Minimal DFAs

A DFA M is *minimal* if there is no DFA N equivalent to M with fewer states.

We will assume a fixed alphabet Σ throughout.

For convenience, we make the following definition: suppose $N = \langle Q, \Sigma, \delta, q_0, F\rangle$ is a DFA. For any state $q \in Q$ and string w over Σ, define $\hat{\delta}(q, w)$ to be the unique state resulting from starting in state q and reading w on the input. That is, $\hat{\delta}$ extends δ to strings of any length.

More formally, we can define $\hat{\delta} : Q \times \Sigma^* \to Q$ inductively:

Base case: $\hat{\delta}(q, \varepsilon) = q$.

Inductive case: $\hat{\delta}(q, aw) = \hat{\delta}(\delta(q, a), w)$ for $a \in \Sigma$ and $w \in \Sigma^*$.

It is clear by the definition that for any strings x and y and state q, $\hat{\delta}(q, xy) = \hat{\delta}(\hat{\delta}(q, x), y)$. Note also that a string w is in $L(N)$ iff $\hat{\delta}(q_0, w) \in F$ (i.e., $\hat{\delta}(q_0, w)$ is an accepting state).

Let L be any language (not necessarily regular) over Σ. For every string w define $L_w = \{x \mid wx \in L\}$. Note that $L_\varepsilon = L$, and if $L_w = L_{w'}$ then $L_{wa} = L_{w'a}$ for any $a \in \Sigma$, because $x \in L_{wa} \iff wax \in L \iff ax \in L_w \iff ax \in L_{w'} \iff w'ax \in L \iff x \in L_{w'a}$. Define

$$\mathcal{C}_L = \{L_w \mid w \in \Sigma^*\}.$$

Theorem 1 (Myhill, Nerode) *Let L be any language. L is regular if and only if $\mathcal{C}_L$ is finite, and in this case, there is a (unique) minimal DFA recognizing L with $|\mathcal{C}_L|$ many states.*

We'll prove Theorem 1 via three lemmata.

The first lemma shows that $|\mathcal{C}_L|$ is a lower bound on the number of states of any DFA recognizing a language L.

Lemma 1 *If a language L is regular, recognized by some DFA N with state set Q, then $|\mathcal{C}_L| \leq |Q|$.*

Proof. Suppose $N = \langle Q, \Sigma, \delta, q_0, F\rangle$. For any state $p \in Q$, define N_p to be the DFA that is the same as N but whose start state is p. That is, $N_p = \langle Q, \Sigma, \delta, p, F\rangle$. (So, for example, $N = N_{q_0}$.) Now let w be any string over Σ, and let $q = \hat{\delta}(q_0, w)$. We claim that

$$L_w = L(N_q).$$

To see this, note that, for any string $x \in \Sigma^*$, we have

$$\begin{aligned}
x \in L_w &\iff wx \in L \ \text{(by definition of } L_w\text{)} \\
&\iff wx \in L(N) \ \text{(since } L = L(N)\text{)} \\
&\iff \hat{\delta}(q_0, wx) \in F \ \text{(definition of acceptance)} \\
&\iff \hat{\delta}(\hat{\delta}(q_0, w), x) \in F \ \text{(since this is the same state)} \\
&\iff \hat{\delta}(q, x) \in F \ \text{(definition of } q\text{)} \\
&\iff x \in L(N_q).
\end{aligned}$$

So, with N fixed, we see that L_w only depends on the state obtained by starting in q_0 and reading w, and thus the number of distinct L_w is no more than the number of states of N. □

The next lemma gives $|\mathcal{C}_L|$ (if $\mathcal{C}_L$ is finite) as an upper bound on the number of states necessary for a DFA to recognize L.

Lemma 2 *Let L be a language. If $\mathcal{C}_L$ is finite, then L is regular, recognized by a DFA M_L with exactly $|\mathcal{C}_L|$ many states.*

Proof. We construct M_L having state set $\mathcal{C}_L$. We let $M_L = \langle \mathcal{C}_L, \Sigma, \delta, q_0, F \rangle$ where

- $q_0 = L_\varepsilon = L$,
- $\delta(L_w, a) = L_{wa}$ (this is well-defined!), and
- $F = \{q \in \mathcal{C}_L \mid \varepsilon \in q\}$.

