分为moduleparser和functionparser

Moduleparser先相对于整个程序进行parse，

code : (function\_def | simple\_decl | using\_directive | water)\*;

function\_def中存在compound statement，即函数体。

获取函数体后，将其交给functionparser，具体对函数语句进行parse。

把每个表达式都当作赋值表达式处理

statements: (pre\_opener

| pre\_closer

| pre\_else {preProcSkipToEnd(); }

| statement)\*;

statement: opening\_curly

| closing\_curly

| block\_starter

| jump\_statement

| label

| simple\_decl

| expr\_statement

| water

;

以函数为单位，必须存在　FunctionName{

C文件

1. test.c 存在结构体定义，只有一个函数，没有基础头文件 可以生成AST
2. mov.c 代码片段，只存在一个独立的函数，存在语句省略[…]，直接调用未定义的函数 可以生成AST
3. novariabledef.c 存在未定义的变量，返回值未定义　可以生成AST
4. nofuncstruct.c 没有函数结构，单纯的一些语句　无法生成AST
5. partfuncstruct.c 部分函数结构，如大括号只有一边　能生成部分AST，即能识别出函数，但最后一条语句无法识别出类型

C++文件

1. 单独的C++文件 mysql中的单独文件

不能识别函数参数中包含#ifdef的函数　AST中直接把函数舍弃

IdentifierDecl \*a

C++程序/版本 Chrome Mysql 大程序 llvm

宏定义

存入数据库

Java语法