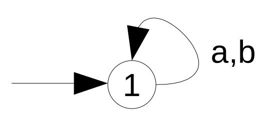
1. TG on page 142, 1ii: This TG has no non-determinism and no NULL transitions, so it is an easy transformation. First, remove edges with strings labeling them. Create a transition table showing the transitions that are present, and use ∅ to represent missing transitions. Assume the states are numbered 1 through 6 in the TG after creating new edges.

	a	b
{1}	{2}	{5}
{2}	∅	{3}
{5}	∅	{6}
∅	∅	∅
{3}	{4}	∅
{6}	{4}	∅
{4}	{4}	{4}

2. TG on page 142, 1iv: This one does not have any NULL transitions, but it does have non-determinism. Because the start state is reentrant, I create a new start with a null transition to state 1. The new start state is labeled 0. Constructing the transition table using the algorithm results in the following. Because states 1 and 4 were final states in the TG, every state in the FA is a final state, since each has both 1 and 4 in it.

	a	b
{0,1}	{1}	{1,2}
{1}	{1}	{1,2}
{1,2}	{1,3}	{1,2}
{1,3}	{1,4}	{1,2}
{1,4}	{1,4,5}	{1,2,4}
{1,4,5}	{1,4,5}	{1,2,4,6}
{1,2,4}	{1,3,4,5}	{1,2,4}
{1,2,4,6}	{1,3,4,5}	{1,2,4}
{1,3,4,5}	{1,4,5}	{1,2,4,6}

Of course the simplest FA accepting the same language is just this one, in which the state is also the final state.



But this is not obtained by any algorithm.

3. NDFA on page 146, 16iii: This one has null transitions, so null closures have to be constructed for each state. First, though, a new start state is created, labeled 0, with a null transition to state 1. The new start state is {0,1,3}. The table is

	a	b
{0,1,3}	{2,3,4}	∅
{2,3,4}	{3,4}	{1,3,4}
{3,4}	{3,4}	{1,3,4}
{1,3,4}	{2,3,4}	{1,3,4}
∅	∅	∅

4. Use the algorithm I gave in class to construct a regular expression defining the language accepted by FA_1 on page 143. Show each of the sets $L_{i,j,k}(M)$.

The table below is constructed by the algorithm I gave in class. Remember that when k = 0, the language accepted is the finite set of symbols that label edges from the first state to the second state, and that if i = j then it must also include the null string. The recursive formula is

$$L(i,j,k+1) = L(i,j,k) + L(i,k+1,k)\ L(k+1,k+1,k)^*\ L(k+1,j,k)$$

Also, there is no need to fill in the last column in any row other than rows that lead from the start state to the final state(s), and that the final expression is the union of all entries in the last column in those rows. In this case only state 2 is a final state.

L(i,j,k)	**k**		
i,j	**0**	**1**	**2**
1,1	Λ + b	b*	
1,2	a	a + b*a = b*a	b*a (b*a)*
2,1	b	b*b	
2,2	Λ + a	Λ + b*a	

5. Use the algorithm I gave in class to construct a regular expression defining the language accepted by FA_2 on page 143. Show each of the sets $L_{i,j,k}(M)$.

Using the same algorithm as above, we get the following table. In this case it is very important to try to simplify the results at each stage, to make the subsequent expressions shorter, and thus easier to understand.

L(i,j,k)	**k**			
i,j	**0**	**1**	**2**	**3**
1,1	Λ + b	b*	b*	
1,2	a	b*a	b*aa*	
1,3	∅	∅	b*aa*b	b*aa*b(a+b)*
2,1	∅	∅	∅	
2,2	Λ + a	Λ + a	a*	
2,3	b	b		
3,1	∅	∅		
3,2	∅	∅		
3,3	Λ + a + b	Λ + a + b	Λ + a + b	