

Notes for CMSC 652
On the Number of Satisfying Assignments
William Gasarch

1 Goal

The work in this section is due to Valiant and Vazirani [3]. Our presentation is largely taken from Kobler, Schoning, Toran [1].

Given a Boolean Formula we usually wonder if it has a satisfying assignment. We 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

$$\begin{aligned} x \in A &\rightarrow \Pr_{|r|=p(n)}(B(x, r)) \geq \frac{3}{4} \\ x \notin A &\rightarrow \Pr_{|r|=p(n)}(\neg B(x, r)) = 1 \end{aligned}$$

(Think of r as being a random string.)

Exercise 1

1. Show that if $\frac{3}{4}$ is replaced by $\frac{99}{100}$ then the resulting class is still R.
2. Show that if $\frac{3}{4}$ is replaced by $1 - \frac{1}{2^{|x|}}$ then the resulting class is still R.

Note 2.2 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.

Note 2.3 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. There are also theoretical reasons to think that $P = R$ (see [2]).

Example 2.4 Let DETPOLYZERO be the set of all square matrices $M(x)$ of polynomials in one variable over the integers such that the $DET(M(x)) \equiv 0$ (that is, for any real a , $DET(M(a)) = 0$).

1. The matrix

$$\begin{matrix} x & x - 1 \\ x + 1 & x^2 - 1 \end{matrix}$$

is NOT in DETPOLYZERO since

$$DET(M_1(x)) = x(x^2 - 1) - (x - 1)(x + 1) = x^3 - x - (x^2 - 1) = x^3 - x^2 - x + 1 \neq 0.$$

2. The matrix

$$\begin{matrix} 1 & x - 1 \\ x + 1 & x^2 - 1 \end{matrix}$$

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 2.5 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.

Exercise 2

1. Show that PRIMES is in R. (You may not use that PRIMES is known to be in P. You may use the Web or any other non-organic source; however, you must hand in a reasonably complete proof.)
2. Find a case in the literature where using randomization seems to improve the performance of something.

We will need the notion of a randomized reduction.

Def 2.6 Let A and B be two sets. We say that $A \leq_r B$ 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 2.7 *If $A \leq_r B$ and $B \in P$ then $A \in R$.*

Proof:

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

1. Input (x, r) . Let n be such that $|x| = n$ and let $|r| = 2p(n)q(n)$. (We denote $r = r_1 r_2 \cdots r_{2q(n)}$ where, for each i , $|r_i| = p(n)$.)
2. Compute $f(x, r_1), \dots, f(x, r_{2q(n)})$. 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

$$\begin{aligned} [\Pr((\forall i)[f(x, r_i) \notin B])] &\leq (1 - \frac{1}{q(n)})^{2q(n)} \\ &\leq (e^{-1/q(n)})^{2q(n)} \\ &\leq (e^{-1})^2 \\ &\leq \frac{1}{e^2} \\ &\leq \frac{1}{4}. \end{aligned}$$

Hence

$$\Pr((\exists i)[f(x, r_i) \in B]) \geq 1 - \frac{1}{4} = \frac{3}{4}.$$

This establishes the desired result.

■

Exercise 3 Show that if $A \leq_r B$ and $B \in R$ then $A \in R$.

3 Our Plan

Given a formula ϕ we want to produce a formula ϕ' such that

$$\begin{aligned} \phi \in SAT &\rightarrow \#(\phi') = 1 \text{ with high probability;} \\ \phi \notin SAT &\rightarrow \#(\phi') = 0. \end{aligned}$$

We view a formula as a set of satisfying assignments. Hence we want to map this set to a much smaller set. How do computer scientists map large sets to small sets? By using Hash Functions! The next section has all we will need.

4 Hash Functions

If a set is ‘large’ then a randomly chosen hash function (which we assume is fairly uniform) is quite likely to map some element to some fixed element. If a set is ‘small’ then this is unlikely. In particular, if a random string r is interpreted as a random hash function and A is large, then it is quite likely that some element of A maps to 0^k .

Def 4.1

1. A *sample space* is the set of things that could happen. In our case it will be the set of possibly hash functions that could be produced.
2. A *random variable* is a mapping from the sample space to numbers. In our case it will be mapping the hash function h to the number $|\{x \mid h(x) = 0^k\}|$.
3. If S is a random variable then $E(S)$ is its expected value and $Var(S)$ is its variance. It is known that $Var(S) = E((S - E(S))^2) = E(S^2) - E(S)^2$.

Important convention: Whenever we have a 0-1 valued matrix apply to a vector we do all of the calculations mod 2.

Lemma 4.2 *Let $k, n \in \mathbb{N}$. Let $X \subseteq \{0, 1\}^n$. Assume $0^n \notin X$. Consider the following random variable: Pick a random $k \times n$ 0-1 valued matrix M .*

$$S = |\{x \in X : M(x) = 0^k\}|.$$

Output S . Then $E(S) = 2^{-k}|X|$ and $Var(S) \leq 2^{-k}|X|$. (Note that neither $E(S)$ nor $Var(S)$ depends on n , just on k and $|X|$.)

Proof: Before looking at $E(S)$ and $Var(S)$ we will need to look at E of some easier random variables

Let $x, y \in X$. Let R_x be the random variable

$$R_x = \begin{cases} 1 & \text{if } M(x) = 0^k \\ 0 & \text{if } M(x) \neq 0^k. \end{cases} \quad (1)$$

Let R_y be similar.

Let $M_i(x)$ be the i th element of the vector $M(x)$.

$$\begin{aligned} E(R_x) &= \Pr(M(x) = 0^k) \cdot 1 + \Pr(M(x) \neq 0^k) \cdot 0; \\ E(R_x) &= \Pr(M(x) = 0^k); \\ E(R_x) &= \prod_{i=1}^k \Pr(M_i(x) = 0). \end{aligned}$$

Recall that x is fixed and that $x \neq 0^n$. The probability that $h_i(x) = 0$ can be phrased as follows: What is the probability that a randomly chosen y will make $x \cdot y \equiv 0 \pmod{2}$? We leave it as an easy exercise that this is $\frac{1}{2}$. Hence

$$E(R_x) = \prod_{i=1}^k \Pr(M_i(x) = 0) = \frac{1}{2^k}.$$

The exact same calculation shows that

$$E(R_x^2) = \frac{1}{2^k}. \text{ (For any 0-1 valued random variable } Z, E(Z) = E(Z^2).)$$

We now compute $E(R_x R_y)$.

$$\begin{aligned} E(R_x R_y) &= \Pr(M(x) = 1 \wedge M(y) = 1) \cdot 1 + \Pr(M(x) = 0)\Pr(M(y) = 1) \cdot 0 + \\ &\quad \Pr(M(x) = 1)\Pr(M(x) = 0) \cdot 0 + \Pr(M(x) = 0)\Pr(M(x) = 0) \cdot 0 \\ &= \Pr(M(x) = 1)\Pr(M(y) = 1) \\ &= \frac{1}{2^k} \frac{1}{2^k} = \frac{1}{4^k} \end{aligned}$$

We are now ready to tackle $E(S)$ and $Var(S)$. Note that $S = \sum_{x \in X} R_x$.

$$E(S) = E\left(\sum_{x \in X} R_x\right) = \sum_{x \in X} E(R_x) = \frac{1}{2^k} |X|.$$

We now look at $Var(S)$. Recall that $Var(S) = E(S^2) - (E(S))^2$.

$$\begin{aligned} E(S^2) &= E\left(\left(\sum_{x \in X} R_x\right)\left(\sum_{y \in X} R_y\right)\right); \\ &= \sum_{x \in X} \sum_{y \in X} E(R_x R_y); \\ &= \sum_{x \in X} E(R_x^2) + \sum_{x \neq y} E(R_x R_y); \\ &= \sum_{x \in X} \frac{1}{2^k} + \sum_{x \neq y} \frac{1}{4^k}; \\ &= \frac{1}{2^k} |X| + \frac{1}{4^k} |X|(|X| - 1); \\ Var(S) &= E(S^2) - (E(S))^2 \\ &= \frac{1}{2^k} |X| + \frac{1}{4^k} |X|(|X| - 1) - \frac{1}{4^k} |X|^2 \\ &= \frac{1}{2^k} |X| + \frac{1}{4^k} |X|^2 - \frac{1}{4^k} |X| - \frac{1}{4^k} |X|^2 \\ &= \frac{1}{2^k} |X| - \frac{1}{4^k} |X| \\ &\leq \frac{1}{2^k} |X| \end{aligned}$$

■

5 If PARITYSAT \in P Then NP = R

Def 5.1 Let $\ell \in \mathbb{N}$. Then SAT_ℓ is

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

We will first show $\text{SAT} \leq_r \text{SAT}_{12}$ and then use this in our reduction $\text{SAT} \leq_r \text{PARITYSAT}$. The reason we use SAT_{12} (as opposed to SAT_{17} or something else) will become evident later.

We use the following lemma which is Chebyshev's inequality. We do not present a proof.

Lemma 5.2 *If S is any random variable and $a > 0$ then*

$$\Pr(|S - E(S)| \geq a) < \frac{\text{Var}(S)}{a^2}.$$

Intuitively this is saying that the probability that S is far away from $E(S)$ is small, and how small depends on $\text{Var}(S)$.

Lemma 5.3 $\text{SAT} \leq_r \text{SAT}_{12}$.

Proof:

Here is the randomized reduction.

1. Input $\phi(\vec{x})$. Let n be the number of variables in ϕ .
2. Evaluate $\phi(\vec{0})$. If equals TRUE then output x (which is a formula in SAT_{12} and stop. If FALSE then goto next step. Note that if X is the set of satisfying assignments for ϕ then $0^n \notin X$.
3. Pick a random $k \in \{0, \dots, n-1\}$ (uniformly).
4. Pick a random $k \times n$ 0-1 valued matrix M .
5. Output the Boolean formula $\psi(\vec{x}) = \phi(x) \wedge M(x) = 0^k$. (This can easily be written as a Boolean formula of size poly in n .)

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

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

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

Case 2: $\#(\phi) \geq 13$. Let m be such that $2^m < \#(\phi) \leq 2^{m+1}$. (Note that $m \in \{3, \dots, n-1\}$.) We look at what happens if $k = m-2$. (Note that $m-2 \in \{1, \dots, n-3\}$.) Let X be the set of satisfying assignments of ϕ . Recall that $0^n \notin X$. 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

$$S = |\{x \in X : h(x) = 0^k\}|.$$

By Lemma 4.2 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 $1 - \Pr(S \notin \{1, \dots, 12\})$.

So what is $\Pr(S \notin \{1, \dots, 12\})$?

Note that

$$S \notin \{1, \dots, 12\} \rightarrow (S = 0) \vee (S \geq 13)$$

$$\text{Since } E(S) \geq 4, S = 0 \rightarrow |S - E(S)| \geq 4.$$

$$\text{Since } E(S) \leq 8, S \geq 13 \rightarrow |S - E(S)| \geq 5.$$

Hence

$$S \notin \{1, \dots, 12\} \rightarrow (S = 0) \vee (S \geq 13) \rightarrow |S - E(S)| \geq 4.$$

Hence

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

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 $\Pr(\psi \in \{1, \dots, 12\}) \geq \frac{1}{2}$. The probability that $k = m - 2$ is $\frac{1}{n}$. Hence $\Pr(\psi \in \text{SAT}_{12}) \geq \frac{1}{2n}$. ■

Theorem 5.4 $\text{SAT} \leq_r \text{PARITYSAT}$.

Proof:

In this theorem we use capital letters for vectors of variables. $X_1 < \dots < X_m$ is the Boolean formula that is true iff, in lexicographic order, $X_1 < \dots < X_m$.

RANDOMIZED REDUCTION

1. Input(ϕ).
2. Apply the transformation from Lemma 5.3 to get a Boolean formula ψ . Note that

$$\begin{aligned} \phi \in \text{SAT} &\rightarrow \psi \in \text{SAT}_{12} \text{ with probability } \geq \frac{1}{2n}; \\ \phi \notin \text{SAT} &\rightarrow \psi \notin \text{SAT}_{12}. \end{aligned}$$

3. Pick a random $m \in \{1, \dots, 12\}$.
4. Output

$$\phi' = \psi(X_1) \wedge \psi(X_2) \wedge \dots \wedge \psi(X_m) \wedge (X_1 < \dots < X_m).$$

END OF RANDOMIZED REDUCTION

If $\phi \notin \text{SAT}$ then $\psi \notin \text{SAT}$ and therefore $\phi' \notin \text{SAT}$. Hence $\#(\phi') = 0 \equiv 0 \pmod{2}$, so $\phi' \notin \text{PARITYSAT}$.

Assume $\phi \in \text{SAT}$. Then then with probability $\frac{1}{2n}$, $\#(\psi) \in \{1, \dots, 12\}$.

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

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

Corollary 5.5

1. If $\text{PARITYSAT} \in \text{P}$ then $\text{NP} = \text{R}$.
2. If $\text{PARITYSAT} \in \text{R}$ then $\text{NP} = \text{R}$.

Proof: We obtain a by combining Theorem 2.7 and 5.4. We obtain b by combining Exercise 3 and 5.4. ■

6 Promise Problems

The reduction from Theorem 5.4, $\text{SAT} \leq_r \text{PARITYSAT}$, actually showed $\text{SAT} \leq_r \text{SAT}_1$. This result is not that interesting because of the following easy exercise.

Exercise 4 Show that if $\text{SAT}_1 \in \text{P}$ then $\text{SAT} \in \text{P}$.

Let us look at the reduction $\text{SAT} \leq_r \text{SAT}_1$ more carefully.

We actually have

$$\phi \in \text{SAT} \rightarrow \phi' \in \text{SAT}_1 \text{ probability } \geq \frac{1}{2^n}$$

$$\phi \notin \text{SAT} \rightarrow \phi' \notin \text{SAT}.$$

So if ϕ' is the output then, with probability $\geq \frac{1}{2^n}$, $\#(\phi') \in \{0, 1\}$.

This motivates the following definition.

Def 6.1 A *promise problem* is a pair (A, PR) where both A and PR are subsets of Σ^* . PR is referred to as *the promise*. The promise problem (A, PR) is in P if there exists a poly time algorithm ALG such that

$$PR(x) = 1 \rightarrow ALG(x) = A(x).$$

Note that we do not care what $ALG(x)$ does if $PR(x) = 0$.

By using Theorem 5.4 and the fact that the output ϕ' of the reduction satisfies (with high probability) the promise $\#(\phi) \in \{0, 1\}$, we can obtain the following.

Theorem 6.2 Let $PR(\phi)$ be the promise that $\#(\phi) \in \{0, 1\}$. If the promise problem (SAT_1, PR) can be solved in P then $\text{NP} = \text{R}$.

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] N. Nisan and A. Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994. Prior version in FOCS88. Full Version at <http://www.math.ias.edu/~avi/PUBLICATIONS/>.
- [3] L. G. Valiant and V. V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986. Earlier version in STOC85.