Practice CS103 Final Exam IV

We strongly recommend that you work through this exam under realistic conditions rather than just flipping through the problems and seeing what they look like. Setting aside three hours in a quiet space with your notes and making a good honest effort to solve all the problems is one of the single best things you can do to prepare for this exam. It will give you practice working under time pressure and give you an honest sense of where you stand and what you need to get some more practice with.

This practice final exam is essentially the final exam from Fall 2015, with a few minor modifications (some of the problems we asked here got converted to problem set questions, so we replaced them with other exam questions) and others covered topics that have since be dropped from CS103 (namely, using self-reference to prove unrecognizability). With the exception of Q5.ii, every question here has appeared on some CS103 exam in the past.

The exam policies are the same for the midterms – closed-book, closed-computer, limited note (one double-sided sheet of 8.5" × 11" paper decorated however you'd like).

You have three hours to complete this exam. There are 48 total points.

Question	Points	Graders
(1) Logic and Relations	/ 6	
(2) Graphs and Sets	/ 6	
(3) Induction and Cardinality	/ 6	
(4) Regular and Context-Free Languages	/ 12	
(5) **R** and **RE** Languages	/ 14	
(6) **P** and **NP** Languages	/ 4	

Problem One: Logic and Relations   (6 Points)

Suppose that you want to prove the implication $P \rightarrow Q$. Here are two possible routes you can take:

- Prove the implication by contradiction.

- Take the contrapositive of the implication, then prove the contrapositive by contradiction.

It turns out that these two proof approaches are completely equivalent to one another.

   i. **(2 Points)** State, in propositional logic, which statements you will end up assuming if you were to use each of the above proof approaches, then *briefly* explain why they're equivalent.

ii. **(4 Points)** Below is a drawing of a binary relation R over a set of people A:
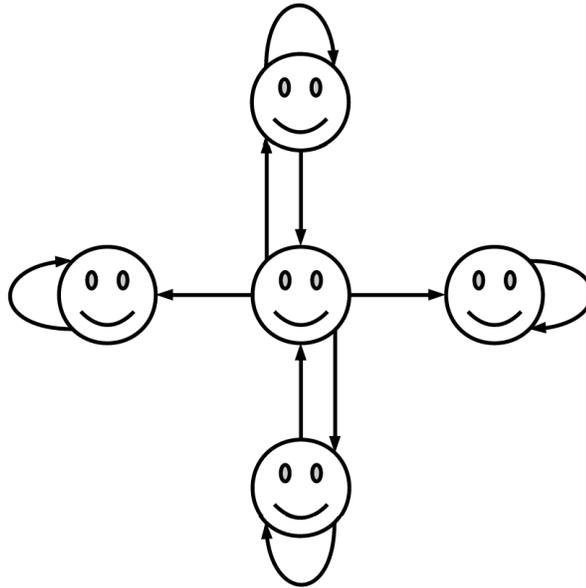
For each of the following first-order logic statements about R, decide whether that statement is true or false. No justification is required, and there is no penalty for an incorrect guess.

1. $\forall p \in A.\ \exists q \in A.\ pRq$

   ☐ True   ☐ False

2. $\exists p \in A.\ \forall q \in A.\ pRq$

   ☐ True   ☐ False

3. $\exists p \in A.\ (pRp \rightarrow \forall q \in A.\ qRq)$

   ☐ True   ☐ False

4. $\neg\forall p \in A.\ \forall q \in A.\ (p \neq q \rightarrow \exists r \in A.\ (pRr \wedge qRr))$

   ☐ True   ☐ False

Problem Two: Graphs and Sets   (6 Points)

Recently, there's been a major development in complexity theory: an "almost" efficient algorithm for the graph isomorphism problem. The algorithm relies on a special class of graphs that are the focus of this problem.

The **_triangular graph of order n_**, denoted T_n, is a graph defined as follows. Begin with the set $\{1, 2, 3, \ldots, n\}$. The nodes in T_n are the two-element subsets of $\{1, 2, 3, \ldots, n\}$, and there's an edge between any two sets that have exactly one element in common. For example, below are the graphs T_3 and T_4:

The graph T_3 The graph T_4
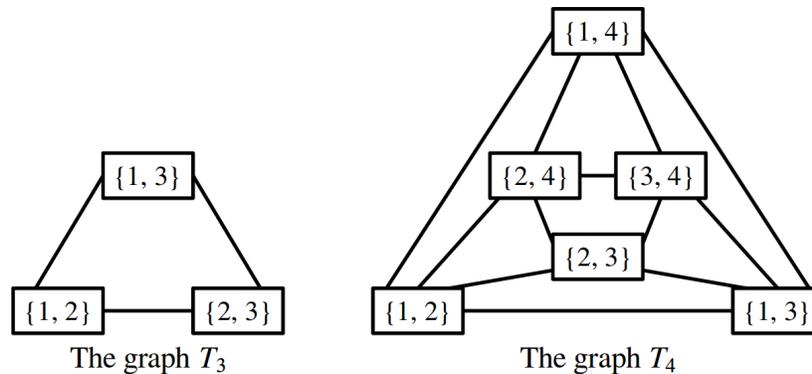
Recall from Problem Set Four that an **_independent set_** in an undirected graph $G = (V, E)$ is a set $I \subseteq V$ such that if $x \in I$ and $y \in I$, then $\{x, y\} \notin E$. Intuitively, an independent set in G is a set of nodes where no two nodes in I are adjacent. The **_independence number_** of a graph G, denoted $\alpha(G)$, is the size of the largest independent set in G.

Prove that if $n \in \mathbb{N}$ and $n \geq 1$, then $\alpha(T_{2n}) = n$. (Hint: You need to prove two separate results: first, that there's an independent set of size n in T_{2n}; second, that no larger independent set exists in T_{2n}.)

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

Problem Three: Induction and Cardinality   (6 Points)

Consider the following series:

$$-1 + 2 - 3 + 4 - 5 + 6 - 7 + 8 - 9 + 10 - 11 + 12 - 13 + 14 - 15 \ldots$$

We can think about evaluating larger and larger number of terms in the summation. For example, the sum of the first five terms is $-1 + 2 - 3 + 4 - 5 = -3$, and the sum of the first eight terms works out to $-1 + 2 - 3 + 4 - 5 + 6 - 7 + 8 = 4$. For notational simplicity, let's define A_n to be the sum of the first n terms in the summation. For example, A_0 is the sum of the first zero terms in the summation (that's the empty sum, which is zero). A_1 is the sum of the first term (-1), A_2 is the sum of the first two terms (-1 + 2 = 1), A_3 is the sum of the first three terms (-1 + 2 − 3 = -2), etc.

When we covered cardinality in lecture, we gave the following piecewise function as an example of a bijection $f : \mathbb{N} \to \mathbb{Z}$:

$$f(n) = \begin{cases} \dfrac{n}{2} & \text{if } n \text{ is even} \\ -\dfrac{n+1}{2} & \text{otherwise} \end{cases}$$

It turns out that this function is closely connected to the above series. Specifically, for every natural number n, the following is true:

$$A_n = f(n)$$

In other words, you can form a bijection from $\mathbb{N}$ to $\mathbb{Z}$ by considering longer and longer alternating sums of the natural numbers. Weird, isn't it?

Prove by induction on n that if $n \in \mathbb{N}$, then $A_n = f(n)$.

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

Problem Four: Regular and Context-Free Languages (12 Points)

Let $\Sigma = \{a, b\}$ and consider the following languages L_1 and L_2 over Σ:

$L_1 = \{ w \in \Sigma^* \mid w$ doesn't contain bb as a substring $\}$

$L_2 = \{ w \in \Sigma^* \mid |w| \geq 3$ and the third-to-last character of w is an a $\}$

This problem concerns the language $L_1 \cap L_2$. As an example, the strings aaa, baaba, and bababa are all in $L_1 \cap L_2$, and the strings ε, ba, abb, bbaab, and bab are all not in $L_1 \cap L_2$.

   i. **(3 Points)** Design an NFA for $L_1 \cap L_2$. No justification is necessary.

   ii. **(3 Points)** Write a regular expression for $L_1 \cap L_2$. No justification is necessary.

The "canonical" example of a nonregular language is the language $L_3 = \{\ \mathsf{a}^n\mathsf{b}^n \mid n \in \mathbb{N}\ \}$. It turns out that, not only is this language not regular, but most of its subsets aren't regular either.

    iii. **(3 Points)** Prove that if $L \subseteq L_3$ and L contains infinitely many strings, then L is not regular.

