**A logo of a globe with yellow rings around it

Description automatically generated**

**GROUP ASSIGNMENT**

**TECHNOLGY PARK MALAYSIA**

**CT111-3-2-COMT**

**COMPUTING THEORY**

**APD2F2309CS(CYB)**

**HANDOUT DATE:  16 OCTOBER 2023**

**HAND-IN DATE:  16 NOVEMBER 2023**

**WEIGHTAGE: 40%**

|  |  |
| --- | --- |
| **NAME:** | **TP NUMBER:** |
| **YAZEN ABOBAKR AHMED AL-MEHDHAR** | **TP069210** |
| **ABDULRAHMAN GAMIL MOHAMMED AHMED** | **TP071012** |
| **ABDULELAH HUSSEIN ABDULRAHMAN AL-KAF** | **TP069319** |
| **HAZIQ IRFAN BIN RADZALI** | **TP072306** |

PART A

[**1.** **Code Sinppit** 3](#_Toc151723238)

[**2.** **NFA Design** 4](#_Toc151723239)

[**3.** **Quintuple and Stat Transition Table** 5](#_Toc151723240)

[**4.** **DFA Conversion** 6](#_Toc151723241)

[**5.** **Regular Expression** 12](#_Toc151723242)

[**6.** **REFERNCES:** 19](#_Toc151723243)

**PART A:**

# **Code Sinppit**

A computer screen shot of numbers

Description automatically generated

The user is prompted to enter numbers from 1 to 5 in ascending order using this Python code. It will not stop requesting input until the user enters five real numbers. The user is prompted to enter a valid number if the number they entered falls outside of the range of 1 to 5. It also makes sure that every number entered is bigger than or equal to the one before it. The application prints the five valid numbers in ascending order and ends when the user enters them successfully

# **NFA Design**

A diagram of a flowchart

Description automatically generated

We have converted the python code to the NFA. The NFA will accept the string in ascending order, and the alphabet: 1-5 can be repeated as many times as necessary. Between every state there is an epsilon, and the state that you will start with allows you to make a loop and move to the other state until you get to the accept state. Having an epsilon transition between the states assists with skipping the numbers that are not required while maintaining.

# **Quintuple and Stat Transition Table**

**Characters: 1, 2, 3, 4, 5**

**q (States):** q0, q1, q2, q3, q4, q5

**Initial State :** q0

**Accept State :** q5

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STATES | 1 | 2 | 3 | 4 | 5 | E |
| q0 | ∅ | ∅ | ∅ | ∅ | ∅ | q1, q2, q3, q4, q5 |
| q1 | q1 | ∅ | ∅ | ∅ | ∅ | q2 |
| q2 | ∅ | q2 | ∅ | ∅ | ∅ | q3 |
| q3 | ∅ | ∅ | q3 | ∅ | ∅ | q4 |
| q4 | ∅ | ∅ | ∅ | q4 | ∅ | q5 |
| q5 | ∅ | ∅ | ∅ | ∅ | q5 | ∅ |

# **DFA Conversion**

**Step 1:** we will put all the initial states, and accepts states that we created in the NFA in the table , for each state that we have, we need to check if the state go an E\* (epsilon closure)

**Step 2:** we will set of the states that we get here, and it have to be checked on which state do they go on getting a particular input

**Step 3:** after the second step we will go through all the states to check on see on which state do they go on E\* (epsilon closure)

**E\* (epsilon closure):** lastly we will look at all the states that can be reached from a particularState that can only be seen by the'E' symbol

A diagram of a flowchart

Description automatically generated

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STATES | 1 | 2 | 3 | 4 | 5 |
| Q1 | q1, q2, q3, q4, q5 | q2, q3, q4, q5 | q3, q4, q5 | q4, q5 | q5 |
| Q2 | ∅ | q2, q3, q4, q5 | q3, q4, q5 | q4, q5 | q5 |
| Q3 | ∅ | q2, q3, q4, q5 | q3, q4, q5 | q4, q5 | q5 |
| Q4 | ∅ | ∅ | q3, q4, q5 | q4, q5 | q5 |
| Q5 | ∅ | ∅ | ∅ | q4, q5 | q5 |
| Q6 | ∅ | ∅ | ∅ | ∅ | q5 |

q0:

|  |  |  |
| --- | --- | --- |
| E\* | 1 | E\* |
| q0  q1  q2  q3  q4  q5 | ∅  q1  ∅  ∅  ∅  ∅ | q1  q2  q3  q4  q5 |

|  |  |  |
| --- | --- | --- |
| E\* | 2 | E\* |
| q0  q1  q2  q3  q4  q5 | ∅  ∅  q2  ∅  ∅  ∅ | q2  q3  q4  q5 |

|  |  |  |
| --- | --- | --- |
| E\* | 3 | E\* |
| q0  q1  q2  q3  q4  q5 | ∅  ∅  ∅  q3  q4  q5 | q3  q4  q5 |

|  |  |  |
| --- | --- | --- |
| E\* | 4 | E\* |
| q0  q1  q2  q3  q4  q5 | ∅  ∅  ∅  ∅  q4  ∅ | q4  q5 |

|  |  |  |
| --- | --- | --- |
| E\* | 5 | E\* |
| q0  q1  q2  q3  q4  q5 | ∅  ∅  ∅  ∅  ∅  q5 | q5 |

q1:

|  |  |  |
| --- | --- | --- |
| E\* | 1 | E\* |
| q1  q2  q3  q4  q5 | q1  ∅  ∅  ∅  ∅ | q1  q2  q3  q4  q5 |

|  |  |  |
| --- | --- | --- |
| E\* | 2 | E\* |
| q1  q2  q3  q4  q5 | ∅  q2  ∅  ∅  ∅ | q2  q3  q4  q5 |

|  |  |  |
| --- | --- | --- |
| E\* | 3 | E\* |
| q1  q2  q3  q4  q5 | ∅  ∅  q3  ∅  ∅ | q3  q4  q5 |

|  |  |  |
| --- | --- | --- |
| E\* | 4 | E\* |
| q1  q2  q3  q4  q5 | ∅  ∅  ∅  q4  ∅ | q4  q5 |

|  |  |  |
| --- | --- | --- |
| E\* | 5 | E\* |
| q1  q2  q3  q4  q5 | ∅  ∅  ∅  ∅  q5 | q5 |

q2:

|  |  |  |
| --- | --- | --- |
| E\* | 1 | E\* |
| q2  q3  q4  q5 | ∅  ∅  ∅  ∅ |  |

|  |  |  |
| --- | --- | --- |
| E\* | 2 | E\* |
| q2  q3  q4  q5 | q2  ∅  ∅  ∅ | q2  q3  q4  q5 |

|  |  |  |
| --- | --- | --- |
| E\* | 3 | E\* |
| q2  q3  q4  q5 | ∅  q3  ∅  ∅ | q3  q4  q5 |

|  |  |  |
| --- | --- | --- |
| E\* | 4 | E\* |
| q2  q3  q4  q5 | ∅  ∅  q4  ∅ | q4  q5 |

|  |  |  |
| --- | --- | --- |
| E\* | 5 | E\* |
| q2  q3  q4  q5 | ∅  ∅  ∅  q5 | q5 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DFA | 1 | 2 | 3 | 4 | 5 |
| q0 | A | B | C | D | E |
| A | A | B | C | D | E |
| B | F | B | C | D | E |
| C | F | F | C | D | E |
| D | F | F | F | D | E |
| E | F | F | F | F | E |

A = {q1, q2, q3, q4, q5} F= Dead state

B = { q2, q3, q4, q5}

C = { q3, q4, q5}

D = { q4, q5}

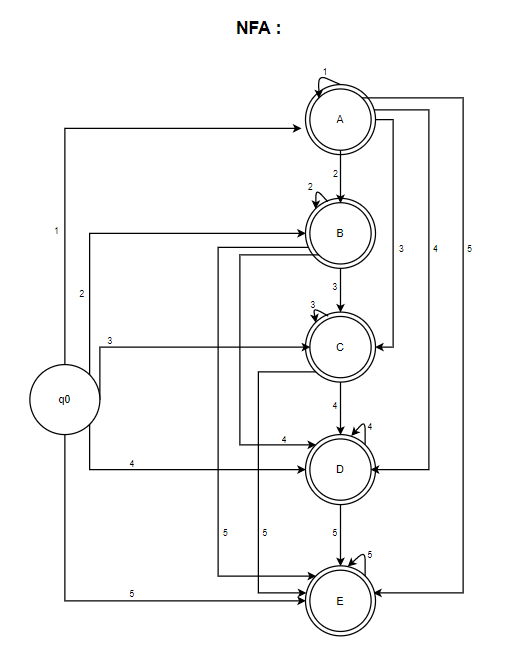
E = { q5}



# **Regular Expression**

In the regular expression we will look at the start and accept state and then we will construct a new state and accept state, then we will remove a single state on by one, every time we remove a single state we combine them again to proceed with the next state and to achieve the final result

**Step 1: Firstly we will identify the start state & the accept state**



**Step 2: We will construct the new start state and the accept state to confirm**

A diagram of a diagram

Description automatically generated

**Step 3: We will need to Delete the q0 state and then combine them to proceed**

**A diagram of a machine

Description automatically generated**

**Step 4: We will need to Delete the A state and then combine them to proceed**

A diagram of a machine

Description automatically generated

**Step 5: then We will Delete the B state and combine them to procced**

A diagram of a diagram

Description automatically generated

**Step 6: Then we will remove C and combine them again to proceed**

A diagram of a diagram

Description automatically generated

**Step 7: We will Delete the D state and combine them again**

A diagram of a loop

Description automatically generated

**Step 8: Lastly, we will Delete the E state to have the last expression**

A white rectangular object with black text

Description automatically generated

# **REFERNCES:**

Automata Conversion from NFA to DFA - Javatpoint. (2021). Retrieved November 24, 2023, from www.javatpoint.com website: https://www.javatpoint.com/automata-conversion-from-nfa-to-dfa

‌