

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Ульяновский государственный технический университет»
Кафедра «Вычислительная техника»

Теория автоматов

Лабораторная работа №1
«Автоматная реализация алгоритма разбора.»

Выполнил
Студент группы ИВТбд-21
Ведин В. А.
Проверил(а):
ст. преподаватель кафедры «ВТ»
Лылова А.В.

Ульяновск
2024

ЗАДАНИЕ

Требуется реализовать автомат типа Мили для разбора и преобразования цикла `for`, на языке C/C++, в `label/goto`. Достаточно обрабатывать один цикл в случае множественных или вложенных циклов во входных данных.

ТЕКСТОВОЕ ОПИСАНИЕ РЕАЛИЗОВАННОГО АВТОМАТА РАЗБОРА

На вход программе подаётся файл “code.txt”, в котором содержится код исходной программы. Затем, файл считывается и построчно записывается в список. После программа переписывает все строчки в файл “code2.txt”, пока не встретит цикл for. Если в программе не было цикла, то программа просто закончит свою работу, когда закончит переписывать код из исходного файла. Однако, если цикл for есть в исходном коде, то начинается его разбиение на элементы (переменная, условие, инкремент), и начинает формироваться label/goto. Программа переписывает всю внутренность цикла, пока не встретит закрывающую скобку цикла (“}”). При встрече формируется условие перехода в label и сам переход goto. Далее программа проверяет есть ли что-то после цикла, если есть - переписывает. Программа не обрабатывает множественные циклы, она обрабатывает только первый встреченный ей цикл.

ТАБЛИЦЫ ОБОЗНАЧЕНИЙ

Таблица 1. Обозначение входных сигналов.

Входной сигнал	Обозначение
x1	for item in data:
x2	for line in data:
x3	if “for” in line:
x4	for idx in range(count, dataLen):
x5	if len(data) == 0:
x6	for line in data:
x7	if “{” in line:
x8	if “}” in line:
x9	if line.strip() == “}” and stack == “”:
x10	for idx in range(count, dataLen):
x11	for line in data:

Таблица 2. Обозначение выходных функций.

Выходная функция	Обозначение
y1	with open("code.txt", "r") as file:
y2	data = file.readlines()
y3	new_data = []
y4	item.strip(“\n”)
y5	new_data.extend(tem.strip(“\n”))
y6	data = new_data.copy()
y7	file.close()
y8	with open("code2.txt", "w") as file:

Выходная функция	Обозначение
y9	count = 0
y10	file.write(line + “\n”)
y11	count += 1
y12	dataLen = len(data)
y13	new_data.extend(data[idx])
y14	exit()
y15	line = data[0]
y16	data.pop(0)
y17	line = line.strip()
y18	line = line[5:-3]
y19	innerFor = line.split(“;”)
y20	cycleVar = innerFor[0].strip()
y21	cycleCondition = innerFor[1].strip()
y22	cycleIncrement = innerFor[2].strip()
y23	file.wrtie(“\t” + cycleVar + “;\n”)
y24	file.write(“label:\n”)
y25	stack = “{”
y26	stack += “{”
y27	stack = stack[:-1]
y28	lessStr = f”if ({cycleCondition} - 1)” + “ {”
y29	moreStr = f”if ({cycleCondition} + 1)” + “ {”
y30	labelCondition = lessStr if “<” in cycleCondition else moreStr

Входная функция	Обозначение
y31	<code>file.write("\t" + labelCondition + ";\n")</code>
y32	<code>file.write("\t\t" + cycleIncrement + ";\n")</code>
y33	<code>file.write("\t\t" + "goto label;" + "\n")</code>
y34	<code>file.write("\t" + "}" + "\n")</code>
y35	<code>modifiedLine = line.replace(" ", "", 4)</code>
y36	<code>file.write(modifiedLine + "\n")</code>

