

场景爆炸问题



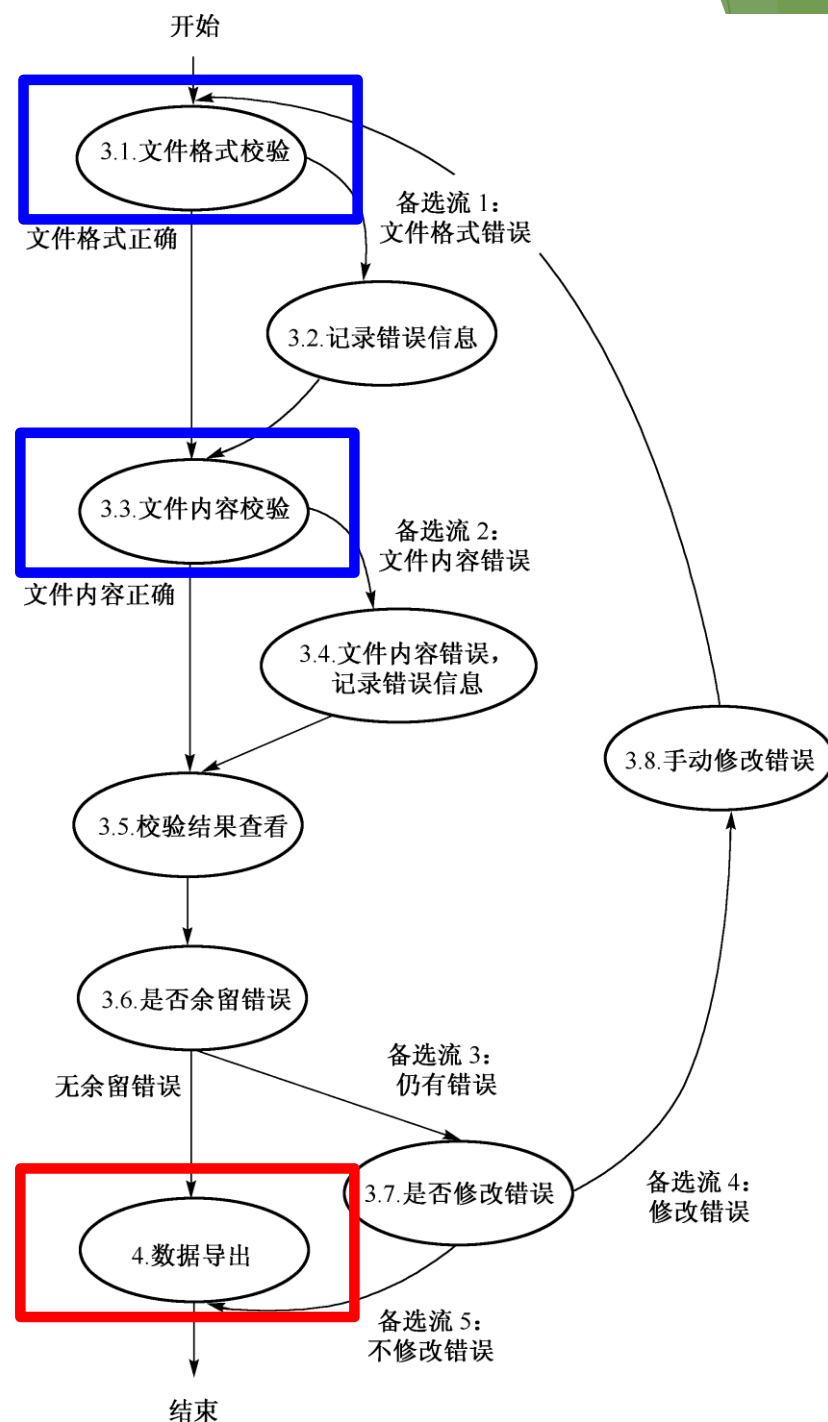
场景爆炸问题

如何从大量场景
中选出有限个典型场景来
测试？

场景爆炸问题

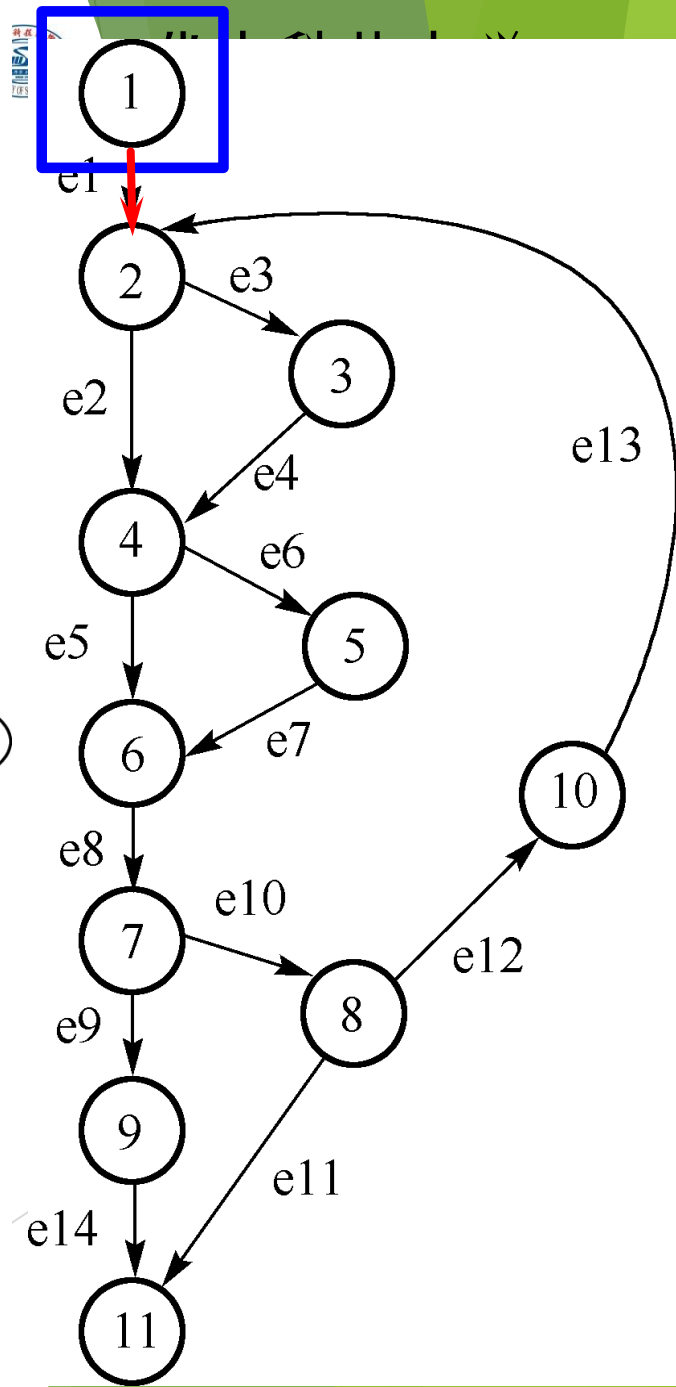
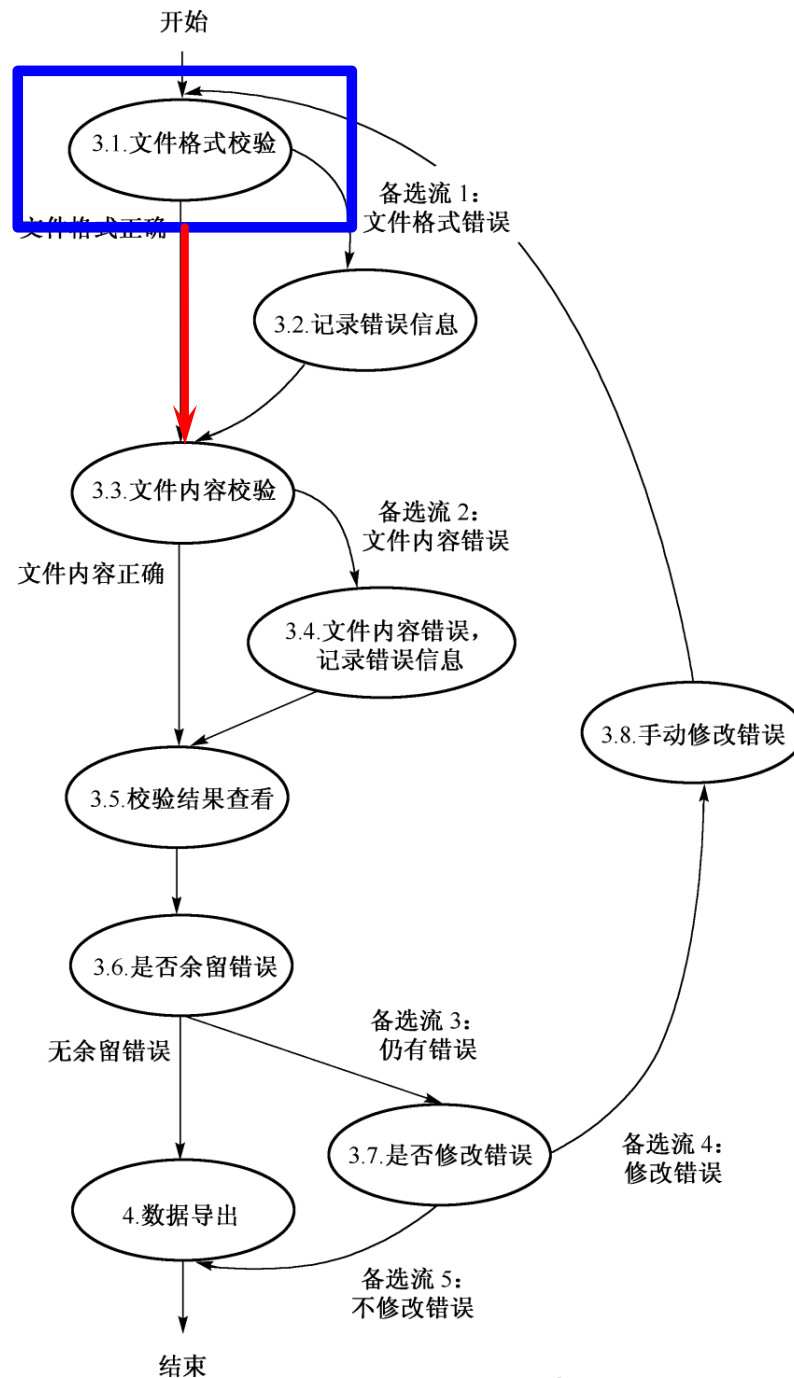
- ▶ 基本流：文件无错，数据成功导出
- ▶ 备选流1：文件格式错误
- ▶ 备选流2：文件内容错误
- ▶ 备选流3：信息文件中仍然存在错误
- ▶ 备选流4：要求修改错误
- ▶ 备选流5：不修改错误

测试最重要、风险最高的
典型场景



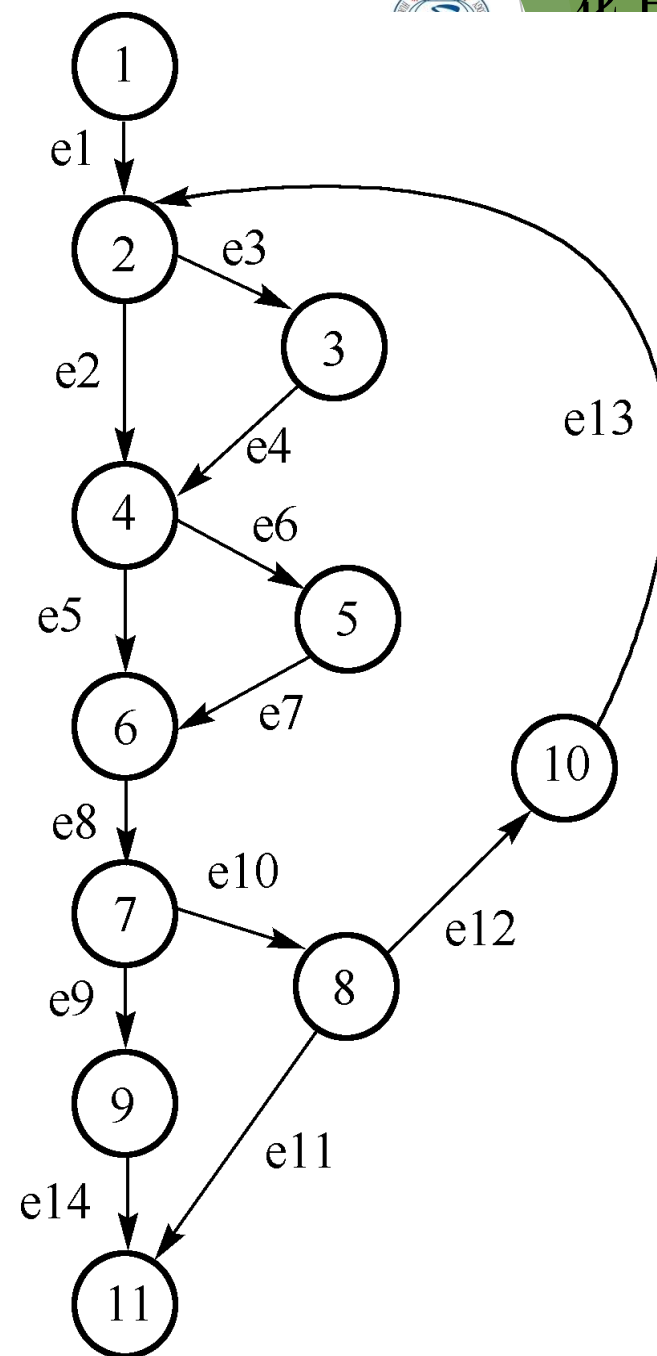
方案1：基于独立路径构建典型场景

► 针对程序图构建最少独立路径集合的问题



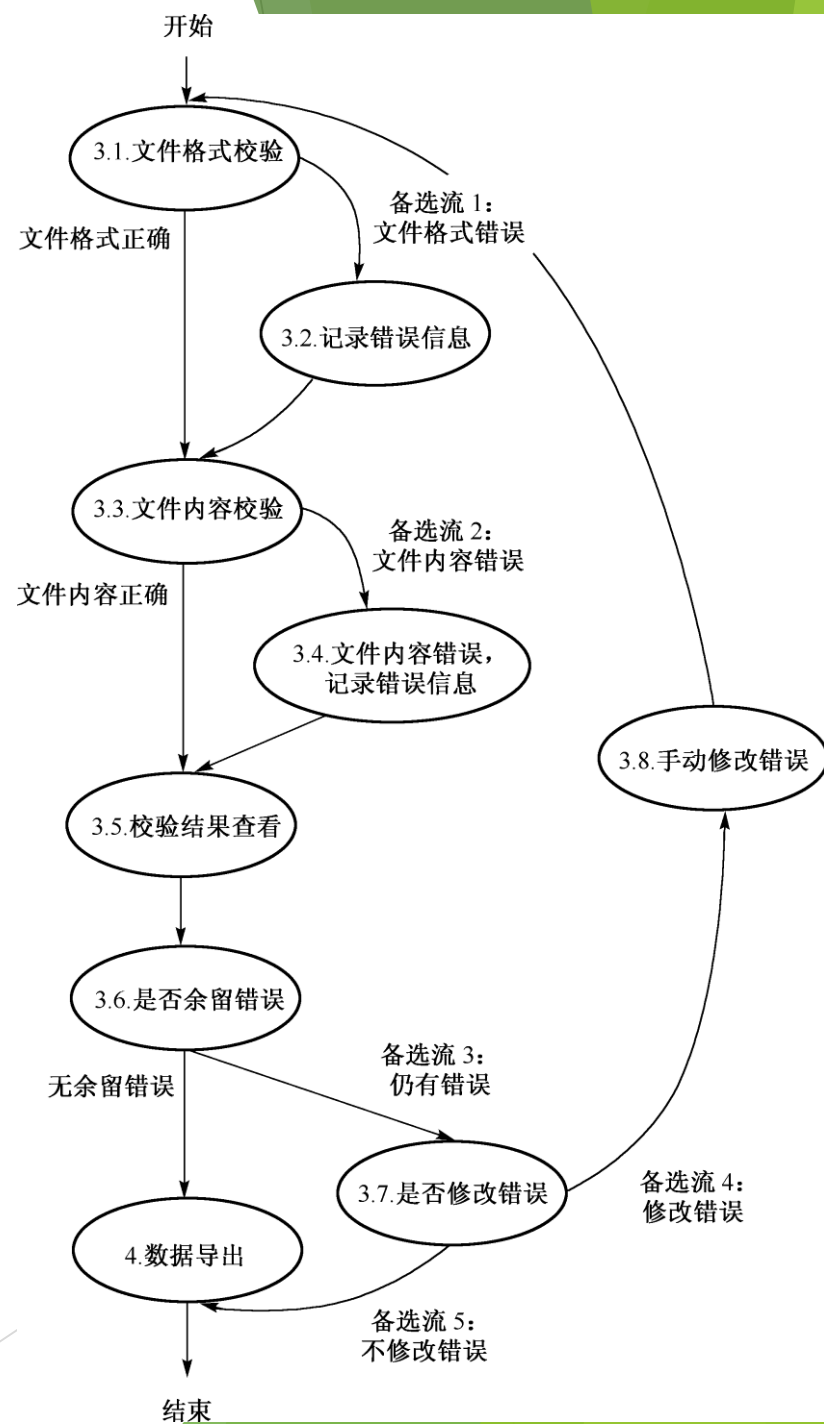
方案1：基于独立路径构建典型场景

- ▶ 路径1：1,2,3,4,5,6,7,8,11
- ▶ 路径2：1,2,4,5,6,7,8,11
- ▶ 路径3：1,2,3,4,6,7,8,11
- ▶ 路径4：1,2,3,4,5,6,7,9,11
- ▶ 路径5：1,2,3,4,5,6,7,8,10,2,4,6,7,9,11



