State minimization

Given DFA $A = (Q, \Sigma, \delta, q_0, F)$, find A' with minimum number of states s.t. $\mathcal{L}(A') = \mathcal{L}(A)$.

Idea: partition Q into equivalence classes and collapse equivalent states

Equivalence relation on states:

$$p \equiv q \quad \text{if for all } w \in \Sigma^* : \hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F$$

The equivalence relation induces a partition of Q

$$Q = C_1 \cup C_2 \cup \cdots \cup C_k$$

for all $p \in C_i$, $q \in C_j$: $p \equiv q \Leftrightarrow i = j$

How do we find the partition? We discover inequivalent states:

$$p \not\equiv q \quad \text{if for some } w \in \Sigma^* \quad \hat{\delta}(p, w) \in F \text{ and } \hat{\delta}(q, w) \notin F$$

or viceversa.

Let $w = a_1 a_2 \cdots a_m$ (i.e. $|w| = m$)

$$p \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \longrightarrow \cdots \xrightarrow{a_{m-1}} p_{m-1} \xrightarrow{a_m} p_m$$

$$q \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \longrightarrow \cdots \xrightarrow{a_{m-1}} q_{m-1} \xrightarrow{a_m} q_m$$

one is final and the other is not

Note: $a_{i+1} \cdots a_m$ is a proof of length $m-i$ of inequivalence of p_i and q_i

Definition: $p \equiv_i q$ if for all w with $|w| \leq i$

$$\hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F$$

(intuitively, there is no inequivalence proof of lenght $\leq i$)

The following is immediate to see:

$p \not\equiv_{i+1} q$ if and only if for some $a \in \Sigma$

$\delta(p, a) \not\equiv_i \delta(q, a)$.

10/12/2007

<u>Algorithm</u> to compute $\not\equiv_i$ inductively on i:

step 0: partition $Q = C_1 \cup C_2$ with $C_1 = F$, $C_2 = Q - F$

justified since $p \not\equiv_0 q$ iff one is final and the other not

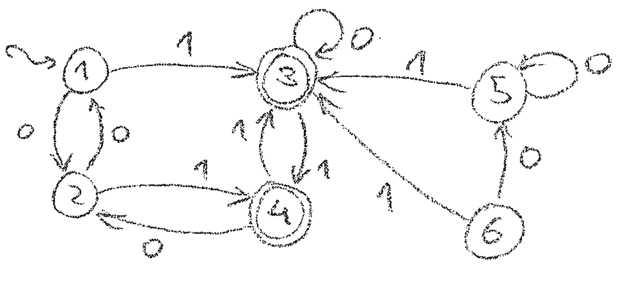
step $i+1$: determine $p \not\equiv_{i+1} q$ iff $\exists a \in \Sigma$

$\delta(p, a) \not\equiv_i \delta(q, a)$

compute refined partition

<u>Algorithm terminates when the refined partition coincides with</u> the one in the previous step (at most $|Q|$ steps)

<u>Example</u>:



step 0: $C_1^0 = \{1, 2, 5, 6\}$ $C_2^0 = \{3, 4\}$

step 1: $C_1^1 = \{1, 2, 5, 6\}$ $C_2^1 = \{3\}$ $C_3^1 = \{4\}$

step 2: $C_1^2 = \{1, 5, 6\}$ $C_2^2 = \{2\}$ $C_3^2 = \{3\}$ $C_4^2 = \{4\}$

step 3: $C_1^3 = \{1\}$ $C_2^3 = \{2\}$ $C_3^3 = \{3\}$ $C_4^3 = \{4\}$ $C_5^3 = \{5, 6\}$

step 4: no change

To construct A'

1) Construct partition $Q = C_1 \cup \dots \cup C_k$ of states of A

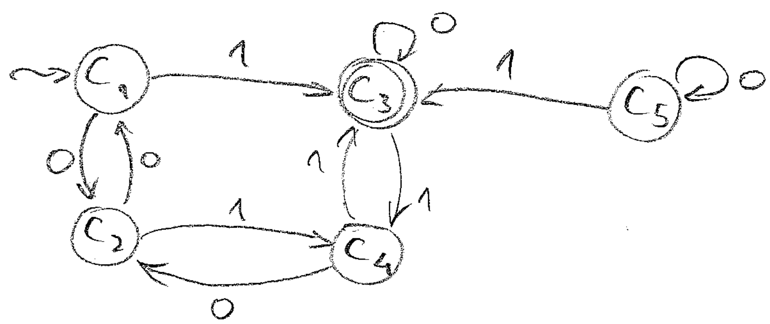
2) Construct $A' = (Q', \Sigma, \delta', q_0', F')$

- states $Q' = \{C_1, C_2, \dots, C_k\}$
- transitions: if $\delta(p, a) = q$ in A
  then $\delta(C[p], a) = C[q]$
  where $C[p]$ is the equivalence class of p
- start state: $C[q_0]$
- final states: $\{C[q_f] \mid q_f \in F\}$

We can verify that A' is a well-defined DFA.

$\boxed{\text{Exercise E4.4}}$

Example:



Note that C_5 is not reachable from the start state and must be removed.

We could show that the DFA constructed in this way is the smallest possible for a given language.

Myhill - Nerode Theorem:

Given $L \subseteq \Sigma^*$, consider the equivalence relation R_L on Σ^* defined as follows: $x R_L y \iff \forall z \in \Sigma^*: xz \in L \iff yz \in L$.
Then L is regular iff R_L induces a finite number of equivalence classes.