15th Triennial World Congress, Barcelona, Spain

A LANGUAGE MEASURE FOR DISCRETE-EVENT AUTOMATA[†]

Asok Ray
Email: axr2@psu.edu

Shashi Phoha
Email: sxp26@psu.edu

The Pennsylvania State University
University Park, PA 16802, USA

Abstract: This paper presents the concept and formulation of a signed real measure of regular languages for analysis of discrete-event supervisory control systems. The measure is constructed based upon the principles of language theory and real analysis for quantitative evaluation and comparison of the controlled behavior for discrete-event automata. The marked (i.e., accepted) states of finite-state automata are classified in different categories such that the event strings leading to good and bad marked states have positive and negative measures, respectively. In this setting, a controlled language attempts to disable as many bad strings as possible and as few good strings as possible. Different supervisors may achieve this goal in different ways and generate a partially ordered set of controlled languages. The language measure creates a total ordering on the performance of the controlled languages, which provides a precise quantitative comparison of the controlled plant behavior under different supervisors. The total variation of this language measure induces a norm on the vector space of sublanguages of the given regular language over the Galois field *GF*(2). *Copyright © 2002 IFAC*

Keywords: Automata theory; Discrete-event systems; Finite automata; Formal languages; Measures

1 INTRODUCTION

An important paradigm for Discrete Event System (DES) control is the Supervisory Control Theory (SCT), originally proposed by Ramadge '87 and subsequently extended by other researchers (for example, see the October 2000 issue of Part B of *IEEE Transactions on Systems, Man, and Cybernetics*). SCT partitions the behavior of a physical plant into legal and illegal categories. The legal behavior of the plant is modeled by a deterministic finite-state automaton, abbreviated as DFSA in the sequel. The DFSA model is equivalent to a regular language. Then, SCT synthesizes a DES controller as another language that guarantees restricted legal behavior of the controlled plant based on the desired specifications. Instead of continuous numerical data, DES controllers process event strings to disable certain controllable events in the physical plant. The algorithms for DES control synthesis have evolved based on the automata theory and formal languages in the discipline of Computer Science.

The controlled behavior of a given DFSA, also referred to as the plant, under different supervisors could vary, as they are designed based on different control specifications. As such the respective controlled sublanguages of the automaton form a partially ordered set that is not necessarily totally ordered. Since the literature on DES control does not apparently provide a language measure, it may not be possible to quantitatively evaluate the performance of a DES controller. Therefore, it is necessary to formulate a mathematically rigorous concept of language measure(s) to quantify performance of individual supervisors such that the measures of controlled plant behavior, described by a partially ordered set of controlled sublanguages, can be structured to form a totally ordered set. From this perspective, the goal of this paper is to construct a signed real measure that can be assigned to any sublanguage of the uncontrolled regular language of the plant to achieve the following objective:

[†] This work was supported in part by Army Research Office (ARO) under Grant No.DAAD19-01-1-0646 and Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under Agreement No. F3062-01-0575.

Given that the relation $\subseteq$ induces a partial ordering on a set of controlled sublanguages $\{L_k\}$ of a regular plant language $L(G)$, the language measure μ induces a total ordering $\leq$ on $\{\mu(L_k)\}$.

2 LANGUAGE MEASURE CONCEPT

Let $G \equiv \langle Q, \Sigma, \delta, q_1, Q_m \rangle$ be a trim (i.e., accessible and co-accessible) DFSA that represents the discrete-event dynamics of a physical plant [Ramadge '87; Kumar '95] where $Q = \{q_1, q_2, \cdots, q_n\}$ is the set of states with q_1 being the initial state; $\Sigma = \{\sigma_1, \sigma_2, \cdots, \sigma_m\}$ is the alphabet of events; $\delta : Q \times \Sigma \to Q$ is the (possibly partial) function of state transitions; and $Q_m \subseteq Q$ is the (non-empty) set of marked (i.e., accepted) states.

Let Σ^* be the Kleene closure of Σ, i.e., the set of all strings made of the events belonging to Σ as well as the empty string ε that is viewed as the identity element of the monoid Σ^* under the operation of string concatenation, i.e., $\varepsilon s = s = s\varepsilon$ $\forall s \in \Sigma^*$. Since δ is allowed to be a partial function, the regular language $L(G)$ generated by the DFSA G is given as: $L(G) \subseteq \Sigma^*$, and $L(G) = \Sigma^*$ iff $\delta : Q \times \Sigma \to Q$ is a total function. Therefore, if δ is a partial function, the set of states can be augmented with an additional non-marked dead-lock state q_{n+1}, called the dump state [Kumar '95], such that the partial function δ can be extended to a total function $\delta_{ext} : (Q \cup \{q_{n+1}\}) \times \Sigma \to (Q \cup \{q_{n+1}\})$.

Definition 1: *A $\sigma-$algebra M of a nonempty language $L(G) \subseteq \Sigma^*$ is a non-empty collection of subsets of $L(G)$ which satisfies the following three conditions:*

(i) $L(G) \in M$;

(ii) If $L \in M$, *then* $(L(G) - L) \in M$;

(iii) $\bigcup_{k=1}^{\infty} L_k \in M$ *if* $L_k \in M$ $\forall k$.

Definition 2: *An at most countable collection $\{L_k\}$ of members of a $\sigma-$algebra M is a partition of a member $L \in M$ if $L = \bigcup_k L_k$ and $L_k \cap L_j = \varnothing$ $\forall k \neq j$.*

Definition 3: *Given a $\sigma-$algebra M of the language $L(G)$, the set function $\mu : M \to \Re \equiv (-\infty, \infty)$, is called a signed real measure if the following two conditions are satisfied:*

(i) $\mu(\varnothing) = 0$;

(ii) $\mu\left(\bigcup_{k=1}^{\infty} L_k\right) = \sum_{k=1}^{\infty} \mu(L_k)$ *for every partition* $\{L_k\}$ *of any member* $L \in M$.

Note that, unlike a positive measure (e.g., the Lebesgue measure), μ is finite (but not bounded) such that the series in part (ii) of Definition 3 converges absolutely in $\Re$ and the result is independent of any permutation of the terms under union.

Definition 4: *Total variation measure $|\mu|$ on a $\sigma-$algebra M is defined as: $|\mu|(L) = Sup \sum_k |\mu(L_k)|$ $\forall L \subseteq M$ where the supremum is taken over all partitions $\{L_k\}$ of L.*

Definition 5: *Relative to the signed real measure μ, a sublanguage $L \in M$ is defined to be:*

(i) null, denoted as $L = 0$, if $\mu(L \cap J) = 0$ $\forall J \in M$;

(ii) positive, denoted as $L > 0$, if $\mu(L \cap J) \geq 0$ $\forall J \in M$;

(iii) negative, denoted as $L < 0$, if $\mu(L \cap J) \leq 0$ $\forall J \in M$.

Proposition 1: Total variation measure $|\mu|$ of any regular language $L(G)$ is non-negative and finite, i.e., $|\mu|(L(G)) \in [0, \infty)$. Hence, $|\mu|(L) \in [0, \infty)$ $\forall L \in M$.

Proof of Proposition 1: The proof follows standard theorems on complex measures [Rudin '88]. ■

Proposition 2: Every sublanguage $L \in M$ can be partitioned as: $L = L^0 \cup L^+ \cup L^-$ where mutually exclusive sublanguages L^0, L^+, and L^- are null, positive, and negative, respectively, relative to a signed real measure μ.

Proof of Proposition 2: The proof is based on the Hahn Decomposition Theorem [Rudin '88]. ■

3 LANGUAGE MEASURE FORMULATION

For a given DFSA $G \equiv \langle Q, \Sigma, \delta, q_1, Q_m \rangle$, we now construct a $\sigma-$algebra M of the regular language $L(G)$ as the power set $2^{L(G)}$. Based on the facts that $L(G) \subseteq \Sigma^*$ is at most countable and that every singleton legal string set belongs to $2^{L(G)}$, Definition 4 is modified as follows.

