

発表概要

パターンマッチに基づくバイトコード変換とその型安全性

川上 祐介[†] 鎌田 十三郎^{††}

プログラムの高速化，機能追加を行うには，プログラム構造に応じた横断的な変更が必要になることがある。我々は，このような拡張操作に必要なコストを軽減するため，バイトコード変換ツール Cusapi を提案している。Cusapi では，バイトコード変換を行う変換モジュールを作成，通常のライブラリのように配布し，ユーザプログラムに対して変換を適用することができる。また，変換モジュール作成者は，パターンマッチによりプログラム構造の特徴を抽出することで，対象プログラムに応じたコード変換を記述できる。ただし，このような変換を行うためには，変換後プログラムの整合性，たとえば変換により追加されたメソッドと，変換前から存在するメソッドのシグネチャが重複する可能性などを考慮しなければならない。変換モジュールを一般ライブラリのように利用するためには，任意のプログラムに対して，安全に変換を適用できることを事前に保証すべきである。本発表では，変換モジュールの安全性として，変換後プログラムが Java 言語仕様違反を犯さないことを，事前に保証するための型システムを提案する。本システムは，パターンとして宣言的に記述された対象プログラムの特徴のみを前提として，任意の変換後プログラムが言語仕様を満たすか否かを判定する。この前提には，変換対象となるプログラムが言語仕様を満たす，つまり型整合性を持ち，メソッド型の重複なども存在しないという性質も含む。変換後プログラムについても同様の性質が導出可能であれば，変換は任意のプログラムに安全に適用可能であると判定される。

Bytecode Translation Based on Declarative Pattern Matching and Its Type-safeness

YUSUKE KAWAKAMI[†] and TOMIO KAMADA^{††}

When programmers customize their programs to optimize or to attach new features, there are situations where the programmers are forced to modify multiple code fragments spread over the programs with adjusting to the program structures. To reduce the cost of manual customizations, we have proposed a bytecode translation tool, Cusapi, which provides an environment to develop 'bytecode translation modules' and to distribute them as libraries that customize users' programs. Our tool features declarative descriptions, where module developers use pattern match syntax to acquire program structures, and apply code modifications based on the structure information. However, this kind of program modification may cause some trouble, such as conflicts of method signatures among newly created or original methods. To distribute a translation module as a type-safe library, we have to confirm that the module can be applied to any program safety. This presentation proposes a type system that confirms type-safeness of translation modules. To judge whether any translated program will satisfy Java language specification, our system assumes only characteristics of target programs from pattern match declarations. This assumption also implies that the target programs satisfy the language specification and do not cause conflicts between method signatures or variable names. When it is derived from this assumption that translated programs also satisfied the language specification, our system judge that the translation module is type-safe, and can be applied to any programs.

(平成 17 年 3 月 18 日発表)

[†] 神戸大学大学院自然科学研究科情報知能工学専攻
Graduate School of Science and Technology, Kobe University

^{††} 神戸大学工学部情報知能工学科
Department of Computer and Systems Engineering,
Faculty of Engineering, Kobe University