Assignment 6: Temporal Logic and Cardinality
Due: 11:00am, Wednesday, October 31

- Emails about this assignment should be directed to cs81help@cs.hmc.edu.
- The usual collaboration rules apply. You may discuss an exercise with any other student(s) currently taking CS 81 as long as:
  - You contribute equally;
  - You come away from this discussion only with understanding in your head — no written materials or computer notes may be retained;
  - Your submission is authored solely by you, on a separate occasion.
- You should refer only to materials from this semester of CS 81 (lecture notes, handouts, textbooks, grutors, profs, etc.).
- Bring a writeup/printout to class on the due date. Illegible answers will get no credit.
- Make sure your submission includes your name!

1 Temporal Logic

(CTL was described in class. It also appears in Section 3.4 of Huth & Ryan.)

<table>
<thead>
<tr>
<th></th>
<th>Eventually True</th>
<th>Invariably True</th>
</tr>
</thead>
<tbody>
<tr>
<td>Some Path</td>
<td>EF $p$</td>
<td>EG $p$</td>
</tr>
<tr>
<td></td>
<td>Possibly</td>
<td>Potentially Always</td>
</tr>
<tr>
<td>All Paths</td>
<td>AF $p$</td>
<td>AG $p$</td>
</tr>
<tr>
<td></td>
<td>Inevitably</td>
<td>Invariably</td>
</tr>
</tbody>
</table>
Consider the possible execution paths in the following piece of C/C++/Java-style code:

```c
int a[100] = { 0, 0, 0, ..., 0 }; // Array is initially all zeros

while (true) {
    int k = user_input(); // Assume this user input will be a number
                             // between 0 and 99 inclusive!
    if (a[k] == 1) {
        // Reset the array back to zeros
        for (int i = 0; i < 100; ++i)
            a[i] = 0;
    } else {
        // Set the k-th element of the array to 1
        a[k] = 1;
    }
}
```

Which of the following logical propositions are true (starting at the beginning of the program)? Explain your answer; you do not need to provide a full formal proof, but your explanation should be clear and convincing.

1. AG (a[42] = 0)
2. AG (\( \sum_{j=0}^{<100} a[j] < 100 \))
3. EF (a[42] = 1)
4. AF (a[42] = 1)
5. EG (a[42] = 0)
6. AG (AF (a[42] = 0))
7. AG (EF (a[42] = 1))
8. EF (AG (a[42] = 0))
2 Cardinality

Carefully argue whether the following sets are countable or not.

1. The set of partial functions from $\mathbb{N}$ to $\mathbb{N}$ whose support is a finite set.

   [A total function is one that, for any input, produces an output. A partial function is one that, for some or all particular inputs, always returns “don’t know” or “undefined” (often written $\perp$ although it’s not a truth value) rather than returning an output. The support of a partial function (sometimes called the domain or domain of definition) is the collection of inputs that provide non-$\perp$ output.

   For example, we could have a partial function $\sqrt{\cdot} : \mathbb{R} \to \mathbb{R}$ defined by

   \[
   \sqrt{x} := \begin{cases} 
   \sqrt{x} & \text{if } x \geq 0; \\
   \perp & \text{otherwise.}
   \end{cases}
   \]

   The support of $\sqrt{\cdot}$ is then the set of all nonnegative real numbers.]

2. The set all valid Hoare Logic proofs.

3. The set of flow networks with rational capacities (directed graphs with finitely many nodes and edges, where each directed edge is labeled with a rational number called the “capacity” of that edge).

4. The set of Java programs that typecheck (i.e., that successfully compile to a Java .class file)