

C 语言声明语句的文字描述和可视化表示

古 辉, 游联启

(浙江工业大学计算机科学与技术学院, 杭州 310023)

摘 要: C 语言复杂的声明语句难以理解。为此, 提出一种程序理解算法, 自动地对 C 语言声明语句进行部分语义文字描述和可视化表示。引入图元和解释 2 个概念, 构成该程序理解算法的理论基础, 这些概念同样可应用于程序理解算法的其他方面。实验结果表明, 声明语句的文字描述和可视化表示能够有效提高程序可读性。

关键词: 程序理解; 声明语句; 可视化; 语义解释; 图元; 基本词汇

Text Description and Visualization Representation for C Language Declaration Statement

GU Hui, YOU Lian-qi

(College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou 310023, China)

【Abstract】 It is difficult to be understood that some complicated declarations in C language. In order to solve this problem, an algorithm of program comprehension which auto charts and describes the declaration of C language is proposed. The concept of chart and description are proposed in this algorithm. They are the key of the algorithm and also used in other researches of program understanding. Experimental results show that the chart and description of C declaration can be understood.

【Key words】 program comprehension; declaration statement; visualization; semantic interpretation; primitive; basic vocabulary

DOI: 10.3969/j.issn.1000-3428.2012.03.016

1 概述

尽管实际应用中可以避免使用过于复杂的 C 语言声明语句, 但懂得如何理解和使用这些复杂的声明也很重要, 有时确实需要这种声明, 因为这种声明可以生成更高效、更紧凑的代码^[1]。

因此, 本文提出一种可以自动实现对 C 语言声明语句进行部分语义描述和可视化表示的算法。该算法思想同样可以被程序理解的其他方面所借鉴^[2-3]。

2 右左法则

右左法则^[1]是常用的 C 语言复杂声明解析方法。右左法则如下: 首先从最里面的圆括号看起, 然后往右看, 再往左看。每当遇到圆括号时, 就应该掉转阅读方向。一旦解析完圆括号里面所有的东西, 就跳出圆括号。重复这个过程直到整个声明解析完毕。

3 图元与解释

定义 1 用于可视化表示 C 语言声明语句的基本图形符号称为图元。

声明语句可视化表示的图元^[4]与组合如表 1 所示。声明语句的语义描述需要相关语句来构成, 如表 2 所示。其中, 图元 2 可根据数组的大小进行调整。从表 1 可以看出, 实现函数的参数 factor 为横坐标因子, 若 factor 加 1, 则相应位置的横坐标加 60, 图元 1 和图元 2 占 2 个 factor, 图元 3 和图元 4 占 5 个 factor。

定义 2 用于构成描述声明语句语意的语句片段称为基本词汇。

定义 3 通过程序理解将一些基本词汇组合为一条声明语

句的解释称为语义描述。

例如: `int *(*p[2])(int,int)`; 可视化后可表示为如图 1 所示的图形。

语义描述为:

- (1) p 是一个具有 2 个元素的数组;
- (2) 这个数组的每个元素都是一个指针;
- (3) 这些指针指向一个参数为 (int, int) 的函数;
- (4) 函数的返回值返回一个指针;
- (5) 这个指针指向一个 int 类型。

表 1 声明语句的可视化表示常用图元符号

图元序号	图元	实现函数
1		Draw_PointOrType(int &factor, CDC *pdc)
2		Draw_Array(int &factor, int sizearray, CDC *pdc)
3		Draw_Func(int &factor, string arguments, CDC *pdc)
4		Draw_Func(int &factor, string arguments, CDC *pdc)

基金项目: 浙江省自然科学基金资助项目“基于最小信息集的程序理解信息抽取模型研究”(93531068)

作者简介: 古 辉(1956—), 男, 教授, 主研方向: 程序理解技术, 智能信息处理; 游联启, 硕士研究生

收稿日期: 2011-08-08 **E-mail:** gh@zjut.edu.cn

表 2 构成描述语句的基本词汇列表

语句序号	语句	实现函数
1	___是一个	Link_Variable(string& wdescription, string arguments)
2	指针; 这个指针指向一个	Link_Point_1(string& wdescription)
3	指针; 这些指针指向一个	Link_Point_2(string& wdescription)
4	数组; 这个数组的每个元素都是一个	Link_Array_1(string& wdescription)
5	具有___个元素的数组; 这个数组的每个元素都是一个	Link_Array_2(string& wdescription, int sizearray)
6	函数; 函数的返回值返回一个	Link_Func_1(string& wdescription)
7	参数为___的函数; 函数的返回值返回一个	Link_Func_2(string& wdescription, string arguments)
8	___类型。	Link_Type(string& wdescription, string arguments)

