William Montgomery
CS 442 - Spring 2014
Homework 1

Problem Statement

The goal of this project was to implement intraprocedural data-flow analysis using the Soot framework. Five programs were generated using the RugRat program generator. The same settings were used to generate all generated code except for the “totalLOC “ key was changed to generate differing lines of code. The effect of this was that the generated code had more lines of code per method when the the total lines of code was increased.

The programs were analyzed, and the def-use, simple, prime, and round-trip paths were output. The maximum memory utilization was recorded for each run, as was the elapsed execution time. JUnit tests were then generated using zzz, and the test coverage was measure using EclEmma. Software metrics were then generated using CodePro.

Archive Contents

Project Report

1. Generated_Code_Analysis_wmontg2.pdf - This document

Raw Output

1. *output500.txt* - Output from running against 500 LOC
2. *output1000.txt* - Output from running against 1000 LOC
3. *output1500.txt* - Output from running against 2000 LOC
4. *output2000.txt* - Output from running against 3000 LOC
5. *output2500.txt* - Output from running against 2500 LOC

Generated Programs

1. Generated500.zip - The source code with ~500 LOC
2. Generated1000.zip - The source code with ~1000 LOC
3. Generated1500.zip - The source code with ~1500 LOC
4. Generated2000.zip - The source code with ~2000 LOC
5. Generated2500.zip - The source code with ~2500 LOC

Program Analysis Code

1. code.zip - The code used to analyze the Java code

Installation Instructions

This project is based on the installation that Guru Devanla provided, with significant modifications to the skeleton code that he provided to support the simple, prime, and round-trip paths.

1. Import the project into your favorite editor
2. Change the edu.uic.Settings.java file:
    a. applicationFolder = points to the Java source directory to be analyzed
    b. classPathNeedBySoot = depends on the system

3. Run as a JUnit test.

Hardware Configuration

All work was done on the following system:

Class	Description
system	Galago UltraPro (Not Applicable)
bus	Galago UltraPro
memory	64KiB BIOS
processor	Intel(R) Core(TM) i7-4750HQ CPU @ 2.00GHz
memory	1MiB L2 cache
memory	256KiB L1 cache
memory	6MiB L3 cache
memory	16GiB System Memory
memory	8GiB SODIMM DDR3 Synchronous 1600 MHz (0.6 ns)
memory	8GiB SODIMM DDR3 Synchronous 1600 MHz (0.6 ns)
bridge	Crystal Well DRAM Controller
display	Crystal Well Integrated Graphics Controller
multimedia	Crystal Well HD Audio Controller
bus	8 Series/C220 Series Chipset Family USB xHCI
communication	8 Series/C220 Series Chipset Family MEI Controller #1
network	Ethernet Connection I217-V
bus	8 Series/C220 Series Chipset Family USB EHCI #2
multimedia	8 Series/C220 Series Chipset High Definition Audio Controller
bridge	8 Series/C220 Series Chipset Family PCI Express Root Port #1
bridge	8 Series/C220 Series Chipset Family PCI Express Root Port #2
generic	RTS5229 PCI Express Card Reader
bridge	8 Series/C220 Series Chipset Family PCI Express Root Port #4
network	AR9462 Wireless Network Adapter
bus	8 Series/C220 Series Chipset Family USB EHCI #1
bridge	HM87 Express LPC Controller
storage	8 Series/C220 Series Chipset Family 6-port SATA Controller 1 [AHC
bus	8 Series/C220 Series Chipset Family SMBus Controller
scsi0	storage
disk	1TB TOSHIBA MQ01ABD1
volume	931GiB EXT4 volume
scsi1	storage
disk	240GB Crucial_CT240M50
volume	1023KiB BIOS Boot partition
volume	219GiB EXT4 volume
volume	4095MiB swap partition

William Montgomery
CS 442 - Spring 2014
Homework 1

Program Metrics

Generated500

EclEmma Metrics

Element	Coverage	Covered Instructions	Missed Instructions
Generated500	99.8%	26,014	48	26,062

