NP Versus *PP* *

ZHAO Yun-lei, ZHU Hong, ZHAO Yi-ming

(*Department of Computer Science, Fudan University, Shanghai* 200433, *China*)

E-mail: 990314@fudan.edu.cn; hzhu@fudan.edu.cn

http://www.fudan.edu.cn

Received October 23, 2000; accepted January 18, 2001

Abstract: In this paper, the authors mainly intend to clarify the relation between NP and PP. A randomized version of NP is given. Based on this equivalent definition of NP, another randomized complexity class is given: $SUPER\text{-}NP$. Although the $SUPER\text{-}NP$ is very close to NP, but it is found surprisingly that $PP \subseteq SUPER\text{-}NP$ and thus $NP \subseteq PP \subseteq SUPER\text{-}NP$. In light of $NP = PCP(\log, O(1))$ and the closeness of NP and $SUPER\text{-}NP$ it is hoped that $PP \subseteq PCP(\log^2, O(1))$ conjecture can be proved by showing that $SUPER\text{-}NP \subseteq PCP(\log^2, O(1))$.

Key words: NP; PP; PCP; randomized computation; complexity theory

1 Introduction

Definition 1.1. [1,2] $NP = \{L$: There exists a polynomial time non-deterministic Turing machine M, such that $L_M = L$, where $L_M = \{x \in \Sigma^* : M(x) = 1\}\}$.

Definition 1.2. [3] $NP_1 = \{L$: There exists a polynomially-bounded relation $R_L \subseteq \{0,1\}^* \times \{0,1\}^*$, such that R_L is polynomial-time decidable and $x \in L$ if and only if there exists a witness w, for which $(x, w) \in R_L$.$\}$

Proposition 1.1. [3] $NP = NP_1$.

Definition 1.3. [3~5] $PP = \{L \subseteq \{0,1\}^*$ ∶ There exists a probabilistic polynomial time Turing machine M s.t. $\forall x$, $\text{Prob}[M(x) = \chi_L(x)] > \frac{1}{2}\}$.

Here

$$\chi_L(x) = \begin{cases} 1 & x \in L \\ 0 & x \notin L \end{cases}$$

Theorem 1.1. [3] $NP \subseteq PP$.

Definition 1.4. [3,6,7] (Probabilistically Checkable Proofs-PCP) A Probabilistically Checkable Proof system for a language L is a probabilistic polynomial-time oracle machine (called verifier), denoted as M, satisfying:

- Completeness: For every $x \in L$ there exists an oracle π_x such that: $\text{Prob}[M^{\pi_x}(x) = 1] = 1$
- Soundness: For every $x \notin L$ and every oracle π:

$$\text{Prob}[M^{\pi}(x) = 1] \leqslant \frac{1}{2}$$

* Supported by the National Natural Science Foundation of China under Grant No. 69973013 (国家自然科学基金)

ZHAO Yun-lei was born in 1974. He is a Ph. D. candidate at the Department of Computer Science, Fudan University. His main research interests are computational complexity theory and foundations of cryptography. **ZHU Hong** was born in 1939. He is a doctoral supervisor at the Department of Computer Science, Fudan University. His main research interests are algorithm and computational complexity theory and foundations of cryptography. **ZHAO Yi-ming** was born in 1961. He is an associate professor at the Department of Computer Science, Fudan University. His main research interests are computer security and cryptography.

where the probability is taken over M's internal coin tosses.

Definition 1.5.[3,6,7] (Complexity measures for PCP) Let r,q: $N \to N$ be integer functions. The complexity class $PCP(r(\ ),q(\ ))$ consists of languages having a probabilistically checkable proof system in which it holds that:

- Randomness Complexity: On input $x \in \{0,1\}^*$, the verifier makes at most $r(|x|)$ coin tosses.
- Query Complexity: On input $x \in \{0,1\}^*$, the verifier makes at most $q(|x|)$ queries.

For sets of integer functions R and Q, we let

$$PCP(R,Q) = \bigcup_{r \in R, q \in Q} PCP(r(\ ), q(\ )).$$

Theorem 1.2.[7] (The PCP Characterization of NP) $NP = PCP(\log, O(1))$.

2 Alternative Definition for *NP* (Randomized Version)

Definition 2.1. NP_2: The complexity class NP_2 is the class of all languages L for which there exist a probabilistic polynomial-time (bounded by a polynomial $p_1(\ )$) Turing machine ($PPTM$) M and a positive polynomial $p()$, such that

$$x \in L \Rightarrow \mathrm{Prob}[M(x)=1] \geq 2^{-p(|x|)}$$
$$x \notin L \Rightarrow \mathrm{Prob}[M(x)=0] = 1.$$

Theorem 2.1. $NP_2 = NP$.

Proof. $NP \subseteq NP_2$.

Suppose that $L \in NP$ is decided by a nondeterministic machine M with a running-time that is bounded by a polynomial $p(|x|)$. The following machine M' then will decide L by means of Definition 2.1:

$$M'(x,(b_1,b_2,\ldots,b_{p(|x|)})) = M(x,(b_1,b_2,\ldots,b_{p(|x|)}))$$

That is M' uses its random coin-tosses as a witness to M. So, combined with Proposition 1.1, then $L \in NP_2$.

$$NP_2 \subseteq NP.$$

For each $L \in NP_2$ is decided by a probabilistic polynomial-time Turing machine ($PPTM$) M (according to Definition 2.1) with a running-time that is bounded by a polynomial $p_1(|x|)$. Without loss of generality, we assume that for each $x \in \{0,1\}^*$ all computations of M use the same length ($p_1(|x|)$) of coin-toss (or 'guess') and that all those computations constitute a binary tree (that is there are just $2^{p_1(|x|)}$ possible coin-tosses (computation paths) for each $x \in \{0,1\}^*$).

We distinguish two cases according to whether $p_1(|x|) \geq p(|x|)$ or not.

First, if $p_1(|x|) \geq p(|x|)$, when $x \in L$ then there exists at least one coin-toss (computation path) which leads to $M(x)=1$. We use the coin-tosses as the witness for $x \in L$. Combined with Proposition 1.1 we get $L \in NP$.

Second, if $p_1(|x|) < p(|x|)$ then we construct another $PPTM$ M' using M as follows:

$$M'(x,(a_1,a_2,\ldots,a_{p(|x|)-p_1(|x|)+1},b_1,b_2,\ldots,b_{p_1(|x|)})) = M(x,(b_1,b_2,\ldots,b_{p_1(|x|)}))$$

That is no matter what $a_1,a_2,\ldots,a_{p(|x|)-p_1(|x|)-1}$ would be, M' just return $M(x,(b_1,b_2,\ldots,b_{p_1(|x|)}))$.

Note that indeed $\mathrm{Prob}[M'(x)=\chi_L(x)] = \mathrm{Prob}[M(x)=\chi_L(x)]$.

Denote $p_2(|x|)$ as $p(|x|)+1$, then $p_2(|x|) > p(|x|)$. According to the arguments in the first case we get that $L \in NP$.

As a conclusion, in both cases above we get that $L \in NP$, thus $NP_2 \subseteq NP$ and the theorem does hold. □

3 The *SUPER-NP* Class

Definition 3.1. ($SUPER\text{-}NP$). The complexity class $SUPER\text{-}NP$ is the class of all languages L for which there exist a probabilistic polynomial-time (bounded by a polynomial $p_1(|x|)$) Turing machine (*PPTM*) M and a positive polynomial $p()$, such that

$$x\in L\Rightarrow \mathrm{Prob}[M(x)=1]>2^{-p(|x|)}$$
$$x\notin L\Rightarrow \mathrm{Prob}[M(x)=0]>1-2^{-p(|x|)}$$

Note that in contrast to Definition 2.1 of NP, the class *SUPER-NP* is indeed very close to the class NP.

4 *SUPER-NP* Versus *PP*

Theorem 4.1. $PP\subseteq SUPER\text{-}NP$.

