Homework 5

COMS W3261, Summer B 2021

This homework is due **Monday, 8/2/2021, at 11:59PM EST**. Submit to GradeScope (course code: X3JEX4).

Grading policy reminder: LaTeX is preferred, but neatly typed or handwritten solutions are acceptable. Your TAs may dock points for indecipherable writing. Proofs should be complete; that is, include enough information that a reader can clearly tell that the argument is rigorous.

The tool `http://madebyevan.com/fsm/` may be useful for drawing finite state machines.

If a question is ambiguous, please state your assumptions. This way, we can give you credit for correct work. (Even better, post on Ed so that we can resolve the ambiguity.)

1 Problem 1 (15 points)

Use the context-free pumping lemma to prove that the following languages are not context-free. (Recall that the CFPL has the form "In any context-free language, all sufficiently long strings have some decomposition that satisfies three properties". To show that a language is not context-free, we need to contradict this statement; i.e., show that "In this particular language, there exists at least one long string such that *no* decomposition of this string satisfies all three properties.")

1. (5 points).
$$L_1 = \{a^n b^n a^n b^n \mid n \geq 1\}.$$

2. (10 points).
$$L_2 = \{a^i b^j c^k \mid i, j, k \geq 1; ij = k\}.$$

2 Problem 2 (8 points)

Provide *implementation-level descriptions* of Turing Machines that recognize the following languages.

(Recall that an implementation-level description is less detailed than a *formal description*, in which we specify every state, transition function, and so on. You do not need to specify specific states or the transition function for this question. An implementation-level description is more detailed than a *high-level description*, in which we describe an algorithm but ignore all details of how we move the machine head or manage storage on the tape. For this question, you should describe in prose how the TM moves its head around and manages memory. Answers should be a few paragraphs (certainly more than one sentence and less than two pages.))

(Example answer for the language $\{w\#w^R : w \in \{0,1\}^*\}$: First, we read the input string from left to right and ensure it contains the # symbol. We reject if it does not. Then, beginning at the hash symbol, we shuttle back and forth between the two substrings on either side. Every time we go to the left, we cross out and remember the first uncrossed symbol we come to. If it matches the first uncrossed symbol on the right, we cross out this symbol also and continue. If it does not match, or if there are no more uncrossed symbols on the right, we reject. Once we have completed this procedure for all symbols on the left substring, we accept if and only if we have crossed out all symbols on the right substring.)

1. (4 points).

$$L_3 = \{0^n 1^n 2^n \mid n \geq 1\}.$$

2. (4 points).

$$L_4 = \{a^n b^{n^2} \mid i \geq 1\}.$$

3 Problem 3 (12 points)

Provide *high-level descriptions* of Turing Machines that recognize the following languages.

(Recall that a high-level description is an algorithm for a Turing Machine described in prose, ignoring implementation details such as the way information is encoded or where the head needs to move. However, your TM's behavior should still be *precisely specified*: it should be clear what the Turing Machine does in every case. These answers should be on the order of a few sentences.)

(Example for the language $\{\langle G\rangle \mid \langle G\rangle$ encodes a connected graph.$\}$: First, check to see if the input encodes a graph and reject if not. Then, begin a list with an arbitrary vertex v. Add every neighbor of v to the list, then every neighbor's neighbor, and so on. Repeat this process until no new vertices are added and accept if the list contains every vertex.)

1. (4 points). The Fibonacci sequence F begins by defining $F_0 = 0$ and $F_1 = 1$, and then recursively defines F_i for all $i \geq 2$ as $F_i = F_{i-1} + F_{i-2}$.

$$L_5 = \{w \mid w \text{ is a number in the Fibonacci sequence}\}.$$

2. (4 points). Let D be any DFA.

$$L_D = \{w \mid D \text{ accepts the input string } w\}.$$

3. (4 points). The following language consists of strings over $\{0, 1\}$ and the symbols $\cup$, *, $^+$, etc., that we usually use to write down regular expressions.

$$L_R = \{w \mid w \text{ is a well-defined regular expression over } \{0, 1\}\}.$$

4 Problem 4 (Extra credit, up to 10 points)

1. (2 points). P and NP refer to classes of languages that we will define next week. The question of whether $P = NP$ is a famous unresolved question in theoretical computer science.

   Consider the language L_6 defined as follows.

   $$L_6 = \begin{cases} \{0\} \text{ if } P \neq NP \\ \{1\} \text{ if } P = NP \ . \end{cases}$$

   Is this language decidable? (Hint: you don't need to know what P and NP are to answer this question.)

2. (3 points). Give a high-level description of a Turing machine that recognizes the following language.

   $$L_7 = \{\langle M\rangle \mid \langle M\rangle \text{ encodes a TM that accepts the input string `ACCEPT ME'}\}.$$

3. (1 point). In the previous part, you showed that L_7 was Turing-recognizable. Why is it harder to show that L_7 is decidable?

4. (4 points). Recall that an *enumerator* is a Turing Machine with an attached printer that can print the contents of its tape at any point. The language of an enumerator is the set of all strings that it prints. Let G be a context-free grammar. Give a high-level description of an enumerator that writes down every string in the language of G.