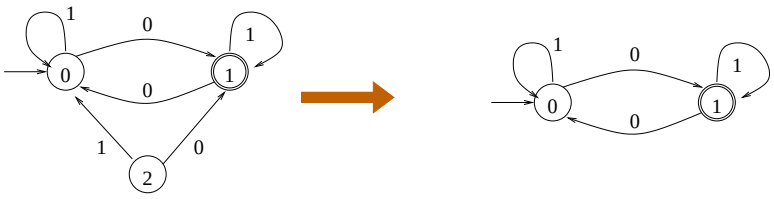
1 Minimization of Finite Automata

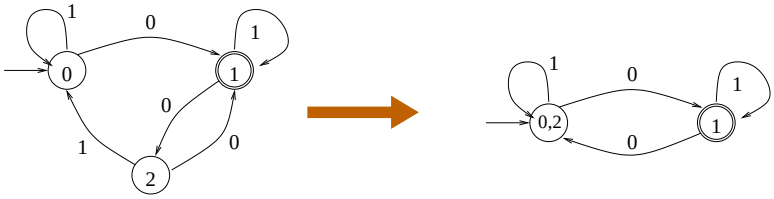
We now consider the following problem: for a given DFA A, find an equivalent DFA with a minimum number of states.

We start with two examples. Consider the automaton on the left:



You can see that it is not possible to ever visit state 2. States like this are called *unreachable*. We can simply remove them from the automaton without changing its behavior. (This will be, indeed, the first step in our minimization algorithm.) In our case, after removing state 2, we get the automaton on the right.

As it turns out, however, removing unreachable states is not sufficient. The next example is a bit more subtle (see the automaton on the left):



Let us compare what happens if we start processing some string w from state 0 or from state 2. I claim that the result will be always the same. (For now I didn't specify what exactly I mean by "the result". It will become clear soon.) This can be seen as follows. If $w = \lambda$, we will stay in either of the two states and λ will not be accepted. If w starts with 0, in both cases we will go to state 1 and both computations will be now identical. Similarly, if w starts with 1, in both cases we will go to state 0 and both computations will be identical. So no matter what w is, either we accept w in both cases or we reject w in both cases. Intuitively, from the point of view of the "future", it does not matter whether we start from state 0 or state 2. Therefore the transitions into state 0 can be redirected into state 2 and vice versa, without changing the outcome of the computation. Alternatively, we can combine states 0 and 2 into one state.

We now formalize the above idea. Let A be a finite automaton. We will say that $w \in \Sigma^*$ *distinguishes* between two states $p, q \in Q$ if either $\delta(p, w) \in F$ & $\delta(q, w) \notin F$, or $\delta(p, w) \notin F$ & $\delta(q, w) \in F$. Two states $p, q \in Q$ are called *distinguishable* iff there is a word that distinguishes between them. States that are indistinguishable will also be sometimes called *equivalent*.

The reasoning above leads to the following method: Start with an automaton A without unreachable states. If A has distinguishable states p, q, combine them into one state. (For instance, remove q and reroute all transitions into q to go into p instead.) Repeat this process until no more distinguishable states can be found. At this point we will not be able to reduce A further. But does it necessarily mean that A is minimum? Conceivably, there could be a completely different automaton B, with a completely different topology but with fewer states, that accepts the same language as A. Quite remarkably, it turns out that this is impossible, namely, any automaton whose all states are pairwise distinguishable must be minimum.

Lemma 1 *Suppose that B is a DFA without unreachable states. Then B is minimum if and only if all pairs of states are distinguishable.*

Proof: ($\Rightarrow$) This implication is easy, for if B has two undistinguishable states, one of them can be eliminated, and the transitions into this state can be changed to go to the other. This will not affect the accepted language. (Formal proof omitted.)

($\Leftarrow$) This implication is more interesting. Assume that in B all states are pairwise distinguishable. Let B have k states. Consider any other $\tilde{B}$ with $l < k$ states. We need to prove that $L(\tilde{B}) \neq L(B)$.

For each state q of B, choose arbitrarily one string w_q such that $\delta^*(q_0, w_q) = q$. That such w_q exists for each q follows from the assumption that all states are reachable. Since $l < k$, there are two states $p \neq q$ of B such that in $\tilde{B}$ we have $\tilde{\delta}^*(\tilde{q}_0, w_p) = \tilde{\delta}^*(\tilde{q}_0, w_q)$. Since p, q are distinguishable in B, there is a string v such that $\delta^*(q, w) \in F$ but $\delta^*(p, w) \notin F$ (or vice versa, but if so, we can always swap p, q.) This means that B accepts $w_p v$ but not $w_q v$. But in $\tilde{B}$, we have $\tilde{\delta}^*(\tilde{q}_0, w_p v) = \tilde{\delta}^*(\tilde{q}_0, w_q v)$, so $\tilde{B}$ either accepts both $w_p v$, $w_q v$ or rejects both. Therefore $L(\tilde{B}) \neq L(B)$, as claimed. $\diamond$

Before we get to the algorithm, we need one more thing. The idea discussed earlier was to repeatedly combine undistinguishable states. But suppose that p, q are indistinguishable, and q and r are indistinguishable. We want to combine p with q, and we also want to combine q with r. Then, it better be that p and r are also indistinguishable. Luckily, they are. This property is formalized in the lemma below.

Lemma 2 *State indistinguishability is an equivalence relation.*

Proof: Write $p \equiv q$ if p is indistinguishable from q. We need to check the following three conditions:

Symmetry: $p \equiv p$.

Reflexivity: $p \equiv q$ implies $q \equiv p$.

Transitivity: $p \equiv q$ and $q \equiv r$ implies $p \equiv r$.

All conditions are trivially true, directly from the definition of the indistinguishability relation. $\diamond$

Lemma 3 *Let $\delta(p, a) = p'$ and $\delta(q, a) = q'$. Then, if p', q' are distinguishable then so are p, q.*

Proof: This is quite simple, for if p' and q' are distinguished by some string w, then p, q are distinguished by string aw. $\diamond$

To minimize A, after removing unreachable states, we will find all equivalence classes of the indistinguishability relation and join all states in each class into one state of the new automaton $\hat{A}$. To determine transitions, if X is an equivalence class, we pick any $q \in X$ and define $\hat{\delta}(X, a) = Y$ where Y is the equivalence class that contains $\delta(q, a)$.

To determine which states are equivalent, we will use the following method. Instead of trying to figure out which states are indistinguishable, we will try to figure out which states are *distinguishable*. Clearly, if $p \in F$ and $q \notin F$ then p, q are distinguishable, since they can be distinguished by λ. Then we iteratively examine all pairs of states. For each pair p, q we do this. If we find a symbol a, such that $\delta(p, a)$ and $\delta(q, a)$ are distinguishable, then we mark p, q as distinguishable as well. We iterate this until no more pairs are marked. When we are done, the non-marked pairs are equivalent. The complete algorithm is given below.

Minimization Algorithm.

Step 1: Remove unreachable states.

Step 2: Mark the distinguishable pairs of states.

To chieve this task, we first mark all pairs p, q, where $p \in F$ and $q \notin F$ as distinguishable. Then, we proceed as follows:

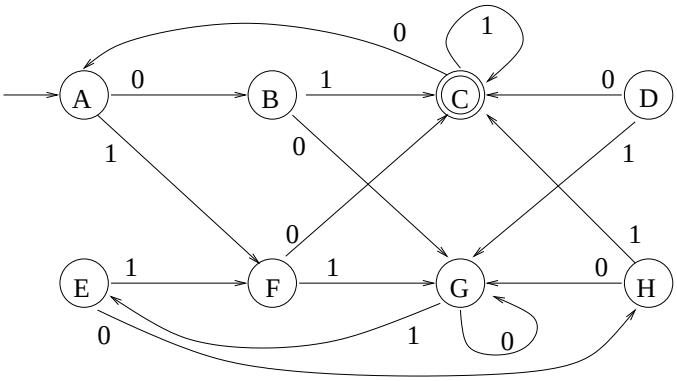
repeat
  for all non-marked pairs p, q do
    for each letter a do
      if the pair $\delta(p, a)$,$\delta(q, a)$ is marked
      then mark p, q
until no new pairs are marked

Step 3: Construct the reduced automaton $\hat{A}$.

We first determine the equivalence classes of the indistinguishability relation. For each state q, the equivalence class of q consists of all states p for which the pair p, q is not marked in Step 2.