Note that if we start in the start state q_0 of M_L and read a string w, then we wind up in state L_w; that is, $\hat{\delta}(q_0, w) = L_w$. You can see this by induction on $|w|$. The base case is when $|w| = 0$, that is, when $w = \varepsilon$. In this case, reading w keeps us in q_0. But $q_0 = L_\varepsilon = L_w$ by definition, so we wind up in state L_w in this case. Now suppse $|w| > 0$ and so $w = ya$ for some string y and alphabet symbol a. By the inductive hypothesis, reading y lands us in state L_y, but then further reading a moves us to state $\delta(L_y, a) = L_{ya} = L_w$.

Intuitively, the state L_w coincides with the set of all strings x such that starting in this state and reading x would lead us to accept wx. This is why we make L_w itself to be an accepting state when we do: if reading ε should make us accept w. Immediately after this proof, we give a concrete example of an M_L.

We must show that $L = L(M_L)$. For any $x \in \Sigma^*$ we have (below, q_0, $\hat{\delta}$, and F are all with respect to M_L):

$$\begin{aligned} x \in L &\iff \varepsilon \in L_x \ \text{(by definition of } L_x\text{)} \\ &\iff L_x \in F \ \text{(by definition of } F\text{)} \\ &\iff \hat{\delta}(q_0, x) \in F \ \text{(since } L_x = \hat{\delta}(q_0, x) \text{ from above)} \\ &\iff x \in L(M_L) \ \text{(by definition of acceptance).} \end{aligned}$$

Thus M_L recognizes L. □

Example. Let $\Sigma = \{a, b\}$ and let L be the set of all strings that contain three consecutive b's somewhere in the string. So L is denoted by the regular expression $(a \cup b)^* bbb (a \cup b)^*$. With a little thought, we can see that $|\mathcal{C}_L| = 4$. In fact, $\mathcal{C}_L = \{L_\varepsilon, L_b, L_{bb}, L_{bbb}\}$. These four languages are all distinct:

- $L_\varepsilon = L$.

- L_b contains bb (which is not in L_ε), but does not contain b.
- L_{bb} contains b (which is neither in L_ε nor in L_b), but does not contain ε.
- L_{bbb} contains all strings over $\{a, b\}$, including ε.

Furthermore, any L_w is equal to one of these four. For example, $L_{abaabb} = L_{bb}$ and $L_{bbbaaaaa} = L_{bbb}$. (For any w, how do you tell which of the four L_w is?) We thus get the following DFA M_L for L, as constructed in the proof of Lemma 2:

- The state set is $\mathcal{C}_L$.
- The start state is L_ε.
- The sole accepting state is L_{bbb}, as this is the only one of the four languages that contains ε.
- The transition function δ is as follows:

$$\begin{aligned}
\delta(L_\varepsilon, a) &= L_a = L_\varepsilon, \\
\delta(L_\varepsilon, b) &= L_b, \\
\delta(L_b, a) &= L_{ba} = L_\varepsilon, \\
\delta(L_b, b) &= L_{bb}, \\
\delta(L_{bb}, a) &= L_{bba} = L_\varepsilon, \\
\delta(L_{bb}, b) &= L_{bbb}, \\
\delta(L_{bbb}, a) &= L_{bbba} = \{a, b\}^* = L_{bbb}, \\
\delta(L_{bbb}, b) &= L_{bbbb} = L_{bbb}.
\end{aligned}$$

□

The third and final lemma proves that M_L is unique, thus completing the proof of Theorem 1. First, we will say that a DFA $N = \langle Q, \Sigma, \delta, q_0, F \rangle$ is *economical* if every state of N is reachable from the start state, that is, for every $q \in Q$ there is a string w such that $q = \hat{\delta}(q_0, w)$. Clearly, a DFA N is equivalent to an economical DFA with the same number or fewer states—just remove any states of N that are unreachable from q_0. These states will never be visited in a computation, so they are completely irrelevant.

