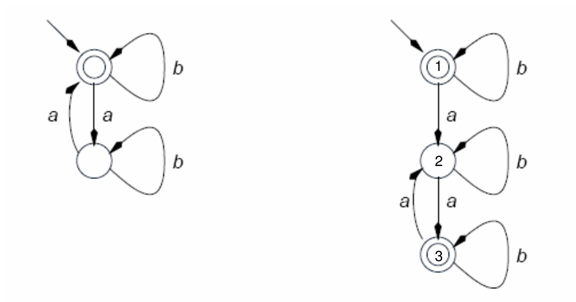
CS520: Lecture 6 handout
State Minimization in Finite Automata

February 12, 2008

DFAs accepting a certain language are not necessarily unique. For example, the two DFAs below encode the same language $L = \{w | w \text{ contains an even number of } a\text{'s}\}$. It is clear that the third state of the automaton M_2 is redundant and can be merged with its first state.



M_1 M_2

In this lecture we will discuss the following questions – Given a language L, what is the smallest (and therefore simplest) DFA recognizing L? Is the smallest such DFA unique? Given a DFA, can we determine which of its states are redundant? How do we eliminate those states?

Theorem 1 *For every regular language L, there exists a unique (up to re-labeling of the states) minimal DFA M, such that $L = L(M)$.*

The key to proving this theorem is to define an equivalence relation between states of a DFA. Intuitively, two states are equivalent if upon starting in one of those states and reading any string, the behavior of the DFA is identical to starting in the other state and reading the same string. In order to formalize this equivalence relation, we first extend the transition function so that its second argument can be a string and not just a symbol from the alphabet.

Definition 1 *Given a DFA $M = (Q, \Sigma, \delta, q_0, F)$, the extended transition function $\hat{\delta} : Q \times \Sigma^* \to Q$ is defined recursively as follows:*

$$\begin{aligned} \hat{\delta}(q, \epsilon) &= q \\ \hat{\delta}(q, a) &= \delta(q, a) && \forall a \in \Sigma \\ \hat{\delta}(q, aw) &= \hat{\delta}(\delta(q, a), w) && \forall a \in \Sigma, w \in \Sigma^* \end{aligned}$$

Intuitively $\hat{\delta}(q, w)$ tells us the state we end up in when we start at q and read the string w.

Definition 2 *A string $w \in \Sigma^*$ distinguishes states q_1 from q_2 if $\hat{\delta}(q_1, w) \in F$ but $\hat{\delta}(q_2, w) \notin F$, or vice versa. Two states q_1 and q_2 are indistinguishable (written $q_1 \sim q_2$), if no string in Σ^* distinguishes them, or in other words, for all $w \in \Sigma^*$, $\hat{\delta}(q_1, w) \in F$ if and only if $\hat{\delta}(q_2, w) \in F$.*

When a string w distinguishes states q_1 and q_2, w acts as witness to the fact that q_1 and q_2 behave differently and are not equivalent to each other. In the figure above, for example, the states 1 and 2 in M_2 are distinguishable because the string ab distinguishes them—starting in 2 we accept the string ab, ending at 3, but starting in 1 we reject it because we end up at 2. On the other hand, the states 1 and 3 are indistinguishable. There is no string that distinguishes them.
$\sim$ is an equivalence relation:

- $q \sim q$ (reflexive)
- $p \sim q \implies q \sim p$ (symmetric)
- $p \sim q$ and $q \sim r \implies p \sim r$ (transitive)

This means that given any DFA $M = (Q, \Sigma, \delta, q_0, F)$, we can partition its states into equivalence classes based on $\sim$. We use $[q]$ to denote the class containing the state q.

$$[q] = \{p | p \sim q\}$$

Now, let us define a new DFA M' that contains one state for every equivalence class in M.

$$\begin{aligned} M' &= (Q', \Sigma, \delta', q'_o, F') \\ Q' &= \{[q] | q \in Q\} \\ q'_0 &= [q_0] \\ F' &= \{[q] | q \in F\} \\ \delta'([q], a) &= [\delta(q, a)] \end{aligned}$$

We claim that the DFAs M and M' are equivalent. That is, $L(M) = L(M')$.

However, before we prove that, we need to argue that M' is properly defined. In particular, suppose that $[p] = [q]$ for two distinct states p and q in Q. Then, δ' must be defined such that $\delta'([p], a) = \delta'([q], a)$ for all symbols $a \in \Sigma$. In other words, we need to show that $[\delta(p, a)] = [\delta(q, a)]$, or $\delta(p, a) \sim \delta(q, a)$, whenever $p \sim q$.

We can prove this by contradiction. In particular, suppose that $p \sim q$ but $\delta(p, a) \not\sim \delta(q, a)$. Specifically, there is a string $w \in \Sigma^*$ that distinguishes $\delta(p, a)$ and $\delta(q, a)$. Without loss of generality, say that $\hat{\delta}(\delta(p, a), w) \in F$ but $\hat{\delta}(\delta(q, a), w) \notin F$. Then, by the definition of $\hat{\delta}$, we have $\hat{\delta}(p, aw) \in F$ but $\hat{\delta}(q, aw) \notin F$. This means that aw distinguishes p and q and contradicts our assumption that $p \sim q$. Therefore, δ' is properly defined.

Now let us argue that M and M' accept the same language. First, we can prove by induction that for any string $w \in \Sigma^*$ and state $q \in Q$, $\hat{\delta'}([q], w) = [\hat{\delta}(q, w)]$. The base case follows from the definition of δ'. The details are left as exercise to the reader.

The rest of the proof is straightforward. Suppose that M accepts a string $x \in \Sigma^*$. This means that $\hat{\delta}(q_0, x) \in F$. Then $\hat{\delta'}([q_0], x) = [\hat{\delta}(q_0, x)] \in F'$ by the definition of F', and M' accepts the string too. Next suppose that M' accepts the string x, that is, $\hat{\delta'}([q_0], x) \in F$. Let $\hat{\delta}(q_0, x) = p$. Then $[p] \in F'$. This means that there exists a state $q \sim p$ such that $q \in F$ (by the definition of F'). But then it must be the case that $p \in F$, otherwise the empty string ϵ would distinguish the states p and q. Therefore, $\hat{\delta}(q_0, x) \in F$ and M accepts the string.

We will now prove that the DFA M' is the unique smallest DFA recognizing the language $L = L(M)$.

A DFA is called minimal if every pair of distinct states in the DFA are distinguishable. It is straighforward to see that M' is minimal. We now claim that any two minimal DFAs recognizing the same language must be equivalent upto a renaming of the states (and in particular have the same number of states).

Suppose that $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ are two minimal DFAs with $L(M_1) = L(M_2)$. Consider a mapping f from the states of M_1 to those of M_2, defined as follows: (1) $f(q_1) = f(q_2)$; (2) if for some string $w \in \Sigma^*$, $\hat{\delta_1}(q_1, w) = p_1$ and $\hat{\delta_2}(q_2, w) = p_2$, then $f(p_1) = p_2$.

We first note that the mapping f is a function, that is, it maps every state $p_1 \in Q_1$ to a unique state in Q_2. Let us prove this by contradiction. Suppose that f maps p_1 to two distinct states p_2 and p_2' in Q_2. That is, by definition, there are two strings x and y such that $\hat{\delta_1}(q_1, x) = \hat{\delta_1}(q_1, y) = p_1$, $\hat{\delta_2}(q_2, x) = p_2$ and $\hat{\delta_2}(q_2, y) = p_2'$. Then, since M_2 is minimal, and therefore, p_2 and p_2' are distinguishable, there is a string z such that $\hat{\delta_2}(p_2, z) \in F_2$ and $\hat{\delta_2}(p_2', z) \notin F_2$. That is, $\hat{\delta_2}(q_2, xz) \in F_2$ and $\hat{\delta_2}(q_2, yz) \notin F_2$. Since M_1 and M_2 accept the same language, it must be the case that $\hat{\delta_1}(q_1, xz) \in F_1$ and $\hat{\delta_1}(q_1, yz) \notin F_1$. But $\hat{\delta_1}(q_1, x) = \hat{\delta_1}(q_1, y) = p_1$. Then the first statement implies that $\hat{\delta_1}(p_1, z) \in F_1$ and the second implies that $\hat{\delta_1}(p_1, z) \notin F_1$. We arrive at a contradiction.

Likewise we can prove that f cannot map two distinct states in Q_1 to the same state in Q_2. This means that f is a one-one mapping, and $|Q_1| = |Q_2|$. Furthermore, we can verify that $f(\delta_1(p_1, a)) = \delta_2(f(p_1), a)$ for all $p_1 \in Q_1$ and $a \in \Sigma$. Therefore, the two DFAs are equivalent.

This concludes the proof of Theorem 1 above.

In the next lecture we will give an algorithm for finding the equivalence classes under $\sim$ for any given DFA.