



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика, искусственный интеллект и системы управления»
КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЁТ

по лабораторной работе № 3
по курсу «Конструирование компиляторов»
на тему: «Синтаксический разбор с использованием метода рекурсивного
спуска»
Вариант № 3

Студент ИУ7-23М
(Группа)

(Подпись, дата)

Волкова А. А.
(Фамилия И. О.)

Преподаватель

(Подпись, дата)

Ступников А. А.
(Фамилия И. О.)

2022 г.

Описание задания

Цель работы: приобретение практических навыков реализации рекурсивного спуска для синтаксического разбора.

Задачи работы:

1. Выбрать грамматику по варианту.
2. Дополнить грамматику блоком, состоящим из последовательности операторов присваивания.
3. Для модифицированной грамматики написать программу нисходящего синтаксического анализа с использованием метода рекурсивного спуска.

Вариант 3. Грамматика G3

Рассматривается грамматика выражений отношения с правилами

<выражение> ->

<арифметическое выражение> <знак операции отношения> <арифметическое

<арифметическое выражение> ->

<терм> |

<знак операции типа сложения> <терм> |

<арифметическое выражение> <знак операции типа сложения> <терм>

<терм> ->

<множитель> |

<терм> <знак операции типа умножения> <множитель>

<множитель> ->

<первичное выражение> |

<множитель> ^ <первичное выражение>

<первичное выражение> ->
 <число> |
 <идентификатор> |
 (<арифметическое выражение>)

<знак операции типа сложения> ->
 + | -

<знак операции типа умножения> ->
 * | / | %

<знак операции отношения> ->
 < | <= | = | >= | > | <>

Замечания.

1. Нетерминалы <идентификатор> и <число> — это лексические единицы (лексемы), которые оставлены неопределенными, а при выполнении лабораторной работы можно либо рассматривать их как терминальные символы, либо определить их по своему усмотрению и добавить эти определения.
2. Терминалы () — это разделители и символы пунктуации.
3. Терминалы + - * / % < <= = >= > <> — это знаки операций.
4. Нетерминал <выражение> — это начальный символ грамматики.

Вариант в стиле Си

<программа> ->
 <блок>

<блок> ->

```

    { <список операторов> }
    ,
<список операторов>
    <оператор> <хвост>

<хвост> ->
    ; <оператор> <хвост> | eps

```

Вариант не содержит левую рекурсию, но имеет ε -правило. Точка с запятой (;) ставится между операторами. Теперь начальным символом грамматики становится нетерминал `<программа>`. Варианта содержит цепное правило `<программа> -> <блок>`. Можно начальным символом грамматики назначить нетерминал `<блок>`. А можно `<блок>` считать оператором, т. е.

```

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

```

В последнем случае возможна конструкция с вложенными блоками. Если между символом присваивания (=) и символом операции отношения (=) возникает конфликт, то можно для любого из них ввести новое изображение, например, `:=`, `<-`, `==` и т. п.

Текст программы

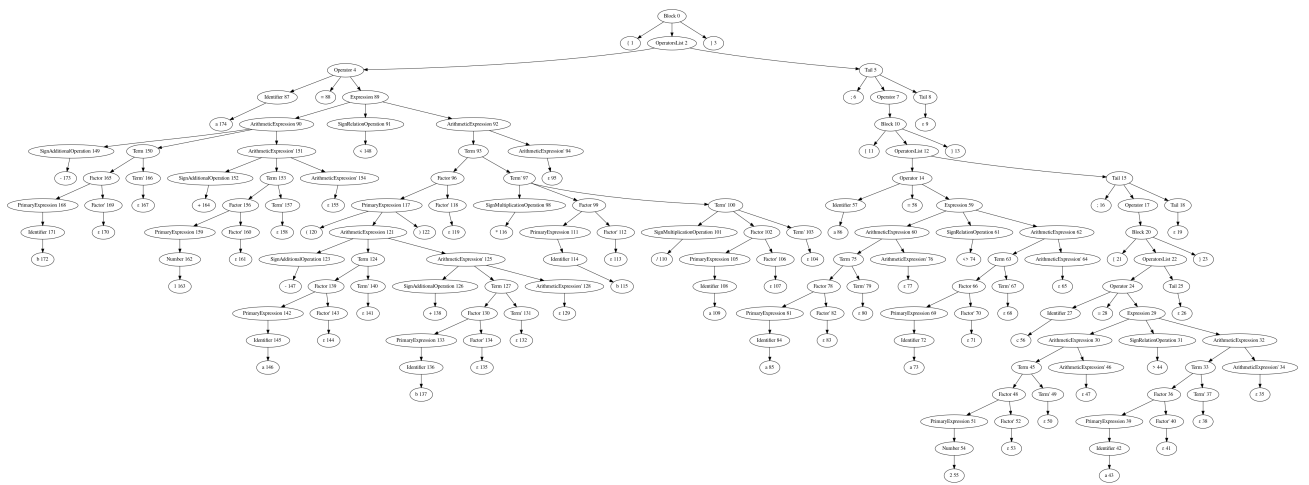
С полным текстом программы можно ознакомиться по адресу: <https://github.com/Volkovaan/CD/tree/lab03/lab03/src>.

Тестирование и результаты

Выражение:

{ a = -b + 1 < (-a + b) * b / a; {a = a <> a; {c = 2 > a}}}

Результат:



Выводы

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