

Java プログラム中の数式表現の揺らぎを抽出するシステムの試作

秀山 祐司*1 樋口 昌宏*1 中田 達也*2

近畿大学大学院総合理工学研究科*1 近畿大学理工学部情報学科*2

1. はじめに

ソフトウェアの保守性を向上させるためにはソースコードの可読性を高めることが重要である。可読性を妨げる要因として同じ意味の数式を異なる表現で記述している場合がある。本研究では、この様な場合を数式表現の揺らぎと呼び、Java プログラム中からこれを検出しプログラマに提示することを目的とする。検出手法として、プログラムから全ての数式を抽出し、それぞれの式が等価であるかの判定を行う。今回は、数式の抽出に際して、変数名やメソッド呼び出しの引数の違いを吸収することで、より多くの数式表現の揺らぎを検出できるように作成を行った。

2. 数式表現の揺らぎ

まず、数式表現の揺らぎに関して記述する。本研究の数式表現における揺らぎとは、プログラム中に出現する2つの数式が論理的に等価であるにも関わらず異なる表現が混在することを指す。例えば、英語・数学に基づいた記述では係数を先に記述する(3*apple+4*orange)場合や、日本語では主語述語の順に記述する(apple*3+orange*4)場合等がある。この様に、プログラマ毎に思考言語、思考方法が異なるため数式表現の揺らぎが現れると考えられる。

次に検出したい数式表現の揺らぎに関して記述する。数式に現れる基本構成要素はフィールド変数、ローカル変数、メソッド呼び出しの戻り値、定数で構成されており、表 1, 2 に揺らぎとして判定したいものを示す。x, y, z はフィールド変数, i, j, k はローカル変数とする。

表 1 における 1) は変数 x, y の順序が入れ替わっているため揺らぎとする。2) は用いられている変数は式 1, 式 2 で共通していないが x, y はフィールド変数, i, j はローカル変数でそれらの順序が入れ替わっているため揺らぎとする。3) についてもローカル変数とメソッド呼び出しの戻り値が現れる順序が入れ替わっているため揺らぎとする。4) はフィールド変数とローカル変数のみからなる部分式の現れる順序が入れ替わっ

表 1 算術式の揺らぎ

	式 1	式 2	揺らぎ
1)	x+y	y+x	○
2)	x+i	j+y	○
3)	k+y.size(i)	z.get(j)+k	○
4)	x*(i+j+k)	(i+j)*y	○
5)	2*(i+j)	(i+j)*4	×
6)	i*(x+z)	i*x+i*z	○

表 2 不等式の揺らぎ

	式 1	式 2	揺らぎ
1)	i>0	0<i	○
2)	i>j	i-j>0	○
3)	i>0	i>=1	○

ているため揺らぎである。5) は括弧を1つの部分式として見ると順序が入れ替わっていると考えられるが、定数が異なるため揺らぎではないとする。6) についても揺らぎとする。

表 2 の不等式は全て互いに論理的に等価であるため揺らぎであると言える。

3. 検出手法

本研究では、プログラム上の数式表現の揺らぎを検出するために、まず対象のプログラムに構文解析を行い、数式を全てリストアップする。リストアップした数式に変数名やメソッド呼び出しの引数の違いを吸収するために、3.2 で述べる抽象化を行う。抽象化した複数の式を SMT ソルバ である z3^[1]形式の数式としてリストに保存し、そのリストを z3 駆動部に渡し、z3 の等価性評価プログラムを実行する。実行結果として、リストに保存された算術式同士、不等式同士がそれぞれ等価であるか、また、数式表現の揺らぎであると判定された場合に、その数式が元のプログラムの何行目に存在しているかを表示する。以上が数式表現における揺らぎを検出するシステムのフローである。

A prototype system for extracting fluctuations in mathematical expressions in Java programs

*1 Graduate School of Science and Engineering, Kindai University

*2 Department of Informatics, Faculty of Science and Engineering, Kindai University

3.1. 構文解析

解析対象プログラムに構文解析を用いて、全ての算術式、不等式の抽出を行いテキストファイルに出力した。また、この時、数式の出現位置の記録を行い、数式表現の揺らぎが検出された位置を出力できる様にした。

3.2. 抽象化

より多くの揺らぎを検出するために構文解析で渡された数式に抽象化を行った。例えば、`height`, `width` をフィールド変数, `input` をローカル変数としてプログラム中に現れる数式 `height*(width+input)` に対し, `height*(width+input)`, `width+input`, の2つの数式が抽出される。しかし、このままでは表1の2), 3), 4)の様な揺らぎの抽出には適さない。そこで、フィールド変数, ローカル変数, メソッド呼び出し, 算術式のそれぞれの名前を抽象化することにした。それぞれの頭文字 `f`, `l`, `m`, `e` を取り, 左から順に出現した変数を, 例えばフィールド変数であれば, `f1`, `f2`, `f3`, ...と変換する。メソッド呼び出しに関しては, 与えられた引数に違いが存在しても同じメソッド呼び出しとして判定することができる。これにより, 上記の抽出された数式はそれぞれ `f1*e1` と `f1+l1` という形に抽象化される。

3.3. z3による等価性評価

`z3` は与えられた論理式が充足可能であるか判定し, 充足可能であればその値を返すことができる。本研究では不等式, 算術式を扱っているため, 不等式と算術式の等価性の判定方法について記述する。

まず, 不等式に関して, 論理式 `A`, `B` において, `A ∧ ¬B` と `¬A ∧ B` が充足不能であることを判定することで `A` と `B` が等価であると判定できる。

次に算術式に関して, 算術式 `C`, `D` において, 2式が等価であることを判定するために `¬(C=D)` が充足不能であることを判定することで `C`, `D` が等価であると判定できる。

4. 揺らぎ検出実験

解析対象プログラムとして, 数式表現の揺らぎとして検出したい算術式対, 不等式対を意図的に含ませたプログラムと, 商店の売上高における売上総利益率や当期純利益率等を計算する数式を含むプログラムを用いた。表3に抽出, 評価結果を示す。今回は異なる変数の種類で順序が異なる場合数式の変数が現れた順に抽象化を行ったがため表1の `x+i` と `i+y` を揺らぎとして検出することができたが, カウンタ変数 `i` と一般的な

表3 揺らぎ検出実験

	意図的	売上高
行数	48行	170行
抽出した数式		
算術式	12個	100個
不等式	7個	4個
判定試行回数	87回	4956回
数式表現の揺らぎ		
算術式	8個	49個
不等式	3個	2個

ローカル変数である `input1` と `input2` を用いた数式 `i+input1` と `input2+i` の様に同じ種類の変数同士の順序の揺らぎを抽象化するとどちらも `l1+l2` となり揺らぎとして検出できなかった。

また, 数式に括弧が現れた場合部分式として `e1` と抽象化を行ったため, 表1の `i*(x+z)` と `i*x+i*z` の様な分配則を含んだ数式を検出できなかった。

5. 考察

今後の課題として, 同じ種類の変数同士の順序の揺らぎの検出が正しく行えなかったため, 変数の種類の分類をより細かく行うべきであると考える。今回の場合で例えるとカウンタ変数をローカル変数とは別の種類として分類を行うことや, ローカル変数をスコープの現れた順に重みをつけることで正しく検出を行うことができると考えられる。

また今回は定数の抽象化は行っていない。定数は計算により値が変わるため等価性判定において結果が変わるためである。しかし実際には, プログラム中に現れる数式に定数同士の計算はあまり見られなかった。そのため, 定数の抽象化を行うことで, 定数の順序を含むより多くの揺らぎの検出を目指す。

参考文献

- [1] 梅村 晃広 SAT ソルバ・SMT ソルバ の技術の応用, コンピュータソフトウェア, Vol.27, No.3(2010)
- [2] GitHub [<https://github.com/Z3Prover/z3>] 2019/12/27 閲覧