ITI8531 - Software Synthesis and Verification LTL Assignment

A railroad company wants to make a new controller for single-track railroad crossings. Naturally, they don't want any accidents with cars at the crossing, so they want to verify their controller. Their propositions include *train_is_approaching*, *train_is_crossing*, *light_is_flashing*, and *gate_is_down*. Using natural English, these are some properties we'd like to have true:

1. Whenever a train passing, the gate is down.

2. If a train is approaching or passing, then the light is flashing.

3. If the gate is up and the light is not flashing, then no train is passing or approaching.

4. If a train is approaching, the gate will be down before the train passes. In other words, if a train is approaching, then this train is not passing *until* the gate is down. However, this still allows that the gate could be back up by the time a train passes (thus only *until* does not suffice), and a train *might never* pass (you need to consider this also as additional option in order to ensure safety).

5. The gate will be up while train not passing infinitely many times.

To formalize such statements, we would start with the primitive propositions involved. These could be:

1. a (a train is approaching the crossing)

2. p (a train is passing the crossing)

3. l (the light is flashing)

4. g (the gate is down)

So, *inputs* are a, p and *outputs* are g, l.

Assignment:

1) Encode properties 1-5 in LTL.
2) Solve it using Acacia+ tool.
3) Write a report and explain in natural language:
    a. Properties 1-5
    b. The result of Acacia+ tool.
    c. See also other guidelines posted on the site.

Hints: 1) You need at least a combination of formulas correctly expressed using a relevant LTL operator in order for some previously defined formulas to be realizable. 2) You need to express each of the properties with the relevant LTL operator for each full formula. This is a safety critical system.