Computational Complexity — tutorial 9

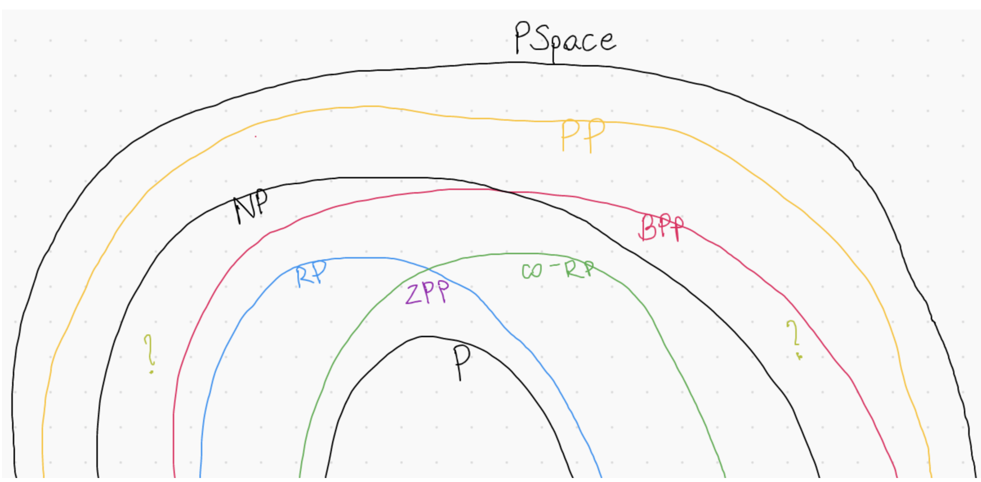
Probabilistic algorithms 1

Probabilistic complexity classes compared with P:

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

Well known open-problem: $\mathsf{RP} \stackrel{?}{=} \mathsf{P}$. The inclusion $\mathsf{P} \subseteq \mathsf{RP}$ is trivial, the other one is hard. In other words: can randomized algorithms be efficiently derandomized?

Schwartz-Zippel lemma *(simplified version)* Fix a prime p. Given a non-zero polynomial Q (over the integers mod p) with variables $x_1, x_2, \ldots, x_n$ and total degree $d \geqslant 0$, the probability that $Q(x_1, x_2, \ldots, x_n) = 0 \mod p$ for a random tuple of variables $x_1, \ldots, x_n$ is bounded from above by $\frac{d}{p}$.



Rysunek 1: Possible hierarchy of the complexity classes. Note that the relation between NP and BPP is **unknown**, but at the same time it's conjectured that BPP = P. Also, the hierarchy or its parts may collapse (it's even possible that PSpace = P).

1. You're given a fair coin ($\frac{1}{2}$ probability of getting heads, $\frac{1}{2}$ probability of getting tails). How to simulate a skewed coin, with $\frac{2}{3}$ probability of getting heads, and $\frac{1}{3}$ probability of getting tails?

2. How to do it the other way around: given a skewed coin ($\frac{2}{3}$ probability of heads, $\frac{1}{3}$ probability of tails), how to produce a fair coin?

3. For an undirected graph G with n vertices, we define the *Tutte matrix* as an $n \times n$ matrix defined as follows:

$$A_{i,j} = \begin{cases} x_{i,j} & \text{if } (i,j) \text{ is an edge of } G \text{ and } i < j, \\ -x_{j,i} & \text{if } (i,j) \text{ is an edge of } G \text{ and } i > j, \\ 0 & \text{otherwise.} \end{cases}$$

Here, each $x_{i,j}$ is a separate variable. For instance, the graph which is a cycle $1-2-3-4$ on 4 vertices has the following Tutte matrix:

$$\begin{pmatrix} 0 & x_{1,2} & 0 & x_{1,4} \\ -x_{1,2} & 0 & x_{2,3} & 0 \\ 0 & -x_{2,3} & 0 & x_{3,4} \\ -x_{1,4} & 0 & -x_{3,4} & 0 \end{pmatrix}$$

It can be proved that the determinant of this matrix is non-zero if and only if G has a perfect matching on n vertices.

Prove that verifying whether a given graph has a perfect matching is in RP.

