プログラム認識のための階層的類似性評価手法の検討 — 式レベルの類似性について —

3H－5

安岡 義弘　　上田 賀一

茨城大学

1 はじめに

理解、認識は、AIの基本テーマであるが、知的ソフトウェア工学においても重要な課題である。プログラム理解とは、プログラムの機能、動き、さらにはそのプログラムを作った人の意図を人間が理解することである。この作業は大変困難であるが、プログラム開発段階、デバッグ段階、保守段階において重要な課題であり、これをコンピュータに支援させるべく二つの観点から研究が行なわれている [4]。一つはデバッグの観点 [1][2]、もう一つはリバースエンジニアの観点 [3] である。前者は主にデバッグ段階、後者は主に保守段階である。

また、プログラム理解、認識においてプログラムの類似性を評価することは、正しいプログラムと与えられたプログラムを比べる上で非常に重要なことである。つまり、類似性を評価するプログラムの機能が高いほど認識率を高めると考えて良い。

本稿では、特に式レベルに注目し類似性を評価する手法を検討する。

2 階層的類似性評価とは

これまでの研究では、図1のようなプログラムの階層に対して、トップダウン的にプログラムを認識する方法 (上の階層から下の階層に向かって認識する方法) が多く、ボトムアップ的にプログラムを認識する方法は見かけなかった。しかし例えば、正しいプログラムと与えられたプログラムの変数の対応の取り方を考えると、「構文から情報を得て、式で使用される変数と対応づける方法」(トップダウン的な方法) もあるが、「式から情報を得て、構文中で使用されている変数に対応づける方法」(ボトムアップ的な方法) も考えられる。つまり、式からの情報抽出と構文からの情報抽出の、どちらを先にすべきか決まっているわけでない。同様にこのことは全ての階層にいえることであり、トップダウン的にもボトムアップ的にも認識は可能であり、どちらを先に行なうかは決まっているわけではないということである。

そこで階層を細かく設け、それぞれの階層から情報を抽出しておき、認識や類似性の評価を行なう方法を考える。

例えば式を標準化する際に一度に標準化せず、階層を設けて使用されている変数の対応を抽出するなどトップダウン的にプログラムを認識する過程で必要と思われる情報を式から抽出し、それぞれの階層で類似性を評価する。このように階層別に類似度を考え評価する方法を「階層的類似性評価」と呼ぶことにする。

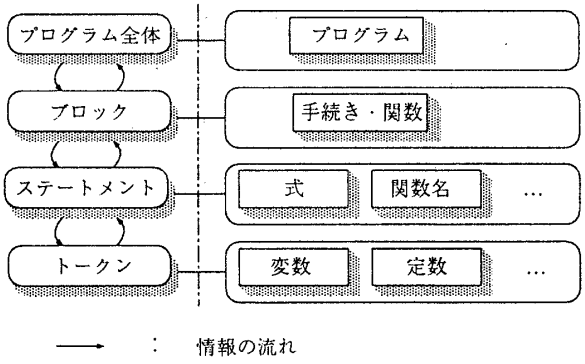

図 1: プログラムの階層

3 式から階層的に抽出できる情報

これまでの式の認識には、式を標準化 (括弧を展開しソートするなど) してから、テンプレートマッチングを行なうなど、先に標準化を行なっていた。しかし式中の括弧にも、プログラムを作った人の意図があると考えられる。

Evaluation Method for Hierarchical Similarity in Program Recognition – especially for expression level –
Yoshihiro Yasuoka,Yoshikazu Ueda
Ibaraki University

例えば、

$$y = 2 \times (x - 1) + 3$$

の場合、括弧を展開すると

$$y = 2 \times x + 1$$

になる。当然上式も下式もyに代入される値は同じだが、上式は括弧の部分から$(x-1)$に何らかの意図があるのではないかと考えられる。つまり直線の式だと考えたとき、上式は$(1,3)$を通る傾き2の直線、下式は$(0,1)$を通る傾き2の直線となる。同じ直線だか、プログラムを作った人の意図が違うのがわかる。このように、括弧を展開する前の式にも情報があることが考えられる。

次に類似性認識の際の変数等の対応であるが、これまでは特定のステートメントのテンプレート中で使用されている変数と、テンプレートとマッチしたプログラムで使用された変数とを対応させる方法を用いているものが多かった。この方法はバグを含まない式には有効であるが、バグがあると対応しない場合がある。例えば、テンプレートでcountというカウントする変数をプログラム内でxという変数に対応するはずであるものが、対応する箇所において一箇所違う変数名にしてしまうと、正しく対応できない場合がある。そこで、ステートメントだけでなく式からも対応をとることができれば、さらに認識率が上がる。その方法は後で述べるが、そこで必要な情報は使用されている変数の相関関係である。この変数の相関関係を抽出するために、プログラムスライシング [5] の手法を用い、複数の式において代入、参照がされたかどうかを調べる必要がある。

4 評価手法

類似性評価のために以下のような階層を設ける。二つのプログラムが一致した階層が上であるほど類似性が高いものとする。

◇ 式

- プログラマが入力したままの式
  使用させている定数、関数、変数の種類とそれぞれの個数、括弧を比較する。
- 定数を代入した式
  定数をまとめて、変数、関数、括弧のみで比較を行なう。
- 括弧を外した式
  同じ項はまとめて、項の係数、個数、符合を比較する。
- 変数の相関関係を基にソートした式
  相関関係から変数の対応関係がわかるので、それに従って並び替え、項の係数、符合を比較する。

◇ 式列 (複数の式)

- 並列的な式列を並び替えた式
  並列的に処理できる式に対しては、順序を入れ換えることが可能であることは当前であるが、一方の式列に他方の式列が対応するように変換を行なう。その変換度合により評価を行なう。
- 順次的な式列を結合した式
  並び替えできない式列において、変数へ代入などを行ない式を結合して、式になるものは、それに対して式の評価を行なう。

5 おわりに

式の類似性を評価する一手法を述べた。現在先の評価手法を踏まえた類似性評価システムをflex,flex++,bison,c,c++を用い作成中である。並列性の抽出手法や、式だけでなくPascalのプログラムの類似性を評価するシステムの構築に向けて検討中である。ここでは、diffなど既存アプリケーションを利用して評価する手法などを考案しシステムを試作する予定である。

参考文献

[1] 海尻賢二、原克巳： ゴール／プランに基づくプログラム理解システム, 信学技報,KBSE92-3,pp.15-22(1992)

[2] 中島歩、上野晴樹：プログラミング知識表現およびプログラム理解システムへの応用, 信学技報,KBSE92-4,pp.23-30(1992)

[3] 関本理佳、海尻賢二: プログラム認識による仕様記述生成について, 信学技報,KBSE93-10,pp.43-50(1993)

[4] 竹下亨：「CASEを探求しよう」, bit Vol.25, No.1-No.12, 共立出版 (1993)

[5] Weiser,M.： Program Slicing, *IEEE Trans. Software Engineering* Vol.SE-10,No.4, pp352-357(1984)