

白盒测试技术

知识点4：数据流测试



内容

- 数据流测试的内容和方法

数据流测试的目标

- ◉ 数据流测试主要是为了发现
 - 定义/引用异常缺陷
 - 变量被定义，但从来没有使用（引用）
 - 所使用的变量没有被定义
 - 变量在使用之前被定义两次。
- ◉ 数据流测试重点关注的是变量的定义与使用测试
 - 调试修改bug时，我们可能会在一段代码中搜索某个变量所有的定义、使用位置，思考在程序运行时该变量的值会如何变化，从而分析bug产生原因
 - 数据流测试是将这种方法形式化，这样也便于构造算法，实现自动化分析

数据流测试概念

- 数据流测试：指关注变量接收值的点和使用（引用）这些值的点的结构性测试形式。数据流测试用作路径测试的“真实性检查”。
- 两种方法：
 - 一种提供一组基本定义和一种统一的测试覆盖指标结构
 - 另一种基于叫做“程序片”的概念。

定义/使用测试

- 程序图： $G(P)$ ，其中 P 是程序
- V ：程序 P 中的一组变量
- $G(P)$ ：有一个单入口节点和一个单出口节点，并且不允许有从某个节点到其自身的边
- $PATH(P)$ ： P 中所有路径的集合

一组定义

- 节点 $n \in G(P)$ 是变量 $v \in V$ 的定义节点，记做 $DEF(v, n)$ ，当且仅当变量 v 的值由对应节点 n 的语句片段处定义。
- 输入语句、赋值语句、循环控制语句和过程调用，都是定义节点语句的例子。
- 执行定义语句，该变量所关联的存储单元的内容就会改变。

一组定义

- 节点 $n \in G(P)$ 是变量 $v \in V$ 的使用节点，记做 $USE(v, n)$ ，当且仅当变量 v 的值在对应节点 n 的语句片段处使用。
- 输出语句、赋值语句、条件语句、循环控制语句和过程调用，都是使用节点语句的例子。
- 执行定义语句，该变量所关联的存储单元的内容保持不变。

一组定义

- 使用节点 $USE(v, n)$ 是一个谓词使用（记做 P-use），当且仅当语句 n 是谓词语句；否则 $USE(v, n)$ 是计算使用（记做 C-use）。
- 对应谓词使用的节点永远有外度 ≥ 2 ，对应于计算使用的节点永远有外度 ≤ 1 。

一组定义

- 关于变量 v 的定义-使用路径（记做du-path）
 - 是PATHS (P) 中的路径，使得对某个 $v \in V$ ，存在定义和使用节点DEF (v , m) 和USE (v , n) ，使得m和n是该路径的最初和最终节点。

一组定义

- 关于变量 v 的定义清除路径（记做dc-path），
 - 是具有最初和最终节点 $DEF(v, m)$ 和 $USE(v, n)$ 的PATHS(P)中的路径，使得该路径中没有其他节点是 v 的定义节点。

定义-使用路径测试覆盖指标

- 集合T满足程序P的全定义准则，
 - 当且仅当所有变量 $v \in V$ ，T包含从v的每个定义节点到v的一个使用的定义清除路径。
- 集合T满足程序P的全使用准则，
 - 当且仅当所有变量 $v \in V$ ，T包含从v的每个定义节点到v的所有使用，以及到所有 $USE(v, n)$ 后续节点的定义清除路径。

