Steady-State Probability Estimation in FSMs Considering High-Order Temporal Effects

Diana Marculescu, Radu Marculescu
and Massoud Pedram

CENG 97-19

Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, California 90089-2562
(213) 740-4458

April 1997

STEADY-STATE PROBABILITY ESTIMATION IN FSMS CONSIDERING HIGH-ORDER TEMPORAL EFFECTS

Abstract

This paper illustrates, analytically and quantitatively, the effect of high-order temporal correlations on steady-state and transition probabilities in Finite State Machines (FSMs). As the main theoretical contribution, we extend the previous work done on steady-state probability calculation in FSMs to account for complex spatiotemporal correlations which are present at the primary inputs when the target machine models real hardware and receives data from real applications. More precisely: 1) using the concept of constrained reachability analysis, the correct set of Chapman-Kolmogorov equations are constructed; and 2) based on stochastic complementation and iterative aggregation/disaggregation techniques, exact and approximate techniques for finding the state occupancy probabilities in the target machine are presented. From a practical point of view, we show that assuming temporal independence or even using first-order temporal models is not sufficient, that is, the inaccuracies induced in steady-state and transition probability calculations are significant for most of the analyzed benchmarks. Experimental results show that, if the order of the source is underestimated, not only the set of reachable sets is incorrectly determined, but also the obtained probability values can be more than 100% off from the correct ones.

1. Introduction

In the last decade, probabilistic approaches have received a lot of attention as a viable alternative to deterministic techniques for analyzing complex digital systems. Logic synthesis [1], verification [2], testing [3] and more recently, low-power design [4] have benefited from using probabilistic techniques. In particular, the behavior of FSMs has been investigated using concepts from the Markov chain (MC) theory.

Studying the behavior of the MC provides us with different variables of interest of the original FSM. In this direction [5][6] are excellent references where steady-state and transition probabilities (as variables of interest) are successfully estimated in large FSMs. Both techniques are analytical in nature and, in order to manage complexity, have made some simplifying assumptions, temporal independence on the primary inputs being the most notable one. These assumptions, however, limit the applicability of the results. As we will prove in this paper, in the case of FSMs, this is not enough for accurate estimation of transition probabilities. Temporal correlations longer than one time step can affect the overall behavior of the FSM and therefore result in very different values for the actual transition probabilities compared to those predicted analytically. More interestingly, it will be shown that, if one ignores the effects of finite-order statistics at the primary inputs of the FSM, it is possible to wrongly predict non-zero steady-state probabilities for some transient states which normally occur in the beginning of operation, but disappear shortly afterwards, once the machine have reached its steady-state regime. This kind of erroneous prediction for the state occupation probabilities is especially harmful in timing verification, when one tries to find the probability of occurrence of an event related to a state or transition in the circuit.

Addressing these issues, the present paper extends the previous work reported in [5] to explicitly include complex spatiotemporal correlations in steady-state and transition probabilities calculations for standard FSMs. The analysis itself relies on time-homogeneous discrete-parameter MCs that are used in two different ways:

- first, a MC is used to model the binary input stream that typify the application data (called also *trace*) feeding the target FSM[1];
- second, a MC dependent on the first one, that characterizes the state lines of the FSM is used to solve the set of Chapman-Kolmogorov equations that give the state occupancy probabilities for the machine itself.

In fact, using the joint transition probabilities of the primary inputs and internal states in the target machine, these two models can be merged. More precisely, if the sequence feeding the target circuit has temporal correlations of order *k*, then a lag-*k* MC model of the sequence will suffice to model correctly the joint transition probabilities of the primary inputs and internal states in the target circuit. As a consequence, we can use this accurate model to solve the set Chapman-Kolmogorov equations and, once the solution is determined, one can derive the state occupancy probabilities in a straightforward manner.

Of course, studying the MC that models the FSM is also related to performing reachability analysis of the target machine. At this point, our work differs substantially from what other researchers have been considered in the past, in the sense that our reachability analysis is constrained by the actual input sequence and accounts for the very specific way in which the input source excites the target FSM.

Last but not least, analysis of MCs involves sophisticated numerical techniques; to date, Gauss-Jacobi and power method have been extensively used in steady-state probability calculation [4],[5]. We present instead two different algorithms, that is *stochastic complementation* and *iterative aggregation/disaggregation*, which provide, aside from a

[1] Although the Markov model is derived for a particular input trace, it is completely general and represents in a compact form the whole class of input sequences having the same characteristics.

deep insight into theoretical aspects of MCs analysis, an efficient solution for a large class of MCs, that is *nearly completely decomposable* systems.

Addressing these issues, the present paper improves the state-of-the-art in three ways:

- first, based on high-order Markov models, it shows an effective analytic way to account for spatial and temporal correlations of the input sequence on steady-state probability calculation in standard FSMs;
- second, based on the stochastic complement concept and iterative aggregation/disaggregation techniques, it presents exact and approximate techniques for steady-state probability calculation in FSMs;
- finally, it provides analytic and quantitative evaluations of the impact of neglecting input temporal correlations in the analysis of FSM behavior.

To conclude, this research is beneficial to general FSM analysis techniques relying on probabilistic premises. The issues brought into attention are new and represent an important step towards understanding of FSM behavior from a probabilistic point of view.

The paper is organized as follows. Section 2 presents the basic definitions and notations on FSMs and MCs that will be used throughout the paper. In Section 3 we formulate the problem we want to solve and present the basic Markov model. Section 4 focuses on constrained reachability analysis issue. In Section 5 we present exact and approximate methods for steady-state probabilities calculation, and we point out some issues regarding complexity and convergence of algorithms. Finally, we present some experimental results for common sequential benchmarks, and we conclude by summarizing our main contribution.

2. Preliminaries

In this section we present the basic definitions and notations. Far from being exhaustive, we restrict our attention to only those concepts that are required by our working hypotheses. For a complete documentation, we refer the reader to references [7][8].

2.1 Finite-order MCs

A *stochastic process* is defined as a family of random variables $\{x(t), t \in \mathrm{T}\}$ defined on a given probability space and indexed by the parameter *t*, where *t* varies over the index set T. The stochastic process is said to be *stationary* when it is invariant under an arbitrary shift of the time origin. In this case, the values assumed by the random variable *x(t)* are called *states*, and the set of all possible states forms the *state space* of the process.

A *Markov process* $\{x(t), t \in \mathrm{T}\}$ is a stochastic process whose future evolution depends only on its current state and not on its past. This is so called "Markov property" and defines a fundamental subclass of stochastic processes. We shall assume that the transitions out of state *x(t)* are independent of time and, in this case, the Markov process is said to be *time-homogeneous*.

If the state space of a Markov process is *discrete*, the Markov process is referred to as a MC. In what follows, we consider only MCs with finite state space. If we assume that the index set T is also discrete, then we have a *discrete-parameter MC*. We may assume without loss of generality that T = {0, 1, 2,...} and denote the MC as $\{x_n\}_{n \geq 0}$.

Definition 1. (lag-one MC) A discrete stochastic process $\{x_n\}_{n \geq 0}$ is said to be a lag-one MC if at any time step $n \geq 1$ and for all states x_n:

$$p(x_n = \alpha_n | x_{n-1} = \alpha_{n-1} x_{n-2} = \alpha_{n-2} \ldots x_0 = \alpha_0) = p(x_n = \alpha_n | x_{n-1} = \alpha_{n-1}) \tag{1}$$

The conditional probabilities $p(x_n = \alpha_n | x_{n-1} = \alpha_{n-1})$ are called *single-step transition probabilities* and represent the conditional probabilities of making a transition from state x_{n-1} to state x_n at time step n. In homogeneous MCs these probabilities are independent of n and consequently written as $p_{ij} = p(x_n = j | x_{n-1} = i)$ for all n = 1, 2,... The matrix Q, formed by placing p_{ij} in row i and column j, for all i and j, is called the *transition probability matrix*. We note that Q is a *stochastic matrix* because its elements satisfy the following two properties: $0 \le p_{ij} \le 1$ and $\sum_j p_{ij} = 1$.

An equivalent description of the MC can be given in terms of its *state transition graph* (STG). Each node in the STG represents a state in the MC, and an edge labelled p_{ij} (from node i to node j) implies that the one-step transition probability from state i to state j is p_{ij}.

Changes of states over $n > 1$ time steps are ruled by probability rules simply expressed in terms of p_{ij}. Let us denote by p_{ij}^n the probability of transition from state i to state j in exactly n steps, namely: $p_{ij}^n = p(x_{m+n} = j | x_m = i)$, whatever the integer m. It is easily recognized that probabilities p_{ij}^n (which are called *n-step transition probabilities*) represent the entries of the Q^n matrix (called *n-step transition matrix)*, $n \ge 1$. The Q^n matrix itself is still a stochastic matrix and satisfies the identity $Q^{m+n} = Q^m \cdot Q^n$, $m, n \ge 0$ (Q^0 is by definition the unit matrix I) or just the system of equations $p_{ij}^{m+n} = \sum_k p_{ik}^m \cdot p_{kj}^n$, known as the *Chapman-Kolmogorov equations*. In words, to go from i to j in *(m+n)* steps, it is necessary to go from i to an intermediate state k in m steps and then from k to j in the remaining n steps. By summing over all possible intermediate states k, we consider all possible distinct paths leading from i to j in *(m+n)* steps.

Definition 2. (lag-k MC) A discrete stochastic process $\{x_n\}_{n \ge 0}$ is said to be a lag-k MC if at any time step $n \ge k+1$:

$$p(x_n = \alpha_n | x_{n-1} = \alpha_{n-1} x_{n-2} = \alpha_{n-2} \ldots x_0 = \alpha_0) = p(x_n = \alpha_n | x_{n-1} = \alpha_{n-1} x_{n-2} = \alpha_{n-2} \ldots x_{n-k} = \alpha_k) \quad (2)$$

It should be noted that any lag-k MC can be reduced to a lag-one MC based on the following result.

Proposition 1. [8]. If $\{u_n\}_{n \ge 1}$ is a lag-k MC then $\{v_n\}_{n \ge 1}$, where $v_n = (u_n, u_{n+1}, \ldots, u_{n+k-1})$, is a multivariate first-order MC.

As a consequence, the study of lag-k MCs is practically reduced to study the properties satisfied by lag-one MCs. For clarity, we will refer subsequently only to lag-one MCs but, by virtue of Proposition 1, all results translate to lag-k MCs.

2.2 Classification of states, decomposability and probability distributions

To better analyze the long-run behavior of a MC we need to distinguish between states that are guaranteed to be visited infinitely often and states to which the system may never return.

Definition 3. (recurrent/transient state) A state in a MC is called *recurrent* if the probability of returning to it after $n \ge 1$ steps is greater than zero. Otherwise, the state is called *transient.*

This definition does not imply that the transient state cannot be visited many times, only that the probability of never returning to the state is nonzero. In our subsequent discussion, we will consider that all states are recurrent since all transient states vanish (that is, their steady-state probabilities become zero) after a finite number of steps.

From Definition 3, if i is a recurrent state, we have that $p_{ii}^{\,n} > 0$, for some $n \geq 1$. If the greatest common divisor over all such integers n is $d > 1$, then the state i is also called *periodic* of period d. Otherwise (i.e., if $d = 1$), the state i is called *aperiodic*. A state that is recurrent and aperiodic is said to be *ergodic*. If all the states of a MC are ergodic, then the MC itself is said to be ergodic.

Definition 4. (nondecomposable/decomposable MC) A MC is said to be *nondecomposable* (or *irreducible*) if every state can be reached from every other state in a finite number of steps. Otherwise, the chain is called *decomposable* (*reducible*) and its transition matrix can be written as a block-diagonal matrix having the form:

$$Q = \begin{bmatrix} Q_1 & 0 & \dots & 0 \\ 0 & Q_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & Q_p \end{bmatrix} \tag{3}$$

where $Q_1, Q_2, \ldots, Q_p$ are square matrices and the p sets of states contain only recurrent states that do not communicate among them.

The above definition, may be used in an approximate sense too; we say that the MC is *nearly completely decomposable* if its transition matrix Q can be partitioned in the form

$$Q = \begin{bmatrix} Q_{11} & Q_{12} & \cdots & Q_{1p} \\ Q_{21} & Q_{22} & \cdots & Q_{2p} \\ \dots & \dots & \dots & \dots \\ Q_{p1} & Q_{p2} & \cdots & Q_{pp} \end{bmatrix} \tag{4}$$

where the non-zero elements in the off-diagonal blocks are small[1] compared to those in the diagonal blocks. As we shall see later, the concept of near decomposability plays a significant part in our framework.

We now turn our attention to distribution defined on the states of a MC. We shall denote by $\pi_i^{\,n}$ the probability that the MC is in state i at step n, that is $\pi_i^{\,n} = p(x_n = i)$. In vector notation we have that $\pi^n = (\pi_1^{\,n}, \pi_2^{\,n}, \ldots, \pi_i^{\,n}, \ldots)$, where π is a row vector. The probability that the MC is in state i at step n is given by $\pi^n = \pi(0) \cdot Q^{\,n}$, where $\pi(0)$ denotes the initial state distribution of the chain. For nondecomposable and aperiodic MCs it may be shown that the *limiting distribution* $\pi = \lim_{n \to \infty} \pi^n$ always exists and it is independent of the initial probability distribution. In addition, the following important results hold.

Proposition 2. [7] For a nondecomposable MC, the equation $\pi \cdot Q = \pi$ has a unique solution that represents the *stationary distribution* (or stationary probability vector) of the MC.

The unique solution of the equation in Proposition 2 can actually be determined by solving the system of equations $\pi_j = \sum_i \pi_i \cdot p_{ij}$ with $\sum_j \pi_j = 1$. The vector π represents the *left eigenvector* corresponding to the unit eigenvalue, that is, it satisfies the standard equation $\pi \cdot Q = \lambda_1 \cdot \pi$ for $\lambda_1 = 1$.

[1] The precise meaning of ‘small’ will be defined later in Section 5.

3. FSM steady-state analysis: problem formulation

In this section, we introduce formally the problem we have to solve and we present two Markov models that we use to solve it: one model is the MC associated to the state lines of the FSM and another one is the MC that models the input sequence that feeds the target FSM. Within this framework, we also analyze the effects of input statistics on FSM behavior.

3.1 FSM characterization

As used by Hachtel et al. in [2], the probabilistic behavior of an FSM can be studied by regarding its state transition graph (STG) as a MC. More precisely, attaching to each out-going edge of each state in the target FSM a transition probability that corresponds to that particular transition, one actually obtains a MC as defined in previous section. Furthermore, studying the behavior of the underlying MC provides us different variables of interest in the original FSM. For instance, we may be interested in defining the MC associated to state lines of the FSM (denoted by $\{s_n\}_{n \geq 0}$) because, by virtue of Property 1 in Section 2, we may find the state occupation probabilities for the machine. To this end, the equation to solve is:

$$\pi \cdot Q_S = \pi \quad \text{with} \quad \sum_{all\ i} \pi_i = 1 \tag{5}$$

where π and $Q_S = (p_{ij})_{1 \leq i, j \leq m}$ denote the stationary distribution and the transition matrix of the chain, respectively.

To set up the above Q_S matrix, the authors in [2] consider that all input combinations are equiprobable during the normal operation of the machine and therefore, the one-step transition probability matrix can be obtained from the transition relation in a straightforward manner. However, in practice, the situation can be quite different: different input sequences may exercise the machine in different ways and thus produce substantially different STG structures. Due to the feedback lines, the behavior of the state lines themselves is strongly dependent on the characteristic of the input sequences present at primary inputs and therefore, to set up the Q_S matrix which actually accounts for the influence of correlations at the primary inputs on the state lines of the FSM is a key (and nontrivial!) task. To construct the exact Q_S matrix that corresponds to the actual input trace, we also associate a finite-order MC to the primary input stream; we assume that this MC is described by the matrix $Q_X = (q_{ij})_{1 \leq i, j \leq l}$. To see what difference can make the actual input sequence in FSM analysis, let us consider the following example.

Example 1: Let S_1 and S_2 be two 2-bit sequences, of length 48, as shown in Fig.1a. These two sequences, have exactly the same set of first-order temporal statistics that is, they cannot be distinguished as far as wordwise one-step transition probabilities are concerned. In fact, in Fig.1b we provide the wordwise transition graph for these two sequences. Each node in this graph is associated to a distinct pattern that occurs in S_1 and S_2 (the upmost bit is the most significant one, e.g. in S_1, v_1 = '1', v_2 = '2', v_3 = '3',..., v_{48} = '1'). Each edge represents a valid transition between any two valid patterns and has a nonzero probability associated with it. For instance, the pattern '3' in S_1 and S_2 is always followed by '1' (thus the edge between nodes '3' and '1' has the probability 1) whereas it is equally likely to have either '0', '2' or '1' after pattern '1'.

S_1

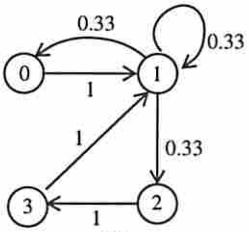
v_{48} … $v_3 v_2 v_1$

0110110110000110000001101100000000000110000000110
1101101101111101101011011010111010111010110101101

S_2

001100001100001100001100001100001100001100001100
011011011011011011011011011011011011011011011011

(a)



(b)

Fig.1: Two sequences with the same first-order characteristics

Starting with different initial states and using a random number generator we may, of course, generate other sequences equivalent with S_1 and S_2 as far as the one-step transition probabilities are concerned. We can then see the graph in Fig.1b as a compact, canonical, characterization of sequences S_1 and S_2. Suppose that we want to compute the occurrence probability of string v = '01 10' that is, the probability that transition $1 \to 2$ is taking place in S_1. To this effect, we just use $p(v) = p(v_1 v_2) = p(v_1) \cdot p(v_2 | v_1)$ which gives us the value of 1/6. If we are interested in finding the two-step transition probability $0 \to 1 \to 1$ in S_2, then we use the formula $p(v) = p(v_1 v_2 v_3) = p(v_1) \cdot p(v_2 | v_1) \cdot p(v_3 | v_2 v_1)$, and then we get the value of 1/6.

If we consider now that the sequence S_1 is applied to the benchmark *dk17*, then using the STG-based calculations when all input combinations are equally likely to occur, one can find the matrix shown in Fig.2b. On the other hand, simulating the actual sequence S_1 and analyzing the state occupancy probabilities of the circuit, one can construct the matrix $Q_S{}^{trace}$ shown in Fig.2c.

S_1

v_{48} … $v_3 v_2 v_1$

0110110110000110000001101100000000000110000000110
1101101101111101101011011010111010111010110101101

(a)

$$Q_S^{equi} = \begin{bmatrix} 0 & 0.5 & 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0.25 & 0 & 0.25 & 0.25 & 0.25 & 0 & 0 \\ 0 & 0.25 & 0 & 0.25 & 0.25 & 0.25 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 \\ 0.75 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0.25 & 0 & 0 & 0.25 & 0 \\ 0.5 & 0 & 0.25 & 0 & 0 & 0 & 0 & 0.25 \\ 0.5 & 0 & 0 & 0.25 & 0.25 & 0 & 0 & 0 \end{bmatrix}$$

(b)

$$Q_S^{trace} = \begin{bmatrix} 0 & 0.47 & 0 & 0 & 0.53 & 0 & 0 \\ 0 & 0.25 & 0.5 & 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0.39 & 0.61 & 0 & 0 & 0 \\ 0.46 & 0 & 0.54 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.77 & 0 & 0 & 0.23 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0.33 & 0 & 0.33 & 0.33 & 0 & 0 & 0 \end{bmatrix}$$

(c)

Fig.2: Building the Q_S matrix in two different hypotheses

Obviously, solving equation (5) for these two matrices will result in two very different solutions: the matrix in Fig.2b gives the solution of equation (5) as [0.22 0.15 0.01 0.25 0.17 0.15 0.04 0.01], while in the second case we get the solution [0.12 0.08 0.42 0.26 0.09 0.02 0.02]. This shows that accounting for correlations at the primary inputs is important for calculating steady-state probability distribution of the FSM. In fact, this type of dependence is mentioned by the authors in [5] but they account for it only from a signal probability perspective. More precisely, they allow certain signals in their analysis to be fixed at 0 or 1 or have a signal probability different that 0.5 and for this new STG (obtained from the original one by deleting some edges), they perform steady-state probability analysis using the power method for the underlying matrix Q_S. Obviously, this solution has only limited applicability because it is very common in practice to have much more complicated temporal correlations among at the primary input lines. In the following, we investigate the effect of input statistics on FSM behavior.

3.2 The effect of input sequence on FSM behavior

Referring to the general FSM in Fig.3, we proposed up to this point two interacting Markov models: one for the primary inputs $\{x_n\}_{n \geq 0}$ (which characterizes the trace) and another one, dependent on the first one, for the state lines $\{s_n\}_{n \geq 0}$ (which characterizes the machine itself). In fact, these two models can be conceptually merged via the joint probabilities $p(x_n s_n)$ and $p(x_n s_n x_{n-1} s_{n-1})$; as we can see in Fig.3, they completely characterize the input that feeds the next state and the output logic of the target circuit.

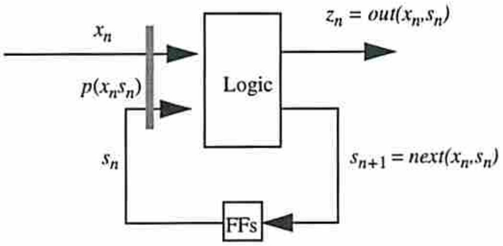


Fig.3: Modeling the target FSM

Under the general assumption of ergodicity we can prove the following result:

Theorem 1. If the sequence feeding a target sequential circuit has order k, then a lag-k MC which correctly models the input sequence, also correctly models the k-step conditional probabilities of the primary inputs and internal states in the target circuit, that is $p(x_n s_n | x_{n-1} s_{n-1} x_{n-2} s_{n-2} \ldots x_{n-k} s_{n-k}) = p(x_n | x_{n-1} x_{n-2} \ldots x_{n-k})$.

Proof: Let $p(x_n s_n x_{n-1} s_{n-1} \ldots x_{n-k} s_{n-k})$ be the joint transition probability for inputs and states over k consecutive time steps. Then we have:

$$p(x_n s_n \ldots x_{n-k} s_{n-k}) = \begin{cases} p(x_n x_{n-1} \ldots x_{n-k} s_{n-k}) & \text{if } next(x_i, s_i) = s_{i+1} \quad i = n-k, \ldots, n-1 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

For the first alternative we have:

$$p(x_n s_n \ldots x_{n-k} s_{n-k}) = p(x_n x_{n-1} \ldots x_{n-k} s_{n-k}) = p(x_n s_{n-k} | x_{n-1} \ldots x_{n-k}) \cdot p(x_{n-1} \ldots x_{n-k}) \quad (7)$$

Since x_n is a lag-k MC, x_n depends only on $x_{n-1}, x_{n-2}, \ldots, x_{n-k}$. On the other hand, s_{n-k} is a function of x_{n-k-1} and s_{n-k-1}. Thus, when $x_{n-1}, x_{n-2}, \ldots, x_{n-k}$ are fixed, x_n and s_{n-k} are independent, that is:

$p(x_n s_{n-k} | x_{n-1} x_{n-2} \ldots x_{n-k}) = p(x_n | x_{n-1} x_{n-2} \ldots x_{n-k}) \cdot p(s_{n-k} | x_{n-1} x_{n-2} \ldots x_{n-k})$ or equivalently, using (7), we obtain: $p(x_n s_n \ldots x_{n-k} s_{n-k}) = p(x_n | x_{n-1} x_{n-2} \ldots x_{n-k}) \cdot p(x_{n-1} x_{n-2} \ldots x_{n-k} s_{n-k})$.

Dividing both sides by $p(x_{n-1} x_{n-2} \ldots x_{n-k} s_{n-k})$ and using the second part of (7) we obtain exactly $p(x_n s_n | x_{n-1} s_{n-1} x_{n-2} s_{n-2} \ldots x_{n-k} s_{n-k}) = p(x_n | x_{n-1} x_{n-2} \ldots x_{n-k})$. For the second alternative, if $x_n s_n x_{n-1} s_{n-1} x_{n-2} s_{n-2} \ldots x_{n-k} s_{n-k}$ is not a valid sequence, then $p(x_n s_n | x_{n-1} s_{n-1} x_{n-2} s_{n-2} \ldots x_{n-k} s_{n-k}) = 0$ and this concludes our proof. ■

We note therefore that preserving order-k statistics implies also that order-k statistics will be captured for inputs and states. In general, modeling an order-*k* source by a lower order source may introduce large inaccuracies as shown in the next example.

Example 2: We consider once again the sequences S_1 and S_2 that feed the benchmark *dk17* and illustrate that indeed, if the input sequence has order two, then modeling it as a lag-one MC will *not* preserve the first-order joint transition probabilities (primary inputs and internal states) in the target circuit. We simulated the benchmark *dk17 (*starting with the same initial state *'19')* for both sequences and we present in Fig.4 (Fig.4a is for S_1 and Fig.4b is for S_2) the wordwise transition graphs obtained for the signal lines (x_ns_n).

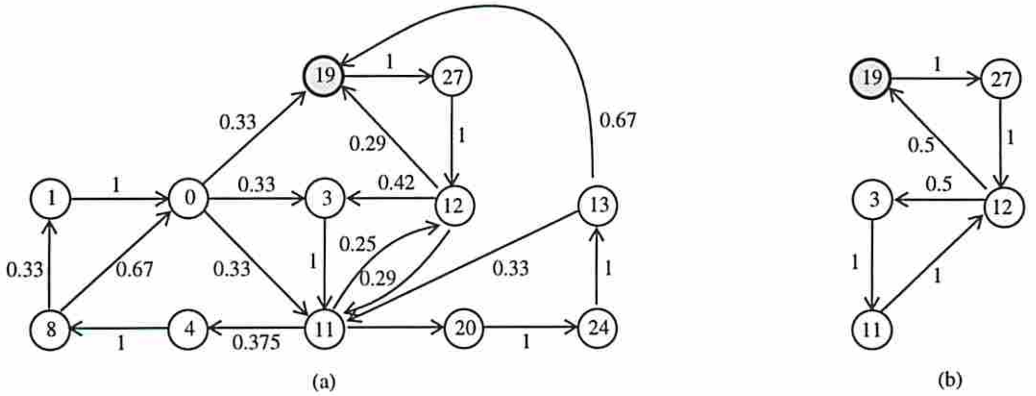


Fig.4: The MCs associated to input and state lines for two different types of input sequences

The benchmark *dk17* has 2 primary inputs and 3 FFs therefore in Fig.4, any node is decimally encoded using 5 bits. For instance, the initial state *'19'* corresponds to the binary code '10011' that is, '10' for primary inputs and '011' for state lines. After that, applying '11' on the primary inputs, the present state becomes '011', therefore we enter the node '11011' = 27 in Fig.4. As we can see, because S_1 can be modeled as a first order Markov source, while S_2 must be modeled as a second order Markov source, the corresponding transition graphs are quite different. From a practical point of view, this means that if one underestimates a high-order source (for instance, assuming that second- or higher-order temporal correlations are not important), then one may end up not preserving even the first-order transition probabilities in the target circuit.

To conclude this section, we note that these findings not only clarify the relationship between the primary inputs and state lines of the FSM, but also provide a theoretical justification for considering high-order temporal models in steady-state probability analysis of FSMs.

4. Sequence-driven reachability analysis

In this section we will present first a theoretical framework for sequence-driven reachability analysis, followed by a practical solution to solving this problem. We point out that sequence-driven reachability analysis differs from classical reachability analysis in that it takes into consideration constraints on the inputs that is, the possible set of input vectors applicable to the circuit and their sequencing.

In [9] it has been shown that to any first-order Markov chain one can associate a *Stochastic Sequential Machine* (SSM) that generates symbols according to the conditional probabilities of the initial Markov chain. Specifically, a synthesis procedure for the SSM modeling the input sequence has been proposed by the authors. Based on this synthesis procedure, the target FSM and its input can be viewed as in Fig.5.

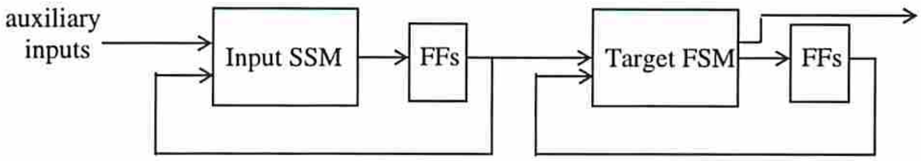


Fig.5: A general framework for FSM analysis using input sequence modeling

The primary inputs of the SSM in Fig.5 (called *auxiliary inputs*) are generated according to a probability distribution such that the states of the SSM have exactly the behavior of the inputs of the target FSM. Thus, the analysis can be done on the *product machine* (*input SSM*, *target FSM*) which has temporally uncorrelated inputs with a prescribed probability distribution. What we are interested in, is the probability distribution on the state lines, that is the probability that the FSM is in a particular state.

To make the presentation more clear, we introduce a new formulation based on matrices, mostly because this is a convenient notation for our purpose. As we can see in Fig.3, the first-order Markov chain at the primary inputs x_n can be characterized by the matrix of conditional probabilities $Q_X = (q_{ij})_{1 \le i,j \le l}$, where $q_{ij} = p(x_n = v_j | x_{n-1} = v_i)$ and $\{v_1, v_2, \ldots, v_l\}$ represents the actual trace at the primary inputs. On the other hand, the Markov chain defined jointly for the primary inputs and state lines $\{x_n s_n\}_{n \ge 0}$ can be characterized by the matrix $Q_{XS} = (r_{ip,jq})_{\substack{1 \le i,j \le l \\ 1 \le p,q \le m}}$, where $r_{ip,jq} = p(x_n s_n = v_j u_q | x_{n-1} s_{n-1} = v_i u_p)$ and $\{u_1, u_2, \ldots, u_m\}$ represents the set of reachable states. From this joint characterization we can easily derive the state probabilities as: $p(s_n) = \sum_{all\ x_n} p(x_n s_n)$ which is actually the variable of interest in our investigation. The following result provides the starting point in finding the correct matrix Q_{XS} which is the stochastic matrix associated with the product machine.

Proposition 3. The matrix Q_{XS} can be written in the following form:

$$Q_{XS} = \begin{bmatrix} q_{11}B_1 & q_{12}B_1 & \cdots & q_{1l}B_1 \\ q_{21}B_2 & q_{22}B_2 & \cdots & q_{2l}B_2 \\ \cdots & \cdots & & \cdots \\ q_{l1}B_l & q_{l2}B_l & \cdots & q_{ll}B_l \end{bmatrix} \tag{8}$$

where $\{B_i\}_{1 \le i \le l}$ is a set of $m \times m$ degenerate[1] stochastic matrices defining the next state function for input v_i; specifically, if $B_i = (b^i_{pq})_{1 \le p,q \le m}$ then $b^i_{pq} = \begin{cases} 1 & \text{if } next(v_i, u_p) = u_q \\ 0 & \text{otherwise} \end{cases}$.

[1] That is, the elements of the matrices B_i are only 0 or 1.

Proof: From Theorem 1 we can deduce that:

$$p(x_n s_n = v_j u_q | x_{n-1} s_{n-1} = v_i u_p) = \begin{cases} p(x_n = v_j | x_{n-1} = v_i) & \text{if } next(v_i, u_p) = u_q \\ 0 & \text{otherwise} \end{cases}$$

from which the representation in (8) follows immediately. ■

Corollary 1. The product machine (*input SSM, target FSM*) (as in Fig.5) is also a SSM whose auxiliary inputs are excited using the same probability distribution as the one for the input SSM.

Proof: From [9] and Proposition 3, if $Q_X = \sum_i p_i \cdot U_i$ where U_i's are degenerate stochastic matrices, we can easily write $Q_{XS} = \sum_i p_i \cdot V_i$ where V_i's are degenerate matrices having the form $V_i = \begin{bmatrix} u^i_{11}B_1 & u^i_{12}B_1 & \dots & u^i_{1l}B_1 \\ u^i_{21}B_2 & u^i_{22}B_2 & \dots & u^i_{2l}B_2 \\ \dots & \dots & & \dots \\ u^i_{l1}B_l & u^i_{l2}B_l & \dots & u^i_{ll}B_l \end{bmatrix}$; B_i's are as in Proposition 3 and u^i_{jk} are the elements of matrix U_i. ■

From now on, in order to compute the steady-state probability for the state lines of the target machine, we could apply any existing approach that computes the probabilities for the states of the product machine (*input SSM, target FSM*) which has the virtue of having temporally uncorrelated inputs. However, this approach could be very inefficient: the task of synthesizing the exact input SSM may require huge memory and computation time. Instead, we assume that the input is completely modeled by a *Dynamic Markov Tree* of order 1 (DMT_1) [10] as described next.

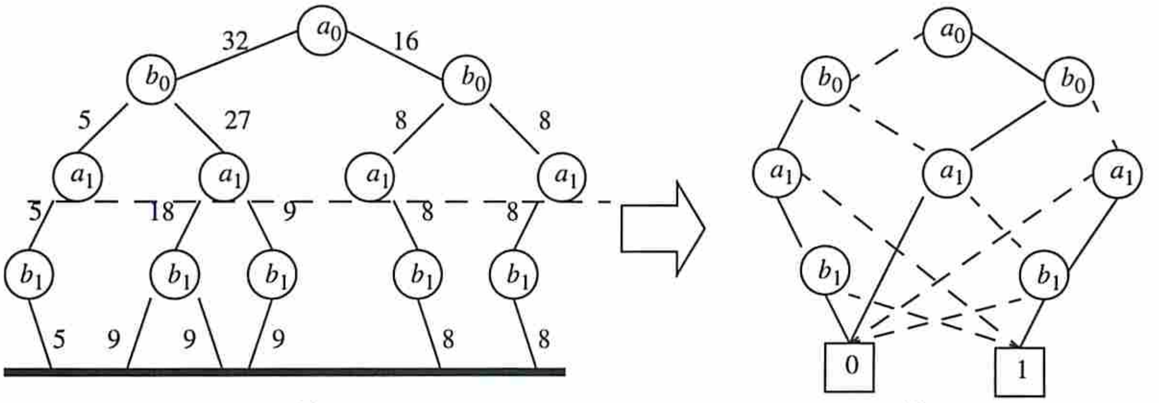
The DMT_1 model contains information about not only the possible binary vectors that can appear on the inputs of the FSM, but also the sequencing of these vectors. Additionally, the wordwise conditional probabilities for the primary inputs are easily extracted from such a model. The benefits of using DMT_1 for input modeling are threefold:

- first, the structure DMT_1 is constructed "on demand", therefore it offers a very compact representation;
- second, the model provides a set of parameters that completely characterizes the input sequences (that is, it completely captures spatiotemporal correlations);
- third, its structure is compatible with that of *Binary Decision Diagrams* (BDDs) [11] which were successfully used in reachability analysis for FSMs [12][13].

To see how these advantages can be exploited, let us consider the following example.

Example 3: For the sequence S_1 in Fig.1, the DMT_1 is given below. The node labels (a_0, b_0) represent the variables encoding the input bits for the current time step, whereas (a_1, b_1) correspond to the next time step. Left edges are associated with bits equal to 0, right edges to bits equal to 1. Each edge that enters a node is labeled with a positive count that represents the number of times the substring from the root to that particular node appeared in the original sequence. The upper half of this DMT_1 captures the statistics of order zero for the input sequence, whereas the lower half captures the sequencing between any two consecutive vectors. Based on this information and on the edge counts, we can easily compute the conditional or steady-state probabilities for the inputs. For example, p (00) = 5/48 (5 denotes the number of times 00 appears in the input sequence and 48 is the total length of the sequence) and p (01|01)

$= p\ (0101)\ /\ p\ (01) = 9/27$. Also, for the above DMT$_1$, the corresponding BDD is depicted in Fig.6b. Every possible combination with non-zero probability of occurrence in DMT$_1$ is part of the ON-set of the corresponding BDD; everything else, represents the OFF-set.



(a)

(b)

Fig.6: The tree DMT$_1$ and the corresponding BDD

The BDD corresponding to a given DMT$_1$ actually represents the transition relation δ of the machine that models the *input* to the target FSM (that is "Input SSM" in Fig.5). Having this representation for the input, our task is then to find all the reachable combinations (*input, state*). Let $B = \{0, 1\}$, N the number of primary input variables and δ: B^Nx $B^N \rightarrow B$ defined as $\delta\ (x^-, x^+) = 1$ if vector x^+ can follow x^- on the input modeled as a lag-one Markov chain (that is, if the corresponding conditional probability is non-zero) and zero otherwise. Also, let M be the number of state variables and *next*: B^Mx $B^N \rightarrow B^M$ be the next state function of the FSM. Then, we may employ the following standard procedure [13] to compute the set C of reachable combinations (*input, state*) for the given FSM and input characterization:

$C_0 = \{(x_0, s_0) \mid x_0$ is any possible initial input, s_0 is any possible initial state$\}$

$C_{i+1} = C_i \cup \{(x, s) \mid \exists\ (x', s') \in C_i$ s.t. $\delta\ (x', x) = 1$ and *next* $(s', x') = s\}$

$C = C_i$ if $C_i = C_{i+1}$.

Example 4: Let's assume that $C_0 = \{(01, 011)\}$ for circuit *dk*17. Applying the above algorithm, we get successively $C_1 = C_0 \cup \{(00, 100), (01, 100), (10, 100)\}$, $C_2 = C_1 \cup \{(01, 000), (00, 011), (10, 011), (11, 000)\}$,..., $C_7 = C_6 = C = \{(01, 000), (11, 000), (00, 001), (01, 001), (10, 001), (00, 011), (01, 011), (10, 011), (11, 011), (00, 100), (01, 100), (10, 100), (01, 101), (11, 101), (01, 110), (00, 111), (01, 111), (10, 111)\}$.

The above steps can be performed completely symbolically with the aid of BDDs [12][13]. After having the complete set C of possible combinations (*input, state*), we can easily build the matrix Q_{XS} based on Theorem 1 and Proposition 3. We should point out that in general, the matrix Q_{XS} 1) may have transient states or 2) may be decomposable. However, these can be dealt with in a similar way to the approach presented in [5] where: 1) transient states are eliminated and 2) the problem is reduced to finding the steady state distribution for each strongly connected component of the underlying Markov chain. In what follows we will thus refer only to irreducible Markov chains or matrices, that is those in which each state is reachable from any other state in a finite number of steps. This hypothesis has no theoretical limitation to be extended to the case of reducible MCs or MCs with multiple components.

5. Steady-state probability computation

We have constructed by now the correct matrix Q_{XS} (which completely characterizes the FSM behavior) and we are therefore ready to solve the basic equation $\pi \cdot Q_{XS} = \pi$ (with $\Sigma\pi_i = 1$). To this end, we present first an exact approach based an stochastic complementation and after that, because in practice Q_{XS} can be a fairly large matrix, we describe an approximate method based on iterative aggregation/disaggregation techniques.

5.1 Classical methods

Finding the stationary distribution of MCs with relatively few states is not a difficult problem and standard techniques based on solving systems of linear equations do exist [15]. In order to find the stationary distribution for a MC, one can always employ direct or iterative methods to solve the Chapman-Kolmogorov equations. The direct methods (e.g Gaussian elimination, LU decomposition) always provide the result in a fixed number of steps. On the other hand, iterative methods (e.g. power method, Jacobi, Gauss-Seidel) start with an initial approximation of the solution and iteratively improve on it. The number of iterations until convergence depends on the actual characteristics of the matrix. However, both types of methods work with the stochastic matrix as a whole. When the matrix size is large, we must resort to decompositional methods that try to solve smaller problems and then aggregate their solutions to find the needed stationary distribution.

5.2 Stochastic complementation

For large-scale problems, it is only natural to attempt to somehow uncouple (or decompose) the original MC into smaller chains (which are therefore easier to analyze) and finally, having these partial solutions, to produce the global stationary distribution that corresponds to the original chain. The stochastic complementation approach provides the theoretical basis for uncoupling Markov chains and furthermore, if applied on the stochastic matrix Q_{XS}, provides as a by-product the matrix Q_S as we shall see later in one of our results.

Definition 5. (stochastic complement) [16] Let Q be a $n \times n$ irreducible stochastic matrix partitioned as

$$Q = \begin{bmatrix} Q_{11} & Q_{12} & \cdots & Q_{1p} \\ Q_{21} & Q_{22} & \cdots & Q_{2p} \\ & \cdots & & \cdots \\ Q_{p1} & Q_{p2} & \cdots & Q_{pp} \end{bmatrix} \tag{9}$$

where all diagonal blocks are square. For a given index i, let Q_i denote the principal block submatrix of Q obtained by deleting the i^{th} row and i^{th} column of blocks from Q, and let Q_{i*} and Q_{*i} designate

$$Q_{i*} = (Q_{i1}Q_{i2}\cdots Q_{i,i-1}Q_{i,i+1}\cdots Q_{ip}) \qquad \text{and} \qquad Q_{*i} = \begin{bmatrix} Q_{1i} \\ \vdots \\ Q_{i-1,i} \\ Q_{i+1,i} \\ \vdots \\ Q_{pi} \end{bmatrix} \tag{10}$$

That is, Q_{i*} is the i^{th} row of blocks with Q_{ii} removed and Q_{*i} is the i^{th} column of blocks with Q_{ii} removed. The stochastic complement of Q_{ii} is defined to be the matrix

$$S_{ii} = Q_{ii} + Q_{i*} \cdot (I - Q_i)^{-1} \cdot Q_{*i} \tag{11}$$

where I is the unit matrix.

