要 旨

C プログラムを対象としたシェープ解析アルゴリズムの実装

水野 雄介

近年, ソフトウェアが複雑化するにつれて, プログラムの実行効率や信頼性を向上させるためのプログラム解析技術がますます重要になってきている. 重要な静的解析技術のひとつとしてポインタ解析がある. ポインタ解析とは, プログラム内の変数が実行時に指す可能性のあるメモリ領域を求める静的解析である. ポインタの指す先に関する情報はメモリ参照式間の依存関係を正確に解析する事で求められる. しかしポインタ解析では通常, リストのような再帰的なデータ構造を一つの領域として解析することから解析精度が低下してしまう. そこで, ポインタ解析よりも精度の高い解析手法の一つとしてシェープ解析が提案されている.

シェープ解析とは, 述語抽象を行う事によりプログラム内のヒープ割り当て構造を求める静的解析である. シェープ解析では, ポインタ解析では行わない再帰的なデータ構造の解析を行う事ができる.

シェープ解析の効率化や解析対象の拡張に関する研究が近年行われているが, 公開されている入手容易な実装がなく, 利用者が容易に解析を行う事ができない. そこで, 本研究では COINS コンパイラ・インフラストラクチャに対して, Reps が提案しているシェープ解析法を実装する. その結果, ポインタ解析では正確な解析結果を得られない連結リストを含むアルゴリズムを正確に解析することができた.

キーワード　ポインタ解析, シェープ解析, COINS, 再帰的データ構造

Abstract

An implementation of a shape analysis algorithm for C programs

Yuusuke MIZUNO

In recent years, the program-analysis technologies for improving the efficiency and reliability of a program is becoming stil more important as software is complicated. The points-to analysis is one of the important static analysis technologies. The points-to analysis statically computes the memory areas to which a variable in a program may point during the execution of that program. That computation can be achieved by analyzing the dependency between memory references correctly. However, in the points-to analysis, a recursive data strucutre like a list is usually abstracted as one element, and it sometimes causes low analysis accuracy. To solve this problem, the shape analysis is proposed as one of the analysis techniques with higher-precision than points-to analysis.

The shape analysis statically analyses heap-allocated structures in a program. It can analyse recursive data structures that cannot be analysed by the points-to analysis.

Although researches on the extension and improvement of the shape analysis have been performed in recent years, there is no freely and easily accessible implementation of the shape analysis.

Therefore, in this study, we implement on the COINS compiler infrastructure a simple shape analysis algorithm that Reps has proposed. As a result, we could correctly analysed an algorithm using a linked list that could not be analysed by the points-to

analysis.

key words Points-to analysis, Shape analysis, COINS, Recursive data structure