CS 21 Decidability and Tractability **Winter 2025**

Midterm

Out: February 5 *Due:* **February 12**

This is a midterm. You may consult only the course notes and the text (Sipser). *You may not collaborate.* The full honor code guidelines can be found in the course syllabus.

There are 5 problems on 2 pages. Please attempt all problems. **Please turn in your solutions via Gradescope, by 1pm on the due date.** Good luck!

1. Identify each of the following languages as either (i) regular, (ii) context-free but not regular, or (iii) not context-free. For each language, prove that your classification is correct, using the techniques we have developed in this course.

   (a) $L_1 = \{a^i b^j c^k : j \leq i + k\}$.

   (b) $L_2 = \{a^i b^j c^k : i = j = k \text{ or } i > 1000\}$.

   (c) $L_3 = \{a^i b^j c^k : i = j = k \text{ or } i < 1000\}$.

2. A state q of a deterministic Turing Machine M is *reachable* if there is some string w for which M's computation on w enters state q. Otherwise, it is *unreachable*. Consider the following language:

   $$L = \{\langle M \rangle : \text{Turing Machine } M \text{ has an unreachable state.}\}$$

   (a) Determine if L is RE or co-RE, and prove that your classification is correct.

   (b) Prove that L is undecidable. Hint: you may want to take a given Turing Machine M, and modify it so that it accepts the same set of strings, but before it enters q_{accept}, it visits *all* of its states. To do this, introduce a new character % to the tape alphabet, and new transitions that take the machine on a tour of its states when it reads this special character... If you are stuck, you can take this transformation as a given and describe a reduction using it, for partial credit.

3. Two (disjoint) languages L_1 and L_2 are called *recursively separable* if there is a decidable language D for which $L_1 \cap D = \emptyset$ and $L_2 \subseteq D$; they are *recursively inseparable* if no such decidable language D exists. Convince yourself that an undecidable language and its complement are recursively inseparable.

   Consider the following languages:

   $$\begin{aligned} L_1 &= \{\langle M \rangle : M \text{ halts and accepts input } \langle M \rangle\} \\ L_2 &= \{\langle M \rangle : M \text{ halts and rejects input } \langle M \rangle\} \end{aligned}$$

   Prove that L_1 and L_2 are recursively inseparable. Hint: your proof will probably involve supplying a Turing Machine its own description as input.

4. A *right-linear* CFG is a context-free grammar in which every production has the form

   - $A \to xB$, or
   - $A \to x$,

   where A and B are non-terminals, and x can be any string of terminals. A CFG is *linear* if productions of the form $A \to Bx$ are allowed in addition to the two types of productions in a right-linear CFG.

   (a) Prove that every language generated by a right-linear CFG is regular.

   (b) Prove that every regular language is generated by some right-linear CFG.

   (c) Give a linear CFG that generates the non-regular language

   $$L = \{a^n b^n : n \geq 0\}$$

   and prove that your grammar indeed generates exactly L (i.e., prove that every string in L is generated by your grammar, and prove that every string generated by your grammar is in L).

5. Given a language L, define AT-LEAST-50_L as follows:

   $$\text{AT-LEAST-}50_L \quad = \quad \{\#x_1\#x_2\#\cdots\#x_k\# : k \geq 0 \text{ and } |\{i : x_i \in L\}| \geq 50.\}$$

   Prove that AT-LEAST-50_L is RE if L is RE. Here the x_i are strings over L's alphabet, and # is a symbol that is not in L's alphabet.