Problem 1 [modeling and verification]. Consider the program given in file `ccp.m`. It contains a description of a cache coherence protocol: \( n \) processes share access to a cache; access is granted via a hub. Access can be shared (read-only; several processes at the same time) or exclusive (read/write; at most one process at a time). The access status of each process \( i \) is indicated by its local state variable \( c[i] \); value \( I \) indicates that the cache copy of that process is invalid.

Each process communicates with the hub via three uni-directional channels per process: requests are sent to the hub through Channel 1, acknowledgments through Channel 3. Channel 2 is for communication from the hub to each process, such as instructions to invalidate the local cache copy, or to grant access to the cache item (either shared or exclusive). Processes do not communicate directly with one another.

The protocol consists of variable declarations, a Startstate declaration, and \( 10 \times n \) rules: ten rules parameterized by the process identity \( i \); there is one instance of each rule per process. Rules consist of a guard and an action. The rule is enabled if the guard is true; firing the rule means to execute its action.

The execution model of the protocol is as follows: first, the initializations under the Startstate keyword are performed. After that, the following is repeated forever: among all \( 10 \times n \) rules that are enabled, one is nondeterministically selected and executed atomically.

The end of the protocol description contains an invariant of interest: that, if any process has exclusive access to the cache data, all other processes are in the invalid state. (In particular, this implies that exclusive is mutually exclusive.)

Model the protocol in a model checker of your choice, such as SPIN or NuSMV, and find out whether the invariant property holds. If it doesn’t, give a counterexample. If it does, give a non-trivial modification to the protocol that breaks the invariant.

Problem 2 [specifications in LTL]. Formulate the following English property specifications in LTL. Use atomic propositions as suggested by keywords in *italics*: these are observables. If you believe that the property cannot be reasonably precisely translated into formal logic, state so and give reasons.

1. There will be no reset for the next four clock cycles.
2. If req is true and ack becomes true one cycle later, then eventually req will become false.
3. Whenever req is followed, after finitely many steps, by an ack, then there will be no further request until restart.
4. The phone rings now and then every other moment.
5. The phone sometimes rings between the start and the end of a speech.

Problem 3 [specifications in CTL]. Formulate the following English property specifications in CTL. Use atomic propositions as suggested by keywords in *italics*: these are observables. If you believe that the property cannot be reasonably precisely translated into formal logic, state so and give reasons.

1. There will be no reset for the next four clock cycles.
2. If req is true, then eventually req will become false.
3. The array will eventually be sorted, and once it is it remains sorted forever (it will never be unsorted).
4. The phone rings now and then every other moment.
5. Along some timeline the phone sometimes rings between the start and the end of a speech.

Problem 4 [equivalence of CTL formulas]. Show that the CTL formulas \( \phi_1 = AX AF p \) and \( \phi_2 = AF AX p \) are not equivalent. To do that, find a Kripke structure \( M \) (some of whose states are labeled \( p \)) and a state \( s_0 \) such that \( M, s_0 \models \phi_1 \) but \( M, s_0 \not\models \phi_2 \), or vice versa. Show \( M \) as a graph structure, including the labeling, and identify \( s_0 \).
Problem 5 [comparing LTL and CTL]. For this problem, let $M = (S, R, L)$ be a Kripke structure. Simplifying, we assume that $M$ has a unique initial state $s_0$. An LTL formula $f$ and a CTL formula $g$ are called equivalent if

$$M \models f \iff M \models g.$$ 

Note that “$\models$” on the left denotes LTL semantics. That is, every path through $M$ that starts at $s_0$ must satisfy $f$. On the right, “$\models$” denotes CTL semantics. That is, the CTL formula $g$ must evaluate to true over $M$ and $s_0$.

1. Show that the LTL formula $GFp$ and the CTL formula $AG AFp$ are equivalent.

2. Show that the LTL formula $FGp$ and the CTL formula $AF AGp$ are not equivalent, by presenting a small Kripke structure $M$ with appropriate labels and a state $s_0$ in it such that $M \models FGp$, but $M, s_0 \not\models AF AGp$.

3. The fact that the LTL formula $FGp$ is not equivalent to the CTL formula $AF AGp$ does not necessarily imply that $FGp$ is not expressible in CTL. Design a suitable definition of expressibility of an LTL formula in CTL. With your definition, is $FGp$ expressible in CTL? (Intuition suffices.) Is $AX EXp$ expressible in LTL? (For this one, try to prove your result.)

4. Among the formulas $XFp$, $FXp$, $AX AFp$, and $AF AXp$, three are equivalent to one another. Which? Prove them equivalent, and give a Kripke structure witnessing that the remaining formula is not equivalent to (any one of) the others.

Problem 6 [fixpoint computations and Tarski-Knaster theorem]. You are to prove that for a finite set $S$ with cardinality $k$ and a total structure $M = (S, R)$, $AFg$ is the least fixpoint of the function $\tau(Z) = g \lor AXZ$, for an arbitrary CTL formula $g$. To this end, we first define:

**Notation.** For $i \in \mathbb{N}$, denote by $AF^{\leq i}g$ the set of all states from which, along any possible future, a state satisfying $g$ can be reached after at most $i$ transitions.

Hence, for example, $AF^{\leq 0}g = g$, and $AF^{\leq 1}g = g \lor AXg$. Now proceed as follows.

1. Prove that $AFg = AF^{\leq k}g$.

2. Prove that for $i \geq 1$, $\tau^i(\emptyset) = AF^{\leq i-1}g$.

3. Use the Tarski-Knaster theorem to conclude that $AFg$ is the least fixpoint of $\tau$. (Hint: the TK theorem doesn’t know anything about CTL, only about fixpoints. You are asked here to make the connection between the two.)