Chapter 1

Finite automata minimization algorithms

1.1 Introduction

1.2 Brzozowski's algorithm

$\varepsilon - free$ FA: $M_0 = (Q_0, V, T_0, \emptyset, S_0, F_0)$
to be minimized DFA: $M_2 = (Q_2, V, T_2, \emptyset, S_2, F_2)$
intermediate NFA: $M_1 = (Q_1, V, T_1, \emptyset, S_1, F_1)$

NFA: $M_1 \to$ DFA: $M_2, M_2 = suseful_s \circ subsetopt(M_1)$

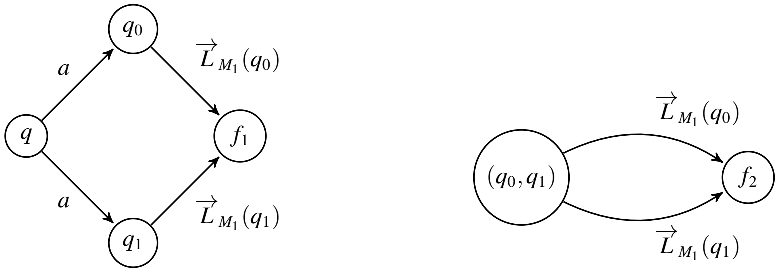
$$
\begin{aligned}
&q_0, q_1 \in Q_1, Q_2 \subseteq \mathbb{P}(Q_1), \forall p \in Q_2, p = (q_0, q_1) \\
&\overrightarrow{L}_{M_2}(p) = \overrightarrow{L}_{M_1}(q_0) \cup \overrightarrow{L}_{M_1}(q_1) \\
&\Rightarrow \\
&\overrightarrow{L}_{M_2}(p) = \bigcup_{q \in p} \overrightarrow{L}_{M_1}(q) \\
&\Rightarrow
\end{aligned}
$$

$$
\begin{tikzcd}
Q_1 \arrow[r, "subsetopt"] \arrow[dr] & P(Q_1) \arrow[d, "useful_s"] \\
& D(Q_2)
\end{tikzcd}
$$

图 1.1: $M_2 = suseful_s \circ subsetopt(M_1)$

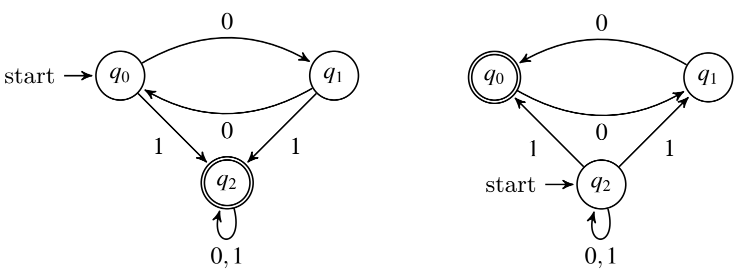
1.3 Minimization by equivalence of states

Let $A = (Q, V, T,, F)$ be a deterministic finite automaton, where Q is a finite set of states, V is a finite set of input symbols, T is a mapping from $Q \times V$ into Q, and $F \subseteq Q$ is the set of final states. No initial state

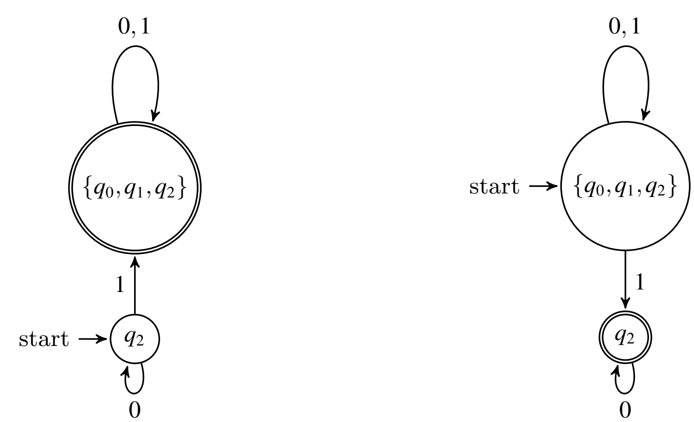


(a) $M_1 = (Q_1, V, T_1, \emptyset, S_1, F_1)$ (b) $M_2 = (Q_2, V, T_2, \emptyset, S_2, F_2)$

图 1.2: $M_2 = useful_s \circ subsetopt(M_1)$



(a) $M_0 = (Q_0, V, T_0, \emptyset, S_0, F_0)$ (b) $M_0^R = (Q_0, V, T_0, \emptyset, S_0, F_0)^R = (Q_0, V, T^R, \emptyset, F_0, S_0)$



(c) $useful_s \circ subsetopt \circ R(M_0)$ (d) $M_1 = R \circ useful_s \circ subsetopt \circ R(M_0)$

图 1.3: $M_1 = R \circ useful_s \circ subsetopt \circ R(M_0)$

is specified since it is of no importance in what follows. The mapping T is extended to $T \times V^*$ in the usual manner where V^* denotes the set of all finite strings (including the empty string ε) of symbols from V

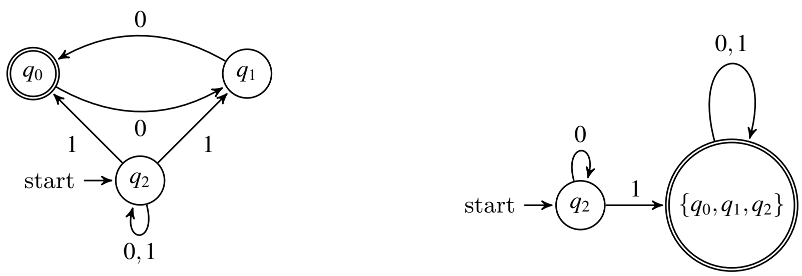
Definition 1.1 (equivalent states). The states s and t are said to be equivalent if for each $x \in V^*, T(s,x) \in F$ if and only if $T(t,x) \in F$.

From [Hopcroft71]

The algorithm for finding the equivalence classes of Q is described below:

Example 1.1. $Q = \{1,2,3,4,5,6\}, V = \{0,1\}, T$ see Fig. 1.7

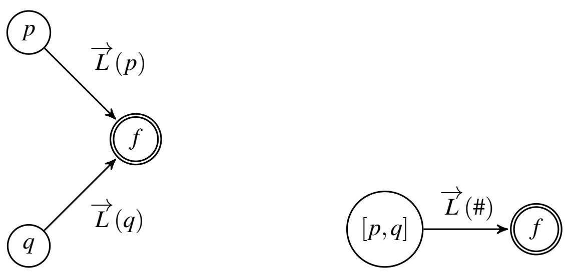
$$\begin{aligned}
\text{start: } & U = \{q_2\} \\
u = q_2 : & T(q_2, 0) = \{q_2\}, T(q_2, 1) = \{q_0, q_1, q_2\} \\
& \text{add new start to } D,\ D = \{q_2, \{q_0, q_1, q_2\}\} \\
u = \{q_0, q_1, q_2\} : & T(\{q_0, q_1, q_2\}, 0) = T(q_0, 0) \cup T(q_1, 0) \cup T(q_2, 0) = \{q_1\} \cup \{q_0\} \cup \{q_2\} = \{q_0, q_1, q_2\} \\
& T(\{q_0, q_1, q_2\}, 1) = T(q_0, 1) \cup T(q_1, 1) \cup T(q_2, 1) = \emptyset \cup \emptyset \cup \{q_0, q_1, q_2\} = \{q_0, q_1, q_2\}
\end{aligned}$$



(a) M (b) $useful_s \circ subsetopt(M)$

图 1.4: $useful_s \circ subsetopt(M)$

Equivalence relation $E \subseteq Q \times Q$
$(p, q) \in E \equiv (\overrightarrow{L}(p) = \overrightarrow{L}(q))$



(a) $(p, q) \in E$ (b) $(p, q) \in E$

图 1.5: Equivalence relation $E \subseteq Q \times Q$

$\overrightarrow{L}(p) = \bigcup_{a \in V}(\{a\} \cdot \overrightarrow{L}(T(p, a)) \cup \{\varepsilon | p \in F\}$

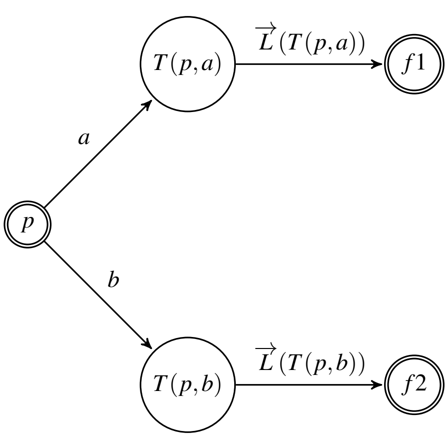


图 1.6: $L(p)$

Algorithm 1 The algorithm for finding the equivalence classes of Q

Input: $M = (Q,V,T,,F)$

Output: The equivalence classes of Q

Step 1. For each $s \in Q$ and each $a \in V$ construct

$T^{-1}(s,a) = \{t|T(t,a) = s\}$

Step 2. construct $B(1) = F, B(2) = Q - F$ and for each $a \in V$ and $1 \leq i \leq 2$ construct

for each $a \in V$ **do**

for $i = 1$; $i < n$; $i++$ **do**

$\hat{B}(B(i),a) = \{s|s \in B(i)$ and $T^{-1}(s,a) \neq \emptyset\}$;

end for

end for

Step 3. Set $k = 3$;

Step 4. For each $a \in V$ construct $L(a)$

for each $a \in V$ **do**

if $|\hat{B}(B(1),a)| \leq |\hat{B}(B(2),a)|$ **then**

$L(a) = \hat{B}(B(1),a)$;

else

$L(a) = \hat{B}(B(2),a)$;

end if

end for

Step 5. Select $a \in V$ and $i \in L(a)$. The algorithm terminates when $L(a) = \emptyset$ for each $a \in V$.

Step 6. Delete i from $L(a)$.

Step 7. For each $j < k$ such that there exists $t \in B(j)$ with $T(t,a) \in \hat{B}(B(i),a)$, perform steps 7a,7b,7c, and 7d.

Step 7a. partition $B(j)$ into

$B'(j) = \{t|T(t,a) \in \hat{B}(B(i),a)\}$ and

$B''(j) = B(j) - B'(j)$

Step 7b. Replace $B(j)$ by $B'(j)$ and constant $B(k) = B''$. Construct the corresponding $\hat{B}(B(j),a)$ and $\hat{B}(B(k),a)$ for each $a \in V$.

Step 7c. For each $a \in V$ modify $L(a)$ as follows.

if $j \notin L(a) \& 0 < |\hat{B}(B(j),a)| \leq |\hat{B}(B(k)),a|$ **then**

$L(a) = L(a) \cup \{j\}$;

else

$L(a) = L(a) \cup \{k\}$;

end if

Step 7d. Set $k = k + 1$.

Step 8. Return to Step 5.

The algorithm for finding the equivalence classes of Q is described below:

Step 1. For each $s \in Q$ and each $a \in V$ construct $T^{-1}(s,a) = \{t|T(t,a) = s\}$

$T^{-1}(1,0) = \emptyset, T^{-1}(2,0) = \{1\}, T^{-1}(3,0) = \{2\}, \cdots, T^{-1}(6,0) = \{5\}$

$T^{-1}(1,1) = \{1\}, T^{-1}(2,1) = \{2\}, \cdots T^{-1}(6,1) = \{6\}$,

Step 2. $B(1) = F = \{6\}, B(2) = Q - F = \{1,2,3,4,5\}$

for each $a \in V$ and $i \in [1,2]$ construct $\hat{B}(B(i),a) = \{s|s \in B(i)$ and $T^{-1}(s,a) \neq \emptyset\}$;

$\hat{B}(B(1),0) = \{6\}, \hat{B}(B(2),0) = \{2,3,4,5\}$

$\hat{B}(B(1),1) = \{6\}, \hat{B}(B(1),1) = \{1,2,3,4,5\}$

Step 3. Set $k = 3$

Step 4. For each $a \in V$ construct $L(a)$

$L(0) = \{6\}$, since $|\hat{B}(B(1),0)| = 1 \le |\hat{B}(B(2),0)| = 4$.

$L(1) = \{6\}$, since $|\hat{B}(B(1),1)| = 1 \le |\hat{B}(B(2),1)| = 5$.

Step 5. Select $a \in V$ and $i \in L(a)$. The algorithm terminates when $L(a) = \emptyset$ for each $a \in V$.

$a = 0$

$i = 1, \hat{B}(B(i),0) = \{6\}$

Step 6. Delete i From $L(a)$.

$L(0) = L(0) - B(i) = \emptyset$

Step 7. For each $j < k$ such that there exists $t \in B(j)$ with $T(t,a) \in \hat{B}(B(i),a)$, perform steps 7a,7b,7c, and 7d.

Step 7a. Partition $B(j)$ into

$B'(j) = \{t | T(t,a) \in \hat{B}(B(i),a)\} = \{5\}$ and

$B''(j) = B(j) - B'(j)$

Step 7b.

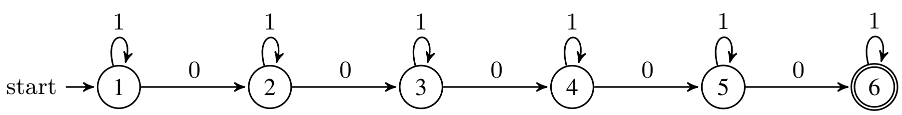


图 1.7: Minimizing example

Example 1.2. Consider the automaton with $Q = \{a,b,c,d,e\}, V = 0,1, F = \{d,e\}$, and T is given by the arcs of diagram of Fig. (1.8).

{a,b} is not equivalent, since $T(a,0) \in F$ but $T(b,0) \notin F$.

{d,e} is not equivalent, since $T(d,0) \in F$ but $T(e,0) \notin F$.

Sets of equivalent states: {a,c},{b},{d},{e}

另外一种描述：

1. $(a,b) \notin E$, since $T(a,0) \in F$ but $T(b,0) \notin F$.
2. $(d,e) \notin E$, since $T(d,0) \in F$ but $T(e,0) \notin F$.
3. $(a,c) \in E$, since $(a,c) \in E \equiv (a \in F \equiv c \in F) \wedge (\forall v \in V, (T(a,v), T(c,v) \in E)$

   $(a \notin F, c \notin F) \Rightarrow (a \in F \equiv c \in F)$

   $T(a,0) = T(c,0) = \{d\} \Rightarrow (T(a,0), T(c,0)) \in E$

   $T(a,1) = T(c,1) = \{c\} \Rightarrow (T(a,1), T(c,1)) \in E$

□.

Algorithm:

1. $B_1 \leftarrow F; B_2 \leftarrow (Q - F)$

   $B_1 = \{d,e\}; B_2 = \{a,b,c\}$
2. $|B_1| = 2, |B_2| = 3. \Rightarrow L \leftarrow (B_1, c)$

   $T(d,0) = \{d\} \in F; T(e,0) = \{c\} \notin F$

$\Rightarrow (d,e)$ is not equivalent states.

$T(d,1) = \{d\} \in F; T(e,1) = \{e\} \in F. \Rightarrow$ 无法判断。

$L = (B_1, 0)$;

3. split (d,c)

   $T(d,0) = \{d\} \in F; T(c,0) = \{d\} \notin F$ 无法判断

   $T(d,1) = \{d\} \in F; T(c,1) = \{c\} \notin F$

   $\Rightarrow (d,c)$ is not equivalent states.

{a,b},{d,e}is not equivalent states.

Sets of equivalent states: {a,c},{b},{d},{e}

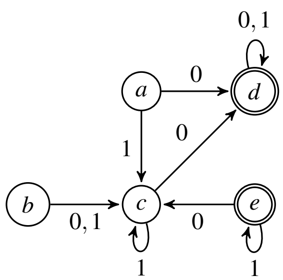


图 1.8: Finite state automaton

References

Hopcroft71. Hopcroft, J.E. *An n log n algorithm for minimizing states in a finite automaton*, in The Theory of Machines and Computations (Z. Kohavi, ed.), pp.180-196, Academic Press, New York, 1971.

Gries73. Gries, D. *Describing an Algorith m by Hopcroft*, Acta Inf. 2:97 109, 173. © by Springer-Verlag 1973