On the Effects of Noise and Speed on Computations

Bernard Delyon
INRIA/IRISA
Campus de Beaulieu
35042 Rennes
France

Oded Maler[†]
VERIMAG
B.P. 53x
38041 Grenoble
France
E-mail: maler@imag.fr

1993

*To appear in *Theoretical Computer Science*.

†This work was done while the author was at INRIA/IRISA, Campus de Beaulieu, 35042, Rennes, France

Abstract

In this paper we propose a model that captures the influence of noise and speed on the correct behavior of a computing device situated in a dynamic environment. Within this model we analyze the relation between structural properties of automata and their immunity to noise. We prove upper- and lower-bounds on the effect of noise for various classes of finite automata. In addition we show similar relatinoships between relative speeds of the automaton and the environment and the accuracy of computation. Our model, combining basic notions from algebraic automata theory and the theory of stochastic processes, can serve as a starting point for a rigorous theory of computational systems embedded in the real world.

1 Introduction

Traditional computer science models try to abstract away as many real-world features as possible. The external world appears in these models only after being converted into a time-less sequence of symbols written on the input tape of a Turing machine. Even if we consider sequential machines that are constrained to process the input in the same order as it arrives, still some simplifying assumption remain, and in particular:

1. The input symbols are immediately and precisely recognized.

2. The durations of the automaton transitions are negligible compared to the arrival rate of input symbols.

When we consider "real" computers embedded in a physical environment, as in the case of robots, controllers or signal processors, this idealization is inadequate. External events do not appear with "labels" taken from the machine's alphabet, but are rather computed approximately from noisy data. Moreover, they are not constrained to occur when the machine is "ready" to process them.

The goal of this paper is to build a framework for comparing the "ideal" behavior of a discrete computational device with its behavior in "realistic" situations. The ideal behavior is the one usually studied in theoretical computer science models, that is, the behavior of a transition system (the language it accepts, its associated sequential function, etc.) when all inputs are correctly interpreted and all state transitions are performed correctly with a negligible duration.

The bridge between the idealized and real world is built by introducing noise: with some probability the system takes a wrong transition. This noise can result from the physical properties of sensors, from limitations of classification algorithms, from unreliability of computational hardware or from insufficient speed of the computer with respect to the arrival rate of input symbols. Whatever the physical reason of the noise is, and no matter what its logical form is (omission, misclassification or duplication of symbols) in our model it is assumed to be reducible to a bound ϵ on the probability of taking the wrong transition.

The noise transforms the original deterministic computational system into a probabilistic one over the same set of states. The deviation of the noisy system from the original "normative" behavior is defined as the expected probability that these two systems are in different states given the same input sequence of external events.

The class of systems we consider are finite-state automata and our main result is in establishing the relation between the properties of the original automaton and its expected asymptotic behavior in the presence of noise. It turns out that some classes of automata are less sensitive to noise than others. These results are finally applied to the case where the speeds of the automaton and the environment are given in quantitative real time terms.

The significance of this work is in establishing a theoretical basis for the performance analysis of embedded systems and in linking together concepts and notions from automata theory, Markov processes and the theory of semi-groups.

It should be noted that unlike other works on fault-tolerant computations, ours is not concerned with the design of computer architectures that minimize the effect of noise on arbitrary computations. In contrast, we try to classify computational tasks according to their inherent immunity to noise, and in particular according to whether they can be performed in a satisfactory manner in spite of temporary errors during execution.

The paper is organized as follows: in section 2 we define formally the noisy version of an automaton and the distance between the ideal and the noisy versions. In section 3, we calculate an upper bound on this distance for a class of automata whose associated transformation semigroup contains a reset. In section 4 we give a lower-bound on this distance for the complementary class of automata. In section 5 we apply these results to real-time and in section 6 we conclude and mention briefly some relations with past and future work.

2 Ideal and Noisy Automata

2.1 The Effect of Noise

The essential behavior of a deterministic finite-state automaton (see [5] for standard definitions and notations) is the mapping of external sequences into internal states. When, for some reason, the automaton fails to take the correct transition (e.g., the current input arrived before the previous one is completely processed) we are at the risk that the intended relation between the input history and the internal state is no longer maintained. When we have a bound on the probability of such a fault, the situation can be viewed as if we work with a noisy version of our intended automaton which is just a probabilistic automaton (in the sense of [9] and [10]. This notion is formalized below.

Definition 1 (Noisy Version) *Let $\mathcal{A} = (\Sigma, Q, \delta)$ be a deterministic automaton and let ϵ, $0 \leq \epsilon \leq 1$ be a probability. An ϵ-noisy version of $\mathcal{A}$ is any probabilistic automaton $\mathcal{A}' = (\Sigma, Q, \delta')$ where δ' is a time-invariant probabilistic transition function such that for every state q and input σ satisfying $\delta(q, \sigma) = q'$ we have $Pr\{\delta'(q, \sigma) = q'\} \geq 1 - \epsilon$* and consequently, $\sum_{q'' \neq q'} Pr\{\delta'(q, \sigma) = q''\} \leq \epsilon$.

An example of an automaton and one of its ϵ-noisy versions is depicted in figure **??**-a. This particular pattern of noise is associated with a probability ϵ of omitting an input symbol and thus not performing a transition. Some insight concerning the nature of this perturbation can be gained by employing the alternative description via state-vector and transition-matrix terminology. The current state can be represented by a probabilistic vector $\bar{v}$ where v_i is the probability that the current state is q_i, and every input letter σ in a stochastic automaton can be associated with a probabilistic transition matrix M^σ such that $M^\sigma_{ij} = Pr\{\delta(q_i, \sigma) = q_j\}$. In the deterministic case we are restricted to $0-1$ vectors and matrices. The matrices for $\mathcal{A}$ and $\mathcal{A}'$ appear in figure **??**-b.

< **Figure ??** >

2.2 The Difference Between Behaviors

A useful conceptual tool for describing the joint behavior of two automata reacting to the same input is their direct product.

Definition 2 (Product of Probabilistic Automata) *Let $\mathcal{A}_1 = (\Sigma, Q_1, \delta_1)$ and $\mathcal{A}_2 = (\Sigma, Q_2, \delta_2)$ be two probabilistic automata. Their cartesian product $\mathcal{A}_1 \times \mathcal{A}_2$ is a probabilistic automaton $\mathcal{A} = (\Sigma, Q, \delta)$ where $Q = Q_1 \times Q_2$ and δ is a probabilistic transition function such that for every $(q_1, q_2), (p_1, p_2) \in Q$ and $\sigma \in \Sigma$*

$$Pr\{\delta((q_1, q_2), \sigma) = (p_1, p_2)\} = Pr\{\delta_1(q_1, \sigma) = p_1\} \cdot Pr\{\delta_2(q_2, \sigma) = p_2\}$$

In the special case of deterministic automata this definition reduces to the usual direct product. The product of $\mathcal{A}$ and $\mathcal{A}'$ from figure **??** appears in figure **??**-a. In terms of matrices this is equivalent to the following construction: for every $\sigma \in \Sigma$, let M_1^σ and M_2^σ be the corresponding matrices in $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively. The matrix associated with σ in $\mathcal{A}_1 \times \mathcal{A}_2$ is defined as $M^\sigma = M_1^\sigma \otimes M_2^\sigma$ where $\otimes$ denotes the Kronecker product of the two matrices. The resulting matrices for $\mathcal{A}' \times \mathcal{A}$ in our example appear in figure **??**-b.

< **Figure ??** >

Now we have a probabilistic automaton where all the trajectories ending in "diagonal" states, i.e., states in $\{(q,q) : q \in Q\}$, represent a good behavior ($\mathcal{A}'$ agrees with $\mathcal{A}$) while other compound states indicate disagreement between the two. In order to quantify this difference we associate with each individual sequence $w \in \Sigma^*$ a distance measure ρ_w defined as

$$\rho_w(\mathcal{A}, \mathcal{A}') = Pr\{\delta(q_0, w) \neq \delta'(q_0, w)\} \tag{1}$$

which indicates the probability of reaching a non-diagonal configuration after reading w starting at an initial state q_0. Next we consider, for every k, a probability distribution μ_k on all the input sequences of length k. This induces an expected distance measure ρ^k defined as:

$$\rho^k(\mathcal{A}, \mathcal{A}') = \sum_{w \in \Sigma^k} \mu_k(w) \cdot \rho_w(\mathcal{A}, \mathcal{A}') \tag{2}$$

indicating the expected probability of error after reading k input symbols. Finally we consider $\{\mu_k\}_{k=1}^{\infty}$ as a sequence of probability distributions on $\{\Sigma^k\}_{k=1}^{\infty}$. The asymptotic expected distance between $\mathcal{A}$ and $\mathcal{A}'$ is

$$\rho(\mathcal{A}, \mathcal{A}') = \lim_{k \to \infty} \rho^k(\mathcal{A}, \mathcal{A}') \tag{3}$$

