Introduction:

We've recently explored regular expressions, their uses, applications and their representation in finite state automata. We've also seen how to covert between various representations of regular grammars. Demonstrate your knowledge and facility with these mechanisms below.

Methodology:

Submit a PDF with your answers to the below on Sakai on or before 2355 30 September.

Since some of the below require drawing FSAs, you can write them by hand and scan them, but recall that a human will need to examine whatever you submit.

0. Let L1 be a formal language using the binary digits 0 and 1 as its character set, such that a string is in L1 if and only if it has 3 or more “1”s in a row somewhere in it.

    A. Draw a deterministic finite state automaton (FSA) that recognizes L1.
    N.B. it may be eaasier to draw a NFA first and convert it

    B. Write a regular expression whose language is L1.

1. Let L3 be a formal language using the letters o, a, and y, such that a string is in L3 if and only if it has the string “yoya” somewhere within it.
N.B.: “yyyoyoyaaa” is in L3, as are; “yyoya”, “yoyoya”, and “yoyaaya”,
but “yoaya” is not in L3.

    A. Draw a nondeterministic FSA that recognizes L3.
    N.B. make sure it has at least five states in operation.

    B. Write a regular expression whose language is L3.

    C. Draw a deterministic FSA that recognizes L3.

2. Describe in English the language of each of the following regular expressions:

   A. a*|b*

   B. (ab)*

3. Draw a Deterministic FSA that accepts a string made up of the characters x and y if and only if it has at most two “x”s and has more “x”s than “y”s. It may or may not accept the empty string “”, whichever you prefer.
   E.G.: your DFA should accept the strings “x”, “xyx” and “yxx” but not “y” or “xy” or “xyxyx”.

Evaluation:

All on time submissions will be graded out of 100 points:

0.	1.	2.	3. 5pts
A. 15pts	A. 5pts	A. 15pts	
B. 15pts	B. 15pts	B. 15pts	
	C. 15pts		