SWE 619 Assignment 9
Fall 2019

Goal: Temporal Logic / Specification Patterns.

Note: You might find the Huth and Ryan text useful for some of these exercises:

M.R.A. Huth and M.D. Ryan. *Logic in Computer Science*. Second Edition. Cambridge University Press, 2004, ISBN 978-0-521-54310-1.
Although this text is no longer available through the university libarary services, enterprising students may be able to locate this document. Relevant material begins on page 183.

The non-bonus part of this assignment is a paper/pencil exercise only. Use the KSU slides covered in the lecture as a resource to help you formulate the following in CTL. You should use the variables listed in `typewriter font` as your primitive propositions.

1. It is possible to get to a state where `started` holds, but `ready` does not hold.
2. For any state, if a `request` (of some resource) occurs, then it will eventually be `acknowledged`.
3. A certain process is `enabled` infinitely often on every computation path.
4. Whatever happens, a certain process will eventually be permanently `deadlocked`.
5. From any state, it is possible to get to a `restart` state.
6. An upwards travelling elevator at the second floor does not change its direction when it has passengers wishing to go to the fifth floor.
   Use predicates such as `direction == up`, `floor == 2`, and `Button5Pressed`.
7. The elevator can remain idle on the third floor with its doors closed.
8. After `switchClosed` becomes true, `valueOpen` is never true.
9. After `q`, `p` is not true until `r`.
10. Variable `toggle` alternates between true and false on successive states.

For those interested in bonus points, consider the traffic light specification. Expand the SMV to model a more complex intersection. For example, you could consider an intersection with a left-turn arrow. Write SPEC clauses to capture the following property:

- (Safety): If the light for a given light is GREEN, then the lights for all conflicting diretions are RED. You should have one SPEC clause for each pair of conflicting directions. (You can, but don't have to, write any other SPEC clauses.)

Finally, verify these properties with SMV. You can download SMV in various forms from Carnegie Mellon or NuSMV.

As an additional example, you might find the mutual exclusion example of interest.

Note: NuSMV has deprecated processes; I'd be happy to have someone figure out the appropriate replacement.

Bonus credit (up to 3 points) **only** available if you demonstrate that you had SMV running.