(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 110114755 A
(43)申请公布日 2019.08.09

(21)申请号 201780081109.4

(22)申请日 2017.12.14

(30)优先权数据
15/394,238 2016.12.29 US

(85)PCT国际申请进入国家阶段日
2019.06.27

(86)PCT国际申请的申请数据
PCT/US2017/066236 2017.12.14

(87)PCT国际申请的公布数据
WO2018/125584 EN 2018.07.05

(71)申请人 微软技术许可有限责任公司
地址 美国华盛顿州

(72)发明人 S·古尔瓦尼 K·M·埃利斯

(74)专利代理机构 北京市金杜律师事务所 11256
代理人 王茂华

(51)Int.Cl.
G06F 8/30(2006.01)
G06F 17/21(2006.01)

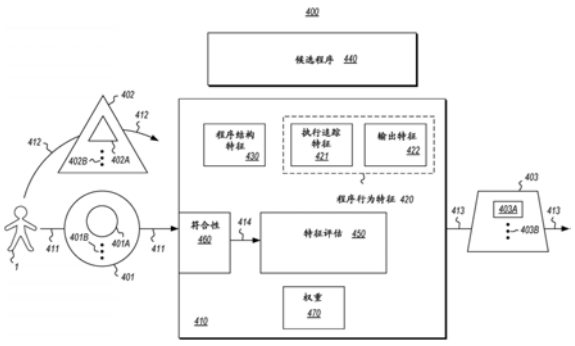
权利要求书2页 说明书10页 附图6页

(54)发明名称

示例编程中的行为特征使用

(57)摘要

用于在示例编程中缩小针对均符合由用户提供的示例行为的程序的选择的技术。即使用户提供不充足的行为示例来精确地标识应当被使用的程序，系统也仍然使用程序的程序行为特征(潜在地连同结构特征)以便标识将符合特定组的行为示例的每个程序的适合性。然后特定程序被选择和启用以用于该用户，使得特定程序执行由一个或多个程序行为示例所例示的行为。在对程序的选择使用用户协助的情况下，针对每个可能程序的适合性可以用于决策应当使复数个可能程序中的哪个可由用户选择。那些较高适合性的程序可以被可视化给用户以供选择。



CN 110114755 A

1.一种计算机系统,包括:

一个或多个处理器;以及

一个或多个计算机可读介质,其上具有计算机可执行指令,所述计算机可执行指令被构造以使得在由所述一个或多个处理器执行时,所述计算系统被使得执行一种用于通过使用由用户提供的一组一个或多个程序行为示例作为输入来启用针对所述用户的程序而执行示例编程的方法,所述方法包括:

标识将符合所述一组一个或多个行为示例的多个程序;

基于特定程序的一个或多个程序行为特征,标识所述多个程序中的特定程序的适合性;以及

至少基于所述特定程序的所标识的所述适合性,启用所述特定程序,使得所述特定程序执行由所述一个或多个程序行为示例所例示的行为。

2.根据权利要求1所述的计算系统,所述方法还包括:

使用针对所述多个程序中的复数个程序中的每个程序的一个或多个程序行为特征来标识所述复数个程序的适合性,所述复数个程序包括所述特定程序;以及

在启用所述特定程序之前,从所述复数个程序之中选择所述特定程序。

3.根据权利要求2所述的计算系统,所述方法还包括:

可视化包括具有较高的所标识的适合性的所述特定程序的所述复数个程序;以及

检测表示对所述特定程序的选择的用户输入,所述特定程序的所述启用响应于表示对所述特定程序的选择的所述用户输入的所述检测而被执行,被可视化的程序基于所标识的所述适合性而在视觉上被排序。

4.根据权利要求1所述的计算系统,所述一个或多个程序行为特征包括多个程序行为特征,所述多个程序行为特征包括至少一个程序追踪特征和至少一个程序输出特征。

5.根据权利要求1所述的计算系统,所述一个或多个程序行为特征包括至少一个程序输出特征。

6.根据权利要求1所述的计算系统,所述特定程序的所标识的所述适合性还基于所述特定程序的至少一个程序结构特征。

7.根据权利要求1所述的计算系统,所述一个或多个程序行为特征包括多个程序行为特征,所述特定程序的所标识的所述适合性通过执行函数而被标识,所述函数使用所述多个程序行为特征中的每个程序行为特征的加权作为至少一个程序结构特征的加权。

8.根据权利要求1所述的计算系统,所述特定程序的所标识的所述适合性通过执行函数而被标识,所述函数使用包括所述一个或多个程序行为特征的多个程序行为特征中的每个程序行为特征的加权。

9.根据权利要求8所述的计算系统,所述加权是机器学习的产物。

10.一种用于通过使用由用户提供的一组一个或多个程序行为示例作为输入来启用针对所述用户的程序而执行示例编程的方法,所述方法包括:

标识将符合所述一组一个或多个行为示例的多个程序;

基于特定程序的一个或多个程序行为特征,标识所述多个程序中的特定程序的适合性;以及

至少基于所述特定程序的所标识的所述适合性,启用所述特定程序,使得所述特定程

序执行由所述一个或多个程序行为示例所例示的行为。

示例编程中的行为特征使用

背景技术

[0001] 传统上，计算机程序的构建需要经训练的计算机程序员的大量且耗时的工作。另一方面，示例编程技术允许用户通过提供所意图行为的示例来构建和运行新程序。换言之，不是要求用户告诉计算机做什么(通常需要显式编程)，示例编程而是允许用户通过提供几个示例来简单地向计算机示出做什么。

[0002] 提供所意图行为的示例对于典型人类而言是直观的。毕竟，人类熟悉通过向他们示出做什么来教彼此怎么做一些事情，并将它留给学生以施加所演示的行为。因此，示例编程不需要编程语言或专门编程技术的知识。示例编程因此具有对几乎不具有或不具有编程知识的更大的用户群体开放灵活计算的新入口。

[0003] 然而，示例编程依赖于用户来提供行为的示例。系统针对可能程序的巨大集合评价示例行为以便标识满足行为的程序。所标识的程序然后用于执行行为。行为的示例可能包括输入和对应的可接受的输出。当越来越多的示例被提供时，系统可以缩小标识满足所演示的行为的程序。

[0004] 然而，所意图行为可能足够复杂使得可能需要大量示例以便以能够标识执行所意图行为的合适程序的充分的确定性来确定所意图行为。典型人类不想要提供大量示例，并且/或者可能不具有大量示例来提供。相反，用户常常仅提供较少数目的示例，并且可能甚至少至一个示例。

[0005] 然而，几个所提供的示例很少足以在可能程序的巨大空间内确定所意图行为。因此，系统挑选尚未明确地被给出的一个示例或多个示例排除的可能程序中的一个。用于挑选程序的先前方法依赖于程序本身的结构特征。例如，可以存在朝向更短、更简单、或更自然结构化的程序的偏向。然而，无论何时所意图行为未通过足够的示例来明确，都存在不执行所意图行为的程序将被选择的机会。

