(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2024 年 12 月 19 日 (19.12.2024)

WIPO | PCT

(10) 国际公布号
WO 2024/255477 A1

(51) 国际专利分类号:
G06F 8/41 (2018.01)

(21) 国际申请号: PCT/CN2024/090980

(22) 国际申请日: 2024 年 4 月 30 日 (30.04.2024)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
202310709070.3 2023年6月14日 (14.06.2023) CN

(71) 申请人: 中兴通讯股份有限公司 (**ZTE CORPORATION**) [CN/CN]; 中国广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦, Guangdong 518057 (CN)。 北京大学 (**PEKING UNIVERSITY**) [CN/CN]; 中国北京市海淀区颐和园路5号, Beijing 100871 (CN)。

(72) 发明人:彭星海 (**PENG, Xinghai**); 中国广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦, Guangdong 518057 (CN)。 刘东 (**LIU, Dong**); 中国广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦, Guangdong 518057 (CN)。 朱琪豪 (**ZHU, Qihao**); 中国广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦, Guangdong 518057 (CN)。 曹继承 (**CAO, Jicheng**); 中国广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦, Guangdong 518057 (CN)。 程圣宇 (**CHENG, Shengyu**); 中国广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦, Guangdong 518057 (CN)。 熊英飞 (**XIONG, Yingfei**); 中国广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦,

(54) **Title:** REPAIR PATCH GENERATION METHOD, STORAGE MEDIUM, AND ELECTRONIC DEVICE

(54) 发明名称: 修复补丁生成方法、存储介质及电子装置

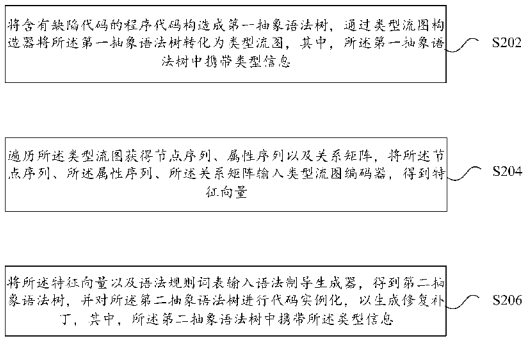


图 2

S202 Construct a program code comprising a defect code into a first abstract syntax tree, and convert the first abstract syntax tree into a type flow graph by means of a type flow graph constructor, wherein the first abstract syntax tree carries type information

S204 Traverse the type flow graph to obtain a node sequence, an attribute sequence and a relation matrix, and input the node sequence, the attribute sequence and the relation matrix into a type flow graph encoder to obtain a feature vector

S206 Input the feature vector and a grammar rule word list into a syntax-directed generator to obtain a second abstract syntax tree, and perform code instantiation on the second abstract syntax tree to generate a repair patch, wherein the second abstract syntax tree carries the type information

(57) **Abstract:** Embodiments of the present disclosure provide a repair patch generation method, a storage medium, and an electronic device. The method comprises: constructing a program code comprising a defect code into a first abstract syntax tree, and converting the first abstract syntax tree into a type flow graph by means of a type flow graph constructor, wherein the first abstract syntax tree carries type information; traversing the type flow graph to obtain a node sequence, an attribute sequence and a relation matrix, and inputting the node sequence, the attribute sequence and the relation matrix into a type flow graph encoder to obtain a feature vector; and inputting the feature vector and a grammar rule word list into a syntax-directed generator to obtain a second abstract syntax tree, and performing code instantiation on the second abstract syntax tree to generate a repair patch, wherein the second abstract syntax tree carries the type information.

(57) 摘要: 本公开实施例提供了一种修复补丁生成方法、存储介质及电子装置。该方法包括：将含有缺陷代码的程序代码构造成第一抽象语法树，通过类型流图构造器将所述第一抽象语法树转化为类型流图，其中，第一抽象语法树中携带类型信息；遍历类型流图获得节点序列、属性序列以及关系矩阵，将节点序列、属性序列、关系矩阵输入类型流图编码器，得到特征向量；将特征向量以及语法规则词表输入语法制导生成器，得到第二抽象语法树，并对第二抽象语法树进行代码实例化，以生成修复补丁，其中，所述第二抽象语法树中携带所述类型信息。

WO 2024/255477 A1

[见续页]

Guangdong 518057 (CN)。 肖元安(**XIAO, Yuanan**); 中国广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦, Guangdong 518057 (CN)。

(74) 代理人: 北京康信知识产权代理有限责任公司(**KANGXIN PARTNERS , P.C.**); 中国北京市海淀区知春路甲48号盈都大厦A座16层, Beijing 100098 (CN)。

(81) 指定国(除另有指明，要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

(84) 指定国(除另有指明，要求每一种可提供的地区保护): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:
— 包括国际检索报告(条约第21条(3))。

修复补丁生成方法、存储介质及电子装置

相关申请的交叉引用

本公开基于2023年6月14日提交的发明名称为“修复补丁生成方法、存储介质及电子装置”的中国专利申请CN202310709070.3，并且要求该专利申请的优先权，通过引用将其所公开的内容全部并入本公开。

技术领域

本公开实施例涉及计算机技术领域，具体而言，涉及一种修复补丁生成方法、存储介质及电子装置。

背景技术

目前，基于深度学习的语法制导的程序缺陷修复方法，仅能保证生成的修复补丁符合语法规则，但深度学习模型中的神经网络无法感知到程序的类型规则，容易生成大量的类型不正确补丁，尽管这些补丁最终可以被编译器过滤掉，但在编译大型工程时，过滤这些类型不正确补丁需要耗费大量的时间，缺陷修复系统的效率较低；而使用搜索算法来确定概率较高的语法规则路径，神经网络还可能未类型不正确的语法规则赋予较高概率，如此，可能将正确补丁排除在候选补丁之外，导致无法修复缺陷。

如何抽象代码的类型、神经网络如何表示并学习类型规则是有待解决的技术问题。

发明内容

本公开实施例提供了一种修复补丁生成方法、存储介质及电子装置，以至少解决相关技术中已有的深度学习缺陷代码修复方法缺少学习类型信息能力的问题。

根据本公开的一个实施例，提供了一种修复补丁生成方法，包括：将含有缺陷代码的程序代码构造成第一抽象语法树，通过类型流图构造器将所述第一抽象语法树转化为类型流图，其中，所述第一抽象语法树中携带类型信息；遍历所述类型流图获得节点序列、属性序列以及关系矩阵，将所述节点序列、所述属性序列、所述关系矩阵输入类型流图编码器，得到特征向量；将所述特征向量以及语法规则词表输入语法制导生成器，得到第二抽象语法树，并对所述第二抽象语法树进行代码实例化，以生成修复补丁，其中，所述第二抽象语法树中携带所述类型信息。

根据本公开的又一个实施例，还提供了一种计算机可读存储介质，所述计算机可读存储介质中存储有计算机程序，其中，所述计算机程序被设置为运行时执行上述任一项方法实施例中的步骤。

根据本公开的又一个实施例，还提供了一种电子装置，包括存储器和处理器，所述存储器中存储有计算机程序，所述处理器被设置为运行所述计算机程序以执行上述任一项方法实施例中的步骤。

附图说明

图 1 是本公开实施例的运行修复补丁生成方法的计算机终端的硬件结构框图；

图 2 根据本公开实施例的修复补丁生成方法的流程图；

图 3 是根据本公开实施例的修复补丁生成装置的结构框图；

图 4 是根据本公开实施例的修复补丁生成流程的示意图；

图 5 是根据本公开实施例的类型流图编码器内部结构的示意图；

图 6 是根据本公开场景实施例的基于类型约束的程序缺陷自动修复方法的流程图；

图 7 是根据本公开实施例的抽象语法树及缺陷代码的示意图；

图 8 是根据本公开实施例的类型流图的示意图。

具体实施方式

下文中将参考附图并结合实施例来详细说明本公开的实施例。

本公开的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象，而不必用于描述特定的顺序或先后次序。

本申请实施例中所提供的方法实施例可以在移动终端、计算机终端或者类似的运算装置中执行。以运行在计算机终端上为例，图 1 是本公开实施例的运行修复补丁生成方法的计算机终端的硬件结构框图。如图 1 所示，计算机终端可以包括一个或多个（图 1 中仅示出一个）处理器 102（处理器 102 可以包括但不限于微处理器或可编程逻辑器件的处理装置）和用于存储数据的存储器 104，其中，上述计算机终端还可以包括用于通信功能的传输设备 106 以及输入输出设备 108。本领域普通技术人员可以理解，图 1 所示的结构仅为示意，其并不对上述计算机终端的结构造成限定。例如，计算机终端还可包括比图 1 中所示更多或者更少的组件，或者具有与图 1 所示不同的配置。

存储器 104 可用于存储计算机程序，例如，应用软件的软件程序以及模块，如本公开实施例中的修复补丁生成方法对应的计算机程序，处理器 102 通过运行存储在存储器 104 内的计算机程序，从而执行各种功能应用以及数据处理，即实现上述的方法。存储器 104 可包括高速随机存储器，还可包括非易失性存储器，如一个或者多个磁性存储装置、闪存、或者其他非易失性固态存储器。在一些实例中，存储器 104 可进一步包括相对于处理器 102 远程设置的存储器，这些远程存储器可以通过网络连接至计算机终端。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

传输设备 106 用于经由一个网络接收或者发送数据。上述的网络具体实例可包括计算机终端的通信供应商提供的无线网络。在一个实例中，传输设备 106 包括一个网络适配器（Network Interface Controller，简称为 NIC），其可通过基站与其他网络设备相连从而可与互联网进行通讯。在一个实例中，传输设备 106 可以为射频（Radio Frequency，简称为 RF）模块，其用于通过无线方式与互联网进行通讯。

在本实施例中提供了一种运行于上述计算机终端的修复补丁生成方法，图 2 根据本公开实施例的修复补丁生成方法的流程图，如图 2 所示，该流程包括如下步骤：

步骤 S202，将含有缺陷代码的程序代码构造成第一抽象语法树，通过类型流图构造器将所述第一抽象语法树转化为类型流图，其中，所述第一抽象语法树中携带类型信息；

在本实施例中，抽象类型系统定义了一套规则，可以对程序源码的变量、常量、操作符

的类型进行抽象，从而将程序源码转换为带类型信息的抽象语法树。因此，在步骤 S202 之前，该方法还包括：通过抽象类型系统将程序源码语料转换为多棵第三抽象语法树，其中，所述第三抽象语法树中携带所述类型信息；遍历所有所述第三抽象语法树得到所述语法规则词表，其中，所述语法规则词表中携带所述类型信息，该词表可用于后续神经网络生成修复代码抽象语法树。其中，该规则也用于将含有缺陷代码的程序代码构造成第一抽象语法树。

在一个实施例中，抽象类型系统定义的规则包括：1）将程序代码或所述程序源码语料转换为抽象语法树，程序代码或所述程序源码语料语句中的变量、常量、操作符是抽象语法树的叶子节点；2）定义五种抽象类型：通用类型、字符串类型、数值类型、布尔类型和空类型，其中，空类型是字符串类型、数值类型、布尔类型的子类型；通用类型是字符串类型、数值类型、布尔类型的父类型；通用类型的父类型是通用类型，空类型的子类型是空类型；3）根据上述定义的类型，程序源码中的变量或常量可抽象为字符串类型、数值类型、布尔类型中的一种，NULL 归入空类型；4）操作符的类型由它的操作数类型决定，如果一个操作符的所有操作数类型相同，则操作符类型与操作数类型一样；如果操作数类型有两种及以上，则操作符类型是所有操作数类型的父类型；5）抽象语法树的非叶子节点类型由其子节点类型决定，如果一个非叶子节点的所有子节点类型相同，则该节点的类型与其子节点类型一样；如果子节点类型有两种及以上，则该节点类型是所有子节点类型的父类型。

通过抽象类型系统定义的规则，可对程序源码或缺陷代码中的变量、常量、操作符的类型进行抽象，并在构造抽象语法树时，将抽象的各类型信息对应地携带于抽象语法树上，从而得到一种带类型的抽象语法树。

在本实施例的步骤 S202 中，通过类型流图构造器将所述第一抽象语法树转化为类型流图，包括：将所述第一抽象语法树中的各节点转换为所述类型流图中抽象语法树节点；将所述程序代码中的变量抽象为所述类型流图中的变量节点；其中，所述抽象语法树节点之间、变量节点之间、对应于同一变量的所述抽象语法树节点和所述变量节点之间，由不同属性的边连接。

其中，所述抽象语法树节点之间的边的属性包括：父-子、子-父、左兄弟、右兄弟；所述变量节点之间的边的属性包括：相同类型、兼容类型、不兼容类型；对应于同一变量的连接所述抽象语法树节点和所述变量节点之间的边的属性包括：申明-变量、使用-变量、变量-申明、变量-使用。

步骤 S204，遍历所述类型流图获得节点序列、属性序列以及关系矩阵，将所述节点序列、所述属性序列、所述关系矩阵输入类型流图编码器，得到特征向量；

在本实施例中，类型流图编码器为多个神经网络层的有机堆叠，其中，多个神经网络层包括：嵌入层、类型敏感的自注意神经元、门控层、线性层，除嵌入层的输入为外界输入的节点序列、属性序列以及关系矩阵外，其余神经网络层的输入为前一个神经网络层的输出。

在一个示例性实施例中，遍历所述类型流图获得节点序列，包括：遍历所述类型流图中所述第一抽象语法树的抽象语法树节点名称，将所有所述抽象语法树节点名称组成第一序列片段；按照所述程序代码上下文中各变量的出现顺序，遍历所述第一抽象语法树中的变量节点名称，将缺陷代码处可访问的变量节点名称组成第二序列片段；将所述第一序列片段和所述第二序列片段进行拼接，得到所述节点序列。

在一个示例性实施例中，遍历所述类型流图获得属性序列，包括：按照所述节点序列中

抽象语法树节点和变量节点的排列顺序，依次获取各所述抽象语法树节点和各所述变量节点对应的节点属性值，并将所有所述节点属性值组合为所述属性序列。

在本实施例中，所述抽象语法树节点包括以下属性：类型属性、可复制性；所述变量节点包括以下属性：类型属性、可访问性。

在本实施例的步骤S204中，将所述节点序列、所述属性序列、所述关系矩阵输入类型流图编码器，得到一个特征向量，包括：通过所述类型流图编码器中的嵌入层，基于输入的所述节点序列、所述属性序列、所述关系矩阵，得到节点嵌入向量、属性嵌入向量和关系嵌入向量；通过所述类型流图编码器中的类型敏感的自注意神经元，对所述节点嵌入向量和所述关系嵌入向量进行融合，得到第一融合向量；通过所述类型流图编码器中的门控层，对所述属性嵌入向量和所述第一融合向量进行融合，得到第二融合向量；通过所述类型流图编码器中的线性层，对所述第二融合向量进行线性变换得到所述特征向量。

通过类型流图及其构造规则，可记录带类型的抽象语法树节点与用户自定义的变量节点之间的邻接关系，实现了将程序源码或缺陷代码的语法规则和类型规则进行了有机融合。

步骤S206，将所述特征向量以及语法规则词表输入语法制导生成器，得到第二抽象语法树，并对所述第二抽象语法树进行代码实例化，以生成修复补丁，其中，所述第二抽象语法树中携带所述类型信息。

通过类型流图编码器及其神经网络架构，通过提取类型流图的节点序列、属性序列和关系矩阵，对程序源码的语法规则和类型规则利用神经网络进行编码，可在补丁生成过程中赋予修复补丁（代码）类型约束。

在一个示例性实施例中，所述特征向量的值为浮点数。

在本实施例的步骤S206中，将所述特征向量以及语法规则词表输入语法制导生成器，得到第二抽象语法树，包括：将所述特征向量和所述语法规则词表输入语法制导生成器，预测第一语法规则，并根据所述第一语法规则进行抽象语法树的编辑操作，得到第一部分抽象语法树，其中，所述第一语法规则携带类型信息；基于所述第一部分抽象语法树、所述特征向量以及所述语法规则词表，预测第二语法规则，并根据所述第二语法规则进行抽象语法树的编辑操作，得到第二部分抽象语法树，其中，所述第二语法规则携带类型信息；迭代进行预测语法规则步骤和编辑抽象语法树步骤，直到得到完整的所述第二抽象语法树。

通过上述步骤，将缺陷代码构造成可携带类型信息的第一抽象语法树，实现了对代码类型进行抽象；而由第一抽象语法树转化而来的类型流图中，可将程序代码中的语法规则与类型规则进行有机融合，进而，基于类型流图得到的特征向量与语法规则词表输入语法制导生成器而得到的第二抽象语法树将携带类型信息，而由第二抽象语法树得到的修复补丁代码便会被赋予类型约束。因此，可以解决相关技术中已有的深度学习缺陷代码修复方法缺少学习类型信息能力的问题，达到扩展已有的深度学习缺陷代码修复方法的能力的效果。

通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到根据上述实施例的方法可借助软件加必需的通用硬件平台的方式来实现，当然也可以通过硬件，但很多情况下前者是更佳的实施方式。基于这样的理解，本公开的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质（如ROM/RAM、磁碟、光盘）中，包括若干指令用以使得一台终端设备（可以是手机，计算机，服务器，或者网络设备等）执行本公开各个实施例所述的方法。

在本实施例中还提供了一种修复补丁生成装置，该装置用于实现上述实施例及优选实施方式，已经进行过说明的不再赘述。如以下所使用的，术语“模块”可以实现预定功能的软件和/或硬件的组合。尽管以下实施例所描述的装置较佳地以软件来实现，但是硬件，或者软件和硬件的组合的实现也是可能并被构想的。

图 3 是根据本公开实施例的修复补丁生成装置的结构框图，如图 3 所示，该装置包括：转化模块 10、遍历模块 20 以及生成模块 30。

转化模块 10，设置为将含有缺陷代码的程序代码构造成第一抽象语法树，通过类型流图构造器将所述第一抽象语法树转化为类型流图，其中，所述第一抽象语法树中携带类型信息；

遍历模块 20，设置为遍历所述类型流图，获得节点序列、属性序列以及关系矩阵，将所述节点序列、所述属性序列、所述关系矩阵输入类型流图编码器，得到一个特征向量；

生成模块 30，设置为将所述特征向量以及语法规则词表输入语法制导生成器，得到第二抽象语法树，并对所述第二抽象语法树进行代码实例化，以生成修复补丁，其中，所述第二抽象语法树中携带所述类型信息。

上述各个模块是可以通过软件或硬件来实现的，对于后者，可以通过以下方式实现，但不限于此：上述模块均位于同一处理器中；或者，上述各个模块以任意组合的形式分别位于不同的处理器中。

为便于对本公开所提供的技术方案的理解，下面将结合具体场景的实施例进行详细的阐述。

图 4 是根据本公开实施例的修复补丁生成流程的示意图，如图 4 所示，该流程中涉及以下模块：抽象类型系统、类型流图构造器、类型流图编码器、语法制导生成器。

在一个实施例中，抽象类型系统定义了一套规则，可以对程序源码的变量、常量、操作符的类型进行抽象，将程序源码转换为带类型信息的抽象语法树。

类型流图构造器，用于将抽象语法树转换为类型流图，用于记录带类型的抽象语法树节点、用户自定义变量节点之间的邻接关系；基于类型流图及其构造规则，本公开实施例可将程序源码的语法规则和类型规则进行有机融合。

类型流图编码器，用于基于类型流图中记录的节点序列、属性序列和关系矩阵生成特征向量，通过提取类型流图的节点序列、属性序列和关系矩阵，对程序源码的语法规则和类型规则利用神经网络进行编码，可在补丁生成过程中对修复补丁赋予类型约束。

在一个实施例中，抽象类型系统定义了以下五条规则：

1）将程序源码转换为抽象语法树，程序源码语句中的变量、常量、操作符是抽象语法树的叶子节点。

2）定义五种抽象类型：通用类型、字符串类型、数值类型、布尔类型和空类型。其中，空类型是字符串类型、数值类型、布尔类型的子类型；通用类型是字符串类型、数值类型、布尔类型的父类型；通用类型的父类型是通用类型，空类型的子类型是空类型。

3）根据上述定义的类型，程序源码中的变量或常量可抽象为字符串类型、数值类型、布尔类型中的一种，NULL 归入空类型。

4）操作符的类型由它的操作数类型决定，如果一个操作符的所有操作数类型相同，则操作符类型与操作数类型一样；如果操作数类型有两种及以上，则操作符类型是所有操作数类型的父类型。

5）抽象语法树的非叶子节点类型由其子节点类型决定，如果一个非叶子节点的所有子节点类型相同，则该节点的类型与其子节点类型一样；如果子节点类型有两种及以上，则该节点类型是所有子节点类型的父类型。

在一个实施例中，类型流图是一种图的数据结构，包含节点集合和边集合。类型流图的构造规则如下：

1）节点集合包含抽象语法树节点子集（其中的元素我们称为 Anode）和变量节点子集（其中的元素我们称为 Vnode）。

2）每个 Anode 拥有两个属性：类型属性和可复制性属性，类型属性的取值是五种抽象类型的一种；可复制属性是一个布尔值，表示以该节点为根节点的抽象语法子树是否依赖于局部上下文的变量，以便该抽象语法子树可以独立复制到其他位置。

3）每个 Vnode 表示上下文的一个用户自定义元素（变量或参数），上下文是指抽象语法树对应的程序源码范围。每个 Vnode 拥有两个属性：类型属性和可访问性属性，类型属性的值是该元素定义时的实际类型；可访问性属性是一个布尔值，表示在缺陷代码位置是否可以访问该元素。

4）边集合包含 Anode-Anode，Anode-Vnode，Vnode-Anode 和 Vnode-Vnode 四个子集。边集合中的边是有向的，拥有一个属性标签，具体的边属性信息如表 1 所示。

表 1

节点x	节点y	边属性序号	边属性标签	描述
Anode	Anode	1 2 3 4	父-子 子-父 左兄弟 右兄弟	x是y的父节点 x是y的子节点 x是y的左兄弟节点 x是y的右兄弟节点
Anode	Vnode	5 6	申明-变量 使用-变量	x是y的申明 x使用y
Vnode	Anode	7 8	变量-申明 变量-使用	y是x的申明 y使用x
Vnode	Vnode	9 10 11	相同类型 兼容类型 不兼容类型	x和y的类型相同 x与y的类型兼容 x与y的类型不兼容

5）Anode-Anode 集合中的边表示的是抽象语法树节点之间的语法关系，两个 Anode 之间的关系是父-子、子-父、左兄弟、右兄弟四种关系之一。

6）Vnode-Anode 集合中的边表示的是上下文的变量和抽象语法树节点之间的关系，如果变量 x 的申明语句对应抽象语法子树的根节点是 y，那么 x 有一条“变量-申明”属性的边连接到 y；同理，如果抽象语法树节点 m 使用到变量 n，那么 n 有一条“变量-使用”属性的边连接到 m。

7）Anode-Vnode 集合中的边是 Vnode-Anode 集合中对应边的反转。

8）Vnode-Vnode 集合中的边表示的是两个变量类型之间的兼容关系。

在一个实施例中，类型流图构造器将带类型的抽象语法树转换为满足上述规则的类型流

图。

在一个实施例中，类型流图编码器是一系列神经网络层的有机堆叠，作用是通过学习将类型流图信息编码为一个特征向量，其内部结构如图 5 所示。

如图 5 所示，类型流图编码器的输入是属性序列、节点序列和关系矩阵，输出是特征向量。三个输入分别通过各自的嵌入层得到属性嵌入向量、节点嵌入向量和关系嵌入向量。节点嵌入向量和关系嵌入向量通过类型敏感的自注意神经元进行融合，然后和属性嵌入向量通过门控层，最后通过线性层做线性变换。

在本实施例中，嵌入层、类型敏感的自注意神经元、门控层和线性层均来自业界公开的神经网络单元，它们一起构成了神经网络块。

在本实施例中，类型流图编码器由 N 个神经网络块串联而成，其中，N 为大于 0 的整数，除第一个神经网络块的输入是属性序列、节点序列和关系矩阵的嵌入向量，其他神经网络块的输入都是前一个神经网络块的输出。

图 6 是根据本公开场景实施例的基于类型约束的程序缺陷自动修复方法的流程图，以图 7(a)中的 Java 示例缺陷代码片段（以下简称为缺陷代码）为例，下面详细描述通过本实施例的程序缺陷自动修复方法修复图 7(a)中的缺陷。

