(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 103218297 A
(43)申请公布日 2013.07.24

(21)申请号 201310180277.2

(22)申请日 2013.05.15

(71)申请人 百度在线网络技术(北京)有限公司
地址 100085 北京市海淀区上地十街 10 号百度大厦三层

(72)发明人 杨咏臻 丁世远 陈菊花 黄延胜

(74)专利代理机构 北京清亦华知识产权代理事务所(普通合伙) 11201
代理人 宋合成

(51)Int.Cl.
G06F 11/36(2006.01)

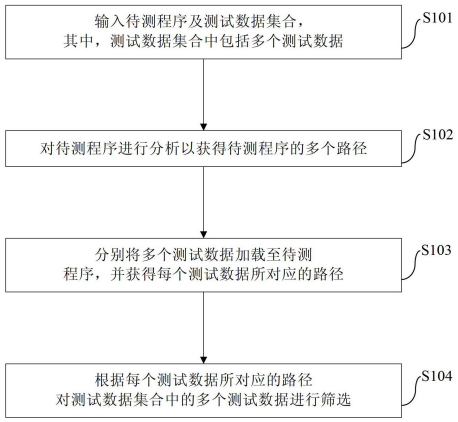
权利要求书1页 说明书7页 附图4页

(54)发明名称

测试数据的筛选方法及装置

(57)摘要

本发明提出一种测试数据的筛选方法及装置,其中所述方法包括:输入待测程序及测试数据集合,其中,测试数据集合中包括多个测试数据;对待测程序进行分析以获得待测程序的多个路径;分别将多个测试数据加载至待测程序,并获得每个测试数据所对应的路径;以及根据每个测试数据所对应的路径对测试数据集合中的多个测试数据进行筛选。根据本发明实施例的方法,无需商业软件提供接口,就可获取测试数据集合中的测试数据对应的路径,用户接入成本小,使用方便,并且筛选模块能根据每个测试数据对应的路径对测试数据集合进行筛选,能够在众多测试数据中提取出覆盖待测程序所有路径并且最精简的测试数据集合,大幅度地提高了测试速度,缩短了测试周期。



CN 103218297 A

1. 一种测试数据的筛选方法，其特征在于，包括以下步骤：

输入待测程序及测试数据集合，其中，所述测试数据集合中包括多个测试数据；

对所述待测程序进行分析以获得所述待测程序的多个路径；

分别将所述多个测试数据加载至所述待测程序，并获得每个测试数据所对应的路径；以及

根据所述每个测试数据所对应的路径对所述测试数据集合中的多个测试数据进行筛选。

2. 如权利要求 1 所述的方法，其特征在于，所述根据所述每个测试数据所对应的路径对所述测试数据集合中的多个测试数据进行筛选进一步包括：

如果两个测试数据所对应的路径相同，则进一步判断所述路径相同的两个测试数据是否具有相同的处理逻辑；

如果所述路径相同的两个测试数据具有相同的处理逻辑，则选择所述两个测试数据中的一个作为最终的测试数据。

3. 如权利要求 1 或 2 所述的方法，其特征在于，其中，每个路径中在主函数及分支节点处均设有断点，通过所述断点获得每个测试数据所对应的路径。

4. 如权利要求 3 所述的方法，其特征在于，还包括：

记录每个测试数据对应的路径，及所述对应的路径的断点信息并作为测试文件输出。

5. 如权利要求 1-4 任一项所述的方法，其特征在于，其中，所述多个测试数据之间具有数据分割点。

6. 一种测试数据的筛选装置，其特征在于，包括：

输入模块，用于输入待测程序及测试数据集合，其中，所述测试数据集合中包括多个测试数据；

分析模块，用于对所述待测程序进行分析以获得所述待测程序的多个路径；

加载模块，用于分别将所述多个测试数据加载至所述待测程序，并获得每个测试数据所对应的路径；以及

筛选模块，用于根据所述每个测试数据所对应的路径对所述测试数据集合中的多个测试数据进行筛选。

7. 如权利要求 6 所述的装置，其特征在于，所述筛选模块在两个测试数据所对应的路径相同，并且所述路径相同的两个测试数据具有相同的处理逻辑时，选择所述两个测试数据中的一个作为最终的测试数据。

8. 如权利要求 6 所述的装置，其特征在于，其中，每个路径中在主函数及分支节点处均设有断点，通过所述断点获得每个测试数据所对应的路径。

9. 如权利要求 8 所述的装置，其特征在于，还包括：

输出模块，用于记录每个测试数据对应的路径，及所述对应的路径的断点信息并作为测试文件输出。

10. 如权利要求 6-9 任一项所述的装置，其特征在于，其中，所述多个测试数据之间具有数据分割点。

测试数据的筛选方法及装置

技术领域

[0001] 本发明涉及计算机技术领域，特别涉及一种测试数据的筛选方法及装置。

背景技术

[0002] 为了提高测试速度缩短测试周期，一般会对测试用例的测试数据进行筛选，目前主要基于分支覆盖率对测试数据进行筛选，需要在安装 ccover 编译程序的前提下，在测试程序代码中插入 ccover 提供的覆盖率事件统计接口，对每条测试数据进行测试后，只要获得覆盖率增加，则说明测试程序进入新分支，即认为该测试数据有效，并存储为最终测试数据，否则认为无效。

[0003] 但是这种数据筛选方法需要安装 ccover 编译程序，使用成本较高，并且需要在测试程序中插入调用覆盖率事件统计接口，使用者需要修改部分文件，使用起来较为繁琐。

发明内容

[0004] 本发明的旨在至少解决上述技术缺陷之一。

[0005] 为此，本发明第一个目的在于提出一种测试数据的筛选方法，用户接入成本小，使用方便，并且大幅度地提高了测试速度，缩短了测试周期。

[0006] 本发明第二个目的在于提出一种测试数据的筛选装置。

[0007] 为实现上述目的，根据本发明第一方面的实施例的测试数据的筛选方法包括以下步骤：输入待测程序及测试数据集合，其中，所述测试数据集合中包括多个测试数据；对所述待测程序进行分析以获得所述待测程序的多个路径；分别将所述多个测试数据加载至所述待测程序，并获得每个测试数据所对应的路径；以及根据所述每个测试数据所对应的路径对所述测试数据集合中的多个测试数据进行筛选。

[0008] 根据本发明实施例的测试数据的筛选方法，无需商业软件提供接口，就可获取测试数据集合中的测试数据对应的路径，用户接入成本小，使用方便，并且能根据每个测试数据对应的路径对测试数据集合进行筛选，能够在众多测试数据中提取出覆盖待测程序所有路径并且最精简的测试数据集合，从而大幅度地提高了测试速度，并且缩短了测试周期。

[0009] 为实现上述目的，根据本发明第二方面的实施例的测试数据的筛选装置，包括：输入模块，用于输入待测程序及测试数据集合，其中，所述测试数据集合中包括多个测试数据；分析模块，用于对所述待测程序进行分析以获得所述待测程序的多个路径；加载模块，用于分别将所述多个测试数据加载至所述待测程序，并获得每个测试数据所对应的路径；以及筛选模块，用于根据所述每个测试数据所对应的路径对所述测试数据集合中的多个测试数据进行筛选。

[0010] 根据本发明实施例的测试数据的筛选装置，无需商业软件提供接口，就可获取测试数据集合中的测试数据对应的路径，用户接入成本小，使用方便，并且能根据每个测试数据对应的路径对测试数据集合进行筛选，能够在众多测试数据中提取出覆盖待测程序所有路径并且最精简的测试数据集合，从而大幅度地提高了测试速度，并且缩短了测试周期。

[0011] 本发明附加的方面和优点将在下面的描述中部分给出，部分将从下面的描述中变得明显，或通过本发明的实践了解到。

附图说明

[0012] 本发明上述的和 / 或附加的方面和优点从下面结合附图对实施例的描述中将变得明显和容易理解，其中 ：

[0013] 图 1 为根据本发明一个实施例的测试数据的筛选方法的流程图 ；

[0014] 图 2 为根据本发明一个具体实施例的测试数据的筛选方法的流程图 ；

[0015] 图 3 为根据本发明另一个实施例的测试数据的筛选方法的流程图 ；

[0016] 图 4 为根据本发明一个实施例的测试数据的筛选装置的结构框图 ；

[0017] 图 5 为根据本发明另一个实施例的测试数据的筛选装置的结构框图。

具体实施方式

[0018] 下面详细描述本发明的实施例，实施例的示例在附图中示出，其中自始至终相同或类似的标号表示相同或类似的元件或具有相同或类似功能的元件。下面通过参考附图描述的实施例是示例性的，仅用于解释本发明，而不能理解为对本发明的限制。相反，本发明的实施例包括落入所附加权利要求书的精神和内涵范围内的所有变化、修改和等同物。

[0019] 在本发明的描述中，需要说明的是，除非另有明确的规定和限定，术语“相连”、“连接”应做广义理解，例如，可以是固定连接，也可以是可拆卸连接，或一体地连接 ；可以是机械连接，也可以是电连接 ；可以是直接相连，也可以通过中间媒介间接相连。对于本领域的普通技术人员而言，可以具体情况理解上述术语在本发明中的具体含义。此外，在本发明的描述中，除非另有说明，“多个”的含义是两个或两个以上。

[0020] 流程图中或在此以其他方式描述的任何过程或方法描述可以被理解为，表示包括一个或更多个用于实现特定逻辑功能或过程的步骤的可执行指令的代码的模块、片段或部分，并且本发明的优选实施方式的范围包括另外的实现，其中可以不按所示出或讨论的顺序，包括根据所涉及的功能按基本同时的方式或按相反的顺序，来执行功能，这应被本发明的实施例所属技术领域的技术人员所理解。

[0021] 下面参考附图描述根据本发明实施例的测试数据的筛选方法。

[0022] 一种测试数据的筛选方法，包括以下步骤 ：输入待测程序及测试数据集合，其中，测试数据集合中包括多个测试数据 ；对待测程序进行分析以获得待测程序的多个路径 ；分别将多个测试数据加载至待测程序，并获得每个测试数据所对应的路径 ；以及根据每个测试数据所对应的路径对测试数据集合中的多个测试数据进行筛选。

