Computer Aided Design and Test: BDDs vs. SAT

28.01.2001 - 02.02.2001

organized by

B. Becker (Freiburg)
M. Fujita (Tokyo)
C. Meinel (Trier)
F. Somenzi (Boulder)

The focus of the sixth workshop in the biannual series *Computer Aided Design and Test* at the IBFI Schloß Dagstuhl was on **BDDs vs. SAT**. The seminar was organized by Bernd Becker (University Freiburg), Masahiro Fujita (University of Tokyo), Christoph Meinel (University Trier), and Fabio Somenzi (University of Colorado). It was attended by 44 scientists.

While after 10 years use of BDDs various BDD-based algorithms have been developed and BDD-techniques have seen dramatic improvements only recently, SAT based techniques are reconsidered with respect to their usability in Electronic Design Automation and in other applications.

The organizers took the opportunity to bring together researchers from different areas in computer science, electrical engineering and industry. During the seminar 31 lectures covering different aspects of the topic were presented and the seminar provided a forum for scientific discussion e.g. on

- both approaches, also on comparisons among various approaches to SAT,
- the advances in BDD and SAT algorithms,
- comparisons between BDDs and SAT for various applications e.g., model checking,
- hybrid approaches that use BDDs and SAT, and
- other approaches to the decision of Boolean formulae.

As always, Schloß Dagstuhl and its staff provided a very convenient and stimulating environment for the workshop. The organizers wish to thank all of those who helped in establishing this excellent research atmosphere.

More detailed information including some full papers can be found on the WWW-pages with the URL:

- `http://www.dagstuhl.de/DATA/Seminars/01/`
- `http://www.bdd-portal.org/dagstuhl-ppt/dagstuhl-talks.htm`

Contents

Schedule

	Monday, 29. January 2001
Masahiro Fujita	Simultaneous Instruction Set Synthesis and Code Generation for Application Specific Processor Based on Finite State Model
Henrik Reif Andersen	Satisfiability checking of BED's
Jim Kukula	When Is SAT Hard?
Per Bjesse	Finding bugs in an Alpha microprocessor using satisfiability solvers
Rainer Feldmann	A Distributed Algorithm to Evaluate Quantified Boolean Formulae
Geert Janssen	A Pointerless BDD Implementation
Rolf Drechsler	Implementation of Read-k-times BDDs

	Tuesday, 30. January 2001
Armin Biere	Beyond BDD based and SAT based Model Checking
Richard Raimi	SAT-Based Model Checking Vs. Traditional Sequential ATPG: A case study with examples
Forrest Brewer	Automata Based Scheduling: MIPS Case Study
Pawel Kerntopf	Decision Diagrams based on New Generalizations of Shannon Expansion
Detlef Sieling	Lower Bounds for Linearly Transformed OBDDs and FBDDs
Beate Bollig	Improved OBDD and FBDD Lower Bounds for Integer Multiplication via Universal Hashing
Martin Sauerhoff	The Wonderful World of Partitioned BDDs
Harald Sack	Heuristics for $\oplus$-OBDD Minimization
Gianpiero Cabodi	Meta-BDDs: A decomposed representation for layered symbolic manipulation of Boolean functions

	Wednesday, 31. January 2001
Aarti Gupta	SAT-Based Image Computation with Application in Reachability Analysis
Ted Stanion	Can SAT Approximate Free BDDs?
Lintao Zhang	A Fast SAT Solver for EDA Applications
Rustam Mubarakzjanov	Lower Bounds on Complexity of Probabilistic Branching Programs
Christoph Meinel	Invitation: www.bdd-portal.org - A Place for Cooperated BDD-Research

Thursday, 01. February 2001	
Eugene Goldberg	A New Method of Checking Satisfiability in Propositional Logic
Christoph Scholl	Checking Equivalence for Partial Implementations
Andreas Kühlmann	Combining BDDs and SAT for Equivalence Checking
Luis Baptista	New Algorithms for Solving Satisfiability in Formal Verification
Ziyad Hanna	Efficient BDD size reduction methods for combinational equivalence verification
Marc Herbstritt	Dynamic Selection of Branching Rules
Alan Hu	Stochastic Planning Using Decision Diagrams

Friday, 02. February 2001	
Karem A. Sakallah	SAT Using ZBDDs
Christian Stangier	A New Partitioning Scheme for Improving Image Computation
Ingo Wegener	On the Complexity of OBDD Synthesis with Optimal Reordering

