(a) Are the two following Linear time Temporal Logic formula equivalent? If yes, give a proof. If not, construct example traces to show that they are not equivalent.

\[ F(p \mathcal{U} q) \Leftrightarrow Fp \mathcal{U} Fq \]

You can assume that \( p \) and \( q \) are atomic propositions.

**Answer:** The two formulae are equivalent. Consider a trace \( \pi \) satisfying \( F(p \mathcal{U} q) \). Then by the definition of the \( F \), \( \mathcal{U} \) operators, there must exist a state in \( \pi \) which satisfies \( q \). Let the first position in \( \pi \) where \( q \) is true be \( k \). Then clearly \( \pi^k \models p \mathcal{U} q \), and hence \( \pi \models F(p \mathcal{U} q) \). Since \( \pi^k \models q \) We see that \( \pi \models Fq \); By definition of the until operator \( \pi \models \varphi \Rightarrow \pi \models \psi \mathcal{U} \varphi \) for any LTL properties \( \varphi, \psi \). Thus, \( \pi \models Fp\mathcal{U}Fq \).

Now, consider any trace \( \pi \) such that \( \pi \models Fp\mathcal{U}Fq \). Again it means that there exists \( k \geq 0 \) such that \( \pi^k \models Fq \) which means that there exists \( m \geq k \geq 0 \) such that \( \pi^m \models q \). Then \( \pi^m \models p\mathcal{U}q \) and hence \( \pi \models F(p\mathcal{U}q) \).

This concludes the proof of equivalence of the two formulae. In fact we see that any trace with at least one state in which \( q \) is true, satisfies both the formulae and vice-versa.
(b) In class, we discussed the nested depth-first search algorithm implemented inside the model checker SPIN. Among other things, this allows us to easily retrieve the counter-example trace from the stack. Suppose we implemented breadth-first search with queues instead for the purpose of model checking. Will the task of counter-example computation become any more difficult? Explain your answer.

**Answer:** In the nested depth-first search, the counter-example trace can be obtained by simply concatenating the two stacks. This will not be the case for the nested breadth-first search. In order to retrieve the counter-example trace in the nested breadth-first search we need to perform more book-keeping during the search. One possibility is to store a link at each state pointing to a predecessor state; this will allow the counter-example trace to be reconstructed when a violation is detected.
(c) Recall the definition of the Until operator $\mathbf{U}$ in Linear-time temporal logic (LTL). Let us now define a new until operator $\mathbf{U}_1$ as follows:

$M, \pi \models \varphi \mathbf{U}_1 \psi \equiv \text{ if there exists a } k \geq 0 \text{ such that } M, \pi^k \models \psi \text{ then for all } 0 \leq j < k \text{ we have } M, \pi^j \models \varphi$

The notation $\pi^k$ was discussed in class (and also appears in the textbook). Express $\varphi \mathbf{U}_1 \psi$ as a Linear-time temporal logic (LTL) formula and give explanation for your answer. You may assume that $\varphi, \psi$ are arbitrary LTL properties.

**Answer:** The definition is

$\varphi \mathbf{U}_1 \psi = (\varphi \mathbf{U} \psi) \lor G \neg \psi$

The only difference between $\mathbf{U}$ and $\mathbf{U}_1$ is that $\psi$ is not required to hold eventually in the definition of $\mathbf{U}_1$. This accounts for the disjunction in the definition of $\mathbf{U}_1$. 
(d) Assume \( p \) is an atomic proposition. Describe the following property in LTL: “along any path, a state satisfying \( p \) occurs at most once”. Explain your answer.

Answer:

\[
G \neg p \lor (\neg p U (p \land XG \neg p))
\]

\( G \neg p \) is true when \( p \) never occurs.

If \( p \) occurs exactly once then the path starting from the state in which \( p \) occurs must satisfy \( p \land XG \neg p \) (i.e. \( p \) occurs at the start and never occurs again). This explains the answer.