2N-02

関数型プログラムのための 回帰推定による停止性判定ツールの試作

村本大起† 佐藤亮介† 鵜林尚靖† 亀井靖高†

† 九州大学

1 はじめに

停止しないプログラムはセキュリティなどの問題の原因になり，停止性を検証することは重要な研究課題である．

プログラムの停止性を判定する方法の一つに，ループまたは再帰ごとに単調減少し，ある一定値以下にはならないという性質を持つランキング関数を発見する手法が存在する．我々は，これまでより多くのプログラムを検証できるようにするため，ランキング関数の推定方法に回帰推定を取り入れた検証手法を提案し，実際に検証できることを評価してきた [1].

本研究では，これまでの検証手法をもとに関数型言語の停止性検証の枠組みを作成するため，停止性を自動で判定するツールを作成した．これにより，プログラムの入力により自動的に停止性を検証することができ，評価を簡単に行えるようになると考えられる．また，ランキング関数の推定方法を変更できるようにすることで今後の検証手法の改善を可能にできると考えている．

2 関数型プログラムにおける停止性検証

本節では，関数型プログラムを対象とした停止性検証手法について述べる．

2.1 関数型における停止性検証

関数型言語における既存の停止性検証では，有力な手法としてプログラムの遷移不変量に基づくもの [2,3] が存在する．我々は，以前より高階関数に対処した手法としてランキング関数の推定によって停止性を検証する研究 [3] に着目し，推定方法を変更することによって検証可能なプログラムを増やす研究 [1] を行ってきた．ここで，ランキング関数 r とは，プログラムのループまたは再帰において単調減少かつ 0 以上の値をとる整数の算術式である．妥当なランキング関数が得られれば，ループまたは再帰で必ず減少し，0 以上であることからプログラムが停止すると判定することができる．

例えば，次のプログラム pow を考える

```
pow n m = if m<1 then 1 else n * pow n (m-1)
```

ここで，変数 m は再帰ごとに減少し 0 以下になった場合にプログラムが終了するので，妥当なランキング関数は $r(n,m) = m - 1$ となり，このプログラムは停止することがわかる．また，妥当なランキング関数が得られない場合は停止しない可能性があると判定する．

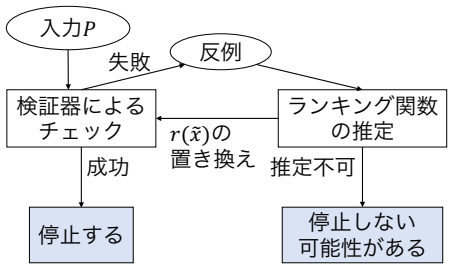

図 1: 停止性検証の自動化手法

2.2 検証手法

関数型言語における停止性検証手法として桑原ら [3] は既存のモデル検査器を用いて反復的にランキング関数の推定と検査を行う手法を提案している．停止性の検証手法の概要を図 1 に示す．

手法の流れは，(1) プログラムを入力する (2) ランキング関数が妥当であるか検証器で判定する (3) 妥当でないパスを反例として受け取りランキング関数を新たに推定する (4) 推定できたランキング関数が妥当になるまで (2) と (3) を繰り返す．

桑原らは，この手法により高階関数を含む停止性が自明でないプログラムに対しても停止性を検証することが可能であることを確かめている．

2.3 ランキング関数の推定方法

桑原らは，2.2 節の手法においてランキング関数の推定方法としては SMT ソルバを使用していた．ただ，関数に制御に直接関係しない引数が含まれる場合などで推定に失敗する例が見られた．そこで，[1] では反例

Prototype of Termination Checker by Regression for Functional Programs

Daiki Muramoto†, Ryosuke Sato†, Naoyasu Ubayashi†and Yasutaka Kamei†

†Kyushu University, 744 Motooka, Nishi-ku, Fukuoka, Japan

†muramoto@posl.ait.kyushu-u.ac.jp

{sato, ubauashi, kamei}@ait.kyushu-u.ac.jp

をもとに線形回帰を用いてランキング関数の推定を行い，検証できない例を検証できることを確認した．

3 停止性検証ツール

本節では本研究で作成した停止性検証を行うツールについて述べる．

3.1 概要

今回作成したツールはプログラムを入力することで，自動でプログラムを推定して検証結果を表示する．プログラムをテキストデータとして入力し，プログラムが停止すると判定できた場合にはその関数のランキング関数を出力し，そうでない場合は停止するかどうか不明という結果を示す．

ツールでは，2.2節の手法においてランキング関数の推定方法に線形回帰を取り入れた検証手法によって停止性を検証しており，制御に関係しない変数を含む場合でも検証可能だと考えられる．

サポートする言語は関数型言語の OCaml から再帰関数について検証を行える最低限のサブセットを対象としている．

3.2 実装と実行環境

ツールの実装は OCaml を用いており，現在は Linux 上のネイティブアプリケーションとして動作する．将来的にはインターフェースを公開し，web から動作できるようにすることを考えている．

3.3 対象言語

対象とする言語は OCaml のサブセットで，プログラムは関数定義によって構成される．条件式 `if` や関数適用，二項演算 (`+`，`-`，`*`等)，変数束縛を含む．また，不確定な分岐等を考慮するため，実行時に非決定的に評価される整数`*int` を含む．定数には整数，真偽値と unit 型 `()` を含む．

例えば，`pow` の検証を行う入力は次のようになる．

```
let rec pow n m = if n<1 then 1
  else n * pow n (m-1)
let main () = pow *int *int
```

非決定的な整数を含むことで検証器によって全ての入力に対して検証を行うことができる．

3.4 推定方法の変更

本研究では，ランキング関数を推定する方法として線形回帰を用いて検証を行っているが，プログラムによってはランキング関数が単純な線形結合式で表すことができないものが存在する．そこで，ランキング関数の推定方法を変更できるようにすることで，今後推定方法の変更で検証可能なプログラムを増やすことができるようになると考えられる．

3.5 実験

本研究の手法にて，桑原らの手法で検証できなかった制御に関係のない変数を含むものを検証できることを確認した．次の末尾再帰で階乗を求めるプログラムは桑原らの手法では検証不可であったが，

```
let rec fac n acc = if n<1 then acc
  else fac (n-1) (n*acc)
let main () = fac *int 1
```

今回の手法ではランキング関数 $r(n, acc) = n - 1$ を推定し停止することを判定できた．

4 おわりに

本論文では，関数型プログラムの停止性検証を自動で行うツールを作成した．検証を自動化することによって，より多くのプログラムに対して実験を行えるようになり，正確な停止性検証の評価を行えるようになると考えられる．また，ランキング関数の推定方法を置き換えることができるため，検証手法を改善することができると考えられる．今後の課題としては，より多くのプログラムを検証することによって既存研究と検証できるプログラムの比較による評価と，ランキング関数の推定方法を変更することによって検証できるものをさらに増やすように改善することが挙げられる．

謝辞

本研究は，JP26240007，18H04097 による助成を受けた．

参考文献

[1] 村本大起, 佐藤亮介, 鵜林尚靖, 亀井靖高. 関数型言語における停止性検証のためのランキング関数の回帰推定 (ソフトウェアサイエンス). 電子情報通信学会技術研究報告, Vol. 117, No. 477, pp. 63–68, 2018.

[2] Ruslán Ledesma-Garza and Andrey Rybalchenko. Binary reachability analysis of higher order functional programs. In *Static Analysis - 19th International Symposium, SAS 2012, Deauville, France, September 11-13, 2012. Proceedings*, pp. 388–404, 2012.

[3] Takuya Kuwahara, Tachio Terauchi, Hiroshi Unno, and Naoki Kobayashi. Automatic termination verification for higher-order functional programs. In *European Symposium on Programming, ESOP 2014, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings*, pp. 392–411, 2014.