在本实施例中，缺陷代码中以“-”开头的是缺陷代码行，以“+”开头的是对应的修复代码行。

如图 7（a）所示，在缺陷代码中定义了四个变量：loc0、loc 1、loc2、loc3，其中，loc 0 为 boolean 类型，其余三个变量是 double 类型。变量 loc0、loc1 和 loc2 在缺陷代码行之前定义，因此在缺陷代码处可以访问；变量 loc3 在缺陷代码行之后定义，在缺陷代码处不能访问。

在一个实施例中，如图 6 所示，本实施例中的基于类型约束的程序缺陷自动修复方法包括以下步骤：

S601，收集大量的程序源码语料，通过抽象类型系统转换为多棵带类型的抽象语法树，通过遍历抽象语法树，得到带类型的语法规则词表。

在本实施例中，对大量的程序源码进行提取，可以得到多棵带类型的抽象语法树，对每一棵带类型的抽象语法树进行遍历，可以得到多条语法规则。将这些语法规则去重后，组合起来得到一个带类型的语法规则词表，该词表可供后续神经网络在生成修复代码的抽象语法树时使用。

S602，缺陷代码通过抽象类型系统得到带类型的抽象语法树，然后通过类型流图构造器得到类型流图。

在本实施例中，使用抽象类型系统将缺陷代码转换为带类型的抽象语法树，得到的抽象语法树如图 7(b)所示，其中，抽象语法树中的类型信息的确定如下：

例如：根据抽象类型系统的定义，操作符的类型要么和所有操作数的类型一致，要么是所有操作数类型的父类型。操作符“-”有两个操作数“loc2”和“1”，“loc2”的实际类型是 double，抽象为数值类型；“1”是数值常量，也抽象为数值类型，因此，操作符“-”的类型与它的两个操作数类型保持一致，即数值类型。抽象语法树中其他操作符类型的推导类似。

又例如，根据抽象类型系统的定义，父节点的类型要么和所有子节点类型相同，要么是所有子节点类型的父类型，如图 7（b）所示，父节点“BinOp_N”有三个子节点“L-Op_N”、“Operator_N”和“R-Op_N”，这三个子节点都是数值类型，因此父节点“BinOp_N”也是数值类型，其他父节点的类型推导类似。

与步骤 S601 相同，对步骤 S602 中得到的带类型的抽象语法树进行子节点遍历，可以得到一系列的语法规则，例如："Assignment_N -> L-Expr_N AssignOp_N R-Expr_N"，它表示可以从"Assignment_N"节点推出"L-Expr_N"、"AssignOp_N"和"R-Expr_N"三个子节点。

根据类型流图的定义，我们将图 7(b)的带类型的抽象语法树转换为类型流图，得到的类型流图如图 8 所示。

如图 8 所示，类型流图中包含两类节点：

(1)抽象语法树节点 Anode（图 8 中以矩形框表示），它拥有类型属性和可复制性属性。

类型属性取值为抽象类型系统定义的五种类型之一，即通用类型、字符串类型、数值类型、布尔类型和空类型；

可复制性属性取值为 0 或 1,取值 0 表示该节点依赖于上下文的变量，以该节点为根节点的子树无法复制到其他地方复用。以"BinOp_N"节点为例，它的类型属性是数值类型，它的可复制性属性为 0，因为该节点依赖于变量"loc2"。

(2)另一类节点是变量节点 Vnode（图 8 中以椭圆框表示），它拥有类型属性和可访问性属性。

类型属性的取值与变量的实际类型一致，可访问性属性取值为 0 或 1，取值 0 表示在缺陷代码行处无法访问该变量。以椭圆节点"loc3"为例，它的类型是 double，它的可访问属性为 0，因为该节点对应的变量语句"double loc3 = 0"定义在缺陷代码"loc1 = loc2 - 1"之后，在缺陷代码处无法访问该变量。

类型流图还包含四类共 11 种属性的边，例如：如图 8 所示：

节点"BinOp_N"到节点"L-Op_N"的边属于 Anode-Anode 父子属性，对应表 1 的属性编号为 1；

矩形节点"loc1"到椭圆节点"loc1"的边属于 Anode-Vnode 使用变量属性，表示矩形节点"loc1"使用了椭圆节点"loc1"，对应表 1 的属性编号为 6；

椭圆节点"loc1"到矩形节点"loc1"的边属于 Vnode-Anode 变量使用属性，表示矩形节点"loc1"使用了椭圆节点"loc1"，对应表 1 的属性编号为 8；矩形节点"loc2"到椭圆节点"loc2"之间的两个边同理；

椭圆节点"loc1"到椭圆节点"loc2"边的边属性、椭圆节点"loc1"到椭圆节点"loc3"边、椭圆节点"loc2"到椭圆节点"loc3"边的边属性均表示类型相同；

椭圆节点"loc0"到椭圆节点"loc1"、椭圆节点"loc2"、椭圆节点"loc3"之间的边属性均表示类型不兼容，例如：椭圆节点"loc1"和椭圆节点"loc2"的类型相同，椭圆节点"loc2"的类型 double 和椭圆节点"loc0"的类型 boolean 不兼容。还有另外一种边属性是类型兼容，例如在 C 语言中，类型 int 兼容类型 int64。

S603，先序遍历类型流图中的节点名称，得到节点序列的第一部分，即 Anode 序列；再按照缺陷代码上下文中变量出现顺序遍历变量名称，基于变量名称确定节点序列的第二部分，即 Anode 序列，将两部分序列拼接得到完整的节点序列。

在一个实施例中，对图 8 的类型流图进行遍历可以得到节点序列、属性序列、关系矩阵三个数据，包括：

对图 8 中的带类型抽象语法树节点进行先序遍历，得到 Anode 序列：

[Assignment_N, L-Expr_N, loc1, AssignOp_N, =, R-Expr_N, BinOp_N, L-Op_N, loc2,

Operator_N，-，R-Op_N，1]；

由于椭圆节点“loc3”的可访问性属性为 0，缺陷代码行无法访问变量 loc3，因此，Vnode 序列不包含椭圆节点“loc3”，即 Vnode 序列为：

[loc0， loc1， loc2]。

即 Vnode 序列为截取缺陷代码处可以访问的变量名称序列片段。

节点序列是 Anode 序列和 Vnode 序列的拼接，即：

[Assignment_N，L-Expr_N，loc1，AssignOp_N，=，R-Expr_N，BinOp_N，L-Op_N，loc2，Operator_N，-，R-Op_N，1，loc0，loc1，loc2]。

S604，按照节点序列的顺序获取各节点的节点属性值，得到属性序列。

基于上述节点序列并按照节点序列的顺序，将节点序列中各变量替换为相对应的类型，可得到一个属性序列：[(数值，0)，(数值，0)，(数值，0)，(数值，1)，(数值，1)，(数值，0)，(数值，0)，(数值，0)，(数值，0)，(数值，1)，(数值，1)，(数值，1)，(数值，1)， (boolean，1)，(double，1)，(double，1)]。

其中，前 13 个为 Anode 节点，包含类型属性和可复制性属性，“0”代表不可复制，“1”代表可复制；后 3 个为 Vnode 节点，包含类型属性和可访问性属性，“1”代表可访问。

S605，获取类型流图节点的关系矩阵；

在一个实施例中，类型流图的关系矩阵是该类型流图的各节点之间的邻接矩阵，矩阵尺寸为 16x16，部分关系矩阵如表 2 所示。关系矩阵的行和列都是节点序列，如果两个节点邻接，则矩阵的值是对应两个节点边的属性编号，取值范围为 1 到 11，如表 1 所示。如果两个节点不邻接，矩阵的取值为 0，节点自身与自身之间视为不邻接。由于类型流图的边是有向的，因此，关系矩阵的取值从行往列看。

例如：表 2 的第 1 行第 2 列（不算表头）的值为 1，表示节点“Assignment_N”到节点“L-Expr_N”是父子节点关系。

表 2

	Assignment_N	L-Expr_N	loc1	……	loc2
Assignment_N	0	1	0	……	0
L-Expr_N	2	0	1	……	0
loc1	0	2	0	……	0
……	……	……	……	……	……
loc2	0	0	0	……	0

S606，将节点序列、属性序列、关系矩阵通过类型流图编码器得到特征向量。

在一个实施例中，将上述得到的节点序列、属性序列、关系矩阵输入图 4 的类型流图编码器，节点序列、属性序列、关系矩阵首先通过嵌入层分别得到节点嵌入向量、属性嵌入向量和关系嵌入向量，这三个嵌入向量均是浮点数矩阵。

节点嵌入向量和关系嵌入向量通过类型敏感的自注意神经元进行融合，融合后的向量与属性嵌入向量通过门控层融合，最后通过线性层做线性变换，输出第一个神经网络块的特征向量。

整个类型流图编码器由 N 个神经网络块串联构成，在本实施例中，N 的取值为 8，后一个神经网络块的三个输入通过复制前一个神经网络块的输出得到，最后一个神经网络块的输出是整个类型流图编码器的输出，即输出一个特征向量，值为浮点数。

S607，将特征向量、语法规则词表输入语法制导生成器，预测得到一条带类型的语法规则，表示一个对抽象语法树节点的编辑操作。

在一个实施例中，语法制导生成器根据输入的特征向量和带类型的语法规则词表，开始生成第一条语法规则，例如："Assignment_N -> L-Expr_N AssignOp_N R-Expr_N"；

S608，预测语法规则和编辑抽象语法树迭代地进行，直到生成一棵完整的抽象语法树，该抽象语法树为一棵不同于步骤 S602 中基于缺陷代码得到的抽象语法树。

在一个实施例中，在步骤 S607 中的第一条语法规则基础上生成第二条语法规则，例如："L-Expr_N -> loc1"，生成过程迭代地进行，直到生成一棵完整的抽象语法树，如图 7(c)所示。

S609，将步骤 S608 生成的抽象语法树进行代码实例化，得到修复补丁。

在一个实施例中，生成的抽象语法树需要进一步还原为程序源码"loc1 = 1;"即为缺陷代码"loc1 = loc2 - 1;"的修复代码（修复补丁）。即这段程序的修复是将 loc1 的值从"loc2-1"改为常量 1。

本公开上述各实施例可应用于程序缺陷修复工具中。

通过本公开上述实施例，可根据输入的缺陷代码，输出能同时满足语法规则和类型约束的修复补丁，扩展了现有语法制导缺陷修复方法的能力，规避生成类型不正确的补丁，提高缺陷修复正确率，从而提高软件开发和维护的效率，降低人力成本。

本公开的实施例还提供了一种计算机可读存储介质，该计算机可读存储介质中存储有计算机程序，其中，该计算机程序被设置为运行时执行上述任一项方法实施例中的步骤。

在一个示例性实施例中，上述计算机可读存储介质可以包括但不限于：U 盘、只读存储器（Read-Only Memory，简称为 ROM）、随机存取存储器（Random Access Memory，简称为 RAM）、移动硬盘、磁碟或者光盘等各种可以存储计算机程序的介质。

本公开的实施例还提供了一种电子装置，包括存储器和处理器，该存储器中存储有计算机程序，该处理器被设置为运行计算机程序以执行上述任一项方法实施例中的步骤。

在一个示例性实施例中，上述电子装置还可以包括传输设备以及输入输出设备，其中，该传输设备和上述处理器连接，该输入输出设备和上述处理器连接。

本实施例中的具体示例可以参考上述实施例及示例性实施方式中所描述的示例，本实施例在此不再赘述。

显然，本领域的技术人员应该明白，上述的本公开的各模块或各步骤可以用通用的计算装置来实现，它们可以集中在单个的计算装置上，或者分布在多个计算装置所组成的网络上，它们可以用计算装置可执行的程序代码来实现，从而，可以将它们存储在存储装置中由计算装置来执行，并且在某些情况下，可以以不同于此处的顺序执行所示出或描述的步骤，或者将它们分别制作成各个集成电路模块，或者将它们中的多个模块或步骤制作成单个集成电路模块来实现。这样，本公开不限制于任何特定的硬件和软件结合。

以上所述仅为本公开的优选实施例而已，并不用于限制本公开，对于本领域的技术人员来说，本公开可以有各种更改和变化。凡在本公开的原则之内，所作的任何修改、等同替换、改进等，均应包含在本公开的保护范围之内。

权 利 要 求 书

1、一种修复补丁生成方法，包括：

将含有缺陷代码的程序代码构造成第一抽象语法树，通过类型流图构造器将所述第一抽象语法树转化为类型流图，其中，所述第一抽象语法树中携带类型信息；

遍历所述类型流图获得节点序列、属性序列以及关系矩阵，将所述节点序列、所述属性序列、所述关系矩阵输入类型流图编码器，得到特征向量；

将所述特征向量以及语法规则词表输入语法制导生成器，得到第二抽象语法树，并对所述第二抽象语法树进行代码实例化，以生成修复补丁，其中，所述第二抽象语法树中携带所述类型信息。

