Decomposition of T-generalized State Machines

Sung-Jin Cho*, Jae-Gyeom Kim**, Seok-Tae Kim***

※Supported by NON DIRECTED RESEARCH FUND, Korea Research Foundation, 1996.

ABSTRACT

In this paper we introduce the notions of T-generalized state machines and primary submachines of T-generalized state machines and obtain a decomposition theorem for T-generalized state machine in terms of primary submachines.

Ⅰ. Introduction

Automata theory is one of basic and important theories in computer science. Following Zadeh [8] who introduced the concept of a fuzzy set, Wee [7] introduced the idea of fuzzy automata. There has been considerable growth in the area of fuzzy automata [2]. The use of algebraic techniques in determining the structure of automata has been significant. However, in fuzzy automata, the algebraic approach is lacking. Cho et al. [1] and Kim et al. [3] introduced the notions of TL-finite state machine and TL-transformation semigroup that are extensions of fuzzy state machine and fuzzy transformation semigroup, respectively. In [5] Malik et al. introduced the notions of submachines, primary submachines of fuzzy finite state machines and obtained a decomposition theorem for fuzzy finite state machine in terms of primary submachines. In this paper we introduce the notions of T-generalized state machines and primary submachines of T-generalized state machines and obtain a decompostion theorem for T-generalized state machine in terms of primary submachines. And we investigate some of algebraic properties of them.

*Dept. of Applied Mathematics, Pukyong National University
**Dept. of Mathematics, Kyungsung University
***Dept. of Telematics Engineering, Pukyong National University

Ⅱ. T-generalized state machines

For a state machine (Q, X, τ), $\tau: Q\times X \to Q$ can be regarded as a fuzzy subset τ of $Q\times X\times Q$ defined by $\tau(p, a, q)=1$ if $\tau(p, a)=q$ and $\tau(p, a, q)=0$ otherwise, and $\sum_{q\in Q} \tau(p, a, q) \leq 1$ for all $p\in Q$ and $a\in X$. Conversely, for a triple (Q, X, τ) with a fuzzy subset τ of $Q\times X\times Q$ such that $\tau(Q\times X\times Q)$ and $\sum_{q\in Q} \tau(p, a, q)$ ≤ 1 for all $p\in Q$ and $a\in X$, τ can be regarded as a partial function $\tau: Q\times X\to Q$ defined by $\tau(p, a)=q$ if $\tau(p, a, q)=1$. So the concept of state machines and the concept of fuzzy subsets with some restrictions can be identified. Now we can naturally fuzzify the concept of state machines.

Definition 2.1. A triple $M=(Q, X, \tau)$ where Q and

X are finite nonempty sets and τ is a fuzzy subset of $Q \times X \times Q$, i.e., τ is a function from $Q \times X \times Q$ to $[0, 1]$, is called a generalized state machine if $\sum_{q \in Q} \tau(p, a, q) \leq 1$ for all $p \in Q$ and $a \in X$. If $\sum_{q \in Q} \tau(p, a, q) = 1$ for all $p \in Q$ and $a \in X$, then M is said to be complete.

Note that the concept of generalized state machines is different from the concept of fuzzy finite state machines of Malik et al. [4] that also is a fuzzification of the concept of state machines. Their notion is based on the concept of fuzzy automata introduced by Wee [7]. While a generalized state machine (Q, X, τ) with $\tau(Q \times X \times Q) \subset \{0, 1\}$ can be always regarded as a state machine, a fuzzy finite state machine (Q, X, τ) with $\tau(Q \times X \times Q) \subset \{0, 1\}$ can not be regarded as a state machine generally. So the concept of generalized state machines is a generalization of the concept of state machines, whereas the concept of fuzzy finite state machines of Malik et al. [4] may not be considered as a generalization of the concept of state machines in a certain sense. This means that the concept of generalized state machines is a more adequate fuzzification of the concept of state machines than the concept of fuzzy finite state machines.

Let $M = (Q, X, \tau)$ be a generalized state machine. Then Q is called the set of states and X is called the set of input symbols. Let X^+ denote the set of all words of elements of X of finite length with empty word λ.

Formally, every incomplete generalized state machine can be extended to a complete generalized state machine as follows:

Definition 2.2. Let $M = (Q, X, \tau)$ be an incomplete generalized state machine. Let z be a state not in Q. The completion M^c of M is the complete generalized state machine (Q', X, τ') given by $Q' = Q \cup \{z\}$ and

$$\tau'(p', a, q') = \begin{cases} \tau'(p', a, q') & \text{if } p', q' \in Q \\ 1 - \sum_{q \in Q} \tau(p', a, q) & \text{if } p' \in Q \text{ and } q' = z \\ 0 & \text{if } p' = z \text{ and } q' \in Q \\ 1 & \text{if } p' = z \text{ and } q' = z \end{cases}$$

for all $a \in X$. The new state z is called the sink state of M^c. If M is complete, then we take M itself as M^c.

Definition 2.3 [6]. A binary operation T on $[0, 1]$ is called a t-norm if

(1) $T(a, 1) = a$,
(2) $T(a, b) \leq T(a, c)$ whenever $b \leq c$,
(3) $T(a, b) = T(b, a)$,
(4) $T(a, T(b, c)) = T(T(a, b), c)$

for all $a, b, c \in [0, 1]$.

The maximum and the minimum will be written as $\vee$ and $\wedge$, respectively. T is clearly $\vee$-distributive, i.e., $T(a, b \wedge c) = T(a, b) \vee T(a, c)$ for all $a, b, c \in [0, 1]$. Define T_0 on $[0, 1]$ by $T_0(a, 1) = a = T_0(1, a)$ and $T_0(a, b) = 0$ if $a \neq 1$ and $b \neq 1$ for all $a, b \in [0, 1]$. Then $\wedge$ is the greatest t-norm on $[0, 1]$ and T_0 is the least t-norm on $[0, 1]$, i.e., for any t-norm T, $\wedge(a, b) \geq T(a, b) \geq T_0(a, b)$ for all $a, b \in [0, 1]$.

T will always mean a t-norm on $[0, 1]$. T is said to be positive-definite if $T(a, b) > 0$ for all $a \neq 0$, $b \neq 0$. Throughout this paper, T shall mean a positive-definite unless otherwise specified. By an abuse of notation we will denote $T(a_1, T(a_2, \cdots, T(a_{n-2}, T(a_{n-1}, a_n)\cdots)))$ by $T(a_1, \cdots, a_n)$ where $a_1, \cdots, a_n \in [0, 1]$. The legitimacy of this abuse is ensured by the associativity of T(Definition 2.3(4)).

Definition 2.4. Let $M = (Q, X, \tau)$ be a generalized state machine. Define $\tau^+ : Q \times X^+ \times Q \to [0, 1]$ by

$$\tau^+(p, \lambda, q) = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$$

and

$$\tau^+(p, a \cdots a_n, q)$$
$$= \vee\{T(\tau(p, a_1, q_1), \tau(q_1, a_2, q_2), \cdots, \tau(q_{n-2}, a_{n-1}, q_{n-1}), \tau(q_{n-1}, a_n, q)) \mid q_i \in Q\}$$

where $p, q \in Q$ and $a_1, \cdots, a_n \in X$. When T is applied to M as above, M is called a T-generalized state machine.

Hereafter a generalized state machine will always be written as a T-generalized state machine because a generalized state machine always induces a T-generalized state machine as in Definition 2.4.

Proposition 2.5. Let $M = (Q, X, \tau)$ be a T-generalized state machine. Then

$$\tau^+(p, xy, q) = \vee\{T(\tau^+(p, x, r), \tau^+(r, y, q)) \mid r \in Q\}$$

for all $p, q \in Q$ and $x, y \in X^+$.

Proof. Let $p, q \in Q$. Let $x = a_1 \cdots a_n$ and $y = b_1 \cdots b_m$ with $a_1, \cdots, a_n, b_1, \cdots, b_m \in X$. Then

$$\vee\{T(\tau^+(p, x, r), \tau^+(r, y, q)) \mid r \in Q\}$$
$$= \vee\{T(\tau^+(p, a_1 \cdots a_n, r), \tau^+(r, b_1 \cdots b_m, q)) \mid r \in Q\}$$
$$= \vee\{T(\vee\{T(\tau(p, a_1, q_1), \cdots, \tau(q_{n-1}, a_n, r)) \mid q_1, \cdots, q_{n-1} \in Q\},$$
$$= \vee\{T(\tau(r, b_1, q_n), \cdots, \tau(q_{n+m-1}, b_m, q)) \mid q_n, \cdots, q_{n+m-1} \in Q\} \mid r \in Q\} \quad \text{by Definition 2.4}$$
$$= \vee\{T(\tau(p, a_1, q_1), \cdots, \tau(q_{n-1}, a_n, r), \tau(r, b_1, q_n), \cdots, \tau(q_{n+m-1}, b_m, q)) \mid q_1, \cdots, q_{n+m-1}, r \in Q\}$$
$$\text{by the } \vee\text{-distributivity of } T$$
$$= \tau^+(p, a_1 \cdots a_n b_1 \cdots b_m, q) \quad \text{by Definition 2.4}$$
$$= \tau^+(p, xy, q).$$

Ⅲ. Decomposition of T-generalized state machines

Definition 3.1. Let $M = (Q, X, \tau)$ be a T-generalized state machine. Let $p, q \in Q$. p is called an immediate successor of q if there exists $a \in X$ such that $\tau(q, a, p) > 0$. p is called a successor of q if there exists $x \in X^+$ such that $\tau^+(q, x, p) > 0$.

Proposition 3.2. Let $M = (Q, X, \tau)$ be a T-generalized state machine. Let $p, q, r \in Q$. Then

(1) q is a successor of q,

(2) if p is a successor of q and r is a successor of p, then r is a successor of q.

Proof. (1) Since $\tau^+(q, \lambda, q) = 1 > 0$, q is a successor of q.

(2) Let $x, y \in X^+$ so that $\tau^+(q, x, p) > 0$ and $\tau^+(p, y, r) > 0$. Then by Propostion 2.5 we have

$$\tau^+(q, xy, r) = \vee\{T(\tau^+(q, x, s), \tau^+(s, y, r)) \mid s \in Q\}$$
$$\geq T(\tau^+(q, x, p), \tau^+(p, y, r))$$
$$> 0.$$

So we have (2).

When $M = (Q, X, \tau)$ be a T-generalized state machine, we denote $S_M(q)$ the set of all successors of q, where $q \in Q$.

Definition 3.3. Let $M = (Q, X, \tau)$ be a T-generalized state machine. Let $R \subset Q$. The set of all successors of R, denoted by $S_M(R)$, in Q is defined to be the set

$$S_M(R) = \cup\{S_M(q) \mid q \in R\}.$$

We will write $S(q)$ and $S(R)$ for $S_M(q)$ and $S_M(R)$, respectively.

Theorem 3.4. Let $M = (Q, X, \tau)$ be a T-generalized state machine. Let $A, B \subset Q$. Then

(1) if $A \subset B$, then $S(A) \subset S(B)$,

(2) $A \subset S(A)$,

(3) $S(S(A)) = S(A)$,

(4) $S(A \cup B) = S(A) \cup S(B)$,

(5) $S(A \cap B) \subset S(A) \cap S(B)$.

Proof. The proofs of (1), (2), (4) and (5) are straightforward.