[0006] 本文中要求保护的技术方案不限于解决例如以上描述的任何缺点或仅操作于例如以上描述的环境中的实施例。相反，该背景仅被提供以说明本文中描述的一些实施例可以被实践在其中的一个示例性技术领域。

发明内容

[0007] 本文中描述的至少一些实施例涉及示例编程的执行，其中程序通过用户简单地提供一组行为示例而被选择并且被启用以用于用户。即使用户提供不足够的行为示例来精确地标识应当被使用的程序，系统也仍然使用程序的行为特征(潜在连同结构特征)以便标识将符合特定一组行为示例的每个程序的适合性。然后特定程序被选择和启用以用于该用户，使得特定程序执行由一个或多个程序行为示例所例示的行为。

[0008] 在对程序的选择中使用了用户协助的情况下，针对每个可能程序的适合性可以用于决策应当使复数个可能程序中的哪个可由用户选择。例如，那些较高适合性的程序可以被可视化给用户，可能按适合性的顺序来排序，从而允许用户智能地选择合适的程序以完成示例编程。

[0009] 在一个示例情况中，行为示例由用户以输入和输出集合的形式提供。即，对于一个示例，用户提供特定输入，并且表达给定该特定输入的期望输出。如果仅存在几个这样的示例，则可能存在能够生成给定那些输入的期望输出的许多程序，一些比另一些更适合于总体一般所意图的功能。本文中描述的原理通过使用程序行为特征(以及可能的程序结构特征)以便标识哪些程序是适合的，由此至少部分地缩小可用于执行功能的程序的选择，来帮助解决这种歧义。在一个实施例中，总体行为域是数据变换(将输入数据变换到输出数据)。这样的变换的一个示例是数据提取(从输入数据中提取数据的一部分以由此生成以结构化表格形式的输出数据)。

[0010] 因此，本文中描述的原理允许在其中用户甚至对于要执行更复杂和/或难以基于较少示例来预测的行为的程序而不提供多于一个示例或几个示例的现实情况下对示例编程的更大技术灵活性。因此，本文中描述的原理通过使示例编程更易于用户参与来改善计算技术的功能。

[0011] 提供本发明内容从而以简化的形式介绍下面在具体实施方式中进一步描述的一系列概念。本发明内容不旨在标识要求保护的技术方案的关键特征或必要特征，也不旨在用作确定要求保护的技术方案的范围的辅助。

附图说明

[0012] 为了描述能够获得本发明的以上所述的优点和特征以及其他优点和特征的方式，将通过参考在附图中图示的其具体实施例来呈现以上简单描述的本发明的更具体的描述。在理解这些附图仅描绘本发明的典型实施例并且因此不被认为是对其范围的限制的情况下，将通过附图的使用利用附加的特异性和细节来描述并解释本发明，在附图中：

[0013] 图1图示了本文中描述的原理可以被采用于其中的一个示例计算机系统；

[0014] 图2图示了示出在选择用于对输入进行操作以生成合适输出的合适程序中的较低水平歧义的示例编程的一个示例；

[0015] 图3图示了示出在选择用于对输入进行操作以生成合适输出的合适程序中的较高水平歧义的示例编程的一个示例；

[0016] 图4图示了本文中描述的原理可以操作于其中的示例编程环境；

[0017] 图5图示了用于执行示例编程的方法的流程图，该方法可以在图4的示例编程环境中被执行；

[0018] 图6图示了跟随执行追踪以找到重叠提取的一个示例；

[0019] 图7示出了示出程序输出特征可以如何由输出相似性表征的电子表格示例；以及

[0020] 图8是语义知识可以如何用于确定程序输出特征的一个示例。

具体实施方式

[0021] 本文中描述的至少一些实施例涉及示例编程的执行，其中程序通过用户简单地提供一组行为示例而被选择并且被启用以用于用户。即使用户提供不足够的行为示例来精确地标识应当被使用的程序，系统也仍然使用程序的行为特征(潜在地连同结构特征)以便标识将符合特定一组行为示例的每个程序的适合性。然后特定程序被选择和启用以用于该用户，使得特定程序执行由一个或多个程序行为示例所例示的行为。

[0022] 在对程序的选择中使用了用户协助的情况下，针对每个可能程序的适合性可以用于决策应当使复数个可能程序中的哪个可由用户选择。例如，那些较高适合性的程序可以被可视化给用户，可能按适合性的顺序来排序，从而允许用户智能地选择合适的程序以完成示例编程。

[0023] 在一个示例情况中，行为示例由用户以输入和输出集合的形式提供。即，对于一个示例，用户提供特定输入，并且表达给定该特定输入的期望输出。如果仅存在几个这样的示例，则可能存在能够生成给定那些输入的期望输出的许多程序，一些比另一些更适合于总体一般所意图的功能。本文中描述的原理通过使用程序行为特征(以及可能的程序结构特征)以便标识哪些程序是合适的，由此至少部分地缩小可用于执行功能的程序的选择，来帮助解决这种歧义。在一个实施例中，总体行为域是数据变换(将输入数据变换到输出数据)。这样的变换的一个示例是数据提取(从输入数据中提取数据的一部分以由此生成以结构化表格形式的输出数据)。

[0024] 因此，本文中描述的原理允许在其中用户甚至对于要执行更复杂和/或难以基于较少示例来预测的行为的程序而不提供多于一个示例或几个示例的现实情况下对示例编程的更大技术灵活性。因此，本文中描述的原理通过使示例编程更易于用户参与来改善计算技术的功能。

[0025] 因为本文中描述的原理操作于计算系统的上下文中，所以将参考图1描述计算系统。然后，缩小针对均符合示例行为的程序的选择的原理参考图2至图8，其中缩小使用每个候选程序的程序行为特征来完成。

[0026] 计算系统现在越来越多地采取各种各样的形式。计算系统可以例如为手持式设备、家电、膝上型计算机、台式计算机、大型主机、分布式计算系统、数据中心、或甚至传统上尚未被认为是计算系统的设备，诸如可穿戴设备(例如，眼镜、手表、带等)。在本说明书中和在权利要求书中，术语“计算系统”被广泛地定义为包括包含至少一个物理且有形处理器和物理且有形存储器的任何设备或系统(或其组合)，物理且有形存储器能够具有存储于其上的可以由处理器执行的计算机可执行指令。存储器可以采取任何形式并且可以取决于计算系统的性质和形式。计算系统可以被分布在网络环境上并且可以包括多个组成计算系统。

