Table-filling algorithm for DFA minimization

In order to minimize DFA-s, we have to be able to recognize if two distinct states of an automaton behave exactly the same, and as a result, can be replaced by a single state. We say that states p and q are *equivalent*, if:

- For **all** input strings w, $\hat{\delta}(p, w)$ is an accepting state if and only if $\hat{\delta}(q, w)$ is an accepting state (these don't have to be the same accepting states).

If two states are not *equivalent*, we call them *distinguishable*:

- State p is distinguishable from state q if there is at least one input strings w such that one of $\hat{\delta}(p, w)$ and $\hat{\delta}(q, w)$ is accepting, and the other is not accepting.

We will use the **table filling algorithm** for testing the equivalence of states. Consider the following automaton:

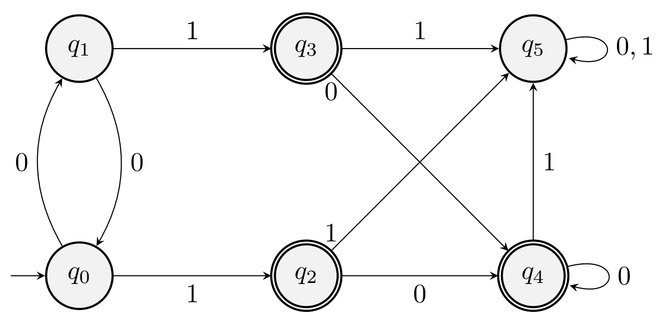


Figure 1: DFA 1

We will need a table representing all the possible pairs of states. We will use this to mark all pairs that turn out to be distinguishable.

	q_0	q_1	q_2	q_3	q_4
q_0					
q_1					
q_2					
q_3					
q_4					
q_5					

The table filling algorithm consists of the following steps:

1. For each state pair (p, q), if p is accepting and q is non-accepting, then they are distinguishable. We mark them.

2. By induction, for any other pair (p, q), they are distinguishable, if and only if for an input symbol a, $r = \delta(p, a)$ and $s = \delta(q, a)$ are distinguishable pairs. In this case, we mark them.

3. We repeat the above step until no more distinguishable pairs are found.

If the (r, s) pair in Step 2 has one accepting and one non-accepting state, then r and s are distinguishable. Otherwise, we have to check our table if (r, s) was previously marked as distinguishable. If yes, then we can mark (p, q) in our table.

Considering DFA 1, we have the following table after Step 1, marking all pairs where one state is accepting, and the other is not.

	q_0	q_1	q_2	q_3	q_4
q_0					
q_1					
q_2	x	x			
q_3	x	x			
q_4	x	x			
q_5			x	x	x

Then we consider all unmarked pairs in the table.

- (q_0, q_1) for input 0: $\delta(q_0, 0) = q_1$ and $\delta(q_1, 0) = q_0$, the (q_0, q_1) pair we got as a result is unknown yet in the table, so we don't mark.
- (q_0, q_1) for input 1: $\delta(q_0, 1) = q_2$ and $\delta(q_1, 1) = q_3$, the (q_2, q_3) pair we got as a result is unknown yet in the table, so we don't mark.
- (q_0, q_5) for input 0: $\delta(q_0, 0) = q_1$ and $\delta(q_5, 0) = q_5$, the (q_1, q_5) pair we got as a result is unknown yet in the table, so we don't mark.
- (q_0, q_5) for input 1: $\delta(q_0, 1) = q_2$ and $\delta(q_5, 1) = q_5$, the (q_2, q_5) pair we got is an accepting/non-accepting pair (and it's also marked in the table), so we **mark** (q_2, q_5).
- (q_1, q_5) for input 0: $\delta(q_1, 0) = q_0$ and $\delta(q_5, 0) = q_5$, the (q_0, q_5) pair we got as a result is unknown yet in the table, so we don't mark.
- (q_1, q_5) for input 1: $\delta(q_1, 1) = q_3$ and $\delta(q_5, 1) = q_5$, the (q_3, q_5) pair we got is an accepting/non-accepting pair (and it's also marked in the table), so we **mark** (q_3, q_5).
- (q_2, q_3) for input 0: $\delta(q_2, 0) = q_4$ and $\delta(q_3, 0) = q_4$, the (q_4, q_4) pair is actually the same state, so this will give us no useful information.
- (q_2, q_3) for input 1: $\delta(q_2, 1) = q_5$ and $\delta(q_3, 1) = q_5$, the resulting (q_5, q_5) pair shows the same, as above.
- (q_2, q_4) and (q_3, q_4) will have the same outcome, as (q_2, q_3).

