

Notes On the Number of Satisfying Assignments

W.I. Gasarch

1 Goal

The work in this section is due to Valiant and Vazarani [2]. Our presentation is largely taken from Kobler, Schoning, Toran [1].

Given a Boolean Formula we usually wonder if it has a satisfying assignment. We can look at questions about the NUMBER of satisfying assignments.

Notation 1.1 If ϕ is a Boolean formula then $\#(\phi)$ is the number of satisfying assignments for ϕ .

We will concentrate on the following set:

$$\text{PARITYSAT} = \{\phi : \#(\phi) \equiv 1 \pmod{2}\}.$$

Is PARITYSAT easy? hard?

Our goal is as follows:

If PARITYSAT \in P, then SAT can be solved with a randomized poly time algorithm.

Since we do not think SAT can be solved with a randomized poly time algorithm, we think PARITYSAT \notin P.

2 Randomized Poly Time

Def 2.1 A set A is in R (Randomized Polynomial Time) if there exists a polynomial p , and a polynomial predicate B so that for all n , and for all x with $|x| = n$, we have

$$\begin{aligned} x \in A &\Rightarrow \Pr_{|r|=p(n)}(B(x, r)) \geq 1 - \frac{1}{2^n} \\ x \notin A &\Rightarrow \Pr_{|r|=p(n)}(\neg B(x, r)) = 1 \end{aligned}$$

(Think of r as being a random string.)

Note 2.2 We could have used $\frac{1}{4}$ instead of $\frac{1}{2^n}$ in the above definition. We leave the proof that they are equivalent to the reader.

Note 2.3 We have defined R with 1-sided error. This is because most randomized algorithms that are actually out there have 1-sided error. Classes with 2-sided error have also been defined and we will look at them later in the semester.

Note 2.4 If a problem is in R, then we think of it as feasible to solve in the real world. This is because there are good (though not provably good) random number generators. Hence it is thought that $R \neq NP$. In fact, there are reasons to think that $P = R$.

Example 2.5 Let *DETPOLYZERO* be the set of all square matrices of polynomials in one variable over the integers $M(x)$ such that the $DET(M(x)) \equiv 0$ (that is, for any real a , $DET(M(a)) = 0$).

1.

$$M_1(x) = \begin{pmatrix} x & x-1 \\ x+1 & x^2-1 \end{pmatrix}$$

is NOT in *DETPOLYZERO* since

$$DET(M_1(x)) = x(x^2-1) - (x-1)(x+1) = x^3 - x - (x^2-1) = x^3 - x^2 - x + 1 \not\equiv 0.$$

2.

$$M_2(x) = \begin{pmatrix} 1 & x-1 \\ x+1 & x^2-1 \end{pmatrix}$$

is IN *DETPOLYZERO* since the determinant is

$$DET(M_2(x)) = x^2 - 1 - (x-1)(x+1) = x^2 - 1 - (x^2 - 1) = 0.$$

Here is a randomized algorithm for *DETPOLYZERO*.

1. Input $M(x)$ (an $n \times n$ matrix of polynomials).
2. Pick random primes p_1, \dots, p_n between n^2 and $2n^2$. (They need not be distinct.)

3. For each i , $1 \leq i \leq n$, pick a random $a_i \in \{0, \dots, p_i - 1\}$.
4. For each i , $1 \leq i \leq n$, calculate $d_i = \text{DET}(M(a_i)) \pmod{p_i}$. If for some i , $d_i \neq 0$ then output NO with certainty. If for all i , $d_i = 0$ then output YES (not certain).

If $M(x) \in \text{DETPOLYZERO}$ then for all a, p $M(a) \equiv 0 \pmod{p}$. Hence, for all i , $M(a_i) \equiv 0 \pmod{p_i}$. If $M(x) \notin \text{DETPOLYZERO}$ then it is unlikely that $M(a) \equiv 0 \pmod{p}$ (we omit a formal analysis).

Note 2.6 In the above algorithm, we use “mod p ” so that the intermediate values do not get so large that computing with them is no longer polynomial in n . We pick random numbers so that an adversary cannot contrive a bad input.

We will also need the notion of a randomized reduction.

Def 2.7 Let A and B be two sets. We say that $A \leq_r^p B$ if there exists a function f computable in poly time (the reduction), and polynomials p, q such that

$$\begin{aligned} x \in A &\Rightarrow \Pr_{|r|=p(n)}(f(x, r) \in B) \geq \frac{1}{q(n)} \\ x \notin A &\Rightarrow \Pr_{|r|=p(n)}(f(x, r) \notin B) = 1 \end{aligned}$$

(Think of r as being a random string.)

This does not look that useful since the probability of being right when $x \in A$ is small. But its usefulness emerges from the following theorem.

Theorem 2.8 *If $A \leq_r^p B$ and $B \in \text{P}$ then $A \in \text{R}$.*

Proof:

Assume $A \leq_r^p B$ via the function f and polynomials p, q .

1. Input (x, r) . Let n be such that $|x| = n$ and let $|r| = p(n)q(n)$. (We denote $r = r_1 r_2 \dots r_{q(n)}$ where, for each i , $|r_i| = p(n)$.)
2. Compute $f(x, r_1), \dots, f(x, r_{q(n)})$. For each i query $f(x, r_i) \in B$. If any of the answers are YES, then output YES. Otherwise output NO.

If $x \notin A$, then for all i $f(x, r_i) \notin B$ hence the algorithm will (correctly) say NO.

If $x \in A$, then for each i , $\Pr(f(x, r_i) \notin B) \leq (1 - \frac{1}{q(n)})$. Hence

$$\begin{aligned} [\Pr((\forall i)[f(x, r_i) \notin B])] &\leq (1 - \frac{1}{q(n)})^{nq(n)} \\ &\leq (e^{-1/q(n)})^{nq(n)} \\ &\leq (e^{-1})^n \\ &\leq e^{-n} \\ &\leq 2^{-n}. \end{aligned}$$

Hence

$$\Pr((\exists i)[f(x, r_i) \in B]) \geq 1 - \frac{1}{2^n}.$$

This establishes the desired result.

■

3 If PARITYSAT \in P Then NP = R

We will use the notions of hash functions and random linear functions discussed earlier.

Def 3.1 Let $k \in \mathbb{N}$. Then SAT_k is

$$\{\phi : 1 \leq \#(\phi) \leq k\}.$$

We will first show $\text{SAT} \leq_r^p \text{SAT}_{12}$ and then use this in our reduction $\text{SAT} \leq_r^p \text{PARITYSAT}$.

Lemma 3.2 $\text{SAT} \leq_r^p \text{SAT}_{12}$.

Proof:

Here is the random reduction.

1. Input $\phi(\vec{x})$. Let n be the number of variables in ϕ .
2. Pick a random $k \in \{0, \dots, n-1\}$ (uniformly).
3. Pick a random $h \in \{0, 1\}^{kn}$ which we view as a $k \times n$ matrix, and hence as a randomized linear hash function which we also denote h .

4. Form a Boolean formula $\psi(\vec{x})$ which is true on \vec{x} iff $h(\vec{x}) = 0^k$.
5. Output $\phi'(\vec{x}) = \phi(\vec{x}) \wedge \psi(\vec{x})$.

Clearly if $\phi \notin \text{SAT}$ then $\phi' \notin \text{SAT}_{12}$.

Assume $\phi \in \text{SAT}$. We show that the $\Pr(\phi' \in \text{SAT}_{12}) \geq \frac{1}{2n}$. There are two cases.

Case 1: $\#(\phi) \leq 12$. If $k = 0$ then $\phi = \phi' \in \text{SAT}_{12}$. The probability that $k = 0$ is $\frac{1}{n} \geq \frac{1}{2n}$.

Case 2: $\#(\phi) \geq 13$. Let m be such that $2^m < \#(\phi) \leq 2^{m+1}$. We look at what happens if $k = m - 2$. Let X be the set of satisfying assignments of ϕ . We have $2^m < |X| \leq 2^{m+1}$. We are picking a random hash function $h : \{0, 1\}^n \rightarrow \{0, 1\}^k$. Let $S = |\{x \in \{0, 1\}^n : h(x) = 0^k\}|$. By previous lemmas we know that $E(S) = 2^{-k}|X| = 2^{-m-2}|X|$ and $\text{Var}(S) \leq 2^{-m-2}|X|$. Hence $2^{-(m-2)+m} < E(S) \leq 2^{-(m-2)+m+1}$, so $4 < E(S) \leq 8$; and $\text{Var}(S) < 8$. We want $\Pr(S \in \{1, \dots, 12\})$. This is $1 - \Pr(S \notin \{1, \dots, 12\})$. We leave it to the reader to show that

$$\Pr(S \notin \{1, \dots, 12\}) \leq \Pr(|S - E(S)| \geq 4).$$

By Chebyshev's inequality

$$\Pr(|S - E(S)| \geq 4) \leq \frac{\text{Var}(S)}{4^2} \leq \frac{8}{16} = \frac{1}{2}.$$

So $\Pr(S \in \{1, \dots, 12\}) > 1 - \frac{1}{2} = \frac{1}{2}$. Hence, given that $k = m - 2$ we have $\Pr(\phi' \in \{1, \dots, 12\}) \geq \frac{1}{2}$. The probability that $k = m - 2$ is $\frac{1}{n}$. Hence $\Pr(\phi' \in \text{SAT}_{12}) \geq \frac{1}{2n}$. ■

Theorem 3.3 $\text{SAT} \leq_r^p \text{PARITYSAT}$.

Proof:

In this theorem we use capital letters for vectors of variables. The Boolean formula $LESS(X, Y)$ will mean that X is less than Y in lexicographic order.

1. Input(ϕ).
2. Apply the transformation from Lemma 3.2 to get a Boolean formula ϕ' .

3. Pick a random $m \in \{1, \dots, 12\}$.
4. Output $\psi(X_1, \dots, X_m)$
 $= \phi'(X_1) \wedge \phi'(X_2) \wedge \dots \wedge \phi'(X_m)$
 \wedge
 $LESS(X_1, X_2) \wedge LESS(X_2, X_3) \wedge \dots \wedge LESS(X_{m-1}, X_m)$

If $\phi \notin \text{SAT}$ then $\phi' \notin \text{SAT}$ and therefore $\psi \notin \text{SAT}$.

We have shown that if $\phi \in \text{SAT}$, then with probability $\frac{1}{2n}$, $\#(\phi') \in \{1, \dots, 12\}$.

There are 12 cases, but we can make them all into one case.

Assume $\#(\phi') = i \in \{1, \dots, 12\}$. If $m = i$ then ϕ' has m different satisfying assignments which we order lexicographically as $B_1 \preceq \dots \preceq B_m$. Note that $\psi(B_1, \dots, B_m)$ is true, and is the only satisfying assignment for ψ . Hence $\#(\psi) = 1 \equiv 1 \pmod{2}$. Hence if $m = i$ then $\psi \in \text{PARITYSAT}$. The probability that $m = i$ is $\frac{1}{12}$. The probability that $\phi' \in \text{SAT}_{12}$ is $\geq \frac{1}{2n}$. Hence the probability that $\psi \in \text{PARITYSAT}$ is $\geq \frac{1}{24n}$. ■

References

- [1] J. Köbler, U. Schöning, and J. Torán. *The Graph Isomorphism Problem: Its Structural Complexity*. Progress in Theoretical Computer Science. Birkhauser, Boston, 1993.
- [2] L. G. Valiant and V. V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Comput. Sci.*, 47:85–93, 1986. Earlier version in STOC85.