调研报告——Doop

MF21330068 沈天琪

Doop是一个针对Java程序的指针分析框架，在实现中使用了Java、Datalog、Groovy三种编程语言。这个框架提供了命令行与Java API两种操作接口。

在Doop框架中，分析的大致流程是：首先从分析的Java程序出发，利用soot工具从字节码中抽取出待分析程序中所需要的静态分析信息；此外，根据程序所接受的不同分析选项，程序将自动化的从模块化的datalog规则中构造出对应分析选项的规则；最终，分析规则与分析输入将被交给具体执行运算的datalog引擎，datalog引擎将具体的结果计算出来并最终保存下来。

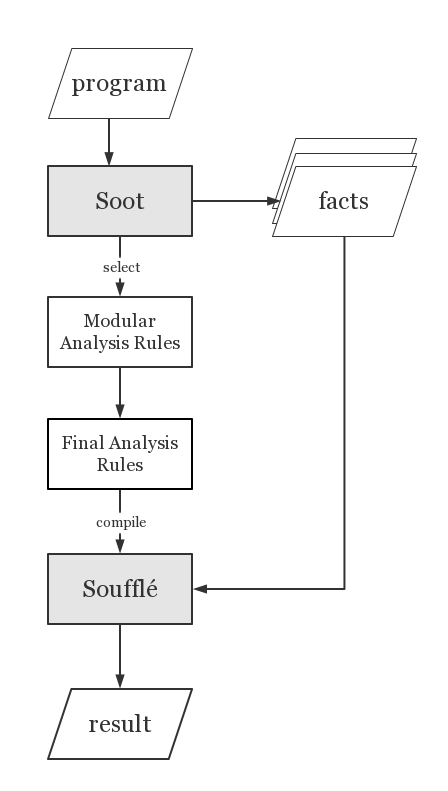
另外从细节上讲，Doop框架实际上还适配了许多其他已有的静态分析工具的结果——譬如Tamiflex工具的结果也可以作为Doop的输入以进一步增强静态分析工具的精度。

Doop自己的定义是一个软件框架。通常而言，软件框架是指一种抽象，就是说，这个框架可以根据第三方代码的更改来实现有选择的具体功能的更改。换句话说，它提供了一种构建和部署应用程序的标准方法，并且是一种通用的、可重用的软件环境。

一般而言，从软件工程角度而言，框架与普通的程序类库的区别体现在如下几点：控制反转、默认行为、可拓展性、不可修改的框架代码。

Doop作为一个静态分析框架，他要解决的任务就是对静态分析中大量的底层实现细节做封装，为了达到这个目标，Doop框架实际上是这样做的：它针对Java程序，将现有的指针分析以及信息流分析以及污点分析之类的算法通过datalog语言进行实现，而将datalog程序的适配交由自己的框架代码负责。

以下我们更加详细的来说明这一点，首先从框架代码开始。在这里我们通过一个具体程序运行的流程来更加形象的展示这些内容。



正如上图所示的，

