

# 要 旨

## 関数型プログラミング学習者のための プログラミング支援システム

佐藤功二

プログラマはより簡潔で誤りのないプログラムを作成するよう心がける必要がある。純粋な関数型プログラムはプログラム全体が関数で構成されており、その関数は副作用がなく参照透明である。そのため、構成されたプログラムは誤りが少なく、簡潔である。しかし、手続き型プログラミングに慣れてしまい、再代入を意識することで関数型プログラミングの思考に馴染めないプログラマも存在する。また、手続き型プログラミングでは馴染みの薄い高階関数や遅延評価も馴染めない要因として考えられる。そこで、再帰関数や高階関数のプログラミングを助けるシステムを提案することでプログラミングの際の負担を軽減できないかと考えた。

本研究では、学習者のソースコードを簡潔性やコード長、実行時間の観点から評価するシステムを提案した。さらに提案システムは、評価結果に基づいてソースコード中の再帰関数に対して高階関数を利用した定義に書き換えることでソースコードの品質を向上させる。プログラムの書き換えは、既存のリファクタリングツールを用いて行う。そして、コード長と簡潔性の観点からの評価機能の動作確認を行い、考察した。その結果、コード長の減少と可読性の向上効果が見られたが、リファクタリングツールの改良や評価箇所の拡大など改善点も多く見受けられた。

キーワード 関数型プログラミング, Haskell, HaRe, プログラミング支援

# Abstract

## A Programming Supporting System for Learners of Functional Programming

Koji Sato

Programmers should endeavor to make a concise and correct program. A functional program consists of only referentially transparent functions that have no side effect. Thanks to this property a functional program tends to be concise and have few faults. However, there are programmers who cannot be accustomed well with the idea of functional programming because they are accustomed with imperative programming, especially the idea of substitution. Providing a system for supporting a user to understand recursive higher-order functions, we can reduce the burden of programming.

In this study, we propose a programming support system that evaluates the user's source code from some viewpoints. Moreover, the proposed system improves the quality of source code by rewriting the definition of recursive functions using higher-order functions. The rewriting of a program is performed by using an existing refactoring tool. We evaluate a sample program from the viewpoint of code length. As a result, reduction in code length and improvement of readability were obtained. However, many problems on the refactoring tool and the evaluation method were also found.

**key words** Functional programming, Haskell, HaRe, Refactoring, Programming support