1. **PL—Programming Languages**

理论：语言设计，类型系统，语义和逻辑

环境：编译器，运行系统

应用：程序分析，程序验证，程序合成

技术：抽象解释（Abstract interpretation），数据流分析（Data-flow analysis, ），Hoare logic，Model checking，Symbolic execution等等

**静态分析作用**：程序可靠性、程序安全性、编译优化、程序理解（调用关系、类型识别）。

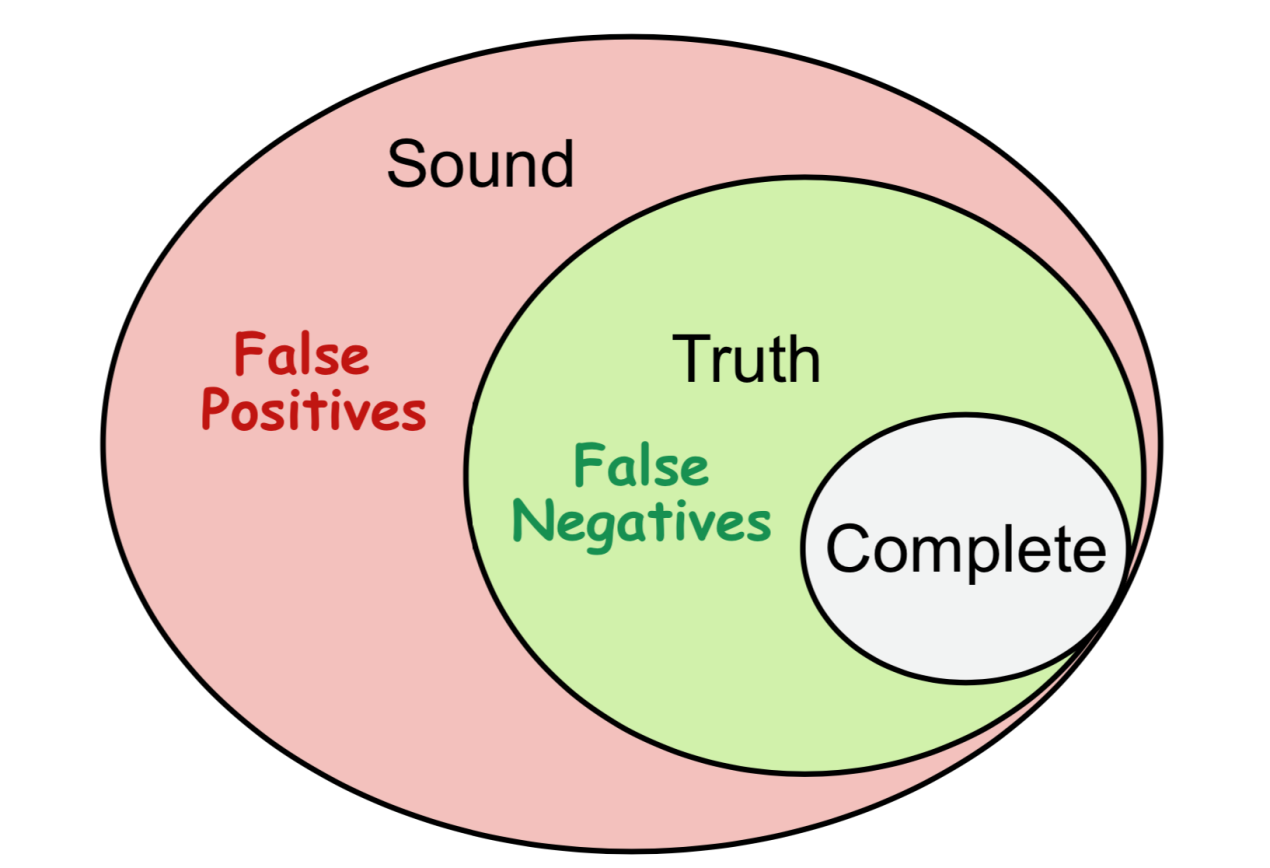
1. **soundness与completeness**

soundness：对程序进行了over-approximate过拟合，不会漏报（有false positives误报）。

completeness：对程序进行了under-approximate欠拟合，不会误报（有false negatives漏报）。

很多重要领域如军工、航天领域，我们追求的是soundness，但是要平衡精度和速度。那么我们绝大多数软件分析方法都做到了completeness，那么只要能证明满足soundness，那么该分析方法就是正确的。

那么什么样的SA是完美的呢？定义是既overapproximate又underapproximate的SA是完美的。overapproximate也叫sound，underapproximate也叫complete，他们之间的关系可以用一个图很好的表示



complete表示报告包含的错误都是真实的错误，但可能并未包含全部的错误，造成了漏报；sound表示报告包含了所有的真实错误，但可能包含了误报的错误，导致误报。

### 数据流分析步骤

abstraction：抽象值，定义集合（类别）

Safe-approximation安全近似

Transfer Functions：对抽象值的操作，转换规则

Control flows