



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 и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 3

Дисциплина Конструирование компиляторов

**Тема Синтаксический разбор с использованием метода рекурсивного
спуска**

Вариант №6 (1)

Студент Иванов В.А.

Группа ИУ7-21М

Преподаватель Ступников А.А.

Москва.
2023 г.

Цель работы: приобретение навыков синтаксического разбора исходного кода программы на примере метода рекуррентного спуска

Задачи работы:

- 1) Изучить метод.
- 2) Преобразовать исходную грамматику к LL(1) путём ранее изученных методов.
- 3) Дополнить грамматику до Си-подобного синтаксиса
- 4) Разработать и протестировать программу рекуррентного спуска по полученной грамматике.

Вариант 1. Грамматика G1.

Грамматика G1 определяется правилами:

<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

Пусть

$\langle \text{идентификатор} \rangle \rightarrow i$

$\langle \text{константа} \rangle \rightarrow C$

Грамматика содержит левую рекурсию, после устранения она приобретает вид:

$\langle \text{выражение} \rangle \rightarrow$

$\langle \text{простое выражение} \rangle \mid$

$\langle \text{простое выражение} \rangle \langle \text{операция отношения} \rangle \langle \text{простое выражение} \rangle$

$\langle \text{простое выражение} \rangle \rightarrow$

$\langle \text{терм} \rangle \langle \text{простое выражение}' \rangle \mid$

$\langle \text{знак} \rangle \langle \text{терм} \rangle \langle \text{простое выражение}' \rangle$

$\langle \text{простое выражение}' \rangle \rightarrow$

$\langle \text{операция типа сложения} \rangle \langle \text{терм} \rangle \langle \text{простое выражение}' \rangle \mid$

ε

$\langle \text{терм} \rangle \rightarrow \langle \text{фактор} \rangle \langle \text{терм}' \rangle$

$\langle \text{терм}' \rangle \rightarrow \langle \text{операция типа умножения} \rangle \langle \text{фактор} \rangle \langle \text{терм}' \rangle \mid \varepsilon$

$\langle \text{фактор} \rangle \rightarrow$

$\langle \text{идентификатор} \rangle \mid$

$\langle \text{константа} \rangle \mid$

$(\langle \text{простое выражение} \rangle) \mid$

$\text{not } \langle \text{фактор} \rangle$

$\langle \text{операция отношения} \rangle \rightarrow = \mid < \mid < = \mid > \mid > =$

$\langle \text{знак} \rangle \rightarrow + \mid -$

$\langle \text{операция типа сложения} \rangle \rightarrow + \mid - \mid \text{or}$

$\langle \text{операция типа умножения} \rangle \rightarrow * \mid / \mid \text{div} \mid \text{mod} \mid \text{and}$

$\langle \text{идентификатор} \rangle \rightarrow i$

$\langle \text{константа} \rangle \rightarrow C$

Вычисление FIRST:

$\text{FIRST}(\langle \text{выражение} \rangle) = \{i, C, (, n, +, -\}$

$\text{FIRST}(\langle \text{простое выражение} \rangle) = \{i, C, (, n, +, -\}$

$\text{FIRST}(\langle \text{простое выражение}' \rangle) = \{+, -, o, \varepsilon\}$

$\text{FIRST}(\langle \text{терм} \rangle) = \{i, C, (, n\}$

$\text{FIRST}(\langle \text{терм}' \rangle) = \{*, /, d, m, a, \varepsilon\}$

$\text{FIRST}(\langle \text{фактор} \rangle) = \{i, C, (, n\}$

$\text{FIRST}(\langle \text{операция отношения} \rangle) = \{=, <, >\}$

$\text{FIRST}(\langle \text{знак} \rangle) = \{+, -\}$

$\text{FIRST}(\langle \text{операция типа сложения} \rangle) = \{+, -, o\}$

$\text{FIRST}(\langle \text{операция типа умножения} \rangle) = \{*, /, d, m, a\}$

$\text{FIRST}(\langle \text{идентификатор} \rangle) = \{i\}$

$\text{FIRST}(\langle \text{константа} \rangle) = \{C\}$

Посмотрев на правые части правил, уверенно заявляем, что теперь G1 является LL(1) грамматикой – FIRST от правых частей правил с одинаковыми левыми частями не пересекаются. Можно двигаться дальше.

В качестве расширенной грамматики выбран вариант в стиле Си.

$\langle \text{программа} \rangle \rightarrow \langle \text{блок} \rangle$

$\langle \text{блок} \rangle \rightarrow \{ \langle \text{список операторов} \rangle \}$

$\langle \text{список операторов} \rangle \rightarrow \langle \text{оператор} \rangle \langle \text{хвост} \rangle$

$\langle \text{хвост} \rangle \rightarrow ; \langle \text{оператор} \rangle \langle \text{хвост} \rangle \mid \varepsilon$

$\langle \text{оператор} \rangle \rightarrow \langle \text{идентификатор} \rangle = \langle \text{выражение} \rangle$

Расширенная грамматика также является LL(1) грамматикой.

Выводы

В ходе лабораторной работы был изучен метод рекуррентного спуска. Получены практические навыки в преобразовании грамматики к нужной форме. Реализована и протестирована программа, реализующая метод.