1. Let L be the set of all strings w in $\{A, B\}^*$ for which $\#(ABBA, w) \geq 2$.

(a) Give a regular expression for L, and briefly argue why your expression is correct.

Solution: $(A+B)^*(ABBA)(A+B)^*(ABBA)(A+B)^* + (A+B)^*(ABBABBA)(A+B)^*$

The first term describes all strings that contain at least two *disjoint* occurrences of the substring ABBA.

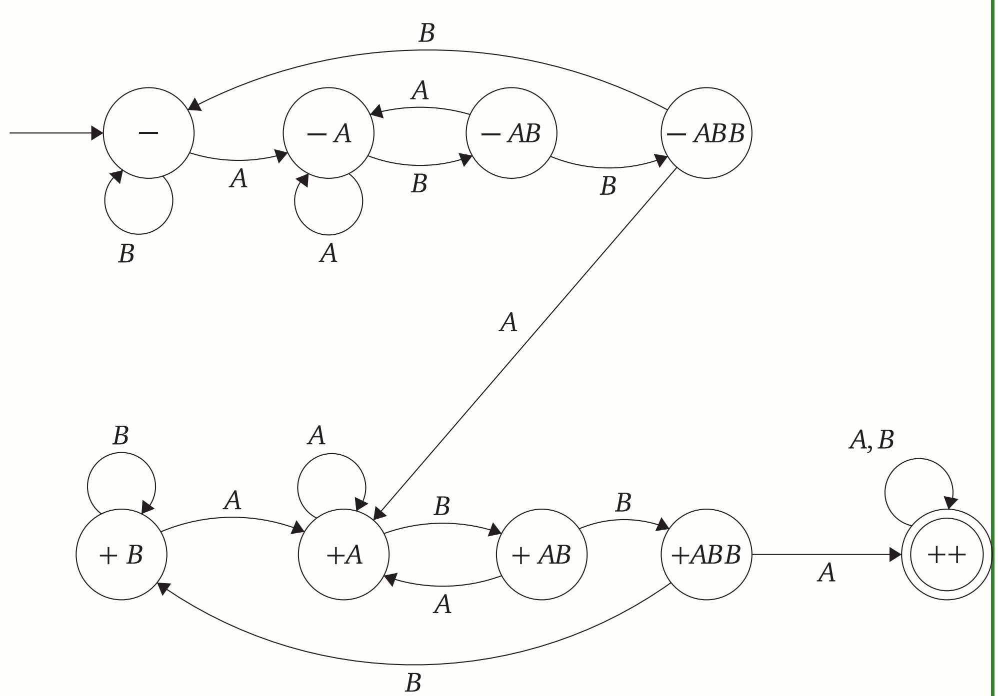
The second term describes all strings that contain the substring ABBABBA, and therefore either contain at least two *overlapping* occurrences of the substring ABBA.

(Neither subexpression attempts the match the *first* two or *last* two occurrences of ABBA, and the first subexpression does not attempt to match *adjacent* occurrences of the substring ABBA.) ■

Rubric: 5 points: standard regular expression rubric (scaled). This is not the only correct solution.

(b) Describe a DFA over the alphabet $\Sigma = \{A, B\}$ that accepts the language L.

Solution:



The nine states of the DFA have the following meanings:

- −: We have not seen the substrings ABBA, and we have not just read a non-empty prefix of ABBA or BAAB. This is the start state.
- −A: We have not seen the substring ABBA, and we just read A.
- −AB: We have not seen the substring ABBA, and we just read A followed by B.
- −ABB: We have not seen the substring ABBA, and we just read A followed by

B followed by another B.

- +A: We have seen the substring ABBA exactly once, and we just read A.
- +B: We have seen the substring ABBA exactly once, and we just read B, but this was not immediately preceded by either A or AB.
- +AB: We have seen the substring ABBA exactly once, and we just read A followed by B.
- +ABB: We have seen the substring ABBA exactly once, and we just read A followed by B followed by another B.
- ++: We have seen the substring ABBA at least twice. This is the only accepting state.

■

Rubric: 5 points: standard DFA rubric (scaled). This is not the only correct solution.

2. Let L denote the set of all strings $w \in \{0, 1\}^*$ that satisfy *at most two* of the following conditions:

- The number of times the substring 01 appears in w is *not* divisible by 3[1].
- The length of w is even.
- The binary value of w equals $2 \pmod 3$.

Formally describe a DFA with input alphabet $\Sigma = \{0, 1\}$ that accepts the language L, by explicitly describing the states Q, the start state s, the accepting states A, and the transition function δ.

Solution (formal description):

$$Q = \{0, 1\} \times \{0, 1, 2\} \times \{0, 1\} \times \{0, 1, 2\}$$
$$s = (1, 0, 0, 0)$$
$$A = \{(a, b, c, d) \mid b = 0 \text{ or } c = 1 \text{ or } d \in \{0, 1\}\}$$

$$\delta((0, b, c, d), 0) = \big(0,\ b,\ (c + 1) \bmod 2,\ (2d) \bmod 3\big)$$
$$\delta((0, b, c, d), 1) = \big(1,\ (b + 1) \bmod 3,\ (c + 1) \bmod 2,\ (2d + 1) \bmod 3\big)$$

$$\delta((1, b, c, d), 0) = \big(0,\ b,\ (c + 1) \bmod 2,\ (2d) \bmod 3\big)$$
$$\delta((1, b, c, d), 1) = \big(1,\ b,\ (c + 1) \bmod 2,\ (2d + 1) \bmod 3\big)$$

The state (a, b, c, d) indicates the following:

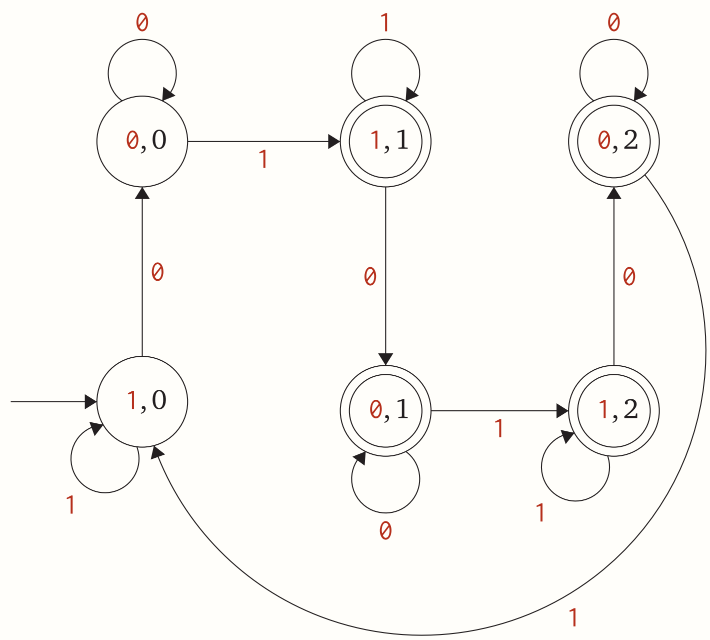
- a is the last symbol read by the DFA, or 1 if the DFA hasn't read anything yet.
- b is the number of times the DFA has read the substring 01, modulo 3.
- c is the number of times the DFA has any symbol, modulo 2.
- d is the binary value of the string red so far, modulo 3.

■

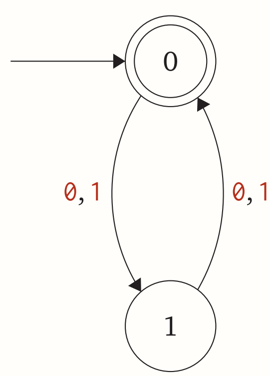
[1]Recall that a is divisible by b if and only if $a \equiv 0 \pmod b$.

Solution (product construction): Our DFA M is the product of three smaller DFAs:

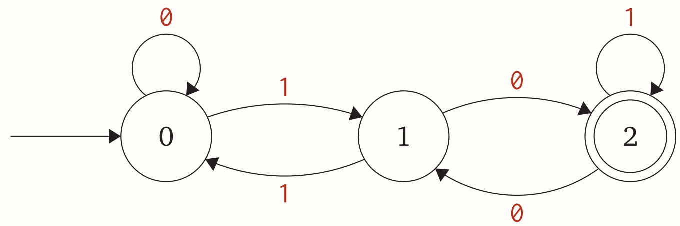
- The first DFA A accepts all strings in which the substring 01 appears $3k+1$ or $3k+2$ times for integer $k \geq 0$. Each state records the last symbol read (or 1 if nothing has been read yet), and whether the DFA has read $3k$ or $3k+1$ or $3k+2$ number of 01s.



- The second DFA B accepts all strings of even length. Each state records whether an even or odd number of symbols have been read so far.



- The third DFA C accepts all strings whose binary value equals $2 \pmod 3$. Each state records the binary value $\pmod 3$ of the string read so far.



Each state of the product DFA M is a triple (a, b, c), where a is a state of A, b is a state of B, and c is a state of C. For example, the start state of M is $((1, 0), 0, 0)$.

Finally, a state (a, b, c, d) of M is accepting if and only if $b = 0$ or $c = 1$ (the string has odd length) or $d \in \{0, 1\}$ (the binary value equals 0 or $1 \pmod 3$). ■

Rubric: 10 points: standard DFA rubric. These are not the only correct solutions.