软件安全入门

第四课 程序切片

目录
CONTENTS

主要对程序切片的基础知识和基本原理进行介绍。

程序切片是旨在从程序中提取满足一定约束条件的代码片段，是一种重要的程序分解技术。控制流信息和数据流信息是实现程序切片所依赖的最重要的信息。

控制流分析

控制流指程序中一系列指令(语句、函数调用)执行的顺序，例如程序中的3种基本控制结构：顺序、条件、循环。

数据流分析

数据流是数据在程序中一系列执行指令间产生、传递、复制和消失的过程。

基本块是满足下列条件的一组连续指令代码：

(1)程序执行时只能从该基本块的第一条指令进入该基本块

(2)程序执行时离开该基本块前的最后一条指令必须是该基本块的最后一条指令。

当程序被划分为基本块后，基本块之间在程序执行流程上互为前驱和后继关系视为一条边，则整个程序能够转换为一个有向图，即控制流图(Control FlowGraph,CFG)。

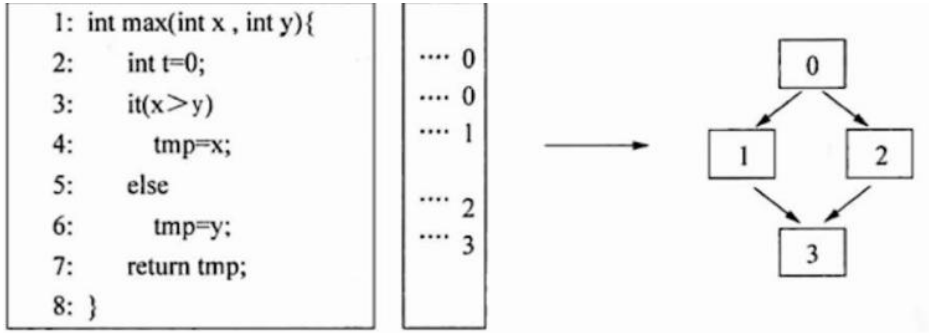

图 4-4 代码片段 2 及其控制流图

数据流分析关注的是跨越多条语句的变量定义、赋值和运算操作。一般以一条语句作为基准点进行分析，一条语句通常会有引用变量和定义变量的双重行为。因此，数据流分析又可分为可到达定义分析和变量活性分析。

可到达定义

可到达定义分析对象为一条程序语句，主要目标在于获取该语句所引用变量的来源，即引用的变量是如何产生、复制和传递的。

活性分析

活性分析是指对某个语句定义的变量是否在后续语句中被引用以及被哪些语句引用的情况的分析。

1.1.3 程序依赖图

控制流分析和数据流分析分别建立了基本块之间的程序执行顺序关系以及程序中不同语句对同一变量的定义-引用关系，也可以称为控制依赖关系和数据依赖关系。

由“控制依赖关系”可以定义控制依赖图G=(V,C),由“数据依赖关系”可以定义数据依赖图G=(V,D),依据两种依赖关系可以构建出程序依赖图(PDG)G=(V.E),其中E=CUDUX。