[0027] 如图1所示，以其最基本配置，计算系统100通常包括至少一个硬件处理单元102和存储器104。存储器104可以是物理系统存储器，其可以是易失性的、非易失性的、或者这两种的某种组合。术语“存储器”也可以在本文用于指代诸如物理存储介质的非易失性大容量存储装置。如果计算系统是分布式的，则处理、存储器和/或存储能力也可以是分布式的。

[0028] 计算系统100在其上具有通常被称为“可执行组件”的多个结构。例如，计算系统100的存储器104被图示为包括可执行组件106。术语“可执行组件”是针对如下结构的名称，该结构对计算领域普通技术人员很好理解为可以为软件、硬件或其组合的结构。例如，当以软件来实现时，本领域普通技术人员将理解，可执行组件的结构可以包括可以在计算系统上执行的软件对象、例程、方法，无论这样的可执行组件是否存在于计算系统的堆中，或者无论可执行组件是否存在于计算机可读存储介质上。

[0029] 在这样的情况下，本领域普通技术人员将认识到，可执行组件的结构存在于计算机可读介质上，使得当由计算系统的一个或多个处理器(例如，由处理器线程)解译时，使计算系统执行功能。这样的结构可以是由处理器直接计算机可读的(在可执行组件是二进制

的情况下是这种情况)。备选地,结构可以被构造为可解译的和/或编译的(无论在单个阶段中还是在多个阶段中),以便生成可直接由处理器解译的这样的二进制。当使用术语“可执行组件”时,可执行组件的示例结构的这样的理解很好地在计算领域普通技术人员的理解内。

[0030] 术语“可执行组件”还由普通技术人员很好地理解为包括专有地或几乎专有地以硬件实现的结构,诸如在现场可编程门阵列(FPGA)、专用集成电路(ASIC)或任何其他专门的电路内实现的结构。因此,术语“可执行组件”是针对由计算领域普通技术人员很好地理解的结构的术语,无论该结构以软件、硬件还是组合来实现。在本说明书中,也可以使用术语“组件”。如在本说明书中并且在本情况中使用的,该术语(不管该术语是否利用一个或多个修饰语来修饰)也旨在与术语“可执行组件”同义或是这样的“可执行组件”的具体类型,并且因此还具有由计算领域普通技术人员很好理解的结构。

[0031] 在随后的描述中,参考由一个或多个计算系统执行的动作来描述实施例。如果这样的动作以软件来实现,则(执行动作的相关联的计算系统的)一个或多个处理器响应于已经执行了组成可执行组件的计算机可执行指令而引导计算系统的操作。例如,这样的计算机可执行指令可以被实施在形成计算机程序产品的一个或多个计算机可读介质上。这样的操作的一个示例涉及对数据的操纵。

[0032] 计算机可执行指令(以及操纵的数据)可以被存储在计算系统100的存储器104中。计算系统100还可以包含允许计算系统100通过例如网络110与其他计算系统通信的通信信道108。

[0033] 尽管不是所有计算系统都需要用户接口,但是在一些实施例中,计算系统100包括用于与用户进行接口交互的用户接口112。用户接口112可以包括输出机制112A以及输入机制112B。本文中描述的原理不限于精确的输出机制112A或输入机制112B,因为这将取决于设备的性质。然而,输出机制112A可以包括例如扬声器、显示器、触觉输出、全息图、虚拟现实等。输入机制112B的示例可以包括例如麦克风、触摸屏、全息图、虚拟现实、相机、键盘、其他指针输入的鼠标、任何类型的传感器等。

[0034] 本文中描述的实施例可以包括或利用包括计算机硬件的专用计算系统或通用计算系统,计算机硬件例如一个或多个处理器和系统存储器,如下面更详细地讨论的。本文中描述的实施例还包括用于携带或存储计算机可执行指令和/或数据结构的物理介质和其他计算机可读介质。这样的计算机可读介质可以是能够由通用计算系统或专用计算系统访问的任何可用介质。存储计算机可执行指令的计算机可读介质是物理存储介质。携带计算机可执行指令的计算机可读介质是传输介质。因此,通过举例而非限制性的方式,实施例可以包括至少两种截然不同种类的计算机可读介质:存储介质和传输介质。

[0035] 计算机可读存储介质包括RAM、ROM、EEPROM、CD-ROM或其他光盘存储装置、磁盘存储装置或其他磁性存储设备、或可以用于存储以计算机可执行指令或数据结构的形式的期望程序代码部件并且可以由通用计算系统或专用计算系统访问的任何其他物理且有形存储介质。

[0036] “网络”被定义为使得电子数据能够在计算系统和/或模块和/或其他电子设备之间传输的一个或多个数据链路。当通过网络或另外的通信连接(硬接线的、无线的或者硬接线或无线的组合)将信息传送或提供到计算系统时,计算系统将连接恰当地视为传输介质。

传输介质可以包括可以用于携带以计算机可执行指令或数据结构的形式的期望程序代码部件并且可以由通用计算系统或专用计算系统访问的网络和/或数据链路。以上的组合也应当被包括在计算机可读介质的范围内。

[0037] 另外，在到达各种计算系统组件后，以计算机可执行指令或数据结构的形式的程序代码部件可以自动地从传输介质被传送到存储介质(或者反之亦然)。例如，通过网络或数据链路而接收到的计算机可执行指令或数据结构可以被缓冲在网络接口模块(例如，“NIC”)内的RAM中，并且然后最终被传送到计算系统RAM和/或被传送到在计算系统处的非易失性存储介质。因此，应当理解，可读介质可以被包括在也(或甚至主要)利用传输介质的计算系统组件中。

[0038] 计算机可执行指令包括例如当在处理器处被执行时使得通用计算系统、专用计算系统、或专用处理设备执行特定功能或特定组的功能的指令和数据。备选地或另外，计算机可执行指令可以将计算系统配置为执行特定功能或特定组的功能。计算机可执行指令可以例如为二进制的或甚至在由处理器直接执行之前经历一些转化(诸如编译)的指令，诸如中间格式指令，诸如汇编语言，或甚至源代码。

[0039] 本领域技术人员将认识到可以在具有许多类型的计算系统配置的网络计算环境中实践本发明，许多类型的计算系统配置包括个人计算机、台式计算机、膝上型计算机、消息处理器、手持式设备、多处理器系统、基于微处理器的或可编程的消费电子设备、网络PC、微型计算机、大型计算机、移动电话、PDA、寻呼机、路由器、交换机、数据中心、可穿戴设备(诸如眼镜或手表)等。还可以在分布式系统环境中实践本发明，在分布式系统环境中通过网络(通过硬接线数据链路、无线数据链路、或通过硬接线数据链路和无线数据链路的组合)而链接的本地计算系统和远程计算系统两者都执行任务。在分布式系统环境中，程序模块可以被定位在本地存储器存储设备和远程存储器存储设备两者中。

[0040] 本领域技术人员还将认识到本发明可以被实践于云计算环境中。云计算环境可以是分布式的，但是这不是要求的。当分布式时，云计算环境可以国际性地分布在组织内和/或具有跨多个组织而拥有的组件。在本说明书和随附权利要求书中，“云计算”被定义为用于启用对可配置计算资源(例如，网络、服务器、存储、应用以及服务)的共享池的按需网络访问的模型。“云计算”的定义不限于当被恰当地部署时可以从这样的模型获得的其他许多优点中的任何。

[0041] 例如，云计算当前在市场中被采用以便提供对可配置计算资源的共享池的无所不在且方便的按需访问。另外，可配置计算资源的共享池可以经由虚拟化被快速提供并以低管理工作或服务提供方交互被释放，并且然后相应地按比例进行调整。

[0042] 云计算模型可以包括各种特性，诸如按需自服务、宽网络访问、资源池化、快速弹性、可度量的服务等。云计算模型还可以以各种服务模型的形式出现，例如软件即服务(“SaaS”)、平台即服务(“PaaS”)以及基础设施即服务(“IaaS”)。云计算模型还可以使用诸如私有云、社团云、公共云、混合云等的不同部署模型而被部署。在本说明书中并且在权利要求书中，“云计算环境”是其中采用云计算的环境。

[0043] 在示例编程中，用户提供一组一个或多个行为示例，并且系统挑选或至少建议将不仅合适地完成由(多个)行为示例演示的确切行为而且还将合适地完成由行为示例所例示的更一般行为的程序。图2和图3图示了行为编程的两个示例。

[0044] 在图2中,用户正在电子表格程序的上下文中操作并且正在编辑两列——列A和列B。电子表格程序可以例如被显示在计算系统(例如,计算系统)的显示器(例如,输出机制112A之一)上。左边部分201示出了仅示出用户已经(例如,经由计算系统100的输入机制112B)提供了什么的电子表格的状态。右边部分202示出了在系统已经选择了用于将剩余输入变换成对应输出的程序之后的电子表格的状态。(由一个示例表示的)程序203表示系统根据系统从第一行的示例行为估计的一般行为而选择以便将剩余输入转换成对应输出的程序。在这种情况下,程序是将输入变换成输出的表达(例如,诸如通过提取输入的部分以生成输出)。

[0045] 在电子表格部分201的第一行中,用户提供行为示例。例如,在该第一行中,用户可以在列A中提供“John Smith”并在列B中提供“JS”。用户因此正在向系统示出关于列A中的其他输入做什么,而不是实际上写入表达。现在用户在列A(或电子表格部分201)的其他行中提供其他输入文本,并且期望系统选择不仅将满足第一行的该行为示例而且还将遵循最可能由示例所例示的针对其他输入的一般意图的功能。

[0046] 在图2的情况下,由示例所例示的一般行为意图仅具有低水平的歧义。很清楚的是,一般意图是输出被表示为由空格分开的每个文本部分的首字母的串联。系统可以因此选择程序“输出=输入A[0:1]+输入A[regexpos(“”,””):-1](output=inputA[0:1]+inputA[regexpos(“”,””):-1])作为程序203。这将被提供作为针对列A中的所有附加输入的程序。

[0047] 因此,如得到的电子表格部分202中所示,通过对表达的评估,列A的第二行中的“Mary Sue”将得到列B中的“MS”。类似地,列A中的“Jane Doe”将评估为列B中的“JD”。“Sumit Gulwani”将评估为列B中的“SG”。因此,示例编程允许通过提供相对少的行为示例而快速找到将输入转换为输出的合适程序。

[0048] 以上是相对简单的示例,其中很少有机会存在期望表达是什么的歧义。然而,仍然存在某种水平的歧义。例如,用户可能意图的是不管输入文本是什么都得到文本“JS”,尽管这看起来是不可能的意图。用户可能已经备选地意指,首字母的提取应当仅应用到头两个词,仅应用到至多头三个词,应用到所有词,等等。然而,如果存在仅提供的几个行为示例,则存在其中很少的行为示例可以实际上得到许多有竞争的且合理的可能程序来选择的其他时候。

[0049] 如图3中所示,假设用户指示输入“缺页数,1993”对应于输出“1993”作为行为示例的一个示例。现在用户提供“64-67,1995”作为另一输入并要求系统挑选用于生成输出的程序。示例本身是清楚的,但是更大的所意图行为实际上是相当有歧义的。实际上给定该单个示例存在无数可能的更大的所意图行为。

[0050] 一个可能所意图行为是从末尾提取第一个数字,在这种情况下输入“64-67,1995”将评估为“1995”的输出。另一可能所意图行为是从开头提取第一个数字,在这种情况下“64-67,1995”将评估为“64”的输出。另一可能所意图行为是提取在任何逗号之后的任何数字,在这种情况下输入“64-67,1995”将评估为“1995”的输出。另一可能所意图行为是仅提取在第一个逗号之后的数字,在这种情况下输入“64-67,1995”将评估为“1995”的输出。另一可能所意图行为是仅提取在最后一个逗号之后的数字,在这种情况下输入“64-67,1995”将评估为“1995”的输出。另一可能所意图行为是不管输入是什么总是生成恒定字符串

“1993”。因此，给定由用户提供的单个输入，存在关于所意图行为是什么(并且因此选择什么程序)的很大程序的歧义。

[0051] 用于解决该歧义的常规机制基于对程序结构特征的分析。对此的简单示例是对程序大小的特征进行加权，以便提供选择更小程序的偏向。其他方法也可能指望根据某个学习的准则来表示程序结构的简单性和自然性的其他结构特征。因此，解决该歧义的常规方法指望程序的结构特征来偏向其语法结构看起来看似合理的程序。这样的方法当预测更可能正确的程序以便解决从具有不足以确定确切程序的行为示例而产生的所意图行为的歧义时具有某种水平的准确性。

[0052] 根据本文中描述的原理，正确程序通过考虑程序的、独立于程序的结构的特征而被估计。即，正确程序基于程序的(多个)行为特征而被估计。这样的程序行为的行为特征可以包括例如程序的执行追踪特征和/或程序的输出特征。这样的行为特征可以与程序结构特征一起用于提供在给定有限组的行为示例的情况下预测正确程序的更高可能性。

[0053] 图4图示了本文中描述的原理可以操作于其中的示例编程环境400。用户1将一组一个或多个行为示例401提供(如由箭头411表示的)给程序缩小系统410。程序缩小系统410可以例如为图1的计算系统100。程序缩小系统410然后基于行为示例输入402来评估候选程序440的程序行为特征420(以及潜在地评估程序结构特征430)，以由此输出(如由箭头413所表示的)一组一个或多个程序403。这是图4的高级描述。现在将参考图5描述图4的剩余元件以及关于图4的所有元件的更多细节。

