

Lecture - 22

Topic: Minimal DFAs

Scribed by: Saksham Rathi

Checked and compiled by:

Disclaimer. Please note this document has not received the usual scrutiny that formal publications enjoy. This may be distributed outside this class only with the permission of the instructor.

1 Minimum States in a DFA

So, far we have dealt with DFA, NFA without ϵ , NFA with ϵ and Regular Expressions. We have also seen that all of them are equivalent and inter-convertible and represent regular languages. Consider the language which consists of all strings which are terminated by one. The regular expression for this will be: $(0 + 1)^*1$. Here is a 2-state DFA for the same language:

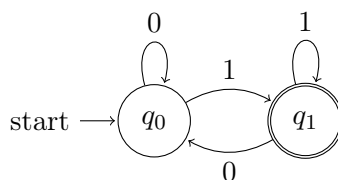


Figure 1: 2-state DFA for the language

The above DFA has two states q_0 and q_1 . q_0 is reached when the last seen letter was 0 (also at the start). q_1 is reached when the last seen letter was 1 (also, it is an accepting state). We can even construct a 4-state DFA for the same language. Here is an example:

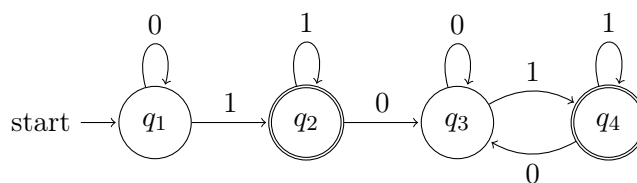


Figure 2: 4-state DFA for the language

The above DFA has four states q_1, q_2, q_3, q_4 .

- q_1 : Last letter 0 and no 1s so far
- q_2 : Last letter 1 and no 101 seen so far
- q_3 : Last letter 0 and more than one 1s seen so far
- q_4 : Last letter 1

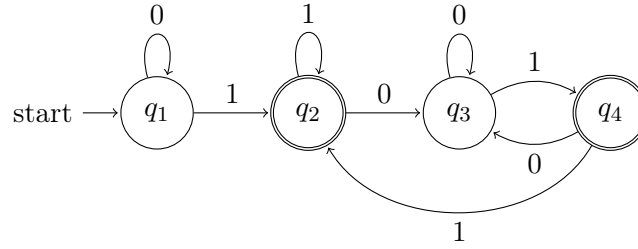


Figure 3: Another 4-state DFA for the same language

One can construct many more 4-state DFAs for the same language. Above is another example.

A natural question which comes to our mind is that can we construct another 2-state DFA for this language. One can find through trial and error, that this is not possible.

Is one state DFA possible for this language? Let us assume that this is possible. Two cases arise. If that state is accepting, then it will also accept ϵ , which is not possible. If that state is not accepting, then the language will be empty. We have arrived at a contradiction. Hence, one state DFA is not possible for this language.

Therefore, for this language, the minimum states in any DFA can be 2. We also observe that number of such 2-state DFAs is 1. So, we will now claim that for every language, the number of minimal DFAs is 1 and try to prove this. Before we prove this, we will introduce the notion of indistinguishability.

2 Indistinguishability

Two states of a DFA q_i and q_j are considered indistinguishable, if $\forall w \in \Sigma^*$, we start with q_i , process w and reach q'_i and we start with q_j , process w and reach q'_j , then either $q'_i \in F$ and $q'_j \in F$ or $q'_i \notin F$ and $q'_j \notin F$, where F is the set of all final states of the DFA. So, we are basically changing the start states and checking whether we reach the same type of states or not through the same string.

This relation is denoted by \equiv . It has the following properties:

- It is **reflexive**. It is clear to see that every state is indistinguishable to itself, as it will reach a particular state on seeing w . (Due to the nature of a DFA)
- Also, it is clear to see that this relation is **symmetric**.
- This relation is also **transitive**. We can prove this by contradiction. Let us assume that $(q_i \equiv q_j) \wedge (q_j \equiv q_k)$ but $q_i \not\equiv q_k$. Then $\exists w$ such that $q'_i \in F$ and $q'_k \notin F$, where q'_i and q'_k are the states we reach from q_i and q_k respectively on seeing w . From the equivalence of q_i and q_j , we have $q'_j \in F$ but from the equivalence of q_j and q_k , we have $q'_j \notin F$, where q'_j is the state that we reach from q_j on seeing w . Hence, we have arrived at a contradiction. Therefore, this relation is transitive.
- A relation which is reflexive, symmetric and transitive, is **equivalent**. Thus the states of the DFA, on which this relation is defined can be partitioned into equivalence classes.

In the above example (Figure: 2), q_1 and q_3 belong to the same equivalence class, and q_2 and q_4 also belong to another equivalence class. Let us now try to construct a 2-state DFA from the above 4-state DFA example (Figure: 2). We will choose, one element each from both of the equivalence classes. Let us take q_1 from the first class and q_4 from the second class.

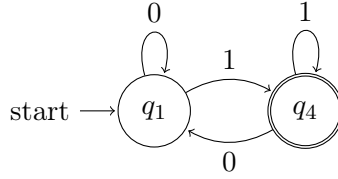


Figure 4: 2-state DFA constructed from the 4-state DFA

From q_1 , if we see a 0, we land at q_1 itself. If we see a 1, we would have landed at q_2 , but since q_2 and q_4 are equivalent, we replace q_2 by q_4 . Similarly, from q_4 , if we see a 1, we remain at q_4 , but if we see a 0, we would have landed at q_3 , but since q_1 and q_3 are equivalent, we replace q_3 by q_1 . Also, q_2 and q_4 both were acceptable earlier, now q_4 is acceptable.

So, through these equivalence classes, we have minimized our 4-state DFA into a 2-state DFA, and also this 2-state DFA is structurally the same as the previous one (Figure: 1), thus again making us think that the claim that there is a single minimal DFA for every language might be correct.

Another interesting thing we observe is that an accepting state cannot be indistinguishable from a non-accepting state. We can take $w = \epsilon$, and observe that the states we reach from this pair of states do not satisfy the definition of indistinguishability relation. However, an initial state and a non-initial state can belong to the same equivalence class. (For example, above q_1 and q_3 belonged to the same class.)

Now, several important questions arise. How can we find the equivalence classes of this relation? When we will come to know that we cannot compress our DFA further (by compress, we mean reducing the number of states of the DFA)? How can we prove our claim that the minimal DFA will be unique?

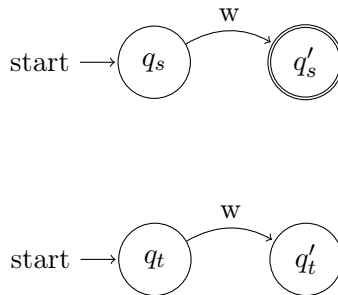
We will try to answer all these questions subsequently. Let us start with the easiest one.

3 Equivalence classes of Indistinguishability relation

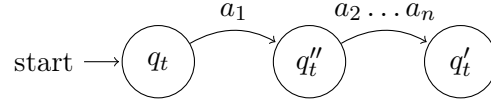
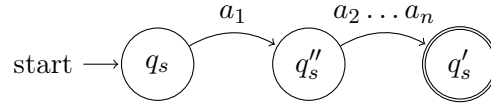
Now, we will develop an algorithm to find the equivalence classes of this relation.

Firstly, we should keep in mind our previous observation that an accepting state cannot be indistinguishable from a non-accepting state. That is $q_i \not\equiv q_j \forall q_i \in F$ and $q_j \in (Q \setminus F)$, where Q is the set of all states of the DFA.

Suppose, we find two states q_s and q_t which are distinguishable. Thus there exists a string w such that q_s leads to an accepting state but q_t leads to a non-accepting state.



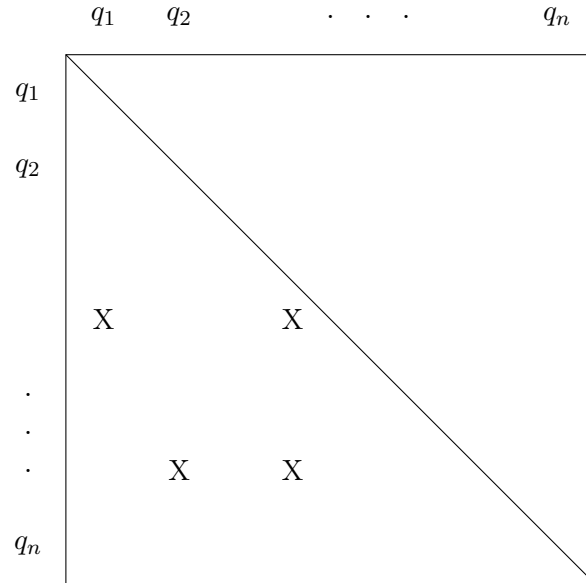
Now, w can be decomposed into $a_1 a_2 \dots a_n$, where $|w| = n$.



From, the above figures, one can observe that q''_s and q''_t are also distinguishable, where $w' = a_2 \dots a_n$ is the string which is making them distinguishable.

Thus, we can observe that for all states q_i and q_j such that $q_i \not\equiv q_j$ and for all $a \in \Sigma$, such that q_s on seeing a lands at q_i and q_t on seeing a lands at q_j then q_s will be distinguishable with q_t , where q_s and q_t are two states of the DFA. (We are basically extending $w' = a + w$.)

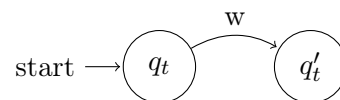
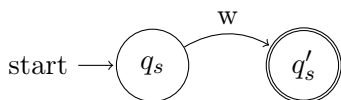
Therefore using a distinguishable pair, we have found another one. This is the basis of our algorithm. We initialize our set with all the pairs, where one state belongs to the set of accepting states and other state does not belong to the set of non accepting states. And then through the above step, we keep on increasing the size of this set. (Note that this algorithm is not exponential, because there are only $\binom{n}{2}$ pairs possible, and we do need to check an already visited pair.)



We stop the process, when no more crosses can be inserted.

Now, it is quite natural for us to ask the question that whether our algorithm is correct or not i.e. can we still find a pair of distinguishable states, which are not detected even after our algorithm finishes?

Proof: Let us assume that there are two states q_s and q_t which are distinguishable but not recognized by our algorithm. By definition, there exists a string w such that q_s leads to an accepting state on seeing w and q_t reaches a non-accepting state.



Now w can be written as $a_1 a_2 \dots a_n$, where $|w| = n$. And also assume that we reach q_s'' and q_t'' on seeing a_1 from q_s and q_t respectively. It is clear that, if our algorithm has not detected q_s and q_t as distinguishable, then it would not even have detected q_s'' and q_t'' as a distinguishable pair. (Because, if it would have done, then the next step would have been to make q_s and q_t as distinguishable.)



Now, we will inductively move forward our algorithm. Thus q_s''' and q_t''' would not also be detected as distinguishable after our algorithm finishes. But clearly, this is not possible, because our algorithm initializes the set of pairs of {accepting, non-accepting} states and then it's first step is to move backwards. So, it would have marked q_s''' and q_t''' as distinguishable in the first step itself.



We have achieved contradiction. Therefore, we can safely conclude that our algorithm terminates and is also correct.

Now, once our algorithm detects all pairs of distinguishable states, we can choose equivalence classes of the indistinguishability relation. (Then, we can choose one representative from each class and move forward with our proof of existence of a unique minimal DFA.)

It is worth noting that distinguishability is not an equivalent relation. It is not even reflexive. It is also not transitive. But, it is indeed symmetric. And also, it proved out to be very useful for finding these equivalence classes.

Now, we will move forward with the remaining two questions, that can we compress a given DFA further and does there exist any other minimal DFA?