1 Satisfiability checking of BED's

Henrik Reif Andersen The IT University of Copenhagen, Denmark

(joint work with Paul Frederick Williams and Henrik Hulgaard)

This talk presented an algorithm for determining satisfiability of Boolean formulas which are not necessarily on conjunctive normal form. The algorithm extends the well-known Davis-Putnam algorithm to work on Boolean formulas represented using Boolean Expression Diagrams (BEDs). The BED data structure allows the algorithm to take advantage of the built-in reduction rules and the sharing of sub-formulas. Furthermore, it is possible to combine the algorithm with traditional BDD construction (using Bryant's APPLY-procedure). By adjusting a single parameter — turning a knob — it is possible to control to what extent the algorithm behaves like the APPLY-algorithm or like a SAT-solver. Thus the algorithm can be seen as bridging the gap between SAT-solvers and BDDs.
Promising experimental results where shown for 566 non-clausal formulas obtained from the multi-level combinational circuits in the ISCAS85 benchmark suite and from performing model checking of a shift-and-add multiplier.

References

[1] `www.it.edu/research/bed/`

2 When Is SAT Hard?

Jim Kukula Synopsys Inc., Beaverton, USA

The difficulty of deciding satisfiability, and of generating a witness, varies greatly across problem instances of the same size. We report here on a set of experiments performed to test two hypotheses about features of instances that determine their difficulty. The first hypothesis is that difficult instances have large bandwidth, the second that their ratio of clauses to variables is near some critical threshold. We randomly generated a set of satisfiable instances, directly controlling the clause / variable ratio and using an underlying mesh to indirectly control bandwidth. We generated a thousand instances at each parameter setting, measuring difficulty by observing the runtime of either Chaff

or Walksat. The distributions of runtimes at fixed parameter settings were observed to have heavy tails, which can make problematic simple statistical measures like average or standard deviation. Therefore we report 90th percentile runtimes for each parameter setting. The main observations from the experiments are that the hard instances had both large bandwidth and a critical ratio clause to variables, that Walksat was much faster than Chaff for these problems (in contrast to what we have observed for typical CAD-derived instances), and that Chaff is much more sensitive than Walksat to topology.

3 Finding bugs in an Alpha microprocessor using satisfiability solvers

Per Bjesse Chalmers University, Göteborg, Sweden

We present the approach we have used to find bugs in the memory subsystem of a next-generation microprocessor. Our methodology is based on two methods that use satisfiability solvers.
The first of these two methods, bounded model checking, has reduced the time necessary for finding certain bugs from days to minutes, when compared to state-of-the-art BDD-based model checking. The second method, symbolic trajectory evaluation based on SAT solvers, can find as deep bugs as bounded model checking with negligible runtimes. The trade off is that we have to spend more time writing specifications.
We also present a methodology for using these methods in heady duty industrial verification.

4 A Distributed Algorithm to Evaluate Quantified Boolean Formulae

Rainer Feldmann University Paderborn, Paderborn, Germany

(joint work with Burkhard Monien, Stefan Schamberger)

We present Psolve, a distributed theorem-prover for Quantified Boolean For-

mulae.
First, we introduce our sequential algorithm Qsolve. We show how to use known heuristics from SAT-solvers and QSAT-solvers and develop new heuristics to prune the search space. As a result, Qsolve is more efficient than the QSAT-solvers previously known.
We have parallelized Qsolve. The resulting distributed QSAT-solver Psolve uses parallel search techniques, which we have developed for distributed game tree search. Psolve runs efficiently on distributed systems, i.e. parallel systems without any shared memory.
We present experiments on randomly generated formulae as well as on formulae which describe problems from the field of autoepistemic logic. Besides their differing structures, Psolve runs with a speedup of about 110 on 128 processors on both classes of formulae.

5 A Pointerless BDD Implementation

Geert Janssen IBM, Yorktown Heights, USA

Inspired by the ICCAD'98 paper [1] of David Long, I have redesigned a BDD package with the intention to achieve full platform independence. For this to work, BDD nodes can no longer be identified by machine addresses (C pointers).
This talk will discuss the implications of this fundamental design decision w. r. t. the choice of data structures and algorithms. I will highlight some of the problems that I have encountered and explain the solutions that I have adopted. An interesting area which was not addressed by the cited paper, is how to implement dynamic variable ordering in the new context.

