BRICS

Basic Research in Computer Science

An Algorithm for Exact Satisfiability Analysed with the Number of Clauses as Parameter

Bolette Ammitzbøll Madsen

BRICS Report Series **RS-04-18**

ISSN 0909-0878 **September 2004**

Copyright © 2004, Bolette Ammitzbøll Madsen.
BRICS, Department of Computer Science
University of Aarhus. All rights reserved.

Reproduction of all or part of this work
is permitted for educational or research use
on condition that this copyright notice is
included in any copy.

See back inner page for a list of recent BRICS Report Series publications.
Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark
Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/04/18/`

An Algorithm for Exact Satisfiability Analysed with the Number of Clauses as Parameter

Bolette Ammitzbøll Madsen
BRICS[*]
Department of Computer Science
University of Aarhus, Denmark
`bolette@brics.dk`

September 2004

Abstract

We give an algorithm for Exact Satisfiability with polynomial space usage and a time bound of $poly(L) \cdot m!$, where m is the number of clauses and L is the length of the formula. Skjernaa has given an algorithm for Exact Satisfiability with time bound $poly(L) \cdot 2^m$ but using exponential space. We leave the following problem open: Is there an algorithm for Exact Satisfiability using only polynomial space with a time bound of c^m, where c is a constant and m is the number of clauses?

Exact Satisfiability (XSAT) is the problem: given a formula F in conjunctive normal form, is there an assignment to all variables in F, such that exactly one literal in each clause is true? In this paper a formula F has m clauses and n variables. A literal is either a variable or the negation of a variable. The length of a formula L is the number of literals in the formula.

XSAT is NP-complete even when restricted to clauses containing at most three literals and all variables occurring only unnegated [7], and various exact algorithms have been given for this problem [6, 5]. So far all algorithms given for XSAT have been analysed using the number of *variables* as parameter. The best known algorithm for Exact Satisfiability (no limit on clause length) has a running time of $poly(L) \cdot 2^{0.2325n}$ [5]. This algorithm (or a variant thereof) also gives a time bound in the number of literals, but no good time bound in the number of *clauses* is known.

This is interestingly different from Satisfiability (no limit on clause length) for which good time bounds in the number of clauses have been proved (the

---

[*] Basic Research in Computer Science (`www.brics.dk`),
funded by the Danish National Research Foundation.

currently best is $poly(L) \cdot 2^{0.30897m}$ [4]), but it is still an open question if a time bound of $poly(L) \cdot 2^{\alpha n}$ with $\alpha < 1$ exists. The best time bounds with the number of variables as parameter are of the form $poly(L) \cdot 2^{n-f(n,m)}$ with $f(n,m) = 2\sqrt{n/\log n}$ [2] and $f(n,m) = n/\log(2m)$ [3].

The abovementioned algorithms all use polynomial space. If we allow the use of exponential space, XSAT can be solved in time $O(n \cdot 2^m)$ using dynamic programming [8]. We present an algorithm for XSAT using only polynomial space with time complexity $poly(L) \cdot m!$. The question remains:

- **Open problem:** Is there an algorithm for XSAT using only polynomial space and running in time c^m, where c is a constant?

We note that there are some interesting analogues to other NP-complete problems. A legal k-colouring of a graph $G = (V, E)$ is a mapping c of the vertices in V into the colours $\{1, \ldots, k\}$ such that no two neighbours have the same colour, i.e. $(v, u) \in E \Rightarrow c(v) \neq c(u)$. The Chromatic Number Problem is: given a graph G find the least k for which there is a legal k-colouring of G. We can solve this problem in time proportional to $n!$, where n is the number of vertices, by simply testing possible colourings. This is the best known time complexity if we only allow polynomial space, but allowing exponential space, a time complexity of $O(2.4023^n)$ can be achieved [1].

- **Open problem:** Is there an algorithm for Chromatic Number using only polynomial space and running in time c^n, where c is a constant?

We see the same picture looking at the Travelling Salesman Problem or the Hamiltonian Circuit Problem. Both problems can be solved in time proportional to $n!$, where n is the number of vertices, using polynomial space, and in time proportional to 2^n using exponential space.

- **Open problem:** Is there an algorithm for the Travelling Salesman Problem using only polynomial space and running in time c^n, where c is a constant?

We present the first analysis in the number of clauses of an algorithm for XSAT using only polynomial space:

Theorem 1. *Exact Satisfiability can be solved in time proportional to $m!$ and polynomial space.*

Proof. First we note that the hard case is when we do not have any negations. If we do have a variable that occurs both unnegated and negated, simply branching on setting this variable to either true or false, will remove a clause in both branches: When a literal is set to true, all other literals in the clause must be set to false to satisfy this clause, and we can remove it. The problem is when we have no negations: If we branch on a variable occurring only unnegated, we can only be sure to remove a clause in one of the branches. Thus we assume from here on that we have no negations (or assume that we have simply branched on all variables occurring negated, as the running time of the algorithm is larger than 2^m).

Now assume we are given an instance of the XSAT problem. The idea of this algorithm is that in some permutation of the clauses, the true variables in a satisfying assignment will occur in intervals of clauses.

Given a permutation of the clauses, number the clauses $1, \ldots, m$. We say that variable x occurs in the clause-interval from i to j ($i \leq j$) if x occurs in all the clauses $i, \ldots, j$ and no others.

In a satisfying assignment to the given instance, there is exactly one true variable (literal) in each clause. Then in some permutation of the clauses, the true variables of a satisfying assignment will occur in non-overlapping clause-intervals, which cover all clauses.

If a satisfying assignment exists, we can find it by checking for each permutation of the clauses, if there is a set of variables occurring in non-overlapping clause-intervals, which cover all clauses. This check can be performed in polynomial time: Say we are given a permutation of the clauses. We number the clauses in this permutation $1, \ldots, m$ for simplicity. We can then build the following directed graph $G = (V, E)$:

$$\begin{aligned} V &= \{s, t\} \cup \{v_x | \text{variable } x \text{ occurs in a clause-interval}\} \\ E &= \{(s, v_x) | x \text{ occurs in a clause-interval starting with clause } 1\} \cup \\ &\quad \{(v_x, v_y) | \exists i.\ x \text{ occurs in a clause-interval ending with clause } i \text{ and} \\ &\qquad\qquad y \text{ occurs in a clause-interval starting with clause } i+1\} \cup \\ &\quad \{(v_x, t) | x \text{ occurs in a clause-interval ending with clause } m\} \end{aligned}$$

In this permutation of the clauses, there is a set of variables occurring in non-overlapping clause-intervals, which cover all clauses, if and only if there is a path from s to t in G.

The graph can be constructed in time $O(n \cdot L + n^2)$ and the existence of a path from s to t can be determined in time $O(n^2)$ (as the graph is acyclic). As there are $m!$ permutations of the clauses, this algorithm then has time complexity $O((n \cdot L + n^2) \cdot m!)$. $\square$

References

[1] J. M. Byskov. Enumerating maximal independent sets with applications to graph colouring. *Operations Research Letters*, 32:547–556, 2004.

[2] E. Dantsin, E. A. Hirsch, and A. Wolpert. Algorithms for SAT based on search in hamming balls. In *Proceedings of the 21st Annual Symposium on Theoretical Aspects of Computer Science, STACS 2004, volume 2996 of Lecture Notes in Computer Science*, pages 141–151, March 2004.

[3] E. Dantsin and A. Wolpert. Derandomization of Schuler's algorithm for SAT. In *Proceedings of the 7th International Conference on Theory and Applications of Satisfiability Testing, SAT 2004*, pages 69–75, May 2004.

[4] E. A. Hirsch. New worst-case upper bounds for SAT. *Journal of Automated Reasoning*, 24(4):397–420, 2000.

[5] B. A. Madsen, J. M. Nielsen, and B. Skjernaa. New algorithms for exact satisfiability. Report Series RS-03-30, BRICS, Department of Computer Science, Aarhus University, October 2003.

[6] B. Monien, E. Speckenmeyer, and O. Vornberger. Upper bounds for covering problems. *Methods of Operations Research*, 43:419–431, 1981.

[7] T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, pages 216–226, 1978.

[8] B. Skjernaa. *Exact Algorithms for Variants of Satisfiability and Colouring Problems*. PhD thesis, BRICS, Department of Computer Science, Aarhus University, September 2004.

Recent BRICS Report Series Publications

RS-04-18 Bolette Ammitzbøll Madsen. *An Algorithm for Exact Satisfiability Analysed with the Number of Clauses as Parameter*. September 2004. 4 pp.

RS-04-17 Mayer Goldberg. *Computing Logarithms Digit-by-Digit*. September 2004. 6 pp.

RS-04-16 Karl Krukow and Andrew Twigg. *Distributed Approximation of Fixed-Points in Trust Structures*. September 2004. 25 pp.

RS-04-15 Jesús Fernando Almansa. *Full Abstraction of the UC Framework in the Probabilistic Polynomial-time Calculus ppc*. August 2004.

RS-04-14 Jesper Makholm Byskov. *Maker-Maker and Maker-Breaker Games are PSPACE-Complete*. August 2004. 5 pp.

RS-04-13 Jens Groth and Gorm Salomonsen. *Strong Privacy Protection in Electronic Voting*. July 2004. 12 pp. Preliminary abstract presented at Tjoa and Wagner, editors, *13th International Workshop on Database and Expert Systems Applications*, DEXA '02 Proceedings, 2002, page 436.

RS-04-12 Olivier Danvy and Ulrik P. Schultz. *Lambda-Lifting in Quadratic Time*. June 2004. 34 pp. To appear in *Journal of Functional and Logic Programming*. This report supersedes the earlier BRICS report RS-03-36 which was an extended version of a paper appearing in Hu and Rodríguez-Artalejo, editors, *Sixth International Symposium on Functional and Logic Programming*, FLOPS '02 Proceedings, LNCS 2441, 2002, pages 134–151.

RS-04-11 Vladimiro Sassone and Paweł Sobociński. *Congruences for Contextual Graph-Rewriting*. June 2004. 29 pp.

RS-04-10 Daniele Varacca, Hagen Völzer, and Glynn Winskel. *Probabilistic Event Structures and Domains*. June 2004. 41 pp. Extended version of an article to appear in Gardner and Yoshida, editors, *Concurrency Theory: 15th International Conference*, CONCUR '04 Proceedings, LNCS, 2004.

RS-04-9 Ivan B. Damgård, Serge Fehr, and Louis Salvail. *Zero-Knowledge Proofs and String Commitments Withstanding Quantum Attacks*. May 2004. 22 pp.