Horn upper bounds of random 3-CNF: a computational study*

Marina Langlois Robert H. Sloan György Turán[1]
Email: mirodo1|sloan|gyt@uic.edu
University of Illinois at Chicago
[1]Also University of Szeged, Hunagary

Abstract

Experiments are reported with computing various Horn upper bounds of random 3-CNF formulas of different densities (i.e., clause to variable ratios). Among four algorithms tested, the most successful one uses renaming of variables, and generates Horn implicates of limited size only. The output sizes and approximation errors exhibit unimodal patterns with maxima in some intermediate range of densities.

1 Introduction

A general formulation of the reasoning problem in propositional logic is to decide if a clause C is implied by a CNF expression φ. Here φ is often viewed as a fixed *knowledge base*, and it is assumed that we have to answer a large number of *queries* C for the same knowledge base. Therefore, it may be useful to preprocess φ into a more tractable form, resulting in a new knowledge base which may be only approximately equivalent to the original one. This approach, called *knowledge compilation*, goes back to the seminal work of Selman and Kautz [20]. (See also [6].)

Selman and Kautz suggested considering *Horn formulas* approximating the initial knowledge base from above and below, and using these formulas to answer the queries. The *Horn least upper bound (Horn-LUB)* of φ is the conjunction of all Horn prime implicates of φ [1]. It can also be obtained as the *closure, under intersections, of the set of satisfying truth assignments*. This natural interpretation suggests that the Horn-LUB may be of interest as a combinatorial object in itself.

Queries to Horn formulas can be answered efficiently, but the approach can have the following drawbacks: it may be inefficient as the resulting Horn upper bound may be large, and it may fail to answer certain queries (those implied by the lower bound, but not implied by the upper bound).

Initial experiments for the evaluation of Horn approximations were performed by Kautz and Selman [14]. Boufkhad [5] gives the results of experiments for Horn lower bounds using an extension to renamable Horn formulas. The main test examples were random 3-CNF formulas with density around 4.26, which are well known to be hard for satisfiability algorithms[2].

In this paper we report some results of a detailed computational study of the Horn upper bound. We compare several modified versions, including a renamable variant which uses an approximation algorithm

*This material is based upon work supported by the National Science Foundation under grant CCF-0431059.

[1]We omit the definition of Horn *greatest lower bounds* (which, as opposed to LUBs, are not unique), as those will not be discussed in this paper.

[2]Kautz and Selman also considered a class of planning problems, and Boufkhad also considered 4-CNF formulas.

of Boros [4] to find a large Horn renamable subformula of the original knowledge base, and approximate versions which resolve clauses of bounded size only. The conclusion is that the best compromise in terms of running time, the size of the Horn upper bound and the quality of the approximation is obtained by *using renaming, and computing resolvents up to a certain bounded size*. In the majority of our experiments we measured approximation quality using the relative error measure corresponding to the combinatorial interpretation of the Horn-LUB.

Concerning output size and approximation error for random 3-CNF of different densities, the emerging pattern is what could perhaps be called the *Horn bump*: as a function of the density, the averages of these quantities are unimodal (i.e., increasing then decreasing) with maxima in some intermediate range of densities. According to previous experiments, the set of satisfying truth assignments at these densities form a single connected cluster (Martin et al. [16]), thus there may be some other structural changes that account for these phenomena.

2 Background

A clause is a disjunction of literals (unnegated or negated variables). It is a Horn clause if it contains at most one unnegated literal. A CNF is a conjunction of clauses; it is a 3-CNF if each clause contains exactly 3 literals. A *Horn formula* or *Horn-CNF* is a conjunction of Horn clauses. A clause C is an *implicate* of a CNF expression φ if every vector satisfying φ also satisfies C; it is a *prime* implicate if none of its sub-clauses is an implicate.

The Horn least upper bound of φ (Horn-LUB(φ)) is the conjunction of all Horn prime implicates of φ. More generally, a *Horn upper bound* is a conjunction of some Horn implicates. Selman and Kautz [20] give an algorithm for computing Horn-LUB(φ). Their algorithm works iteratively, starting from φ, performing all possible resolutions between two clauses, at least one of which is non-Horn, and simplifying whenever possible (for a detailed proof of the correctness of the algorithm, see del Val [8]). We refer to this algorithm also as Horn-LUB.

The *intersection* of vectors $(x_1, \ldots, x_n)$, $(y_1, \ldots, y_n) \in \{0, 1\}^n$ is $(x_1 \wedge y_1, \ldots, x_n \wedge y_n)$. A Boolean function can be described by a Horn formula if and only if its set of satisfying truth assignments is closed under intersection [12, 17]. Thus Horn-LUB(φ) can also be obtained by taking $T(\varphi)$, the set of satisfying truth assignments of φ, and closing it under intersections.

A *renaming* of a variable x in a formula is obtained by switching the occurrences of x and $\bar{x}$. A CNF is *Horn renamable* if one can rename some of its variables to turn it into a Horn formula. It can be decided in polynomial time if a CNF is Horn renamable [2, 15], but finding a largest Horn renamable sub-CNF is NP-hard (Crama et al. [7]). Boros [4] gave an approximation algorithm for finding a large Horn renamable sub-CNF in an arbitrary CNF. Note that the Horn-LUB obtained after a renaming corresponds to intersection closure with respect to a reorientation of the hypercube, using another vertex as the minimum instead of the all 0 vector. Van Maaren and van Norden [21] considered the connection between the efficiency of satisfiability algorithms and the size of a largest renamable sub-CNF for random 3-CNF.

An n-variable *random 3-CNF* formula of *density* α is obtained by selecting $\alpha \cdot n$ clauses of size 3, selecting each clause from the uniform distribution over all such clauses. There is a vast literature on random 3-CNF (and, more generally, on k-CNF) formulas, including both theoretical and experimental results. Experiments show that there is a *phase transition* from the formulas being almost surely satisfiable to being almost surely unsatisfiable around $\alpha = 4.26$. It has been proved that there is indeed a sharp transition [11], and various bounds are known for its location [1, 9].

3 Computational results

Besides Horn-LUB, we considered three other algorithms to compute Horn approximations. The algorithm Renamed-Horn-LUB finds a renaming of the variables using a heuristic algorithm of [4], and then applies the Horn-LUB algorithm. The algorithm 4-Horn-LUB works as the Horn-LUB algorithm, but only performs resolution steps that produce clauses of size at most 4. Finally, Renamed-4-Horn-LUB is the combination of the last two algorithms: it first performs a renaming, and then does those resolution steps that produce clauses of size at most 4.

Table 1 gives *running times*, and Table 2 *output sizes*, for Horn-LUB, for different numbers of variables and different densities. (All running times reported in this paper were measured on a Dell laptop with a 2.40 GHz CPU and 256MB RAM.) As we wished to perform exhaustive testing (over all truth assignments) in several cases, it was not feasible to go much above 20 variables. For Horn LUB, in any event, running times become prohibitive as one gets significantly above 20 variables, as indicated by the last column of Figure 1. As the case of 20 variables is considered in more detail below, the output size for this case is computed with a smaller step size. The output size is unimodal, with maximum around density 2.5.

	Number of variables					
α	12	14	16	18	20	22
1	0.01	0.02	0.07	0.19	0.96	7.53
2	0.07	0.55	2.74	19.71	50.49	$>$ 15 min
3	0.14	0.79	2.91	24.27	126.81	$>$ 15 min
4	0.11	0.42	4.90	27.07	224.561	$>$ 15 min

Table 1: Mean running time to compute the Horn-LUB of random 3-CNF formulas φ in CPU sec as function of number of variables and density α of φ, averaged over 50 runs.

	Number of variables			
α	16	18	20	22
1	36.28	66.48	96.14	146.60
1.5	$--$	$--$	517.3	$--$
2	306.46	565.00	1044.72	$>$ 15 min
2.5	$--$	$--$	1255.06	$>$ 15 min
3	330.04	585.42	898.76	$>$ 15 min
3.5	$--$	$--$	599.60	$>$ 15 min
4	195.40	305.40	409.34	$>$ 15 min

Table 2: Mean size (number of clauses) of the Horn-LUB as function of density and number of variables, averaged over 50 runs. ($--$ means not computed; $>$ 15 min means not run to completion due to long running time).

Tables 3 and 4 contain similar data for the 4-Horn-LUB algorithm. The running times of this algorithm are significantly smaller. The output sizes are also smaller, and again unimodal, with maximum between density 2.5 and 3.

As all these algorithms produce a conjunction of some implicates of the original formula φ, their output is implied by φ; that is, each algorithm's output has a one-sided error. The *relative error* of such an algorithm

α	Number of variables 20	30	40
1	0.00	0.03	0.01
2	0.16	0.27	0.21
3	1.49	21.41	100.88
4	0.909	28.29	261.17

Table 3: Running time to compute Renamed-4-Horn-LUB in CPU sec. Averaged over 50 runs.

