Homework 8: CFGs, Relations and DFAs

Due date: Wednesday May 26 at 11:00 PM (Seattle time, i.e. GMT-7)
If you work with others (and you should!), remember to follow the collaboration policy.
In general, you are graded on both the clarity and accuracy of your work. Your solution should be clear enough that someone in the class who had not seen the problem before would understand it.
We sometimes describe approximately how long our explanations are. These are intended to help you understand approximately how much detail we are expecting. Whenever an exercise asks to "*prove*" or "*show*" a statement without further specification, you are free to choose any proof method that you have learned in 311.

Be sure to read the grading guidelines for more information on what we're looking for.

While working on the assignment, please (roughly) keep track of the time you spent on each problem. This is so that you have an idea of how to answer the feedback questions!

1. Collaborators

List any collaborators (i.e., other students) you worked with and which problems you worked on together, or state that you worked alone.

2. Context Is Everything. Except for Context-Free Grammars (Online) [20 points]

For each of the following languages, construct a context-free grammar that generates exactly the given set of strings. The start symbol is S. You should submit (and check!) your answers online at https://grin.cs.washington.edu/

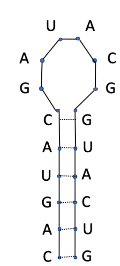
Test your grammar carefully before submitting; you only have 5 submissions. Because these are auto-graded, we will not award partial credit.

(a) $\{1^m 0^n 1^{m+n} : m, n \geq 0\}$

(b) Binary strings matching the regular expression "$(1 \cup 01 \cup 001)^*(0 \cup \varepsilon)$".
Hint: You can use the procedure described in lecture to convert the RE to a CFG.

(c) RNA strings that can can fold into a lollipop shape, as defined below.

Each RNA string is a string over the alphabet $\{A, C, G, U\}$. The string CAGUACGAUACGGUACUG can fold into a lollipop shape like this:



The prefix CAGUAC and suffix GUACUG can together form the handle of the lollipop because they line up in such a manner that As are across from Us (or vice versa) and Cs are across from Gs (or vice versa). The "candy" part of the lollipop consists of the string GAUACG, which has length 6.

Definition. In general, an RNA string can fold into a lollipop shape if it can be written as xyz, where x and z contain at least two characters, y contains at least four characters, and the characters of x match up with those of z^R, in the sense that the the corresponding pairs of letters fall in the set $\{(A, U), (U, A), (C, G), (G, C)\}$. (The reverse z^R of a string is defined here.)

3. University Relations [15 points]

Consider the set of all students at UW. In each part of this problem, we define a relation R on this set. For each one, state whether R is or is not reflexive, symmetric, antisymmetric, and/or transitive. (No proofs.)

(a) $(a, b) \in R$ if students a and b take exactly the same set of classes in the Spring 2021 quarter.

(b) $(a, b) \in R$ if every class that is taken by a is also taken by student b.

(c) $(a, b) \in R$ if students a and b have no classes in common this quarter.

4. Distant Relations [10 points]

Prove or disprove each of the following claims:

(a) If R and S are transitive relations on the set A, then $R \cap S$ is transitive.

(b) If R and S are transitive relations on the set A, then $R \cup S$ is transitive.

5. Paths vs. Relations [13 points]

Let $R \subseteq A \times A$ be a relation on a set A and let $n \in \mathbb{N}$. Prove by induction that there is a path of length n from a to b in the directed graph (A, R) if and only if $(a, b) \in R^n$.
Hint. You need to prove a bi-conditional here. Also recall that each single vertex corresponds to a path of length 0. Note that $\mathbb{N} = \{0, 1, 2, \ldots\}$.

6. Suffixes [7 points]

For a language $L \subseteq \Sigma^*$, define $\sim_L$ to be the following relation on Σ^*:
$x \sim_L y$ if and only if $\forall z(xz \in L \leftrightarrow yz \in L)$

(a) Convince yourself $\sim_L$ is reflexive. You do not have to write anything for this part. [0 points]

(b) Convince yourself $\sim_L$ is symmetric. You do not have to write anything for this part. [0 points]

(c) Write a proof that $\sim_L$ is transitive. [7 points]

Fun fact: in the minimum DFA for any regular language, $x \sim_L y$ if and only if the DFA puts them in the same state! Recognizing this fact is the key to proving that the minimum DFA for a regular language is unique.

7. Build DFAs (Online) [15 Points]

For each of the following languages, construct a DFA that accepts exactly the given set of strings. You should submit (and check!) your answers online at https://grin.cs.washington.edu

Think carefully before entering your DFA; you only have 5 guesses. Because these are auto-graded, we will not award partial credit.

(a) Binary strings with at least two 1s.

(b) Binary strings that have at least one 1 and an even number of 0s.

(c) Binary strings such that none of their runs of 1s have odd length.
A "run" of 1s is a string of consecutive 1s with edges at the start of the string, end of the string, or adjacent to a 0. "11" contains exactly one run of 1s (of length 2).
"111101" contains two runs of 1s (one of length 4, the other of length 1).
For example 1110011 is not accepted because the first three characters are an odd-length run of 1s. But 110011110 is accepted because the first run of 1s has length two, and the second run of 1s has length 4.

8. Extra Credit: Finishing Strong

Definition. A language is regular if there is a regular expression that matches it. Equivalently, a language is regular if there is a DFA that accepts it.

Let L be a regular language over the alphabet Σ. Define $\texttt{ends}(L) = \{v \in \Sigma^* : \exists u \in \Sigma^*(uv \in L)\}$. Prove that $\texttt{ends}(L)$ is regular.

9. Feedback

Please keep track of how much time you spend on this homework and answer the following questions. This can help us calibrate future assignments and future iterations of the course, and can help you identify which areas are most challenging for you.

- How many hours did you spend working on this assignment?
- Which problem did you spend the most time on?
- Which problem did you find to be the most confusing?
- Any other feedback for us?