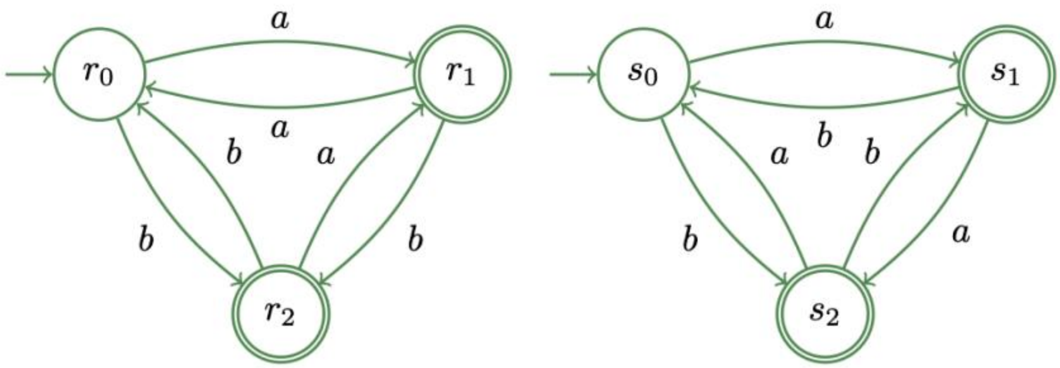
1. The two DFAs do not accept the same language, DFA R accepts the string ab and DFA S does not.

Proof:

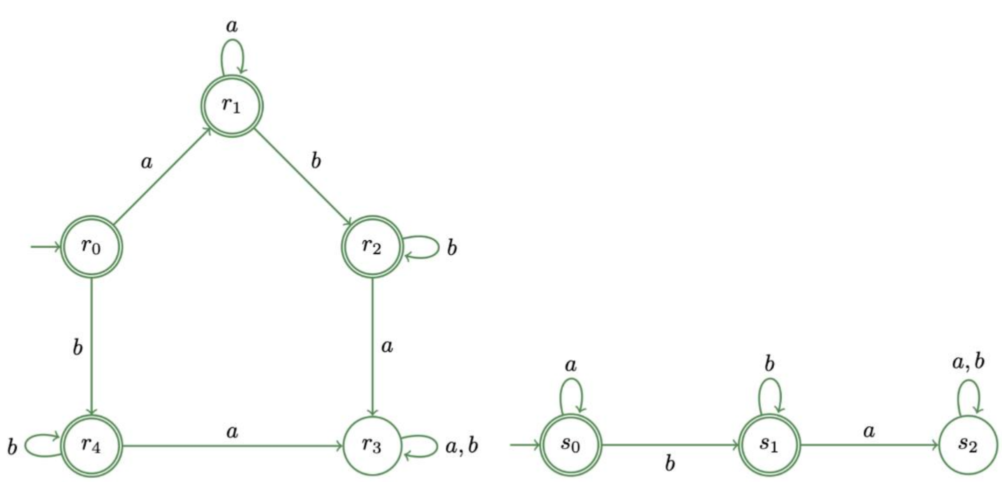


These are the two DFAs that I am trying to prove are different, DFA with r_0-r_2 is DFA R. The other one is DFA S. If you enter the string ‘ab’ into DFA R you reach the accepting state r_2. If you enter the same string ‘ab’ into DFA S you reach the *non*-accepting state s_0. Since the DFAs have at least one string that is different in their language then they do not accept the same language.

2. The two DFAs do accept the same language

Proof:

Prove whether or not the following two DFAs accept the same language.



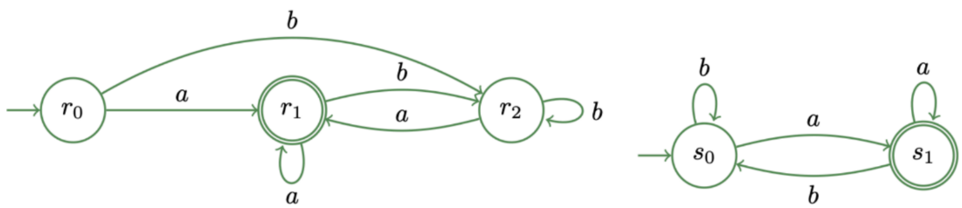
These are the two DFAs that I am trying to prove are the same, DFA with r0-r4 is DFA R. The other one is DFA S.

States	a	b

+r0,s0	r1,s0	r4,s1
+r1,s0	r1,s0	r2,s1
+r4,s1	r3,s2	r4,s1
+r2,s1	r3,s2	r2,s1
r3,s2	r3,s2	r3,s2

This is a product table of the two DFAs above, the '+' means that state is an accepting state. r0,r1,r4,r2,s0,s1 are all accepting states. All of the product table accepting states are pairs from r0,r1,r4,r2,s0,s1. This means that when you do the symmetric difference on the two DFAs you do not come up with any accepting states, therefore the two DFAs are equal.

3. The language accepted by the DFA R is a subset of the language accepted by the DFA S



These are the two DFAs that I am trying to prove are the same, DFA with r0-r2 is DFA R. The other one is DFA S.

States	a	b
r0,s0	r1,s1	r2,s0
+r1,s1	r1s1	r2,s0
r2,s0	r1,s1	r2,s0

This is a product table of the two DFAs above, the '+' means that state is an accepting state.

r1,s1 are all accepting states. All of the product table accepting states are pairs from r1,s1. This means that when you do the symmetric difference on the two DFAs you do not come up with any accepting states, therefore the two DFAs are equal. And because they are equal that means they are subsets of each other, therefore the language accepted by DFA R is a subset of the language accepted by DFA S.

4. No, in order to prove this I am going to use the following method

1. Mark the start state
2. Highlight all edges that the Marked states lead to
3. Remove all Marked states
4. The highlighted states are now the marked states
5. Repeat steps 2-4 until you reach an accepting state

6. If you reach an accepting state then you found a string that is accepted, if you don’t then the DFA doesn’t accept any string

using this method

1. Mark = r0
2. Highlight = r4,r5
3. Mark = r4,r5
4. Highlight = none
5. Mark = none
6. No accepting state found