

# Extra Credit Project: Testing of Input Generation Tools

Helena Bales

CS362-001  
Winter 2017

March 21, 2017

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Selection of Open Source Java Project</b>	<b>4</b>
2.1	Project Details . . . . .	4
2.2	Classes to Test . . . . .	4
<b>3</b>	<b>Selection of Input Generation Tool</b>	<b>4</b>
<b>4</b>	<b>Using the Input Generation Tool</b>	<b>5</b>
4.1	Generated Tests . . . . .	5
4.1.1	Class 1 - Option . . . . .	5
4.1.2	Class 2 - OptionValidator . . . . .	5
4.1.3	Class 3 - GnuParser . . . . .	5
4.1.4	Class 4 - CommandLine . . . . .	6
4.2	Evaluation of Selected Input Generation Tool . . . . .	6
4.2.1	Implementation Details . . . . .	6
4.2.2	Effective Functionality . . . . .	7
4.2.3	Ineffective Functionality . . . . .	7
4.2.4	Tool Use Cases . . . . .	7
4.2.5	Suggested Possible Improvements . . . . .	7
4.3	Coverage of Generated Test Inputs . . . . .	7
<b>5</b>	<b>Fault Insertion</b>	<b>7</b>
<b>6</b>	<b>Conclusion</b>	<b>7</b>

# 1 Introduction

This is my documentation of testing an input generation tool on an Open Source Java project. The project that I have selected is Commons-cli. I will provide a description of this project and a link to the project's webpage below. I will test this project using a test input generation tool called AVMf. Further evaluation of this tool will also be included in this document. I will evaluate the code coverage of these generated tests in order to judge the effectiveness of AVMf. Finally, I will evaluate AVMf by inserting five faults in the commons-cli project and running the tests to see if the faults are caught.

## 2 Selection of Open Source Java Project

The project that I have selected to test is Commons-cli. The project's source code is available at:

---

[https://commons.apache.org/proper/commons-cli/download\\_cli.cgi](https://commons.apache.org/proper/commons-cli/download_cli.cgi)

---

### 2.1 Project Details

The Apache Commons-cli is an API for parsing command line arguments. It is developed and supported by the Apache Foundation. It provides support for many different types of command line arguments. I will be using this project to gain familiarity with the selected Test Generation Tool.

### 2.2 Classes to Test

I will be testing the following four classes:

1. Option
2. OptionValidator
3. GnuParser
4. CommandLine

## 3 Selection of Input Generation Tool

I have selected to use AVMf as my Input Generation Tool to test. AVMf is a Java framework for AVM. It automatically generates unit tests for Java programs. I will evaluate this tool by generating tests for Apache Commons-cli, then evaluating the code coverage of those tests, and inserting faults into the code to see if the tests catch them.

## 4 Using the Input Generation Tool

### 4.1 Generated Tests

#### 4.1.1 Class 1 - Option

Command:

---

```
java -cp target/avmf-1.0-jar-with-dependencies.jar  
org.avmframework.examples.GenerateInputData Option 1T
```

---

Output:

---

```
ERROR: Unable to instantiate test object "Option"  
USAGE: java class org.avmframework.examples.GenerateInputData testobject branch  
[search]  
where:  
- testobject is a test object to generate data for (e.g., "Calendar", "Line" or  
  "Triangle")  
- branch is a branch ID of the form X(T|F) where X is a branching node number (e.g.,  
  "5T")  
- [search] is an optional parameter denoting which search to use (e.g.,  
  "IteratedPatternSearch", "GeometricSearch" or "LatticeSearch")
```

---

#### 4.1.2 Class 2 - OptionValidator

Command:

---

```
java -cp target/avmf-1.0-jar-with-dependencies.jar  
org.avmframework.examples.GenerateInputData OptionValidator 1T
```

---

Output:

---

```
ERROR: Unable to instantiate test object "OptionValidator"  
USAGE: java class org.avmframework.examples.GenerateInputData testobject branch  
[search]  
where:  
- testobject is a test object to generate data for (e.g., "Calendar", "Line" or  
  "Triangle")  
- branch is a branch ID of the form X(T|F) where X is a branching node number (e.g.,  
  "5T")  
- [search] is an optional parameter denoting which search to use (e.g.,  
  "IteratedPatternSearch", "GeometricSearch" or "LatticeSearch")
```

---

#### 4.1.3 Class 3 - GnuParser

Command:

```
java -cp target/avmf-1.0-jar-with-dependencies.jar
    org.avmframework.examples.GenerateInputData GnuParser 1T
```

---

#### Output:

---

```
ERROR: Unable to instantiate test object "GnuParser"
USAGE: java class org.avmframework.examples.GenerateInputData testobject branch
      [search]
where:
- testobject is a test object to generate data for (e.g., "Calendar", "Line" or
  "Triangle")
- branch is a branch ID of the form X(T|F) where X is a branching node number (e.g.,
  "5T")
- [search] is an optional parameter denoting which search to use (e.g.,
  "IteratedPatternSearch", "GeometricSearch" or "LatticeSearch")
```

---

#### 4.1.4 Class 4 - CommandLine

##### Command:

---

```
java -cp target/avmf-1.0-jar-with-dependencies.jar
    org.avmframework.examples.GenerateInputData CommandLine 1T
```

---

#### Output:

---

```
ERROR: Unable to instantiate test object "CommandLine"
USAGE: java class org.avmframework.examples.GenerateInputData testobject branch
      [search]
where:
- testobject is a test object to generate data for (e.g., "Calendar", "Line" or
  "Triangle")
- branch is a branch ID of the form X(T|F) where X is a branching node number (e.g.,
  "5T")
- [search] is an optional parameter denoting which search to use (e.g.,
  "IteratedPatternSearch", "GeometricSearch" or "LatticeSearch")
```

---

## 4.2 Evaluation of Selected Input Generation Tool

I have selected to use AVMf as my Input Generation Tool. AVMf is a framework and Java implementation of the AVM heuristic local search algorithm. The project's source is available on Github.

### 4.2.1 Implementation Details

AVMf is a Java and Maven project. AVMf implements the Alternating Variable Method in a Java framework for use in test generation. AVM is a heuristic local search algorithm and was first applied to test data generation in 1990.

#### **4.2.2 Effective Functionality**

The build process for this tool is very straightforward and easy to use. I was able to follow the Eclipse installation instructions with no issues.

Running the examples was also a very easy process. The instructions for how to set up AVMf and run the examples all flowed together very easily for a smooth introduction to the tool.

#### **4.2.3 Ineffective Functionality**

After running the examples, there is no instruction on where to go from there. It says in the README that the examples show how to apply the GenerateInputData to other classes, but when I try to run AVMf on other classes, it fails every time.

#### **4.2.4 Tool Use Cases**

This tool is effective for use with Maven projects in Eclipse because it is very easy to install using Maven and Eclipse. It is used to generate test data.

#### **4.2.5 Suggested Possible Improvements**

In order to make this tool useable I would suggest adding a walkthrough on how to add another class and generate test data for that added class. I feel that the examples alone, without a walkthrough for adding a new class, do not describe how to use this tool fully.

Since I was not able to get this tool to run on my classes, I am not able to comment on other possible improvements.

### **4.3 Coverage of Generated Test Inputs**

I am unable to evaluate the coverage of the test generated by AVMf since AVMf failed to generate any tests for the selected project and classes.

## **5 Fault Insertion**

I was also not able to test the output of AVMf by inserting faults in the project since there is no output of AVMf to test.

## **6 Conclusion**

I had significant difficulties with this assignment. I attempted to install and use all four of the tools linked to in the assignment statement. After successfully installing AVMf, I read all documentation that I could find on the subject. Most of it was simply repeating the User's Guide, but despite this I was eventually able to figure out where to put my classes from commons-cli to test them. I was

not able to get the tool to run on any of my classes, only on the example classes given in the AVMf source code. Given documentation on how to run AVMf on a new class, other than those provided in the source code, I feel that I would have had no trouble with this project.