Probabilistic State Space Search

Andreas Kuehlmann
IBM T. J. Watson Research Center
Yorktown Heights, NY, U.S.A.

Kenneth L. McMillan
Cadence Design Systems
Berkeley, CA, U.S.A.

Robert K. Brayton
University of California at Berkeley
Berkeley, CA, U.S.A.

Abstract

This paper describes a probabilistic approach to state space search. The presented method applies a ranking of the design states according to their probability of reaching a given target state based on a random walk model. This ranking can be used to prioritize an explicit or partial symbolic state exploration to find a trajectory from a set of initial states to a set of target states. A symbolic technique for estimating the reachability probability is described which implements a smooth trade-off between accuracy and computing effort. The presented probabilistic state space search complements incomplete verification methods which are specialized in finding errors in large designs.

1 Introduction

A key problem in functional design verification is to decide whether a set of target states can be reached from a set of initial states, and if so, to compute a trajectory to demonstrate it. A large number of techniques and refinements have been developed to make reachability analysis practical for larger designs. They can be classified in three major categories: (1) Inductive methods use a given invariant to prove that no initial state can reach a target state. They avoid the need for state space exploration and can utilize the rich set of practical SAT solvers. However, the intrinsic difficulty in determining a sufficient invariant significantly limits the application of inductive methods and they fail completely if a target state is reachable. (2) Forward state space traversal techniques explicitly or symbolically execute the machine from the initial states. If the enumeration of new states saturates without encountering a target, unreachability is proven, otherwise the execution trace provides a trajectory from an initial to a target state. (3) Similarly, backward state space traversal starts from the target states and symbolically executes the machine in reverse mode to check whether an initial state can be reached backwards.

The application of state space traversal techniques for hardware designs is fundamentally handicapped by the potentially exponential gap between the design size and the number of states. Explicit methods [1] execute one state transition at a time and can utilize the rich bag of tricks from simulation based verification methods. Unless special reduction rules are applicable, their usage is limited to designs with a small number of reachable state transitions. Symbolic methods [2, 3] are based on compact representation and manipulation techniques for large sets of states and are less sensitive to the actual number of reachable transitions or states. However, depending on the size and structure of the state space, the symbolic representation may grow excessively large which limits its predictable application to smaller designs.

A number of approximation techniques have been suggested to address the capacity limitations of explicit and symbolic state traversal techniques. Overapproximating the set of reachable states provides a conservative method that is useful in proving unreachability if all target states remain outside the approximation. In [4] an overapproximation is described which is based on partitioning the design structure into subcomponents that are tractable for symbolic traversal. For the computation of the reachable states of the individual components the actual correlation of their interface signals is disregarded, which results in additional global state transitions and potentially additional reachable states. In [5] a refinement of this approximation approach using overlapping subcomponents is described which yields a significantly tighter bound. A combination of overapproximating the set of reachable states and exact backward traversal is described in [6], however this approach is fundamentally limited by the exact backward traversal step. Overall, overapproximation techniques are useful if unreachability can be proven, but they are unable to demonstrate true reachability if a target state lies in the approximation.

Conversely, methods to underapproximate the set of reachable states are used to discover reachable targets whereas they are incapable of proving unreachability. A straightforward underapproximation can be computed by

0-7803-5832-X /99/$10.00 ©1999 IEEE.

a partial explicit or symbolic state traversal. For example, symbolic simulation [7] is used to explore the state space to a fixed depth. In order to reduce the memory requirements in explicit state space traversal, hash compaction for tracking the reached states was introduced at the expense of potentially missing parts of the state space [8]. In [9] a pruning method for the BDD representing the set of reachable states is described. Its application during state space traversal is targeted for "sparse" subgraphs of the original BDD which require more BDD nodes to represent a comparable number of states. All of the above mentioned methods do not consider the actual target states for constraining the state space exploration. Since in many practical applications only a tiny fraction of the design space can be traversed, this shortcoming significantly reduces the value of those approaches.

More recent research in the area of incomplete reachability analysis focuses on methods to guide the exploration specifically toward the target states. Guided state space search requires a scoring mechanism that prioritizes the state traversal according to the potential to reach a target. In [10] a method to exercise all transitions of the control part of the design is described. It is assumed that this will expose all functional corner cases and therefore increase the chance of hitting a target state. Based on the same concept, a technique called saturated simulation is presented in [11]. After applying an incomplete forward and backward traversal a state distance function is used to search for a "short" trajectory between the two resulting sets of states. The proposed method applies the Hamming distance of the state encodings to measure their distance in state space. In [12, 13] an extension of the explicit model checker Murφ to guide the state space traversal is described. Here the reported results on using the Hamming distance to guide the search confirm the practical intuition that this metric does not correlate with the actual distance in state space and is therefore of limited value in this context. A second proposed distance function is based on the length of the counter example trajectory in the overapproximated state transition graph using overlapping subcomponents. This metric provides a better scoring function for the search, but due to the assumed unconstrained execution of the individual subcomponents the accuracy decreases dramatically with increasing distance from the target states.

In this paper we describe a probabilistic approach for guiding the state space search. The scoring of the individual states is based on their reachability probability to lead to a target state in a random walk model. Since the accurate computation of this probability is as complex as the original reachability decision, an estimation technique is proposed which implements a trade-off between accuracy and computing effort. In contrast to previous methods in the area of guided state space search, the presented work models the actual probabilistic nature of the search process and uses it to focus the search toward target areas. For practical verification tools, the described probabilistic state space search provides a building block that can effectively complement other heuristics. This paper is focused on the actual probabilistic method and will not elaborate on its combination with other methods.

2 Reachability Probability

A random simulation run can be characterized by a Markov Chain [14] with a one-to-one correspondence of its states to the states of the FSM model of the design. For a random walk the events at each input are chosen randomly and independently. Therefore, the transition probability p_{ij} between states i and j corresponds to the fraction of the Boolean input space that exercises this transition in the FSM. More precisely:

$$p_{ij} = \frac{|T(x,s,s')_{s=i,s'=j}|}{|T(x,s,s')_{s=i}|}$$

where $|.|$ denotes the cardinality of a set, $T(x,s,s')$ represents the transition relation of the FSM from the current states s to the next states s' for inputs x, and the subscript denotes a projection onto the given values. The given definition for p_{ij} can easily be modified for a non-equal distribution of the inputs. Let P be the probability matrix composed of all p_{ij}. Note, that for the given definition the following invariant holds:

$$\forall i : \sum_{\forall j} p_{ij} = 1.$$

To denote the set of target states, we introduce a vector $t = (t_1, \cdots, t_k)^T$ for the k states of the FSM, where $t_i = 1$ if i is a target, 0 otherwise. Similarly, we use $\bar{t}$ to denote the non-target states. We are interested in the probability $r_i^{(m)}$ that a random simulation run of length m starting from state i reaches a target state. Let $r^{(m)} = (r_1^{(m)}, \cdots, r_k^{(m)})^T$ be the vector of the reachability probability for all states. $r^{(m)}$ cannot be determined by simply computing the sum over all powers of P. This would double count recurring visits of target states and yield incorrect results. Instead we compute:

$$r^{(m)} = \sum_{n=0}^{m} \mathcal{P}^n \; t$$

where $\mathcal{P} = \bar{t}\;\bar{t}^T P$. In essence, this scheme first removes the outgoing transitions from all target states and then adds the individual probabilities to reach a target in n steps.

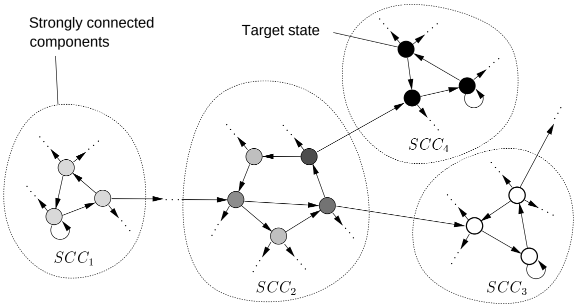

Figure 1: General distribution of the probability $r_i^{(\infty)}$ to reach a target state (black: $r_i^{(\infty)} = 1$, white: $r_i^{(\infty)} = 0$, gray: intermediate).

Note, that for $m = \infty$ the given formula resembles the traditional backward reachability analysis: If the summation is replaced by disjunction and $p_{ij} = 1$ for all nonzero transition probabilities, the resulting Boolean vector $r^{(\infty)}$ denotes the reachability of a target from the individual states.

A random walk model based on a simulation run with a particular length m causes a dilemma: A fixed finite length will tailor the model toward a specific exploration depth and mask targets that are beyond that limit. An infinite length of the simulation run is not realistic and reduces the discrimination power of the model. As illustrated in Figure 1, the variation of $r^{(\infty)}$ is based on the strongly connected components (SCC) of the state transition graph and the transitions between them. For example, assuming that SCC_4 constitutes a terminal SCC without outgoing edges, the $r_i^{(\infty)}$ of its states cannot be distinguished. However, the transition from SCC_2 to SCC_3 induces a "leakage" of reachability probability and causes a discrimination of the internal states of SCC_2.

A more accurate modeling of a random walk is based on a set of finite simulation runs with a given distribution of their lengths. Basically any distribution could be used. In this context we assume an exponential distribution for which the number of simulation steps m is given by the following density function:

$$d(m) = \frac{1-\eta}{\eta}\,\eta^m \quad \text{for } m > 0 \text{ and } \eta < 1, \text{ constant}$$

where $E = \frac{1}{1-\eta}$ is the mean of the number of simulation steps. Let r_i be the probability that state i reaches a target in the given simulation setting. The following formula computes the reachability probability $r = (r_1, \cdots, r_k)^T$ for all states:

$$\begin{aligned} r &= \frac{1-\eta}{\eta} \sum_{m=1}^{\infty} \eta^m \sum_{n=0}^{m} \mathcal{P}^n \; t \\ &= t + \frac{1}{\eta} \sum_{m=1}^{\infty} \eta^m \mathcal{P}^m \; t. \end{aligned}$$

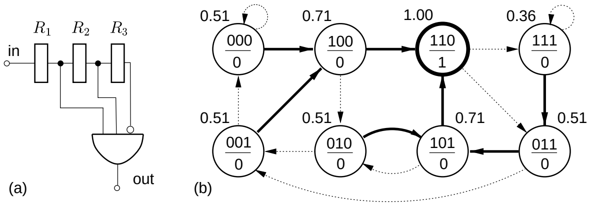

Figure 2: Circuit example of a shift register with signature detection: (a) circuit structure, (b) state transition graph with probabilities to reach target state (110).

Note, that this scheme is similar to the "discounting" method applied in dynamic programming [15].

As an example, Figure 2 shows design example consisting of an n-bit shift register with signature detection. This general structure poses a particular challenge for many reachability algorithms. The reachability probabilities of the drawn state transition graph are based on a uniform transition probability of 0.5 and an expected mean simulation length of $E = 6$. To demonstrate the effect of a directed search scheme, all transitions that compose a trajectory to the target state with increasing reachability probabilities are drawn in bold.

The computation of the reachability probability can be done symbolically using the "analog" version of BDDs [16]. Due to their algorithmic similarity, many heuristics developed for symbolic state traversal (e.g., variable ordering) can be adopted to the symbolic computation of the reachability probability resulting in comparable application ranges for the design size. In section 4 we describe an approximation algorithm for the reachability probability using overlapping subcomponents.

