published in *Proceedings of the 4th Workshop on Tools for System Design and Verification (FMTOOLS'2000)*, Reisenburg (Germany), 2000, pp. 195-196.

Functional Extension of Decision Diagrams in Practice

Harald Sack, Christoph Meinel
FB IV - Informatik
Universität Trier
D-54286 Trier
{sack,meinel}@uni-trier.de

a b s t r a c t

In computer aided design of very large scale integrated circuits (CAD for VLSI) Ordered Binary Decision Diagrams (OBDDs) [1] have been established as the state-of-the-art data structure. They are applied in synthesis as well as in formal verification of combinatorial or sequential designs. This is due to the fact that almost every design step can be mapped to the task of manipulating Boolean functions. For performing these tasks efficiently in an automated way with a computer, OBDDs are very well suited, because they are compact, efficiently to manipulate, and canonical, i.e. there exists a unique OBDD for every Boolean function. The compactness property of OBDDs holds for most Boolean functions that are used in practice, but unfortunately not for all. The multiplication of two binary encoded numbers can only be represented with an OBDD of exponential size related to the number of inputs. This restriction is responsible for the research and development of more general data structures, based on extensions of OBDDs.

Besides relaxing the ordering restriction [2], easing the read-once property of the input variables, or the usage of different decomposition types for Boolean functions, we are focusing on the extension of OBDDs with functional operator nodes, esp. Parity-OBDDs (POBDDs), i.e. OBDDs with additional operator nodes computing the Boolean parity of their successors [3]. By introducing parity nodes the representation has the potential of being more compact while on the other hand giving up canonicity. Therefore, the identification of two POBDDs representing the same Boolean function becomes an essential operation. We present an efficient probabilistic equivalence test for POBDDs that admits working with POBDDs in an professional environment [4]. Due to the fact that the size of Decision Diagrams crucially depends on the order of the input variables we show how to apply heuristics for POBDD minimization based on dynamic changes in the variable order and the relocation of parity operator nodes inside the data structure.

Many problems in practice require the transformation of symbolic variables to a binary encoding for getting accessible with OBDDs or POBDDs. Extending the OBDD data structure from the binary domain to a finite domain results in so called Multi-valued Decision Diagrams (MDDs) [5] and a binary encoding of symbolic variables is not necessary anymore. The already introduced POBDDs can now be extended towards Mod-p-Decision Diagrams (Mod-p-DDs), i.e. MDDs with additional operator nodes representing an integer addition modulo p, p -

prime. Such decision diagrams have a potential of being more space-efficient than MDDs. However, they are not a canonical representation and thus, the equivalence test of two Mod-p-DDs is more difficult than the test of two MDDs. To overcome this problem, we design a fast probabilistic equivalence test for Mod-p-DDs based on the transformation of integer functions represented by Mod-p-OBDDs to polynomials over a finite domain [6] and show how to apply heuristics for their minimization.

References

[1] R.E. Bryant, Graph-based algorithm for Boolean function manipulation, *IEEE Transactions on Computers* **C-35** No. 8 (1986), 677-691.

[2] J. Gergov, C. Meinel, Frontiers of feasible and probabilistic feasible Boolean manipulation with branching programs, *Proc. 10th Annual Symp. on Theoretical Aspects of Computer Science*, **665** of LNCS, Springer (1993), 576-585.

[3] J. Gergov, C. Meinel, Mod2-OBDDs: A data structure that generalizes EXOR-sum-of-products and Ordered Binary Decision Diagrams, *in Formal Methods in System Design* **8** (1996), Kluwer Academic Publishers, 273-282.

[4] C. Meinel, H. Sack, $\oplus$-OBDDs - a BDD structure for probabilistic verification, *in Proc. Workshop on Probabilistic Methods in Verification (PROBMIV'98)* (1998), 141-151.

[5] D. M. Miller, Multiple-valued logic design tools, *Proc. 23rd Int. Symp. on MVL* (1993), 2-11.

[6] E. Dubrova, H. Sack, Probabilistic verification of multiple-valued functions, Tech. Report 99-23, University of Trier, FB IV-Informatik, Nov. 1999.