

Increasing the Accuracy of Shape and Safety Analysis of Pointer-based Codes

Pedro Diniz

To appear at the *16th Workshop on Languages and Compilers for Parallel
Computing (LCPC03)*, College Station, TX, 2-4 October 2003

Abstract

Analyses and transformations of programs that manipulate pointer-based data structures rely on understanding the topological relationships between the nodes i.e., the overall *shape* of the data structures. Current static *shape* analyses either assume correctness of the code or trade-off accuracy for analysis performance, leading in most cases to shape information that is of little use for practical purposes.

This paper introduces four novel analysis techniques, namely *structural fields*, *scan loops*, *assumed/verified shape properties* and *context tracing*. Analysis of *structural fields* allows compilers to uncover node configurations that play key roles in the data structure. Analysis of *scan* loops allows compilers to establish accurate relationship between variables that traverse the data structure. *Assumed/verified* property analysis derives sufficient shape properties that guarantee termination of loops. These properties must be verified during shape analysis for consistency. *Context tracing* allows the compiler to isolate data structure nodes by evaluation of conditional statements and hence preserve shape accuracy.

We believe that future static shape and safety analysis algorithms will have to include some if not all of these techniques to attain a high level accuracy. In this paper we illustrate the application of the proposed techniques to codes that build (correctly as well as incorrectly) sophisticated data structures that are beyond the reach of current approaches.