3 State Space Search Guided by Reachability Probability

The reachability probabilities of the states have an interesting property that can be exploited in guiding a state space search toward target states:

Theorem:

$$\forall i: \; (t_i = 0 \wedge r_i > 0) \;\Rightarrow\; r_i < \max_{\forall j,\; p_{ij} > 0} r_j.$$

Proof:

For all non-target states with a non-zero reachability probability the computation of the reachability probability yields:

$$r_i < \sum_{\forall j} p_{i,j}\; r_j \leq \sum_{\forall j} p_{i,j} \big(\max_{\forall j,\; p_{ij} > 0} r_j \big) = \max_{\forall j,\; p_{ij} > 0} r_j.$$

■

In other words, for all states that can reach a target, there exists a trajectory from that state to a target state such that the reachability probability of the states along this trajectory is strictly monotonically increasing. Once a target is reached, the probability is 1.0 and cannot further increase. If a state cannot reach a target, the reachability probability of that state and its successors is zero. The algorithm of Figure 3 outlines a guided state space search based on this monotonic property.

```
Algorithm Search_State_Space ($init, target$) {
      $present$ = get_states_with_max_probability ($queue$);
      $next$    = get_partial_image ($present, r$);
      if (($next \cap target$) != $empty$) return reachable;
      add_to_queue ($queue, next, r$);
  }
  return undecided;
}
```

Figure 3: State space search guided by reachability probability.

The algorithm is based on a priority queue which holds the states that are to be further explored. The queue is prioritized by the reachability probabilities of the entries. If no initial states has a nonzero reachability probability, `unreachability` is proven and returned. Otherwise, in each iteration, the states with the highest priority are further explored by generating a set of successor states which are added back to the queue. If a target state is encountered during the search, the algorithm returns `reachable`, otherwise `undecided`. The function of the priority queue is to heuristically avoid dead ends during the guided state space search which can be caused by the inaccuracy of an approximated state ranking or the incomplete image computation.

4 Approximation of the Reachability Probability

The calculation of the exact reachability probabilities is computationally as hard as the original reachability decision problem and therefore it does not simplify its solution. However, in the context of incomplete verification methods such as directed search, an approximation of the reachability probabilities, is valuable as long as it preserves the monotonicity of the probabilities along a trajectory to a target from as many states as possible. In the following we describe an approximation technique that is based on a set of subcomponents of the design for which the computation of the reachability probability is tractable.

Similar to [4, 5], a subcomponent of a hardware design is defined as a subset of its state registers and their next state functions. The resulting bisection of the state registers defines a division of the set of global states s into two parts $\hat{s}$ and $\check{s}$ such that $s = \hat{s} \times \check{s}$. Let $\hat{s}$ and $\hat{x}$ be the states and inputs of the subcomponent, respectively. Further, let $\tau : (x \times \check{s}) \rightarrow \hat{x}$ denote the mapping of the global inputs and "environment" states onto the subcomponent inputs. Using capital letters for individual inputs and states, the transition relation $\hat{T}$ of the subcomponent is derived as follows:

$$\hat{T}(\hat{x}, \hat{s}, \hat{s}') = \{(\hat{X}, \hat{S}, \hat{S}') \mid \quad \exists \check{S} : \hat{X} = \tau(X, \check{S}) \wedge \\ (X, (\hat{S}, \check{S}), (\hat{S}', \check{S}')) \in T\}.$$

Let $\pi : s \rightarrow \hat{s}$ denote the projection of each global state onto its corresponding state of the subcomponent. Similar to the formula of Section 2, the projected transition probability $\hat{p}_{ij}$ between two global states i and j is defined as:

$$\hat{p}_{ij} = \frac{|\hat{T}(\hat{x}, \hat{s}, \hat{s}')_{\hat{s}=\pi(i), \hat{s}'=\pi(j)}|}{|\hat{T}(\hat{x}, \hat{s}, \hat{s}')_{\hat{s}=\pi(i)}|}.$$

$\hat{t} = (\hat{t}_1, \cdots, \hat{t}_k)^T$ denotes the projected target vector such that:

$$\hat{t}_i = \begin{cases} 1 & \text{if } \exists j \ (t_j = 1 \wedge \pi(i) = \pi(j)) \\ 0 & \text{otherwise .} \end{cases}$$

