COSC 362: The State Minimization Problem

[Note: the procedure described in this handout is adapted from "Theory of Computing: A Gentle Introduction" by Kinber & Smith, 2001, Prentice-Hall]

Suppose M is DFA with alphabet Σ, state set Q, final state set F, and transition function δ.

Definition: two states $s, t \in Q$ are called "indistinguishable" if, for all strings in $w \in \Sigma^*$, the computation on w starting from state s ends in a final state iff the computation on w starting from state t ends in a final state. In other "words," we say s is indistinguishable from t if the condition $\delta^*(w,s) \in F \leftrightarrow \delta^*(w,t) \in F$. holds for all strings w.

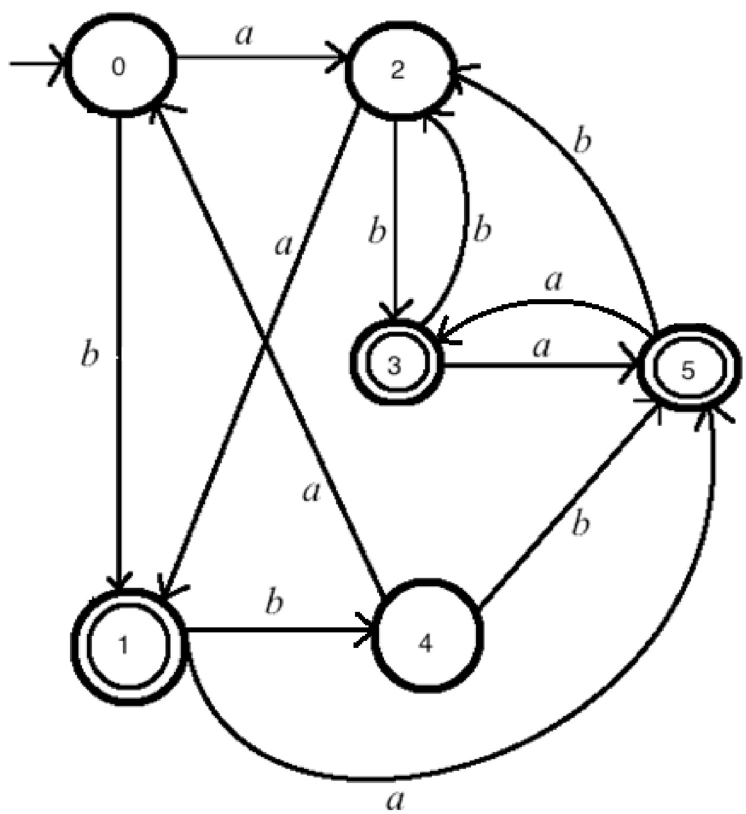
Notation: we write $s \equiv t$ to indicate that s and t are indistinguishable states of a DFA.

Definition: two states $s, t \in Q$ are said to be "indistinguishable by strings of length n or less" if the condition $\delta^*(w,s) \in F \leftrightarrow \delta^*(w,t) \in F$ holds for all strings w of length n or less.

Notation: we write $s \equiv_n t$ to indicate that s and t are indistinguishable by strings of length n or less.

For example: if states s and t are indistinguishable by the strings $\lambda, a, b, aa, ab, ba,$ and bb, then s and t would be "indistinguishable by strings of length 2 or less," so we could write $s \equiv_2 t$.

Example: Consider the DFA described by the diagram below:



We have $\Sigma = \{a, b\}, Q = \{0,1,2,3,4,5\}, F = \{1,3,5\}$, and δ according to the diagram.

For each value of n, we wish to find the equivalence classes of states relative to the relation $\equiv_n$. We do so inductively, starting with $\equiv_0$. Two states are indistinguishable by strings of length zero when both are final states or when both are non-final states. Therefore, our $\equiv_0$ equivalence classes are F ={1,3,5} and $F' = \{0,2,4\}$.

Now, when $n > 0$, the rule is that $s \equiv_n t$ iff both of the following are true:

(a) $s \equiv_{n-1} t$ and
(b) for all $x \in \Sigma$, $\delta(s, x) \equiv_{n-1} \delta(t, x)$

In other words – s,t are indistinguishable by strings of length n-1 or less; and, for each symbol in Σ, the states to with s and t are "mapped" are indistinguishable by strings of length n-1 or less.

We'll proceed by testing conditions (a) and (b) above for each pair of states in $\equiv_{n-1}$ in order to find the equivalence classes for $\equiv_n$. We'll know we're done when, for some value of n, the equivalence classes for $\equiv_n$ are the same as those for $\equiv_{n-1}$.

Let's see what all of this means by applying it to the example introduced on the preceding page. We start with $\equiv_0$ equivalence classes $F = \{1,3,5\}$ and $F' = \{0,2,4\}$. Now, here's how to find the $\equiv_1$ equivalence classes…

Start with F={1,3,5}. For each state in this set, let's investigate the computations on a and b:

$\delta(1, a) = 5 \in F$ $\quad\delta(3, a) = 5 \in F$ $\quad\delta(5, a) = 3 \in F$
$\delta(1, b) = 4 \in F'$ $\quad\delta(3, b) = 2 \in F'$ $\quad\delta(5, b) = 2 \in F'$

For each state q in F, $\delta(q, a) \in F$ and $\delta(q, b) \in F'$. Therefore, no pair of states in {1,3,5} is distinguishable by either a or b. As a result, {1,3,5} will be an equivalence class of $\equiv_1$.

Now, let's carry out the same analysis of {0,2,4}:

$\delta(0, a) = 2 \in F'$ $\quad\delta(2, a) = 1 \in F$ $\quad\delta(4, a) = 0 \in F'$
$\delta(0, b) = 1 \in F$ $\quad\delta(2, b) = 3 \in F$ $\quad\delta(4, b) = 5 \in F$

This is a bit more interesting. First, note that states 0 and 4 are not distinguishable by either a or b; for each, a maps to F', while b maps to F. However, state 2 does not follow the same rules – specifically, $\delta(2, a) \in F$, while $\delta(0, a)$ and $\delta(4, a) \in F'$. Therefore, state 2 is distinguishable from states 0 and 4 by the symbol a. The result of all of this is that the $\equiv_0$ equivalence class {0,2,4} gets broken up into two distinct $\equiv_1$ equivalence classes: {0,4} and {2}.

Summary: Our $\equiv_1$ equivalence classes are {1,3,5}, {0,4}, and {2}. For the sake of notation, let's call these sets F (as before), A_1, and B_1, respectively. (The subscript is to connect these sets to $\equiv_1$, but that's a cosmetic choice – call them whatever you want!)

Since the $\equiv_1$ classes are not all identical to the $\equiv_0$ classes, we must continue this analysis for at least one more step. On the next page, we'll proceed to compute the $\equiv_2$ equivalence classes from the $\equiv_1$ equivalence classes…

Computation of $\equiv_2$ equivalence classes from F={1,3,5}, $A_1 = \{0,4\}$, $B_1 = \{2\}$.

(Note: since B_1 is a singleton set, it is already "broken up" as much as possible; we certainly can't distinguish state 2 from itself! So, we'll focus on the other two sets…)

First, look at F={1,3,5}. For inputs a and b, we'll see which $\equiv_1$ equivalence class each of these states is mapped to. (NOTE: this is NOT the same as just tracking whether each state goes to a final or non-final state! That's an important distinction, as we will see shortly…)

$\delta(1,a) = 5 \in F$ $\quad$ $\delta(3,a) = 5 \in F$ $\quad$ $\delta(5,a) = 3 \in F$
$\delta(1,b) = 4 \in A_1$ $\quad$ $\delta(3,b) = 2 \in B_1$ $\quad$ $\delta(5,b) = 2 \in B_1$

Note that our results are not all the same this time. While states 3 and 5 are not distinguishable by either a or b, we see that state 1 is distinguishable from the others. In particular, for the symbol b, 1 is mapped to a state in A_1, while 3 and 5 are both mapped to a state in B_1. This implies that {1,3,5} is not a $\equiv_2$ equivalence class; rather, it must be broken up into two smaller classes, {1} and {3,5}. For the sake of notation, let's call these A_2 and B_2, respectively. (Again, you can name your equivalence classes whatever you like, as long as you keep track!) So, we have $\equiv_2$ equivalence classes $A_2 = \{1\}$ and $B_2 = \{3,5\}$.

