Отчет по лабораторной работе № 11 по курсу “Фундаментальная информатика”

Студент группы М80-109Б-22 Федоров Алексей Алексеевич, № 20
Контакты:
Email: hedgefog@yandex.ru, Telegram: @hedgefo9

Работа выполнена: «08» декабря 2022г.

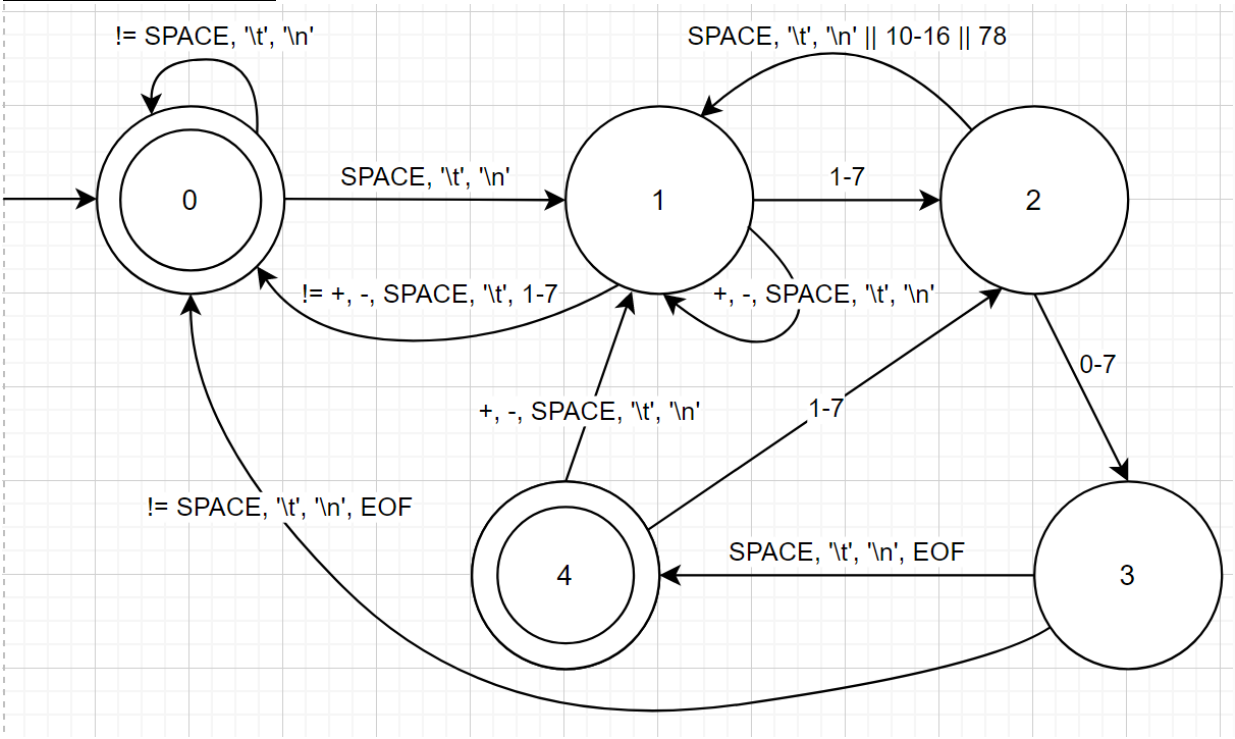
Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан « » _____20__ г., итоговая оценка _____

Подпись преподавателя

_____

1. **Тема:** Обработка последовательности литер входного текстового файла. Простейшие приёмы лексического анализа. Диаграммы состояний и переходов.
2. **Цель работы:** Научиться обрабатывать последовательности литер входного текстового текста.
3. **Задание (Вариант №33)**
   Выделить все восьмеричные числа от 17 до 77 по модулю и распечатать их значения в словесной форме по-английски.
4. **Оборудование (студента):**
   Процессор *Ryzen 5 3500U @ 8x 2.1GH* с ОП *14900* Мб, НМД *677* Гб. Монитор *2160x1440*
5. **Программное обеспечение (студента):**
   Операционная система семейства: *linux*, наименование: *ubuntu*, версия *18.10 cinnamon* интерпретатор команд: *bash* версия *4.4.19*.
   Система программирования -- версия --, редактор текстов *emacs* версия *25.2.2*
6. **Идея, метод, алгоритм**



7. **Сценарий выполнения работы**
   - Изучить лекционные материалы
   - Составить диаграмму переходов состояний
   - Написать программу, удовлетворяющую условию
   - Провести unit-тестирование
   - Написать отчёт
8. **Распечатка протокола:**

Код на языке C

```c
#include <stdio.h>
#include <assert.h>

typedef enum {
  state_zero,
  state_first,
  state_second,
  state_third,
  state_fourth
} state;

int is_sp(char c) {
    return (c == ' ');
}

int is_comma(char c) {
    return (c == ',');
}

int is_tab(char c) {
    return (c == '\t');
}

int is_newline(char c) {
    return (c == '\n');
}

int is_it_num_8(char c) {
    return (c >= '0' && c <= '7');
}

int is_pm(char c) {
    return (c == '+' || c == '-');
}

int num2word(char a, char b) {
    if (is_it_num_8(a) && is_it_num_8(b)) {
        switch (a) {
            case '1':printf("seven");
                break;
            case '2':printf("twenty ");
                break;
            case '3':printf("thirty ");
                break;
            case '4':printf("forty ");
                break;
            case '5':printf("fifty ");
                break;
            case '6':printf("sixty ");
                break;
            case '7':printf("seventy ");
                break;
            default:return 0;
                break;
        }
        switch (b) {
            case '0':printf("\n");
                break;
            case '1':printf("one\n");
                break;
            case '2':printf("two\n");
                break;
            case '3':printf("three\n");
                break;
            case '4':printf("four\n");
                break;
            case '5':printf("five\n");
                break;
            case '6':printf("six\n");
```

```
                break;
            case '7':
                if (a == '1') {
                    printf("teen\n");
                } else {
                    printf("seven\n");
                }
                break;
            default:return 0;
                break;
        }
        return 1;
    } else {
        return 0;
    }
}

void test_is_sp() {
    assert(is_sp(' ') == 1);
    assert(is_sp('d') == 0);
    assert(is_sp(EOF) == 0);
}

void test_is_comma() {
    assert(is_comma(',') == 1);
    assert(is_comma('d') == 0);
    assert(is_comma(EOF) == 0);
}

void test_is_tab() {
    assert(is_tab('\t') == 1);
    assert(is_tab('d') == 0);
    assert(is_tab(EOF) == 0);
}

void test_is_newline() {
    assert(is_newline('\n') == 1);
    assert(is_newline('\t') == 0);
    assert(is_newline(EOF) == 0);
}

void test_is_num_8() {
    assert(is_it_num_8('0') == 1);
    assert(is_it_num_8('4') == 1);
    assert(is_it_num_8('7') == 1);
    assert(is_it_num_8('8') == 0);
    assert(is_it_num_8('\t') == 0);
    assert(is_it_num_8(EOF) == 0);
}

void test_is_pm() {
    assert(is_pm('+') == 1);
    assert(is_pm('-') == 1);
    assert(is_pm('d') == 0);
    assert(is_pm(EOF) == 0);
}

void test_num2word() {
    assert(num2word('4', '0') == 1);
    assert(num2word('d', '7') == 0);
    assert(num2word('7', '3') == 1);
}

void test_all() {
    test_is_sp();
    test_is_comma();
    test_is_tab();
    test_is_num_8();
    test_is_newline();
```

```
    test_is_pm();
}
int main() {
    test_all();

    state current_state = state_zero;
    char c = ' ', a = ' ', b = ' ';
    int current_sign = 0;

    while ((c = getchar()) != EOF) {

        switch (current_state) {
            case state_zero:
                if (is_sp(c) || is_newline(c) || is_tab(c) || is_comma(c)) {
                    current_state = state_first;
                }
                break;

            case state_first:
                if (is_it_num_8(c) && c != '0' && c != '8') {
                    a = c;
                    current_state = state_second;
                    current_sign = 0;

                } else if (is_it_num_8(c) && (c == '0' || c == '8')) {
                    current_state = state_zero;
                    current_sign = 0;
                } else if (is_pm(c) && current_sign == 0) {
                    current_sign = 1;
                } else if (is_pm(c) && current_sign == 1) {
                    current_state = state_zero;
                    current_sign = 0;
                } else if (is_sp(c) || is_newline(c) || is_tab(c) || is_comma(c)) {
                    current_sign = 0;
                } else {
                    current_state = state_zero;
                    current_sign = 0;
                }
                break;

            case state_second:
                if (is_it_num_8(c) && !((a <= '1') && (c < '7')) && !((a == '7') && (c >
'7'))) {
                    b = c;
                    current_state = state_third;
                } else {
                    current_state = state_first;
                }
                break;

            case state_third:
                if (is_sp(c) || is_newline(c) || is_tab(c) || is_comma(c)) {
                    current_state = state_fourth;
                    num2word(a, b);
                    a = ' ';
                    b = ' ';
                } else
                    current_state = state_zero;
                break;

            case state_fourth:
                if (is_it_num_8(c) && (c != '0') && (c != '8')) {
                    a = c;
                    current_state = state_second;
                    current_sign = 0;

                } else if (is_it_num_8(c) && (c == '0' || c == '7')) {
                    current_state = state_zero;
                    current_sign = 0;
```

```
                } else if (is_pm(c) && current_sign == 0) {
                    current_sign = 1;
                    current_state = state_first;
                } else if (is_pm(c) && current_sign == 1) {
                    current_state = state_zero;
                    current_sign = 0;
                } else if (is_sp(c) || is_newline(c) || is_tab(c) || is_comma(c)) {
                    current_sign = 0;
                    current_state = state_first;
                } else {
                    current_sign = 0;
                    current_state = state_zero;
                }
                break;

        }
    }
    if (current_state == state_third) {
        num2word(a, b);
        current_state = state_fourth;
    }
    return 0;
}
```

9. **Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. **Замечания автора**

Нет замечаний

11. **Выводы**

Благодаря этой лабораторной работе я потренировался работать с конечным автоматами и их реализацией в Си. Мне было интересно, я узнал новые методы, которые возможно пригодятся мне в будущем на работе, т. к. конечные автоматы тесно связаны с регулярными выражениями, которые в свою очередь нужны для поиска и осуществления действий с подстроками в тексте.

Подпись студента _____