[0023] 图 1 为根据本发明一个实施例的测试数据的筛选方法的流程图。

[0024] 如图 1 所示，根据本发明实施例的测试数据的筛选方法包括下述步骤。

[0025] S101，输入待测程序及测试数据集合，其中，测试数据集合中包括多个测试数据。

[0026] 在本发明的一个实施例中，可通过 GNU 程序调试器 GDB（The GNU Project Debugger）对待测程序进行测试并对测试数据集合进行筛选，首先需要将待测程序及测试数据集合输入 GDB 中，通过 Linux 管道技术与 GDB 的机器界面 MI(Machine Interface) 接口建立连接后，根据命令可直接调用待测程序和测试数据集合中的多个测试数据。

[0027] S102，对待测程序进行分析以获得待测程序的多个路径。

[0028] 具体地，首先通过静态分支预测方法查找程序源文件中的函数信息，并以函数为单位，在源代码中分析各个函数的分支关键词，并获取各个函数的起始行、终止行和函数内各分支间的层级关系，举例来说，对于一个二进制的待测程序，包含一个 main 函数，main 函数中又包含 if/else/for/while 等函数语句，可以将其看做一个树结构，其中，根节点为 main 函数，树节点为程序的分支节点，即 if/else/for/while 等，叶子节点为一个测试数据的测试结束点，从根节点到叶子节点的多个不同执行路径既为这个二进制待测程序的多个路径。

[0029] 其中，静态分支预测方法为公知方法，在本发明中不再赘述。

[0030] S103，分别将多个测试数据加载至待测程序，并获得每个测试数据所对应的路径。

[0031] 其中，多个测试数据之间具有数据分割点，用以对每个测试数据进行区分，待测程序的每个路径中在主函数及分支节点处均设有断点，通过断点获得每个测试数据所对应的路径。具体地，可以通过 GDB 的 MI 接口获取每个测试数据对应的路径，首先通过 Linux 管道技术与 GDB 的 MI 接口建立连接后，载入待测程序，并设定程序运行参数，同时在待测程序的 main 函数处插入一个断点，并利用静态分支预测方法扫描待测程序的源代码，找出所有的分支节点，并在分支节点处以及多个测试数据间的数据分割点处插入断点，然后运行待测程序，每次在断点处停下，记录断点位置，包括行号和节点名称(如 if/else/for/while 等)。当待测程序经过数据分割点时，表明一个测试数据执行路径已经结束，既可获取此测试数据对应的路径，并进行保存。

[0032] S104，根据每个测试数据所对应的路径对测试数据集合中的多个测试数据进行筛选。

[0033] 具体地，可在对应的路径相同的测试数据中选取一个作为最终测试数据，由此实现对多个测试数据进行筛选。

[0034] 根据本发明实施例的测试数据的筛选方法，无需商业软件提供接口，就可获取测试数据集合中的测试数据对应的路径，用户接入成本小，使用方便，并且能根据每个测试数据对应的路径对测试数据集合进行筛选，能够在众多测试数据中提取出覆盖待测程序所有路径并且最精简的测试数据集合，从而大幅度地提高了测试速度，并且缩短了测试周期。

[0035] 图 2 为根据本发明一个具体实施例的测试数据的筛选方法的流程图。

[0036] 如图 2 所示，根据本发明实施例的测试数据的筛选方法包括下述步骤。

[0037] S201，输入待测程序及测试数据集合，其中，测试数据集合中包括多个测试数据。

[0038] 在本发明的一个实施例中，可通过 GNU 程序调试器 GDB 对待测程序进行测试并对测试数据集合进行筛选，首先需要将待测程序及测试数据集合输入 GDB 中，通过 Linux 管道技术与 GDB 的机器界面 MI 接口建立连接后，根据命令可直接调用待测程序和测试数据集合中的多个测试数据。

[0039] S202，对待测程序进行分析以获得待测程序的多个路径。

[0040] 具体地，首先通过静态分支预测方法查找程序源文件中的函数信息，并以函数为单位，在源代码中分析各个函数的分支关键词，并获取各个函数的起始行、终止行和函数内各分支间的层级关系，举例来说，对于一个二进制的待测程序，包含一个 main 函数，main 函数中又包含 if/else/for/while 等函数语句，可以将其看做一个树结构，其中，根节点为

main 函数，树节点为程序的分支节点，即 if/else/for/while 等，叶子节点为一个测试数据的测试结束点，从根节点到叶子节点的多个不同执行路径既为这个二进制待测程序的多个路径。

[0041] 其中，静态分支预测方法为公知方法，在本发明中不再赘述。

[0042] S203，分别将多个测试数据加载至待测程序，并获得每个测试数据所对应的路径。

[0043] 其中，多个测试数据之间具有数据分割点，用以对每个测试数据进行区分，待测程序的每个路径中在主函数及分支节点处均设有断点，通过断点获得每个测试数据所对应的路径。具体地，可以通过 GDB 的 MI 接口获取每个测试数据对应的路径，首先通过 Linux 管道技术与 GDB 的 MI 接口建立连接后，载入待测程序，并设定程序运行参数，同时在待测程序的 main 函数处插入一个断点，并利用静态分支预测方法扫描待测程序的源代码，找出所有的分支节点，并在分支节点处以及多个测试数据间的数据分割点处插入断点，然后运行待测程序，每次在断点处停下，记录断点位置，包括行号和节点名称(如 if/else/for/while 等)。当待测程序经过数据分割点时，表明一个测试数据执行路径已经结束，既可获取此测试数据对应的路径，并进行保存。

