

# 过程模型的测试用例自动生成研究<sup>①</sup>

## Study of Automated process model test case Generation

朱令娴 周伯生 罗文劼 (北京航空航天大学计算机学院 北京 100083)

**摘要:** 在软件测试中,测试用例的自动生成问题近来是一个研究的热点,基于代码的测试用例的自动生成技术目前已很成熟,但基于过程模型的测试用例的自动生成技术却没有起步。为了解决这一问题,本文提出了双向过程切片的概念并使用该技术,在北航软件工程研究所开发的企事业建模系统 EPMS(Enterprise Process Modeling System) 中的过程模型建造环境 PMBE(Process Building Environment) 基础上,实现了过程模型的测试用例的自动生成技术,并通过了数个实际项目的验证。

**关键词:** 测试用例自动生成 过程模型 双向过程切片

### 1 引言

软件测试是度量收集的一种重要手段,特别是质量度量数据,它们基本都是由软件测试采集的。而在软件测试中,测试用例的自动生成问题近来是一个研究的热点,基于代码的测试用例的自动生成技术目前

化语言描述需求,从该描述提炼出测试用例的,其中文献<sup>[7]</sup>是依赖模型检测的方法生成测试用例的。但基于过程模型的测试用例的自动生成技术却没有起步,下节本文将简述自动生成过程模型的测试用例集合的必要性。

### 2 自动生成过程模型的测试用例集合的必要性

对于使用 PMBE、Rational Rose、UML Designer 或任何一种过程建模工具构建的过程模型而言,模型的构建者和模型的使用者往往不是同一个人,当过程模型中包含一些条件分支和循环流程时,模型的流程是不甚清晰的,尤其是当模型包含众多复杂的条件分支和循环流程时,模型的使用者往往难于理解或不能理解业务逻辑,更不用说对该模型进行正确性验证和优化处理了。为了解决这一问题,迫切需要建模工具同时提供自动生成测试用例集合的功能,使每个

测试用例对一个判断表达式或两个可同时访问的判断表达式(即存在某路径,使两个判断表达式都被遍历

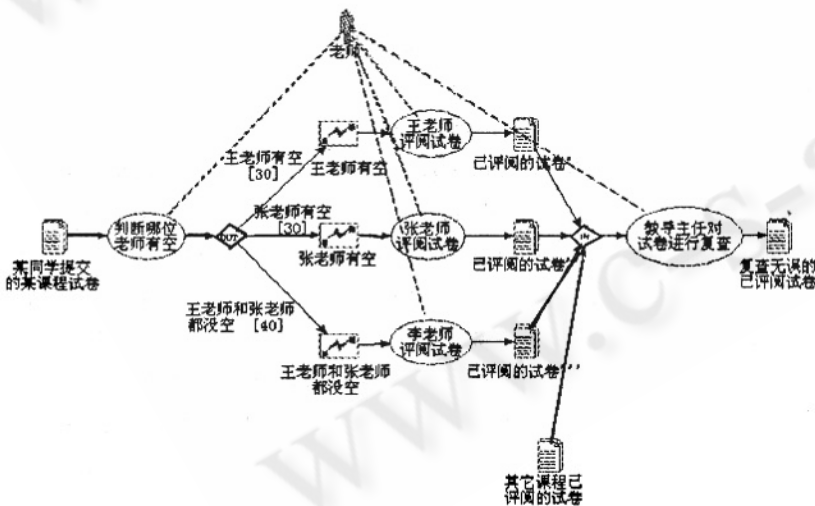


图 1 评阅试卷流程

已很成熟,它主要使用的是前向和后向程序切片技术<sup>[1][2]</sup>。基于形式规格说明的自动生成技术方面,也涌现出了大量文献<sup>[3-7]</sup>,它们基本都是使用某种形式

① 基金项目:北京市科技计划资助项目(y0105001040521)

