1. CTL equivalence/non-equivalence

Solutions:

a) $\text{EF}p \land \text{EG}q$, $\text{EF}(p \land \text{EG}q)$
Not equivalent
Counter-example: $\text{EF}p \land \text{E}gq$ satisfied, but not $\text{EF}(p \land \text{EG}q)$

b) $\text{AF}p \land \text{AG}q$, $\text{AF}(p \land \text{AG}q)$
Not equivalent
Counter-example: $\text{AF}(p \land \text{AG}q)$ satisfied, but not $\text{AF}p \land \text{AG}q$

c) $\text{AF}p \land \text{AG}q$, $\text{AG}(\text{AF}p \land q)$
Not equivalent
Counter-example: $\text{AF}p \land \text{Ag}q$ is satisfied, but not $\text{AG}(\text{AF}p \land q)$
d) \( AFAGp \land AFAGq, AF(AGp \land AGq) \)
Equivalent
Justification:
   i) \( AFAGp \land AFAGq \Rightarrow AF(AGp \land AGq) \)
Suppose \( AFAGp \land AFAGq \) holds, then there is a state somewhere in all future paths at which \( p \) is true, and all states on all paths from that state have \( p \) true as well. Furthermore, we know that there is a state somewhere in all future paths with \( q \) true, and that all states on all paths from that state have \( q \) true as well. Then we see that it must be true that somewhere on all future paths there must be “an intersection”, that is, there must be a state where both \( p \) and \( q \) are true, and all paths from that state have both \( p \) and \( q \) true as well. Thus \( AF(AGp \land AGq) \) must also hold.

   ii) \( AF(AGp \land AGq) \Rightarrow AFAGp \land AFAGq \)
Suppose that \( AF(AGp \land AGq) \) holds, then it must be true that there is a state somewhere in all future paths at which \( p \) and \( q \) hold, and all states on all paths from that state have both \( p \) and \( q \) true as well. Thus, for each future path, we can choose the latter described state, and then it is true that \( p \) holds globally at that state, it is also true that \( q \) holds globally at that state. Therefore, \( AFAGp \land AFAGq \) must also hold.
Therefore, \( AFAGp \land AFAGq \) and \( AF(AGp \land AGq) \) are equivalent.

e) \( E[pUq] \land E[qUr], E[pUr] \)
Not equivalent
Counter-example: \( E[pUq] \land E[qUr] \) is satisfied, but not \( E[pUr] \)

f) \( A[pUq] \land A[qUr], A[pUr] \)
Not equivalent
Counter-example: \( A[pUq] \land A[qUr] \) is satisfied, but not \( A[pUr] \)
2. CTL formulas and M

\begin{figure}
\centering
\begin{tikzpicture}
  \node (s0) at (0,0) [circle, draw] {p,q};
  \node (s1) at (2,1) [circle, draw] {r};
  \node (s2) at (2,2) [circle, draw] {q,r};
  \node (s3) at (0,2) [circle, draw] {p,t};

  \path[->]
    (s0) edge [loop above] node {s_0} (s0)
    (s0) edge [bend right] node {s_1} (s1)
    (s1) edge [loop right] node {s_1} (s1)
    (s1) edge [bend right] node {s_2} (s2)
    (s2) edge [loop above] node {s_2} (s2)
    (s2) edge [bend right] node {s_3} (s3)
    (s3) edge [loop right] node {s_3} (s3)
    (s3) edge [bend right] node {s_0} (s0);
\end{tikzpicture}
\end{figure}

**Solutions:**

a) $\text{AF}q$

Holds.

$q$ is true at $s_0$, and the future includes the present, thus all future paths contain $q$.

b) $\text{AG}(\text{EF}(p \lor q))$

Holds.

This can be seen by noting that states $s_0$, $s_1$, $s_2$, and $s_3$ all satisfy $\text{EF}(p \lor q)$ – there is some state reachable from those states where either $p$ or $q$ is satisfied.

c) $\text{EX}(\text{EX}r))$

Holds.

Look at path $s_0$, $s_1$, $s_1$ - this path shows the existence of a state following $s_0$, immediately after which there is a state with $r$ true.

d) $\text{AG}(\text{AF}q))$

Does not hold.

To see this consider the path $s_0$, $s_1$, $s_1$, $s_1$, $s_1$, $s_1$, ($s_1$ repeating)…

e) $\text{AGEXE}[(p \lor r)Uq]$

Holds.