[0044] S204，判断是否存在两个测试数据所对应的路径相同。

[0045] 具体地，当一个测试数据执行路径结束时，首先判断是否存在与该测试数据对应的路径相同的已存储的路径，即判断该测试数据对应的路径中包含的路径号是否相同，如果不存在，则此测试数据有效，并存储此测试数据的路径。

[0046] S205，如果存在，则进一步判断路径相同的两个测试数据是否具有相同的处理逻辑。

[0047] 如果存在与此测试数据对应的路径相同的已存储的路径，则进一步判断路径相同的两个测试数据是否具有相同的处理逻辑，即判断路径相同的两个测试数据所测试的目标功能是否相同。

[0048] S206，如果是，则选择两个测试数据中的一个作为最终的测试数据。

[0049] 如果路径相同的两个测试数据具有相同的处理逻辑，即路径相同的两个测试数据所测试的目标功能也相同，则此测试数据无效，并选择两个测试数据中的一个作为最终的测试数据，由此，所有对应路径相同并且处理逻辑相同的测试数据中只选取一个作为最终的测试数据，减少了不必要的测试数据，能够提高测试速度以及缩短测试周期。

[0050] 根据本发明实施例的测试数据的筛选方法，能够在多个对应路径相同的测试数据中筛选一个作为最终的测试数据，进一步精简了测试数据集合，提高测试速度，并且缩短测试周期。

[0051] 图 3 为根据本发明另一个实施例的测试数据的筛选方法的流程图。

[0052] 如图 3 所示，根据本发明实施例的测试数据的筛选方法包括下述步骤。

[0053] S301，输入待测程序及测试数据集合，其中，测试数据集合中包括多个测试数据。

[0054] 在本发明的一个实施例中，可通过 GNU 程序调试器 GDB 对待测程序进行测试并对测试数据集合进行筛选，首先需要讲待测程序及测试数据集合输入 GDB 中，通过 Linux 管道技术与 GDB 的机器界面 MI 接口建立连接后，根据命令可直接调用待测程序和测试数据集合中的多个测试数据。

[0055] S302，对待测程序进行分析以获得待测程序的多个路径。

[0056] 具体地，先通过静态分支预测方法查找程序源文件中的函数信息，并以函数为单位，在源代码中分析各个函数的分支关键词，并获取各个函数的起始行、终止行和函数内各分支间的层级关系，举例来说，对于一个二进制的待测程序，包含一个 main 函数，main 函数中又包含 if/else/for/while 等函数语句，可以将其看做一个树结构，其中，根节点为 main 函数，树节点为程序的分支节点，即 if/else/for/while 等，叶子节点为一个测试数据的测试结束点，从根节点到叶子节点的多个不同执行路径既为这个二进制待测程序的多个路径。

[0057] 其中，静态分支预测方法为公知方法，在本发明中不再赘述。

[0058] S303，分别将多个测试数据加载至待测程序，并获得每个测试数据所对应的路径。

[0059] 其中，多个测试数据之间具有数据分割点，用以对每个测试数据进行区分，待测程序的每个路径中在主函数及分支节点处均设有断点，通过断点获得每个测试数据所对应的路径。具体地，可以通过 GDB 的 MI 接口获取每个测试数据对应的路径，首先通过 Linux 管道技术与 GDB 的 MI 接口建立连接后，载入待测程序，设定程序运行参数，同时在待测程序的 main 函数处插入一个断点，并利用静态分支预测方法扫描待测程序的源代码，找出所有的分支节点，并在分支节点处以及多个测试数据间的数据分割点处插入断点，然后运行待测程序，每次在断点处停下，记录断点位置，包括行号和节点名称(如 if/else/for/while 等)。当待测程序经过数据分割点时，表明一个测试数据执行路径已经结束，既可获取此测试数据对应的路径，并进行保存。

[0060] S304，判断是否存在两个测试数据所对应的路径相同。

[0061] 具体地，当一个测试数据执行路径结束时，首先判断是否存在与该测试数据对应的路径相同的已存储的路径，即判断该测试数据对应的路径中包含的路径号是否相同，如果不存在，则此测试数据有效，并存储此测试数据的路径。

[0062] S305，如果存在，则进一步判断路径相同的两个测试数据是否具有相同的处理逻辑。

[0063] 如果存在与此测试数据对应的路径相同的已存储的路径，则进一步判断路径相同的两个测试数据是否具有相同的处理逻辑，即判断路径相同的两个测试数据所测试的目标功能是否相同。

[0064] S306，如果是，则选择两个测试数据中的一个作为最终的测试数据。

[0065] 如果路径相同的两个测试数据具有相同的处理逻辑，即路径相同的两个测试数据所测试的目标功能也相同，则此测试数据无效，并选择两个测试数据中的一个作为最终的测试数据，由此，所有对应路径相同并且处理逻辑相同的测试数据中只选取一个作为最终的测试数据，减少了不必要的测试数据，能够提高测试速度以及缩短测试周期。

[0066] S307，记录每个测试数据对应的路径，及对应的路径的断点信息并作为测试文件输出。

[0067] 在本发明的一个实施例中，在每个测试数据测试结束时记录其对应的路径及其断点信息，并根据路径断点信息生成程序执行路径信息，以及将测试数据对应的路径信息和程序执行路径信息作为输出文件进行输出，以展示给用户。

