Problem 1. **Software Modeling, Regular Expressions, LTL Model Checking.** Consider a fragment of a concurrent program with two threads, $P$ and $Q$, which share the variable $n$:

<table>
<thead>
<tr>
<th>thread $P$</th>
<th>thread $Q$</th>
</tr>
</thead>
</table>
| 1 repeat
2 begin print $a$ ;
3 $n := n + 1$ ;
4 if $n = 2$
5 begin print reset ;
6 $n := 0$
7 end | 1 repeat
2 begin print $b$ ;
3 $n := n - 1$ ;
4 if $n = -2$
5 begin print target ;
6 $n := 0$ ; stop thread $Q$
7 end |

We take a begin-end block in each thread above as a “critical section” which is entirely executed without any interference from the other thread. This means that, once lines 2-8 in thread $P$ starts executing, there is no interleaving with any of lines 2-8 in thread $Q$, and vice-versa. Interleaving only occurs between full begin-end blocks, and not between portions of them.

(1) Define a transition system $M$ modeling the behavior of the two threads $P$ and $Q$ by drawing its diagram. In this part (1), ignore the printouts \{a, b, reset, target\}.

**Hint for (1):** Draw the diagram of $M$ as a transition system with 8 states:

- 5 states when both $P$ and $Q$ are running and variable $n$ stores an integer $i \in \{0, 1, 2, -1, -2\}$. Denote by $s[P,Q,i]$ the state of $M$ when $P$ and $Q$ are running and $n$ stores $i$.
- 3 states when only $P$ is running and variable $n$ stores an integer $i \in \{0, 1, 2\}$. Denote by $s[P,i]$ the state of $M$ when only $P$ is running and $n$ stores $i$.

Take $s[P,Q,0]$ as the start state of $M$.

**Answer:**

![Diagram of transition system M]
We can incorporate the printouts \{a, b, \text{reset, target}\} as labels in the model \(M\) in two different ways, in parts (2) and (3) first, and then in parts (4) and (5).

(2) Consider the case when \{a, b, \text{reset, target}\} are labels for the transitions.\footnote{Use \{a, b, r, t\} instead of \{a, b, \text{reset, target}\} for simplicity.} For example, we can use label \(a\) to identify the transition from state \(s[P, Q, 0]\) to state \(s[P, Q, 1]\) by writing \(s[P, Q, 0] \xrightarrow{a} s[P, Q, 1]\), and we can use label \(b\) to identify the transition from state \(s[P, Q, 0]\) to state \(s[P, Q, -1]\) by writing \(s[P, Q, 0] \xrightarrow{b} s[P, Q, -1]\), etc.

Your task is to write a regular expression \(E\) over the alphabet \{a, b, r, t\} that denotes all finite sequences of transitions from \(s[P, Q, 0]\) back to \(s[P, Q, 0]\).

\textbf{Hint for (2)}: For every \(i \in \{1, 2, -1\}\), define a regular expression \(E[P, Q, i]\) where \(i \neq 0\) which denotes the sequences of transitions from \(s[P, Q, 0]\) to \(s[P, Q, i]\) without visiting these two states more than once each, followed by the sequences of transitions from \(s[P, Q, i]\) back to \(s[P, Q, 0]\) without visiting these two states more than once each. Try to write the desired \(E\) by using the three regular expressions \(E[P, Q, 1]\), \(E[P, Q, 2]\), and \(E[P, Q, -1]\).

\textbf{Answer}: Following the hint,

\[ E[P, Q, 1] = ab + aar, \quad E[P, Q, 2] = aar, \quad E[P, Q, -1] = ba, \]

\[ E = (E[P, Q, 1] + E[P, Q, 2] + E[P, Q, -1])^* = (ab + aar + ba)^* \]

(3) Consider again the case when \{a, b, \text{reset, target}\} are labels for the transitions. Your task is to write an \(\omega\)-regular expression \(F\) over the alphabet \{a, b, r, t\} that denotes all infinite sequences of transitions that start at state \(s[P, Q, 0]\) and visit state \(s[P, 0]\) infinitely often.

