CSE 311: Foundations of Computing I

Homework 7 (due Wednesday, August 16th at 11:59 PM)

Directions: *Write up carefully argued solutions to the following problems. Each solution should be clear enough that it can explain why it works to someone who does not already understand the answer. If you work with others, remember to follow the collaboration policy outlined in the syllabus. Be sure to read the Typesetting and Grading guidelines prior to submitting.*

You may not use late days on this homework. All parts of this assignment are auto-graded on Grin, so you do not need to submit anything to Gradescope. There is a walk-through video on Panopto here of how to use Grin.

1. Regular Expressions [Online] (20 points)

For each of the following languages, construct a regular expression that matches exactly the given set of strings.

You will submit (and check!) your answers online at https://grin.cs.washington.edu/. Think carefully before entering a submission; you only have 5 guesses.

Because these are auto-graded, we will not award partial credit.

(a) [5 Points] Strings over $\{a, b, c\}$ where every a is immediately followed by a b or c.

(b) [5 Points] Binary strings where every instance of 11 is followed by 00.

(c) [5 Points] Strings over $\{a, b\}$ with an even number of b's

(d) [5 Points] Binary strings that begin with a 1 and which have length ℓ such that $\ell \,\%\, 4 = 3$

2. Context-Free Grammars [Online] (20 points)

For each of the following languages, construct a context-free grammar that generates exactly the given language.

You will submit (and check!) your answers online at https://grin.cs.washington.edu/. Think carefully before entering a submission; you only have 5 guesses.

Because these are auto-graded, we will not award partial credit.

(a) [5 Points] Strings from the alphabet $\Sigma = \{a, b\}$ with an equal number of a's and b's.

(b) [5 Points] The language $\{a^x b^y a^{3x+y} : x, y \geq 0\}$.

(c) [5 Points] Binary strings with an odd number of 0's.

(d) [5 Points] Strings from the alphabet $\Sigma = \{0, 1, 2\}$ that have an equal number of 0's and 1's, and whose consecutive digits differ by exactly 1. For example, 0121010 is in the language.

3. DFAs [Online] (20 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) [5 Points] Strings from the alphabet $\Sigma = \{a, b\}$ with an even number of a's **or** an odd number of b's.

(b) [5 Points] Binary strings with at least three 0s.

(c) [5 Points] Binary strings not containing the substring 1011.

(d) [5 Points] Binary strings such that none of their runs of 0s have an even length.

A "run" of 0s is a string of consecutive 0s with the following properties:

- A run of 0s must have length 1 or greater.
- The first 0 in the run must either be the start of the string, or must be preceeded by a 1.
- The last 0 in the run must either be the end of the string, or must be followed by a 1.

For example, "00" contains exactly one run of 0s (of length 2). "000010" contains two runs of 0s (one of length 4, the other of length 1). The string "111111" contains no runs of 0s.

Now consider some example strings. The string "0001100" is not accepted because it has a run of 0s of length 2 at the end. The string "01110001" is accepted because the first run of 0s has length 1, and the second run of 0s has length 3. "111111" is also accepted because it contains no run of 0s.

4. NFAs [Online] (15 points)

For each of the following languages, construct an NFA 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 NFA; you only have 5 guesses.

Because these are auto-graded, we will not award partial credit.

(a) [5 Points] The set of binary strings that contain 00 **and** do not contain 11.

(b) [5 Points] The set of binary strings that contain 00 **or** do not contain 11.

(c) [5 Points] The same language as the regular expression $(1 \cup 01 \cup 001)^*(\varepsilon \cup 0 \cup 00)$.