Discovering the Impact of Group Structure on 3-SAT

James M. McQuillan
Department of Computer Science
Western Illinois University
Macomb, IL, 61455, USA
email: JM-McQuillan@wiu.edu

Craig A. Damon
Department of Computer Science
University of Vermont
Burlington, VT, 05405, USA
email: cdamon@cs.uvm.edu

Abstract — Boolean SAT problems are widespread in current demanding applications, most notably model checking for both hardware and software. It has long been known that the clause-to-variable ratio describes the likely behavior of randomly generated boolean formulae. This paper empirically investigates how the grouping of variables in these formulae further affects their satisfiability.

keywords: normalized boolean formulae structure, 3-SAT, satisfiability.

1.0 Introduction

Boolean satisfiability problems (SAT) are a core problem in computer science. SAT forms the core definition for NP complete [4]. Perhaps more practically, the ability to find a solution to a SAT problem is the limiting factor for many computationally challenging applications.

For more than a decade, the demands of model checking [3] have driven researchers to pursue more efficient mechanisms to solve SAT problems. In model checking (and other related problems), important characteristics of a hardware or software design are encoded in a boolean formula. This formula is extended to incorporate the beliefs about the expected behavior of the system being designed. Any satisfying assignments to the final boolean formula represent contradictions to the designers' expectations.

These formulae can easily contain thousands of distinct boolean variables. While most formulae can be reasonably solved by some approach, no single current approach can efficiently solve all such formulae appearing in practice. Choosing the best approach is an inexact science at best.

Determining a priori the behavior of a boolean formula with regards to these approaches can assist in choosing the appropriate choice. Little is known about the characteristics of formulae that make binary decision diagrams (BDD) [2] efficient or inefficient. Local search and other hill climbing approaches [15] are extremely efficient, but work only with formulae that are satisfiable. Determining whether a formula is likely to be satisfied can encourage or discourage the choice of such an approach.

Empirical investigations of the behavior of SAT problems will hopefully lead to a better understanding of when and why particular classes of SAT problems are likely to be satisfied or not. These investigations may also lead to a better understanding of when and why particular approaches work well or poorly, hopefully leading to new, improved SAT solvers.

Many approaches require the SAT formula to be in a standard format. Most commonly, the formula is expected to be in conjunctive-normal-form (CNF), meaning that each clause ors its terms and all clauses must be satisfied. Each term in a clause is either a variable or a negated variable. Formulae with exactly three terms in every clause are called 3-CNF boolean formulae.

Franco et. al. [11] discovered the initial structural property of SAT formulae. For 3-CNF boolean formulae, the ratio of the number of clauses to the number of distinct variables in a randomly chosen formula is highly correlated to the likelihood that the formula can be satisfied. A randomly chosen boolean formula with a clause-to-variable ratio of about 4.25 is equally likely to be satisfiable or not satisfiable. The likelihood of satisfaction monotonically rises as this ratio falls, with a formula with a ratio of 3.52 ([12]) almost certainly being satisfiable. Similarly, the higher the ratio, the less likely is the formula to be satisfiable; a ratio of 4.506 ([9]) indicating statistical near certainty. These bounds are constantly being improved, reducing the range of uncertainty.

All these determinations assume a 3-CNF formula, as do many of the approaches to solving boolean formulae. Unfortunately, many practical problems do not naturally yield formulae in 3-CNF format. Standard approaches exist to convert arbitrary boolean formulae to 3-CNF, increasing the number of clauses and/or variables in the process. The normalizing algorithm in [5] introduces one additional variable for each binary operator in the original formula. The resultant 3-CNF formula contains 4 clauses for each such operator (plus an additional 4 clauses). The four clauses corresponding to a particular operator contain the same three variables: the newly introduced variable and the two variables appearing as operands of the original operator.

This transformation has obvious impact on the structure of the formula. In terms of previously discovered properties, all 3-CNF formula generated by this algorithm must have a clause-to-variable ratio below 4.0. This observation can be seen in the following relationship:

$$\frac{C}{V} = \frac{4 * op + 4}{op + v} < 4 \qquad \text{if } v > 1,$$

where C is the number of clauses in the resultant 3-CNF formula, V the number variables in the 3-CNF formula, op the number of operators in the original formula and v is the number of original variables. Going the other way, knowing the ratio of operators to variables in the original formula will tell us the clause-to-variable ratio in the normalized 3-CNF formula:

$$\begin{aligned} \tfrac{C}{V} &\approx \tfrac{4*op}{op+v}, \qquad \text{if } v \text{ is large} \\ &= \tfrac{4}{1+\frac{1}{(op/v)}}. \end{aligned}$$

This operator-to-variable ratio (op/v) is essentially the average number of times each variable appears in the original formula.

Table I shows the relationship between the operator-variable ratio in an arbitrary formulae and the clause-to-variable ratio in the corresponding normalized formula.

TABLE I
CLAUSE-TO-VARIABLE RATIO FOR THE 3-CNF EQUIVALENT OF A BOOLEAN EXPRESSION WITH A GIVEN OPERATOR-TO-VARIABLE RATIO.

op/v	3	4	5	6	7	8
C/V	3.0	3.2	3.33	3.43	3.5	3.56

op/v	9	10	12	15	20
C/V	3.6	3.64	3.69	3.75	3.81

One could assume that essentially all original formulae with the average variable appearing less than 7 times should be satisfiable. Furthermore, the vast majority of all formula are expected to be satisfiable. Clearly, this assumption is invalid. Since the mid 1980's, many different structures have been investigated with respect to satisfiability. See [10] for some examples. HornSAT ([8]) is just one such example.

Table II shows how the satisfiability of normalized formulae diverges from the satisfiability of random formulae with the same clause-to-variable ratio. The first row shows the measured probability that a random 3-CNF formula with 40 variables and the indicated clause-to-variable ratio will be satisfiable. The second row shows the measured probability that an equivalent random normalized formula with the same clause-to-variable ratio can be satisfied. All the normalized formulae had 40 variables.

TABLE II
PERCENT OF RANDOM AND NORMALIZED RANDOM FORMULAE THAT ARE SATISFIABLE AT VARIOUS C/V RATIOS.

	3.0	3.2	3.3	3.5	3.7
random	100	100	99.9	99.5	97.3
normalized	92.8	92.5	91.6	88.2	81.4

The normalized formulae were 10 to 100 times more likely to be unsatisfiable than the random 3-CNF formula with the same clause-to-variable ratio. Something else must be affecting the satisfiability of the normalized formulae.

The normalization has introduced other structure into the formula that masks the properties usually associated with the clause-to-variable ratio. The most obvious structure comes from the generation of clauses as groups of four, with each clause in the group sharing all three variables. Furthermore, one variable in each of these clauses never appears again after this group of four clauses. For the remainder of this paper, we investigate the structural property introduced during the normalization process from boolean expression to 3-CNF. We empirically investigate the properties of structural families of formulae, each of which include all normalized formulae.

1.1 Methodology

To measure the behavior of each set of formulae, we randomly generate 1000 formulae with the desired properties. A complete SAT algorithm is used to determine whether each is satisfiable.

The actual generation of each family of formulae is slightly different, but they all follow the same basic structure. The appropriate number of clauses are generated, each having 3 terms. These terms are randomly (50% each) chosen as variables or negated variables. With exceptions noted in certain families, the variables themselves are chosen randomly from the pool of available variables, with each variable having an equal chance of being chosen.

Following this procedure will occasionally generate a formula that does not make use of all the variables defined to be of interest. We skip those formulae, generating additional formulae until the full 1000 formulae of the appropriate size are available. For some families, only a few formulae, if any, needed to be skipped. Others required a lot of skipping. Table III indicates generation sets that required significant skipping.

TABLE III
THE FAMILY, NUMBER OF VARIABLES AND CLAUSE-TO-VARIABLE RATIO WHERE RANDOM GENERATION REQUIRED SKIPPING MORE THAN 5% OF THE GENERATED FORMULAE.

family	vars	C/V	total skipped
S_3	30	3.0	304
S_3	30	3.2	183
S_3	30	3.3	90
S_3	40	3.0	648
S_3	40	3.2	291
S_3	40	3.3	161
S_3	40	3.5	50
S_3	50	3.0	818
S_3	50	3.2	369
S_3	50	3.3	254
family	vars	C/V	total skipped
S_4	30	3.0	24,391
S_4	30	3.2	16,862
S_4	30	3.3	11,182
S_4	30	3.5	8,334
S_4	30	3.7	5,007
S_4	40	3.0	109,512
S_4	40	3.2	50,469
S_4	40	3.3	35,026
S_4	40	3.5	19,166
S_4	40	3.7	11,540
S_4	50	3.0	356,534
S_4	50	3.2	150,641
S_4	50	3.3	106,325
S_4	50	3.5	41,683
S_4	50	3.7	25,705
family	vars	C/V	total skipped
normalized	30	3.0	201
normalized	30	3.2	73
normalized	40	3.0	477
normalized	40	3.2	113
normalized	50	3.0	549
normalized	50	3.2	163
normalized	50	3.3	66

2.0 Random Formulae

As a baseline, we define the family of all 3-CNF formulae, which we call S_0. All further families of formulae are subsets of S_0. To provide a baseline to which other families can be compared, we measure the performance of 1000 randomly generated formulae with 30, 40 and 50 variables and clause-to-variable ratios of 3.0, 3.2, 3.3, 3.5 and 3.7. Table IV shows the satisfiability of the formulae with these characteristics.

TABLE IV
THE PROBABILITY (AS A PERCENTAGE) OF BEING SATISFIABLE FOR A RANDOM FORMULA IN S_0 WITH EITHER 30, 40 OR 50 VARIABLES, AND CLAUSE-TO-VARIABLE RATIO EITHER 3.0, 3.2, 3.3, 3.5 OR 3.7.

	3.0	3.2	3.3	3.5	3.7
30	100	100	99.8	99.3	95.9
40	100	100	99.9	99.5	97.3
50	100	100	99.9	99.7	98.8

This data agrees with the previously published work. The likelihood of satisfaction grows monotonically with the clause-to-variable ratio. For a given clause-to-variable ratio, an increasing number of variables slightly increases the chance of being satisfiable.

The other interesting baseline for comparison is the behavior of normalized arbitrary boolean formulae. We need to use a somewhat different approach for generating random formulae in this family. We generate random boolean formula with the desired number of variables and operators. Each binary operator is randomly chosen from *and* (50%) or *or* (50%). Starting from the top operator, each child is randomly chosen to be another operator (75%) or a leaf (25%) until the desired number of operators have been selected, when all children are forced to be leaves. Each leaf is either a variable (50%) or a negated variable (50%).

TABLE V
PROBABILITY OF BEING SATISFIABLE FOR A RANDOM FORMULA NORMALIZED TO A 3-CNF FORMULAE WITH THE INDICATED FINAL NUMBER OF VARIABLES AND CLAUSE-TO-VARIABLE RATIO. WHEN NECESSARY, THE CLAUSE-TO-VARIABLE RATIO WAS ADJUSTED, AS INDICATED IN PARENTHESES.

	3.0	3.2	3.3	3.5	3.7
30	91.6(3.07)	91.8	88.3(3.33)	86.9(3.47)	79.7(3.73)
40	92.8	92.5	91.6	88.2	81.4
50	94.1(3.04)	92.6	91.9(3.28)	89.3(3.52)	85.3(3.68)

As noted previously, these formulae are far more difficult to satisfy than the corresponding random 3-CNF formulae. The probability of being satisfying generally falls as the clause-to-variable ratio climbs. Conversely, the chance of being satisfiable generally climbs as the number of variables grows.