Example 5: For an irreducible stochastic matrix $Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}$ in which the diagonal blocks are square, the stochastic complement of Q_{11} is given by $S_{11} = Q_{11} + Q_{12} \cdot (I - Q_{22})^{-1} \cdot Q_{21}$ and the stochastic complement of Q_{22} is $S_{22} = Q_{22} + Q_{21} \cdot (I - Q_{11})^{-1} \cdot Q_{12}$.

It can be shown that every stochastic complement in Q is also an irreducible matrix. In addition, the following theorem has been proved [16]:

Theorem 2. Let Q be an $n \times n$ irreducible stochastic matrix as in (9) whose stationary probability vector π can be written as

$$\pi = (\xi_1 \Phi_1, \xi_2 \Phi_3, \ldots, \xi_p \Phi_p) \tag{12}$$

with $\Phi_i \cdot e = 1$ for $i = 1, 2, \ldots, p$[1]. Then Φ_i is the unique stationary probability vector for the stochastic complement S_{ii} and $\xi = (\xi_1, \xi_2, \ldots, \xi_p)$ is the unique stationary probability vector for the $p \times p$ irreducible stochastic matrix A (the *coupling matrix*) whose entries are defined by $a_{ij} = \Phi_i \cdot Q_{ij} \cdot e$.

The coupling matrix A corresponds to the Markov chain in which states belonging to the same set of the partition are collapsed in a single state. Thus, ξ describes the steady-state probability of being in such a set of states. Using this important theorem, the following exact procedure can be used to compute the stationary probability vector. (The input for this procedure is the Q matrix defined in Definition 5).

1. Partition Q into a $p \times p$ block matrix with square diagonal blocks.
2. Form the stochastic complement of each diagonal block:
   $S_{ii} = P_{ii} + P_{i*} (I - P_i)^{-1} P_{*i}$, $i = 1, 2, \ldots, p$.
3. Compute the stationary probability vector of each stochastic complement:
   $\Phi_i \cdot S_{ii} = \Phi_i$; $\Phi_i \cdot e = 1$, $i = 1, 2, \ldots, p$.
4. Form the coupling matrix A whose elements are given by:
   $a_{ij} = \Phi_i \cdot Q_{ij} \cdot e$.
5. Compute the stationary probability vector of A:
   $\xi \cdot A = \xi$; $\xi \cdot e = 1$.
6. Construct the stationary probability vector of Q as:
   $\pi = (\xi_1 \Phi_1, \xi_2 \Phi_2, \ldots, \xi_p \Phi_p)$.

Fig.7: The stochastic complementation algorithm

[1] e is a column vector defined as: $e = (1,1,\ldots,1)^T$; the product $Q_{ij} \cdot e$ does basically the sum of elements on each row of matrix Q.

Example 6: Let's consider the following matrix described in [17]:

$$Q = \begin{bmatrix} 0.85 & 0.0 & 0.149 & 0.0009 & 0.0 & 0.00005 & 0.0 & 0.00005 \\ 0.1 & 0.65 & 0.249 & 0.0 & 0.0009 & 0.00005 & 0.0 & 0.00005 \\ 0.1 & 0.8 & 0.0996 & 0.0003 & 0.0 & 0.0 & 0.0001 & 0.0 \\ 0.0 & 0.0004 & 0.0 & 0.7 & 0.2995 & 0.0 & 0.0001 & 0.0 \\ 0.0005 & 0.0 & 0.0004 & 0.399 & 0.6 & 0.0001 & 0.0 & 0.0 \\ 0.0 & 0.00005 & 0.0 & 0.0 & 0.00005 & 0.6 & 0.2499 & 0.15 \\ 0.00003 & 0.0 & 0.00003 & 0.00004 & 0.0 & 0.1 & 0.8 & 0.0999 \\ 0.0 & 0.00005 & 0.0 & 0.0 & 0.00005 & 0.1999 & 0.25 & 0.55 \end{bmatrix}$$

Assuming that the states are partitioned as {(1, 2, 3), (4, 5), (6, 7, 8)}, we compute the stochastic complements as in step 2. For example, $S_{11} = Q_{11} + [Q_{12} \;\; Q_{13}] \cdot \begin{bmatrix} I - Q_{22} & -Q_{23} \\ -Q_{32} & I - Q_{33} \end{bmatrix}^{-1} \cdot \begin{bmatrix} Q_{21} \\ Q_{31} \end{bmatrix} = \begin{bmatrix} 0.8503 & 0.0004 & 0.1493 \\ 0.1003 & 0.6504 & 0.2493 \\ 0.1001 & 0.8002 & 0.0997 \end{bmatrix}$ and its left eigenvector is $\Phi_1 = \begin{bmatrix} 0.4012 & 0.4168 & 0.1819 \end{bmatrix}$. Doing the same for all stochastic complements (S_{22} and S_{33}), we can proceed to steps 4 and 5 which compute the coupling matrix and its left eigenvector ξ. We get $A = \begin{bmatrix} 0.99911 & 0.00079 & 0.00010 \\ 0.00061 & 0.99929 & 0.00010 \\ 0.00006 & 0.00004 & 0.99990 \end{bmatrix}$ and ξ = [0.22333 0.27667 0.50000]. Using ξ and Φ_i's, we can now compute the exact stationary probability distribution as π = [0.0893 0.0928 0.0405 0.1585 0.1189 0.1204 0.2778 0.1018].

We point out that the analysis based on stochastic complementation does not depend in any way on the matrix Q being nearly completely decomposable and when implementing the stochastic complement approach, we may choose a partition that is convenient for us. For instance, based on the functionality of the FSM, we may partition the matrix Q_{XS} such that all combinations XS having the same S are clustered in the same block. The reason of doing so is that this way the matrix A in the stochastic complement algorithm becomes precisely the matrix Q_S (that is, the transition matrix associated with the state lines in the FSM) and ξ represents therefore the state occupancy probability we are looking for.

Corollary 2. If the stochastic complementation algorithm is applied to matrix Q_{XS} and the partitioning is done such that combinations (x_i, s_i), (x_j, s_j) are clustered in the same block if and only if $s_i = s_j$, then A is the matrix associated to the Markov chain for the state lines Q_S and the corresponding probability distribution is given by ξ.

Proof: According to the interpretation of the coupling matrix, A represents the stochastic matrix for the Markov chain when all states (in our case, (x_i, s_i) combinations) belonging to one subset are collapsed into a single state. Since (x_i, s_i) and (x_j, s_j) are in the same subset if and only if $s_i = s_j$, the states of the collapsed Markov chain are exactly $\{s_n\}_{n \geq 0}$, that is, the states of the target FSM. Also, their probability distribution is given by ξ.■

This important result justifies basically the applicability of stochastic complementation to FSM analysis. We also note that this method has the important feature of being *exact,* but unfortunately, in the worst case is computationally inefficient. More precisely, step 2 is the bottleneck because it involves the inversion of a large matrix, that is $(I - P_i)$. Its contribution lies primarily in the insight it provides into theoretical aspects of nearly decomposable systems.

5.3 Iterative aggregation/disaggregation

As shown in the previous section, the stochastic complement approach may be used to compute the stationary probability of an arbitrary irreducible stochastic matrix (not necessarily nearly completely decomposable), but the computational cost may be excessive. In this section, we present an iterative algorithm based on approximate decomposition that rapidly converges to the exact solution when the MC is nearly completely decomposable (NCD).

The pioneering work on NCD systems comes from Simon and Ando [18] who studied the dynamic behavior of linear systems. The concept was later extended to performance analysis of computer systems by Courtois [17]. The whole idea behind the dynamic behavior of NCD systems is the existence of two regimes:

- a *short-run dynamics,* when strong interactions within each subsystem are dominant and quickly force each subsystem to a local equilibrium almost independently of what is happening in the other subsystems;
- a *long-run dynamics*, when weak interactions among groups begin to become important and the whole system moves toward a global equilibrium; in this global equilibrium the relative values attained by the states at the end of the short-run dynamics period are maintained.

As a consequence, the state space of the global MC can be partitioned into disjoint sets, with strong interactions among the states of a subset but with weak interactions among the subsets themselves. This way, each subproblem can be solved separately and the global solution is then constructed from partial solutions. However, the assumption that the subsystems are independent and can therefore be solved separately does not usually hold. Consequently, an error arises but studies show that the error is small if the assumption is approximately true [14].

Iterative aggregation/disaggregation (IAD) methods are particularly useful when the global MC is NCD. More precisely, IAD methods work on partitioned state space as an *aggregation* (or *coupling*) step followed by a *disaggregation* one. The coupling step involves generating a stochastic matrix of block transition probabilities and then determining its stationary probability vector. The disaggregation step, computes an approximation to the probability of each state aggregated in the previous step within the same block. The basic iterative algorithm referred in the literature of numerical methods is called KMS (named after its authors Khoury-McAllister-Stewart) [14] and is described in Fig.8. (Once again, the input of the algorithm is the matrix Q as Definition 5.)

```
1. Let $\pi^{(0)} = (\pi_1{}^{(0)}, \pi_2{}^{(0)},...,\pi_p{}^{(0)})$ be a given initial approximation to the solution $\pi$, and set $m = 1$.
2. Compute $\Phi^{(m-1)}$: for $i = 1, 2,..., p$ do
      $\Phi_i^{(m-1)} = \pi_i^{(m-1)} / \|\pi_i^{(m-1)}\|_1$ (the norm is defined as: $\|x\|_1 = \sum_j |x_j|$).
3. Construct the aggregation matrix $A^{(m-1)}$ whose elements are given by:
      $(A^{(m-1)})_{ij} = \Phi_i^{(m-1)} . Q_{ij} . e$.
4. Solve the eigenvector problem:
      $\xi^{(m-1)} . A^{(m-1)} = \xi^{(m-1)}; \quad \xi^{(m-1)} . e = 1$.
5.(a) Compute the row vector:
      $z^{(m)} = (\xi_1{}^{(m-1)}\Phi_1{}^{(m-1)}, \xi_2{}^{(m-1)}\Phi_2{}^{(m-1)},\dots, \xi_p{}^{(m-1)}\Phi_p{}^{(m-1)})$.
  (b) Find $\pi^{(m)}$: for $k = 1, 2,..., p$ solve the system of equations:
      $\pi_k^{(m)} = \pi_k^{(m)} \cdot Q_{kk} + \sum_{j<k} \pi_j^{(m)} \cdot Q_{jk} + \sum_{j>k} z_j^{(m)} \cdot Q_{jk}$
6. Test for convergence: if the accuracy is sufficient, then stop and take $\pi^{(m)}$ as the solution vector;
      otherwise, set $m = m + 1$ and go to step 2.
```

Fig.8: The KMS algorithm

In this case, the partitioning criterion is purely numerical. States are aggregated within the same block if they interact strongly enough to favor short-run dynamics. This way, the off-diagonal elements in the global matrix are smaller than those in the blocks on the main diagonal and therefore, the interactions among subsets are minimized. In practice, finding the partitioning of a NCD stochastic matrix is not a trivial task. One way to do it would be to ignore the entries in the matrix that are less than some threshold ε and then find the strongly connected components of the underlying Markov chain. Next, the threshold may be increased and the same analysis is done on the components already found. We should note that the lower the threshold, the higher the rate of convergence, at the expense of larger blocks. On the other hand, if we proceed in partitioning the state space further until it becomes manageable, the convergence rate slows down due to the larger threshold used. More formally, under fairly general conditions[1], the following result hold for NCD stochastic matrices:

Theorem 3. [14] If the matrix Q is partitioned such that $\|Q_{ii}\|_1 = O(1)$ and $\|Q_{ij}\|_1 = O(\varepsilon)$ for $i \neq j$, then the error in the approximate solution using the iterative aggregation/disaggregation algorithm is reduced by a factor of ε at each iteration.

Example 7: Considering again the matrix from Example 6, we can see that taking 0.001 as a threshold value, we find three strongly connected components in the underlying MC: (1, 2, 3), (4, 5) and (6, 7, 8). Considering this partitioning for matrix Q (see the form of matrix Q in Example 6), applying the KMS algorithm produces the solution with error less than 10^{-10} in only 3 iterations. More specifically, considering as initial approximation $\pi^{(0)} = \begin{bmatrix}\frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8}\end{bmatrix}$, we get $\Phi_1^{(0)} = \begin{bmatrix}\frac{1}{3} & \frac{1}{3} & \frac{1}{3}\end{bmatrix}$, $\Phi_2^{(0)} = \begin{bmatrix}\frac{1}{2} & \frac{1}{2}\end{bmatrix}$ and $\Phi_3^{(0)} = \begin{bmatrix}\frac{1}{3} & \frac{1}{3} & \frac{1}{3}\end{bmatrix}$. Next, in step 3, based on these values, we can compute the entries of the approximate aggregation matrix.

For example, $(A^{(0)})_{11} = \Phi_1^{(0)} \cdot Q_{11} \cdot e = \begin{bmatrix}\frac{1}{3} & \frac{1}{3} & \frac{1}{3}\end{bmatrix} \cdot \begin{bmatrix} 0.85 & 0.0 & 0.149 \\ 0.1 & 0.65 & 0.249 \\ 0.1 & 0.8 & 0.0996 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = 0.9992$. Doing this for all entries, we get the following aggregation matrix: $A^{(0)} = \begin{bmatrix} 0.9992 & 0.0007 & 0.0001 \\ 0.00065 & 0.9992 & 0.00015 \\ 0.000053 & 0.00016 & 0.99978 \end{bmatrix}$ which is also a stochastic matrix. Its stationary distribution is given by $\xi^{(0)} = [0.2979 \;\; 0.3369 \;\; 0.3651]$ and thus, in step 5(a) we compute the row vector:

$$z^{(1)} = (\xi_1^{(0)}\Phi_1^{(0)} \;\; \xi_2^{(0)}\Phi_2^{(0)} \;\; \xi_3^{(0)}\Phi_3^{(0)}) = [0.0993 \;\; 0.0993 \;\; 0.09993 \;\; 0.1684 \;\; 0.1684 \;\; 0.1217 \;\; 0.1217 \;\; 0.1217].$$

In step 5(b) we solve for the block components of $\pi^{(1)}$. For example, ${\pi_1}^{(1)}$ is the solution of the equation:

$\pi_1^{(1)} = \pi_1^{(1)} \cdot Q_{11} + \sum_{j=2}^{3} z_j^{(1)} \cdot Q_{j1}$. Doing this for all components of $\pi^{(1)}$, we get as a first approximation for π: $\pi^{(1)} =$ [0.0874 0.0908 0.0396 0.1564 0.1174 0.1184 0.2731 0.1001]. After this step, the error of this estimate (defined as $\|\pi Q - \pi\|_2{}^2$) is 9.3581e-6, that is, less than 10^{-5}. Continuing this process, the error becomes less than 10^{-10} after 3 iterations.

[1] A detailed analysis on convergence is presented in [14].

In practice, the KMS algorithm offers the attractive feature of working on individual blocks instead of the whole matrix Q_{XS}. As a consequence, its complexity per iteration is given by $O(n_{max}{}^3)$, where n_{max} is the maximum order over all diagonal blocks in the partitioned matrix Q_{XS}. Compared to the classical power method (which is also iterative in nature) [5], the KMS algorithm has the significant advantage of being applicable to *any irreducible* MC (aperiodic or periodic) and also having a *higher rate of convergence* for NCD systems. In these cases, a few iterations will suffice for all practical purposes. We should point out that we can always trade-off space versus time complexity: if the partitioning is such that n_{max} is still too large, we can use a higher value for the threshold such that the size of the largest subset (i.e. n_{max}) becomes manageable. In this case, the convergence rate will be smaller and thus, the time needed for convergence will increase. Also, the size of the coupling matrix will be larger and hence step 4 in KMS algorithm becomes critical. However, there is a solution to this latter problem: in solving the Chapman-Kolmogorov equations for the coupling matrix, we can apply the KMS algorithm in a recursive manner. This approach is called the *hierarchical KMS algorithm*.

6. Experimental results

In this section we provide our experimental results for some common benchmark circuits. In particular, we compare the probability distribution for the states where the order of the input sequence is assumed to be one, against the case where the actual order of the source is taken into consideration.

We provide in Table 1 the results obtained for stochastic complementation when a Fibonacci sequence (that is, a second-order sequence) is applied at the primary inputs of some *mcnc*'91 and ISCAS'89 benchmarks. We were forced to consider at this point only small benchmarks because, as pointed out in Section 5, the exact method based on stochastic complementation is computationally inefficient. For each example, an appropriate dynamic Markov tree was built and based on it, a sequence-driven reachability analysis has been performed. Using the obtained set of reachable combinations (*input, state*) (denoted by #(x, s)), and sparse matrix techniques, the matrix Q_{XS} has been built and further used in the stochastic complementation method presented in Section 5. The partitioning was determined by the same functional criterion as in Corollary 2. We also report for each benchmark the number of reached states (#s) and their corresponding probabilities.

In Table 2, we report our results obtained when applying KMS algorithm for an extended set of benchmarks. For comparison, we also provide the results when the input is considered as having order one. For each case, the sequence-driven reachability analysis is carried out as above and then, using the Q_{XS} matrix, the KMS algorithm is applied using a numerical partitioning criterion with a threshold of 0.001. In all cases, we report the number of (*input, state*) combinations reached and the number of reached states. The number of iterations needed to converge for an error less than 10^{-5} was ≤ 3 for the second-order model and between 7 and 15 for the first-order model. For the larger sized matrices, the hierarchical KMS algorithm was used. For comparison, for first-order models we also provide the maximum and mean percentage errors obtained when comparing the results of lower order models to the actual order two model (MAX% and MEAN%).

[2] By definition, $\|x\|_2 = \sqrt{\sum_j x_j^2}$.

Table 1: Steady-state distributions obtained with stochastic complementation for second-order input sequences

Circuit	PIs/FFs	#(x,s)	#s	State probability distribution
bbara	4/4	28	3	[0.5 0.25 0.25]
bbtas	2/3	18	6	[0.0556 0.3333 0.3333 0.0556 0.0556 0.1667]
dk17	2/3	6	2	[0.6667 0.3333]
donfile	2/5	9	5	[0.1667 0.1667 0.1667 0.1667 0.3333]
s400	3/21	12	3	[0.6667 0.1667 0.1667]
s526	3/21	17	9	[0.3333 0.0833 0.0833 0.0833 0.0833 0.0833 0.0833 0.0833 0.0833]

Table 2: First-order model vs. the second-order model using the KMS algorithm

		Second-order model		First-order model			
Circuit	PIs/ FFs	#(x,s)	#s	#(x,s)	#s	MAX%	MEAN%
bbara	4/4	28	3	86	10	49.40	35.91
bbtas	2/3	18	6	20	6	55.70	24.28
dk17	2/3	6	2	18	7	37.48	29.96
donfile	2/5	9	5	14	6	52.70	26.76
ex1	9/5	769	11	1596	11	50.00	18.03
planet	7/6	192	34	3527	48	513.79	67.14
sand	11/5	977	32	17169	32	103.57	24.49
s1196	14/18	1536	329	1918	342	133.71	4.17
s1238	14/18	1538	330	1919	343	134.82	4.21
s1494	8/6	384	6	1372	40	28.69	8.92
s400	3/21	8	3	8	3	0.00	0.00
s526	3/21	17	9	12155	4137	97.17	43.03
s820	18/5	680	8	3056	24	24.78	10.78

As we can see, considering the input of the target FSM as having only one-step temporal correlations may significantly impair the ability of predicting the correct number of reached states. Also, for the subset of states correctly found as being reached, there is still a significant error in the value of steady-state probabilities. For example, when excited using a second-order type of input, benchmark *planet* has a number of 34 reached states, whereas if the order is incorrectly assumed one, the number of reached states becomes 48. Moreover, the error in predicting the steady-state probability can be as high as 513% for the first-order model. Generally speaking, a lower order model covers all the states from the original one, but it may also introduce a significant number of extra states and, furthermore, the quality of the results in estimating the steady-state probabilities is seriously impaired.

7. Conclusion

In this paper we investigated from a probabilistic point of view the effect of finite-order statistics of the input sequence on FSMs behavior. In particular, the effect of temporal correlations longer that one clock-cycle was analyzed for steady-state and transition probabilities calculations. The results presented in this paper can be used in synthesis and verification and represent an important step towards understanding the FSM behavior from a probabilistic point of view.

References

[1] K. Parker and E.J. McCluskey, 'Probabilistic Treatment of General Combinational Networks', in *IEEE Trans. on Computers*, vol. C-24, June 1975.

[2] G. Hachtel, E. Macii, A. Pardo, and F. Somenzi, 'Probabilistic Analysis of Large Finite State Machines', in *Proc. ACM/IEEE Design Automation Conference*, pp. 270-275, June 1994.

[3] J. Savir, G.S. Ditlow, and P.H. Bardell, 'Random Pattern Testability', in *IEEE Trans. on Computers*, vol. C-33, Jan. 1984.

[4] G. Hachtel, M. Hermida, A. Pardo, M. Poncino and F. Somenzi, 'Reencoding sequential circuits to reduce power dissipation', in *Proc. ACM/IEEE Intl. Conf. Computer-Aided Design,* pp. 70-73, Nov. 1994.

[5] G.D. Hachtel, E. Macii, A. Pardo, and F. Somenzi, 'Markovian Analysis of Large Finite State Machines', in *IEEE Trans. on CAD*, vol.15, no.12, Dec. 1996.

[6] C.-Y. Tsui, J. Monteiro, M. Pedram, S. Devadas, A. M. Despain, and B. Lin, 'Power Estimation Methods for Sequential Logic Circuits', in *IEEE Trans. on VLSI Systems*, vol.3, no.3, Sept. 1995.

[7] J.G. Kemeny, J.L. Snell, and A.W. Knapp, 'Denumerable Markov Chains', D.Van Nostrand Company, Inc., 1966.

[8] A. Papoulis, 'Probability, Random Variables, and Stochastic Processes', McGraw-Hill Co., 1984.

[9] D. Marculescu, R. Marculescu, and M. Pedram, 'Stochastic Sequential Machine Synthesis Targeting Constrained Sequence Generation', in *Proc. ACM/IEEE Design Automation Conference*, pp. 696-701, June 1996.

[10] R. Marculescu, D. Marculescu, and M. Pedram, 'Adaptive Models for Input Data Compaction for Power Simulators', in *Proc. IEEE Asian-South Pacific Design Automation Conference*, Japan, Jan. 1997.

[11] R. E. Bryant, 'Graph Based Algorithms for Boolean Function Manipulation', in *IEEE Trans. on Computers*, vol. C-35, no. 8, pp. 677-691, Aug. 1986.

[12] O. Coudert, C. Berthet, and J.C. Madre, 'Verification of Sequential Machines Based on Symbolic Execution', in *Proc. of the Workshop on Automatic Verification Methods for Finite State Systems*, Grenoble, France, 1989.

[13] H.J. Touati, H. Savoj, B. Lin, R.K. Brayton, and A. Sangiovanni-Vincentelli, 'Implicit State Enumeration of Finite State Machines Using BDDs', in *Proc. ACM/IEEE Intl. Conf. Computer-Aided Design,* pp. 130-133, Nov. 1990.

[14] R. Koury, D.F. McAllister, and W.J. Stewart, 'Methods for Computing Stationary Distributions of Nearly-Completely-Decomposable Markov Chains', *SIAM Journal of Algebraic and Discrete Mathematics*, vol.5, no.2, pp. 164-186, 1984.

[15] G.W. Stewart, 'Introduction to Matrix Computations', Academic Press, New York, 1973.

[16] C.D. Meyer, 'Stochastic Complementation, Uncoupling Markov Chains and the Theory of Nearly Reducible Systems', *SIAM Reviews*, vol.31, no.2, pp. 240-272, 1989.

[17] P.J. Courtois, 'Decomposability: Queueing and Computer System Applications', Academic Press, New York, 1977.

[18] H.A. Simon and A. Ando, 'Aggregation of variables in dynamic systems', *Econometrica*, vol.29, pp. 111-138, 1961.