到) 及其后跟随的某条条件分支进行遍历, 以期达到该测试用例集合对所有判断表达式的所有条件分支进行

术还没有起步的局面, 为了解决这一问题, 本文提出了双向过程切片的概念并使用该技术, 在北航软件工程

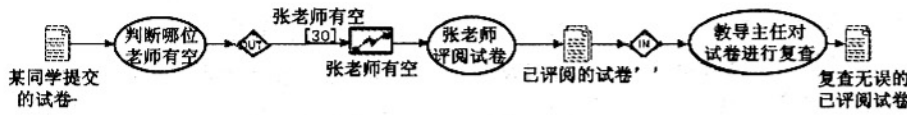


图 2 与已评阅的试卷相关的评阅试卷流程的过程切片

研究所开发的企事业建模系统 EPMS 中的过程模型建造环境 PMBE 基础上, 实现了过程模型的测试用例的自动生成技术。下节将详

述过程切片的概念。

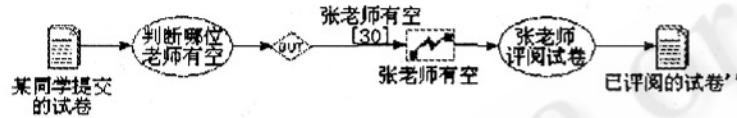


图 3 与已评阅的试卷相关的评阅试卷流程的前向过程切片

### 3 过程切片的定义

仿照 Weiser<sup>[8]</sup>、S. Horwitz<sup>[9]</sup> 给出的程序切片的定义, 本文给出过程切片的相应定义:

过程切片是感兴趣的产品(或产品集)和与该产品(或产品集)通过一条或多条顺序连接的连接符连接的构件集合(含连接符)。图 1 是一个评阅试卷流程, 该模型表示: 对某同学提交的某课程试卷, 首先察看是否王老师有空, 如王老师有空, 则由王老师进行试卷的评阅工作并产生已评阅的试卷; 如王老师没空, 察看是否张老师有空, 如张老师有空, 则由张老师进行试卷的评阅工作并产生已评阅的试卷; 如张老师没空, 由李老师进行试卷的评阅工作并产生已评阅的试卷; 对该课程和其它课程已经评阅的试卷, 教导主任对试卷进行复查并产生复查无误的已评阅的试卷。与已评阅的试卷相关的评阅试卷流程的过程切片如图 2 所示。

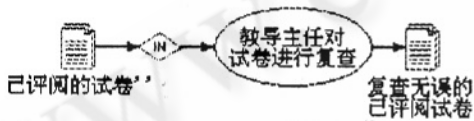


图 4 与已评阅的试卷相关的评阅试卷流程的后向过程切片

前向过程切片是感兴趣的产品(或产品集)和与该产品(或产品集)通过一条或多条顺序的指向该产品(或产品集)的连接符连接的构件集合(含连接符)。与已评阅的试卷相关的评阅试卷流程的前向过程切片如图 3 所示。

后向过程切片是感兴趣的产品(或产品集)和与该产品(或产品集)通过一条或多条顺序的背离该产品(或产品集)的连接符连接的构件集合(含连接符)。与已评阅的试卷相关的评阅试卷流程的后向过程切片如图 4 所示。

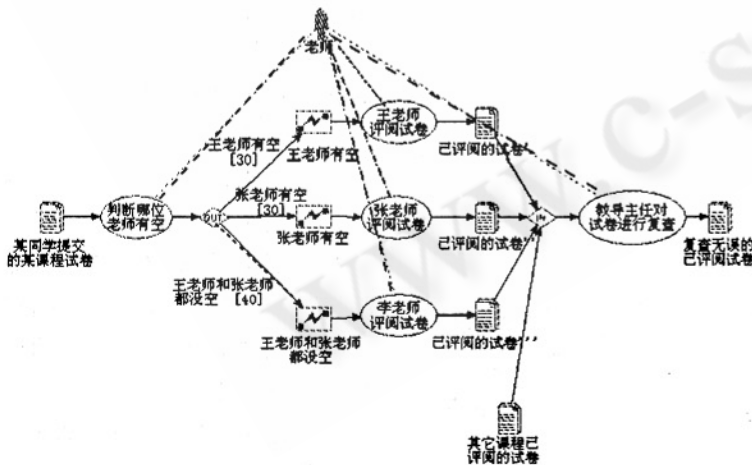


图 5 不含循环的评阅试卷流程

遍历, 这样生成的测试用例集合我们称其满足条件分支覆盖的原则。

针对目前基于过程模型的测试用例的自动生成技

双向过程切片是感兴趣的产品(或产品集)和与该产品(或产品集)通过一条或多条顺序的指向或背

离该产品(或产品集)的连接符连接的构件集合(含连接符)。与已评阅的试卷“相关的评阅试卷流程的后向过程切片如图 1 所示。可以看出双向过程切片是对应的前向过程切片和后向过程切片的并集,同对应的过程切片是等价的,反过来说,为求得通过某一产品(或产品集)的路径,我们需要了解该产品(或产品集)的过程切片,即该产品(或产品集)的双向过程切片。

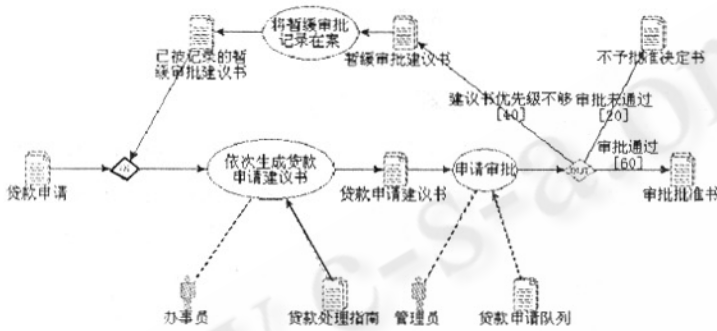


图 6 包含循环的贷款审批流程

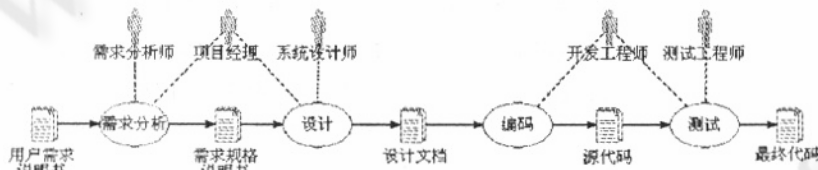


图 7 传统瀑布模型流程

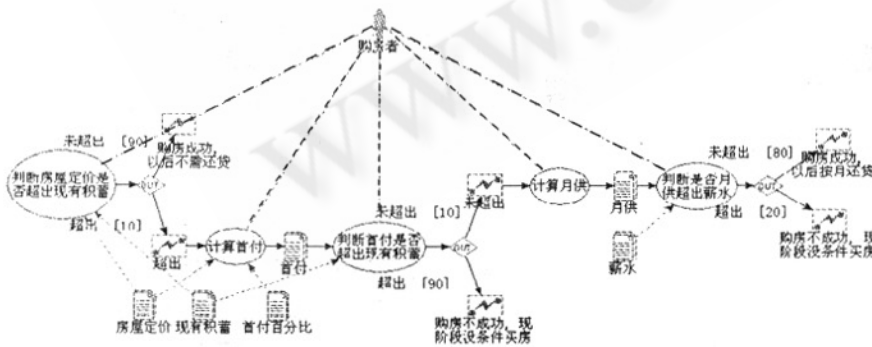


图 8 仅包含输出或的买房流程

下节本文将阐述主要依赖双向切片技术构造满

足条件分支覆盖的测试用例路径集合的方法。

### 4 基于双向切片技术构造满足条件分支覆盖的测试用例路径集合的方法

在 PMBE 环境中,路径是指 PMBE 构建的模型自模型输入(源产品,它是由当前所描述的过程之外的其它过程所生产的产品,不应有输入数据流连接到该产品<sup>[10]</sup>)产生开始至生成模型输出为止涉及的构件序列,更具体地说,它包括执行的活动、活动输入或输出的产品、活动使用的资源和时钟以及它们之间的各种连接符。它的子集叫做子路径。条件分支是指根据判断表达式的取值不同而导致模型执行不同的截至判断表达式作用范围为止的子路径。一个模型允许出现多个判断表达式和其对应的条件分支,同时一个分支也允许嵌套判断表达式和其对应的条件分支。根据执行的条件分支的不同,可以确定不同的路径。反过来说,由于本文构建的测试用例集合