定义-使用路径测试覆盖指标

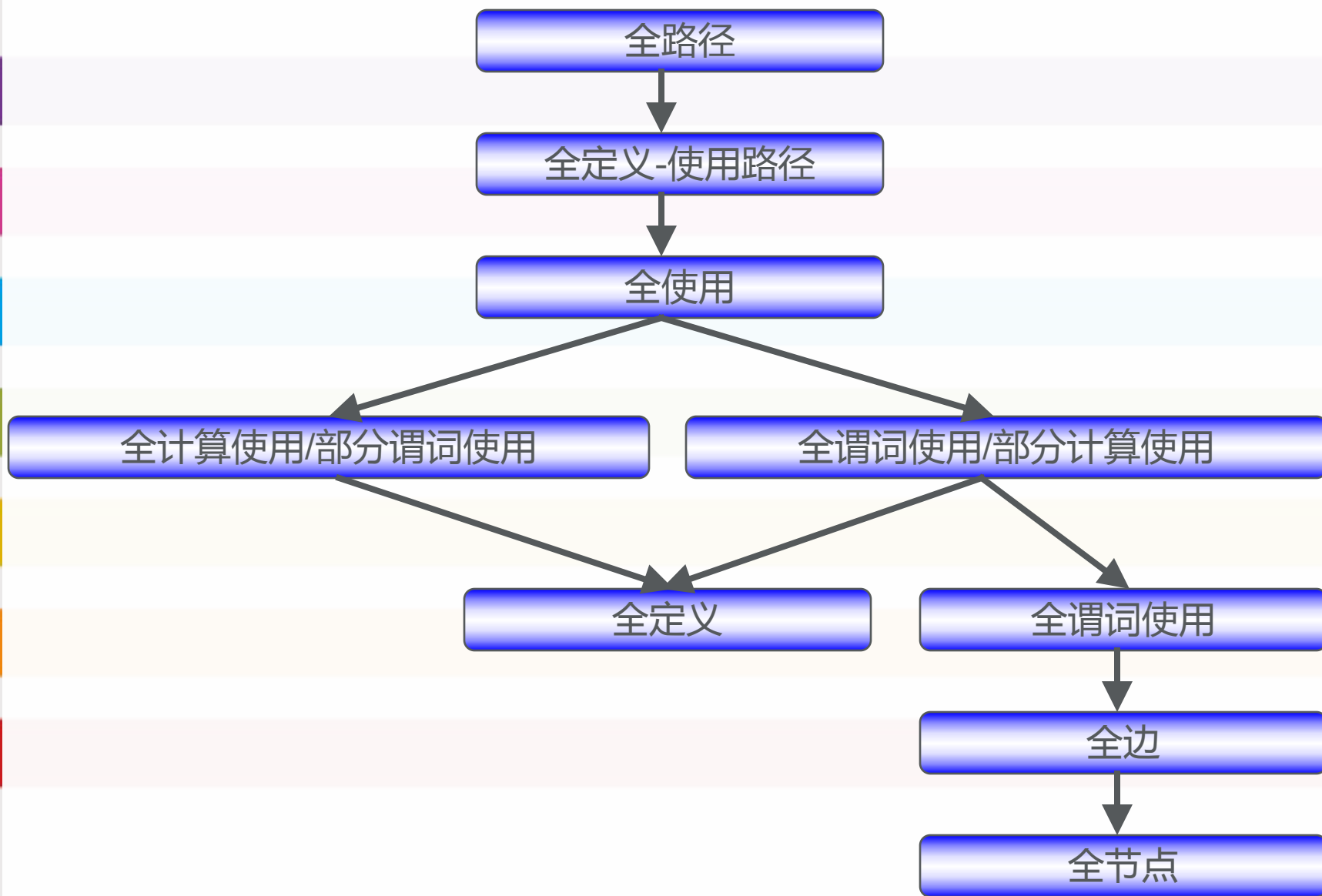
- 集合 T 满足程序 P 的全谓词使用/部分计算使用准则，
 - 当且仅当所有变量 $v \in V$ ， T 包含从 v 的每个定义节点到 v 的所有谓词使用的定义清除路径，并且如果 v 的一个定义没有谓词使用，则定义清除路径导致至少一个计算使用。

定义-使用路径测试覆盖指标

- 集合 T 满足程序 P 的全计算使用/部分谓词使用准则，
 - 当且仅当所有变量 $v \in V$ ， T 包含从 v 的每个定义节点到 v 的所有计算使用的定义清除路径，并且如果 v 的一个定义没有计算使用，则定义清除路径导致至少一个谓词使用。

定义-使用路径测试覆盖指标

- 集合T满足程序P的全定义-使用路径准则，
 - 当且仅当所有变量 $v \in V$ ，T包含从v的每个定义节点到v的所有使用，以及到所有USE(v, n)后续节点的定义清除路径，并且这些路径要么有一次的环经过，要么没有环路。



谢 谢