[0054] 图5图示了用于执行示例编程的方法500的流程图，该方法可以在图4的示例编程环境400中被执行。方法500通过访问一组一个或多个行为示例作为输入来开始(动作501)。在图4中，这由用户1将(多个)行为示例401提供(如由箭头411所表示的)给程序缩小系统410来表示。(多个)行为示例由图4中的圆形符号化。可以存在由用户1提供的少至仅单个的行为示例401A。然而，椭圆401B表示可以存在由用户1提供的附加的行为示例。所提供的非冗余行为示例越多，对所意图的一般行为的估计越好，但是以对用户的减少的方便为代价。毕竟，提供行为示例的确需要一些用户工作。

[0055] 另外，用户还可以可选地提供对于其不存在指定的精确行为的一些另外的输入(动作502)。代替地，用户可以依靠系统来根据由(多个)行为示例所例示的一般意图来选择程序以作用于输入。在图4中，这由用户1将(多个)输入402提供(如由箭头412所表示的)给程序缩小系统410来表示。这些输入由图4中的三角形符号化。可以存在少至零个这样的输入，尽管存在图4中明确示出的一个输入402A。椭圆402B表示在该数上可以存在从少至零个到任何数目一或更大的灵活性。

[0056] 作为一个示例，在图2中，用户提供输入“John Smith”和输出“JS”的组合作为行为示例(例如，行为示例401A)，而“Mary Sue”、“Jane Doe”和“Sumit Gulwani”中的每一个是输入(例如，输入402)的示例。在图3中，用户提供输入“缺页数，1993”和输出“1993”的组合作为行为示例(例如，行为示例401A)，而“64-67，1995”是输入(例如，输入402A)的一个示例。

[0057] 程序缩小系统410标识将符合(多个)行为示例的一组程序(动作503)。程序当其执行示例中示出的确切行为时“符合行为示例”。因此，在图2的示例中，基于输入“John Smith”而生成输出“JS”的任何程序是符合的程序。在图3的示例中，基于输入“缺页数，

1993”来生成输出“1993”的任何程序是符合的程序。也就是说，如先前提到的，不是符合示例的所有符合的程序实际上都遵循由行为示例所例示的一般意图。其中存在更有效地且高效地解决关于使用或建议符合的程序中的哪一个的歧义的需要。在图4中，符合性组件460基于行为示例来标识一组符合的程序，并将该组符合的程序提供(如由箭头414所表示的)给特征评估组件450。

[0058] 特征评估组件450然后评估各种符合的程序的特征以缩小该组可能的程序(动作505)。符合的程序的一些特征可以提前确定。例如，预构建程序的结构特征可以很好地提前被标识。

[0059] 备选地或另外，符合的程序的特征可以直到在符合的程序被标识之后才被标识。例如，对于未被预构建但响应于从用户接收到示例行为而被构建的符合的程序，程序结构和行为的方面将不必提前知晓。因此，存在发生在特征的评估之前的对程序进行特征化的动作(动作504)。对于一些类型的程序行为特征，诸如输出特征，候选程序实际上在(多个)输入上运行以验证输出的特征。因此，在这种情况下，具体地，输出的特征化不被提前执行，因为输出的特征取决于程序的运行来标识输出将是什么。

[0060] 传统上，特征评估组件仅指望程序结构特征以便缩小来自候选程序的初始列表的程序的列表。然而，根据本文中描述的原理，特征评估组件450评估程序的独立于程序结构的特征。例如，在图4中，特征评估组件450评估程序行为特征420，其可以包括(多个)执行追踪特征421和(多个)程序输出特征422中的一个或两者。(多个)程序结构特征430也可以作为因素被引入到该缩小操作中。

[0061] 作为缩小过程的一部分(在动作505中)，特征评估组件450可以将适合性水平分配给候选程序中的每一个。这样的适合性水平可以是一个或多个程序行为特征以及潜在的一个或多个程序结构特征的函数。每个特征可以例如具有与其相关联的加权。加权可以通过例如对一些训练数据应用机器学习以便合适地调节权重来确定。机器学习可以是导致加权被更新的恒定或频繁过程。在图4中，加权由权重470表示。

[0062] 最终，所缩小的程序列表中的程序之一被选择(动作506)，并且被启用以用于(动作507)，使得所选择的程序针对(多个)输入402运行以生成输出。例如，在图2的示例中，输出电子表格202可以使用所选择的表达203来生成。在图3的示例中，合适的程序可以针对输入“64-77,1995”来运行以生成文本“1995”。

[0063] 可以存在与合适的程序的选择(动作506)相关联的一些用户介入。毕竟，是用户他自己或她自己最熟悉关于要由程序执行并且他/她针对其提供行为示例的功能的一般意图。另外，对于用户而言通过审查程序(或关于程序的元数据)来解译程序正在做什么可能比使用户实际上写程序容易得多。

[0064] 如果所缩小的程序列表仅是单个程序，则系统可以在没有用户介入的情况下自动地实现程序，或者可以询问用户批准程序。例如，在图4中，在所缩小的一组403可能的程序中可以仅存在单个程序403A。

[0065] 如果所缩小的列表包括复数个程序(在所缩小的一组403可能的程序中存在复数个程序)，则程序可以被可视化给用户(可能根据所估计的适合性来排序)。椭圆403B表示可以存在被包括于所缩小的一组403程序中的各种不同数目的程序。

[0066] 如果甚至在所缩小的列表中存在太多这样的程序，则可能仅具有较高适合性的那

些程序被可视化。用户可以然后选择期望的程序。可视化可以包括程序做什么的一些描述以允许用户验证程序将根据用户的意图来工作。如果用户介入被使用，那么在检测到表示对所缩小的程序列表中的一个程序的选择的用户交互后，所选择的程序被启用并为用户针对剩余输入来运行(动作507)。

[0067] 因为更多信息用于缩小可能程序的列表，所以选择正确地遵循用户的一般意图的程序的机会更大。这将从以下对行为特征的描述中变得显而易见。

[0068] 程序行为特征包括例如执行追踪特征和程序输出特征。例如，在图4中，程序行为特征420包括(多个)执行追踪特征421和(多个)程序输出特征422。关于程序输出，一些程序输出比其他类型的程序输出更可能由合适的程序来产生。执行追踪特征的加权允许具有关于所提供的输入的病态行为的、看起来自然的程序被排除。在执行追踪特征上的偏向允许基于计算的序列的偏向，而不是仅生成其所产生的那些序列或输出的程序。

[0069] 可以经由对程序的执行的评估来提取任何数目的执行追踪特征。在一个具体示例中，使用三个执行追踪特征，包括：1)执行追踪是否具有重叠的子字符串提取(具有朝向具有该执行追踪特征的程序的选择的否定偏向)，2)执行追踪是否具有重复的子字符串提取(具有朝向具有该执行追踪特征的程序的选择的否定偏向)，3)执行追踪是否具有相邻的子字符串提取(具有朝向具有该执行追踪特征的程序的选择的肯定偏向)。