It is reasonable to assume additional restrictions on $\{\mu_k\}$, such as

$$\mu_k(w) = \sum_{\sigma \in \Sigma} \mu_{k+1}(w\sigma)$$

In the rest of this paper we will assume μ_k as induced by a Bernoulli process: for every position in the sequence, the probability of a letter $\sigma_i \in \Sigma$ is a fixed probability p_i. We will denote $\min\{p_i\}$ by $\hat{p}$. Following this assumption, the expected behavior of $\mathcal{A} \times \mathcal{A}'$ can be modeled as an ordinary input-less Markov chain ([6]) where the labels on the transitions are replaced by their corresponding probabilities. In matrix terms we replace the input-dependent matrices $M^{\sigma_1}, M^{\sigma_2}, \ldots, M^{\sigma_m}$ by a common averaged matrix

$$M = \sum_{i=1}^{m} p_i \cdot M^{\sigma_i}$$

representing the expected transition probabilities. In our example, by assuming probabilities p for a and $1-p$ for b, the probabilistic automaton of figure **??** becomes the chain of figure **??**.

< **Figure ??** >

The distance between $\mathcal{A}$ and $\mathcal{A}'$, now becomes:

$$\rho(\mathcal{A}, \mathcal{A}') = \limsup_{k \to \infty} \bar{v}_0 \cdot M^k \cdot \bar{u} \tag{4}$$

where $\bar{v}_0$ is a row vector indicating the initial state and $\bar{u}$ is a $0-1$ vector with 1's in the entries corresponding to non-diagonal states in $\mathcal{A}' \times \mathcal{A}$. The limit exists if M is aperiodic (which is the case for most ϵ-noisy versions). The question we answer in this paper is the following: *What is the relation between the structure of $\mathcal{A}$ and $\rho(\mathcal{A}, \mathcal{A}')$?*

3 Robustness of Synchronizing Automata

The essential observation underlying our results is the following: Suppose that for two states $q, q' \in Q$ there exists a sequence $w \in \Sigma^+$ that merges them, i.e., $\delta(q, w) = \delta(q', w) = q''$. Then, whenever we are in an error configuration (q, q') or (q', q) in $\mathcal{A} \times \mathcal{A}'$, an application of w will bring us back to a correct diagonal configuration (q'', q''), and the effect of the past error will be cancelled. If we had such a merging sequence for every pair of states then we could recover from every error with a high probability.

Definition 3 (Synchronizing Automata) *An automaton is synchronizing if there exists a sequence $w \in \Sigma^*$ and a state q' such that for all $q \in Q$, $\delta(q, w) = q'$.*

We call such sequence a *reset* – some authors (e.g., [2]) use the term synchronizing sequence, or synchronizer. Note that if w is a reset then so is wu for every $u \in \Sigma^*$. By $\ell(\mathcal{A})$ we denote the length of the minimal reset in $\mathcal{A}$ if there exists one or ∞ otherwise. It can be shown that $\ell(\mathcal{A}) < \infty$ implies $\ell(\mathcal{A}) < |Q|^3$.

Claim 1 *An automaton is synchronizing if and only if every pair of states has a merging sequence.*

Proof: One direction is obvious by the definition of resets. The other can be proved inductively based on the following argument: Suppose w merges q_1 and q_2 but not necessarily q_3, that is, $\delta(q_1, w) = \delta(q_2, w) = q$ and $\delta(q_3, w) = q'$. But q and q' have a merging sequence w' so $\delta(q_1, ww') = \delta(q_2, ww') = \delta(q_3, ww')$. Thus, if any pair of states has a merging sequence we can construct a global reset. ∎

Definition 4 (Reset Probability) *For every $k > 0$ we let $R(k)$ denote the probability that $w \in \Sigma^k$ is a reset.* Obviously if $\mathcal{A}$ is reset-free then $R(k) = 0$ for every k.

Claim 2 (Probability of Resets) *If $\mathcal{A}$ is synchronizing then*

1. $R(\ell(\mathcal{A})) \geq \hat{p}^{\ell(\mathcal{A})}$.
2. *Moreover,* $\lim_{k \to \infty} R(k) = 1$.

Proof:

1. Trivial, follows from the existence of a reset induced by a sequence of length $\ell(\mathcal{A})$.
2. We take the transformation semigroup $(Q, S, \cdot)$ generated by Σ and convert it into an automaton (Σ, S, γ) with $\gamma(s, \sigma) = s \cdot \sigma$, as is done in the proof of Cayley theorem. We replace the Σ-labeling of the edges by their corresponding probabilities and get a Markov chain over the space of transformations. The set of resets, which is the minimal right ideal of S (see [2],[7]) is an absorbing subset and its probability goes to one. ∎

Unfortunately we cannot make use of the asymptotic convergence of S to resets because as $|w|$ grows, the probability that $\delta(q,w) = \delta'(q,w)$ decreases. In fact we have a trade-off between an increasing probability for a reset in $\mathcal{A}$ and a decreasing probability of an equivalent error-free behavior in $\mathcal{A}'$. Our main result is:

Theorem 3 (Robustness of Synchronizing Automata) *Let $\mathcal{A}$ be a synchronizing automaton with n states and let $\mathcal{A}'$ be an ϵ-noisy version of $\mathcal{A}$. Then, for any $k \geq 0$,*

$$\rho(\mathcal{A}, \mathcal{A}') \leq \frac{1-\eta}{1-\eta+R(k)\eta} \tag{5}$$

where $\eta = (1-\epsilon)^k$. By letting $k = \ell(\mathcal{A})$ we obtain

$$\rho(\mathcal{A}, \mathcal{A}') \leq \frac{1-(1-\epsilon)^{\ell(\mathcal{A})}}{1-(1-\epsilon)^{\ell(\mathcal{A})}+\hat{p}^{\ell(\mathcal{A})}(1-\epsilon)^{\ell(\mathcal{A})}} \tag{6}$$

Proof: We partition the state-space of $\mathcal{A} \times \mathcal{A}'$ into two sets, the "good" diagonal pairs G, and the "bad" error states B. We consider the transition probabilities between G and B after reading k symbols, for some k, $k \geq \ell(\mathcal{A})$. The probability of staying in a diagonal state is at least the probability of having k non-noisy transitions while the probability of returning from B to G is the latter multiplied by the probability of a reset in $\mathcal{A}$. Thus for every t,

$$P_{t+k}(G) \geq (1-\epsilon)^k (P_t(G) + R(k)(1-P_t(G))) \tag{7}$$

where $P_t(G)$ denotes the probability of being at some element of G after reading t symbols. The result follows from the well-known fact that if a positive sequence $\{x_k\}_{k=1}^{\infty}$ satisfies $x_{n+k} > \alpha x_k + \beta$, $0 < \alpha < 1$, then

$$\liminf_{n\to\infty} x_n > \frac{\beta}{1-\alpha}$$

∎

Corollary 4 *For every synchronizing automaton $\mathcal{A}$*

$$\lim_{\epsilon\to 0} \rho(\mathcal{A}, \mathcal{A}') = 0 \tag{8}$$

The significance of this result is in showing that the "global" accuracy of computations with resets can always be improved by decreasing the "local" noise. This means that computational tasks that fall into this category can be made more and more reliable by improving the components realizing them, e.g, by using redundant sensors, faster processors, etc. In the next section we will see that in other cases the presence of a local noise, no matter how small, causes a large global deviation from the correct behavior.

4 Non-Robustness of Reset-Free Automata

After establishing an upper-bound on the distance for synchronizing automata, we would like to set a lower-bound for the complementary class of reset-free automata. In the special case of permutation automata, i.e., those in which all the input letters induce permutations, we have the following lower-bound:

Theorem 5 (Non-Robustness of Permutation Automata) *Let $\mathcal{A}$ be any n-state permutation automaton $(n > 1)$. Then*

1. *For any noisy version $\mathcal{A}'$ such that for every q, σ, $Pr\{\delta'(q,\sigma) \neq \delta(q,\sigma)\} \geq \epsilon$ we have*
$$\rho(\mathcal{A}, \mathcal{A}') \geq \frac{1}{2} \tag{9}$$

2. *There exist an ϵ-noisy version $\mathcal{A}'$ such that*
$$\rho(\mathcal{A}, \mathcal{A}') \geq \frac{2n-1}{2n} \tag{10}$$

Proof:

1. The proof is similar to the previous one. This time we note that the probability of a transition from G to B is at least ϵ while the probability of moving back from B to G is at most ϵ (because of the lack of any merging sequence, errors can only be corrected by subsequent errors). Thus we have a symmetric chain that converges to 1/2.

2. We use the same argument but consider a noise pattern such that every letter that induces a permutation in $\mathcal{A}$, induces in $\mathcal{A}'$ with probability ϵ, a permutation completely different from the original one. Thus, the probability of moving from G to B is the same but the probability of correcting an error decreases from ϵ to ϵ/n. ∎

If we look at n asymptotically we see that for large permutation automata there exists noise patterns that can make them being wrong *most* of the time.

Our last result concerns the whole class of reset-free automata. The analysis here is a bit more complicated because the set B of non-diagonal states divides into two subsets: W containing all the pairs which cannot be merged by any sequence, and U containing those that are correctable. The synchronizing case corresponds to $W = \emptyset$, while the permutation case corresponds to $U = \emptyset$. We will denote by $R'(k)$ the probability over Σ^k of those sequences leading from U to G.

Theorem 6 (Non-Robustness of Reset-Free Automata) *For every reset-free automaton $\mathcal{A}$ with n states there exists an ϵ-noisy version $\mathcal{A}'$ such that*

$$\lim_{\epsilon \to 0} \rho(\mathcal{A}, \mathcal{A}') \geq \frac{1}{1 + \frac{1}{\hat{p}}} \tag{11}$$

Proof: Our analysis is based on the following observations: 1) There exist at least two states q, q' that cannot be merged and since the automaton is strongly-connected,[1] there exists at least one $\sigma \in \Sigma$ and $q^* \in Q$ such that $\delta(q^*, \sigma) = q$. Then we define a noisy version in which $\delta'(q^*, \sigma) = q'$ with probability ϵ. This means that from $(q^*, q^*) \in G$ we can go to $(q, q') \in W$ with a probability not smaller than $\epsilon\hat{p}$. 2) The probability of leaving W in one step is smaller then ϵ (as in the permutation case). 3) The probability of going from G to W in k steps is at least the probability of getting from every $(q, q) \in G$ to (q^*, q^*) in $k-1$ steps multiplied by $\epsilon\hat{p}$. From all this we obtain:

$$P_{t+k}(W) \geq (1-\epsilon)^k P_t(W) + \epsilon\hat{p} P_t(G) \hat{p}^{k-1} (1-\epsilon)^{k-1} \tag{12}$$

and

$$P_{t+k}(G) \geq (1-\epsilon)^k [P_t(G) + R'(k) P_t(U)] \tag{13}$$

[1]When considering asymptotic probabilistic behavior we should only care about strongly-connected components.

Summing up (12) and (13) and using the fact that $P_t(U) = 1 - P_t(G) - P_t(W)$, we obtain an equation that we treat like (7) in order to show that $P_t(U)$ is negligible when ϵ is small. Thus we can replace $P_t(G)$ by $1 - P_t(W)$ in (12), let $k = 1$ and obtain the result. ∎

5 The Price of Being Slow

In this section we apply the previous results to real-time situations. In order to do this we extend the model by using notions of timed sequences and timed automata (see [1]). A timed sequence is $w = (s_1, t_1)(s_2, t_2)\ldots$ where for every i, $s_i \in \Sigma$, $t_i \in \mathbb{R}$, $t_0 = 0$ and $t_i < t_{i+1}$. Intuitively each t_i denotes the "time-stamp" of the arrival of s_i.

A timed automaton is an automaton such that each of its transitions is augmented with a number d indicating the time that must elapse between the arrival of the input and the execution of the transition. Several results have been proved concerning the timed sequences that can be accepted or generated by various types of timed automata (the version described here is a simplified one). We are concerned here with the opposite problem: given a set of timed sequences that does not necessarily obey the timing constraints – what can be said about the expected behavior of the automaton?

In order to be able to speak quantitatively we make the following assumptions:

1. For every timed sequence and for every $i > 0$, $x_i = t_{i+1} - t_i$ is an exponential random variable with a parameter α. Thus the mean time between two arrivals of input is $\frac{1}{\alpha}$.

2. Similarily for every run and for every $i > 0$ the time y_i between the arrival of the i^{th} input and the execution of the i^{th} transition is also distributed exponentially with a parameter $\beta \leq \alpha$, so the mean duration of a transition is $\frac{1}{\beta}$.

We also assume that any input arriving before the previous transition has been executed is lost. The probability of missing a symbol s_i is the probability of $x_i < y_i$ which is given by

$$\int_{x=0}^{\infty} \int_{y=x}^{\infty} \alpha e^{-\alpha x} \beta e^{-\beta y} dx dy = \int_{x=0}^{\infty} \alpha e^{-\alpha x} e^{-\beta x} dx = \frac{\alpha}{\alpha + \beta} \qquad (14)$$

Thus, such a real-time situation converts the ideal automaton (infinitely faster than the environment) into an ϵ-noisy version with $\epsilon = \frac{\alpha}{\alpha+\beta}$, where in probability ϵ the automaton misses an input symbol and makes a self-loop instead (as in the example of figure **??**). So all our previous results can be transferred to this situation: for synchronizing automata, by increasing β (that is, taking a faster automaton) we can decrease the error as much as we want. For reset-free automata, no matter how fast they operate, the errors eventually accumulate and their behavior becomes random with respect to the intended one.

6 Discussion

6.1 Past

In this paper we have built a model that captures an intuitive property of computations in the presence of noise: the longer is the past history upon which a computation depends, the larger is the probability of error. The essence of the model is in considering a class of probability distributions on Σ^* such that the notion of expected distance between behaviors becomes meaningful. This idea, inspired by an old paper on language identification ([13]), is in contrast with traditional treatment of stochastic automata in computer science (see [9], [10]) where such automata are used as acceptors of *individual* sequences whose probability of reaching a terminal state is above some threshold. Observations in the same spirit as ours have been made in [3], in the context of certain codes which can "self-synchronize" in high probability. This notion of expected correctness relative to some probability on the input also underlies the PAC-learnability model ([12]) and we believe that investigating its properties can contribute to the general shift from worst-case to average-case analysis of computational phenomena.

The notion of comparing the ideal and the noisy behavior appears already in Von Neumann's seminal paper ([14]). In that paper a similar question of obtaining global correctness in spite of local noise is discussed and the solution of redundancy is devised. It is interesting to note that Von Neumann considered local/global relations in space, i.e., some logical gates can be faulty but the output of the whole circuit is correct, while we consider the same relationship with respect to time. Another association which comes in mind is with Dijkstra's notion of self-stabilizing system ([4]) where the system can go from any incorrect configuration into a correct one after finitely-many steps.

6.2 Future

We will mention briefly several research direction that can follow this work.

- It might be interesting to investigate specific classes of automata that arise in the modeling of realistic situations, for example, automata whose state-space is embedded in a metric space and the transitions have some arithmetical or geometrical interpretation. In such a case the distance between the behaviors will be more refined than the distance we used in this paper which was induced by the discrete metric on the state-space.

- The real-time model introduced in the previous section can be extended to include a bounded buffer – in this case an input symbol is lost only if it arrives when the buffer is full. Such a model will require alternative definitions of distance that takes into account the difference between logical (i) and real (t_i) time. For example, the automaton can be in the correct state with respect to the sequence it has read so far, but in the wrong state if we consider additional symbols that have already arrived and wait in the buffer. A reasonable measure would be the average length of a real-time interval in which the ideal and the timed automaton agree. Within such a model, trade-offs between speed, accuracy and buffer size could be computed.

- In our model we have only considered the task of mapping classes of input histories into internal states. This model can be extended into a full control model by specifying the dynamics of the environment, the structure of observations (the relation between the states of the environment and the input of the program), and the effect of the automaton's output on the environment. For a discrete environment, such an extension will add a robustness dimension to recent models ([11]) dealing with the control of discrete-event dynamical systems. If on the other hand we consider automata interacting with a continuously changing environment we come into the realm of *hybrid* systems ([8])

having much more intricate relationships between time, change, observation and noise. The modeling of such systems requires a broader synthesis of computational and control-theoretic models.

Acknowledgements

We would like to thank A. Benveniste, A. Juditsky, A. Pnueli and an anonymous referee for their attention and comments.

References

[1] R. Alur and D. Dill, The theory of timed automata, in: J. de Bakker, C. Huizing, W. de Roever, and G. Rozenberg, eds., *Proceedings of the REX Workshop "Real-Time: Theory in Practice"*, Lect. Notes in Comp. Sci., Vol. 600 (Springer, Berlin, 1992) 28-44.

[2] J. Berstel and D. Perrin, *Theory of Codes* (Academic Press, New-York, 1985).

[3] R.M. Capocelli, L.G. Gargano and U. Vaccaro, On the Characterization of Statistically Synchronizable Variable-Length Codes, *IEEE Trans. on Information Theory*, **34** (1988) 817-825.

[4] E.W. Dijkstra, Self-stabilizing systems in spite of distributed control, *Comm. of the ACM* **17** (1974) 643-644.

[5] J.E. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages and Computation* (Addison-Wesley, Reading, MA, 1979).

[6] J.G. Kemeny and J.L. Snell, *Finite Markov Chains* (Van Nostrand, New-York, 1960).

[7] G. Lallement, *Semigroups and Combinatorial Applications* (Wiley, New-York, 1979).

[8] O. Maler, Z. Manna, and A. Pnueli. From timed to hybrid systems, in: J. de Bakker, C. Huizing, W. de Roever, and G. Rozenberg, eds., *Proceedings of the REX Workshop "Real-Time: Theory in Practice"*, Lect. Notes in Comp. Sci., Vol. 600 (Springer, Berlin, 1992) 28-44.

[9] A. Paz, *Introduction to Probabilistic Automata* (Academic Press, New-York, 1970).

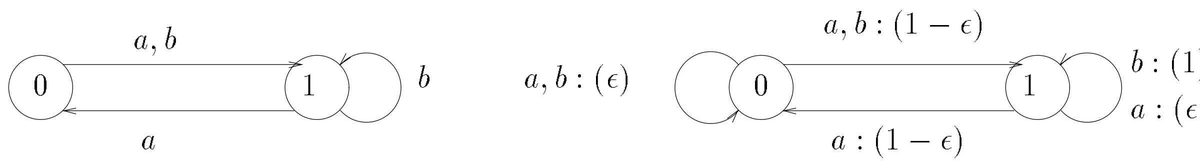
[10] M.O. Rabin, Probabilistic automata, *Information and Control* **6** (1963) 230-245.

[11] P.J.G. Ramadge and W.M. Wonham, The control of discrete event systems, *Proc. of the IEEE* **77** (1989) 81-98.

[12] L.G. Valiant, A theory of the learnable, *Comm. of the ACM* **27** (1984) 1134-1142.

[13] R.M. Wharton, Approximate language identification, *Information and Control* **26** (1974) 236-255.

[14] J. Von Neumann, Probabilistic logics and the synthesis of reliable organisms from unreliable components, in: C. Shannon and J. McCarthy, eds., *Automata Studies* (Princeton University Press, Princeton, 1956) 205-228.

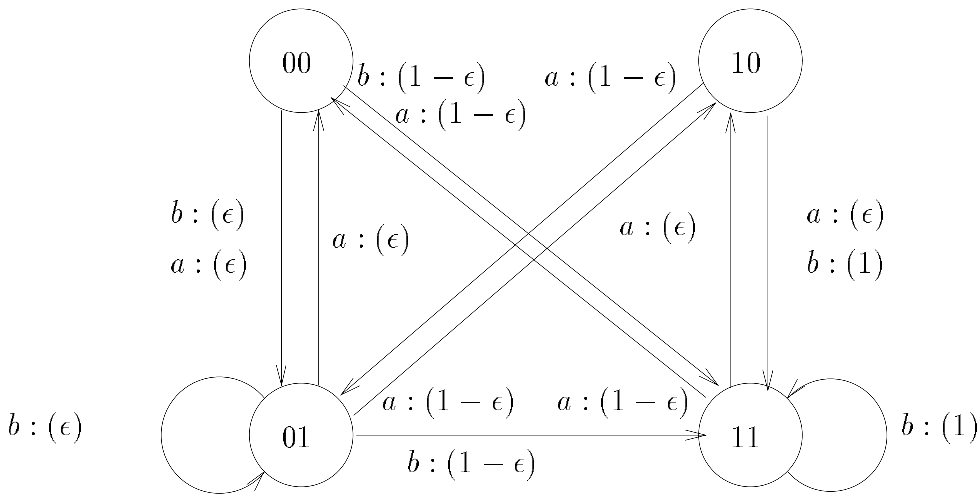


(a)

	$\mathcal{A}$	$\mathcal{A}'$
M^a	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} \epsilon & 1-\epsilon \\ 1-\epsilon & \epsilon \end{pmatrix}$
M^b	$\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} \epsilon & 1-\epsilon \\ 0 & 1 \end{pmatrix}$

(b)

Figure 1: *(a) A deterministic automaton $\mathcal{A}$ and one of its ϵ-noisy versions $\mathcal{A}'$. (b) A matrix representation of $\mathcal{A}$ and $\mathcal{A}'$.*



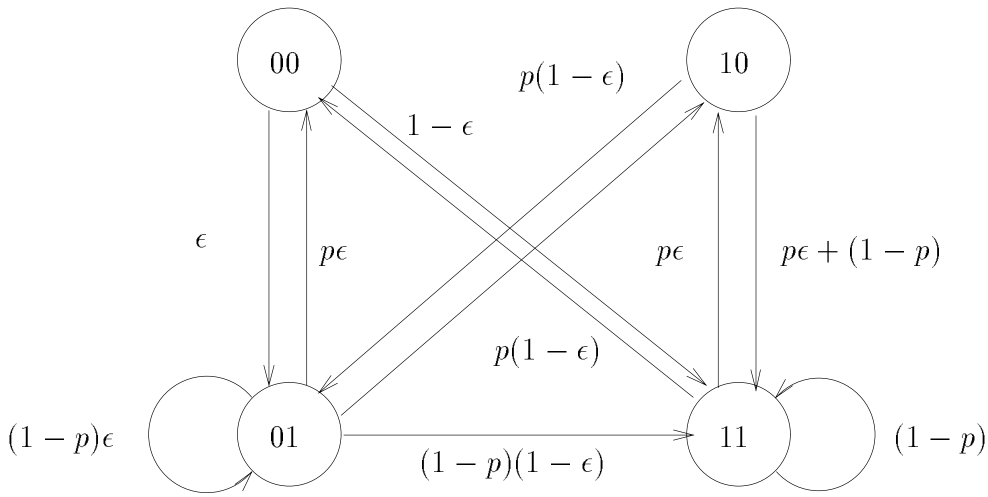
(a)

$$M^a \qquad\qquad M^b$$

$$\begin{pmatrix} 0 & \epsilon & 0 & 1-\epsilon \\ \epsilon & 0 & 1-\epsilon & 0 \\ 0 & 1-\epsilon & 0 & \epsilon \\ 1-\epsilon & 0 & \epsilon & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & \epsilon & 0 & 1-\epsilon \\ 0 & \epsilon & 0 & 1-\epsilon \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(b)

Figure 2: *(a) The product* $\mathcal{A}' \times \mathcal{A}$. *(b) A matrix representation of* $\mathcal{A}' \times \mathcal{A}$. *The indices of the rows and columns correspond to the pairs* $(0,0), (0,1), (1,0)$ *and* $(1,1)$.



(a)

$$\begin{pmatrix} 0 & \epsilon & 0 & 1-\epsilon \\ p\epsilon & (1-p)\epsilon & p(1-\epsilon) & (1-p)(1-\epsilon) \\ 0 & p(1-\epsilon) & 0 & p\epsilon + (1-p) \\ p(1-\epsilon) & 0 & p\epsilon & 1-p \end{pmatrix}$$

(b)

Figure 3: *(a) The Markov chain associated with $\mathcal{A}' \times \mathcal{A}$. (b) A matrix representation.*