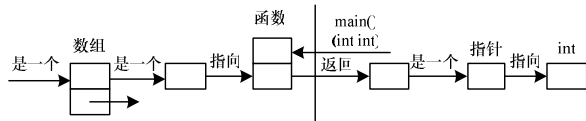


图 1 声明语句可视化表示

4 算法设计

4.1 词法分析

在对声明语句进行可视化表示和语义描述之前, 需要对源程序进行词法分析。其中, 词法标志的结构为:

```
struct Token{int syn;//expression code
string text;//expression content};
```

将声明语句的词法标志存储在向量 TokenList 中。

4.2 声明合法性判断

在对声明语句进行可视化表示和语义描述之前, 还需要判断这个声明语句是否合法。如果声明语句不合法, 则终止对这个错误的声明进行分析。有些错误很明显, 例如: 括号不匹配、声明中有不能识别的符号(如@、\$), 但有些错误却比较隐蔽, 例如: int p[]();是非合法的, 因为数组元素必须具有相同的长度, 但不同的函数显然可能具有不同的长度。int p0[]; 也是非法的, 因为函数只能返回标量值, 不能返回数组^[5]。

由于篇幅有限, 算法 1 只给出判断声明语句是否合法的部分算法。

算法 1 合法性算法 IsLegal

```
procedure IsLegal (TokenList) return bool
input: TokenList : a vector of Token
begin
position : integer
while position < TokenList.size do
begin
if 11 == TokenList[position].syn &&
12 == TokenList[position+1].syn
return false;
else if position 大于标识符位置
&& 13 == TokenList[position].syn
&& 10 == TokenList[position+1].syn
return false;
/*符号 “[” 的 syn 等于 10;
符号 “]” 的 syn 等于 11;
符号 “(” 的 syn 等于 12;
符号 “)” 的 syn 等于 13 */
position++;
end
return true;
end
```

4.3 可视化表示和语义描述算法

算法 2 的功能是实现声明语句到可视化图形和语义描述的转换, 其中包括 6 个子算法, 算法 2 的函数流程(即子算法 1)如图 2 所示。

算法 2 可视化表示和文字描述

变量

- factor : integer 初始值为 0
- n_needdraw : integer 初始值为 0
- n_drawn : integer 初始值为 0
- position : integer 初始值都为 0
- wdescription: string 初始值为空
- IsArray : bool 初始值为 false

输入 TokenList : a vector of Token

子算法 1 声明语句合法

- (1)当前指针指向变量在向量中的下一个位置;
- (2)在坐标(0, 100)上输出变量名;
- (3)调用函数 Link_Variable(factor, TokenList[position]. text)并且 factor 加 1。

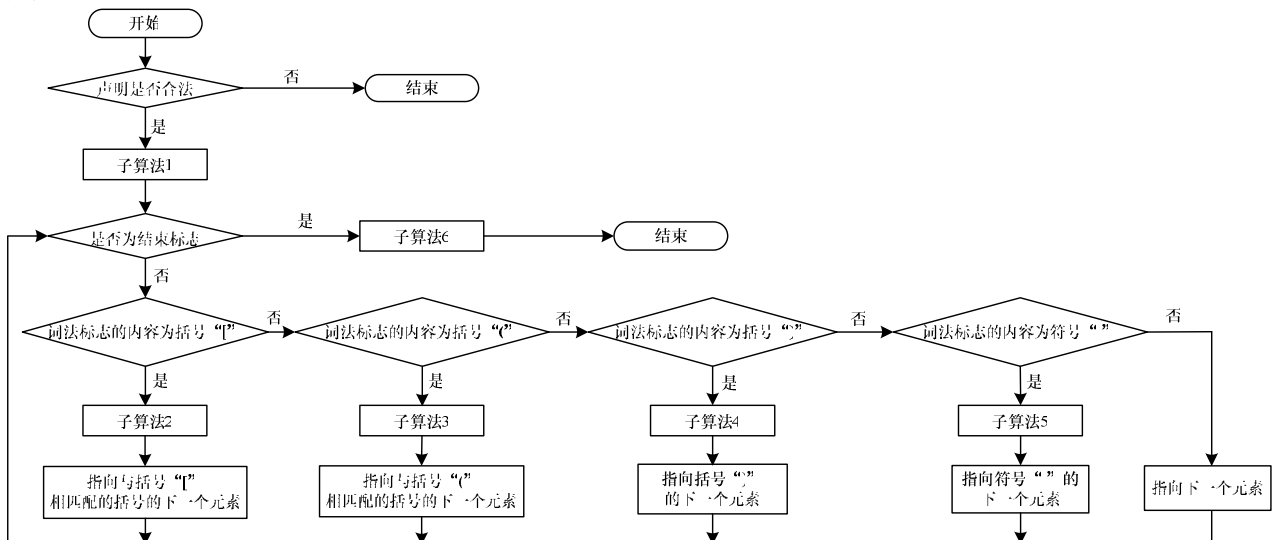


图 2 可视化表示和语义描述函数流程

