Class 21: Review

Exam 2

Exam 2 will be in class on Thursday, 9 Nov. See Class 20 notes for details on the exam.

Main Topics for Review

Today we review the topics that we learned after Exam 1 with the exception of number theory (which will not be included in Exam 2).

– State Machines and how to argue about correctness of programs.

– Recursive Definitions and how to prove statements about them using structural induction.

– Infinite Sets and Cardinalities, and how to show sets are finite, infinite, countable, or uncountable.

State Machines

$M = (S, G \subset S \times S, q_0 \in S)$ defines a state machine.

P is a *preserved invariant* if:

$$\forall q \in S.(P(q) \wedge (q \to r) \in G) \implies P(r)$$

Invariant Principle: If P is a *preserved invariant* and $P(q_0)$ is true, then property P is true for all **reachable states**.

Proving Program Correctness

To prove a program R produces the correct output:

1. Model it as a state machine, M.
2. Show that M eventually terminates.
3. Show partial correctness:
   - Find a suitable preserved invariant P for M.
   - Show that $P(q)$ for all final states implies the output correctness property. (Final states are states where the execution terminates.)
   - Show $P(q_0)$ — the perserved invariant holds for the start state.

Recursive Data Types

To define a recursive data type D:

– Define one or more **base** objects, $d \in D$.

– Define one or more **constructor** cases that specify how to construct a new object $d \in D$ from one or more previoulsy-constructed objects, $d_1, d_2, \ldots \in D$.

