Minimizing DFAs

CSC320

DFA State Minimization

- Given a DFA $M = (Q, \Sigma, \delta, q_0, F)$, for a language L, there is a procedure for constructing a *minimal* DFA with as few states as possible which is unique up to isomorphism (i.e., renumbering of the states).
- The process has two stages:
  - Get rid of inaccessible states.
  - Collapse “equivalent” states.
- We need a procedure to *partition* Q into a collection Π of classes of equivalent states

The partition procedure

$\Pi \leftarrow \{F, Q - F\}$; refined $\leftarrow$ $true$;

While refined **do**

refined $\leftarrow$ $false$;

For $B \in \Pi$ **do**

If $\exists\ a, \exists\ p, q \in B, \exists\ B' \in \Pi$, **s.t.** $\delta(p, a) \in B'$ **and** $\delta(q, a) \notin B'$ **then**

$B_1 \leftarrow \{r \in B \mid \delta(r, a) \in B'\}; B_2 \leftarrow B - B_1;$

$\Pi \leftarrow (\Pi - \mathrm{B}) \cup B_1 \cup B_2$; refined $\leftarrow$ $true$;

Partitions

- A *partition* of Q is a collection $\Pi = \{B_1, \dots, B_k\}$ of disjoint subsets (called *classes*) of Q such that $\mathrm{Q} = B_1 \cup \cdots \cup B_k$
- Clearly, the procedure produces a partition of Q
- Write $p \equiv q$ if p, q are in the same class
- We say that Π is *compatible* with M if:

1. $p \equiv q \Rightarrow \delta(p, a) \equiv \delta(q, a)$ for all $a \in \Sigma$
2. $p \equiv q$ and $p \in F \Rightarrow q \in F$

- It is again clear that the procedure produces a compatible partition

The quotient construction

- For $q \in Q$, we write $[q]$ to denote the class of Π that contains q

- Define M' to be $(Q', \Sigma, \delta', [q_0], F')$, where
  - $Q' = \{[q] \mid q \in Q\}$
  - $F' = \{[q] \mid q \in F\}$
  - $\delta' : Q' \times \Sigma \rightarrow Q'$ is defined by $\delta'([q], a) = [\delta(q, a)]$
- We need compatibility for this to work

The language of M'

- This is a general result about any compatible partition
- Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA and Π a partition of Q

- **Theorem:** If Π is compatible then $L(M') = L(M)$
- **Proof:** By induction, $\widehat{\delta}'([q], w) = [\hat{\delta}(q, w)]$ But then

$$w \in L(M') \Leftrightarrow \widehat{\delta}'([q], w) \in F' \Leftrightarrow [\hat{\delta}(q, w)] \in F' \Leftrightarrow \hat{\delta}(q, w) \in F \Leftrightarrow w \in L(M)$$

Minimality

- We've shown that $L(M') = L(M)$. But how do we know that M' is minimal? We will need the following Lemma
- **Lemma:** Suppose $p, q \in Q$. Then $p \not\equiv q$ iff there is a string w such that either $\hat{\delta}(p, w) \in F$ or $\hat{\delta}(q, w) \in F$, *but not both.*
- **Proof:** ($\Leftarrow$) Use induction on w. When $w = \epsilon$, $\hat{\delta}(p, w) = p$ and $\hat{\delta}(q, w) = q$, so at line (1) they end up in different classes. Now suppose $w = av$. Then $\hat{\delta}(p, w) = \hat{\delta}(\delta(p, a), v)$ and $\hat{\delta}(q, w) = \hat{\delta}(\delta(q, a), v)$, so it follows by IH that $\delta(p, a) \not\equiv \delta(q, a)$. It then follows by lines (5)—(7) that $p \not\equiv q$.

Minimality

- **Lemma:** Suppose $p, q \in Q$. Then $p \not\equiv q$ iff there is a string w such that either $\hat{\delta}(p, w) \in F$ or $\hat{\delta}(q, w) \in F$, *but not both.*
- **Proof:** ($\Rightarrow$) Suppose $p \not\equiv q$. Consider the point at which p, q were put into different classes. If this happened at line (1) then clearly exactly one of $\hat{\delta}(p, \epsilon)$ or $\hat{\delta}(q, \epsilon)$ is in F. Otherwise, assume inductively that the Lemma holds for any two states that have been placed into separate classes of the partition. We have that p, q are placed into separate classes at lines (5)—(7). This means there is some a such that $\delta(p, a)$ and $\delta(q, a)$ were already placed into separate classes, and inductively this means there is some v such that either $\hat{\delta}(\delta(p, a), v) \in F$ or $\hat{\delta}(\delta(q, a), v) \in F$, but not both. But then either $\hat{\delta}(p, av) \in F$ or $\hat{\delta}(q, av) \in F$, but not both.

Minimality

- Suppose there is a DFA M'' such that $L(M'') = L(M')$ and M'' has fewer states. Let q_0'' be the start state of M''
- There must be states $[p] \neq [q]$ in M' and a state r in M'' for which there are strings u, v such that $\delta'([q_0], u) = [p]$, $\delta'([q_0], v) = [q]$, and $\delta''(q_0'', u) = \delta''(q_0'', v) = r$ (Why? This is just because M' is a DFA and M'' has fewer states than M'.)
- So for any x, $ux \in L(M'')$ iff $vx \in L(M'')$
- But since p and q were not collapsed in M', by the Lemma we have $ux \in L(M')$ or $vx \in L(M')$, but not both.