Name: ______________________________

Student ID: ______________________________

CS181 Winter 2018 - Final

Due Friday, March 16, 11:59 PM

- This exam is open-book and open-notes, but any materials not used in this course are prohibited, including any material found on the internet. **Collaboration is prohibited.** Please avoid temptation by not working on the final while you are in the presence of any other student who has taken or is currently taking CS181. **Be extra careful if you live with or meet regularly with a student of this class.** If you have any questions about the exam, ask the TA or Professor Sahai by email or after class. **Do not ask other students.** You are allowed to use any theorem shown in class or in the textbook, as long as you clearly cite it. Please monitor Piazza for any clarifications. **Do not** post any questions on piazza.
- We suggest that you spend approximately 12 hours (not necessarily contiguous) to take this exam. Start early so that you have time to understand and think about the problems. **The solutions must be submitted on CCLE by 11:59 PM on Friday, March 16.**
- Place your name and UID on every page of your solutions. Retain this cover sheet and the next sheet with the table as the first pages of your solutions. **Please use separate pages for each question. All problems require clear and well-written explanations.**
- There are 4 questions worth a total of 255 points and an extra credit question worth 40 points.
- For each part (except for the extra credit), if you describe a non-trivial approach that you tried using to solve the problem but realized it doesn't work and explain correctly why it doesn't work and then write "I don't know" you will get 20% points for that problem. You will **not** get 20% points for just writing "I don't know". Whether your stated approach was indeed non-trivial is solely at the discretion of the grader.
- 5% extra credit will be awarded to solutions writen in LaTeX.

Please **handwrite** the following honor code agreement and sign and date in the spaces provided.
Honor Code Agreement: I promise and pledge my honor that during the exam period, I did not and will not talk to any person about CS 181 material except for the professor or the TA, nor will I refer to any material except for the class textbook and my own class notes. I will abide by the CS181 Honor Code.

Signature: ______________________________

Date: ______________________________

Question	**Points**	
1		
2		
3		
4		
EC		
Total		

1. **nLRA Problem (40 points)**
Recall the definition of a *Lovesick Robot Automaton (LRA)*. A *Lovesick Robot Automaton* is a type of machine which takes a **finite** string as input, and generates an **infinite** pattern. We think of the LRA as starting at position 1 on an infinite tape and moving right in a sequence of steps, and at each step it decides to either mark the current cell on the tape with a $\heartsuit$ or to not mark anything, i.e. it leaves a blank character, which we denote with "␣". The LRA has the following characteristics:

   - Just as in the case of a DFA, a LRA is a five tuple $(Q, \Sigma, \delta, q_0, F)$. However, after a LRA has processed the last character of the input string, it goes back to the first character. So a LRA processes the string repeatedly in an infinite loop. Each step of the robot corresponds to a state transition.
   - The set F is now called the set of *marking states.* At each step, if the new state is a marking state then the robot marks the new cell with a $\heartsuit$. Otherwise, the robot does not make a mark in this position, i.e. it leaves it blank (␣)
   - It is only after processing the first symbol of the input that the LRA either creates a mark $\heartsuit$ or leaves a blank ␣, depending on what state the LRA enters after processing this first symbol.

   Define a *New Lovesick Robot Automaton (nLRA)* to be an LRA which can decide at each step which direction to move on the infinite tape. In other words, the transition function is now defined as $\delta : Q \times \Sigma \to Q \times \{L, R\}$. If $\delta(q, a) = (q', R)$ then the robot moves **right** on the tape and leaves a $\heartsuit$ if q' is a marking state. Likewise, if $\delta(q, a) = (q', L)$ then the robot moves **left** on the tape and leaves a $\heartsuit$ if q' is a marking state. (If the robot is at position 1 and tries to move left then it just stays at position 1.)

   We will explore the question of using a Turing Machine to decide questions about nLRAs. To that end, denote by $\langle\langle R\rangle\rangle$ an encoding of an nLRA R.

   (a) **(20 points)** Rigorously prove the following new "pumping lemma" for nLRAs:

   Lemma 1. *For any nLRA M and any string s, there exist $w, z \in \{L, R\}^*$ such that the* ***pattern of directions taken by the robot*** *when executing $M(s)$ is of the form $wzzzzz\dots$.*

   (b) **(20 points)** Show the language

   $$L = \{(\langle\langle R\rangle\rangle, s) \mid \text{For all } n \in \mathbb{N},\ R(s) \text{ eventually visits the } n\text{-th cell on the infinite tape}\}$$

   is decidable.

