区間限定スライスを用いた再利用部品抽出手法の評価

5M－10

丸山 勝久　島 健一　高橋 直久
{maru,kshima,naohisa}@slab.ntt.jp
NTT ソフトウェア研究所

1 はじめに

部品を用いたソースコード再利用において，部品の抽出を容易にする手法が望まれている[1]．これに対して，我々はプログラムスライシング[2]に区間という概念を導入した**区間限定スライス** (bounded slice) を定義し，このスライスを用いて既存のソースコードから再利用可能な部品を容易に抽出する手法を提案した[3]．本部品抽出手法は従来の手法と比較して，以下に示す利点を持つ．

- 部品作成者は抽出する部品の機能を自由に決定することが可能となり，不要な記述コードを部品から半自動的に削除できる．

我々は，本手法に基づき実際のプログラムから再利用部品を抽出する実験を行い，上記の利点を確認した．本稿では，実験結果と考察について述べる．

2 スライシングによる部品抽出手法

従来のスライシングでは，部品作成者がスライシングの対象範囲を自由に指定することはできず，スライスを部品とみなすと抽出した部品は部品抽出対象のプログラム構造に大きく依存する．よって，部品の機能を変数の値を決定することとおくと，部品作成者の意図する機能だけを持つ部品を抽出することができない．

これに対して，区間限定スライスを用いた部品抽出手法では，既存のプログラムにおいてスライシング対象範囲を部品作成者が任意に指定可能である．本手法により部品を抽出する際には，部品作成者が部品作成基準 $C_C = (N_u, N_l, V)$ を指定する．N_u, N_l は部品抽出対象プログラムの制御フローグラフ (CFG) 上で指定する上限及び下限節点で，2節点の制約付き到達可能経路 (CRP)[3] がスライシング対象区間である．V は部品の出力として着目する変数である．部品作成基準の指定後は，部品 C_b = 区間限定スライス $\hat{S}_b(N_u, N_l, V)$ が自動抽出される．本手法で抽出した部品は適当な入力データのもとで実行可能で，部品内の変数に関してはもとのプログラムを実行させたときと等価な値を出力することが保証される．

図1に部品抽出対象プログラムと抽出した部品を示す．節点番号の左側の○は従来のスライシングにより抽出した部品，●は本手法により抽出した部品を表す．図1において，本手法による部品は従来手法による部品と比較して節点 {1,2,3,4,9,12} を含まない．このように，本手法では不要なコードが削除でき，従来手法では抽出不可能であった部品が抽出可能である．

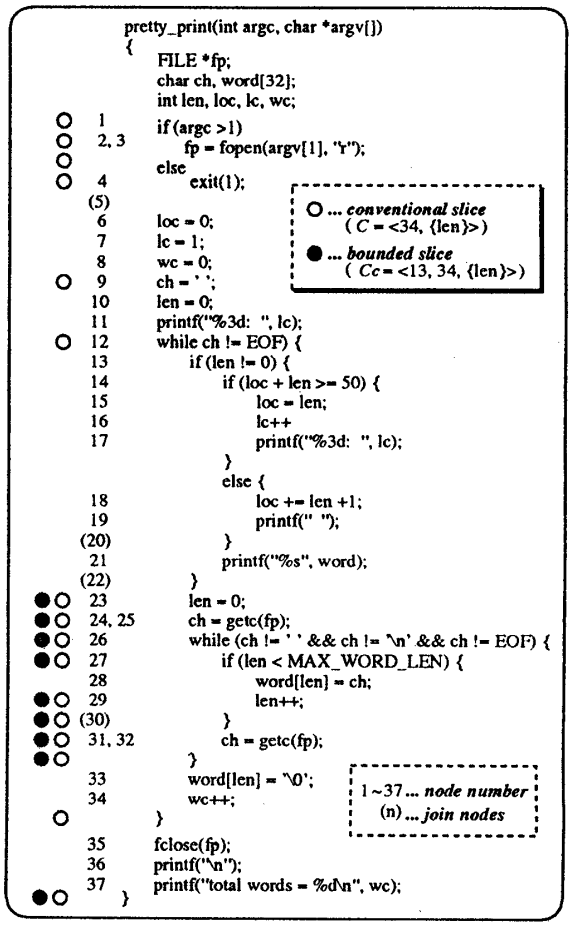

図 1: 部品抽出対象プログラムと抽出した部品

3 評価実験

本部品抽出手法に基づいてワークステーション上に部品作成実験システムを構築し，既存のプログラムから部品を抽出する実験を行った．部品抽出対象プログラムは，表1に示す NetBSD のソースコードである．

表 1: 実験対象プログラム (ソースコード)

プログラム	LOC	関数の数	節点数	抽出部品
whois	131	2	81	A1~A4
finger	1240	24	799	B1~B5
ftp	5595	128	4399	C1~C6

表1のプログラムから抽出した部品の節点数及び入力パラメータ (入力引数) を表2に示す．表2において，対象関数とは実験対象プログラムを関数単位で分割したもの，従来部品とは対象関数に対して従来のスライシング手法を適用して抽出した部品を指す．

Evaluation of a Reusable Component Extraction Method by Bounded Slices
Katsuhisa MARUYAMA, Ken-ichi SHIMA and Naohisa TAKAHASHI, NTT Software Laboratories

