A New System and Methodology for Generating Random Modal Formulae

Peter F. Patel-Schneider[1] and Roberto Sebastiani[2]

[1] Bell Labs Research, 600 Mountain Ave., 2A-427 Murray Hill, NJ 07974, USA
pfps@research.bell-labs.com
[2] Dept. of Mathematics, University of Trento, via Sommarive 14, I-38050 Trento, Italy
rseba@science.unitn.it

Abstract. Previous methods for generating random modal formulae (for the multi-modal logic $\mathbf{K}_{(\mathbf{m})}$) result either in flawed test sets or formulae that are too hard for current modal decision procedures and, also, unnatural. We present here a new system and generation methodology which results in unflawed test sets and more-natural formulae that are better suited for current decision procedures.

Most empirical testing of decision procedures for propositional modal logics, usually for the multi-modal logic $\mathbf{K}_{(\mathbf{m})}$, employs randomly generated formulae. This style of testing was initially proposed by Giunchiglia *et al* [3] and later improved by them and also by Hustadt and Schmidt [6, 1]. Other kinds of randomly generated formulae have been proposed by Massacci [7]. Randomly generated formulae have been used with all the recent, highly optimised modal decision procedures, including DLP [8], FaCT [4], KSatC [1], *SAT [2], and TA [6], and have been used in several comparisons of these systems [7].

The basic idea behind the random generator in [3, 6, 1] is to generate a number of clauses. Each clause has the same number of disjuncts. As most testing as been on clauses with three disjuncts, the generated formulae are often called $3\mathrm{CNF}_{\Box_m}$ formulae. Each disjunct in each clause is independently either a possibly negated propositional variable or a modal formula consisting of a possibly negated box over a clause. The embedded clauses are generated in the same way as the top-level clauses, except that at some maximum modal depth all the disjuncts are possibly negated propositional variables. Care is taken to not repeat disjuncts in a clause, nor to have complementary disjuncts in a clause.

Testing consists of setting various parameters, usually the maximum modal depth, d, the probability that a disjunct is a propositional literal, p, and the number of propositional variables, N. Most tests use standard values for the other parameters, namely three disjuncts in a clause, C, and only one kind of modal box, m. Then several values are picked for the number of top-level clauses in a formula, L, and for each value for L many (usually about 100) formulae are generated and tested for satisfiability.

This testing methodology as described above provides a decent test for current modal decision procedures, allowing good comparisons of their behaviour.

However, problems have remained with these methods, most notably that the generated formulae can contain pieces that make the entire formula easy to solve. Because even the fastest current modal decision procedures can only handle formulae with a few propositional variables, the presence of even a small number of top-level clauses with only propositional disjuncts can easily cover all the combinations of the propositional literals and make the entire formula unsatisfiable. (We call such formulae trivially unsatisfiable.)

Previous attempts to eliminate this trivial unsatisfiability have concentrated on eliminating top-level propositional disjuncts by setting $p = 0$ [6]. However, formulae with propositional literals showing up only at the deepest modal depth are extremely hard to solve for modal depths greater than 1. Moreover, the resulting formule are not very natural [5].

We have devised a new generation methodology for generating modal formulae for $\mathbf{K_{(m)}}$ that eliminates or reduces the problems with the previous generation methods. The first new idea of our approach, first suggested in [5], is to eliminate strictly-propositional clauses except at the maximum modal depth by requiring that the number of propositional literals in each clause be less than 1 away from the average value, while still maintaining the overall ratio. For three disjuncts per clause ($C = 3$) and propositional probability one-half ($p = 0.5$) this means that half the clauses have 2 propositional literals and half have 1.

To show the benefit of this change, we present two experimental runs for maximum modal depth $d = 2$ and propositional probability $p = 0.5$. In the old method the time curves are dominated by a "half-dome" shape, whose steep side shows up where the number of trivially unsatisfiable formulae becomes large before the formulae become otherwise easy to solve, as shown in Figure 1 (left). In fact, nearly all the unsatisfiable formulae here are propositionally unsatisfiable. With our new method, as shown in Figure 1 (right), the formulae are much more difficult to solve than the old method, because there is no abrupt drop-off from propositional unsatisfiability, but they are much easier to solve than those generated with $p = 0$. Further, trivially unsatisfiable formulae do not appear at all in the interesting portion of the test sets.

The second new idea of our approach is to allow the number of disjuncts in a clause C to vary in a manner similar to the number of propositional disjuncts. We then determine the number of propositional literals in each clause based on the number of disjuncts in that particular clause.

We tested this generation methodology for several values of C. The results for $C = 2.5$ are given in Figure 2 (left). These formulae are much easier than those generated with $C = 3$, although they are still quite hard and form a reasonable source of testing data. Trivially unsatisfiable formulae appear in large numbers well after all the formulae are all unsatisfiable and relatively easy. By reducing the number of disjuncts we are thus able to create interesting and unflawed test sets for higher modal depths.

The last new idea of our approach is that the generator allows (optionally) direct specification of the probability distribution of the number of propositional atoms in a clause, and allows the distribution to be different for each modal depth

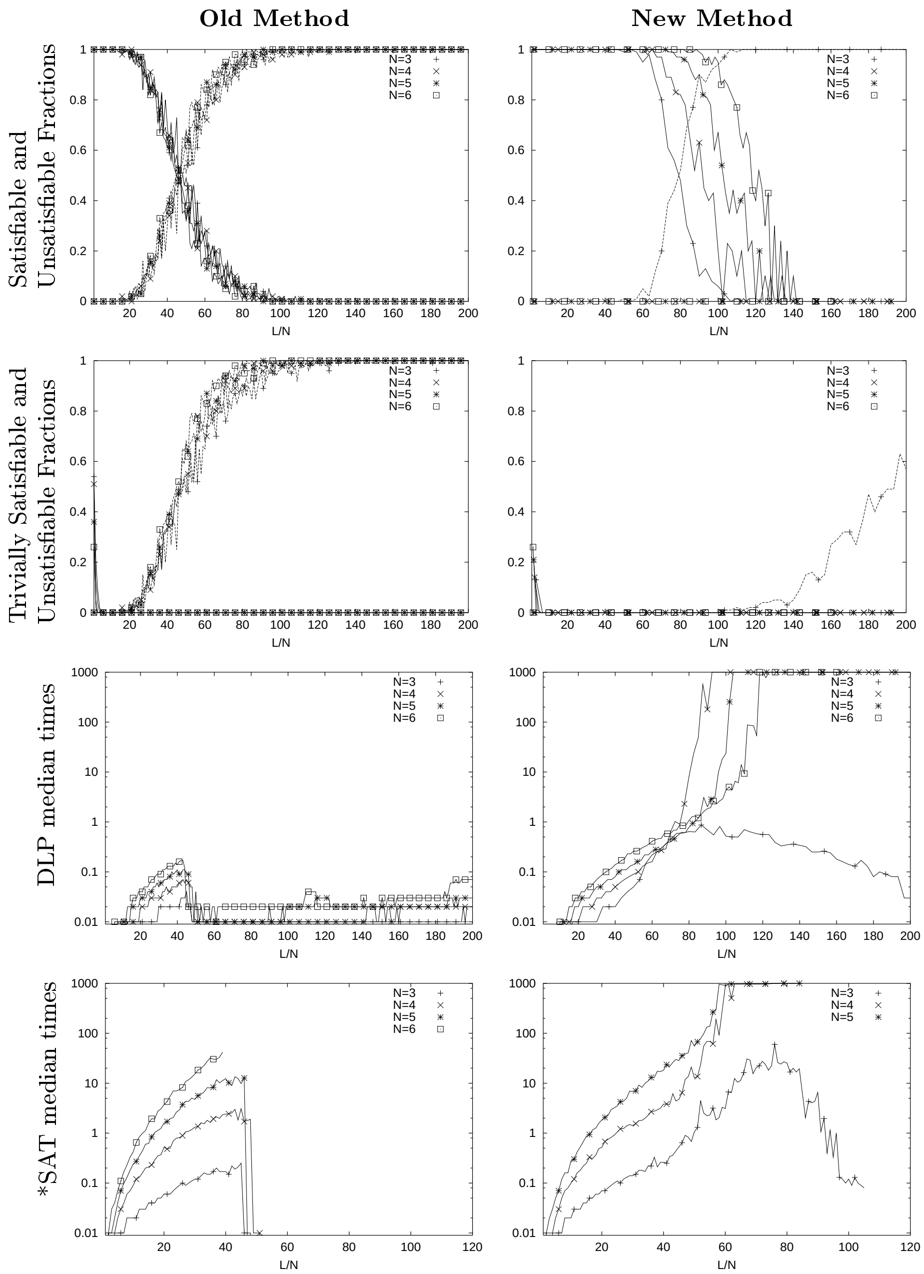

Fig. 1. Results for $m = 1$, $C = 3$, $d = 2$, $N = 3, ..., 6$, and $p = 0.5$.

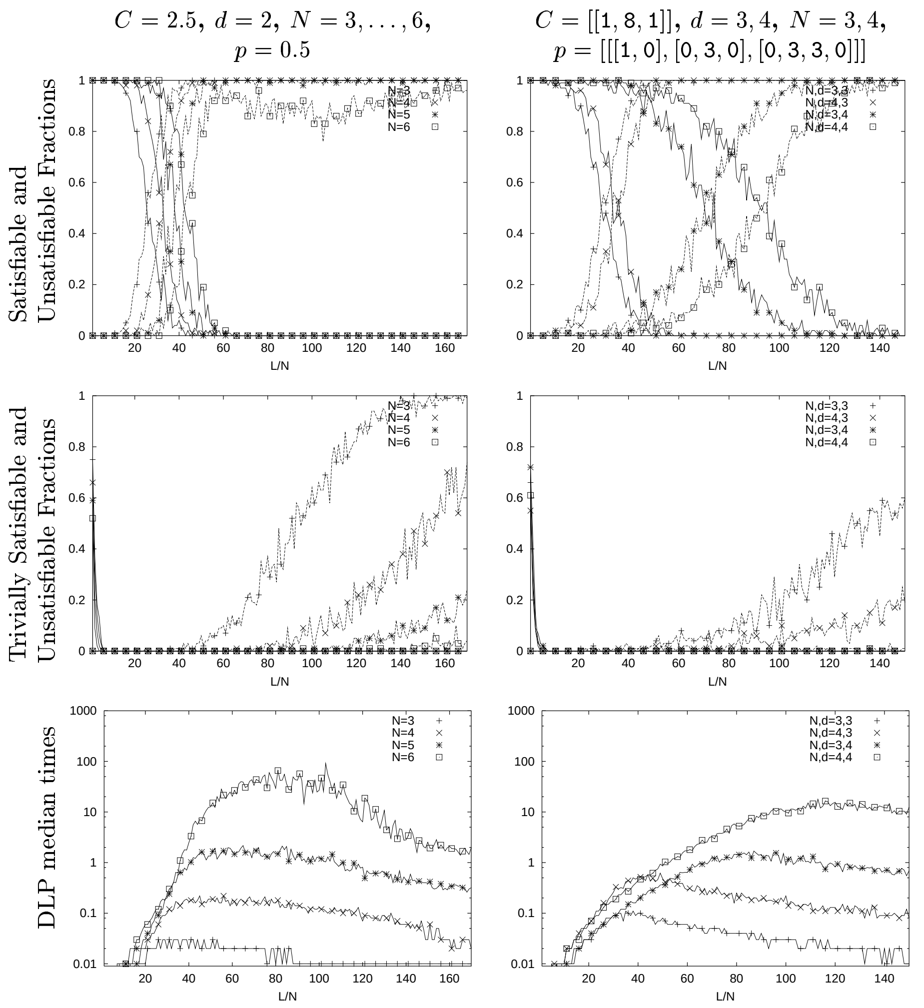

Fig. 2. New method. Left column: $m = 1$, $C = 2.5$, $d = 2$, $N = 3, ..., 6$, and $p = 0.5$. Right column: $m = 1$, $C = [[1, 8, 1]]$, $d = 3, 4$, $N = 3, 4$, $p = [[[1, 0], [0, 3, 0], [0, 3, 3, 0]]]$.

from the top level to $d - 1$. It also allows direct specification of the probability distribution for the number of literals in a clause at each modal depth. Thus, the probability distribution for the number of propositional atoms depends on both the modal depth and the number of literals in the clause.

As an example we present a set of tests with $m = 1$, $C = [[1, 8, 1]]$, $N = 3, 4$, $d = 3, 4$, and $p = [[[1, 0], [0, 3, 0], [0, 3, 3, 0]]]$; C represents the probability distribution for the number of literals in a clause, meaning "1/10, 8/10 and 1/10 clauses are of length 1, 2, 3 respectively"; p represents the probability distribution for the number of propositional atoms in a clause, meaning "1/1 and 0/1 1-literal clauses have 0 and 1 propositional literal respectively; 0/3, 3/3 and 0/3 2-literal clauses have 0, 1 and 2 propositional literals respectively; 0/6, 3/6, 3/6 and 0/6 3-literal clauses have 0, 1, 2 and 3 propositional literals respectively". In this example, both distributions do not vary with the modal depth. This set of tests introduces a small fraction of single-literal clauses that contain a modal literal (except at the greatest modal depth, where they contain, of course, a single propositional literal). The results of tests are given in Figure 2 (right). Notice that we can generate test sets of interesting difficulty even with modal depth 4.

Our generator also takes extreme care to not disturb the probablility distributions of the generated cluses as the result of rejecting clauses that have repeated or contradictory disjuncts. It first selects a "shape" for the formula and only when this is determined does it select the rest of the formula. This care is necessary because otherwise larger disjuncts would be preferentially selected because they have less chance of a repetition or contradiction.

Our generator is available for download at

`http://www-db.research.bell-labs.com/user/pfps/dlp`.

It can output formulae in various syntaxes and can be used to capture statistics from various modal decision procedures.

References

1. E. Giunchiglia, F. Giunchiglia, R. Sebastiani, and A. Tacchella. SAT vs. Translation based decision procedures for modal logics: a comparative evaluation. *Journal of Applied Non-Classical Logics*, 10(2):145–172, 2000.
2. E. Giunchiglia, F. Giunchiglia, and A. Tacchella. SAT Based Decision Procedures for Classical Modal Logics. To appear in *Journal of Automated Reasoning.*, 2001.
3. F. Giunchiglia and R. Sebastiani. Building decision procedures for modal logics from propositional decision procedures - the case study of modal K. In *Proc. CADE13*, number 1104 in Lecture Notes in Artificial Intelligence, pages 583–597, Berlin, August 1996. Springer-Verlag.
4. I. Horrocks. Using an expressive description logic: FaCT or fiction? In *Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 636–647, 1998.
5. Ian Horrocks, Peter F. Patel-Schneider, and Roberto Sebastiani. An analysis of empirical testing for modal decision procedures. *Logic Journal of the IGPL*, 8(3):293–323, 2000.
6. Ullrich Hustadt and Renate A. Schmidt. An empirical analysis of modal theorem provers. *Journal of Applied Non-Classical Logics*, 9(4):479–522, 1999.
7. F. Massacci. Design and Results of Tableaux-99 Non-Classical (Modal) System Competition. In *Automated Reasoning with Analytic Tableaux and Related Methods: International Conference (Tableaux'99)*, number 1617 in Lecture Notes in Artificial Intelligence, pages 14–18, Berlin, 1999. Springer-Verlag.
8. Peter F. Patel-Schneider. DLP system description. In E. Franconi, G. De Giacomo, R. M. MacGregor, W. Nutt, C. A. Welty, and F. Sebastiani, editors, *Collected Papers from the International Description Logics Workshop (DL'98)*, pages 87–89, 1998. Available as CEUR-WS/Vol-11 from http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS.