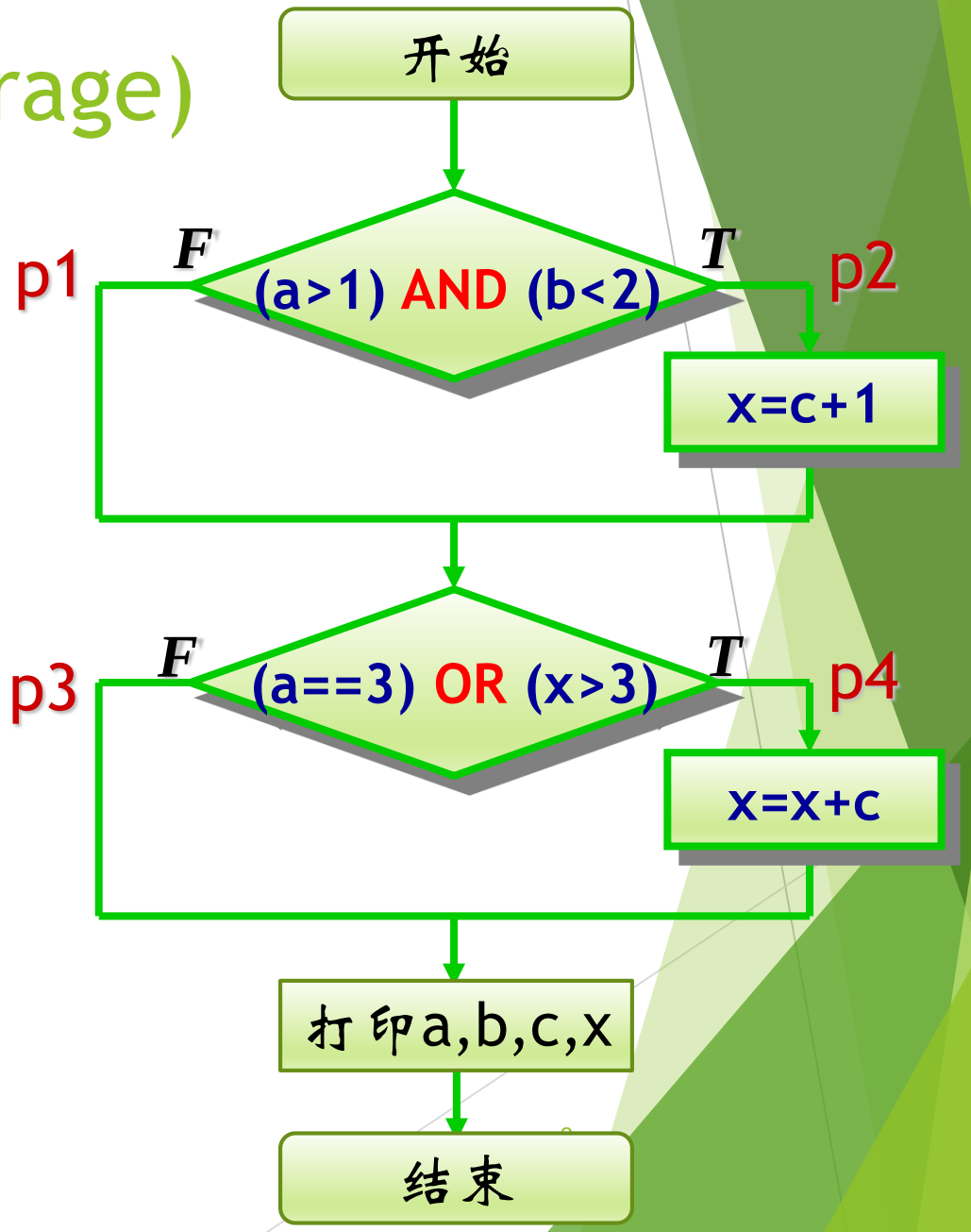
对判定的测试（中）

3. 条件覆盖(Condition coverage)

- 设计测试用例时应保证程序中每个复合判定表达式中，每个简单判定条件的取真和取假情况至少执行一次

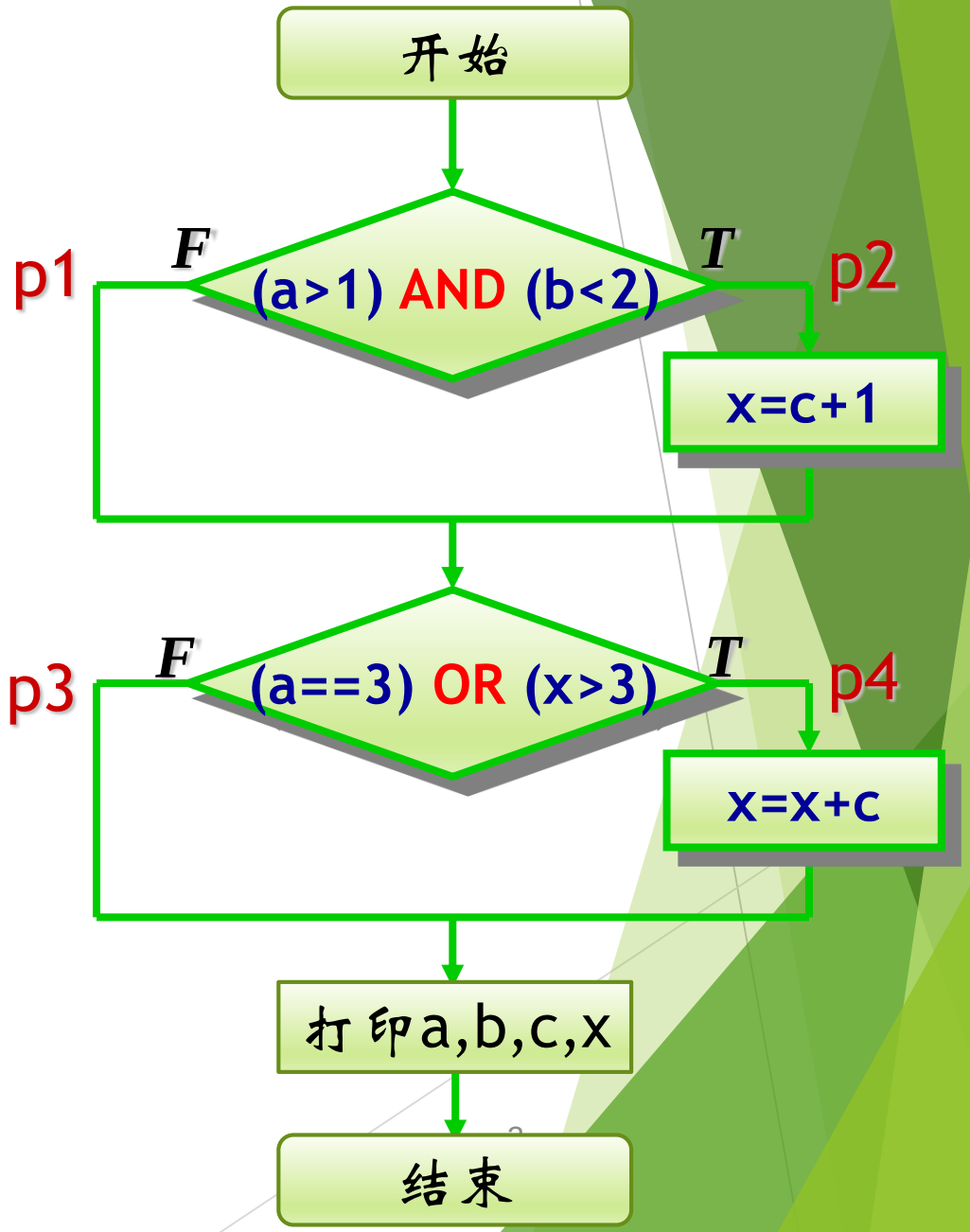
条件覆盖一定能满足判定覆盖吗？
条件覆盖真的没有漏洞吗？



测试用例设计

输入条件	取值	取值
T1:a>1	T	F
T2:b<2	F	T
