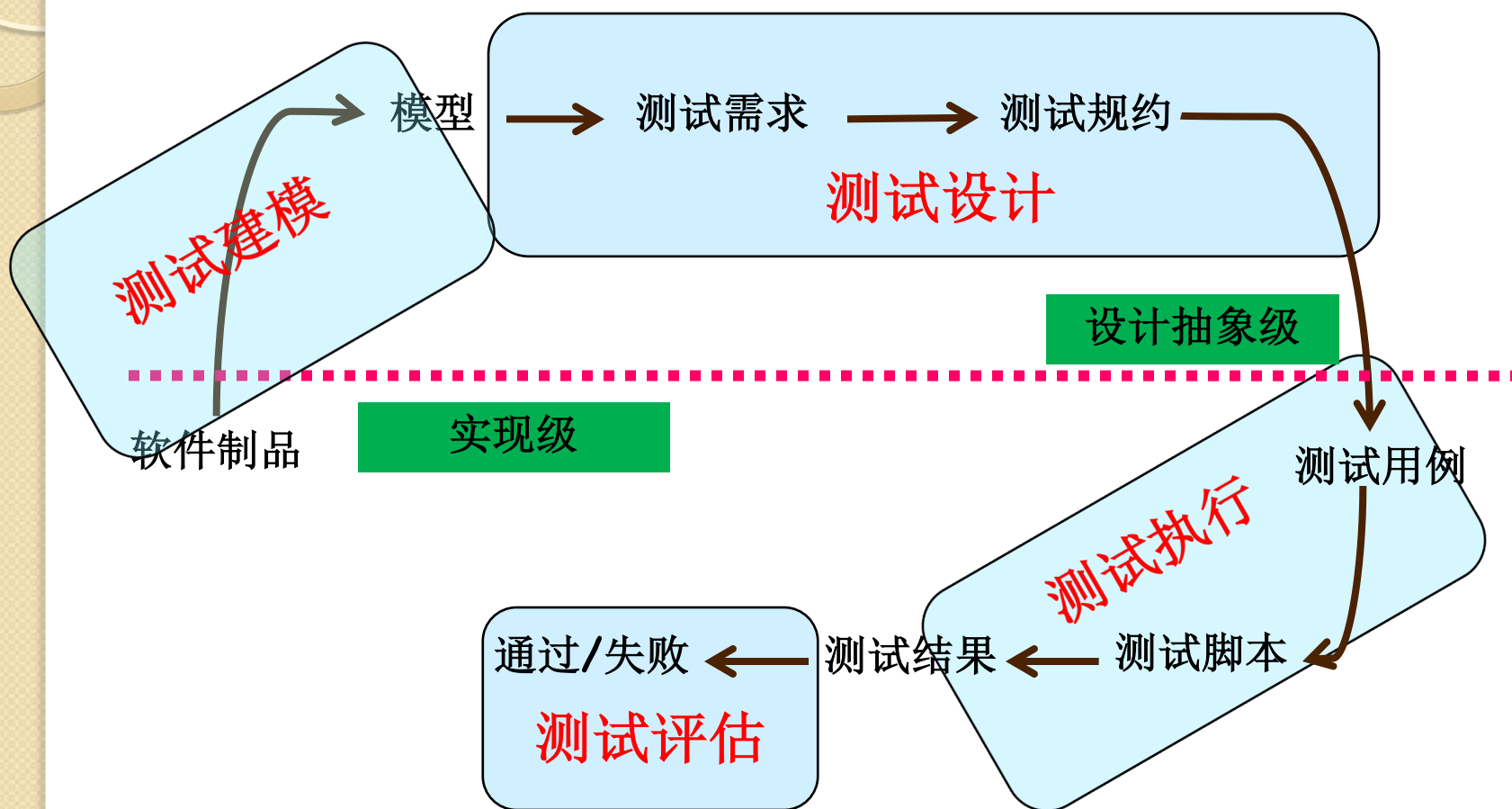
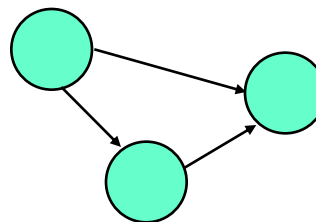


基于模型的测试



4种模型

图



逻辑表达式

$(\text{not } X \text{ or not } Y) \text{ and } A \text{ and } B$

输入域特性

A: $\{0, 1, >1\}$

B: $\{600, 700, 800\}$

C: $\{\text{swe, cs, isa, infs}\}$

语法结构

if $(x > y)$

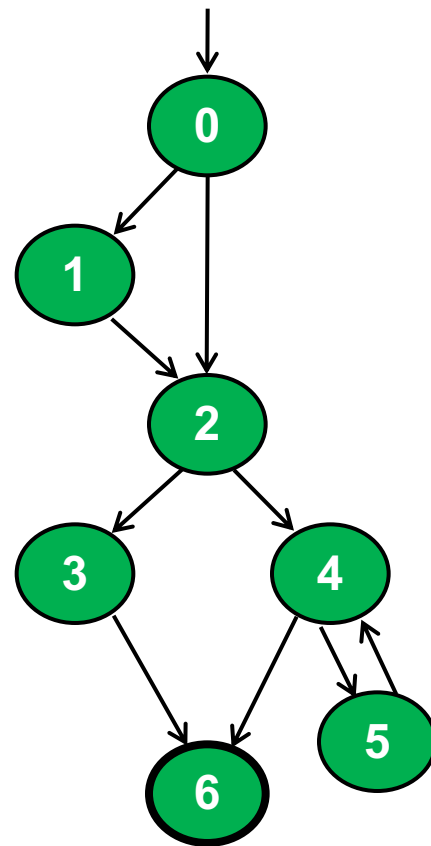
$z = x - y;$

else

$z = 2 * x;$

图覆盖

图: $G = (N, N_0, N_f, E)$



- 路径: $\langle n_1, n_2, \dots, n_m \rangle, \langle n_i, n_{i+1} \rangle \in E$
- 路径长度: 路径中的边数
- 子路径: 路径中的节点子序列
- 测试路径: 从开始节点到终止节点的路径

图覆盖

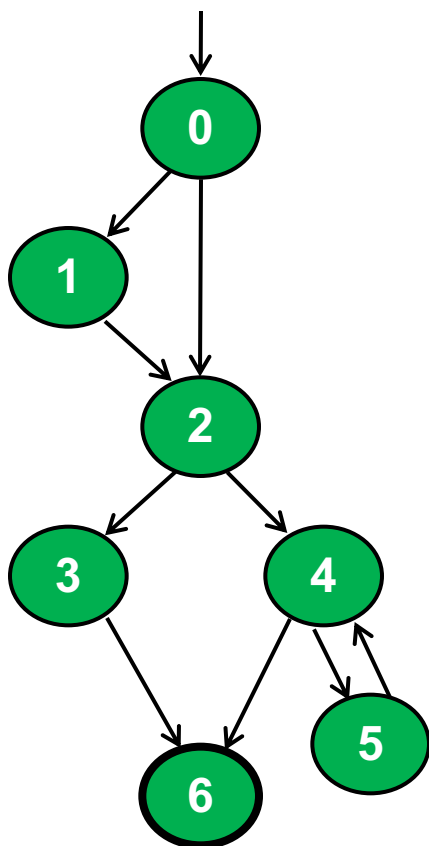
■ 图结构覆盖

覆盖图的节点、边或子路径

■ 数据流覆盖

用变量的引用标注图，覆盖变量的定义、使用或它们之间的关系

图覆盖—结构覆盖



节点覆盖

测试需求: {0, 1, 2, 3, 4, 5, 6}

测试路径集

{[0, 1, 2, 3, 6], [0, 1, 2, 4, 5, 4, 6]}

边覆盖

测试需求

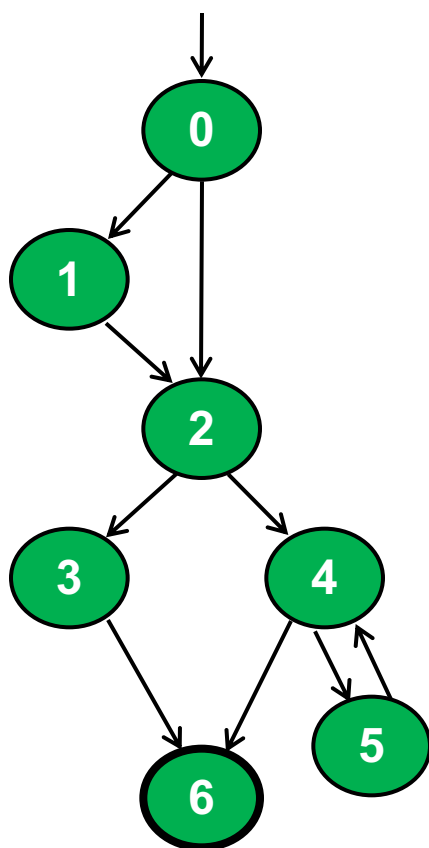
{[0, 1], [0, 2], [1, 2], [2, 3], [2, 4], [3, 6], [4, 5], [4, 6], [5, 4]}

