

5D-3

開発支援システム REFINE を用いた C プログラム依存関係解析ツールの試作

栗飯原述宏[†] 高田智規[‡] 井上克郎[†][†]奈良先端科学技術大学院大学 [‡]大阪大学 基礎工学部

1 はじめに

プログラムのデバッグを行なうことは、非常に労力の多い作業である。単にプログラムを眺めるだけでは、わずかなプログラミングのミスを発見するのに多大な時間を費やすことも珍しくない。

特定の変数に影響を与える文(プログラムスライスと呼ばれる)のみをプログラム全体から抜き出せば、ミスを効率良く発見できることが期待される。これには、ある変数が参照された時、その変数がプログラム中のどこで定義される可能性があるのかわらなければならない。すなわち、変数の定義・参照の関係(依存関係)を正確に理解する必要がある。

本研究では、応用範囲の広い C 言語を対象として、プログラムの依存関係の解析を行うツールを試作した。試作には REFINE システムを用い、これにより、依存関係はグラフ化されて表示される。グラフ表示することで、複雑な依存関係を視覚的にとらえることができ、バグの発見に重要なプログラム依存関係の認識の向上が期待できる。

2 プログラム依存グラフとスライス

2.1 プログラム依存グラフ

プログラム依存グラフ(Program Dependence Graph, 以下 PDG) [1] とは、プログラム内の文の間の依存関係を表す有向グラフである。PDG の節点はプログラムに含まれる条件判定文や他の一般の文を表し、有向辺は 2 つの節点の間の依存関係を表している。2 つの節点の間の依存関係には、条件文や繰り返し文などの条件判定によるプログラムの制御の依存関係を表す制御依存関係と、変数の値の依存関係を表すデータ依存関係がある。

このように、PDG はプログラムの依存関係をグラフ化したものであるため、PDG を表示することができれば、依存関係のより良い理解が可能となる。

2.2 プログラムスライス

プログラムスライス(Program Slice, 以下 スライス) [2] とは、プログラム中で参照されたある変数の値に影響を与える全ての文、もしくは、定義されたある変数の値が影響を与える全ての文のことをいい、PDG から求めることができる。スライスは、デバッグや保守などに利用されることが多い。

スライスには、静的スライスと動的スライスがあるが、本研究では静的スライスのみ扱う。

3 REFINE システム

REFINE システムは、ソースコードの解析・変更を行うツールの開発を支援するシステムである。

このシステムで用いられる REFINE 言語は、豊富なデータ型や機能を備えた言語であり、集合や木構造を扱えるなどソースコードの解析を行なうのに適している。また、付属のサブシステムを利用することにより、構文木の構築・変更や、その図式表示などといった操作を比較的容易に行うことができる。

4 依存関係解析ツール

4.1 ツールの概要

入力されたプログラムに対して、その PDG とソースコードをあわせて表示する、プログラム依存関係解析ツールを試作した。本ツールはまた、スライスの計算も行なうことができる。構文解析やグラフの表示には、REFINE のサブシステムを用いた。

スライス計算などの結果は、その都度グラフおよびコード上に同時に表示されるため、そこから依存関係の対応を容易に読みとることができる。また PDG において、複合文のような複数の節点をまとめて 1 つの節点にすることにより、複雑な PDG を見やすくすることができる。

Developing a C Program Analysis Tool Using a Support Environment REFINE

Nobuhiro Aihara[†], Tomonori Takada[‡], Katsuro Inoue[‡]

[†]Nara Institute of Science and Technology (NAIST)
8916-5 Takayama, Ikoma, Nara 630-01, Japan.

[‡]Faculty of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560, Japan.

```

1:  include<stdio.h>
2:
3:  int g;
4:
5:  int f(p)
6:  int p;
7:  {
8:      int l;
9:
10:     if( p > 1 ){
11:         l = 2 * f( p - 1 );
12:         g = g + 1;
13:         return( l );
14:     } else {
15:         g = 1;
16:         return( 1 );
17:     }
18: }
19:
20: void nth()
21: {
22:     int n, a;
23:
24:     g = 1;
25:     n = 0;
26:     a = f( n );
27:     printf( "%d", a );
28: }
29:
30: void main()
31: {
32:     g = 0;
33:     nth();
34:     printf( "%d", g );
35:     exit( 0 );
36: }
    
```

図1: 入力プログラムの例

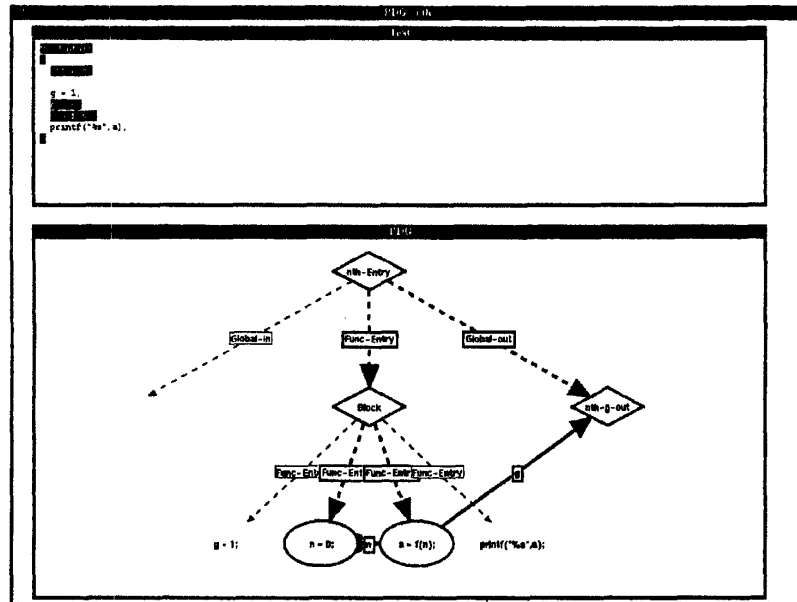


図2: スライスの計算結果(関数 nth() に対して)

4.2 ツールの機能

- 入力されたプログラムから関数呼び出しグラフ (Call Graph) を作成し、その表示を行なう。
- PDG をグラフ表示部に、ソースコードをコード表示部に表示する。これらは、Call Graph で指定した関数ごとに表示される。
- 文中の任意の変数名、スライスの種類 (前向き、後向き) を指定することによって、スライスの計算を行なう。スライス計算の起点は、コード表示部では文を、またグラフ表示部では節点を指定することにより得られる。計算結果は PDG, ソースプログラム両方に反映される。
- PDG において複数の文の節点は、まとめて1つの節点として表示することができる。

4.3 ツールの実行

実際に、図1のプログラムの34行目の大域変数 g に対して、後向きにスライスを計算した結果を図2に示す。

スライスは、コード表示部では反転表示として、またグラフ表示部では回りを囲まれた接点として表されている。依存関係は、この図から容易に読みとることができる。

5 考察

本ツールを実行させると、関数ごとに PDG とソースコードが同時に表示される。これにより、依

存関係の対応が取りやすくなり、その理解に役立つものと思われる。

本ツールは、現段階ではスカラー変数しか扱っていないため、ポインタ等を用いたプログラムは正確に解析できないが、初級者クラスのプログラミングにおけるデバッグ支援ツールとしては十分利用し得るものと考えられる。

6 おわりに

本研究では、REFINE システムを用い、C プログラムの依存関係解析を行うツールの試作を行なった。試作にあたり、REFINE のサブシステムを用いることで、既存の言語では非常に困難である PDG などの表示を比較的容易に行うことができた。

しかし、本ツールはまだ試作段階であるために課題も多い。改良すべき点としては、ポインタ変数に対する依存関係解析、ユーザインタフェース機能の充実、ツール上でのCプログラムの作成・実行、などが挙げられる。

参考文献

- [1] S. Horwitz and T. Reps: "The Use of Program Dependence Graphs in Software Engineering", Proc. 14th Int. Conf. on Software Engineering, pp.392-411, 1992.
- [2] 下村 隆夫: "プログラムスライシング技術と応用", 共立出版, 1995.
- [3] 植田, 練, 井上, 鳥居: "再帰を含むプログラムの依存関係解析とそれに基づくプログラムスライシング", 信学技報, Vol.SS93, No.24, pp.33-40, 1993.
- [4] 植田, 練, 井上, 鳥居: "再帰を含むプログラムのスライス計算法", 信学論 (D-1), Vol.J78-D-1, No.1, pp.11-22, 1995.