Instruction to students: Attempt all questions

Question 1. (10 Marks)

Prove that the set of strings that are finite prefixes of this infinite string
-0-1-10-11-100-101-110-111-1000-1001-1010-1011- $\cdots$
is not regular.

Question 2. (10 Marks)

Consider the following attempt at transforming 3-SAT to 2-SAT. Replace each clause

$$(A \vee B \vee C)$$

where A, B, C are, as usual, variables or negated variables, by

$$(A \vee B) \wedge (B \vee C) \wedge (A \vee C)$$

What is wrong with this "transformation"?

Question 3. (20 Marks)

For each of the five statements below, state whether it is `TRUE` or `FALSE`. Don't explain your answer.

1. If A and B are two problems in NP, A is NP-complete, and there is a polynomial-time transformation <u>from A to B</u> then B is NP-complete as well.
2. If A and B are two problems in NP, A is NP-complete, and there is a polynomial-time transformation <u>from B to A</u> then B is NP-complete as well.
3. If A and B are two NP-complete problems then there is a polynomial-time transformation from A to B and also a polynomial-time transformation from B to A.
4. If A is a problem that cannot be solved in polynomial time and there is a polynomial-time transformation from A to B then B cannot be solved in polynomial time either.
5. Any problem A that can be transformed into CIRCUIT SATISFIABILITY by a polynomial-time transformation is NP-complete.

Question 4. (20 Marks)

Ex. 4.5. Which of the following statements are true?
("Polynomial-time" is understood to mean "deterministic polynomial-time.")

1. No polynomial-time algorithm for Vertex Cover is known.
2. No polynomial-time algorithm for Graph 3-Coloring is known.
3. No polynomial-time algorithm for Graph 2-Coloring is known.
4. If a polynomial-time algorithm exists for SAT then polynomial-time algorithms exist for all problems in NP.
5. There is a polynomial-time transformation from 3-Sat to Vertex Cover.
6. There is a polynomial-time transformation from Vertex Cover to 3-Sat.
7. If a polynomial-time algorithm exists for Vertex Cover then polynomial-time algorithms exist for all problems in NP.
8. If A and B are two problems in NP, if we know that B is NP-complete, and we know that there is a polynomial-time transformation of A into B, then we know that A is NP-complete as well.
9. Vertex Cover is known to be NP-complete.

Question 5. (20 Marks)

Ex. 4.7. Given a polynomial-time solution to the *decision* version of Satisfiability, i.e. given a Boolean function that tells you whether or not a circuit you pass to it is satisfiable, how could you program a polynomial-time solution to the full version of the problem? (The full version of the problem means that if a satisfying assignment exists you have to find one, not just say that one exists.)

Question 5. (20 Marks)

Ex. 4.8. *(a)* Apply the transformation shown in class from 3SAT to VERTEX COVER to this expression:

$$(x \vee \neg y \vee z) \wedge (x \vee y \vee w)$$

(b) If in a Boolean expression in 3-CNF the same variable shows up unnegated in all clauses, what property of the graph does that translate into (again, in the transformation discussed in class)?

(c) Still about the same transformation, if there are 10 solutions to the vertex cover problem, does this imply that there are 10 ways to satisfy the expression? Explain briefly.

End of exam. Good luck