T3:a=3	T	F
T4:x>3	F	T

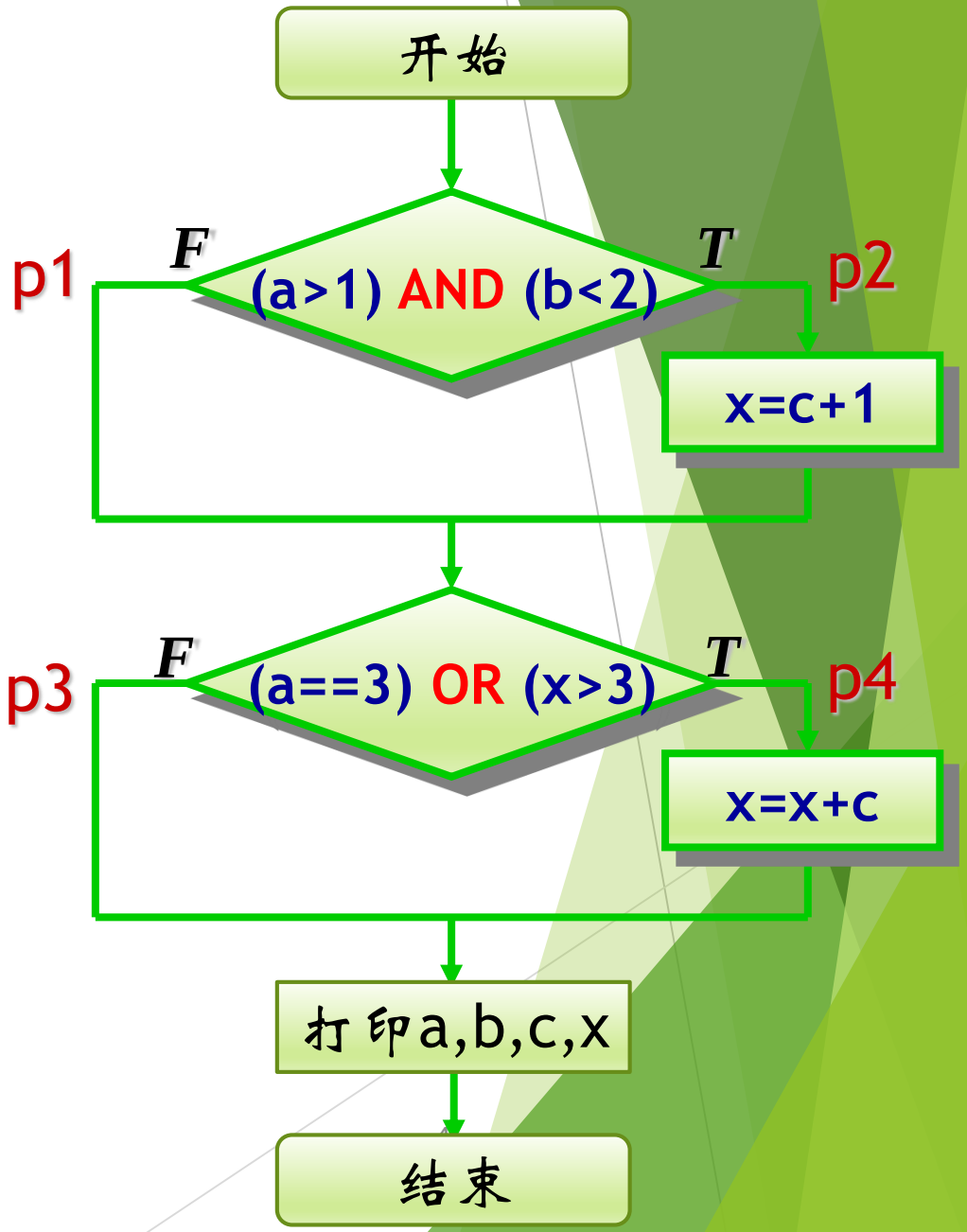
ID	输入				预期输出	通过路径	条件覆盖	判定覆盖
	a	b	c	x				
TC7	3	2	1	0	1	L14	100%	50%
TC8	1	1	1	4	5	L14		



