**CS 81 (Spring 2018)** **Assignment 11** **Name**
**Computability and Logic** **Due Monday April 23 at 6:00 PM**

**Notice:** When you submit this assignment, you are certifying therewith that you understand and accept the following policy, which applies to all assignments.

**Collaboration Policy: The writeup that you submit must be your own work.** You are encouraged to get help from the professor and grutors. You may discuss the problems with classmates, but if you do so, it should be in groups of no more than three. You are not allowed to copy or transcribe solutions from other sources, including the work of other students, the internet, previous solution sets, and images photographed from a whiteboard or blackboard. There is to be no "partnering" where two or more students submit the same writeup. If you get help on a problem, you should say who provided the help on a per-problem basis. Blanket statements such as "worked with John and Mary" are not allowed. Detected infractions may impact your academic career.

**Formatting Policy** All work must be typeset in electronic media and submitted as a **single pdf file**, one problem on each page as shown in the following pages. Retain this header page. **Handwritten and photographed or scanned work is not allowed.** Do not use inverse video (light typography on dark background). Do not rotate images. **You will not get credit for difficult-to-read submissions.**

Once your document is complete, make a pdf and submit to **Gradescope**.

1. [20 points] As shown in lecture, it is possible to construct a DFA with input alphabet $\{0, 1\}$ that decides the language of strings in $\{0, 1\}^\star$ that, when interpreted as binary numerals most-significant bit first, represent numbers divisible by 3. [As a convention, we will take the empty string $\varepsilon$ to represent number $0$ as well, even though technically it is not a binary numeral.]

   a. Describe how for any $n > 0$, the set $L_n$ of strings of `0`'s and `1`'s that are binary numerals of numbers divisible by $n$ is regular (i.e. there is a DFA that decides it.) [Hint: What is the effect on the modulus of appending an additional `0` vs. an additional `1`?]

   b. Illustrate your construction by showing the exact rules for $n = 7$.

   c. Run your $\bmod 7$ machine on the simulator `dfa.pro` and show the output for all input strings of length 7.

   Note: It is **not** recommended that you use induction on $n$ here. Instead, describe the state-transitions for any given $n$ numerically, using arithmetic modulo $n$.

---

2. [30 points] By the reverse $x^{\mathsf{R}}$ of a string, we mean the string consisting of the same symbols as in $x$, but in the reverse order. For example, $01011^{\mathsf{R}} = 11010$. By the reverse of a language $L$, we mean the language consisting of the reverses of strings in $L$:

$$L^{\mathsf{R}} = \left\{ x^{\mathsf{R}} \mid x \in L \right\}.$$

   a. Show by construction that regular languages are closed under reversal, i.e. if $L$ is regular, then so is $L^{\mathsf{R}}$.

   b. If $L$ represents numerals most-significant-bit first, then $L^{\mathsf{R}}$ represents numerals least-significant-bit first. Construct the reverse of the $\bmod 3$ machine presented in the lecture notes. What is interesting about this example?

   c. Construct the reverse of the $\bmod 7$ machine from part c.. Show the output of the simulator `dfa.pro` for input strings of length 7.

---

3. [10 points] The reverse $x^{\mathsf{R}}$ of a string was defined in problem 2. Show that if $L$ is regular then so is

$$L' = \left\{ x \in L \mid x^{\mathsf{R}} \in L \right\}.$$

$L'$ is thus the subset of $L$ such that for any string in the subset, the reverse of the string is also in $L$. For example, if $L = \{0, 01\}^\star$, then $L' = \{\varepsilon\} \cup \{0\}\{0, 10\}^\star$, as

$$\begin{aligned} L &= \{\varepsilon, 0, 00, 01, 000, 001, 010, 0000, 0010, 0100, 0101, 00000, \dots\}, \\ L' &= \{\quad 0, 00, \quad 000, \quad 010, 0000, 0010, 0100, \quad 00000, \dots\}, \end{aligned}$$

where we've aligned the elements of each to show which are in the subset.

---

4. [10 points] Consider the language $T = \{1^t \mid t \text{ is a triangular number}\} = \{\varepsilon, 1, 111, 111111, 1111111111, \dots\}$ $(0, 1, 3, 6, 10, 15, \dots)$. Use the Myhill-Nerode theorem to show that $T$ is not regular.

---

5. [10 points] Let $\Sigma$ and $\Delta$ be alphabets. A *string homomorphism* is a function $h\colon \Sigma^\star \to \Delta^\star$ such that for any $x \in \Sigma^\star$ there is a string $h(x) \in \Delta^\star$ as follows:

   For single letters $\sigma \in \Sigma$, $h(x) \in \Delta^\star$ is specified as a *base*. Then for strings:

   $$h(\sigma_1\sigma_2\sigma_3 \ldots \sigma_n) = h(\sigma_1)h(\sigma_2)h(\sigma_3)\ldots h(\sigma_n), \quad \text{each } \sigma_i \in \Sigma.$$

   (For $n = 0$, $h(\varepsilon) = \varepsilon$.)

   For example, suppose $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and $\Delta = \{0, 1, 2\}$. One string homomorphism is given by $h(\mathtt{a}) = 0112, h(\mathtt{b}) = \varepsilon$. Then, for example,

   $$h(\mathtt{aba}) = 0112\varepsilon 0112 = 01120112.$$

   A string homomorphism is applied to an entire language $L$ as

   $$h(L) = \left\{h(x) \mid x \in L\right\}.$$

   Show that the regular languages are closed under string homomorphism, i.e. if $L$ is regular, and $h$ is a string homomorphism, then $h(L)$ is regular.

   [Suggestion: Maybe use the idea of an NFA.]

---

6. [20 points] Let $\Sigma$ and $\Delta$ be alphabets. A *substitution* is a function $h\colon \Sigma \to 2^{\Delta^\star}$ (where $2^{\Delta^\star}$ is the set of all languages of $\Delta$). That is, for each $\sigma \in \Sigma$, $h(\sigma) \subseteq \Delta^\star$, a language. Then for strings,

$$h(\sigma_1\sigma_2\sigma_3\ldots\sigma_n) = h(\sigma_1)h(\sigma_2)h(\sigma_3)\ldots h(\sigma_n), \quad \text{each } \sigma_i \in \Sigma,$$

where the concatenation here is the concatenation of languages rather than strings. (For $n = 0$, $h(\varepsilon) = \{\varepsilon\}$.)

A substiution is applied to an entire language $L$ as

$$h(L) = \bigcup \{h(x) \mid x \in L\}.$$

That is, to get $h(L)$ for a language, we take the union of $h$ applied to individual strings in the language.

A *regular* substitution is one where each $h(\sigma)$ is a regular language. Show that a regular substitution applied to a regular langauge results in a regular language.

[Hint: Maybe use regular expressions.]

---