

# Iterative Abstraction using SAT-based BMC with Proof Analysis

Aarti Gupta, Malay Ganai, Zijiang Yang, Pranav Ashar

NEC Laboratories America  
Princeton, NJ 08540, U.S.A.  
{agupta, malay, jyang, ashar}@nec-labs.com

## Abstract

*Resolution-based proof analysis techniques have been proposed recently to identify a sufficient set of reasons for unsatisfiability derived by a CNF-based SAT solver. We have adapted these techniques to work with a hybrid SAT solver. We use the proof analysis technique with SAT-based BMC, in order to generate useful abstract models. Our abstraction procedure is used iteratively in a top-down framework, starting from the concrete design, where we apply BMC on increasingly more abstract models. We apply various SAT-based and BDD-based verification methods on these abstract models, in order to obtain proofs of correctness, or to perform deeper searches for counterexamples. We demonstrate the effectiveness of our prototype implementation on several large industry designs.*

## 1. Introduction

Symbolic model checking techniques [1, 2], based on the use of Binary Decision Diagrams (BDDs) [3], offer the potential of exhaustive coverage and the ability to detect subtle bugs. However, these techniques do not scale very well in practice due to the state explosion problem. A recent alternative is Bounded Model Checking (BMC) [4], which focuses on the search for counterexamples of bounded depth. Effectively, the problem is translated to a propositional formula, such that the formula is satisfiable if and only if there exists a counterexample of depth  $k$ . In practice, the depth  $k$  can be increased incrementally to find the shortest counterexample. However, additional reasoning is needed to ensure completeness of the proof of correctness, when no counterexample can be found [4, 5].

The satisfiability check in the BMC method is typically performed by a backend SAT solver. Due to the many advances in SAT solving techniques [6-9], BMC can handle much larger designs than BDD-based methods. A related important development has been the use of resolution-based proof analysis techniques to check the unsatisfiability result of a SAT solver [10, 11]. As part of the check, these techniques also identify a set of clauses from the original problem, called the *unsatisfiable core*, such that the clauses are sufficient for implying the unsatisfiability. Similar SAT-based proof analysis techniques have also been proposed independently in the context of refinement, and abstraction-based verification methods [12, 13]. The existing resolution-based proof analysis techniques have been described for SAT solvers that use a CNF (Conjunctive Normal Form) representation of the Boolean problem. We have adapted these techniques to work with a circuit SAT solver [14], or a hybrid SAT solver [9].

We use the resolution-based proof analysis technique in the SAT solver used for checking BMC problems – we call this *SAT-*

*based BMC with Proof Analysis*. Note that unsatisfiable SAT instances in BMC correspond to the absence of a counterexample of (or up to) a given depth. For each such depth, we identify an unsatisfiable core, and use it to generate an abstract model. In particular, we propose a latch-based abstraction, such that the resulting abstract models are guaranteed to not have a counterexample of (or up to) that depth.

Our overall verification methodology centers around a top-down *iterative abstraction* framework. Starting from the concrete design, we apply SAT-based BMC with Proof Analysis on a *seed model* in each iteration. Our abstraction, based on identification of unsatisfiable cores, is used to choose a seed model for the next iteration. Under certain practical conditions, we allow a refinement step, which can potentially increase the size of the seed model. In each iteration, we also generate existentially abstract models, again based on the unsatisfiable cores. These models are known to be conservative for LTL properties [1, 15]. We use various BDD-based and SAT-based methods for performing unbounded model checking on these models. A proof of correctness on any of these models guarantees correctness on the concrete design, while a counterexample may require a refinement, or going back to a previous iteration in our iterative flow. In practice, we iterate the loop until convergence of the seed model, or until a conclusive result is obtained on some abstract model.

The key contribution of our work is that the overall flow is targeted at reducing the size of the seed models across successive iterations. The potential benefit is that for properties that are false, BMC search for deeper counterexamples is performed on successively smaller models, thereby increasing the likelihood of finding them. For properties that are true, the successive iterations help to reduce the size of the abstract models, thereby increasing the likelihood of completing the proof by unbounded verification methods.

We have implemented these ideas in a prototype verification framework called *DiVer* [16], which includes other BDD-based and SAT-based verification methods. We report on our experience on some large industry designs. For some of these, we have been able to complete proofs of correctness for the first time ever. In most of these, we have been able to perform deeper searches for counterexamples. We have also observed that our iterative abstraction typically gives an order of magnitude reduction in the final model sizes. For many examples, this reduction was crucial in enabling the successful application of the unbounded verification methods.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICCAD '03, November 11-13, 2003, San Jose, California, USA.

Copyright 2003 ACM 1-58113-762-1/03/0011 ...\$5.00.













