Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №1**

Конечные автоматы

тема

Преподаватель \_\_\_\_\_\_\_\_\_ А. С. Кузнецов

подпись, дата инициалы, фамилия

Студент КИ18-17/1б 031830504 \_\_\_\_\_\_\_\_\_ Е.В. Железкин

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2021

# Цель работы

Необходимо построить ДКА и НКА в системе JFLAP и произвести программную реализацию. В коде программы обязательно наличие сущностей и процедур, относящихся к табличному представлению автомата.

Использование функций обработки строковых данных запрещено. Результат работы, выдаваемый программой на экран, внешне должен быть схож, а фактически эквивалентен результату, выдаваемому JFLAP на тех же тестовых цепочках.

# Задача работы

В каждом варианте задания в части,

а) задается цепочка или набор цепочек для распознавания ДКА. В части

б) задается цепочка или набор цепочек для распознавания НКА.

*Вариант 7*

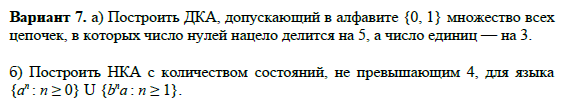


Рисунок 1 - Задание

# Ход работы

1. Была составлена таблица состояний (множество всех состояний Q) с использованием условия «остаток от деления количества единиц на 3 и нулей на 5»:



Рисунок 2 – Таблица состояний

Где первый столбец отвечает за остаток от деления количества единиц в данном состоянии, а второй столбец остаток от деления кол-ва нулей в данном состоянии.

По условию, входное состояние q0 у нас (0, 0);

По условию, в качестве выходного состояния (F) нас устраивает только (0,0);

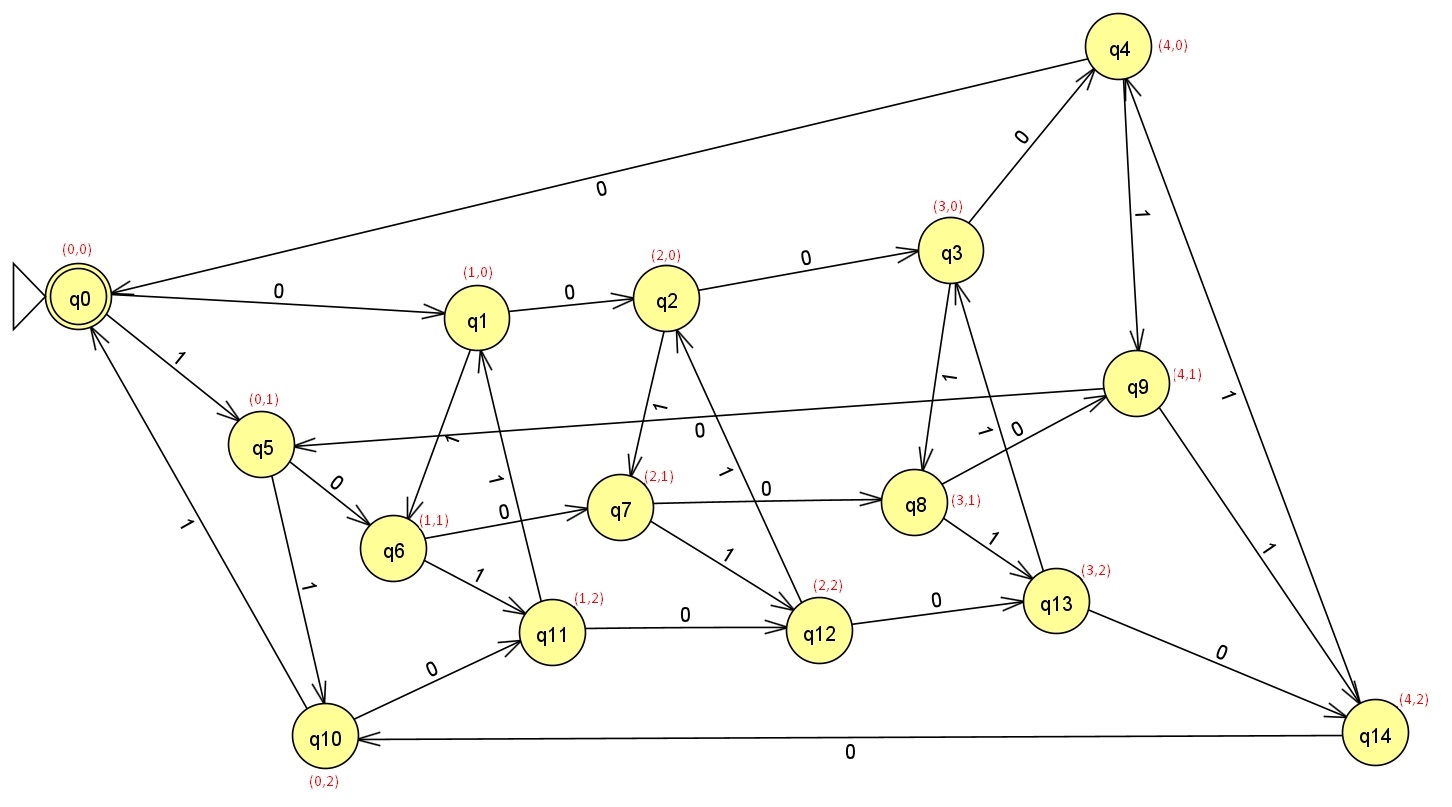


Рисунок 3 – Диаграмма переходов ДКА (у вершин подписаны их состояния из таблицы)

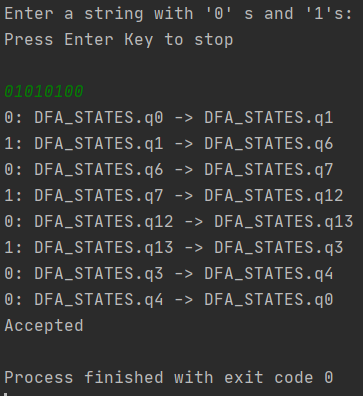


Рисунок 4 – Пример выполнения ДКА 1

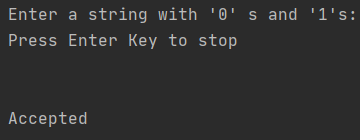


Рисунок 5 – Пример выполнения ДКА 2

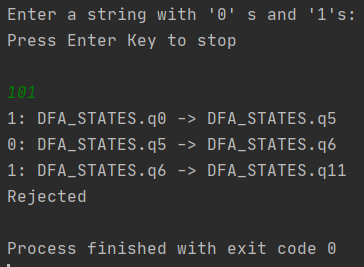


Рисунок 6 – Пример выполнения ДКА 3

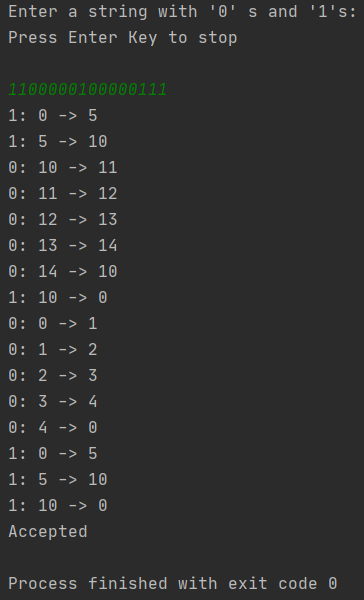


Рисунок 7 - Пример выполнения ДКА 4

1. Для выполнения данного задания была составлена диаграмма переходов в соответствии с языком, описанным в задании:

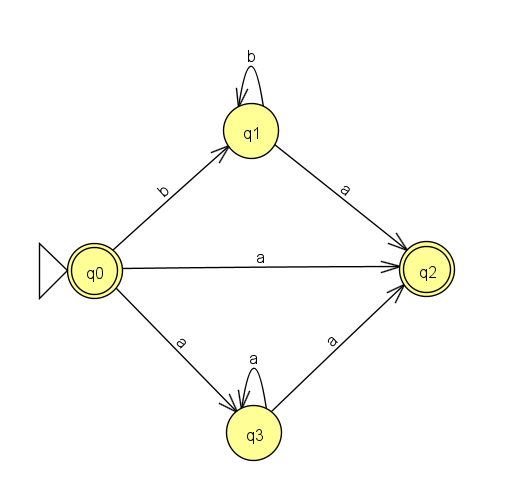


Рисунок 8 – Диаграмма переходов НКА

Далее была составлена таблица функции переходов:

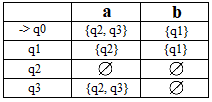


Рисунок 9 – Таблица функции переходов

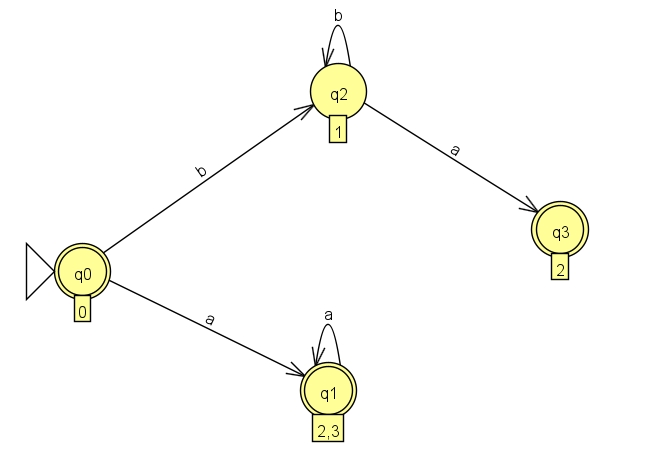


Рисунок 10 – НКА, конвертированный в ДКА с помощью средств JFLAP

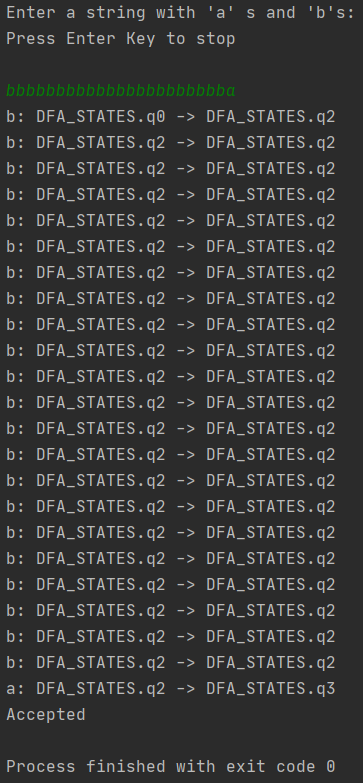


Рисунок 11 - Пример выполнения НКА 1

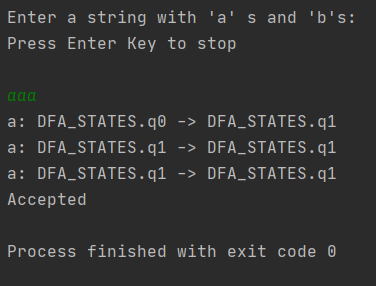


Рисунок 12 - Пример выполнения НКА 2

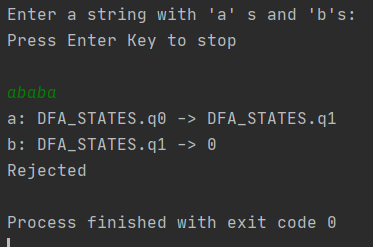


Рисунок 13 - Пример выполнения НКА 3

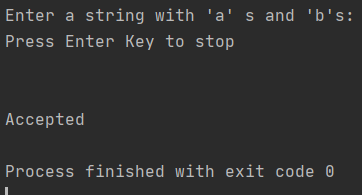


Рисунок 14 - Пример выполнения НКА 4

# Вывод

В ходе данной лабораторной работы были изучены и реализованы в систему JFLAP, а также программно детерминированные и недетерминированные конечные автоматы.

**ПРИЛОЖЕНИЕ А**

Листинг 1 – файл PyDFA.py

import sys  
from enum import Enum  
  
TOTAL\_STATES = 15  
FINAL\_STATES = 1  
ALPHABET\_CHARCTERS = 2  
  
UNKNOWN\_SYMBOL\_ERR = 0  
NOT\_REACHED\_FINAL\_STATE = 1  
REACHED\_FINAL\_STATE = 2  
  
  
class DFA\_STATES(Enum):  
 q0 = 0  
 q1 = 1  
 q2 = 2  
 q3 = 3  
 q4 = 4  
 q5 = 5  
 q6 = 6  
 q7 = 7  
 q8 = 8  
 q9 = 9  
 q10 = 10  
 q11 = 11  
 q12 = 12  
 q13 = 13  
 q14 = 14  
  
  
class Input(Enum):  
 zero = 0  
 one = 1  
  
  
gAcceptedStates = [DFA\_STATES.q0]  
gAlphabet = (**'0'**, **'1'**)  
gTransitionTable = [[0] \* ALPHABET\_CHARCTERS for i in range(TOTAL\_STATES)]  
gCurrentState = DFA\_STATES.q0  
  
  
def setDFATransitions() -> None:  
 gTransitionTable[DFA\_STATES.q0.value][Input.zero.value] = DFA\_STATES.q1  
 gTransitionTable[DFA\_STATES.q0.value][Input.one.value] = DFA\_STATES.q5  
 gTransitionTable[DFA\_STATES.q1.value][Input.zero.value] = DFA\_STATES.q2  
 gTransitionTable[DFA\_STATES.q1.value][Input.one.value] = DFA\_STATES.q6  
 gTransitionTable[DFA\_STATES.q2.value][Input.zero.value] = DFA\_STATES.q3  
 gTransitionTable[DFA\_STATES.q2.value][Input.one.value] = DFA\_STATES.q7  
 gTransitionTable[DFA\_STATES.q3.value][Input.zero.value] = DFA\_STATES.q4  
 gTransitionTable[DFA\_STATES.q3.value][Input.one.value] = DFA\_STATES.q8  
 gTransitionTable[DFA\_STATES.q4.value][Input.zero.value] = DFA\_STATES.q0  
 gTransitionTable[DFA\_STATES.q4.value][Input.one.value] = DFA\_STATES.q9  
 gTransitionTable[DFA\_STATES.q5.value][Input.zero.value] = DFA\_STATES.q6  
 gTransitionTable[DFA\_STATES.q5.value][Input.one.value] = DFA\_STATES.q10  
 gTransitionTable[DFA\_STATES.q6.value][Input.zero.value] = DFA\_STATES.q7  
 gTransitionTable[DFA\_STATES.q6.value][Input.one.value] = DFA\_STATES.q11  
 gTransitionTable[DFA\_STATES.q7.value][Input.zero.value] = DFA\_STATES.q8  
 gTransitionTable[DFA\_STATES.q7.value][Input.one.value] = DFA\_STATES.q12  
 gTransitionTable[DFA\_STATES.q8.value][Input.zero.value] = DFA\_STATES.q9  
 gTransitionTable[DFA\_STATES.q8.value][Input.one.value] = DFA\_STATES.q13  
 gTransitionTable[DFA\_STATES.q9.value][Input.zero.value] = DFA\_STATES.q5  
 gTransitionTable[DFA\_STATES.q9.value][Input.one.value] = DFA\_STATES.q14  
 gTransitionTable[DFA\_STATES.q10.value][Input.zero.value] = DFA\_STATES.q11  
 gTransitionTable[DFA\_STATES.q10.value][Input.one.value] = DFA\_STATES.q0  
 gTransitionTable[DFA\_STATES.q11.value][Input.zero.value] = DFA\_STATES.q12  
 gTransitionTable[DFA\_STATES.q11.value][Input.one.value] = DFA\_STATES.q1  
 gTransitionTable[DFA\_STATES.q12.value][Input.zero.value] = DFA\_STATES.q13  
 gTransitionTable[DFA\_STATES.q12.value][Input.one.value] = DFA\_STATES.q2  
 gTransitionTable[DFA\_STATES.q13.value][Input.zero.value] = DFA\_STATES.q14  
 gTransitionTable[DFA\_STATES.q13.value][Input.one.value] = DFA\_STATES.q3  
 gTransitionTable[DFA\_STATES.q14.value][Input.zero.value] = DFA\_STATES.q10  
 gTransitionTable[DFA\_STATES.q14.value][Input.one.value] = DFA\_STATES.q4  
  
  
def dfa(current\_symbol: chr) -> int:  
 global gAcceptedStates  
 global gCurrentState  
 global gTransitionTable  
  
 if current\_symbol not in gAlphabet:  
 return UNKNOWN\_SYMBOL\_ERR  
  
 old\_state = gCurrentState  
 gCurrentState = gTransitionTable[gCurrentState.value][gAlphabet.index(current\_symbol)]  
 print(str(old\_state) + **" -> "** + str(gCurrentState)) *# route info* if gCurrentState in gAcceptedStates:  
 return REACHED\_FINAL\_STATE  
  
 return NOT\_REACHED\_FINAL\_STATE  
  
  
if \_\_name\_\_ == **"\_\_main\_\_"**:  
 result = RuntimeError  
  
 setDFATransitions()  
  
 print(**"Enter a string with '0' s and '1's:**\n**Press Enter Key to stop**\n**"**)  
  
 symbol = sys.stdin.read(1)  
  
 while symbol != **'**\n**'**:  
 result = dfa(symbol)  
  
 symbol = sys.stdin.read(1)  
  
 if REACHED\_FINAL\_STATE == result:  
 print(**'Accepted'**)  
  
 if NOT\_REACHED\_FINAL\_STATE == result:  
 print(**'Rejected'**)

Листинг 2 – файл PyNFA.py

import sys  
from enum import Enum  
  
TOTAL\_STATES = 3  
FINAL\_STATES = 1  
ALPHABET\_CHARCTERS = 2  
  
UNKNOWN\_SYMBOL\_ERR = 0  
NOT\_REACHED\_FINAL\_STATE = 1  
REACHED\_FINAL\_STATE = 2  
  
  
class DFA\_STATES(Enum):  
 q0 = 0  
 q1 = 1  
 q2 = 2  
 q3 = 3  
  
  
class Input(Enum):  
 a = 0  
 b = 1  
  
  
gAcceptedStates = [DFA\_STATES.q0, DFA\_STATES.q1, DFA\_STATES.q3]  
gAlphabet = (**'a'**, **'b'**)  
gTransitionTable = [[0] \* ALPHABET\_CHARCTERS for i in range(TOTAL\_STATES)]  
gCurrentState = DFA\_STATES.q0  
  
  
def setDFATransitions() -> None:  
 gTransitionTable[DFA\_STATES.q0.value][Input.a.value] = DFA\_STATES.q1  
 gTransitionTable[DFA\_STATES.q0.value][Input.b.value] = DFA\_STATES.q2  
 gTransitionTable[DFA\_STATES.q1.value][Input.a.value] = DFA\_STATES.q1  
 gTransitionTable[DFA\_STATES.q2.value][Input.a.value] = DFA\_STATES.q3  
 gTransitionTable[DFA\_STATES.q2.value][Input.b.value] = DFA\_STATES.q2  
  
  
def dfa(current\_symbol: chr) -> int:  
 global gAcceptedStates  
 global gCurrentState  
 global gTransitionTable  
  
 if current\_symbol not in gAlphabet:  
 return UNKNOWN\_SYMBOL\_ERR  
  
 old\_state = gCurrentState  
 try:  
 gCurrentState = gTransitionTable[gCurrentState.value][gAlphabet.index(current\_symbol)]  
 except Exception as e:  
 return NOT\_REACHED\_FINAL\_STATE  
  
 print(current\_symbol + **": "** + str(old\_state) + **" -> "** + str(gCurrentState)) *# route info* if gCurrentState in gAcceptedStates:  
 return REACHED\_FINAL\_STATE  
  
 return NOT\_REACHED\_FINAL\_STATE  
  
  
if \_\_name\_\_ == **"\_\_main\_\_"**:  
 result = REACHED\_FINAL\_STATE  
  
 setDFATransitions()  
  
 print(**"Enter a string with 'a' s and 'b's:**\n**Press Enter Key to stop**\n**"**)  
  
 symbol = sys.stdin.read(1)  
  
 while symbol != **'**\n**'**:  
 result = dfa(symbol)  
  
 if UNKNOWN\_SYMBOL\_ERR == result:  
 print(**'Unknown symbol error'**)  
 break  
  
 symbol = sys.stdin.read(1)  
  
 if REACHED\_FINAL\_STATE == result:  
 print(**'Accepted'**)  
  
 if NOT\_REACHED\_FINAL\_STATE == result:  
 print(**'Rejected'**)