Homework 3 Group 10
Nathan Daily daily043
Wyatt Kormick kormi001
Ross Faber faber047

Problem 1

1. Requirements:
    - At most one set of traffic lights(vertical or horizontal) shall be green or yellow at one point in time. The other lights must be red.
    - A pedestrian must be able to request access to walk across the street
    - The system shall sense when cars arrive at the intersection.
    - While an emergency vehicle is sensed, all lights shall be red.
    - A walk light for pedestrians shall be shown after the request access to walk button is pressed and there is a green light in the same direction
    - A no walk light for pedestrians shall be shown when there is a red light in the same direction as the pedestrian is walking and the request access to walk button has not been pressed
    - When a vehicle is sensed waiting at a light, and there are not other vehicles passing through the intersection, a trigger to change the other lights to yellow and then red and then the car's direction of lights will turn green.
    - A traffic light shall turn yellow before turning red and after turning green (ie Green→ Yellow → Red)

Assumptions: Emergency vehicles will go through red lights.
Walk sign will only show when the request walk button is pressed.
Left turns will be yield only

2. In file: traffic.smv included in our submission.

3. A G (!(hori_traf = green ∧ vert_traf = green))
   "At most one set of lights shall be green at any point in time"
   A G(!(hori_walk = green ∧ vert_traf = green) ∨ (vert_walk = green ∧ hori_traf = green))
   "A walk light for pedestrians shall be shown… there is a green light in the same direction" - This means that a pedestrian walk sign will only show when there is a green light going the same way.

4. G(hori_traf = red → F(hori_traf = green))
   When is a vehicle is sensed waiting at a light.. A trigger to change the the car's direction of lights to turn green
   G((hori_walk = red ∧ walk_request = true) → F(hori_walk = green ∧ request = false))
   A walk light for pedestrians shall be shown after the request access to walk button is pressed and there is a green light in the same direction, otherwise a no

Walk light will be shown

5. Anti-property: A (vert_traf = green U (!A X(vert_traf = yellow)))
   A yellow light will always turn green.
   Counter-example from NuSMV:
   -> State: 1.1 <-
   time_passed = 5
   hori_sensor = FALSE
   vert_sensor = FALSE
   walk_request_vert = FALSE
   walk_request_hori = FALSE
   emergency = FALSE
   mc.mode = HORIGREEN
   mc.walkmode = NOWALK
   mc.emergency_drive = FALSE
   mc.vert_cross = FALSE
   mc.hori_cross = FALSE
   mc.vert_slow = FALSE
   mc.hori_slow = FALSE
   mc.vert_drive = FALSE
   mc.hori_drive = TRUE
   -> State: 1.2 <-
   time_passed = 6
6. In File: transcript.txt
   Some of our conditions fail when they should not.

Problem 2

1. Statement Coverage

| Input (Number of Elements, {Sequence of Ints}) | Expected Value |
|---|---|
| (3,{1,2,3}) | {3, 2, 3, 1, 2, 3} |

2.

| Input (Number of Elements, {Sequence of Ints}) | Expected Value |
|---|---|
| (5 {0, 1, 0, 2, 0}) | {5, 4, 4} |

| | |
|---|---|
| (3,{1,2,3}) | {3, 2, 3, 1, 2, 3} |

3.

| Input (Number of Elements, {Sequence of Ints}) | Expected Value |
|---|---|
| (5 {0, 1, 0, 2, 0}) | {5, 4, 4} |
| (3,{1,2,3}) | {3, 2, 3, 1, 2, 3} |
| (1,{1}) | {1} |

For MC/DC coverage, all we need is (2,{0,2)}) with an expected value of 2 2, which covers the true false condition of the while loop, which then gives us MC/DC coverage.

Problem 3

13.1:

a)

i) All p-use, some c-use: A subpath from the definition of some variable to each use of the variable in conditionals, and if the variable is not used in any conditionals, a subpath from the definition to at least one computation using the variable.

ii) All c-use, some p-use: A subpath from the definition of some variable to each use of the variable in computations, and if the variable is not used in any computations, a subpath from the definition to at least one conditional using the variable.

b) All p-use, some c-use would require more test cases as it needs to cover all paths from definitions to use in conditionals, and the same is true for all c-use, some p-use but for use in computations.

13.2:

All DU Paths > All c-use, some p-use / All p-use, some c-use / All DU Pairs > All-Defs

All c-use, some p-use and All p-use, some c-use both require All Defs coverage to be considered covered but does not use all DU Pairs, as they only need all the paths for their respective uses, but could leave out ones for the other use.

All of these coverage types are checked on all definitions, so they all subsume All-Defs.

All DU Paths is all of the subpaths while the all/some coverages are certain subpaths, so they will be in the subpaths covered by All DU Paths.