Note: we can also prove that this problem is in P *by utilizing Edmonds' blossom algorithm, but we don't talk about this here.*

4. Consider the String Matching problem: given two strings s, t ($|s| \geqslant |t|$) of lowercase English characters, decide if t is a substring of s.

Let's solve it using Rabin-Karp algorithm: let $s = s_1 s_2 \dots s_n$, $t = t_1 t_2 \dots t_m$. For simplicity, assume that s_i, t_i are the 0-based indices of the corresponding characters in the English alphabet. For some prime $p \gg \max(26, n^2)$ and a random number $x \in [0, p-1]$, we define the polynomial hashes:

$$T = \left(t_1 + t_2 x + t_3 x^2 + \dots + t_m x^{m-1}\right) \mod p,$$

$$S_i = \left(s_i + s_{i+1} x + s_{i+2} x^2 + \dots + s_{i+m-1} x^{m-1}\right) \mod p \qquad \text{for } i \in \{1, 2, \dots, n-m+1\}.$$

(With some care, T and all $S_1, S_2, \dots, S_{n-m+1}$ can be computed in linear time.) We now guess that t is a substring of s if and only if $T \in \{S_1, S_2, \dots, S_{n-m+1}\}$.

Show that this algorithm proves that String Matching is in $\mathsf{co\text{-}RP}$.

Note: yes, it's obvious that the problem is in P*, but we're getting used to the complexity classes.*

5. Consider the Permutation Path problem: you're given a directed/undirected graph G with n vertices where each vertex is colored with one of k colors (called $1, 2, \dots, k$). Does there exist a simple path with k vertices such that the color of the first vertex is 1, the color of the second vertex is 2, ..., the color of the k-th vertex is k?

Prove that Permutation Path can be solved in polynomial time (without randomization).

6. Consider the Colorful Path problem: you're given a directed/undirected graph G with n vertices where each vertex is colored with one of k colors (called $1, 2, \ldots, k$). Does there exist a simple path with k vertices hitting each color exactly once?

Prove that Permutation Path can be solved in time $2^k \cdot \text{poly}(|G|)$ (without randomization).

7. Consider the k-Path problem: you're given a directed/undirected graph G with n vertices. Does there exist a simple path with k distinct vertices?

Prove that k-Path can be solved in time $(2e)^k \cdot \text{poly}(|G|)$ with randomization. Conclude that k-Path for $k = O(\log n)$ is in RP.

Hint: use Stirling's approximation: $k! \approx \left(\frac{k}{e}\right)^k \sqrt{2\pi k}$.

8. Prove the *amplification lemma* for RP: if we replace „$\geqslant \frac{1}{2}$", in the definition of RP with „$\geqslant \varepsilon$" for any constant $0 < \varepsilon < 1$, we'll get exactly the same definition of RP.

In other words: if we run the probabilistic algorithm over and over again, we'll be more and more confident about its answer.

9. Prove the *amplification lemma* for BPP: if we replace „$\geqslant \frac{3}{4}$", „$\leqslant \frac{1}{4}$" in the definition of BPP with „$\geqslant 1 - \varepsilon$", „$\leqslant \varepsilon$" for any constant $0 < \varepsilon < \frac{1}{2}$, we'll get exactly the same definition of BPP.

Hint: you can (but don't have to) use a variant of Chernoff bound for Bernoulli variables — for independent variables $X_1, X_2, \ldots, X_n \in \{0, 1\}$, $\mu = \mathbb{E}[X_1 + \cdots + X_n]$, $\delta \in (0, 1)$*:*

$$\mathbb{P}[X_1 + \cdots + X_n \leqslant (1 - \delta)\mu] \;\leqslant\; e^{-\frac{1}{2}\delta^2\mu}.$$

10. As above, but we assume that $\varepsilon = \frac{1}{2} - \frac{1}{n}$ where n is the length of the input.

11. Prove that $\mathsf{RP} \cap \mathsf{co\text{-}RP} = \mathsf{ZPP}$.

12. Prove that $\mathsf{NP} \subseteq \mathsf{PP}$.

13. Prove that RP is closed under union, concatenation and Kleene star.

14. Prove that BPP is closed under union, complementation, concatenation and Kleene star.