Problem 1

Algorithm 1

Input: Two integers \( M > 0 \) and \( N \geq 0 \)
Output: An integer \( k \) such that \( k = M^N \)

\[
\begin{align*}
  a & \leftarrow M \\
  b & \leftarrow N \\
  k & \leftarrow 1 \\
  \text{while } b > 0 \text{ do} \\
  & \quad \text{if } \text{even}(b) \text{ then} \\
  & \quad \quad a \leftarrow a \times a \\
  & \quad \quad b \leftarrow b/2 \\
  & \quad \quad \text{else} \\
  & \quad \quad k \leftarrow k \times a \\
  & \quad \quad b \leftarrow b - 1 \\
  & \quad \text{end if} \\
  \text{end while}
\end{align*}
\]

Note that \( / \) is an integer division. For instance, the predicate \( \text{even}(b) \) can be implemented as \( b = 2 \times (b/2) \).

Find an invariant for each node of the control-flow of the program. Use your invariants to prove the correctness of the program.

Puzzle 2 (Bonus problem on automata theory)

Four glasses are placed on the corners of a square-shaped rotating table. Some of the glasses are upright (up) and some upside-down (down). A blindfolded person is seated next to the table and is required to re-arrange the glasses so that they are all up or all down, either arrangement being acceptable, which will be signalled by the ringing of a bell. The glasses may be re-arranged in turns subject to the following rules. Any two glasses may be inspected in one turn and after feeling their orientation the person may reverse the orientation of either, neither or both glasses. After each turn the table is rotated through by (arbitrarily) a quarter turn, half-turn, three quarters of a turn, or not at all. The puzzle is to devise a (non-stochastic) algorithm which allows the blindfolded person to ensure that all glasses have the same orientation (either up or down) in a finite number of turns.

Suggestion 3

Q & A session on the first five lectures.