

软件质量保证与软件测试

--数据流测试

主讲教师:

Email:

西安交大软件学院

什么是数据流测试

- 数据流测试是从关注程序中数据及其使用的角度，来设计测试用例的。像是一种路径测试覆盖，但关心的是数据变量而不是程序结构
- 数据流测试指关注变量接收值的点和使用（或引用）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”
- 数据流测试两种基本方法：
 - 提供一组基本定义和一种统一的测试覆盖指标结构测试
 - 基于“程序片”的概念的测试

西安交大软件学院

数据流测试

- 早期的数据流分析常常集中于现在叫做定义/引用异常的缺陷：
 - 变量被定义但是从来没有使用
 - 所使用的变量没有被定义
 - 变量在使用之前被定义两次
- 这些异常可以通过程序的索引表发现。由于索引表信息是由编译器生成的，因此这些异常可以通过所谓静态分析发现，即在不执行被测程序的情况下发现源代码失效

主要内容

- 定义/使用测试
- 定义/使用路径覆盖指标
- 基于程序片的测试
- 指导方针和观察

■ 定义/使用测试

- 对于程序 P ，有程序图 $G(P)$ ， V 是程序中变量的集合，节点表示语句片段，边表示节点序列
- $G(P)$ 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
- $PATHS(P)$ ： P 中所有路径的集合

西安交大软件学院

■ 定义

- 定义节点：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
- 如果执行对应这种语句的节点，那么与该变量关联的存储单元的内容就会改变
- 即：能改变变量值的语句

西安交大软件学院

■ 定义

- 使用节点：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
- 如果执行对应这种语句的节点，那么与该变量关联的存储单元的内容会保持不变
- 即：使用变量但不会改变变量的值

西安交大软件学院

■ 定义

- 谓词使用和计算使用：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**）
- 对应于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

西安交大软件学院

定义

■定义-使用路径：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

■定义-清除路径：

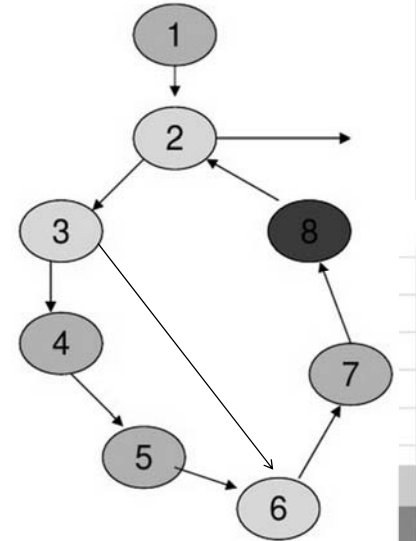
关于变量 v 的定义清除路径（记做 **dc-path**），是具有最初和最终节点 **DEF** (v, m) 和 **USE** (v, n) 的 **PATHS** (P) 中的路径，使得该路径中没有其他节点是 v 的定义节点

不是定义清除的定义-使用路径，是潜在的有问题的地方

西安交大软件学院

示例-变量的定义与使用

```
1. a=5; //定义a
2. while(C1) {
3.     if (C2) {
4.         b = a*a; //定义b、使用a
5.         a = a - 1; //定义a、使用a
6.     }
7.     print(a); //使用a
8.     print(b); //使用b }
```



西安交大软件学院

示例-变量的定义与使用

■定义/使用节点

变量	定义节点	使用节点
a	1,5	4,5,7
b	4	8

■定义/使用路径

变量	路径（开始、结束）节点	是否定义清除
a	1,4	是否否是
	1,5	
	1,7	
	5,7	
b	4,8	是

示例--佣金问题的代码(1)

```
1 Program Commission (INPUT, OUTPUT)
2 Dim locks, stocks, barrels As Integer
3 Dim lock Price, stock Price, barrel Price As Real
4 Dim total Locks, total Stocks, total Barrels As Integer
5 Dim lock Sales, stock Sales, barrel Sales As Real
6 Dim sales, commission As Real
7 lock Price = 45.0
8 stock Price = 30.0
9 barrel Price = 25.0
10 total locks = 0
11 total Stocks = 0
12 total Barrels = 0

13 Input (locks)
14 While NOT (locks = -1)
15     Input (stocks, barrels)
16     total Locks = total Locks + locks
17     total Stocks = total Stocks + stocks
18     total Barrels = total Barrels + barrels
19     Input (locks)
20 End While
```