\textbf{Hint for (3)}: Use regular expression \(E\) from part (2).

\textbf{Answer}: Following the hint,

\[ F = Ebbt (aar)^\omega = (ab + aar + ba)^* bbt (aar)^\omega \]

(4) For this part, we view \{a, b, r, t\} as atomic propositions which are labels for the states (not the transitions) in \(M\). For example, if \(S\) is the set of states and \(L\) is the labelling function,

\[ S = \{s[P, Q, 0], \ldots, s[P, 2]\} \quad \text{and} \quad L : S \rightarrow \{\text{propositional WFF’s over \{a, b, r, t\}}\}, \]

then we can write \(L(s[P, Q, 0]) = \{a \land b\}\) to mean that the propositional WFF \(a \land b\) is true at state \(s[P, Q, 0]\); and \(L(s[P, Q, 1]) = \{a\}\) to mean that the atom \(a\) is true at \(s[P, Q, 1]\); etc.

Your task is to complete the definition of the labelling function \(L\).

\textbf{Answer}: Following the hint,

\[ L(s[P, Q, 0]) \triangleq \{a \lor b \lor r\} \quad L(s[P, Q, 1]) \triangleq \{a\} \quad L(s[P, Q, 2]) \triangleq \{a\} \quad L(s[P, Q, -1]) \triangleq \{b\} \]

\[ L(s[P, Q, -2]) \triangleq \{b\} \quad L(s[P, 0]) \triangleq \{r \lor t\} \quad L(s[P, 1]) \triangleq \{a\} \quad L(s[P, 2]) \triangleq \{a\} \]

A more verbose labelling for \(s[P, Q, 0]\) is \(L(s[P, Q, 0]) \triangleq \{(a \land b) \lor a \lor b \lor r\}\), but this latter WFF is equivalent to \(\{a \lor b \lor r\}\).
This is a continuation of part (4). We are given the three LTL formulas:

\[ \varphi_1 \triangleq GFr, \quad \varphi_2 \triangleq G((r \lor t) \rightarrow (Xa \land XXa)), \quad \varphi_3 \triangleq ((a \lor b \lor r) U t). \]

For each LTL formula \( \varphi_i \) above, with \( i \in \{1, 2, 3\} \), your task is twofold:

1. Find a path \( \pi_i \) in \( \mathcal{M} \) (whose first state is the start state \( s[P,Q,0] \)) such that \( \mathcal{M}, \pi_i \models \varphi_i \).
2. Decide whether \( \mathcal{M} \models \varphi_i \).

**Answer:** Consider each formula in turn:

- For \( \varphi_1 \), using the labelling chosen in part (4), there is no path such that \( \mathcal{M}, \pi_1 \models \varphi_1 \). Hence, a fortiori, \( \mathcal{M} \not\models \varphi_1 \).

  On the other hand, if you choose a labelling such that \( \mathcal{L}(s[P,Q,0]) = \{a \land b\} \), as suggested in the hint for (4), then there are paths \( \pi_1 \) such that \( \mathcal{M}, \pi_1 \models \varphi_1 \), e.g., let \( \pi_1 \triangleq (s[P,Q,0] s[P,Q,1] s[P,Q,2])^\omega \) and there many others. Nonetheless, it is still the case that \( \mathcal{M} \not\models \varphi_1 \).

- For \( \varphi_2 \), it is easy to see that every infinite path in \( \mathcal{M} \) will satisfy it. We can therefore let \( \pi_2 \) be any infinite path in \( \mathcal{M} \). Hence, also, \( \mathcal{M} \models \varphi_2 \).

- For \( \varphi_3 \), using the labelling chosen in part (4), there is no path such that \( \mathcal{M}, \pi_3 \models \varphi_3 \). Hence, a fortiori, \( \mathcal{M} \not\models \varphi_3 \).

  However, if you choose a labelling such that \( \mathcal{L}(s[P,0]) = \{t\} \), instead of \( \mathcal{L}(s[P,0]) = \{r \lor t\} \), then there is a path \( \pi_3 \) such that \( \mathcal{M}, \pi_3 \models \varphi_3 \), e.g., let \( \pi_3 \) be an infinite path that starts with \( s[P,Q,0] s[P,Q,-1] s[P,Q,-2] \cdots \). Nonetheless, it is still the case that \( \mathcal{M} \not\models \varphi_3 \) because not every infinite path that begins from the start state \( s[P,Q,0] \) satisfies \( \varphi_3 \).

**Problem 2. CTL Model Checking.** Consider the transition system \( \mathcal{A} \) in Figure 1, whose set of states is \( S \triangleq \{s_0, s_1, s_2, s_3, s_4\} \) and atomic propositions are \( \{a, b\} \). There are two parts in this problem, (6) and (7). For each CTL formula \( \psi_i \) below, with \( i \in \{1, 2\} \), your task is twofold:

1. Determine the satisfaction set \( \text{Sat}(\psi_i) \), i.e., the set of all the states \( s \in S \) such that \( \mathcal{A}, s \models \psi_i \).
2. Decide whether \( \mathcal{A} \models \psi_i \).

(6) CTL formula \( \psi_1 \triangleq \forall (a U b) \lor \exists X (\forall G b) \)
**Answer:** We can follow the inside-out construction of the satisfaction sets:

- \( \text{Sat}(b) = \{s_2, s_3, s_4\} \)
- \( \text{Sat}(\forall G b) = \{s_4\} \)
- \( \text{Sat}(\exists X (\forall G b)) = \{s_0, s_4\} \)
- \( \text{Sat}(a) = \{s_1, s_2\} \)
- \( \text{Sat}(\forall (a \mathcal{U} b)) = \{s_1, s_2, s_3, s_4\} \)
- \( \text{Sat}(\psi_1) = \{s_1, s_2, s_3, s_4\} \cup \{s_0, s_4\} = \{s_0, s_1, s_2, s_3, s_4\} \)

Since the start states, \( s_0 \) and \( s_3 \), are in \( \text{Sat}(\psi_1) \), we conclude that \( A \models \psi_1 \).

A more direct approach is to first observe that \( s \models A \) is one of the initial states of system \( A \), and then observe that \( A, s \models \forall (a \mathcal{U} b) \) because all paths from \( s_3 \) start with \( b \). With these two observations, we conclude \( A \models \psi_1 \).

(7) CTL formula \( \psi_2 \triangleq \forall G (\forall (a \mathcal{U} b)) \)

**Answer:** If \( \pi \triangleq t_0 t_1 t_2 \cdots \) is an infinite execution path, we denote the \( i \)-th state in \( \pi \) by writing \( \pi[i] \) for every \( i \geq 0 \). We start by noting a sequence of equivalences:

\[
A, s \models \psi_2 \iff \text{for every path } \pi \text{ from } s \text{ we have } A, \pi \models G (\forall (a \mathcal{U} b))
\]

\[
\iff \text{for every path } \pi \text{ from } s, \text{ for every } i \geq 0, \text{ we have } A, \pi[i] \models \forall (a \mathcal{U} b)
\]

\[
\iff \text{for every path } \pi \text{ from } s, \text{ for every } i \geq 0, \text{ for every path } \pi' \text{ from } \pi[i], \text{ we have } A, \pi' \models a \mathcal{U} b
\]

Consider the initial state \( s_0 \) and the path \( \pi \triangleq s_0 s_1^\omega \). By the equivalences above, \( A, s_0 \models \psi_2 \) will hold iff all the suffixes of \( \pi \) (including \( \pi \) itself) satisfy \( (a \mathcal{U} b) \). By inspection of the transition system \( A \), it is clear that \( A, \pi \not\models a \mathcal{U} b \). Hence, \( A, s_0 \not\models \psi_2 \) and, hence also, \( s_0 \not\in \text{Sat}(\psi_2) \). Since \( s_0 \) is one of the initial states of \( A \), it follows that \( A \not\models \psi_2 \).

**Hint for (6) and (7):** Follow the inside-out construction of the satisfaction sets, going from smaller sub-formulas to larger sub-formulas (as in the example in Handout 23, pages 3-8).

**Problem 3. LTL Definability.**

(8) Write a LTL formula which enforces the following requirement in a transition system: *At every state \( s \), if \( p \) is true, then at every state \( s' \) reachable from \( s \) in one transition or more, if \( q \) is true, then \( r \) is false until \( t \) becomes true (for all continuations of the execution path starting at \( s' \)).*

**Answer:** Although the translation of informal requirements into formal specifications is generally fraught with ambiguities, in this case the informal requirement can be directly translated to:

\[
G \left( p \rightarrow X G (q \rightarrow \neg r \mathcal{U} t) \right)
\]

(9) Write a LTL formula which enforces the following requirement in a a transition system: *For every state \( s_i \) along an execution path \( s_1 s_2 s_3 \cdots \), unless \( s_i \) is the first state \( s_1 \), if \( p \) is true
in $s_i$, then $p$ must be true in at least one of the two states just before $s_i$, i.e., in the states $s_{i-1}$ and $s_{i-2}$.

**Answer:** The informal requirement in English can be directly translated to:

$$(Xp \rightarrow p) \land G (XXp \rightarrow p \lor Xp)$$

(10) Write a LTL formula which enforces the following requirement in a transition system: *In every odd state along an execution path $\pi = s_1s_2s_3\cdots$ the atom $p$ is true, and in every even state of the same path $\pi$ the atom $p$ is false.*

**Answer:** The informal requirement in English can be directly translated to:

$$p \land G (p \leftrightarrow X\neg p)$$

**Problem 4. CTL Definability.**

(11) Write a CTL formula which enforces the following requirement in a transition system: *At every state $s$, if $p$ is true, then at every state $s'$ reachable from $S$ in one transition or more, if $q$ is true, then $r$ is false until $t$ becomes true (for all continuations of the execution path starting at $s'$).*

**Answer:** In this case, it suffices to insert the appropriate quantifiers into LTL formula in (8):

$$\mathsf{AG} \left( p \rightarrow \mathsf{AXAG} (q \rightarrow \mathsf{A[\neg U t]}) \right)$$

(12) Write a CTL formula which enforces the following requirement in a transition system: *There exists a path $\pi$ such that, for every state $s$ on $\pi$, there exists a path $\pi'$ starting at $s$, which eventually enters a state $s'$ where $p$ is true and which, immediately after $s'$, enters another state $s''$ where $\neg p$ is true.*

**Answer:** The informal requirement in English can be directly translated to:

$$\mathsf{EF} \left( p \land \mathsf{EX} \neg p \right)$$

(13) Write a CTL formula which enforces the following requirement in a transition system: *There exists a state $s$ where atom $p$ is true and on all paths starting at $s$, atom $r$ is true as long as atom $q$ is not true.*

**Answer:** We can express that $r$ is true as long as $q$ is not true by the formula $rWq$, rather than by the formula $rUq$ which is more restrictive than the stated requirement. To say that $p$ is true and that all paths satisfy $rWq$, we write $p \land A (rWq)$. Because we need only one state in which this formula holds, we existentially range over all paths and require the formula to eventually hold in some state. We thus obtain:

$$\mathsf{EF} \left( p \land A (rWq) \right)$$
Write a CTL formula which enforces the following requirement in a transition system: For every path $\pi$ and in every state $s$ on $\pi$, atom $p$ is true iff two conditions: (1) atom $q$ is true and (2), in the state immediately preceding $s$, atom $r$ is true.

**Answer:** Atom $p$ cannot be true in the first state at which a path starts. We thus obtain:

$$\neg p \land AG \left( (r \rightarrow AX(p \leftrightarrow q)) \land (\neg r \rightarrow AX\neg p) \right)$$