1. Finite state automata and regular expressions.
   (a) Design a deterministic finite state automaton that recognizes the language
   \[ L = \{ w \in \{a, b\}^* \mid w \text{ has an odd number of both } a \text{ and } b \} \]
   5 p.
   (b) Design a regular expression that describes the language
   \[ L = \{ w \in \{a, b\}^* \mid w \text{ has an odd number of } a \text{ or } b \} \]
   5 p.
   (c) Find the minimal deterministic finite state automaton that accepts the language
   \[ L = \{ w \in \{a, b\}^* \mid w \text{ has an odd number of either } a \text{ or } b \} \]
   5 p.

2. Let \( \Sigma = \{0, 1\} \). We examine the language
   \[ L = \{ w \mid w = x0y, \text{ where } |x| = |y| \text{ and } x, y \in \Sigma^* \} \]
   (a) Show that the language \( L \) is not regular. 7 p.
   (b) Design a pushdown automaton that recognizes \( L \). Present the pushdown automaton as a state chart. Additionally, give an accepting computation for the input strings 01001 and 11011. 8 p.

3. Design a deterministic Turing machine that decides the language
   \[ L = \{ a^ib^jc^k \mid i \geq j \geq k \geq 0, i - j = k \} \]
   If you wish, the machine you design may have multiple tapes and it may keep one or more tape heads stationary in a transition. Give a short description of your algorithm and present the machine as a state chart. 15 p.

4. Let \( L \) be a formal language over the alphabet \( \Sigma \).
   (a) Show that if \( |L| = n \) for some \( n \in \mathbb{N} \), then \( L \) is regular. 7 p.
   (b) Show that if there exists an \( n \in \mathbb{N} \) such that \( |w| \leq n \) for all \( w \in L \), then \( L \) is regular. 8 p.

Total 60 p.