Programming Project

Please submit your solution via email to the instructor with CC to yjk337@nyu.edu.

The deadline for Phase 1 is Tuesday, April 10.

Phase 1  Alloy Model of Minesweeper Game and Solver

In Phase 1 of the project you will develop an Alloy Model of a minesweeper game and solver.

(a) Develop an Alloy model of minesweeper board configurations for abstract minesweeper games. In an abstract minesweeper game the board is an undirected graph with nodes representing cells and edges indicating which cells are neighbors. Other than that, abstract minesweeper games are played exactly like normal minesweeper games. Think about what properties you want the field encoding the neighbor relation to hold and add appropriate facts to your model. Simulate some board configurations, e.g., you can write a simulation predicate that is true for boards that correspond to the two-dimensional grids of classical minesweeper games.

(b) Think about invariants of board configurations that can occur while playing a minesweeper game and add appropriate predicates to your model. For instance, during a minesweeper game, marked cells are never revealed. Another important property is that the number of marked cells never exceeds the number of mines on the board. There are more such invariants that you want to specify. Simulate some board configurations both satisfying and violating your specified invariants.

(c) Specify the operations on board configurations that you need for playing the game. These are: (1) mark a cell on the board, (2) unmark a cell on the board, and (3) reveal a cell on the board. You will also need to write a function that computes, for a given cell, the number of neighboring cells that contain a mine. Make sure that revealing a cell is propagated, i.e., if a cell with no neighboring mines is revealed then all neighboring cells are also revealed.

(d) Write assertions checking whether your operations preserve all the invariants of board configurations that you have specified. If some invariant is violated by an operation, add an appropriate pre-condition to the operation.

(e) Write a predicate that holds true for board configurations that are safe, i.e. do not contain a revealed mine. Write another predicate that holds true for configurations that are won, i.e., on which all unrevealed cells are marked. Use these predicates and your operations on minesweeper boards to simulate some minesweeper plays, both winning and losing.

(f) Write a predicate that holds true for consistent board configuration according to the Minesweeper Consistency Problem. Simulate some consistent and inconsistent boards. Use your consistency predicate to write a perfect minesweeper solver and use your solver to solve some minesweeper boards. Make sure that your solver does not cheat by directly
accessing the fields that encode which cells contain mines. You can do this by declaring these fields \textit{private} and putting your consistency predicate and the solver into a separate module.