测试用例设计

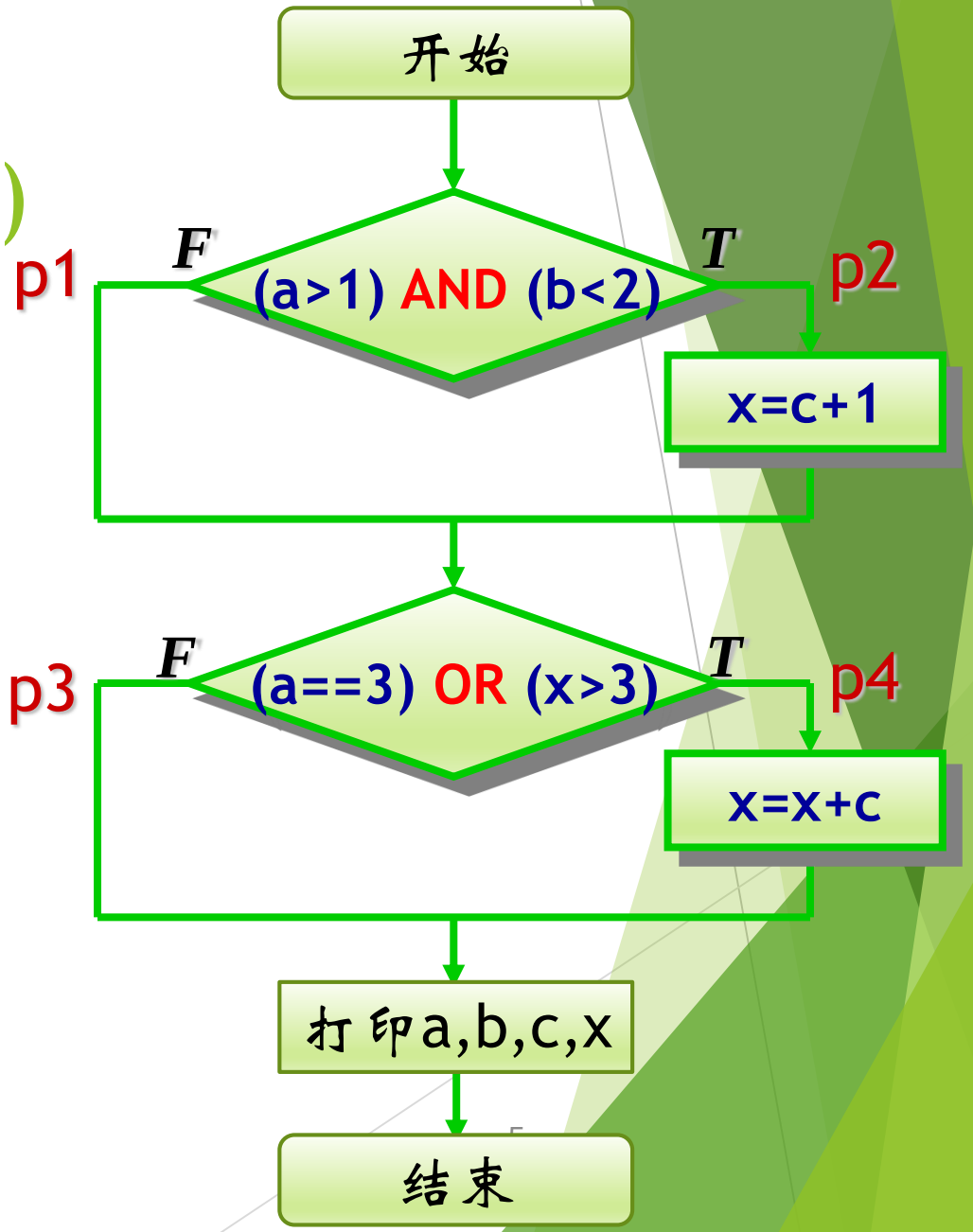
输入条件	取值	取值
T1:a>1	T	F
T2:b<2	T	F
T3:a=3	T	F
T4:x>3	T	F

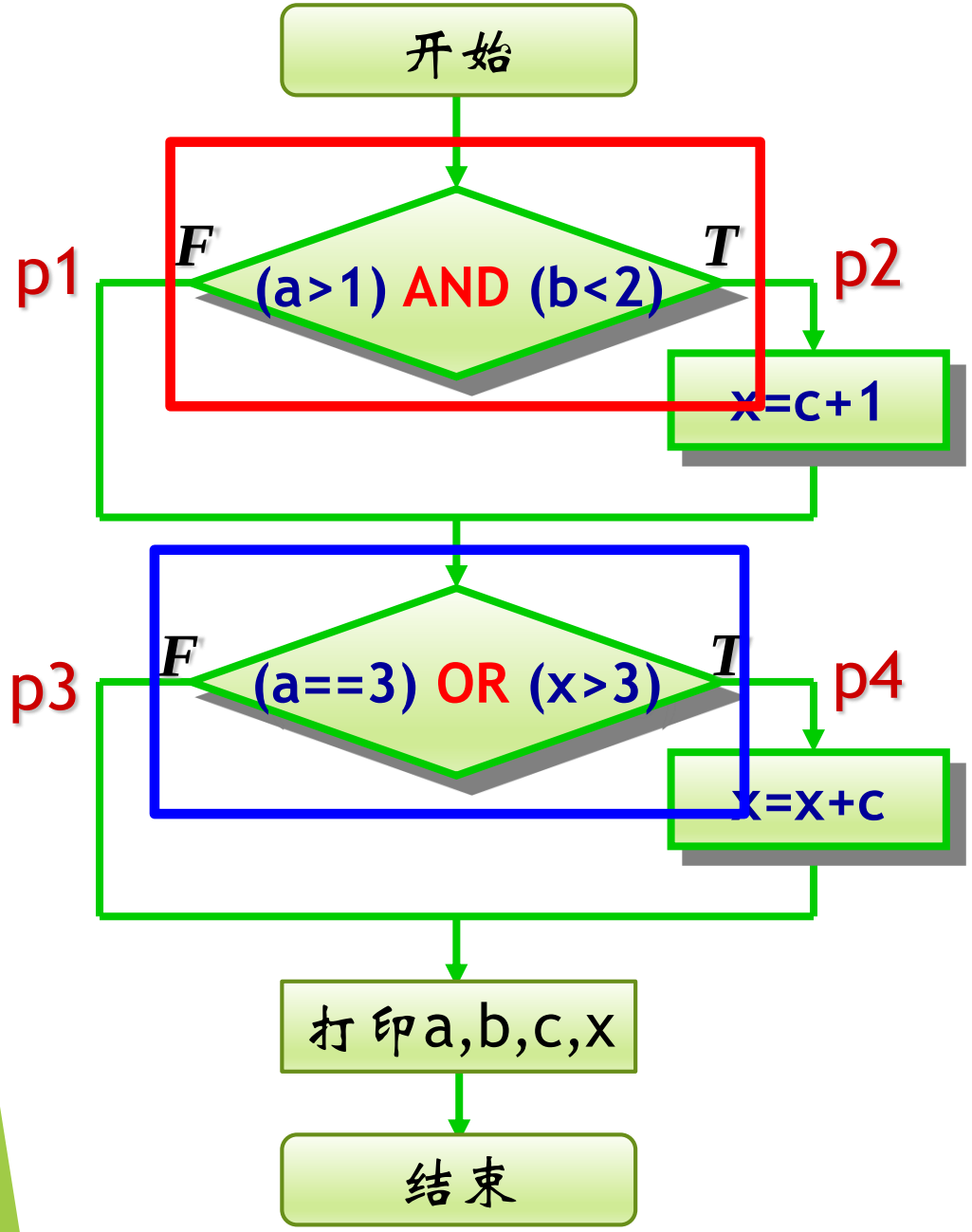
ID	输入				预期输出	通过路径	条件覆盖	判定覆盖
	a	b	c	x				
TC9	3	1	3	0	7	L24	100%	100%
TC10	1	2	2	0	0	L13		

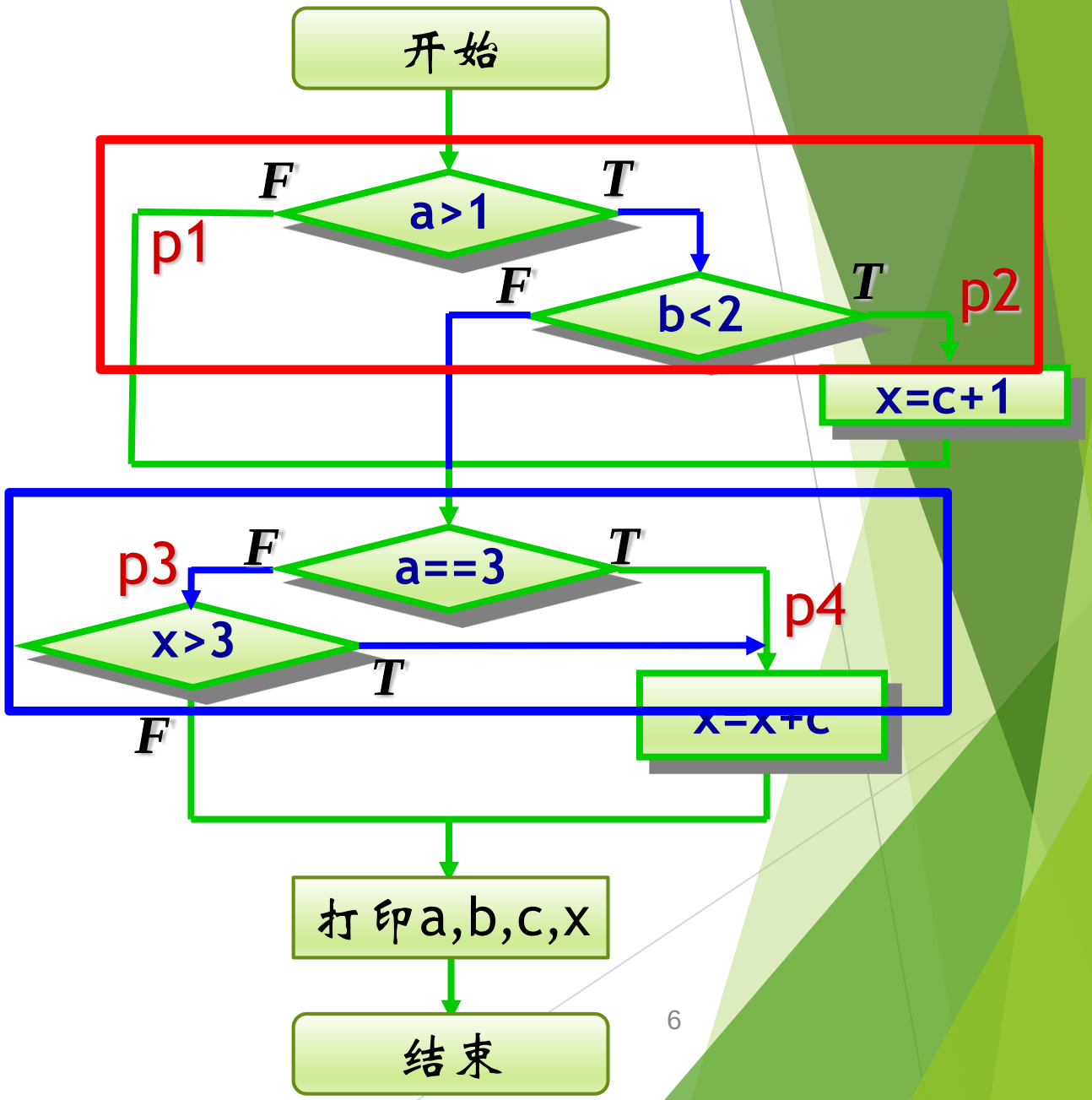


4. 判定/条件覆盖 (Branch/Condition coverage)

