CS 6156 Fall 2020
Homework 3

1 Deadline

This homework assignment is due on 12/15/2020 at 11:59pm Anywhere on Earth.

2 Goals and Overview

This homework is an exercise in debugging runtime verification violations. You will

- run JavaMOP on provided specifications and record violations;
- inspect one violation and determine whether the violation is a true bug or a false alarm; and
- discuss, based on your experience, if/how RV should be improved to make it easier to inspect violations.

3 Introduction

A serious but often unstated problem in Runtime Verification is the human time cost of inspecting violations that result from executions that do not satisfy the specifications. The goals of this homework are to provide you with experience in inspecting runtime verification violations and to encourage you to think about how to make it easier for real-world developers to inspect violations. Parts of this homework are deliberately underspecified and open-ended.

4 Tasks

4.1 Preliminaries (20 points)

In the Docker container from HW-0:

1. Fetch the JavaMOP specifications required for this homework:

```
$ mkdir hw-3 && cd hw-3
$ wget https://www.cs.cornell.edu/courses/cs6156/2020fa/resources/hw-3-bundle.tgz
$ tar xf hw-3-bundle.tgz
```

2. Make and install a JavaMOP Java agent from the specifications in the `hw-3-bundle` directory.

3. Use your JavaMOP Java agent to monitor a test in `jetty-util`:

```
$ git clone https://github.com/eclipse/jetty.project.git
$ cd jetty.project/jetty-util
$ git checkout df1f709ea2f883ffb7f0a87d63aac506ae3fedd5
$ mvn test -Dtest=QueuedThreadPoolTest#testMaxStopTime -Denforcer.skip
```

4. Put the resulting `violation-counts` file in the `hw-3-bundle` directory.

4.2 Inspect a Violation (40 points)

You should see a violation of `Collection_UnsafeIterator` on line 128 of `QueuedThreadPool.java`.

1. Inspect this violation and explain why it occurred.
2. Describe the steps that you took in conducting your inspection
3. Do you think this violation is a true bug or a false alarm? Why?
4. Put your answers for this task in `hw-3-bundle/inspection.txt`.

HINT: If at first you do not see the violation referred to in the preamble of this task, re-run the test mentioned in Section 4.1 until the violation appears.

4.3 Reflecting on your experience (30 points)

1. Discuss any difficulties that you faced in carrying out the inspection in Section 4.2.
2. Do you agree that manual inspection of violations is a problem that runtime verification researchers should pay more attention to?
3. Discuss three ways in which you think runtime verification researchers can make violation inspection easier for real-world users. Note that a real-world user of runtime verification will not have received the hint that you got in Section 4.2.
4. Put your answers for this task in `hw-3-bundle/discussion.txt`.

4.4 Debriefing (10 points)

Answer the following questions in a file, `hw-3-bundle/debriefing.txt`. These questions are reused from CS 6114 [1].

1. How many hours did you spend on this homework?
2. Would you rate it as easy, moderate, or difficult?
3. How deeply do you feel you understand the material it covers (0%–100%)?
4. If you have any other comments, I would like to hear them! Please write them down or send email to legunsen@cornell.edu

4.5 Deliverable

Submit a single archive file with all the files created and edited during this homework.

```
$ tar czvf hw-3-submission.tgz hw-3-bundle
$ # upload hw-3-submission.tgz to CMS under "Homework 3"
```

References

[1] Nate Foster. *CS 6114 Homework 1*. `https://cornell-pl.github.io/cs6114/homework01.html`. 2019.