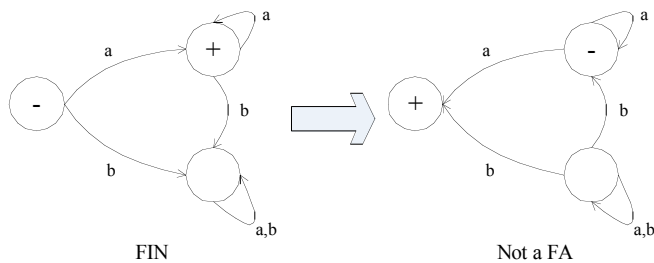


Solution of Assignment #2

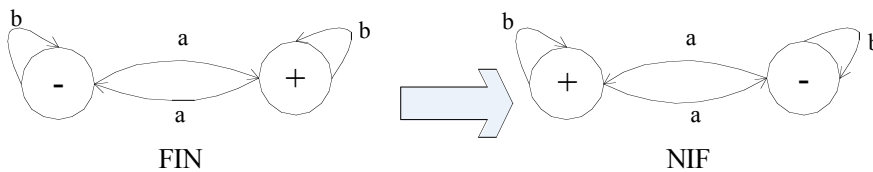
Chapter 5, Exercise 12, pg 72

i)



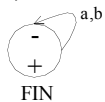
The changed FIN is not a FA because the start state has more than one path for a.

ii)



iii) The language accepted by NIF is the reverse of every word in the language accepted by FIN. If there was a path from one state to another on FIN by a sequence of letters, then there must be a path on NIF using the sequence backwards.

iv)



The words accepted by this FA are not limited to palindromic words. It accepts the non-palindrome words *ab* and $FIN = NIF$.

Chapter 5, Exercise 18, pg 74

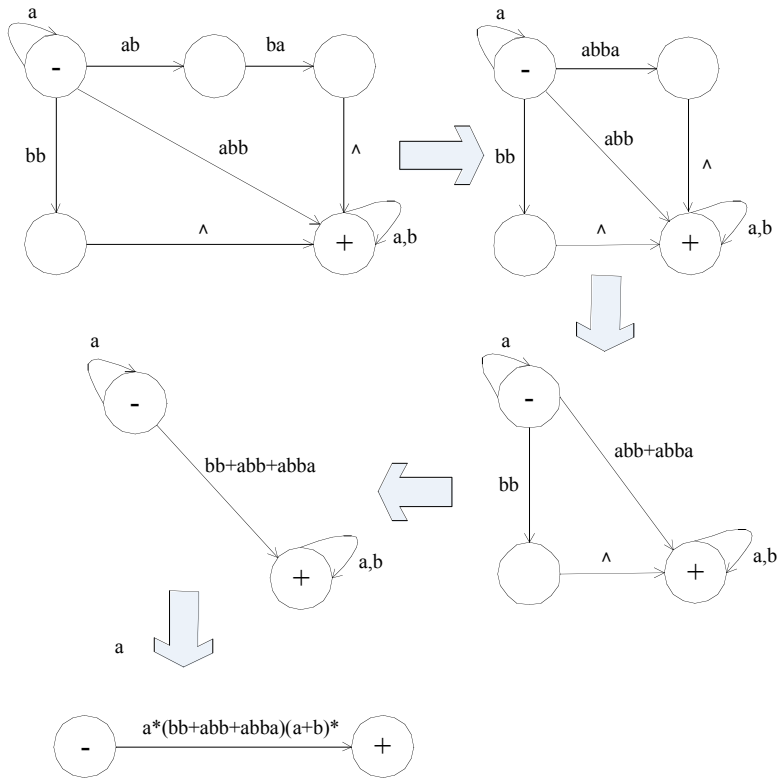
We start in state 1 and remain there until we encounter an *a*. State 2 = we have just read an *a*. Scan any *a*'s and return to state 1 on reading *c*. State 3 = we have read a *b* following an *a*. Reading an *a* puts us back to state 2 and reading a *b* sets us back to state 1. However, state 4 = we have just found a substring *abc*, and if the whole sequence was read the string is accepted.

Now states 4, 5 and 6 exactly mirror states 1, 2 and 3. Returning to state 1 indicates that we just found another occurrence of the substring *abc*. Being in one of the first three states means that we have read an even number of *abc* substrings (if any) and are in the midst of finding another one. Ending in an accepting state, 4, 5 or 6, means that we have read an odd number of *abc*'s.

