Министерство образования и науки российской федерации

(минобрнауки россии)

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Санкт-Петербургский государственный политехнический университет» (ФГБОУ ВПО «СПбГПУ»)

**Институт менеджмента и информационных технологий**

(филиал)федерального государственного бюджетного образовательного учреждения высшего профессионального образования

«Санкт-Петербургский государственный политехнический университет» в г. Череповце (ИМИТ «СПбГПУ»)

Кафедра ПО ВТ и АС

ЛАБОРАТОРНАЯ РАБОТА №4

Дисциплина: «Теория языков программирования и методы трансляции»

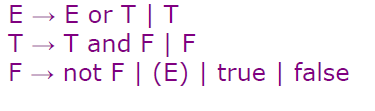
Тема: «МП-автоматы для нисходящего синтаксического анализа»

|  |  |  |
| --- | --- | --- |
| Выполнил студент группы о.291: | Шанин Игнат Леонидович | |
|  | № зачетной книжки: о2080127 | |
| Проверил: | Михайлов Андрей Евгеньевич | |
|  | «\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. | |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | отметка о зачете | подпись преподавателя |

# Задание

На основе порождающей КС-грамматики построить МП-автомат для нисходящего синтаксического анализа без возвратов и реализовать его на любом ЯП. Алгоритм построения должен основываться на функциях FIRST() и FOLLOW(). Программа должна принимать на вход файл, содержащий строки символов. На выходе формируется сообщение о том, принадлежат или нет строки языку, порождаемому грамматикой.

Грамматика 1:



Грамматика 2:

R R^F | RT | R\* | (R) | F

F T | b

T +a | -a

# Теоретические сведения

#### Основная работа по построению анализатора – это создание множества , которое удобно представить в виде таблицы.

Нисходящий анализ, как правило, включает в себя только одно состояние, и поэтому требуется одна таблица. Такая таблица содержит столбцы с входными символами и строки с символами магазина. В ячейки таблицы помещаются правила вида выполнение которых приводит к удалению из магазина символа Ф и добавления в него цепочки , начиная с последнего символа. Для построения таблицы необходимы функции FIRST и FOLLOW.

## Функции FIRST и FOLLOW

Пусть - это КС-грамматика. Тогда для произвольной цепочки  можно определить функцию FIRST() как множество терминальных символов, с которых начинаются строки, выводимые из . Если из  можно вывести пустую строку , то она также принадлежит этому множеству.

Функция FOLLOW() определяет для некоторого нетерминала  множество терминальных символов, которые могут появиться непосредственно после  в некой сентенциальной форме грамматики, то есть это множество символов таких, что существует вывод вида  Если  самым правым некой цепочки вывода, то к множеству FOLLOW() добавляется символ окончания строки.

Для вычисления функции FIRST() вводится функция FIRST(), где .

## Построение таблицы предикативного анализатора

Для построения таблицы предиктивного анализатора используется алгоритм, основанный на следующем принципе. Предположим, что правило выводаграмматики и есть некий терминальный символ . Тогда анализатор заменяется  на  если входной символ в данный момент является .

# Решение

## Преобразование грамматик

### Грамматика 1

* E → E or T | T
* T → T and F | F
* F → not F | (E) | true | false

Устранение левой рекурсии:

* E → E or T | T
  + E → ТE’
  + E’→ or ТЕ’ | λ
* T → T and F | F
  + Т → FТ’
  + T’→ and FT’ | λ

Результат:

* E → ТE’
* E’→ or ТЕ’ | λ
* Т → FТ’
* T’→ and FT’ | λ
* F → not F | (E) | true | false

### Грамматика 2

* R → R^F | RT | R\* | (R) | F
* F → T | b
* T → +a | -a
* Устранение левой рекурсии:
* R → R^F | RT | R\* | (R) | F
  + R → FR’ | TR’ | (R)
  + R’→ ^FR’ | TR’ | \*R’| λ

Результат:

* R → FR’ | TR’ | (R)
* R’→ ^FR’ | TR’ | \*R’| λ
* F → T | b
* T → +a | -a

## Листинг программы

# Результат работы программы

# Вывод