**Звіт**

**до лабораторної роботи №2**

Оформив студент групи ТТП-31

Казмірчук Володимир

Викладач: Шишацький Андрій

Постановка задачі

Автомат A на вході програми (та на виході, де потрібно) подається у вигляді текстового файлу наступної структури:

||A|| ||S|| s0 ||F||

FS1 ∈ F … fS||F|| ∈ F

s a s’ // всі такі трійки, що (s, a, s’) ∈ f, через проміжок, по одній на рядок – до кінця файлу.

**Встановити, які літери вхідного алфавіту не сприймає скінчений детермінований автомат.**

Розв’язок

Для вирішення цієї задачі було написано наступний код на мові C++:

#include <iostream>

#include <fstream>

#include <list>

using namespace std;

// transition structure

struct transition {

    int from;

    int to;

    char c;

};

int main () {

    // open file

    ifstream fin;

    while (true) {

        cout << "Enter file name: ";

        string fname;

        cin >> fname;

        fin.open (fname);

        if (fin) {

            cout << "The file was opened successfully" << endl;

            break;

        }

        else { cout << "Couldn't open the file!" << endl; }

    }

    // read data

    int alfsize, statesize, first, fstatenum;

    fin >> alfsize >> statesize >> first >> fstatenum;

    int final [fstatenum];

    for (int i = 0; i < fstatenum; i ++) { fin >> final [i]; }

    list < transition > transitions;

    transition t;

    while (fin >> t.from >> t.c >> t.to) {

        transitions.push\_back (t);

    }

    fin.close ();

    // perform BFS on graph, setting 0 if unreached, 1 if deadend, and 2 if reached

    char states [statesize] = {0};

    list < int > queue;

    queue.push\_back (first);

    while (!queue.empty ()) {

        int state = queue.front ();

        queue.pop\_front ();

        if (states [state] != 0) { continue; }

        states [state] = 1;

        for (auto it = transitions.begin (); it != transitions.end (); it ++) {

            if (it -> from == state) {

                states [state] = 2;

                queue.push\_back (it -> to);

                it = transitions.erase (it);

                it --;

            }

        }

    }

    // print lists of unreachable and deadend states

    cout << "Unreachable states: ";

    for (int i = 0; i < statesize; i ++) {

        if (states [i] == 0) { cout << i << ' ';}

    }

    cout << endl;

    cout << "Deadend states: ";

    for (int i = 0; i < statesize; i ++) {

        if (states [i] == 1) {

            bool found = false;

            for (int j = 0; j < fstatenum && !found; j ++) {

                if (final [j] == i) { found = true; }

            }

            if (!found) { cout << i << ' '; }

        }

    }

    cout << endl;

    return 0;

}

Даний код виконує аналіз графу, представленого автоматом зі станами та переходами між ними. Програма шукає недосяжні та тупикові стани в автоматі.

Основні етапи програми:

1. Відкриття файлу:

Користувачу пропонується ввести ім'я файлу. Після цього програма намагається відкрити вказаний файл. Якщо відкриття вдається, виводиться повідомлення "Файл успішно відкрито". Якщо ні, виводиться повідомлення про невдале відкриття.

2. Читання даних:

Програма зчитує з файлу параметри автомата, такі як розмір алфавіту, кількість станів, початковий стан, кількість заключних станів та їх перелік, та переходи між станами. Зчитані дані зберігаються у відповідних змінних.

3. Пошук недосяжних та тупикових станів:

Використовуючи алгоритм BFS (Breadth-First Search) для обходу графу програма позначає недосяжні стани як 0, тупикові як 1 - якщо тупикові, та 2 якщо досяжні і не тупикові.

4. Вивід результатів:

Виводяться списки недосяжних та тупикових станів у вигляді "Unreachable states" та "Deadend states" відповідно.

Загалом, програма служить для виявлення та виведення недосяжних та тупикових станів у графі, що представляє автомат – чого і вимагає технічне завдання.

Приклади роботи програми

Приклад №1

Вхідний файл:

26 10 0 3

7 6 9

0 a 1

0 b 2

1 c 3

2 c 3

3 d 7

2 o 9

2 i 8

5 u 6

4 a 7

6 a 7

Вивід:

Enter file name: t1.txt

The file was opened successfully

Unreachable states: 4 5 6

Deadend states: 8

Приклад №2

Вхідний файл:

26 7 0 2

3 6

0 a 1

0 b 2

2 a 3

5 b 4

4 a 3

4 b 6

Вивід:

Enter file name: t2.txt

The file was opened successfully

Unreachable states: 4 5 6

Deadend states: 1

Ілюстрація для другого прикладу:

