# Pattern Classification In Symbolic Streams via Semantic Annihilation of Information

Ishanu Chattopadhyay \*, Yicheng Wen<sup>†</sup> and Asok Ray<sup>‡</sup>

Abstract—We propose a technique for pattern classification in symbolic streams via selective erasure of observed symbols, in cases where the patterns of interest are represented as Probabilistic Finite State Automata (PFSA). We define an additive abelian group for a slightly restricted subset of probabilistic finite state automata (PFSA), and the group sum is used to formulate pattern-specific semantic annihilators. The annihilators attempt to identify pre-specified patterns via removal of essentially all inter-symbol correlations from observed sequences, thereby turning them into symbolic white noise. Thus a perfect annihilation corresponds to a perfect pattern match. This approach of classification via information annihilation is shown to be strictly advantageous, with theoretical guarantees, for a large class of PFSA models. The results are supported by simulation experiments.

Index Terms—Probabilistic Finite State Machines, Machine Learning, Pattern Classification

#### I. INTRODUCTION AND MOTIVATION

The principal focus of this work is the development of an efficient algorithm for identifying pre-specified patterns of interest in observed symbolic data streams, where the patterns are represented as Probabilistic Finite State Automata (PFSA) over pre-defined symbolic alphabets.

A finite state automaton (FSA) is essentially a finite graph where the nodes are known as states and the edges are known as transitions, which are labeled with letters from an alphabet. A string or a symbol string generated by a FSA is a sequence of symbols belonging to an alphabet, which are generated by stepping through a series of transitions in the graph. Probabilistic finite state automata, considered in this paper, are finite state machines with probabilities associated with the transitions. PFSA have extensively studied as an efficient framework for learning the causal structure of observed dynamical behavior [1]. This is an example of inductive inference [2], defined as the *process of hypothesizing a general rule from examples*. In this paper, we are concerned with the special case, where the *inferred general rule* takes the form of a PFSA, and the examples are drawn from a

Authors are with the department of Mechanical Engineering, The Pennsylvania State University, University Park, PA 16802, USA

This work has been supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under Grant No. W911NF-07-1-0376 and by the Office of Naval Research under Grant No. N00014-09-1-0688. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsoring agencies. stochastic regular language. Conceptually, in such scenarios, one is trying to learn the *structure inside of some black box*, *which is continuously emitting symbols* [1]. The system of interest may emit a continuous valued signal; which must be then adequately partitioned to yield a symbolic stream. Note that such partitioning is merely *quantization* and not *data-labeling*, and several approaches for efficient symbolization have been reported [3].

Probabilistic automata are more general compared to their non-probabilistic counterparts [4], and are more suited to modeling stochastic dynamics. It is important to distinguish between the PFSA models considered in this paper, and the ones considered by Paz [5], and in the detailed recent review by Vidal *et al.* [6]. In the latter framework, symbol generation probabilities are not specified, and we have a distribution over the possible end states, for a given initial state and an observed symbol. In the models considered in this paper, symbol generation is probabilistic, but the end state for a given initial state, and a generated symbol is unique. Unfortunately, authors have referred to both these formalisms as *probabilistic finite state automata* in the literature. The work presented here specifically considers the latter modeling paradigm considered and formalized in [1], [7], [8], [9].

The case for using PFSA as pattern classification tools is compelling. Finite automata are simple, and the sample and time complexity required for learning them can be easily characterized. This yields significant computational advantage in time constrained applications, over more expressive frameworks such as belief (Bayesian) networks [10], [11] or Probabilistic Context Free Grammars (PCFG) [12], [13] (also see [14] for a general approach to identifying PCFGs from observations) and hidden Markov models (HMMs) [15]. Also, from a computational viewpoint, it is possible to come up with provably efficient algorithms to optimally learn PFSA, whereas "optimally learning HMMs is often hard" [1]. Furthermore, most reported work on HMMs [15], [16], [17] assumes the model structure or topology is specified in advance, and the learning procedure is merely training, *i.e.*, finding the right transition probabilities. For PFSA based analysis, researchers have investigated the more general problem of learning the model topology, as well as the transition probabilities, which implies that such analysis can then be applied to domains where there is no prior knowledge as to what the correct structure might look like [18].

Although the reported PFSA construction algorithms [8], [7], [1] (*referred to as the direct compression algorithms in the sequel*) are asymptotically efficient, time critical applica-

<sup>\*</sup> Corresponding Author, email: ixc128@psu.edu

<sup>†</sup> email: yxw167@psu.edu

<sup>&</sup>lt;sup>‡</sup> email: axr2@psu.edu



Fig. 1: Concept of information annihilation: Addition of symbolic streams to yield symbolic white noise

tions (e.g. pattern classification in sensing and surveillance networks) often demand faster identification to what the state of the art can provide. This motivates the key problem investigated in this paper:

Given a set of PFSA models representing patterns of interest (i.e. a PFSA based pattern dictionary or library), the problem is to identify (in real or near-real time) if any of the specified patterns of interest exist in an observed symbol sequence, without resorting to direct compression and subsequent comparison [9] of the constructed PFSA model against the library elements.

We propose a novel classification technique based on selective erasure of observed symbols leading to perfect information annihilation as illustrated in Figure 1. Specifically, we construct an additive abelian group over a slightly restricted subset of all PFSA (over a fixed alphabet), and show that it is possible to define pattern-specific semantic annihilators as a function of the group inverses. These annihilators can then operate on the sensed stream in a symbol-by-symbol fashion attempting to eliminate all inter-symbol correlations. The annihilation is shown to be perfect if and only if the annihilator corresponds exactly to the inverted PFSA model of the underlying generating process. Thus we need to only check if the annihilated stream (corresponding to a particular PFSA) is free from any emergent pattern, *i.e.*, if the symbols are equi-probable in an history-independent manner (denoted as symbolic white noise in the sequel) to infer the existence of that pattern in the original observed stream.

The proposed approach is computationally efficient for direct compression of the symbol stream, since it is significantly easier to check if a symbolic stream is in fact symbolic white because the underlying PFSA model has a single state with equal symbol generation probabilities as seen in Figure 2b and 2c. It is also shown that the proposed technique is provably faster if the cardinality of the alphabet is not greater than the number of states in a particular pattern of interest, *which represents almost all PFSA models encountered in practice*.

The rest of the paper is organized in additional ten sections. Section II is a brief overview of preliminary concepts, and related work. Section III presents the construction of the additive abelian group for probability measures on symbol strings which is then shown to induce an abelian group on a restricted set of PFSA. Section IV develops a practical implementation of the PFSA sum which is then used to formulate the notion of the semantic annihilators in Section V. Section VI identifies the theoretical conditions under which we can guarantee classification via semantic annihilation to be faster than direct compression. Section VIII establishes asymptotic bounds on the run-time complexity of annihilators. Simulation results are presented in Section IX, and pertinent discussions, intuitive interpretations, and potential applications are delineated in Section X. The paper is concluded in Section XI with recommendations for future work.

#### II. PRELIMINARY CONCEPTS AND RELATED WORK

A string x over an alphabet (*i.e.* a non-empty finite set)  $\Sigma$  is a finite-length string of symbols in  $\Sigma$  [19]. The length of a string x is the number of symbols in x and is denoted by |x|. The Kleene closure of  $\Sigma$ , denoted by  $\Sigma^*$ , is the set of all finite-length strings of symbols including the null string  $\epsilon$ . The set of all strictly infinite-length strings of symbols is denoted as  $\Sigma^{\omega}$ . The string xy is the concatenation of strings x and y. Therefore, the null string  $\epsilon$  is the identity element of the concatenative monoid.

Definition 1 (PFSA): A probabilistic finite state automaton (PFSA) is a tuple  $G = (Q, \Sigma, \delta, q_0, \widetilde{\Pi})$ , where Q is a (nonempty) finite set, called the set of states;  $\Sigma$  is a (nonempty) finite set, called the input alphabet;  $\delta : Q \times \Sigma \to Q$  is the state transition function;  $q_0 \in Q$  is the start state;  $\widetilde{\Pi} : Q \times \Sigma \to$ [0, 1] is an output mapping, known as the probability morph function that specifies the state-specific symbol generation probabilities, and satisfies  $\forall q_i \in Q, \sigma \in \Sigma, \widetilde{\Pi}(q_i, \sigma) \geq 0$ , and  $\sum_{\tau \in \Sigma} \widetilde{\Pi}(q_i, \tau) = 1$ .

Notation 1: In the sequel, we would often use a matrix representation  $\Pi$  (denoted as the morph matrix) of the morph function, with the  $ij^{th}$  element given by  $\Pi(q_i, \sigma_j)$ . Note, that  $\Pi$  is, in general, a rectangular non-negative matrix with row sums equal to unity. Also, from a knowledge of the morph matrix  $\Pi$ , and the transition map  $\delta$ , one can compute the stochastic state transition matrix  $\Pi$ , as:

$$\Pi_{ij} = \sum_{\sigma_k:\delta(q_i,\sigma_k)=q_j} \widetilde{\Pi}(q_i,\sigma_k)$$
(1)

Note that  $\Pi$  is a square non-negative stochastic matrix.

Notation 2: The transition map  $\delta$  naturally induces an extended transition function  $\delta^* : Q \times \Sigma^* \to Q$  such that  $\delta^*(q, \epsilon) = q$  and  $\delta^*(q, x\tau) = \delta(\delta^*(q, x), \tau)$  for  $q \in Q$ ,  $x \in \Sigma^*$  and  $\tau \in \Sigma$ .

We assume that the underlying graph for the PFSA models considered in this paper is irreducible, *i.e.*, is strongly connected. This implies that the transition probability matrix  $\Pi$ is an irreducible stochastic matrix, and in particular, has an unique stationary distribution [20] irrespective of the the initial distribution. This assumption is motivated by the association of PFSA with emerging patterns in statistically stationary symbolic streams, because it makes little sense to represent such dynamical systems with models whose stationary behavior would depend on the initial state. Furthermore, the theoretical development in the sequel, necessitates this assumption for technical reasons.

*Notation 3:* In the sequel, we denote the PFSA constructed by directly compressing a symbol string  $\omega \in \Sigma^*$  as  $\mathbb{C}(\omega)$ .

The specific algorithm used is not important for the analysis presented in this paper.

