Extended Abstracts of the 38th Iranian International Conference on Mathematics
3-6 September 2007, University of Zanjan, Zanjan, Iran, pp 377-379

ACCEPTING MORE GENERAL FORMS OF QBFS

MOHAMMAD GHASEMZADEH[1] AND CHRISTOPH MEINEL[2*]

ABSTRACT. Almost all existing QSAT solvers only accept QBFs represented in prenex-CNF. Formulating problems into QBFs usually does not yield prenex-CNF. Therefore, they are required to transform the obtained formula into prenex-CNF before launching it to a QSAT solver. This task usually cannot be done efficiently. In this paper we show how Zero-Suppressed Binary Decision Diagram (abbreviated by ZBDD or ZDD) can be used to represent QBFs given in prenex-NNF and evaluate them efficiently. Transforming a QBF into its equivalent in prenex-NNF usually can be done efficiently.

1. INTRODUCTION

Quantified Boolean formula (QBF) is a language that extends propositional logic in such a way that many advanced forms of reasoning could be formulated and evaluated easily [1]. For this purpose, the problem must first be translated into a QBF. For example Rintannen has shown how conditional planning problems can be translated into QBFs. The next step is solving the obtained formula. Almost all existing QBF evaluators (also known as QSAT solvers) accept QBFs

2000 *Mathematics Subject Classification.* Primary 68Q15; Secondary 68Q25.
Key words and phrases. Quantified Boolean Formula (QBF), Zero-Suppressed Binary Decision Diagram (ZDD), Satisfiability, Negation Normal Form (NNF).

* Ghasemzadeh.

represented in prenex-CNF. Therefore one needs to transform the obtained formula into its equivalent prenex-CNF. This transformation may be done in two ways: the structure preserving transformation or the application of distributivity laws. The time and space complexity of the 'structure preserving normal form transformation' method is polynomial in the size of the formula, but it produces a lot of new variables. This property usually lifts the amount of time needed to evaluate the resulting formula considerably. The second method, i.e. the application of the distributivity laws does not produce new variables but in its worst case it is exponential time and the size of the formula can grow up exponentially. On the other hand, transforming any prenex QBF into its equivalent in prenex-NNF can be done efficiently without producing new variables. This fact led us to look for a method and a suitable data structure in which we could represent prenex-NNF formulas directly and perform the evaluation step directly and efficiently.

ZDD (Zero-Suppressed Binary Decision Diagram) is a variant of BDD (Binary decision diagram). While BDDs are more suited for representing Boolean functions, ZDDs are better suited for storing and implementing sets of subset. This data structure has been used successfully in a number of research works. We found this data structure also suitable for representing and evaluating QBFs (given in prenex-CNF or prenex-NNF). In our research, we first implemented a QSAT solver based on ZDDs and an adopted version of the DPLL algorithm for solving QBFs given in standard prenex-CNF format. It was comparable and in many cases much faster than the existing methods. Afterwards, we invented a method in order to accept prenex-NNF formulas, represent them directly in a ZDD and evaluate them efficiently.

2. Accepting NNF formulas can be Exponentially Beneficial

There are some small size Boolean functions where their equivalents in CNF have exponential size in the number of variables. We know that processing of an exponential size formula is at least exponential time. We can show that there is a straight forward way to represent those formulas in linear time using ZDDs [2].

Theorem 2.1. (1) *There are Boolean functions* $f : \{0,1\}^n \rightarrow \{0,1\}$ *in NNF with linear size,* $size(f) = n$*, whose logically equivalent minimal CNF representation* f' *are of exponential size,* $size(f') = 2^{\frac{n}{2}} \cdot \frac{n}{2}$.

(2) f *can be represented by ZDDs of linear size* n *and there exists a synthesis algorithm for their ZDDs such that each ZDD occurring during the synthesis process of* f *would never be larger than* n.

The proof of this theorem is based on two lemmas which can be obtained from the authors.

References

[1] T. Eiter, V. Klotz, H.Tompits and S. Woltran, *Efficient Encodings for the Basic Reasoning Tasks*, Proc. of the Eleventh Conf. on Automated Reasoning, 2002.

[2] M. Ghasemzadeh, V. Klotz, and Ch. Meinel, *Splitting Versus Unfolding*, Int. Symp. on Representations and Methodology of Future Computing Technologies, (Tokyo, Japan), 2005, pp. 30–34.

[1] Department of Computer, Yazd University, P. O. Box 89195-741 , Yazd, Iran.

E-mail address: m.ghasemzadeh@yazduni.ac.ir

[2] Hasso Plattner Institute, University of Potsdam, P. O. Box 900460, 14440 Potdam, Germany.

E-mail address: meinel@hpi.uni-potsdam.de