This gives us the following table:

	q_0	q_1	q_2	q_3	q_4
q_0					
q_1					
q_2	x	x			
q_3	x	x			
q_4	x	x			
q_5	x	x	x	x	x

As we could not decide in the first iteration for (q_0, q_1), (q_2, q_3), (q_2, q_4) and (q_3, q_4), we check them again. However, we again get no new information. We checked all unmarked pairs, and did not modify our table, meaning that our algorithm terminates. All unmarked pairs will be equivalent state pairs. This means that we can combine (q_0, q_1) into a new state, and we will have a combination of $q_2, q_3 and q_4$ as well because of the equivalence of (q_2, q_3), (q_2, q_4) and (q_3, q_4). This results in the following automaton:

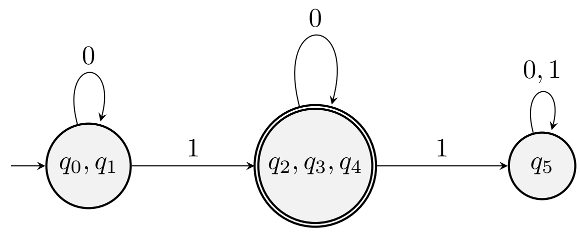


Figure 2: DFA 1 - Minimized

Another example

Minimize the following automaton:

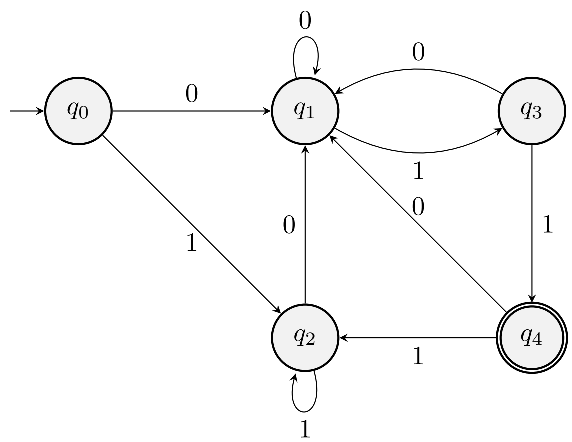


Figure 3: DFA 2

This gives us the following table after Step 1 (all accepting/non-accepting state pairs marked):

	q_0	q_1	q_2	q_3
q_0				
q_1				
q_2				
q_3				
q_4	x	x	x	x

Then we check the transition for each unmarked pair on both input symbols:

- (q_0, q_1) has no useful information on either 0 or 1 input (we get the state pairs (q_1, q_1) and (q_2, q_3), which are both unmarked).
- (q_0, q_2) is the same, as above, with state pairs (q_1, q_1) and (q_2, q_2), which are both unmarked. (as both pairs result in the same states, this will actually never be distinguishable pair)
- (q_0, q_3) for input 1 results in the state pair (q_2, q_4). This is an accepting/non-accepting pair, and is also marked in our table, se we **mark** (q_0, q_3).
- (q_1, q_2) gives state pairs (q_1, q_1) (for input 0) and (q_3, q_2) (for input 1). As these are unmarked in the table, we get no useful information.

- (q_1, q_3) for input 1 results in the state pair (q_3, q_4). This is an accepting/non-accepting pair, and is also marked in our table, se we **mark** (q_1, q_3).
- (q_2, q_3) for input 1 gives us the state pair (q_2, q_4). As we've seen before, this is an accepting/non-accepting pair, and is also marked in our table, se we **mark** (q_2, q_3).

We get the following table:

	q_0	q_1	q_2	q_3
q_0				
q_1				
q_2				
q_3	x	x	x	
q_4	x	x	x	x

We are still unsure of the pairs (q_0, q_1), (q_0, q_2) and (q_1, q_2), so we should check them again.

- As we've seen above, (q_0, q_1) for input 1 results in (q_2, q_3). While (q_2, q_3) is not an accepting/non-accepting pair, it is marked in our table, so we **mark** (q_0, q_1) as a result.
- The status of (q_0, q_2) is the same as above.
- (q_1, q_2) has the same status as (q_0, q_1): it gives us (q_2, q_3) for input 1, and as it is now marked in our table, we should **mark** (q_1, q_2) as well.

Which leads us to this new table:

	q_0	q_1	q_2	q_3
q_0				
q_1	x			
q_2		x		
q_3	x	x	x	
q_4	x	x	x	x

As the table was modified, we should check all unmarked pairs again. This only means (q_0, q_2), and its status won't change after the check. We checked all unmarked pairs, and did no modifications to the table, meaning that the algorithm is over. This time we can only combine q_0 and q_2 to a single state:

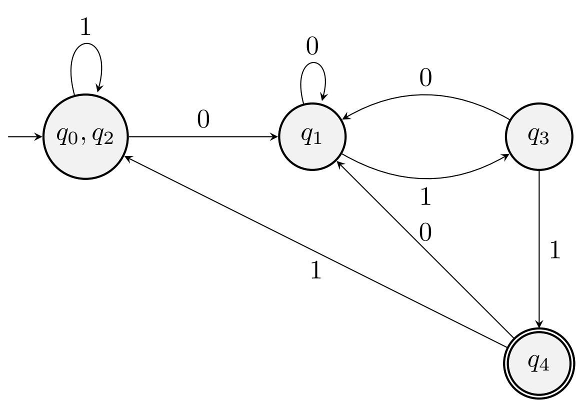
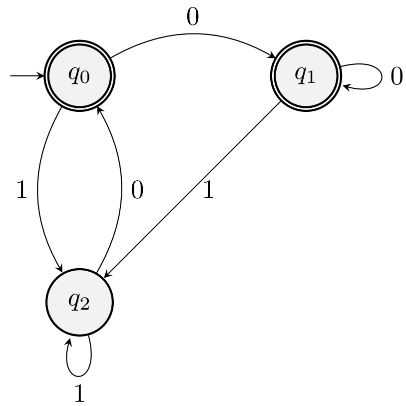
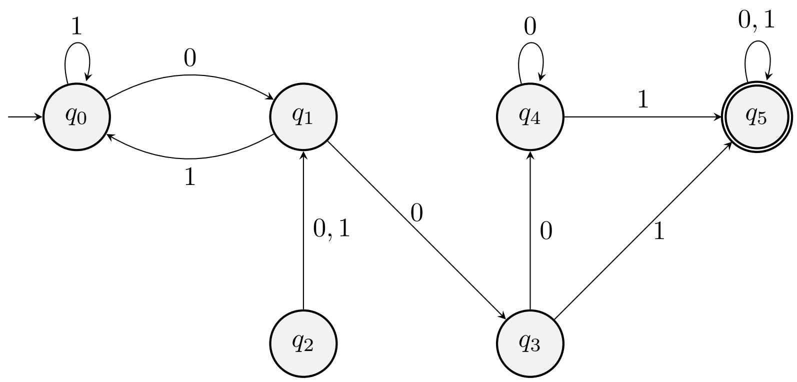


Figure 4: DFA 2 - Minimized

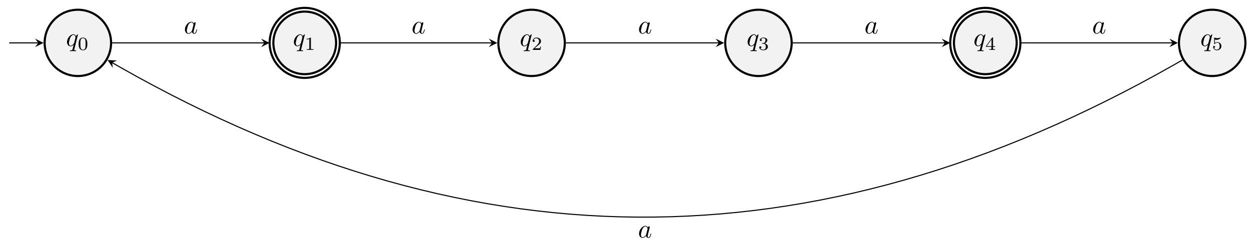
Exercises



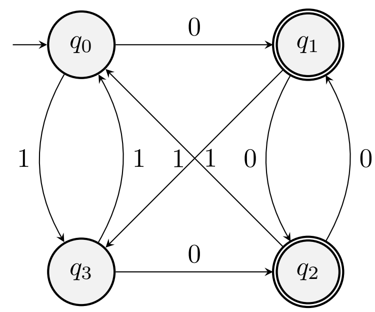
Exercise 1



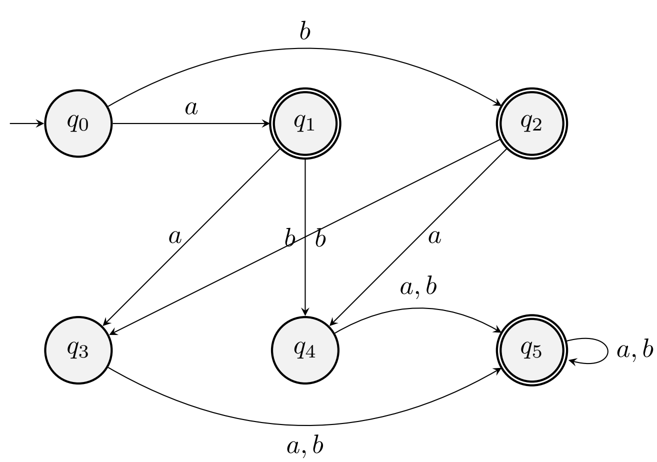
Exercise 2



Exercise 3



Exercise 4



Exercise 5