2. **Libra Machines (85 points)**
In this problem, we define a Turing machine variant called a Libra Machine (LM). A Libra Machine has a total of three tapes: in addition to its regular tape, it has two *machine tapes.* The LM can write to the three tapes in the ordinary way, but it has an extra trick up its sleeve. At any given moment, the LM can go into a special *query state* and perform a *language equality query.* Suppose an ordinary Turing Machine representation $\langle M_1 \rangle$ is stored on the first machine tape of an LM T, and another ordinary Turing Machine representation $\langle M_2 \rangle$ is stored on the second machine tape of T. If T enters the query state, then immediately T learns whether $L(M_1) = L(M_2)$. In particular, if $L(M_1) = L(M_2)$, then T enters a special state called the *yes state* without consuming any input or moving the tape. If $L(M_1) \neq L(M_2)$ then T enters another special state called the *no state.* We say that a language $L \subseteq \Sigma^*$ is Libra recognizable (respectively, Libra decidable) if there exists an LM T recognizing (respectively, deciding) L.
Note: Throughout this problem, $\langle M \rangle$ **denotes a description of an *ordinary* Turing machine.**
Example: A Libra machine can easily decide the language $\{(\langle M_1 \rangle, \langle M_2 \rangle) \mid L(M_1) = L(M_2)\}$. The machine simple writes $\langle M_1 \rangle$ to the first machine tape and $\langle M_2 \rangle$ to the second tape and enters the query state, and accepts if it enters the yes state.

(a) **(15 points).** Show that the language

$$\mathsf{HALT} = \{(\langle M \rangle, x) \mid M \text{ halts on input } x\}$$

is Libra decidable.

(b) **(20 points).** Show that the language

$$\mathsf{MINIMAL} = \{\langle M \rangle \mid M \text{ is a minimal turing machine}\}$$

is Libra decidable. (This language is defined in Section 6.1 in Sipser.)

(c) **(20 points).** Show that there exists a language L which is not Libra recognizable.
Hint: Think about infinities.

(d) **(30 points).** Show that there exists a language L which is Libra recognizable but not Libra decidable.

3. **Decidability and Recognizability (90 points)**
Consider the following language:

$$\mathsf{DISAGREE} = \{(\langle M_1 \rangle, \langle M_2 \rangle) \mid \text{there exists an } x \text{ such that } x \in L(M_1) \text{ but } x \notin L(M_2)\}$$

(a) **(15 points).** Show $\mathsf{DISAGREE}$ is undecidable.

(b) **(15 points).** Show $\mathsf{DISAGREE}$ is unrecognizable.

Now we define a machine called a *fibonacci enumerator*. A two-tape Turing Machine is a *fibonacci enumerator* if it runs forever and for any fibonacci number n, the binary representation of n is eventually written to the work tape.

Consider the following language:

$$\mathsf{FIB} = \{\langle M \rangle \mid M \text{ is a fibonacci enumerator}\}$$

(c) **(25 points).** Is FIB recognizable? Prove or disprove.

Finally, consider the language

$$\mathsf{SMALLANDPICKY} = \{(\langle M \rangle, x) \mid M \text{ is a minimal Turing machine where } L(M) = \{x\}\}$$

(d) **(10 points).** Show that the set

$$\{\langle M \rangle \mid \text{there is some } x \text{ such that } (\langle M \rangle, x) \in \mathsf{SMALLANDPICKY}\}$$

is infinite.

(e) **(25 points).** Show that $\mathsf{SMALLANDPICKY}$ is unrecognizable.

4. **Language Properties (40 points)** In this problem we study variants of Rice's theorem.

(a) **(20 points).** Define a *strong nontrivial property of languages* to be a nontrivial property of languages P where $P(\varnothing) = 1$. Show that for any such P the language

$$L_P = \{\langle M \rangle \mid P(L(M)) = 1\}$$

is unrecognizable.

(b) **(20 points).** Define a *2-dimensional nontrivial property of languages* to be a function $P : \mathcal{P}(\Sigma^*) \times \mathcal{P}(\Sigma^*) \to \{0, 1\}$, such that the following property holds:

(i) There exist Turing Machines $M, M_{\mathsf{yes}}, M_{\mathsf{no}}$ where $P(L(M), L(M_{\mathsf{yes}})) = 1$ but $P(L(M), L(M_{\mathsf{no}})) = 0$.

Show that for any such P, the language

$$L_P = \{(\langle M_1 \rangle, \langle M_2 \rangle) \mid P(L(M_1), L(M_2)) = 1\}$$

is undecidable.

(c) **(40 points extra credit).** Now consider a variant of the previous problem where (i) is replaced by the following:

(i) There exist Turing Machines M_1, M_2, M_1', M_2' where $P(L(M_1), L(M_2)) = 1$ but $P(L(M_1'), L(M_2')) = 0$.

Show that it is still the case that for any such P, the language L_P is undecidable.