References

[1] David E. Long, "The design of a Cache-Friendly BDD library", ICCAD, 1998.

6 Implementation of Read-k-times BDDs

Rolf Drechsler Siemens AG, München, Germany

(joint work with Wolfgang Günther, University of Freiburg, Germany)

Ordered Binary Decision Diagrams (OBDDs) are the state-of-the-art data structure in VLSI CAD for representation and manipulation of Boolean functions. But due to the ordering restriction, many Boolean functions cannot be represented efficiently. As one alternative read-k-times BDDs have been proposed. They are a generalization of OBDDs in the way that variables may occur up to k times on each path, while they may only occur once in OBDDs. More functions can be represented by read-k-times BDDs in polynomial space than by OBDDs, while many operations, like synthesis and satisfiability, still have polynomial worst case behavior.

We present a new technique for implementation of read-k-times BDD packages on top of standard OBDD implementations. Thus, highly optimized OBDD packages can be used and only few changes in the code are needed, while the new type of decision diagram allows much smaller representations. Experimental results are given to demonstrate the efficiency of the approach.

7 Beyond BDD based and SAT based Model Checking

Armin Biere ETH Zürich, Zürich, Switzerland

In recent years there has been an increasing interest in applying technology from the domain of Satisfiability Checking (SAT) to the model checking problem. One of the starting points was Bounded Model Checking (BMC). This SAT based technique helps to tackle certain large designs where traditional BDD based symbolic traversal techniques fail. However, while BDD based methods are gaining more and more acceptance, though for model checking smaller design, SAT based methods do not deliver the same degree of robustness on the same range of models. In particular SAT is often only used for finding bugs and fails to actually prove desired properties. We tried to argue, that one of the reasons is the incompleteness of SAT based methods from a practical point of view.

In this talk we gave our vision on how to merge techniques from Automatic

Test Pattern Generation (ATPG), i.e. sequential ATPG, with techniques from the SAT community in order to bring SAT based model checking closer to completeness and as consequence increase its robustness. As key ingredients of our new methodology, termed Complete Bounded Model Checking (CBMC), we identified propagation of justification frontiers to derive partial assignments of state variables in combination with a clause data base for fast unit propagation and adding of clauses. Relevance learning and conflict directed backtracking, which were recently shown to have a large impact on the performance of SAT tools, fit nicely into our framework. We also presented a tool set for synthesis and model checking that is based on the SMV language and we gave a demo of our initial implementation.

8 SAT-Based Model Checking Vs. Traditional Sequential ATPG: A case study with examples

Richard Raimi BOPS Inc., Austin, USA

SAT-based model checking appears to be highly superior to BDD-based model checking for reachability checks over short, bounded time frames, in that it can handle designs with many more state variables. But, not much work has been done to compare bounded model checking to a method of state reachability checking which has been in use for a number of years, this being sequential ATPG.

In this talk, I review experiments in which certain restrictions are imposed on the commercial, Sunrise sequential ATPG tool such that it checks whether a state setting a given combinational circuit node to a 1 is reachable from an initial state of all state variables (i.e., latches) set to 0. The circuits used were from the ISCAS benchmark series. The results of Sunrise are compared to those of a bounded model checker using backward traversal (i.e., starting at a state where the combinational node is 1 and working backwards to the intiial state). On 21 problems, bounded model checking with SAT was able to find a solution 18 times, Sunrise only 9 times. The bounded model checker found the given state unreachable 9 times, Sunrise only 6, and the bounded model checker was able to find input sequences leading to the desired state 9 times, Sunrise only 3. The 3 sequences Sunrise found were all much longer than the bounded model checker found for the same problems: 3, 4, and 8 cycles long for the bounded model checker, 12, 15 and 15 cycles for Sunrise.

While these experiments indicate that SAT-based model checking may be su-

perior for simple state reachability checking to sequential ATPG tools, they do not necessarily mean that SAT-based model checking can be used to solve the entire sequential ATPG problem. Further experiments the author did to add on propagation sequences (propagating the combinational node's value to an observable circuit output) proved disappointing. However, the encouraging results on simple state reachability checking give hope that bounded model checking may eventually be used in sequential ATPG.

9 Automata Based Scheduling: MIPS Case Study

Forrest Brewer University California, Santa Barbara, USA

Automata based scheduling techniques comprise a new synthesis technique for assembling NFA models with sequential and functional constraints. The techniques are a superset of existing scheduling algorithms in that they support all existing forms of dependency, concurrency and sequential constraints while providing a design scale unmatched by alternative techniques. In this talk, we describe the application of ABS to the MIPS-4 processor via behavioural synthesis of all optimal pipelined schedules. ABS makes use of NFA representations for memory cache behaviour and for interface sequencing constraints as well as conventional functions unit behaviours. To our knowledge this is the first exact scheduler for general looping, control dominated behaviour. Moreover, it can exactly schedule over 470000 control paths — an improvement of 4 orders of magnitude from earlier results. In the future, we hope these techniques prove applicable to commercial CAD flows and a create a path to practical HLS.

10 Decision Diagrams based on New Generalizations of Shannon Expansion

Pawel Kerntopf Warsaw University, Warsaw, Poland

During the last 10 years many new decision diagrams have been proposed in search for a better representation of Boolean functions. The key idea behind proposing the variants like FDDs, KFDDs, EVBDDs, BMDs, etc. is relaxing

limitations imposed on BDDs. The above mentioned variants of decision diagrams were based on the new modifications of the so-called Shannon's expansion of a Boolean function:

$$f = \overline{x}_i f_i^0 + x_i f_i^1,$$

where x_i is a variable of f, f_i^0 and f_i^1 are subfunctions of f obtained by replacing x by constants 0 and 1, respectively.
The author has proposed new generalizations based on the following formula:

$$f = \overline{g}_i f_{i,g}^0 + g_i f_{i,g}^1,$$

where

$$\begin{aligned} f_{i,g}^o &= F(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) \\ f_{i,g}^1 &= F(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) \\ F &= f(x_1, \ldots, x_{i-1}, g_i, x_{i+1}, \ldots, x_n) \end{aligned}$$

and g_i is a Boolean function over variable set $\{x_1, \ldots, x_n\}$ with the property

$$g(x_1, \ldots, x_{i-1}, \overline{x}_i, x_{i+1}, \ldots, x_n) = \overline{g}(x_1, \ldots, x_n)$$

called self-duality with respect to x_i.
A function-driven decompositon type list has been introduced:

$$g = (g_1, g_2, \ldots, g_n)$$

where $g_i(x_1, \ldots, x_n)$ is a self-dual function for $i = 1, 2, \ldots, n$ and g is an independent set of functions. Decision diagrams (called function-driven DDs) have been defined on the basis of a function-driven decomposition type list. Preliminary results of minimization of HWB functions of small number of variables has been presented.

11 Lower Bounds for Linearly Transformed OBDDs and FBDDs

Detlef Sieling University Dortmund, Dortmund, Germany

Linearly Transformed Ordered Binary Decision Diagrams (LTOBDDs) have been suggested as a generalization of OBDDs for the representation and manipulation of Boolean functions. Instead of variables as in the case of OBDDs,

linear tests, i.e. tests of parities of variables, may be performed at the nodes of an LTOBDD, where an ordering of the linear tests has to be respected. By this extention it is possible to represent functions in polynomial size that do not have polynomial size OBDDs, e.g. the characteristic functions of linear codes. We present a method for proving exponential lower bounds for LTOBDDs, and apply this method to an explicitly defined function. The method also works for several variants of LTOBDDs, e.g., the linearly transformed variants of $\oplus$-OBDDs or OFDDs.
We also consider two possibilities to introduce linear transformations into FBDDs (Free Binary Decision Diagrams) and call the resulting variants of FBDDs LTFBDDs and strong LTFBDDs. We separate these two variants by proving a polynomial upper bound for strong LTFBDDs and an exponential lower bound for LTFBDDs for a modified version of the matrix storage access function. By all the upper and lower bound results we also separate the classes of functions with polynomial size LTOBDDs, LTFBDDs and strong LTFBDDs from the corresponding complexity classes for several other variants of BDDs.

12 Improved OBDD and FBDD Lower Bounds for Integer Multiplication via Universal Hashing

Beate Bollig University Dortmund, Dortmund, Germany

(joint work with Philipp Woelfel)

Binary Decision Diagrams (BDDs) are graph representations for Boolean functions. Besides the complexity theoretical viewpoint people have used restricted BDDs in applications where the complexity of fundamental functions is of interest. FBDDs are BDDs where on each path from the source to a sink each variable is tested at most once. OBDDs, one of the most popular representations in applications, have the additional restriction that on all paths the variables are tested according to a given variable ordering.
Bryant (1991) has shown that any OBDD representation for the function $\mathrm{MULT}_{n-1,n}$, which computes the middle bit of the product of two n-bit numbers, requires at least $2^{n/8}$ nodes. This bound would still allow the possibility that one can construct 64-bit multipliers represented by OBDDs containing only 256 nodes, where on the other hand it is widely conjectured that OBDDs computing $\mathrm{MULT}_{n-1,n}$ have a size of at least 2^n. In this talk a stronger lower bound of $\frac{1}{61}2^{n/2}$ is proven by a new technique using a recently found universal

family of hash functions.

