

基于二叉树的 PLC 梯形图转化为指令表的算法

黄晶晶, 陈文芄, 叶军君, 林志民

(厦门大学 物理与机电学院 福建 厦门 361005)

摘要: 针对 PLC 梯形图利于编程, 但不利于被执行的特点, 利用二叉树这个中间工具, 提出了一种将梯形图转化为指令表的转化算法, 转化以梯级为单位, 采取“从左至右, 从上至下”的扫描顺序构造二叉树结构, 然后对树进行后序遍历得到指令表。并简要介绍了梯形图编辑软件的各个组成部分, 以及梯形图的编辑环境; 描述了算法的主要数据结构。

关键词: 梯形图; 指令表; 二叉树; 后序遍历

中图分类号: TP311

文献标识码: A

文章编号: 1004-373X(2010)04-125-03

Algorithm from PLC Ladder Diagram to Instruction Table Based on Binary-Tree

HUANG Jingjing, CHEN Wenxiang, YE Junjun, LIN Zhimin

(School of Physics and Mechanical & Electrical Engineering, Xiamen University, Xiamen, 361005, China)

Abstract: Owing to PLC ladder diagram's traits of being programmed easily and processed inconveniently, an algorithm of transformation from PLC ladder diagram to instruction table are presented using the tool of binary tree. Based on a single rung as a conversion unit, the binary-tree data structure are created following the scanning order of from left to right, from top to bottom through the conversion process. Then the instruction table are obtained through the post-order traversal of the binary-tree. In addition, various components of PLC ladder diagram software and the editing environment of PLC ladder diagram are introduced, and main data structure of the algorithm are described simply.

Keywords: ladder diagram; instruction table; binary-tree; post-order traversal

0 引言

国际电工委员会制定的工业控制编程语言标准^[1]规定了五种 PLC 的编程语言: 梯形图(LD), 指令表(IL), 功能模块图语言(FBD), 顺序功能流程图语言(SFC)以及结构化文本语言(ST)。其中, 以梯形图最为常用^[2], 因为它与继电器线路类似, 与电气操作原理图相对应, 而电气设计人员对继电器控制较为熟悉。即使是 PLC 编程的初学者, 也因为 PLC 梯形图的直观和易于掌握而选择用梯形图来编程, 但是梯形图不能被 PLC 的 CPU 识别。指令表编程语言是与汇编语言类似的一种助记符编程语言, 和汇编语言一样有操作码和操作数组成。汇编语言是与计算机硬件联系非常紧密的低级语言, 易于被通用 PC 的 CPU 执行, 指令表也有类似特点, 它也是比较接近机器码的语言, 但是使用者要学习和记忆 PLC 的编程指令。如果把梯形图的逻辑通过人工翻译的模式转化成为指令表, 这大大加大了开发人员的工作量(事实上这已经是用指令表编程了), 而且较为复杂。基于此, 为了方便 PLC 应用程序开发人

员, 研究 PLC 梯形图转化成为指令表的算法, 开发出一种指令表自动生成模块, 也就是梯形图转换成指令表的编译模块, 是非常有意义的。

1 梯形图与指令表

1.1 梯形图

梯形图与继电器控制线路图非常类似, 不过梯形图本身也有自己的设计规则^[3]: 梯形图程序应按自下而上, 从左至右的方式进行编制; 触点一般不画在垂直分支线上; 合理的组织串联电路块的并联和并联电路块的串联, 比如, 在有几个串联回路相并联时应将触点最多的那个串联回路放在该梯级的最上面, 而在有几个并联回路相串联时应将触点最多的并联回路放在梯形图的最左面(靠近左母线的一端); 输入触点只允许画在输出线圈的左边等。

1.2 指令表

指令表其实就是有逻辑的指令序列, 类似于汇编语言。各个大厂商的 PLC 有其自己的指令集, 不过大同小异, 只是命名方式和各厂提供的功能指令不同。梯形图转化成指令表后指令的排列顺序都保持一致, 正是基于此, 梯形图与指令表之间一一对应, 可以相互转化。

