A new method for testing decision procedures in modal and terminological logics

Fausto Giunchiglia
IRST, 38050 Povo (TN), Italy.
DISA, via Inama 5, 38100 Trento
ph: ++39.461.314436
fausto@irst.itc.it

Marco Roveri and **Roberto Sebastiani**
DIST, v. Causa 13, 16146 Genoa, Italy.
ph: ++39.10.3532811
{marco,rseba}@mrg.dist.unige.it

Abstract

We present a new methodology for testing decision procedures for modal and terminological logics which extends the *fixed-clause-length* test model, commonly used for propositional satisfiability testing. The new method is easy to implement and to use, and it allows for a statistical control of some important features, e.g., hardness and satisfiability rate, of the formulas generated.

1 Introduction

In the area of automated reasoning in modal and terminological logics there seems to be very little bibliography on both problem sets and test methodologies. Most authors do not present empirical tests. Some authors test their systems on groups of single formulas, mostly taken from textbooks (see, e.g., [Catach, 1991; Demri, 1995]). Few others use formulas which are derived from or simulate realistic problems (see, e.g., [Baader *et al.*, 1994]). As far as we know, there has been little study on classes of test formulas and on their intrinsic properties (e.g., hardness, satisfiability likelyhood). Modal S4 is a noteworthy exception, as it is possible to exploit the Goedel-Tarski translation from intuitionistic logic into S4. A second noteworthy exception can be found in [Halpern and Moses, 1992]. In this paper the authors describe a class of modal formulas which are intrinsically hard, as they are provably satisfiable only in exponential-size Kripke structures. An empirical test based on these formulas is presented in [Giunchiglia and Sebastiani, 1996b].

In propositional and first-order theorem proving, instead, there is a wide bibliography on both problem sets (see, e.g., [Pellettier, 1986; Suttner and Sutcliffe, 1995]) and test-generating methods (see, e.g., [Buro and Buning, 1992; Mitchell *et al.*, 1992]). In this paper we describe a new methodology for testing decision procedures for modal and terminological logics, in particular for K, K(m) and $\mathcal{ALC}$, which generalizes a test model commonly used in propositional satisfiability (SAT from now on). The new method is easy to implement and to use, and it allows for a statistical control of some important features features, e.g, hardness and satisfiability rate, of the formulas generated.

The paper is organized as follows. In Section 2 we briefly describe the test method used in SAT. In Section 3 we describe our new test methodology for K, K(m)/$\mathcal{ALC}$. In Section 4 we present an example of empirical test which highlights the features and effectiveness of our method. Finally, in Section 5 we give hints about how our proposed methodology can be extended to other logics.

2 The fixed-clause-length SAT method

We start from the *fixed clause-length* SAT test model (see, e.g., [Mitchell *et al.*, 1992; Buro and Buning, 1992]) with clause length $K = 3$ (*3-clause-length* from now on), briefly described below. Given a number N of propositional variables, for increasing values of the clause number L, sufficiently many (100, 500, 1000,...) random 3CNF wffs are generated and given in input to the procedure under test. After the computation, a statistical analysis of the results is performed. The resulting statistical values, like satisfiability percentages, mean/median CPU times or mean/median size of the search space, are plotted against the L/N ratio. This process can be repeated for different numbers of propositional variables. Random 3CNF wffs are generated as follows: "for given L and N, an instance of a random 3SAT is produced by randomly generating L clauses of length 3. Each clause is produced by randomly choosing a set of 3 propositional variables from a set of N, and negating each with probability 0.5." (Quote from [Mitchell *et al.*, 1992]).

The success of this method is due, in our opinion, to three main features. First, 3CNF wffs represent all propositional formulae. In fact, it is well-known (see, e.g., [Garey and Johnson, 1979]) there is a satisfiability-preserving way of converting all propositional wffs into 3CNF. Second, 3CNF wffs can be randomly generated according to only 2 parameters:

(i) the number of clauses L;
(ii) the number of propositional variables N.

