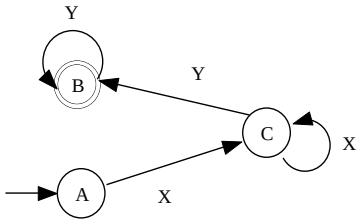
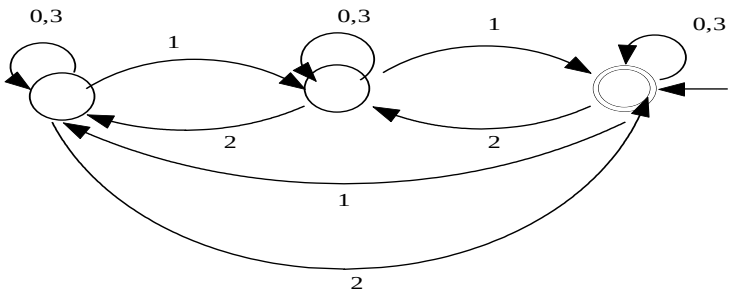
Regular Expressions and FSAs

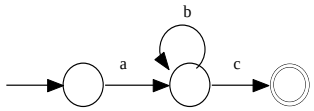
A Finite State Automation (FSA) has four components: an input alphabet (those letters or strings which are legal inputs); a set of transition rules to advance from state to state (given a current state and an element of the input alphabet, what is the next state); a unique start state; and one or more final states. We can draw the FSA, as shown below, by representing each state as a circular node; the final state as a double circle; the start state as the only node with an incoming arrow; and the transition rules by the strings on the edges connecting the nodes. When labels are assigned to states, they appear inside the circle representing the state.



If there is a path from the start state to a final state by which an input string can be parsed, then the input string is said to be "accepted" by the FSA. The FSA above will accept strings composed of one or more *x*'s followed by one or more *y*'s (e.g., *xy*, *xxy*, *xxxyy*, *xyyy*). A more complicated example is given below. The input alphabet is the digits 0, 1, 2 and 3. This FSA accepts those strings whose base 4 value is a multiple of 3. It does this by summing the value of the digits: when in the left node, the running sum of the digits has a value, modulo 3, of "1"; in the middle node, "2", and in the right node, "0". A base four number, like a base ten number, is a multiple of 3 only when its digits sum to a multiple of 3.



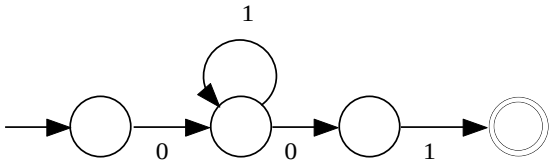
Just like Boolean Algebra is a convenient algebraic representation of Digital Electronic Circuits, a regular expression is an algebraic representation of an FSA. For example, the regular expression corresponding to the first FSA given above is xx^*yy^*. The regular expression for the second FSA is extremely complex!



ab^*c

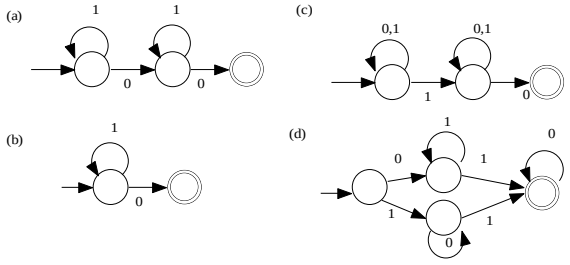
Sample Problems

Find a simplified regular expression for the following FSA:



The expression 01*01 is read directly from the FSA. It is in its most simplified form.

List all the following FSAs which represent 1*01*0.



Only choice (a) is correct: the other FSAs correspond to the following regular expressions:

(b) 1*0

(c) (0**U**1)*1(0**U**1)*0

(d) 01*10***U**10*10*

Which regular expressions are accepted by the following regular expression (u=A,…,Z and l=a,…,z)?

ull*_u._ul*

A) Ullyses_S._Grant B) Ms_ C._Blackstone C) Emmaus_H._School
D) U_S._A E) Abe_Lincoln F) Madam_Im_Adam

Solution: A, B, C

D doesn't have an 'l' after the first 'u'. E has no 'u._' and F has no '.' after the l.

Which regular expressions are equivalent to the following regular expression?

(ab*ba*)*

A) a*b*a* B) (a*b*)* C) ab*ba*
D) (ab*)*(ba*)* E) (ab)* F) a*b*a*

Solution: D

A, C, and F don't accept 'abab'. E doesn't accept 'abba'.

Which regular expressions are equivalent to the following regular expression?

1*01(01)*1100*

A) 0010100 - fails - must start with 01 not 00
B) 101011100 - matches
C) 01010101100 - fails - missing a 1, should end with 11100
D) 1010110 - fails - should end with 1110
E) 01110 - matches

B, E