



简易翻译器

王雨婷 王华清 孙浩

CONTEXT



实验选题



文法设计



具体实现



效果展示



Part 1

实验选题

C++ -> javascript

- 源语言：C++
- 目标语言：javascript
- 测试程序：四则运算计算

C++与javascript语法对比

	C++	javascript
函数声明	<code>returnType functionID (paraType para, ..., paraType para){}</code>	<code>functionID = function(para list){}</code>
函数传参	需要传递参数类型，支持引用类型&。	无需传递参数的类型，除对object 类型之外不支持引用类型。
变量类型	明确区分数据类型。	除了对数组是array 类型之外基本不区分数据类型，都为var 类型。
数组	<code>arrayType arrayID[] = {para}</code>	<code>var arrayID = new Array(para)</code>
整体结构	main 函数作为程序的入口，对语法规则限制更多，更加规范	无特定的main 函数作为程序的起始地址，整个程序布局由function 定义语句块作为分界，相对来说限制较少，语言更灵活。



Part 2

文法设计

设计原则

以代码嵌套进行分层，以代码作用进行模块划分。

利用消除左递归，合并左公因子等方法生成LL(*)文法。

语句群

program(程序) -> statements(语句群)

statements (语句群) -> statement (语句) statements(语句群) | e

statement (语句) -> type func_or_var(函数或者变量)

| "#include<" ID ">"

| 'using' 'namespace' ID ';' ;

| COMMENT (注释)

func_or_var -> dec_func(函数) | dec_var (变量)

函数&变量

dec_fun -> dec_func_name '{' func_implement (函数实现) '}' semi_colon
dec_func_name -> ID '(' dec_param(参数们) ')'

dec_param (参数们) -> type ID dec_param_(参数递归) | e
dec_param_ (参数递归) -> , dec_param | e

func_implement(函数实现) -> block

dec_var -> ID dec_array ';' (数组定义)
 | ID dec_array '=' array_value ';' (数组定义+赋值)
 | ID dec_expression dec_var_ (正常变量)

dec_array -> [INT]

array_value -> { INT ints }

dec_var_ -> ';' | ',' dec_var

dec_expression -> expr_value | e

基本功能模块

block -> stat block | e

stat -> type dec_var(声明变量)

| ie_stat | for_stat | while_stat | switch_stat | break ;(语句块)

| call_function ; (函数调用)

| 'return' return_sentence ; (返回)

| normalExp ;(表达式)

| cin input ; | cout outputs ;(输入输出)

| COMMENT (注释)



Part 3

具体实现

工具使用

使用开源的语法分析器——antlr，由上述的文法设计编译成文法文件，通过antlr处理.g文件可生成对应的Lexer（词法分析器）和Parser（语法分析器）的java文件。

