# ITI8531 - Software Synthesis and Verification
# LTL Exercises

A) Give an English translation of the following LTL formulae. Try to give a natural wording for each, not just a transliteration of the logical operators.

1. $(\Diamond r \rightarrow (p U r))$
2. $\Box(q \rightarrow \Box\neg p)$

B) In the following, give an LTL formula that formalizes the given English wording. If the English is subject to any ambiguity, as it frequently is, describe how you are disambiguating it, and why.

1. ``p is true.''
2. ``p becomes true before r.''
3. "p will happen at most once.''
4. ``p will happen at most twice.''
5. ``The light always blinks.'' Use the following proposition: p = the light is on.
6. p is false until r occurs, but r may not occur at all.
7. Always, after q occurs p must remain false.

C) A railroad 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_fashing, and gate_is_down.

Using natural English, these are some properties we'd like to have true:

1. 1. Whenever a train passing, the gate is down.
2. 2. If a train is approaching or passing, then the light is ashing.
3. 3. If the gate is up and the light is not ashing, then no train is passing or approaching.
4. 4. If a train is approaching, the gate will be down before the next train passes.
5. 5. If a train has nished passing, then later the gate will be up.
6. 6. The gate will be up infinitely many times.
7. 7. If a train is approaching, then it will be passing, and later it will be done passing with no train approaching.

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 ashing)
4. **g** (the gate is down)

Encode properties 1-7 in LTL.