

NP IS AS EASY AS DETECTING UNIQUE SOLUTIONS

L.G. VALIANT*

Aiken Computation Laboratory, Harvard University, Cambridge, MA 02138, U.S.A.

V.V. VAZIRANI**

Computer Science Department, Cornell University, Ithaca, NY 14853, U.S.A.

Communicated by R.M. Karp

Received August 1984

Revised January 1986

Abstract. For every known NP-complete problem, the number of solutions of its instances varies over a large range, from zero to exponentially many. It is therefore natural to ask if the inherent intractability of NP-complete problems is caused by this wide variation. We give a negative answer to this question using the notion of randomized polynomial time reducibility. We show that the problems of distinguishing between instances of SAT having zero or one solution, or of finding solutions to instances of SAT having a unique solution, are as hard as SAT, under randomized reductions. Several corollaries about the difficulty of specific problems follow. For example, computing the parity of the number of solutions of a SAT formula is shown to be NP-hard, and deciding if a SAT formula has a unique solution is shown to be D^P -hard, under randomized reduction. Central to the study of cryptography is the question as to whether there exist NP-problems whose instances have solutions that are unique but are hard to find. Our result can be interpreted as strengthening the belief that such problems exist.

1. Introduction

Several authors have observed that, among most known NP-complete problems, reductions can be found that preserve the number of solutions [3, 15, 18]. Such reductions are called *parsimonious reductions*. A characteristic of each such NP-complete problem is that its instances have widely varying numbers of solutions. Their number may be zero, one, several, or exponentially many. It is natural to ask whether the inherent difficulty in solving NP-complete problems is caused by this wide variation.

In this paper we give a negative answer to this question, in the following sense. Let A be any NP-complete problem to which satisfiability is parsimoniously reducible. Let $\#A(x)$ denote the number of solutions to instance x . For each Boolean

* Supported by National Science Foundation under Grant MCS-83-02385.

** Supported by the National Science Foundation under Grants MCS-81-21431, BCR-85-03611 and an IBM Faculty Development Award. Present affiliation: AT&T Bell Laboratories, Murray Hill, NJ 07974, U.S.A.

predicate Q , define the problem UA_Q :

$$UA_Q(x) = \begin{cases} 0 & \text{if } \# A(x) = 0, \\ 1 & \text{if } \# A(x) = 1, \\ Q(x) & \text{if } \# A(x) > 1. \end{cases}$$

Our main result is that there is a randomized polynomial time reduction from A to UA_Q that is valid for *any* predicate Q . This shows that if there is a random polynomial-time (i.e., RP) decision procedure that gives the correct answer on instances of A having zero or one solution, and an *arbitrary* answer on the remaining instances, then $NP = RP$.

Equivalently, we prove that the following 'promise problem' [10] is NP-hard, under randomized reductions (see Corollary 3.4):

Input: an instance x of A ,

Output: a solution to x ,

Promise: $\# A(x) = 1$.

A procedure for this problem is required to give the correct answer only if $\# A(x) = 1$. It must, however, halt on each input.

Via our theorem, we can establish for several problems for the first time that they are NP-hard, under randomized polynomial-time reductions. Let A be as defined above. Then $PARITY-A$, which gives the parity of the number of solutions of an instance of A , is NP-hard. The problem of computing the permanent of a $(0, 1)$ -matrix mod k , for any k which is not an exact power of 2, is NP-hard.

One further motivation for studying UA_Q is that every problem in UP is polynomial time reducible to it (for any Q). UP is the class of sets recognized by nondeterministic polynomial-time Turing machines that for all inputs have either zero or one solution. The class appears to be especially relevant in cryptography and has been studied before [8, 11, 12, 14, 19]. Note that it is not known whether, for any Q , $USAT_Q$ is itself a member of UP. Hence, our proof does not show the existence of a hard problem in UP.

As remarked earlier, most of the known polynomial-time reductions among NP-complete problems are parsimonious. For reducing SAT to $USAT_Q$, this is exactly the property that our reducibility should *not* have. Instead, we shall use a randomized reduction. Such reductions have been used in the past for classifying the complexity of certain NP problems which have not yielded to proofs of NP-completeness in the standard sense [1, 2, 22]. Our reduction is of the following kind: We shall say that a problem A is reducible to B by a randomized polynomial-time reduction if there is a randomized (coin flipping) polynomial-time Turing machine T , and a polynomial p , such that:

(1) $\forall x[x \in A \rightarrow T[x] \in B]$. (Even though the output of T , on input $x \in A$, may depend on the coin flips, it will definitely not be in B .)