佣金问题的代码(2)

```

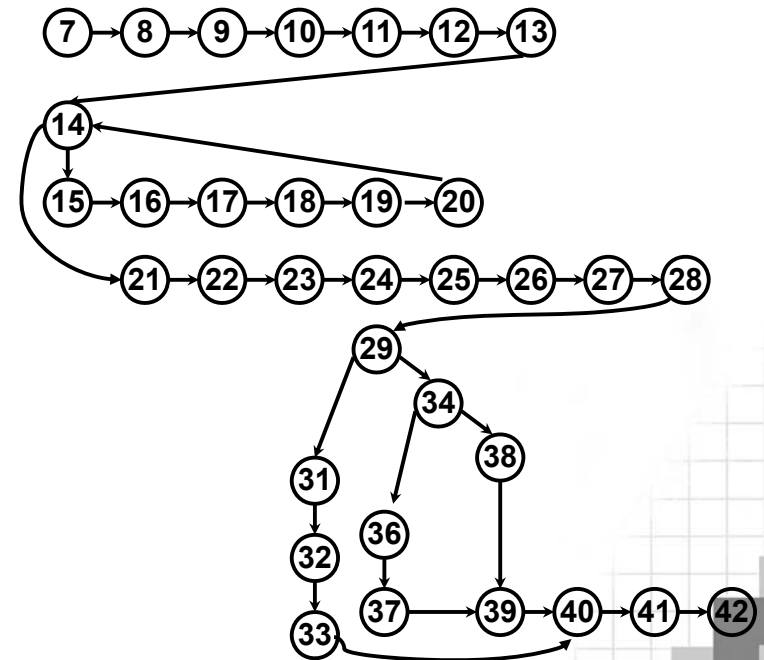
21 Output ("Locks sold: ", total Locks)
22 Output ("Stocks sold: ", total Stocks)
23 Output ("Barrels sold: ", total Barrels)
24 lock Sales = lock Price * total Locks
25 stock Sales = stock Price * total Stocks
26 barrel Sales = barrel Price * total Barrels
27 sales = lock Sales + stock Sales + barrel Sales
28 Output ( "Total sales:" , sales)

29 If (sales > 1800.0)
30   Then
31     commission = 0.1 * 1000.0
32     commission = commission + 0.15 * 800.0
33     commission = commission + 0.20 * (sales-1800.0)
34   Else
35     If (sales > 1000.0)
36       Then
37         commission = 0.10 * 1000.0
38         commission = commission + 0.15 * (sales-1000.0)
39       Else
40         commission = 0.10 * sales
41     End If
42 End If
41 Output ("Commission is $" , commission)
42 End Commission

```

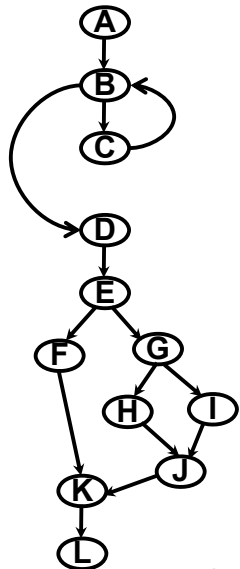
学院

佣金问题的程序图



西安交大软件学院

佣金问题的DD-路径图



DD-路径	节点
A	7、8、9、10、11、12、13
B	14
C	15、16、17、18、19、20
D	21、22、23、24、25、26、27、28
E	29
F	31、32、33
G	34
H	36、37
I	38
J	39
K	40
L	41、42

西安交大软件学院

佣金问题变量的定义/使用节点

变量	定义节点	使用节点
lockPrice	7	24
stockPrice	8	25
barrelPrice	9	26
totalLocks	10, 16	16, 21, 24
totalStocks	11, 17	17, 22, 25
totalBarrels	12, 18	18, 23, 26
locks	13, 19	14, 16
stocks	15	17
barrels	15	18
lockSales	24	27
stockSales	25	27
barrelSales	26	27
sales	27	28, 29, 33, 34, 37, 38
commission	31, 32, 33, 36, 37, 38	32, 33, 37, 41

Downloader: 王博玉

西安交大软件学院

佣金问题中的部分定义/使用路径

变量	路径(开始, 结束)节点	是定义清除吗?
lock Price	7, 24	是
stock Price	8, 25	是
barrel Price	9, 26	是
total Stocks	11, 17	是
total Stocks	11, 22	否
total Stocks	11, 25	否
total Stocks	17, 17	是
total Stocks	17, 22	否
total Stocks	17, 25	否
locks	13, 14	是
locks	19, 14	是
locks	13, 16	是
locks	19, 16	是
sales	27, 28	是
sales	27, 29	是
sales	27, 33	是
sales	27, 34	是
sales	27, 37	是
sales	27, 38	是

stocks的定义-使用路径

■ 变量stocks:

- DEF (stocks, 15)
- USE (stocks, 17)
- 路径<15,16, 17>

■ stocks没有其他定义节点，因此这条路径是定义清除路径

西安交大软件学院

locks的定义-使用路径

■ 变量locks:

- DEF(locks, 13)、DEF(locks, 19)、USE (locks, 14)和USE (locks, 16)

■ p1=<13,14>

■ p2=<13,14,15,16>

■ p3=<19,20,14>

■ p4=<19,20,14,15,16>

扩展路径:

p1'=<13,14,21>

p3'=<19,20,14,21>

p1',p2,p3',p4构成while循环测试用例的非常完备的集合---旁路循环、开始循环、退出循环、重复循环，所有这些定义-使用路径都是定义清除路径

!仅限个人使用*请勿上传至互联网*违者必究!

西安交大软件学院

totalLocks的定义-使用路径

■ 两个定义节点:

- DEF (totalLocks, 10)、DEF (totalLocks, 16)

■ 三个使用节点:

- USE (totalLocks, 16)、USE(totalLocks, 21)和USE (totalLocks, 24)

变量	路径节点(开始, 结束)	是定义清除吗?
totalLocks	10, 16	是
totalLocks	10, 21	否
totalLocks	10, 24	否
totalLocks	16, 16	退化, 不允许这种情况作为定义-使用路径
totalLocks	16, 21	是
totalLocks	16, 24	是

Downloader: 王博玉

西安交大软件学院

sales的定义-使用路径

sales只使用了一个定义节点，因此关于sales的所有定义-使用路径都必须是定义清除的

变量	路径节点(开始, 结束)	是定义清除吗?
sales	27, 28	是
sales	27, 29	是
sales	27, 33	是
sales	27, 34	是
sales	27, 37	是
sales	27, 38	是

西安交大软件学院

commission的定义-使用路径(1)

变量	路径节点(开始, 结束)	是否可行?	是定义清除吗?
commission	31, 32	是	是
commission	31, 33	是	否
commission	31, 37	否	—
commission	31, 41	是	否
commission	32, 32	是	是
commission	32, 33	是	是
commission	32, 37	否	—
commission	32, 41	是	否
commission	33, 32	否	—
commission	33, 33	是	是
commission	33, 37	否	—

西安交大软件学院

commission的定义-使用路径(2)

变量	路径(开始, 结束)节点	是否可行?	是定义清除吗?
commission	33, 41	是	是
commission	36, 32	否	—
commission	36, 33	否	—
commission	36, 37	是	是
commission	36, 41	是	否
commission	37, 32	否	—
commission	37, 33	否	—
commission	37, 37	是	是
commission	37, 41	是	是
commission	38, 32	否	—
commission	38, 33	否	—
commission	38, 37	否	—
commission	38, 41	是	是

西安交大软件学院

commission的定义-使用路径

■ 语句31~33使用存储单元保存中间值，累加佣金值。这些中间值在定义-使用路径中表现为定义节点和使用节点。这里将像31、32、36的赋值语句从定义-使用路径中排除

■ 只考虑三个"实际"定义节点

- DEF (commission,33)
- DEF(commission, 37)
- DEF (commission, 38)

■ 一个使用节点:

- USE (commission, 41)

变量	路径(开始, 结束)节点	是否可行?	是定义清除吗?
commission	33, 41	是	是
commission	37, 41	是	是
commission	38, 41	是	是

西安交大软件学院

佣金问题的定义-使用路径

编号	变量	路径(开始, 结束)节点
1	lockPrice	7,24
2	stockPrice	8,25
3	barrelPrice	9,26
4	stocks	15,17
5	barrels	15,18
6	locks	13,14
7	locks	13,16
8	locks	19,14
9	locks	19,16
10	totalLocks	10, 16
11	totalLocks	10, 21
12	totalLocks	10, 24
13	totalLocks	16, 21
14	totalLocks	16, 24
15	totalStocks	11, 17
16	totalStocks	11, 22
17	totalStocks	11, 25
18	totalStocks	17, 22

编号	变量	路径(开始, 结束)节点
19	totalStocks	17, 25
20	totalBarrels	12, 18
21	totalBarrels	12, 23
22	totalBarrels	12, 26
23	totalBarrels	18, 23
24	totalBarrels	18, 26
25	lockSales	24, 27
26	stockSales	25,27
27	barrelSales	26,27
28	sales	27, 28
29	sales	27, 29
30	sales	27, 33
31	sales	27, 34
32	sales	27, 37
33	sales	27, 38
34	commission	33, 41
35	commission	37, 41
36	commission	38, 41

主要内容

- 定义/使用测试
- 定义/使用路径覆盖指标
- 基于程序片的测试
- 指导方针和观察

西安交大软件学院

定义

■ 定义1:

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

■ 定义2:

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3:

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

!仅限个人使用, 请勿上传至互联网*违者必究!

西安交大软件学院

定义

■ 定义4:

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

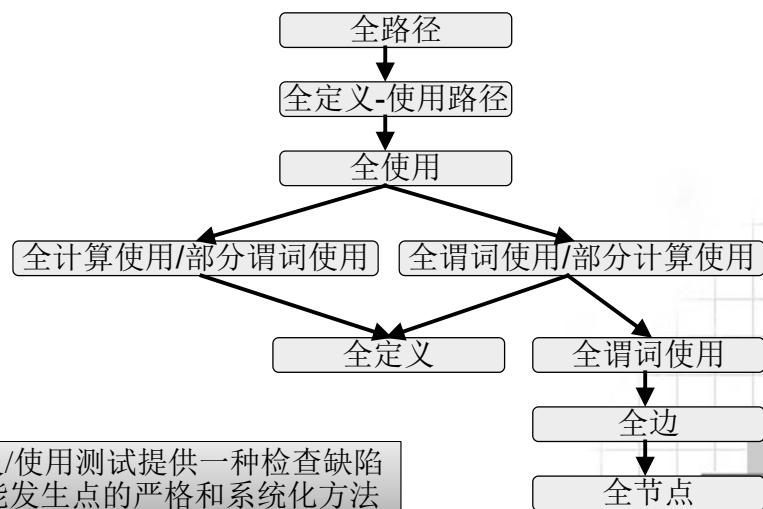
■ 定义5:

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

西安交大软件学院

Downloader: 王博玉

数据流覆盖指标Rapps/Weyuker层次结构



西安交大软件学院

佣金问题的DD-路径覆盖情况（决策表方法）

Case	locks	stocks	barrels	sales	commission	DD-Paths
1	5	5	5	500	50	A-E,G,I,J,K,L
2	15	15	15	1500	175	A-E,G,H,J,K,L
3	25	25	25	2500	360	A-E,F,K,L