Lemma 3 *Let L be a regular language. The DFA M_L of Lemma 2 is the unique (up to renaming of states) minimal DFA recognizing L.*

Proof. Let $N = \langle Q, \Sigma, \delta, q_0, F \rangle$ be any economical DFA recognizing L. We can map the states of N surjectively onto the states of M_L in a way that preserves the transition function, start state, and accepting states. For every $q \in Q$, define $f(q) = L(N_q)$, where N_q is as in the proof of Lemma 1. This defines a mapping f with domain Q, mapping states to languages. We will verify the following properties of f:

1. The range of f is exactly $\mathcal{C}_L$, that is, f maps Q onto $\mathcal{C}_L$ (the state set of M_L).

2. $f(q_0) = L_\varepsilon = L$ (the start state of M_L).

3. Let $q \in Q$ be a state, and let w be any string such that $f(q) = L_w$ (such a w exists by item (1)). For any $a \in \Sigma$, we have $f(\delta(q, a)) = L_{wa}$ (which is the result of the transition function of M_L being applied to the state $L_w = f(q)$ and a).

4. For any $q \in Q$, we have $q \in F$ if and only if $\varepsilon \in f(q)$ (that is, if and only if $f(q)$ is an accepting state of M_L).

For (1): First, let q be any state in Q. Since N is economical, there is some string w such that $q = \hat{\delta}(q_0, w)$. In the proof of Lemma 1 we showed that $L_w = L(N_q)$. Since $L(N_q) = f(q)$ by definition, this shows that $f(q) \in \mathcal{C}_L$, and thus the range of f is a subset of $\mathcal{C}_L$. Lastly, we must show that every element of $\mathcal{C}_L$ is equal to $f(q)$ for some $q \in Q$. Given any string w, we define $q = \hat{\delta}(q_0, w)$. Then again by the proof of Lemma 1, we have $L_w = L(N_q)$, which also equals $f(q)$. Thus every L_w is in the range of f, and so f maps Q surjectively onto $\mathcal{C}_L$.

For (2): Clearly, $f(q_0) = L(N_{q_0}) = L(N) = L = L_\varepsilon$.

For (3): Let $r = \delta(q, a)$. For any string x, we have

$$\begin{aligned} x \in f(r) &\iff x \in L(N_r) \\ &\iff \hat{\delta}(r, x) \in F \\ &\iff \hat{\delta}(q, ax) \in F \\ &\iff ax \in L(N_q) \\ &\iff ax \in f(q) \\ &\iff ax \in L_w \\ &\iff wax \in L \\ &\iff x \in L_{wa}. \end{aligned}$$

Therefore $f(r) = L_{wa}$.

For (4): Clearly, $q \in F$ if and only if $\varepsilon \in L(N_q)$, because $\hat{\delta}(q, \varepsilon) = q$.

Now we're ready to prove the lemma. Let $M_L = \langle \mathcal{C}_L, \Sigma, \delta', L_\varepsilon, F' \rangle$ be the minimal DFA for L constructed in the proof of Lemma 2. Suppose $N = \langle Q, \Sigma, \delta, q_0, F \rangle$ is *any* minimal DFA recognizing L. We show that N and M_L are the same DFA after relabelling states. Since N is minimal, it must also be economical (otherwise, we could remove at least one unreachable state to get a smaller equivalent DFA). Thus we have the surjective mapping $f : Q \to \mathcal{C}_L$ defined above. Also, by Lemmata 1 and 2 we must also have $|Q| = |\mathcal{C}_L|$ (since N is minimal). Thus f must be a bijection (perfect matching) between Q and $\mathcal{C}_L$. This is the relabelling we want: f maps the start state of N to the start state of M_L (by (2)); it maps accepting states to accepting states and nonaccepting states to nonaccepting states (by (4)); finally, it preserves the transition function, that is, for any $q \in Q$ and $a \in \Sigma$,

$$f(\delta(q, a)) = L_{wa} = \delta'(f(q), a)$$

by (3), where w is any string such that $f(q) = L_w$.

This shows that N and M_L are identical up to the relabelling f, which proves the lemma. □

Minimizing a DFA

Nice as they are, the results in the last section do not give an explicit general procedure to construct a minimal DFA equivalent to a given DFA. Often one can determine M_L by inspection, as we did in the Example, above. Nevertheless, we would like a general algorithm that, given a DFA $N = \langle Q, \Sigma, \delta, q_0, F \rangle$, constructs M_L, where $L = L(N)$.

