Harvard University
Computer Science 20

In-Class Problems 16

Friday, March 4, 2016

Authors: Crystal Chang and Ben Zheng

Executive Summary

1. A **state machine** is a binary relation called a *transition relation* on a set of elements S called *states*, visually depicted as nodes.
   - A transition from state q to r is denoted $q \to r$.
   - A state machine has a start state $q_0 \in S$, which is denoted with an arrow pointing to the start state's node.
   - A state machine may have one or more final states, which are denoted with double-circles around the nodes of the final states.
2. An *execution* of a state machine with a set of states S is a plausible sequence of states (possibly an infinite one) with the property that
   - the sequence starts with the start state $q_0 \in S$, and
   - for all consecutive states q and r, $q \to r$.
3. A state $s \in S$ is *reachable* if some execution of a state machine includes s.
4. A state machine *accepts* a string a if the execution of a state machine with input a ends on a final state of the state machine.
5. Predicate P is a preserved invariant of a state machine if for all $q, r \in S$,

$$P(q) \wedge (q \to r) \implies P(r)$$

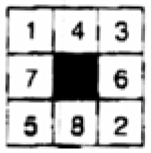
6. **The Invariant Principle**: If a preserved invariant of a state machine is true for the start state, then it is true for all reachable states.
7. A state machine is *deterministic* if for all states s in the machine, there is at most one transition out of s.
8. A state machine is *non-deterministic* if there exists a state s in the machine such that there is more than one transition out of s.

PROBLEM 1

Draw a state machine with states, labelled transitions, start state, and final state(s) that accepts exactly
(A) the binary string 101.
(B) binary strings that begin with the sequence 101.
(C) binary strings that begin and end with 1 (i.e. $1, 11, 101, 111, \ldots$).
(D) no strings (the empty set).

PROBLEM 2

We have a puzzle that looks like the following graph where we start on the black square in the center. Each move consists of moving to an adjacent square directly above, to the left, to the right, or below our current square.



(A) What's the size of the state space after 1 move?
(B) What's the size of the state space after 2 moves?

PROBLEM 3

In this problem we will use loop invariants to prove the correctness property that $y = c$ (for $c > 0$) after the following loop terminates.

```
• x=c; y=0;
• while ($x > 0$):
      $x--$ (subtract 1 from x)
      $y++$ (add 1 to y)
```

(A) Construct a loop invariant for the proof.
(B) Use induction to prove that the loop preserves the invariant.
(C) Use loop invariants to prove the correctness property that $y = c$ (for $c > 0$) after the loop terminates.

PROBLEM 4

(BONUS) Consider the following piece of code where n is a positive integer:

```
• $y = 0$; $i = 0$
• while ($i < n$):
      $y += 2^i$
      $i$++
```

(A) Compute the value of y after the 0th, 1st, 2nd, and 3rd iterations. From these results, can you guess what the value of y would be after the loop terminates?
(B) Use your result from (A) to construct a loop invariant.
(C) Use induction to prove that the loop preserves the invariant.
(D) Use the loop invariant to prove the correctness of your guess in (A).