Implicit representations and algorithms for the logic and stochastic analysis of discrete–state systems*

Gianfranco Ciardo

Department of Computer Science and Engineering
University of California, Riverside
Riverside, CA 92521, USA
ciardo@cs.ucr.edu

As discrete–state systems are pervasive in our society, it is essential that we model and analyze them effectively, both prior to putting them in operation and during their useful life. The size of their state space, however, is a huge obstacle in practice. Often, the "easy" way to tackle this problem is to use some type of simulation, but this technique has obvious limitations. For performance analysis, simulation can at best offer only a statistical approximation, i.e., confidence intervals, while, for logic analysis, the situation is even worse, as it can only find errors, not prove correctness. Ultimately, these limitations stem from the same source: simulation only visits a fraction of the reachable states. Indeed, the fraction of the states that can actually be explored in a reasonable amount of time becomes exponentially smaller as the complexity of the system being modeled (measured in number of components, parts, etc.) increases.

One way to attack this problem is to employ *implicit representations* whose memory and time requirements are often much less than linear in the number of states of the system under study. In this context, two techniques emerged in the mid '80s are particularly relevant.

- Since the introduction of *binary–decision diagrams* (BDDs) [1], *symbolic* algorithms have proven themselves very effective for the verification of discrete–state systems, especially digital hardware and protocols. Systems with 10^{20} or more states have become amenable to exact logic analysis, leading to the wide adoption of *model checking* by major hardware and software vendors.
- In an apparently distant area of research, the structure of a Markovian discrete system has been exploited to develop a *Kronecker algebra* representation of the infinitesimal generator matrix of the system [11]. By avoiding the need to store this matrix, which is the largest data structure required for a steady–state or transient analysis of a Markov model, systems with a factor of 10 or more states can be studied than with ordinary sparse–storage methods. While this improvement is not as impressive as that of symbolic model checking, the idea has spurred much research and further advancements in the two decades following its introduction.

* Work supported in part by the National Science Foundation under grants CNS-0501747 and CNS-0501748.

BDDs, Kronecker representations, and several other variants that have been introduced in the literature can all be seen as *implicit* representations of vector or matrices, as opposed to *explicit* representations; the fundamental difference between the two being that, when using the latter, the memory, hence time, requirements are always at least proportional to the number of (nonzero) entries in the vector or matrix being stored, while the former is enormously more efficient in many cases.

In this talk, we first survey and organize several types of decision diagrams which can be employed for logic or Markov analysis, and briefly illustrate important algorithms that manipulate them, from fixed–point algorithms used in state–space generation [3, 7] and CTL model checking [6] to vector–matrix multiplications [2] needed for the numerical solution of a Markov model: *multiway decision diagrams* (MDDs) [12], *multi–terminal decision diagrams* (MTBDDs, MTMDDs) [8], *additive or multiplicative edge–valued decision diagrams* (EVBDDs, EV^{+}MDDs and EV^{*}MDDs) [9, 5], *matrix diagrams* (MxDs) [4, 10], and a new variant particularly efficient when storing matrices, *identity–reduced decision diagrams*.

Then, we focus on the need for approximate solutions of Markov models that have been encoded implicitly. This is an important challenge to be tackled, because, while we can analyze the logic behavior and even encode the infinitesimal generator of systems with huge state spaces, the exact numerical solution of the underlying Markov model still requires the storage of an explicit probability vector in practice. We present some known results, and discuss potential research directions.

References

1. R. E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Trans. Comp.*, 35(8):677–691, Aug. 1986.
2. P. Buchholz, G. Ciardo, S. Donatelli, and P. Kemper. Complexity of memory-efficient Kronecker operations with applications to the solution of Markov models. *INFORMS J. Comp.*, 12(3):203–222, 2000.
3. G. Ciardo, R. Marmorstein, and R. Siminiceanu. Saturation unbound. In H. Garavel and J. Hatcliff, editors, *Proc. Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, LNCS 2619, pages 379–393, Warsaw, Poland, Apr. 2003. Springer-Verlag.
4. G. Ciardo and A. S. Miner. A data structure for the efficient Kronecker solution of GSPNs. In P. Buchholz, editor, *Proc. 8th Int. Workshop on Petri Nets and Performance Models (PNPM'99)*, pages 22–31, Zaragoza, Spain, Sept. 1999. IEEE Comp. Soc. Press.
5. G. Ciardo and R. Siminiceanu. Using edge-valued decision diagrams for symbolic generation of shortest paths. In M. D. Aagaard and J. W. O'Leary, editors, *Proc. Fourth International Conference on Formal Methods in Computer-Aided Design (FMCAD)*, LNCS 2517, pages 256–273, Portland, OR, USA, Nov. 2002. Springer-Verlag.
6. G. Ciardo and R. Siminiceanu. Structural symbolic CTL model checking of asynchronous systems. In W. Hunt, Jr. and F. Somenzi, editors, *Computer Aided*

Verification (CAV'03), LNCS 2725, pages 40–53, Boulder, CO, USA, July 2003. Springer-Verlag.
7. G. Ciardo and J. Yu. Saturation-based symbolic reachability analysis using conjunctive and disjunctive partitioning. In *Proc. CHARME*, Saarbrücken, Germany, Oct. 2005. Springer-Verlag. To appear.
8. M. Fujita, P. C. McGeer, , and J. C.-Y. Yang. Multi-terminal binary decision diagrams: an efficient data structure for matrix representation. *Formal Methods in System Design*, 10:149–169, 1997.
9. Y.-T. Lai, M. Pedram, and B. K. Vrudhula. Formal verification using edge-valued binary decision diagrams. *IEEE Trans. Comp.*, 45:247–255, 1996.
10. A. S. Miner. Efficient solution of GSPNs using canonical matrix diagrams. In R. German and B. Haverkort, editors, *Proc. 9th Int. Workshop on Petri Nets and Performance Models (PNPM'01)*, pages 101–110, Aachen, Germany, Sept. 2001. IEEE Comp. Soc. Press.
11. B. Plateau. On the stochastic structure of parallelism and synchronisation models for distributed algorithms. In *Proc. ACM SIGMETRICS*, pages 147–153, Austin, TX, USA, May 1985.
12. A. Srinivasan, T. Kam, S. Malik, and R. K. Brayton. Algorithms for discrete function manipulation. In *Int. Conference on CAD*, pages 92–95. IEEE Comp. Soc. Press, 1990.