CS 6156 Fall 2020
Homework 1

1 Deadline

This homework assignment is due on 11/05/2020 at 11:59pm Anywhere on Earth.

2 Goals and Overview

This homework assignment asks you to reflect on the computational complexity of runtime verification and check your understanding of a monitor synthesis algorithm. You will

- argue for or against statements about the computational complexity of runtime verification; and
- work through a monitor synthesis algorithm.

3 Introduction

A learning outcome of CS 6156 is to provide exposure to theoretical and practical aspects of runtime verification. In this homework, you will investigate the complexity of runtime verification and demonstrate your understanding of monitor synthesis for a specification language that we discussed in class. Parts of this homework are deliberately under-specified and open-ended.

4 Tasks

4.1 Computational Complexity of Runtime Verification (60 points)

In your own words, explain why each of the following statements is true or false. Let P be a safety property.

Statement 1.

The complexity of monitoring P is the functional complexity of function M, where $\mathcal{M} = (S, s_0, M : S \times \Sigma \rightharpoonup S)$ is the "best" monitor for P.

Statement 2.

P is typically infinite, so the complexity of monitoring P should be a function of the size of some finite specification, or representation, of P.

Statement 3.

The complexity of monitoring P is nothing but the complexity of checking, for an input word $w \in \Sigma^\star$, whether $w \in \mathsf{prefixes}(P)$.

Statement 4.

Monitoring P is arbitrarily hard.

You may read slides 33–36 from Lecture 3 [1] or Sections 4.2 and 4.3 in the associated paper [2] prior to providing a response.

4.2 CFG Monitor Synthesis (40 points)

Refer to the journal paper on CFG monitor synthesis [3] in answering this question. That journal paper is an extension of the conference paper that we read in class [4].

Describe the execution of the CFG monitoring algorithm in Figure 6 for the SafeLock property in Section 1.1 with its LALR(1) table in Table 2, on the following observed trace: begin begin acquire acquire release release end begin acquire release end end begin end. Explain only the major steps. The purpose of this exercise is to demonstrate that you understand the CFG monitoring algorithm, including the handling of the $ event.

References

[1] Owolabi Legunsen. *Lecture Slides: Safety Properties and their monitoring*. `https://www.cs.cornell.edu/courses/cs6156/2020fa/resources/Safety-Properties-and-Monitoring.pdf`. 2020.

[2] Grigore Rosu. "On safety properties and their monitoring". In: *Scientific Annals of Computer Science* 22.2 (2012), pages 327–365.

[3] Patrick O'Neil Meredith, Dongyun Jin, Feng Chen, and Grigore Roşu. "Efficient monitoring of parametric context-free patterns". In: *Automated Software Engineering* 17.2 (2010), pages 149–180.

[4] Patrick O'Neil Meredith, Dongyun Jin, Feng Chen, and Grigore Rosu. "Efficient Monitoring of Parametric Context-Free Patterns". In: *International Conference on Automated Software Engineering*. 2008, pages 148–157.