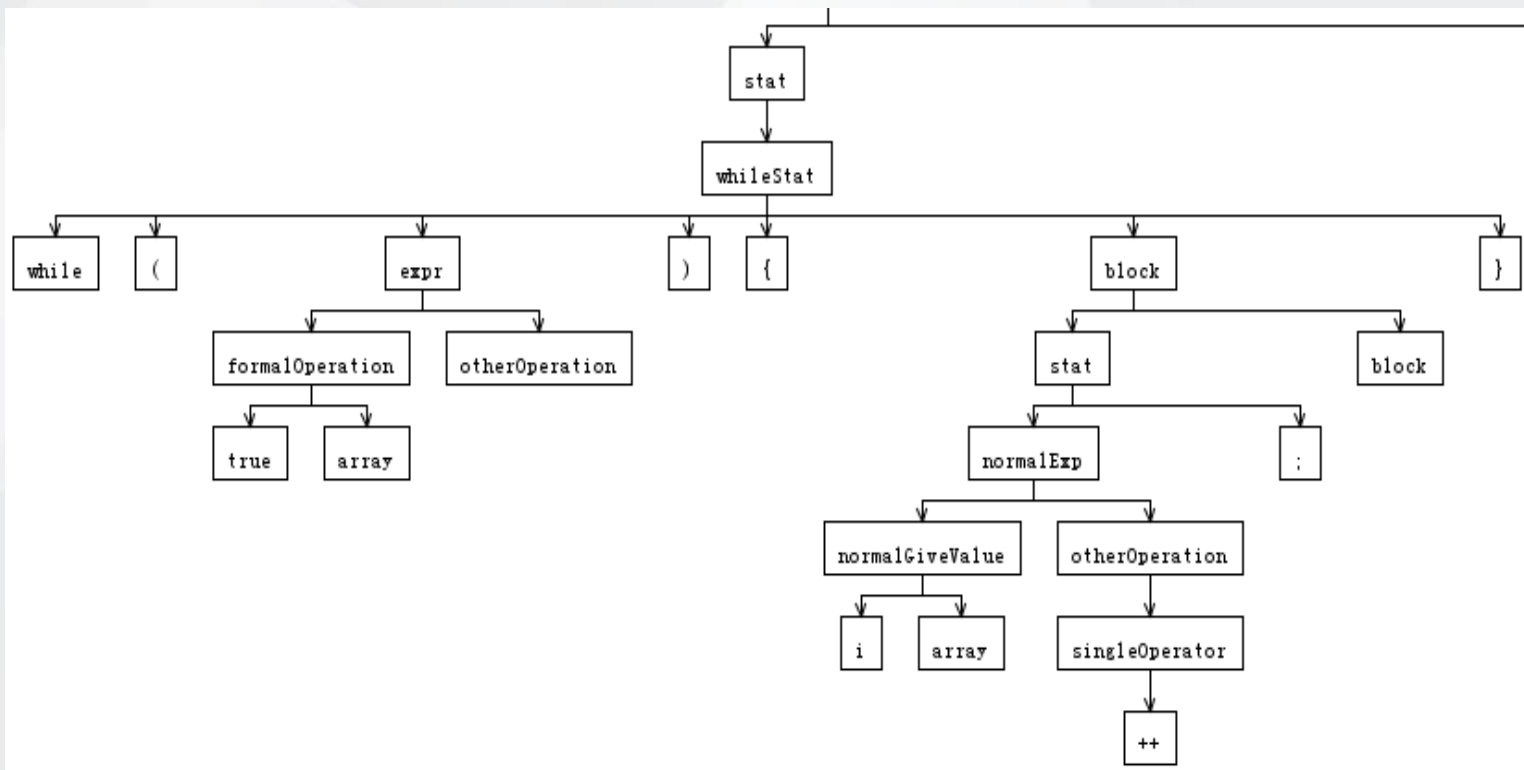
之后利用antlrworks生成输入程序的语法分析树，对文法进行检测、验证。

支持语法

- 函数定义、调用
- 全局变量、局部变量
- 数组定义、初始化
- 数组下标取值
- for、while循环结构
- include、using
- if-elseif-else选择分支结构
- switch-case语句
- 多种函数、变量、参数类型
- 逻辑和算术运算（单目、双目）
- 注释
- 输入、输出

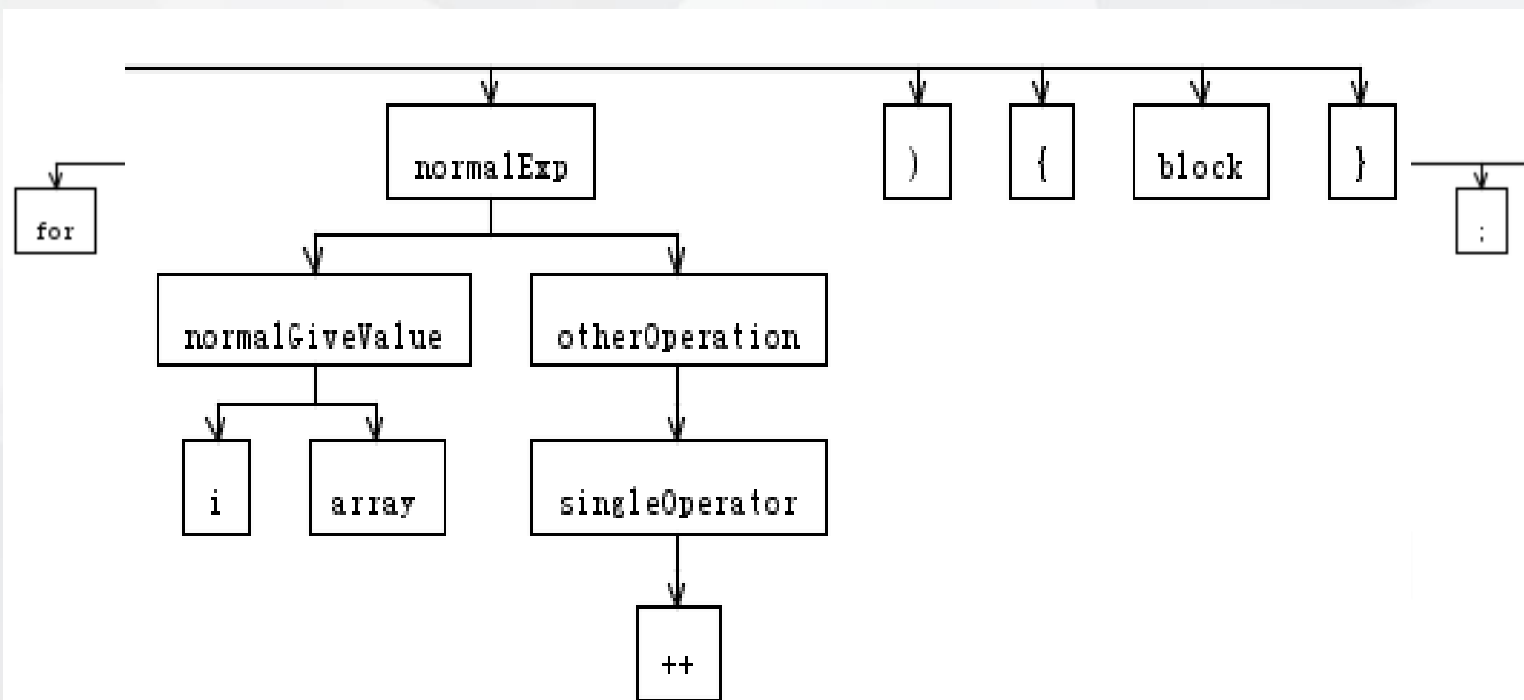
while循环语法树

while(true) {i+ +;}



for循环语法树

for(int i = 0; i < 2; i++)



语义规则

综合属性 value：用于产生目标JavaScript代码

继承属性 tab：用于自动处理生成代码的缩进



Part 4

效果展示

THANKS FOR LISTENING