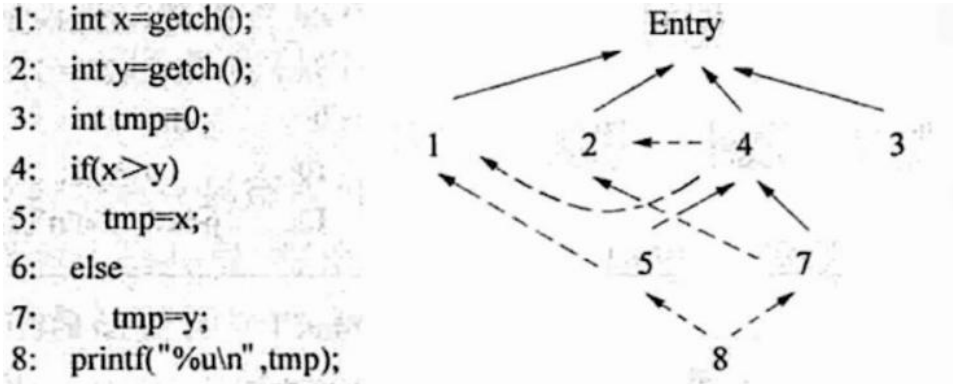

图 4-8　代码片段 3 及其程序依赖图

程序切片通常包括三个步骤：

1.程序依赖关系提取：主要从程序中提取各类信息，包括控制流和数据流信息，形成程序依赖图。

2.切片规则制定：主要是依据具体的程序分析需求设计切片准则(包含两个要素，切片目标变量以及开始切片的代码位置)。

3.切片生成：主要是依据切片准则选择相应的程序切片方法，然后对第一步中提取的依赖关系进行分析处理，从而生成切片。

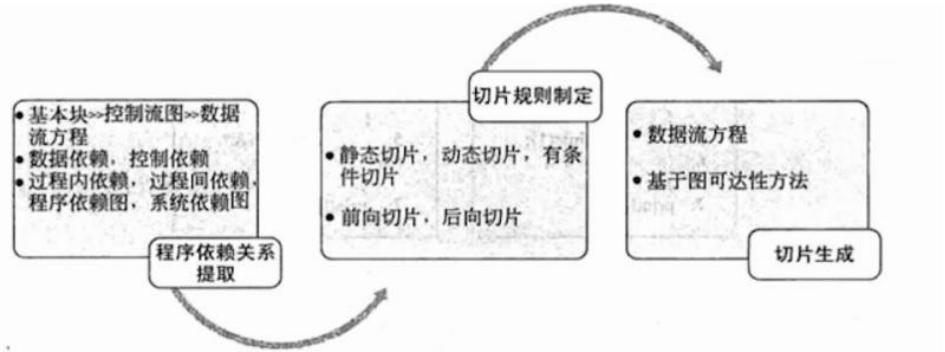

图 4-11 切片的一般过程

主要内容包括基于数据流方程的切片方法和基于程序依赖图可达性的切片方法。

2 静态程序切片的两种方法

方法一

基于数据流方程的切片方法

通过迭代计算控制流图中每个节点的相关变量集合，迭代分析语句间的数据依赖关系和控制依赖关系，最终获得每条语句中与切片准则C相关的变量的集合。

基于数据流方程的方法提取切片的策略较为保守，构造的程序切片往往较大，可能造成求解空间爆炸分析人员还是无法基于该切片进行有效分析。

方法二

基于图可达性算法的切片方法

首先需要构建程序依赖图，然后从切片准则所对应的节点出发沿着数据依赖边和控制依赖边进行图遍历，所有遍历可达的节点都加入到切片中。

优点：能够计算含有过程的程序切片，并且使程序切片的计算过程变得简单且可视化。缺点：不能计算含有多个过程的程序切片。

主要内容包括基于程序依赖图的动态切片方法和基于动态依赖图的动态切片方法。

基于程序依赖图的动态切片方法步骤如下：

①构造程序依赖图

②依据切片准则的输入得到程序动态执行历史

③在程序依赖图中删除在执行历史中不存在的节点

④对于当前依赖图中剩余节点之间的每条边，如果这条边对应的控制和数据依赖关系并没有在执行历史中出现，则将这条边删除。

⑤从切片准则对应的节点开始进行图的遍历，将所有可达的节点加入到切片中。

3.2 基于动态依赖图的动态切片方法

DDG的构造方法是遍历执行历史，依次为其中每个语句的每一次出现都创建一个新的节点，同时节点之间只因为程序执行而导致有实质的控制和依赖关系时才建立一条依赖边。

基于DDG的动态切片计算比较简单，只需要从切片准则对应的节点开始遍历动态依赖图，将多有可达到的节点都添加到切片中。

动态依赖图(DDG),不但能够表示静态程序依赖关系，同时也能表示程序的动态执行过程。

课程小结

·1.程序切片介绍

·主要内容包括：程序切片的基础知识和基本原理。

· 2.静态程序切片

·主要内容包括：基于数据流方程的切片方法和基于程序依赖图可达性的切片方法。

· 3.动态程序切片

·主要内容包括：基于程序依赖图的动态切片方法和基于动态依赖图的动态切片方法。

谢谢观赏