表 2: 表 1 のプログラムから抽出した部品の部品作成基準，機能，節点数及び入力パラメータ

部品	部品作成基準	機能	対象関数		従来部品		本手法による部品			
			節点数	入力引数	N	入力引数	CRP	C_b	C_b/N	C_{in}
A-1	⟨20,49,{connect}⟩	ソケットの結合	81	argv,argc	44	argv,argc	13	12	27%	host
A-2	⟨17,29,{s}⟩	ソケットのオープン	81	argv,argc	27	argv,argc	5	4	15%	host,argc
A-3	⟨30,69,{sfi}⟩	データポインタの設定	81	argv,argc	49	argv,argc	19	10	20%	s,hp
A-4	⟨20,75,{ch}⟩	データの読み込み	81	argv,argc	57	argv,argc	30	19	33%	host
B-1	⟨9,42,{connect}⟩	ソケットの結合	83	name	37	name	22	21	57%	host
B-2	⟨26,36,{s}⟩	ソケットのオープン	83	name	23	name	7	2	9%	hp
B-3	⟨9,78,{fp}⟩	データポインタの設定	83	name	32	name	53	23	72%	host
B-4	⟨54,76,{c}⟩	データの読み込み	83	name	64	name	25	24	38%	s
B-5	⟨35,76,{c}⟩	データの読み込み	83	name	64	name	35	27	42%	sin,hp
C-1	⟨f9,f30,{connect}⟩	ソケットの結合	96	host,port	21	host,port	18	15	71%	host,port
C-2	⟨c40,f30,{connect}⟩	ソケットの結合	213	argc,argv	42	argc,argv	27	24	57%	argc,argv
C-3	⟨f2,f66,{s}⟩	ソケットのオープン	96	host,port	65	host,port	40	26	40%	host,port
C-4	⟨f22,f91,{cin}⟩	データポインタの設定	96	host,port	48	host,port	30	17	35%	histaddr,hp
C-5	⟨m68,c52,{port}⟩	ポート番号の取得	203	argc,argv	52	argc,argv	16	13	25%	argc,argv
C-6	⟨m68,f92,{hostname}⟩	ホスト名の取得	299	argc,argv	85	argc,argv	121	24	28%	argc,argv

部品作成基準の節点 m, f, c : main.c, ftp.c, cmds.c

4 考察

本手法で抽出した部品と従来のスライシングを用いて抽出した部品を比較し，本手法の利点を確認する．本手法に対する評価項目として以下の 3 点を設定した．

(1) 任意の区間を指定した際，抽出した部品は不要なコードを含まないか (節点数の比較).

(2) 区間を変化させることで，機能の異なる部品が抽出可能か (入力パラメータの比較).

(3) 異なるプログラムから抽出した同じ機能を持つ部品のコードは等しいか (機能とコードとの比較).

4.1 従来部品と本手法による部品の節点数

表 2において，本手法による部品のすべての節点は従来部品に含まれ，本手法による部品の節点数 C_b は従来手法による部品の節点数 N より必ず小さくなる (節点数の割合 C_b/N = 9%~71%)．これは，部品の抽出対象が区間 CRP に限定されているためであり，削除されたコードは着目した区間の外に存在していたコードか例外処理のため部品外部に制御が移るコードであった．以上より，評価項目 (1) に関して，次に示す効果が確認できる．

(a) 任意の区間を指定して部品を抽出することが可能で，指定した区間において着目した変数に影響を与えない記述コードが部品から削除される．

4.2 指定する区間と入力パラメータ

表 2より，従来部品の入力引数は対象関数の入力引数に支配されている．これに対して，本手法は入力パラメータ C_{in} をさまざまな変数に設定できる．これは，対象関数の入力引数と C_{in} を比較すれば明確である．さらに，本手法では着目する変数を同じものに指定しても，指定する上限節点により C_{in} を変化させることができる．例えば，B-4 と B-5 は着目する変数が節点 76 の {c} で同じであるが，上限節点の位置 (54 と 35) によって C_{in} は異なる．入力パラメータは部品の機能を決定する要素の一つであるとみなせるので，評価項目 (2) に関して，次に示す効果が確認できる．

(b) 入力パラメータとなる変数の選択肢が広がり，従来のスライシングでは抽出できない機能を持つ部品が抽出可能である．

4.3 抽出した部品の機能とコード

本手法により抽出した部品の機能とコードとの関係を調べるために，同じ機能を持つように部品作成基準を設定して部品を抽出した．表 2の A-1 と B-1 では，部品の節点数 C_b 及びコードは大きく異なる．逆に，A-1 と C-1 では C_{in} が異なりはするが，C_b はほぼ等しく，そのコードはほとんど同じであった．これは，抽出対象のプログラムの構造が似ていたためである．以上より，評価項目 (3) に関して，次に示すことがいえる．

(c) 異なるプログラムからでも同じ機能を持つ部品を抽出することは可能であるが，部品のコードは抽出対象のプログラム構造に影響を受ける．

5 おわりに

区間限定スライスを用いて既存のプログラムから再利用部品を抽出する手法に基づき評価実験を行い，その実験結果と考察を述べた．本手法は，既存プログラムの任意の区間を指定して，部品抽出が可能であるという点で従来の手法より有利である．今後の課題として，部品作成基準を指定する際の支援について検討する予定である．

謝辞 日頃御指導御討論いただく伊藤正樹リーダーはじめ，グループの皆様に深く感謝します．

参 考 文 献

1) Abd-El-Hafiz,S.K., Basili,V.R. and Caldiera,G.: *Towards Automated Support for Extraction of Reusable Components*, Proc. Conf. Softw. Maintenance, pp.212-219 (Oct. 1991)

2) Weiser,M.: *Program Slicing*, IEEE Trans. Softw. Eng., Vol.10, No.4, pp.352-357 (Jul. 1984)

3) 丸山勝久, 高橋直久: プログラムスライシングによるソフトウェア部品の作成, 情報処理学会ソフトウェア工学研究会 (94-SE-98), Vol.94, No.43, pp.1-8 (May 1994)