The algorithm to do this works in two steps: (1) remove all states of N that are not reachable from q_0, making N economical, and (2) collapse remaining states together that we find are *equivalent*. We'll assume that we have already performed (1), so that N is economical.

For any state $q \in Q$, define N_q as in the proof of Lemma 1, above.

State equivalence and distinguishability

We say that two states p and q in Q are *distinguishable* if $L(N_p) \neq L(N_q)$, that is, there is some string x such that one of $\hat{\delta}(p, x)$ and $\hat{\delta}(q, x)$ is accepting and the other is not. Such a string x, if it exists, *distinguishes* p from q. (Such an x is in $L(N_p) \triangle L(N_q)$, the symmetric difference of $L(N_p)$ and $L(N_q)$.[1])

If p and q are indistinguishable (i.e., not distinguishable by any string, i.e., $L(N_p) = L(N_q)$), then we say p and q are *equivalent* and write $p \approx q$. This is obviously an equivalence relation. Intuitively, $p \approx q$ means that our acceptance or rejection of any string w does not depend on whether we start in state p or in state q before reading w. (Although it's not necessary for the current development, you may observe that $p \approx q$ iff p and q are mapped to the same state by the f defined in the proof of Lemma 3.)

The meat of the algorithm is to determine which pairs of states are equivalent. This is done using a table-filling technique. We methodically find *distinguishable* pairs of states; when we can't find any more, the pairs of states left over must be equivalent. For convenience, assume that $Q = \{1, \ldots, n\}$. We use a two-dimensional array T with entries $T[p, q]$ for all $1 \leq p, q \leq n$. We'll keep T symmetric in what follows, i.e., any value assigned to $T[p, q]$ will automatically also be assigned to $T[q, p]$ implicitly. (Also, we will have no use for diagonal entries $T[p, p]$, so actually, only the proper upper triangle of T need be stored: those entries $T[p, q]$ where $p < q$.) We proceed as follows:

Initially, all entries of T are blank;

[1] For any sets A and B, we define $A \triangle B = (A - B) \cup (B - A) = (A \cup B) - (A \cap B)$, i.e., the set of all z such that $z \in A$ or $z \in B$ but not both.

FOR each pair (p, q) of states with $p < q$, DO
    IF $p \in F$ or $q \in F$ but not both, THEN
        $T[p, q] \leftarrow$ X;
REPEAT
    FOR each pair (p, q) with $p < q$ and each $a \in \Sigma$, DO
        IF $T[p, q]$ is blank and $T[\delta(p, a), \delta(q, a)] =$ X, THEN
            $T[p, q] \leftarrow$ X
UNTIL no entries are marked X in one full iteration of this loop

When finished, it will be the case that an entry $T[p, q]$ is blank if and only if $p \approx q$.

To see the "if" part, notice that we only mark an entry with X if we have *evidence* that the two states in question are distinguishable. If $T[p, q]$ gets marked in the initial FOR-loop, it is because ε distinguishes p from q. For entries marked in the REPEAT-loop, we can proceed by induction on the number of steps taken by the algorithm when an entry is marked to show that the corresponding states are distinguishable: if $T[p, q]$ is marked in the REPEAT-loop, it is because $T[\delta(p, a), \delta(q, a)]$ was marked sometime previously, for some $a \in \Sigma$. By the inductive hypothesis, $\delta(p, a)$ and $\delta(q, a)$ are distinguished by some string w. But then, aw clearly distinguishes p from q, so marking $T[p, q]$ is correct. This proves the "if" part.

To show the "only if" part, we need to show that all distinguishable pairs of states are eventually marked with X. Suppose this is not the case. We call (p, q) a *bad pair* if $p \not\approx q$ but $T[p, q]$ is left blank by the algorithm. Our assumption is that there is at least one bad pair. Let w be a string distinguishing some bad pair (p, q), and assume that w is as short as any string distinguishing any bad pair. It must be that $w \neq \varepsilon$, since otherwise, ε distinguishes p from q, which in turn implies that either $p \in F$ or $q \in F$ but not both; but then $T[p, q]$ is marked X in the initial FOR-loop, so (p, q) is not a bad pair. So we must have $w = ay$ for some $a \in \Sigma$ and $y \in \Sigma^*$. Then, y distinquishes $r = \delta(p, a)$ from $s = \delta(q, a)$, and since $|y| < |w|$, (r, s) is not a bad pair (by the minimality of w). So $T[r, s]$ is marked with an X at some point. But then, on the first iteration of the REPEAT-loop following the marking of $T[r, s]$, we mark $T[p, q]$ with X, which contradicts the assumption that (p, q) is a bad pair. Thus there are no bad pairs.

Now that we can tell whether any two states are equivalent, we are ready to construct the minimum DFA M for L. The let the state set of M be the set $Q/\approx$ of equivalence classes of Q under the $\approx$-relation. For any $q \in Q$ we let $[q]$ denote the equivalence class containing q.

We define the transition function δ_M for M as follows: for any $q \in Q$ and $a \in \Sigma$, let

$$\delta_M([q], a) = [\delta(q, a)].$$

This is a legitimate definition, because $p \approx q$ clearly implies $\delta(p, a) \approx \delta(q, a)$, and so the transition does not depend on which representative of the equivalence class we use. Also note that when we extend δ_M to $\hat{\delta}_M$ acting on all strings, we can routinely check that

$$\hat{\delta}_M([q], w) = [\hat{\delta}(q, w)]$$

for all $q \in Q$ and $w \in \Sigma^*$.

We define the start state of M to be $[q_0]$, and the set of accepting states of M to be $F_M = \{[r] \mid r \in F\}$.

We must verify two things:

1. M recognizes L, and
2. M is minimal.

For (1), let w be any string.

$$\begin{aligned} w \in L &\iff \hat{\delta}(q_0, w) \in F \\ &\iff [\hat{\delta}(q_0, w)] \in F_M \\ &\iff \hat{\delta}_M([q_0], w) \in F_M \\ &\iff w \in L(M). \end{aligned}$$

Thus $L = L(M)$.

For (2), first note that since N is economical, so is M: if $[r]$ is any state of M, let w be such that $\hat{\delta}(q_0, w) = r$; then $\hat{\delta}_M([q_0], w) = [\hat{\delta}(q_0, w)] = [r]$, and thus $[r]$ is reachable from the start state $[q_0]$ via w. Now we show that $\mathcal{C}_L = \{L(M_{q'}) \mid q' \in Q/\!\approx\}$. By adapting the proof of Lemma 1 we get that $L_w = L(M_{q'})$ for any string w, where $q' = \hat{\delta}_M([q_0], w)$. But since M is economical, every state of M is of this form for some w, so indeed $\mathcal{C}_L = \{L(M_{q'}) \mid q' \in Q/\!\approx\}$. We'll be done if we can show that any two distinct states of M are distinguishable, for then $L(M_{p'}) \neq L(M_{q'})$ for all $p', q' \in Q/\!\approx$, and so it must be that $|Q/\!\approx| = |\{L(M_{q'}) \mid q' \in Q/\!\approx\}| = |\mathcal{C}_L|$, which implies that M is minimal by Lemmata 1 and 2.

To see that all distinct states of M are distinguishable, notice that if $[p] \neq [q]$, then $p \not\approx q$, and so p and q are distinguished (in N) by some string x. We claim that the same string x also distinguishes $[p]$ from $[q]$ in M. Let $r = \hat{\delta}(p, x)$ and let $s = \hat{\delta}(q, x)$. Then either $r \in F$ or $s \in F$ but not both. We also have $\hat{\delta}_M([p], x) = [r]$ and $\hat{\delta}_M([q], x) = [s]$ by definition, so we're done if either $[r]$ or $[s]$ is accepting (in M) but not both. Suppose, WLOG, that $r \in F$. Then $[r]$ is accepting in M by definition. Further, $s \notin F$. This implies that $[s]$ cannot be accepting in M: otherwise, there is some $s' \in [s] \cap F$, but then $s' \approx s$, and so $s' \notin F$ because $s \notin F$—contradiction. Thus $[s]$ is not accepting in M. (Any equivalence class in $Q/\!\approx$ either contains only accepting states or only nonaccepting states.) A similar argument works assuming $s \in F$. Thus $[p]$ and $[q]$ are distinguishable in M.