Proof. For each $L\in PP$, then there exists a probabilistic polynomial time (bounded by $p(|x|)$, where $p()$ is a polynomial) Turing machine M, s.t. $\forall x$, $\mathrm{Prob}[M(x)=\chi_L(x)]>\frac{1}{2}$. Using M we can define another *PPTM* M' as follows:

$$M'(x,(a_1,a_2,\ldots,a_{p(|x|)-1},b_1,b_2,\ldots,b_{p(|x|)}))=\begin{cases}\text{if } a_1a_2\ldots a_{p(|x|)-1}\neq 0 & \text{then return 'NO'}\\ \text{else return} & M(x,(b_1,b_2,\ldots,b_{p(|x|)}))\end{cases}$$

This gives us that:

$$x\in L\Rightarrow \mathrm{Prob}[M'(x)=1]=2^{-(p(|x|)-1)}\cdot \mathrm{Prob}[M(x)=1]\geqslant 2^{-p(|x|)}$$
$$x\notin L\Rightarrow \mathrm{Prob}[M'(x)=0]=(1-2^{-(p(|x|)-1)})-2^{-(p(|x|)-1)}\cdot \mathrm{Prob}[M(x)=0]$$
$$>(1-2^{-(p(|x|)-1)})+2^{-p(|x|)}=1-2^{-p(|x|)}$$

So M' satisfies Definition 3.1, and thus $L\in SUPER\text{-}NP$. □

5 Conclusions

In this paper we give a randomized version of NP. Based on this equivalent definition we give another randomized complexity class: *SUPER-NP*. Although the *SUPER-NP* is very close to NP, but we surprisingly find that $PP\subseteq SUPER\text{-}NP$ and thus $NP\subseteq PP\subseteq SUPER\text{-}NP$. We hope our work can contribute to the clarification between NP and PP.

In Ref. [3] O. Goldreich conjectured that $PP\subseteq PCP(\log^2, O(1))$. In light of $NP=PCP(\log, O(1))$ and the closeness of NP and *SUPER-NP* we hope we can finally solve this conjecture by showing that $SUPER\text{-}NP\subseteq PCP(\log^2, O(1))$. Indeed, the work in this line is currently under investigation.

Acknowledgements We would like to thank professor O. Goldreich for his careful reading of this paper, and especially for his suggestions on further research.

References:

[1] Garey, M.R., Johnson, D.S. Computers and Intractability: A Guide to the Theory of NP-Complete. New York: W.H. Freeman and Company, 1979. 19~39.

[2] Hopcroft, J.E., Ullman, J.D. Introduction to Automata Theory, Languages and Computation. Addison-Wesley Publishing Company, 1979. 285~320.

[3] Oded, Goldreich. Introduction to Complexity Theory. 1999. Available from http://www1.wisdom.weizmann.ac.il/~oded/ or http://theory.lcs.mit/~oded. 73~163.

[4] Papadimitriou, H. Computational Complexity. Addison-Wesley Publishing Company, 1994. 329~332.

[5] Gill, J. Computational complexity of probabilistic turing machines. SIAM Journal on Computing, 1977,6(4):675~695.

[6] Oded, Goldreich. Modern cryptography, probabilistic proofs and pseudorandomness. Algorithms and Combinatorics Series, Springer, 1998,17:31~73.

[7] Arora, S., Safra, S. Probabilistic checkable proofs: a new characterization of NP. JACM, 1998,45:70~122.

NP 对 PP

赵运磊， 朱 洪， 赵一鸣

（复旦大学 计算机科学系，上海 200433）

摘要：主要目的是研究 NP 与 PP 的关系.引入了一个 NP 的等价的随机定义.基于此等价定义，定义了另一个随机复杂性类：$SUPER\text{-}NP$.虽然 $SUPER\text{-}NP$ 与 NP 非常接近，但令人吃惊的是发现了 $PP \subseteq SUPER\text{-}NP$，从而 $NP \subseteq PP \subseteq SUPER\text{-}NP$.考虑到 $NP=PCP(\log, O(1))$ 以及 NP 和 $SUPER\text{-}NP$ 的相似性，也希望能通过证明 $SUPER\text{-}NP \subseteq PCP(\log^2, O(1))$ 来解决 $PP \subseteq PCP(\log^2, O(1))$ 的猜想.

关键词：NP；PP；PCP；随机计算；复杂性理论

中图法分类号：TP301 文献标识码：A