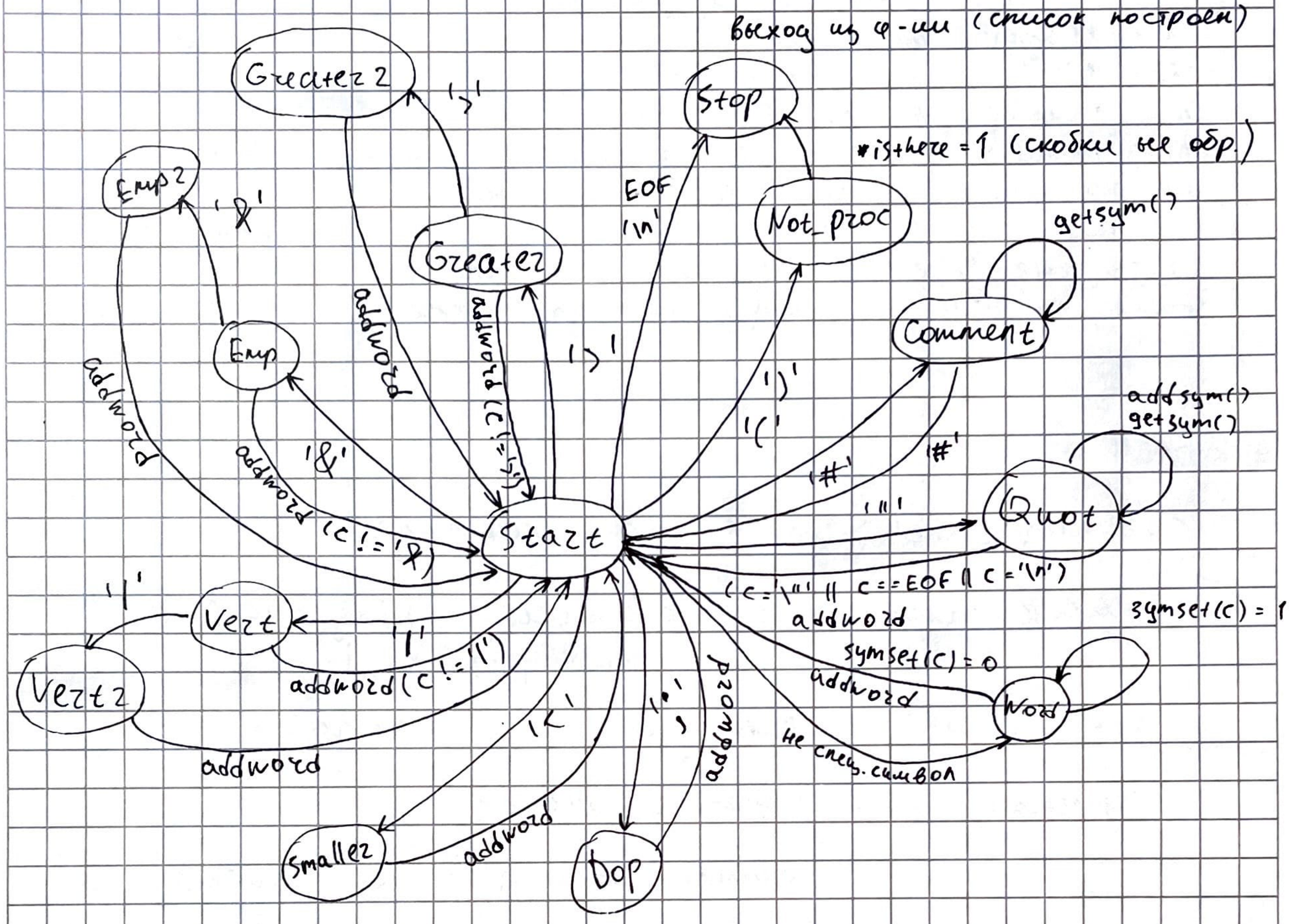
Горина 212 Отчет

1) Лексический этап (список)



2) Синтаксический этап (дерево)

- create_tree() → list_of_com() →

Вызывает list_of_com()
возвр постр. дерево

Создает дерево, заполняет tnext (||, &, ;)
заполняет поле ссылок next на след. ком.,
заполняет поле backgrnd. Вызывает conveyor()

→ conveyor() → simple_command() →

Создает конвейер, вызывает simple_command
для добавл. простой команды (аргументы,
ввод / вывод в файл). Заполняет поле ссылок
pipe. Обработка ошибок

Вызывает add_arg(),
заполняет поле ввода / вывода
в файл. Обработка ошибок

→ add_arg
добавляет в дерево аргументы
и их кол-во

3) Описание дерева:

```
struct tree
{
    int argc ;      // кол-во аргументов
    char **argv;    // список из имени команды и аргументов
    char *infile ;  // файл ввода
    char *outfile ; // файл вывода
    int append ;    // флаг для записи в конец файла
    int backgrnd ;  // флаг фонового режима
    int tnext    ;  // ; = 1 , && = 2 , || = 3 , иначе 0
    struct tree *pipe; // след команда после |
    struct tree *next; // след. команда после ;, &, &&, ||
}
```

4) БНФ

< команда_shell > ::= < список_команд >

< список_команд > ::= < конвейер > { [один из &, &&, ||, ;]
< конвейер > } [&, ;]

< конвейер > ::= < команда > { | < команда > }

< команда > ::= < простая_команда > |
(< список_команд >) [<{имя-файла}]
[[один из >, >>] < имя_файла >]

< простая_команда > ::= имя_файла { < аргумент > }
[< {имя_файла}] [[один из >, >>]
{имя_файла}]

5) Тесты

а) верные

```
cd ..
pwd > fpwd.txt
sleep 10 ; pwd
cat < file_in.txt > file_out.txt
echo a b c > f ; cat f & ls
cat f ; date ; pwd > zz
cat nosuchname && pwd
cat nosuchname || pwd
```

б) ошибочные

```
( pwd )
ls >
echo ||
abc
cat -n < file.txt |
||
wc <
```