

科技期刊信息处理类流程图的编辑加工

童天添

《江苏科技大学学报》编辑部, 212003, 江苏镇江

摘要 参照现行的 GB/T 1526—1989 和关于用流程图表示算法的方法的相关文献,发现不同类型流程图,可使用的流程图符号范围和约定存在差异。在科技期刊文稿编辑加工过程中,针对流程图的规范问题,首先根据稿件内容,将该流程图按算法流程图、数据流程图、程序流程图、系统流程图做细分,再采用不同图的方法对流程图做编辑加工,可提高流程图的编辑加工质量。文中以某程序流程图的编辑加工为例,分析该程序流程图的编辑加工方法。

关键词 流程图符号;流线;判断符号;规范;程序流程图

Editing of information processing flowcharts in sci-tech journals// TONG Tianian

Abstract According to the existing GB/T 1526-1989 and other literatures, we find that symbols and conventions are different in various kinds of flowchart. As for normalizing of flowchart when we are editing flowcharts in sci-tech journals, firstly we subdivide the flowcharts into algorithm, data, program and system flowchart according to the content of the paper, and then we use different methods to edit the flowcharts, which could improve the quality of editing. As last, we show an example of editing program flowcharts, and analyze the method of editing program flowcharts.

Keywords flowchart symbol; streamline; decision symbol; standard; program flowchart

Author's address Editorial Department of Journal of Jiangsu University of Science and Technology, 212003, Zhenjiang, Jiangsu, China

DOI:10.16811/j.cnki.1001-4314.2017.01.010

流程图是对于一个问题的定义、分析或解法的图形表示,图中用各种符号来表示操作、数据、流向以及装置等。流程图包括确定含义的符号、简单的说明性文字和各种连线,被广泛用于描绘各种类型的信息处理问题^[1-3]。随着计算机学科以及与计算机学科有交叉的其他学科的迅速发展,流程图在诸多领域得到广泛应用。计算机流程图有其规范的表达与约定,科技期刊编辑,更加应该按照标准与规范,严格根据流程图中可使用的各种符号,以及使用这些符号的约定,有原则地对流程图做编辑加工,以增强图的可读性。

美国国家标准化协会(American National Standards Institute)规定了一些常用流程图符号,已为世界各国程序人员使用。文献[4]介绍了怎样通过流程图

来表示一种算法。此外,1973年美国学者 Nassi 和 Shneiderman 提出了 N-S 结构化流程图,去掉了带箭头的流程线,将全部算法写在一个矩形框内,在该框内还可以包含其他的从属于它的框。文献[5-6]从编辑角度出发,分析了在科技期刊中出现的、用流程图表示算法的常见问题,有效促进了编辑人员正确掌握算法流程图的编辑加工技能,保证了算法流程图的科学性和规范性。流程图不只局限于表示算法,还包括数据流程图、程序流程图、系统流程图等。关于这些不同类型流程图的编辑加工的相关研究却较少。

流程图存在问题会严重影响论文的科学性、逻辑性和规范性^[7]。本文对照现行的 GB/T 1526—1989《信息处理 数据流程图、程序流程图、系统流程图、程序网络图和系统资源图的文件编制符号及约定》,发现不同类型流程图,可使用的流程图符号范围和约定存在差异。在科技期刊编辑加工过程中,对于流程图类插图,首先应根据稿件内容,将该流程图进行细分,再采用不同图的处理方法来加工流程图。文中详细分析了 1 例程序流程图的编辑加工方法。

1 流程图常见问题

1.1 不一致问题 不一致问题包括:插图内容是算法流程图,而图题却是“×××程序流程图”;插图内容是程序流程图,而图内符号却未按符号使用约定,未以端点符标示开始和结束;插图内容是数据流程图,而图内符号却未按符号使用约定,未以数据符号开始和结束,数据流程图内出现判断符号;等等。

1.2 流线可读性差 造成流线可读性差的原因包括:流线出现长线;流线出现交叉;多条流线汇集为 1 条流线时,各连接点没有相互错开;流线未对准符号的中心;流线未使用箭头来指示方向。

1.3 多出口判断符号问题 程序流程图和系统流程图的判断符号可以有多个输出,3 个以上出口的判断情况虽然少,但是却极易出错。与处理符号混乱使用、符号间未能均匀分配空间等问题比较常见^[8]。

2 流程图的标准与规范

2.1 常用符号 文献[1]等同采用 ISO 5807:1985,标准中规定了 25 种流程图符号,包括基本符号和特定

符号。常用的流程图符号如图 1 所示,符号下方配有相应的符号名。

例如判断符号,表示判断或开关类型功能,只有 1 个入口;但可以有若干个可选择的出口,在对符号中定义的条件进行求值后,有 1 个且仅有 1 个出口被激活。求值结果可在表示路径的流线附近写出。判断符号可以用于系统流程图、程序流程图,但不能用于系统资源图、程序网络图和数据流程图。

