Project Introduction

CSE105 Spring 2022

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:
https://drive.google.com/open?id=1KROMAQuTCk40zwrEFotlYSJJQdcG_GUU.
The video that was produced from that recording session in Zoom is here:
https://drive.google.com/open?id=1MxJN6CQcXqIbOekDYMxjh7mTt1TyRVMl

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 vidoes, 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.

Project Part 1: Regular Expressions

Please respond to all 3 tasks below. As you do so, explain your ideas and constructions so that a student who is in the class with you but about a week behind could fully understand your project. In other words, you can assume that the reader knows the prerequisite material and you should include justifications and explanations for anything you use or refer to that was taught in CSE 105. The work you submit for the project needs to be your own, and any resources you consult while doing this work must be cited. You will submit your project to Gradescope, under

the assignment titled ***Project Part 1***.

Task 1: Explain a review quiz question (Written)

1. Select one question from one of the review quizzes from 3/30/22 (Wednesday of Week 1) to 4/13/22 (Wednesday of Week 3) 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 Review sections of the relevant notes documents, and you must address **all** of its parts.

2. 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"

3. 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: Formulate a regular expression (Written)

1. Consider the alphabet Σ = {x, y, z}. Choose and write out a regular expression containing all three operations allowed in regular expressions-- Kleene star, union, and concatenation. *Tip: use the notation and conventions listed in the book and used in class.*

2. List 3 strings that do not match the pattern described by your regular expression, and 3 strings that do. Explain why each string does or does not match the pattern.

3. Explain whether the language described by your regular expression is finite or infinite. Justify why this is the case based on your regular expression's syntax, using precise and correct terminology from class as applicable.

Extra exploration (not for credit) Will this be the case for any language which has a Kleene star, union, and concatenation? List 1 counterexample! HINT: We discussed an example in-class during Week 1.

Task 3: Investigate your regular expression (Video)

1. Many programming languages and other familiar tools give us the ability to work with regular expressions. But, they often use different syntax than what we have seen.

Research how one of the following options represents the syntax of regular expressions: Java, Python, JavaScript, grep, Excel/Google Sheets. Clearly describe the tool or language you chose to research. Then, write the regular expression that you created in Task 2 in this new syntax. Make sure to replace the syntax for all relevant operations (union, Kleene star, concatenation) with the new syntax. Write out test cases that demonstrate that your choice of strings that matched the pattern described by your regular expression in Task 2 work, and those that didn't match the pattern don't. Discuss how you set up these test cases and what output / feedback the tool/ language produces to indicate whether a string matches a regular expression.

2. 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.
    - State which programming language or tool you chose to work with in step 1, and briefly tell us why.
    - Show the entire process for how you translated between the first regular expression that you generated in Task 2, and the one with new syntax that you created in this task. To do this, you should start the video showing only your original regular expression. You will write the new one during the video, pointing out any syntax differences between the two. To explain how you translated each part, show the sources you used to figure out each part as you go! Please make sure that URLs of any documentation or other resources you used are clearly legible in the video.
    - Demonstrate the test cases that that you set up, describing each one, its expected behavior, and what output / feedback the tool/ language produces to indicate whether the expected behavior matches the actual behavior.
    - Discuss any difficulties or confusions you encountered during this translation process. Or, explain at least 2 difficulties that another student might come across during this process.

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

Extra exploration (not for credit) Think about whether the definition of "regular expressions" that we've covered in class is equivalent to the definition of the "regular expressions" from the language/tool that you researched in the third step of Task 2. What would it even mean for these to be equivalent? How could we tell?

Checklist (this is how we will grade this part):

Task 1: Explain a review quiz question

- 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 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: Formulate a regular expression

- Submitted regular expression satisfies given constraints and is written with precise notation, consistent with the textbook
- Lists 3 strings that don't match the pattern described by the created regular expression and explains why each of these strings doesn't match the pattern
- Lists 3 strings that do match the pattern described by the created regular expression and explains why each of these strings does match the pattern
- Submission clearly labels the set described by the regular expression as finite or infinite, and justifies this choice. The justification is precise and complete.

Task 3: Investigate your regular expression (Video) Only the video will be graded

Logistics Items

- Video loads correctly
- Video is between 3 and 5 minutes
- Video shows the student's face and ID, and they introduce themself audibly while on screen

Content Items

- States which programming language/tool was chosen and explains why
- Shows the process of translating their original regular expression into the new syntax that they researched. Must start out with the original expression and show exactly how they translated each part of the expression to the new syntax, including showing the references that they used to figure out each part.
- Video demonstrates all six test cases
- Video discusses difficulties and confusions that occurred/may occur while solving this problem