1. Consider the following program with boolean variables $v$ and $w$:

\[
\begin{align*}
v & := w \lor v; \quad w := v \land \neg w; \quad v := w \land \neg v
\end{align*}
\]

(a) Using standard operational semantics, draw the complete 16 state transition system describing the behaviour of this program. Note that $v$ and $w$ may start either at true or false. You will have to introduce a program counter variable to keep track where in the program you are.

(b) The program was meant to swap the values of $v$ and $w$. By inspecting the transition system, identify when the program does not work.

(c) The program can be verified by adding two new variables $v'$ and $w'$ initially equal to $v$ and $w$ respectively, and then checking that whenever the program has terminated (by looking at the program counter), the values of $v$ and $w$ match the values of $v'$ and $w'$ respectively. Express this property using logic. Is it a safety or liveness property? (Explain your answer).

(d) Draw the reachable portion of the transition system with the new state variables $v'$ and $w'$. (The new system still has four initial states, with the values of $v$ and $v'$ coinciding, similarly those of $w$ and $w'$).
(e) Explain the distinction between forward and backward model checking and illustrate your explanation by explaining how they work in discovering the bug in the transition system you gave in (d) with respect to the property you gave in (c).

(f) Explain abstraction, its power and pitfalls.