

Отчет по лабораторной работе №3

по курсу «Конструирование компиляторов»

Выполнил: Камакин А.С. ИУ7-23М

Преподаватель: Ступников А.А.

Москва, 2020г

Цели и задачи

Цель работы: приобретение практических навыков реализации одного из методов синтаксического анализа - рекурсивного спуска.

Задачи работы:

1. Выбрать грамматику по варианту и реализовать для нее синтаксический разбор с использованием метода леворекурсивного спуска.

Текст программы

В программе 2 части:

- *parser* - непосредственно сам метод, который строит дерево разбора.
- *printTree* - метод, который печатает вывод дерева в формате dot.

Полный текст программы находится по адресу: https://github.com/lieroz/CourseOnCompilers/tree/master/lab_03

Тесты и результаты

Грамматика:

<выражение> -> <простое выражение> | <простое выражение> <операция отношения> <простое выражение>

<простое выражение> -> <терм> | <знак> <терм> | <простое выражение> <операция типа сложения> <терм>

<терм> -> <фактор> | <терм> <операция типа умножения> <фактор>

<фактор> -> <идентификатор> | <константа> | (<простое выражение>) | **not** <фактор>

<операция отношения> -> = | <> | < | <= | > | >=

<знак> -> + | -

<операция типа сложения> -> + | - | **or**

<операция типа умножения> -> * | / | **div** | **mod** | **and**

<программа> -> <блок>

<блок> -> { <список операторов> }

<список операторов> -> <оператор> <хвост>

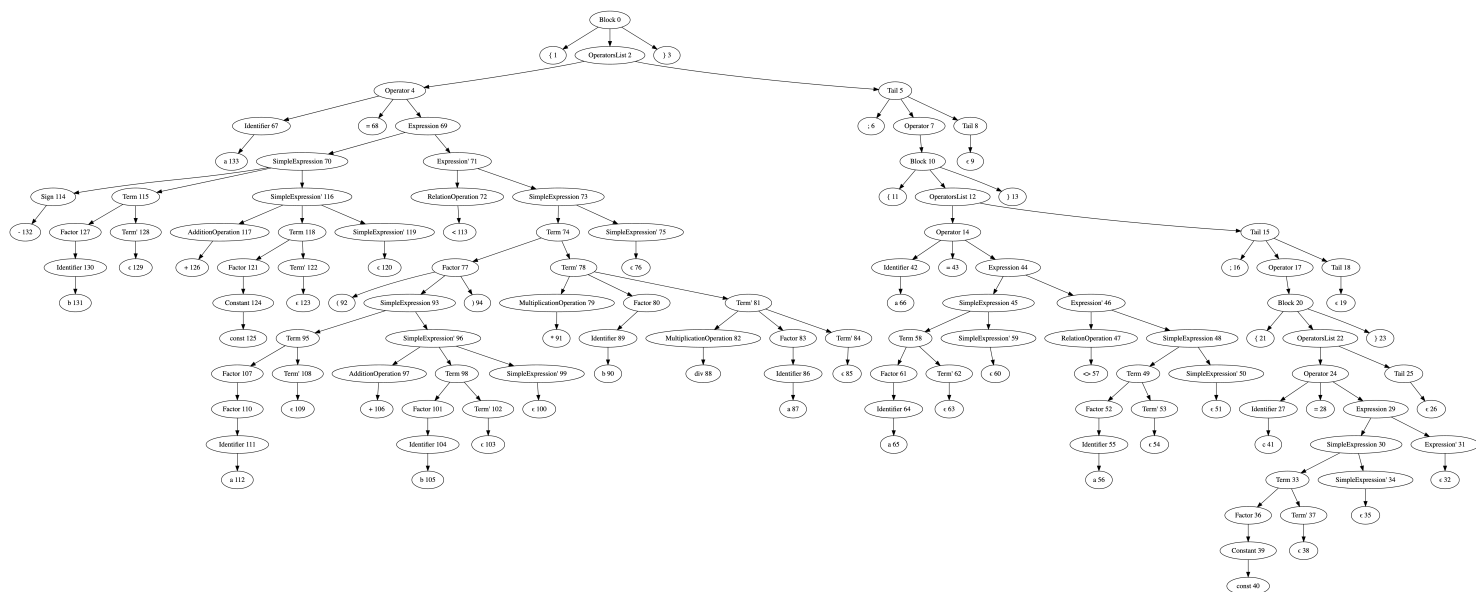
<хвост> -> ; <оператор> <хвост> | ε

<оператор> -> <идентификатор> = <выражение> | <блок>

Пример:

“{a=-b+const<(nota+b)*bdiva;{a=a<>a;{c=const}} }”

Результат:



Выводы

В данной лабораторной работе был реализован метод нисходящего рекурсивного анализа выражения. Были получены навыки реализации леворекурсивного спуска и проанализированы различные входные данные.