

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ
И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра информатики

Дисциплина: Методы трансляции

Отчет по лабораторной работе №3
Синтаксический анализатор.

Выполнил:
студент гр. 953504 Басенко К. А.

Проверил:
Шиманский В.В.

Минск 2022

Содержание:

1. Постановка задачи. 3
2. Теория. 5
3. Результат работы программы. 6
4. Код с ошибками. 14
5. Выводы. 15

1. Постановка задачи

Освоение работы с существующими синтаксическими анализаторами. Разработать свой собственный синтаксический анализатор, выбранного подмножества языка программирования. Построить синтаксическое дерево.

В качестве анализируемого подмножества языка программирования будет использован язык программирования C#.

Для написания анализатора использован язык программирования Python.

Анализируемый код программы на языке C#:

```
1  _ = "string string";
2
3  _ =          5 + 5;
4
5  _ = 1 + 2;
6  _ = 0 -          3;
7  _ = 9 / 5;
8  _ = 4 * 3          ;
9
10 _ = (1 +2) * 3;
11 _ = (9 - 4)/7;
12 _ = 5 + 3 - 2;
13
14 float prevar;
15 int variable = 2;
16
17 const int _1 =1; const int _2= 2;const int _3=3;
18
19 var a = 3d;
20 double b = a + 5;
21 var c=(1.5435e6+b)*0.543f;
22
23
24 (_, _) = (2f, 2d);
25 (_, _) = (2m, 2u);
26 (_, _) = (2L, 2L);
27 (_, _) = (2uL, 2UL);
28
29 a += c - b;
30 b -= (a + c) * 0.11e2D;
31 c *= a * b;
32
33 _ = a < b;
34 _ = b >= c;
35
```

```

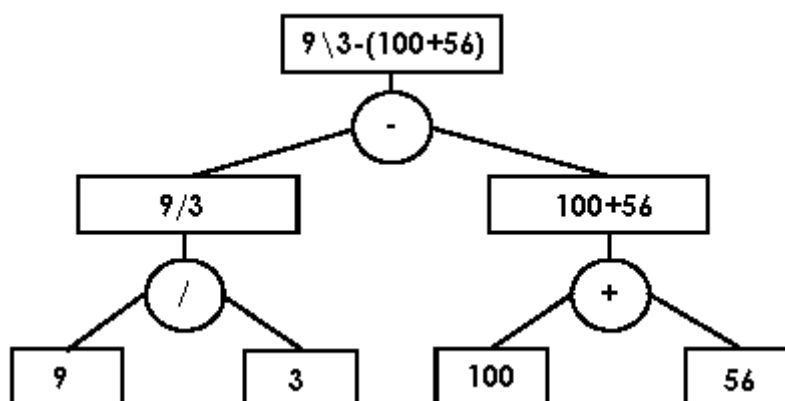
36 int func(int a, int b, int c)
37 {
38     for (int i = 0; i < b; i++)
39     {
40         a--;
41     }
42
43     while (c > 0)
44     {
45         a *= a;
46         c--;
47     }
48
49     return a;
50 }
51
52 _ = func(4, 5, 2);
53
54
55 void f1()
56 {
57
58 }
59
60 f1(
61
62 int a = 2
63
64 f1()
65
66

```

2. Теория

Синтаксический анализ — это процесс сопоставления линейной последовательности лексем (слов, токенов) естественного или формального языка с его формальной грамматикой. Результатом обычно является дерево разбора (синтаксическое дерево). Обычно применяется совместно с лексическим анализом.

Синтаксический анализатор — это программа или часть программы, выполняющая синтаксический анализ.



В ходе синтаксического анализа исходный текст преобразуется в структуру данных, обычно — в дерево, которое отражает синтаксическую структуру входной последовательности и хорошо подходит для дальнейшей обработки.

Как правило, результатом синтаксического анализа является синтаксическое строение предложения, представленное либо в виде дерева зависимостей, либо в виде дерева составляющих, либо в виде некоторого сочетания первого и второго способов представления.

Типы синтаксических анализаторов:

- 1) **LL-анализатор** (*LL parser*) — в информатике для подмножества. Он анализирует входной поток слева направо, и строит грамматики. Класс грамматик, для которых можно построить LL-анализатор, известен как .
- 2) **LR-анализатор** (*LR parser*) — для программ, написанных на некотором, который читает входной поток слева направо и производит наиболее правую продукцию .

3. Результат работы программы

В результате дерево программы имеет следующий вид:

```
{
|=
||'_-'
||'"string string"'
|=
||'_-'
||+
|||'5'
|||'5'
|=
||'_-'
||+
|||'1'
|||'2'
|=
||'_-'
||-
|||'0'
|||'3'
|=
||'_-'
||/
|||'9'
|||'5'
|=
```

```

||| '-'
||| *
||| '4'
||| '3'
|=
||| '-'
||| *
||| multiple values
||| +
||| '1'
||| '2'
||| '3'
|=
||| '-'
||| /
||| multiple values
||| -
||| '9'
||| '4'
||| '7'
|=
||| '-'
||| +
||| '5'
||| -
||| '3'
||| '2'
|declare var
||| 'prevar'

```

```

||<class 'data_types.TypeFloat'>
||''
|declare var
||'variable'
||<class 'data_types.TypeInt'>
||'2'
|decorate with type
||<class 'data_types.TypeConst'>
||declare var
|||'_1'
|||<class 'data_types.TypeInt'>
|||'1'
|decorate with type
||<class 'data_types.TypeConst'>
||declare var
|||'_2'
|||<class 'data_types.TypeInt'>
|||'2'
|decorate with type
||<class 'data_types.TypeConst'>
||declare var
|||'_3'
|||<class 'data_types.TypeInt'>
|||'3'
|define var define type
||a
||'3d'
|declare var
||'b'

```



```

||<class 'data_types.TypeDouble'>
||+
|||'a'
|||'5'
|define var define type
||c
||*
||multiple values
|||+
|||'|1.5435e6'
|||'|b'
|||'|0.543f'
|=
|multiple values
|||'|_
|||'|_
|multiple values
|||'|2f'
|||'|2d'
|=
|multiple values
|||'|_
|||'|_
|multiple values
|||'|2m'
|||'|2u'
|=
|multiple values
|||'|_

```

```

||| '_'
||multiple values
||| '2l'
||| '2L'
|=
||multiple values
||| '_'
||| '_'
||multiple values
||| '2ul'
||| '2UL'
|=
|| 'a'
||+
||| 'a'
||| -
||| 'c'
||| 'b'
|=
|| 'b'
|| -
||| 'b'
||| *
|||multiple values
||||+
||||| 'a'
||||| 'c'
||||| '0.11e2D'

```

```

| =
| | 'c'
| | *
| | | 'c'
| | | *
| | | 'a'
| | | 'b'
| =
| | '_'
| | <
| | | 'a'
| | | 'b'
| =
| | '_'
| | >=
| | | 'b'
| | | 'c'
| declare func
| | <class 'data_types.TypeInt'>
| | 'func'
| | multiple values
| | declare var
| | | 'a'
| | | <class 'data_types.TypeInt'>
| | | ''
| | declare var
| | | 'b'
| | | <class 'data_types.TypeInt'>
| | | ''

```

```

|||declare var
|||'c'
|||<class 'data_types.TypeInt'>
|||''
||{
||loop
|||(|
|||declare var
|||'|i'|
|||<class 'data_types.TypeInt'>
|||'|0'|
|||<
|||'|i'|
|||'|b'|
|||=
|||'|i'|
|||+
|||'|i'|
|||'|1'|
|||)
|||{
|||=
|||'|a'|
|||'-
|||'|a'|
|||'|1'|
|||}
||loop
|||(|

```

```

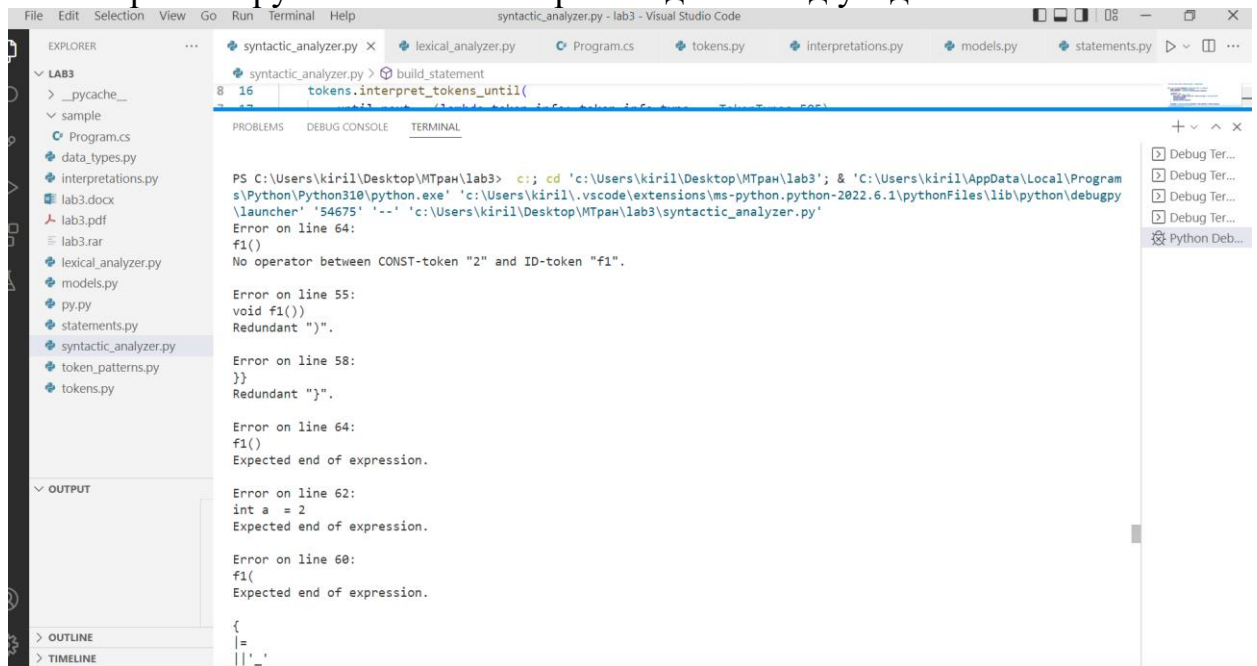
|>
|'c'
|'0'
|)
|{
|=
|'a'
|*
|'a'
|'a'
|=
|'c'
|-
|'c'
|'1'
|}
|return
|'a'
|}
|=
|'_ '
|call
|'func'
||multiple values
|'4'
|'5'
|'2'
|{
|}

|call
|'f1'
||multiple values
|'2'
||declare var
|'a'
|<class 'data_types.TypeInt'>
|call
|'f1'
||multiple values
}

```

Код с ошибками:

При обнаружении ошибок происходит вывод уведомления:



Выводы

В результате работы были получены знания о синтаксических анализаторах. А также синтаксических деревьях, способах их построения, их предназначения в системе интерпретаторов.

Для выделения синтаксических структур из кода был использован нисходящий парсер.

В итоге работы был построен простой синтаксический анализатор на основе уже имеющегося лексического, который способен не только строить и выводить синтаксическое дерево, но и выделять ошибки, уведомлять о них, выводя их на консоль.