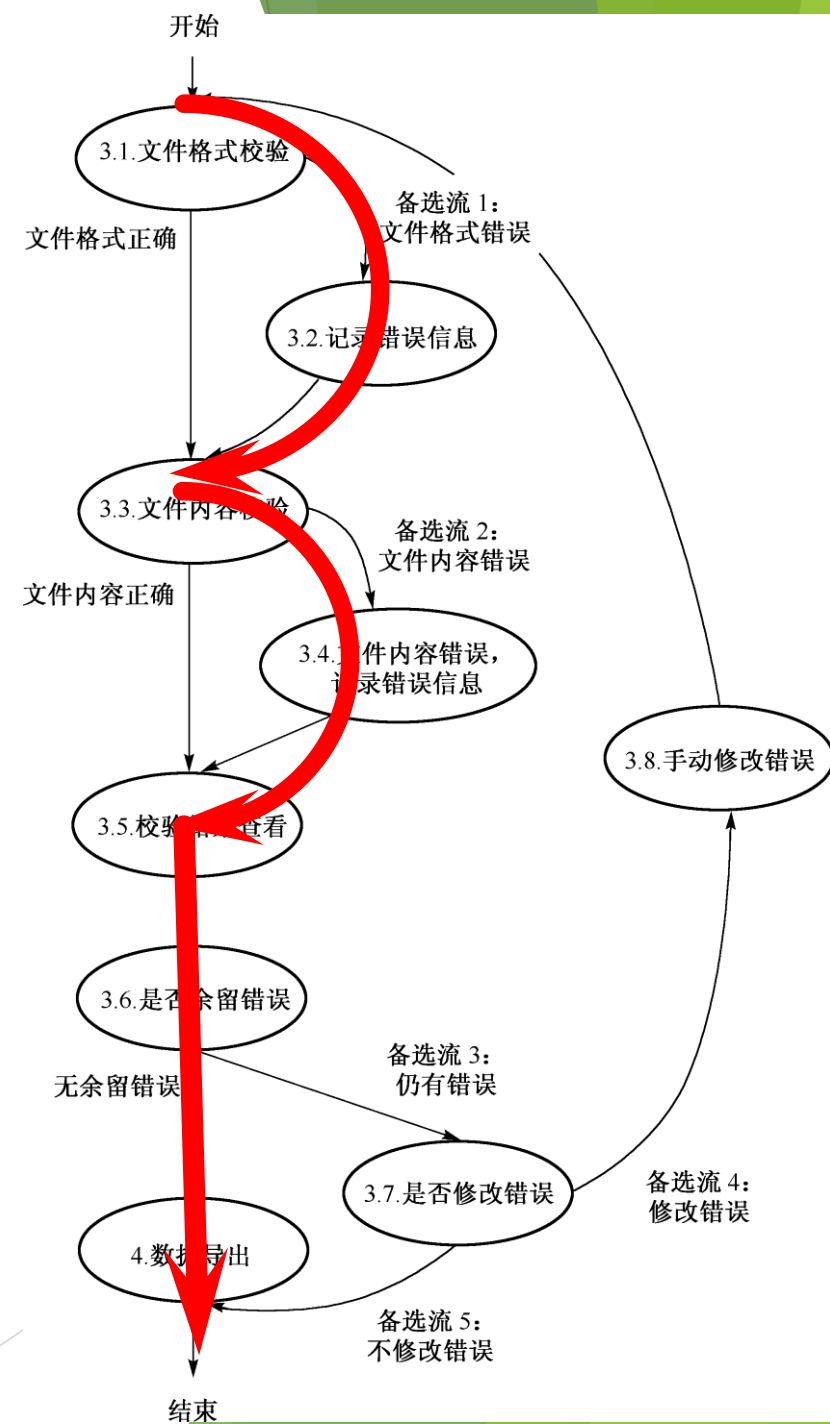
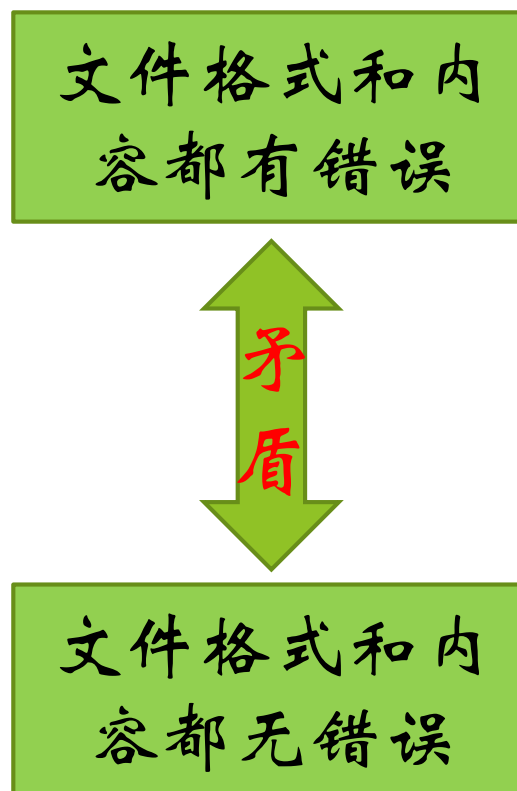
方案1：基于独立路径 构建典型场景

