

発表概要

高水準動的型付け言語間 FFI における Type Description Helper の構築

池崎 翔哉^{1,a)} 山崎 徹郎^{1,b)} 千葉 滋^{1,c)}

2020年3月13日発表

他言語インタフェース (FFI) はある言語から他の言語のライブラリを使用するための仕組みである。プログラミング言語が実用的であるために、すでに他言語で書かれたライブラリ資産を使用可能であることが重要である。近年では豊富な型を有する動的型付け言語にライブラリ資産が溜まってきているため、このような言語との FFI が望まれている。しかし型が豊富であるがゆえに、ホスト言語とライブラリ言語間の型変換規則が複雑なものとなり型変換規則の記述量が多くなってしまいう問題がある。本発表は Type Description Helper という型変換規則の半自動的な導出器を提案する。この Type Description Helper を用いることでユーザが記述する必要のある型変換規則の量をおさえることができる。Type Description Helper による導出は不完全であるため、一部の導出できない型変換規則を手動で記述する必要がある。Type Description Helper による型変換規則の導出には Log-based 法と Type-inference-based 法の 2 種類の方法がある。Log-based 法は他言語関数呼び出し時のログから動的に型変換規則を導出する。Type-inference-based 法はソースコードを静的に解析することで型変換規則を導出する。また、この静的解析によって明らかに適用不可能な型の引数を検出することも可能である。本発表では具体例として Python および Euslisp の 2 言語を選定し、この間の FFI を作成するとともに Type Description Helper の実装を行い、その有用性を確認した。

Presentation Abstract

Building Type Description Helper in FFI between High-level Dynamically Typed Languages

SHOYA IKEZAKI^{1,a)} TETSUROU YAMAZAKI^{1,b)} SHIGERU CHIBA^{1,c)}

Presented: March 13, 2020

A foreign function interface (FFI) is a mechanism that enables a programming language to use libraries written in another foreign language. The FFI is important since the language that does not have access to the rich libraries that already exist is not considered practical. Recently, the FFI between dynamically typed languages that have various types is required because a great number of useful libraries are written in those languages. However, it takes a high cost to describe the rule of type conversion between host and library languages since the rule of it is complex on account of type-richness. This presentation proposes Type Description Helper that derives the rule of type conversion semi-automatically. The Type Description Helper reduces the amount of the rule of type conversion that users have to write. As the rule of type conversion that is derived by Type Description Helper is incomplete, users have to write the rule of type conversion that has not yet derived. Type Description Helper derives the rule of type conversion in two ways: log-based approach and type-inference-based approach. The log-based approach derives the rule of type conversion dynamically from the log of foreign function calls. The type-inference-based approach derives the rule of type conversion by analyzing the source code statically. At the same time, it can detect some invalid arguments that cannot be applied by foreign function. In this presentation, we choose Python and Euslisp as an example and implement FFI and Type Description Helper, then checked that it is useful.

This is the abstract of an unrefereed presentation, and it should not preclude subsequent publication.

¹ 東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology,
The University of Tokyo, Bunkyo, Tokyo 113-8656, Japan

a) ikezaki@csg.ci.i.u-tokyo.ac.jp

b) yamazaki@csg.ci.i.u-tokyo.ac.jp

c) chiba@acm.org