Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное учреждение высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

Кафедра «Информационных технологий и систем»

**«Проектирование синтаксического анализатора»**

Лабораторная работа

по дисциплине:

«Теория языков программирования и методы трансляции»

Проверил:

\_\_\_\_\_\_\_\_ / Макаров В. А.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2021г.

Выполнил:

студент группы 8091:

\_\_\_\_\_\_/ Васильев И. В.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2021 г.

**Великий Новгород**

**2021**

# Постановка задачи

Формулировка:

Разработать синтаксический анализатор. Синтаксический блок компилятора должен быть объединён с лексическим блоком. На входе - программа на определенном заданием языке, разработанная в соответствии с правилами грамматики. На выходе – сообщения в виде таблицы о входных данных и операций над ними, в случае ошибки вывод сообщение об ошибке.

Описание грамматики:

<STAT> -> <CREATE\_STM><STAT>

<STAT> -> <DROP\_STM><STAT>

<STAT> -> <ALTER\_STM><STAT>

<STAT> ->E

<ALTER\_STM> -> alter table ID add column <FIELD\_DEF\_LIST>;

<ALTER\_STM> -> alter table ID drop column ID ;

<CREATE\_STM> -> create table ID ( <FIELD\_DEF\_LIST>) ;

<DROP\_STM> -> drop table ID ;

<FIELD\_DEF\_LIST> -> <FIELD\_DEF> , <FIELD\_DEF\_LIST>

<FIELD\_DEF\_LIST> -> <FIELD\_DEF>

<FIELD\_DEF> -> ID TYPE

Пояснения к грамматике:

ID – идентификатор, полученный из лексического блока (например: itsNameTable)

TYPE– тип, полученный из лексического блока (например: integer, date)

# Определение

Для реализации поставленной задачи используется нисходящий синтаксический анализ, реализуемый методом рекурсивного спуска. Построение синтаксического дерева начинается с корня и переходит к листьям. Данный метод распознаёт грамматики типа LL (1).

Для удобного представление грамматики, создадим таблицу:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Left | Right(1) | Right(2) | Right(3) | Right(4) | Right(5) | Right(6) | Right(7) |
| **STAT** | **CREATE\_STM** | **STAT** |  |  |  |  |  |
| **STAT** | **DROP\_STM** | **STAT** |  |  |  |  |  |
| **STAT** | **ALTER\_STM** | **STAT** |  |  |  |  |  |
| **STAT** | **E** |  |  |  |  |  |  |
| **ALTER\_STM** | alter | table | ID | add | column | **FIELD\_DEF\_LIST** | ; |
| **ALTER\_STM** | alter | table | ID | drop | column | ID | ; |
| **CREATE\_STM** | create | table | ID | ( | **FIELD\_DEF\_LIST** | ) | ; |
| **DROP\_STM** | drop | table | ID | ; |  |  |  |
| **FIELD\_DEF\_LIST** | **FIELD\_DEF** | , | **FIELD\_DEF\_LIST** |  |  |  |  |
| **FIELD\_DEF\_LIST** | **FIELD\_DEF** |  |  |  |  |  |  |
| **FIELD\_DEF** | ID | TYPE |  |  |  |  |  |

# Описание реализации

Для реализации метода рекурсивного спуска нужно для каждого нетерминального символа создать соответствующую функцию.

Главной функцией является функция STAT, которая является вершиной синтаксического дерева.

Суть алгоритма состоит в анализе входящего слова, выданное лексическим анализатором, после чего функция выбирает поведение, в зависимости от входного значения.

# Алгоритм

Пример из одной команды:

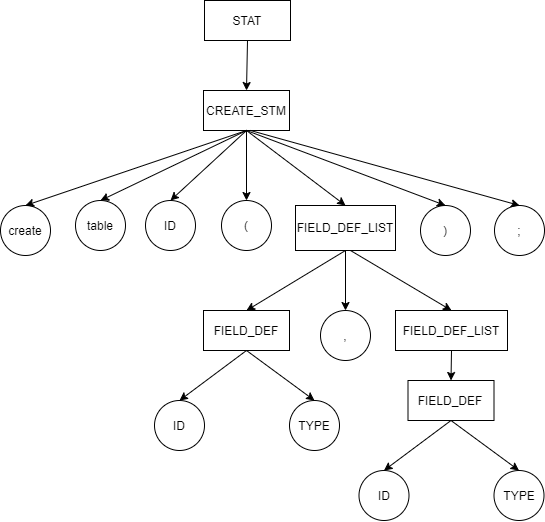
Входная цепочка имеет вид:

create table hello ( next integers, hh integer );

