

手続き型言語からの Haskell コード変換

竹岡 尚三¹ 松村 哲郎² 関口 渚² 倉光 君郎^{2,3}

概要：関数型言語は、並列化がしやすい等様々なメリットをもっている。多くのプログラマが親しんでいる手続き的な記述から、関数型の記述へ変換することができれば、手軽に関数型言語のメリットを享受することができ有用であると考えられる。我々は、コード生成器を切り替えることのできるスクリプト言語 GreenTeaScript の処理系に対して、関数型言語である Haskell のソースコードを生成するコード生成モジュールを実装し、手続き型言語である GreenTeaScript から関数型言語への変換を可能にした。本稿では、GreenTeaScript から Haskell への変換のコンセプトを述べる。

キーワード：スクリプト言語、関数型言語、コード変換、Haskell

Haskell Code Translation from Procedural Language

SHOZO TAKEOKA¹ TETSURO MATSUMURA² NAGISA SEKIGUCHI² KIMIO KURAMITSU^{2,3}

Abstract: Functional languages have the advantage of a easier parallelization. If procedural language codes can be converted to Functional languages, procedural language programmers can enjoy the benefits of functional languages easily. So the conversion is believed to be useful. We implement the code generation module that generates source code of Haskell, a functional language, from a procedural language GreenTeaScript. GreenTeaScript is capable of switching a code generator, and our Haskell generator is developed as one of code generation module. In this paper, We introduce the concept of conversion to Haskell from GreenTeaScript.

Keywords: Script Language, Functional Programing, Code Translation, Haskell

1. はじめに

GreenTeaScript^{*1}は、マルチプラットフォーム開発で利用されることを目指してオープンソースで開発が進められている、オブジェクト指向静的型付けスクリプト言語である。GreenTeaScript では、コード生成器がモジュール化されており、現在、GreenTeaScript コードから JavaScript, Python, Bash, Java, C# のソースコードを生成するモジュールが存在する。これらの言語は手続き型言語であったが、我々は、新たに GreenTeaScript コードを関数型言語である Haskell のソースコードへ変換するモジュールを

開発した。

本論文の構成は次の通りである。2章ではスクリプト言語 GreenTeaScript について説明する。3章で実際の変換例を紹介する。4章で関連研究、5章で全体をまとめる。

2. GreenTeaScript

本章では、GreenTeaScript のアーキテクチャについて述べる。図 1 に示すように GreenTeaScript で記述されたコードは、パーサによって抽象構文木 (AST) に変換される。抽象構文木は、デフォルトではコード生成器によってバイトコード (Java Byte コード) に変換され、実行される。GreenTeaScript のコード生成器はモジュール化されているため、すでに述べた通り、他の言語のソースコードを生成するモジュールに切り替えることができる。

3. 変換例

本章では、GreenTeaScript から Haskell への変換例のう

¹ 株式会社アックス
AXE, Inc.

² 横浜国立大学
Yokohama National University

³ 科学技術振興機構/CREST
Japan Science and Technology Agency/CREST

*1 <https://github.com/GreenTeaScript/GreenTeaScript>

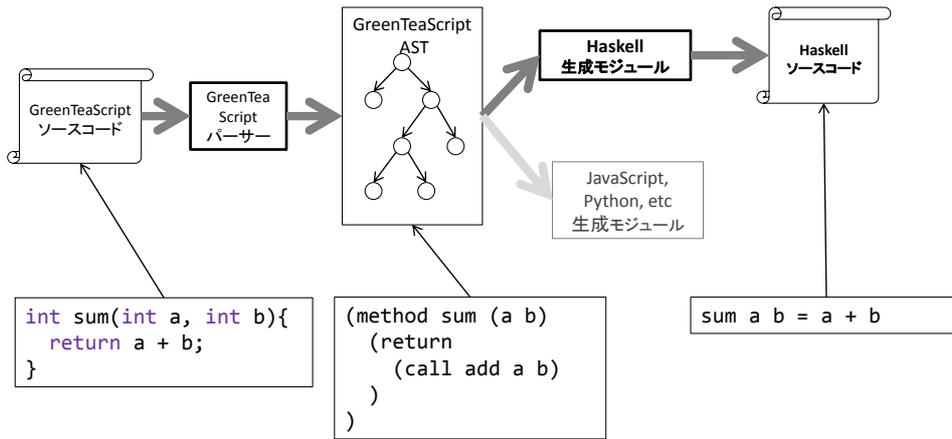


図 1 GreenTeaScript のアーキテクチャ
Fig. 1 GreenTeaScript Architecture

ち、代表的なものを紹介する。変換例は次の書式で表す。

(変換前のソースコード (GreenTeaScript))
(変換後のソースコード (Haskell))

3.1 四則演算

四則演算は、整数型と実数型（浮動小数点数型）で分けている。Haskell では演算子は中置記法で書くことができるため、四則演算子は基本的にそのまま利用できるが、整数型の場合は、除算と剰余の演算子をそれぞれ 'div' と 'mod' に変換する。

$$\frac{+ \ - \ / \ * \ \%}{+ \ - \ 'div' \ * \ 'mod'}$$

実数型の場合は同じ四則演算子を利用できるので特別な変換は必要ない。ただし GreenTeaScript, Haskell 共に実数型には剰余演算が定義されていない。

3.2 関数定義

ここでは紙面の都合上簡単な例のみ示す。return 文のみの単に値を返す関数は以下のように変換できる。

$$\frac{\text{int sum(int a, int b) \{ return a + b; \}}}{\text{sum a b = a + b}}$$

3.3 制御構造

GreenTeaScript の手続き的な制御構造は、モナドを利用するコードに変換していく。

$$\frac{\text{for(a = 0; a < 10; a++) \{ ... \}}}{\text{forM [0..10-1] \$ a -> do ...}}$$

4. 関連研究

Kelsey[1] や森下 [2] は静的単一代入 (SSA) 形式を経由して手続き型プログラムと関数型プログラムを相互に変換する手法を発表している。新妻 [3] は本手法とは逆に、実行の高速化を目的として関数型言語の Scheme から C++ へのコード変換を行っている。

5. まとめ

本研究では、スクリプト言語 GreenTeaScript の処理系上に関数型言語 Haskell のソースコード生成モジュールを実装し、手続き型言語である GreenTeaScript から関数型言語への変換を可能にした。

謝辞 本研究は、JST/CREST「実用化を目指した組込みシステム用ディペンダブル・オペレーティングシステム」領域の研究課題「実行時の安全性を確保する SecurityWeaver と P-SCRIPT」の一部として行われた。

参考文献

- [1] Kelsey, R. A.: A Correspondence Between Continuation Passing Style and Static Single Assignment Form, *SIG-PLAN Not.*, Vol. 30, No. 3, pp. 13-22 (online), DOI: 10.1145/202530.202532 (1995).
- [2] 森下大輔: 手続き型言語と関数型言語の相互変換手法, 全国大会講演論文集, Vol. 2013, No. 1, pp. 359-361 (オンライン), 入手先 (<http://ci.nii.ac.jp/naid/110009580383/>) (2013).
- [3] 新妻弘崇: Scheme から C++ への単純な変換, 第 96 回プログラミング研究会, PRO-2013-3, 情報処理学会・プログラミング研究会 (2013).