
発表概要

AspectJ プログラムの Points-to 解析

徳 門 敦[†] 趙 建 軍[†]

現在、ロギングなどの複数の箇所にわたる処理の分散を「アスペクト」という単一のモジュールにまとめることができるアスペクト指向プログラミングが注目されてきている。アスペクトによる横断処理のモジュール化は、プログラムの保守性、再利用性を高めることができ、その有用性も示されつつある。AspectJ はオブジェクト指向言語である Java を拡張した言語であり、アスペクト指向プログラミングのモジュール機能「アスペクト」を利用することができる。points-to 解析は、プログラムを静的に解析し、参照変数、またはフィールドとそれによって指し示されるオブジェクトのセットを決定する。解析で得られるセット情報はコンパイルの最適化やソフトウェア工学ツールに対して様々な利点を持つ。AspectJ では、クラスとアスペクトを織り込むことで、アスペクト側を動作させることができ、Points-to 解析はこの両方にわたって行われるべきである。この場合、解析の対象となるクラスから織り込まれるアスペクトに向けて解析処理の移行が必要となるが、言語の性質上、クラス側にアスペクトへの動作移行の目安となるものは明示的に記述されおらず、Java に対しての Points-to 解析手法と同様な解析時における移行目安の検出からの処理移行は困難である。本発表では、AspectJ に対して、クラス-アスペクト間の関係情報を事前構築し、その情報の参照に基づいて、従来の Java の points-to 解析手法から AspectJ への拡張を提案する。

Points-to Analysis for AspectJ Programs

ATSUSHI TOKUMON[†] and JIANJUN ZHAO[†]

Points-to analysis is a fundamental analysis technique whose results are useful in compiler optimization and software engineering tools. Although many points-to analysis algorithms have been proposed for procedural and object-oriented languages such as C and Java, there is no points-to analysis algorithm for aspect-oriented languages until now. In this presentation, we propose a flow- and context-insensitive points-to analysis algorithm for AspectJ programs. Our algorithm is based on the points-to analysis algorithm for Java.

(平成 16 年 5 月 19 日発表)

[†] 福岡工業大学情報工学部

Department of Computer Science and Engineering,
Fukuoka Institute of Technology