Problem 11

Evan Burton, ID: 010945129, Undergraduate

November 8, 2016

1 Problem

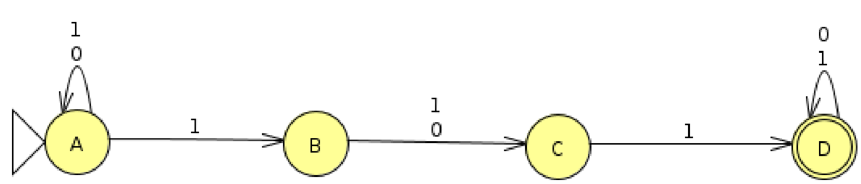
How many subsets of $\{0, 1, \cdots, 31\}$ do not contain two elements which differ by 2?

2 A Nondeterministic Finite State Machine

The question is equivalent to "how many bit strings of length 32 do not contain 101 or 111" since we can model choosing subsets as constructing a bit string by placing a 1 if an element is in the subset or 0 if it is not.

Finite state machines have 5 features: A set of states, an initial state, a language, a transition function, and a final state. Here, states are represented as circles, the initial state has a large triangle, the language is {"0", "1"}, the transition functions will be given by the arrows, and the final state(s) will have concentric circles. To be formal, one should define the transition functions as a state table, but that would take far too much room and is not as clear as graphs.

Proceeding forward, we can construct a nondeterministic machine (multiple choices exist for a transition) which recognizes bit strings which contain 101 or 111:

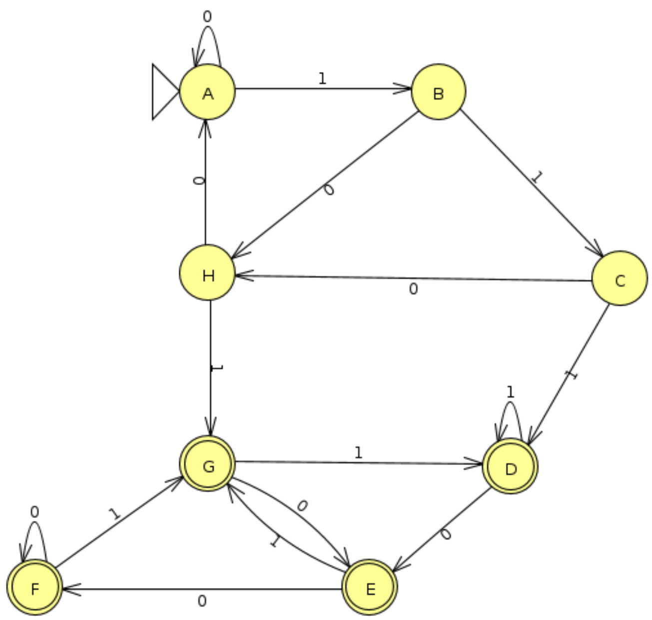


However, this is not directly helpful in its present form. Luckily, every nondeterministic finite state machine has a deterministic counterpart, and therefore, we can construct another equivalent machine using the standard conversion algorithm[1].

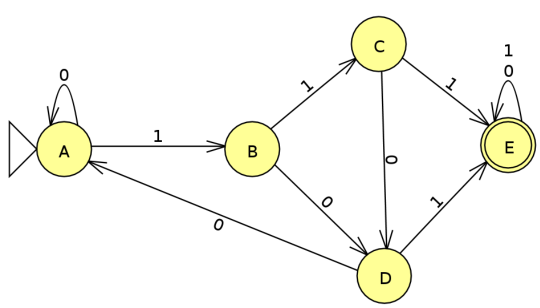
[1]See *Introduction to the Theory of Computation, 3^{rd} ed.* by Michael Sipser page 57.

3 The Deterministic Finite Automaton

After going through the conversion algorithm, we are fortunate to end with a finite state machine with only 8 states, since there was a possibility of 16.



This can be represented by a matrix just as any other directed graph can, but the resulting matrix would be 8 by 8 and we can do better. Notice that the only way to get to a final state is by getting a "1" at C or H, therefore we can remove all but one final state and make both C and H transition to it on "1". This results in the 5 state system:



The only thing left to do is to represent the new DFA as a matrix which can be done easily.

Using the smaller graph, we obtain the matrix representation of the DFA (each row must sum to 2 since each state must branch for both "0" and "1"):

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

We can find the walks of length 32 along the DFA by calculating A^{32}, and the amount of bit strings which contain 101 or 111 will be in the top-right entry (that is, the entry corresponding to the path Initial→Final State). Consequently, the number of bit strings which do <u>not</u> contain 101 or 111 is the sum of the entries a_{11} to a_{14} and the sum of all the entries of the first row of A^n is always 2^n when n is a nonnegative integer.

Calculating A^{32}, we finally obtain:

$$A^{32} = \begin{bmatrix} 255049 & 1576239 & 974169 & 1576239 & 4288290240 \\ 1576239 & 974169 & 602070 & 974169 & 4290840648 \\ 974169 & 602070 & 372100 & 602070 & 4292416887 \\ 1576239 & 974169 & 602070 & 974170 & 4290840648 \\ 0 & 0 & 0 & 0 & 4294967296 \end{bmatrix}$$

The sum of the first 4 entries in the first row is $6,677,056$ and the sum of all the entries of that row is: $4,294,967,296 = 2^{32}$. Therefore, the number of subsets of $\{0, 1, \cdots, 31\}$ which do not contain elements which differ by 2 is the amount of paths which begin at the start state and do not end in the final state, which is 6,677,056.