IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 11, Issue 6 (May. - Jun. 2013), PP 51-54
www.iosrjournals.org

Static Slicing Technique with Algorithmic Approach

Sonam Agarwal[1], ArunPrakash Agarwal[2]
[1](M.Tech CSE, Amity University, Noida, India)
[2](Assistant Professor, Amity University, Noida, India)

Abstract:*In order to improve the accuracy of the static programslicing which is used to locate the faults that cause an exception,we propose a new algorithm which isstored when an exception occurred. The proposed approachconsists of various steps and figure out those methods and statements thathave not been executed. Then, these methods which are notexecuted will be ignored when building the system dependencegraph. Finally, the accurate slice will be got by adopting theimproved program slicing algorithm. And the result shows that using our approachthe slice is 8 percent less than using the general static programslicing algorithm on average.One approach to improve the comprehension ofprograms is to reduce the amount of data to be observedand inspected. Programmers tend to focus andcomprehended selected functions (outputs) and those partsof a program that are directly related to that particularfunction rather than all possible program functions. Oneapproach is to utilize program slicing, a program decomposition technique that transforms a large programinto a smaller one that contains only statements relevantto the computation of a selected function.*
Keywords: *Program slicing, Static slicing, Static slicing algorithm*

I. Introduction

Program slicing is a decomposition technique to solve program analysis problem. Program slicing is used for slicing the large program into small and simple programsthat contain only that statement that is necessary to complete any execution or execute any program without any error. It is widely used for program analysis, understanding, testing, debugging, metricetc. The concept and principle of program slicing were first established by M.Weiser [6] in 1979, and after thatmany other slicing techniques has been implemented. Program slicing is a feasible method to restrict the focus of a task to specific sub-components of a program.
The program slicing can be classified as static slicing anddynamic slicing. For a specific slicing criterion, the maindifference between them is that the information they required.Static program slicing only needs static information which is the source code, while the other requires the entire execution trace which is corresponding to a specific program input. In other words, static slicing considers all possible executions but dynamic slicing only considers the execution of a specific input [16]. Therefore, dynamic slicing contains fewer statements and is more precise than the static one. The optimizeddynamic slicing will take several minutes to perform the slicingand the preparing time is even longer [2]. With complex program and long execution traces, dynamic slicing is not appropriate.

Applying slicing technique to software architectures can benefit software development in two main ways [18]. The first one concerns maintenance of a component-based software. By using slicing tools on an architectural description, we can determine which components might be affected when a given component is modified. Second, architectural reuse can be facilitated. While reuse of code is important, reuse of software design and patterns are expected to offer greater productivity benefits and reliability enhancements[22].

II. Paper Work

In this paper we firstlywritethe simple program along with the sliced program. Sliced program is also calledthe simple slicing or static slicing of a program.

```
(a) Void main()
(b) {
(c) inti=0;
(d) int n;
(e) int sum=0;
(f) printf("enter the value of n");
(g) scanf("%d",&n);
(h)     for(i=0; i<n; i++)
(i)     {
(j)   sum=sum+i;
(k) printf("the current value of i is",i);
```

```
(l)     }
(m) printf("the sum is",sum);
(n) getch();
(o) }
```

Program: Simple Program

This program consists of three variables. Sohere we have three variables which can remove. Now we describe all the three conditions that satisfy thecriteria[20]. After removing one by one each variable which program remains that is called sliced program or static sliced program.

Condition 1: Remove variable i

```
(a) Void main()
(b) {
(c)
(d) int n;
(e) int sum=0;
(f) printf("enter the value of n");
(g) scanf("%d",&n);
(h)
(i)
(j)
(k)
(l)
(m) printf("the sum is",sum);
(n) getch();
(o) }
```

Program: Static Sliced Program

Condition 2: Remove variable n

```
(a) Void main()
(b) {
(c) inti=0;
(d)
(e) int sum=0;
(f) printf("enter the value of n");
(g)
(h)
(i)     {
(j)         sum=sum+i;
(k) printf("the current value of i is",i);
(l)     }
(m) printf("the sum is",sum);
(n) getch();
(o) }
```

Program: Static Sliced Program

Condition 3: Remove variable sum

```
(a) Void main()
(b) {
(c) inti=0;
(d) int n;
(e)
(f) printf("enter the value of n");
(g) scanf("%d",&n);
(h)     for(i=0; i<n; i++)
(i)     {
(j)
(k) printf("the current value of i is",i);
```

```
(l)     }
(m)
(n)  getch();
(o)  }
```

Program: Static Sliced Program

Now after understanding the static slicing concept we describe a static slicing algorithm which is clearly understandable and gives the solution of simple program to the static sliced program. This algorithm is very simple and gives the precise solution of any problem. In order to extract a slice from a program, the dependencies between the statements must be computed first. The control flow graph (CFG) is a data structure which makes the control dependencies for each operation in a program explicit [3][4].

2.1 Algorithm

```
P: Simple Program
Ch: variable which we want to remove
Static_Slice(p,ch)
{
int l;
str s1=" ";
```

1. Open program P in read mode
2. Fetch the input variable Ch
3. Find l i.e. length of program P
4. while(!EOF)
5. {
6. for(i=0;i<=l;i++)
7. {
8. if(Ch==getc(p))
9. {
10. Delete the variable;
11. }
12. else
13. {
14. s1=s1+Ch;
15. }
16. }
17. }

III. Conclusion

This paper, presents general program slicing algorithms. The algorithms compute correctprogram slices for all language constructs found in major object-oriented programming languages, e.g., polymorphism, inheritance, late binding, exception handling, local and global variables. This paper provides the general static slicing algorithm computes correct and executable static slices.Static slicing methods have been proposed for maintenance and program understanding because this way certainparts of the program can be "sliced away" that are of nointerest with respect to the slicing criterion.

The application of slicing in various areas likes debugging, cohesion measurement, comprehension, maintenance and re-engineering and testing are highlighted.In this paper, we have proposed and validated a technique, which is an extension of an existing technique proposed by us in an analogous study. On carefullyanalyzing the code the behavior of techniques we observed that proposed technique gives better result.

IV. Future Scope

As part of our future work, we will expand ourframework to extend the algorithmic support within ourstructure and derive new slicing related concepts, as wellas new visualization techniques. We also plan to integrateforward slicing algorithm and backward slicing algorithm within our static slicingframework.

In future we plan to make a more robust implementation of the algorithms and to perform some measurementson real, industrial applications with real case studies. Forthis purpose we need to make the implementation more stable and improve the performance, especially regarding theslowdown of the instrumented code. Another on-going work is to implement the algorithm for the C++ language.

References

[1] Tao Wang and Abhik Roy Choudhury," Dynamic Slicing on Java Bytecode Traces",Singapore International Conference on Software Engineering (ICSE), 2004

[2] Hiralalagarwal,richard a. demillo and eugene h. spafford," Debugging with Dynamic Slicing andBacktracking", Software— Practice And Experience, Vol. 23, no. 6, pp. 589–616, JUNE 1993

[3] SwarnenduBiswas and Rajib Mall, "Regression Test Selection Techniques: A Survey", Information and Software Technology, Vol. 52, no. 1, January 2010

[4] Jaiprakash T Lalchandani, R Mall, "Regression TestingBased-on Slicing of Component-based SoftwareArchitectures", ISEC ,vol. 79, no. 06, pp. 19-22, 2008

[5] Rajiv Gupta, Mary Jean Harrold, Mary Lou Soffa, "AnApproach to Regression Testing using Slicing", ACMTransactions on Programming Languages and Systems,vol. 12, no. 1, pp. 26-60, January 1990

[6] Yogesh Singh, ArvinderKaur and BhartiSuri, "A Hybrid Approach for Regression Testingin Interprocedural Program", Journal of Information Processing Systems, Vol.6, No.1, March 2010

[7] Hongchang Zhang, Shujuan Jiang, Rong Jin, "An Improved Static Program Slicing Algorithm Using Stack Trace",IEEE,2011

[8] N.Sasirekha, A.Edwin Robert, Dr.M.Hemalatha,"Program slicing techniques and its applications",International Journal of Software Engineering &Applications (IJSEA), Vol. 2, No. 3, pp.85-92, July 2011

[9] MithunAcharya, Brian Robinson, "Practical ChangeImpact Analysis Based on Static Program Slicing forIndustrial Software Systems", ICSE, vol. 11, pp. 21–28,may 2011

[10] BaowenXu, JuQian, Xiaofang Zhang, Zhongqiang, WuLin Chen," A Brief Survey Of Program Slicing", ACMSIGSOFT Software Engineering, Vol. 30, no. 2, pp. 1-36, March 2005

[11] Josep Silva, "A Vocabulary of Program Slicing-BasedTechniques", ACM Computing Surveys, Vol. 44, No. 3,Article 12, June 2012

[12] S. Horwitz, T. Reps, and D. Binkley, "InterproceduralSlicing Using Dependence Graphs", ACM Transactionon Programming Languages and Systems, 1990, pp. 26-61

[13] SwateeRekhaMohanty,Durga Prasad Mohapatra, HimansuSekharBehara, "A Novel Approach for Static Slicing of Inter-Procedural Programs",9th International Conference on Information Technology (ICIT'06), 2006

[14] Frank Tip, "A Survey of Program Slicing Techniques",Journal of Programming Languages, Vol. 3, No. 3, pp.121– 189

[15] David Binkley, "The Application of Program Slicing toRegression Testing"

[16] DebasisMohapatra, "GA Based Test Case GenerationApproach for Formation of Efficient Set of DynamicSlices", International Journal on Computer Science andEngineering (IJCSE),Vol. 3, No. 9, September 2011

[17] AmoghKatti, SujathaTerdal, "Program Slicing forRefactoring: Static Slicer using Dynamic Analyser", International Journal of Computer Applications, Vol. 9,No. 6, November 2010

[18] HiralalAgrawal, Joseph R. Horgan, "Dynamic ProgramSlicing", ACM SIGPLAN Notices, Vol. 25, No. 6, pp.246-256, June 1990

[19] Z. Chen, B. Xu, and J. Zhao, "An Overview of MethodsforDependence Analysis of Concurrent Programs", ACMSIGPLAN Notices, Vol. 37, No. 8, pp. 45-52, 2002

[20] Lei Xu, BaowenXu, Zhenqiang Chen, Jixiang Jiang, andHuowang Chen, "Regression testing for web applicationsbased on slicing. In Proceedings of the 27th AnnualInternational Computer Software and ApplicationsConference",IEEE Computer Society, pages 652–656,Los Alamitos, CA, USA, November 2003

[21] J. Bible, G. Rothermel, and D. Rosenblum, "Acomparative study of coarse- and fine-grained saferegression test-selection techniques", ACM Transactionson Software Engineering and Methodology, Vol. 10, No.2, pp. 149–183, April 2001

[22] S.S. Anju, P. Harmya, NoopaJagadeesh, R. darsana,"Malware detection using assembly code and controlflow graph optimization", ACM Digital library, No. 52,2010

[23] J.Zhao, J.Cheng, and K.Ushijima, "Static Slicing ofConcurrent Object-Oriented Programs", Proc. of the 20thIEEE Annual International Computer Software andApplications conference, IEEE, pp.312-320, August, 1996