Ponzio (1995, 1998) has presented a lower bound of $2^{\Omega(n^{1/2})}$ on the size of FBDDs for $\text{MULT}_{n-1,n}$. Combining results and methods for universal hashing with lower bound techniques for FBDDs the first strongly exponential lower bound of $\Omega(2^{n/4})$ is proven for the middle bit of integer multiplication.

13 The Wonderful World of Partitioned BDDs

Martin Sauerhoff University Dortmund, Dortmund, Germany

Recently, Jain, Mohanram, Moudanos, Wegener, and Lu (2000) proposed a new heuristics for the automated generation of partitioned BDDs (PBDDs) from circuits. The output of their algorithm fulfills two structural constraints often imposed on PBDDs for algorithmic reasons: each part of the PBDD has a corresponding *window function*, and different parts are *disjoint* (which means that the conjunction of their respective functions is the 0-function).

This motivates the theoretical study of the role which these two restrictions play with respect to the size of PBDDs. In the talk, the following two main results have been shown:

- With respect to the case of PBDDs with the same variable order for all parts, requiring disjoint parts and window functions together may lead to a *superpolynomial* blow-up of the size of PBDDs compared to the same scenario without disjointness (more precisely, size $n^{\log n/\log\log n}$ compared to polynomial size in the input length n).
- In the general scenario of arbitrary variable orders for the different parts, requiring disjointness may increase the size of PBDDs *exponentially* compared to the most general model without any restrictions.

With respect to the automated generation of PBDDs, this implies that it may be worthwhile to also consider the generation of PBDDs with *non-disjoint* parts (and, especially, non-disjoint window functions).

14 Heuristics for $\oplus$-OBDD Minimization

Harald Sack University Trier, Trier, Germany

$\oplus$-OBDDs are a true extension of OBDDs, the state-of-the-art data structure in CAD / VLSI for the representation of Boolean functions. By adding nodes that represent an XOR-function computed from the successor nodes, $\oplus$-OBDDs are a more powerful and concise representation than OBDDs are. Despite the fact that they are not a canonical representation for Boolean functions, efficient manipulation based on a probabilistic equivalence test is possible. The size of a $\oplus$-OBDD is determined by the following three factors: (1) frequency of $\oplus$-nodes in the $\oplus$-OBDD (2) their placement inside the data structure and (3) the chosen variable order.
In the talk two heuristics are presented including efficient techniques for restructuring the $\oplus$-OBDD. The first heuristic takes place during synthesis and decides, whether to introduce $\oplus$-nodes for a new gate to be computed, or not. The second heuristic uses dynamic restructuring techniques and is able to find a well suited position for a $\oplus$-node that is already included in the diagram. Experimental results are giving a proof for the efficiency of the two heuristics.

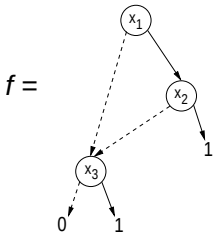
15 Meta-BDDs: A decomposed representation for layered symbolic manipulation of Boolean functions

Gianpiero Cabodi Politecnico di Torino, Torino, Italy

I propose a BDD based representation for Boolean functions which extends conjunctive / disjunctive decompositions.
The model introduced (Meta-BDD) can be considered as a symbolic representation of k-layer automata describing boolean functions. A layer is the set of BDD nodes labeled by a given variable (or a set of variables), and it is represented by an uncompletely specified function, capturing zeros and ones reached by the represented BDD at that level:

$$f =< f_1, \ldots, f_i, \ldots, f_n > .$$

So the i-th component (f_i) partially specifies the f function by represented BDD edges to 0 / 1 from BDD nodes of the i-th layer. E.g.



$$\Longrightarrow \quad < (0,0),\ (x_1x_2,0),\ \overline{x_1x_2}\cdot x_3, \overline{x_1x_2}\cdot\overline{x_3}) >$$

- no edges to terminal at x_1 layer
- 1 edge to terminal at x_2 layer $(x_1 = 1, x_2 = 1)$
- edges to 1 $(x_1, x_2 \neq 11, x_3 = 1)$ and to 0 $(x_1x_2 \neq 11, x_3 = 0)$ at the x_3 layer.

16 SAT-Based Image Computation with Application in Reachability Analysis

Aarti Gupta NEC, Princeton, USA

(joint work with Zijiang Yang, Pranar Ashar (NEC USA, CCRL), Anubhav Gupta (CMU))

Image computation finds wide application in VLSI CAD, such as state reachability analysis in formal verification and synthesis, combinational verification, combinational and sequential test. Existing BDD-based symbolic algorithms for image computation are limited by memory resources in practice, while SAT-based algorithms that can obtain the image by enumerating satisfying assignments to a CNF representation of the Boolean relation are potentially limited by time resources.
We propose new algorithms that combine BDDs and SAT in order to exploit their complementary benefits, and to offer a mechanism for trading off space vs. time. In particular,

1. our integrated algorithm uses BDDs to represent the input and image sets, and a CNF formula to represent the Boolean relation,
2. a fundamental enhancement called BDD Bounding is used whereby the SAT solver uses the BDDs for the input set and the dynamically changing image set to prune the search space of all solutions,
3. BDDs are used to compute all solutions below intermediate points in the SAT decision tree,

4. a fine-grained variable quantification schedule is used for each BDD subproblem, based on the CNF representation of the Boolean relation.

These enhancements coupled with more engineering heuristics lead to an overall algorithm that can potentially handle larger problems. This is supported by our preliminary results on exact reachability analysis of ISCAS benchmark circuits.

17 Can SAT Approximate Free BDDs?

Ted Stanion Synopsys Inc., Beaverton, USA

We start with the assertion that Davis-Putnam (DP)-type SAT algorithms can mimic the behaviour of several other types of decision procedures and data structures for an appropriate ordering of decisions. In particular we are interested in the case of free BDD's (FBDD's). Theoretically, it is possible for a DP algorithm to have similar runtime characteristics as an FBDD algorithm.
To test this hypothesis, we generate a family of randomized circuits with the property that they have good FBDD representations but no godd ROBDD representation, formulate an equivalence problem in CNF and solve these with a state-of-the-art SAT solver. The results we get are in line with what we would expect from an optimal FBDD implementation.

18 A Fast SAT Solver for EDA Applications

Lintao Zhang Princeton University, Princeton, USA

Boolean satisfiabilty (SAT) problems are widely studied in both AI and EDA communities. There are a lot of applications of SAT in EDA problems. However, previous SAT solvers are mainly targeted and tuned for AI problems, thus not suitable for actual problems rised in EDA applications.
Chaff is a SAT solver specialized in solving SAT problems fast and efficiently. Chaff has a novel decision strategy and efficient learning method, more importantly, Chaff employed the new BCP algorithm which is asymptoically faster than most SAT solvers. By careful implementation, we achieve a speed up of

up to 2 orders of magnitude over existing SAT solvers such as GRASP and SATO. Experimental results show that Chaff is currently the best SAT solver for EDA applications.

19 Lower Bounds on Complexity of Probabilistic Branching Programs

Rustam Mubarakzjanov University Trier, Trier, Germany

There are exponential lower bounds on the complexity of bounded error probabilistic OBDDs. For more general model of read-once branching programs, no such bound is known for an arbitrary computation error. There are no lower bounds of complexity of unbounded error probabilistic OBDDs either. We describe in this work some property of functions in terms of communication matrix. For functions satisfying the defined property we present lower bounds on the complexity of more general models than bounded error probabilistic OBDDs.

20 Invitation: www.bdd-portal.org - A Place for Cooperated BDD-Research

Christoph Meinel University Trier, Trier, Germany

The great success in practical applications has made BDDs to a favourite object in EDA and research. There are a lot of new insights in the nature of BDDs, various methods were developed to optimize the use of BDDs and a lot of powerful software tools have been created in recent years. What seems to be missing is one central attraction point, that links all the researchers, conferences and workshops in the field, as well as providing easy access to latest tools and benchmarks for evaluating newly developed algorithms and heuristics.

www.bdd-portal.org aims to be such an one-stop site, that provides all this information and in addition permits easy an fair online evaluation on various BDD-based tools.

21 A New Method of Checking Satisfiability in Propositional Logic

Eugene Goldberg Cadence Labs, Berkeley, USA

We present a new algorithm for checking conjunctive normal form (CNF) satisfiability called *successive clause replacement* (SCR). The SCR-algorithm is based on the fact that if a CNF is satisfiable there must exist a solution which is in the 1-neighborhood of a clause of the CNF. Since explicit 1-neighborhood exploration is very inefficient we introduce a way of implicit 1-neighborhood exploration. A distinction of the SCR-algorithm from the existing deterministic resolution algorithms is in the way it generate resolvents. The generation of new clauses is guided by the objective of 1-neighborhood exploration. This allows one to avoid producing a great number of redundant resolvents which is typical for the previous approaches. One more remarkable feature of the SCR-algorithm is that it can prove CNF satisfiability ”locally” without deducing an empty clause.

22 Checking Equivalence for Partial Implementations

Christoph Scholl Albert-Ludwigs-University, Freiburg, Germany

We consider the problem of checking whether a partial implementation can (still) be extended to a complete design which is equivalent to a given full specification.
Several algorithms trading off accuracy and computational resources are presented: Starting with a simple 0,1,X-based simulation, which allows approximate solutions, but is not able to find all errors in the partial implementation, we consider more and more exact methods finally covering all errors detectable in the partial implementation. The exact algorithm reports no error if and only if the current partial implementation conforms to the specification, i.e. it can be extended to a full implementation which is equivalent to the specification.
We give a series of experimental results demonstrating the effectiveness and feasibility of the methods presented.

23 Combining BDDs and SAT for Equivalence Checking

Andreas Kühlmann Cadence Labs, Berkeley, USA

(joint work with Malay Ganai and Viresh Paruthi)

Many taks in CAD, such as equivalence checking, property checking, logic synthesis, and false path analysis, require efficient Boolean reasoning for problems that are originally derived from circuits. Traditionally, canonical representations, e.g. BDDs, or SAT based search methods are alternatively used to solve a particular class of problems.
In this talk we present a combination of techniques for Boolean reasoning based on BDDs, structural transformations, and a SAT procedure natively working on a shared graph representation of the problem. The described intertwined integration of the individual techniques results in a robust summation of their orthogonal strengths. A large number of experiments demonstrates the overall effectiveness of the approach.

24 New Algorithms for Solving Satisfiability in Formal Verification

Luis Baptista INESC, Lisboa, Portugal

(joint work with Joao Marques-Silva)

Recent work on the Satisfiability Problem (SAT) has provided strong empirical and theoretical evidence of the advantages of applying randomization and restarts in solving satisfiable problem instances.
This talk addresses the interaction between randomization, with restart strategies, and learning, an often crucial technique for proving unsatisfiability.
We use instances of SAT from the hardware verification domain to provide evidence that randomization can indeed be essential in solving real-world satisfiable instances of SAT. More interestingly, our results indicate that randomized restarts and learning may cooperate in proving both satisfiability and unsatisfiability. Finally, we utilize and expand the idea of algorithm portfolio design to propose an alternative approach for solving hard unsatisfiable instances of SAT.

25 Efficient BDD size reduction methods for combinational equivalence verification

Ziyad Hanna Intel, Haifa, Israel

BDDs rapidly blow up when we build monolithic cones in the equivalence verification domain. In this paper we propose a new novel method for building BDDs using cut point method which guarantees no false negative results. The divide-and-conquer method is based on normalizing the cut point functions by eliminating all the logic paths produced by non-convergent fanout variables. This technique proved to be correct and preserves the equivalence relation property at the outputs of the two verified functions.
In addition, this talk proposed a new concept of computing static variable ordering heuristics based on SAT computations. This method is using functional relations among the circuit nodes, and thus computes an efficient variable order while keeping related variables together and important variables at the top of the order.
At the end we presented very promising results that show the efficiency of this method on Intel and ISCAS85 circuits.

26 Dynamic Selection of Branching Rules

Marc Herbstritt Albert-Ludwigs-University, Freiburg, Germany

Current SAT solver (e.g. GRASP) consist of three ”engines”: the deduction engine, the diagnosis engine, and the decision engine. Branching rules are applied in the decision engine to select a variable and an assignment to this variable to guide the search process. In the last years several branching rules were developed, but there is no ”best-of-all” branching rule. Another powerful technique to speed up search is non-chronological backtracking which is part of the diagnosis engine. Due to non-chronological backtracking it can be avoided to search ”senseless” parts of the search tree.
In this talk we present a method to combine information from non-chronological backtracking and the pool of available branching rules. The intuition behind our approach is that the branching rule which caused a conflict and thus led to a backtrack should be ”punished”. Therefore we maintain preference values for all branching rules which model the probability to be selected when a decision assignment is made. To punish a branching rule we count how of-

ten it was used and how often it triggered a conflict. This values are used to diminish the preference value of the branching rule. To select a branching rule during decision assignment we use well known selection methods (roulette-wheel, linear ranking, tournament selection).
Our approach results in a faster and more robust behaviour of the SAT solver.

27 Stochastic Planning Using Decision Diagrams

Alan Hu University of British Columbia, Vancouver, Canada

(joint work with Jesse Hoey, Robert St. Aubin, Craig Boutilier)

The talk described the use of ADDs (a.k.a. MTBDDs) to solve a basic formulation of stochastic planning:

> Given finite set of states S, finite set of actions A, each of which specifies a transition probability matrix $P_a : S \times S \to \mathbf{R}$, an initial state, a reward function $R : S \to \mathbf{R}$, a discount factor $\beta, 0 < \beta < 1$, compute a policy $\pi : S \to A$ that maximizes expected total discounted reward.

Our solution used value iteration, which computes a series of vectors $V : S \to \mathbf{R}$, where $V_i(s)$ is the expected total discounted reward starting in state s and running for i cycles. These vectors can be computed iteratively:

$$\begin{aligned} V_0(s) &= R(s) \\ V_{i+1}(s) &= R(s) + \beta \cdot \max_{a \in A} \left\{ \sum_{t \in S} P_a(s,t) \cdot V_i(t) \right\}. \end{aligned}$$

ADDs can be used to represent the P_a, V_i, and R. To reduce blow-up, the P_a must be represented in a factored form, by assuming independence of various dimensions of the state space. The resulting tool is the fastest and largest capacity tool for this problem.

28 SAT Using ZBDDs

Karem A. Sakallah University of Michigan, Ann Arbor, USA

Zero-suppressed Binary Decision Diagrams (ZBDDs) were proposed several years ago as an alternative to "regular" BDDs for representing sets. ZBDDs are a particularly compact representation for sets of sparse combinations. A recent application of ZBDDs has been for storing and symbolically processing sets of clauses that represent Boolean functions in Conjunctive Normal Form (CNF). In particular, it was demonstrated by Chatalic and Simon (ICTAI 2000) that, when used as the underlying data structure for the Davis-Putnam resolution-based satisfiability (SAT) algorithm, ZBDDs can achieve impressive compression ratios (e.g. $10^8 0$ clauses stored in a 25K-node ZBDD).

In this talk I described some initial ideas on how ZBDDs can be used as the data structure for backtrack search SAT algorithms. An interesting insight is that a judicious combination of search and resolution using ZBDDs may be an effective approach for tackling very large and difficult SAT instances that have defied either method separately.

I also demonstrated VisualSAT, a graphical browser for the SATIRE incremental conflict-based backtrack search program. VisualSAT uses a ZBDD to collect conflict clauses identified during the search; this ZBDD is then used to update a progress bar indicating the percentage of the search space that has been proved to contain no solution.

29 A New Partitioning Scheme for Improving Image Computation

Christian Stangier University Trier, Trier, Germany

Image computation is the core operation for optimization and formal verification of sequential systems like controllers and protocols. State exploration techniques based on OBDDs use a partitioned representation of the transition relation to keep the OBDDs-sizes manageable. This talk presented a new approach building up on previous work using RTL-information resulting in a significant performance increase — 60th in time and memory consumption. The heuristic has been successfully applied to symbolic model checking of real life designs, The approach is also general enough to be applied in a non-BDD environment.

30 On the Complexity of OBDD Synthesis with Optimal Reordering

Ingo Wegener University Dortmund, Dortmund, Germany
(joint work with Beate Bollig)

It was known that the synthesis of two π-OBDDs may lead to a π-OBDD whose size is of the order of the product of the sizes of the given π-OBDDs. However, can this also hold after an optimal reordering of the resulting π-OBDD? An example with such a behavior is presented. The lower bound proof has some new features. Moreover, the rule of thumb that control variables should be tested before data variables is falsified for the multiplexer and ZBDDs and quasi-reduced OBDDs.