Assignment 12: (Un)decidability

Due: Wednesday, December 7

- 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. Review Chapter 21 of Rich, paying particular attention to the examples. Come up with (at least) two questions about the reading where you're not sure of the answer. These may relate to points where the book is confusing, or simply to some related question or conjecture that occurs to you while doing the reading.

2. Consider the following languages (where M ranges over TMs). Are they semidecidable? Are they decidable? Prove each answer.

   $L_1 = \{ \langle M \rangle \mid M \text{ accepts at least 999 strings} \}$
   $L_2 = \{ \langle M, w \rangle \mid M \text{ never moves right twice in a row while running on input } w \}$
   $L_3 = \{ \langle M \rangle \mid M\text{'s program includes a transition that writes a on the tape} \}$
   $L_4 = \{ \langle M \rangle \mid M \text{ eventually writes a non-blank symbol to the tape when given input } \varepsilon \}$
   $L_5 = \{ \langle M \rangle \mid M \text{ accepts the string a} \}$
   $L_6 = \{ \langle M \rangle \mid M \text{ is the only TM accepting } L(M) \}$
   $L_7 = \{ \langle M \rangle \mid M \text{ halts on input } \epsilon \text{ in no more than 1000 steps} \}$

3. We know that A_{TM} is semidecidable but not decidable, and its complement $\neg A_{TM}$ is neither semidecidable nor decidable. Is the third language

   $$R_{TM} = \{ \langle M, w \rangle \mid M \text{ rejects } w \}$$

   decidable, semidecidable, or neither? Prove your answer.

4. Although the language

$$\mathit{ADD} = \{\, x{=}y{+}z \mid x, y, z \text{ are binary integers, and } x \text{ is the sum of } y \text{ and } z \,\}$$

can easily be shown not to be regular, you can surely write a program (in Python, or Java, or Turing-Machine, etc.) that takes a string and decides whether it's one of these correct sums of binary numbers; *ADD* is a decidable language.[1]

Now, even if you restrict yourself to a single programming language, (we'll use Turing Machines), there might be many programs to decide *ADD*; they each decide whether the input is a correct binary sum or not, each in slightly different (or very different) ways. We can then consider the set *ADDERS* of all such programs.

(a) Prove that the language

$$\mathit{ADDERS} = \{\, \langle M \rangle \mid M \text{ is a Turing Machine that accepts the language } \mathit{ADD} \,\}$$

is not decidable.

(b) Prove that *ADDERS* is also not semidecidable by showing $\neg A_{TM} \leqslant \mathit{ADDERS}$. That is, reduce $\neg A_{TM}$ to *ADDERS* by showing that if *ADDERS* were *semidecidable*, then $\neg A_{TM}$ would be semidecidable too. (You may assume that *ADD* is decidable, i.e., there is at least one Turing Machine that decides it.)

(c) If we assume our encoding of Turing Machines is such that every string represents a Turing Machine, then the complement of *ADDERS* is the set

$$\neg\mathit{ADDERS} = \{\, \langle M \rangle \mid L(M) \neq \mathit{ADD} \,\}$$

Is $\neg$*ADDERS* decidable? If not, is it semidecidable? Prove your answers.

[1] As we've discussed a couple of times now, if *ADD* is decidable then addition is computable; if you want to add 3 and 5, consider all binary number s in increasing order, check whether "$s{=}\ 11 + 101$" is in *ADD*, and stop when you find an s where the answer is yes.