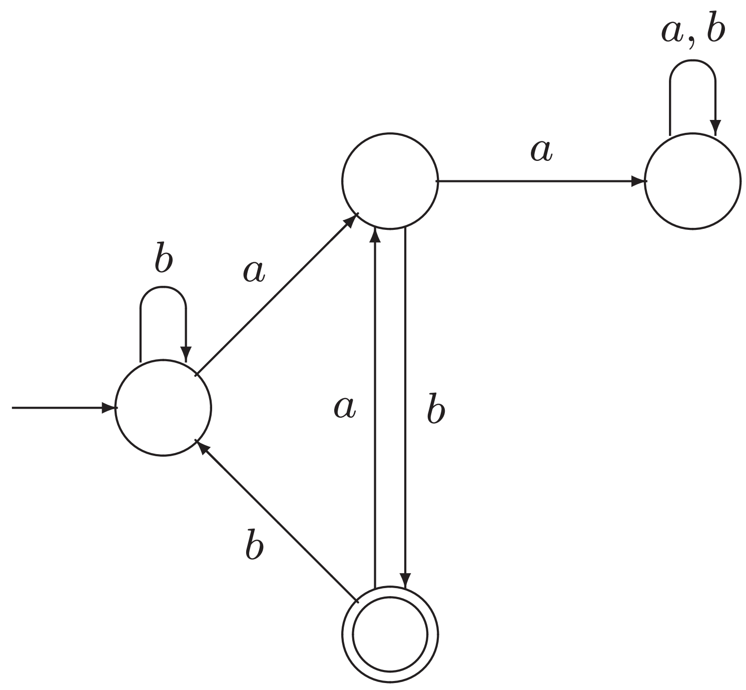
From exercise class 3:

Exercise 2.13. ♣

For the FA pictured below, show that there cannot be any other FA with fewer states accepting the same language.



Exercise 2.33. ♣

Let x be a string of length n in $\{a, b\}^*$, and let $L = \{x\}$.

How many equivalence classes does $\equiv_L$ have? Describe them.

Hint: first draw an FA accepting L.

Exercise 2.36. ♣

For a certain language $L \subseteq \{a, b\}^*$, $\equiv_L$ has exactly four equivalence classes. They are [Λ], $[a]$, $[ab]$ and $[b]$.

It is also true that the three strings a, aa, and abb are all equivalent,
and that the two strings b and aba are equivalent.

Finally, $ab \in L$, but Λ and a are not in L, and b is not even a prefix of any element of L.

Draw an FA accepting L.

Exercise 2.37.

Suppose $L \subseteq \{a, b\}^*$ and $\equiv_L$ has three equivalence classes. Suppose they can be described as the three sets $[a]$, $[aa]$, and $[aaa]$, and also as the three sets $[b]$, $[bb]$, and $[bbb]$.

How many possibilities are there for the language L? For each one, draw a transition diagram for an FA accepting it.

Exercise 2.38.

In each part, find every possible language $L \subseteq \{a, b\}^*$ for which the equivalence classes of $\equiv_L$ are the given three sets.

a. ♣ ♠ $\{a, b\}^*\{b\}$, $\{a, b\}^*\{ba\}$, $\{\Lambda, a\} \cup \{a, b\}^*\{aa\}$

Exercise 2.40. ♣

Consider the language

$$L = \mathit{AEqB} = \{x \in \{a,b\}^* \mid n_a(x) = n_b(x)\}$$

Let x and y be arbitary elements of $\{a,b\}^*$ (not necessarily in L)

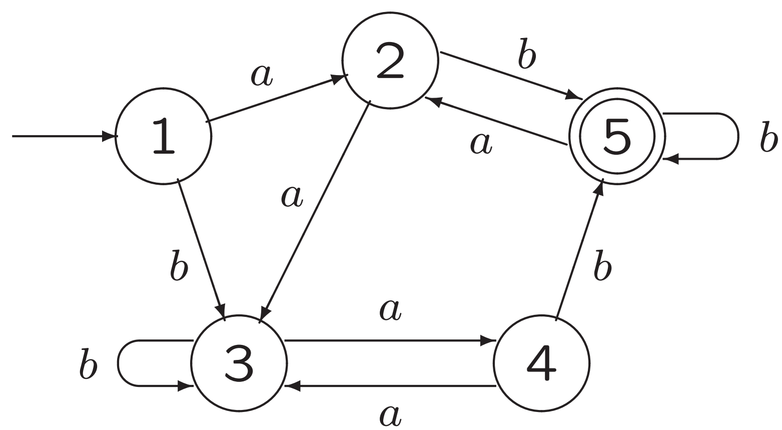
a. Show that if $n_a(x) - n_b(x) = n_a(y) - n_b(y)$, then $x \equiv_L y$.

b. Show that if $n_a(x) - n_b(x) \neq n_a(y) - n_b(y)$, then x and y are L-distinguishable.

c. Describe all the equivalence classes of $\equiv_L$.

Exercise 2.55.
For each of the FAs below, use the minimization algorithm described in Section 2.6 to find a minimum-state FA accepting the same language. (It's possible that the given FA is already minimal.)

a. ♣



Exercise 2.55.
For each of the FAs below, use the minimization algorithm described in Section 2.6 to find a minimum-state FA accepting the same language. (It's possible that the given FA is already minimal.)

c.