(2) $\forall x[x \in A \rightarrow T[x] \in B \text{ with probability at least } 1/p(|x|)]$.

Define RP to be the class of sets accepted by randomized Turing machines with one-sided error. In particular, $A \in \text{RP}$ if there is a randomized Turing machine T such that:

- (1) $\forall x[x \notin A \rightarrow T \text{ rejects } x]$.
- (2) $\forall x[x \in A \rightarrow T \text{ accepts } x \text{ with probability at least } \frac{1}{2}]$.

Notice that if, in the above definition, " $\frac{1}{2}$ " is replaced by " $[p(|x|)]^{-1}$ ", for any polynomial p , we still obtain the same class. Clearly, if A is reducible to B by a randomized polynomial-time reduction and if $B \in \text{RP}$, then $A \in \text{RP}$.

In the remainder of this paper we shall take the conjunctive normal form SAT itself as representative of the class of NP-complete problems parsimoniously inter-reducible with it. Hence, our main result can be stated as follows.

Theorem 1.1. *There is a randomized polynomial-time reduction from SAT to USAT_Q that is valid for any predicate Q .*

Corollary 1.2. *If, for some predicate Q , $\text{USAT}_Q \in \text{RP}$, then $\text{NP} = \text{RP}$.*

Our technique can be used to establish certain problems to be complete in the class D^P [13]. In particular, it can be shown that UNIQUE SAT, the problem of determining whether a formula has exactly one solution or not, is complete in D^P under randomized polynomial reduction. This contrasts with the result of Blass and Gurevich [4] that states that completeness here does not hold under any deterministic reduction that relativizes.

2. The proof

We shall show that for any Q there is a randomized polynomial-time reduction from SAT to USAT_Q . The idea of the reduction is the following: given instance f of SAT we will successively conjoin constraints to f to obtain a series of formulae f_1, \dots, f_n that will have decreasing numbers of solutions. We shall prove that if f is satisfiable, then with probability at least $\frac{1}{4}$ one of these formulae will have a unique solution. Hence, if we pick one of these formulae at random it will have a unique solution with probability at least $(4n)^{-1}$. On the other hand, if f is not satisfiable, then each of these formulae will be unsatisfiable, including the randomly chosen one.

Since we do not know what the solutions of f are, we will pick constraints at random from a suitable set. Ideally, we would like to knock out each solution independently with a certain probability. This is not possible with only polynomially many random choices. Surprisingly, the use of GF[2] inner products with polynomially few $\{0, 1\}$ vectors suffices. This use can be viewed as an application of universal hash functions [5]. Such inner products have been used previously in complexity theory by Sipser [16] and Stockmeyer [17].

We shall view truth assignments to the variables x_1, \dots, x_n as n -dimensional $\{0, 1\}$ vectors from the vector space $\text{GF}[2]^n$. The solutions to f form a set of vectors from this space $\{0, 1\}^n$. For $u, v \in \{0, 1\}^n$ we denote by $u \cdot v$ the inner product over $\text{GF}[2]$ of u and v .

First we observe the following:

Lemma 2.1. *If f is any CNF formula in x_1, \dots, x_n and $w_1, \dots, w_k \in \{0, 1\}^n$, then one can construct in linear time a formula f'_k whose solutions v satisfy f and the equations $v \cdot w_1 = \dots = v \cdot w_k = 0$. Furthermore, one can construct a polynomial-size CNF formula f'_k in variables $x_1, \dots, x_n, y_1, \dots, y_m$ for some m such that there is a bijection between solutions of f_k and f'_k , defined by equality on the x_1, \dots, x_n values.*

Proof. It is sufficient to show the lemma for $k = 1$. Then, f'_1 is

$$f \wedge (x_{i_1} \oplus x_{i_2} \oplus \dots \oplus x_{i_j} \oplus 1),$$

where \oplus denotes exclusive-or and i_1, \dots, i_j are the indices of the x_i that have values 1 in w . Also, f_1 is the CNF equivalent of the formula

$$\begin{aligned} f \wedge (y_1 \Leftrightarrow x_{i_1} \oplus x_{i_2}) \wedge (y_2 \Leftrightarrow y_1 \oplus x_{i_3}) \\ \wedge \dots \wedge (y_{j-1} \Leftrightarrow y_{j-2} \oplus x_{i_j}) \wedge (y_{j-1} \oplus 1). \quad \square \end{aligned}$$

The intuition behind our main theorem can be explained as follows. Let S be any subset of $\{0, 1\}^n$. Define the sets

$$S_1 = \{v \mid v \in S, v \cdot w = 0\} \quad \text{and} \quad S'_1 = \{v \mid v \in S, v \cdot w = 1\}.$$

The surprising fact is that if we pick w randomly, then any S will be partitioned in this way into roughly equal halves with high probability. By letting S be the set of solutions of f , choosing w_1, \dots, w_k at random and constructing f'_k we obtain a formula with roughly $2^{-k}|S|$ solutions.

The randomized polynomial-time reduction of Theorem 1.1 from SAT to USAT_Q is simply the following: Given an instance f of SAT, randomly choose an integer k from $\{1, \dots, n\}$, randomly choose vectors $w_1, \dots, w_k \in \{0, 1\}^n$, and output f'_k .

The phenomenon that a random w partitions S into about equal halves is expressed by the following result from [21].

Theorem 2.2. *For all $\varepsilon > 0$ there exists a constant $\alpha > 1$ such that for all $x \geq 1$ if $S \subseteq \{0, 1\}^n$ with $|S| \geq n^{\alpha x}$, then the probability that either $|S_1|$ or $|S'_1|$ is smaller than $(\frac{1}{2} - \varepsilon)|S|$ is less than n^{-x} .*

While this result is sufficient to establish Theorem 1.1, it does not yield the simplest proof. Sharp bounds on the tails of the distribution are not necessary, while bounds applying to all sizes $|S|$ are advantageous. Mark Jerrum observed that the following remarkably precise statement is true and easily proved.

Theorem 2.3. If $S \subseteq \{0, 1\}^n - \{0^n\}$, w_1, \dots, w_n are randomly chosen from $\{0, 1\}^n$ and S_i is defined as

$$\{v \mid v \in S, v \cdot w_1 = v \cdot w_2 = \dots = v \cdot w_i = 0\},$$

then the expectation and variance of S_i are as follows:

$$E(|S_i|) = 2^{-i}|S| \quad \text{and} \quad V(|S_i|) = 2^{-i}(1 - 2^{-i})|S|.$$

Sharper bounds still on the required probabilities can be established by a direct argument due to Michael Rabin. The main idea is to dispense with discussing the size of S_i and do induction on its rank. We shall restrict ourselves to giving this last argument which is encapsulated in Theorems 2.4 and 2.5. The final conclusions are the following.

Theorem 2.4. Let $S \subseteq \{0, 1\}^n$. Suppose w_1, \dots, w_n are chosen at random, for each $i \leq n$, $S_i = \{v \mid v \in S, v \cdot w_1 = \dots = v \cdot w_i = 0\}$ and $P_n(S)$ is the probability that, for some $i \leq n$, $|S_i| = 1$. Then:

- (i) $P_n(S) \geq \frac{1}{4}$;
- (ii) if w_1, \dots, w_n are chosen to be linearly independent in addition, then $P_n(S) \geq \frac{1}{2}$.

Clearly, Theorem 1.1 follows directly from either one of the above two statements. In order to prove them, we first state and prove the following.

Theorem 2.5. Let $S \subseteq \{0, 1\}^n$, $S \neq \emptyset$. Randomly choose vectors w_1, w_2, \dots from $\{0, 1\}^n$. Define $H_i = \{v \mid v \cdot w_i = 0\}$, $S_0 = S$, and $S_i = S \cap H_1 \cap \dots \cap H_i$. Then:

$$P(S) \stackrel{\text{def}}{=} \text{Prob}(\exists i \mid |S_i| = 1) \geq \frac{1}{2}.$$

Proof. Let $T_k = \min\{P(S)\}$, where S ranges over all nonempty sets with $\text{rank}(S) \leq k$. We shall show by induction on k that $T_k \geq 2^{k-1}/(2^k - 1)$.