测试路径集

{[0, 1, 2, 3, 6], [0, 2, 4, 5, 4, 6]}

图覆盖—结构覆盖

边对覆盖：测试覆盖每一个可达的长度不超过2的路径



边对覆盖

测试需求（边对集）

{ [0,1,2], [0,2,3], [0,2,4],
[1,2,3], [1,2,4], [2,3,6], [2,4,5],
[2,4,6], [4,5,4], [5,4,5], [5,4,6] }

测试路径集

{ [0, 1, 2, 3, 6],
[0, 1, 2, 4, 6],
[0, 2, 3, 6],
[0, 2, 4, 5, 4, 5, 4, 6] }

图覆盖—结构覆盖

全路径覆盖：测试覆盖所有可能的路径

不幸的是，如果图包含循环，全路径覆盖可能不切实际

指定路径覆盖：测试覆盖用户指定的路径集

图覆盖—结构覆盖

■ 测试循环

- 1970s : 执行1次循环
- 1980s : 执行每个循环1次
- 1990s : 执行循环0次、1次和多次
- 2000s : 覆盖主路径

图覆盖—结构覆盖

■ 简单路径

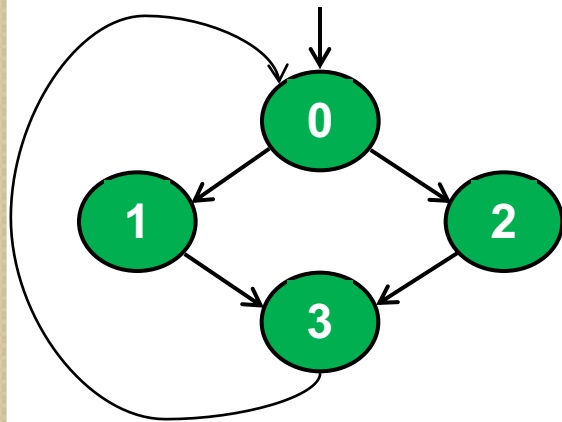
除开始和结束节点可能相同外没有重复节点

- 无内部循环
- 简单路径的所有的子路径也是简单路径
- 一个循环是简单路径

■ 主路径

不是其它简单路径的真子路径

图覆盖—结构覆盖



简单路径集

$\{ [0, 1, 3, 0], [0, 2, 3, 0], [1, 3, 0, 1], [2, 3, 0, 2], [3, 0, 1, 3], [3, 0, 2, 3], [1, 3, 0, 2], [2, 3, 0, 1],$

$[0, 1, 3], [0, 2, 3], [1, 3, 0], [2, 3, 0], [3, 0, 1], [3, 0, 2],$

$[0, 1], [0, 2], [1, 3], [2, 3], [3, 0],$

$[0], [1], [2], [3] \}$

主路径集

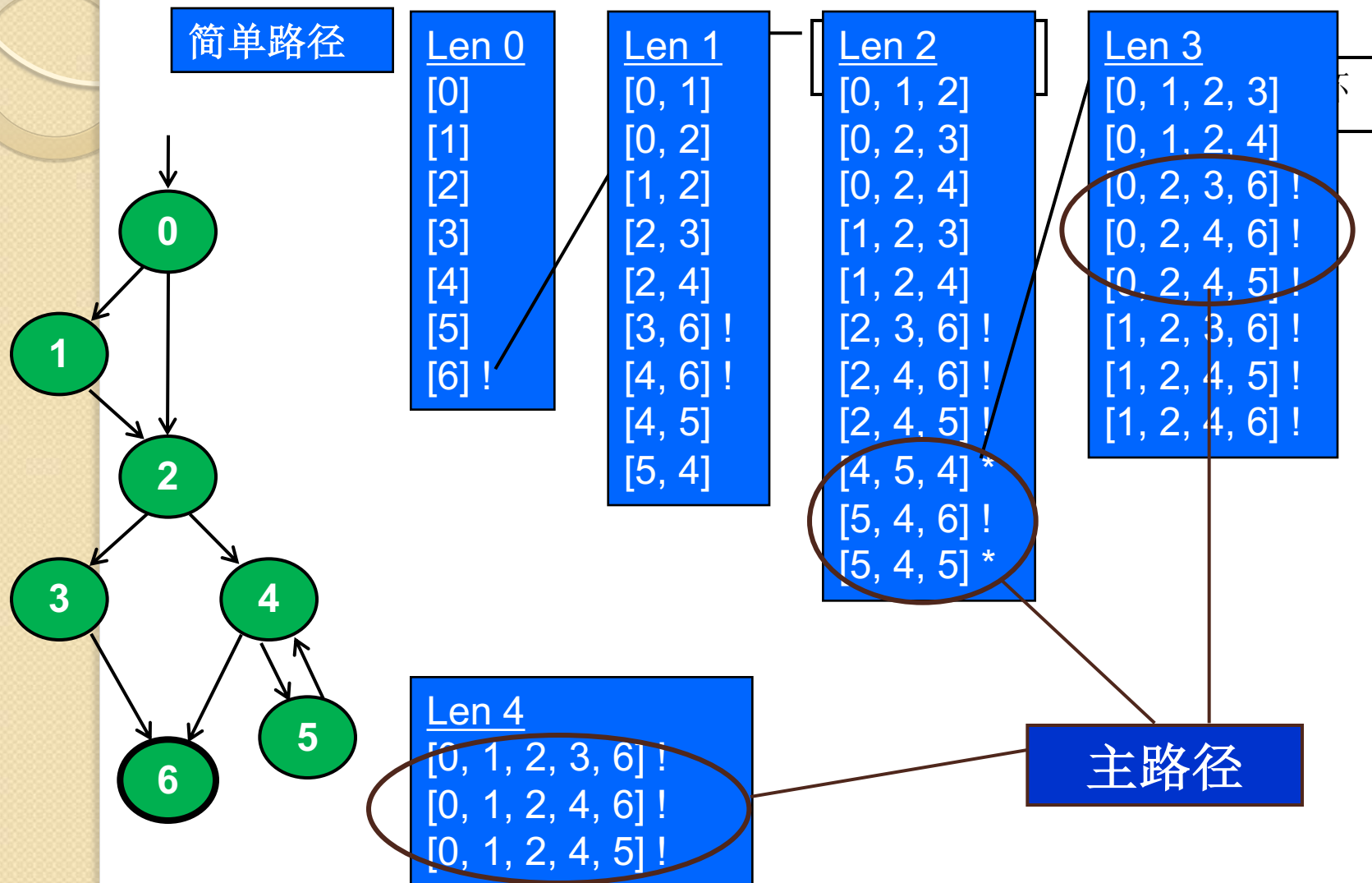
$\{ [0, 1, 3, 0], [0, 2, 3, 0], [1, 3, 0, 1], [2, 3, 0, 2], [3, 0, 1, 3], [3, 0, 2, 3], [1, 3, 0, 2], [2, 3, 0, 1] \}$

图覆盖—结构覆盖

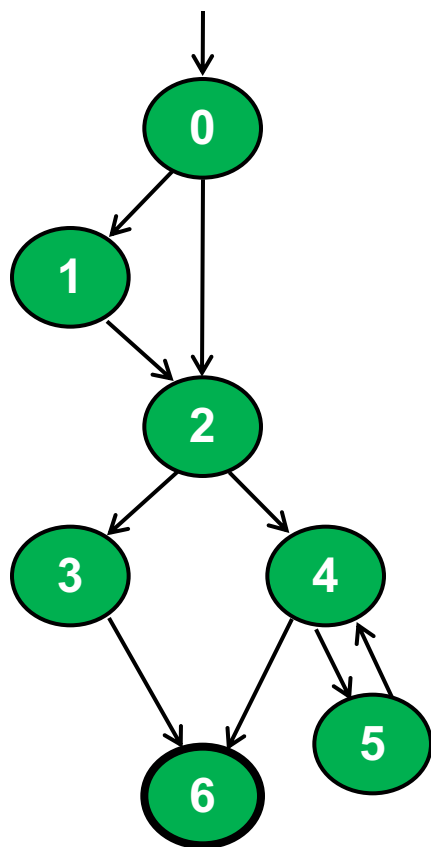
主路径覆盖：测试覆盖每个主路径

- 遍历长度为 $0, 1, \dots, n$ 的所有路径
(n 为指定的主路径最大长度)
- 主路径覆盖包含节点、边和边对覆盖

图覆盖—结构覆盖



图覆盖—结构覆盖



- 38条简单路径

- 9条主路径

测试需求（主路径集）

[0, 1, 2, 3, 6]

[0, 1, 2, 4, 5]

[0, 1, 2, 4, 6]

[0, 2, 3, 6]

[0, 2, 4, 5]

[0, 2, 4, 6]

[5, 4, 6]

[4, 5, 4]

[5, 4, 5]

0次循环

1次循环

多次循环

测试路径集

{[0,1,2,3,6], [0,1,2,4,6],

[0,2,4,6], [0,2,3,6],

[0,2,4,5,4,6], [0,1,2,4,5,4,5,4,6]}

图覆盖—结构覆盖

问题

■ 测试用例的实例化（可执行的测试用例）

语法可达：图中存在一条满足需求的子路径

语义可达：测试能真正执行到一条子路径

■ 不可行的测试需求

不可达的节点

不能严格按子路径执行

■ 非确定性

■ 模型组合

图覆盖—结构覆盖

测试用例的实例化

■ 游历 (Touring)

测试路径 p 游历路径 q ，如果 q 是 p 的子路径

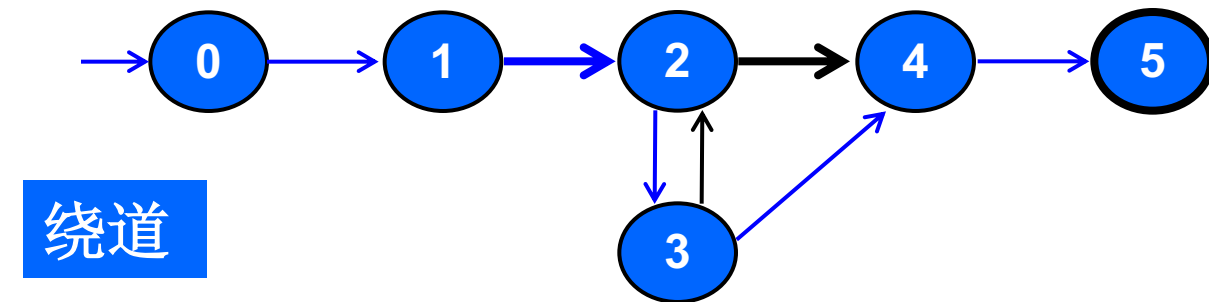
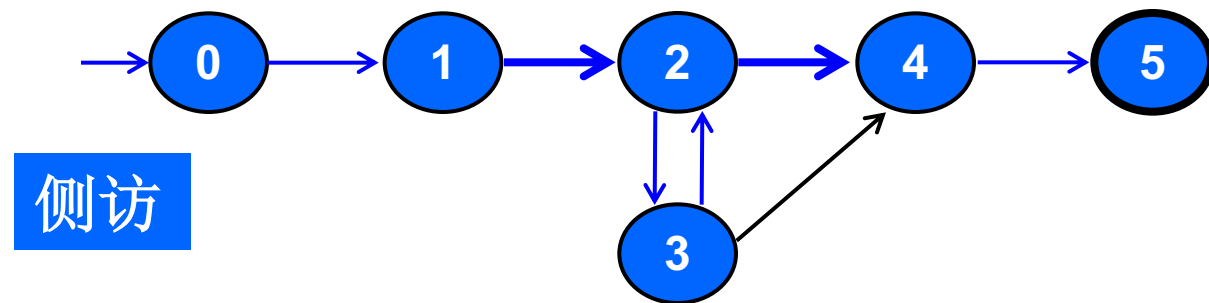
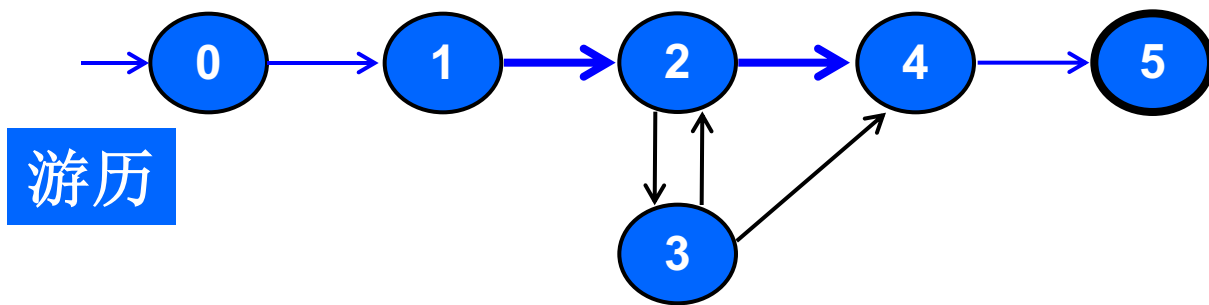
■ 侧访游历 (Sidetrips)