CodePro Metrics

Metric results for src for Generated500 at 3/28/14 11:15 PM

Abstractness 21.2%
Average Block Depth 0.95
  minimum 0
  maximum 1
Average Cyclomatic Complexity 1.00
  minimum 1
  maximum 1
Average Lines Of Code Per Method 4.85
  minimum 1
  maximum 50
Average Number of Constructors Per Type 0.00
  minimum 0
  maximum 0
Average Number of Fields Per Type 2.23
  minimum 0
  maximum 10
Average Number of Methods Per Type 29.12
  minimum 1
  maximum 49
Average Number of Parameters 5.18
  minimum 1
  maximum 10
Comments Ratio 0%
Efferent Couplings 44
Lines of Code "6,909"
Number of Characters "503,626"
Number of Comments 0
  end-of-line 0
  multi-line 0
  javadoc 0
Number of Constructors 0

public	0	
protected	0	
package	0	
private	0	
Number of Fields		105
instance	38	
public	0	
protected	0	
package	38	
private	0	
static	67	
public	0	
protected	0	
package	47	
private	20	
public	0	
instance	0	
static	0	
protected	0	
instance	0	
static	0	
package	85	
instance	38	
static	47	
private	20	
instance	0	
static	20	
Number of Lines		"9,723"
Number of Methods		"1,369"
instance	703	
public	703	
protected	0	
package	0	
private	0	
static	666	
public	666	
protected	0	
package	0	
private	0	
public	1369	
instance	703	
static	666	
protected	0	

- instance 0
- static 0
- package 0
  - instance 0
  - static 0
- private 0
  - instance 0
  - static 0

Number of Packages 7
- compilation units 47
  - minimum 0
  - average 6
  - maximum 47
- class files 0
  - minimum 0
  - average 0
  - maximum 0

Number of Semicolons "4,117"
Number of Types 47
- interface 10
  - public 10
  - protected 0
  - package 0
  - private 0
- class 37
  - public 37
  - protected 0
  - package 0
  - private 0
- public 47
  - interface 10
  - class 37
- protected 0
  - interface 0
  - class 0
- package 0
  - interface 0
  - class 0
- private 0
  - interface 0
  - class 0

Weighted Methods "1,369"

Generated1000

EclEmma Metrics

Element	Coverage	Covered Instructions	Missed Instructions
Generated1000	1.4 %	276	19,124	19,400

CodePro Metrics

Metric results for com.accenture.lab.carfast.test at 3/28/14 11:02 PM

Abstractness 21.2%
Average Block Depth 1.63
  minimum 0
  maximum 7
Average Cyclomatic Complexity 2.54
  minimum 1
  maximum 14
Average Lines Of Code Per Method 9.07
  minimum 1
  maximum 47
Average Number of Constructors Per Type 0.00
  minimum 0
  maximum 0
Average Number of Fields Per Type 2.46
  minimum 0
  maximum 10
Average Number of Methods Per Type 9.89
  minimum 1
  maximum 19
Average Number of Parameters 5.12
  minimum 1
  maximum 10
Comments Ratio 0%
Efferent Couplings 44
Lines of Code "4,489"
Number of Characters "258,458"
Number of Comments 0
  end-of-line 0
  multi-line 0
  javadoc 0
Number of Constructors 0
  public 0
  protected 0
  package 0

private	0
Number of Fields	116
instance	45
public	0
protected	0
package	45
private	0
static	71
public	0
protected	0
package	51
private	20
public	0
instance	0
static	0
protected	0
instance	0
static	0
package	96
instance	45
static	51
private	20
instance	0
static	20
Number of Lines	"5,528"
Number of Methods	465
instance	212
public	212
protected	0
package	0
private	0
static	253
public	253
protected	0
package	0
private	0
public	465
instance	212
static	253
protected	0
instance	0
static	0
package	0

- instance 0
- static 0
- private 0
  - instance 0
  - static 0

Number of Packages 1

- compilation units 47
  - minimum 47
  - average 47
  - maximum 47
- class files 0
  - minimum 0
  - average 0
  - maximum 0

Number of Semicolons "2,498"

Number of Types 47

- interface 10
  - public 10
  - protected 0
  - package 0
  - private 0
- class 37
  - public 37
  - protected 0
  - package 0
  - private 0
- public 47
  - interface 10
  - class 37
- protected 0
  - interface 0
  - class 0
- package 0
  - interface 0
  - class 0
- private 0
  - interface 0
  - class 0

Weighted Methods "1,182"

Generated1500

EclEmma Metrics

Element	Coverage	Covered Instructions	Missed Instructions
Generated1500	81.1 %	14,435	3,363

CodePro Metrics

Metric results for src for Generated1500 at 3/28/14 11:10 PM

Abstractness 21.2%
Average Block Depth 1.25
  minimum 0
  maximum 7
Average Cyclomatic Complexity 2.07
  minimum 1
  maximum 23
Average Lines Of Code Per Method 7.90
  minimum 1
  maximum 47
Average Number of Constructors Per Type 0.00
  minimum 0
  maximum 0
Average Number of Fields Per Type 2.31
  minimum 0
  maximum 10
Average Number of Methods Per Type 10.59
  minimum 1
  maximum 36
Average Number of Parameters 5.03
  minimum 1
  maximum 10
Comments Ratio 0%
Efferent Couplings 44
Lines of Code "4,150"
Number of Characters "259,530"
Number of Comments 0
  end-of-line 0
  multi-line 0
  javadoc 0
Number of Constructors 0
  public 0
  protected 0
  package 0

private	0
Number of Fields	109
instance	47
public	0
protected	0
package	47
private	0
static	62
public	0
protected	0
package	42
private	20
public	0
instance	0
static	0
protected	0
instance	0
static	0
package	89
instance	47
static	42
private	20
instance	0
static	20
Number of Lines	"5,296"
Number of Methods	498
instance	229
public	229
protected	0
package	0
private	0
static	269
public	269
protected	0
package	0
private	0
public	498
instance	229
static	269
protected	0
instance	0
static	0
package	0

instance	0
static	0
private	0
instance	0
static	0
Number of Packages	1
compilation units	47
minimum	47
average	47
maximum	47
class files	0
minimum	0
average	0
maximum	0
Number of Semicolons	"2,356"
Number of Types	47
interface	10
public	10
protected	0
package	0
private	0
class	37
public	37
protected	0
package	0
private	0
public	47
interface	10
class	37
protected	0
interface	0
class	0
package	0
interface	0
class	0
private	0
interface	0
class	0
Weighted Methods	"1,034"

Generated2000

EclEmma Metrics

Element	Coverage	Covered Instructions	Missed Instructions
Generated2000	67.4 %	38,219	18,504

CodePro Metrics

Metric results for src for Generated2000 at 3/28/14 11:12 PM

Abstractness 21.2%
Average Block Depth 2.37
  minimum 0
  maximum 7
Average Cyclomatic Complexity 3.81
  minimum 1
  maximum 19
Average Lines Of Code Per Method 12.46
  minimum 1
  maximum 47
Average Number of Constructors Per Type 0.00
  minimum 0
  maximum 0
Average Number of Fields Per Type 2.29
  minimum 0
  maximum 10
Average Number of Methods Per Type 20.38
  minimum 1
  maximum 29
Average Number of Parameters 5.24
  minimum 1
  maximum 10
Comments Ratio 0%
Efferent Couplings 44
Lines of Code "12,151"
Number of Characters "703,512"
Number of Comments 0
  end-of-line 0
  multi-line 0
  javadoc 0
Number of Constructors 0
  public 0
  protected 0
  package 0

private	0
Number of Fields	108
instance	47
public	0
protected	0
package	47
private	0
static	61
public	0
protected	0
package	41
private	20
public	0
instance	0
static	0
protected	0
instance	0
static	0
package	88
instance	47
static	41
private	20
instance	0
static	20
Number of Lines	"14,345"
Number of Methods	958
instance	512
public	512
protected	0
package	0
private	0
static	446
public	446
protected	0
package	0
private	0
public	958
instance	512
static	446
protected	0
instance	0
static	0
package	0

- instance 0
- static 0
- private 0
  - instance 0
  - static 0

Number of Packages 1

- compilation units 47
  - minimum 47
  - average 47
  - maximum 47
- class files 0
  - minimum 0
  - average 0
  - maximum 0

Number of Semicolons "6,431"

Number of Types 47

- interface 10
  - public 10
  - protected 0
  - package 0
  - private 0
- class 37
  - public 37
  - protected 0
  - package 0
  - private 0
- public 47
  - interface 10
  - class 37
- protected 0
  - interface 0
  - class 0
- package 0
  - interface 0
  - class 0
- private 0
  - interface 0
  - class 0

Weighted Methods "3,654"

Generated2500

EclEmma Metrics

Element	Coverage	Covered Instructions	Missed Instructions
Generated2500	71.5 %	45,126	17,9451

CodePro Metrics

Metric results for src for Generated2500 at 3/28/14 11:13 PM

Abstractness 21.2%
Average Block Depth 2.42
  minimum 0
  maximum 7
Average Cyclomatic Complexity 3.70
  minimum 1
  maximum 30
Average Lines Of Code Per Method 12.13
  minimum 1
  maximum 47
Average Number of Constructors Per Type 0.00
  minimum 0
  maximum 0
Average Number of Fields Per Type 2.27
  minimum 0
  maximum 10
Average Number of Methods Per Type 23.06
  minimum 1
  maximum 34
Average Number of Parameters 5.16
  minimum 1
  maximum 10
Comments Ratio 0%
Efferent Couplings 45
Lines of Code "13,368"
Number of Characters "789,257"
Number of Comments 0
  end-of-line 0
  multi-line 0
  javadoc 0
Number of Constructors 0
  public 0
  protected 0

package	0
private	0
Number of Fields	107
instance	39
public	0
protected	0
package	39
private	0
static	68
public	0
protected	0
package	48
private	20
public	0
instance	0
static	0
protected	0
instance	0
static	0
package	87
instance	39
static	48
private	20
instance	0
static	20
Number of Lines	"15,797"
Number of Methods	"1,084"
instance	545
public	545
protected	0
package	0
private	0
static	539
public	539
protected	0
package	0
private	0
public	1084
instance	545
static	539
protected	0
instance	0
static	0

package	0
instance	0
static	0
private	0
instance	0
static	0
Number of Packages	1
compilation units	47
minimum	47
average	47
maximum	47
class files	0
minimum	0
average	0
maximum	0
Number of Semicolons	"7,078"
Number of Types	47
interface	10
public	10
protected	0
package	0
private	0
class	37
public	37
protected	0
package	0
private	0
public	47
interface	10
class	37
protected	0
interface	0
class	0
package	0
interface	0
class	0
private	0
interface	0
class	0
Weighted Methods	"4,021"

William Montgomery
CS 442 - Spring 2014
Homework 1

Analysis Metrics

Lines of Code	Time to Complete	Maximum Memory Usage
500	13 seconds	429 MB
1000	45 seconds	295 MB
1500	5 minutes 13 seconds	437 MB
2000	6 minutes 19 seconds	542 MB
2500	23 minutes 5 seconds	736 MB

Results

Working with Soot is like wrestling a drunken bear. The documentation is lacking and tutorials are generally more confusing than helpful. The mailing list seems full of half answers, so even getting a basic analysis working is challenging. Despite this, I was able to generate all the required functionality, although my dignity suffered along the way.

It is obvious from the time required to complete the analyses, that the time required is superlinear and probably exponential. This makes logical sense, as more lines of code means more nodes in the control flow graph. Improvements in the generation of prime paths could improve this, as the calculation of subpaths was time consuming and a brute force method was used.