西安交大软件学院

佣金问题的du-路径覆盖情况（决策表方法）

du-Path	Case 1	Case 2	Case 3
1	X	X	X
2	X	X	X
3	X	X	X
4	X	X	X
5	X	X	X
6	X	X	X
7	X	X	X
8	X	X	X
9			
10	X	X	X
11	X	X	X
12	X	X	X
13	X	X	X
14	X	X	X
15	X	X	X
16	X	X	X
17	X	X	X
18	X	X	X

du-Path	Case 1	Case 2	Case 3
19	X	X	X
20	X	X	X
21	X	X	X
22	X	X	X
23	X	X	X
24	X	X	X
25	X	X	X
26	X	X	X
27	X	X	X
28	X	X	X
29	X	X	X
30			X
31	X	X	
32		X	
33	X		
34			X
35		X	
36	X		

西安交大软件学院

主要内容

- 定义/使用测试
- 定义/使用路径覆盖指标
- 基于程序片的测试
- 指导方针和观察

Downloader: 王博玉

西安交大软件学院

■ 程序(切)片的基本思想

- 对程序进行裁剪，使得所得到的程序代码仍能反映原程序的部分特征
- 程序切片专门针对这类问题，它按切片准则来裁剪程序，使人们能把注意力集中在相关的程序代码上
- Weiser的切片准则
 - $C = \langle q, v \rangle$ ，其中 q 是程序中的一条语句， v 是一个程序变量
 - 当程序的下一执行语句为 q 时，删去源程序中不会影响变量 v 值的语句，所得到的就是关于 $\langle q, v \rangle$ 的静态切片

西安交大软件学院

■ 程序(切)片的举例

1 input(k);	1 input(k);
2 t:=0;	2 t:=0;
3 f:=1;	
4 while (k > 0) do	4 while (k > 0) do
5 { t:=t + k;	5 { t:=t + k;
6 f:=f * k;	
7 k:=k -1;	7 k:=k -1;
8 output(t);	8 output(t);
9 output(f); }	}

(a) 程序例子

(b) 程序例子关于切片准则
 $C = \langle 8, t \rangle$ 的静态切片

西安交大软件学院

■ 程序片的定义

程序 P 有程序图 $G(P)$ 和一个程序变量集合 V

■ 定义一：

给定一个程序 P 和 P 中的一个变量集合 V ，变量集合 V 在语句 n 上的一个片，记做 $S(V, n)$ ，是 P 中对 V 中的变量值作出贡献的所有语句集合

■ 定义二：

给定一个程序 P 和一个给出语句及语句片段编号的程序图 $G(P)$ ，以及 P 中的一个变量集合 V ，变量集合 V 在语句片段 n 上的一个片，记做 $S(V, n)$ ，是 P 中在 n 以前对 V 中的变量值作出贡献的所有语句片段编号的集合

片的思想，是将程序分成具有某种(功能)含义的组件

!仅限个人使用*请勿上传至互联网*违者必究!

西安交大软件学院

■ 使用节点和定义节点

■ USE关系适合五种形式的使用：

- 谓词使用 用在谓词(判断)中
- 计算使用 用在计算中
- 输出使用 用于输出
- 定位使用 用于定位(指针、下标)
- 迭代使用 迭代(内部计数器、循环指示)

■ 两种定义节点：

- 输入定义 通过输入定义
- 赋值定义 通过赋值定义

Downloader: 王博玉

西安交大软件学院

■ 使用节点和定义节点

■假设片 $S(V, n)$ 是一个变量上的片，即集合 V 由单一变量 v 组成

◆如果语句片段 n 是 v 的一个定义节点，则 n 包含在该片中

◆如果语句片段 n 是 v 的使用节点，则 n 不包含在该片中

■其他变量(不是片集合 V 中的 v)的谓词使用和计算使用，要包含，（只要其执行会影响变量 v 的取值）

■指导方针：如果不管是否包含语句片段， v 的值都保持不变，那么排除该语句片段

西安交大软件学院

■ 举例：佣金问题数据流测试

变量 $locks$ 上的片：

■ 变量 $locks$ 有2个使用节点14、16，2个定义节点13、19，则：

$S_1: S(locks, 13) = \{13\}$

$S_2: S(locks, 14) = \{13, 14, 19, 20\}$

$S_3: S(locks, 16) = \{13, 14, 19, 20\}$

$S_4: S(locks, 19) = \{13, 14, 19, 20\}$

西安交大软件学院

■ 变量 $stocks$ 和 $barrels$ 上的片

■ 变量 $stocks$ 和 $barrels$ 要受循环变量 $locks$ 的影响,所以

$S_5: S(stocks, 15) = \{13, 14, 15, 19, 20\}$

$S_6: S(stocks, 17) = \{13, 14, 15, 19, 20\}$

$S_7: S(barrels, 15) = \{13, 14, 15, 19, 20\}$

$S_8: S(barrels, 18) = \{13, 14, 15, 19, 20\}$

西安交大软件学院

■ $totalLocks$ 的片

■ 变量 $totalLocks$,节点10是赋值定义,节点16既是赋值又是计算,且存在循环。而节点21和24分别是输出和计算节点。所以：

$S_9: S(totalLocks, 10) = \{10\}$

$S_{10}: S(totalLocks, 16) = \{10, 13, 14, 16, 19, 20\}$

$S_{11}: S(totalLocks, 21) = \{10, 13, 14, 16, 19, 20\}$

$S_{11-1}: S(totalLocks, 24) = \{10, 13, 14, 16, 19, 20\}$

西安交大软件学院

totalStocks和totalBarrels的片

$S_{12}: S(\text{totalStocks}, 11) = \{11\}$

$S_{13}: S(\text{totalStocks}, 17) = \{11, 13, 14, 15, 17, 19, 20\}$

$S_{14}: S(\text{totalStocks}, 22) = \{11, 13, 14, 15, 17, 19, 20\}$

$S_{14-1}: S(\text{totalStocks}, 25) = \{11, 13, 14, 15, 17, 19, 20\}$

$S_{15}: S(\text{totalBarrels}, 12) = \{12\}$

$S_{16}: S(\text{totalBarrels}, 18) = \{12, 13, 14, 15, 18, 19, 20\}$

$S_{17}: S(\text{totalBarrels}, 23) = \{12, 13, 14, 15, 18, 19, 20\}$

$S_{17-1}: S(\text{totalBarrels}, 26) = \{12, 13, 14, 15, 18, 19, 20\}$

西安交大软件学院

赋值定义

- 变量lockPrice、stockPrice、barrelPrice、lockSales、stockSales、barrelSales都是1个定义，1个使用，不涉及循环

$S_{18}: S(\text{lockPrice}, 24) = \{7\}$

$S_{19}: S(\text{stockPrice}, 25) = \{8\}$

$S_{20}: S(\text{barrelPrice}, 26) = \{9\}$

$S_{21}: S(\text{lockSales}, 24) = \{7, 10, 13, 14, 16, 19, 20, 24\}$

$S_{22}: S(\text{stockSales}, 25) = \{8, 11, 13, 14, 15, 17, 19, 20, 25\}$

$S_{23}: S(\text{barrelSales}, 26) = \{9, 12, 13, 14, 15, 18, 19, 20, 26\}$

西安交大软件学院

Sales上的片

- sales变量只有1个定义节点27，有28、29、33、34、37、38共6个使用节点，但影响其值的变量是除commission以外的所有变量

$S_{24}: S(\text{sales}, 27) = \{7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 24, 25, 26, 27\}$

$S_{25}: S(\text{sales}, 28) = \{7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 24, 25, 26, 27\}$