[0068] 根据本发明实施例的测试数据的筛选方法，能够记录每个测试数据对应的路径以及对应路径的断点信息，并据此生成测试文件输出，是用户能够让用户了解测试数据的执

行顺序，方便测试工程师对测试进行跟踪分析，使其能够根据未被覆盖的分支语句反推并添加测试用例，提高测试函数覆盖率。

[0069] 为了实现上述实例，本发明还提出一种测试数据的筛选装置。

[0070] 一种测试数据的筛选装置，包括：输入模块，用于输入待测程序及测试数据集合，其中，测试数据集合中包括多个测试数据；分析模块，用于对待测程序进行分析以获得待测程序的多个路径；加载模块，用于分别将多个测试数据加载至待测程序，并获得每个测试数据所对应的路径；以及筛选模块，用于根据每个测试数据所对应的路径对测试数据集合中的多个测试数据进行筛选。

[0071] 图 4 为根据本发明一个实施例的测试数据的筛选装置的结构框图。

[0072] 如图 4 所示，根据本发明实施例的测试数据的筛选装置包括：输入模块 100、分析模块 200、加载模块 300 以及筛选模块 400。

[0073] 具体地，输入模块 100 用于输入待测程序及测试数据集合，其中，测试数据集合中包括多个测试数据。

[0074] 在本发明的一个实施例中，可通过 GNU 程序调试器 GDB 对待测程序进行测试并对测试数据集合进行筛选，首先通过输入模块 100 将待测程序及测试数据集合输入 GDB 中，通过 Linux 管道技术与 GDB 的机器界面 MI 接口建立连接后，根据命令可直接调用待测程序和测试数据集合中的多个测试数据。

[0075] 分析模块 200 用于对待测程序进行分析以获得待测程序的多个路径。更具体地，分析模块 200 首先通过静态分支预测方法查找程序源文件中的函数信息，并以函数为单位，在源代码中分析各个函数的分支关键词，并获取各个函数的起始行、终止行和函数内各分支间的层级关系。举例来说，对于一个二进制的待测程序，包含一个 main 函数，main 函数中又包含 if/else/for/while 等函数语句，可以将其看做一个树结构，其中，根节点为 main 函数，树节点为程序的分支节点，即 if/else/for/while 等，叶子节点为一个测试数据的测试结束点，从根节点到叶子节点的多个不同执行路径既为这个二进制待测程序的多个路径。其中，静态分支预测方法为公知方法，在此不再赘述。

[0076] 加载模块 300 用于分别将多个测试数据加载至待测程序，并获得每个测试数据所对应的路径。其中，多个测试数据之间具有数据分割点，用以对每个测试数据进行区分，待测程序的每个路径中在主函数及分支节点处均设有断点，通过断点获得每个测试数据所对应的路径。更具体地，可以通过 GDB 的 MI 接口获取每个测试数据对应的路径，首先通过 Linux 管道技术与 GDB 的 MI 接口建立连接后，加载模块 300 载入待测程序，并设定程序运行参数，同时在待测程序的 main 函数处插入一个断点，并利用静态分支预测方法扫描待测程序的源代码，找出所有的分支节点，并在分支节点处以及多个测试数据间的数据分割点处插入断点，然后运行待测程序，每次在断点处停下，记录断点位置，包括行号和节点名称(如 if/else/for/while 等)。当待测程序经过数据分割点时，表明一个测试数据执行路径已经结束，既可获取此测试数据对应的路径，并进行保存。

[0077] 筛选模块 400 用于根据每个测试数据所对应的路径对测试数据集合中的多个测试数据进行筛选。更具体地，筛选模块 400 在两个测试数据所对应的路径相同，并且路径相同的两个测试数据具有相同的处理逻辑时，选择两个测试数据中的一个作为最终的测试数据由此实现对多个测试数据进行筛选。当一个测试数据执行路径结束时，首先判断该测试

数据对应的路径是否与已经存储的路径相同，即判断该测试数据对应的路径中包含的路径号是否相同，如果不同，则此测试数据的路径有效，并存储此测试数据的路径，如果相同，则进一步判断路径相同的两个测试数据是否具有相同的处理逻辑，即判断路径相同的两个测试数据所测试的目标功能是否相同。如果路径相同的两个测试数据具有相同的处理逻辑，即路径相同的两个测试数据所测试的目标功能也相同，则选择两个测试数据中的一个作为最终的测试数据，由此，所有对应路径相同并且处理逻辑相同的测试数据中只选取一个作为最终的测试数据，减少了不必要的测试数据，能够提高测试速度以及缩短测试周期。

[0078] 根据本发明实施例的测试数据的筛选装置，无需商业软件提供接口，就可获取测试数据集合中的测试数据对应的路径，用户接入成本小，使用方便，并且筛选模块能根据每个测试数据对应的路径对测试数据集合进行筛选，能够在众多测试数据中提取出覆盖待测程序所有路径并且最精简的测试数据集合，从而大幅度地提高了测试速度，并且缩短了测试周期。

[0079] 图 5 为根据本发明另一个实施例的测试数据的筛选装置的结构框图。

[0080] 如图 5 所示，根据本发明实施例的测试数据的筛选装置在图 4 所示的基础上还包括 :输出模块 500。

[0081] 具体地，输出模块 500 用于记录每个测试数据对应的路径，及对应的路径的断点信息并作为测试文件输出。在本发明的一个实施例中，输出模块能够在每个测试数据测试结束时记录其对应的路径及其断点信息，并根据路径断点信息生成程序执行路径信息，以及将测试数据对应的路径信息和程序执行路径信息作为输出文件进行输出，以展示给用户。

