5CCS2FC2: Foundations of Computing II

Probability, Expectation & Probabilistic Complexity Classes

Week 9

Dr Christopher Hampson

Department of Informatics

King's College London

Probabilistic Complexity Classes

And their relation to P vs NP

Probabilistic Complexity Classes

- **Bounded-error Probabilistic Polynomial Time (BPP)**
  - A language L is said to be solvable in **bounded-error probabilistic polynomial time** if there is *some* machine M that is:
    - **Probabilistically "Sound"** — If $w \notin L$ then $\textbf{Prob}\,(M(w) = 1) \leq 1/3$
    - **Probabilistically "Complete"** — If $w \in L$ then $\textbf{Prob}\,(M(w) = 1) \geq 2/3$
    - **Polynomial-time** — $T(n) \in O(n^k)$ for some $k > 0$

Probabilistic Complexity Classes

- **Zero-error Probabilistic Polynomial Time (ZPP)**
  - A language L is said to be solvable in **zero-error probabilistic polynomial time** if there is *some* machine M that is:
    - **Sound** — If $w \notin L$ then $\textbf{Prob}\,(M(w) = 1) = 0$
    - **Complete** — If $w \in L$ then $\textbf{Prob}\,(M(w) = 1) = 1$
    - **Expected Polynomial-time** — $\mathbf{E}\,[T(n)] \in O(n^k)$ for some $k > 0$

Monte Carlo vs Las Vegas

Monte Carlo Algorithms

A randomized algorithm with deterministic runtime but may return incorrect answers due to their probabilistic nature.

$$\mathbf{BPP} = \left\{ \begin{array}{c} \text{problems solvable in polynomial time} \\ \text{with a Monte Carlo Algorithm} \end{array} \right\}$$

Monte Carlo vs Las Vegas

Las Vegas Algorithms

A randomized algorithm that always sound and complete, but whose runtime may vary due to their probabilistic nature.

$$\mathbf{ZPP} = \left\{ \begin{array}{c} \text{problems solvable in polynomial time} \\ \text{with a Las Vegas Algorithm} \end{array} \right\}$$

Monte Carlo vs Las Vegas

Theorem **ZPP $\subseteq$ BPP**

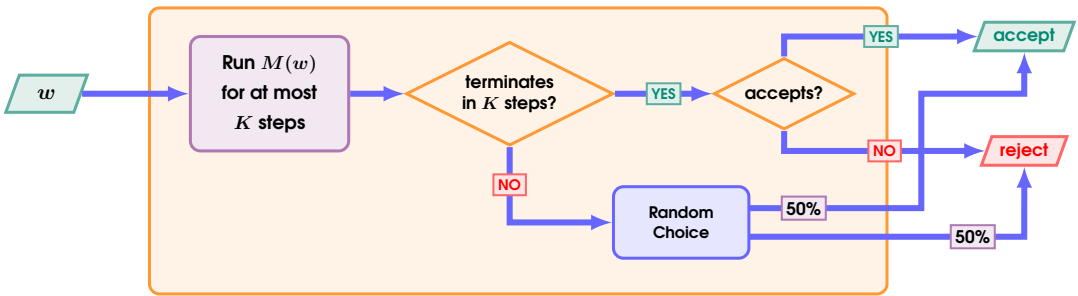
Proof:

Step 1) Suppose that $L \in$ **ZPP** then there is some PTM M that is **sound** and **complete** with repect to L with an **expected polynomial** runtime

$$\mathbf{E}[T(n)] < c \cdot n^k \qquad \text{for some } c, k > 0.$$

Monte Carlo vs Las Vegas

Step 2) Construct a new PTM M' as follows, where $K = 3 \cdot \mathbf{E}[T(n)]$



Step 3) Clearly M' always **terminates** in $O(n^k)$ steps, by design.

Step 4) We claim that this algorithm is **probabilistically sound**

$$w \notin L \quad \Longrightarrow \quad \textbf{Prob}\left(M'(w) = 1\right) \leq 1/3$$

Step 3.1) Suppose that $w \notin L$

Step 3.2) Since, M is sound M' will only **incorrectly accept** w if it had to guess

$$\textbf{Prob}\left(M'(w) = 1\right) \leq \textbf{Prob}\left(T(n) > 3 \cdot \textbf{E}\left[T(n)\right]\right) \leq 1/3$$

Step 5) We claim that this algorithm is **probabilistically complete**

$$w \in L \quad \Longrightarrow \quad \textbf{Prob}\left(M'(w) = 1\right) \geq 2/3$$

Step 4.1) Suppose that $w \in L$

Step 4.2) Since, M is complete M' will only **incorrectly reject** w if it had to guess

$$\textbf{Prob}\left(M'(w) = 1\right) \quad \geq \quad 1 - \textbf{Prob}\left(T(n) > 3 \cdot \textbf{E}\left[T(n)\right]\right) \geq 2/3$$

Monte Carlo vs Las Vegas

Step 6) Hence our machine M' is a **polynomial-time Monte Carlo** algorithm.

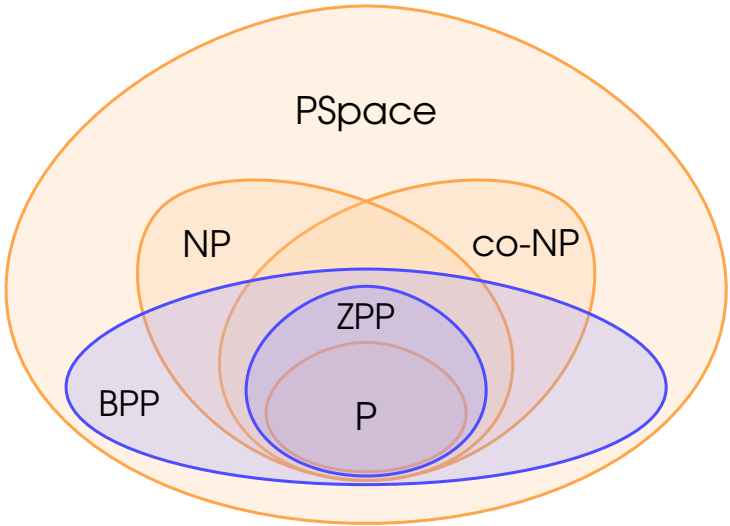
Therefore $L \in \mathbf{BPP}$.

Q.E.D.

Complexity Hierarchy

(Extended)

Complexity Hierarchy (Extended)



$$\mathbf{BPP} \overset{?}{\subseteq} \mathbf{NP}$$

$$\mathbf{BPP} \overset{?}{\supseteq} \mathbf{NP}$$

$$\mathbf{ZPP} \overset{?}{=} \mathbf{P}$$

$$\mathbf{BPP} \overset{?}{=} \mathbf{P}$$

End of Slides!