Project - CSE 105 Spring 2022

CSE105Sp22

Part 3 due 6/2/22 at 5pm (no penalty late submission until 8am next day)

The project component of this class will be an opportunity for you to extend your work on assignments and explore applications of your choosing.

Why? To go deeper and explore the material from Theory of Computation and how it relates to other aspects of CS and beyond.

How? During emergency remote instruction last academic year, we discovered that video assessment and some open-ended personalized projects help ensure fairness and can be less stressful for students than in-person midterm exams. Asynchronous project submission also gives flexibility and allows more physical distancing.

Your videos: We will delete all the videos we receive from you after assigning final grades for the course, and they will be stored in a university-controlled Google Drive directory only accessible to the course staff during the quarter. Please send an email to the instructor (minnes@eng.ucsd.edu) if you have concerns about the video / screencast components of this project or cannot complete projects in this style for some reason.

You may produce screencasts with any software you choose. One option is to record yourself with Zoom; a tutorial on how to use Zoom to record a screencast (courtesy of Prof. Joe Politz) is here: Tutorial URL The video that was produced from that recording session in Zoom is here: Video produced in tutorial .

What resources can you use? This project must be completed individually, without any help from other people, including the course staff (other than logistics support if you get stuck with screencast). You can use any of this quarter's CSE 105 offering (notes, readings, class videos, supplementary videos, homework feedback). You may additionally search online to respond to project parts that explicitly ask you to do so, and you must cite all resources (online or offline) that you consult as part of this search. Link directly to the resource and include the name of the author / video creator and the reason you consulted this reference. The work you submit for the project needs to be your own.

The written portion of the project is expected to be clearly legible, and should preferably be typed.

Tasks for Project Part 3

Task 1: Explain a review quiz question (Written)

(a) Select one question from one of the review quizzes from 5/2/22 (Monday of Week 6) to 5/27/22 (Friday of Week 9) to revisit. Include the problem description, why you picked this question (e.g. what is interesting about it, what is hard about it, or why you wanted to take a second look at it), and your solution. Question selection: you can pick any one question listed in the Gradescope review quizzes, and you must address all of its parts.

(b) For each part of your chosen question: prepare a complete solution (you can use the homework solutions we post for guidance about the style). Your submission will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should explain how you arrived at your conclusions, using mathematically sound reasoning. Your goal should be to convince the reader that your results and methods are sound. Imagine you are preparing these solutions for someone else taking CSE 105 who missed that week and is "catching up".

(c) Include at least 2 potential mistakes that a student may have made while attempting to solve the quiz problem that you selected. Explain why the reasoning behind these mistakes is flawed so that a student reading this may learn from these mistakes. It's a good idea to include mistakes that you made when you first tried to solve this problem!

Style guidelines: your written submission for Task 1 should clearly label the three parts: *Question Selection*, *Solution*, and *Potential Mistakes*.

Task 2: Proving undecidability with mapping reductions (Written)

Define:

$$L_n = \{\langle M \rangle \mid M \text{ is a Turing machine and } |L(M)| = n\}$$
$$X_y = \{\langle M \rangle \mid M \text{ is a Turing machine and } y \in L(M)\}$$

Option 1: Pick a specific positive integer n and you will show that L_n is undecidable.

Option 2: Pick a specific string y over $\{0, 1\}$ and you will show that X_y is undecidable.

For either option:

(a) Clearly specify whether you chose Option 1 or Option 2, and specify the value of n or y you picked.

(b) Give two specific examples of strings in the set L_n or X_y, and two specific examples of strings not in the set. Justify your examples with specific connections between the strings and the definition of the set.

(c) Pick whether you will mapping reduce A_{TM}, $HALT_{TM}$, $\overline{A_{TM}}$, or $\overline{HALT_{TM}}$ to your set. Define **two different** computable functions that can witness the mapping reduction. Prove that each of these functions witnesses the mapping reduction.

If you get stuck:

We want you to demonstrate your knowledge about mapping reductions in this part of the project. As professionals, it's important to realize when we don't know or unsure about something. In grading your work on this part of the project, some partial credit will be available for partial correct progress on the task and then explanations of where you got stuck and what you did to try to get unstuck.

Task 3: Video about computable functions

To relate the difficulty level of one language to another we use mapping reduction, which relies on the notion of computable function. In this part of the project, you will define and explain a specific computable function from $\{0,1\}^*$ to $\{0,1\}^*$.

Presenting your reasoning and demonstrating it via screenshare are important skills that also show us a lot of your learning. Getting practice with this style of presentation is a good thing for you to learn in general and a rich way for us to assess your skills. Create a 3-5 minute screencast video with the following components:

- Start with your face and your student ID for a few seconds at the beginning, and introduce yourself audibly while on screen. You don't have to be on camera for the rest of the video, though it's fine if you are. We are looking for a brief confirmation that it's you creating the video and doing the work you submitted.

- Present the function you will be working with. You can pick any function you like so long as:

    - Its domain is $\{0,1\}^*$ and its codomain is $\{0,1\}^*$
    - It is not the identity map (that sends every string to itself), and it is not the function we worked through in class $f_1 : \{0,1\}^* \to \{0,1\}^*$ where $f_1(x) = x0$.

    Your video should include a clear and precise definition of the function.

- Give a high-level description of a Turing machine witnessing that your function is computable.

- Present the state diagram and formal definition of a Turing machine witnessing that your function is computable.

- Trace the computation of the Turing machine whose state diagram you gave on an input of length 3.

You will submit this video along with a written version of Tasks 1 and 2 to Gradescope.

Extra exploration (not for credit): What would it take to implement your computable function in code in a programming language of your choosing? Could you use this computable function to witness any mapping reductions?

1 Grading Criteria and checklists

Task 1

Submission covers a complete review quiz question from the relevant weeks (all parts of the question must be addressed for multi-part questions).

Submission clearly labels review questions, including which day it's from and the problem description.

Submission includes why the student picked the problem/ what they found interesting.

Solution is written (or typed) out in detail step-by-step, with clear and correct logic and justification.

Submission includes 2 potential mistakes that a student might make while solving this question and explains why they are wrong.

Task 2

Submission clearly specify whether Option 1 or Option 2 is chosen, and clearly specifies the value of n or y as appropriate.

Two specific examples of strings in the set and two specific examples of strings not in the set are included. Justifications of membership / non-membership are complete, clear, correct, and precise. Explanations include specific refereece to the example and to relevant definitions.

Each of the two mapping reductions clearly identify the sets involved and include a high-level definition for a Turing machine witnessing the mapping reduction, an analysis of the output of the function for possible inputs, and a connection with the definition of mapping reduction. Definitions and explanations are complete, clear, correct, and precise.

Task 3

Logistics Items

- Video loads correctly

- Video is between 3 and 5 minutes

- Video shows the student's face and ID, and they introduce themselves audibly while on screen.

The video clearly presents a function which is well-defined and computable.

The video presents a correct high-level description of a Turing machine that computes this function.

The video presents a complete and correct formal definition of a Turing machine that computes this function, including a state diagram.

The video includes a trace of the computation of this Turing machine on an input of length 3, where each step of the trace is included and shows the contents of the tape, the location of the read-write head, and the control state of the machine. The trace compares the Turing machine behavior with the expected output of the function on this input string.