Finally, the parameters L and N allow for a coarse "tuning" of the probability of satisfiability and of the hardness of random 3CNF wffs. In fact, L [N] monotonically

```
function rand_wff(d,m,L,N,p)
    for i := 1 to L do
        $C_i$ := rand_clause(d,m,N,p);
    return $\bigwedge_{i=1}^{L} C_i$;

function rand_clause(d,m,N,p)
    for j := 1 to 3 do
        $l_j$ := rand_lit(d,m,N,p);
    return $\bigvee_{j=1}^{3} l_j$;

function rand_lit(d,m,N,p)
    $\varphi$ := rand_atom(d,m,N,p);
    if flip_coin(0.5)
    then return $\varphi$;
    else return $\neg\varphi$;

function rand_atom(d,m,N,p)
    if (d=0 or flip_coin(p))
    then return rand_propositional_atom(N);
    else $\Box_r$ := rand_box(m);
         C := rand_clause(d-1,m,N,p);
         return $\Box_r C$ ;
```

Figure 1: The algorithm of the random generator.

increases [decreases] the level of constraintness. Thus, varying the L/N ratio, we pass from a situation where wffs are underconstrained (and thus mostly satisfiable) to one where wffs are overconstrained (and thus mostly unsatisfiable). As a consequence, the plot of the satisfiability percentages draws a transition from 100% satisfiability to 100% unsatisfiability [Mitchell *et al.*, 1992], with the 50% crossover point always located around the fixed value $L/N \approx 4.3$. Moreover, the mean and median CPU time plots reveal a easy-hard-easy pattern always centered in the "100% satisfiable-100% unsatisfiable" transition zone. Increasing N the plots become sharper. This phenomenon, known as "phase transition" for some analogies with thermodynamics, has been widely investigated both empirically (see, e.g., [Kirkpatrick and Selman, 1994]) and theoretically (see, e.g., [Williams and Hogg, 1994]). Therefore, suitable choices of L and N allow us to generate very hard wffs with near 50% satisfiability probability.

3 The 3CNF$_{K(m)}$ test method

Consider the modal logic K(m), as it is described, e.g., in [Halpern and Moses, 1992] [1]. In order to extend the fixed-clause-length test methodology,, we first give a suitable definition of CNF wffs for K(m), CNF$_{K(m)}$ from now on.

- a CNF$_{K(m)}$ wff is a conjunction of CNF$_{K(m)}$ clauses;
- a CNF$_{K(m)}$ clause is a disjunction of CNF$_{K(m)}$ literals, i.e., CNF$_{K(m)}$ atoms or their negations;
- a CNF$_{K(m)}$ atom is either a propositional atom or a wff of the form $\Box_r C$, where $\Box_r \in \{\Box_1, \ldots, \Box_m\}$ and C is a CNF$_{K(m)}$ clause.

Notice that conjunctions appear only at the top level of a CNF$_{K(m)}$ wff. Without loss of generality, we can fix the number of literals per clause. A CNF$_{K(m)}$ wff is called a 3CNF$_{K(m)}$ wff iff the number of literals per clause is fixed to 3. 3CNF$_{K(m)}$ wffs can be randomly generated according to only 5 parameters:

(i) the modal depth d;
(ii) the number of distinct boxes m;
(iii) the number of top-level clauses L;
(iv) the number of propositional variables N;
(v) the probability p with which any random 3CNF$_{K(m)}$ atom is propositional.

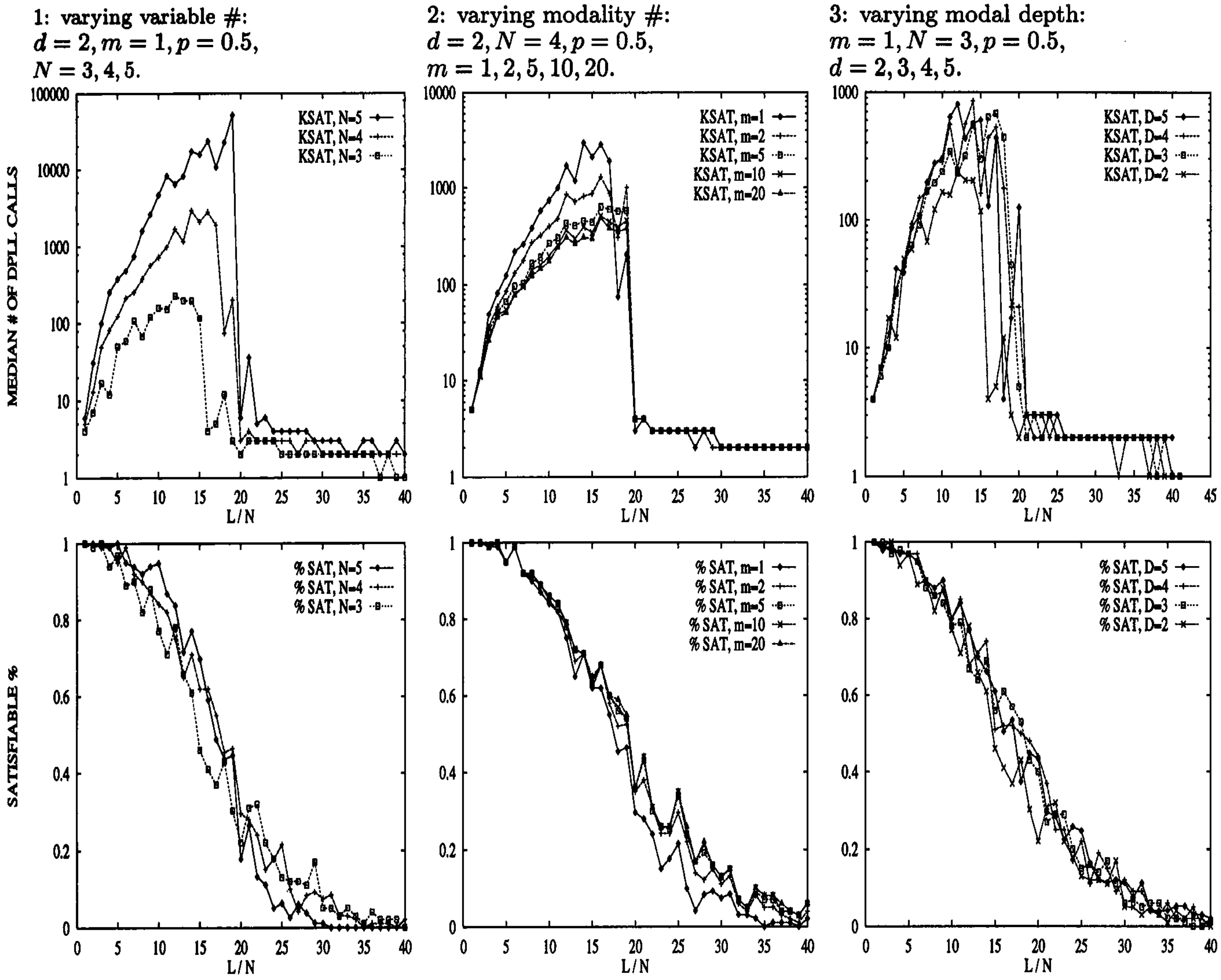
The algorithm of the random generator is presented in Figure 1. A 3CNF$_{K(m)}$ modal wff of depth d is produced by randomly generating L modal 3CNF$_{K(m)}$ clauses of depth d. A 3CNF$_{K(m)}$ modal clause of depth d is produced by randomly generating three 3CNF$_{K(m)}$ modal atoms of depth d, and negating each of them with probability 0.5. (The function *flip_coin(p)* returns *True* with probability p, *False* otherwise.) A 3CNF$_{K(m)}$ modal atom of depth d is produced in the following way. If either $d = 0$ or *flip_coin(p)* returns *True*, then the function *rand_propositional_atom(N)* picks randomly an atom $A_k \in \{A_1, \ldots, A_N\}$, which is returned. (Intuitively, the parameter p establishes the mean ratio of the propositional atoms at every level of the wff tree.) Otherwise a box $\Box_r$, picked randomly from $\{\Box_1, \ldots, \Box_m\}$ by the function *rand_box(m)*, followed by a 3CNF$_{K(m)}$ modal clause of depth $d - 1$, is returned.

The modal 3-clause-length test method is then defined in analogy with the propositional case. For fixed N, d, m and p, for increasing values of L, a certain number (100, 500, 1000...) of random 3CNF$_{K(m)}$ wffs is generated, internally sorted, and then given in input to the procedure under test. Satisfiability percentages, mean/median CPU times or mean/median search space sizes are plotted against the L/N ratio [2].

The methodology proposed preserves the three main features of the SAT method. First, CNF$_{K(m)}$ [3CNF$_{K(m)}$] wffs represent all K(m) wffs, as there is

[1] As it is well known, the terminological logic $\mathcal{ALC}$ is a notational variant of the modal logic K(m), that is, K with m distinct modalities [Schild, 1991]. In this paper we always refer to K(m) rather than to $\mathcal{ALC}$. In particular, we speak of "wffs" rather than "concepts", "modalities" rather than "roles", "satisfiability" rather than "coherence", and so on.