- 设计测试用例时应满足判定节点的取真、取假分支至少执行一次，且每个简单判定条件的取真和取假情况也至少执行一次
- 判定覆盖+条件覆盖

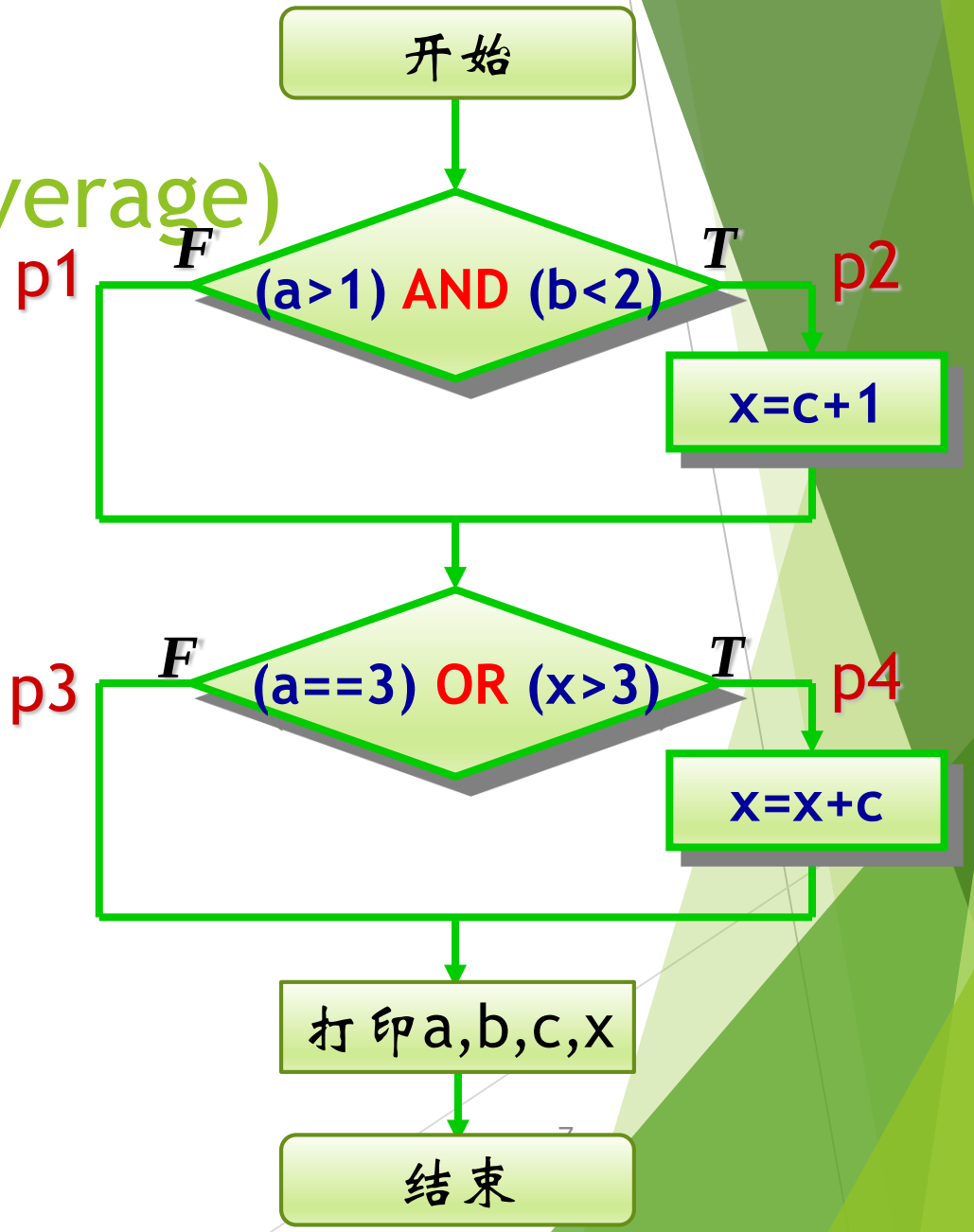



开始
p1
F
(a>1) AND (b<2)
T
p2
x=c+1
p3
F
(a==3) OR (x>3)
T
p4
x=x+c
打印a,b,c,x
结束


开始
p1
F
a>1
T
F
b<2
T
p2
x=c+1
p3
F
a==3
T
p4
x>3
T
F
x=x+c
打印a,b,c,x
结束

5. 条件组合覆盖 (Condition combination coverage)