Informally, a state of the subcomponent is a projected target state if there is at least one global state that is projected onto it. Similar to section 2, let $\hat{P}$ and $\hat{\mathcal{P}}$ denote the projected probability matrix and its adjusted version without outgoing transitions from projected targets, respectively. This results in the following scheme to compute the projected reachability probability for a subcomponent:

$$\hat{r} = \hat{t} + \frac{1}{\eta} \sum_{m=1}^{\infty} \eta^m \hat{\mathcal{P}}^m \ \hat{t}.$$

The computation of the global approximated reachability probability is based on a set of subcomponents, which are not necessarily disjoint. Let $(\hat{r}^1, \cdots, \hat{r}^q)$ denote the projected reachability probabilities of the q components. The approximate reachability probability $\tilde{r} = (\tilde{r}_1, \cdots, \tilde{r}_k)$ is defined as:

$$\forall i : \tilde{r}_i = \prod_{j=1}^{q} \hat{r}_i^j.$$

In other words, the approximate reachability probability of a state is computed by the product of the projected reachability probabilities of the subcomponents. Using the product to combine the individual probabilities assumes that the subcomponents operate independently, neglecting that their structural composition constrains possible transitions. However, this heuristic scheme is based on the fact that a higher reachability probability reflects a greater chance that the environment of a subcomponent can cooperate for a given transition.

In practice, the reachability probability of a subcomponent can be directly computed from the subset of state

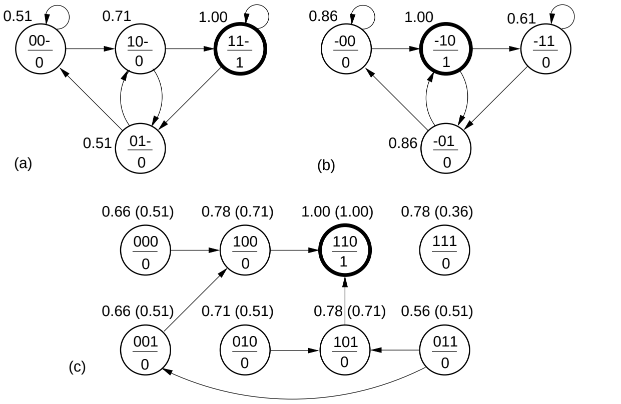

Figure 4: Approximation of the reachability probability of the circuit of Figure 2 using two subcomponents: (a) and (b) state transition graphs and projected reachability probabilities $\hat{r}$ for the subcomponents $\{R_1, R_2\}$ and $\{R_2, R_3\}$, respectively, (c) full state transition graphs and resulting approximate reachability probabilities $\tilde{r}$ (r are given in brackets).

registers belonging to the component and the corresponding next state functions. Moreover, the generation of the subcomponents can be combined with the construction of its transition relation. By incrementally adding one state bit at a time, the BDD size of the transition relation can be monitored and used to control the actual size of the component by setting a limit on the memory requirements. This scheme ensures a maximum size of the subcomponents for which the projected reachability probabilities can be computed with given resources.

For a practical application in a directed search algorithm, the projected reachability probabilities $\hat{r}_i$ are precalculated. However, the computation of the approximate reachability probability $\tilde{r}$ is deferred until it is needed to evaluate a state or set of states. An up front computation for all states is computationally too expensive and would diminish the benefits of the partitioned computation. A concrete state is evaluated by first projecting it onto the subcomponents. For the resulting state projections, the values of the vector $(\hat{r}^1, \cdots, \hat{r}^q)$ are determined and then combined to the global approximate reachability probability of this particular state.

Figure 4 illustrates the resulting approximate reachability probability for the design of Figure 2 using two subcomponents. To illustrate the effect of a directed search, all transitions that do not preserve the strict monotonicity of the reachability probabilities are removed from the global state transition graph (c). As shown, except for state (111) the remaining seven states would still reach the target.

