On The Comparison of Incompletely Specified Fintie Automata Models

by

C. C. Yang.

Associate Professor, National Chiao-Tung University

In the area of automata theory, two intuitive models, namely, recognition and transformation devices, are well known [1]-[2]. In this note, finite automata models refer to automata [3], and Mealy model [4] and Moore model [5] sequential machines which are deterministic and finite.

The equivalence of Mealy and Moore model machines without specifiying start states has appeared in the literature [6]-[11]. However, only Miller [9] has considered the equivalence between incompletely specified machines. After thorough review of these articles, the relationships among incompletely specified finite automata models are further investigated. Some explicit results including the differences between Moore and Mealy models and alternatively proved equivalence theorems among incompletely specified automata models will be developed.

First of all, we need several necessary definitions. Definitions One and Two involve two models of transformation device and Definition Three is related to recognition device.

Definition One: A Mealy model machine A is a 6-tuple

$$A=\langle S, \Sigma, \Delta, M, P, s_0\rangle$$

satisfying the following conditions:

(i) S, Σ, and Δ are finite nonempty sets (of states, inputs, and outputs respectively);

(ii) M is a function from a subset D_M of $S\times\Sigma$ into S (next state function);

(iii) P is a function from a subset D_P of $S\times\Sigma$ onto Δ (output function);

(iv) s_0 is in S (start or initial state).

When $D_M=D_P=S\times\Sigma$, i. e., $M(s_i, \sigma_j)$ and $P(s_i, \sigma_j)$ are defined for every s_i in S and every σ_j in Σ, A is a complete machine. If only $D_M=S\times\Sigma$, A is complete in state-transition. Otherwise, A is incompletly specified. Complete sequential machines are special cases of some incompletely specified models. In an incompletely specified machine, whenever $M(s_i, \sigma_j)$ or $P(s_i, \sigma_j)$ for any s_i in S and any σ_j in Σ is undefined, it is represented by a dash "—" in this note.

Definition Two: A Moore model machine B is a 6-tuple

$$B=\langle T, \Sigma, \Delta, N, Q, t_0\rangle$$

where T is the state set, Σ and Δ are defined as before, N is the next state function from a subset D_N of $T\times\Sigma$ into T, t_0 is the start state, and the output function Q is defined from a subset T_Q of T onto Δ such that

$$Q(N(t_i, \sigma_j))=P_B(t_i, \sigma_j) \qquad (1)$$

where P_B is an alternative output function from a subset D_{P_B} of $T\times\Sigma$ onto Δ and each (t_i, σ_j) is in $T\times\Sigma$.

The differences between Mealy and Moore models arise from the output functions P and Q (or P_B). In a Mealy model machine, the output defined by $P(s_i, \sigma_j)$ depends on the present state s_i in S and the present input σ_j in Σ. However, the ouptput defined by $P_B(t_i, \sigma_j)=Q(N(t_i, \sigma_j))$ of a Moore model for each t_i in T and each σ_j in Σ depends only on the next state defined by $N(t_i, \sigma_j)$. This has been shown by Krohn and Rhodes [12]. When the functions M, N, P, and Q (or P_B) are extended by considering an input tape of length longer than one, we can clearly see the differences between both these models.

Let Σ^* (Δ^*) denote the free semigroup with identity Λ (input tape of zero length) generated by Σ (Δ). Then the functions M and P of a Mealy model can be uniquely extended to the following mappings:

$$M: S\times\Sigma^*\rightarrow S$$

$$P: S\times\Sigma^*\rightarrow\Delta^*$$

by letting

$$M(s, \Lambda)=s \qquad (2a)$$

$$M(s, x\sigma)=M(M(s, x), \sigma) \qquad (2b)$$

$$P(s, \Lambda)=\Lambda \qquad (3a)$$

and

$$P(s, x\sigma)=P(s, x)\ P(M(s, x), \sigma) \qquad (3b)$$

for each (s, x, σ) in $S\times\Sigma^*\times\Sigma$.

For a Moore model machine, N can be extended in the same way as

M of a Mealy model machine by defining

$N(t, \Lambda)=t$ (4a)

$N(t, x\sigma)=N(N(t, x), \sigma)$ (4b)

for each (t, x, σ) in $T\times\Sigma^*\times\Sigma$. However, the extension of the function Q (or P_B) requires the understanding of the following facts first. By (1) and (4a), we have

$P_B(t, \Lambda)=Q(t)$ (5)

for each t in T. Since Q(t) defines the output associated with state t, or it may be undefined in some incompletely specified machine, $P_B(t, \Lambda)$ is not always equal to Λ as (3a) for a Mealy model machine. Secondly, by use of (1) and (4a), we have

$P_B(t, x\sigma)=P_B(N(t, x), \sigma)$ (6)

for each (t, x, σ) in $T\times\Sigma^*\times\Sigma$. This relation implies that P_B can be naturally extended only to a mapping from $T\times\Sigma^*$ into Δ instead of into Δ^* as (3b) for a Mealy model machine. In other words, P_B cannot be naturally extended in the same manner as P. In order to preserve the transformational capability of a Moore model machine, we need to define a new output function which will map $T\times\Sigma^*$ into Δ^*. Let H denote this function which should satisfy the following conditions:

$H(t, \Lambda)=P_B(t, \Lambda)$ (7a)

$H(t, \sigma)=P_B(t, \sigma)$ (7b)

$H(t, x\sigma)=H(t, x)\ H(N(t, x), \sigma)$ (7c)

for each (t, x, σ) in $T\times\Sigma^*\times\Sigma$. By use of (6) and (7b), $H(t, x\sigma)$ can be alternatively shown as the following form:

$H(t, x\sigma)=H(t, x)\ P_B(t, x\sigma)$ (8)

Furthermore, the extended output function P of a Mealy model machine is length preserving because of (3a). However, it is easily seen, from (5) and (7a), H is not length preserving.

Whenever only input-output tapes not containing Λ are of interest, we can let $\Sigma_-^*=\Sigma^*-\{\Lambda\}$, and $\Delta_-^*=\Delta^*-\{\Lambda\}$ and define $\hat{P}$ and $\hat{H}$ being the restrictions of P and H respectively to the following mappings:

$\hat{P}$: $S\times\Sigma_-^*\rightarrow\Delta_-^*$,

$\hat{H}$: $T\times\Sigma_-^*\rightarrow\Delta_-^*$.

Moreover, we define P_s ($\hat{P}_s$) and H_t ($\hat{H}_t$) as being (restricted) input-output functions from Σ^* (Σ_-^*) into Δ^* (Δ_-^*) for each s in S and each t in T respectively.

Definition Three: An automaton C is a 5-tuple

$C = \langle T, \Sigma, N, t_o, F \rangle$,

where T, N, and t_o are defined as before and F is a subset of T called the terminal (or final) state set.

An automaton defined above has in fact an understood natural output function P_B, namely

$P_B(t_o, x) = 1$, if $N(t_o, x)$ is defined and in F,

$= 0$, if $N(t_o, x)$ is defined and not in F,

$P_B(t_o, x)$ is undefined, if $N(t_0, x)$ is undefined,

for each x in Σ^*. Therefore the equivalence, in the sense of recognition capability, between an automaton C and a Moore machine B with $\Delta = \{0, 1\}$ is obvious and trivial.

Before giving the equivalence theorems of sequential machine models, some relevant concepts are briefly reviewed as entailed in the following:

Definition Four: Two states s and s′ of a Mealy model machine are equivalent, written $s \equiv s'$, if and only if $\hat{P}_s(y) = \hat{P}_{s'}(y)$ for all y in Σ_-^*. Here $\hat{P}_s$ instead of P_s is used because Λ is of no practical interest in a Mealy model machine due to its property described by (2a) and (3a).

In a Moore model machine, $t \equiv t'$ if and only if $H_t(x) = H_{t'}(x)$ for all x in Σ^*. Since a Moore machine has the property described by (5), it has an output Q(t) associated with its first state t when Λ is applied to the machine at t. Thus we use H_t instead of $\hat{H}_t$.

In both these machines, two states are equivalent if and only if they produce the same output sequence for every input tape.

An incompletely specified sequential machine is reduced if and only if no two distinct states are equivalent.

It should be noted that an output tape may contain some undefined symbols in an incompletely specified machine. Thus, two output tapes are said to be same if and only if their output symbols being either defined or undefined are identical at corresponding positions.

Definition Five: For some states t and t′ in T of a Moore model machine, tRt′ means that

$N(t, \sigma) = N(t', \sigma) = t''$, if $N(t, \sigma)$ and $N(t', \sigma)$ are both defined,

$= -$, if $N(t, \sigma)$ and $N(t', \sigma)$ are both undefined, for every σ in Σ.

