Notes:

- Read Course Information: Section 7 (Miscellaneous) and Section 9 (Academic Dishonesty or Misconduct).
- When you are giving a construction, example, etc., provide a justification with your argument. Your solutions to numerical problems must contain the derivation of your answers. In all of your presentations, strive for correctness, completeness, and clarity. When in doubt about the assumptions of problems, the interpretations of wording, etc., consult the instructor.
- You should strive to complete all problems assigned, and a subset of them will be graded.

1. Read the notes above carefully.

2. Study the context-freedom of each of the following languages. Prove your answers.
   
   (a) \( L_1 = \{0^i1^j : 0 \leq i \leq j \} \).
   
   (b) \( L_2 = \{ww^r w \mid w \in \{0,1\}^*\} \).

3. Do [Sip13] Chapter 3, exercise 3.8 (c). Give its formal tuple-definition (including the definition of the deterministic transition function in implementation-level details) and brief interpretations for the states/ transitions.

4. Let \( \Sigma = \{0,1\} \), and assume that \( A \) and \( B \) are languages over \( \Sigma \) such that \( A \) is Turing-recognizable (that is, recursively enumerable) and \( B \) is finite. Prove that the symmetric difference \( A \Delta B = (A \cap \overline{B}) \cup (\overline{A} \cap B) \) is Turing recognizable. Notes: (1) No credit will be given for using closure properties of Turing-recognizable languages, and (2) A direct construction of a deterministic Turing machine is required – in the form of high-level description (with comments).

5. We have covered [Sip13] Theorem 4.1 in class. Read Theorems 4.2 – 4.5 and their proofs.
   
   Prove that the following language:
   
   \[ \text{DISJ} := \{ (A, B) \mid A \text{ and } B \text{ are deterministic finite automata with } L(A) \cap L(B) = \emptyset \} \]
   
   is Turing-decidable (recursive).

6. Let \( \Sigma = \{0,1\} \). Assume that \( A \) and \( B \) are Turing-recognizable languages such that \( A \cup B = \Sigma^* \). Prove that there exists a Turing-decidable language \( C \subseteq \Sigma^* \) such that:

   \[ A \cap \overline{B} \subseteq C \text{ and } \overline{A} \cap B \subseteq \overline{C}. \]