CSE 322
Introduction to Formal Models in Computer Science

**Minimizing DFAs**

Given a DFA $M = (Q, \Sigma, \delta, s, F)$ such that $A = L(M)$ how do we find a DFA $M'$ of minimal size such that $A = L(M')$? We know that such a DFA has precisely one state for each equivalence class of $\equiv_A$ and that each equivalence class of $\equiv_A$ is a union of equivalence classes of $\equiv_M$. All that we would need to do is to figure out how to group the states of $M$ so that their corresponding equivalence classes lie in a single equivalence class of $\equiv_A$. In order to do this, rather than starting with individual states and clumping them together we will instead begin by grouping states together and separate them only when necessary. When we have finished we will have produced the minimal DFA for $A$.

We can represent the final grouping of the states of $M$ as an equivalence relation $\equiv_{min}$ on the states of $M$.

DEFINITION 0.1. *Define an equivalence relation* $\equiv_{min}$ *on the set of* states $Q$ *of* $M$ *by* $p \equiv_{min} q$ *if and only if for all strings* $z \in \Sigma^*$, $\delta^*(p, z) \in F \Leftrightarrow \delta^*(q, z) \in F$.

Now let's get some intuition for this definition. The following lemma says that if states $p$ and $q$ are $\equiv_i$ for all $i$, then the inputs that reach $p$ and $q$ should all go to the same state in a minimal DFA for $L(M)$ and otherwise those inputs should go to different states in that minimal DFA.

**Lemma 1.** *Let* $A = L(M)$. *Let* $p, q \in Q$ *and suppose that every state of* $M$ *is reachable from the start state* $s$. *Then* $p \equiv_{min} q$ *if and only if the equivalence classes of* $\equiv_M$ *corresponding to* $p$ *and* $q$ *lie in the same equivalence class of* $\equiv_A$.

*Proof.* Suppose that $p \equiv_{min} q$. Then for any $z \in \Sigma^*$, if $\delta^*(s, x) = p$ and $\delta^*(s, y) = q$ then by the properties of $\delta^*$ and the fact that $p \equiv_{min} q$, we know that $\delta^*(s, xz) \in F \Leftrightarrow \delta^*(s, yz) \in F$, but this means that $xz \in A \Leftrightarrow yz \in A$. Therefore any two strings that reach $p$ and $q$ are in the same equivalence class of $\equiv_A$.

Suppose that $p \not\equiv_{min} q$. Then there is some $z$ with such that only one of $\delta^*(p, z)$, $\delta^*(q, z)$ is in $F$. Since $p$ and $q$ are both reachable from $s$, there are strings $x$ and $y$ with $\delta^*(s, x) = p$ and $\delta^*(s, y) = q$. Therefore only one of $\delta^*(s, xz), \delta^*(yz)$ is in $F$ and so only one of $xz$, $yz$ is in $A$. This means that the classes of $\equiv_M$ corresponding to $p$ and $q$ are not in the same equivalence class of $\equiv_A$. $\square$

That's all very well but how can we figure out which states are equivalent to each other under $\equiv_{min}$?

Observe first that by considering $z = \varepsilon$ we know that if $p \equiv_{min} q$ and $p \in F$ then we must have $q \in F$. Therefore, whenever we have one state in $F$ and the other in $Q \setminus F$ then $(p, q)$ cannot be equivalent.

We will maintain a (growing) subset $B$ of "bad" pairs pairs of states of $M$ that we know cannot be equivalent to each other in $\equiv_{min}$. We can do this with table with $\binom{|Q|}{2}$ entries, each of which either has a blank or an X for each $(p, q)$ pair where the X denotes a member now known to be in $B$.

**Algorithm:** Initially, as described above, $(p, q)$ is put in $B$ if one is a final state and the other is not.

At each step of the algorithm if $(p, q)$ is in not in $B$ and there is some $a \in \Sigma$ such that the pair of states $(p', q')$ is in $B$ where $p' = \delta(p, a)$ and $q' = \delta(q, a)$ then we add $(p, q)$ to $B$.

**Start of Correctness:** Since the pair $(p', q')$ is in $B$ there is some $z \in \Sigma^*$ such that exactly one of $\delta^*(p', z)$ and $\delta^(q', z)$ is in $F$. Now $\delta^*(p, az) = \delta^*(\delta(p, a), z) = \delta^*(p', z)$ and $\delta^*(q, az) = \delta^*(\delta(q, a), z) = \delta^*(q', z)$ so exactly one of $\delta^*(p, az)$ and $\delta^*(q, az)$ is in $F$. Therefore $p \not\equiv_{min} q$ which justifies putting $(p, q)$ in $B$.

**Stopping:** Since there are only $\binom{|Q|}{2}$ possible pairs of states and the set $B$ always grows, this will terminate in at most $\binom{|Q|}{2} - 1$ steps.

**Lemma 2.** *Every pair $(p, q)$ such that $p \not\equiv_{min} q$ will be added to $B$ in the above minimization algorithm.*

*Proof.* We prove this by induction on the length of a string $z$ such that exactly one of $\delta^*(p, z)$ and $\delta^*(q, z)$ is in $F$. (Such a string must exist since $p \not\equiv_{min} q$.)

**Base Case:** $|z| = 0$. In this case $z = \varepsilon$. Therefore exactly one of $\delta^*(p, \varepsilon) = p$ and $\delta^*(q, \varepsilon) = q$ is in $F$. Then this pair was initially put in $B$.

**Induction Hypothesis:** Suppose this is true for all $(p', q')$ such that $p' \not\equiv_{min} q'$ and there is a string $z'$ of length $i$ such that exactly one of $\delta^*(p', z')$ and $\delta^*(q', z')$ is in $F$.

**Induction Step:** Now consider some pair $(p, q)$ such that $p \not\equiv_{min} q$ and there is a string $z$ of length $i+1$ such that exactly one of $\delta^*(p, z)$ and $\delta^*(q, z)$ is in $F$. Write $z = az'$ and let $p' = \delta(p, a)$ and $q' = \delta(q, a)$. Now by construction exactly one of $\delta^*(p', z')$ and $\delta^*(q', z')$ is in $F$ so by the inductive hypothesis the pair $(p', q')$ will eventually be added to $B$. Once $(p', q')$ is in $B$ if $(p, q)$ is not already in $B$ then the while loop of the algorithm will add the pair $(p, q)$ to $B$ since we have $p' = \delta(p, a)$ and $q' = \delta(q, a)$ and $(p', q')$ is in $B$.

Therefore the result follows by induction. $\square$

**Example:** Consider the DFA in Figure 1 on the next page.

We first remove state 4 because it isn't reachable.

Initially we include the pairs (1,3),(2,3),(3,5),(3,6),(3,7),(3,8) in $B$.
Since $\delta(6,a) = 3$ and $\delta(1,a) = 2$ and (2,3) is in $B$ we add (1,6) to $B$.
Since $\delta(6,a) = 3$ and $\delta(2,a) = \delta(7,a) = \delta(8,a) = 7$ and $(3,7)$ is in $B$ we add (2,6), (6,7) and (6,8) to $B$.
Since $\delta(6,a) = 3$ and $\delta(5,a) = 8$ and (3,8) is in $B$ we add (5,6) to $B$.
Since $\delta(2,b) = \delta(8,b) = 3$ and $\delta(1,b) = \delta(5,b) = 6$ and $(3,6)$ is in $B$ then we add all the pairs (1,2),(1,8),(2,5),(2,8) to $B$.
Since $\delta(2,b) = \delta(8,b) = 3$ and $\delta(7,b) = 5$ and (3,5) is in $B$ then we add (2,7),(7,8) to $B$.

At this point (since state 4 has been removed) the only pairs not yet in $B$ are (1,5),(1,7),(5,7) and (2,8).
Now $\delta(1,b) = \delta(5,b) = 6$ and $\delta(7,b) = 5$ and $(5,6)$ is in $B$ so we add (1,7) and (5,7) to $B$.

We now only have (1,5) and (2,8) not yet in $B$. Now $\delta(1,b) = \delta(5,b)$. Also $\delta(1,a) = 2$ and $\delta(5,a) = 8$ but (2,8) is not in $B$ so there is no reason to include (1,5) in $B$. We also have $\delta(2,a) = \delta(8,a) = 7$ and $\delta(2,b) = \delta(8,b) = 3$ so there is no reason to include (2,8) in $B$.

Therefore the procedure halts and $\equiv_{min}$ has classes: $\{$1,5$\}$,$\{$7$\}$,$\{$2,8$\}$,$\{$6$\}$,$\{$3$\}$

The minimal DFA has 5 states and is shown in Figure 2.

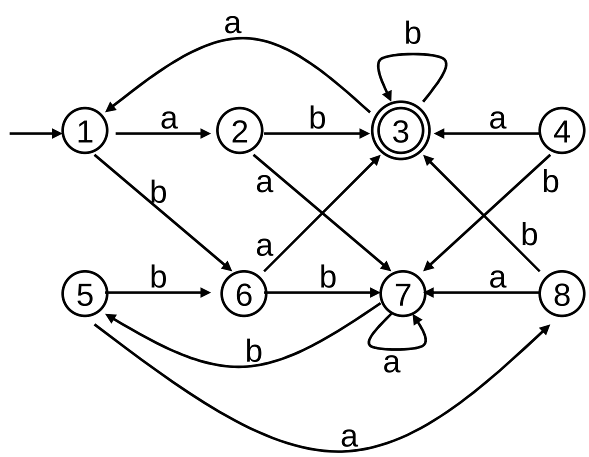


Figure 1: Original DFA

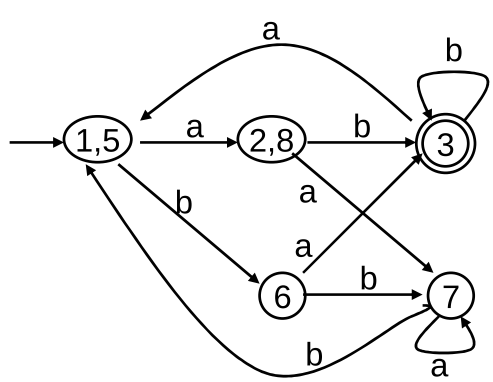


Figure 2: Minimized DFA