{10\} =$ f
~~" (move [illegible]~~

$$\begin{tikzcd} \arrow[r] & a \arrow[r, "B"] & b \arrow[r, "U"] & c \arrow[r, "T"] & d \arrow[r, "9"] & e \arrow[r, "F"] & f \end{tikzcd}$$

Punctuators

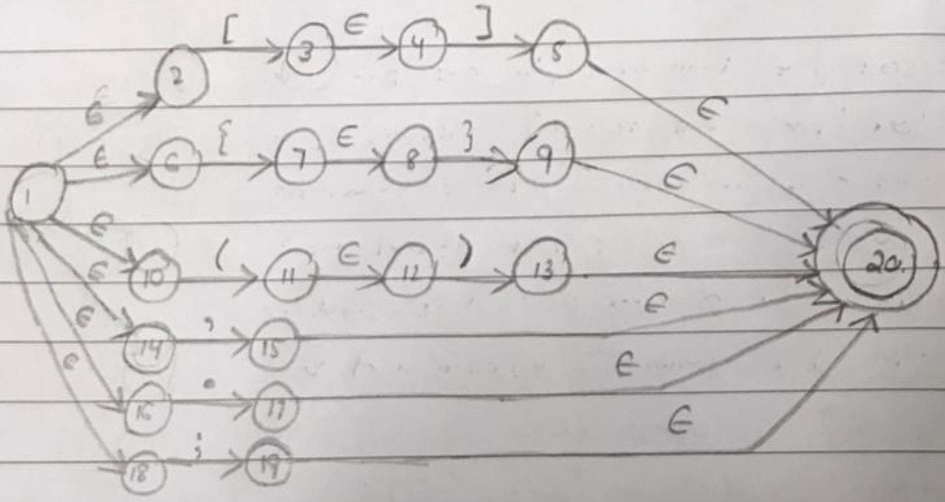


Date

Punctuators / Operaters :-

* [ ], { }, ( ), “ , ” , “.” , “;”

For NFA:-



For DFA :-

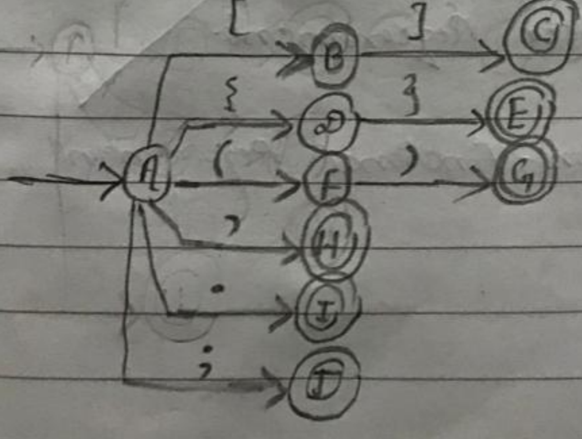
Eclosure {1} = {1} $\longrightarrow$ A
{2} = {2}
{3} = {3, 4}
{4} = {4}
{5} = {5, 20}
{6} = {6}
{7} = {7, 8}

Eclosure {8} = {8}
" {9} = {9, 20}
" {10} = {10}
{11} = {11, 12}
{12} = {12}
{13} = {13, 20}
{14} = {14}