[0082] 根据本发明实施例的测试数据的筛选装置，通过输出模块能够记录每个测试数据对应的路径以及对应路径的断点信息，并据此生成测试文件输出，是用户能够让用户了解测试数据的执行顺序，方便测试工程师对测试进行跟踪分析，使其能够根据未被覆盖的分支语句反推并添加测试用例，提高测试函数覆盖率。

[0083] 在本说明书的描述中，参考术语“一个实施例”、“一些实施例”、“示例”、“具体示例”、或“一些示例”等的描述意指结合该实施例或示例描述的具体特征、结构、材料或者特点包含于本发明的至少一个实施例或示例中。在本说明书中，对所述术语的示意性表述不一定指的是相同的实施例或示例。而且，描述的具体特征、结构、材料或者特点可以在任何的一个或多个实施例或示例中以合适的方式结合。

[0084] 尽管已经示出和描述了本发明的实施例，对于本领域的普通技术人员而言，可以理解在不脱离本发明的原理和精神的情况下可以对这些实施例进行多种变化、修改、替换和变型，本发明的范围由所附权利要求及其等同限定。

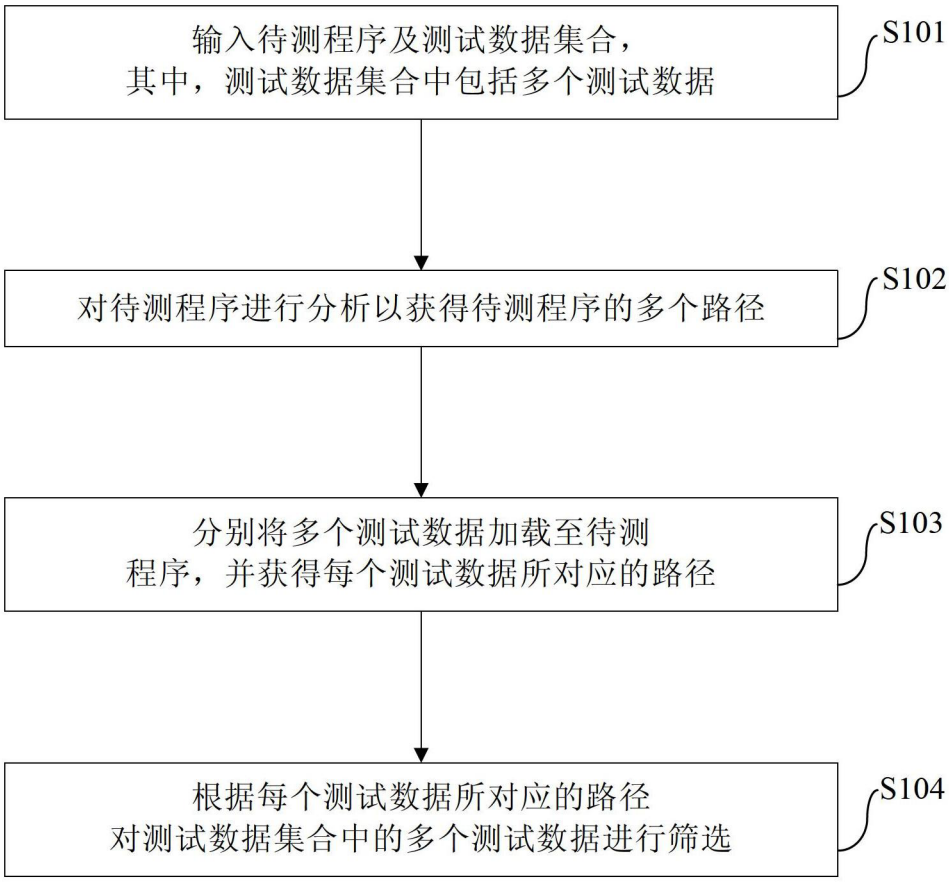

输入待测程序及测试数据集合，
其中，测试数据集合中包括多个测试数据
S101
对待测程序进行分析以获得待测程序的多个路径
S102
分别将多个测试数据加载至待测
程序，并获得每个测试数据所对应的路径
S103
根据每个测试数据所对应的路径
对测试数据集合中的多个测试数据进行筛选
S104

图 1

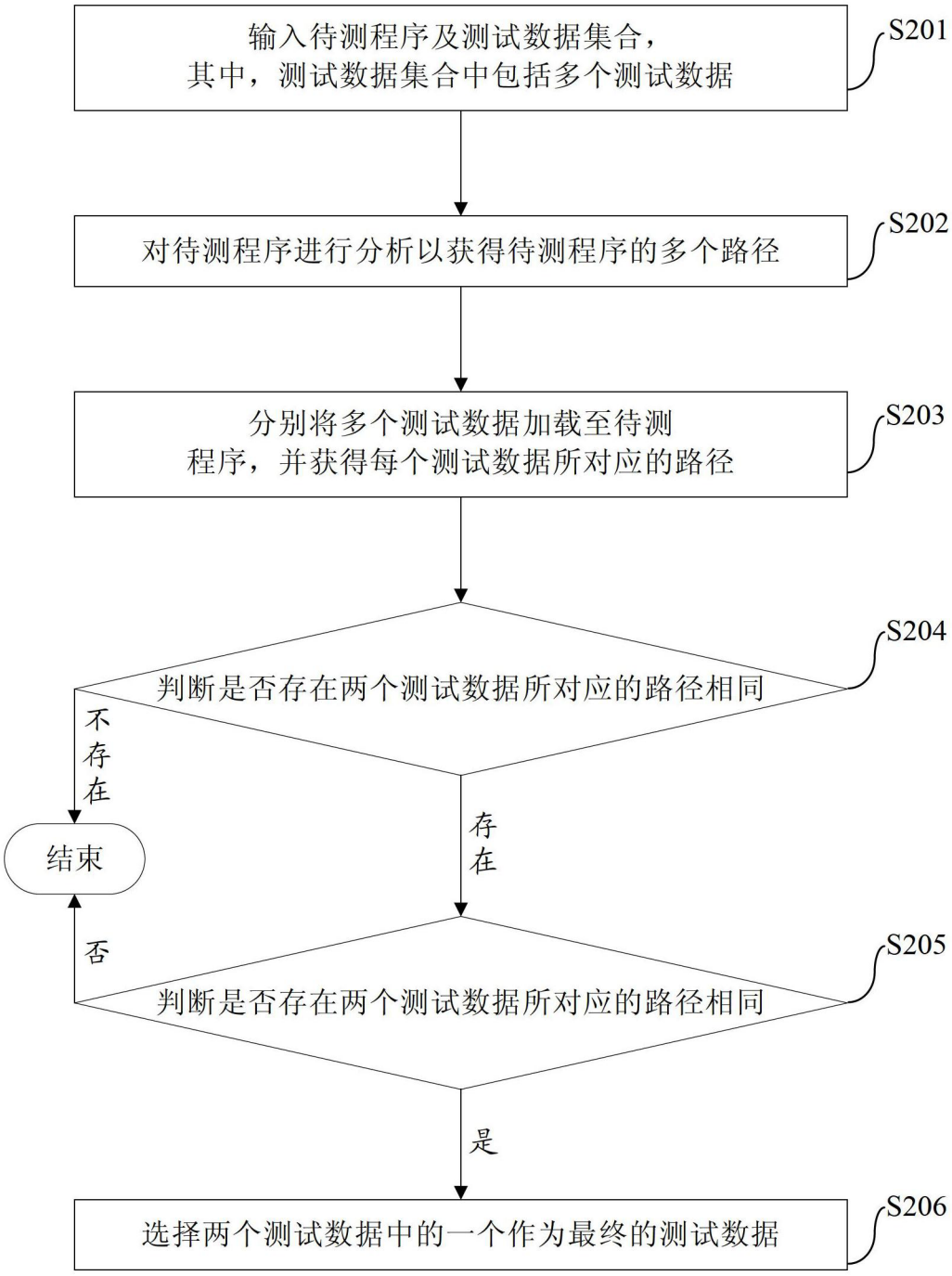

输入待测程序及测试数据集合，
其中，测试数据集合中包括多个测试数据
S201
对待测程序进行分析以获得待测程序的多个路径
S202
分别将多个测试数据加载至待测
程序，并获得每个测试数据所对应的路径
S203
判断是否存在两个测试数据所对应的路径相同
S204
不存在
存在
结束
否
判断是否存在两个测试数据所对应的路径相同
S205
是
选择两个测试数据中的一个作为最终的测试数据
S206