[2] As a test rule we often introduce a timeout of 1000s on each sample wff. If the decision procedure under test exceeds the timeout, a failure value is returned and the CPU time value is conventionally set to 1000s. The satisfiability percentage is then evaluated on the number of samples which terminate within the timeout. Under this test strategy, the satisfiability data produced should be considered only as a coarse indication.



1: varying variable #: $d = 2, m = 1, p = 0.5$, $N = 3, 4, 5$.

2: varying modality #: $d = 2, N = 4, p = 0.5$, $m = 1, 2, 5, 10, 20$.

3: varying modal depth: $m = 1, N = 3, p = 0.5$, $d = 2, 3, 4, 5$.

Figure 2: The results of the three experiments.

a K(m)-satisfiability-preserving way of converting any K(m) wff into $\text{CNF}_{K(m)}$ $[\text{3CNF}_{K(m)}]^3$. Second, the usage of the $\text{3CNF}_{K(m)}$ form minimizes the number of parameters to handle. In fact, we need only one "and-branch" parameter L and no "or-branch" parameters, independently of the modal depth d. Finally, the parameters L and N allow for a coarse "tuning" of both the satisfiability probability and the hardness of random $\text{3CNF}_{K(m)}$ modal wffs. In fact, L $[N]$ monotonically increases [decreases] the level of constraintness so that, again, varying the L/N ratio, the plot of the satisfiability percentages draws a transition from 100% satisfiability to 100% unsatisfiability. In [Giunchiglia and Sebastiani, 1996a; 1996b] we have observed easy-hard-easy patterns in all the CPU-times/search-space-size mean/median values plots, which were centered in the "100% satisfiable-100% unsatisfiable" transition zone (see also Section 4.2). This showed the existence of a phase transition phenomenon also for K(m). Therefore, it is possible to generate very hard $\text{3CNF}_{K(m)}$ problems with near 0.5 satisfiability probability.

4 An example of $\text{3CNF}_{K(m)}$ testing

As an example of the effectiveness of the $\text{3CNF}_{K(m)}$ test method, we present here the results of an empirical test taken from [Giunchiglia and Sebastiani, 1996b] [4] and use them to analyze the behaviour of the formulas generated

[3] The conversion works recursively on the depth of the wff, from the leaves to the root, each time applying to sub-wffs the propositional CNF [3CNF] conversion and the transformation $\Box_r \bigwedge_j \bigvee_i \varphi_{ij} \Longrightarrow \bigwedge_j \Box_r \bigvee_i \varphi_{ij}$.

[4] The test code and all the results presented in this paper are available via anonymous FTP at `ftp.mrg.dist.unige.it` in `pub/mrg-systems/ksat/ksat1/`.

by the $3\mathrm{CNF}_{K(m)}$ method. In particular, we focus on the hardness and satisfiability rate of such formulas, and on how the latter values are affected by the parameters of the generators.

Figure 2 describes the results of three experiments, for a total amount of 48,000 randomly generated wffs, obtained by running KSAT [Giunchiglia and Sebastiani, 1996a; 1996b], a "SAT-based" decision procedure for K(m)/$\mathcal{ALC}$. All curves represent 100 sample wffs per point. The range $1 \ldots 40$ for the X-axis parameter L/N has been chosen empirically to cover coarsely the "100% satisfiable - 100% unsatisfiable" transition. In each experiment we investigate the effects of varying one parameter while fixing the others. In Experiment 1 (left column) we fix $d = 2$, $m = 1$, $p = 0.5$ and plot different curves for increasing numbers of variables $N = 3,4,5$. In Experiment 2 (center column) we fix $d = 2$, $N = 4$, $p = 0.5$ and plot different curves for increasing number of distinct modalities $m = 1,2,5,10,20$. In Experiment 3 (right column) we fix $m = 1$, $N = 3$, $p = 0.5$ and plot different curves for increasing modal depths $d = 2,3,4,5$. For each experiment, we present two distinct sets of curves, each corresponding to a distinct row. In the first (top row) we plot the median size of the space searched by KSAT. This gives an indication of the hardness of the formulae. In the second (bottom row) we plot the percentage of satisfiable wffs evaluated by KSAT. This gives an indication of the satisfiability likelyhood of the formulae.

Despite the big noise, due to the small samples/point rate (100), the results indicated in Figure 2 provide interesting indications. We report below (Section 4.1) a first pass, experiment by experiment, analysis of the results. This gives us an idea of how efficiency and satisfiability are affected by each single parameter. In Section 4.2 we report a global analysis of the results we have.

4.1 A testwise analysis

The results of the first experiment (left column) show that increasing N (and L accordingly) causes a relevant increase in the hardness of the test formulae. This should not be a surprise, as in K/K(m), adding few variables may cause an exponential increase of the search space. Each variable may in fact assume distinct truth values inside distinct states/possible worlds, that is, each variable must be considered with an "implicit multiplicity" equal to the number of states of a potential Kripke model for the input formula φ, which is $O(|\varphi|^d)$ [Halpern, 1995].

The results of the second experiment (center column) present two interesting aspects. First, the complexity of the search monotonically decreases with the increase of the number m of modalities. At first this may sound like a surprise, but it should not be so. In fact, the search tree is "divided and conquered" into m non-interfering search trees, each restricted to a single $\Box_r$. Therefore, the bigger is m, the more partitioned is the search space, and the easier is the problem to solve. Second, a careful look reveals that the satisfiability percentage increases with m. Again, there is no mutual dependency between subwffs occurring under the scope of different $\Box_r$'s. Therefore the bigger is m, the less constrained — and thus the more likely satisfiable — are the randomly-generated wffs.

The results of the third experiment (right column) provide evidence of the fact that complexity increases with the modal depth d. This is rather intuitive: the higher is d, the deeper are the Kripke models to be searched, and the higher is the complexity of the search.

4.2 A global analysis

We highlight now some experimental evidence which is shared by all three experiments. First, consider the satisfiability plots (bottom row). Despite the noise and the approximations due to timeouts, it is easy to notice that the 50% satisfiability point is centered around $L/N = 15 \sim 20$ in all the experiments. Moreover, similarly to [Mitchell *et al.*, 1992], a careful look to the first experiment reveals that the satisfiability transition becomes steeper when increasing N (e.g., compare the $N = 3$ and $N = 5$ plots). Second, consider the search space plots (top row). In all the experiments considered, the curves draw an easy-hard-easy pattern, whose peak is centered around the satisfiability transition — although, unlike, e.g., [Mitchell *et al.*, 1992], they seem to anticipate a little the 50% crossover point. Moreover the locations of the easy-hard-easy zones do not seem to vary significantly, neither with the number of variables N (left column), nor with the number of modalities m (center column), nor with the depth d (right column).

The results of this test suggest some considerations. First, we may conjecture (to be verified!) the existence for K(m)/$\mathcal{ALC}$ of a phase transition phenomenon, similar to that already known for SAT and other NP-complete problems. As far as we know, this is the first time this phenomenon is revealed with modal formulas. Second, the $3\mathrm{CNF}_{K(m)}$ test method allows us to generate formulae "as hard as we like them", with near 50% satisfiability probability. In fact, once we have fixed m, d and p, we can "tune" the hardness by choosing N and then the satisfiability rate by choosing L accordingly. Finally, the test shows that the size of the search space *decreases* after a certain size of the formula under test. This may surprise whoever is used to tableau-based systems, where the space searched always grows with the size of formulae, even with fixed d and N. As described in more detail in [Giunchiglia and Sebastiani, 1996b], this is due to the fact that, unlike tableaux, SAT procedures prune a branch as soon as it violates a constraint of the formula; the more constrained the formula is, the more likely an uncomplete branch violates a constraint, the higher the search tree is pruned. The $3\mathrm{CNF}_{K(m)}$ method has enabled us not only to evidence a big quantitative performance gap between two procedures, but also to detect an intrinsic qualitative wickedness of one of them.

5 Beyond K(m)/$\mathcal{ALC}$

Although we do not yet have any empirical evidence backing our claim, we believe that our method will work for a wide class of logics, and, more specifically with most (all?) normal modal logics. First, independently on the logic considered, L monotonically increases the constraintness of the formulae, decreasing their satisfiability probability, causing thus a satisfiability transition. Different logics can only affect the location and the steepness of the transition itself: the higher the logic in the hierarchy of normal modal logics, the more likely unsatisfiable the formula, the earlier the transition. Second, the last consideration in Section 4.2 (see also [Williams and Hogg, 1994]) suggests that easy-hard-easy patterns in the size of the search space are also a direct consequence of the monotonic increase of constraintness, independently on the kind of logic considered. Again, different logics can only affect the location and the steepness of the peak itself. We have performed an analogous (unpublished) set of experiments for modal S5, in which we have obtained results qualitatively similar — but a little shifted to the left, as expected — to those of [Giunchiglia and Sebastiani, 1996a]. We expect easy-hard-easy patters centered in the satisfiability transition zone will exist for most (all?) modal normal logics.

There is a natural way to extend our method to logics which extend the syntax of K(m)/$\mathcal{ALC}$ with other prefixed constructs C_i's — like, e.g., number restriction, inverse, composition, ... For every new construct C_i we need a function *rand_*$\mathcal{C}_i(v_1^{C_i}, v_2^{C_i}, \ldots)$, which we use to randomly generate C_i according to some specific parameters $v_1^{C_i}, v_2^{C_i}, \ldots$. The random generator *rand_wff* is thus extended by simply substituting *rand_box(m)* with a function which either (i) selects, with probability P_{C_i}, one C_i, and returns *rand_*$\mathcal{C}_i(v_1^{C_i}, v_2^{C_i}, \ldots)$, or (ii) returns *rand_box(m)*, with probability $1 - \sum_i P_{C_i}$. The parameters $P_{C_i}, v_1^{C_i}, v_2^{C_i}, \ldots$ are added to the parameter list of the generator, for every construct C_i. To perform a test, all parameters except L and one parameter $v \in \{d, m, p, N, P_{C_1}, v_1^{C_1}, v_2^{C_1}, \ldots\}$ are set to fixed values. Then the experiment can be run similarly to the ones in Figure 2 (where $v \in \{N, m, d\}$ and $P_{C_i} = 0$, for all i).

References

[Baader *et al.*, 1994] F. Baader, E. Franconi, B. Hollunder, B. Nebel, and H.J. Profitlich. An empirical analysis of optimization techniques for terminological representation systems or: Making KRIS get a move on. *Applied Artificial Intelligence. Special Issue on Knowledge Base Management*, 4:109–132, 1994.

[Buro and Buning, 1992] M. Buro and H. Buning. Report on a SAT competition. Technical Report 110, University of Paderborn, Germany, November 1992.

[Catach, 1991] L. Catach. TABLEAUX: a general theorem prover for modal logics. *Journal of Automated Reasoning*, 7, 1991.

[Demri, 1995] S. Demri. Uniform and Nonuniform strategies for tableaux calculi for modal logics. *Journal of Applied Nonclassical Logics*, 5(1):77–98, 1995.

[Garey and Johnson, 1979] M. R. Garey and D. S. Johnson. *Computers and Intractability.* Freeman and Company, New York, 1979.

[Giunchiglia and Sebastiani, 1996a] F. Giunchiglia and R. Sebastiani. Building decision procedures for modal logics from propositional decision procedures - the case study of modal K. In *Proc. of the 13th Conference on Automated Deduction*, August 1996.

[Giunchiglia and Sebastiani, 1996b] F. Giunchiglia and R. Sebastiani. A SAT-based decision procedure for ALC. In *Proc. of the 5th International Conference on Principles of Knowledge Representation and Reasoning - KR'96*, Cambridge, MA, USA, November 1996.

[Halpern and Moses, 1992] J.Y. Halpern and Y. Moses. A guide to the completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(3):319–379, 1992.

[Halpern, 1995] J.Y. Halpern. The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic. *Artificial Intelligence*, 75(3):361–372, 1995.

[Kirkpatrick and Selman, 1994] S. Kirkpatrick and B. Selman. Critical behaviour in the satisfiability of random boolean expressions. *Science*, 264:1297–1301, 1994.

[Mitchell *et al.*, 1992] D. Mitchell, B. Selman, and H. Levesque. Hard and Easy Distributions of SAT Problems. In *Proc. of the 10th National Conference on Artificial Intelligence*, pages 459–465, 1992.

[Pellettier, 1986] F. J. Pellettier. Seventy-Five Problems for Testing Authomatic Theorem Provers. *Journal of Automated Reasoning*, 2:191–216, 1986.

[Schild, 1991] K. D. Schild. A correspondence theory for terminological logics: preliminary report. In *Proc. of the 12th International Joint Conference on Artificial Intelligence*, pages 466–471, Sydney, Australia, 1991.

[Suttner and Sutcliffe, 1995] C. B. Suttner and G. Sutcliffe. The TPTP Problem Library. Technical Report TR 95/6, James Cook University, Australia, August 1995.

[Williams and Hogg, 1994] C. P. Williams and T. Hogg. Exploiting the deep structure of constraint problems. *Artificial Intelligence*, 70:73–117, 1994.