(3) Clearly $S(A) \subset S(S(A))$. Let $q \in S(p)$ for some $p \in S(A)$. Since $p \in S(A)$, $p \in S(r)$ for some $r \in A$. So q

$\in S(r)$ by Proposition 3.2(2). Thus $q\in S(A)$ because $r \in A$. Hence $S(S(A))\subset S(A)$.

Definition 3.5. Let $M=(Q, X, \tau)$ be a T-generalized state machine. Let $R\subset Q$. Let ν be a fuzzy subset of $R\times X\times R$ and let $N=(R, X, \nu)$. The T-generalized state machine N is called a submachine of M if

(1) $\tau|_{R\times X\times R}=\nu$,

(2) $S_M(R)\subset R$.

For the convenience sake, we assume that $\phi=(\phi, X, \nu)$ is a submachine of a T-generalized state machine M. A submachine $N=(R, X, \nu)$ of $M=(Q, X, \tau)$ is called proper if $R\neq Q$ and $R\neq\phi$. Clearly, if K is a submachine of N and N is a submachine of M, then K is a submachine of M. Note that the number of all submachines of M is finite because Q is finite.

Definition 3.6. Let $M=(Q, X, \tau)$ be a T-generalized state machine. Let $R\subset Q$ and $\{N_i=(Q_i, X, \tau_i)|i\in\Lambda\}$ be the collection of all submachines of M whose state set contains R. Define $\langle R\rangle=\bigcap_{i\in\Lambda}\{N_i|i\in\Lambda\}=(\bigcap_{i\in\Lambda} Q_i, X, \wedge_{i\in\Lambda}\tau_i)$. Then $\langle R\rangle$ is called the submachine generated by R.

In Definition 3.6, $\langle R\rangle$ is clearly the smallest submachine of M whose state set contains R. The union $\bigcup_{i\in\Lambda} N_i$ of a collection $\{N_i=(Q_i, X, \tau_i)|i\in\Lambda\}$ of submachines of M is $(\bigcup_{i\in\Lambda} Q_i, X, \nu)$ where $\nu=\tau|_{(\bigcup_{i\in\Lambda} R_i\times X\times\bigcup_{i\in\Lambda} Q_i)}$. The union of submachines of M is clearly a submachine of M.

Definition 3.7. Let $M=(Q, X, \tau)$ be a T-generalized state machine. Let P be a submachine of M. Then P is called a primary submachine of M if

(1) There exists $q\in Q$ such that $P=\langle q\rangle$,

(2) For all $s\in Q$ if $P\subset\langle s\rangle$, then $P=\langle s\rangle$.

Lemma 3.8. Let $M=(Q, X, \tau)$ be a T-generalized state machine. Let $R\subset Q$. Then $(S(R), X, \tau|_{S(R)\times X\times S(R)})$ is a submachine of M.

Proof. It is clear.

Lemma 3.9. Let $M=(Q, X, \tau)$ be a T-generalized state machine. Let $R\subset Q$. Then $\langle R\rangle=(S(R), X, \tau|_{S(R)\times X\times S(R)})$.

Proof. By Definition 3.6 $\langle R\rangle=(\bigcap_{i\in\Lambda} Q_i, X, \wedge_{i\in\Lambda}\tau_i)$, where $\{N_i|i\in\Lambda\}$ is the collection of all submachines of M whose state set contains R and $N_i=(Q_i, X, \tau_i)$, $i\in\Lambda$. It suffices to show that $S(R)=\bigcap_{i\in\Lambda} Q_i$. Since $(S(R), X, \tau|_{S(R)\times X\times S(R)})$ is a submachine of M such that $R\subset S(R)$ by Lemma 3.8, $\bigcap_{i\in\Lambda} Q_i\subset S(R)$. Let $p\in S(R)$. Then there exist $r\in R$ and $x\in X^+$ such that $\tau^+(r, x, p)>0$. Now $r\in\bigcap_{i\in\Lambda} Q_i$. Since $\langle R\rangle$ is a submachine of M, $p\in S(\bigcap_{i\in\Lambda} Q_i)\subset\bigcap_{i\in\Lambda} Q_i$. Thus $S(R)\subset\bigcap_{i\in\Lambda} Q_i$. Hence $S(R)=\bigcap_{i\in\Lambda} Q_i$.

Theorem 3.10. Let $M=(Q, X, \tau)$ be a T-generalized state machine. Let $P=\{P_1, P_2, \cdots, P_n\}$ be the set of all distinct primary submachines of M. Then

(1) $M=\bigcup_{i=1}^{n} P_i$,

(2) $M\neq\bigcup_{i=1, i\neq j}^{n} P_i$ for any $j\in\{1, 2, \cdots, n\}$.

Proof. (1) Let $q\in Q$. Then by Lemma 3.8 and Lemma 3.9 $\langle q\rangle=(S(q), X, \tau|_{S(q)\times X\times S(q)})$ is a submachine of M. Thus either $\langle q\rangle\in P$ or there exists $p\in Q\backslash S(q)$ such that $\langle q\rangle\in\langle p\rangle$. Since Q is finite, either $\langle q\rangle\in P$ or there exists an integer $k(1\leq k\leq n)$ such that $\langle q\rangle\in\langle p_k\rangle\in P$. Thus $q\in\bigcup_{i=1}^{n} S(p_i)$ where $P_i=\langle p_i\rangle$. Hence $M=\bigcup_{i=1}^{n} P_i$.

(2) Let $N=\bigcup_{i=1, i\neq j}^{n} P_i$ and let $P_i=\langle p_j\rangle$. If $p_j=\bigcup_{i=1, i\neq j}^{n} S(p_i)$, then $p_j\in S(p_i)$ for some $i\neq j$. Hence $P_j=\langle p_j\rangle\subset P_j$. This is a contradiction because $P_j\neq P_i$. Hence $M\neq N$.

Let $M=(Q, X, \tau)$ be a T-generalized state machine. M is called singly generated if there exists $q\in Q$ such that $M=\langle\{q\}\rangle$.

Corollary 3.11. Let $M=(Q, X, \tau)$ be a T-generalized state machine. Then every singly generated submachine of $M\neq\phi$ is a submachine of a primary submachine of M.

Proof. By Theorem 3.10 $M=\bigcup_{i=1}^{n} P_i$ where P_i is a primary submachine of M. Moreover $Q=\bigcup_{i=1}^{n} S(p_i)$ where $P_i=\langle p_i\rangle$, $i=1, 2, \cdots, n$. Let A be a singly generated submachine of M. Then $A=\langle a\rangle$, $a\in S(p_i)$ for some i. Hence A is a submachine of P_i.

Definition 3.12. A T-generalized state machine $M=(Q, X, \tau)$ is said to be T-generalized retrievable if it satisfies the following; for $p, q\in Q$ if there exists $y\in X^+$ such that $\tau^+(q, y, p)>0$, then there exists $x\in X^+$ such that $\tau^+(p, x, q)>0$; or equivalently, $q\in S(p)$ if and only if $p\in S(q)$ where $p, q\in Q$.

Definition 3.13. A nonempty submachine $N=(R, X, \nu)$ of a T-generalized state machine $M=(Q, X, \tau)$ is said to be T-generalized separated if $S(Q\setminus R)=Q\setminus R$.

Proposition 3.14. Let $N=(R, X, \nu)$ be a nonempty submachine of a T-generalized state machine $M=(Q, X, \tau)$. Then N is T-generalized separated if and only if $S(Q\setminus R)=Q\setminus R$.

Proof. Suppose N is T-generalized separated. Let $q\in S(Q\setminus R)$. Then $q\neq R$. Thus $q\in Q\setminus R$. So $S(Q\setminus R)=Q\setminus R$. Hence $S(Q\setminus R)\cap R=\phi$. Conversely, suppose that $S(Q\setminus R)=Q\setminus R$. Then $S(Q\setminus R)\cap R=\phi$. So N is T-generalized separated.

Definition 3.15. A T-generalized state machine $M=(Q, X, \tau)$ is said to be T-generalized connected if M has no T-generalized separated proper submachines.

Definition 3.16. A T-generalized state machine $M=(Q, X, \tau)$ is called strongly T-generalized connected if $p\in S(q)$ for all $p, q\in Q$.

Proposition 3.17. Let $M=(Q, X, \tau)$ be a T-generalized state machine. Then M is strongly T-generalized connected if and only if M has no proper submachines.

Proof. Suppose M is strongly T-generalized connected. Let $N=(R, X, \nu)$ be a submachine of M such that $R\neq\phi$. Then there exists $q\in R$. Let $p\in Q$. Since M is strongly T-generalized connected, $p\in S(q)$. Hence $p\in S(q)\subset S(R)\subset R$ because N is a submachine of M. Thus $R=Q$ and so $M=N$. Conversely, suppose M has no proper submachines. Let $p, q\in Q$ and let $N=(S(q), X, \nu)$ where $\nu=\tau|_{S(q)\times X\times S(q)}$. Then N is a submachine of M and $S(q)\neq\phi$. Hence $S(q)=Q$. Thus $p\in S(q)$. Hence M is strongly T-generalized connected.

Theorem 3.18. Let $M=(Q, X, \tau)$ be a T-generalized state machine. Then the following are equivalent:

(1) M is T-generalized retrievable.

(2) M is the union of strongly T-generalized connected submachines of M.

Proof. The proof is similar to the proof of Theorem 4.8 [3].

Theorem 3.19. Let $M=(Q, X, \tau)$ be a T-generalized state machine. Then the following are equivalent:

(1) M is T-generalized retrievable.

(2) Every primary submachine of M is strongly T-generalized connected.

Proof. (1)$\Rightarrow$(2): Let P be a primary submachine of M. Then $P=\langle p\rangle$ for some $p\in Q$. Let $r, t\in S(p)$. Then there exist $x, y\in X^+$ such that $\tau^+(p, x, r)>0$ and $\tau^+(p, y, t)>0$. Since M is T-generalized retrievable, there exist $u, v\in X^+$ such that $\tau^+(r, u, p)>0$ and $\tau^+(t, v, p)>0$. Thus by Proposition 2.5 $\tau^+(t, vx, r)=\vee\{T(\tau^+(t, v, s), \tau^+(s, x, r))\,|\,s\in Q\}>0$. Hence r

$\in S(t)$, i.e., P is strongly T-generalized connected.

(2)⇒(1): From Theorem 3.10 $M=\bigcup_{i=1}^{n} P_i$ where P_i are primary submachines of M. Since P_i are strongly T-generalized connected, M is the union of strongly T-generalized connected submachines of M. Hence by Theorem 3.18 M is T-generalized retrievable.

Lemma 3.20. Let $M=(Q, X, \tau)$ be a T-generalized state machine. Then M has a strongly T-generalized connected submachine.

Proof. We prove the result by induction on $|Q|=n$. If $n=1$, then the result is obvious. Suppose the result is true for all T-generalized state machines $N=(R, X, \nu)$ such that $|R|<n$. Let $q\in Q$. Then $M'=(S(q), X, \tau|_{S(q)\times X\times S(q)})$ is a submachine of M by Lemma 3.8. If M' is strongly T-generalized connected, the result follows. Suppose that M' is not strongly T-generalized connected. Then there exists $p\in S(q)$ such that $q\notin S(p)$ and hence $S(p)\subset S(q)$. Now $|S(p)|<n$. Hence by induction hypothesis the T-generalized state machine $(S(p), X, \tau|_{S(p)\times X\times S(p)})$ has a strongly T-generalized connected submachine.

Theorem 3.21. Let $M=(Q, X, \tau)$ be a T-generalized state machine. Then the following are equivalent:

(1) M is T-generalized retrievable.

(2) Every singly generated submachine of M is primary.

(3) Every nonempty T-generalized connected submachine of M is primary.

Proof. (1)⇒(2): Let $N=\langle q\rangle$ be a singly generated submachine of M. From Theorem 3.10 $M=\bigcup_{i=1}^{n} P_i$ where the P_i are primary submachines of M. By Theorem 3.19, the P_i are strongly T-generalized connected. Then $\langle q\rangle\subset P_i$ for some i. By Proposition 3.17 $\langle q\rangle=P_i$. Thus N is primary.

(2)⇒(1): Since every singly generated submachine of M is primary, every singly generated submachine of M is strongly T-generalized connected. Thus every primary submachine of M is strongly T-generalized connected. Hence by Lemma 3.20 M is T-generalized retrievable.

(2)⇒(3): Let $N=(R, X, \nu)$ be a nonempty T-generalized connected submachine of M. Let $q\in Q$. Suppose $S(q)\neq R$. Since N is T-generalized connected, $S(R\backslash S(q))\cap S(q)\neq\phi$. Let $r\in S(R\backslash S(q))\cap S(q)$. Then $r\in S(t)$ for some $t\in R\backslash S(q)$ and $r\in S(q)$. Now $\langle r\rangle\subset\langle t\rangle$ and $\langle r\rangle\subset\langle q\rangle$. Since $\langle r\rangle$ is primary, $\langle t\rangle=\langle r\rangle=\langle q\rangle$. Thus $t\in S(q)$ which is a contradiction. Hence $N=\langle q\rangle$ and so N is primary.

(3)⇒(2): Let $N=\langle s\rangle$ be a singly generated submachine. By Lemma 3.20 N has a strongly T-generalized connected submachine $B=\langle r\rangle$, say. Then B is T-generalized connected and hence primary. Thus $\langle r\rangle=\langle s\rangle=N$. Hence N is primary.

Lemma 3.22. Let $M=(Q, X, \tau)$ be a T-generalized state machine and let $N=(R, X, \nu)$ be a T-generalized separated submachine of M. Then every primary submachine of N is also a primary submachine of M.

Proof. Let $\langle q\rangle$ be a primary submachine of N. Suppose $\langle q\rangle$ is not a primary submachine of M. Then there exists $p\in Q\backslash S(q)$ such that $\langle q\rangle\subset\langle p\rangle$. Clearly $p\notin R$. Thus $p\in Q\backslash R$. Since $q\in S(p)$, $q\in S(Q\backslash R)$. Thus $q\in S(Q\backslash R)\cap R$. This is a contradiction because N is T-generalized separated. Hence $\langle q\rangle$ is a primary submachine of M.

Theorem 3.23. Let $M=(Q, X, \tau)$ be a T-generalized state machine and let $N_i=(R_i, X, \nu_i)$, $i=1, 2, \cdots, n$ be the primary submachines of M. Then a proper submachine $N=(R, X, \nu)$ of M is T-generalized separated if and only if for some $J\subset\{1, 2, \cdots, n\}$, $J\neq\phi$, $Q\backslash R=\bigcup_{i\in J} R_i$.

Proof. Suppose $N=(R, X, \nu)$ be a proper T-generalized separated submachine of M. Then $S(Q\backslash R)=$

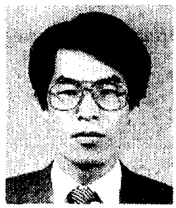
$Q\backslash R$. Since N is proper, $\langle Q\backslash R\rangle$ is nonempty. Thus $\langle Q\backslash R\rangle$ is the union of all its primary submachines by Theorem 3.10. Since $\langle Q\backslash R\rangle$ is T-generalized separated, by Lemma 3.22 every primary submachine of $\langle Q\backslash R\rangle$ is a primary submachine of M. Thus $S(Q\backslash R) = \bigcup_{i\in J} R_i$ for some $J\subset\{1, 2, \cdots, n\}$, $J\neq\phi$. Since $Q\backslash R = S(Q\backslash R)$ by Proposition 3.14, $Q\backslash R=\bigcup_{i\in J} R_i$. Conversely, let $N=(R, X, \nu)$ be a proper submachine of M such that $Q\backslash R=\bigcup_{i\in J} R_i$ for some $J\subset\{1, 2, \cdots, n\}$, $J\neq\phi$. Since $Q\backslash R=S(Q\backslash R)$, $Q\backslash R=\bigcup_{i\in J} R_i$. Since N_i is a submachine of M, $S(Q\backslash R)=S(\bigcup_{i\in J} R_i)=\bigcup_{i\in J} S(R_i)=\bigcup_{i\in J} R_i=Q\backslash R$ by Theorem 3.4. Hence N is T-generalized separated.

Corollary 3.24. Let $M=(Q, X, \tau)$ be a T-generalized state machine. Then M is T-generalized connected if and only if M has no proper submachine $N=(R, X, \nu)$ such that $Q\backslash R$ is the union of the sets of states of all primary submachines of M.

Proof. It is straightforward.

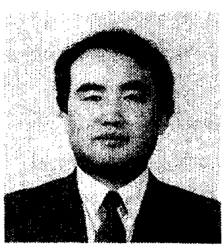
References

1. S.J. Cho, J.G. Kim and S.T. Kim, *TL*-state machines and *TL*-transformation semigroups, *J. Fuzzy Logic* and *Intelligent Systems* 5(4)(1995), 3-11.
2. A. Kandel and S.C. Lee, *Fuzzy switching* and *automata*: Theory and applications, (Crane Russak, New York, 1980).
3. J.G. Kim, S.J. Cho and S.T. Kim, On *TL*-finite state machines, *Korean J. Com.* and *Appl. Math.* 3 (1)(1996), 1-10.
4. D.S. Malik, J.N. Mordeson and M.K. Sen, Semigroups of fuzzy finite state machines, *Advances in Fuzzy Theory* and *Technology* 2, ed. P.P. Wang, 87-98, 1994.
5. D.S. Malik, J.N. Mordeson and M.K. Sen, Submachines of fuzzy finite state machines, *J. Fuzzy Math.* 2(1994), 781-792.
6. B. Schweizer and A. Sklar, Satistical metric spaces, *Pacific J. Math.* 10(1960) 313-334.
7. W.G. Wee, On generalizations of adaptive algorithm and applicaiton of the fuzzy sets concept to pattern classification, *Ph.D. Thesis,* Purdue Univ., June, 1967.
8. L.A. Zadeh, Fuzzy sets, *Inform. Control* 8(1965) 338-365.



조 성 진(Sung Jin Cho) 정회원
1979년 : 강원대학교 수학교육과 (이학사)
1981년 : 고려대학교 수학과 대학원 (이학석사)
1988년 : 고려대학교 수학과 대학원 (이학박사)
1988년~현재 : 부경대학교 자연과학대학 응용수학과 재직(부교수)

김 재 겸(Jae Gyeom Kim) 정회원
1981년 : 고려대학교 수학과(이학사)
1983년 : 고려대학교 수학과 대학원(이학석사)
1987년 : 고려대학교 수학과 대학원(이학박사)
1989년~현재 : 경성대학교 이과대학 수학과 재직(부교수)



김 석 태(Seok Tae Kim) 정회원
1983년 : 광운대학교 전자공학과 졸업(공학사)
1988년 : 경도공예섬유대학 전자공학과 졸업(공학석사)
1991년 : 오사카대학 통신공학과 졸업(공학박사)
1991년~현재 : 부경대학교 정보통신공학과 재직(조교수)