Цепочка, обработанная лексическим анализатором:

create table ID ( ID TYPE , ID TYPE ) ;

Синтаксическое дерево:



Пример из трёх команд:

Входная цепочка:   
create table journal ( count\_name integer,price integer,count date);

alter table journal add column issued date,count integer;

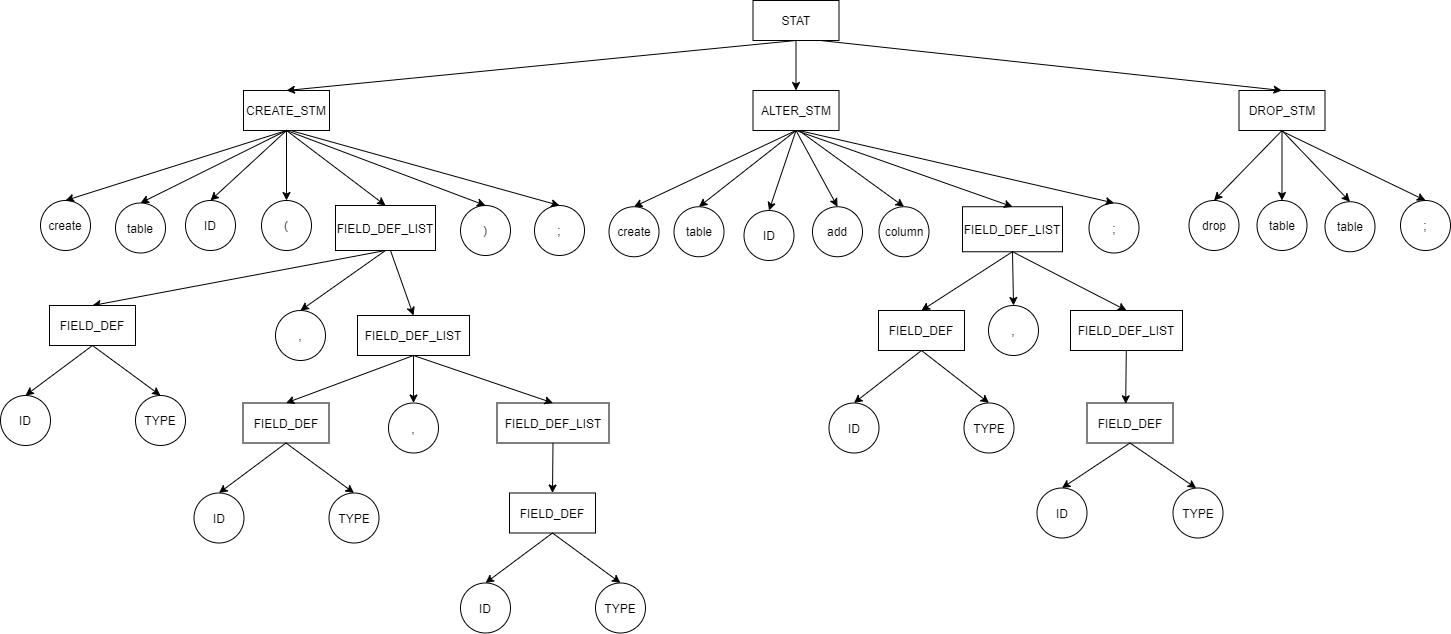
drop table journal;

Цепочка, обработанная лексическим анализатором:

create table ID ( ID TYPE,ID TYPE,ID TYPE);

alter table ID add column ID TYPE,ID TYPE;

drop table ID;



Обработчик ошибок

Для входной строки: create table hello\_world ) next integers(;

Обработчик выдаст сообщение:

Error, incorrect ‘(‘ entry!

The CREATE is syntactically incorrect

Вывод

В ходе работы на основе выбранной грамматики был разработан синтаксический анализатор.