2 梯形图的编辑环境

梯形图的编辑要有一个平台, 采用 VC++ 作为开发工具, 搭建一个梯形图的编辑环境^[4], 也就是开发出一个 PLC 编程软件, 这个软件平台的搭建必须界面友好, 还要利于编译。

整个软件系统分为三个子窗口, 有梯形图编辑窗口, 指令表生成窗口, 编译信息提示窗口; 整个软件分为四个模块, 分别是梯形图编辑模块, 编译模块, 文件管理模块, 仿真模块, 其中, 文件管理模块主要用于用户编制的梯形图的保存和读取^[5]。

为了方便绘制梯形图, 将梯形图的绘制窗口设计成为一个网格区, 每个网格可以画一个元件, 水平连接线是一个特殊元件, 它也属于元件的范畴, 但是它不是图符, 也就是不属于后面所说的节点类的对象。梯形图的绘制主要是元件的选择和连接。

梯形图的编辑环境和梯形图的编译是联系非常紧密的, 好的编辑环境非常利于编译和排查语法错误, 将梯形图的绘制窗口设计成网格区就是利于编译的一种做法, 因为在扫描过程中分成一个个单元进行扫描, 这样做利于设计扫描算法。

3 梯形图的数据结构

3.1 二叉树的构造

通过对 PLC 梯形图程序设计规则的研究可知, 用梯形图编制程序的过程实际上就是用梯形图图符来表示操作指令, 用图符的串并联关系以及位置顺序来表示操作指令之间逻辑关系的过程^[6]。

既然梯形图与指令表之间是一一对应的, 那么在将 PLC 梯形图转化为指令表的过程中势必会有规律可循。通过手工翻译大量的梯形图后(可以对照梯形图 1 以及它的指令表 1, 指令表命名方式基于三菱 PLC 系统^[7]) 发现, 梯形图中每个梯级的图符的逻辑关系是一种层次结构, 而这种层次结构容易用数据结构中的树来表达, 把每个梯级用二叉树这种结构表达出来后, 那么整个梯形图就是多个二叉树的集合, 也就是森林。现在的问题是怎么样来构造这棵二叉树。

利用面向对象的编程思想, 由于图符是梯形图的基本构成元素, 先把图符抽象成一个类, 也就是一个节点对象的类(这里的节点就是指图符)。这个对象的类包含数据和一些函数, 数据代表要存储的节点的内容, 下面是这个类的数据成员的描述:

```
class Node
{private:
    int x, y;           // 节点所在窗格在绘图区域的位置
```

```
bool BehindBranch;    // 节点后面是否有向下线
bool FrontBranch;     // 节点前面是否有向上线
char name[ 10];       // 与该图符对应的指令
int index;            // 图符的索引号
...
}
```

除了节点类以外, 还需要一个表示梯级的类, 由这个类实例化的对象含有一个梯级所有的节点, 由于梯级是二叉树结构, 这个类取名 BinaryTree 类。这个类中提供了一些函数, 它们用来扫描节点, 合并树和构造树结构。

```
class BinaryTree
{
    ...
public:
    int ScanNode();    // 从左至右, 从上至下扫描图符
    bool IsBehindBranch(); // 判断是否扫描到向下线
    bool IsFrontBranch(); // 判断是否找到节点向上线
    ...
}
```

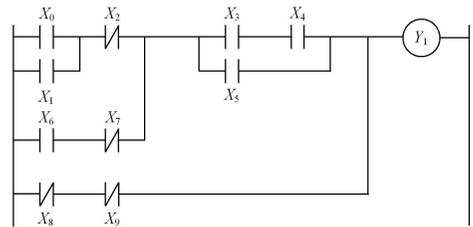


图 1 复杂串并联关系的梯形图

表 1 梯形图 1 的指令表

地址	指令	操作数	地址	指令	操作数
00	LD	X ₀	08	OR	X ₅
01	OR	X ₁	09	ANB	
02	ANI	X ₂	10	LDI	X ₈
03	LD	X ₆	11	ANI	X ₉
04	ANI	X ₇	12	ORB	
05	ORB		13	OUT	Y ₁
06	LD	X ₃	14	END	
07	AND	X ₄			

