

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:

- *buildTree* - 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:

```
Grammar grammar = {
    {"program", {{{"block"}}}},
    {"block", {{{"", "operators list", ""}}}},
    {"operators list", {{{"operator", "tail"}}}},
    {"tail", {{{";", "operator", "tail"}, {":"}, {"e"}}}},
    {"operator", {{{"identifier", "=", "expression"}, {"{", "operators list", ""}}}},
    {"expression", {{{"simple expression", "expression"}}}},
    {"simple expression", {{{"term", "simple expression"}, {"sign", "term", "simple
expression"}}}},
    {"term", {{{"factor", "term"}}}},
    {"factor", {{{"identifier"}, {"constant"}, {"(", "simple expression", ")"}, {"not", "factor"}}}},
    {"relation operation", {{{"=="}, {"!="}, {"<"}, {"<="}, {">"}, {">="}}}},
    {"sign", {{{"+"}, {"-"}}}},
    {"addition type operation", {{{"+"}, {"-"}, {"or"}}}},
    {"multiplication type operation", {{{"*"}, {"/"}, {"div"}, {"mod"}, {"and"}}}},
    {"identifier", {{{"i"}}}},
```

$\} ;$

“{i=-c; {i=c<(not i+c*c); {i=c!(i*c);}} }”

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