HW 2

Due on **Wednesday, September 7, 2022 at 9am CST**

CS/ECE 374A: Intro. Algorithms & Models of Computation, Fall 2022 Version: **1.0**

Submission instructions as in previous homeworks.

3 (100 PTS.) 374 Balanced.

A string s over $\Sigma = \{0, 1\}$ is ***balanced*** if (i) $\#_0(s) = \#_1(s)$, and for any prefix p of s we have that $\#_0(p) \geq \#_1(p)$. Here, for any character $c \in \Sigma$, and any string $w \in \Sigma^*$, the quantity $\#_c(w)$ is the number of times the character c appears in w. Thus, the strings

0101010101, 00101101001011, 00011101, and 010011,

are balanced, while

10, 001, 001110, 0001110111111, and 01001110,

are not balanced. A string w is 374 ***balanced*** if w is balanced, and for any prefix p of w, we have that $0 \leq \#_0(p) - \#_1(p) \leq 374$.

For both languages specified below, describe *formally* a DFA that accepts them. In addition, explain informally and precisely the idea beyond your DFA and how it works.

3.A. (50 PTS.) Let L_1 be the language of all 374 balanced strings.

3.B. (50 PTS.) Let L_2 be the language of all binary strings w, such that:

(i) w is 374 balanced,
(ii) $|w|$ is divisible by 16, and
(iii) w contains 0000 as a substring.

(The language of all balanced strings is not regular, so this question is interesting because the more restricted languages L_1 and L_2 are regular.)

4 (100 PTS.) Freedom of regular expressions.

For each of the following languages over the alphabet $\{0, 1\}$, give a regular expression that describes that language, and briefly argue why your expression is correct.

4.A. (30 PTS.) The language containing all strings that do not contain 000 as a substring.

4.B. (70 PTS.) All strings that do *not* contain 0110 as a subsequence.

(Hint: (A) Break the input string into runs – a ***run*** of a string w is a maximal substring s all made of the same character. (B) You might want to solve an easier version of this question first, where 010 is a forbidden subsequence.)