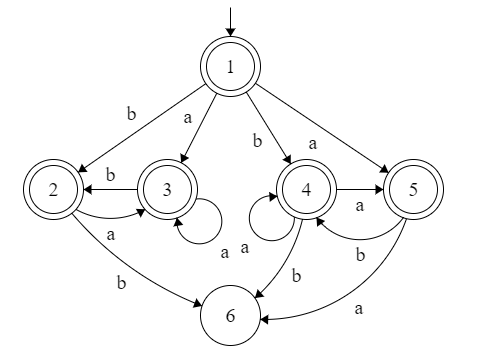
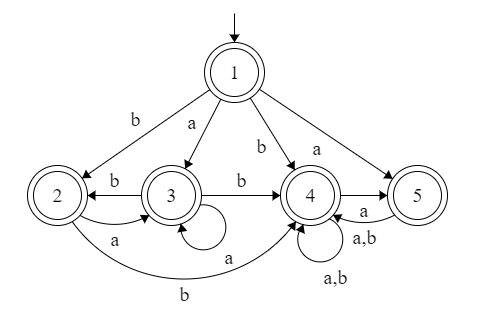
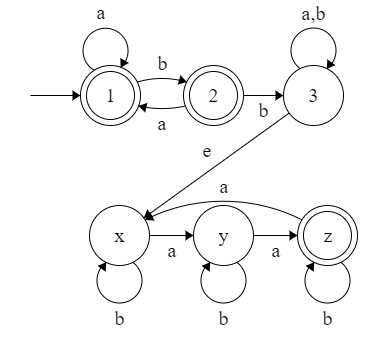
1. E = new state: 6

|  | a | b |
| --- | --- | --- |
| 1 | 35 | 24 |
| 2 | 3 | E |
| 3 | 3 | 2 |
| 4 | 45 | E |
| 5 | E | 4 |
| 6 | 6 | 6 |

Different version, is this still ok?

|  | a | b |
| --- | --- | --- |
| 1 | 35 | 24 |
| 2 | 3 | E |
| 3 | 3 | 2 |
| 4 | 45 | E |
| 5 | E | 4 |
| 35 | 3 | 4 |
| 24 | 3 | 45 |
| 45 | 45 | 4 |

1. 
2. If L is regular, then bw(w) is regular. By reversing the DFA for w, you can get a working DFA to ding the reverse string. Edits to the w DFA:
   1. Make the acceptance state the start state
   2. Make the original start state the acceptance state.
   3. Reverse arrow directions between states.

There would not be multiple acceptance states or start states if it's looking for a single string or substring so the start state and acceptance state is always 1 to 1 which can be flipped easily. The arrow transitions would swap direction a -> b to a <- b, this does not introduce or remove any states assuming there were no dead states.

1. RS is a regular expression whose language is L. R\* is a regular expression whose language is L\*. All strings in the language L are accepted to the given DFA which makes it regular.