BORING BOOLEAN CIRCUIT

THINH D. NGUYEN

Abstract. We survey the problem of deciding whether a given Boolean circuit is boring.

1. Definition and NP-hardness

Definition 1. (Boring Circuit problem): We say that a Boolean circuit is boring if it returns the same result for $> \frac{3}{4}$ fractions of possible inputs, where we have n input gates. Hence, boring circuit returns the same output (0 or 1) for $> \frac{3}{4}2^n$ inputs. The decision problem asks if a Boolean circuit is boring.

Claim 2. *We have that* 3-CNF-Sat $\leq_p$ Boring Circuit

Proof. Given a Boolean formula φ, let x, y be two fresh variables, and consider the circuit computing the function

$$x \vee y \vee \varphi.$$

This circuit is boring iff φ is satisfiable. □

2. Proof of PP-hardness

Claim 3. *We have that* MajSat $\leq_p$ Boring Circuit

Proof. The problem is PP-hard. This means that unless the polynomial hierarchy collapses to NP, then deciding whether a circuit is boring is not in NP (and consequently is not NP-complete). The collapse follows from the fact that PP is closed under complement, so $\mathsf{PP} = \mathsf{NP}$ implies $\mathsf{NP} = \mathsf{coNP}$. We now show that deciding whether or not a circuit is boring is PP-hard.

Suppose $L \in PP$, then there exists a probabilistic polynomial Turing machine M such that $x \in L \iff \Pr[M(x) = L(x)] > \frac{2}{3}$. Now consider the machine M' which tosses two coins, if both output "heads" accept, otherwise execute M. If $x \in L$ then M' accepts x with probability $\frac{1}{4} + \frac{3}{4}\Pr[M(x) = L(x)] > \frac{3}{4}$. Now, let C_x be the corresponding circuit for M' on input x (its inputs are the coins for M'), then $x \mapsto C_x$ is a reduction from L to our problem, since $x \in L \iff C_x$ is boring. To see why, note that if $x \in L$ then C_x accepts more than $\frac{3}{4}$ of the possible inputs, and if $x \notin L$ then C_x accepts $\frac{1}{4} \leq t \leq \frac{3}{4}$ of the possible inputs.

In fact, our problem is PP-complete. To see why this language lies in PP, Let L be the language of circuits who accept more than a $\frac{3}{4}$ fraction of their input, and L' be the language of circuits who reject more than a $\frac{3}{4}$ fraction of their inputs. Clearly $L, L' \in PP$, and since PP is closed under union, deciding whether a circuit is boring, i.e is in $L \cup L'$, is in PP. Note that both here and in the above we used

Key words and phrases. boring, circuit, hard, probability.

Perebor.

the fact that the constant in the definition of PP can be changed to any rational number.

If we want to avoid using closure properties of PP, then an argument along the following lines should work. Consider the machine M, which upon given an n-input circuit C, approximates the acceptance probability of C by evaluating it on different uniformly distributed random length n strings. Let $\hat{p}$ denote our evaluation. If $\hat{p} \geq \frac{1}{2}$, then M evaluates C on a random string and accept iff C accepts, otherwise M accepts iff C rejects. Let E denote the event that $|p - \hat{p}| \leq \frac{1}{4}$, where p is the real acceptance probability of C. By evaluating C on polynomially many strings, we can make sure that $\Pr[E] \geq 1 - \frac{1}{2^{2n}}$. If C is boring, then conditioned on E, M obtained the right evaluation of the majority of C and accepts with probability $> \frac{3}{4}$. Thus, if C is boring M accepts with probability $> \frac{3}{4}\left(1 - \frac{1}{2^{2n}}\right)$. If C isn't boring, then C outputs either 0 or 1 with probability of at most $\frac{3}{4} - \frac{1}{2^n}$. Thus, regardless of our evaluation $\hat{p}$, M accepts C with probability of at most $\frac{3}{4} - \frac{1}{2^n} < \frac{3}{4}\left(1 - \frac{1}{2^{2n}}\right)$, which puts our language in PP. □

3. Conclusion

As long as we study a mathematical conjecture, we should encourage ourselves of having enough labouring hours on Prasolov and Sharygin maths books. Then, reading some articles on theory of computing like [2] is a good practice. Only after that, could we think of the ultimate final for all mathematics sciences.

References

1. Michael R. Garey, David S. Johnson, *Computers and Intractability: A Guide to the Theory of* **NP**-*Completeness*
2. Phan Dinh Dieu, Le Cong Thanh, Le Tuan Hoa, *Average Polyno-mial Time Complexity of Some NP-Complete Problems*, Theor. Comput. Sci. 46(3): 219-237 (1986)

Current address: Department of Mathematics, Moscow State University
Email address: kosmofarmer@yandex.com