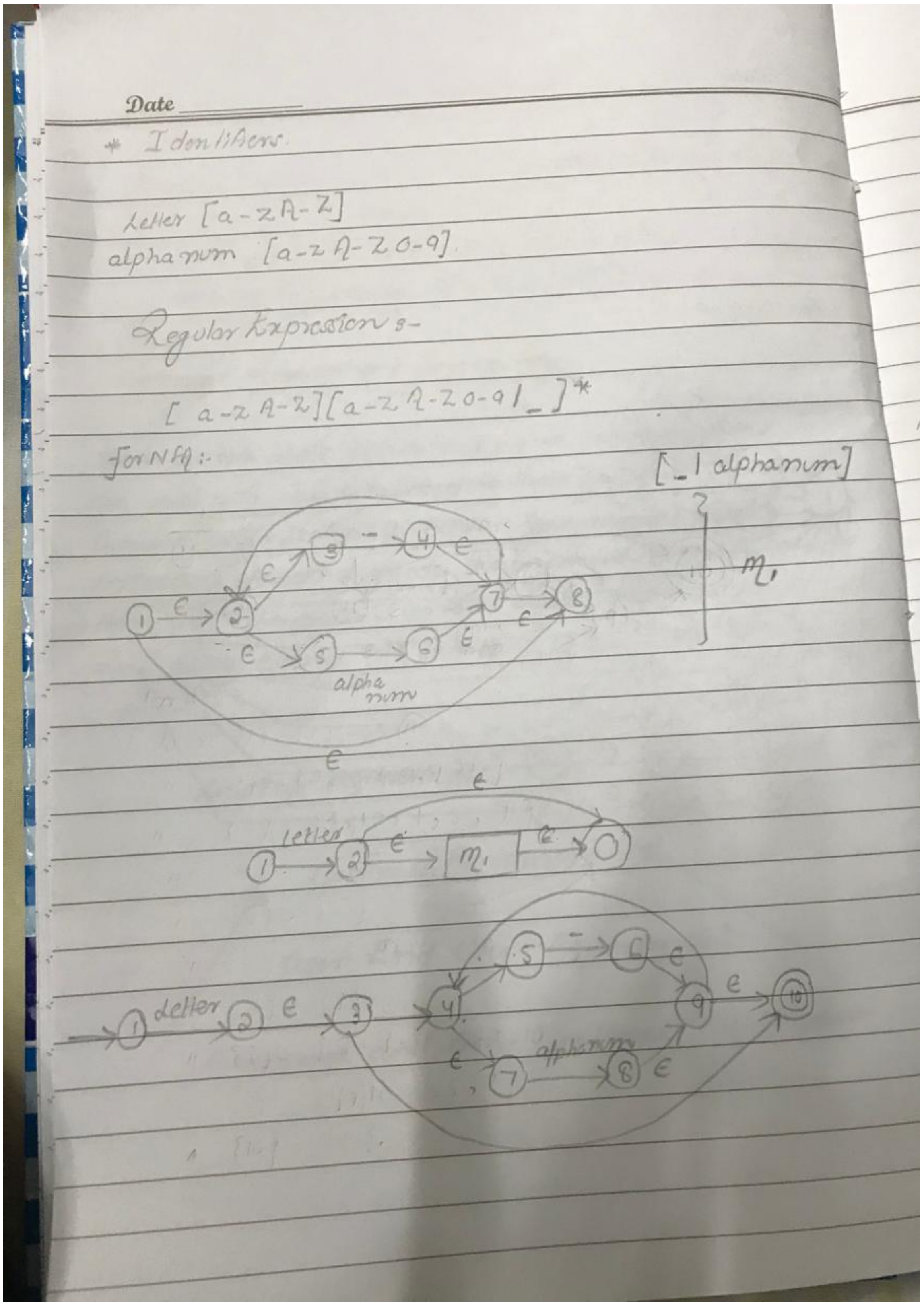
Date

Eclosure {15} = {15, 20}
" {16} = {16}.
" {17} = {17, 20}.
" {18} = {18}.
" {19} = {19, 20}.
" {20} = {20}.

Move (A, [) = Eclosure {3} = {3, 4} $\rightarrow$ B
(B,]) = Eclosure {5} = {5, 20} $\rightarrow$ C
(C, {) = Eclosure {7} = {7, 8} $\rightarrow$ D
(D, }) = Eclosure {9} = {9, 20} $\rightarrow$ E
(E, () = Eclosure {11} = {11, 12} $\rightarrow$ F
(F,)) = Eclosure {13} = {13, 20} $\rightarrow$ G
(G, ,) = Eclosure {15} = {15, 20} $\rightarrow$ H
(H, ·) = Eclosure {17} = {17, 20} $\rightarrow$ I
(I, ;) = Eclosure {19} = {19, 20} $\rightarrow$ J
(J) = ϕ



Identifiers



* Identifiers.

letter [a-z A-Z]
alphanum [a-z A-Z 0-9].

Regular Expression:-

[a-z A-Z][a-z A-Z 0-9|_]*

for NFA:-

[_| alphanum]

Date

Eclosure {1} = {1} → Ⓐ

" {2} = {2, 3, 4, 5, 7, 10}

{3} = {3, 4, 5, 7, 10}

{4} = {4, 5, 7}.

{5} = {5}.

Eclosure {6} = {6, 9, 10, 4, 5, 7}

" {7} = {7}

{8} = {8, 9, 10, 4, 5, 7}

{9} = {4, 9, 10, 5, 7}

{10} = {10}

Move (A, letter) = Eclosure {2} = {2, 3, 4, 5, 7, 10} → Ⓑ

" (B, -) = ∅

(A, alphanum) = ∅

Move (B, Letter) = ∅

" (B, -) = Eclosure (6) = {4, 5, 7, 6, 9, 10} → Ⓒ

(B, alphanum) = " (8) = {4, 5, 7, 8, 9, 10} → Ⓓ

Move (C, Letter) = ∅

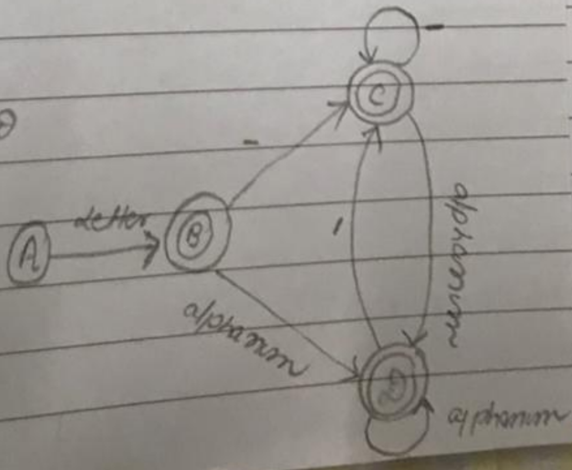
" (C, -) = Eclosure (6) = {4, 5, 7, 6, 9, 10} → Ⓒ

" (C, alphanum) = " (8) = {4, 5, 7, 8, 9, 10} → Ⓓ

Move (D, Letter) = ∅

" (D, -) = Eclosure (6) = C

" (D, alphanum) = " (8) = D


A
B
C
D
Letter
-
-
-
alphanum
alphanum
alphanum