CSE 531: Computability and Complexity — Autumn 2002

Problem Set #1 — Instructor: Venkatesan Guruswami

Due on **October 17, 2002** in class.

Reminder: If you haven't done so already, subscribe to CSE 531 email group ASAP by visiting `http://majordomo.cs.washington.edu/mailman/listinfo/cse531`.

Instructions: You are allowed to collaborate with fellow students taking the class in solving problem sets. If you do so, please indicate for each problem the people you worked with on that problem. You are expected to refrain from looking up solutions from websites or other literature.

The solutions do not have to be *completely* formal (to the extent of writing down Turing machine code, for example), but the arguments must still be convincing. Most of the problems only require one or two key ideas for their solution – spelling out these ideas should give you most of the credit for the problem even if you err in some finer details. So, make sure you clearly write down the main idea(s) behind your solution even if you could not figure out a complete solution.

---

1. Problem 3.13, Sipser's book (Turing machines with stay put instead of left)
2. Problem 3.15, Sipser's book (Closure of Turing-recognizable languages under union, concatenation, star, intersection)
3. Problem 3.9, Sipser's book (The power of k-PDA's for $k = 0, 1, 2, 3$)
4. Problem 3.16, Sipser's book (A language is decidable if and only if some enumerator enumerates it in standard order)
   - Note that Sipser's book uses the terminology lexicographic order to mean the familiar dictionary order except that shorter strings precede longer strings. Thus, the lexicographic ordering of all strings over $\{0, 1\}$ is $\{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$. We refer to this ordering as the standard order.
5. Define the language

$$A = \{\langle M\rangle |\ M \text{ is a DFA that accepts } \textit{some} \text{ binary string with an equal number of 0's and 1's}\} .$$

   Prove that A is decidable.
6. Problem 4.17, Sipser's book. (A language C is Turing-recognizable iff there exists a decidable language D such that $C = \{x : \exists y(\langle x, y\rangle \in D)\}$)
7. * **(Optional problem)** In this problem, we strengthen the result of (one part of) Problem 3. Instead of k-PDAs, we consider k-counter machines.

   A k-counter machine (k-CM, for short) is a Turing machine with a read-only input tape that contains the input to the machine (assume that the ends of the input tape are marked with special symbols). The k-CM can move its head on the input tape in either direction (i.e. left or right), but it cannot write anything onto the tape. For its writable memory, a k-CM is given access to k counters $C_1, C_2 \ldots, C_k$ each of which can hold a non-negative integer. The k-CM cannot access the contents of the counters, but can check whether $C_i = 0$ for each i, $1 \leq i \leq k$. It can thus find out which subset of the counters, if any, are zero. A k-CM computes as follows: initially, the tape head is at the left end of the input, the control is in the start state, and all counters equal zero. In a single step, depending upon its current state, the symbol under the head on the input tape, and the subset of counters which are zero, a k-CM:

(i) changes its finite control to the appropriate next state;

(ii) moves its input tape head to the left or right by one cell; and

(iii) decrements or increments one of the k counters by 1 (assume that attempts to decrement a zero counter do not affect the counter)

As with a Turing machine, a k-CM accepts by entering a special accept state, rejects by entering a special reject state, and could also run forever without halting. Now, to your exercise.

- Prove that a 2-counter machine can simulate an arbitrary Turing machine. Conclude that 2-counter machines recognize precisely the class of Turing-recognizable languages.

Suggestion: First, show how two counters can be used to simulate an arbitrary stack. Then use what you showed in Problem 3 to conclude that 4-counter machines can simulate Turing machines. Finally, does one really need four counters? – try to simulate their functionality using just two counters (hint: use a one-to-one map from $\mathbb{N}^4$ to $\mathbb{N}$, where $\mathbb{N}$ is the set of positive integers).