*Definition 2 (\sigma-Algebra):* A collection  $\mathfrak{M}$  of subsets of a non-empty set X is said to be a  $\sigma$ -algebra [21] in X if  $\mathfrak{M}$  has the following properties:

- 1)  $X \in \mathfrak{M}$
- If A ∈ M, then A<sup>c</sup> ∈ M where A<sup>c</sup> is the complement of A relative to X, i.e., A<sup>c</sup> = X \ A
- 3) If  $A = \bigcup_{n=1}^{\infty} A_n$  and if  $A_n \in \mathfrak{M}$  for  $n \in \mathbb{N}$ , then  $A \in \mathfrak{M}$ .

Definition 3 (Measure): A (non-negative) measure is a countably additive function  $\mu$ , defined on a  $\sigma$ -algebra  $\mathfrak{M}$ , whose range is  $[0, \infty]$ . Countable additivity means that if  $\{A_i\}$  is a pairwise disjoint countable collection of members of  $\mathfrak{M}$ , then  $\mu(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mu(A_i)$ 

Definition 4 (Probability Measure): A probability measure on a non-empty set with a specified  $\sigma$ -algebra  $\mathfrak{M}$  is a finite (non-negative) measure on  $\mathfrak{M}$ . Although not required by the theory, a probability measure is defined to have the unit interval [0, 1] as its range.

Definition 5 (Measure Space): A probability measure space is a triple  $(X, \mathfrak{M}, p)$  where X is a non-empty set,  $\mathfrak{M}$ is a  $\sigma$ -algebra in X, and p is a finite non-negative measure on  $\mathfrak{M}$ .

Definition 6 ( $\sigma$ -Algebra for Symbolic Strings): Given an alphabet  $\Sigma$ , the set  $\mathfrak{B}_{\Sigma} \triangleq 2^{\Sigma^{\star}}\Sigma^{\omega}$  is defined to be the  $\sigma$ -algebra generated by the set  $\{L : L = x\Sigma^{\omega} \text{ where } x \in \Sigma^{\star}\}$ , i.e., the smallest  $\sigma$ -algebra on the set  $\Sigma^{\omega}$ , which contains the set  $\{L : L = x\Sigma^{\omega} \text{ s.t. } x \in \Sigma^{\star}\}$ .

For brevity, the probability  $p(x\Sigma^{\omega})$  is denoted as  $p(x), \forall x \in \Sigma^*$  in the sequel. In other words, p(x) is the probability of the occurrence of all the strings with x as a prefix.

Definition 7 (Probabilistic Nerode Relation): Given an alphabet  $\Sigma$ , any two strings  $x, y \in \Sigma^*$  are said to satisfy the probabilistic Nerode relation  $\mathcal{N}_p$  on a probability space  $(\Sigma^{\omega}, \mathfrak{B}_{\Sigma}, p)$ , denoted by  $x\mathcal{N}_p y$ , if either of the following conditions is true:

1) 
$$p(x) = p(y) = 0;$$
  
2)  $\forall \sigma \in \Sigma^{\star}, \frac{p(x\sigma)}{p(x)} = \frac{p(y\sigma)}{p(y)}$  provided that  $p(x) \neq 0, p(y) \neq 0.$ 

It has been proved in [9] that the probabilistic Nerode relation defined above is a right-invariant equivalence relation [19] which means that if two strings x, y are equivalent, so are any right extensions of the strings, *i.e.*,

$$\forall x, y, u \in \Sigma^{\star}, \ x \mathcal{N}_p y \Rightarrow x u \mathcal{N}_p y u$$

In the sequel, this is referred to as probabilistic Nerode equivalence and we denote the Nerode equivalence class of a string x on  $\Sigma^*$  by  $[x]_p$ , i.e.,  $[x]_p = \{z \in \Sigma^* : x \mathcal{N}_p z\}$ . The right invariance property induces the notion of states and hence is crucial to the definition of probabilistic state machines; by this property two equivalent strings have probabilistically indistinguishable future evolution and therefore can be visualized as terminating on the same state as seen in Figure 2a. In this context, we make the following observation:



Fig. 2: Linguistic Concepts: (a) Concept of PFSA states from Probabilistic Nerode Equivalence: Nerode equivalent strings  $\omega_1, \omega_2$  have probabilistically indistinguishable future evolution, thus leading to the same state  $q_1$ . (b) Symbolic White Noise (See Eq. (2) for formal definition) for alphabet  $\Sigma = \{\sigma_0, \sigma_1, \sigma_2\}$ ; (c) for alphabet  $\Sigma = \{\sigma_0, \sigma_1\}$ 

A symbolic dynamical process has a probabilistic finite state description if and only if the corresponding Nerode equivalence has a finite index.

Definition 8 (Space of PFSA): The space of all PFSA over a given symbol alphabet is denoted by  $\mathscr{A}$  and the space of all probability measures p that induce a finite-index probabilistic Nerode equivalence on the corresponding measure space  $(\Sigma^{\omega}, \mathfrak{B}_{\Sigma}, p)$  is denoted by  $\mathscr{P}$ .

As expected, there is a close relationship between  $\mathscr{A}$  and  $\mathscr{P}$ , which is made explicit in the sequel.

Definition 9 (PFSA Map  $\mathbb{H}$ ): Let  $p \in \mathscr{P}$  and  $G = (Q, \Sigma, \delta, q_0, \widetilde{\Pi}) \in \mathscr{A}$ . The map  $\mathbb{H} : \mathscr{A} \to \mathscr{P}$  is defined as  $\mathbb{H}(G) = p$  such that the following condition is satisfied:

$$\forall x = \sigma_1 \cdots \sigma_r \in \Sigma^*,$$
$$p(x) = \widetilde{\Pi}(q_0, \sigma_1) \prod_{k=1}^{r-1} \widetilde{\Pi}(\delta^*(q_0, \sigma_1 \cdots \sigma_k), \sigma_{k+1})$$

where  $r \in \mathbb{N}$ , the set of positive integers.

Definition 10 (Right Inverse  $\mathbb{H}_{-1}$ ): The right inverse of the map  $\mathbb{H}$  is denoted by  $\mathbb{H}_{-1} : \mathscr{P} \to \mathscr{A}$  such that

$$\forall p \in \mathscr{P}, \ \mathbb{H}(\mathbb{H}_{-1}(p)) = p$$

An explicit construction of the map  $\mathbb{H}_{-1}$  is reported in [9] and is not presented in this paper, because we only require that such a map exists.

Definition 11 (Perfect Encoding): Given an alphabet  $\Sigma$ , a PFSA  $G = (Q, \Sigma, \delta, q_0, \widetilde{\Pi})$  is said to be a perfect encoding of the measure space  $(\Sigma^{\omega}, \mathscr{B}_{\Sigma}, p)$  if  $p = \mathbb{H}(G)$ .

There are possibly many PFSA realizations that encode the same probability measure on  $\mathscr{B}_{\Sigma}$  due to existence of non-minimal realizations and state relabeling; neither of them affect the underlying encoded measure. From this perspective, a notion of PFSA equivalence is introduced as follows:

Definition 12 (PFSA Equivalence): Two PFSA  $G_1$  and  $G_2$  are defined to be equivalent if  $\mathbb{H}(G_1) = \mathbb{H}(G_2)$ . In this case, we say  $G_1 = G_2$ .

Remark 1: In the sequel, a PFSA G implies the equivalence class of G, *i.e.*,  $\{P \in \mathscr{A} : \mathbb{H}(P) = \mathbb{H}(G)\}$ . This concept is similar to the equivalence class of almost everywhere equal functions being a unique vector in the  $L_r$ -space [21].

Definition 13 (Structural Equivalence): Two PFSA  $G_i = (Q_i, \Sigma, \delta_i, q_0^i, \widetilde{\Pi}_i) \in \mathscr{A}, i = 1, 2$ , are defined to have the equivalent (or identical) structure if  $Q_1 = Q_2, q_0^1 = q_0^2$  and  $\delta_1(q, \sigma) = \delta_2(q, \sigma), \forall q \in Q_1 \ \forall \sigma \in \Sigma$ .

Definition 14 (Synchronous Composition of PFSA): The binary operation of synchronous composition of two PFSA  $G_i = (Q_i, \Sigma, \delta, q_0^{(i)}, \widetilde{\Pi}_i) \in \mathscr{A}$  where i = 1, 2, denoted by  $\otimes : \mathscr{A} \times \mathscr{A} \to \mathscr{A}$  is defined as

$$G_1 \otimes G_2 = \left(Q_1 \times Q_2, \Sigma, \delta', (q_0^{(1)}, q_0^{(2)}), \widetilde{\Pi}'\right)$$

where  $\delta'$  and  $\Pi'$  is computed as follows:

$$\begin{aligned} \forall q_i \in Q_1, q_j \in Q_2, \sigma \in \Sigma, \\ \delta'((q_i, q_j), \sigma) &= (\delta_1(q_i, \sigma), \delta_2(q_j, \sigma)) \\ \widetilde{\Pi}'((q_i, q_j), \sigma) &= \widetilde{\Pi}_1(q_i, \sigma) \end{aligned}$$

*Remark 2:* In general, the operation  $\otimes$  of synchronous composition is non-commutative.

Proposition 1 (Synchronous Composition of PFSA): Let  $G_1, G_2 \in \mathscr{A}$ . Then,  $\mathbb{H}(G_1) = \mathbb{H}(G_1 \otimes G_2)$  and therefore  $G_1 = G_1 \otimes G_2$  in the sense of Definition 12.

*Proof:* See Theorem 4.5 in [9].

Synchronous composition of PFSA allows transformation of PFSA with disparate structures to non-minimal descriptions that have the same underlying graphs. This assertion is crucial for the development in the sequel, since any binary operation defined for two PFSA with an identical structure can be extended to the general case on account of Definition 14 and Proposition 1.

#### III. ABELIAN GROUP OF PFSA

This section shows that a subspace of PFSA can be assigned the algebraic structure of an abelian group. We first construct the abelian group on a subspace of probability measures, and then induce the group structure on this subspace of PFSA via the isomorphism between the two spaces.

Definition 15 (Restricted PFSA Space): Let  $\mathscr{A}^+ = \{G = (Q, \Sigma, \delta, q_0, \widetilde{\Pi}) : \widetilde{\Pi}(q, \sigma) > 0 \ \forall q \in Q \ \forall \sigma \in \Sigma \}$  that is a proper subset of  $\mathscr{A}$ . It follows that the transition map of any PFSA in the subset  $\mathscr{A}^+$  is a total function. We restrict the map  $\mathbb{H} : \mathscr{A} \to \mathscr{P}$  on a smaller domain  $\mathscr{A}^+$ , that is,  $\mathbb{H}^+ : \mathscr{A}^+ \to \mathscr{P}^+$ , i.e.,  $\mathbb{H}^+ = \mathbb{H}|_{\mathscr{A}^+}$ .

Definition 16 (Restricted Probability Measure): Let  $\mathscr{P}^+ \triangleq \{p \in \mathscr{P} : p(x) \neq 0, \forall x \in \Sigma^*\}$  that is a proper subset of  $\mathscr{P}$ . Each element of  $\mathscr{P}^+$  is a probability measure that assigns a non-zero probability to each string on  $\mathfrak{B}_{\Sigma}$ . Similar to Definition 15, we restrict  $\mathbb{H}_{-1}$  on  $\mathscr{P}^+$ , i.e.,  $\mathbb{H}_{-1}^+ = \mathbb{H}_{-1}|_{\mathscr{P}^+}$ .

Since we do not distinguish PFSA in the same equivalence class (See Definition 12), we have the following result.

Proposition 2 (Isomorphism of  $\mathbb{H}^+$ ): The map  $\mathbb{H}^+$  is an isomorphism between the spaces  $\mathscr{A}^+$  and  $\mathscr{P}^+$ , and its inverse is  $\mathbb{H}^+_{-1}$ .

Definition 17 (Abelian Operation on  $\mathscr{P}^+$ ): The addition operation  $\oplus$  :  $\mathscr{P}^+ \times \mathscr{P}^+ \to \mathscr{P}^+$  is defined by  $p_3 \triangleq p_1 \oplus p_2, \forall p_1, p_2 \in \mathscr{P}^+$  such that

1)  $p_3(\epsilon) = 1.$ 

2)  $\forall x \in \Sigma^* \text{ and } \tau \in \Sigma, \ \frac{p_3(x\tau)}{\sum_{\alpha \in \Sigma} p_1(x\alpha)p_2(x\alpha)} = \frac{p_1(x\tau)p_2(x\tau)}{\sum_{\alpha \in \Sigma} p_1(x\alpha)p_2(x\alpha)}$ 

 $p_3$  is a well-defined probability measure on  $\mathscr{P}^+$ , since

$$\begin{aligned} \forall x \in \Sigma^{\star}, \\ \Sigma_{\tau \in \Sigma} p_3(x\tau) &= \Sigma_{\tau \in \Sigma} \frac{p_1(x\tau) p_2(x\tau)}{\sum_{\alpha \in \Sigma} p_1(x\alpha) p_2(x\alpha)} p_3(x) = p_3(x) \end{aligned}$$

Proposition 3 (abelian Group of PFSA): The algebra  $(\mathscr{P}^+, \oplus)$  forms an abelian group.

*Proof:* Closure property and commutativity of  $(\mathcal{P}^+, \oplus)$  are obvious. The associativity, existence of identity and existence of inverse element are established next.

(1) Associativity i.e.  $(p_1 \oplus p_2) \oplus p_3 = p_1 \oplus (p_2 \oplus p_3)$ . We note, that  $\forall x \in \Sigma^*, \tau \in \Sigma$ ,

$$\frac{((p_1 \oplus p_2) \oplus p_3)(x\tau)}{((p_1 \oplus p_2) \oplus p_3)(x)} = \frac{(p_1 \oplus p_2)(x\tau)p_3(x\tau)}{\sum_{\beta \in \Sigma} (p_1 \oplus p_2)(x\beta)p_3(x\beta)}$$

$$= \frac{\frac{p_1(x\tau)p_2(x\tau)}{\sum_{\alpha \in \Sigma} p_1(x\alpha)p_2(x\alpha)}p_3(x\tau)}{\sum_{\beta \in \Sigma} \frac{p_1(x\tau)p_2(x\tau)}{\sum_{\alpha \in \Sigma} p_1(x\alpha)p_2(x\alpha)}p_3(x\beta)}$$

$$= \frac{p_1(x\tau)p_2(x\tau)p_3(x\tau)}{\sum_{\beta \in \Sigma} p_1(x\beta)p_2(x\beta)p_3(x\beta)}$$

$$= \frac{p_1(x\tau)\frac{p_2(x\tau)p_3(x\tau)}{\sum_{\alpha \in \Sigma} p_2(x\alpha)p_3(x\alpha)}}{\sum_{\beta \in \Sigma} p_1(x\beta)\frac{p_2(x\beta)p_3(x\beta)}{\sum_{\alpha \in \Sigma} p_2(x\alpha)p_3(x\alpha)}}$$

$$= \frac{p_1(x\tau)(p_2 \oplus p_3)(x\tau)}{\sum_{\beta \in \Sigma} p_1(x\beta)(p_2 \oplus p_3)(x\beta)}$$

$$= \frac{(p_1 \oplus (p_2 \oplus p_3))(x\tau)}{(p_1 \oplus (p_2 \oplus p_3))(x)}$$

(2) Existence of identity: Let us introduce a probability measure  $\mathbf{i}_{\circ}$  of symbol strings such that:

$$\forall x \in \Sigma^{\star}, \ \mathbf{i}_{\circ}(x) = \left(\frac{1}{|\Sigma|}\right)^{|x|}$$
 (2)

where |x| denotes the length of the string x. Then,  $\forall \tau \in \Sigma$  that  $\frac{\mathbf{i}_0(x\tau)}{\mathbf{i}_0(x)} = \frac{1}{|\Sigma|}$ . For a measure  $p \in \mathscr{P}^+$  and  $\forall \tau \in \Sigma$ ,

$$\frac{(p \oplus \mathbf{i}_{\circ})(x\tau)}{(p \oplus \mathbf{i}_{\circ})(x)} = \frac{p(x\tau)\mathbf{i}_{\circ}(x\tau)}{\sum_{\alpha \in \Sigma} p(x\alpha)\mathbf{i}_{\circ}(x\alpha)} = \frac{p(x\tau)}{p(x)}$$

This implies that  $p \oplus \mathbf{i}_{\circ} = \mathbf{i}_{\circ} \oplus p = p$  by Definition 17 and by commutativity. Therefore,  $\mathbf{i}_{\circ}$  is the identity of the monoid  $(\mathscr{P}^+, \oplus)$ .

(3) Existence of inverse:  $\forall p \in \mathscr{P}^+$ ,  $\forall x \in \Sigma^*$  and  $\forall \tau \in \Sigma$ , let -p be defined by the following relations:

$$(-p)(\epsilon) = 1 \tag{3}$$

$$\frac{(-p)(x\tau)}{(-p)(x)} = \frac{p^{-1}(x\tau)}{\sum_{\alpha \in \Sigma} p^{-1}(x\alpha)} \tag{4}$$

Then, we have:

$$\frac{(p\oplus(-p))(x\tau)}{(p\oplus(-p))(x)} = \frac{p(x\tau)(-p)(x\tau)}{\sum_{\alpha\in\Sigma} p(x\alpha)(-p)(x\alpha)} = \frac{1}{|\Sigma|}$$
(5)

This gives  $p \oplus (-p) = \mathbf{i}_{\circ}$  which completes the proof.

In the sequel, we denote the zero-element  $\mathbf{i}_{\circ}$  of the abelian group  $(\mathscr{P}^+, \oplus)$  as the *symbolic white noise*. The concept of symbolic white noise has been illustrated in Figure 2b and 2c.

### A. Explicit Computation of the abelian Operation $\oplus$

The isomorphism between  $\mathscr{P}^+$  and  $\mathscr{A}^+$  (See Proposition 2) induces the following abelian operation on  $\mathscr{A}^+$ .

Definition 18 (Addition Operation on PFSA): Given any  $G_1, G_2 \in \mathscr{P}^+$ , the addition operation  $+ : \mathscr{A}^+ \times \mathscr{A}^+ \to \mathscr{A}^+$  is defined as:

$$G_1 + G_2 = \mathbb{H}^+_{-1}(\mathbb{H}^+(G_1) \oplus \mathbb{H}^+(G_2))$$

If the summand PFSA have identical structure (i.e., their underlying graphs are identical), then the explicit computation of this sum is stated as follows.

Proposition 4 (PFSA Addition): If two PFSA  $G_1, G_2 \in \mathcal{A}^+$  are of the same structure, i.e.,  $G_i = (Q, \Sigma, \delta, q_0, \widetilde{\Pi}_i), i = \{1, 2\}$ , then we have  $G_1+G_2 = (Q, \Sigma, \delta, q_0, \widetilde{\Pi})$  where

$$\widetilde{\Pi}(q,\sigma) = \frac{\Pi_1(q,\sigma)\Pi_2(q,\sigma)}{\sum_{\alpha\in\Sigma}\widetilde{\Pi}_1(q,\alpha)\widetilde{\Pi}_2(q,\alpha)}$$
(6)

*Proof:* Let  $p_i = \mathbb{H}^+(G_i)$ ,  $i = \{1, 2\}$  and since  $G_1, G_2$  have the same structure, we have from Eq. (9):

$$\forall \sigma \in \Sigma, \forall x \text{ s.t. } \delta^{\star}(q_0, x) = q \in Q,$$
  
$$\frac{p_i(x\sigma)}{p_i(x)} = \widetilde{\Pi}_i(\delta^{\star}(q_0, x), \sigma) = \widetilde{\Pi}_i(q, \sigma)$$
(7)

Now, by Definition 17 and Definition 9,

$$\widetilde{\Pi}(q,\sigma) = \frac{(p_1 \oplus p_2)(x\sigma)}{(p_1 \oplus p_2)(x)} = \frac{p_1(x\sigma)p_2(x\sigma)}{\sum_{\alpha \in \Sigma} p_1(x\alpha)p_2(x\alpha)}$$
$$= \frac{\frac{p_1(x\sigma)p_2(x\sigma)}{p_1(x)p_2(x)}}{\sum_{\alpha \in \Sigma} \frac{p_1(x\alpha)p_2(x\alpha)}{p_1(x)p_2(x)}} = \frac{\widetilde{\Pi}_1(q,\sigma)\widetilde{\Pi}_2(q,\sigma)}{\sum_{\alpha \in \Sigma} \widetilde{\Pi}_1(q,\alpha)\widetilde{\Pi}_2(q,\alpha)}$$

The extension to the general case is achieved by using synchronous composition of probabilistic machines.

Proposition 5 (PFSA Addition (General case)): Given two PFSA  $G_1, G_2 \in \mathscr{A}^+$ , the sum  $G_1+G_2$  is computed via Proposition 4 and Definition 14 as follows:

$$G_1 + G_2 = (G_1 \otimes G_2) + (G_2 \otimes G_1) \tag{8}$$

*Proof:* Noting that  $G_1 \otimes G_2$  and  $G_2 \otimes G_1$  have the same structure up to state relabeling, it follows from Proposition 1:

$$\mathbb{H}^+(G_1 + G_2) = \mathbb{H}^+(G_1) \oplus \mathbb{H}^+(G_2) \quad \text{(See Definition 18)} \\
= \mathbb{H}^+(G_1 \otimes G_2) \oplus \mathbb{H}^+(G_2 \otimes G_1) \\
= \mathbb{H}^+\left((G_1 \otimes G_2) + (G_2 \otimes G_1)\right)$$

which completes the proof.

*Example 1:* Let  $G_1$  and  $G_2$  be two PFSA with identical structures, such that the probability morph matrices are:

$$\widetilde{\Pi}_1 = \begin{pmatrix} 0.2 & 0.8\\ 0.4 & 0.6 \end{pmatrix}$$
 and  $\widetilde{\Pi}_2 = \begin{pmatrix} 0.1 & 0.9\\ 0.6 & 0.4 \end{pmatrix}$  (9)

Then the  $\widetilde{\Pi}$ -matrix for the sum  $G_1+G_2$ , denoted by  $\widetilde{\Pi}_{12}$ , is

$$\widetilde{\Pi}_{12} = \begin{pmatrix} 0.1 \times 0.2 & 0.9 \times 0.8 \\ 0.6 \times 0.4 & 0.4 \times 0.6 \end{pmatrix} \xrightarrow[rows]{Normalize} \begin{pmatrix} 0.027 & 0.973 \\ 0.5 & 0.5 \end{pmatrix}$$

#### IV. A MACHINE REPRESENTATION OF PFSA SUM

In this section, we investigate the implementation of the sum of two PFSA by a sequentially controlled interaction of individually generated symbol strings, which form the conceptual basis of designing a semantic annihilator. Referring to Figure 3, we will call this the Plus-machine.



Fig. 3: Sum of two PFSA: The Plus-machine  $\mathcal{M}(G_1+G_2)$ . The machines generate symbols independently, but is allowed to change states only if the generated symbols match.

## A. Functional Description of the Plus-Machine

For a given pair of PFSA  $G_1$  and  $G_2$ , the Plus-machine denoted as as  $\mathcal{M}(G_1+G_2)$  has the following components:

• Copies of the component machines  $G_1$  and  $G_2$ . We assume, without loss of generality, that  $G_1$  and  $G_2$  have the same structure (See Definition 14), since this can be always arranged (See Proposition 1).

• A logical AND gate AND :  $\Sigma \times \Sigma \rightarrow \{0, 1\}$  which operates as follows:

$$\forall \sigma_i, \sigma_j \in \Sigma, \ \sigma_i \text{ AND } \sigma_j = \begin{cases} 0 \text{ (false)}, & \text{if } \sigma_i \neq \sigma_j \\ 1 \text{ (true)}, & \text{otherwise} \end{cases}$$

#### B. Operational Description of the Plus-Machine

The + machine  $\mathcal{M}(G_1+G_2)$  operates as follows:

- Each of the component machines,  $G_1$  and  $G_2$ , is initialized to the same state  $q_0$  in the underlying graph.
- Each of the component machines, G<sub>1</sub> and G<sub>2</sub>, operates in a statistically independent manner to generate symbols from the alphabet Σ.
- However, to activate a state transition, the generated symbols must be passed through the AND gate, upon which they must yield a true output. Formally,

$$\begin{aligned} \forall \sigma_i, \sigma_j \in \Sigma, q_i, q_j \in Q, \\ \delta \Big( (q_i, q_j) (\sigma_i, \sigma_j) \Big) &= \begin{cases} & (q_i, q_j), \\ & \text{if } \sigma_i \text{ AND } \sigma_j = 0 \\ & \left( \delta(q_i, \sigma_i), \delta(q_j, \sigma_j) \right), \\ & \text{otherwise} \end{cases} \end{aligned}$$

• The machine is assumed to function inside a "black box", with an external observer. The observable output string generated as follows: A generated symbol is observable if and only if it causes a sMargtate transition.

The sequential functioning of  $\mathcal{M}(G_1+G_2)$  is illustrated in Figure 3. We have the following result:

Proposition 6 (Semantic Compression): For a given pair of PFSA  $G_1, G_2$ , if the output string from the  $\mathscr{M}(G_1+G_2)$  is denoted as  $x \in \Sigma^{\omega}$ , then, the PFSA  $\mathbb{C}(x)$  obtained by semantically compressing x is given by the sum  $G_1+G_2$ .

*Proof:* It follows from the functional description, and the following considerations:

- 1) The component machines  $G_1$  and  $G_2$  are always state synchronized (follows from operational description).
- 2) The components generate symbols in a statistically independent manner.
- 3) The probability for  $\mathcal{M}(G_1+G_2)$  to emit a particular symbol  $\sigma \in \Sigma$ , while being at state  $(q_i, q_i)$ , (*i.e.* both components are at state  $q_i$ ), is given by the probability of generating  $\sigma$  simultaneously (and independently) by **Sensed** both components; and the probability of the stage populate metrical symbol (marginalized by the probability of generating identical symbols on both machines) is :

$$\widetilde{\Pi}_{12}((q_i, q_i), (\sigma, \sigma)) = \frac{\widetilde{\Pi}_1(q_i, \sigma)\widetilde{\Pi}_2(q_i, \sigma)}{\sum_{\sigma}\widetilde{\Pi}_1(q_i, \sigma)\widetilde{\Pi}_2(q_i, \sigma)} \xleftarrow{Compound Event}{\leftarrow Marginalization}$$

which matches exactly with Proposition 4.

4) Since the internal states of *M*(G<sub>1</sub>+G<sub>2</sub>) are always of the form (q<sub>i</sub>, q<sub>i</sub>), it is straightforward to see that for any correct semantic compression algorithm, the structure of the identified PFSA matches with the component machines, G<sub>1</sub> and G<sub>2</sub>. The proof is now complete. It follows from Proposition 6, that the Plus-Machine can be used to annihilate information in the symbol string generated by a PFSA in the following manner:

$$G+H = \mathbb{H}_{-1}(\mathbf{i}_{\circ}) \Rightarrow \mathscr{M}(G+H) = \mathbb{H}_{-1}(\mathbf{i}_{\circ})$$
(10)

which implies that if G is the underlying PFSA for the sensed process, and we can compute H such that  $G + H = \mathbb{H}_{-1}(\mathbf{i}_{\circ})$ , and subsequently modify the incoming sensed data stream via the Plus-machine construction, we would end up with symbolic white noise in the output, which then can be identified easily. This, however, is not directly achievable in practice for the following reasons:

- 1) Impossibility of state synchronization with sensed stream.
- Impossibility of disabling state transitions in the sensed physical process.

The next section presents modifications to this basic construction to admit a physically realizable implementation of a semantic annihilator.

#### V. SEMANTIC ANNIHILATION

In this section, we assume that we are given a preidentified (during the training phase) pattern library  $\mathbb{G} \triangleq \{G^i : G^i \in \mathscr{A}^+\}$  containing a finite number of patterns of interest, represented as PFSA. We would construct a *semantic annihilator* for each pattern in  $\mathbb{G}$ , which would be used in online classification.

We need the following function that operates symbol-wise on streams, typically implementing a selective erasure of the two input streams ( $\epsilon$  is the null symbol, *i.e.*, the identity in the concatenative free monoid over the alphabet  $\Sigma$ ):

Definition 19 (Erasing Function): The erasing function  $\xi$ :  $\Sigma \times \Sigma \rightarrow \Sigma \bigcup \{\epsilon\}$  is defined as follows:

$$\xi(\sigma_1, \sigma_2) = \begin{cases} \sigma_1 & \text{if } \sigma_1 \text{ AND } \sigma_2 = 1\\ \epsilon & \text{otherwise} \end{cases}$$
(11)

#### A. Construction of the Semantic Annihilator



Fig. 4: The block design for a semantic annihilator

The component machines are set as follows:

• Let the  $G \in \mathbb{G}$  be one element of the pattern library.

• Construct the additive inverse for G, i.e. compute H s.t.

$$G + H = \mathbb{H}_{-1}^+(\mathbf{i}_\circ) \tag{12}$$

Let the state set for H be  $Q_H$ , and let  $|Q_H| = m$ .

 Create m copies for H, each initialized at a distinct state. Let H<sup>j</sup> be the copy of the PFSA H initialized at state q<sub>i</sub> ∈ Q<sub>H</sub>.

#### B. Operational Description of the Annihilator

The semantic annihilator operates as follows:

- 1) Read symbol  $\sigma_{sensor}$  from sensor
- 2) Independently generate symbols  $\sigma^j$  for each component  $H^j$ .
- 3) Transition each  $H^{j}$  using the same symbol  $\sigma_{sensor}$ .
- Construct *m* symbol streams ω<sup>j</sup> ∈ Σ<sup>\*</sup> : j ∈ {1, · · · , m} recursively using the erasing function ξ:

$$\omega_{UPDATE}^{j} = \omega^{j} \xi(\sigma_{sensor}, \sigma^{j}) \tag{13}$$

5) Check if any  $\omega^j$  is in fact symbolic white noise.

Next we present the main result (Proposition 7) which rigorously establishes the annihilation concept as a viable tool for pattern classification.

Proposition 7 (Main Result): At least one of the constructed streams  $\omega^j$  will semantically compress to symbolic white noise if and only if  $G+H = \mathbb{H}^+_{-1}(\mathbf{i}_\circ)$ , *i.e.*,

$$G + H^{j} = \mathbb{H}_{-1}^{+}(\mathbf{i}_{\circ}) \Leftrightarrow \exists j \ \left(\mathbb{C}(\omega^{j}) = \mathbb{H}_{-1}(\mathbf{i}_{\circ})\right)$$
(14)

(See Notation 3, and note that  $H^j$  is a copy of H initialized at the  $j^{th}$  state.)

**Proof:** (Left to Right:) Let the sensed process be generated by the underlying PFSA G, such that  $G+H = \mathbb{H}_{-1}(\mathbf{i}_{\circ})$ . We note that, by construction, there exists  $j_{\star}$  such that  $H^{j_{\star}}$ is always state synchronized with G. However, we only see symbols in the output stream  $\omega^{j_{\star}}$ , if the generated symbols are identical. It follows that, on compression  $\omega^{j_{\star}}$  would yield a modified PFSA (denote by  $G^{mod}$ ) with structure identical to G, but each row of the  $\Pi^{mod}$  matrix would be modified as follows:

$$\begin{split} \widetilde{\Pi}^{mod}(q_i,\sigma) &= \frac{\widetilde{\Pi}^G(q_i,\sigma)\widetilde{\Pi}^H(q_i,\sigma)}{\sum_{\sigma}\widetilde{\Pi}^G(q_i,\sigma)\widetilde{\Pi}^H(q_i,\sigma)} \\ &= \frac{\widetilde{\Pi}^G(q_i,\sigma)(\widetilde{\Pi}^G(q_i,\sigma))^{-1}K(q_i)}{\sum_{\sigma}\widetilde{\Pi}^G(q_i,\sigma)(\widetilde{\Pi}^G(q_i,\sigma))^{-1}K(q_i)} = \frac{1}{|\Sigma|} \end{split}$$

=

where  $K(q_i) = \sum_{\sigma} (\widetilde{\Pi}^G(q_i, \sigma))^{-1}$  is the normalizing constant, implying each row is identical and uniform which in turn implies that the identified model is symbolic white noise.

(**Right to Left:**) We show this by contradiction as follows: Let the sensed process is generated by G such that

$$G + H \neq \mathbb{H}_{-1}(\mathbf{i}_{\circ}) \tag{15}$$

and assume if possible, that there exists a constructed stream  $\omega^{j_*}$  which compresses to white noise. Although, we cannot assume that any  $H^j$  is state synchronized with G directly, we can consider the structure of both G and H to be represented (without loss of generality) by the one for  $G \times H$ , in which

they can be assumed to be synchronized (since state  $q_i$  in G and  $q_k$  in H can be mapped to state  $(q_i, q_k)$  in  $G \times H$ ). Denoting the machines modified by the synchronous product as  $G^{\times}$  and  $H^{\times}$  respectively, we note:

$$G^{\times} + H^{\times} = \mathbb{H}_{-1}(\mathbf{i}_{\circ})$$
 (By Assumption) (16)

But since  $\mathbb{H}(G^{\times}) = \mathbb{H}(G)$  and  $\mathbb{H}(H^{\times}) = \mathbb{H}(H)$  (since the underlying measures are not modified by going to a non-minimal realization via synchronous product), it follows:

$$\mathbb{H}(G) + \mathbb{H}(H) = \mathbf{i}_{\circ} \Rightarrow G + H = \mathbb{H}_{-1}(\mathbf{i}_{\circ})$$
(17)

which contradicts Eq. (15). This completes the proof.

Our key motivation for developing the annihilator was to be able to classify PFSA-based patterns faster and in a more robust fashion in real-time or near-real-time field operation. The argument for robustness is pretty obvious, since one state models, especially with uniform generation probabilities of the symbols (*i.e.* white noise) are the easiest ones to identify reliably for any compression algorithm. The argument for fast identification is more involved, primarily due to the fact that the annihilators selectively erase symbols leading to a decrease in the lengths of the observed symbol strings. Thus, although we only need to check for white noise in the outputs (which is significantly faster compared to directly identifying the original pattern), the fact that now we are dealing with a shorter string, implies that there is the possibility that the increased speed of identification is offset by the slow down of the rate of symbol production at the outputs. In the next section, we investigate this issue in more details, and derive rigorous performance guarantees.

#### VI. PERFORMANCE OF SEMANTIC ANNIHILATORS

We need the notion of a stationary distribution on the states of a given PFSA. This is in fact the stationary distribution for the stochastic transition probability matrix that can be computed from the connectivity graph and the symbol generation probabilities  $\tilde{\Pi}$ . Also, as stated before, we assume that all PFSA considered in this paper are irreducible, *i.e.*, have a strongly connected graph and hence yields an irreducible transition probability matrix.

Definition 20 (Stationary Distribution): For a given PFSA  $G = (Q, \Sigma, \delta, \widetilde{\Pi})$ , the stationary distribution  $\wp^G \in [0, 1]^{|Q|}$ ,  $\sum_i \wp_i^G = 1$  is defined as:

1) Construct the transition probability matrix  $\Pi$  as:

$$\forall q_i, q_j \in Q, \ \Pi \Big|_{ij} = \sum_{\sigma_k : \delta(q_i, \sigma) = q_j} \widetilde{\Pi}(q_i, \sigma)$$
(18)

Noting that Π is an irreducible stochastic matrix, compute the stationary distribution ℘<sup>G</sup> as the stationary probability distribution for the state transition matrix Π, *i.e.*, ℘<sup>G</sup> is the unique sum-normalized left eigenvector for Π satisfying ℘<sup>G</sup>Π = ℘<sup>G</sup>.

It follows from the irreducibility assumption, that the stationary distribution is unique for a given PFSA, and has no dependence on the initial state [20].

Notation 4: In the sequel, we use the notation:  $\wp^G_{\star} = \min_{q_i \in Q} \wp^G$ .

Also, our assumption of irreducible models leads to the following property for the stationary distribution:

Proposition 8: For any PFSA  $G = (Q, \Sigma, \delta, \Pi)$  with an irreducible underlying graph,  $\wp^G_* > 0$ .

**Proof:** Since  $\Pi$  is irreducible for such G, no non-negative left eigenvector of  $\Pi$  has a zero coordinate [20]. We want to estimate the shortening experienced by the sensed symbol strings due to the annihilation operation. We require the notion of the auxiliary PFSA  $\mathbb{A}(G)$  for a given PFSA G, which captures the simultaneous operation of the two machines, without erasure of the non-matching symbols.

Definition 21 (Auxiliary PFSA): For a given PFSA  $G = (Q, \Sigma, \delta, \widetilde{\Pi})$ , the auxiliary PFSA  $\mathbb{A}(G)$  is defined as:  $\mathbb{A}(G) = (Q, \Sigma \bigcup \Sigma', \delta^{\mathbb{A}}, \widetilde{\Pi}^{\mathbb{A}})$ , where  $\Sigma'$  is a *distinct* isomorphic copy of  $\Sigma$ , with  $\mathscr{I} : \Sigma \to \Sigma'$  being the (bijective) isomorphism, and:

$$\delta^{\mathbb{A}}(q_i, \sigma) = \begin{cases} \delta(q_i, \sigma) & \text{if } \sigma \in \Sigma \\ \delta(q_i, \mathscr{I}^{-1}\sigma) & \text{otherwise} \end{cases}$$
(19a)

$$\widetilde{\Pi}^{\mathbb{A}}(q_i, \sigma) = \begin{cases} \frac{1}{|\Sigma|} \mathscr{H}_i & \text{if } \sigma \in \Sigma \\ \widetilde{\Pi}(q_i, \sigma) - \frac{1}{|\Sigma|} \mathscr{H}_i & \text{otherwise} \end{cases}$$
(19b)

where  $\mathscr{H}_i$  is the harmonic mean of the  $i^{th}$  row of the  $\Pi$  matrix for G.

Proposition 9 (Properties of the Auxiliary Automaton):

The auxiliary automaton  $\mathbb{A}(G) = (Q, \Sigma \bigcup \Sigma', \delta^{\mathbb{A}}, \widetilde{\Pi}^{\mathbb{A}})$  has the following properties:

1)  $\wp^{\mathbb{A}(G)} = \wp^G$ 

 If H is the annihilator component that is correctly statesynchronized with G (where G is the correct PFSA corresponding to the annihilator), then A(G) correctly tracks H (state-wise and symbol-wise), if we consider that all σ ∈ Σ' are unobservable.

**Proof:** (1) follows immediately from Definition 21, by noting that the probability transition matrix is left unaltered in the construction of  $\mathbb{A}(G)$ . For (2), we note that the transition structure for H (and hence G) is recovered if we map  $\forall \sigma \in$  $\Sigma', \sigma \mapsto \mathscr{I}^{-1}\sigma$ . Next, we compute the probability  $p_{obs}(q_i, \sigma)$ of an observable  $\sigma$  when H is at state  $q_i$  as:

$$\forall \sigma \in \Sigma,$$

$$p_{obs}(q_i, \sigma) = \widetilde{\Pi}^G(q_i, \sigma) \widetilde{\Pi}^H(q_i, \sigma)$$

$$= \widetilde{\Pi}^G(q_i, \sigma) (\widetilde{\Pi}^G(q_i, \sigma))^{-1} \frac{1}{\sum_{\sigma} (\widetilde{\Pi}^G(q_i, \sigma))^{-1}}$$

$$= \frac{1}{|\Sigma|} \mathscr{H}_i$$
(20)

It follows from above, that the probability of an unobservable  $\sigma$  when H is at state  $q_i$  is given by:

$$\forall \sigma \in \Sigma', p_{unobs}(q_i, \sigma) = \widetilde{\Pi}^G(q_i, \sigma) - \frac{1}{|\Sigma|} \mathscr{H}_i \qquad (21)$$

which completes the proof.

Corollary 1: (To Proposition 9) If the length of the symbol string generated by G is denoted by  $L_G$ , then the length  $L_{ann}$  of the correctly annihilated string satisfies:

$$\lim_{L_G \to \infty} \frac{L_{ann}}{L_G} = \sum_{i}^{|Q|} \wp_i^G \mathscr{H}_i$$
(22)

*Proof:* We first note that the stationary frequency distribution  $\vartheta_{\Sigma}$  of the symbols (over alphabet  $\Sigma$ ) in a string generated by an arbitrary irreducible PFSA  $G_{arb}$  is given by:

$$\vartheta_{\Sigma} = \wp^{G_{arb}} \widetilde{\Pi}^{G_{arb}} \tag{23}$$

where the independence from the initial state follows from the irreducibility of  $G_{arb}$ . It then follows from Proposition 9, that the frequency distribution for the auxiliary automaton  $\mathbb{A}(G)$  is given by:

$$\vartheta_{\Sigma \bigcup \Sigma'} = \wp^{\mathbb{A}(G)} \widetilde{\Pi}^{\mathbb{A}(G)} = \wp^G \widetilde{\Pi}^{\mathbb{A}(G)}$$
(24)

which in turn implies (See Eq. (19b)) that the probability  $\lambda$  that any symbol generated by G is observable is given by:

$$\lambda = \sum_{i:\sigma_i \in \Sigma} \wp^G \widetilde{\Pi}^{\mathbb{A}(G)} \big|_i = \sum_i^{|Q|} \wp_i^G \mathscr{H}_i$$
(25)

This completes the proof.

Next, we define the coefficient of annihilation advantage:

Definition 22 (Coefficient of Annihilation Advantage): For a given PFSA  $G = (Q, \Sigma, \delta, \widetilde{\Pi})$ , let  $L_d$  be the string length required for direct identification via semantic compression, and let  $L_w$  be the string length required for identifying symbolic white noise. Then the Coefficient of Annihilation Advantage ( $\beta$ ) is defined as the ratio:

$$\beta = \frac{L_w}{L_d \sum_i^{|Q|} \wp_i^G \mathscr{H}_i} \tag{26}$$

*Remark 3:* It follows that when we have enough data to do a direct compression (of say length  $L_d$ ), then the expected length of the correctly annihilated string is given by  $L_d \sum_{i}^{|Q|} \varphi_i^G \mathscr{H}_i$ . Since we are required to identify symbolic white noise at the annihilator output, and if the string length for identification of symbolic white noise is denoted by  $L_w$ , then identification via semantic annihilation is advantageous if we have  $L_w < L_d \sum_{i}^{|Q|} \varphi_i^G \mathscr{H}_i$ , *i.e.*, if we have  $\beta < 1$ .

In the sequel, we compute upper bounds on the Coefficient of Annihilation Advantage  $\beta$ . In order to do so, it is obvious that we need to relate the lengths  $L_d$  and  $L_w$ . However, we wish to achieve this without reference to any specific algorithm for semantic compression, *i.e.*, we want the computed bounds to hold true irrespective of the manner we construct PFSA models out of symbol strings. We note that if we are to compress a string from a symbolic white noise, then we would expect to obtain a single state PFSA with equi-probable symbols. However, since we are talking about probabilistic generators, observing 1 symbol each from the alphabet would not be sufficient; or rather would be a very bad way of inferring that the symbol string is generated from the symbolic white noise. Since we assume that  $L_w$  is the string length required for the identification (for the particular algorithm, whichever that may be), the number of symbols of each label that we need to observe would be at least  $\frac{1}{|\Sigma|}L_w$ . In the sequel, we assume that for an arbitrary PFSA, the number of symbols of each label that we need to observe at each state must also be of at least this value  $\frac{1}{|\Sigma|}L_w$ , since the chosen algorithm apparently requires this many observations for statistical inference. Proposition 10 (Upper Bound for  $\beta$ ): For a given PFSA  $G = (Q, \Sigma, \delta, \widetilde{\Pi})$  with an irreducible underlying graph, which is not a realization of symbolic white noise, we have the following upper bounds:

1) 
$$\beta = \frac{L_w}{L_d \sum_i^{|Q|} \wp_i^G \mathscr{H}_i} < \frac{|\Sigma| |Q| \wp_\star^G}{\sum_{i=1}^{|Q|} \wp_i^G \mathscr{H}_i \sum_{j=1}^{|Q|} \mathscr{H}_j^{-1}} \triangleq \beta_1$$
(27a)

2) 
$$\beta_1 \leq \frac{|\Sigma|}{|Q|}$$
 (27b)

*Proof:* We note for each state  $q_i \in Q$ , we have:

$$\wp_{\star}^{G} L_{d} \min_{\sigma \in \Sigma} \widetilde{\Pi}(q_{i}, \sigma) \geqq \frac{1}{|\Sigma|} L_{w}$$
(28)

which follows from noting that  $\wp^G_{\star}L_d$  is at least the number of times state  $q_i$  is visited, and hence  $\wp^G_{\star}L_d \min_{\sigma \in \Sigma} \widetilde{\Pi}(q_i, \sigma)$ is at least the expected number of the least likely symbols generated at  $q_i$ . It follows:

$$L_{d} \ge \frac{L_{w}}{|\Sigma|\wp_{\star}^{G}} \frac{1}{\min_{\sigma \in \Sigma} \widetilde{\Pi}(q_{i}, \sigma)} \ge \frac{L_{w}}{|\Sigma|\wp_{\star}^{G}} \mathscr{H}_{i}^{-1}$$
(29)

$$\Rightarrow |Q|L_d > \frac{L_w}{|\Sigma|\wp^G_\star} \sum_{i=1}^{|Q|} \mathscr{H}_i^{-1}$$
(30)

$$\Rightarrow \beta = \frac{L_w}{L_d \sum_{i}^{|Q|} \wp_i^G \mathscr{H}_i} < \frac{|\Sigma||Q| \wp_\star^G}{\sum_{i=1}^{|Q|} \wp_i^G \mathscr{H}_i \sum_{j=1}^{|Q|} \mathscr{H}_j^{-1}} \quad (31)$$

Note the strict bound in the second inequality in Eq. (30) follows from the fact that G is not a realization of symbolic white noise, implying  $\exists q_i \in Q$ ,  $\mathscr{H}_i > \min_{\sigma \in \Sigma} \tilde{\Pi}(q_i, \sigma)$ , which completes the proof of Statement (1). For Statement (2), we first note that for any sequence of real numbers, the harmonic mean of the sequence is bounded above by its arithmetic mean. Hence, it follows that:

$$\begin{split} \frac{|Q|}{\sum\limits_{j=1}^{|Q|} \mathscr{H}_j} &\leq \frac{\sum\limits_{j=1}^{|Q|} \mathscr{H}_j}{|Q|} \Rightarrow \frac{1}{\sum\limits_{j=1}^{|Q|} \mathscr{H}_j \sum\limits_{j=1}^{|Q|} \mathscr{H}_j^{-1}} \leq \frac{1}{|Q|^2} \\ \Rightarrow \frac{1}{\sum\limits_{j=1}^{|Q|} \mathscr{H}_j^G} \frac{1}{\sum\limits_{j=1}^{|Q|} \mathscr{H}_j^{-1}}} &\leq \frac{1}{\sum\limits_{j=1}^{|Q|} \mathscr{H}_j^G} \frac{1}{\sum\limits_{j=1}^{|Q|} \mathscr{H}_j^{-1}}} \leq \frac{1}{\wp_\star^G \mathscr{H}_j} \frac{1}{\sum\limits_{j=1}^{|Q|} \mathscr{H}_j^{-1}}} \\ \Rightarrow \frac{|\Sigma||Q| \wp_\star^G}{\sum\limits_{i=1}^{|Q|} \wp_i^G} \frac{|Q|}{|Q|^2} \leq \frac{|\Sigma||Q| \wp_\star^G}{|Q|^2} = \frac{|\Sigma|}{|Q|} \\ \sum_{i=1}^{|Q|} \wp_i^G \mathscr{H}_i \sum\limits_{j=1}^{|Q|} \mathscr{H}_j^{-1}} \end{cases}$$

where the last step follows from the fact that irreducibility of G guarantees  $\wp_{\star}^{G} > 0$ . This completes the proof.

*Remark 4:* Note that although we assume that G is not a realization of symbolic white noise, we could not assume  $\exists q_i \in Q \ \wp_i^G > \wp_\star^G$ , which would have made the bound in

Statement (2) strict. The reason is that it is possible for a PFSA to have non-uniform symbol generation probabilities from some states, and yet end up having an uniform stationary distribution over its states. Note here that the property of being white (in the way we defined) has to do with the uniformity of the rows of the  $\Pi$  matrix, and not the stationary probabilities.

*Remark 5:* Proposition 10 is a strong result which implies that pattern classification via semantic annihilators is in fact advantageous for most PFSA encountered in practice, where typically one has a relatively small number of alphabet symbols and a possibly large number of machine states.

*Remark 6:* The bounds computed in Proposition 10 are not tight. Specifically, note that we neglected the fact that for a general PFSA, the string length for identification could be significantly greater due to issues relating to adequately achieving statistical stationarity of the observed stream. Thus even for models for which  $|\Sigma| > |Q|$ , it is not automatic that identification via annihilation is slower compared to direct compression.

# VII. SUMMARIZED ALGORITHMS FOR CLASSIFICATION VIA SEMANTIC ANNIHILATION

For each pattern in the specified pattern library, we first compute the inverse PFSA using Algorithm 1. Note, that step 4 in Algorithm 1 is well-defined (and does not encounter a divide-by-zero overflow) on account of our assumption of the restricted set  $\mathscr{A}^+$  (See Definition 15). Once the inverse patterns are computed, we need to set up the pattern-specific annihilators. Namely, for each pattern with |Q| states, we need |Q| copies of the inverse, each initialized to a distinct state, as stated before. The annihilation process requires sequential generation of symbols from these initialized PFSA, in accordance to their computed morph matrices. This is done as follows:

- Given the current state, we first select the corresponding row of the morph matrix, which specifies the probability distribution of the to-be-generated symbol over the alphabet.
- 2) We generate a symbol in accordance to this distribution.

There are standard reported ways of selecting an outcome in accordance to a specified distribution. We explicitly state one method involving a uniform random number generator with range [0, 1], which guarantees that the asymptotic timecomplexity of this choice is  $O(\log_2(|\Sigma|))$  (See Algorithm 2). The stated approach involves considering the cumulative distribution for the symbol. Since this has to be done each time a symbol is generated, we compute the cumulative morph matrix  $\widetilde{\Pi}_{cum}$  for the inverted models offline as follows:

Definition 23 (Cumulative Morph Matrix): The cumulative morph matrix  $\widetilde{\Pi}_{cum}$  is computed as follows:

$$\widetilde{\Pi}_{cum}\Big|_{ij} = \sum_{r=1}^{j} \widetilde{\Pi}\Big|_{ir}$$
(32)

The sequential symbol generation then uses rows of the cumulative morph matrix instead, as the input  $\nu$  to Algorithm 2. Algorithm 1: Computation of Inverse Pattern

**input** : PFSA  $G = (Q, \Sigma, \delta, \Pi)$ output: PFSA -G1 begin 2 for i = 1 : |Q| do for  $j = 1 : |\Sigma|$  do 3  $\widetilde{\Pi}'_{ij} = \frac{1}{\widetilde{\Pi}_{ij}};$ 4 5 endfor for  $j = 1 : |\Sigma|$  do 6  $\widetilde{\Pi}'_{ij} = \frac{\Pi'_{ij}}{\sum_{i} \widetilde{\Pi}'_{ij}}$ 7 endfor 8 endfor 9  $-G = (Q, \Sigma, \delta, \widetilde{\Pi}');$ 10 11 end

*Lemma 1:* Assuming that uniform random numbers in the range [0, 1] can be generated in constant time, the asymptotic time-complexity of Algorithm 2 is  $O(\log_2(|\Sigma|))$ .

*Proof:* We note that the possible number of choices for the to-be-generated symbol reduces by half its previous value in each iteration, implying that the number of iterations I satisfies:

$$2^{I} \leq |\Sigma| \Rightarrow I \leq \log_{2}(|\Sigma|)$$

which completes the proof.

Each copy of the inverted model in the annihilator accesses the sensed symbol, generates its own symbol in accordance to its current state, reports the symbol if there is a match, and finally updates the current state using the sensed symbol. The sequence of moves for each component (or copy) is enumerated in Algorithm 3. The |Q| reported streams are individually compressed to check if any is in fact white noise.

# VIII. ASYMPTOTIC COMPLEXITY ANALYSIS

We ascertain the asymptotic time complexity per sensed symbol of the online portion of the annihilation process, assuming the pattern corresponding to the annihilator is indeed present in the sensed stream. This analysis is important since the annihilator is processing a multi-stream input, and we need to convince ourselves that the work required per observed symbol is not too great, particularly since an overtly complex algorithm will be unable to handle high data rates.

We assume, as before, that random keys can be generated in constant time. Then, we have the following result:

Proposition 11: For a given PFSA  $G = (Q, \Sigma, \delta, \Pi)$ , the asymptotic time-complexity  $\mathscr{A}_G$  of classification via annihilation, per sensed symbol, is bounded as:

$$\mathscr{A}_G = O(\log_2(|\Sigma|)) \tag{33}$$

*Proof:* Time-complexity of identification  $C_1$ , considering all |Q| components of the annihilator, satisfies:

$$C_1 \leq T_R |Q| O(\log_2(|\Sigma|)) L_w C_0 \tag{34}$$

Algorithm 2: Probabilistic Symbol Generation **input** : Non-negative vector  $\nu$  of length  $|\Sigma|$  such that  $\nu|_i \leq \nu|_{i+1}, \nu|_{|\Sigma|} = 1$ , Alphabet  $\Sigma = \{\sigma_1, \cdots, \sigma_{|\Sigma|}\}$ output: Generated Symbol  $\sigma$ 1 begin Generate random key  $K_r \in [0, 1]$ ; 2 **lowerB** = 1; 3 **upperB** =  $|\Sigma|$ ; 4 while upperB > (lowerB + 1) do 5  $M = \left\lceil \frac{upperB-lowerB}{2} \right\rceil; /* Rounding to Next$ 6 Integer \*/ if  $K_r \leq \nu|_M$  then 7 upperB = M;8 else 9 10 **lowerB** = M; endif 11 12 endw if  $k_r \leq \nu|_{lowerB}$  then 13  $\sigma = \sigma_{\text{lowerB}};$ 14 15 else 16  $\sigma = \sigma_{\mathbf{upperB}};$ endif 17 18 end

Algorithm 3: Componentwise Annihilation Operation
<b>input</b> : Cumulative Morph Matrix $\widetilde{\Pi}_{cum}$ , Initial state
$q_{\text{init}}$ , Transition function $\delta$
output: Reported symbol stream
1 begin
2 Set current state $q_{curr} = q_{init}$ ;
/* Infinite loop */
3 while true do
4 Observe sensed symbol $\sigma_{sensed}$ ;
5 Generate random symbol $\sigma_{gen}$ using row of $\widetilde{\Pi}_{cum}$
corresponding to q <sub>curr</sub> ; /* Algorithm 2 */
6 if $\sigma_{\text{sensed}} == \sigma_{\text{gen}}$ then
7 Report $\sigma_{gen}$ ; /* Annihilated Stream
*/
8 endif
9 Update current state $q_{curr} = \delta(q_{curr}, \sigma_{sensed});$
10 endw
11 end

where  $T_R$  is the complexity of generating random keys in the range [0, 1],  $L_w$  is the string length required for identifying the symbolic white noise, and  $C_0$  is the time complexity of identifying symbolic white noise (using some given direct compression algorithm, and assuming we check for white noise on each stream after each sensed symbol observation). Hence, assuming that we have the total sensed string length as  $L_d$ , it follows that the time complexity per sensed symbol

#### PSfrag replacements



Fig. 5: PFSA models used for simulation, real numbers are symbol probabilities, integers are symbol labels: (a) M2 is a subshift of finite type having the structure of a suffix automaton, (b)S1 is a generalized even shift process, is a strictly Sofic system and has no synchronizing string

is bounded by:

$$\mathscr{A}_G \leq T_R C_0 |Q| O(\log_2(|\Sigma|)) \frac{L_w}{L_d}$$
(35a)

Using Definition 22, we have:

$$\mathscr{A}_{G} \leq T_{R}C_{0}|Q|O(\log_{2}(|\Sigma|))\beta\sum_{i}^{|Q|}\wp_{i}^{G}\mathscr{H}_{i}$$
 (35b)

Using Proposition 10, and noting  $\mathscr{H}_i \leq \frac{1}{|\Sigma|}$ , we have:

$$\mathscr{A}_{G} \leq T_{R}C_{0}|Q|O(\log_{2}(|\Sigma|))\frac{|\Sigma|}{|Q|}\sum_{i}^{|Q|}\wp_{i}^{G}\frac{1}{|\Sigma|}$$
(35c)

$$\Rightarrow \mathscr{A}_G \leq T_R C_0 O(\log_2(|\Sigma|)) \tag{35d}$$

Neglecting constant time factors, we have:

$$\mathscr{A}_G = O(\log_2(|\Sigma|)) \tag{35e}$$

Note that Eqn. (35a) is exact and not an averaging, since we do the same work every time a symbol is sensed. This completes the proof.

This is a strong result showing that the asymptotic timecomplexity of classification via annihilation, per symbol, is independent of complexity of the pattern and the number of PFSA states, and is only mildly dependent on the cardinality of the alphabet. Again, since the alphabet sizes are relatively small, and recalling that the proposed technique is provably faster compared to direct compression for most models, it follows that classification via annihilation is indeed highly advantageous for online operation.

# IX. VERIFICATION & VALIDATION

In this section, we validate the preceding theoretical developments in simulation. The PFSA models selected for generating the simulated symbol string is illustrated in Figure 5. The model (M2) shown in Figure 5(a) has the structure of a suffix automaton [1]. PFSA which have such structures are easier to identify from symbolic strings; primarily due to the existence of synchronizing strings [8] (strings which lead to a particular state irrespective of the starting state). For example, in the model M2, the states  $q_1, q_2, q_3, q_4$  can be easily seen to represent sets of symbol strings ending in 00,01,10,11 respectively [22], [1]. Although, the state structure is not available a priori to the compression algorithm, nevertheless, such so-called *d*-Markov machines [22] are significantly easier to identify. For examples of physical situations in anomaly detection which give rise to, or are effectively modeled by such d-Markov machines, the reader is referred to [23], [3]. The second model (S1) (Figure 5(b)) has only two states. However, S1 represents a generalization of the even-shift process, and its underlying graph is an example of a strictly Sofic shift process (and not a sub-shift of finite type [24]). Specifically, S1 does not have any synchronizing strings, *i.e.*, without the knowledge of the initial state one cannot infer the current state in a deterministic sense even from arbitrary long observation strings. Such models are significantly more difficult to identify (See [8] for discussion) for any of the compression algorithms reported in the literature [1], [22].

The algorithm used for direct compression is a modified version of CSSR [8], [7]. We compare PFSA models using the metric  $\theta$  proposed in [9], which is capable of computing distances between PFSA models with different underlying graphs (with identical alphabets). Note that while the output of the direct compression algorithm is compared against the original model, the annihilator output is compared against symbolic white noise.

In the two simulation runs reported, we generate data from the models and compare the string lengths required by direct compression versus classification via annihilation. The annihilators were constructed from the knowledge of the particular model used in the simulation using the formulation presented in Section V-A. Note, that the annihilation technique is not meant for identification of an unknown pattern (i.e. pattern identification), but detecting if the sensed symbol string is actually being generated by a known library pattern (*i.e.* pattern classification). Figure 6(a) illustrates the results for M2, and we note that the annihilator is significantly faster. The principal advantage of using annihilators is better illustrated for S1, where, for reasons explained above, the direct compression algorithm has a hard time, and has failed to correctly identify S1 even after 7000 symbols (convergence was observed at around 10000 symbols). The annihilator identifies S1 at just over 1000 symbols as shown in Figure 6(b). Finally, Figure 6(c) compares the response of a annihilator which does not correspond to the process generating the observed symbols, with one that does. Note, that in both cases, the responses are very stable; with the incorrect annihilator converging to 0.3 and the correct one to (very nearly) 0, which reflects a match.



Fig. 6: (a) Simulation result for model M2: Note that direct compression converges at 3500 symbols, while the annihilator converges at just over 1000 symbols. A few intermediate models identified by the direct compression algorithm is also shown. (b) Simulation result for model S1: Note that direct compression has failed to converge at 7000 symbols; (c) Comparison of distance of annihilated strings from symbolic white noise for correct and incorrect annihilators, *i.e.*, where we have a pattern match and where we do not.

In general, evaluation of such a pattern match involves using a specified detection threshold, the implications of which are discussed in the next section.

Implication of Proposition 10 is illustrated explicitly in Figure 7a (a snapshot of the annihilation process in the above described simulation runs), where we note that the annihilation process essentially erases symbols selectively in the incoming data stream, and hence yields a significantly shorter observed sequence. Although, we now have the relatively easier task of identifying if this annihilated sequence is indeed white; but even such an identification cannot be effectively done with too few symbols. Proposition 10 guarantees that the length shortening cannot offset this advantage in practical scenarios, where we are most likely to have more states than the total number of symbols in the alphabet.

# X. INTUITIVE INTERPRETATION & POTENTIAL APPLICATIONS

An important intuitive insight on why the annihilators are able to classify the streams faster can be given as follows: by avoiding direct compression of the observed symbol sequence, we are essentially solving a classification problem, which is, in general, easier compared to a full blown identication problem, involving discovery of new patterns. Direct compression is capable of telling us not only if there is a match, but also yields the new PFSA model of the observed sequence when there is no match with the existing templates. Annihilation only indicates the matching template if there is one, and indicates a "no match" otherwise. Thus, the increased efficiency is not surprising. This is particularly useful for templates that have no synchronizing strings (such as the model S1), where for





(b) Fig. 7: (a) Snapshot of the annihilation process (b) Pattern classification scheme in symbolized sensory data streams

direct compression, one needs to distinguish between the states using long observation sequences, that disambiguate possible future evolutions based on the small deviations in the observed probability distributions on the future strings. If the spectral gap of the corresponding Markov chain is small, the sequences required can turn out to be unacceptably long (since smaller the spectral gap, longer is the mixing time). This is what we see manifested in Figure 2b, where direct compression has a hard time. For the annihilation, such complexities are absent: the spectral gap does play a role in the degree of shortening of the annihilated sequence (See Proposition 10), but one always looks for symbolic white noise at the annihilator output, irrespective of the complexity of the template.

The idea of pattern classification via controlled information erasure may seem somewhat counter-intuitive at the first reading. However, the key notion exploited here has a clear analogue in communication theory, particularly in the theory of matched filters [25]. A matched filter is a theoretical construct (and not the name of a specific filter family) which processes a received signal to minimize the effect of noise, *i.e.* maximizes the signal to noise ratio (SNR), and simultaneously minimizes the probability of bit error rate (BER). It can be shown that, under the assumption of additive white Gaussian noise (AWGN) in the communication channels, an optimum filter for receiver-end demodulation exists, and is a function only of the transmitted pulse shape. Because of this direct relationship to the transmitted pulse, it is called a *matched filter*. The derivation of a classical matched filter is essentially based on a direct application of Schwartz inequality [21], and leads to a very simple and remarkable conclusion:

For AWGN channels, the signal to noise ratio is maximized when the impulse response of that filter is exactly a reversed and time delayed copy of the transmitted signal.

Since the bit error rate experienced by a signal during demodulation is a function of the signal to noise ratio [26], a matched filter which maximizes SNR will automatically provide the lowest possible BER. The analogy of semantic annihilation with matched filters is compelling: instead of using a timereversed copy of the signal template, we are using the symbol stream generated by an inverse probabilistic automata. Just as a matched filter functions by convolving the signal with its reversed and delayed copy, the annihilator carries out symbolwise comparisons between the given symbol stream, and the state-specific ones generated by the inverted template; erasing symbols that do not match. The fact that we can carry out this procedure in a deterministic fashion should not be surprising: the convolution in the case of matched filters is generally carried out using Fourier transforms (FFT), which is also a rather straightforward deterministic operation. In the latter case, the filtered signal must still be recognized, but this decisionmaking task is now significantly easier due to the filtrationenhanced SNR. In our case, the annihilator does not output an enhanced signal, but reduces it to white noise if the correct template is used. However, the task of recognizing symbolic white noise is significantly easier compared to recognizing the template pattern directly; thus reinforcing the analogy. The recognition of symbolic white noise does involve the use of a detection threshold, since in practical scenarios, we do not expect the signal and the template to match exactly, given finite-length observation sequences. Thus, when the distance between at least one of the PFSA models computed from the annihilator output falls within a pre-specified distance to the white noise model (in the sense of the PFSA metric  $\theta$  [9]), we conclude a positive match. Using arbitrarily small thresholds may require long data streams, and most likely will result in negative matches due to small noise-mediated mismatch between the streams.

The key application that the authors have in mind is pattern classification in symbolized (or quantized) sensory data streams. This particular approach of pattern detection in sensory data has been shown to be significantly more efficient to classical continuous domain techniques, exhibiting remarkable insensitivity to spurious noise and exogenous disturbances; primarily due to the quantization-mediated coarsegraining, and as a consequence of repeated recurrences of paths in the graph of the finite state machine with relatively few states and a large number of sample points in the (fast scale) time series data [22]. Recent applications of such PFSAbased pattern classification has been effectively applied to anomaly detection problems in complex electro-mechanical machines [23], and tracking targets via large-scale multi-modal urban sensor networks [27]. The basic philosophy is illustrated in Figure 7b. Continuous valued data from sensor(s) is quantized via an appropriately chosen partitioning scheme [3] to yield a symbolic sequence over a pre-specified alphabet (depending on the coarseness of the chosen partition). In the absence of annihilators, one is then required to algorithmically compress a sufficiently long symbolic sequence to extract the underlying causal generative model in the form of a probabilistic finite automata. The classifier is provided with a template library consisting of PFSA models that encode the pertinent patterns of interest. Once the observed sequence is compressed to a PFSA, this can then be compared against the individual library elements to compute a possible match. The compression algorithms, however, are often expensive; particularly if the underlying PFSA is not a subshift of finite type [24]. Annihilation offers a significantly simple solution, which skips the compression step altogether. The observed stream can be symbol-wise annihilated using the inverted templates in the library, requiring less data, and significantly simpler implementations.

A second promising application is the design of PFSA-based novel modulation-demodulation schemes for communication

over noisy channels. In this paper, we considered the special case where the symbol stream generated by a PFSA G is annihilated by the inverse model -G. However, in general, one can apply similar ideas to encode a stream from PFSA G using an encoding PFSA  $G_e$  as  $G+G_e$ , and demodulate by "adding" the inverse stream  $G+G_e+(-G_e) = G$ . Such avenues will be explored in future, where careful choice of the encoding PFSA may lead to greater resilience to noise corruption, or even to unauthorized message access.

#### XI. SUMMARY, CONCLUSIONS & FUTURE WORK

We defined an additive abelian group for probability measures on symbolic strings, which induces an abelian group on a slightly restricted set of PFSA. The defined PFSA sum is then used to formulate semantic annihilators, which identify pre-specified patterns of interest via perfect removal of all inter-symbol correlations from observed strings, turning them to symbolic white noise. This approach of classification via annihilation is shown to be advantageous, with theoretical guarantees, for a large class PFSA models. The results are supported by simulation experiments.

Future work will extend the formulation to models where not all symbols satisfy the condition that the generation probabilities are strictly non-zero from each model state. The effect of noise corruption on observed strings need to be investigated, with particular emphasis on the comparative effect of noisy observations on direct compression and semantic annihilation. Furthermore, implementation in actual experimental scenarios will further validate the proposed classification technique.

#### REFERENCES

- K. Murphy, "Passively learning finite automata," Santa Fe Institute, Tech. Rep., 1996. [Online]. Available: citeseer.ist.psu.edu/murphy95passively.html
- [2] D. Angluin and C. H. Smith, "Inductive inference: Theory and methods," ACM Comput. Surv., vol. 15, no. 3, pp. 237–269, 1983.
- [3] V. Rajagopalan and A.Ray, "Symbolic time series analysis via waveletbased partitioning," *Signal Processing*, vol. 86, no. 11, pp. 3309–3320, 2006.
- [4] S. Lucas and T. Reynolds, "Learning deterministic finite automata with a smart state labeling evolutionary algorithm," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 7, pp. 1063– 1074, July 2005.
- [5] A. Paz, Introduction to probabilistic automata (Computer science and applied mathematics). Orlando, FL, USA: Academic Press, Inc., 1971.
- [6] E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta, and R. Carrasco, "Probabilistic finite-state machines - part i," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 7, pp. 1013–1025, July 2005.
- [7] C. R. Shalizi, K. L. Shalizi, and J. P. Crutchfield, "An algorithm for pattern discovery in time series," *Technical Report, Santa Fe Institute*, October 2002. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0210025
- [8] C. R. Shalizi and K. L. Shalizi, "Blind construction of optimal nonlinear recursive predictors for discrete sequences," in AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence. Arlington, Virginia, United States: AUAI Press, 2004, pp. 504–511.
- [9] I. Chattopadhyay and A. Ray, "Structural transformations of probabilistic finite state machines," *International Journal of Control*, vol. 81, no. 5, pp. 820–835, May 2008.
- [10] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.
- "Learning [11] D. Heckerman and D. Geiger, Bayesian WA, Networks," Microsoft Research, Redmond, Tech. December MSR-TR-95-02, 1994 [Online]. Available: Rep. citeseer.ist.psu.edu/article/heckerman95learning.html

- [12] F. Jelinek, J. D. Lafferty, and R. L. Mercer, "Basic methods of probabilistic context free grammars," *Speech Recognition and Understanding*. *Recent Advances, Trends, and Applications*, vol. F75, pp. 35–360, 1992.
- [13] E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta, and R. Carrasco, "Probabilistic finite-state machines - part ii," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 7, pp. 1026–1039, July 2005.
- [14] A. Corazza and G. Satta, "Probabilistic context-free grammars estimated from infinite distributions," *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on*, vol. 29, no. 8, pp. 1379–1393, Aug. 2007.
- [15] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech processing," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [16] D. Kulp, D. Haussler, M. G. Reese, and F. H. Eeckman, "A generalized hidden markov model for the recognition of human genes in dna." AAAI Press, 1996, pp. 134–142.
- [17] A. Brazma, I. Jonassen, I. Eidhammer, and D. Gilbert, "Approaches to the automatic discovery of patterns in biosequences," *J Comput Biol.*, vol. 5, no. 2, pp. 279–305, 1998.
- [18] J. P. Crutchfield and K. Young, "Inferring statistical complexity," *Physical Review Letters*, vol. 63, pp. 105–108, 1989.
- [19] J. E. Hopcroft, R. Motwani, and J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, 2nd ed. Addison-Wesley, 2001.

- [20] A. Berman and R. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1994, corrected republication, with supplement, of work first published in 1979 by Academic Press.
- [21] W. Rudin, Real and Complex Analysis, 3rd ed. McGraw Hill, New York, 1988.
- [22] A. Ray, "Symbolic dynamic analysis of complex systems for anomaly detection," *Signal Processing*, vol. 84, no. 7, pp. 1115–1130, 2004.
- [23] S. Chin, A. Ray, and V. Rajagopalan, "Symbolic time series analysis for anomaly detection: A comparative evaluation," *Signal Processing*, vol. 85, 9, pp. 1859–1868, 2005.
- [24] D. Lind and M. Marcus, An Introduction to Symbolic Dynamics and Coding. Cambridge University Press, United Kingdom, 1995.
- [25] D. North, "An analysis of the factors which determine signal/noise discrimination in pulsed-carrier systems," *Proceedings of the IEEE*, vol. 51, no. 7, pp. 1016–1027, July 1963.
- [26] J. Proakis, *Digital Communications*, 4th ed. Boston, MA: McGraw-Hill Science/Engineering/Math, August 2000.
- [27] I. Chattopadhyay, Y. Wen, S. Phoha, and A. Ray, "Mathematical foundations of sensor network design based on linguistic informatics," in *American Control Conference*, Baltimore, MD, 2010.