图 2

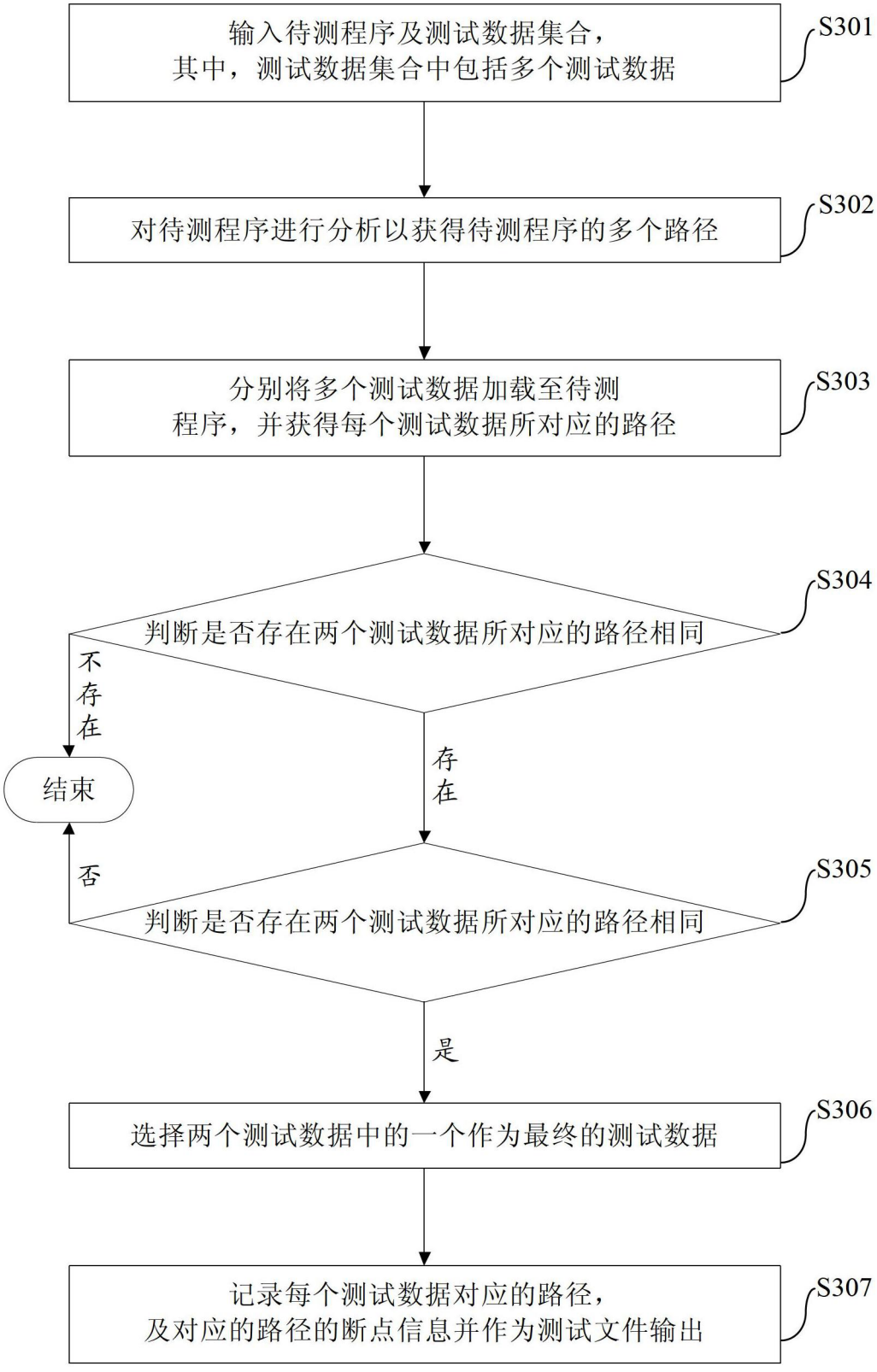

输入待测程序及测试数据集合，
其中，测试数据集合中包括多个测试数据
S301
对待测程序进行分析以获得待测程序的多个路径
S302
分别将多个测试数据加载至待测
程序，并获得每个测试数据所对应的路径
S303
判断是否存在两个测试数据所对应的路径相同
S304
不存在
存在
结束
否
判断是否存在两个测试数据所对应的路径相同
S305
是
选择两个测试数据中的一个作为最终的测试数据
S306
记录每个测试数据对应的路径，
及对应的路径的断点信息并作为测试文件输出
S307

图 3

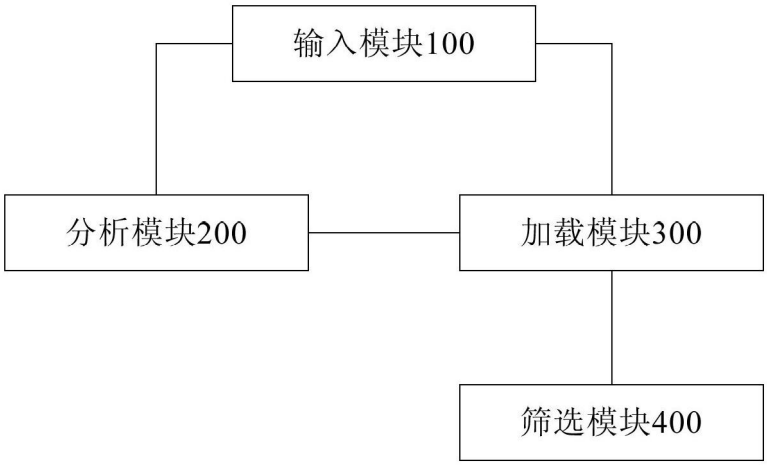

输入模块100
分析模块200
加载模块300
筛选模块400

图 4

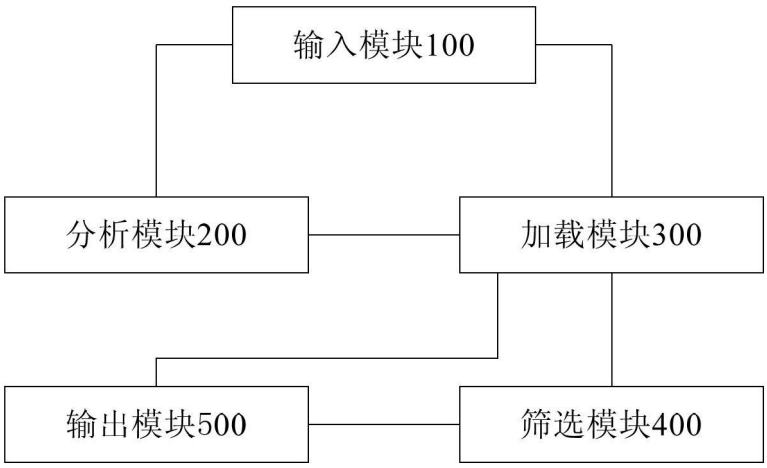

输入模块100
分析模块200
加载模块300
输出模块500
筛选模块400

图 5