> Another way of thinking about what we've just shown: since 1 is not in the same $\equiv_2$ equivalence class as 3 and 5, there must be some string of length 2 which distinguishes state 1 from states 3 and 5. If you run through the possibilities (there are only four such strings), you will find that the string ba is the culprit, since $\delta^*(1,ba) = 0 \notin F$ while $\delta^*(3,ba) = \delta^*(5,ba) = 1 \in F$.

Now, let's look at A_1={0,4}, to see whether it survives as a $\equiv_2$ equivalence class or needs to be broken up into smaller sets… again, we're looking at which $\equiv_1$ equivalence class each mapping takes us to:

$\delta(0,a) = 2 \in B_1$ $\quad$ $\delta(4,a) = 0 \in A_1$
$\delta(0,b) = 1 \in F$ $\quad$ $\delta(4,b) = 5 \in F$

Again, we see distinguishability here – specifically, states 0 and 4 are distinguished by the letter a. This means A_1 must be broken into two smaller $\equiv_2$ equivalence classes: {0} and {4}. Let's call thee C_2 and D_2, respectively. So, we now have $\equiv_2$ equivalence classes $B_1 = \{2\}, C_2 = \{0\}$, and $D_2 = \{4\}$.

> Again, what this is telling us is that states 0 and 4 are distinguished from one another by some string of length 2. One such example is the string aa, since $\delta^*(0,aa) = 1 \in F$, while $\delta^*(4,aa) = 2 \notin F$.

In summary, our $\equiv_2$ equivalence classes are

$$A_2 = \{1\}, B_2 = \{3,5\}, B_1 = \{2\}, C_2 = \{0\}, D_2 = \{4\}$$

All of these "singleton" classes tell us that states 1, 2, 0, and 4 are essential to the DFA's transition function; in other words, there is no redundancy among them. So, the only remaining question is whether states 3 and 5 are distinguishable. To find out for sure, we'll continue our procedure one more step…

Computation of $\equiv_3$ equivalence classes from the $\equiv_2$ classes:

$$A_2 = \{1\}, B_2 = \{3,5\}, B_1 = \{2\}, C_2 = \{0\}, D_2 = \{4\}$$

Again, the only remaining question is whether B_2 is a $\equiv_3$ equivalence class. If it is, then we'll be able to conclude that states 3 and 5 are indistinguishable in general. If not, then it will follow that the original DFA was minimal all along! Let's find out…

$\delta(3,a) = 5 \in B_2$ $\quad$ $\delta(5,a) = 3 \in B_2$
$\delta(3,b) = 2 \in B_1$ $\quad$ $\delta(5,b) = 2 \in B_1$

At this point, we are done – we can conclude that states 3 and 5 are indistinguishable for strings of any length.

But wait – how do we know we're done? That is, how do we know for sure there isn't some much longer string that distinguishes states 3 and 5? Well, let's imagine (but not actually carry out) the process of using $\equiv_3$ equivalence classes to calculate the $\equiv_4$ equivalence classes. If we were to continue in this manner, we'd end up doing exactly the same thing we just did above – that is, we'd find that states 3 and 5 are indistinguishable relative to the $\equiv_3$ equivalence classes, and are thus $\equiv_4$ equivalence classes. For strings of length 5, 6, and so on, the steps would be exactly the same; it would become an endless, repeating loop.

(What we've just done here is essentially the inductive step of a proof by induction – in general, if we find some number, k, such that every $\equiv_{k-1}$ equivalence class is also a $\equiv_k$ equivalence class, it *must* follow that they will remain equivalence classes for $\equiv_n$ for *all* higher values of n.)

Conclusion: the original DFA can be minimized by combining states 3 and 5 into one state, and leaving all of the other states as they are. Therefore, the minimized DFA is as follows: