PRACTICE Exam 1

Name: ________________________________

NOTES:

- This is a practice exam. It will not be graded. The aim is to give you a sense of the types of questions that may appear on the exam.
- As a reminder, this is not an exhaustive list of types of problems, anything through Turing Machines (class on February 20) is fair game.
- I recommend preparing your notes sheet before attempting these problems, then revising it to see if it needs to be adapted.
- Attempt all problems. On the exam, it is possible to receive partial credit for showing your work.

Reference Page

This page does not have any questions. Instead it is a reference for relevant definitions and theorems for this exam.

The formal definition of a **Deterministic Finite Automata** (DFA) is:

$$D = (Q, \Sigma, \delta, q_0, F),$$

where Q is a finite set of states, Σ is the finite alphabet, δ is the transition function which is of the form $Q \times \Sigma \rightarrow Q$, q_0 is the start state, and F is the set of accept states.

The formal definition of a **Non-deterministic Finite Automata (NFA)** is:

$$N = (Q, \Sigma, \delta, q_0, F),$$

where Q is a set of states, Σ is the alphabet, δ is the transition function which is of the form $Q \times \Sigma_\epsilon \rightarrow P(Q)$, q_0 is the start state, and F is the set of accept states.

R is a **Regular Expression** if R is:

1. a for some a in the alphabet Σ
2. ϵ
3. $\emptyset$
4. $(R_1 \cup R_2)$,where R_1 and R_2 are regular expressions,
5. $(R_1 \cdot R_2)$,where R_1 and R_2 are regular expressions, or
6. $(R_1)^*$,where R_1 is a regular expression.

The formal definition of a **Context-Free Model** is:

$$C = (V, \Sigma, R, S),$$

where V is a finite set of variables, Σ is a finite set of terminals disjoint from V, R is a finite set of rules, each rule being a variable and string of variables and terminals, and $S \in V$ is the start variable.

The **pumping lemma** states:

If A is a regular language, then there exists a number p (the pumping length) where if s is any string in A of length at least p, then s may be divided into three pieces, $s = xyz$, satisfying the following conditions:

1. For each $i \geq 0, xy^iz \in A$,
2. $|y| > 0$, and
3. $|xy| \leq p$.

The **Myhill-Nerode theorem** states:
A language L is regular if and only if $\equiv_L$ has a finite number of equivalence classes. Furthermore, the number of equivalence classes corresponds to the number of states in the minimal DFA that recognizes L.

Let $L \subset \Sigma^*$ be a language, and $x, y \in \Sigma^*$. x and y are *distinguishable*, written $x \equiv_L y$, if there does not exist some distinguishing extension of x and y with respect to L.

Problem 1	Score: /

Give a finite automata (deterministic or non-deterministic) that recognizes the following language over the alphabet $\Sigma = \{0, 1\}$:

$$L = \{ w \mid w \text{ is an even length string with an odd number of 0s} \}.$$

Problem 2	Score: /

Let $\mathsf{flip}(L)$ be the language of strings drawn from alphabet $\Sigma^* = \{0, 1\}$ such that if $w \in L$ then $w' \in \mathsf{flip}(L)$, where w' is w but every 0 is a 1 and every 1 is a 0.

For example if $0100 \in L$ then $1011 \in \mathsf{flip}(L)$.

Show that if L is regular, then $\mathsf{flip}(L)$ is also regular. You do not have to give a fully-symbolic proof, but your argument should cover all relevant details of the construction.

Problem 3	Score: /

Consider the following language over the alphabet $\Sigma = \{\, (,)\, \}$, *i.e.*, open and close parentheses.

$$L = \{\, w \mid w \text{ is a string of balanced parentheses} \,\}$$

Prove that L is irregular using any method you prefer.

Problem 4	Score: /

Consider the following context-free grammar G with start symbol S:

$$S \to XYX$$
$$X \to X0 \mid 0$$
$$Y \to 1Y \mid \epsilon$$

(a) Give a derivation demonstrating that $S \overset{*}{\Rightarrow} 00100$.

(b) Give a formal definition for $L(G)$.

Problem 5	Score: /

Give a **textual** implementation-level description of a TM that recognizes the following language over $\Sigma = \{ 0, 1 \}$.

$$L = \{ w \mid w \text{ does not contain twice as many 1s as 0s} \}.$$

Make sure in your description to describe any pre-processing you do to the tape, *e.g.*, shifting and marking, as well as the precise movement of the head throughout the operation of the machine.

Problem 6	Score: /

Let a *Right-skip Turing Machine* (RTM) be a (deterministic) Turing Machine that operates like a normal Turing machine except that when it moves its head to the right, it moves *two* tape cells instead of just one. A RTM cannot move its tape head a single cell to the right in a single step of computation.

Prove that an RTM is equivalent to a TM. You do not have to give a fully symbolic account of your constructions, but your prose should cover all relevant details.

(*Hint*: make sure to prove both sides of the equivalence! They are not symmetric constructions!)