$S_{26}: S(\text{sales}, 29) = \{7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 24, 25, 26, 27\}$

$S_{27}: S(\text{sales}, 33) = \{7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 24, 25, 26, 27\}$

$S_{28}: S(\text{sales}, 34) = \{7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 24, 25, 26, 27\}$

$S_{29}: S(\text{sales}, 37) = \{7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 24, 25, 26, 27\}$

$S_{30}: S(\text{sales}, 38) = \{7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 24, 25, 26, 27\}$

$S_{24} = S_{10} \cup S_{13} \cup S_{16} \cup S_{18} \cup S_{19} \cup S_{20}$

或: $S_{24} = S_{21} \cup S_{22} \cup S_{23}$

注: 除节点27

!仅限个人学习使用,请勿上传至互联网*违者必究!

Commission上的片

- 变量commission有6个定义节点，4个使用节点，所以：

$S_{31}: S(\text{commission}, 31) = \{31\}$

$S_{32}: S(\text{commission}, 32) = \{31, 32\}$

$S_{33}: S(\text{commission}, 33) = \{7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 24, 25, 26, 27, 29, 30, 31, 32, 33\}$

$S_{34}: S(\text{commission}, 36) = \{36\}$

$S_{35}: S(\text{commission}, 37) = \{7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 24, 25, 26, 27, 29, 34, 35, 36, 37\}$

$S_{36}: S(\text{commission}, 38) = \{7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 24, 25, 26, 27, 29, 34, 38\}$

不管进行什么计算，最后都会汇集到最后片上：

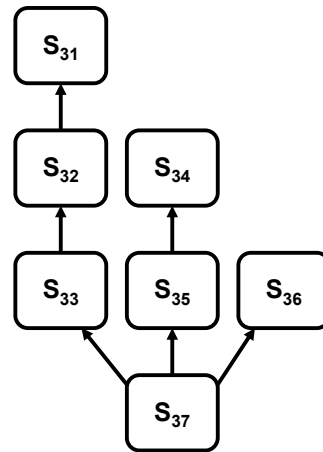
$S_{37}: S(\text{commission}, 41) = \{7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38\}$

西安交大软件学院

Downloader: 王博玉

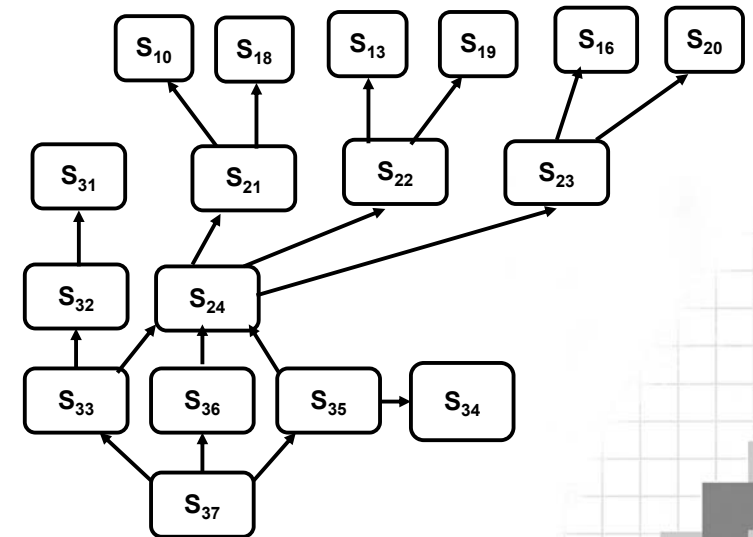
Commission上的片格

- 该图所示的格是一种有向无环路图，其中：节点是片，边表示合适的子集关系
- 这种格使片节点的位置大致对应其源代码所在的位置



西安交大软件学院

Sales和commission上的格



西安交大软件学院

讨论

- 程序片的方法,将程序进行了分段划分
- 如佣金问题的commission是最终结果，它的计算依赖于sales的计算；而sales的计算，又依赖于lockSales、stockSales、barrelSales的计算；而lockSales、stockSales、barrelSales的计算又分别依赖于totalLocks、totalStocks、totalbarrels的计算等等
- 如果所有变量的定义和使用是正确的，当然程序就是正确的！

