Exercises

We will attempt some of these in class (probably 3/31), and I’ll ask for your ideas. You are welcome to discuss these with your classmates, and you do not have to write these up.

1. Exercise 5.2 (EQ_{CFG} is co-Turing-recognizable).
2. Exercise 5.3 (solve a small PCP example).
3. Exercise 5.4 (no, find a counterexample).
4. Exercise 5.6 (answered in book!).
5. Supposing $L$ is decidable, show $L^*$ is decidable.
6. Read about Rice’s Theorem (Problem 5.38, solved in the book). Does it apply to the following language?
   $REC_{TM} = \{ \langle M \rangle : M \text{ is a TM and } L(M) \text{ is recognizable} \}$.
7. Suppose an enumerator for $L$ prints its strings in length sorted order. That is, it never prints a string which is shorter than a previously printed string. Argue that $L$ is decidable.

Problems

Submit written solutions to four of the following problems by the duedate above. Again you may have a partner, and turn in a single problem set with both names, but you cannot use a partner you had before. You may also ask me (MG) for clarifications, basic ideas, and guidance on proof-writing. This work should otherwise be your own; do not discuss solutions with your classmates, nor seek solutions elsewhere. Try to keep your work neat and well organized, mathematical arguments should be convincing.

Problem 1. Argue that if $L$ is recognizable, then so is $L^*$.

Problem 2. In class we looked over the proof of Theorem 5.3, and saw that (with a bit of re-interpretation) it describes a reduction $A_{TM} \leq_m REGULAR_{TM}$. Find another reduction, $A_{TM} \leq_m REGULAR_{TM}$. Conclude that REGULAR_{TM} is neither recognizable nor co-recognizable.
Problem 3. Argue that some unary language $L \subseteq \{a\}^*$ is recognizable but not decidable. (Hint: $A_{TM}$ is a subset of $\{0,1\}^*$. Find a computable bijection between $\{a\}^*$ and $\{0,1\}^*$, and exploit it somehow.)

Problem 4. Let $A$ and $B$ be recognizable languages such that $A \cup B = \Sigma^*$. Show there exists a decidable language $C$ such that $B \subseteq C \subseteq A$. (Hint: mimic the proof of Theorem 4.22, which handles the special case $A = B$.)

Problem 5. Given languages $A, B \subseteq \Sigma^*$, define their “quotient” as the language $A/B = \{x \in \Sigma^* : \exists y \in B, xy \in A\}$.

5(a). Show that if $A$ is regular, then so is $A/B$. (Hint: modify the DFA for $A$. $B$ is arbitrary, it could be undecidable!)

5(b). Show that if $A$ and $B$ are recognizable, then so is $A/B$.

5(c). Show an example where $A$ and $B$ are decidable, but $A/B$ is not. (Hint: history strings. See if you can make $B$ regular.)

Problem 6. Let $\mathbb{N} = \{0, 1, 2, \ldots\}$, the natural numbers. Here we define a function $B : \mathbb{N} \rightarrow \mathbb{N}$ (similar to problem 5.16). For $k \in \mathbb{N}$, consider the finitely many TM’s with state set $Q = \{1, \ldots, k\}$ and tape alphabet $\Gamma = \{0, 1, \_|\}$. Consider computations of such machines which halt, when started on a blank tape. Let $B(k)$ be the maximum number of steps in such a computation (or $B(k) = 0$, when there are no such machines or computations). Argue that the function $B$ is not computable. (Hint: go read “busy beaver” on wikipedia, and argue that if we could compute $B$, then we could solve the halting problem.)

Problem 7. Suppose $K(x)$ is as defined in Chapter 6, and we have access to an oracle for $A_{TM}$. Describe an algorithm which computes $K(x)$ from $x$, using the oracle. I will sketch how you can do it using $O(2^k)$ oracle calls, where $k = K(x)$. See if you can do better than that: do it using $O(k)$ oracle calls. Or even better, use only $O(\log k)$ oracle calls.