构造二叉树的主要步骤是:

(1) 从左至右, 从上至下扫描梯级, 若未发现节点后面有向下线就一直向右扫描, 直到发现向下线时停止, 记下向下线的位置和扫描过的节点, 若扫描过的节点有多个, 每扫描到两个就新建一个 AND 节点, 它是这两个节点的父节点; 新建的这个 AND 节点继续参与构造二叉树, 作为后序 AND 节点(如果有的话) 的子节点。比如有三个节点, 解决方法是, 前两个节点新建一个 AND 节点, 再建一个 AND 节点, 它是刚才 AND 节点和第三个节点的父节点, 如图 2 所示。

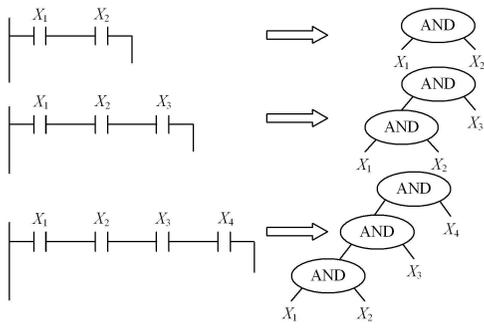


图 2 构造二叉树结构

(2) 从当前行的下一行, 向下线的位置开始向左扫描, 是否有节点存在, 若有则新建一个 OR 节点, 继续向左扫描, 直到发现节点之前有向上线存在, 如果有, 这一层扫描结束, 扫描过的节点如果有两个或是多于两个, 仍然按照第 1 步的方法建立 AND 节点, OR 节点是上一行已构造好的子树的根节点和新建立起来的 AND 节点的父节点。

(3) 若当前行的下一行还有向下线, 以当前行的下一行作为当前行, 递归进行第 (2) 步。

(4) 第(2)步中, 从当前行的下一行, 向下线的位置开始向左扫描时, 若左边没有元件存在, 说明向下线连接的是右边的元件, 此时也建立一个 AND 节点, 它是右边并联块和已扫描过的节点建立起来的子树的跟节点的父节点, 如图 3 所示二叉树中的第三层左边的 AND 节点。

(5) 节点排列的顺序需要注意, 最先扫描到的节点作为新建的 AND 节点或是 OR 节点的左子节点。

这样就建立起梯形图的二叉树结构, 二叉树只是表示了图符间的逻辑关系, 不表示具体的指令含义。

所有图符节点都是所构成的二叉树的叶子节点, 而 AND 节点或 OR 节点都是非叶节点, AND 节点和 OR 节点分别表示左右子树的串联和并联关系^[8]。

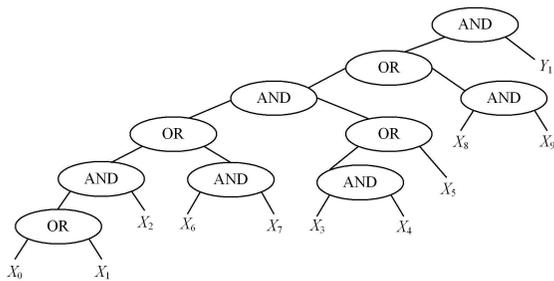


图 3 梯形图 1 的二叉树结构

3.2 二叉树转化为指令表

经过扫描和错误检查后, 建立起类似图 3 所示的梯级的二叉树结构后, 对二叉树的遍历即可转化为指令表。根据先前构造二叉树时所建立起来的顺序可知, 树的遍历顺序是后序遍历^[9], 也就是先访问左子节点, 然后访问右子节点, 最后是根节点。对图 3 的二叉树结构

进行遍历所得到的原始指令表如表 2 所示, 对比表 1 可知, 它还有一些需要修改的地方。比如, 由于二叉树只是表示了图符间的逻辑关系, 不表示具体的指令含义^[10], 指令的含义存储在指令所属的 Node 对象中的 name 成员数据中, 对于指令 LD X1, 若其后面的指令是 AND 或 OR, 将指令合并成一条指令 AND X1 或 OR X1, 表 2 中地址为 01 和 02 的指令就需要合并成一条 OR X1 的指令; 对于两个并联块的串联或是两个串联块的并联需要将指令改为 ORB 或是 ANB, 如表 2 所示中地址为 08, 14, 18 的指令; 对于最后一条指令, 如表 2 中的 AND 节点, 也就是二叉树的根节点, 它没有实际意义, 将其舍去。

表 2 遍历图 3 所得的原始指令表

地址	指令	操作数	地址	指令	操作数
00	LD	X ₀	11	AND	
01	LD	X ₁	12	LD	X ₅
02	OR		13	OR	
03	LDI	X ₂	14	AND	
04	AND		15	LDI	X ₈
05	LD	X ₆	16	LDI	X ₉
06	LDI	X ₇	17	AND	
07	AND		18	OR	
08	OR		19	OUT	Y ₁
09	LD	X ₃	20	AND	
10	LD	X ₄			

4 编译模块的流程图

对于每一个梯级, 首先经过扫描, 在进行编译, 看其是否有语法错误; 如果有, 在编译信息提示窗口输出提示信息, 如果没有错误, 说明通过了编译, 利用所设计的算法建立起用户梯形图的二叉树森林并保存, 然后对二叉树森林进行遍历, 得出原始的指令表, 最后对指令表做相应的修改及优化得出最终指令表, 并在指令表生成窗口显示出来。如图 4 所示。

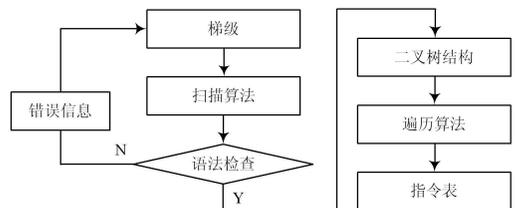


图 4 编译算法的流程图

5 结语

这里主要探讨梯形图转化为指令表的算法, 借助二叉树这个中间工具, 从图 1, 表 1 及表 2 给出的指令表实例可以看出, 该算法能够正确的将梯形图转化为指令

(下转第 134 页)

```

traverse(l); // 访问单链表
return 0;
}

```

这样,就可以正确建立单链表。因为经过初始化,已经建立了一个空的带头结点的单链表,creat 函数将实参 l 的值传递给形参 la, la 接受了实参 l 的值,在 creat 函数中修改实参 l(头指针)所指单元内的内容,即修改实参 l 所指单元的值,而并没有改变实参 l 的值。

2 结 语

从以上单链表建立算法的详细分析可见,关键问题还是需要深刻理解 C/C++ 程序设计语言中的参数传递方式、变量有效范围、指针等,并能灵活应用。这样在实际应用编程中就不会出错,也不会调试软件过程中出现一些莫名其妙的错误。

参 考 文 献

- [1] 严蔚敏,吴伟民. 数据结构(C语言版)[M]. 北京:清华大学出版社,2006.
- [2] 耿国华. 数据结构——C语言描述[M]. 北京:高等教育出版社,2006.
- [3] [美] Robert L Kruse, Clovis L Tondo, Bruce P Leung. Data Structures & Program Design in C[M]. Second Edition. Prentice-Hall International Inc., 1998.

作者简介 谢娟英 女,1971年出生,陕西西安人,博士研究生,副教授。主要研究方向为模式识别、智能信息处理等。

(上接第127页)

表,方便了PLC编程人员的应用程序开发过程。但是编译的效率方面有待提高,因为如果梯形图串并联关系复杂,二叉树的层次将较多且不平衡(多为左子树层次远大于右子树),遍历花费的时间较多,现在的CPU运算能力强,主频一般都能达到1GHz,所以编译的时间不是问题。除了正确性,编译模块还给出了错误信息报告,而错误报告和调试等的可用性方面也是很重要的。

参 考 文 献

- [1] Karl-Heinz J, Tiegelkamp M. IEC61131-3: Programming Industrial Automation Systems[M]. Berlin: Springer-Verlag Company, 2001.
- [2] 白艳艳,陈卫福,于春娜. 开放式数控系统中软PLC的研究开发[J]. 机械管理开发,2005(3): 24-27.
- [3] 钟肇新,范建东. 可编程控制器原理及应用[M]. 广州:华南

作者简介 黄晶晶 男,1986年出生,湖北荆州人,硕士研究生。主要研究方向为计算机控制技术、嵌入式控制。

陈文芾 男,1955年出生,福建漳州人,教授,博士生导师。主要研究方向为电子技术、计算机控制、智能仪器仪表、工业自动控制。

叶军君 男,1962年出生,福建莆田人,副教授。主要研究方向为检测及数控应用技术。

- [4] [美] William Ford, William Topp. Data Structures with C++ [M]. Prentice-Hall International Inc., 1997.
- [5] Wikipedia, The freeencyclopedia Linked List [EB/OL]. <http://en.wikipedia.org/wiki/Linked-list>, 2008.
- [6] 张敬敏,王培崇,路凤佳. 道路网络中的移动对象索引方法研究[J]. 计算机工程与应用,2009,45(12): 144-146.
- [7] 孙玉强,顾玉宛,张聪品,等. 基于PRAM模型的二叉树A序列并行算法的研究[J]. 计算机科学,2009,36(3): 256-257,294.
- [8] 刘小龙,杨维芳. 基于最小距离简单多边形的Delaunay三角剖分算法[J]. 计算机工程与设计,2009,30(5): 1270-1271,1275.
- [9] 王小兵,段振华. 框架投影时序逻辑程序设计语言中的指针[J]. 西安电子科技大学学报:自然科学版,2008,35(6): 1069-1074.
- [10] 李广元,唐稚松. 时序逻辑语言XYZ/E中指针的形式化表示与验证[J]. 软件学报,2000,11(3): 285-292.
- [11] 何煦岚,何晓岚. 基于多链表结构的嵌入式系统内存管理[J]. 计算机应用与软件,2008,25(4): 58-59,81.
- [12] [美] Nell Dale, Chip Weems, Mark Heading. Programming and Problem Solving with C++ [M]. Third Edition. Jones and Bartlett Publishers, 2003.
- [13] [美] Brian W Kernighan, Dennis M Ritchie. The C Programming Language[M]. Second Edition. Prentice-Hall International, Inc., 2008.

理工大学出版社,2008.

- [4] 章德平. 软件PLC开发平台的设计和实现[D]. 武汉:武汉理工大学,2007.
- [5] 潘勇. 数控系统中梯形图编程软件的研究及实现[D]. 武汉:武汉理工大学,2005.
- [6] 莫易敏,章德平,周哲. PLC梯形图转化为指令表算法及实现[J]. 控制工程,2006,13(6): 573-576.
- [7] 龚仲华,史建成,孙毅. 三菱FX/Q系列PLC应用技术[M]. 北京:人民邮电出版社,2006.
- [8] 葛芬,吴宁. 基于AOV图及二叉树的梯形图与指令表互换算法[J]. 南京航空航天大学学报,2006,38(6): 754-758.
- [9] Robert Lafore. Java 数据结构和算法[M]. 2版. 北京:中国电力出版社,2004.
- [10] Lee G B, Han Z D, Lee J S. Automatic Generation of Ladder Diagram with Control Petri Net[J]. Journal of Intelligent Manufacturing, 2004,15(2): 245-252.