CISS362: Introduction to Automata Theory, Languages, and Computation
Final Exam

The takehome final exam is due Thursday 5PM. You must turn in a printed pdf in person, not by email.

There are two version of the final exam. You either attempt Q1-7 or you attempt Q8 but not both. For Q8, you only need to complete one of the three parts.

You cannot attempt both Q1-Q7 and Q8. This means that if you have *any* attempt at all for Q1-Q7 and *any* attempt for Q8, you will get an immediate zero.

You must writing clearly, concisely, and in complete sentences, using proper mathematical notation. In case of ambiguity, I get to choose the meaning of what you're trying to say!

Q1. Let $\Sigma = \{0, 1\}$. Define the following language:

$$L = \{x \mid x \text{ contains an equal number of occurrences of 01 and 10}\}$$

Either prove L is regular (by constructing a DFA/NFA or a regex) or prove that it is not regular using the Pumping Lemma for regular languages.

SOLUTION.

Q2. Given languages L, L' (over Σ), define the shuffle of L, L' to be

$$\text{Shuffle}(L, L') = \{x_1 y_1 x_2 y_2 \cdots x_n y_n \mid x_i \in L \text{ and } y_i \in L' \text{ for } i = 1, \ldots, n\}$$

Suppose that $M = (\Sigma, Q, q_0, \delta, F)$ is a DFA for L and $M' = (\Sigma, Q', q'_0, \delta', F')$ is a DFA for L'. Construct an NFA that accepts Shuffle(L).

[This proves that Shuffle is a closed regular operator, i.e., this proves that if L, L' are regular, then Shuffle(L, L') is also regular.]

SOLUTION.

Q3. Let $\Sigma = \{a, b, c\}$. Prove that the following language (defined over Σ)

$$L = \{uvuvu \mid u \in \{a,b\}^*, v \in \{b,c\}^*\}$$

is not regular.

[HINT: Clean up and then use the pumping lemma for regular languages.]

SOLUTION.

Q4. Prove that the following languages are context-free by constructing the a context-free grammar for L_1 and a push-down automata for L_2.

(a) $L_1 = \{a^m b^m c^n \mid m \geq 0, n \geq 0\}$
(b) $L_2 = \{a^m b^n c^n \mid m \geq 0, n \geq 0\}$

Finally:

(c) What is $L_1 \cap L_2$? [Use set notation.]
(d) TRUE or FALSE: The intersection of any two context-free languages gives a context-free language.

SOLUTION.

Q5. Prove that

$$L = \{a^m b^n \mid m \neq n \text{ and } 2m \neq n\}$$

is a context-free language.

SOLUTION.

Q6. Suppose L is a context-free language and L' is regular. Show that $L \cap L'$ is a context-free language. Specifically, if you're given a PDA diagram of L and a DFA diagram for L', describe informally how to draw a PDA diagram for $L \cap L'$.

[... it's even better if you can describe the construction of $L \cap L'$ formally but you don't have to.]

SOLUTION.

Q7. (a) A Turing machine M is considered a DFA1 if the read/write head always moves to the right and it halts (i.e., enters its q_{accept} or q_{reject}) when it reads a space (or blank) character on the input tape. Note that a DFA1 is like a DFA except that it must have exactly 1 accept state. (A DFA can have any number of accept states.) Is the following language:

$$\text{ACCEPT}_{\text{DFA1}} = \{\langle M\rangle\#\langle w\rangle \mid M \text{ is a DFA1 and accepts } w\}$$

a Turing–decidable language? If it is, explain clearly how to construct a Turing decider (i.e., a Turing machine that always halts) that accepts the language. Otherwise prove that it's not decidable.

(b) Consider the following language:

$$\text{NONEMPTY}_{\text{TM}} = \{\langle M\rangle\#\langle w\rangle \mid M \text{ is a Turing machine and } L(M) \neq \emptyset\}$$

Is $\text{NONEMPTY}_{\text{TM}}$ a Turing–decidable language?

[This is the only TM question but it actually requires very little TM knowledge. You have everything you need to solve this problem if have a general understanding of TMs and you have paid attention to the Wednesday and Friday classes during the last week.]

SOLUTION.

(a)

(b)

Q8. This is the question for the second version of the final exam. You only need to attempt one of the three parts. Do *not* turn in any work for Q1-Q7 if you plan to turn in work for Q8, or you will get an immediate 0.

1. Given two words $x, y \in \Sigma^*$, a DFA M separates x, y if L accepts either x or y but not both. What is the smallest number of states you need to construct a DFA that can separate any pair of distinct words of length $\leq n$?

2. Consider the following problem: Given a PDA M and an integer n, is it true that $\Sigma^n \subseteq L(M)$? The corresponding language is

$$L = \{\langle M \rangle \# \langle n \rangle \mid M \text{ is a PDA and } \Sigma^n \subseteq L(M)\}$$

   Is L regular, context-free, Turing–decidable, Turing–recognizable, not Turing–recognizable?

3. Either prove P = NP or P $\neq$ NP.

SOLUTION.