The states of $\hat{A}$ are the equivalence classes. The initial state $\hat{q}_0$ is this equivalence class that contains q_0. The final states $\hat{F}$ are these equivalence classes that consist of final states of A. The transition function $\hat{\delta}$ is defined as follows. To determine $\hat{\delta}(X, a)$, for some equivalence class X, pick any $q \in X$, and set $\hat{\delta}(X, a) = Y$, where Y is the equivalence class that contains $\delta(q, a)$.

Example: We apply our algorithm to the automaton given below:



Step 1: We have one unreachable state, state D. We remove this state and proceed to Step 2.

Step 2: We first mark pairs of final and non-final states, getting the following tableau:

B						
C	X	X				
E			X			
F			X			
G			X			
H			X			
	A	B	C	E	F	G

In the first iteration we examine all unmarked pairs. For example, for pair A, B, we get $\delta(A, 1) = F$ and $\delta(B, 1) = C$, and the pair C, F is marked, so we mark A, B too. After doing it for all pairs, we get the following tableau.

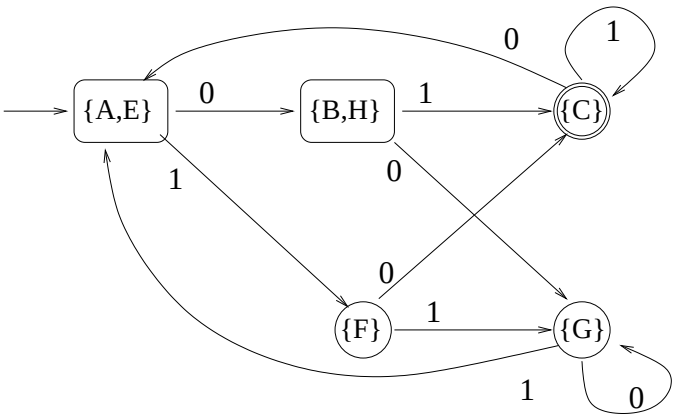
B	X					
C	X	X				
E		X	X			
F	X	X	X	X		
G		X	X		X	
H	X		X	X	X	X
	A	B	C	E	F	G

In the next iteration, we examine the remaining pairs. For example, we will mark pair A, G because $\delta(A, 0) = B$ and $\delta(G, 0) = G$, and the pair B, G is marked. When we're done we get the following tableau:

B	X					
C	X	X				
E		X	X			
F	X	X	X	X		
G	X	X	X	X	X	
H	X		X	X	X	X
	A	B	C	E	F	G

One more iteration will be executed now, but no new distinguishable pairs will be discovered. So we are done with Step 2 now.

Step 3: We group the states into the equivalence classes. Since A, E are equivalent and B, H are equivalent, the classes are: $\{A, E\}$, $\{B, H\}$, $\{C\}$, $\{F\}$, $\{G\}$. The minimal automaton $\hat{A}$ is:



Correctness. To justify correctness, we need to prove a couple of things. First, we need to show that we compute the equivalence classes correctly and that $\hat{A}$ is a well-defined DFA. Then, we need to show that it accepts the same language as A. Finally, we need to show that there is no DFA A' equivalent to A with fewer states.

Theorem 1 *Our minimization algorithm is correct, that is $L(\hat{A}) = L(A)$ and $\hat{A}$ is minimum.*

Proof: We prove all the required conditions, one by one.

Why are the equivalence classes computed correctly? Here, we need to show that the pair p, q is marked iff p, q are distinguishable. The proof of this is quite easy, by induction on the length of the shortest string that distinguishes p, q, using Lemma 3.

Why is $\hat{A}$ well defined? That follows from Lemma 2. The fact that indistinguishability is an equivalence relation implies that the states of $\hat{A}$ are well-defined. We also have that in each equivalence class either all

states are final or all states are non-final. The second condition implies that the transitions are well-defined. More specifically, if X is any equivalence class, then for all $p \in X$ and for all letters a, all states $\delta(p, a)$ are in the same equivalence class Y.

Why $L(\hat{A}) = L(A)$? To prove this, we show that for each w we have $\delta(q_0, w) \in \hat{\delta}(\hat{q}_0, w)$. (This can be proved by induction on the length of w.) Thus, by the definition of the final states of $\hat{A}$, $\hat{A}$ accepts w iff A accepts w.

Why is $\hat{A}$ minimal? This is where Lemma 1 is helpful, for we only need to show that in $\hat{A}$ all states are distinguishable. This is quite obvious from the construction. $\diamond$