БЛОК-СХЕМА

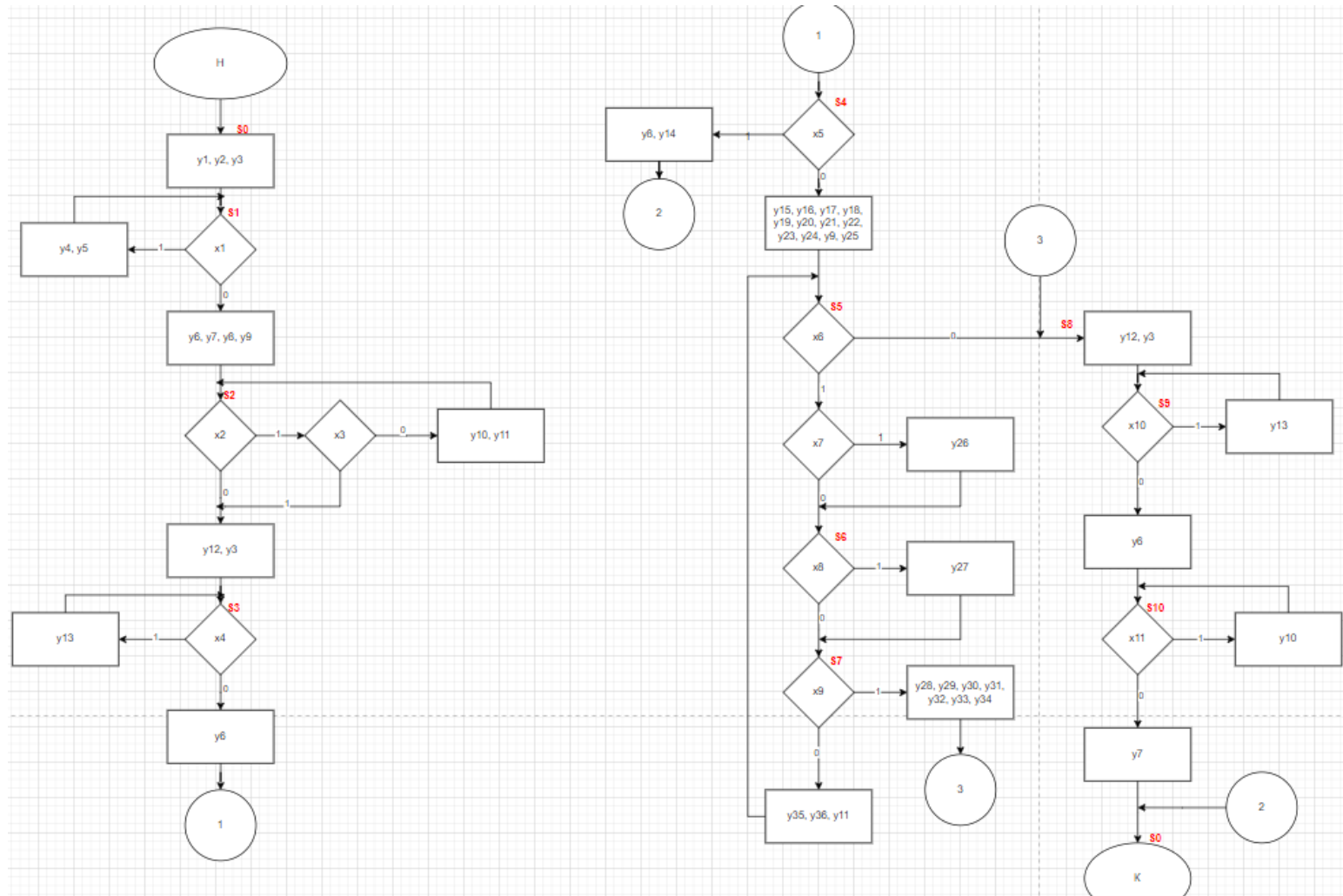


Рис 1. Блок-схема автомата типа Мили для разбора и преобразования цикла.

ГРАФ АВТОМАТА

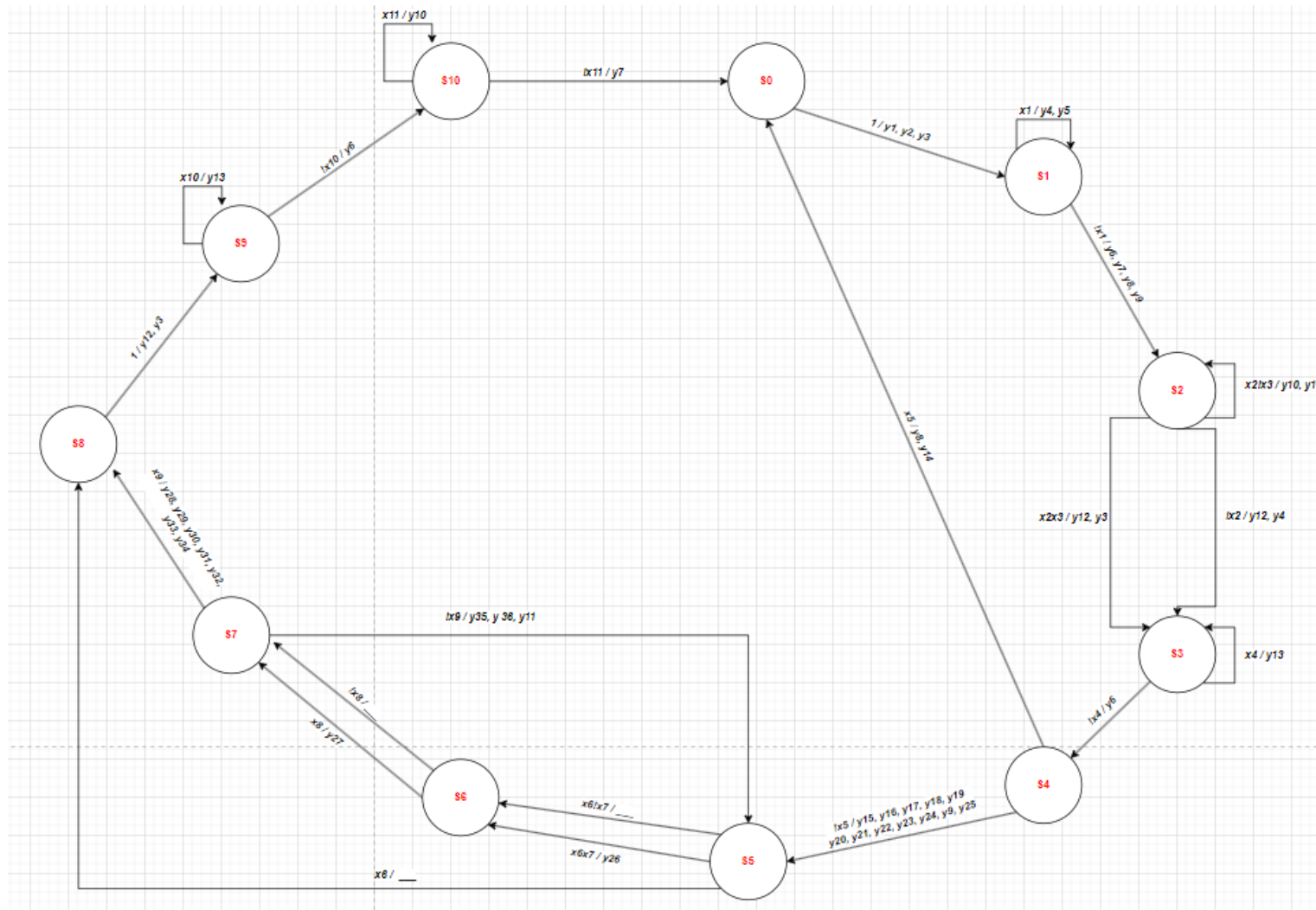
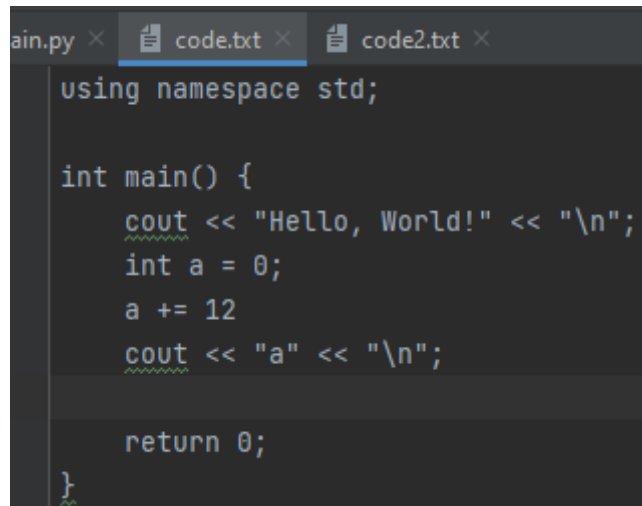


Рис 2. Граф автомата Мили для разбора и преобразования цикла.