An extreme case of the approximate reachability probability is based on the complete set of single bit subcomponents. This essentially results in a guidance metric similar to the Hamming distance. In addition to just counting the "non-matching" state bits, this scheme also weights them according to their probability to assume the "matching" value in the future. This is a surprisingly meaningful extension of the Hamming distance metric and illustrates the overall validity of the proposed method. As mentioned before, a symbolic computation of the reachability probabilities can be based on fairly large subcomponents, resulting in a significantly higher accuracy of the global approximation. Figure 5 gives the resulting approximate reachability probability for the example of Figure 2 based on three single bit subcomponents. As shown, the accuracy is considerably lower, resulting in three states that do not have a strictly monotonic trajectory to the target.

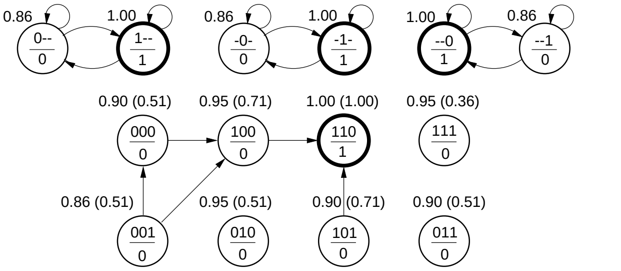

Figure 5: Approximation of the reachability probability of the circuit of Figure 2 using three subcomponents, resembling a guidance metric based on Hamming distance.

The tracking approach presented in [13] can be viewed as another special case of the presented method. Assuming a uniform transition probability of 1.00 and a simulation distribution consisting of a single long simulation run, the state ranking resulting from the approximate reachability probability resembles the results of the tracking method.

5 Results

For an evaluation of the effectiveness of the presented technique, we compared the approximate reachability probability with the exact values for several instances of the shift register example of Figure 2 with random signatures. To demonstrate the influence of the subcomponent size on the accuracy of the approximation, we performed this experiment with different sizes of the subcomponents. During each experiment, these components were systematically generated by selecting a consecutive subset of state bits of the shift register.

First, we computed the exact and the approximate reachability probabilities for all states. Then we determined for each state, the successor state with the highest probability using the exact and the approximate metric and count the instances for which the successor states are identical. This number provides an insight into the effectiveness of the approximate reachability probability in preserving potential trajectories to target states in a directed search scheme.

Table 1 reports the fraction of the transitions that are identical for both metrics. As shown, the size of the subcomponents significantly influences the accuracy of the approximate reachability probability. The results also confirm the intuitive ineffectiveness of the Hamming distance metric which is similar to a subcomponent size of one.

	Subcomponents size (bits)							
	1	2	3	4	5	6	7	8
4	68.7	68.7	81.2	100				
8	62.9	87.9	87.9	87.9	97.3	98.8	93.4	100
12	72.6	78.9	78.9	78.9	86.7	82.8	92.9	96.1
16	67.0	84.6	84.6	86.9	96.2	92.4	92.4	96.2

Table 1: Fraction (in %) of identical highest probable next states in a guided search using the exact and approximate reachability probability for various sizes of the shift register example.

6 Conclusions and Future Work

The paper presents a probabilistic approach for guiding a state space search from a set of initial states to a set of target states. The described technique uses a metric which is based on the reachability probability that a given state can get to a target state in a set of random simulation runs. Since the exact computation of the reachability probability is intractable, a symbolic approximation technique is presented that implements a smooth trade-off between accuracy and computing effort.

The resulting probabilistic metric provides a significantly better discrimination scheme for ranking states in a guided search than previously published approaches. Further, several existing methods such as the Hamming distance metric, tracking metric, and reachability approximations based on design partitioning or projections fold into the presented model as special cases, yielding a more global view on a wide set of distance metrics.

Future work will mainly focus on improving the accuracy and efficient computation of the presented approximation technique and its effective combination with other incomplete verification methods. In particular, we are interested in techniques for probabilistic design block characterization that allow a compositional approach for guided state space search. Further, we are investigating methods that gather statistics while performing the state space search with the objective to correct the approximate reachability probabilities on the fly.

7 Acknowledgments

The authors would like to thank Luca de Alfaro and Amit Mehrotra from UC Berkeley for the valuable input.

References

[1] C. H. West and P. Zafiropulo, "Automated validation of a communications protocol: the CCITT X.21 recommendation," *IBM Journal of Research and Development*, vol. 22, pp. 60–71, January 1978.

[2] O. Coudert, C. Berthet, and J. C. Madre, "Verification of synchronous sequential machines based on symbolic execution," in *International Workshop on Automatic Verification Methods for Finite State Systems*, June 1989.

[3] J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill, and J. Hwang, "Symbolic model checking: 10^{20} states and beyond," in *Proceedings of the Fifth Annual Symposium on Logic in Computer Science*, June 1990.

[4] H. Cho, G. D. Hachtel, E. Macii, B. Plessier, and F. Somenzi, "Algorithms for approximate FSM traversal," in *Proceedings of the 30th ACM/IEEE Design Automation Conference*, (Dallas, TX), pp. 25–30, IEEE, June 1993.

[5] S. G. Govindaraju, D. L. Dill, A. J. Hu, and M. A. Horowitz, "Approximate reachability with BDDs using overlapping projections," in *Proceedings of the 35th ACM/IEEE Design Automation Conference*, (San Francisco), pp. 451–455, ACM/IEEE, June 1998.

[6] G. Cabodi, P. Camurati, and S. Quer, "Symbolic exploration of large circuits with enhanced forward/backward traversals," in *Proceedings of The European Conference on Design Automation*, pp. 22–27, IEEE, February 1994.

[7] R. E. Bryant, "Symbolic simulation - techniques and applications," in *Proceedings of the 27th ACM/IEEE Design Automation Conference*, (Orlando), pp. 517–521, ACM/IEEE, June 1990.

[8] G. J. Holzmann, "On limits and possibilities of automated protocol analysis," in *Proceedings of the 7th International Conference on Protocol Specification, Testing, and Verification*, pp. 339–44, 1987.

[9] K. Ravi and F. Somenzi, "High-density reachability analysis," in *Digest of Technical Papers of the IEEE International Conference on Computer-Aided Design*, (San Jose), pp. 154–158, ACM/IEEE, November 1995.

[10] R. C. Ho, C. H. Yang, M. A. Horowitz, and D. L. Dill, "Architecture validation for processors," in *Proceedings of the 22nd Annual International Symposium on Computer Architecuture*, pp. 404–423, ACM, June 1995.

[11] J. Yuan, J. Shen, J. Abraham, and A. Aziz, "On combining formal and informal verification," in *Proceedings of the 9th International Conference on Computer Aided Verification*, (Haifa, Israel), pp. 376–387, Springer, June 1997.

[12] C. H. Yang and D. L. Dill, "SpotLight: Best-first search of FSM state space," in *IEEE International High Level Design Validation and Test Workshop (HLDVT'96)*, November 1996.

[13] C. H. Yang and D. L. Dill, "Validation with guided search of the state space," in *Proceedings of the 35th ACM/IEEE Design Automation Conference*, (San Francisco), pp. 599–604, ACM/IEEE, June 1998.

[14] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains.* Princeton, New Jersey: Princeton University Press, 1994.

[15] D. Bertsekas, *Dynamic Programming and Optimal Control.* Athena Scientific, 1995. Volumes I and II.

[16] R. I. Bahar, E. A. Frohm, C. M. Gaona, G. D. Hachtel, E. Macii, A. Pardo, and F. Somenzi, "Algebraic decision diagrams and their applications," in *Digest of Technical Papers of the IEEE International Conference on Computer-Aided Design*, (Santa Clara, CA), pp. 188–191, IEEE, November 1993.