Quiz 12 Solutions

written by Alvin Wan . alvinwan.com/cs70

Thursday, March 3, 2016

This quiz does not count towards your grade. It exists to simply gauge your understanding. Treat this as though it were a portion of your midterm or final exam. "Intuition Practice" might be tricky; watch out for subtleties. "Proofs" will be challenging to start; develop an arsenal of *approaches* to starting a problem.

1 Proofs

For each of the following, determine whether or not the problem is decidable, and justify using a formal proof.

Solution: In this quiz, note that we model all solutions as reductions *from* the Halting Problem. This means that

1. We assume for contradiction that this program exists.
2. We solve the Halting Problem using this mystical program.
3. The halting problem cannot be solved. Contradiction. Thus, this mystical program cannot exist.

1. Determining if your CS61A quiz submission S is correct.

   Solution: Undecidable

   Assume for contradiction that such a program `OKPY` exists. Then, we can construct the following program Q:

```
def Q(P):
    P()
    return 'Correct answer'
```

`OKPY` called on Q returns `True` iff Q returns the correct answer, which happens iff P halts. Thus, we can call `OKPY(Q, P)` to test if any program P halts. We have solved the halting problem. Contradiction.

2. Determining if a program P terminates on line L

   Solution: Undecidable

   Assume for contradiction that such a program `TERMINATE-ON-LINE` exists. Let us construct a program Q.

```
1 def Q(P):
2     P()
3     return True
```

   `TERMINATE-ON-LINE` returns `True` iff Q reaches line 2, which only occurs if P halts. Thus, call `TERMINATE-ON-LINE(Q, P, 2)` to test if any program P halts. We have solved the halting problem. Contradiction.

3. Determining if a program is a virus

   Solution: Undecidable The logic here is identical to that of 1. Replace `OKPY` with `VIRUS-CHECKER`.

4. Determining if a program contains a loop, given finite memory.

   Solution: We can write a program that enumerates all possible states for a computer program with finite memory. Track all states that a computer's memory goes through; if there is ever a repeated state, the program has a loop.

5. Determining if a program properly shuts an automatic door.

   Solution: Undecidable The logic here is identical to that of 1. Replace `OKPY` with `DOOR-SHUT-CHECKER`.