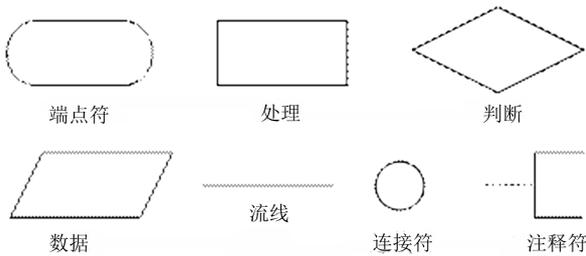


图 1 常用流程图符号

2.2 流程图的细分

2.2.1 算法流程图 算法是对操作的描述,即操作步骤。算法必须满足有穷性、确定性、有零个或多个输入、有 1 个或多个输出、有效性这 5 大特性^[9]。算法流程图用于表示算法,N-S 结构化流程图也用于表示算法。N-S 结构化流程图没有带箭头的流程线,全部算法写在一个矩形框内,在该框内还可以包含其他的从属于它的框,适合结构化程序设计。

文献[1]中未涉及用来表示算法的算法流程图,只对程序流程图、数据流程图和系统流程图做了规定。

2.2.2 程序流程图 表示程序中的操作顺序,可包括表示实际处理操作的处理符号、根据逻辑条件确定要执行路径的判断符号、表示控制流的流线符号,以及便于读写程序流程图的特殊符号。

2.2.3 数据流程图 表示求解某一问题的数据通路,图中同时规定了处理的主要阶段和所用的各种数据媒体。在处理符号的前后都应是数据符。数据流程图可包括表示数据存在的数据符号、对数据执行处理的处理符号、表示数据流的流线符号,以及便于读写数据流程图的特殊符号。数据流程图以数据符号开始和结束。

2.2.4 系统流程图 表示系统的操作控制和数据流,可包括表示数据存在的数据符号、定义执行逻辑路径的判断符号、表示对数据执行操作的处理符号、表示处理或数据媒体间数据流的流线符号,以及便于读写系统流程图的特殊符号。

3 问题分析

3.1 一致性问题 流程图的内容、符号使用、图题应

保持某类图的一致性。不同类型流程图可使用的流程图符号及其约定都是不相同的。与程序流程图相比,算法流程图是程序实现的核心与灵魂;与系统流程图相比较,程序流程图不应使用特定的数据符号;与系统流程图相比,数据流程图要求以数据符号开始和结束,且不应有定义执行逻辑路径的判断符号。

图题问题应纠正,不管是“算法流程图”“程序流程图”,还是“数据流程图”“系统流程图”,都统称“流程图”是错误的。图题的正确写法是根据图的具体内容,写成“×××算法”“×××程序流程图”“×××数据流程图”“×××系统流程图”。

在科技期刊编辑加工过程中,对于流程图类插图,首先根据稿件内容,将流程图做细分,再采用不同图的方法对流程图做编辑加工,可提高流程图的编辑加工质量。

图 2 为实现 $n!$ 功能的信息处理类流程图的对比。a 为算法流程图,表示了 $n!$ 算法的实现思想;b 为 N-S 结构化流程图,既体现了算法思想,结构也很紧凑;c 为程序流程图,表示了整个程序的操作顺序;d 为数据流程图,表示了程序中的数据走向;e 为系统流程图,表示了系统中的处理或数据媒体的工作关系,会有特定数据符号。各类流程图的内容、符号使用、约定等均符合一致性标准。

3.2 流线问题 在数据流程图、程序流程图、系统流程图中,流线可以指示数据流或控制流。连线应保持合理长度,要尽量少使用长线,并尽量避免流线的交叉。即使出现流线交叉,交叉的流线之间也没有任何逻辑关系,不对流向产生任何影响。2 条或更多的进入线可以汇集为 1 条输出线,当 2 条或更多流线汇集为 1 条流线时,各连接点应相互错开,以提高清晰度,并在必要时使用箭头表示流向。一般情况下,流线应从符号的左边或顶端进入,从右边或底端离开。它们都应对准符号的中心。只有在流程图是标准流向,即流向是从左到右和自上而下的情况下,流线可以不使用箭头;当流程不遵循标准流向时,要使用箭头来指示方向。

3.3 多出口判断符号的表达 判断符号会出现多个出口的情况,在对判断符号中定义的条件进行求值后,有 1 个且仅有 1 个出口被激活。图 3-a 不适合用于多出口的判断的表达,因为这样的符号再加上每条出口流线的条件值,使得整个图的符号分布不均匀,严重影响了流程图的可读性;图 3-b 直接从判断符号的 3 个菱形的角各引出通向其他符号的流线,适合 3 出口的判断的表达;图 3-c 从判断符号引出 1 条流线,再分支成若干条流线,这种方法适合 3 个以上的出口的情况。