是基于路径产生的,我们依据的路径必须对所有判断表达式的所有条件分支进行遍历,此即我们依据的路径必须满足条件分支覆盖。

使用 PMBE 构建模型时可以很容易地表述条件分支。判断表达式与对应的数个条件分支之间是一对多的连接关系,由上节所述,可用输入或表示;数个条件分支与相应判断表达式作用范围的后继节点之间(无循环时,如图 5 所示)或多条顺序执行的子路径与这些子路径执行后都必须处理的判断表达式之间(有循环时,如图 6 所示)是多对一的连接关系,由上节所述,可用输入或表示。

通过对图 5 和图 6 的研究发现,针对不同条

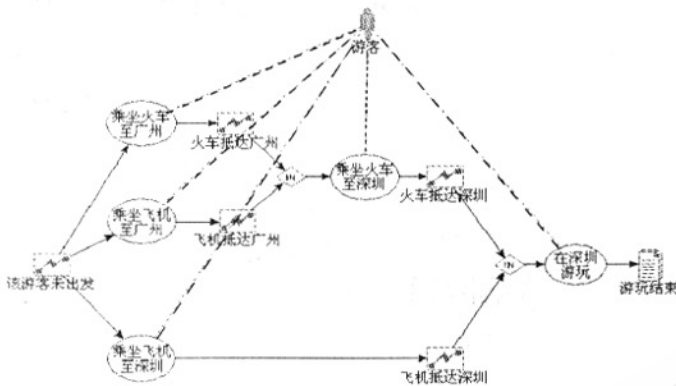


图 9 仅包含输入或的某北京游客去深圳旅游的流程

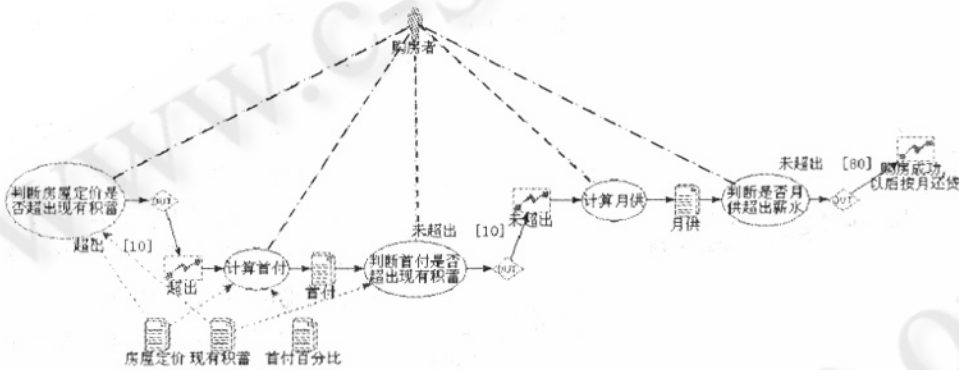


图 10 (1) 算法生成的购房流程的测试路径一

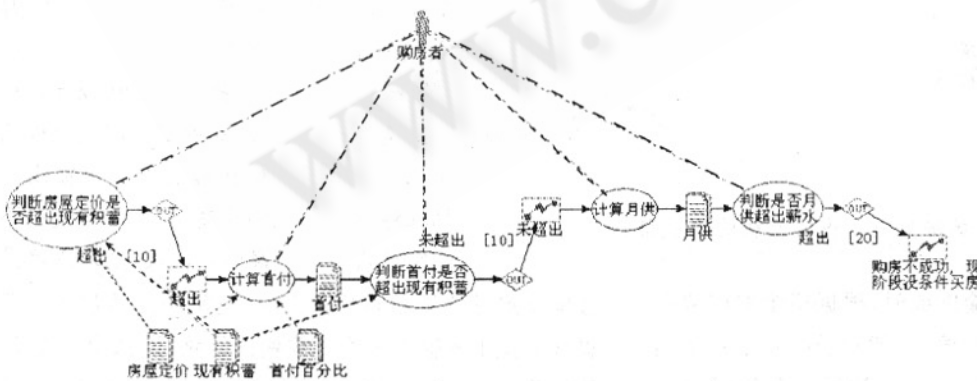


图 10 (2) 算法生成的购房流程的测试路径二

件表达式选取不同条件分支造成程序执行不同路径，也即输出或后选择哪个产品或输入或前选择哪个产品

产品进行双向遍历即可获得唯一的测试路径，它与模

作为路径经历的节点导致形成了不同的路径，那么遍历所有输入与和输入或的相应后继产品和前驱产品就可以产生满足条件分支覆盖的测试用例所产生的路径集合。

下面我们对此发现进行扩充。如模型中既不包含输出或也不包含输入或，模型执行时经过的路径只有一条，即该模型本身，如图 7 所示；当模型中仅包含输出或时（设包含的输出或为  $n$  个，各输出或对应的输出分别为  $out_1, out_2, \dots, out_n$  个），依据这些输出，最多有  $2^{out_1+out_2+\dots+out_n}$  条路径，如图 8 所示；当模型中仅包含输入或时（设包含的输入或为  $m$  个，各输入或对应的输入分别为  $in_1, in_2, \dots, in_m$  个），依

据这些输入，最多有  $2^{in_1+in_2+\dots+in_m}$  条路径，如图 9 所示；当模型中同时包含输出或和输入或时（设包含的输出或为  $n$  个，各输出或对应的输出分别为  $out_1, out_2, \dots, out_n$  个，包含的输入或为  $m$  个，各输入或对应的输入分别为  $in_1, in_2, \dots, in_m$  个），

生成的模型有不含循环和包含循环的两种，它们分别由图 5 和图 6 反映，这两种模型依据这些输出和输入，最多有  $2^{out_1+out_2+\dots+in_m}$  条路径。基于上面的分析，我们给出了如下测试路径的算法：

(1) 当模型既不包含输出或也不包含输入或时，对模型中的任一

型是等价的;

游流程,算法生成的测试路径如图 11(1)、图 11(2)、图

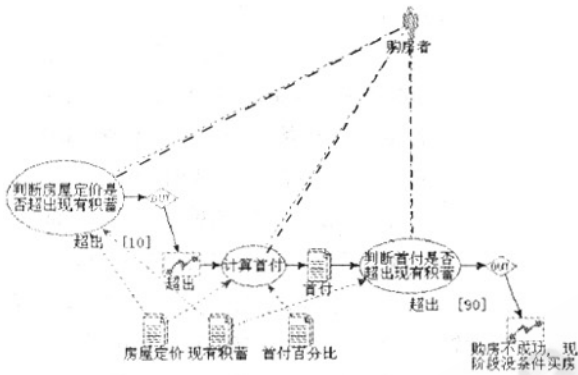


图 10 (3) 算法生成的购房流程的测试路径三

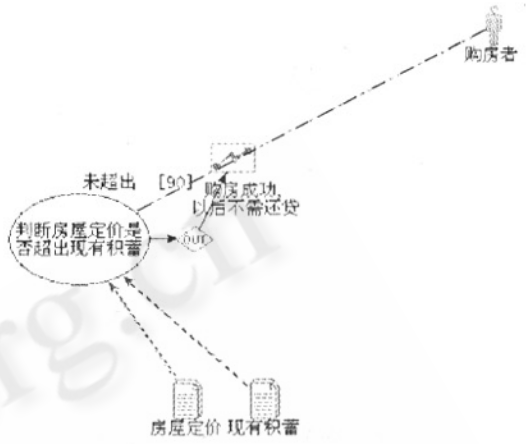


图 10 (4) 算法生成的购房流程的测试路径四

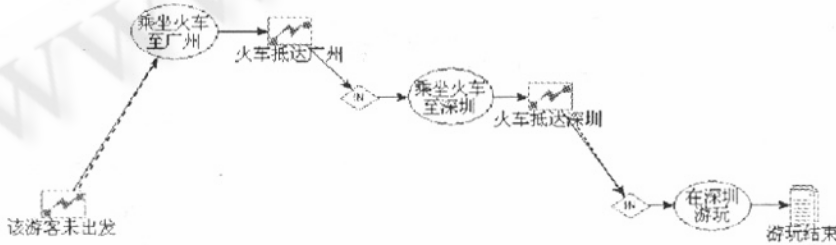


图 11 (1) 算法生成的旅游流程的测试路径一

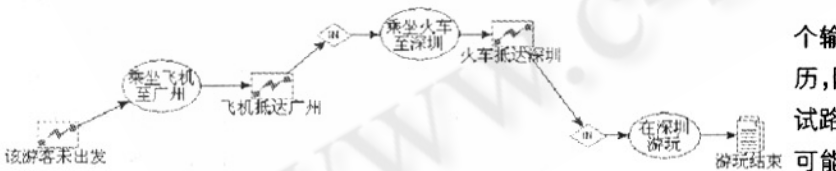


图 11 (2) 算法生成的旅游流程的测试路径二

(2) 当模型仅包含输出或时,根据每个输出或后的每个输出产品进行双向遍历,如对于图 8 所示的购房流程,算法生成的测试路径如图 10(1)、图 10(2)、图 10(3)、图 10(4) 所示。

(3) 当模型仅包含输入或时,根据每个输入或前的每个输入产品进行双向遍历,如对于图 9 所示的旅

11(3) 所示。

(4) 当模型既包含输出或,也包含输入或时,情况是很复杂的,我们对每个输入或和每个输出或构成的不同组合来考虑路径生成问题,即:

① 当输入或是输出或的前向遍历经历节点时,可能不构成循环(如图 12 的某书店发货流程所示),根据该输出或后的每个输出产品和输入或前的每个输入产品构成的不同组合进行双向遍历,限于篇幅,本文不显示算法生成的测试路径,感兴趣的读者参阅文献[215];也可能构成循环(如图 6 的贷款审批流程所示),此时根据该输出或后的每个输出产品和输入或前的每个输入产品构成的不同组合进行双向遍历外,当这个输入产品

