CSE 20 Project

Summer Session 2, 2021

Description

This project replaces small exams given in past summers. This will be an individual assessment, designed to have you demonstrate problem-solving skills *and* to improve your technical writing. Clear communication is an invaluable skill, and the project will give each of you the chance to develop it. The first two parts of the project will be written math problems. We will grade them and give you feedback on your technical writing. The third part of the project will allow you to incorporate this feedback by revising a prior submission. We will drop the lowest score when computing your final grade – that means your grade on the revised submission will replace the original grade.

In general, your focus should be on clarity. The reader should be able to follow your reasoning without any doubt as to how a particular conclusion was reached.

All components are due at 11:59pm on the deadline.

Part 1: Due 8.11.21

We define the following functions on pairs of 3-tuples:

$$d_1((x_1,x_2,x_3),(y_1,y_2,y_3)) = \sum_{i=1}^{3} ((|x_i - y_i| + 1) \ \textbf{div}\ 2)$$

$$d_2((x_1,x_2,x_3),(y_1,y_2,y_3)) = \sqrt{\sum_{i=1}^{3} (x_i - y_i)^2}$$

These definitions can be generalized to pairs of n-tuples. Your written submission should include the following information:

1. Pick 4-6 movies or TV shows. Give them numerical ratings on a scale from 1-10.
2. Construct an n-tuple where the first element is the rating of the first movie/show, the second element is the rating of the second movie/show, and so on. Specify n.
3. Write down new functions d_1' and d_2' that compute the same quantities for n-tuples (for your choice of n).
4. Calculate the result of applying d_1' to your n-tuple and an n-tuple consisting of all zeros. Explain your work.
5. Calculate the result of applying d_2' to your n-tuple and an n-tuple consisting of all zeros. Explain your work.

Requirements:

- Names of shows/movies are listed, and there are 4-6 of them
- The n-tuple is defined appropriately, n is given correctly, and all notation makes sense.
- The calculation of d_1 is correct and explained clearly.
- The calculation of d_2 is correct and explained clearly.

Part 2: Due 8.18.21

RNA strands are nonempty *strings* from the alphabet $B = \{\texttt{A}, \texttt{G}, \texttt{C}, \texttt{U}\}$. Let S denote the set of RNA strands.

We define the following functions on RNA strands:

$$\texttt{mutate}(b_1\ b_2 \dots b_n, k, b) = b_1\ b_2 \dots b_{k-1}\ b\ b_{k+1} \dots b_n$$

$$\texttt{insert}(b_1\ b_2 \dots b_n, k, b) = \begin{cases} b_1\ b_2 \dots b_n\ b & k > n \\ b_1\ b_2 \dots b_{k-1}\ b\ b_k \dots b_n & k \leq n \end{cases}$$

$$\texttt{delete}(b_1\ b_2 \dots b_n, k) = \begin{cases} b_1\ b_2 \dots b_n & k > n \\ b_1\ b_2 \dots b_{k-1}\ b_{k+1} \dots b_n & k \leq n \end{cases}$$

In each function above, the input $b_1\ b_2 \dots b_n$ is an RNA strand of length n, k is a non-negative integer index, and $b \in B$ is a character. `mutate` modifies a strand by replacing the kth character, `insert` inserts a new character at position k, and `delete` deletes the kth character.

Now, consider the following logical predicates:

$$Mut(s_1, s_2) = \exists k \in \mathbb{Z}^+\ \exists b \in B.\ \texttt{mutate}(s_1, k, b) = s_2$$

$$Ins(s_1, s_2) = \exists k \in \mathbb{Z}^+\ \exists b \in B.\ \texttt{insert}(s_1, k, b) = s_2$$

$$Del(s_1, s_2) = \exists k \in \mathbb{Z}^+.\ \texttt{delete}(s_1, k) = s_2$$

In this assignment, you will come up with a logical statement about RNA strands and analyze it. Below is an example statement:

$$\forall s_1 \in S\ \forall s_2 \in S.\ Ins(s_1, s_2) \implies Del(s_2, s_1)$$

Less formally, this means that for all RNA strands s_1 and s_2, if s_2 can be formed by inserting a character into s_1, then s_1 can be formed by deleting a character from s_2.

To complete the assignment, do the following:

1. Using appropriate symbolic notation, write a quantified statement about RNA strands. If you'd like, define extra predicates for use in your statement. For example, it might be useful to reason about the length or contents of the string. Your quantified statement should have the following features:

(a) Nested quantifiers, including at least one existential quantifier and one universal quantifier.

(b) One negation and one binary operator (and, or, etc)

(c) Negations appear only on predicates (that is, no negation is outside a quantifier or on an expression involving logical connectives)

(d) Use at least two of Mut, Ins, Del

2. Translate your statement to english.

3. Negate the whole statement and rewrite it such that negations appear only on predicates (that is, no negation is outside a quantifier or on an expression involving logical connectives)

4. Prove or disprove your statement

As always, show your work when necessary and justify each step.

Requirements:

- Quantified statement is clearly stated, well-defined, syntactically correct, and meets all requirements.
- Translation to english is clear, correct, and complete.
- The negation of the statement is clear, well-defined, syntactically correct, and meets all requirements.
- The proof (or disproof) of the statement is clear, correct, and complete.

Part 3: Due 9.1.21

For this part you will revise your submission for *either* part 1 or part 2 of the project. Taking into account the feedback we gave you, submit an improved version. If you aced it the first time, don't do anything! Please inclue a brief summary of the changes you made – either by describing the changes separately, or by highlighting the changes in-text.

Requirements: This will be graded accrding to the original requirements for the part you revise. Make sure it's clear to the reader what you revised.