Non-State Data Struct

Continuing on: Sudoku

- We'd defined neighborhoods
- Solved predicate: what does it mean for a board to be solved
    - Every row has every number represented in a column & v.v.
- To check that two predicates are equivalent:
    - Check that they imply each other (`iff`)
    - We can use the `check` syntax
    - What about when the predicates take arguments?
        - We might want to use a quantifier to talk about a board (`some b: Board`)
    - If we're looking for a counterexample with `run`, we need to stick a `not` in there
    - Alternatively, we can use `check`
- We can now generate two boards:
    - One as the original puzzle
    - One as the solution for the sudoku puzzle
    - Specifying that these two boards are not the same speeds up the evaluator

Notice that when we don't find a counterexample: Forge says "Assertion *may* be valid"

- Only true up to some bounds
- Forge is still useful! - it can look for some simple counterexamples
- It builds confidence