

B-12 関数型プログラミング学習のための高階関数の振る舞いの可視化

ビジュアルインタフェース研究室 志村 匠

1. 背景と目的

関数型プログラミング言語とは、同じ引数で呼び出されたときには常に同じ値を返す参照数学的な意味に近い関数の組み合わせでプログラムを記述する言語であり、変数の値の変更を原則として禁止することも特徴である。

関数型プログラミングの学習においては、既に命令型のプログラミング言語を習得している者でも、プログラムの処理の流れの理解が難しいと言われている。1行ずつ命令を実行するJavaやC言語などの命令型プログラミング言語と異なり、個々の関数が独立しており処理の流れを推測しづらいことが理由の1つである。

また、関数型プログラミングは、高階関数や遅延評価など、多くの独特の性質を持っていることも習得が難しい理由といわれている。高階関数 (higher-order function) とは、別の関数をその引数や戻り値として扱う関数のことであり、遅延評価 (lazy evaluation) とは、プログラム内で値が必要になった時に初めて式や関数が実際に評価される評価方法である。

そこで、本研究では、関数型プログラミングの初学者に対して、関数型プログラミングの特性や関数の状態を可視化することで、関数型プログラミングの学習効果の向上を図ることを目指した。

2. 関連研究

プログラミング学習では、プログラムを記述してその実行結果から処理の概念を学んでいく学習方法が用いられることが多い。岡本ら[1]は、C言語で書かれたプログラムの処理を可視化、識別容易化、予測可能化、分離化の4

つの方法で可視化する実験を行った。その結果、被験者の3分の1以上から学習効果があったことが確認された。

大城ら[2]は、プログラミング初学者を対象に、学習者が慣れていないプログラミング言語のプログラムが実行される様子と処理の構造を提示してから、学習者にそれに対応するソースコードを書かせる実験を提案した。

さらに、竹内ら[3]は、関数型プログラミング言語について空欄補充問題を自動生成し、回答に対して正誤判定を行う学習支援方法を提案した。それによれば、他の言語で同じ処理のプログラムを書いて比較することにより、学習効果の向上が期待できることが示唆された。

3. 本研究の提案

本研究は、関数型プログラミング言語の学習支援のために、関数型プログラミング言語で用いられる関数やリストを図式化し、アニメーションを利用して処理の流れを動的に可視化する学習システムを提案する。

その中でも、関数型プログラミングで頻繁に用いられる高階関数と遅延評価の仕組みを動的に可視化する手法を開発する。対象とするプログラミング言語はHaskellとし、Haskellのソースコードの例を示すことにした。

4. システム概要

本システムの開発にはProcessingを用いた。

4.1 処理の振る舞いの可視化

本システムは、ユーザが入力した変数名、引数、リストに対して、黒線で表現した関数と引数に図式化したリストを適用したアニメーション

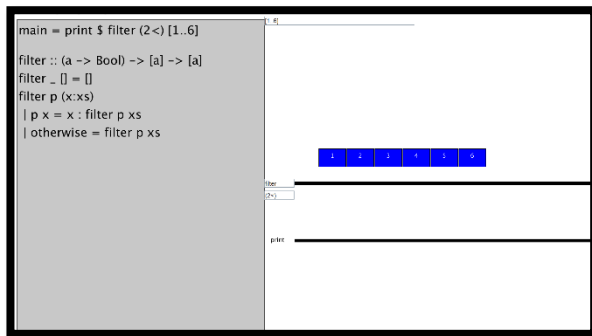


図 1 プログラムとリストデータの可視化
 ヲンを生成し、入力に応じた Haskell のソース
 コードを生成する。また、filter 関数、map 関
 数、foldl 関数の 3 つの関数の処理の流れを動
 的に可視化した。

図 1 に示した画面例は、システムに入力した
 値をもとに図式化したリスト、入力した関数を
 表す黒い横線、入力した内容の Haskell のプロ
 グラム、生成した Haskell のプログラムで使用
 される print 関数を表す黒い横線である。リス
 トが黒い横線の上に現れることで関数にリス
 トが与えられることを表現する。

本システムは、図 2 のようにリストが黒い横
 線に向かって落下することでリストの要素が
 評価されていることを表現する。図 2 では、
 filter 関数によって 2 より大きい要素が黒い横
 線を通してリストとして出力される。filter
 関数から出力されたリストは、print 関数に引
 数として与えられる。print 関数から出力され
 たものがプログラムの出力する文字列となる。

4.2 遅延評価の表現方法

本システムでは、実際に値が評価される要素
 のみをアニメーションで落下させることで、要
 素が評価されるタイミングを表現した。図 3 の
 ようにアニメーションが進んでも落下しない
 要素は、まだ評価されていないことを表す。

5. まとめ

本研究では、関数型プログラミング言語の
 Haskell で利用される基本的な関数を動的に

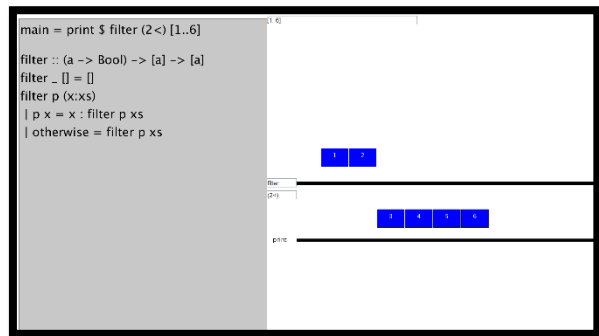


図 2 filter 関数によって評価されたリスト

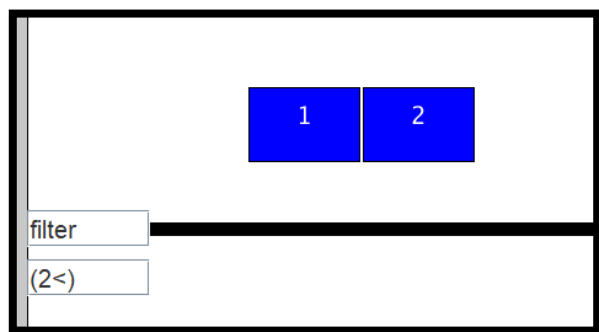


図 3 遅延評価によって評価されていない要素
 可視化するシステムを開発した。これにより、
 高階関数と遅延評価の基本的な処理の流れを
 動的に可視化することができた。

今後の展望として、今回実装できなかった関
 数の実装や参照透明性などの関数型プログラ
 ミング言語の性質の表現が必要である。また、
 ユーザによる使用評価も必要であろう。

参考文献

- [1] 岡本雅子, 村上正行, 吉川直人, 喜多一,
 「視覚的顕在化」に着目したプログラミング学
 習教材の開発と評価, 日本教育工学会論文誌,
 Vol.37, No.1, pp35-45, 2013.
- [2] 大城正典, 永井保夫, 初学者向けプログラ
 ミング学習のための初等アルゴリズム視覚化
 システム, 情報処理学会 情報教育シンポジウ
 ム 2018, pp104-111, 2018.
- [3] 竹内亮太郎, 大久保弘崇, 粕谷英人, 山本
 晋一郎, 空欄補充問題の自動生成による
 Haskell プログラミング学習支援環境, 情報処
 理学会 ソフトウェア工学研究会, Vol.2011-
 SE-171, No.15, pp.1-8, 2011.