Dimensions of Precision in Reference Analysis of Object-Oriented Programming Languages

In object-oriented programming languages, the program representation is dependent on the analysis solution, but the reverse is also true. This interdependent relationship makes the analysis quite different of the analysis of procedural languages.

The dimensions of reference analysis lead to variations in the precision of the solution that can be obtained using different methods. Examining these dimensions will illustrate similarities and differences between analyses, and identify sources of precision and tradeoffs in cost.

Class analysis involves calculating the set of classes (i.e., types) associated with the objects to which a reference variable can refer during program execution. One way to think about this, is a reference analysis in which one abstract object represents all the instantiations of a class.

Points-to analyses are often designed as extensions to earlier pointer analysis for C.

Refers-to analysis is a term sometimes used to distinguish a points-to analysis for object-oriented languages from a points-to analysis for general-purpose pointers in C.

A lot of other analyses require reference analysis in order to obtain a representation of interprocedural flow for a program. Thus, reference analyses are crucial to any analysis of object-oriented code.

The following dimensions affect reference analysis precision:

* Flow sensitivity: If an analysis considers the order of execution of statements in a program.
* Context sensitivity: If an analysis distinguishes between different calling contexts of a method.
* Program representation: There is an interdependence between possible program calling structure and reference analysis solutions in object-oriented languages. Because of this, a simple analysis can obtain an approximation, or the call structure can be calculated lazily, on-the-fly.
* Object representation: Sometimes one abstract object is used to represent all instantiations of a class, and sometimes a representative of each creations site (e.g., new) is used to represent all objects created at that site. Other schemes exist, but are less used.
* Field sensitivity: An object may have its fields represented distinctly in the solution, this is called a field-sensitive analysis.
* Reference representation:
* Directionality: