

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<sup>1,a)</sup> 山崎 徹郎<sup>1,b)</sup> 千葉 滋<sup>1,2,c)</sup>

2021年3月16日発表

他言語インタフェース (FFI) はある言語から他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の実用上、他言語で書かれたライブラリ資産を使用でき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。しかし型の豊富さゆえに、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。本発表では 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。これが導出した型変換規則の分、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 による型変換規則の導出手法として2つの方法を考案した。ログベース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し、型推論ベース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。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<sup>1,a)</sup> TETSUROU YAMAZAKI<sup>1,b)</sup> SHIGERU CHIBA<sup>1,2,c)</sup>

Presented: March 16, 2021

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 type conversion rules between host and library languages since the rules of it are complex on account of type-richness. We propose a Type Description Helper that derives the type conversion rules semi-automatically. The Type Description Helper reduces the number of type conversion rules that users have to write. As the type conversion rules that are derived by Type Description Helper are incomplete, users have to write the type conversion rules that have not yet been derived. Type Description Helper derives the type conversion rules in two ways: log-based approach and type-inference-based approach. The log-based approach derives the type conversion rules dynamically from the log of foreign function calls. The type-inference-based approach derives the type conversion rules by analyzing the source code statically. At the same time, it can detect some invalid arguments that cannot be applied by foreign functions. In this research, 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.

<sup>1</sup> 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, Tokyo 113-8656, Japan

<sup>2</sup> 情報処理学会プログラミング研究会  
Information Processing Society of Japan Special Interest  
Group on Programming, Chiyoda, Tokyo 101-0062, Japan

---

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

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

c) chiba@acm.org