In Problem Set Eight, you designed a CFG for the following language:

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

Now, consider the following language over the alphabet $\{1, +, \approx\}$, which is a variation on *ADD*:

$$NEAR = \{\ 1^m + 1^n \approx 1^p \mid m, n, p \in \mathbb{N} \text{ and } m + n = p + 1\ \}$$

Intuitively, *NEAR* is the set of all arithmetic expressions where the left-hand side is exactly one greater than the right-hand side. For example:

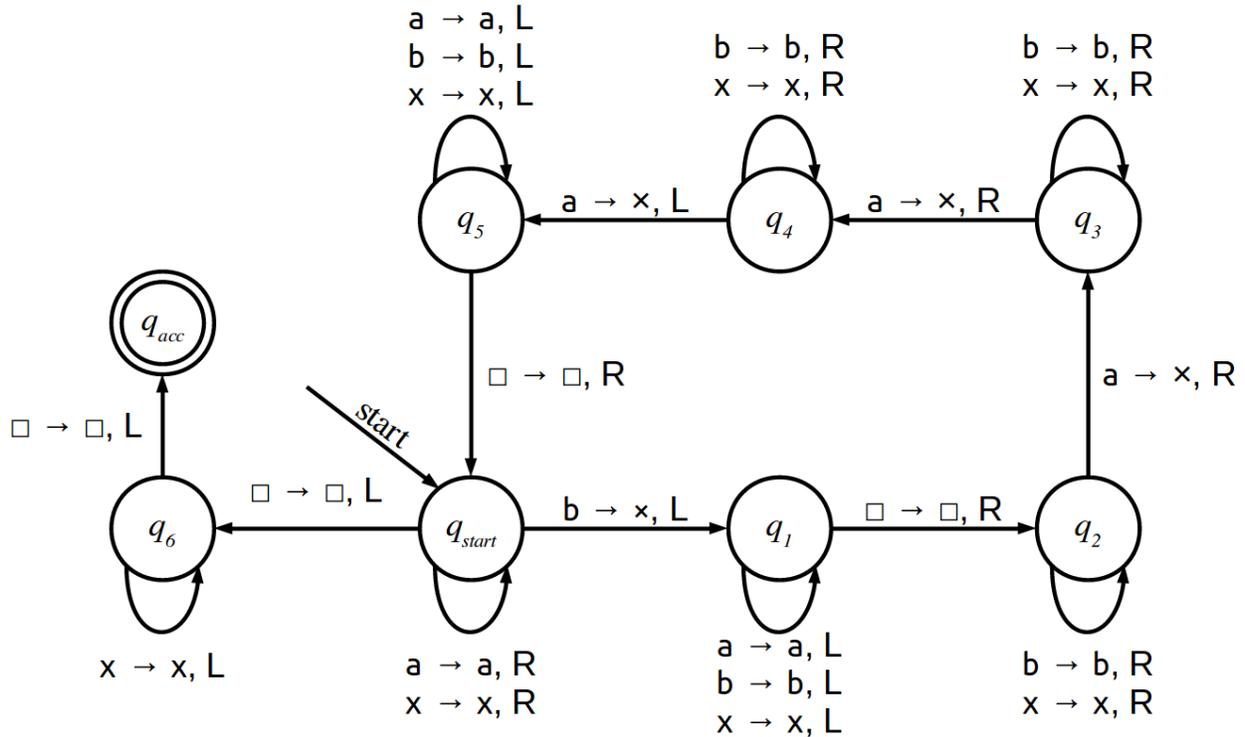|	
111+1≈111 ∈ *NEAR*	+≈ ∉ *NEAR*
11+111≈1111 ∈ *NEAR*	1+1≈11 ∉ *NEAR*
1+≈ ∈ *NEAR*	1+1≈111 ∉ *NEAR*
+1≈ ∈ *NEAR*	1+1+1≈11 ∉ *NEAR*

This language turns out to be context-free.

  iv. **(3 Points)** Write a CFG for *NEAR*.

Problem Five: R and RE Languages   (14 Points)

Consider the following TM, which we'll call TM_6:

Here, q_{start} is the start state, and q_{acc} is the accepting state. As usual, we assume that all missing transitions implicitly cause M to reject.

TM_6's input alphabet is $\Sigma = \{a, b\}$ and its tape alphabet is $\Gamma = \{a, b, \times, \square\}$.

i. **(3 Points)** Fill in the following blank to let us know what the language of TM_6 is. You may find it useful to run this TM on a few small sample inputs to get a feel for how it works. No justification is necessary.

$\mathscr{L}(TM_6) = \{\ w \in \Sigma^* \mid$ _____ $\}$

Let Σ be an arbitrary alphabet and consider the following language:

$$A_{ALL} = \{\ \langle M \rangle \mid M \text{ is a TM and } \mathscr{L}(M) = \Sigma^*\ \}$$

In other words, A_{ALL} is the language of all descriptions of TMs that accept every string.

  ii. **(5 Points)** Prove that $A_{ALL} \notin \mathbf{R}$.

(Extra space for your answer to Problem 5.ii, if you need it.)

iii. **(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 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 there is no penalty for an incorrect guess.

1. Σ^*
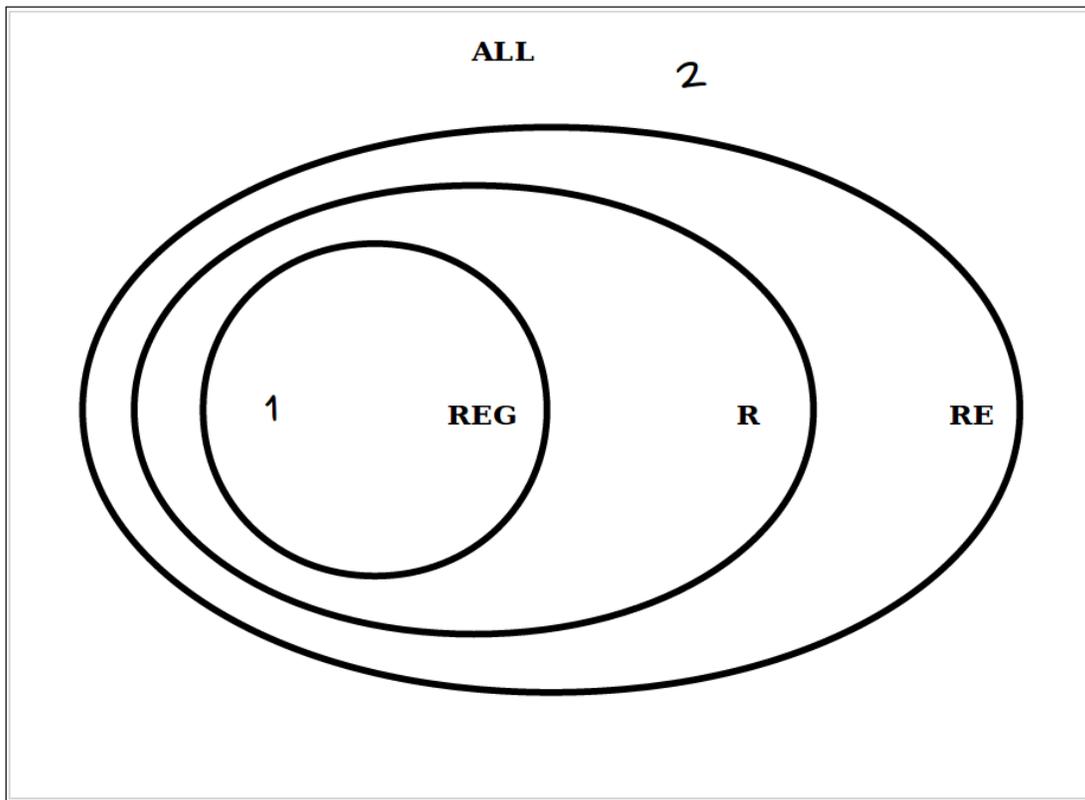
2. L_D

3. $\{\ w \in \{a, b\}^* \mid |w| \geq 100$ and the first 50 characters of w are the same as the last 50 characters of $w\ \}$

4. $\{\ \langle M_1, M_2, M_3 \rangle \mid M_1, M_2,$ and M_3 are TMs over the same alphabet Σ and every string in Σ^* belongs to exactly one of $\mathscr{L}(M_1), \mathscr{L}(M_2),$ or $\mathscr{L}(M_3)\ \}$

5. $HALT - A_{TM}$

6. $A_{TM} - HALT$

7. $\{\ \langle V, w \rangle \mid V$ is a TM and there is a string c such that V accepts $\langle w, c \rangle\ \}$

8. $\{\ w \in \{r, d\}^* \mid w$ has more r's than d's $\}$

Problem Six: P and NP Languages   (4 Points)

Below is a series of four statements. For each statement, decide whether it's true or false. No jus-
tification is necessary. There is no penalty for an incorrect guess.

   i. If **P** = **NP**, there are no **NP**-*complete* problems in **P**.

      ☐ True   ☐ False

   ii. If **P** = **NP**, there are no **NP**-*hard* problems in **P**.

      ☐ True   ☐ False

   iii. If **P** ≠ **NP**, there are no **NP**-*complete* problems in **P**.

      ☐ True   ☐ False

   iv. If **P** ≠ **NP**, there are no **NP**-*hard* problems in **P**.

      ☐ True   ☐ False

We have one final question for you: do *you* think **P** = **NP**? Let us know in the space below. There
are no right or wrong answers to this question – we're honestly curious to hear your opinion!

      ☐ I think **P** = **NP**   ☐ I think **P** ≠ **NP**