

# VYPE - abstract

## 12.5 Context-Insensitive Interprocedural Analysis

This chapter explains how points-to analysis can be used to compute a precise call graph, which is useful in computing precise points-to results. There is also described on-the-fly call-graph discovery and explained how can be Datalog used to describe the analysis succinctly.

First chapter (12.5.1 Effects of a Method Invocation) contains information about how can be effects of a method call computed in Java programming language. Computation process is composed of determining the type of the receiver object, next the formal parameters are assigned to the objects pointed to by the actual parameters and then the returned object is assigned to the left-hand-side variable of the assignment statement. Determining the type of the receiver object can be a problem. This problem is discussed at the end of this chapter.

Next chapter (12.5.2 Call Graph Discovery in Datalog) describes how call graph discovery works in Datalog. It introduces three EDB predicates, each of which is obtainable easily from the source code.

Last chapter (12.5.3 Dynamic Loading and Reflection) describes that in languages like Java, which allows dynamic loading of classes, it is impossible to analyze all the possible code. Next problem discussed in this chapter is reflection, which does not determine the type of object dynamically.

## 12.6 Context-Sensitive Pointer Analysis

There is discussed a cloning-based context-sensitive analysis in this section, which simply clones the methods, one for each context of interest. The discussion of context sensitivity is separated into two parts - how to handle context sensitivity logically and how to represent the exponentially many contexts.

First chapter (12.6.1 Contexts and Call Strings) describes how to create a cloned call graph. Next it defines context, call string and relationship between them. Chapter also discusses various ways how we can limit the number of distinct contexts because if there are recursive functions in the program, then the number of possible call strings is infinite.

Second chapter (12.6.2 Adding Context to Datalog Rules) contains information how to add context to Datalog rules. It also contains example of Datalog program for context-sensitive points-to analysis.

Final chapter (12.6.3 Additional Observations About Sensitivity) describes how to handle the largest of Java applications with context sensitivity considering object-factory idiom and object sensitivity.