Minimizing Deterministic Finite Automata

University of Victoria

CSC 320
Foundations of Computer Science

Oct 2011

An equivalence relation over a set A, denoted $\sim$, is a subset of $(A \times A)$ such that

For all $a, b, c \in A$:

$$(a, a) \in \sim, \qquad \text{ie } a \sim a \qquad \text{(reflexive)}$$

$$\Big((a, b) \in \sim\Big) \Rightarrow \Big((b, a) \in \sim\Big) \qquad \text{ie } a \sim b \Rightarrow b \sim a \qquad \text{(symmetric)}$$

$$\Big((a, b) \in \sim\Big) \text{ and } \Big((b, c) \in \sim\Big) \Rightarrow \Big((a, c) \in \sim\Big)$$

$$a \sim b \text{ and } b \sim c \Rightarrow a \sim c \qquad \text{(transitive)}$$

An equivalence class of $a \in A$ under $\sim$ is the set

$$[a] = \{b \in A \mid a \sim b\}$$

Suppose we have language $L \subseteq \Sigma^*$,
$x, y \in \Sigma^*$,
a DFA $M = (K, \Sigma, \delta, s, F)$ with $L(M) = L$,
states $q, p \in K$ and $f_1, f_2 \in F$.

We need the following three equivalence relations:

$\approx_L$ over Σ^*:	$\forall z \in \Sigma^*$	$xz \in L \Leftrightarrow yz \in L$
$\sim_M$ over Σ^*:	$\exists q$	$(s, x) \vdash_M^* (q, e)$ and $(s, y) \vdash_M^* (q, e)$
$\equiv$ over K:	$\forall z \in \Sigma^*, \exists f_1, f_2$	$(p, z) \vdash_M^* (f_1, e) \Leftrightarrow (q, z) \vdash_M^* (f_2, e)$

The text denotes equivalence classes of $\approx_L$ as $[x]$, for $x \in \Sigma^*$
equivalence classes of $\sim_M$ as $E_q \subseteq \Sigma^*$, for states q of M.

Theorem 2.5.1

For any DFA M and any strings $x, y \in \Sigma^*$

$$x \sim_M y \;\Rightarrow\; x \approx_{L(M)} y$$

This implies $\Big($for x such that $(s, x) \vdash_M^* (q, e)\Big)$ that $E_q \subseteq [x]$.

Ultimately, then any DFA we construct for L must have at least as many states as the number of equivalence classes of Σ^* under $\approx$.

The standard automaton for L (read Myhill-Nerode Theorem)

$M = (K, \Sigma, \delta, s, F)$ such that

$$K = \{[x] \mid x \in \Sigma^*\}$$

For any $[x] \in K$ and any $a \in \Sigma$, define $\delta([x], a) = [xa]$

$$s = [e]$$

$$F = \{[x] \mid x \in L\}$$

Corollary

A language L is regular if and only if $\approx_L$ partitions Σ^* into finitely many equivalence classes.

For a given language L, in general, it isn't clear how to compute $\{[x] \mid x \in \Sigma^*\}$

We need some DFA M of L and the equivalence relation

$\equiv$ over K : $\quad \forall z \in \Sigma^*, \exists f_1, f_2 \quad (p,z) \vdash_M^* (f_1, e) \Leftrightarrow (q,z) \vdash_M^* (f_2, e)$

The equivalence classes of $\equiv$ describe how to adjust M to the standard automaton for L.

Get them with the sequence of relations

$\equiv_n$ over K : $\quad \forall z \in \Sigma^*, |z| \leq n, \exists f_1, f_2 \quad (p,z) \vdash_M^* (f_1, e) \Leftrightarrow (q,z) \vdash_M^* (f_2, e)$

Lemma 2.5.1:

For any two states $p, q \in K$ and any integer $n \geq 1$, $q \equiv_n p$ if and only if

(a) $q \equiv_{n-1} p$ $\qquad$ (b) for all $\forall a \in \Sigma$, $\delta(q,a) \equiv_{n-1} \delta(p,a)$

For $\equiv_0$, equivalence classes are $K - F$ and F.
Get equivalence classes of $\equiv_n$ by using Lemma 2.5.1.

Repeat until there is no change in the partitioning of states.