α	Number of variables 16	18	20	22
1	22.78	23.64	28.24	30.36
1.5	$--$	$--$	70.52	$--$
2	164.06	189.26	236.24	243.36
2.5	$--$	$--$	547.6	$--$
3	302.16	479.26	704.84	968.50
3.5	$--$	$--$	639.26	$--$
4	244.05	333.22	452.74	567.16

Table 4: Mean size (number of clauses) of the Renamed-4-Horn-LUB as function of density and number of variables, averaged over 50 runs ($--$ means not computed).

A on an input formula φ is measured by

$$r_A(\varphi) = \frac{|T(A(\varphi))|}{|T(\varphi)|},$$

where $A(\varphi)$ denotes the formula output by A on φ, and $|T(\varphi)|$ denotes the number of satisfying truth assignments of φ.

Figures 1 and 2 present computational results for the relative errors of the four algorithms for different densities on 20 variables. Figure 1 gives statistics for the Horn-LUB, resp., the Renamed-Horn-LUB algorithm, and Figure 2 gives statistics for the 4-Horn-LUB, resp., Renamed-4-Horn-LUB algorithm. The error curves are again unimodal, with maximum around 2.4. Experiments for fewer variables show similar values of the maxima.

The overall conclusion is that Renamed-4-Horn-LUB is the best algorithm for 20 variables.[3] It is significantly faster than either Horn-LUB or Renamed-Horn-LUB, and it is even somewhat faster than 4-Horn-LUB. Its output size is significantly smaller than those of Horn-LUB or Renamed-Horn-LUB, but larger than that of 4-Horn-LUB. On the other hand, its relative error is only slightly worse than that of Renamed-Horn-LUB, which has the smallest relative error. The running times and output size of Renamed-Horn-LUB and 4-Horn-LUB are omitted, and will be included in the full version of the paper. Replacing the limit 4 on clause size with 3, or even using all implicates of size at most 3, results in a large increase in the relative error for densities below the satisfiability threshold.

It is to be expected, and supported by some experimental evidence, that as the number of variables increases, the limit on the clause size required for producing reasonable relative error will also increase.

[3] Preliminary experiments show Renamed-4-Horn-LUB also performing relatively well for up to at least 40 variables, but for 40 variables, simply *measuring* performance is computationally expensive.

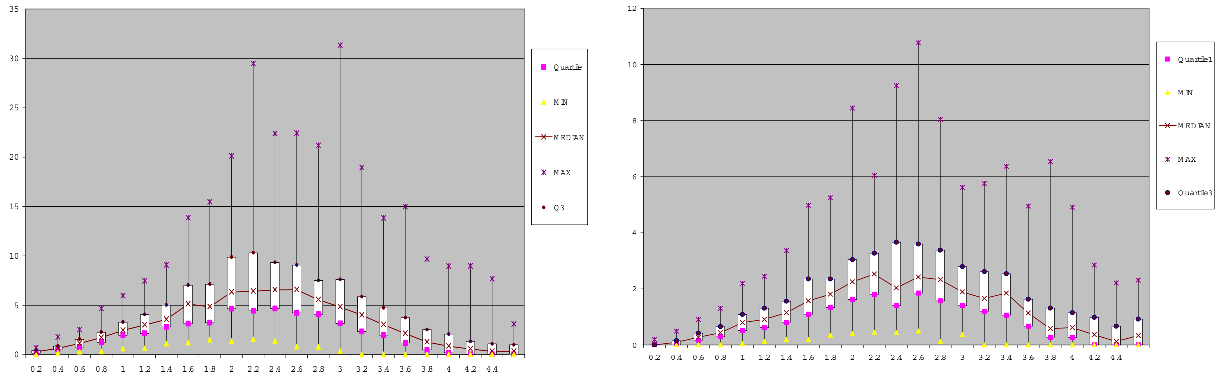

Figure 1: Relative errors $r_A(\varphi)$ of Horn-LUB (left) and Renamed-Horn-LUB (right) for φ with 20 variables as function of density. Measured by exhaustive examination of all length 20 vectors. Notice that the two scales for $r_A(\varphi)$ are different; Renamed-Horn-LUB has much lower relative error. Averaged over 100 runs.

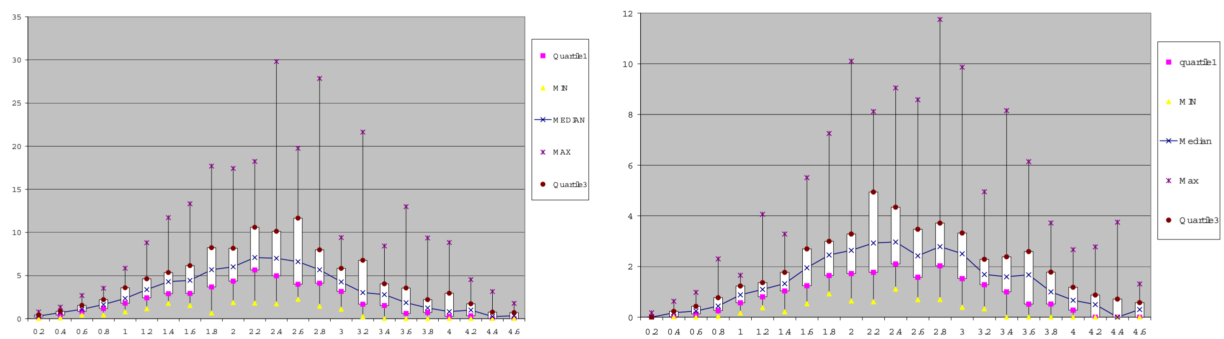

Figure 2: Relative errors of 4-Horn-LUB (left) and Renamed-4-Horn-LUB (right) for formulas with 20 variables as function of density. Measured by exhaustive examination of all length 20 vectors. Again, the two scales are different; Renamed-4-Horn-LUB has much lower relative error. Averaged over 100 runs.

Another way to evaluate the algorithms is to consider the number of queries that are answered incorrectly by their output. In Figure 3 we present such results for the Horn-LUB algorithm. Horn-LUB's output answers all Horn queries correctly, and all non-Horn prime implicate queries incorrectly. Figure 3 shows the fraction of non-Horn prime implicates compared to the total number of prime implicates, without and with renaming, for 20 variables. We again observe a similar unimodal behavior, but the maximum is lower this time, around density 1.6. Similar comparisons for the other algorithms will be included in the full version of the paper.

In their experiments, Selman and Kautz [20] considered the case of prime implicates of size 1 and 2. Prime implicates of size 1 correspond to the notion of the *backbone* of a CNF, considered by Monasson et al. [18]. The backbone of a CNF consists of those variables which have a forced value in every satisfying truth assignment. Monasson et al. present interesting experimental results on the size of the backbone, and give an argument showing that asymptotically, for a random 3-CNF of density α below the SAT/UNSAT threshold, the backbone cannot contain a positive fraction of the variables with positive probability. Figure 4 shows the fraction of clauses of size 1 and 2 that are implicates of a random 3-CNF, for different densities.

For a larger number of variables it is not feasible to search the whole space exhaustively. In order to

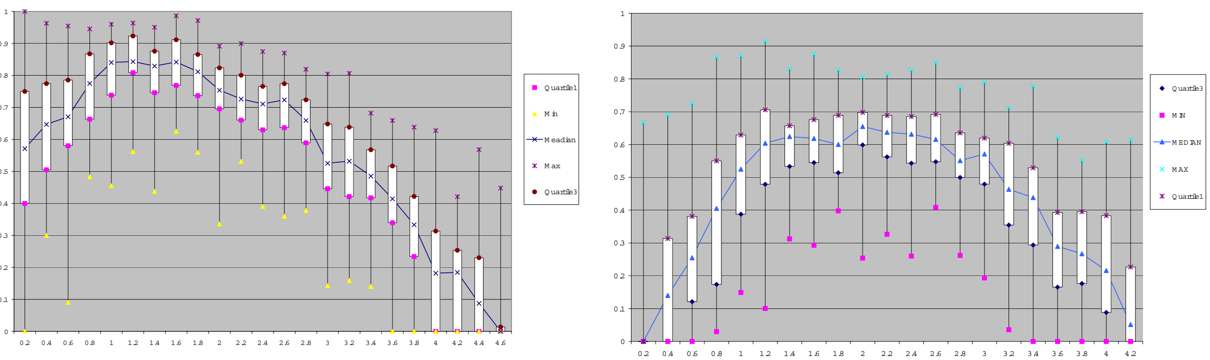

Figure 3: Fraction of all prime implicates of a 3-CNF formula on 20 variables that are *not* Horn as a function of α. Left figure shows statistics for 3-CNFs; right figure shows statistics for 3-CNFs that had renaming applied first. Averaged over 50 runs.

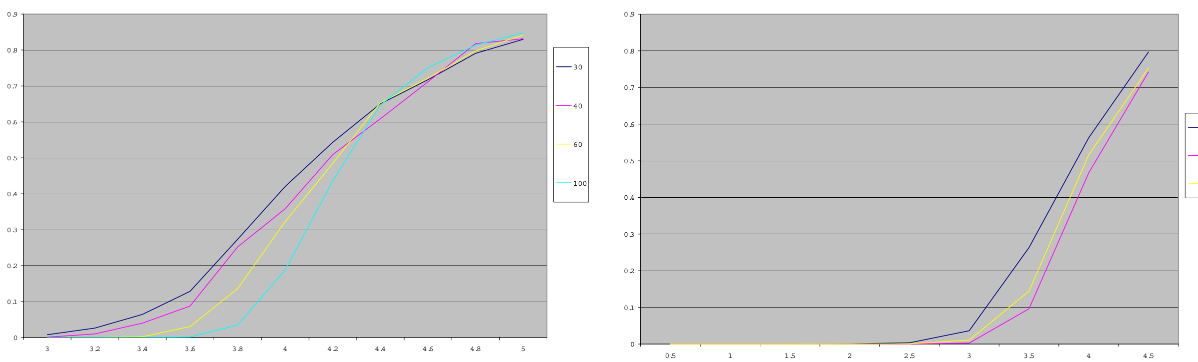

Figure 4: Left: Ratio of number of size 1 implicates of an n-variable 3-CNF formula to n (for $n = 30, 40, 60, 100$) . Right: Ratio of number of size 2 implicates of an n-variable 3-CNF formula to $3\binom{n}{2}$ (which is an upper bound on the maximum number of possible size 2 implicates) (for $n = 30, 40, 50$). All curves (both sides) generated by considering 200 randomly generated formulas for a range of values of α.

estimate the relative error, one could try to use random sampling by generating a random satisfying truth assignment of the Horn upper bounds. This raises the question whether a random satisfying truth assignment of a Horn formula can be generated uniformly in polynomial time. As far as we know, this is open. In related work, Roth [19] showed that it is NP-hard to approximate the number of satisfying truth assignments of a Horn formula within a multiplicative factor of $2^{n^{1-\epsilon}}$ (for any ϵ) in polynomial time, even if the clauses have size 2 and every variable occurs at most 3 times, and Jerrum et al. [13] established a connection between almost uniform generation and randomized approximate counting.

We have started to do some initial experiments with various heuristics for randomly generating a satisfying truth assignment of a Horn formula. Table 5 compares the relative error of the Horn-LUB algorithm with the estimate of the relative error obtained by "random" sampling. The algorithm is a naive one, randomly selecting variables to be fixed (and deriving all assignments that are forced by the previous choices). Uniform random generation would require weighting the choices of the two values by the number of satisfying truth assignments corresponding to each value. As Table 5 shows, the estimates are rather close.

α	0.2	0.6	1	1.4	1.8	2.2	2.6	3	3.4	3.8	4.2
%	1.3	5.4	12	12.9	11.8	8.4	4.2	2.3	1.3	0.0	0.0

Table 5: Percentage error in measuring $r_A(\varphi)$ using “random” sampling of vectors versus exhaustive. Generates 100,000 “random” vectors; stops early if 50,000 *distinct* vectors obtained; uses only distinct vectors to measure error. Averaged over 10 runs.

4 Further remarks

We have considered four different algorithms for generating the Horn-LUB of a random 3-CNF for different densities, and concluded that over 20 variables the algorithm Renamed-4-Horn-LUB provides the best compromise in terms of running time, output size and relative error. We observed a Horn bump for the different performance measures in an intermediate range of densities.

There are several directions for further work that we plan to pursue. The extension of the experiments to error measures corresponding to the number of queries answered correctly, and to larger sizes using more computing power is in progress. This could perhaps give some information on how the relative errors grow with the number of variables. It would also be interesting to incorporate similar experiments concerning Horn greatest lower bounds, extending Boufkhad’s work [5]. The question of almost uniform random generation of a satisfying truth assignment seems to be of interest in itself.

A very interesting, and little understood, problem related to the phase transition of random 3-CNF is the evolution of random 3-CNF (see [16]), in analogy to the classic work of Erdős and Rényi [10] on the evolution of random graphs, and to the study of the evolution of random Boolean functions (see, e.g., Bollobás et al. [3]). In this context, it would perhaps be of interest to perform more experiments on the existence of the Horn bump, and to try to get some theoretical results.

The interpretation of the Horn least upper bound as the intersection closure of the set of satisfying truth assignments leads to the following general question: what is the expected size of the intersection closure of a random subset of $\{0,1\}^n$, given a probability distribution on the subsets? We are not aware of results of this kind. (Another closure problem, the dimension of the subspace spanned by a random set of vectors has been studied in great detail.) A basic case to consider would be when m random vectors are generated, each component of which is set to 1 with probability p. The size of the Horn-LUB of a random 3-CNF with a given density is a special case of the general question, when the distribution is generated by picking a random formula.

Acknowledgment We would like to thank Eli Ben-Sasson and Bart Selman for useful comments, and an anonymous referee for excellent feedback.

References

[1] D. Achlioptas. Lower bounds for random 3-SAT via differential equations. *Theor. Comput. Sci.*, 265:159–185, 2001.

[2] B. Aspvall. Recognizing disguised NR(1) instances of the satisfiability problem. *Journal of Algorithms*, 1:97–103, 1980.

[3] B. Bollobás, Y. Kohayakawa, and T. Łuczak. On the evolution of random Boolean functions. In P. Frankl, Z. Füredi, G. Katona, and D. Miklós, editors, *Extremal Problems for Finite Sets* (Visegrád),

volume 3 of *Bolyai Society Mathematical Studies*, pages 137–156, Budapest, 1994. János Bolyai Mathematical Society.

[4] E. Boros. Maximum renamable Horn sub-CNFs. *Discrete Appl. Math.*, 96-97:29–40, 1999.

[5] Y. Boufkhad. Algorithms for propositional KB approximation. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98) and of the 10th Conference on Innovative Applications of Artificial Intelligence (IAAI-98)*, pages 280–285, 1998.

[6] M. Cadoli. *Tractable Reasoning in Artificial Intelligence*. Number 941 in Lecture Notes In Artificial Intelligence. Springer, Berlin, Heidelberg, New York, 1995.

[7] Y. Crama, O. Ekin, and P. L. Hammer. Variable and term removal from Boolean formulae. *Discrete Applied Mathematics*, 75(3):217–230, 1997.

[8] A. del Val. First order LUB approximations: characterization and algorithms. *Artif. Intell*, 162(1-2):7–48, 2005.

[9] O. Dubois. Upper bounds on the satisfiability threshold. *Theor. Comput. Sci.*, 265:187–197, 2001.

[10] P. Erdős and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5:17–61, 1960.

[11] E. Friedgut. Sharp thresholds of graph properties, and the k-sat problem. *J. American Mathematical Society*, 12:1017–1054, 1999. Appendix by J. Bourgain.

[12] A. Horn. On sentences which are true on direct unions of algebras. *J. Symbolic Logic*, 16:14–21, 1951.

[13] M. R. Jerrum, L. G. Valiant, and V. V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theor. Comput. Sci.*, 43:169–188, 1986.

[14] H. Kautz and B. Selman. An empirical evaluation of knowledge compilation by theory approximation. In *Proceedings of the 12th National Conference on Artificial Intelligence*, pages 155–161, 1994.

[15] H. R. Lewis. Renaming a set of clauses as a Horn set. *J. ACM*, 25:134–135, 1978.

[16] O. C. Martin, R. Monasson, and R. Zecchina. Statistical mechanics methods and phase transitions in optimization problems. *Theor. Comput. Sci.*, 265:3–67, 2001.

[17] J. C. C. McKinsey. The decision problem for some classes without quantifiers. *J. Symbolic Logic*, 8:61–76, 1943.

[18] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky. Determining computational complexity from characteristic ‘phase transitions’. *Nature*, 400:133–137, 1999.

[19] D. Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82:273–302, 1996.

[20] B. Selman and H. Kautz. Knowledge compilation and theory approximation. *J. ACM*, 43:193–224, 1996.

[21] H. van Maaren and L. van Norden. Hidden threshold phenomena for fixed-density SAT-formulae. In *Int. Conf. Theory and Applications of Satisfiability Testing (SAT), LNCS*, volume 6, pages 135–149, 2003.