## Problem 1.

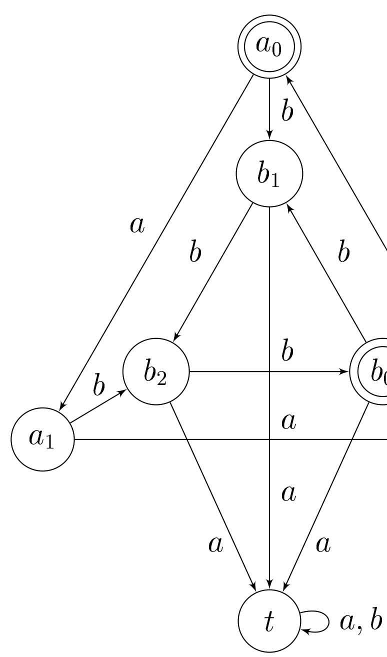
Define a DFA, simplified to the best of your abilities, to recognize the language
$L = \{a^i b^j : (i + j) \bmod 3 = 0\}$.

### Solution

Note that the length of a string in the language will be (i+j), thus the specification of the language is simply looking for an sequence of a's followed by b's, such that the string has a length that is a multiple of 3.

$$M = (\{a_0, a_1, a_2, b_0, b_1, b_2, t\}, \{a, b\}, \delta, a_0, \{a_0, b_0\})$$

Where $\delta : \{a_0, a_1, a_2, b_0, b_1, b_2, t\} \times \{a, b\} \to \{a_0, a_1, a_2, b_0, b_1, b_2)\}$ is defined by the following directed graph:



## Problem 2.

Describe in a short English sentence the language accepted by the DFA, and give a regular expression for it. Then, define a 5 state NFA that accepts the same language.

### Solution

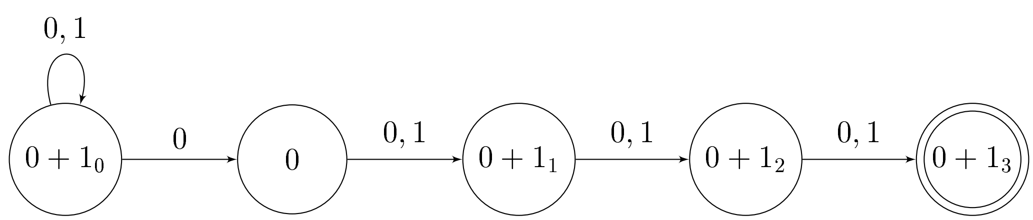
The NFA defines strings of bits, who have a length of at least 4, and who's last 4 bits form a binary number between 0 and 7. That is to say, of the last 4 bits, the first will be 0, and the other 3 will be either 0 or 1.

$$(0+1)^*0(0+1)(0+1)(0+1)$$

We can define an NFA based on that regular expression...

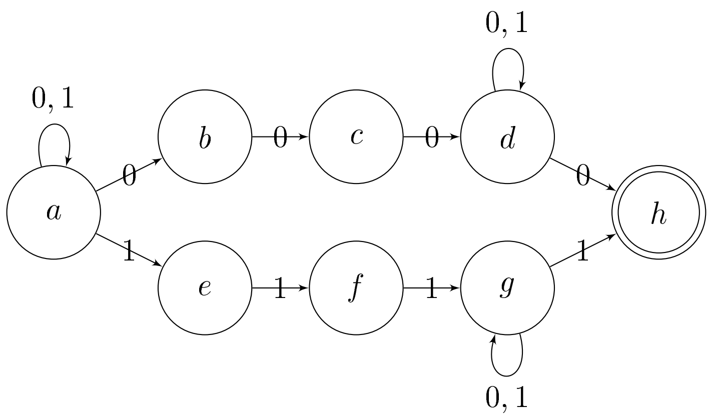
$$M = (\{0+1_0, 0, 0+1_1, 0+1_2, 0+1_3\}, \delta, \{0+1_0\}, \{0+1_3\})$$

Where $\delta : \{0+1_0, 0, 0+1_1, 0+1_2, 0+1_3\} \times \{0,1\} \to \{0+1_1, 0, 0+1_1, 0+1_2, 0+1_3\}$ is defined by the following directed graph:



## Problem 3.

Define a DFA equivalent to the following NFA.




### Solution

solution goes here