Computational Complexity — tutorial 11

Probabilistic algorithms 3, fine-grained complexity 1

Class	$\mathbb{P}$[algorithm accepts x] if		**Running time**	**Why is it named so?**
	$x \in L$	$x \notin L$		
P	1	0	polynomial	**P**olynomial
RP	$\geqslant \frac{1}{2}$	0	polynomial	**R**andomized **P**olynomial
co-RP	1	$\leqslant \frac{1}{2}$	polynomial	
BPP	$\geqslant \frac{3}{4}$	$\leqslant \frac{1}{4}$	polynomial	**B**ounded-error **P**robabilistic **P**olynomial
PP	$\geqslant \frac{1}{2}$	$< \frac{1}{2}$	polynomial	**P**robabilistic **P**olynomial
ZPP	1	0	**expected** polynomial	**Z**ero-error **P**robabilistic **P**olynomial

1. Prove that RP is closed under union and concatenation.

2. Prove that BPP is closed under union, complementation and concatenation.

3. *(exam '17)* Assume that there exists a polynomial time deterministic algorithm A which approximates with $\frac{2}{5}$ error the probability that a given circuit C with n inputs accepts a random n-bit input. Formally, given a circuit $C(x_1, \ldots, x_n)$, the algorithm computes a rational number $A(C)$ such that

$$|\mathbb{P}[C(x_1, \ldots, x_n) = 1] - A(C)| \leqslant \frac{2}{5}.$$

Prove that the existence of such an algorithm implies P = BPP.

Fine-grained complexity starts on the next page.

Let s_k be the smallest real number such that k-CNF-SAT with n variables and m clauses can be solved in $2^{s_k n} \cdot \mathrm{poly}(m)$ time[1].

We know that $s_2 = 0$ (as 2-CNF-SAT can be solved in polynomial—and even linear—time).

We know that $0 \leqslant s_3 \leqslant s_4 \leqslant s_5 \leqslant \cdots \leqslant 1$.

Exponential Time Hypothesis (ETH). $s_3 > 0$; that is, 3-CNF-SAT has no subexponential algorithm.

Strong Exponential Time Hypothesis (SETH). $\lim_{k\to\infty} s_k = 1$. It *follows* that CNF-SAT cannot be solved in $(2-\varepsilon)^n$ time for any $\varepsilon > 0$.

Orthogonal Vectors Conjecture (OVC). The following decision problem cannot be solved in $O(n^{2-\varepsilon} \cdot \mathrm{poly}(d))$ time for any $\varepsilon > 0$:

> ORTHOGONAL VECTORS
> INPUT: two sets $A = \{v_1, v_2, \ldots, v_n\}$, $B = \{w_1, w_2, \ldots, w_n\}$ of bit vectors, each of length d
> OUTPUT: are there two vectors $v_i \in A$, $w_j \in B$ such that $\langle v_i, w_j \rangle = 0$? *That is, on each position, at least one of these two vectors should have a 0 bit.*

We know that SETH $\Rightarrow$ OVC and that SETH $\Rightarrow$ ETH $\Rightarrow$ $\mathsf{P} \neq \mathsf{NP}$.

There are other various similar conjectures: 3SUM, 3XOR, APSP etc.

4. Prove that the following statements are equivalent. Note that (a) is equivalent to the negation of OVC.

(a) For some $\varepsilon > 0$, there exists an $O(n^{2-\varepsilon} \cdot \mathrm{poly}(d))$ algorithm solving ORTHOGONAL VECTORS.

(b) For some $\varepsilon > 0$, there exists an $O(n^{2-\varepsilon} \cdot \mathrm{poly}(d))$ algorithm for ORTHOGONAL VECTORS which additionally returns a pair of orthogonal vectors if it exists.

(c) For some $\varepsilon > 0$, there exists an $O(n^{2-\varepsilon} \cdot \mathrm{poly}(d))$ algorithm for ORTHOGONAL VECTORS constrained to $A = B$.

(d) For some $\varepsilon > 0$, there exists an $O(n^{1.5-\varepsilon} \cdot \mathrm{poly}(d))$ algorithm solving the following problem:

> SQUARE ROOT ORTHOGONAL VECTORS
> INPUT: two sets $A = \{v_1, \ldots, v_n\}$, $B = \{w_1, \ldots, w_{\sqrt{n}}\}$ of bit vectors, each of length d
> OUTPUT: are there two vectors $v_i \in A$, $w_j \in B$ such that $\langle v_i, w_j \rangle = 0$?

(e) For some $\varepsilon > 0$, there exists an $O(n^{2-\varepsilon} \cdot \mathrm{poly}(d))$ algorithm for the following problem:

> SUBSET VECTORS
> INPUT: two sets $A = \{v_1, v_2, \ldots, v_n\}$, $B = \{w_1, w_2, \ldots, w_n\}$ of bit vectors, each of length d
> OUTPUT: are there two vectors $v_i \in A$, $w_j \in B$ such that for each bit set in v_i, the corresponding bit in w_j is also set?

[1] This is slightly oversimplified. Formally, s_k is the infimum of the set of real numbers δ such that k-CNF-SAT can be solved in $2^{\delta n} \cdot \mathrm{poly}(m)$ time. If, for each $p \geqslant 1$, there existed a $2^{(1.15+1/p)n} n^p$ algorithm solving 7-CNF-SAT, we would have $s_7 = 1.15$ although there's no $2^{1.15n}\mathrm{poly}(m)$ algorithm solving the problem.

5. Assuming SETH, prove that the following problem cannot be solved in $O(n^{k-\varepsilon} \cdot \text{poly}(d))$ time for any $\varepsilon > 0$:

> k-ORTHOGONAL VECTORS
> INPUT: k sets $A_1, A_2, \ldots, A_k$ of n bit vectors, each of length d
> OUTPUT: are there k vectors $w_1 \in A_1$, $w_2 \in A_2$, $\ldots$, $w_k \in A_k$ such that, on each of d positions, at least one of the chosen vectors has a 0 bit?

Note that 2-ORTHOGONAL VECTORS is exactly ORTHOGONAL VECTORS.