3.0 Grouped Variable Formulae

The first family of interest captures the most basic structuring of the normalization, partitioning the clauses into sets of four with each group of four sharing at least one variable. We call this family S_1. Formally, we define this family as

$S_1 \subset S_0$ where each formula $\phi_1 \in S_1$ has a multiple of 4 clauses, with each group of 4 clauses having the form $(l_{1,1} \vee l_{1,2} \vee l_{1,3}) \wedge (l_{2,1} \vee l_{2,2} \vee l_{2,3}) \wedge (l_{3,1} \vee l_{3,2} \vee l_{3,3}) \wedge (l_{4,1} \vee l_{4,2} \vee l_{4,3})$, where $l_{i,1}$ is of the form $\neg\, v_X$ or v_X and $l_{i,j}$ is of the form $\neg\, v_{i,j}$ or $v_{i,j}$ where $j > 1$ and $v_{i,j} \neq v_X$ (and $v_{i,2} \neq v_{i,3}$).

Note that a formula in S_1 must satisfy all the properties of a formula in S_0, in addition to the new properties introduced in S_1, as $S_1 \subset S_0$.

An example formula from S_1 is

$$\begin{aligned}\phi_1 = \Big(&(v_1 \vee v_2 \vee v_3) \wedge (v_1 \vee \neg v_3 \vee \neg v_4)\\ &\wedge (\neg v_1 \vee v_3 \vee v_4) \wedge (\neg v_1 \vee \neg v_3 \vee v_4)\\ \wedge &(v_2 \vee v_1 \vee \neg v_4) \wedge (\neg v_2 \vee v_3 \vee \neg v_4)\\ &\wedge (\neg v_2 \vee \neg v_3 \vee \neg v_4) \wedge (v_2 \vee \neg v_1 \vee v_3)\\ \wedge &(v_3 \vee \neg v_1 \vee \neg v_4) \wedge (\neg v_3 \vee \neg v_2 \vee v_4)\\ &\wedge (\neg v_3 \vee v_1 \vee \neg v_2) \wedge (v_3 \vee \neg v_2 \vee v_4)\Big).\end{aligned}$$

Variable v_1 is the common variable for the first group of four clauses, v_2 for the second group of four clauses and v_3 for the final four clauses.

Table VI shows the satisfiability of various clause and variable combinations of this family. For this and all later families, not all combinations of variables and clauses are possible, as each formula must have a multiple of four clauses. Where necessary, we have adjusted the clause-to-variable ratio to the closest possible value.

TABLE VI

PROBABILITY OF BEING SATISFIABLE FOR A RANDOMLY GENERATED MEMBER OF S_1 WITH THE INDICATED NUMBER OF VARIABLES AND CLAUSE-TO-VARIABLE RATIO. BECAUSE S_1 REQUIRES MULTIPLES OF FOUR CLAUSES IN EACH FORMULA, THE CLAUSE-TO-VARIABLE RATIO WAS ADJUSTED IN SOME CASES, AS INDICATED IN PARENTHESES.

	3.0	3.2	3.3	3.5	3.7
30	100(3.07)	100	99.9(3.33)	99.5(3.47)	97(3.73)
40	100	100	100	99.6	98.0
50	100(3.04)	100	100(3.28)	99.8(3.52)	99.1(3.68)

This family is nearly indistinguishable from the more general S_0 in terms of satisfiability. The satisfiability declines as the clause-to-variable ratio grows and climbs more gently with the number of variables. Simply grouping clauses around variables is not sufficient structure to significantly alter the satisfiability of the formulae.

The second notable structural characteristic of the normalized formula is the limitations on the further appearance of the distinguished variable for each group of clauses. This variable can appear in any clause coming before the group of four clauses distinguishing it, but never appears in any clause coming after that group. The next family, which we call S_2, captures this constraint. As with all other families considered, S_2 is a superset of all normalized formulae.

$S_2 \subset S_1$ where each formula $\phi_2 \in S_2$ has a multiple of 4 clauses, with each group of 4 clauses having the form $(l_{1,1} \vee l_{1,2} \vee l_{1,3}) \wedge (l_{2,1} \vee l_{2,2} \vee l_{2,3}) \wedge (l_{3,1} \vee l_{3,2} \vee l_{3,3}) \wedge (l_{4,1} \vee l_{4,2} \vee l_{4,3})$, where $l_{i,1}$ is of the form $\neg v_X$ or v_X and $l_{i,j}$ is of the form $\neg v_{i,j}$ or $v_{i,j}$ where $j > 1$ and $v_{i,j} \neq v_X$ and v_X does not appear in any clause appearing after this group of four in ϕ_2.

An example formula from S_2 is

$$\begin{aligned}\phi_2 = \Big(&(v_1 \vee v_2 \vee v_6) \wedge (v_1 \vee \neg v_3 \vee \neg v_4)\\ &\wedge (\neg v_1 \vee v_6 \vee v_4) \wedge (\neg v_1 \vee \neg v_6 \vee v_4)\\ \wedge &(v_2 \vee v_6 \vee \neg v_4) \wedge (\neg v_2 \vee v_3 \vee \neg v_4)\\ &\wedge (\neg v_2 \vee \neg v_4 \vee \neg v_5) \wedge (v_2 \vee \neg v_5 \vee v_6)\\ \wedge &(v_3 \vee \neg v_5 \vee \neg v_4) \wedge (\neg v_3 \vee \neg v_6 \vee v_4)\\ &\wedge (\neg v_3 \vee v_5 \vee \neg v_6) \wedge (v_3 \vee \neg v_5 \vee v_4)\Big).\end{aligned}$$

Note that v_1 does not appear beyond the first four clauses that distinguish it, while v_2 does not appear after the second group of four clauses.

Table VII shows the satisfiability of random elements of S_2 with various clause and variable combinations.

TABLE VII

PROBABILITY OF BEING SATISFIED FOR A RANDOMLY GENERATED MEMBER OF S_2 WITH THE INDICATED NUMBER OF VARIABLES AND CLAUSE-TO-VARIABLE RATIO.

	3.0	3.2	3.3	3.5	3.7
30	99.8(3.07)	99.5	98.0(3.33)	91.1(3.47)	60.8(3.73)
40	100	99.8	98.4	90.5	62.1
50	100(3.04)	100	99.2(3.28)	87.4(3.52)	66.3(3.68)

While closer to the satisfiability of normalized arbitrary formulae, this family is still not a good match for them. Although being less satisfiable for high clause-to-variable ratios, this family is significantly easier to satisfy at lower clause-to-variable ratios.

The next family, called S_3 invokes the full set of constraints from the normalization to attempt to match its behavior more precisely. More precisely, S_3 is the subset of S_2 where all four clauses in each grouping use the same three variables.

$S_3 \subset S_2$ where each formula $\phi_3 \in S_3$ has a multiple of 4 clauses, with each group of 4 clauses having the form $(l_{1,1} \vee l_{1,2} \vee l_{1,3}) \wedge (l_{2,1} \vee l_{2,2} \vee l_{2,3}) \wedge (l_{3,1} \vee l_{3,2} \vee l_{3,3}) \wedge (l_{4,1} \vee l_{4,2} \vee l_{4,3})$, where $l_{i,j}$ is of the form $\neg v_j$ or v_j

and each v_j is one of the three variables associated with this group of clauses.

An example formula from S_3 is

$$\phi_3 = \Big((v_1 \vee v_4 \vee v_3) \wedge (v_1 \vee \neg v_4 \vee \neg v_3) \wedge (\neg v_1 \vee v_4 \vee v_3) \wedge (\neg v_1 \vee \neg v_4 \vee v_3) \wedge (v_2 \vee v_4 \vee \neg v_7) \wedge (\neg v_2 \vee v_4 \vee \neg v_7) \wedge (\neg v_2 \vee \neg v_4 \vee \neg v_7) \wedge (v_2 \vee \neg v_4 \vee v_7) \wedge (v_3 \vee \neg v_6 \vee \neg v_7) \wedge (\neg v_3 \vee \neg v_6 \vee v_7) \wedge (\neg v_3 \vee v_6 \vee \neg v_7) \wedge (v_3 \vee \neg v_6 \vee v_7)\Big).$$

Table VIII shows the satisfiability of various clause and variable combinations of this family.

TABLE VIII

PROBABILITY OF BEING SATISFIABLE FOR A RANDOMLY GENERATED MEMBER OF FAMILY S_3 WITH THE INDICATED NUMBER OF VARIABLES AND CLAUSE-TO-VARIABLE RATIOS.

	3.0	3.2	3.3	3.5	3.7
30	86.6(3.07)	81.6	72.2(3.33)	64.2(3.47)	39.2(3.73)
40	86.5	76.1	67.5	51.8	31.9
50	82.5(3.04)	72.3	66.7(3.28)	38.5(3.52)	24.8(3.68)

Despite having the same grouping structure as the normalized formulae, random elements of this family are significantly harder to satisfy than corresponding random normalized formulae. Notably, S_3 has a negative correlation between number of variables and satisfiability, reversing the behavior of all previous families considered.

The final family considered in this paper, called S_4 reduces the constraints placed on S_3, leaving only the need for each group of four clauses to utilize the same variables. Note that S_4 is a subset of S_0, but not S_1 or any later family. Formally

$S_4 \subset S_0$ where each formula $\phi_4 \in S_4$ has a multiple of 4 clauses, with each group of 4 clauses having the form $(l_{1,1} \vee l_{1,2} \vee l_{1,3}) \wedge (l_{2,1} \vee l_{2,2} \vee l_{2,3}) \wedge (l_{3,1} \vee l_{3,2} \vee l_{3,3}) \wedge (l_{4,1} \vee l_{4,2} \vee l_{4,3})$, where $l_{i,j}$ is of the form $\neg v_j$ or v_j and v_j is one of three variables associated with group of 4 clauses.

An example formula from S_4 is

$$\phi_4 = \Big((v_4 \vee v_1 \vee v_5) \wedge (v_4 \vee \neg v_1 \vee \neg v_5) \wedge (\neg v_4 \vee v_1 \vee v_5) \wedge (\neg v_4 \vee \neg v_1 \vee v_5) \wedge (v_2 \vee v_4 \vee \neg v_7) \wedge (\neg v_2 \vee v_4 \vee \neg v_7) \wedge (\neg v_2 \vee \neg v_4 \vee \neg v_7) \wedge (v_2 \vee \neg v_4 \vee v_7) \wedge (v_7 \vee \neg v_1 \vee \neg v_3) \wedge (\neg v_7 \vee \neg v_1 \vee v_3) \wedge (\neg v_7 \vee v_1 \vee \neg v_3) \wedge (v_7 \vee \neg v_1 \vee v_3)\Big).$$

Table IX shows the satisfiability of various clause and variable combinations of this family

TABLE IX

PROBABILITY OF BEING SATISFIABLE FOR RANDOM ELEMENTS OF FAMILY S_4 HAVING THE INDICATED NUMBER OF VARIABLES AND CLAUSE-TO-VARIABLE RATIO.

	3.0	3.2	3.3	3.5	3.7
30	89.4(3.07)	82.1	77.8(3.33)	71.8(3.47)	58.9(3.73)
40	87.3	78.5	73.2	62.9	51.7
50	85.7(3.04)	75.5	68.1(3.28)	55.5(3.52)	45.1(3.68)

We emphasize that a formula in S_3 must satisfy all the properties of a formula in S_0, S_1 and S_2 in addition to the new properties introduced in S_3, as $S_3 \subset S_2 \subset S_1 \subset S_0$. However, S_4 only inherits the properties from S_0 as $S_4 \subset S_0$.

While being in general less satisfiable than the family of normalized formulae, the family S_4 provides the best match to it. It seems plausible that the relative difficulty in satisfying normalized formulae comes from their membership in S_4, although other factors are also clearly at work. Like S_3, this family shows a negative correlation between number of variables and satisfiability,

4.0 Conclusions

Significant previous work, both empirical and theoretic, has described the impact of the clause-to-variable ratio on the satisfiability of random 3-CNF formula. Unfortunately, arbitrary boolean formula that have been normalized to 3-CNF largely stand as an exception to this generality.

This paper empirically investigates four possible structural descriptions of normalized formulae in an attempt to better model their satisfiability. Although no simple structural property investigated thus far can completely account for the satisfiability profile of normalized formulae, the family S_4 comes far closer to matching these characteristics than the clause-to-variable ratio alone can.

4.1 Future Work

An obvious future tack is to investigate how changing the size of the groups changes the expected satisfiability. It seems likely that increasing group size will tend to decrease satisfiability, at least up to a point. The degree to which this is true remains an open question.

We would also like to develop a theoretical model of why this grouping changes the satisfiability so significantly.

We know from preliminary investigations that the mixture of ands versus ors in the pre-normalized formula has a significant impact on the satisfiability of the formula.

Anecdotally, we know that this ratio varies significantly between formulae found in the wild. This ratio may provide an equally valid, but more tractable, predictor of satisfiability of formulae.

5.0 References

[1] P. Beame, R. Karp, T. Pitassi, M. Saks, "On the Complexity of Unsatisfiability Proofs for Random k-CNF Formulas", *Proceedings of the* 30^{th} *Annual ACM Symposium on the Theory of Computing* (1998) 561-571.

[2] R.E. Bryant, "Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams", *ACM Computing Surveys*, Vol. 24, No. 3 (1992) 294-318.

[3] J.R. Burch, E.M. Clarke, K.L. McMillan, D.L. Dill, and J. Hwang, "Symbolic model checking: 10E20 states and beyond", in *5th Symposium on Logic in Computer Science (LICS '90).*

[4] S.A. Cook, "The Complexity of Theorem Proving Procedures", *Proceedings of the Third Annual ACM Symposium on the Theory of Computing*, ACM, NY (1971) 151-158.

[5] T. Cormen, C. Leiserson, R. Rivest and C. Stein, *Introduction to Algorithms*, second edition, MIT Press and McGraw-Hill, 2001.

[6] M. Davis and H. Putnam, "A Computing Procedure for Quantification Theory" *JACM 7* (1960) 201-205.

[7] M. Davis, G. Logemann and D. Loveland, "A Machine Program for Theorem Proving", *Communications of the ACM 5* (1962) 394-397.

[8] W.F. Dowling and J.H. Gallier, "Linear-time algorithms for testing the satisfiability of Horn formulas", *Journal of Logic Programming 1* (1984) 267-284.

[9] O. Dubois, Y. Boufkhad and J. Mandler, "Typical Random 3-SAT Formulae and the Satisfiability Threshold", Electronic Colloquium on Computational Complexity (ECCC 2003) 2003.

[10] J. Franco, "Some Interesting Research Directions in Satisfiability", *Annals of Mathematics and Artificial Intelligence 28* (2000) 7-15.

[11] J. Franco and M. Paull, "Probabilistic Analysis of the Davis-Putnam Procedure for Solving the Satisfiability Problem", *Discrete Applied Mathematics 5* (1983) 77-87.

[12] A.C. Kaporis, L.M. Kirousis and E. Lalas, "Selecting Complementary Pairs of Literals", *Electronic Notes in Discrete Mathematics 16* (2003) 47-70.

[13] D.G. Mitchell and H.J. Levesque, "Some Pitfalls for Experimenters with Random SAT", *Artificial Intelligence*, *81* (1996) 111-125.

[14] B. Selman, "Understanding Problem Hardness: Recent Developments and Directions", ISAT meeting, April 1999, Seattle, WA.

[15] B. Selman, H. Kautz and B. Cohen, "Local Search Strategies for Satisfiability Testing" in *Cliques, Coloring and Satisfiability*, D. Johnson and M. Trick eds., DIMACS Series in Discrete Mathematics and Theoretical Computer Science, V. 26, 1996, pp. 521-532.