CMPT 407: Computational Complexity **Summer 2017**

Lecture 6: Randomized Computation (12 - 16 June)

Lecturer: Valentine Kabanets *Scribe: Lily Li*

The complexity classes we are going to consider in the next little while:

$$\mathsf{P} \subseteq \mathsf{BPP} \subseteq \mathsf{BQP}$$

where BPP is randomized polynomial time algorithm (assuming access to uniform random bits) and BQP is quantum polynomial time (assuming Quantum Mechanics is accurate and we can build large quantum systems.

Consider an example in Communication Complexity. Consider two parties *Alice* and *Bob* with strings a and b respectively. They both want to know if $a = b$. Let the cost of a protocol is the number of bits sent between the two parties. Consider the following deterministic protocol: *Alice* sends *Bob* string a. *Bob* compares a and b and sends *Alice* one bit denoting if they are equal or not. The complexity of the protocol is $n + 1$ for strings of length n. As it turns out, the best we can do with a deterministic protocol is n.

What if we use a randomized protocol? *Alice* picks a random prime number p in the range (n^2, n^3) and a random number r in the range $(1, p)$. For $a = a_1 \cdots a_n$, *Alice* calculates $A(r) = a_1 r^{n-1} + \cdot + a_n \mod p$. *Alice* sends *Bob* the string $(q, r, A(r))$ (actually p is not necessary but you might as well send it anyways). For $b = b_1 \cdots b_n$, *Bob* computes $B(r) = b_1 r^{n-1} + \cdots + b_n \mod p$ and compares $A(r)$ and $B(r)$ returning *Alice* one bit. The cost of this protocol is $9 \log n$. To see how good this protocol is

6.1 Randomized Complexity Class

A more granular subdivision of Randomized Complexity classes:

$$\mathsf{ZPP} \subset \mathsf{RP} \subset \mathsf{BPP}$$

Definition 6.1 *A language $L \in \mathsf{BPP}$ if there is a polynomial time DTM $M(x, r)$ such that $\forall x \in L$.*

Similarly the one sized randomized complexity class RP is defined as: .

Finally the class ZPP is defined as

To construct a randomized algorithm we need to ensure that

Theorem 6.2 *If $L \in \mathsf{RP}$ (recall $\frac{1}{2}$ chance of error for accepting) then we can reduce the error to $\frac{1}{2^n}$ for any $n \in \mathbb{N}$.*

Proof: Quite straight forward. Just run the RTM M on input x a bunch of times. If any trial accepts then $x \in L$ since M cannot be wrong on rejecting inputs. ∎

Theorem 6.3 *If $L \in \mathsf{BPP}$ (recall $\frac{1}{4}$ chance of error for both accepting and rejecting) then we can reduce the error to $\frac{1}{2^n}$ for any $n \in \mathbb{N}$.*

Now lets consider ZPP in-depth:

Theorem 6.4 $L \in \mathsf{ZPP} \iff L \in \mathsf{RP} \cap \mathsf{coRP} \iff$ *there exists a randomized algorithm that is always correct and has expected polynomial running time. With random variable $T(x, r)$ be the running time on input x with randomness r. The expected running time on x is $Exp_r[T(x, r)] = \sum_r T(x, r) \cdot Pr(r)$.*

Proof: First show the first $\iff$ holds. In particular we will show that $L \in \mathsf{RP} \cap \bar{L} \in \mathsf{RP} \implies L \in \mathsf{ZPP}$. Next we have $L \in \mathsf{ZPP}$ and need to show that $L \in \mathsf{RP} \cap \bar{L} \in \mathsf{ZPP}$.

For the next we show that $L \in \mathsf{ZPP} \iff$ ■