1. For each of the properties below, provide a formal (detailed) proof or give a counterexample.

   Strengthening:
   (a) \(\text{transient}(P) \land [P' \implies P] \implies \text{transient}(P')\) [5 pts]

   Weakening:
   (b) \(\text{transient}(P) \land [P \implies P'] \implies \text{transient}(P')\) [5 pts]

2. For each of the statements below, indicate whether the statement is true or false. If true, give a formal (detailed) proof. If false, give a counterexample.

   Constants: [5 pts]
   (a) \(P \rightsquigarrow P\)

   Stable strengthening: [5 pts]
   (b) \(\text{stable}(P) \land \text{transient}(P \land \lnot Q) \implies P \rightsquigarrow (P \land Q)\)

   Conjunctivity: [5 pts]
   (c) \((P \rightsquigarrow Q) \land (P' \rightsquigarrow Q') \implies (P \land P') \rightsquigarrow (Q \land Q')\)

3. Give a full proof for the earliest meeting time algorithm in Sivilotti, Section 4.4 (similar to the proof provided for FindMax in Section 4.2). [25 pts]

4. [Nondeterministic Iteration – Shortest Paths]
   Last week, we completed a part of the proof for the correctness of this algorithm. We will now complete the proof in this homework. The program is restated again for your convenience.

   Somebody proposes the following algorithm to find the shortest path between every pair of vertices in a finite directed graph. Let \(W\) be the edge-weight matrix, i.e., \(W[j, k]\) is the weight of edge \((j, k)\). Weights are real numbers. Assume that the graph is completely connected,
and therefore $W[j, k]$ exists for all $j, k$. Also $W[j, j] = 0$ for all $j$. The graph has no cycles of negative weight.

Let $D$ be a matrix with the same dimensions as $W$. The proposed algorithm is as follows:

(a) **initially:** $D = W$

(b) There is a command for every triple $(i, j, k)$ of vertices, and the command is:

$$\text{IF } D[i, k] > D[i, j] + D[j, k] \text{ THEN } D[i, k] := D[i, j] + D[j, k]$$

The claim is that the algorithm will terminate with $D$ being the matrix of shortest path lengths, i.e., $D[j, k]$ will be the length of the shortest path from vertex $j$ to vertex $k$.

(a) For each pair of vertices $(i, k)$, consider the set of the lengths of all possible paths from $i$ to $k$ with length less than or equal to $W[i, k]$. (Note this is a set of path lengths, not a set of paths.)

Let $L[i, k]$ be the set ordered in decreasing order. For each state of the algorithm, you have shown that $D[i, k]$ is the length of some path from $i$ to $k$. So, $D[i, k]$ is a value in the list $L[i, k]$. Let $f_{i,k}(D[i, k])$ be the index of $D[i, k]$ in $L[i, k]$.

Let

$$F(D) = \sum_{i,k} f_{i,k}(D[i, k]).$$

Explain why $F$ is a metric for this program. A formal proof at a similar level of detail to Sivilotti’s proofs in Chapter 4 (after FindMax) is sufficient, but not required. However, you should provide sufficient details that it is clear there are no missed steps.

(b) Write down the invariant and fixed point condition. No proof is necessary since you’ve already proved that. Prove that the program terminates at the fixed point using he invariant, the fixed point condition, and the metric above.