Chapter 6, Exercise 20, pg 91

- i) In a machine without Λ -edges, every transition consumes at least one letter of input. Therefore, we know that Step 2 is true. The list in Step 3 is exhaustive; it includes all possible paths through the machine (where a path is represented by the sequence of integers that number its consecutive edges) and also sequences of integers that cannot represent paths because the edges are discontinuous in the graph. There is an upper bound on the magnitude of the integers used as edge numbers and on the length of the sequence. Since the list is finite, we can actually check it for the two required characteristics: 1) Does the sequence represent a path from a start state to a final state, and 2) when we concatenate the edge labels, does the result match the word we are testing? Since all possible paths are listed, if our test word has a path we must find it. Since we test paths for validity at Step 4, we are not deceived by matching edge labels on an invalid path. The list and the procedures we apply to it are finite and so this algorithm must terminate.
- (ii) In a machine with Λ -edges, it is not true that every transition consumes at least one letter of input. We can use infinitely many Λ -transitions without consuming any letters at all, and therefore we cannot set an upper bound on the length of the path that gives us the test word.

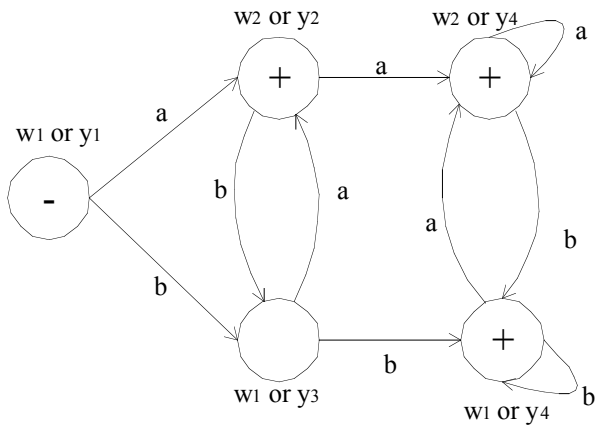
Chapter 7, Exercise 1(v), pg 142



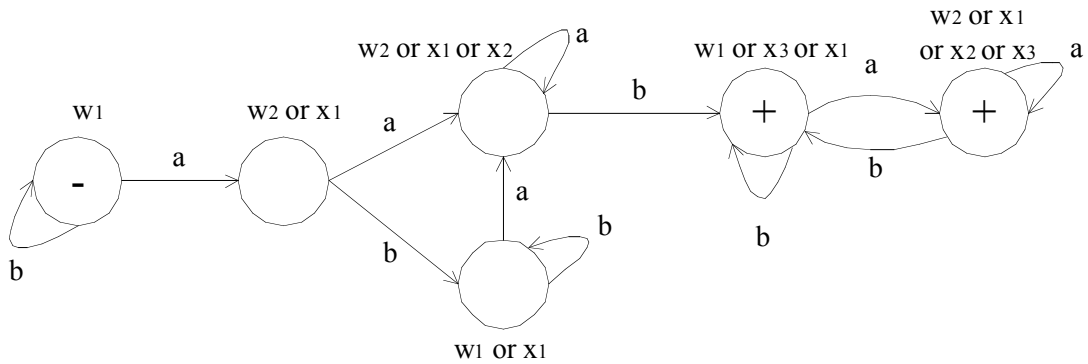
The regular expression : $a^*(bb+abb+abba)(a+b)^*$
 $= a^*bb(a+b)^*$

Chapter 7, Exercise 3(ii), pg 143

Old state	New State	a	New State	b	New State
W1, y1	Z1-	W2, y2	Z2+	W1,y3	Z3
W2,y2	Z2+	W2,y4	Z4+	W1,y3	Z3
W1,y3	Z3	W2,y2	Z2+	W1,y4	Z5+
W2,y4	Z4+	W2,y4	Z4+	W1,y4	Z5+
W1,y4	Z5+	W2,y4	Z4+	W1,y4	Z5+

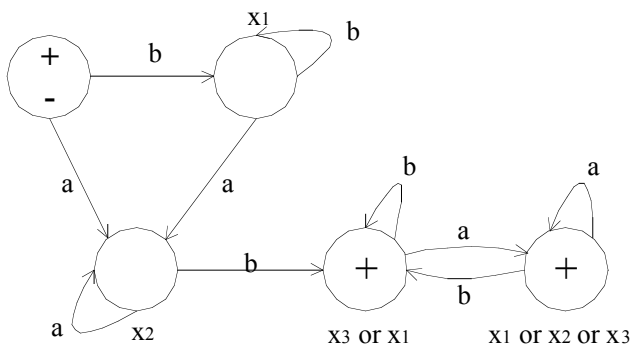


Chapter 7, Exercise 5(i), pg 143



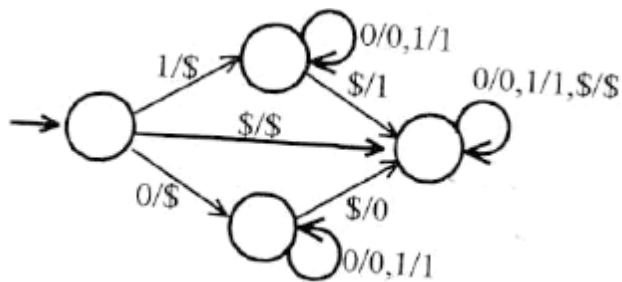
Chapter 7, Exercise 6(ii), pg 144

Old state	New state	a	New state	b	New state
	Z0+/-	X2	Z2	X1	Z1
X1	Z1	X2	Z2	X1	Z1
X2	Z2	X2	Z2	X3,x1	Z3+
X3,x1	Z3+	X3,x1,x2	Z4+	X3,x1	Z3+
X3,x1,x2	Z4+	X3,x1,x2	Z4+	X3,x1	Z3+



Chapter 8, Exercise 13, pg 167

i)

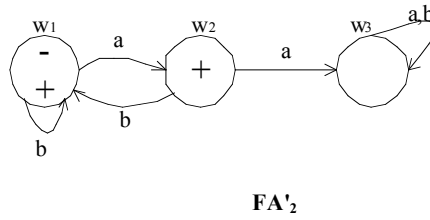
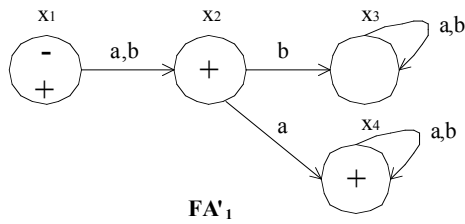
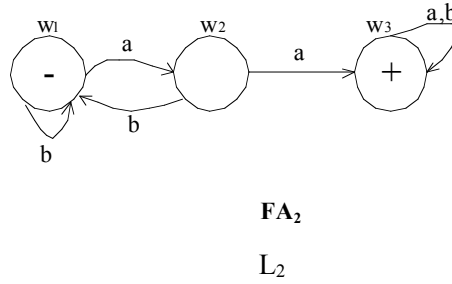
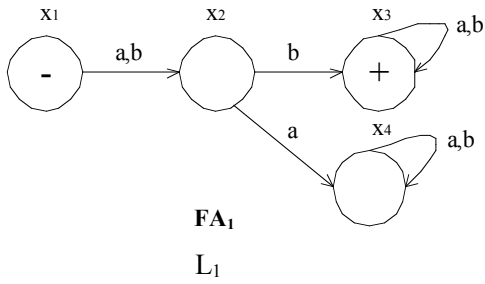


ii) There is no way of knowing when the last letter is read and its time to print the letter that was read first.

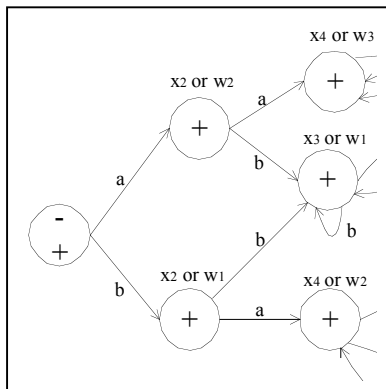
Chapter 9, Exercise 5, pg 185

L₁: (a+b)b(a+b)*, it means all strings that has the second letter to be **b**;

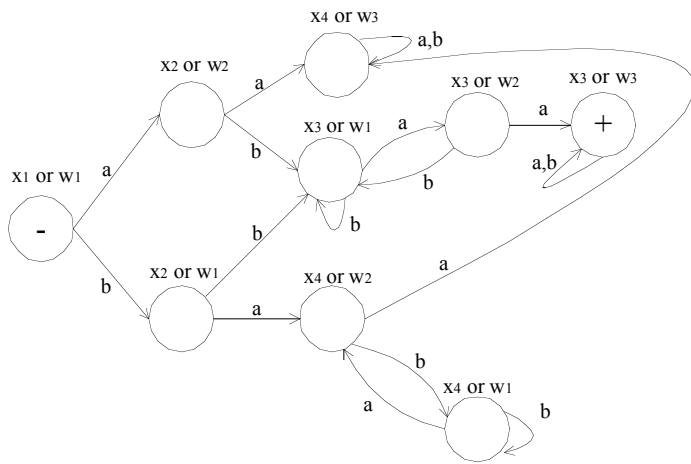
L₂: (a+b)*aa(a+b)*, it means all the strings that contain the substring **aa**;



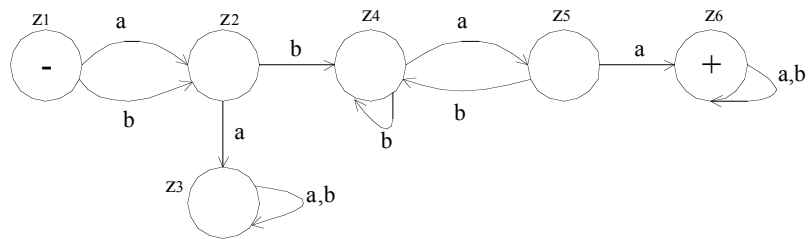
The FA of **L'₁ + L'₂**:



The FA of $L_1 \cap L_2 = (L'_1 + L'_2)'$ is:



Simplifying the FA of $L_1 \cap L_2$,



Regular Expression:

$(a+b)b(a+b)^*aa(a+b)^*$