For $k=0$ we have $S = \{0^n\}$ and $T_0 = 1$. If $k=1$, then $|S| \leq 2$ and $T_1 = 1$. Assume the statement true for all $0 \leq m \leq k-1$, and let $\text{rank}(S) = k$. A nonsingular linear transformation on $\{0, 1\}^n = \text{GF}[2]^n$ permutes the vectors as well as hyperplanes of $\text{GF}[2]^n$. We may therefore assume w.l.o.g. that $\{e_1, \dots, e_k\} \subseteq S$, where e_i is the i th unit vector, and these vectors span S .

A hyperplane $H = \{v \mid v \cdot w = 0\}$ satisfies $\text{rank}(S \cap H) < k$ iff $w \notin 0^k\{0, 1\}^{n-k}$. There are $2^n - 2^{n-k}$ such vectors w . Also, $e_i \notin S \cap H$ for all $1 \leq i \leq k$ iff $w \in 1^k\{0, 1\}^{n-k}$. If $e_i \in S \cap H$ for some i , then certainly $S \cap H \neq \emptyset$. Hence, the probability that $S \cap H \neq \emptyset$, relative to the event that $\text{rank}(S \cap H) < k$, satisfies

$$\text{Prob}(S \cap H \neq \emptyset \mid \text{rank}(S \cap H) < k) \geq 1 - \frac{2^{n-k}}{2^n - 2^{n-k}} = \frac{2^k - 2}{2^k - 1}. \quad (1)$$

When randomly choosing w_1, w_2, \dots , we encounter with probability 1 an index j such that $\text{rank}(S \cap H_1 \cap \dots \cap H_j) < k$. Let i be the smallest such index, thus $S \cap H_1 \cap \dots \cap H_{i-1} = S$, and H_i satisfies $\text{rank}(S \cap H_i) < k$. By (1), the probability that $S \cap H_i \neq \emptyset$ is at least $(2^k - 2)/(2^k - 1)$, thus

$$T_k \geq T_{k-1} \frac{2^k - 2}{2^k - 1} \geq \frac{2^k - 2}{2^{k-1} - 1} \frac{2^k - 2}{2^k - 1} = \frac{2^{k-1}}{2^k - 1}. \quad \square$$

Proof of Theorem 2.4. (i) Clearly, $P_n(S) \geq P(S) \text{Prob}(H_1 \cap \dots \cap H_n = \{0^n\})$. Now, $H_1 \cap \dots \cap H_n = \{0^n\}$ iff the vectors w_1, \dots, w_n are linearly independent. The probability of that is

$$(2^n - 1)(2^n - 2) \dots (2^n - 2^{n-1}) / 2^{n^2} \geq \frac{1}{4}.$$

Hence, $P_n(S) \geq \frac{1}{2} \times \frac{1}{4}$. To obtain the claimed constant of $\frac{1}{4}$ we consider the cases of $0^n \in S$ and $0^n \notin S$ separately. In these two cases,

$$P_n(S) \geq \text{Prob}(H_1 \cap \dots \cap H_n = \{0^n\}) \geq \frac{1}{4} \quad \text{and}$$

$$P_n(S) \geq P(S) \text{Prob}(\text{rank}(H_1 \cap \dots \cap H_{n-1}) = 1) \geq \frac{1}{2} \times \frac{1}{2},$$

respectively.

(ii) Satisfaction of the condition " $\exists i |S_i| = 1$ " by the random process only depends on the indices $i_1 (= 1), i_2, \dots, i_n$ for which strictly

$$H_{i_1} \supset H_{i_1} \cap H_{i_2} \supset \dots \supset H_{i_1} \cap \dots \cap H_{i_n} = \{0^n\}.$$

Therefore, $P(S)$ equals

$$\text{Prob}(\exists i |S_i| = 1 \mid H_1 \supset H_1 \cap H_2 \supset \dots \supset H_1 \cap \dots \cap H_n = \{0^n\}).$$

Also, if $0^n \notin S$, then it suffices to go down to H_{n-1} since now $|S \cap H_1 \cap \dots \cap H_{n-1}| \leq 1$. Hence, by choosing w_1, \dots, w_n to be linearly independent we get a process utilizing just n hyperplanes and achieving the same bound of $\frac{1}{2}$ as $P(S)$. \square

3. Applications

By deterministic polynomial-time reductions from USAT_Q , we can show for several problems for the first time that they are NP-hard under randomized polynomial-time reductions. Let A be any NP-complete problem to which SAT is parsimoniously reducible and define UA_Q as in Section 1.

Corollary 3.1. *For any choice of Q , UA_Q is NP-hard, under randomized reductions.*

Important examples of UA_Q are the following, defined for each positive integer k :

$$\#_k A(x) = \begin{cases} 0 & \text{if } \# A(x) = 0 \pmod k, \\ 1 & \text{if } \# A(x) \neq 0 \pmod k. \end{cases}$$

Corollary 3.2. *For any $k > 1$, $\#_k A$ is NP-hard, under randomized reductions.*

For the following problem, testing for the existence of solutions is in P (by a reduction to finding perfect matchings in bipartite graphs); yet, counting solutions mod k is hard.

Corollary 3.3. *For any $k > 1$ that is not an exact power of two, computing the permanent mod k of a $(0, 1)$ -matrix is NP-hard, under randomized reductions.*

Proof. [20, Lemmas 3.1, 3.2, and 3.3] give a transformation from a SAT formula f to a $(0, 1)$ -matrix B such that $\text{Perm}(B)$ equals $\# \text{SAT}(f) \cdot 4^t \bmod k$ (where t is a polynomial-time computable function of f). Hence, $\# \text{SAT}(f) = 0 \bmod k$ iff $\text{Perm}(B) = 0 \bmod k$. \square

Corollary 3.4. *Consider the problem of finding solutions to instances of SAT with the relaxation that for instances having no solution or more than one solution the output can be arbitrary (the algorithm still has to terminate). This problem is NP-hard under randomized reductions.*

Proof. For any instance of this problem (i.e., with the arbitrariness resolved) there is some Q such that USAT_Q reduces to it trivially. Apply this problem to an instance f of SAT and output 1 or 0 according to whether the output is a solution of f . \square

By self-reducibility, the converse of the above also holds.

The next application, due to Selman, identifies a problem to be complete in the class D^{P} introduced by Papadimitriou and Yannakakis [13],

$$\text{D}^{\text{P}} = \{L_1 - L_2 \mid L_1, L_2 \in \text{NP}\}.$$

The problem UNIQUE SAT is defined as follows: Given an instance of SAT, does it have a unique solution?

Corollary 3.5. *UNIQUE SAT is complete in D^{P} under randomized polynomial-time reductions.*

Proof. In [13] it was shown that the following problem SAT-UNSAT is complete for D^{P} : Given a pair (f_1, f_2) of formulae in CNF, determine whether it is the case that “ f_1 is satisfiable but f_2 is not”. We achieve the required randomized reduction by transforming (f_1, f_2) to $f'_1 \wedge f'_2$ as follows: f'_2 is such that it has exactly one more solution than f_2 has (see [4]). f'_1 is obtained from f_1 exactly as f_k in Theorem 1.1. The variables of f'_1 and f'_2 are made distinct. \square

Our final application involves the problem of APPROXIMATE COUNTING. Sipser and Gacs [16] show that this problem is solvable by a randomized polynomial-time TM with a SAT oracle. Our hash function yields a simpler proof. Once again, this result seems to be the best achievable; Stockmeyer [17] shows an oracle relative to which this problem is not in Δ_2^{P} .

Corollary 3.6. *For any $\varepsilon > 0$ there is a randomized polynomial-time TM with a SAT oracle, which given a SAT formula f outputs a number k such that $(1 - \varepsilon) \cdot \#f \leq k \leq (1 + \varepsilon) \cdot \#f$ with probability at least $1 - \varepsilon$.*

Proof. As in Theorem 1.1, we obtain f_1, \dots, f_n from f . Using the SAT oracle, we determine i such that $\#f_i > 0$ and $\#f_{i+1} = 0$, and output 2^i . The corollary follows from Theorem 2.3 and standard methods for reducing the probability of error in polynomial time. \square

4. Open problems

The obvious open problem is whether there are deterministic reductions from SAT to USAT_Q , for each Q . Blass and Gurevich [4] give an oracle relative to which UNIQUE SAT is not complete in D^P under deterministic reductions. On the other hand, a deterministic reduction from SAT to UNIQUE SAT will make UNIQUE SAT complete in D^P . Therefore, any deterministic reduction that relativizes could not work for reducing SAT to UNIQUE SAT .

We have shown that PARITY SAT is NP-hard under randomized reductions. The problem is clearly in PSPACE; however, we would like to determine where in the hierarchy it lies.

A possible direction for further work is to try to narrow the gap between plausible cryptographic functions and NP-completeness. An ideal cryptographic function would have exactly one solution but it would be NP-hard to find the solution. The following question may be a more tractable first step: Find an NP-complete problem in which for some fixed polynomial $p(n)$ the instances either have no solution, or the number of solutions of an instance of size n is in the range $[s(n), t(n)]$, where $t(n)/s(n) \leq p(n)$. In this regard it is known that not all NP-complete problems are parsimoniously reducible from SAT. Graph edge-coloring is a natural counter-example [7]. Whether the number of solutions of all NP-complete problems is nevertheless polynomial-time interreducible (i.e., whether NP-completeness implies $\#P$ completeness) is also open.

Acknowledgment

We are thankful to Umesh Vazirani for several valuable comments.

References

- [1] L.M. Adleman and K. Manders, Reducibility, randomness and intractability, in: *Proc. 9th ACN Symp. on Theory of Computing* (1977) 151-163.

- [2] L.M. Adleman and K. Manders, Reductions that lie, in: *Proc. 20th IEEE Symp. on Foundations of Computer Science* (1979) 397–410.
- [3] L. Berman and J. Hartmanis, On isomorphisms and density of NP and other complete sets, *SIAM J. Comput.* **6** (1977) 305–322.
- [4] A. Blass and Y. Gurevich, On the unique satisfiability problem, *Inform. and Control* **55** (1982) 80–82.
- [5] J.L. Carter and M.N. Wegman, Universal classes of hash functions, *J. Comput. System Sci.* **18**(2) (1979) 143–154.
- [6] S.A. Cook, The complexity of theorem proving procedures, *Proc. 3rd ACM Symp. on Theory of Computing* (1971) 151–158.
- [7] K.J. Edwards and D.J.A. Welsh, On the complexity of unique problems, Unpublished manuscript, 1984.
- [8] S. Even, A.L. Selman and Y. Yacobi, Hardcore theorems for complexity classes, *J. ACM* **32**(1) (1985) 205–217.
- [9] S. Even, A.L. Selman and Y. Yacobi, The complexity of promise problems with applications to public-key cryptography, *Inform. and Control* **61**(2) (1984) 159–173.
- [10] S. Even and Y. Yacobi, Cryptography and NP-completeness, in: *Proc. 7th Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science **85** (Springer, Berlin, 1980) 195–207.
- [11] J. Geske and J. Grollmann, Relativization of unambiguous and random polynomial time classes, Unpublished manuscript, 1983.
- [12] J. Grollmann and A.L. Selman, On the existence of secure public-key cryptosystems, *Proc. 25th IEEE Symp. on Foundations of Computer Science* (1984) 495–503.
- [13] C.H. Papadimitriou and M. Yannakakis, The complexity of facets (and some facets of complexity), *J. Comput. System Sci.* **28** (1984) 244–259.
- [14] C. Rackoff, Relativized questions involving probabilistic computations, *J. ACM* **29** (1982) 261–268.
- [15] J. Simon, On the difference between one and many, in: *Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science **52** (Springer, Berlin, 1977) 480–491.
- [16] M. Sipser, A complexity theoretic approach to randomness, in: *Proc. 15th ACM Symp. on Theory of Computing* (1983) 330–335.
- [17] L.J. Stockmeyer, The complexity of approximate counting, in: *Proc. 15th ACM Symp. on Theory of Computing* (1983) 118–126.
- [18] L.G. Valiant, A reduction from satisfiability to Hamiltonian circuits that preserves the number of solutions, Unpublished manuscript, Leeds, 1974.
- [19] L.G. Valiant, Relative complexity of checking and evaluating, *Inform. Process. Lett.* **5** (1976) 20–23.
- [20] L.G. Valiant, The complexity of computing the permanent, *Theoret. Comput. Sci.* **8** (1979) 189–201.
- [21] L.G. Valiant and V.V. Vazirani, Unpublished manuscript, 1984.
- [22] U.V. Vazirani and V.V. Vazirani, A natural encoding scheme proved probabilistic polynomial complete, *Theoret. Comput. Sci.* **24** (1983) 291–300.

ERROR: undefined
OFFENDING COMMAND:
STACK: