JPF - EndlessLoopDetector - Enhancement

EECS 4315: MISSION CRITICAL SYSTEMS
Final Project Report
April 23, 2018

Authors

Varsha Ragavendran

Nisha Sharma

Table of Contents

JPF-EndlessLoopDetector-Enhancement

Introduction

The aim of this project is to enhance the functionality of the existing JPF listener, EndlessLoopDetector. Currently this listener detects if any program is caught in a non-terminating loop based on a threshold value which could be set by a user, if not, it has a default value of 500 set in the IdleFilter class (parent class of the EndlessLoopDetector). When detecting an endless loop, the listener does not provide any valuable information, such as, what causes the program to go into an endless loop (i.e. a line of code in the program or the state where the endless loop occurs). Reviewing the existing documentation for the listener, we noticed that there is no documentation regarding the configuration that allows a user to set this threshold value. Also, there are edge cases where this listener does not detect an endless loop (e.g. infinite recursion, infinite loops, etc.).

Therefore, the purpose of this project is to enhance the EndlessLoopDetector listener class within jpf-core and address the issues mentioned above.

Features

This enhancement of EndlessLoopDetector listener is able to detect and report when a program is caught in an infinite recursion. A new configuration "endlessloopdetector.max_loops" has been added to allow users to set the maximum number of times a method can execute before considering current execution to be non-terminating, thereby an endless loop. The enhanced EndlessLoopDetector listener also provides the configuration details in documentation.

Return analysis has been added to produce more informative report. When the program is detected to be caught in an endless loop, the listener would report the following

- Line of code (with clickable hyperlink)
- Method/instruction Name
- Memory

Deliverables

1. EndlessLoopDetector listener class (Enhancements implemented and documented).
2. Test suite to test the enhanced functionality of the listener (fully documented).
3. README file that describes how the listener can be used and configured.

Limitations

The listener does not produce an informative report when used with "RandomSearch" as the search class, as this class is reported to have bugs and does not seem to work correctly.

Installation and Usage

Directory structure

- jpf-core/src/main/gov/nasa/jpf/listener/ contains the EndlessLoopDetector listener class
- EnhancedModelTesting/src/ contains the about 16 different classes with and without infinite recursion and infinite loops to verify the correctness of proposed enhancement; each class has its own jpf configuration file.

Building

- Download and install jpf following the instructions in "notes.pdf".
- After downloading the "EndlessLoopDetector.java" file, copy its contents and paste it in jpf-core/src/main/gov/nasa/jpf/listener/EndlessLoopDetector.java replacing all the content.
- Build jpf again. Try to verify one of the example .jpf files that come with installation to see that the system is working correctly.

Usage

- Import the "EnhancedModelTesting" java project in the same workspace and configure the classpath to point to place where respective jpf jars are in your system. Modify the classpath elements in the jpf files in the package. Make sure you clean this project to recreate .class files locally for jpf to detect. After this setup is done you may go ahead with running verification on these .jpf file and check if you achieve relevant results.