Obviously, R is an equivalence relation. In this case, we call that states t and t′ are R-equivalent and also we define that

$$[t]=\{t' \mid t'Rt\}$$

is an R-equivalence class.

Note that $t \equiv t'$ implies that tRt'. However, the converse may not be true, unless $Q(t)=Q(t')$.

When the flow table of a Moore model machine is provided as shown below:

t	σ: σ_0	$\sigma_1 \cdots\cdots$	$\sigma_j \cdots\cdots$	δ
t_0				$Q(t_0)$
t_1		Next-state entries		$Q(t_1)$
⋮				⋮
t_i		$N(t_i, \sigma_j)$		$Q(t_i)$
⋮				⋮

we can easily find the R-equivalence classes by examining the next-state entries of the flow table. Whenever two rows of the next-state entries are identical for a reduced Moore machine their corresponding states are R-equivalent.

Definition Six: A state s in S (or t in T) of a machine A (or B) is said to be accessible if and only if $M(s_0, x)=s$ (or $N(t_0, x)=t$) for some x in Σ^*. Otherwise it is not accessible. If every s in S (or t in T) of A (or B) is accessible, then A (or B) is a connected machine.

Since incompletely specified machines are considered, some necessary restrictions in the transformation of machine models are briefly discussed:

(1) A Moore model machine B with specified start state t_0 as defined in Definition Two should have the following additional property. If there exists a nonaccessible state t other than t_0 in T, then t is redundant and should be deleted. If the start state t_0 is not accessible, then $Q(t_0)$ should be undefined because the first output symbol produced by the first input $\sigma \neq \Lambda$ is $Q(N(t_0, \sigma))$ instead of $Q(t_0)$. On the other hand, when the input is Λ and t_0 is not accessible, then

$$H(t_o, \Lambda)=Q(t_o) \tag{9}$$

Clearly, if $Q(t_o)$ is not undefined, then there exists a number of non-equivalent Moore model machines which are equivalent to a same Mealy machine as mentioned by Hartmanis [8 or 10]. However, a number of nonequivalent Moore machines whose start states having different defined outputs are accessible would not be equivalent to an identical Mealy machine. This fact will be easily seen from the proof of our equivalence theorm.

(2) In a Mealy model machine A defined in Definition One, if $P(s, \sigma) =\delta$ but $M(s, \sigma)$ is undefined for any s in S, any σ in Σ, and any δ in Δ, we can add an extra state s_σ to S by redefining $M(s, \sigma)=s_\sigma$.

Definition Seven: Two incompletely specified sequential machines A and B under previously described restrictions are equivalent, written $A\equiv B$, if and only if the output tapes generated by both machines due to a same input tape y are identical for every y in Σ^*.

In this definition the length of input tape y, written $\ell g(y)$, being never equal to zero is considered because either every state of a Moore model is assumed to be accessible or the only nonaccessible state of a Moore machine is the start state with undefined output under previously described restrictions and Λ is of no practical interest in a Mealy model machine.

Theorem 1 Let $B=\langle T, \Sigma, \Delta, N, P_B, t_o\rangle$ be a Moore model machine under the previously described restrictions for transformation. There exists a Mealy model machine $A=\langle S, \Sigma, \Delta, M, P, s_o\rangle$ such that $A\equiv B$. Moreover, if B is reduced, then so is A.

Proof First of all, find all possible R-equivalence classes of the given machine B by Definition Five. Then construct the equivalent Mealy model machine A by defining

$$S=\{[t] \mid t \text{ in } T\} \tag{10}$$

$$s_o=[t_o] \tag{11}$$

$$\begin{aligned} M([t], \sigma) &= [N(t, \sigma)], \text{ if } N(t, \sigma) \text{ in } T \\ &= - \quad , \text{ if } N(t, \sigma) = - \end{aligned} \tag{12}$$

$$\begin{aligned} P([t], \sigma) &= P_B(t, \sigma)=Q(N(t, \sigma)), \text{ if } N(t, \sigma) \text{ in } T \text{ and } Q(N(t, \sigma)) \text{ in } \Delta \\ &= - \quad , \text{ otherwise} \end{aligned} \tag{13}$$

for all t in T and all σ in Σ. It is easy to show that both functions M and P are well defined because of Definition Five.

To show the equivalence between A and B, let

$$Z(B)=\{\hat{H}_t(y) \mid y \text{ in } \Sigma_-^*\} \tag{14}$$

and

$$Z(A)=\{\hat{P}_s(y) \mid s=[t],\ y \text{ in } \Sigma_-^*\} \tag{15}$$

and then show that $Z(A)=Z(B)$. Let $y={}_0y_m$ be an arbitrary input tape with $\ell g(y)=m \geq 1$. Assume that there is a sequence of states $t_{i_0}, t_{i_1}, \cdots\cdots, t_{i_m}$ such that

$$N(t_{i_j}, {}_jy_{j+1})=t_{i_{j+1}},\ 0\leq j\leq m-1 \tag{16}$$

By inductive argument, we can easily show that

$$N(t_{i_0}, {}_0y_{j+1})=t_{i_{j+1}} \tag{17}$$

$$M([t_{i_0}], {}_0y_{j+1})=[t_{i_{j+1}}] \tag{18}$$

and

$$P([t_{i_j}], {}_jy_{j+1})=P_B(t_{i_0}, {}_0y_{j+1})=Q(t_{i_{j+1}}) \tag{19}$$

Assume that $\hat{H}_{t_{i_0}}(y)$ be in $Z(B)$. By use of (1), (6), (7b), (7c), (16), (17) and the definition of $\hat{H}_{t_{i_0}}$,

$$\hat{H}_{t_{i_0}}(y)=P_B(t_{i_0}, {}_0y_1)\ P_B(t_{i_0}, {}_0y_2)\cdots P_B(t_{i_0}, {}_0y_m) \tag{20a}$$

$$=Q(t_{i_1})\ Q(t_{i_2})\cdots Q(t_{i_m}) \tag{20b}$$

Assume that the start state t_0 is accessible, it is sufficient to apply arbitrary input tapes $y={}_0y_m$ whose lengths do not necessarily exceed $|T|-1$ (where $|T|$ indicates the cardinality of the set T) in an arbitrary state t_{i_0} in T and to check both output tapes due to a same input tape for the given machine B and the corresponding machine A.

Case 1 Assume that (16) holds for $0\leq j\leq m-1$ and $Q(t_{i_j})$ for $0\leq j\leq m$ are all defined, This is the case for a complete machine. Then by use of (3d), (13), (18), (19), (20a) and the definition of $\hat{P}_{[t_{i_0}]}$, we have

$$\begin{aligned}\hat{H}_{t_{i_0}}(y)&=P_B(t_{i_0}, {}_0y_1)\ P_B(t_{i_0}, {}_0y_2)\cdots P_B(t_{i_0}, {}_0y_m)\\ &\Rightarrow P([t_{i_0}], {}_0y_1)\ P([t_{i_1}], {}_1y_2)\cdots P([t_{i_{m-1}}], {}_{m-1}y_m)\\ &\Rightarrow P([t_{i_0}], {}_0y_1)\ P(M([t_{i_0}], {}_0y_1), {}_1y_2)\cdots\\ &\quad\cdots P(M([t_{i_0}], {}_0y_{m-1}), {}_{m-1}y_m)\\ &=\hat{P}_{[t_{i_0}]}(y)\end{aligned}$$

where $\ell g(\hat{H}_{t_{i_0}}(y))=\ell g(\hat{P}_{[t_{i_0}]}(y))=m$

Case 2 Assume that (16) holds for $0 \leq j \leq m-1$ and an arbitrary output $Q(i_k)$ for $0<k<m$ is undefined. Then by use of (20b) and similar argument of Case 1, we have

$$\hat{H}_{t_{i_0}}(y)=Q(t_{i_1})\cdots Q(t_{i_{k-1}})\ (-)\ Q(t_{i_{k+1}})\cdots Q(t_{i_m})$$
$$\Rightarrow P([t_{i_0}],\ {}_0y_1)\cdots P([t_{i_{k-2}}],\ {}_{k-2}y_{k-1})\ (-)$$
$$P([t_{i_k}],\ {}_ky_{k+1})\cdots P([t_{i_{m-1}}],\ {}_{m-1}y_m)$$
$$\Rightarrow \cdots\cdots$$
$$=\hat{P}_{[t_{i_0}]}(y)$$

where $\ell g(\hat{H}_{t_{i_0}}(y))=\ell g(\hat{P}_{[t_{i_0}]}(y))=m-1$.

Case 3 Assume that (16) is defined only for $0 \leq j<k<m$. Then states t_{i_j} and outputs $Q(t_{i_j})$ for $k \leq j \leq m$ are all undefined. In this case, despite the input tape y not being applicable [13] to the machine B, we still have

$$\hat{H}_{t_{i_0}}(y)=\hat{P}_{[t_{i_0}]}(y)$$

with $\ell g(\hat{H}_{t_{i_0}}(y))=\ell g(P_{[t_{i_0}]}(y))=k-1$

by use of similar arguments as in Cases 1 and 2. Hence $Z(B) \subseteq Z(A)$. Similarly we can show that $Z(A) \subseteq Z(B)$. Therefore $Z(A)=Z(B)$ implies that $A \equiv B$. If the start state t_0 is not accessible and if its output $Q(t_0)$ is undefined, then there do not exist non-equivalent machines of Moore type which are equivalent to a same Mealy machine.

The proof of machine reduction preservation is essentially the same as given by Hartmanis [8] or Ibarra [11]. We do not repeat here.

Since the R-equivalence classes are utilized, the number of states of the equivalent Mealy model may be fewer than that of the given Moore machine and is in fact equal to the number of R-equivalent classes.

Illustrative Example 1 The given Moore model machine under transformation is shown in the following flow table:

t	σ		δ
	σ_0	σ_1	
t_0	t_1	—	—
t_1	t_2	t_1	0
t_2	t_2	t_1	1

The R-equivalence classes contain the following two sets:

$[t_0]=\{t_0\}$

$[t_1]=[t_2]=\{t_1, t_2\}$

The state set S and the start state s_0 are found as shown below:

$S=\{[t_0], [t_1]\}$,

$s_0=[t_0]$

The values of the next state function M and the output function P are shown in the following flow table:

S	σ	
	σ_0	σ_1
$[t_0]$	$([t_1], \delta_0)$	—
$[t_1]$	$([t_1], \delta_1)$	$([t_1], \delta_0)$

Theorem 2 Let $A=(S, \Sigma, \Delta, M, P, s_0)$ be a Mealy model machine under the previously described possible addition of redefined states for transformation. There exists a Moore type $B=(T, \Sigma, \Delta, N, Q, t_0)$ such that $B\equiv A$. Moreover, if A is reduced then so is B.

Proof Provide the flow table of the given Mealy model machine as shown below:

S	σ			
	σ_0	σ_1	$\sigma_2 \cdots$	$\sigma_j \cdots$
s_0				
s_1	Next-state-and-output-entries			
⋮				
s_i	$(M(s_i, \sigma_j), P(s_i, \sigma_j))$			
⋮				

Define state set $T=T_1 \cup T_2$ where

(1) T_1 contains all distinct defined ordered pairs

$(s_k, \delta)=(M(s_i, \sigma_j), P(s_i, \sigma_j))$

by examining the next-state-and-output entries of the flow table;

(2) T_2 contains all distinct ordered-pairs (s, $-$) for all s other than s_k in S such that states does not appear as a defined oredered-pair in the flow table. By this construction, it is clear that

$$|T| \leq |S| \times |\Delta|.$$

Let ϕ denote either the undefined symbol $-$ or a symbol in Δ.
For all (s, ϕ) in T and every σ in Σ, define

$$N((s, \phi), \sigma) = (M(s, \sigma), P(s, \sigma)) \quad (21)$$

$$Q((s, \phi)) = \phi \quad (22)$$

From (1), (21) and (22) we can easily show that

$$P_B((s, \phi), \sigma) = P(s, \sigma) \quad (23)$$

Note that the functions N and P_B are independent from ϕ. This important property permits us to define the start state

$$t_0 = (s_0, \phi)$$

for any ϕ among the set $\{(s_0, \phi) \mid (s_0, \phi) \text{ in } T\}$.

For convenience in proving the equivalence theorem, we want to establish the extended definitions of N and P_B for $y = {}_0y_m$ in Σ^* as follows:

$$N((s, \phi)), {}_0y_m), (M(s, {}_0y_m), P(M(s, {}_0y_{m-1}), {}_{m-1}y_m)) \quad (24)$$

$$P_B((s, \phi), {}_0y_m) = P(M(s, {}_0y_{m-1}), {}_{m-1}y_m)) \quad (25)$$

Now we want to show that Z(B)=Z(A). Let $y = {}_0y_m$ in Σ^* with length not necessarily exceeding $|S| \times |\Delta| - 1$ be an arbitrary input tape and $\hat{H}_{(s, \phi)}(y)$ be in Z(B). Then by use of (3b), (7b), (23), and (25) we have

$$\hat{H}_{(s, \phi)}(y) = P_B((s, \phi), {}_0y_1)\, P_B((s, \phi), {}_0y_2) \cdots P_B((s, \phi), {}_0y_m)$$

$$\Rightarrow P(s, {}_0y_1)\, P(M(s, {}_0y_1), {}_1y_2) \cdots P(M(s, {}_0y_{m-1}), {}_{m-1}y_m)$$

$$= \hat{P}_s(y)$$

where $\ell g(\hat{H}_{(s, \phi)}(y)) = \ell g(\hat{P}_s(y)) = m$. Hence $Z(B) \subseteq Z(A)$. Similarly we can show that $Z(A) \subseteq Z(B)$. Therefore Z(B)=Z(A) implies that $B \equiv A$. The proof of machine reduction preservation is essentially the same as that given by Ibarra [11].

Illustrative Example 2 The given Mealy model machine under transformation is shown in the following flow table:

s	σ	
	σ_0	σ_1
s_0	(s_1, δ_1)	(s_3, δ_0)
s_1	$(s_2, -)$	—
s_2	(s_1, δ_0)	$(s_0, -)$
s_3	—	—

The state set T and the start state t_0 ase found to be

$$T = T_1 \cup T_2 = \{(s_1, \delta_0), (s_1, \delta_1), (s_3, \delta_0)\} \cup \{(s_0, -), (s_2, -)\}$$

$$= \{(s_0, -), (s_1, \delta_0), (s_1, \delta_1), (s_2, -), (s_3, \delta_0)\}$$

$$t_0 = (s_0, -)$$

The values of the next state function N and the output function Q are found as shown in the following flow table:

t	σ		δ
	σ_0	σ_1	
$(s_0, -)$	(s_1, δ_1)	(s_3, δ_0)	—
(s_1, δ_0)	$(s_2, -)$	—	δ_0
(s_1, δ_1)	$(s_2, -)$	—	δ_1
$(s_2, -)$	(s_1, δ_0)	$(s_0, -)$	—
(s_3, δ_0)	—	—	δ_0

REFERENCES

1. S. Ginsburg, An Introduction to Mathematical Machine Theory, Addison-Wesley, 1962, p. 108.
2. M. A. Harrison, Introduction to Switching and Automata Theory, McGraw-Hill, 1965, p. 368.

3. M. O. Rabin and D. Scott, "Finite Automata and Their Decision Problems" IBM J. Res. Dev., Vol 3, pp. 114-125; April, 1959.
4. G. H. Mealy, "A Method for Synthesizing Sequential Circuits," Bell Sys. Tech. J., Vol. 34, pp. 1045-1079; September, 1955.
5. E. F. Moore, "Gedanken Experiments on Sequential Machines," Automata Studies, C. E. Shannon and J. McCarthy, Eds., Princeton University Press, 1956; pp. 129-153.
6. W. J. Cadden, "Equivalent Sequential Circuits," IRE Trans. on Circuit Theory, Vol. CT-6, pp. 30-34; March, 1959.
7. A. Gill, "Comparsion of Finite-State Models," IRE Trans. on Circuit Theory, Vol. CT-7, pp. 178-179; June, 1960.
8. J. Hartmanis, "The Equivalence of Sequential Machine Models," IEEE Trans. on Electronic Computers, Vol EC-12, pp. 18-19, Feb. 1963.
9. R. E. Miller, Switching Theory, Vol. 2, John-Wiley, 1965: pp. 29-34 and 48.
10. J. Hartmanis and R. E. Stearns, Algebraic Theory of Sequential Machines, Prentice-Hall, 1966, pp. 23-27.
11. O. H. Ibarra, "On the Equivalence of Finite-State Sequential Machine Models," IEEE Trans. on Electronic Computers, Vol. EC-16, pp. 88-90; Feb., 1967.
12. K. B. Krohn and J. L. Rhodes, "Algebraic Theory of Machines," Proc. Symposium on Automata Theory, Polytechnic Institute of Brooklyn, 1962; pp. 341-384.
13. M. C. Paull and S. H. Unger, "Minimizing the Number of States in Incompletely Specified Sequential Switching Functions," IRE Trans. on Electronic Computers, Vol. EC-8, No. 3, pp. 356-367; September, 1959.