Essay

Discrete Structures

Regulations

This is a group-of-2 assignment. Groups of only 1 student are not accepted.

The duration of this essay is 14 days, from the beginning of April 18th 2024 to the end of May 2nd 2024.

Only 1 student who represents the group will submit the essay to your ELIT classroom. Late submissions are not accepted. Submissions via email are not accepted. You need to submit a compressed file named with your Student IDs, eg. 52200123_52201001.zip/rar, including this structure:

- The document file is in Word format (.doc/docx), named by your Student IDs, eg. 52200123_52201001.docx, using our faculty's format, from 15 to 25 pages.
    - The structure of this document should be:
        - Chapter 1: The tasks of each member and self-evaluation of your group should be declared at the end of this report.
        - Chapter 2: Truth table
        - Chapter 3: Quantified Reasoning over Real-World Data Using Predicate Logic
        - Chapter 4: RSA cryptosystem
        - References: Using the faculty's format.
    - English is required for high-quality classes.
    - Format violations will cost from 10% to 50% of your total scores.
    - Any case of plagiarism will get 0.
- The source code files are named by your Student ID and Task Number, eg. 52200123_52201001_1.py.

Tasks

1. Truth table

- Write function def Infix2Postfix(Infix):
    - Input: Infix is a string of logical operators and alphabet characters from "A" to "Z" express a logic expression.
        - "(": Open parenthesis
        - "~": Not
        - "&": And
        - "|": Or
        - ">": Implies
        - "=": If and only if

        - “)”: Close parenthesis
    - Output: Postfix is a string calculated from Infix using Reverse Polish notation.
    - Write the theory of Reverse Polish and Basic logic used on calculation of Truth tables in the report document.
- Write function def Postfix2Truthtable(Postfix):
    - Input: Postfix from (1.)
    - Output: The truth table from the input logic expression Infix.
    - Explain your program by doing step by step each function on these testcases in the report document.
        1. R|(P&Q)
        2. ~P|(Q&R)>R
- Shows your experimental result on 5 testcases (run the code on 5 testcases and capture the screen picture).
    1. R|(P&Q)
    2. ~P|(Q&R)>R
    3. P|(R&Q)
    4. (P>Q)&(Q>R)
    5. (P|~Q)>~P=(P|(~Q))>~P

2. Quantified Reasoning over Real-World Data Using Predicate Logic

- Create a small dataset in CSV file with **at least 20 records**. It contains these fields: StudentID, StudentName, DayOfBirth, Math, CS, Eng.
- Define these predicates based on your dataset, each predicate should return a boolean value for a given input.
    - is_passing(student): all scores are greater than or equal to 5.
    - is_high_math(student): math score is greater than or equal to 9.
    - is_struggling(student): math and cs score is less than 6.
    - improved_in_cs(student): cs score is greater than math score.
- Implement Python functions that evaluate whether each statement is true or false over your dataset.
    - 2 Universal quantifications (e.g., ∀x P(x))
        - "All students passed all subjects"
        - "All students have a math score higher than 3"
    - 2 Existential quantifications (e.g., ∃x P(x))
        - "There exists a student who scored above 9 in math"
        - "There exists a student who improved in CS over Math"
    - 2 Combined/nested statements (e.g., ∀x ∃y Q(x, y))
        - "For every student, there exists a subject in which they scored above 6"
        - "For every student scoring below 6 in Math, there exists a subject where they scored above 6"
- Write Python functions to evaluate the **negation** of the quantified statements above and explain their meaning in plain English.

3. RSA cryptosystem

- Implement a Python program to encrypt and decrypt a message with the RSA cryptosystem. Cryptography libraries are allowed.
- Test the implemented RSA cryptosystem using sample messages and verify the results. Capture your screen results and explain them in your report document.
- Measure encyption time and decryption time for different plaintext message lengths. With x-coordinates are plaintext message lengths, and y-coordinates are time consumptions, draw a graph to prepresent the changes. Discuss the limitation(s) of the RSA cryptosystem.
- Conclude with recommendations for improving the RSA cryptosystem implementation.

Rubric

Criteria		Scale	0 score	1/2 score	Full score	Self-evaluation	Reason
Task 1	Implementation	1	Error	Correct but bad performance	Correct and good performance		
	Theory of Reverse Polish and Basic logic	0.5	Do nothing or wrongly	Not enough details, no example, no comment	Correct calculations, detailed explanations		
	Explain testcases 1 and 2	1	Do nothing or wrongly	Explain only 1 testcase correctly.	Explain 2 testcases correctly.		
	Run all 5 testcases	1.25	Do nothing or wrongly	Explain only <= 3 testcases correctly.	Explain 5 testcases correctly.		
Task 2	Create dataset	0.5	No data	Not enough data	Good data		
	Determine truth value	1.5	Do nothing or wrongly	Run only <= 3 statements correctly	Run only <= 6 statements correctly		
	Negation	0.75	Do nothing or wrongly	Run only <= 3 statements correctly	Run only <= 6 statements correctly		
Task 3	Implementation	0.5	Error	Correct but bad performance	Correct and good performance		
	Test	0.5	No test	Test without verification	Test and verification		
	Discussion	1	Do nothing or wrongly	Not enough details, no example, no comment	Correct, detailed explanations		
	Recommendation	1	Do nothing or wrongly	Not enough details, no example, no comment	Correct, detailed explanations		

Reference	0.5	No reference	Wrong format	Right format		
Total	10	Result			0	