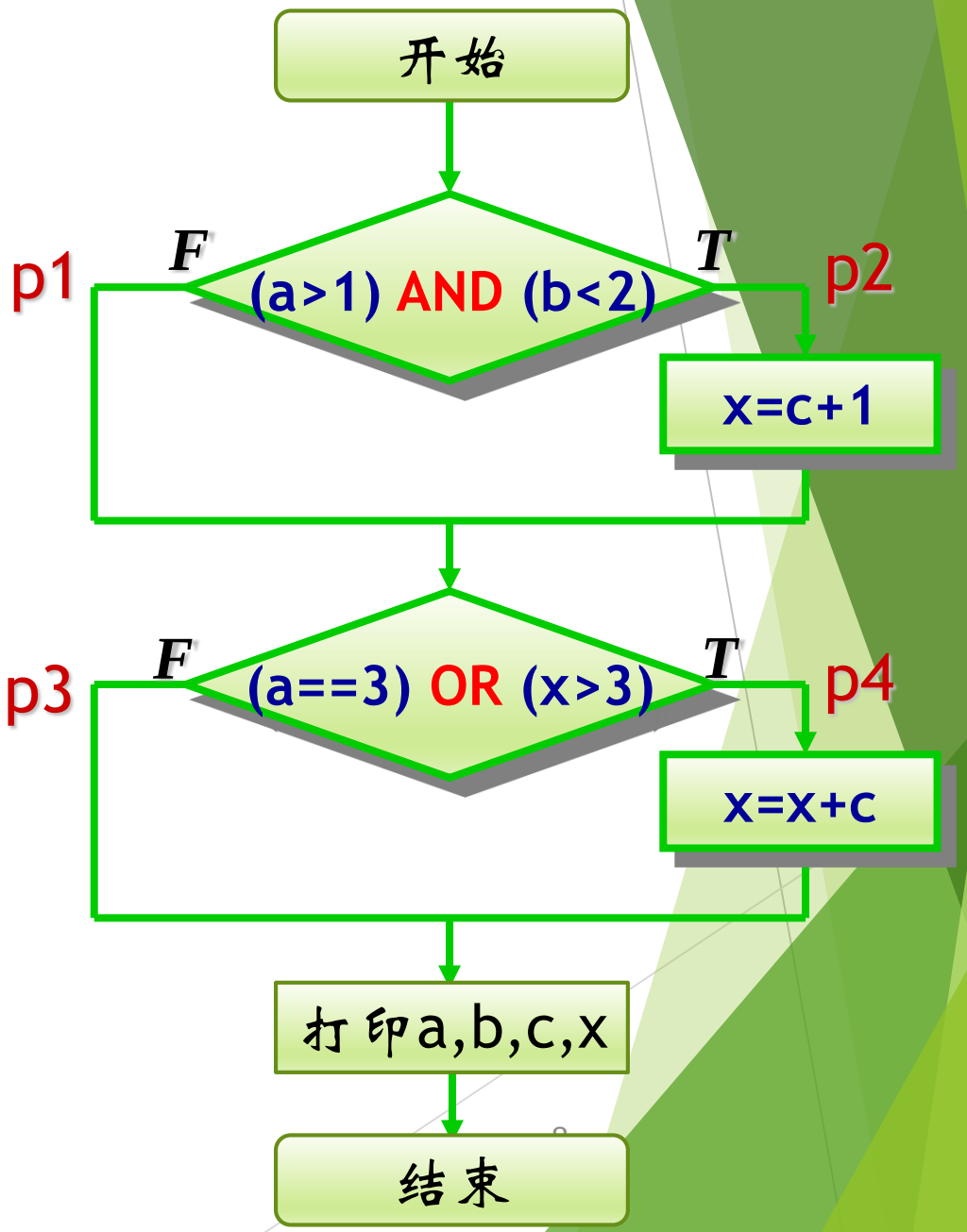
- 设计测试用例时应满足每个判定节点中，所有简单判定条件的所有可能的取值组合情况至少执行一次
- 本质：真值表



测试用例设计

	简单判定条件	
取值	T1:a>1	T2:b<2
	T	T
	T	F
	F	T
	F	F

	简单判定条件	
取值	T3:a==3	T4:x>3
	T	T
	T	F
	F	T
	F	F



测试用例设计（1）

序号	输入				判定条件				路径	备注
	a	b	c	x	T1	T2	T3	T4		
1	3	1	3	0	T	T	T	T	L24	
2	3	1	2	0	T	T	T	F	L24	
3	2	1	3	0	T	T	F	T	L24	
4	2	1	2	0	T	T	F	F	L23	
5	3	2	3	4	T	F	T	T	L14	
6	3	2	1	3	T	F	T	F	L14	
7	2	1	1	4	T	F	F	T	L14	
8	2	2	1	3	T	F	F	F	L13	

测试用例设计（2）

序号	输入				判定条件				路径	备注
	a	b	c	x	T1	T2	T3	T4		
					F	T	T	T		不存在
					F	T	T	F		不存在
9	1	1	1	4	F	T	F	T	L14	
10	2	2	1	3	F	T	F	F	L13	
					F	F	T	T		不存在
					F	F	T	F		不存在
11	1	2	1	4	F	F	F	T	L14	
12	1	2	1	3	F	F	F	F	L13	

条件组合覆盖

- 优势：方法简单
- 局限性：测试用例太多，冗余严重