Practice CS103 Final Exam

This practice exam is closed-book and closed-computer but open-note. You may have a double-sided, 8.5” × 11” sheet of notes with you when you take this exam. Please hand-write all of your solutions on this physical copy of the exam.

On the actual exam, there'd be space here for you to write your name and sign a statement saying you abide by the Honor Code. We're not collecting or grading this practice exam (though you're welcome to step outside and chat with us about it when you're done!) and this exam doesn't provide any extra credit, so we've opted to skip that boilerplate.

You have three hours to complete this practice exam. There are 48 total points. This practice exam is purely optional and will not directly impact your grade in CS103, but we hope that you find it to be a useful way to prepare for the final exam. You may find it useful to read through all the questions to get a sense of what this exam contains before you begin.

Question	**Points**	**Graders**
(1) Induction	/ 4	
(2) Graphs and Cardinality	/ 5	
(3) Sets and Functions	/ 6	
(4) Regular Languages	/ 10	
(5) Context-Free Languages	/ 4	
(6) **R** and **RE** Languages	/ 15	
(7) **P** and **NP** Languages	/ 4	
	/ 48	

Good Luck!

Problem One: Induction (4 Points)

Prove by induction that if $n \in \mathbb{N}$ and $n \geq 2$, then

$$\log_2 3 \cdot \log_3 4 \cdot \log_4 5 \cdot \ldots \cdot \log_n (n+1) = \log_2 (n+1).$$

You might want to use the change of base formula for logarithms: for any $c \in \mathbb{R}$ with $c > 1$,

$$\log_a b = \frac{\log_c b}{\log_c a}$$

Problem Two: Graphs and Cardinality (5 Points)

Below is a series of statements. For each statement, decide whether it's true or false. No justification is necessary.

i. **(1 Point)** There is a tournament T that does not contain a cycle of length three.

ii. **(1 Point)** If G is a planar graph, then every node in G has degree five or less.

iii. **(1 Point)** If G is an undirected graph that contains a simple cycle of length $k+1$, then G is not k-colorable.

iv. **(1 Point)** If A and B are sets and $A \neq B$, then $|A| < |B|$ or $|B| < |A|$.

v. **(1 Point)** $|\mathbb{N}^{137}| = |\mathbb{Z}^{103}|$.

Problem Three: Sets and Functions (6 Points)

Let $f : A \to B$ be an arbitrary function with domain A and codomain B. Normally, we've talked about what happens when you apply f to a specific element $a \in A$ (this is the value $f(a)$). We can generalize this to talk about what happens when you apply f to multiple elements of A, then gather the resulting elements into a set.

Let $f : A \to B$ be any function and let S be any subset of A. The ***image of S under f***, denoted $f[S]$, is the set of values produced by applying f to each element of S. Formally:

$$f[S] = \{\ b \in B \mid \text{there is some } a \in S \text{ such that } f(a) = b\ \}$$

Here are some examples:

- If $f : \mathbb{N} \to \mathbb{N}$ is the function $f(n) = n + 2$, then $f[\{1, 2, 3\}] = \{3, 4, 5\}$ because $f(1) = 3$, $f(2) = 4$, and $f(3) = 5$.
- If $g : \mathbb{Z} \to \mathbb{N}$ is the function $g(x) = x^2$, then $g[\{-1, 0, 1, 2\}] = \{0, 1, 4\}$ because $g(-1) = 1$, $g(0) = 0$, $g(1) = 1$, and $g(2) = 4$.
- If $h : \mathbb{N} \to \mathbb{N}$ is the function $h(n) = 103$, then $h[\emptyset] = \emptyset$ because there are no elements in $\emptyset$ to which we can apply f.

In this question, we'll ask you to explore two related sets, the set $f[S_1 \cap S_2]$ (the image of $S_1 \cap S_2$), and the set $f[S_1] \cap f[S_2]$ (the intersection of the images of S_1 and S_2).

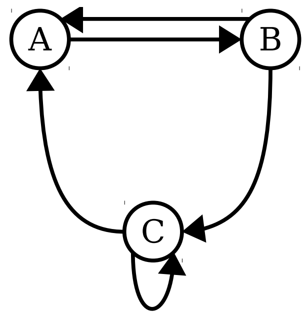
Let $f : A \to B$ be an injection. Prove that if $S_1 \subseteq A$ and $S_2 \subseteq A$, then $f[S_1 \cap S_2] = f[S_1] \cap f[S_2]$. As a hint, remember that to prove two sets are equal to one another, you can show that each set is a subset of the other.

(Extra space for your answer to Problem Three, if you need it.)

Problem Four: Regular Languages (10 Points)

Recall that a *path* in a graph is a series of nodes $v_1, v_2, \ldots, v_n$ such that each pair of adjacent nodes in the path is connected by an edge.

Consider the following graph G:



Let $\Sigma = \{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$. We can represent a path in G as a nonempty string where the letters spell out the path in the graph. For example, the path **A**, **B**, **C**, **C** would be represented by the string **ABCC**.

Let $L = \{\ w \in \Sigma^* \mid w \text{ represents a path in } G\ \}$, where G is the graph given above. For example:

$\mathbf{A} \in L$	$\mathbf{ACC} \notin L$
$\mathbf{ABC} \in L$	$\varepsilon \notin L$
$\mathbf{BCC} \in L$	$\mathbf{BBA} \notin L$
$\mathbf{CCABA} \in L$	$\mathbf{ABBC} \notin L$

i. **(3 Points)** Design a DFA for L.

Let $\Sigma = \{\mathbf{1}, \mathbf{2}, \mathbf{\leq}\}$ and let L be the language defined as follows:

$$L = \{\ w \in \Sigma^* \mid w \text{ is a valid chain of inequalities relating the numbers } \mathbf{1} \text{ and } \mathbf{2}\ \}.$$

For example:

$\mathbf{1{\leq}2} \in L$	$\mathbf{2{\leq}{\leq}} \notin L$
$\mathbf{1{\leq}1{\leq}2{\leq}2} \in L$	$\mathbf{{\leq}2} \notin L$
$\mathbf{2{\leq}2{\leq}2} \in L$	$\varepsilon \notin L$
$\mathbf{1{\leq}1{\leq}1{\leq}1} \in L$	$\mathbf{1} \notin L$
$\mathbf{1{\leq}1{\leq}2} \in L$	$\mathbf{12{\leq}22} \notin L$

Note in particular that inequalities involving numbers like 12, 222, 121212, etc. whose digits are 1 and 2 aren't allowed (the inequality should only relate the numbers 1 and 2) and any individual number itself isn't allowed.

ii. **(2 Points)** Write a regular expression for L.

Let $\Sigma = \{a, b\}$ and consider the following language over Σ:

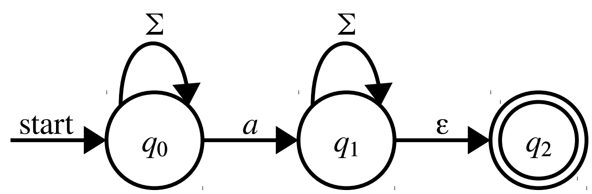
$$L = \{\ w \in \Sigma^* \mid w \text{ has odd length and its middle character is } a\ \}$$

iii. **(4 Points)** Prove that L is not a regular language.

As a reminder, the language L over $\Sigma = \{a, b\}$ from the previous page was defined as follows:

$$L = \{\ w \in \Sigma^* \mid w \text{ has odd length and its middle character is } a\ \}$$

You just proved that this language is not regular. However, below is an NFA that purportedly has language L:



Here is a line of reasoning that claims that this NFA has language L:

> "Intuitively, this NFA will sit in state q_0 following its Σ transition until it nondeterministically guesses that it's about to read the middle a character. When it does, it transitions to q_1, where it keeps following the Σ transition as long as more characters are available. Finally, once it's read all the characters of the input, the NFA follows the ε transition from q_1 to q_2, where the NFA then accepts."

Of course, this reasoning has to be incorrect, since L is not a regular language.

iv. **(1 Point)** Without using the fact that L is not a regular language, explain why the above NFA is not an NFA for the language L.

Problem Five: Context-Free Languages (4 Points)

On Problem Set 6, you explored the language *ADD* over the alphabet $\{ \mathtt{1}, \mathtt{+}, \stackrel{?}{=} \}$, which was defined as follows:

$$ADD = \{ \mathtt{1}^m\mathtt{+}\mathtt{1}^n\stackrel{?}{=}\mathtt{1}^{m+n} \mid m, n \in \mathbb{N} \}$$

Consider the following generalization of *ADD*, which we will call *MULTIADD*, which consists of all strings describing unary encodings of two sums that equal one another. For example:

1 + 3 = 4	would be encoded as	$\mathtt{1+111}\stackrel{?}{=}\mathtt{1111}$
4 = 1 + 3	would be encoded as	$\mathtt{1111}\stackrel{?}{=}\mathtt{1+111}$
2 + 2 = 1 + 3	would be encoded as	$\mathtt{11+11}\stackrel{?}{=}\mathtt{1+111}$
2+0+2+0=0+4+0	would be encoded as	$\mathtt{11++11+}\stackrel{?}{=}\mathtt{+1111+}$
0=0	would be encoded as	$\stackrel{?}{=}$

Notice that there can be any number of summands on each side of the $\stackrel{?}{=}$, but there should be exactly one $\stackrel{?}{=}$ in the string; thus $\mathtt{1}\stackrel{?}{=}\mathtt{1}\stackrel{?}{=}\mathtt{1} \notin$ *MULTIADD*.

Write a CFG that generates *MULTIADD*.

Problem Six: R and RE Languages (15 Points)

Consider the following language L:

$$L = \{ \langle M \rangle \mid M \text{ is a TM with input alphabet } \{a, b\} \text{ and } M \text{ accepts exactly one string of each length } \}$$

i. **(5 Points)** Prove that $L \notin \mathbf{R}$.

Recall that a ***verifier*** for a language L is a TM V such that

- V halts on all inputs, and
- $\forall w \in \Sigma^*. \ (w \in L \quad \leftrightarrow \quad \exists c \in \Sigma^*. \ V \text{ accepts } \langle w, c \rangle)$

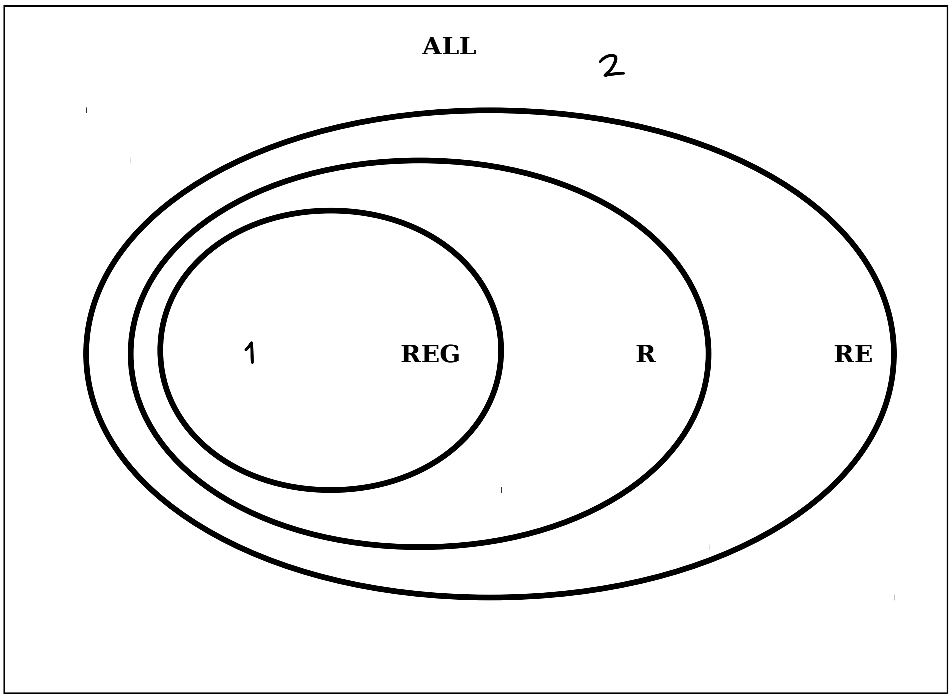
Let's say that a ***weak verifier*** for a language L is a TM X such that

- $\forall w \in \Sigma^*. \ (w \in L \quad \leftrightarrow \quad \exists c \in \Sigma^*. \ X \text{ accepts } \langle w, c \rangle)$

In other words, a weak verifier for a language L is like a normal verifier for L, except that it's not required to halt on all inputs.

ii. **(4 Points)** Prove that if X is a weak verifier for a language L, then $L \in$ **RE**.

ii. **(6 Points)** Below is a Venn diagram showing the overlap of different classes of languages we've studied so far. We have also provided you a list of eight numbered languages. For each of those languages, draw where in the Venn diagram that language belongs. As an example, we've indicated where Language 1 and Language 2 should go. No proofs or justifications are necessary, and each language is worth exactly one point.



1. Σ^*
2. EQ_{TM}
3. $\{ a^n b^n c^n \mid n \in \mathbb{N} \}$
4. The concatenation of language (1) and language (3). Assume $\Sigma = \{ a, b, c \}$
5. The union of language (1) and language (3). Assume $\Sigma = \{ a, b, c \}$
6. $\{ \langle w_1, w_2 \rangle \mid$ either w_1 is the encoding of a TM M that accepts w_2 or w_2 is the encoding of a TM M that accepts w_1 $\}$
7. The complement of language (3).
8. The complement of language (6).

Problem Seven: P and NP Languages (4 Points)

Prove that if L is a language, then L is **NP**-complete if and only if $L \leq_p$ 3SAT and 3SAT $\leq_p L$.