[0070] 图6图示了跟随执行追踪以找到重叠提取的一个示例。在这种情况下，具有重叠提取的程序提供与用户的意图不一致的结果。在图6中，存在包括列A(其包括一组名字)的电子表格，并且列B是输出。第一行示出了行为示例。“Richard”要被转换成“Mr.Richard”。用户可能想要通过此演示的一个一般行为是在名字前面放置合适的前缀(Mr.或Ms.)。

[0071] 在图6中，左边部分601示出了由用户输入的字段，并且右边部分602示出了在文本变换(由箭头603表示)之后得到的部分602的一个示例。在这种情况下，假设系统选择的变换是首先提取字符1，并且然后通过首先将“M”引入在所提取的字符的小写形式之后并且然后添加句点“.”符号来形成前缀。然后，字符1至7被提取并在一个空格之后被添加到该前缀。这当然符合所给出的示例，因为这将导致“Richard”被变换成“Mr.Richard”。然而，当相同的变换被应用到输入“Steve”时，结果是“Ms.Steve”。对输入“Angela”的变换将读取为“Ma.Angela”。

[0072] 该变换涉及重叠的提取。字符1被提取，然后提取字符1至7。字符1因此被提取两次。这是重叠的提取。然而，当仅对程序结构评估时，提取首字母的程序常常是优选的。访问数据的重叠区域很少是所意图的。通常，程序抽出它们想要的数据并且然后关于其进行一些处理，而不是重复地返回到已经看到的数据。简单地引入针对访问输入的重叠区域的归纳偏向足以使错误的程序不优选。

[0073] 类似地，用户通常不意图重复子字符串提取，尽管该否定偏向比重叠的提取的否定偏向少。另一方面，用户常常意图执行相邻字符串提取，并且因此肯定偏向可以被分配给这样的执行追踪。因此，系统可以应用取决于用户通常期望程序如何执行的适当偏向。因此，程序的执行追踪特征的使用可以改进示例编程中对适合程序的标识。

[0074] 程序输出特征的使用允许基于程序对由用户提供的输入所产生的输出来对适合程序的良好预测。这基于以下前提：一些组的输出比其他组的输出更不可能是适合程序的结果。更一般地，用户通常期望程序产生类似格式的输出，诸如全都是日期、自然数、或地

址。因此，可以存在被分配给表示输出中的不相似性的输出特征的否定偏向。系统对提供的所有输入(例如，图4中的(多个)输入402)运行每个候选程序，以生成系统然后对其特征化的一组输出。

[0075] 图7再次示出了一个电子表格示例——左边部分701表示由用户输入的部分，并且右边部分702表示在对输入的文本变换之后的部分的一个示例。变换由箭头703符号化。

[0076] 被选择用于执行变换的程序简单地将右方括号“]”添加在每项输入上以便形成输出。尽管用户所意图的是正确地括出所有输入，但是基线系统对最后一个输入不正确地插入了附加括号。该错误输出是看起来不同于其他程序输出的“异常值”。可能使不正确的输出不优选，因为其包含“异常值”。作为限制性情况场景，考虑与所提供的输入/输出示例一致的程序，但是对其他输入中的一些抛出异常。经验上这种情况常常发生，并且几乎确定不是用户所意图的程序。

[0077] 在一个实施例中，确定程序输出的聚类。具有特定输出模式的那些输出被分配到簇中。如果存在大量簇，那么生成那些许多簇的程序是不优选的。作为另一示例，如果存在恒定字符串(例如，许多输出对应于相同字符串)，那么其可以是不优选的。如果存在输出中的简单正则表达式，那么其对于程序输出可以是优选的。如果存在特殊正则表达式，如时间，那么该输出对于程序输出可以是非常优选的。

[0078] 输出特征的评估可以使用特定语义知识。例如，考虑图8。给定常见人类读者的语义知识，我们知道所意图的程序行为是从程序输入中提取城市和州。将输入识别为地址，并且将输出识别为城市和州(由连字符分开)。给定该语义知识，针对第二行输入的“Los-CA”的输出看上去是错误的，因为“Los”不是城镇的真实名称。同样地，“Angeles-CA”是错误的，因为“Angeles”同样不是城镇的真实名称。然而，“Los Angeles-CA”可能是正确的，因为“Los Angeles”是城市，并且是位于加利福尼亚的大城市。因此，输出的错误性可以基于语义知识来确定。

[0079] 因此，本文中描述的原理提供了用于基于示例编程环境中的相对少的示例来解决选择程序时的歧义的高效机制。这样的歧义具有在给定将符合的程序缩小到更可管理的列表时适当考虑程序行为特征的情况下解析出正确程序的更高机会。

[0080] 本发明可以在不脱离其精神或必要特性的情况下以其他具体形式来实施。所描述的实施例应在所有方面仅被认为是说明性的而非限制性的。本发明的范围因此由所附权利要求而非由前述描述指示。在权利要求的等价性的意义和范围内出现的所有改变应被包含在其范围内。