- 场景1：基本流+备选流1+备选流2+备选流3+备选流5
- 场景2：基本流+备选流2+ +备选流3+备选流5
- 场景3：基本流+备选流1+备选流3+备选流5
- 场景4：基本流+备选流1+备选流2
- 场景5：基本流+备选流1+备选流2+备选流3+备选流4



方案1：基于独立路径构建典型场景

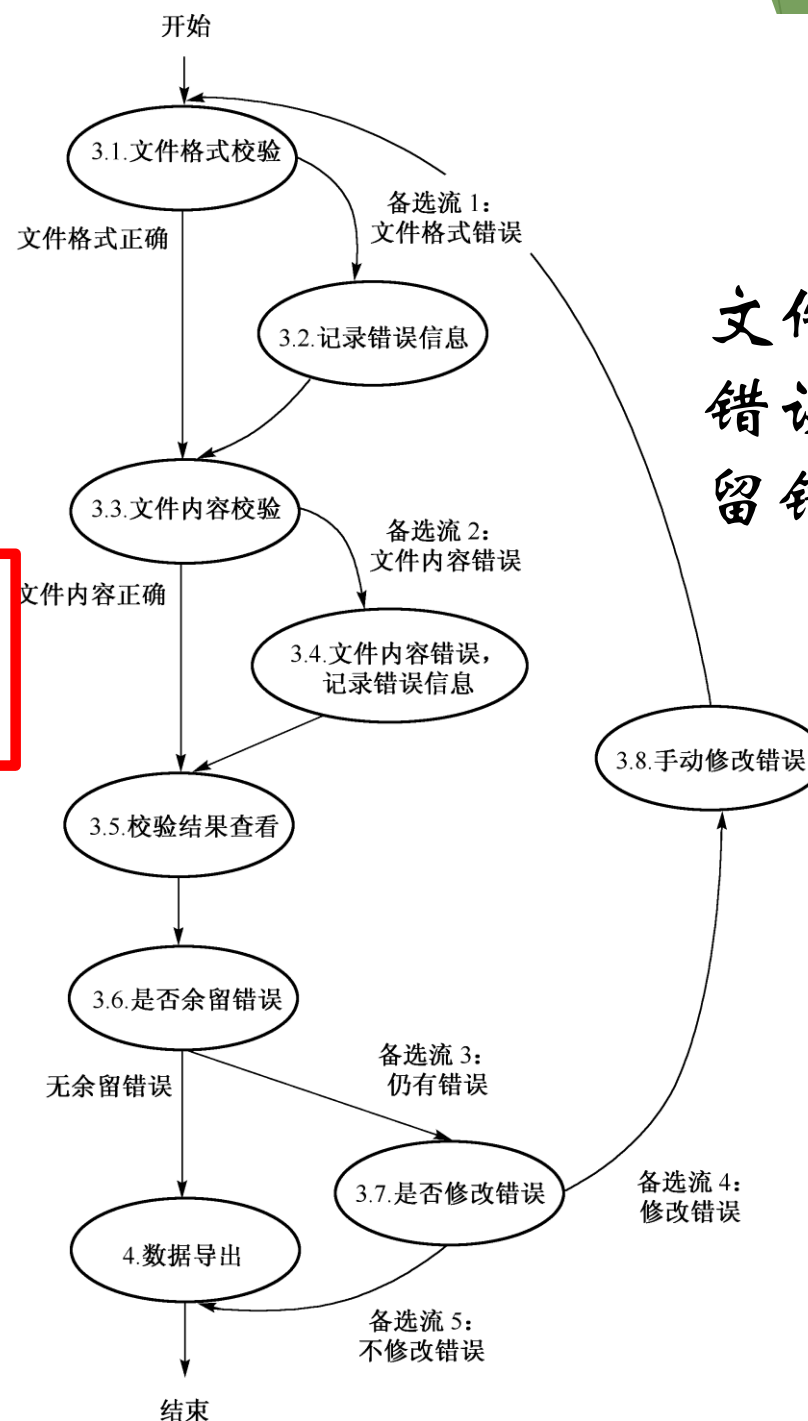
- 场景4：基本流+备选流1+备选流2
- 不可行场景
- 独立路径测试固有的局限性



方案2：基本事件流的个数构建典型场景

- 场景1：基本流
- 场景2：基本流+备选流1
- 场景3：基本流+备选流2
- 场景4：基本流+备选流3+备选流4
- 场景5：基本流+备选流3+备选流5

不可行场景

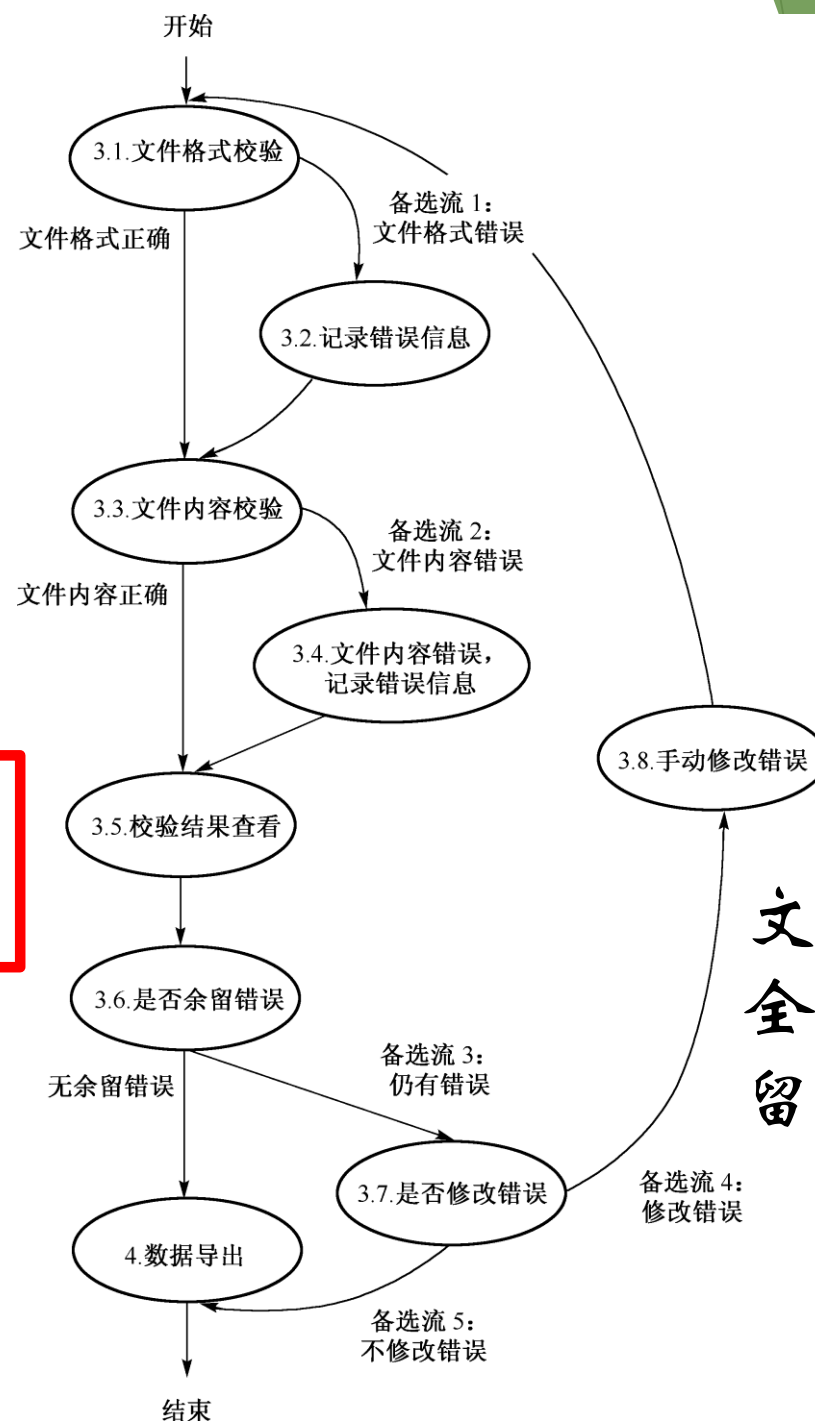


文件格式或内容有错误，则不存在余留错误

方案2：基本事件流的个数构建典型场景

- 场景1：基本流
- 场景2：基本流+备选流1
- 场景3：基本流+备选流2
- 场景4：基本流+备选流3+备选流4
- 场景5：基本流+备选流3+备选流5

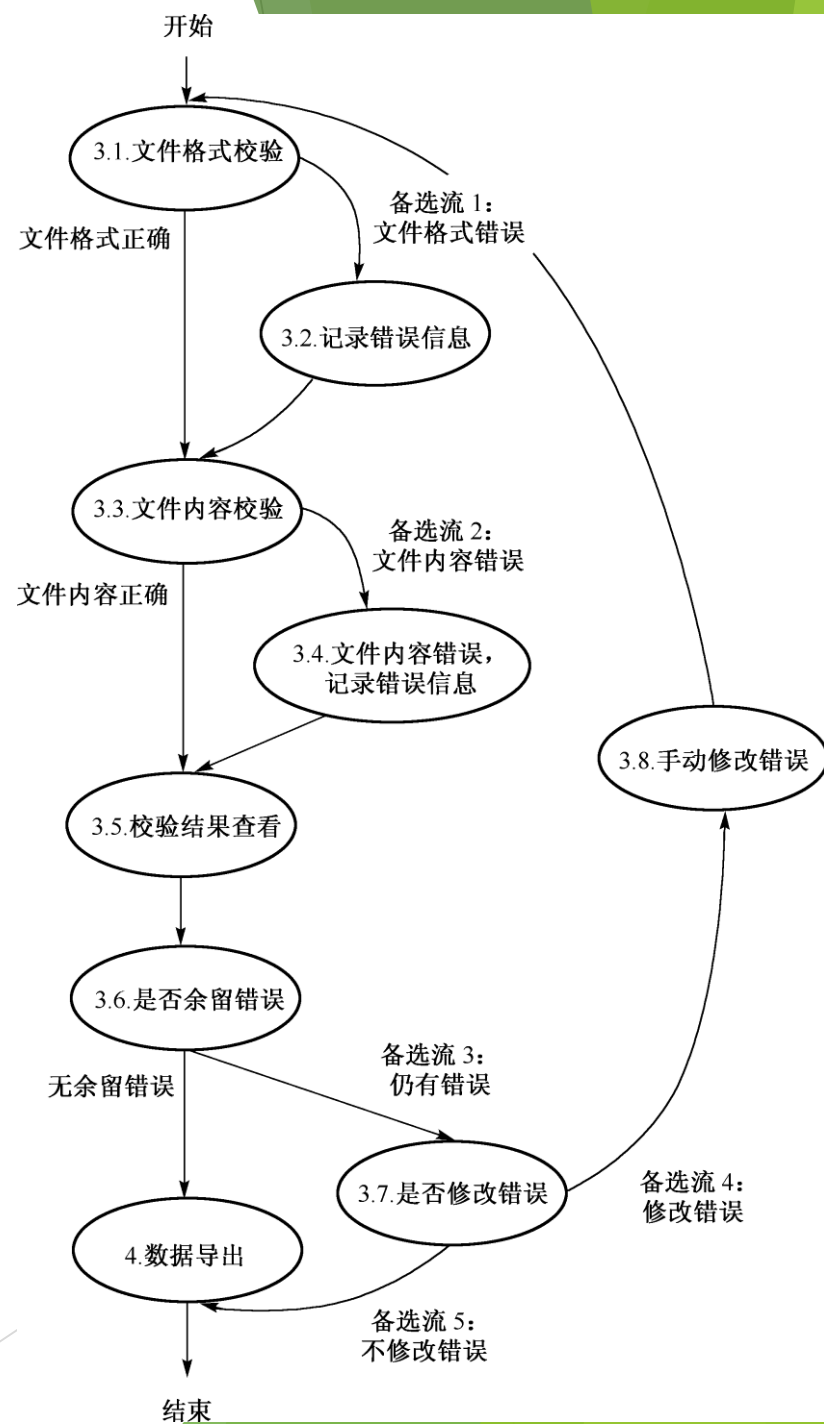
不可行场景



文件格式和内容完全正确，则存在余留错误

方案3：基于需求 构建典型场景

- 场景1：基本流
- 场景2：基本流+备选流1+备选流3+备选流5
- 场景3：基本流+备选流2+备选流3+备选流5
- 场景4：基本流+备选流1+备选流3+备选流4
- 场景5：基本流+备选流2+备选流3+备选流4
- 场景6：基本流+备选流1+备选流2+备选流3+备选流5
- 场景7：基本流+备选流1+备选流2+备选流3+备选流4



方案的比较

- ▶ 方案1：基于独立路径
 - ▶ 保证场景独立，无漏洞；但可能有不可行场景
- ▶ 方案2：基于事件流的个数
 - ▶ 易于缺陷定位；但可能有不可行场景，不保证场景独立和完备
- ▶ 方案3：基于需求
 - ▶ 保证场景可行；但不保证场景独立和完备



建议

- ▶ 事件流之间相互独立，可采用方案1：基于独立路径
- ▶ 事件之间存在关联时，先基于独立路径设计场景，然后基于需求补充场景