或输出产品位于循环分支上时,还需分别添加不位于循环分支上的输入子路径或输出子路径,如对于图 6,输出产品暂缓建议书位于循环分支上,对贷款申请和暂缓建议书进行双向遍历后,应添加审批批准书所在的输出子路径构成测试路径,限于篇幅,本文在此并不显示算法生成的测试路径。

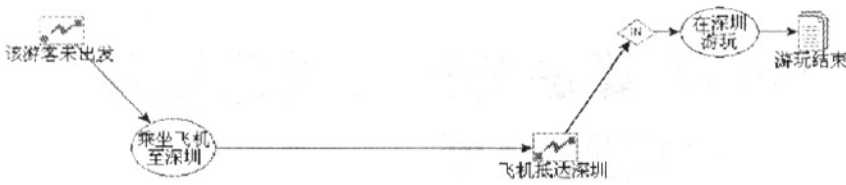


图 11 (3) 算法生成的旅游流程的测试路径三

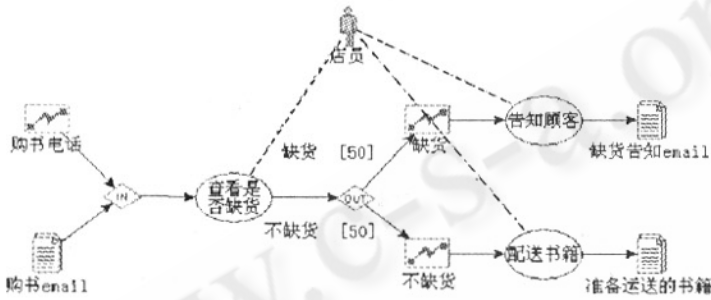


图 12 某书店发货流程

经历节点), 根据该输出或后的每个输出产品和输入或前的每个输入产品分别进行双向遍历, 如对于图 13 所示的景点游览流程, 已达景点 7 和已达景点 7 后的输入或与已达景点 2 判断路线后的输出或是不连通的, 此时应对每个相应的输出产品和输入产品分别进行双向遍历, 限于篇幅, 本文在此并不显示算法生成的测试路径。

在上述生成各条测试路径的过程中, 往往会出现重复生成某条测试路径的情况, 为了处理这种情况, 我们使用伪双向切片步骤, 在此步骤生成一条测试路径, 但并不将此路径绘制在 EPMS 系统中; 其后查询已有的测试路径集合, 看该条测试路径是否已有重复, 如没有才将此路径绘制在 EPMS 系统中, 此步骤我们才称在使用双向切片技术; 用户只

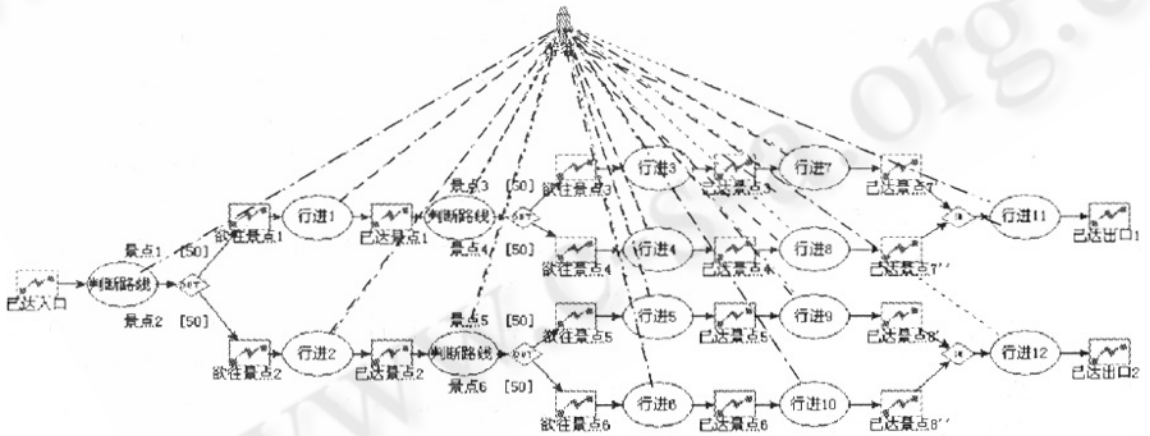


图 13 景点游览流程

② 当输入或是输出或的后向遍历经历节点时, 不可能构成循环, 根据该输出或后的每个输出产品和输入或前的每个输入产品分别进行双向遍历, 如图 5 所示的评阅试卷流程, 限于篇幅, 本文在此并不显示算法生成的测试路径。

③ 当输入或和输出或不连通时 (即输入或既不是输出或的前向遍历经历节点, 也不是输出或的后向遍

能在 EPMS 系统中看到实际生成的各路径图, 而不知道中间的各步骤。

### 5 结束语

本文提出并采用了双向过程切片和伪双向过程切片的方法, 对数个符合第三节所述范畴内的过程模型

(下转第 93 页)

皆输出了正确的满足条件分支覆盖的测试用例集合,证明其具有一定的参考和推广价值。进一步的研究需要处理更复杂的过程模型,如包含分支嵌套的过程模型。

### 参考文献

- 1 王雪莲,程序切片技术研究及其在软件测试数据生成中的应用[D],北京:北京化工大学,2005.
- 2 单锦辉,面向路径的测试数据自动生成方法研究[D].长沙:国防科学技术大学,2002.
- 3 刘玲,基于面向对象形式规格说明的测试用例生成技术[D],上海:上海大学,2004.
- 4 虞凡、覃征、贾晓琳、何坚,基于XYZ/E规范的软件测试用例自动生成方法[J],计算机工程,2005,10:76-78.
- 5 兰毓华、毛法尧、曹化工,基于Z规格说明的软件测试用例自动生成[J],计算机学报,1999,9:963-969.
- 6 S. J. Cuning and J. W. Rozenblit. Automatic Test Case Generation from Requirements Specification to Real-time Embedded Systems, 1999 IEEE International Conference on Systems, Man, and Cybernetics: 784-789.
- 7 Sanjai Rayadurgam. Automatic test-case generation from formal models of software[D]. USA: University of Minnesota, 2006.
- 8 Weiser M. Program slicing. IEEE Transactions on Software Engineering, 1984, 10(4), pp.352-357.
- 9 Christon Steindl. Program Slicing for Object-Oriented Programming Languages[D]. Austria: Johannes Kepler University Linz, 1999.
- 10 周伯生、张社英,可视化过程建模语言VPML[J],软件学报,1997,8(增刊):535-545.