Министерство образования и науки Кыргызской Республики

Кыргызско-Германский институт прикладной информатики

Лабораторная работа №4

по теме «Выполнение арифметических операций на машине Тьюринга»

Выполнила: студентка группы AIH-1-22

Тахирова К.

Проверила:

Доцент КГИПИ

Кошоева Б. Б.

Оглавление

[1 Цель работы 3](#_Toc185199451)

[2 Постановка задачи 3](#_Toc185199452)

[3 Совокупность команд 3](#_Toc185199453)

[4 Таблица соответствия 4](#_Toc185199454)

[5 Граф 4](#_Toc185199455)

[6 Исходный код 4](#_Toc185199456)

# Цель работы

Целью данной лабораторной работы является разработка и реализация машины Тьюринга для выполнения арифметических операций, а также исследование принципов работы данной машины. Изучение способов представления машины Тьюринга, а именно:

* совокупностью команд;
* в виде графа;
* в виде таблицы соответствия.

# Постановка задачи

|  |  |
| --- | --- |
| № | Задание |
| 12 | Дано десятичное натуральное число H. Разработать МТ, которая уменьшала бы это число на 1. В состоянии q1 обозревается некая цифра числа. |

* Расписать совокупности команд.
* Построить таблицу соответствия заданного автомата.
* Реализовать МТ на одном из языков программирования.

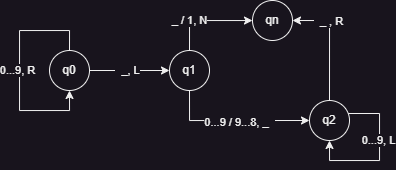
# Совокупность команд

|  |  |  |
| --- | --- | --- |
| Состояние q0: | Состояние q1: | Состояние q2: |
| q0, 0 → q0, 0, R  q0, 1 → q0, 1, R  q0, 2 → q0, 2, R  q0, 3 → q0, 3, R  q0, 4 → q0, 4, R  q0, 5 → q0, 5, R  q0, 6 → q0, 6, R  q0, 7 → q0, 7, R  q0, 8 → q0, 8, R  q0, 9 → q0, 9, R  q0, \_ → q0, \_, L | q1, 1 → q1, 0, L  q1, 0 → q1, 9, L  q1, \_ → q1, \_, R | q2, \_ → q2, \_, H  q2, 0 → q2, \_, H |

# Таблица соответствия

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Состояние  /Символ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | \_ |
| q0 | R | R | R | R | R | R | R | R | R | R | \_  L  q1 |
| q1 | 9  L  q1 | 0  N  q2 | 1  N  q2 | 2  N  q2 | 3  N  q2 | 4  N  q2 | 5  N  q2 | 6  N  q2 | 7  N  q2 | 8  N  q2 | 1  H |
| q2 | L | L | L | L | L | L | L | L | L | L | \_  R |

# Граф



# Исходный код

public class Transition {  
 String currentState;  
 char readSymbol;  
 char writeSymbol;  
 char moveDirection; // 'L' - влево, 'R' - вправо  
 String nextState;  
  
 Transition(String currentState, char readSymbol, char writeSymbol, char moveDirection, String nextState) {  
 this.currentState = currentState;  
 this.readSymbol = readSymbol;  
 this.writeSymbol = writeSymbol;  
 this.moveDirection = moveDirection;  
 this.nextState = nextState;  
 }  
}  
  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Scanner;  
  
public class TMService {  
  
 private List<String[]> table = new ArrayList<>();  
  
  
 public void tmWork(char[] tape, List<Transition> transitions){  
  
 int headPosition = tape.length - 2; // Указатель на последнюю цифру  
 String currentState = "q0";  
 while (!currentState.equals("halt")) {  
 char currentSymbol = tape[headPosition];  
 boolean transitionFound = false;  
  
 for (Transition transition : transitions) {  
 if (transition.currentState.equals(currentState) && transition.readSymbol == currentSymbol) {  
 // Выполнение перехода  
 table.add(new String[]{currentState, String.*valueOf*(currentSymbol), String.*valueOf*(transition.writeSymbol), String.*valueOf*(transition.moveDirection), transition.nextState});  
 tape[headPosition] = transition.writeSymbol; // Запись символа  
 headPosition += (transition.moveDirection == 'R' ? 1 : -1); // Движение указателя  
 currentState = transition.nextState; // Переход в следующее состояние  
 transitionFound = true;  
 break;  
 }  
 }  
  
 if (!transitionFound) {  
 System.*out*.println("Ошибка: переход не найден!");  
 return;  
 }  
 }  
 }  
  
 public void printResult(char[] tape){  
 System.*out*.println("Результат работы машины Тьюринга:");  
 for (char cell : tape) {  
 if (cell != '\_') { // Игнорирование символов границы  
 System.*out*.print(cell);  
 }  
 }  
 }  
  
 public void printTable(){  
  
 System.*out*.println("\nТаблица переходов:");  
 for (String[] row : table) {  
 System.*out*.printf("%-20s %-20s %-20s %-20s %-20s%n", row[0], row[1], row[2], row[3], row[4]);  
 }  
 }  
  
 private List<Transition> transitionList(){  
 List<Transition> transitions = new ArrayList<>();  
 // Переходы для состояния q0 (уменьшение)  
 transitions.add(new Transition("q0", '0', '9', 'L', "q1"));  
 transitions.add(new Transition("q0", '1', '\_', 'H', "halt")); // Удаление последней единицы  
 transitions.add(new Transition("q0", '2', '1', 'H', "halt"));  
 transitions.add(new Transition("q0", '3', '2', 'H', "halt"));  
 transitions.add(new Transition("q0", '4', '3', 'H', "halt"));  
 transitions.add(new Transition("q0", '5', '4', 'H', "halt"));  
 transitions.add(new Transition("q0", '6', '5', 'H', "halt"));  
 transitions.add(new Transition("q0", '7', '6', 'H', "halt"));  
 transitions.add(new Transition("q0", '8', '7', 'H', "halt"));  
 transitions.add(new Transition("q0", '9', '8', 'H', "halt"));  
  
 // Переходы для состояния q1 (заимствование)  
 transitions.add(new Transition("q1", '1', '0', 'L', "q1"));  
 transitions.add(new Transition("q1", '0', '9', 'L', "q1"));  
 transitions.add(new Transition("q1", '\_', '\_', 'R', "q2"));  
  
 // Переходы для состояния q2 (завершение)  
 transitions.add(new Transition("q2", '\_', '\_', 'H', "halt")); // Завершение на символе '\_'  
 transitions.add(new Transition("q2", '0', '\_', 'H', "halt")); // Удаление ведущих нулей  
 return transitions;  
 }  
  
 public void start(){  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Введите натуральное число для уменьшения: ");  
 String input = scanner.nextLine();  
  
 if (!input.matches("\\d+") || input.equals("0")) {  
 System.*out*.println("Ошибка: введите натуральное число больше нуля.");  
 System.*exit*(0);  
 }  
 char[] tape = ("\_" + input + "\_").toCharArray();  
 List<Transition> transitions = transitionList();  
 table.add(new String[]{"Текущее состояние", "Символ на ленте", "Новый символ", "Направление", "Следующее состояние"});  
 tmWork(tape, transitions);  
 printResult(tape);  
 printTable();  
 }  
}

public class Main{  
 public static void main(String[] args) {  
  
 TMService service = new TMService();  
 service.start();  
 }  
}