

Updated Syntax Analysis Grammar

Xavier

2025 年 3 月 7 日

声明

此文法仅针对样例 C++ 编写。规定大写字母为首的单词为非终结符，小写字母为首的为终结符，'...' 表示可出现一次或多次，? 表示可出现 0 次或 1 次，被 \$\$ 包裹的部分为一个整体，etc. 表示省略。

1. 主结构

Program \rightarrow Decl \dots (程序由一个或多个声明组成)

2. 声明部分 (Decl)

Decl \rightarrow StructDecl | Type id Suffix (结构体声明或“类型 *id* + 后续”)

VarDecl \rightarrow Type id VarDeclSuffix

2.1 结构体声明 (StructDecl)

StructDecl \rightarrow struct id? { Decl \dots } id? ;

(仅示例，表示结构体内可以继续声明变量、函数等)

2.2 声明后缀 (Suffix)

Suffix \rightarrow (Param) FunctionBody | VarDeclSuffix

这里是关键!

如果遇到 "(", 说明它是函数声明;

否则走 *VarDeclSuffix* 路径处理变量声明。

2.2.1 函数体 (FunctionBody)

FunctionBody \rightarrow StmtNoDecl | ; (可以是函数定义或函数原型)

2.2.2 变量声明后缀 (VarDeclSuffix)

VarDeclSuffix \rightarrow = Expression ; | ;

如果下一个符号是“=”，则初始化；否则直接分号结束。

3. 函数参数 (Param)

Param $\rightarrow \epsilon \mid \text{ParamList}$ (参数可为空, 也可有若干个)

ParamList $\rightarrow \text{Type id} (, \text{Type id}) \cdots$ (简化表示, 多个 *type id* 逗号分隔)

4. 语句部分 (Statement)

注意! *Expression* 不含分号, *ExpressionStatement* 含有分号

Statement	\rightarrow CompoundStatement	块语句
	SelectionStatement	选择语句
	IterationStatement	循环语句
	ReturnStatement	返回语句
	ExpressionStatement	表达式
	IOStmt	输入输出语句
	VarDecl	变量声明语句
	StructDecl	结构体声明语句
	break ; continue ;	<i>break</i> 语句和 <i>continue</i> 语句

StmtNoDecl	$\rightarrow \{ ? \text{ StmtNoDecl } \cdots \} ?$	非声明语句
	SelectionStatement	选择语句
	IterationStatement	循环语句
	ReturnStatement	返回语句
	ExpressionStatement	表达式
	IOStmt	输入输出语句
	break ; continue ;	<i>break</i> 语句和 <i>continue</i> 语句

CompoundStatement \rightarrow { Statement \dots }
 SelectionStatement \rightarrow if (Expression) StmtNoDecl \$ else StmtNoDecl \$?
 IterationStatement \rightarrow while (Expression) StmtNoDecl
 ReturnStatement \rightarrow return ExpressionStatement
 DeclInFunction \rightarrow VarDecl | StructDecl
 ExpressionStatement \rightarrow Expression ;
 IOSstmt \rightarrow cin | cout StreamOpSequence ;
 StreamOpSequence \rightarrow >> | << Expression \dots

循环语句日后还将支持 *for* 循环

5. 表达式部分 (Expression)

Expression	→	AssignmentExpr	表达式入口
AssignmentExpr	→	LogicalOrExpr \$ = AssignmentExpr \$?	赋值表达式
LogicalOrExpr	→	LogicalAndExpr \$ LogicalAndExpr \$?	逻辑或表达式
LogicalAndExpr	→	EqualityExpr \$ && EqualityExpr \$?	逻辑与表达式
EqualityExpr	→	RelationalExpr \$ \$ == != \$ RelationalExpr \$?	相等性表达式
RelationalExpr	→	ShiftExpr \$ < > <= >= ShiftExpr \$?	关系表达式
ShiftExpr	→	AddExpr \$ << >> AddExpr \$?	位移表达式
AddExpr	→	Term \$ + - Term \$?	加减法表达式
Term	→	UnaryExpr \$ * / % UnaryExpr \$?	乘除法表达式
UnaryExpr	→	\$! - UnaryExpr \$ PostfixExpr	一元表达式
PostfixExpr	→	Factor \$ Argument \$?	后缀表达式, 处理函数调用
Factor	→	identifier Literal (Expression)	因子
Literal	→	integer floating character stringLiteral boolean	字面量
Argument	→	Expression \$, Expression \$?	