测试路径 p 侧访路径 q 当且仅当 q 的每条边在 p 中，且边次序相同

■ 绕道游历 (Detours)

测试路径 p 绕道路径 q 当且仅当 q 的每个节点在 p 中，且节点次序相同

图覆盖—结构覆盖



图覆盖 – 数据流

目标：检测变量的定义和使用

定义(def)：变量被赋值

使用(use)：变量被用于计算或判定

$\text{def}(n) = \{x \mid \text{节点}n\text{包含}x\text{的定义}\}$

$\text{def}(e) = \{x \mid \text{边}e\text{包含}x\text{的定义}\}$

$\text{use}(n) = \{x \mid \text{节点}n\text{包含}x\text{的使用}\}$

$\text{use}(e) = \{x \mid \text{边}e\text{包含}x\text{的使用}\}$

图覆盖 - 数据流

DU-对

位置对 (l_i, l_j) , 一个变量在 l_i 被定义, 在 l_j 被使用

du-路径

从变量的一个定义到它的一个使用的路径, 且中间没有对该变量重新定义 (**定义清纯, def-clear**)

du (n_i, n_j, v)

从 n_i 到 n_j 的变量 v 的 du-路径集合

du (n_i, v)

从 n_i 开始的变量 v 的 du-路径集合

图覆盖 - 数据流

1, 全定义覆盖 (all-defs)

对每个 du -路径集合 $S = du(n, v)$, 至少测试 S 中的一条路径

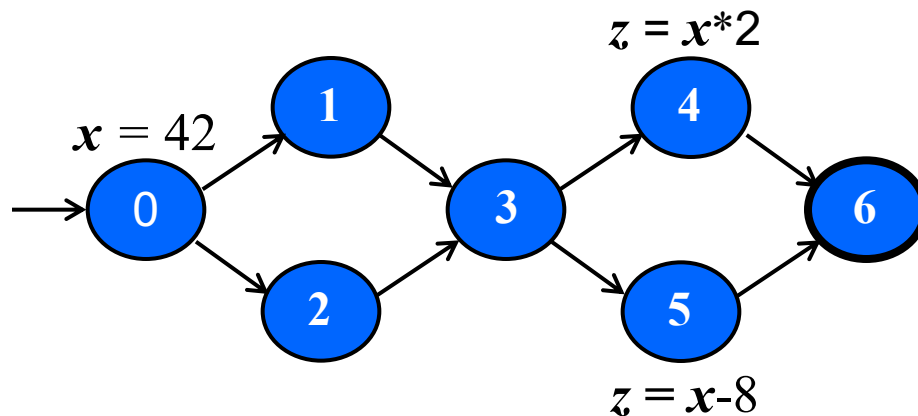
2, 全使用覆盖 (all-uses)

对每个 du -路径集合 $S = du(n_i, n_j, v)$, 至少测试 S 中的一条路径

3, 全 du -路径覆盖 (all-du-paths)

对每个 du -路径集合 $S = du(n_i, n_j, v)$, 测试 S 中的所有路径

图覆盖 - 数据流



全定义 (x)

[0, 1, 3, 4]

全使用 (x)

[0, 1, 3, 4]

[0, 1, 3, 5]

全定义-使用 (x)

[0, 1, 3, 4]

[0, 2, 3, 4]

[0, 1, 3, 5]

[0, 2, 3, 5]