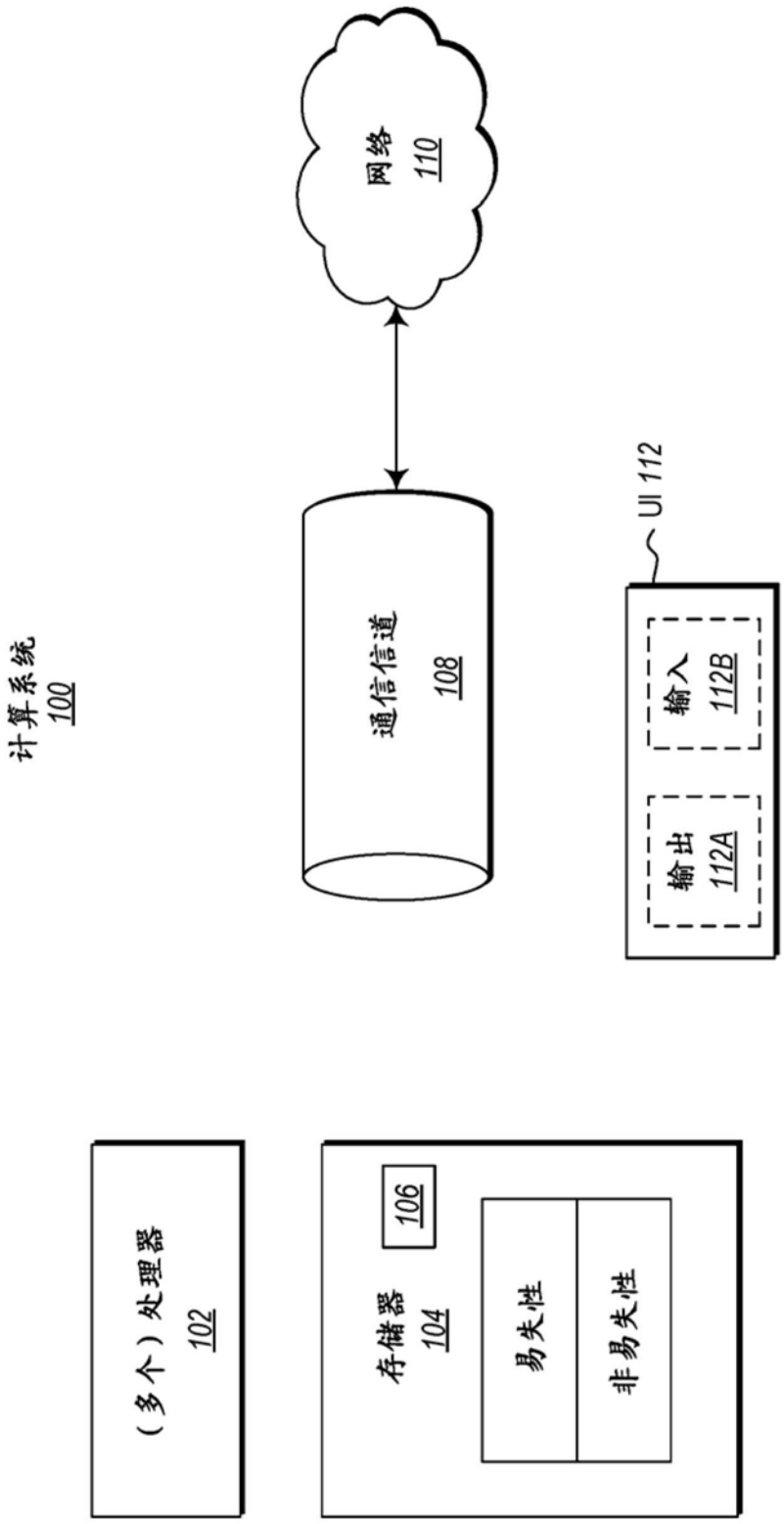

计算系统
100
网络
110
通信信道
108
UI 112
输出
112A
输入
112B
(多个)处理器
102
存储器
104
106
易失性
非易失性

图1

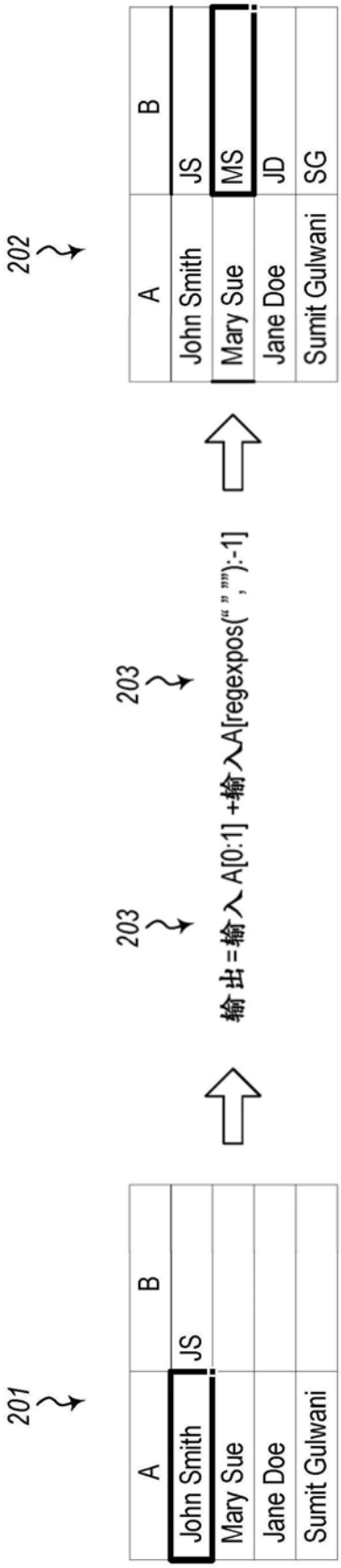

201
A
B
John Smith
JS
Mary Sue
Jane Doe
Sumit Gulwani
203
输出=输入A[0:1] +输入A[regexpos(" ","."):-1]
203
202
A
B
John Smith
JS
Mary Sue
MS
Jane Doe
JD
Sumit Gulwani
SG

图2

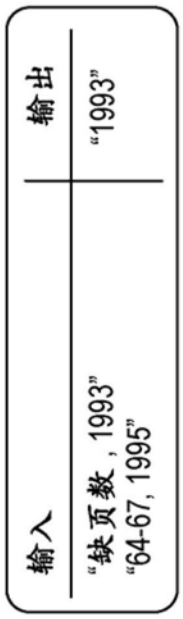


图3

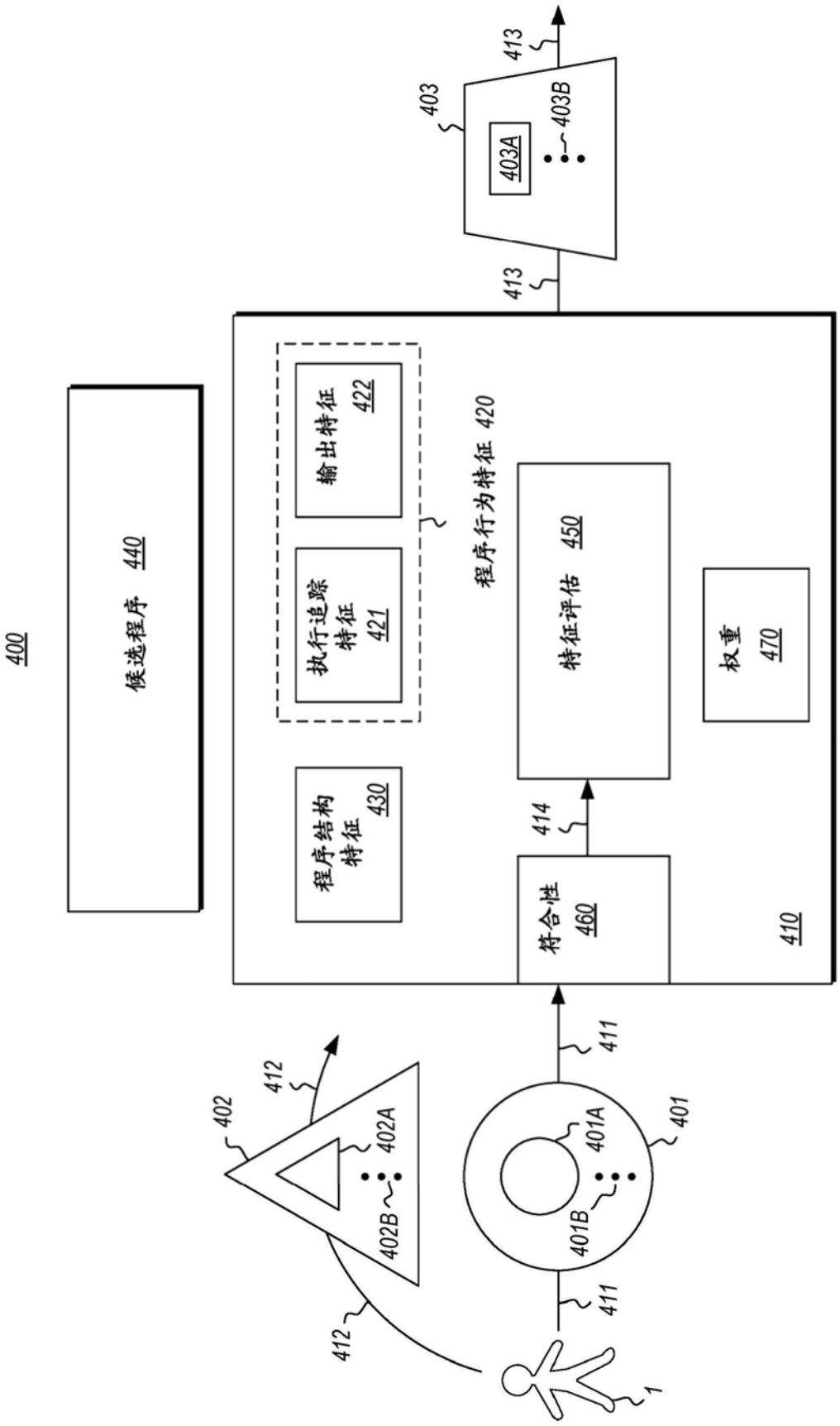

400
候选程序 440
程序行为特征 420
执行追踪特征 421
输出特征 422
程序结构特征 430
特征评估 450
权重 470
符合性 460
410
403
403A
403B
413
402
402A
402B
412
401
401A
401B
411
414
1

图4

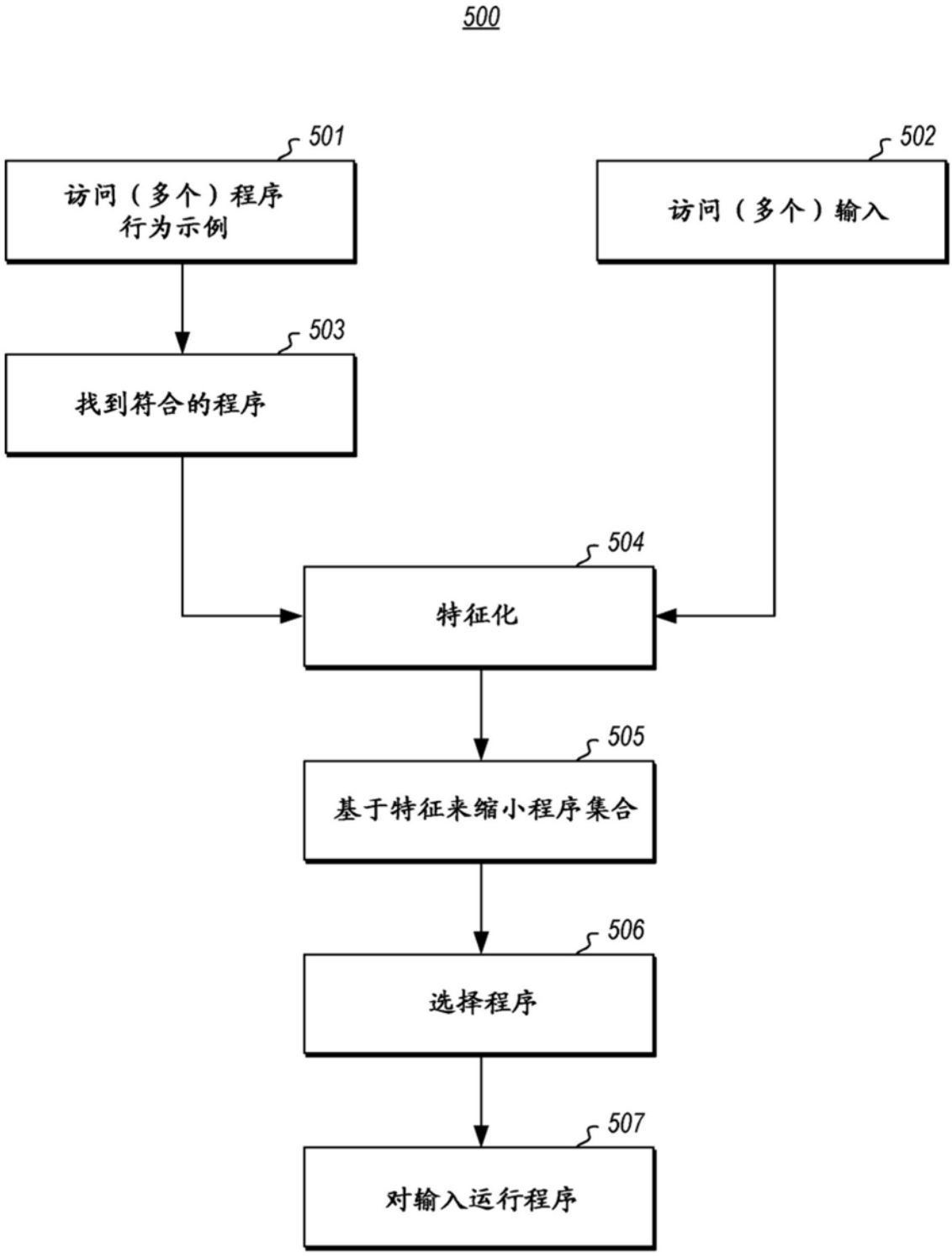

500
501
访问（多个）程序行为示例
502
访问（多个）输入
503
找到符合的程序
504
特征化
505
基于特征来缩小程序集合
506
选择程序
507
对输入运行程序

图5

601

D	E
Richard	Mr. Richard
Steve	

603

602

D	E
Richard	Mr. Richard
Steve	Ms. Steve

图6

701

	A	B
1	[CPT-00350	[CPT-00350]
2	[CPT-00340	
3	[CPT-11536	
4	[CPT-11222	
5	[CPT-115]	

703

702

	A	B
1	[CPT-00350	[CPT-00350]
2	[CPT-00340	[CPT-00340]
3	[CPT-11536	[CPT-11536]
4	[CPT-11222	[CPT-11222]
5	[CPT-115]	[CPT-115]]

图7

程序输入	程序输出
457 124th St S, Seattle, WA 98111 98743 Edwards Ave, Los Angeles, CA 78911 One Microsoft Way, Redmond, WA 98052 11 Main, Bizmark, ND 54891	Seattle-WA

图8