To see this, notice that $\text{EXE}[(p \lor r)Uq]$ holds for all states:

- $s_0$ – next state is $s_1$, then $E[(p \lor r)Uq]$ holds as $s_1$, $s_2$, $s_0$
- $s_1$ – next state is $s_1$, then $E[(p \lor r)Uq]$ holds as $s_1$, $s_2$, $s_0$
- $s_2$ – next state is $s_3$, then $E[(p \lor r)Uq]$ holds as $s_3$, $s_0$
- $s_3$ – next state is $s_0$, then $E[(p \lor r)Uq]$ holds as $s_0$, $s_3$

**Corrected Solution:**

f) $\text{AF}[(p \rightarrow r)Uq]$

Holds.

$A[(p \rightarrow r)Uq]$ is equivalent to $A[(\neg p \lor r)Uq]$.
To see this, we show that $A[(\neg p \lor r)Uq]$ all paths from $s_0$ satisfy this formula. The trick is that formally “$A[p U q]$” means that on all paths, $p$ occurs 0 or more times until $q$. Then we just note that $q$ is asserted in state $s_0$, and so $A[(\neg p \lor r)Uq]$ holds on every path.

3. CTL formulas for English properties

   **Solution:**

   a) “The event $p$ always precedes the event $q$.”
   $\neg E[\neg p U (q \land \neg p)]$

   b) “After $p$, $q$ is never true.”
   $AG(p \rightarrow AXAG\neg q)$

   c) “Between the events $q$ and $r$, $p$ is never true.”
   $[AG(q \rightarrow \neg EF(p \land EFr))] \land [AG(r \rightarrow \neg EF(p \land E))]$

4. Pseudo-code for TRANSLATE

   **Solution:**

   ```c
   function translate(formula F) {
     case (F) {
       F is T : return T;
       F is (bottom) : return \neg T;
       F is an atomic proposition : return F;
       F is \neg F1 : return ( \neg TRANSLATE(F) );
       F is F1 \land F2 : return (TRANSLATE(F1) \land TRANSLATE(F2) );
       F is F1 | F2 : return (\neg(TRANSLATE(\neg F1) \land TRANSLATE(\neg F2)));
       F is F1 \rightarrow F2 : return (TRANSLATE(\neg F1 \land F2) );
       F is AX F1 : return (TRANSLATE(\neg EX\neg F1) );
       F is EX F1 : return (EX (TRANSLATE(F1)));
       F is A[F1 U F2] : return (A[TRANSLATE(F1) U TRANSLATE(F2)]);
       F is E[F1 U F2] : return (E[TRANSLATE(F1) U TRANSLATE(F2)]);
       F is EF F1 : return ( E [T U TRANSLATE(F1)] );
       F is EG F1 : return (TRANSLATE(\neg AF\neg F1) );
       F is AF F1 : return ( A [T U TRANSLATE(F1)] );
       F is AG F1 : return (TRANSLATE(\neg EF\neg F1) );
     }
   }
   ```

5. Microwave modeling

   AG(Start $\rightarrow$ AF Heat)

   **Solutions:**

   a) Formula meaning

   - “In all states, it is true that if start holds in a state, the in some state on all future paths from that state, heat will eventually hold also”
   - We’re checking that if start is pressed, then the heat will eventually turn on.
b) Equivalent to \( \neg EF(\text{Start} \land \text{EG} \neg \text{Heat}) \)

\[
AG(\text{Start} \rightarrow \text{AF Heat}) = \neg EF(\neg(\text{Start} \rightarrow \text{AF Heat})) \quad \sim \text{Translate AG to EF}
\]

\[
= \neg EF(\neg(\neg \text{Start} \lor \text{AF Heat})) \quad \sim \text{Substitute} \rightarrow
\]

\[
= \neg EF(\text{Start} \land (\neg \text{AF Heat})) \quad \sim \text{DeMorgan’s law}
\]

\[
= \neg EF(\text{Start} \land \text{EG} \neg \text{Heat}) \quad \sim \text{Translate AF to EG}
\]

c) Does \( M,1 \models \phi \) hold?

<table>
<thead>
<tr>
<th>Subformula</th>
<th>Satisfied States</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heat</td>
<td>4, 7</td>
</tr>
<tr>
<td>( \neg \text{Heat} )</td>
<td>1, 2, 3, 5, 6</td>
</tr>
<tr>
<td>( \text{EG} \neg \text{Heat} )</td>
<td>1, 2, 3, 5</td>
</tr>
<tr>
<td>Start</td>
<td>2, 5, 6, 7</td>
</tr>
<tr>
<td>( \text{Start} \land \text{EG} \neg \text{Heat} )</td>
<td>2, 5</td>
</tr>
<tr>
<td>( EF(\text{Start} \land \text{EG} \neg \text{Heat}) )</td>
<td>1, 2, 3, 4, 5, 6, 7</td>
</tr>
<tr>
<td>( \neg EF(\text{Start} \land \text{EG} \neg \text{Heat}) )</td>
<td>none</td>
</tr>
</tbody>
</table>

So, the formula does not hold for state 1.