讨论

- 不同变量程序片的补集，为变量的分段划分提供了方法
 - 如：S (commission, 41) - S (sales, 28) = {29~41}
 - 将commission的计算分为sales计算和之后的计算，为错误的定位提供了依据

风格与技术

- 1. 永远不要在不出现在语句片段 n 里的 V 的变量 v 上建立片 $S(V, n)$
- 2. 在一个变量上建立片。片 $S(V, n)$ 中的集合 V 可以包含多个变量，有时这样的片也是有用的，但通常 V 只包含一个变量
- 3. 对所有赋值定义节点都建立片
- 4. 对谓词使用节点建立片
- 5. 考虑使片可编译
 - 在片定义中并不要求语句集合是可编译的，但是如果使片可编译，则意味着编译器指令和数据说明语句集合是每个片的子集

西安交大软件学院

主要内容

- 定义/使用测试
- 定义/使用路径覆盖指标
- 基于程序片的测试
- 指导方针和观察

西安交大软件学院

指导方针与观察

- 1. 数据流测试适用于计算密集的程序
 - 定义-使用路径和片的定义，使我们能非常准确地描述要测试的程序部分
- 2. 片不能很好地映射到测试用例上
 - 因为其他非相关代码仍然在执行路径中
- 片提供了一种消除变量之间交互的方便手段。使用片合成方法来开发困难的代码部分，在把这些片接合(合成)到其他片上之前进行单独测试

西安交大软件学院

指导方针与观察

- 3. 片的相对补可提供诊断能力

集合 B 关于另一个集合 A 的相对补，是 A 中所有不在 B 中的元素集合，叫做 $A-B$

考虑相对补集 $S(\text{commission}, 48) - S(\text{sales}, 35)$:

$S(\text{commission}, 48) = \{3, 4, 5, 36, 18, 19, 20, 23, 24, 25, 26, 27, 34, 38, 39, 40, 44, 45, 47\}$

$S(\text{sales}, 35) = \{3, 4, 5, 36, 18, 19, 20, 23, 24, 25, 26, 27\}$

$S(\text{commission}, 48) - S(\text{sales}, 35) = \{34, 38, 39, 40, 44, 45, 47\}$

如果代码48行的`commission`存在问题，则可以将程序分成两部分：34行的`sales`计算以及35和48行的`commission`计算。如果34行的`sales`没有问题，则问题一定出在相对补中；如果不是，则可能是其他部分的问题。

西安交大软件学院

指导方针与观察

■4.在片和DD-路径之间存在多对多关系:一个片中的语句可以在多条DD-路径中,一条 DD-路径中的语句可能在多个片中。精心选择片的补可能与DD一路径相同

■4.如果要开发片的格,假设第一条语句是片很方便,这样片格会永远在一个根节点上终止。用双向箭头表示相等的片

■5.片可以反映出定义/引用信息

考虑totalLocks上的以下片:

S9: $S(\text{totalLocks}, 10) = \{10\}$

S10: $S(\text{totalLocks}, 16) = \{10, 13, 14, 16, 19, 20\}$

S11: $S(\text{totalLocks}, 21) = \{10, 13, 14, 16, 19, 20\}$

■如果片是相等的,则对应的路径就是定义清除的

西安交大软件学院

总结

- 定义-使用路径测试思想是什么?
- 基于程序片的测试思想是什么?
- 数据流测试适用范围是什么?

西安交大软件学院

作业(可选)

- 查阅相关文献,写一篇关于基于程序片测试的研究报告
- 要求:正文字数不少于**5000**字
- 需要提交报告、相关文献
- 需要自己进行语言组织、不允许直接抄袭
- 如果抄袭,分数为**0**