2、根据权利要求 1 所述的方法，其中，将含有缺陷代码的程序代码构造成第一抽象语法树之前，所述方法还包括：

通过抽象类型系统将程序源码语料转换为多棵第三抽象语法树，其中，所述第三抽象语法树中携带所述类型信息；

遍历所有所述第三抽象语法树得到所述语法规则词表，其中，所述语法规则词表中携带所述类型信息。

3、根据权利要求 1 或 2 所述的方法，其中，所述第一抽象语法树中的叶子节点由抽象所述程序代码中的常量、变量、操作符而得到；

所述第三抽象语法树中的叶子节点由抽象所述程序源码语料中的常量、变量、操作符而得到。

4、根据权利要求 3 所述的方法，其中，所述类型信息包括以下类型：通用类型、字符串类型、数值类型、布尔类型和空类型；

所述空类型是所述字符串类型、所述数值类型、所述布尔类型的子类型；

所述通用类型是所述字符串类型、所述数值类型、所述布尔类型的父类型；

所述通用类型的父类型是通用类型，空类型的子类型是空类型。

5、根据权利要求 4 所述的方法，其中，

所述程序代码或所述程序源码语料中的常量的类型包括以下之一：字符串类型、数值类型、布尔类型；

所述程序代码或所述程序源码语料中的变量的类型包括以下之一：字符串类型、数值类型、布尔类型；

所述程序代码或所述程序源码语料中的操作符的类型由其操作数类型决定；

所述程序代码或所述程序源码语料中的 NULL 归类为所述空类型。

6、根据权利要求 5 所述的方法，其中，所述程序代码或所述程序源码语料中的操作符的类型由其操作数类型决定，包括：

在所述操作符的所有操作数类型相同的情况下，所述操作符类型与其操作数的类型相同；

在所述操作符的操作数的存在多种类型的情况下，所述操作符类型为其所有操作数类型的父类型。

7、根据权利要求 4 所述的方法，其中，

所述第一抽象语法树或所述第三抽象语法中的非叶子节点类型由其子节点类型决定；

在所述非叶子节点的所有子节点的类型相同的情况下，所述非叶子节点的类型与其子节点的类型相同；

在所述非叶子节点的子节点存在多种类型的情况下，所述非叶子节点的类型为其所有子节点类型的父类型。

8、根据权利要求 1 所述的方法，其中，通过类型流图构造器将所述第一抽象语法树转化为类型流图，包括：

将所述第一抽象语法树中的各节点转换为所述类型流图中抽象语法树节点；

将所述程序代码中的变量抽象为所述类型流图中的变量节点；

其中，所述抽象语法树节点之间、变量节点之间、对应于同一变量的所述抽象语法树节点和所述变量节点之间，由不同属性的边连接。

9、根据权利要求 8 所述的方法，其中，

所述抽象语法树节点之间的边的属性包括：父-子、子-父、左兄弟、右兄弟；

所述变量节点之间的边的属性包括：相同类型、兼容类型、不兼容类型；

对应于同一变量的连接所述抽象语法树节点和所述变量节点之间的边的属性包括：申明-变量、使用-变量、变量-申明、变量-使用。

10、根据权利要求 8 所述的方法，其中，遍历所述类型流图获得节点序列，包括：

遍历所述类型流图中所述第一抽象语法树的抽象语法树节点名称，将所有所述抽象语法树节点名称组成第一序列片段；

按照所述程序代码上下文中各变量的出现顺序，遍历所述第一抽象语法树中的变量节点名称，将缺陷代码处可访问的变量节点名称组成第二序列片段；

将所述第一序列片段和所述第二序列片段进行拼接，得到所述节点序列。

11、根据权利要求 10 所述的方法，其中，遍历所述类型流图获得属性序列，包括：

按照所述节点序列中抽象语法树节点和变量节点的排列顺序，依次获取各所述抽象语法树节点和各所述变量节点对应的节点属性值，并将所有所述节点属性值组合为所述属性序列。

12、根据权利要求 11 所述的方法，其中，所述抽象语法树节点包括以下属性：类型属性、可复制性；所述变量节点包括以下属性：类型属性、可访问性。

13、根据权利要求 1 所述的方法，其中，将所述节点序列、所述属性序列、所述关系矩阵输入类型流图编码器，得到一个特征向量，包括：

通过所述类型流图编码器中的嵌入层，并基于输入的所述节点序列、所述属性序列、所述关系矩阵，得到节点嵌入向量、属性嵌入向量和关系嵌入向量；

通过所述类型流图编码器中的类型敏感的自注意神经元，对所述节点嵌入向量和所述关系嵌入向量进行融合，得到第一融合向量；

通过所述类型流图编码器中的门控层，对所述属性嵌入向量和所述第一融合向量进行融合，得到第二融合向量；

通过所述类型流图编码器中的线性层，对所述第二融合向量进行线性变换得到所述特征

向量。

14、根据权利要求 1 所述的方法，其中，所述特征向量的值为浮点数。

15、根据权利要求 1 所述的方法，其中，将所述特征向量以及语法规则词表输入语法制导生成器，得到第二抽象语法树，包括：

将所述特征向量和所述语法规则词表输入语法制导生成器，预测第一语法规则，并根据所述第一语法规则进行抽象语法树的编辑操作，得到第一部分抽象语法树，其中，所述第一语法规则携带类型信息；

基于所述第一部分抽象语法树、所述特征向量以及所述语法规则词表，预测第二语法规则，并根据所述第二语法规则进行抽象语法树的编辑操作，得到第二部分抽象语法树，其中，所述第二语法规则携带类型信息；

迭代进行预测语法规则步骤和编辑抽象语法树步骤，直到得到完整的所述第二抽象语法树。

16、一种计算机可读存储介质，其中，所述计算机可读存储介质中存储有计算机程序，其中，所述计算机程序被处理器执行时实现所述权利要求 1 至 15 任一项中所述的方法的步骤。

17、一种电子装置，包括存储器、处理器以及存储在所述存储器上并可在所述处理器上运行的计算机程序，其中，所述处理器执行所述计算机程序时实现所述权利要求 1 至 15 任一项中所述的方法的步骤。

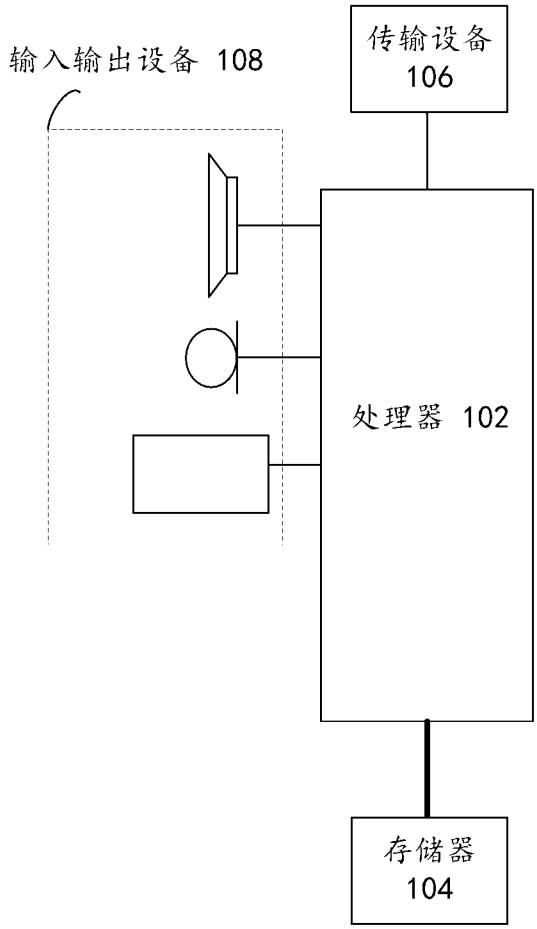


图 1

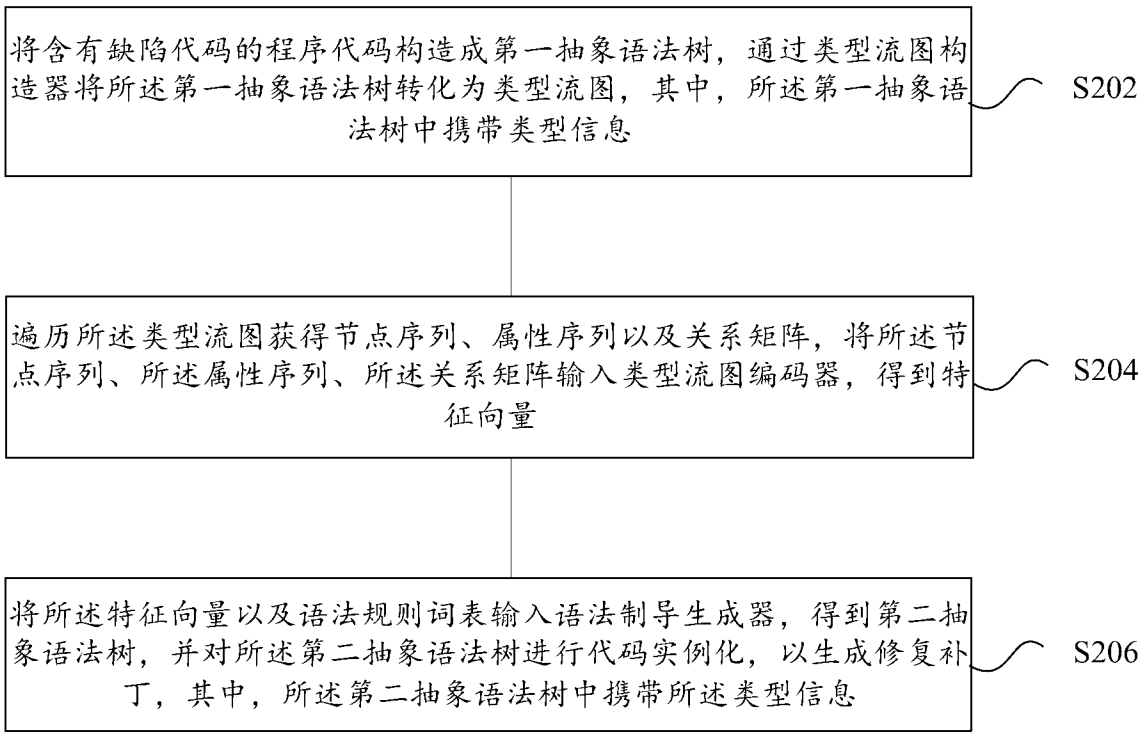

将含有缺陷代码的程序代码构造成第一抽象语法树，通过类型流图构造器将所述第一抽象语法树转化为类型流图，其中，所述第一抽象语法树中携带类型信息
S202
遍历所述类型流图获得节点序列、属性序列以及关系矩阵，将所述节点序列、所述属性序列、所述关系矩阵输入类型流图编码器，得到特征向量
S204
将所述特征向量以及语法规则词表输入语法制导生成器，得到第二抽象语法树，并对所述第二抽象语法树进行代码实例化，以生成修复补丁，其中，所述第二抽象语法树中携带所述类型信息
S206

图 2

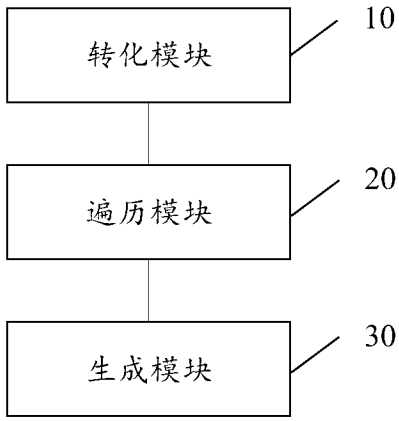

转化模块
10
遍历模块
20
生成模块
30

图 3

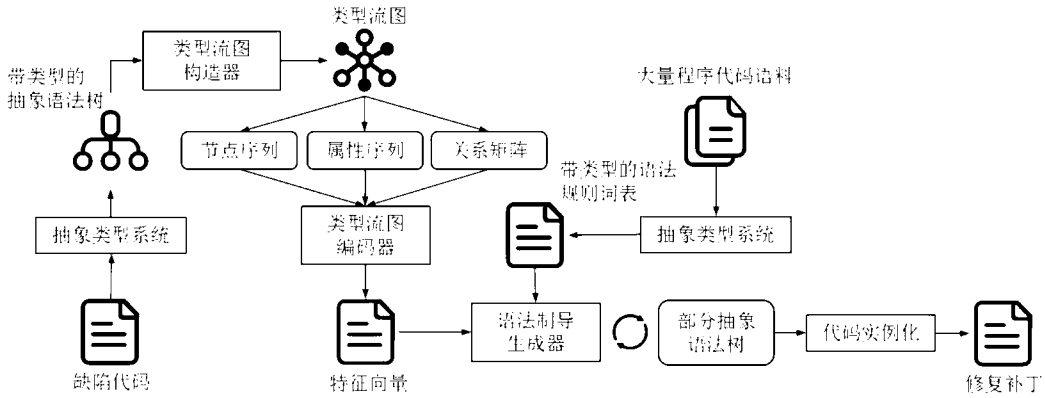

类型流图
类型流图
构造器
带类型的
抽象语法树
节点序列
属性序列
关系矩阵
大量程序代码语料
带类型的语法
规则词表
抽象类型系统
类型流图
编码器
抽象类型系统
缺陷代码
特征向量
语法制导
生成器
部分抽象
语法树
代码实例化
修复补丁

图 4

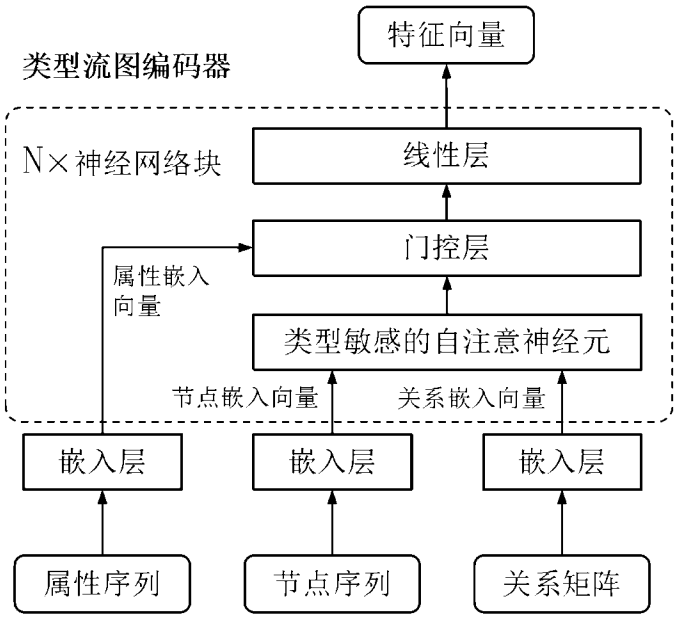

特征向量
类型流图编码器
N×神经网络块
线性层
门控层
属性嵌入
向量
类型敏感的自注意神经元
节点嵌入向量
关系嵌入向量
嵌入层
嵌入层
嵌入层
属性序列
节点序列
关系矩阵

图 5

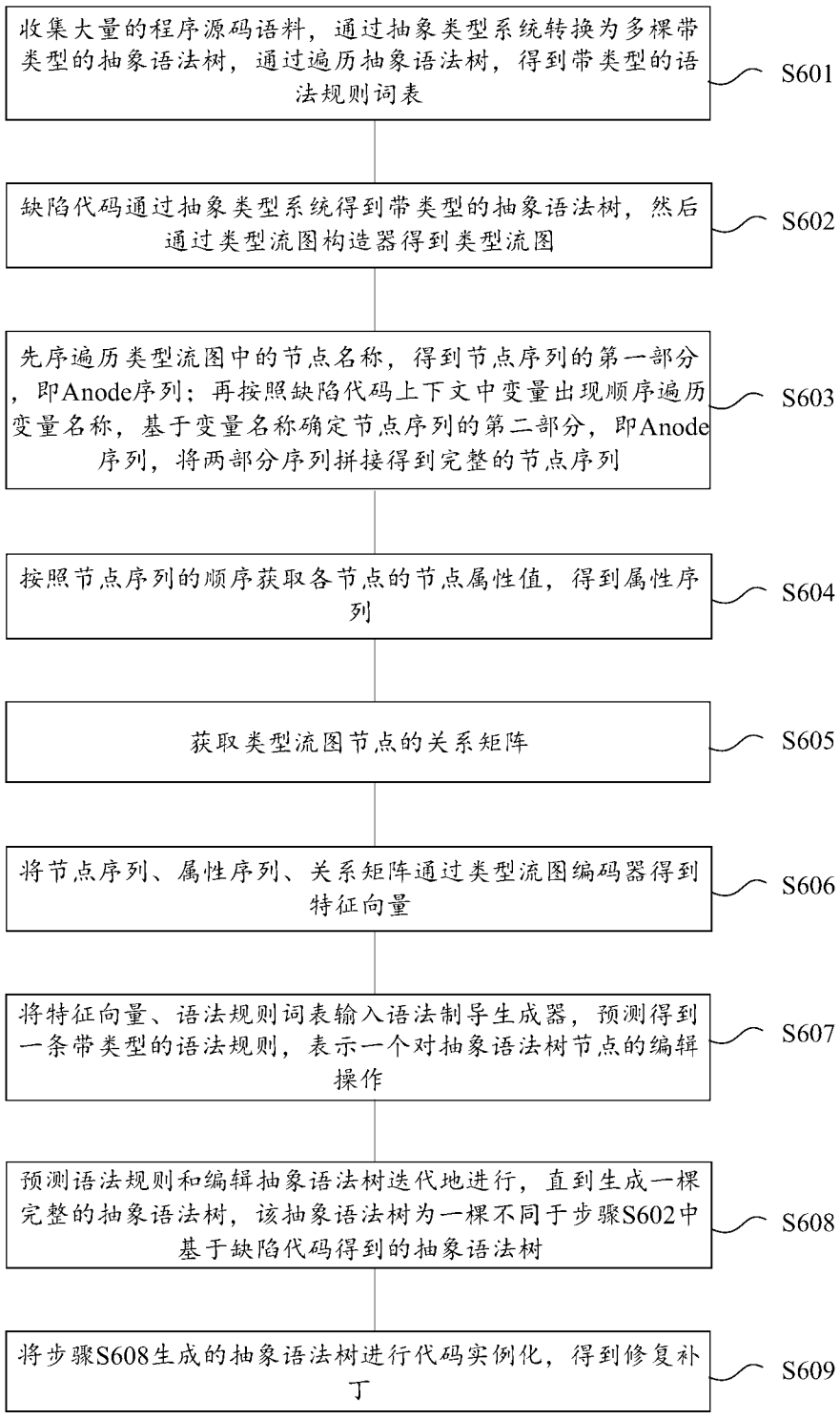

收集大量的程序源码语料，通过抽象类型系统转换为多棵带类型的抽象语法树，通过遍历抽象语法树，得到带类型的语法规则词表
S601
缺陷代码通过抽象类型系统得到带类型的抽象语法树，然后通过类型流图构造器得到类型流图
S602
先序遍历类型流图中的节点名称，得到节点序列的第一部分，即Anode序列；再按照缺陷代码上下文中变量出现顺序遍历变量名称，基于变量名称确定节点序列的第二部分，即Anode序列，将两部分序列拼接得到完整的节点序列
S603
按照节点序列的顺序获取各节点的节点属性值，得到属性序列
S604
获取类型流图节点的关系矩阵
S605
将节点序列、属性序列、关系矩阵通过类型流图编码器得到特征向量
S606
将特征向量、语法规则词表输入语法制导生成器，预测得到一条带类型的语法规则，表示一个对抽象语法树节点的编辑操作
S607
预测语法规则和编辑抽象语法树迭代地进行，直到生成一棵完整的抽象语法树，该抽象语法树为一棵不同于步骤S602中基于缺陷代码得到的抽象语法树
S608
将步骤S608生成的抽象语法树进行代码实例化，得到修复补丁
S609

图 6

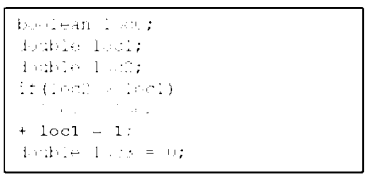


图7（a）示例代码，“-”表示缺陷代码，
“+”表示修复代码

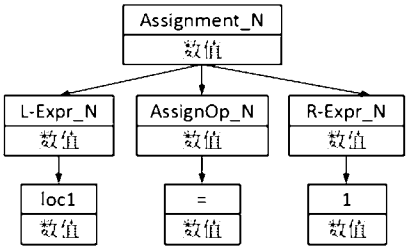


图7（c）修复代码的带类型抽象语法树

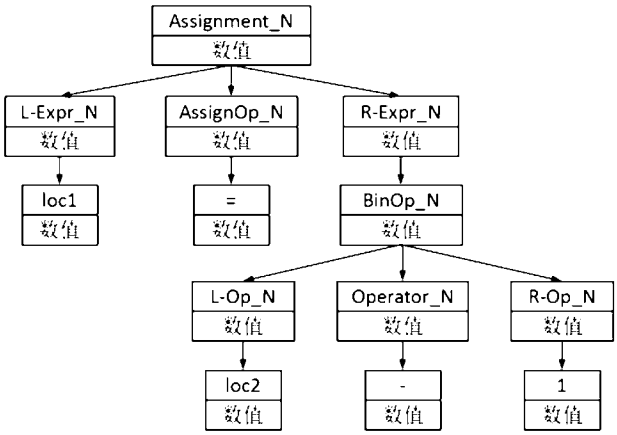


图7（b）缺陷代码的带类型抽象语法树

图 7

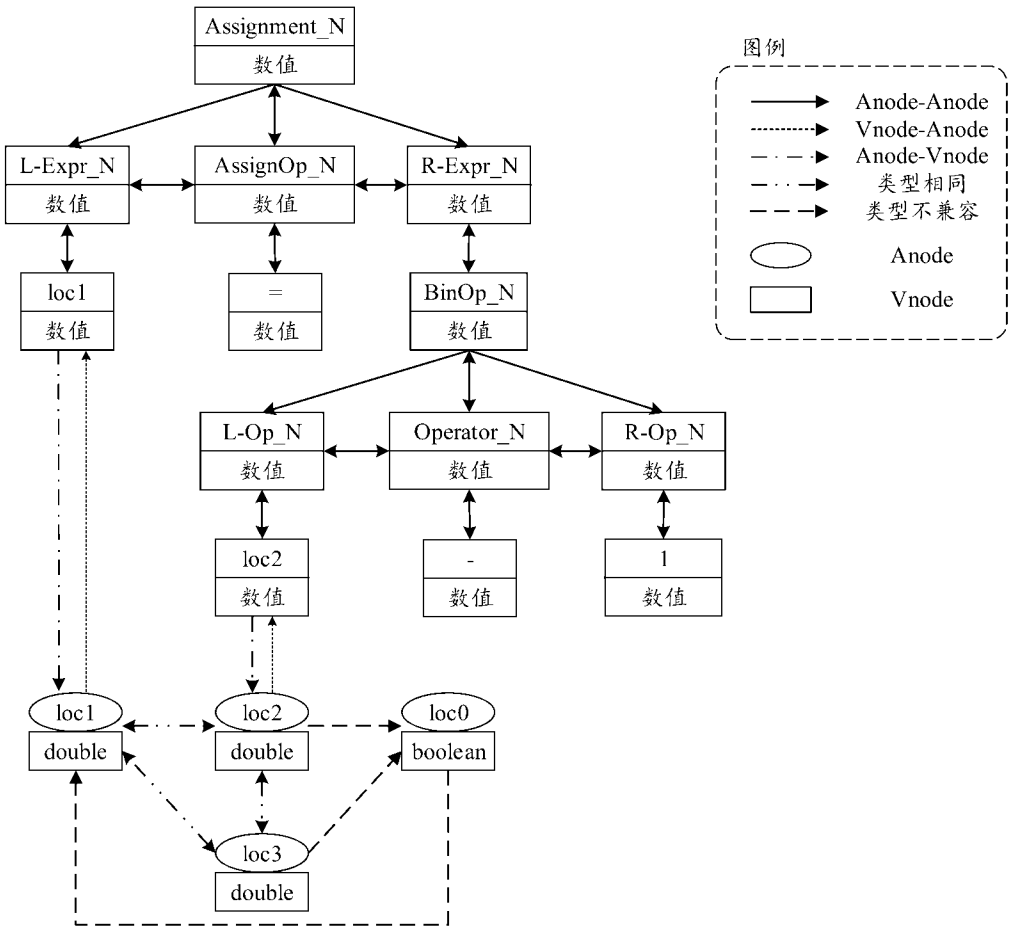


图 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2024/090980

A. CLASSIFICATION OF SUBJECT MATTER

G06F 8/41(2018.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC:G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNTXT, CNABS, DWPI, VEN, CNKI, USTXT, EPTXT, WOTXT, 百度, BAIDU: 抽象语法树, 类型, 代码, 缺陷, 补丁, 深度学习, 神经网络, abstract syntax tree, type, code, bug, patch, deep learning, neural network

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	CN 115934147 A (ZTE CORP. et al.) 07 April 2023 (2023-04-07) description, paragraphs [0019]-[0084]	1, 3, 14, 16-17
Y	曹英魁 等 (CAO, Yingkui et al.). "软件学报 (Journal of Software)" *一种结构信息增强的代码修改自动转换方法 (Structurally-Enhanced Approach for Automatic Code Change Transformation)*, Vol. 32, No. 4, 22 January 2021 (2021-01-22), ISSN: 1000-9825, pages 1006-1022	1, 3, 14, 16-17
Y	CN 113064586 A (NANJING UNIVERSITY) 02 July 2021 (2021-07-02) description, paragraphs [0028]-[0064]	1, 3, 14, 16-17
A	CN 114489669 A (NANJING UNIVERSITY) 13 May 2022 (2022-05-13) entire document	1-17
A	US 2015135166 A1 (MICROSOFT CORP.) 14 May 2015 (2015-05-14) entire document	1-17

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
21 June 2024	**20 July 2024**
Name and mailing address of the ISA/CN **China National Intellectual Property Administration (ISA/CN)** **China No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing 100088**	Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2024/090980

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 115934147 A	07 April 2023	None	
CN 113064586 A	02 July 2021	CN 113064586 B	22 April 2022
CN 114489669 A	13 May 2022	None	
US 2015135166 A1	14 May 2015	US 9928040 B2	27 March 2018

国际检索报告

国际申请号
PCT/CN2024/090980

A. 主题的分类

G06F 8/41(2018.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

IPC:G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用））

CNTXT,CNABS,DWPI,VEN,CNKI,USTXT,EPTXT,WOTXT,百度:抽象语法树,类型,代码,缺陷,补丁,深度学习,神经网络,abstract syntax tree,type,code,bug,patch,deep learning,neural network

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
Y	CN 115934147 A (中兴通讯股份有限 等) 2023年4月7日 (2023 - 04 - 07) 说明书第[0019]-[0084]段	1,3,14,16-17
Y	曹英魁 等. "软件学报" 一种结构信息增强的代码修改自动转换方法, 第32卷，第4期，2021年1月22日 (2021 - 01 - 22)， ISSN：1000-9825, 第1006-1022页	1,3,14,16-17
Y	CN 113064586 A (南京大学) 2021年7月2日 (2021 - 07 - 02) 说明书第[0028]-[0064]段	1,3,14,16-17
A	CN 114489669 A (南京大学) 2022年5月13日 (2022 - 05 - 13) 全文	1-17
A	US 2015135166 A1 (MICROSOFT CORP) 2015年5月14日 (2015 - 05 - 14) 全文	1-17

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

“A” 认为不特别相关的表示了现有技术一般状态的文件

“D” 申请人在国际申请中引证的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2024年6月21日	2024年7月20日
ISA/CN的名称和邮寄地址 中国国家知识产权局 中国北京市海淀区蓟门桥西土城路6号 100088	受权官员 邓丽婉 电话号码 (+86) 020-28958946

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2024/090980

检索报告引用的专利文件	公布日 (年/月/日)	同族专利	公布日 (年/月/日)
CN 115934147 A	2023年4月7日	无	
CN 113064586 A	2021年7月2日	CN 113064586 B	2022年4月22日
CN 114489669 A	2022年5月13日	无	
US 2015135166 A1	2015年5月14日	US 9928040 B2	2018年3月27日