Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STDO

CB-FF

IG-PR-IOOCC

Conclusions

Automatic Test Data Generation using Constraint Programming and Search Based Software Engineering Techniques

Abdelilah Sakti

Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Department of Computer and Software Engineering École Polytechnique de Montréal, Québec, Canada

PhD Dissertation Defense December 16^{th} , 2014, Montréal QC







Pattern Trace Identification, Detection, and Enhancement in Java SOftware Cost-effective Change and Evolution Research Lab

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions

CPA-S2751

Motivation

Every year inadequate infrastructure software costs the U.S. economy around \$60 billion.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing Goal

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Motivation

Every year inadequate infrastructure software costs the U.S. economy around \$60 billion.



Between 1985 and 1987 : the radiotherapy machine Therac-25 sent to patients a X-ray dose 100 times greater than expected.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Motivation .

Every year inadequate infrastructure software costs the U.S. economy around \$60 billion.



Between 1985 and 1987 : the radiotherapy machine Therac-25 sent to patients a X-ray dose 100 times greater than expected.

- At least five deaths;
- Several other patients were severely affected by radiation.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing Strategies

White-box Testing Goal

Guai

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Motivation

Every year inadequate infrastructure software costs the U.S. economy around \$60 billion.



Between 1985 and 1987 : the radiotherapy machine Therac-25 sent to patients a X-ray dose 100 times greater than expected.

- At least five deaths;
- Several other patients were severely affected by radiation.

Only a high quality software can reduce errors costs.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions

CPA-S3 7 51

Software testing

Expensive

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions

CPA-S3 7 51

Software testing

Expensive

▶ Costs more than 50% of the budget of critical software;

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions

CPA-S3751

Software testing

Expensive

▶ Costs more than 50% of the budget of critical software;

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions

CPA-ST 7 51

Software testing

Expensive

▶ Costs more than 50% of the budget of critical software;

Difficult

Exhaustive testing!

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions

CPA-ST 7 51

Software testing

Expensive

▶ Costs more than 50% of the budget of critical software;

Difficult

Exhaustive testing! is not possible

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions

CPA-ST 7 51

Software testing

Expensive

▶ Costs more than 50% of the budget of critical software;

- Exhaustive testing! is not possible
- Random testing!

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions

CPA-ST 7 51

Software testing

Expensive

▶ Costs more than 50% of the budget of critical software;

- Exhaustive testing! is not possible
- Random testing! is insufficient

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions

CPA-ST 7 51

Software testing

Expensive

► Costs more than 50% of the budget of critical software;

- Exhaustive testing! is not possible
- Random testing! is insufficient
- Selecting and generating an adequate subset of test data!

Introduction

Motivation

- Software testing
- Strategies
- White-box Testing
- Goal
- Requirements
- Test-Data Generation for Unit Class Testing
- Test-Data Generation for
- Unit Class Testing
- Test-Data Format of OOP
- Automation
- Search Based
- Genetic Algorithms
- Constraint Based
- Symbolic Execution
- SB-STDG versus CB-STDG
- Thesis
- Contributions

CPA-ST 7 51

Software testing

Expensive

▶ Costs more than 50% of the budget of critical software;

- Exhaustive testing! is not possible
- Random testing! is insufficient
- Selecting and generating an adequate subset of test data! is hard

Introduction

Motivation

- Software testing
- Strategies
- White-box Testing
- Goal
- Requirements
- Test-Data Generation for Unit Class Testing
- Test-Data Generation for
- Unit Class Testing
- Test-Data Format of OOP
- Automation
- Search Based
- Genetic Algorithms
- Constraint Based
- Symbolic Execution
- SB-STDG versus CB-STDG
- Thesis
- Contributions



Software testing

Expensive

▶ Costs more than 50% of the budget of critical software;

Difficult

- Exhaustive testing! is not possible
- Random testing! is insufficient
- Selecting and generating an adequate subset of test data! is hard

Challenge

Automation of test-data generation

Introduction

Motivation

- Software testing
- Strategies
- White-box Testing
- Goal
- Requirements
- Test-Data Generation for Unit Class Testing
- Test-Data Generation for
- Unit Class Testing
- Test-Data Format of OOP
- Automation
- Search Based
- Genetic Algorithms
- Constraint Based
- Symbolic Execution
- SB-STDG versus CB-STDG
- Thesis
- Contributions



Software testing

Expensive

▶ Costs more than 50% of the budget of critical software;

Difficult

- Exhaustive testing! is not possible
- Random testing! is insufficient
- Selecting and generating an adequate subset of test data! is hard

Challenge

Automation of test-data generation can significantly reduce the cost of software

Software testing

Strategies

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Functional Criteria

 Focus on the specification;



Software testing

Strategies

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Functional Criteria	S
 Focus on the specification; 	
(Black-box Testing)	

~

Structural Criteria

 Focus on the internal structure;



Automatic Test Data Generation using Constraint Programming and Search Based Software Engineering Techniques Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc Motivation Software testing Strategies

White-box Testing(Goal)

```
int foo(int x, int y, float z, String s1, String s2) {
 1
       if(y=z)
 2
          if(y>0)
 3
              if(x==10)
 4
                 return s1.length()+s2.length();
 5
 6
       v = x < v:
       x=y+x/y;
 7
       String s=s1+s2;
8
       if((s.equals("OK") && (x>0) && s.length()>x)
 9
           return y/s1.length();
10
       return 0;
11
12
```

White-box Testing Goal Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Automatic Test Data Generation using Constraint Programming and Search Based Software Engineering Techniques Abdelilah Sakti Supervised by Gilles Pesant and

Yann-Gaël

Guéhéneuc

White-box Testing(Goal)

```
int foo(int x, int y, float z, String s1, String s2) {
 1
       if(y=z)
 2
           if(v > 0)
 3
              if(x==10)
 4
                 return s1.length()+s2.length();
                                                       澎
 5
       v= x<<v; (*)
 6
       x = v + x/v:
 7
       String s=s1+s2; (*
8
       if((s.equals("OK") && (x>0) && s.length()>x)
 9
            return y/s1.length(); (*)
10
       return 0;
11
12
```

Unhandled Exceptions

- ▶ Line 5: undefined behavior if *s*1 or *s*2 is null.
- Line 6: undefined behavior if y is negative or >31;
- Line 7: undefined behavior if y = 0;
- ▶ Line 8: undefined behavior if *s*1 or *s*2 is null;
- Line 10: undefined behavior if s1 ="" or is null.

White-box Testing Goal Requirements

Motivation

Strategies

Software testing

Test-Data Generation for Unit Class Testing Test-Data Generation for Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Automatic Test White-box Testing(Requirements) Data Generation using Constraint Programming and Control Flow Graph of foo Search Based Software Engineering Techniques Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc Motivation Software testing Strategies White-box Testing Goal Requirements Test-Data Generation for 驀 Unit Class Testing Test-Data Generation for Unit Class Testing Test-Data Format of OOP 激 Automation Search Based Genetic Algorithms Constraint Based 灪 Symbolic Execution SB-STDG versus CB-STDG Thesis Contributions CPA-S6751

Coverage Criteria

- All-statement;
- All-branch:
- All-paths.

Automatic Test White-box Testing(Requirements) Data Generation using Constraint Programming and Control Flow Graph of foo **Coverage** Criteria Search Based Software Engineering All-statement: Techniques All-branch: Abdelilah Sakti Supervised by All-paths. Gilles Pesant and Yann-Gaël Guéhéneuc Motivation Software testing Strategies White-box Testing Goal Requirements Test-Data Generation for 濝 Unit Class Testing Test-Data Generation for Unit Class Testing Test-Data Format of OOP 憑 Automation Search Based Genetic Algorithms Constraint Based Symbolic Execution SB-STDG versus CB-STDG Thesis Contributions CPA-S6751

Automatic Test White-box Testing(Requirements) Data Generation using Constraint Programming and Control Flow Graph of foo Coverage Criteria Search Based Software All-statement; Engineering Techniques All-branch: Abdelilah Sakti Supervised by All-paths. Gilles Pesant and Yann-Gaël Guéhéneuc Motivation Software testing Strategies White-box Testing Goal Requirements Test-Data Generation for 灪 Unit Class Testing Test-Data Generation for Unit Class Testing Test-Data Format of OOP 激 Automation Search Based Genetic Algorithms Constraint Based 灪 Symbolic Execution SB-STDG versus CB-STDG Thesis Contributions CPA-S6751

Automatic Test White-box Testing(Requirements) Data Generation using Constraint Programming and Control Flow Graph of foo Coverage Criteria Search Based Software All-statement; Engineering Techniques All-branch: Abdelilah Sakti Supervised by ► All-paths. Gilles Pesant and Yann-Gaël Guéhéneuc Motivation Software testing Strategies White-box Testing Goal Requirements Test-Data Generation for 灪 Unit Class Testing Test-Data Generation for Unit Class Testing Test-Data Format of OOP 激 Automation Search Based Genetic Algorithms Constraint Based 灪 Symbolic Execution SB-STDG versus CB-STDG Thesis Contributions CPA-S6751



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Difficulties

- Encapsulation: the state of a class is accessible only through some public methods.
- Abstraction: some classes are not directly instantiable;
- Anonymous Classes: some classes are not visible outside of the method where they are declared;

Test-Data Format of Object-Oriented Programming (OOP) is complex.

Automatic Test Data Generation using Constraint	-	Test-Data Gene
Programming and		
Search Based Software	1	<pre>public class FooClass{</pre>
Engineering	2	private int y;
Techniques	3	private float z;
Abdelilah Sakti	4	<pre>private String s2;</pre>
Supervised by	5	public void setY(int v1
Gilles Pesant and	6	public void set7(float
Yann-Gaël	7	public void setS2(Stri
Gueheneuc	1	public vold sets2(stri
	8	public int roocaller(in
	9	foo(a,str);
Motivation	10	}
Software testing	11	private int foo(int x, S
Strategies White box Tecting	12	if(y=z)
Goal	13	if(v > 0)
Requirements	14	f(y > 0)
Test-Data Generation for	14	II(x==10)
Unit Class Testing	15	return s1.
Test-Data Generation for Unit Class Testing	16	y= x< <y;< td=""></y;<>
Test-Data Format of OOP	17	x=y+x/y; (激)
Automation	18	String s=s1+s2;
Search Based	19	if((s.eguals("OK")
Genetic Algorithms	20	return v/s1 ler
Constraint Based	20	recting of
SP STDC vorus CR STDC	21	return U;
Thesis	22	}
Contributions	23	}

Test-Data Generation for Unit Class Testing

private float z; private String s2; public void setY(int y1) {y=y1}; public void setZ(float z1) {z=z1}; public void setS2(String s) {s2=s}; public int fooCaller(int a, String str){ foo(a,str);... private int foo(int x, String s1){ if(y=z)if(y>0)if(x = 10)濝 return s1.length()+s2.length(); y= x<<y; (激) x=y+x/y; String s=s1+s2; (* if((s.equals("OK") && (x>0) && s.length()>x) return y/s1.length(); () return 0:

CPA-STP 51

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guébéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Test-Data Format of OOP















CPA-STP 51





Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Test-Data Format of OOP



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for Unit Class Testing

Test-Data Format of OOP

Automation Search Based

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Challenge

- 1. Finding an instance of the class under test;
- 2. Finding a sequence of method calls;
- 3. Finding an instance of each required argument.
- The number of sequence of method calls may be unlimited;
- The number of different instances of a given class may be large.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Challenge

- 1. Finding an instance of the class under test;
- 2. Finding a sequence of method calls;
- 3. Finding an instance of each required argument.
- The number of sequence of method calls may be unlimited;
- The number of different instances of a given class may be large.

Pb1: The search space is a problem
Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for Unit Class Testing

Test-Data Format of OOP

Automation

Search Based Genetic Algorithms

Constraint Based

onstraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Automation

Source Code

Instrumentation

Search Based (SB-STDG)

Instrumented Source Code

Meta-heuristic and Fitness Function

Test Data

Constraint Based (CB-STDG)



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Search Based(Genetic Algorithms)



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc 1

2

3

4

5

6

7

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



int foo(int x,int y,float z, String ↔ s1, String s2){ if(y=z)if(y>0)if(x==10)return s1.length()+s2.↔ length();//Target

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing 7

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions







Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing 7

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions







Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing 7

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions







Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing 7

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions







Test candidate	f_{AL}	f_{SE}
i_1	3	
i_2		

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing 7

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions







Test candidate	f_{AL}	f_{SE}
i_1	$3 + \frac{90}{91}$	
i_2		

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing 7

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions







Test candidate	f_{AL}	f_{SE}
i_1	$3 + \frac{90}{91}$	$\frac{90}{91}$
i_2		

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing 7

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions







Test candidate	f_{AL}	f_{SE}
i_1	$3 + \frac{90}{91}$	$\frac{90}{91} + \frac{31}{32}$
i_2		

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing 7

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions







Test candidate	f_{AL}	f_{SE}
i_1	$3 + \frac{90}{91}$	$\frac{90}{91} + \frac{31}{32} + 0$
i_2		

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing 7

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions







Test candidate	f_{AL}	f_{SE}
i_1	$3 + \frac{90}{91} = 3.9890$	$\frac{90}{91} + \frac{31}{32} + 0 = 1.9577$
i_2	$2 + \frac{21}{22} = 2.9545$	$0 + \frac{21}{22} + \frac{20}{21} = 1.9068$

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing 7

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions







Test candidate	f_{AL}	f_{SE}
i_1	$3 + \frac{90}{91} = 3.9890$	$\frac{90}{91} + \frac{31}{32} + 0 = 1.9577$
i_2	$2 + \frac{21}{22} = 2.9545$	$0 + \frac{21}{22} + \frac{20}{21} = 1.9068$

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

5

7

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Search Based





Test candidate	f_{AL}	f_{SE}
i_1	$3 + \frac{90}{91} = 3.9890$	$\frac{90}{91} + \frac{31}{32} + 0 = 1.9577$
i_2	$2 + \frac{21}{22} = 2.9545$	$0 + \frac{21}{22} + \frac{20}{21} = 1.9068$

Pb2: SB-STDG may fail to generate test data to reach Line 5

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Constraint Based(Symbolic Execution)



 $\mathsf{EXP}{=}((\mathsf{S1+S2}).\mathsf{equals}("\mathsf{OK"}) \ \&\& \ ((\mathsf{X}{\ll}\mathsf{Y}){+}\mathsf{X}{/}(\mathsf{X}{\ll}\mathsf{Y}){>}0) \ \&\& \ (\mathsf{S1+S2}).\mathsf{length}(){>}(\mathsf{X}{\ll}\mathsf{Y}){+}\mathsf{X}{/}(\mathsf{X}{\ll}\mathsf{Y}))$

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Constraint Based(Symbolic Execution)



 $\mathsf{EXP}{=}((\mathsf{S1+S2}).\mathsf{equals}("\mathsf{OK"}) \ \& \ ((\mathsf{X}{\ll}\mathsf{Y}){+}\mathsf{X}/(\mathsf{X}{\ll}\mathsf{Y}){>}0) \ \& \ (\mathsf{S1+S2}).\mathsf{length}(){>}(\mathsf{X}{\ll}\mathsf{Y}){+}\mathsf{X}/(\mathsf{X}{\ll}\mathsf{Y}))$

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Constraint Based(Symbolic Execution)



 $\mathsf{EXP}{=}((\mathsf{S1+S2}).\mathsf{equals}("\mathsf{OK"}) \ \&\& \ ((\mathsf{X}{\ll}\mathsf{Y}){+}\mathsf{X}{/}(\mathsf{X}{\ll}\mathsf{Y}){>}0) \ \&\& \ (\mathsf{S1+S2}).\mathsf{length}(){>}(\mathsf{X}{\ll}\mathsf{Y}){+}\mathsf{X}{/}(\mathsf{X}{\ll}\mathsf{Y}))$

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG Thesis

Contributions



Constraint Based(Symbolic Execution)



 $\mathsf{EXP}{=}((\mathsf{S1}{+}\mathsf{S2}).\mathsf{equals}("\mathsf{OK}") \ \& \ ((\mathsf{X}{\ll}\mathsf{Y}){+}\mathsf{X}/(\mathsf{X}{\ll}\mathsf{Y}){>}0) \ \& \ (\mathsf{S1}{+}\mathsf{S2}).\mathsf{length}(){>}(\mathsf{X}{\ll}\mathsf{Y}){+}\mathsf{X}/(\mathsf{X}{\ll}\mathsf{Y}))$

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG Thesis

Contributions



Constraint Based(Symbolic Execution)



 $\mathsf{EXP}{=}((\mathsf{S1}{+}\mathsf{S2}).\mathsf{equals}("\mathsf{OK"}) \&\& ((\mathsf{X}{\ll}\mathsf{Y}){+}\mathsf{X}/(\mathsf{X}{\ll}\mathsf{Y}){>}0) \&\& (\mathsf{S1}{+}\mathsf{S2}).\mathsf{length}(){>}(\mathsf{X}{\ll}\mathsf{Y}){+}\mathsf{X}/(\mathsf{X}{\ll}\mathsf{Y}))$

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for

Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG Thesis Contributions

CPA-STDG51

Constraint Based(Symbolic Execution)



 $\mathsf{EXP}{=}((\mathsf{S1+S2}).\mathsf{equals}("\mathsf{OK}") \ \& \ ((\mathsf{X}{\ll}\mathsf{Y}){+}\mathsf{X}/(\mathsf{X}{\ll}\mathsf{Y}){>}0) \ \& \ (\mathsf{S1+S2}).\mathsf{length}(){>}(\mathsf{X}{\ll}\mathsf{Y}){+}\mathsf{X}/(\mathsf{X}{\ll}\mathsf{Y})) \ (\mathsf{X}{\ll}\mathsf{Y}){+}\mathsf{X}/(\mathsf{X}/\mathsf{Y}){+}\mathsf{X}/(\mathsf{X}){+}\mathsf{X}/(\mathsf{X}){+}\mathsf{X}/(\mathsf{X}){+}\mathsf{X}/(\mathsf{X}){+}\mathsf{X}/(\mathsf{X}){+}\mathsf{X}/(\mathsf{X}){+}\mathsf{X}/(\mathsf{X}){+}\mathsf{X}/(\mathsf{X}){+}\mathsf{X}/(\mathsf{X}){+}\mathsf{X}/(\mathsf{X}){+}\mathsf{X}/(\mathsf{X}){+}\mathsf{X}/(\mathsf{X}){+}\mathsf{X}/(\mathsf{X}){+}\mathsf{X}/(\mathsf{X}){+}\mathsf{X}/(\mathsf{X}){+}\mathsf{X}/(\mathsf{X}){+}\mathsf{X}/(\mathsf{X})){+}\mathsf{X}/(\mathsf{X}){+}\mathsf{X}/(\mathsf{X})){+}\mathsf{X}/(\mathsf{X}$

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing Goal

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for Unit Class Testing

Test-Data Format of OOP

Test-Data Format of O

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



SB-STDG versus CB-STDG

SB-STDG	CB-STDG
 + Scalable; 	 – Not Scalable;
 Incomplete; 	 + Complete;
 Depends on many parameters. 	 + Precision in test data generation.
Derives its advantages from the dynamic analysis.	Derives its advantages from the static analysis.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

- Software testing
- Strategies
- White-box Testing Goal
- Requirements
- Test-Data Generation for Unit Class Testing
- Test-Data Generation for
- Unit Class Testing
- Test-Data Format of OOP
- Automation
- Search Based
- Genetic Algorithms
- Constraint Based
- Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



SB-STDG versus CB-STDG

SB-STDG	CB-STDG
► + Scalable;	 – Not Scalable;
 Incomplete; 	 + Complete;
 Depends on many parameters. 	 + Precision in test data generation.
Derives its advantages from the dynamic analysis.	Derives its advantages from the static analysis.

These approaches use complementary analyses that give them a potential for combination.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Thesis

The Goal

Improving the efficiency and effectiveness of SB-STDG for object-oriented testing and revealing bugs at an earlier stage.

The Way

- Using the main advantages of SB-STDG and CB-SDTG;
- Dynamically analyzing source code to monitor the actual execution;
- Statically analyzing source code to identify and extract relevant information;
- Exploiting relevant information to guide SB-STDG process either directly or through CB-STDG;

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

Motivation

Software testing

Strategies

White-box Testing

Goal

Requirements

Test-Data Generation for Unit Class Testing

Test-Data Generation for Unit Class Testing

Test-Data Format of OOP

Automation

Search Based

Genetic Algorithms

Constraint Based

Symbolic Execution

SB-STDG versus CB-STDG

Thesis

Contributions



Contributions

Combining CB-STDG and SB-STDG

- CPA-STDG, a new CB-STDG approach;
- CSB-STDG, a new approach that combines CPA-STDG and SB-STDG;
- CB-FF, a novel constraint based fitness functions ;

Lightweight Static Analyses to improve SB-STDG

 IG-PR-IOOCC, a new SB-STDG approach for unit-class testing.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDG

Idea of CPA-STDG Constraints Generation Results

CSB-STDG

CB-FF

IG-PR-IOOCC

Conclusions

Constraint Programming Approach for Software Test Data Generation (CPA-STDG)

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDO

Idea of CPA-STDG

Constraints Generation Results

CSB-STDC

CB-FF

IG-PR-IOOCC

Conclusions

CPA-STDG Idea of CPA-STDG

Idea

- Delegating the exploration of the symbolic tree to a CP Solver;
- Creating one CSP for the whole program under test and its CFG.

Advantages

- + Avoid the symbolic evaluation cost;
- + Benefit from the large number of search heuristics that are implanted in a Solver;
- + Avoid exploring infeasible sub-paths;
- + Easy identification of the program structure.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

Idea of CPA-STDG

Constraints Generation

Results

CSB-STD(

CB-FF

IG-PR-IOOCO

Conclusions





Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDO

Idea of CPA-STDG

Constraints Generation

Results

CSB-STD

CB-FF

IG-PR-IOOC

Conclusions





Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDG

Idea of CPA-STDG

Constraints Generation

Results

CSB-STD

CB-FI

IG-PR-IOOC

Conclusions



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDG

Idea of CPA-STDG

Constraints Generation

Results

CSB-STD

CB-FF

IG-PR-IOOCO

Conclusions



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDG

Idea of CPA-STDG

Constraints Generation

Results

CSB-STD

CB-FF

IG-PR-IOOCO

Conclusions

CPA-STDG Constraints Generation



CSP of foo Var int $foo_0, x_0, y_0,$ z_0, max_{0-3} Var int nD_{0-2} in {-1,0,1} **Var** int nS_{0-4} in {0,1} $nS_1 = 1 \Rightarrow max_1 = y_0$ $nS_1 = 0 \Rightarrow max_1 = max_0$

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDG

Idea of CPA-STDG

Constraints Generation

Results

CSB-STD

CB-FF

IG-PR-IOOCO

Conclusions

foo CFG	C.	SP of foo
$ \begin{array}{c} foo_0, x_0, y_0, z_0\\ max_0 = x_0\\ \end{array} $	1 2 3	Var int $foo_0, x_0, y_0, z_0, max_{0-3}$ Var int nD_{0-2} in {-1,0,1} Var int nS_{0-4} in {0,1}
$\begin{array}{c c} max 0 < y 0 \\ \hline \\ max_1 = y_0 \\ \hline \\ max_0 < z_0 \end{array}$	4 5 6 7 8	$nS_1 = 1 \Rightarrow max_1 = y_0$ $nS_1 = 0 \Rightarrow max_1 = max_0$
$\begin{array}{c c} max_1 < z_0 \\ \hline max_3 = z_0 \\ \hline max_2 = z_0 \end{array}$	9 10 11 12 13	$nD_0 = 1 \Rightarrow max_0 < y_0$ $nD_0 = -1 \Rightarrow \neg(max_0 < y_0)$
return max3	14 15 16	

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDG

Idea of CPA-STDG

Constraints Generation

Results

CSB-STD

CB-FF

IG-PR-IOOCO

Conclusions

CPA-STDG Constraints Generation



CSP of foo Var int $foo_0, x_0, y_0,$ z_0 , max_{0-3} Var int nD_{0-2} in {-1,0,1} 2 **Var** int nS_{0-4} in {0,1} 3 $nS_1 = 1 \Rightarrow max_1 = y_0$ 6 $nS_1 = 0 \Rightarrow max_1 = max_0$ 7 9 10 $nD_0 = 1 \Rightarrow max_0 < y_0$ $nD_0 = -1 \Rightarrow \neg(max_0 < y_0)$ 11 12 13 $nD_0 = 1 \Leftrightarrow nS_1 \neq 0$ 14 15 $nD_0 = 1 \Leftrightarrow nD_1 \neq 0$ $nD_0 = -1 \Leftrightarrow nD_2 \neq 0$ 16

CPA-STDG Results

Comparing CPA-STDG to a Dynamic SE approach (PathCrawler)

Program	k-path	# feasible Paths	CPA-STDG		CPA-STDG PathCrawler	
			# T. D.	T. (s)	# T. D.	T. (s)
tri_type	-	10	10	0.001	14	0.010
Sample	3	240	240	0.060	241	0.270
Merge	5	321	321	0.148	337	0.780
Merge	10	20,481	20,481	28.640	20,993	116.000

Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Abdelilah Sakti

Introduction

CPA-STDG

Idea of CPA-STDG

Constraints Generation

Results

CSB-STDG

CB-FF

IG-PR-IOOCC

Conclusions

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guébéneuc

Introduction

CPA-STDG

Idea of CPA-STDG

Constraints Generation

Results

CSB-STDG

CB-F

IG-PR-IOOCO

Conclusions

CPA-STDG

Results

Comparing CPA-STDG to a Dynamic SE approach (PathCrawler)

Program	k-path	# feasible Paths	CPA-STDG		PathCrawler	
			# T. D.	T. (s)	# T. D.	T. (s)
tri_type	-	10	10	0.001	14	0.010
Sample	3	240	240	0.060	241	0.270
Merge	5	321	321	0.148	337	0.780
Merge	10	20,481	20,481	28.640	20,993	116.000

CPA-STDG is more efficient and effective than PathCrawler.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDG

Idea of CPA-STDG

Constraints Generation

Results

CSB-STDG

CB-F

IG-PR-IOOCC

Conclusions

CPA-STDG

Results

Comparing CPA-STDG to a Dynamic SE approach (PathCrawler)

Program	k-path	# feasible Paths	CPA-STDG		PathCrawler	
			# T. D.	T. (s)	# T. D.	T. (s)
tri_type	-	10	10	0.001	14	0.010
Sample	3	240	240	0.060	241	0.270
Merge	5	321	321	0.148	337	0.780
Merge	10	20,481	20,481	28.640	20,993	116.000

CPA-STDG is more efficient and effective than PathCrawler.
Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDG

Idea of CPA-STDG

Constraints Generation

Results

CSB-STDG

CB-F

IG-PR-IOOCO

Conclusions

CPA-STDG

Results

Comparing CPA-STDG to a Dynamic SE approach (PathCrawler)

Program	k-path	# feasible Paths	CPA-STDG		PathCrawler	
			# T. D.	T. (s)	# T. D.	T. (s)
tri_type	-	10	10	0.001	14	0.010
Sample	3	240	240	0.060	241	0.270
Merge	5	321	321	0.148	337	0.780
Merge	10	20,481	20,481	28.640	20,993	116.000

CPA-STDG is more efficient and effective than PathCrawler.

The scalability issue remains an open research question.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STDG

Search Based Software Test Data Generation

The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCC

Conclusions

22 / 51

Constrained Search Based Software Test Data Generation (CSB-STDG).

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STDG

Search Based Software Test Data Generation

The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCC

Conclusions

23 / 51

Search Based Software Test Data Generation



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDO

CSB-STDG

Search Based Software Test Data Generation

The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCC

Conclusions

Search Based Software Test Data Generation



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDO

CSB-STDG

Search Based Software Test Data Generation

The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCC

Conclusions

Search Based Software Test Data Generation





IG-PR-IOOC



IG-PR-IOOCO

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDO

CSB-STDC

Search Based Software Test Data Generation

The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCC

Conclusions

24 / 51

The Idea of CSB-STDG



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDO

CSB-STD0

Search Based Software Test Data Generation

The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCC

Conclusions

24 / 51

The Idea of CSB-STDG



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDO

CSB-STDC

Search Based Software Test Data Generation

The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCC

Conclusions

24 / 51

The Idea of CSB-STDG



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc Introduction CPA-STDG CSB-STDG Sacrh Based Software Test Data Generation The Idea of CSB-STDG

1

2

3

4

5

6

7 8

g

10

11

12

Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCC

Conclusions

Example: Unit Under Test Relaxation

```
int foo(int x,int y , float z, String s1,String s2) {
    if(y== z)
        if(y>0)
            if(x==10)
            return s1.length()+s2.length()
        y= x<<y;
        x=y+x/y;
        String s=s1+s2;
    if( (x>0) && R2>x)
        return y/ s1.length()
    return 0;
}
```

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc Introduction CPA-STDG CSB-STDG Sacrh Based Software Test Data Generation The Idea of CSB-STDG

1

2

3

4

5

6

7 8

g

10

11

12

Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCC

Conclusions

Example: Unit Under Test Relaxation

```
int foo(int x,int y , float z, String s1,String s2) {
    if(y== z)
        if(y>0)
            if(x==10)
            return s1.length()+s2.length()
        y= x<<y;
        x=y+x/y;
        String s=s1+s2;
    if( (x>0) && R2>x)
        return y/ s1.length()
    return 0;
}
```

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc Introduction CPA-STDG

Search Based Software Test Data Generation The Idea of CSB-STDG Example: Unit Under Test Relaxation Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCC

Conclusions

Example: Unit Under Test Relaxation

int foo(int x,int y) {
if(y = R0;)	
if(y>0)	
if(x==10)	
return R1;	
y = x << y;	
x=y+x/y;	
if((x>0) && R2>x)	
return y/R3;	
return 0;	
}	
	<pre>int foo(int x,int y if(y==R0;) if(y>0) if(x==10) return R1; y= x<<y; (x="" if(="" x="y+x/y;">0) && R2>x) return y/R3; return 0; }</y;></pre>

Automatic Test Data Generation Constrained Population Generator(CPG)

Data Generation using Constraint Programming and Search Based Software Engineering Techniques

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDG

CSB-STDG

Search Based Software Test Data Generation

The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCC



Constrained Population Generator(CPG)

Data Generation using Constraint Programming and Search Based Software Engineering Techniques

Automatic Test

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STDG

Search Based Software Test Data Generation

The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCO



Constrained Population Generator(CPG)

Automatic Test Data Generation using Constraint Programming and Search Based Software Engineering Techniques

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STDG

Search Based Software Test Data Generation

The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCO



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STDG

Search Based Software Test Data Generation

The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCC

Conclusions

Constrained Evolution Operator (CEO)



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STDG

Search Based Software Test Data Generation

The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCC

Conclusions

Constrained Evolution Operator (CEO)



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STDG

Search Based Software Test Data Generation

The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects Results

CB-FF

IG-PR-IOOCO

Conclusions

Evaluation

-					
NI 1	h		\sim	tc	
Ju	υı	C	L	ιs	

Subject	# LOC	# Branches
Integer	1244	38
BitSet	755	145
ArithmticUtils	959	102
All	2958	285

analysed approaches

- SB-STDG alone: eToc;
- CB-STDG alone: CPA-STDG;
- CSB-STDG: SB-STDG+CPG; SB-STDG+CEO.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDG

CSB-STDG

Search Based Software Test Data Generation The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects Results

CB-FF

IG-PR-IOOCO

Conclusions

29 / 51

Evaluation

Results



Time (s)

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDG

CSB-STDG

Search Based Software Test Data Generation The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects Results

CB-FF

IG-PR-IOOCC

Conclusions

29 / 51

Evaluation

Results



Time (s)

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDG

CSB-STDG

Search Based Software Test Data Generation The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects Results

CB-FF

IG-PR-IOOCO

Conclusions

29 / 51

Evaluation





Time (s)

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDG

CSB-STDG

Search Based Software Test Data Generation The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects Results

IG-PR-IOOCC

Conclusions

29 / 51

Evaluation







Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STDG

Search Based Software Test Data Generation

The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCC

Conclusions

30 / 51



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STDG

Search Based Software Test Data Generation

The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCC

Conclusions



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STDG

Search Based Software Test Data Generation

The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCC

Conclusions



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STDG

Search Based Software Test Data Generation

The Idea of CSB-STDG

Example: Unit Under Test Relaxation

Constrained Population Generator(CPG)

Constrained Evolution Operator (CEO)

Evaluation

Subjects

Results

CB-FF

IG-PR-IOOCC

Conclusions



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STDC

CB-FF

- The idea of CB-FF Branch-Hardness Metrics
- Branch-Hardness Fitness Functions
- Functions
- Evaluation
- Subjects and configuration
- EA Results
- SA Results

IG-PR-IOOCC

Conclusions

Constraint Based Fitness Function (CB-FF).

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDO

CSB-STD(

CB-FF

The idea of CB-FF

Branch-Hardness Metrics Branch-Hardness Fitness Functions Evaluation

Subjects and configuration

EA Results

SA Results

IG-PR-IOOCC

Conclusions

Prioritizing branches according to how hard it is to

The idea of CB-FF

satisfy them.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STD

CB-FF

The idea of CB-FF

Branch-Hardness Metrics Branch-Hardness Fitness Functions Evaluation Subjects and configuration

EA Results

SA Results

IG-PR-IOOCC

Conclusions

The idea of CB-FF

Prioritizing branches according to how hard it is to satisfy them.

Hardness

 Defining the difficulty to satisfy a constraint in terms of its arity and its projection tightness;

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STD

CB-FF

The idea of CB-FF

Branch-Hardness Metrics Branch-Hardness Fitness Functions Evaluation Subjects and configuration

EA Results

SA Results

IG-PR-IOOCC

Conclusions

The idea of CB-FF

Prioritizing branches according to how hard it is to satisfy them.

Hardness

- Defining the difficulty to satisfy a constraint in terms of its arity and its projection tightness;
 - 1. The lower the arity of the constraint, the less freedom we have to choose some of its variables in order to evolve the test candidate.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STD

CB-FF

The idea of CB-FF

Branch-Hardness Metrics Branch-Hardness Fitness Functions Evaluation Subjects and configuration

EA Results

SA Results

G-PR-IOOCC

Conclusions

Prioritizing branches according to how hard it is to satisfy them.

Hardness

- Defining the difficulty to satisfy a constraint in terms of its arity and its projection tightness;
 - 1. The lower the arity of the constraint, the less freedom we have to choose some of its variables in order to evolve the test candidate.
 - 2. A projection tightness close to 0 will indicate high constrainedness and hardness to satisfy a constraint.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STDC

CB-FF

The idea of CB-FF

Branch-Hardness Metrics

Branch-Hardness Fitness Functions

Evaluation

Subjects and configuration

EA Results

SA Results

IG-PR-IOOCC

Conclusions

Branch-Hardness Metrics

Difficulty Coefficient (DC)

 DC is a possible representation of the hardness of a branch;

►
$$DC(c) = B^2 \cdot \frac{1}{arity_c} + B \cdot (1 - tightness) + 1$$
.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDO

CSB-STD

CB-FF

The idea of CB-FF

Branch-Hardness Metrics

Branch-Hardness Fitness Functions

Evaluation

Subjects and configuration

EA Results

SA Results

IG-PR-IOOCC

Conclusions

Difficulty Coefficient (DC)

 DC is a possible representation of the hardness of a branch;

$$\blacktriangleright DC(c) = B^2 \cdot \frac{1}{arity_c} + B \cdot (1 - tightness) + 1 \ .$$

Difficulty Level (DL)

- DL is based on DC ranking (r);
- DL is a representation of a relative hardness level of a constraint in a set of constraints;

•
$$DL(c,C) = \begin{cases} |C|, & \text{if } r = 0\\ 2^{r-1} \cdot (|C|+1), & \text{if } r > 0 \end{cases}$$

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

The idea of CB-FF

Branch-Hardness Metrics

Branch-Hardness Fitness Functions

Evaluation

Subjects and configuration

EA Results

SA Results

IG-PR-IOOCC

Conclusions

Branch-Hardness Fitness Functions

DC Fitness Function (f_{DC})

- DC is used as a penalty coefficient for breaking a constraint;
- The target of this fitness function is determining a standard-branch-distance.

•
$$f_{DC}(i,C) = \sum_{c \in C} DC(c) \cdot \eta(i,c)$$

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

The idea of CB-FF

Branch-Hardness Metrics Branch-Hardness Fitness

Functions

Evaluation

Subjects and configuration

EA Results

SA Results

IG-PR-IOOCC

Conclusions

DC Fitness Function (f_{DC})

- DC is used as a penalty coefficient for breaking a constraint;
- The target of this fitness function is determining a standard-branch-distance.

•
$$f_{DC}(i,C) = \sum_{c \in C} DC(c) \cdot \eta(i,c)$$
.

DL Fitness Function (f_{DL})

 DL is used as a constant penalty for breaking a constraint in a set of constraints to satisfy;

$$f_{DL}(i,C) = \sum_{c \in C} \ell(i,c) + \eta(i,c),$$
where $\ell(i,c) = \begin{cases} 0, & \text{if } \eta(i,c) = 0 \\ DL(c,C), & \text{if } \eta(i,c) \neq 0 \end{cases}$
Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

The idea of CB-FF Branch-Hardness Metrics Branch-Hardness Fitness Functions

Evaluation

Subjects and configuration

EA Results

SA Results

IG-PR-IOOCC

Conclusions

Evaluation

Subjects and configuration

analyzed meta-heuristics and fitness functions

- Two widely used meta-heuristic algorithms are analyzed: Simulated Annealing (SA) and Evolutionary Algorithm (EA).
- f_{AL} and f_{SE} from the literature;
- Our fitness functions f_{DC} and f_{DL} .

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

The idea of CB-FF Branch-Hardness Metrics Branch-Hardness Fitness Functions

Evaluation

Subjects and configuration

EA Results SA Results

G-PR-IOOCC

Conclusions

Evaluation

Subjects and configuration

analyzed meta-heuristics and fitness functions

- Two widely used meta-heuristic algorithms are analyzed: Simulated Annealing (SA) and Evolutionary Algorithm (EA).
- f_{AL} and f_{SE} from the literature;
- Our fitness functions f_{DC} and f_{DL} .

Subjects

- 440 synthetic test targets that were randomly generated.
- 20 executions for every combination of fitness function and meta-heuristic algorithm.
- If test data was not found after 25,000 (respectively 100,000) fitness evaluations for EA (respectively SA), the search was terminated.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDG

CSB-STDO

CB-FF

The idea of CB-FF Branch-Hardness Metrics Branch-Hardness Fitness Functions

Evaluation

Subjects and configuration

EA Results

SA Results

G-PR-IOOCC

Conclusions



EA Results

EA: Comparing all fitnesses on 440 test targets



Evaluations

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDG

CSB-STD

CB-FF

The idea of CB-FF Branch-Hardness Metrics Branch-Hardness Fitness Functions Evaluation Subjects and configuration

EA Results

SA Results

G-PR-IOOCC

Conclusions



SA Results

Branch Coverage %





Evaluations

CB-FF

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

The idea of CB-FF Branch-Hardness Metrics Branch-Hardness Fitness Functions

Evaluation

Subjects and configuration EA Results

SA Results

G-PR-IOOCC

Conclusions



CB-FF

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

The idea of CB-FF Branch-Hardness Metrics Branch-Hardness Fitness Functions

Evaluation

Subjects and configuration EA Results

SA Results

G-PR-IOOCC

Conclusions



CB-FF

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

The idea of CB-FF Branch-Hardness Metrics Branch-Hardness Fitness Functions

Evaluation

Subjects and configuration EA Results

SA Results

G-PR-IOOCC

Conclusions



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STD

CB-FF

IG-PR-IOOCC

The idea of IG-PR-IOOCC

Example

Evaluation I

Subjects

Results

JTExpert: A Large Scale Experimentation

Exceptions-oriented Test-data Generation

Subjects

Results

Conclusions

39 / 51

Further Static Analyses to Improve SB-STDG (IG-PR-IOOCC).

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

IG-PR-IOOCO

The idea of IG-PR-IOOCC

Example

Evaluation I

Subjects

Results

JTExpert: A Large Scale Experimentation

Exceptions-oriented Test-data Generation

Subjects

Conclusion

IG-PR-IOOCC The idea of IG-PR-IOOCC

Exploring relevant sequences

- Methods that may modify a data member or can reach the test target;
- A static analysis can determine the relevant methods.

Efficiently generate, diversify, and enrich instances of classes

- Instantiate a class with a minimum cost;
- Diversify instances of classes by using different means-of-instantiation;
- Seed instances of classes by using the constants existing in the source code.

Automatic Test Data Generation using Constraint Programming and	IG-PR-IOOCC The idea of IG-PR-IOOCC					
Search Based Software Engineering Techniques						
Abdelilah Sakti Supervised by	• _	Requirements	To analyze	To extract		
Gilles Pesant and Yann-Gaël Guéhéneuc	(I)	A means-of-instantiation of the class under test and each	 Constructors; Factory methods; Data Members; 	Set of means-of-instantiatio		
Introduction		required argument	Derived Classes			
CPA-STDG			(SubClasses).			
CSB-STDG			Data Member ₁	State Modifier1		
CB-FF			Data Membera	State Modifiera		
IG-PR-IOOCC The idea of IG-PR-IOOCC	(11)	A sequence of method calls to make the class under test in a relevant state.	:	State Mountery		
Example			Data Mandan	Charles Mar Il Care		
Evaluation I Subjects			Data Wember _m	State Modifier _m		
Results	-					
JTExpert: A Large Scale Experimentation Exceptions-oriented Test-data Generation	(111	A method that can reach the test target.	Method Under Test	Public Methods caller		
Subjects						
Results						

means-of-instantiation.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CSB-STD

CB-FF

IG-PR-IOOCC

The idea of IG-PR-IOOCC Example Evaluation I Subjects Results JTExpert: A Large Scale Experimentation Exceptions-oriented Test-data Generation Subjects Results

Conclusions

IG-PR-IOOCC

Example

Source Code

1	public class FooClass{
2	private int y;
3	private float z;
4	private String s2;
5	<pre>public void setY(int y1) {y=y1};</pre>
6	<pre>public void setZ(float z1) {z=z1};</pre>
7	<pre>public void setS2(String s) {s2=s};</pre>
8	<pre>public int fooCaller(int a, String str){</pre>
9	
10	foo(a,str);
11	
12	}
13	<pre>private int foo(int x, String s1){</pre>
14	
15	}
16	
17	}

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

IG-PR-IOOCC

The idea of IG-PR-IOOCC Example Evaluation I Subjects Results JTExpert: A Large Scale Experimentation Exceptions-oriented Test-data Generation

16

17

Subjects

Conclusions

IG-PR-IOOCC

Example

Source Code

public class FooClass{ 1 private int y; 2 private float z; 3 private String s2; 4 public void setY(int y1) {y=y1}; 5 public void setZ(float z1) $\{z=z1\};$ 6 public void setS2(String s) {s2=s}; 7 public int fooCaller(int a, String str){ 8 9 foo(a,str); 10 11 12 13 private int foo(int x, String s1){ 14 15

Static Analyses

MIn

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

IG-PR-IOOCO

The idea of IG-PR-IOOCC Example Evaluation I Subjects Results JTExpert: A Large Scale Experimentation Exceptions-oriented Test-data Generation

16

17

Subjects

Results

Conclusions

IG-PR-IOOCC

Example

Source Code

1	public class FooClass{
2	private int y;
3	private float z;
4	private String s2;
5	<pre>public void setY(int y1) {y=y1};</pre>
6	<pre>public void setZ(float z1) {z=z1};</pre>
7	<pre>public void setS2(String s) {s2=s};</pre>
8	<pre>public int fooCaller(int a, String str){</pre>
9	
10	foo(a,str);
11	
12	}
13	<pre>private int foo(int x, String s1){</pre>
14	
15	}

Static Analyses MIn FooClass()

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STE

CB-FF

IG-PR-IOOCO

The idea of IG-PR-IOOCC Example Evaluation 1 Subjects Results JTExpert: A Large Scale Experimentation Exceptions-oriented Test-data Generation

Subjects

Results

Conclusions

17

IG-PR-IOOCC

Example

Source Code

public class FooClass{ 1 private int y; 2 private float z; 3 private String s2; 4 public void setY(int y1) {y=y1}; 5 public void setZ(float z1) $\{z=z1\};$ 6 public void setS2(String s) {s2=s}; 7 public int fooCaller(int a, String str){ 8 9 foo(a,str); 10 11 12 13 private int foo(int x, String s1){ 14 15 16

Static Analyses

MIn

У

FooClass()

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

IG-PR-IOOCO

The idea of IG-PR-IOOCC Example Evaluation I Subjects Results JTExpert: A Large Scale Experimentation Exceptions-oriented Test-data Generation Subjects

17

Results

Conclusions

IG-PR-IOOCC

Example

Source Code

1	public class FooClass{
2	private int y;
3	private float z;
4	private String s2;
5	<pre>public void setY(int y1) {y=y1};</pre>
6	<pre>public void setZ(float z1) {z=z1};</pre>
7	<pre>public void setS2(String s) {s2=s};</pre>
8	<pre>public int fooCaller(int a, String str)</pre>
9	
10	foo(a,str);
11	
12	}
13	<pre>private int foo(int x, String s1){</pre>
14	
15	}
16	



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

IG-PR-IOOCO

The idea of IG-PR-IOOCC Example Evaluation 1 Subjects Results JTExpert: A Large Scale Experimentation Excentions-oriented

16

17

Test-data Generation Subjects

Results

Conclusions

IG-PR-IOOCC

Example

Source Code

public class FooClass{ 1 private int y; 2 private float z; 3 private String s2; 4 public void setY(int y1) {y=y1}; 5 public void setZ(float z1) $\{z=z1\};$ 6 public void setS2(String s) {s2=s}; 7 public int fooCaller(int a, String str){ 8 9 foo(a,str); 10 11 12 13 private int foo(int x, String s1){ 14 15



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STE

CB-FF

IG-PR-IOOCO

The idea of IG-PR-IOOCC

```
Example
Evaluation I
Subjects
Results
JTExpert: A Large Scale
Experimentation
Exceptions-oriented
Test-data Generation
Subjects
```

Results

Conclusions

IG-PR-IOOCC

Example

Source Code

1	public class FooClass{
2	private int y;
3	private float z;
4	private String s2;
5	<pre>public void setY(int y1) {y=y1};</pre>
6	<pre>public void setZ(float z1) {z=z1};</pre>
7	<pre>public void setS2(String s) {s2=s};</pre>
8	<pre>public int fooCaller(int a, String str){</pre>
9	
10	foo(a,str);
11	
12	}
13	<pre>private int foo(int x, String s1){</pre>
14	
15	}
16	
17	}



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

IG-PR-IOOCC

The idea of IG-PR-IOOCC Example Evaluation 1 Subjects Results JTExpert: A Large Scale Experimentation Exceptions-oriented Test-data Generation Subjects

Results

Conclusions

17

IG-PR-IOOCC

Example

Source Code

public class FooClass{ 1 private int y; 2 private float z; 3 private String s2; 4 public void setY(int y1) {y=y1}; 5 public void setZ(float z1) $\{z=z1\};$ 6 public void setS2(String s) {s2=s}; 7 public int fooCaller(int a, String str){ 8 9 foo(a,str); 10 11 12 13 private int foo(int x, String s1){ 14 15 16



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STE

CB-FF

IG-PR-IOOCC

The idea of IG-PR-IOOCC Example Evaluation I Subjects Results JTExpert: A Large Scale Experimentation Exceptions-oriented Test-data Generation

Subjects

Conclusions

17

IG-PR-IOOCC

Example

Source Code

public class FooClass{ 1 private int y; 2 private float z; 3 private String s2; 4 public void setY(int v1) $\{y=y1\}$; 5 public void setZ(float z1) $\{z=z1\};$ 6 public void setS2(String s) {s2=s}; 7 public int fooCaller(int a, String str){ 8 9 foo(a,str); 10 11 12 13 private int foo(int x, String s1){ 14 15 16



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STE

CB-FF

IG-PR-IOOCO

The idea of IG-PR-IOOCC Example Evaluation I Subjects Results JTEsper: A Large Scale Experimentation Exceptions-oriented Test-data Generation Subjects Poeults

Conclusion

42 / 51

IG-PR-IOOCC

Example

Source Code

public class FooClass{ 1 private int y; 2 private float z; 3 private String s2; 4 public void setY(int y1) {y=y1}; 5 public void setZ(float z1) $\{z=z1\};$ 6 public void setS2(String s) {s2=s}; 7 public int fooCaller(int a, String str){ 8 9 foo(a,str); 10 11 12 13 private int foo(int x, String s1){ 14 15 16 17



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STE

CB-FF

IG-PR-IOOCO

The idea of IG-PR-IOOCC Example Evaluation I Subjects Results JTExpert: A Large Scale Exceptions-oriented Test-data Generation Subjects

16

17

Results

Conclusions

IG-PR-IOOCC

Example

Source Code

public class FooClass{ 1 private int y; 2 private float z; 3 private String s2; 4 public void setY(int y1) {y=y1}; 5 public void setZ(float z1) $\{z=z1\};$ 6 public void setS2(String s) {s2=s}; 7 public int fooCaller(int a, String str){ 8 9 foo(a,str); 10 11 12 13 private int foo(int x, String s1){ 14 15



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

IG-PR-IOOCO

The idea of IG-PR-IOOCC

Example

Evaluation I Subjects

Results

JTExpert: A Large Scale Experimentation

Exceptions-oriented Test-data Generation

Subjects

Results

Conclusions

43 / 51

Evaluation I Subjects

Compared systems

- EvoSuite;
- ► JTExpert.
 - Instance Generator;
 - Generator of sequences of method calls.

Libraries	Files	Classes	Methods	Branches	Lines	Instructions
Joda-Time	50	87	1,411	3,052	5,876	25,738
Barbecue	18	18	161	323	1,041	14,558
Commons-lang	5	6	366	1,942	2,134	9,139
Lucene	2	4	58	202	262	1,364
All	75	115	1,998	5,519	9,313	50,799



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

IG-PR-IOOCC

The idea of IG-PR-IOOCC

Example

Evaluation I Subjects

Results

JTExpert: A Large Scale Experimentation

Exceptions-oriented Test-data Generation

Subjects

Results

Conclusions

Results

Branch Coverage %





Search Time (s)



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STDC

CSB-STD

CB-FF

IG-PR-IOOCC

The idea of IG-PR-IOOCC

Example

Evaluation I Subjects

Subjects

Results

JTExpert: A Large Scale Experimentation

Exceptions-oriented Test-data Generation

Subjects

Results

Conclusions

Results

Branch Coverage %





Search Time (s)







```
CPA-STDO
```

CSB-STD(

CB-FF

IG-PR-IOOCC

The idea of IG-PR-IOOCC

Example

Evaluation I

Subjects

Results

JTExpert: A Large Scale Experimentation

Exceptions-oriented Test-data Generation

Subjects

Results

Conclusions

Results

Branch Coverage %





Search Time (s)





```
CPA-STDO
```

```
CSB-STD(
```

```
CB-FF
```

IG-PR-IOOCC



- Example
- Evaluation I
- Subjects

Results

JTExpert: A Large Scale Experimentation

Exceptions-oriented Test-data Generation

Subjects

Results

Conclusions

Results

Branch Coverage %





Search Time (s)

Data Generation using Constraint Programming and Search Based Software Engineering 1 Techniques 2 Abdelilah Sakti 3 Supervised by 4 Gilles Pesant and Yann-Gaël 5 Guéhéneuc 6 7 8 9 10 11 12 13 The idea of IG-PR-IOOCC 14 Example 15 Evaluation I 16 Subjects 17 Results JTExpert: A Large Scale 18 Experimentation 19 Exceptions-oriented Test-data Generation 20 Subjects 21 Reculte 22 23 45 / 51

Automatic Test

JTExpert: A Large Scale Experimentation

Exceptions-oriented Test-data Generation

public class FooClass{ private int y; private float z; private String s2: public void setY(int v1) $\{y=y1\}$; public void setZ(float z1) $\{z=z1\}$; public void setS2(String s) {s2=s}; public int fooCaller(int a, String str){ foo(a.str):... private int foo(int x, String s1){ if(v = z)if(v > 0)if(x==10)濝 return s1.length()+s2.length(); v= x<<v: 激 x = y + x/y;String s=s1+s2: 👗 if((s.equals("OK") && (x>0) && s.length()>x) return y/s1.length(); * return 0;



Automatic Test Data Generation using Constraint Programming and Search Based Software Engineering Techniques Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc The idea of IG-PR-IOOCC Example Evaluation I Subjects Results JTExpert: A Large Scale Experimentation Exceptions-oriented

Test-data Generation

45 / 51

23

Subjects

Reculte

JTExpert: A Large Scale Experimentation

Exceptions-oriented Test-data Generation

public class FooClass{ 1 2 private int y; private float z; 3 private String s2; 4 public void setY(int v1) $\{y=y1\}$; 5 public void setZ(float z1) $\{z=z1\}$; 6 public void setS2(String s) {s2=s}; 7 public int fooCaller(int a, String str){ 8 foo(a.str):... 9 10 private int foo(int x, String s1){ 11 12 if(v = z)if(v > 0)13 if(x = 10)14 return s1.length()+s2.length(); 15 y= x<<y; (▓ 16 x = y + x/y;濝 17 String s=s1+s2; (* 18 if((s.equals("OK") && (x>0) && s.length()>x))19 return y/s1.length(); (*) 20 return 0; 21 22



濛

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

- Introduction
- CPA-STDO
- CSB-STD
- CB-FF

IG-PR-IOOCC

- The idea of IG-PR-IOOCC
- Example
- Evaluation I
- Subjects
- Results
- JTExpert: A Large Scale Experimentation
- Exceptions-oriented Test-data Generation
- Subjects

Results

Conclusions

JTExpert: A Large Scale Experimentation Subjects

The Apache Hadoop framework

The Apache Hadoop framework allows the distributed processing of large data sets across clusters of computers using simple programming models.

- 43 sub-modules;
- 3,545 Java files;
- 7,531 classes;
- 88,971 methods;
- ▶ 148,691 branches;
- 421,032 lines of code;
- 1,698,650 statements.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

IG-PR-IOOCO

The idea of IG-PR-IOOCC

Example

Evaluation I

Subjects

Results

JTExpert: A Large Scale Experimentation

Exceptions-oriented Test-data Generation

Subjects

Results

Conclusions

JTExpert: A Large Scale Experimentation Results

Excontion	Alerted	Alerted in different			
Exception		Lines	Methods	Classes	
NullPointerException	13,663	3,604	3,377	1,343	
ClassCastException	658	213	206	165	
NoClassDefFoundError	602	87	86	86	
IllegalArgumentException	326	104	102	100	
ArrayIndexOutOfBoundsException	296	119	108	71	
Fatal Errors	9	2	2	2	
Other Exceptions	492	193	180	153	
All Exceptions	16,046	4,322	4,061	1,920	

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STD

CB-FF

IG-PR-IOOCO

The idea of IG-PR-IOOCC

Example

Evaluation I

Subjects

Results

JTExpert: A Large Scale Experimentation

Exceptions-oriented Test-data Generation

Subjects

Results

Conclusions

JTExpert: A Large Scale Experimentation Results

Excontion	Alerted	Alerted in different			
Exception		Lines	Methods	Classes	
NullPointerException	13,663	3,604	3,377	1,343	
ClassCastException	658	213	206	165	
NoClassDefFoundError	602	87	86	86	
IllegalArgumentException	326	104	102	100	
ArrayIndexOutOfBoundsException	296	119	108	71	
Fatal Errors	9	2	2	2	
Other Exceptions	492	193	180	153	
All Exceptions	16,046	4,322	4,061	1,920	

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

- Introduction
- CPA-STDO
- CSB-STDC
- CB-FF

IG-PR-IOOCC

- The idea of IG-PR-IOOCC
- Example
- Evaluation I
- Subjects
- Results
- JTExpert: A Large Scale Experimentation
- Exceptions-oriented Test-data Generation
- Subjects
- Results

Conclusions

JTExpert: A Large Scale Experimentation Results

The source of Fatal Errors

- Package: org.apache.hadoop.io;
- Class: WritableComparator;
- Method: compareBytes(byte[] b1, int s1, int l1,byte[] b2, int s2, int l2);
- How to reproduce it : use b1=null or b2=null;
- The root: The class FastByteComparisons uses reflection to get an instance of the private class sun.misc.Unsafe;
- Discussed at: http://comments.gmane.org/gmane. comp.java.hadoop.hbase.devel/39017.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

IG-PR-IOOCC

The idea of IG-PR-IOOCC

Example

Evaluation I

Subjects

Results

JTExpert: A Large Scale Experimentation

Exceptions-oriented Test-data Generation

Subjects

Results

Conclusions

49 / 51

JTExpert: A Large Scale Experimentation

- New instance generator;
- New representation of the test-data generation problem;
- Strategy of revealing bugs;
- JTExpert is available for download at https: //sites.google.com/site/saktiabdel/JTExpert.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STD

CB-FF

IG-PR-IOOCC

Conclusions

Conclusions

CPA-STDG (A. SAKTI et al. JFPC' 11);

 CP is a promising approach for automatic test-data generation.

CSB-STDG (A. SAKTI et al. SSBSE' 12) and CB-FF (A. SAKTI et al. CPAIOR' 13);

- CP is useful for SB-STDG;
- IG-PR-IOOCC (A. SAKTI et al. TSE' 14);
 - Lightweight static analyses is a way to improve SB-STDG.

Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD0

CSB-STD

CB-FF

IG-PR-IOOC

Conclusions

Conclusions

CPA-STDG (A. SAKTI et al. JFPC' 11);

 CP is a promising approach for automatic test-data generation.

CSB-STDG (A. SAKTI et al. SSBSE' 12) and CB-FF (A. SAKTI et al. CPAIOR' 13);

- CP is useful for SB-STDG;
- ► IG-PR-IOOCC (A. SAKTI et al. TSE' 14);
 - Lightweight static analyses is a way to improve SB-STDG.

Future work

- Further experiments;
- Broadening and enlarging internal static analyses;
- Statically analyzing new external sources to derive information that are relevant for test-cases generation.



Abdelilah Sakti Supervised by Gilles Pesant and Yann-Gaël Guéhéneuc

Introduction

CPA-STD

CSB-STD

CB-FF

IG-PR-IOOC

Conclusions

Thank you, Questions?



51 / 51