Definition 6: *Total variation measure $|\mu|$ on $2^{L(G)}$ is defined as: $|\mu|(L) = \sum_{s \in L} |\mu(\{s\})|$ $\forall L \subseteq L(G)$.*

The accepted or marked language $L_m(G)$ of a trim DFSA G has the following properties:

$\varnothing \subset L_m(G) \subseteq L(G)$; and $L_m(G) = L(G)$ iff $Q_m = Q$. Let the marked states be designated as: $Q_m \equiv \{q_{m_1}, q_{m_2}, \cdots, q_{m_\ell}\} \subseteq Q$ where $q_{m_k} = q_j$ for some $j \in \{1, 2, \cdots, n\}$.

Definition 7: *For a state $q \in Q$ of a given DFSA $G \equiv \langle Q, \Sigma, \delta, q_1, Q_m \rangle$, the regular language $L(q)$ is defined to be the set of all strings that terminate at q starting from the initial state q_1. Equivalently, $L(q)$ is the sublanguage of all legal event strings terminating at q starting from the initial state q_1.*

The Myhill-Nerode Theorem is now applied to construct the following state-based partitions [Hopcroft '01; Kumar '95; Martin '97]:

$$L(G) = \bigcup_{q \in Q} L(q); \text{ and } L_m(G) = \bigcup_{q \in Q_m} L(q)$$

where the sublanguage $L(q_k)$ is uniquely labeled by the state q_k $\forall k \in \{1, 2, \cdots, n\}$.

In order to obtain a quantitative measure of the marked language $L_m(G)$, the set of marked states is partitioned as: $Q_m = Q_m^+ \cup Q_m^-$ and $Q_m^+ \cap Q_m^- = \varnothing$. The positive set Q_m^+ contains *good* marked states that we desire to reach, and the negative set Q_m^- contains *bad* marked states that we want to avoid, although it may not always be possible to completely avoid the bad states while attempting to reach the good states. In general, the marked language $L_m(G)$ consists of both good and bad event strings that, starting from the initial state q_1, respectively lead to Q_m^+ and Q_m^-. Any event string belonging to the language $L(G) - L_m(G)$ leads to one of the non-marked states belonging to $(Q - Q_m)$ and does not contain any one of the good or bad strings.

The objective is to construct a performance measure of sublanguages of a regular language for discrete-event control and to define quantitative metrics of the controlled plant performance. To this end, the following definitions are introduced to construct a signed real measure of sublanguages of the regular language. This measure is not restricted to regular sublanguages of the original regular language based on which the measure is constructed.

In view of Definition 5, we proceed to construct a signed real measure $\mu : 2^{L(G)} \to \Re \equiv (-\infty, \infty)$ to allow state-based decomposition of $L(G)$ into null, positive, and negative sublanguages such that:

(i) $\mu(L(q)) = 0 \quad \forall q \notin Q_m$, i.e., the sublanguage corresponding to every non-marked state has zero measure;

(ii) Partitioning of Q_m into Q_m^+ and Q_m^- yields the following properties: $\mu(L(q)) > 0 \; \forall q \in Q_m^+$ and $\mu(L(q)) < 0 \; \forall q \in Q_m^-$, which is in agreement with Proposition 2 in the sense that $L^0 = \bigcup_{q \notin Q_m} L(q)$; $L^+ = \bigcup_{q \in Q_m^+} L(q)$; and $L^- = \bigcup_{q \in Q_m^-} L(q)$.

Partitioning the marked language $L_m(G)$ into a positive language $\bigcup_{q \in Q_m^+} L(q)$ and a negative language $\bigcup_{q \in Q_m^-} L(q)$ is equivalent to partitioning Q_m into the positive set Q_m^+ and the negative set Q_m^-. Each state belonging to Q_m^+ is characterized by a positive weight and each state belonging to Q_m^- by a negative weight. These weights are chosen by the designer based on his/her perception of each marked state's role in the system performance.

Definition 8: *The characteristic function $\chi : \{L(q_k) : k \in \{1, 2, \cdots, n\}\} \to [-1, 1]$ that assigns a signed real weight to state-partitioned sublanguages is defined as:*

$$\chi(L(q)) \in \begin{cases} [-1, 0) \text{ if } q \in Q_m^- \\ \{0\} \quad \text{if } q \notin Q_m \\ (0, 1] \text{ if } q \in Q_m^+ \end{cases}$$

The implication of the characteristic function is that a string belonging to a sublanguage $L(q_k)$, which is labeled by the state q_k, has a zero measure if q_k is not a marked state; a positive measure if q_k is a good marked state; and a negative measure if q_k is a bad marked state. For any accessible DFSA G, $L(q_k)$ is a nonempty language $\forall k \in \{1, 2, \cdots, n\}$.

We now introduce the cost of event strings belonging to $L(G)$. The cost assignment procedure is conceptually similar to that for conditional probability to events of a string. Since the consecutive events in a string may not be statistically independent, it is necessary to find the joint probability mass functions of arbitrarily large order. This makes the probability space of Σ^* ever expanding as there is no finite upper bound on the length of strings in Σ^*. This problem is circumvented by using the state transition function δ of G, which has finitely many Markov states.

Definition 9: *The event cost generated at a DFSA state is defined as $\tilde{\pi} : \Sigma^* \times Q \to [0, 1)$ such that $\forall q_j \in Q$, $\forall \sigma, \sigma_k \in \Sigma$, $\forall s \in \Sigma^*$,*

- $\forall \sigma_k \in \Sigma$, $\tilde{\pi}[\sigma_k | q_j] \equiv \tilde{\pi}_{jk} \in [0,1)$; $\sum_{j=1}^{m} \tilde{\pi}_{jk} < 1$;
- $\tilde{\pi}[\sigma | q_j] = 0$ *if* $\delta(q_j, \sigma)$ *is undefined;* $\tilde{\pi}[\varepsilon | q_k] = 1$;
- $\tilde{\pi}[\sigma s | q_j] = \tilde{\pi}[\sigma | q_j]\, \tilde{\pi}[s | \delta(q_j, \sigma)]$.

The event cost function $\tilde{\pi}$ for an event string $s \in L(q_k)$ starting from the initial state q_1 and terminating at q_k as the product of the respective conditional probabilities. For example, if $s = \sigma_i \sigma_j \sigma_k$, then $\tilde{\pi}(s) \equiv \tilde{\pi}(s|q_1) = \tilde{\pi}_{i1} \tilde{\pi}_{ja} \tilde{\pi}_{kb}$ where the state transition function δ of the DFSA $G \equiv \langle Q, \Sigma, \delta, q_1, Q_m \rangle$ defines $q_a = \delta(q_1, \sigma_i)$ and $q_b = \delta(q_a, \sigma_j)$ that are Markov states.

Definition 10: *The state transition cost of the DFSA is defined as a function* $\pi : Q \times Q \to [0,1)$ *such that* $\forall q_j,\ q_k \in Q,\ \ \pi(q_k | q_j) = \sum_{\sigma \in \Sigma : \delta(q_j, \sigma) = q_k} \tilde{\pi}(\sigma | q_k) \equiv \pi_{jk}$ *and* $\pi_{jk} = 0$ if $\{\sigma \in \Sigma : \delta(q_j, \sigma)\} = \varnothing$.

Remark 1: Definition 10 implies that, for an accessable language:

$$\mu(\{s\}) < \begin{cases} = 0 & \text{if } s \in L(q) \text{ for } q \notin Q_m \\ > 0 & \text{if } s \in L(q) \text{ for } q \in Q_m^+ \\ < 0 & \text{if } s \in L(q) \text{ for } q \in Q_m^- \end{cases}$$

Now we assign a signed measure μ to each string belonging to $L(G)$ that is partitioned by the sublanguages $L(q_k) : k \in \{1, 2, \cdots, n\}$ in terms of the signed weight of the characteristic function χ and the non-negative cost $\tilde{\pi}$.

Definition 11: *The signed measure* μ *of every singleton string set in* $2^{L(G)}$ *is defined as:* $\mu(\{s\}) \equiv \chi(L(q))\tilde{\pi}(s)$, *where* $s \in L(q)$.

From the perspective of performance evaluation of controlled automata under different DES supervisors, the role of the language measure is explained below:

A discrete-event non-marking supervisor S restricts the marked behavior of an uncontrolled plant automaton G such that $L_m(S/G) \subseteq L_m(G)$. The uncontrolled marked language $L_m(G)$ consists of good strings leading to Q_m^+ and bad strings leading to Q_m^-. A controlled language $L_m(S/G)$ should disable as many bad strings as possible and as few good strings as possible. Different supervisors may achieve this goal in different ways and generate a partially ordered set of controlled languages, $\{L_m(S_j/G) : j \in \{1, 2, \cdots, n_s\}\}$. The real signed measure μ provides a precise quantitative comparison of the controlled plant behavior under different supervisors because the set $\{\mu(L_m(S_j/G)) : j \in \{1, 2, \cdots, n_s\}\}$ is totally ordered.

4 LANGUAGE MEASURE COMPUTATION

The previous section formulated the real signed measure μ based on Definitions 8, 9 and 11. This section validates the construction of the measure μ in view of Proposition 1 by showing that $|\mu|(L(q)) < \infty \ \ \forall q \in Q_m$. This implies that the total variation $|\mu|(L(G)) < \infty$. Next, the measure of the marked language is quantified. To this end, we introduce several definitions and propositions.

Definition 12: *Given* $q_i, q_k \in Q$, *a string of events starting from* q_i *and terminating at* q_k *is called a path. A path* v *from* q_i *to* q_k *is said to pass through* q_j *if* $\exists$ *strings* $s \neq \varepsilon$ *and* $t \neq \varepsilon$ *such that* $v = st$; $\delta^*(q_i, s) = q_j$ *and* $\delta^*(q_j, t) = q_k$ *where* $\delta^* : Q \times \Sigma^* \to Q$.

Definition 13: *A path language* p_{ik}^j *is defined to be the set of all paths from* q_i *to* q_k, *which do not pass through any state* q_r *for* $r > j$. *The path language* p_{ik} *is defined to be the set of all paths from* q_i *to* q_k.

Based on the above definitions, we present the following propositions and lemmas to validate and quantify the language measure μ.

Proposition 3: Let $G \equiv \langle Q, \Sigma, \delta, q_1, Q_m \rangle$ be a DFSA with $Q = \{1, 2, \cdots, n\}$. Then, $p_{jk}^i = p_{jk} \ \ \forall i \geq n$.

Proof of Proposition 3: The proof relies on the fact that no string can pass through a state numbered higher than n. ■

Proposition 4: For a DFSA, every path language is regular.

Proof of Proposition 4: Since p_{jk}^0 is a finite language and hence regular, it follows from the proof of Kleene's Theorem [Martin '97, p. 123] by the induction hypothesis that p_{jk}^{i+1} is regular if p_{jk}^i is regular for all $0 \leq i \leq n$. ■

Proposition 5: Let u and v be two known regular expressions and let r be an unknown regular expression that is governed by the implicit equation: $r = ur + v$. Then, $\exists$ solutions $r = u^* v + \theta$ where θ satisfies the condition: $u\theta + v = \theta + v$; and the solution $r = u^* v$ is unique if $\varepsilon \notin u$.

Proof of Proposition 5: Existence is established by substituting $r = u^*v + \theta$ in $r = ur + v$ and then using the identity $u\theta + v = \theta + v$:

$$ur + v = u(u^*v + \theta) + v = uu^*v + (u\theta + v)$$
$$= uu^*v + (\theta + v) = (uu^* + \varepsilon)v + \theta = u^*v + \theta = r$$

If $\varepsilon \notin u$, then $u^* = u^*u + \varepsilon$ is a partition of u^*, which implies $u^*r = u^*ur + r$ is a partition of u^*r. It follows from $r = ur + v$ that $u^*r = u^*ur + u^*v \Rightarrow r \subseteq u^*v$.

Suppose $r \subset u^*v$. Partitioning of u^*v yields $u^*v = r + \varphi$ for some $\varphi \neq \varnothing$. It follows from $r = ur + v$ that $r + \varphi = u^*v = uu^*v + v$. Therefore, $u(r + \varphi) + v = r + u\varphi \Rightarrow \varphi \subseteq u\varphi$ which is a contradiction because $\varepsilon \notin u$. Hence, the solution $r = u^*v$ is unique if $\varepsilon \notin u$. An alternative proof is given in [Drobot '89]. ■

Proposition 6: For a given DFSA $G \equiv \langle Q, \Sigma, \delta, q_1, Q_m \rangle$, the following recursive relation holds for $0 \le i \le n-1$:

$$p^0_{jk} = \{\sigma \in \Sigma : \delta(q_j, \sigma) = q_k\} \text{ and}$$
$$p^{i+1}_{jk} = p^i_{jk} \cup p^i_{j,i+1}(p^i_{i+1,i+1})^* p^i_{i+1,k}$$

Proof of Proposition 6: Since the states are numbered from 1 to n in increasing order, $p^0_{jk} = \{\sigma \in \Sigma : \delta(q_j, \sigma) = q_k\}$ follows directly from the state transition map $\delta : Q \times \Sigma \to Q$ that is allowed to be a partial function. Given $p^i_{jk} \subseteq p^{i+1}_{jk}$, let us consider the set $p^{i+1}_{jk} - p^i_{jk}$ in which each string passes through q_{i+1} in the path from q_j to q_k and no string must pass through q_ℓ for $\ell > (i+1)$. Then, it follows that $p^{i+1}_{jk} - p^i_{jk} = p^i_{j,i+1}\, p^{i+1}_{i+1,k}$ where $p^{i+1}_{i+1,k}$ can be further expanded as: $p^{i+1}_{i+1,k} = (p^i_{i+1,i+1}\, p^{i+1}_{i+1,k}) \cup p^i_{i+1,k}$ that has a unique solution by Proposition 5 because $\varepsilon \notin p^i_{i+1,i+1}$. Therefore, $p^{i+1}_{jk} = p^i_{jk} \cup p^i_{j,i+1}(p^i_{i+1,i+1})^* p^i_{i+1,k}$. ■

Lemma 1: $\pi\left((p^0_{kk})^* \bigcup_{j \neq k} p^0_{kj}\right) \in [0,1)$.

Proof of Lemma 1: Following Definition 9, $\pi(p^0_{kk}) \in [0,1)$. Therefore, by convergence of a geometric series,

$$\pi\left((p^0_{kk})^* \bigcup_{j \neq k} p^0_{kj}\right) = \frac{\sum_{j \neq k} \pi(p^0_{kj})}{1 - \pi(p^0_{kk})} \in [0,1)$$

because $\sum_j \pi(p^0_{kj}) < 1 \Rightarrow \sum_{j \neq k} \pi(p^0_{kj}) < 1 - \pi(p^0_{kk})$. ■

Lemma 2: $\pi\left((p^i_{i+1,i+1})\right) \in [0,1)$.

Proof of Lemma 2: The path $p^i_{i+1,i+1}$ may contain at most i loops, one around the states $q_1, q_2, \cdots, q_i$. If the path $p^i_{i+1,i+1}$ does not contain any loop, then $\pi(p^i_{i+1,i+1}) \in [0,1)$ because it is a product of π_{jk}'s, each of which is a non-negative fraction. Next suppose there is a loop around q_j that does not contain any other loop; this loop must be followed by one or more events σ_k generated at q_j and leading to some other states q_ℓ where $\ell \in \{1, \cdots, i+1\}$ and $\ell \neq j$. By Lemma 1, $\pi(p^i_{i+1,i+1}) \in [0,1)$. Proof follows by starting from the innermost loop and ending with all loops at q_i. ■

Corollary to Lemma 2: $\pi\left((p^i_{i+1,i+1})^*\right) \in [1, \infty)$

Proof of Corollary to Lemma 2: Since $\pi(p^i_{i+1,i+1}) \in [0,1)$ from Lemma 2,

$$\pi\left((p^i_{i+1,i+1})^*\right) = \frac{1}{1 - \pi(p^i_{i+1,i+1})} \in [1, \infty)$$ ■

Proposition 7: For a given DFSA $G \equiv \langle Q, \Sigma, \delta, q_1, Q_m \rangle$, the following recursive relations hold for $0 \le i \le n-1$:

$$\pi(p^{i+1}_{jk}) = \pi(p^i_{jk}) + \frac{\pi(p^i_{j,i+1})\, \pi(p^i_{i+1,k})}{1 - \pi(p^i_{i+1,i+1})} \in [0, \infty)$$

Proof of Proposition 7: It follows from Definition 9 that $\pi\left(p^{i+1}_{jk} - p^i_{jk}\right) = \pi\left(p^i_{j,i+1}(p^i_{i+1,i+1})^* p^i_{i+1,k}\right)$. Since the languages $p^i_{j,i+1}, p^i_{i+1,i+1}$ and $p^i_{i+1,k}$ are mutually disjoint and $p^i_{jk} \subseteq p^{i+1}_{jk}$, it follows that:

$$\pi(p^{i+1}_{jk}) = \pi(p^i_{jk}) + \pi(p^i_{j,i+1})\, \pi\left((p^i_{i+1,i+1})^*\right) \pi(p^i_{i+1,k}).$$

The proof follows from Corollary to Lemma 2. ■

Remark 2: In view of Proposition 6, $L(q_k)$ in Definition 7 is obtained in terms of the path language p_{1k} as:

$$L(q_k) = \begin{cases} p_{11} \cup \{\varepsilon\} & \text{if } k = 1 \\ p_{1k} & \text{if } k > 1 \end{cases}$$
$$\Rightarrow \pi(L(q_k)) = \begin{cases} \pi(p_{11}) + 1 & \text{if } k = 1 \\ \pi(p_{1k}) & \text{if } k > 1 \end{cases}$$

Remark 3: The above recursive algorithm is generated in polynomial time. Specifically, an algorithm for numerically solving $\pi(p_{1k})$ requires three for-loops and hence, for a n-state automaton, the computation time is in the order of n^3.

5 NORMED SPACE FOR A REGULAR LANGUAGE

This section makes use of the language measure to construct a normed vector space of sublanguages for a given DFSA. The norm also induces a distance function between any two sublanguages of the regular language representing the DFSA.

Definition 14: *Let* $L(G)$ *be a regular language. The distance function* $d: 2^{L(G)} \times 2^{L(G)} \to [0,\infty)$ *is defined in terms of the total variation measure* $|\mu|$ *as:*

$$d(K_1, K_2) = |\mu|((K_1 \cup K_2) - (K_1 \cap K_2))$$
$$\forall K_1, K_2 \subseteq L(G).$$

Remark 4: The above distance function $d(\bullet,\bullet)$ satisfies the criteria for a pseudo-metric that can be converted to a metric by clustering of languages having zero distance from each other into the same equivalence class. This metric determines the distance between two controlled languages to quantify the distance between two supervisors relative to a given DFSA model of a plant.

Proposition 8: Let $L(G)$ be the language of a DFSA $G \equiv \langle Q, \Sigma, \delta, q_1, Q_m \rangle$. Let the binary operation of exclusive-OR $\oplus: 2^{L(G)} \times 2^{L(G)} \to 2^{L(G)}$ be defined as:

$$(K_1 \oplus K_2) \equiv (K_1 \cup K_2) - (K_1 \cap K_2)$$

$\forall K_1, K_2 \subseteq L(G)$. Then, $\langle 2^{L(G)}, \oplus \rangle$ is a vector space over the Galois field $GF(2)$.

Proof of Proposition 8: We notice that $\langle 2^{L(G)}, \oplus \rangle$ is an Abelian group where $\varnothing$ is the zero element of the group and the unique inverse of every element $K \in 2^{L(G)}$ is K itself because $K_1 \oplus K_2 = \varnothing$ if and only if $K_1 = K_2$. The associative and distributive properties of the vector space follows by defining the scalar multiplication of vectors as: $0 \otimes K \equiv \varnothing$ and $1 \otimes K \equiv K$. ■

Remark 5: The set $L(G)$ is isomorphic to a basis for the vector space $\langle 2^{L(G)}, \oplus \rangle$ over $GF(2)$.

Proposition 9: Total variation measure $|\mu|: 2^{L(G)} \to [0,\infty)$ is a seminorm on the vector space $\langle 2^{L(G)}, \oplus \rangle$ over $GF(2)$.

Proof of Proposition 9: We notice that $|\mu|(K) \geq 0$, $|\mu|(0 \otimes K) = 0$, and $|\mu|(1 \otimes K) = |\mu|(K) \ \ \forall K \in 2^{L(G)}$.

The remaining property of the triangular inequality: $|\mu|(K_1 \oplus K_2) \leq |\mu|(K_1) + |\mu|(K_2)$ follows from Definition 3 and the facts that $(K_1 \oplus K_2) \subseteq (K_1 \cup K_2)$ and $|\mu|(K_1) \leq |\mu|(K_2) \ \ \forall K_1 \subset K_2$. ■

Remark 6: The above semi-norm $|\mu|(\bullet)$ can be converted to a norm by clustering of all languages having zero total variation measure into the null equivalence class $N \equiv \{K \in 2^{L(G)} : |\mu|(K) = 0\}$ conceptually similar to what is done for defining norms in the L_p spaces [Rudin '88]. In that case, $N = \varnothing \cup \left(\bigcup_{q \notin Q_m} L(q) \right)$ includes all strings leading to the unmarked states from the initial state.

Remark 7: The norm $|\mu|(\bullet)$ can be generated from $d(\bullet,\bullet)$ as: $|\mu|(K) = d(K, J) \ \ \forall K \in 2^{L(G)} \ \forall J \in N$.

Remark 8: The normed space $\langle 2^{L(G)}, \oplus \rangle$ is transformed to a Banach space by completion of the language $L(G)$ to Σ^* by augmenting the state set Q by the additional dump state q_{n+1} that is unmarked as discussed earlier. In that case, the state transition function δ becomes a total function and the characteristic function $\chi(q_{n+1}) = 0$ following Definition 8. Therefore, a Banach space of sublanguages can be generated for a regular language by extending the domain of the language measure to 2^{Σ^*} from $2^{L(G)}$. Non-zero values of the measure remain unchanged and the null equivalence class becomes:

$$N = \{K \in 2^{\Sigma^*} : |\mu|(K) = 0\}.$$

REFERENCES

V. Drobot (1989), Formal Languages and Automata Theory, Computer Science Press.

H.E. Hopcroft, R. Motwani and J.D. Ullman (2001), *Introduction to Automata Theory, Languages, and Computation*, 2nd ed., Addison Wesley.

R. Kumar and V.K. Garg (1995), *Modeling and Control of Logical Discrete Event Systems*, Kluwer Academic.

J.C. Martin (1997), *Introduction to, Languages and the Theory of Computation*, 2nd ed, McGraw Hill.

P.J. Ramadge and W.M. Wonham (1987), "Supervisory Control of a Class of Discrete Event Processes," *SIAM Journal of Comtrol and Opimization*, **25 (1)**, pp. 206-230.

W. Rudin (1988), *Real and Complex Analysis*, 3rd ed, McGraw Hill, New York.