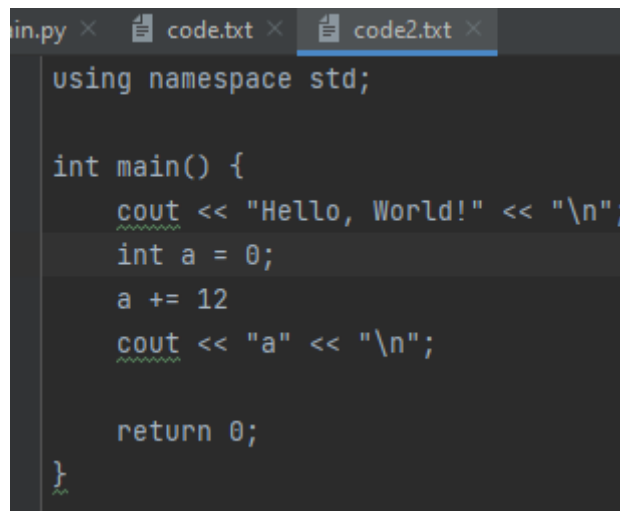
ПРИМЕРЫ РАБОТЫ ПРОГРАММЫ

Отсутствие в файле циклов:



```
main.py × code.txt × code2.txt ×  
using namespace std;  
  
int main() {  
    cout << "Hello, World!" << "\n";  
    int a = 0;  
    a += 12  
    cout << "a" << "\n";  
  
    return 0;  
}
```

Рис 3. Исходный файл без цикла



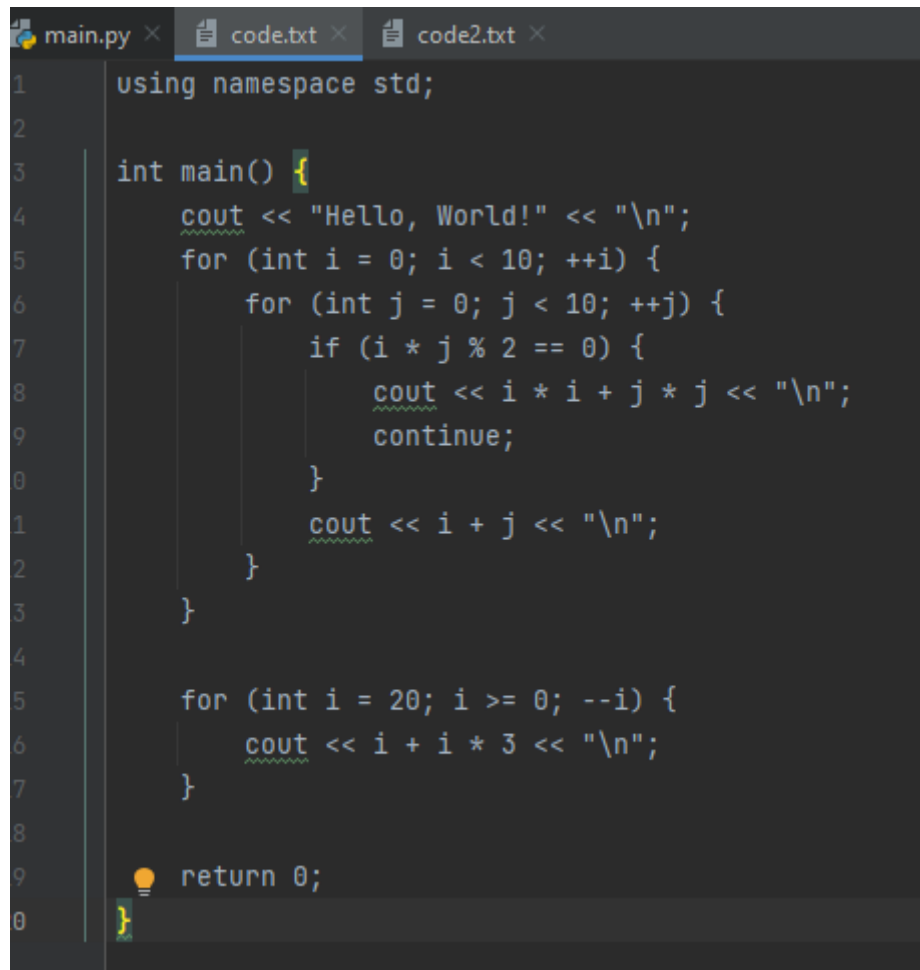
```
main.py × code.txt × code2.txt ×  
using namespace std;  
  
int main() {  
    cout << "Hello, World!" << "\n";  
    int a = 0;  
    a += 12  
    cout << "a" << "\n";  
  
    return 0;  
}
```

Рис 4. Итоговый файл

S0 -> S1 -> S2 -> S2 -> S2 -> S2 -> S2 -> S2 -> S2 -> S2 -> S2 -> S2 -> S3 -> S4 -> S0

Рис 5. Состояния автомата во время выполнения программы с отсутствующими в файле циклами

Наличие в файле вложенного и подряд идущих циклов:



```
1 using namespace std;
2
3 int main() {
4     cout << "Hello, World!" << "\n";
5     for (int i = 0; i < 10; ++i) {
6         for (int j = 0; j < 10; ++j) {
7             if (i * j % 2 == 0) {
8                 cout << i * i + j * j << "\n";
9                 continue;
10            }
11            cout << i + j << "\n";
12        }
13    }
14
15    for (int i = 20; i >= 0; --i) {
16        cout << i + i * 3 << "\n";
17    }
18
19    return 0;
20 }
```

Рис 6. Исходный файл с вложенными и подряд идущими циклами.

ИТОГИ

По итогу данной работы был спроектирован и реализован автомат типа Мили для разбора и преобразования цикла `for` в `label/goto` на языке C/C++. Также, были изображены блок-схема алгоритма и граф автомата Мили.

ПРИЛОЖЕНИЕ

```
#y1
print("S0 -> ", end="")
with open("code.txt", "r") as file:
    #y2
    data = file.readlines()
    # #y3
    # new_data = []
    # #x1
    # for item in data:
    #     #y4, y5
    #     print("S1 -> ")
    #     new_data.extend(item.strip("\n"))
    # #y6
    # data = new_data.copy()
    print("S1 -> ", end="")
    data = [item.strip("\n") for item in data]
#y7
file.close()
#y8
with open("code2.txt", "w") as file:
    #y9
    count = 0
    #x2
    for line in data:
        print("S2 -> ", end="")
        #x3
        if "for" in line:
            break
        #y10
        file.write(line + "\n")
        #y11
        count += 1
    #y12
    dataLen = len(data)
    # #y4
    # new_data = []
    # #x4
    # for idx in range(count, dataLen):
```

```

# #y13
#     new_data.extend(data[idx])
# #y6
#data = copy(new_data)
print("S3 -> ", end="")
data = [data[idx] for idx in range(count, dataLen)]
#x5
print("S4 -> ", end="")
if len(data) == 0:
    #y8
    file.close
    #y14
    print("S0")
    exit()
#y15, y16, y17
line = data[0]
data.pop(0)
line = line.strip()
#y18, y19
line = line[5:-3]
innerFor = line.split(";")
#y20, y21, y22
cycleVar = innerFor[0].strip()
cycleCondition = innerFor[1].strip()
cycleIncrement = innerFor[2].strip()
#y23, y24
file.write("\t" + cycleVar + ";\n")
file.write("label:\n")
#y9, y25
count = 0
stack = "{"
#x6
for line in data:
    print("S5 -> ", end="")
    #x7, y26
    if "{" in line:
        stack += "{"
    #x8, y27
    print("S6 -> ", end="")
    if "}" in line:

```

```

        stack = stack[:-1]
#x9, y28, y29, y30, y31, y32, y33, y34
print("S7 -> ", end="")
if line.strip() == "}" and stack == "":
    lessStr = f"if ({cycleCondition} - 1)" + " {"
    moreStr = f"if ({cycleCondition} + 1)" + " {"
    labelCondition = lessStr if "<" in cycleCondition else moreStr
    file.write("\t" + labelCondition + "\n")
    file.write("\t\t" + cycleIncrement + ";\n")
    file.write("\t\t" + "goto label;" + "\n")
    file.write("\t" + "}" + "\n")
    break
#y35, y36, y11
modifiedLine = line.replace(" ", "", 4)
file.write(modifiedLine + "\n")
count += 1
#y12
print("S8 -> ", end="")
dataLen = len(data)
# #y3
# new_data = []
# #x10
# for idx in range(count, dataLen):
#
# #y13
#     new_data = data[idx]
# #y6
# data = copy(new_data)
print("S9 -> ", end="")
data = [data[idx] for idx in range(count + 1, dataLen)]
#x10, y10
for line in data:
    print("S10 - > ", end="")
    file.write(line + "\n")
#y7
file.close()
print("S0", end="")

```