Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Теория автоматов**

(название дисциплины)

**Лабораторная (практическая) работа №1**

**Автоматная реализация алгоритма разбора**

(название (тема) работы)

**Вариант №8**

Выполнил

Студент группы ИВТАСбд-21

Зюзин Г.А.

(Фамилия И.О.)

Проверил(а):

ст. преподаватель кафедры «ВТ»\_\_\_\_

(должность)

\_\_\_\_\_\_\_\_\_Лылова А.В.\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия И.О.)

Ульяновск

2023

# Постановка задачи

Реализовать автомат Мура, осуществляющий преобразование цикла repeat в do {} while(). Допускается обработка только одного цикла в случае множественных или вложенных циклов во входных данных.

# Текстовое описание реализованного автомата разбора

В ходе работы программа два раза проходит по файлу:

В исходном файле находятся элементы цикла repeat: инициализация индекса цикла, выражение для проверки критериев завершения цикла, выражение приращения индексов. При нахождении последнего элемента обработка файла прерывается. После получения индексов начала и конца цикла выписывается тело функции, которое будет продублирован и записан в цикле.

# Блок схема

# Кодирование входных сигналов

|  |  |
| --- | --- |
| X1 | line in fr |
| X2 | "repeat" in line |
| X3 | "until" in line |
| X4 | len(index\_repeat) == 0 |
| X5 | i in text |
| X6 | i in range (index\_repeat[0] + 1, index\_until[len(index\_until) - 1]) |
| X7 | i in range(index\_repeat[0]) |
| X8 | i in range(len(body\_of\_cycle)) |
| X9 | i in range(index\_repeat[0], len(text) |
| X10 | I in range(new\_code) |
| X11 | 'until' in new\_code[i] and i = index\_repeat[0] + len(new\_code) - len(text) |
| X12 | 'repeat' in new\_code[i] and i = index\_until[len(index\_until) - 1] + len(new\_code) - len(text) |
| X13 | i in new\_code |

# Кодирование выходных сигналов

|  |  |
| --- | --- |
| Y1 | text = [] |
| Y2 | index\_repeat = [] |
| Y3 | index\_until = [] |
| Y4 | fr = open(“source.txt”, “r”) |
| Y5 | text.append(line) |
| Y6 | index\_repeat.append(text.index(line)) |
| Y7 | Index\_until.append(text.index(line)) |
| Y8 | fr.close() |
| Y9 | fw = open(“out.txt”, “w”) |
| Y10 | fw.write(i) |
| Y11 | body\_of\_cycle = [] |
| Y12 | s = text[i][1:] |
| Y13 | text[index\_repeat[0], text[index\_until[len(index\_until)-1] = text[index\_until[len(index\_until)-1], text[index\_repeat[0]] |
| Y14 | new\_code = [] |
| Y15 | new\_code.append(text[i]) |
| Y16 | new\_code.append(body\_of\_cycle[i]) |
| Y17 | new\_code[i] = new\_code.replace('until', 'while not(') |
| Y18 | new\_code[i] = new\_code.replace(';', 'do begin') |
| Y19 | new\_code[i] = new\_code.replace('repeat', 'end;') |
| Y20 | body\_of\_cycle.append(s) |

# Использование входных и выходных сигналов в блок схеме, расстановка состояний автомата Мура

# Граф с указанными переходами и выходами

# Тестирование программы

Тест №1

Входной файл:

var i, N:

integer;

begin

i := 1;

readln(N);

repeat

write(i);

i := i + 1;

until i = N + 1;

end.

Выходной файл:

var i, N:

integer;

begin

i := 1;

readln(N);

write(i);

i := i + 1;

while not(i = N + 1) do begin

write(i);

i := i + 1;

end;

end.

Вывод состояний:

S0 S1 S2 S2 S2 S2 S2 S2 S2 S2 S3 S2 S2 S2 S4 S2 S5 S8 S9 S9 S10 S11 S12 S12 S12 S12 S12 S12 S12 S13 S13 S14 S14 S14 S14 S14 S15 S16 S17 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S0

Тест №2

Входной файл:

var i, N:

integer;

begin

i := 1;

readln(N);

repeat

write(i);

i := i + 1;

repeat

write(j);

j := j + 1;

until j = N + 1;

until i = N + 1;

end.

Выходной файл:

var i, N:

integer;

begin

i := 1;

readln(N);

write(i);

i := i + 1;

repeat

write(j);

j := j + 1;

until j = N + 1;

while not(i = N + 1) do begin

write(i);

i := i + 1;

repeat

write(j);

j := j + 1;

until j = N + 1;

end;

end.

Вывод состояний:

S0 S1 S2 S2 S2 S2 S2 S2 S2 S2 S3 S2 S2 S2 S3 S2 S2 S2 S4 S2 S4 S2 S5 S8 S9 S9 S9 S9 S9 S9 S10 S11 S12 S12 S12 S12 S12 S12 S12 S13 S13 S13 S13 S13 S13 S14 S14 S14 S14 S14 S14 S14 S14 S14 S15 S16 S17 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S18 S0

Тест №3

Входной файл:

var i, N:

integer;

begin

i := 1;

readln(N);

end.

Выходной файл:

var i, N:

integer;

begin

i := 1;

readln(N);

end.

Вывод состояний: S0 S1 S2 S2 S2 S2 S2 S2 S2 S5 S6 S7 S7 S7 S7 S7 S7 S7 S0

# Листинг кода:

states = "S0"  
if \_\_name\_\_ == "\_\_main\_\_":  
 text = []  
 index\_repeat = []  
 index\_until = []  
 with open("source.txt", "r") as fr:  
 states += " S1"  
 for line in fr:  
 text.append(line)  
 states += " S2"  
 if 'repeat' in line:  
 index\_repeat.append(text.index(line))  
 states += " S3"  
 if 'until' in line:  
 index\_until.append(text.index(line))  
 states += " S4"  
 fr.close()  
 states += " S5"  
  
 if len(index\_repeat) == 0:  
 with open("out.txt", "w") as fw:  
 states += " S6"  
 for i in range(len(text)):  
 fw.write(text[i])  
 states += " S7"

states += " S0"  
 else:  
 body\_of\_cycle = []  
 states += " S8"  
 for i in range(index\_repeat[0] + 1, index\_until[len(index\_until) - 1]):  
 s = text[i][1:]  
 body\_of\_cycle.append(s)  
 states += " S9"  
  
 text[index\_repeat[0]], text[index\_until[len(index\_until) - 1]] = \  
 text[index\_until[len(index\_until) - 1]], text[index\_repeat[0]]  
 states += " S10"  
 new\_code = []  
 states += " S11"  
  
 for i in range(index\_repeat[0]):  
 new\_code.append(text[i])  
 states += " S12"  
  
 for i in range(len(body\_of\_cycle)):  
 new\_code.append(body\_of\_cycle[i])  
 states += " S13"  
  
 for i in range(index\_repeat[0], len(text)):  
 new\_code.append(text[i])  
 states += " S14"  
  
 for i in range(len(new\_code)):  
 if 'until' in new\_code[i] and i == index\_repeat[0] + len(new\_code) - len(text):  
 new\_code[i] = new\_code[i].replace('until ', 'while not(')  
 new\_code[i] = new\_code[i].replace(';', ') do begin')  
 states += " S15"  
 if 'repeat' in new\_code[i] and i == index\_until[len(index\_until) - 1] + len(new\_code) - len(text):  
 new\_code[i] = new\_code[i].replace('repeat', 'end;')  
 states += " S16"  
  
 with open("out.txt", "w") as fw:  
 states += " S17"  
 for i in new\_code:  
 fw.write(i)  
 states += " S18"  
 fw.close()  
 states += " S0"  
 print(states)