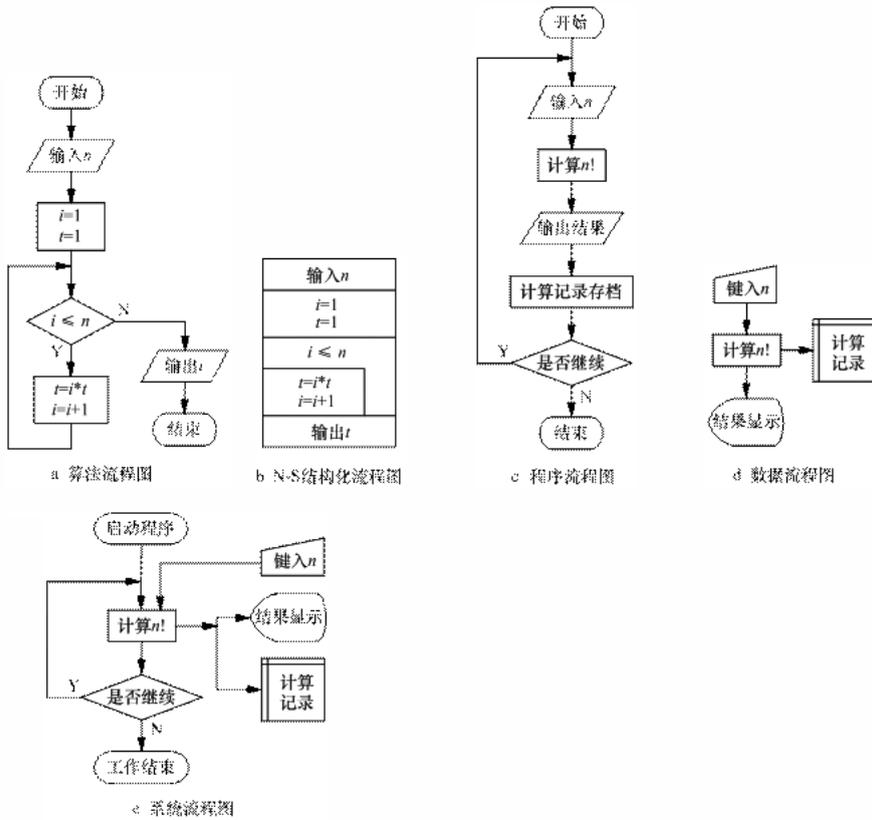


图 2 实现 $n!$ 功能的信息处理类流程图对比

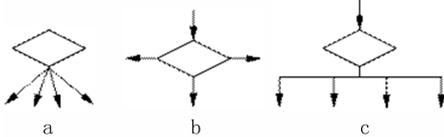


图 3 多个出口判断符号的编辑加工

根据图 4-a 的内容,确定其为程序流程图。对于程序流程图的起始与结束应使用端点符号,而不使用处理符号;对于判断功能,应使用菱形的判断符号,而不使用处理符号;判断符号中定义的条件值应写在流线附近,而不写在处理符号内;对于 4 出口情况,应从

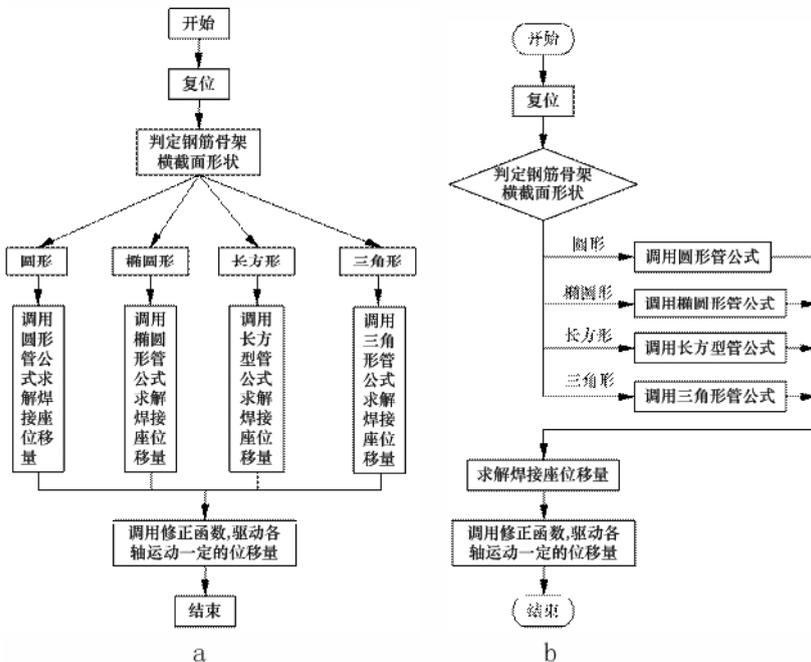


图 4 编辑加工后的程序流程