Lesson 8: Randomized protocols

Theme: Randomized protocols with private and public coins.

1 Protocols with private coins

A (randomized) protocol with private coins with domain \(X \times Y\) and range \(Z\) is defined as a triplet \((P, \Pr_a, \Pr_b)\), where

- \(P\) is a deterministic protocol with domain \((X \times \{0,1\}^n_a) \times (Y \times \{0,1\}^n_b)\) and range \(Z\).
- \(\Pr_a\) and \(\Pr_b\) are probability distribution on \(\{0,1\}^n_a\) and \(\{0,1\}^n_b\), respectively.

Intuitively, it works as follows. On input \((x,y)\in X \times Y\), Alice and Bob do the following.

- Alice and Bob choose independently the strings \(r_A \in \{0,1\}^n_a\) and \(r_B \in \{0,1\}^n_b\).
- Then, they run the protocol with inputs \((x,r_A)\) and \((y,r_B)\), respectively.

The strings \(r_A\) and \(r_B\) are called Alice’s and Bob’s random strings. Note that the probability that Alice and Bob choose \(r_A\) and \(r_B\), respectively, is \(\Pr_a[r_A] \cdot \Pr_b[r_B]\).

We will use the following terminologies.

- \(P\) computes a function \(f\) with zero error, if for every \((x,y)\in X \times Y\),
  \[\Pr[P(x,y) = f(x,y)] = 1.\]

- \(P\) computes a function \(f\) with \(\epsilon\) error, if for every \((x,y)\in X \times Y\),
  \[\Pr[P(x,y) = f(x,y)] \geq 1 - \epsilon.\]

- \(P\) computes a function \(f\) with one-sided \(\epsilon\) error, if the following holds.
  - For every \((x,y)\in X \times Y\) such that \(f(x,y) = 0\), \(\Pr[P(x,y) = 0] = 1.\)
  - For every \((x,y)\in X \times Y\) such that \(f(x,y) = 1\), \(\Pr[P(x,y) = 1] \geq 1 - \epsilon.\)

The worst case running time of a randomized protocol \(P\) is defined as follows.

- The worst case running time of \(P\) on input \((x,y)\) is:
  \[\text{worst}(P, x, y) := \max_{r_A, r_B} \text{the cost of } P \text{ on input } ((x, r_A), (y, r_B))\]

- The worst case running time of \(P\) is:
  \[\text{worst}(P) := \max_{(x,y)} \text{worst}(P, x, y)\]

Note that on input \((x,y)\) a randomized protocol \(P\) may yield different costs on different random strings. We define a random variable \(\text{cost}(P, x, y)\) as follows.

\[\text{cost}(P, x, y) := \text{the cost of } P \text{ on input } (x, y) \text{ on random strings } r_A, r_B.\]
• The average case running time of $\mathcal{P}$ on input $(x, y)$ is:
  \[
  \text{average}(\mathcal{P}, x, y) := E[\text{cost}(\mathcal{P}, x, y)]
  \]
  That is, the expectation of $\text{cost}(\mathcal{P}, x, y)$.

• The average case running time of $\mathcal{P}$ is:
  \[
  \text{average}(\mathcal{P}) := \max_{(x,y)} \text{average}(\mathcal{P}, x, y)
  \]

**Definition 8.1** Let $f : X \times Y \to \{0, 1\}$ be a function.

- $R_0(f) := \min_{\mathcal{P}}$ computes $f$ with zero error $\text{average}(\mathcal{P})$.
- $R_\epsilon(f) := \min_{\mathcal{P}}$ computes $f$ with $\epsilon$ error $\text{worst}(\mathcal{P})$, where $0 < \epsilon < 1/2$.
- $R(f) := R_{1/3}(f)$
- $R_1^\epsilon(f) := \min_{\mathcal{P}}$ computes $f$ with one-sided $\epsilon$ error $\text{worst}(\mathcal{P})$, where $0 < \epsilon < 1$.

Note that for every function $f : X \times Y \to \{0, 1\}$,

\[
\min_{\mathcal{P}} \text{ computes } f \text{ with zero error } \text{worst}(\mathcal{P}) = D(f).
\]

**Proposition 8.2** Let $f : X \times Y \to \{0, 1\}$ be function. For every (randomized) protocol $\mathcal{P}$ that computes $f$ with $\epsilon$ error, there is a protocol $\mathcal{P}'$ that computes $f$ with $2\epsilon$ error such that

\[
\text{worst}(\mathcal{P}') \leq (1/\epsilon) \cdot \text{average}(\mathcal{P}).
\]

Moreover, if $\mathcal{P}$ is one-sided, then so is $\mathcal{P}'$.

Recall the function $\text{EQ}(x, y) = 1$ if and only if $x = y$, where $x, y \in \{0, 1\}^n$.

- $R_{1/\alpha}(\text{EQ}) = O(\log n)$.
- $R_{1/\alpha}(\text{NEQ}) = O(\log n)$, where NEQ is the complement of EQ.

2 Protocols with public coins

A protocol with public coins $\mathcal{P}$ is one with a probability distribution on $\{0, 1\}^n$. On input $(x, y)$, both Alice and Bob together choose a random string $r \in \{0, 1\}^n$ (independent of $(x, y)$) and runs the protocol with both Alice’s and Bob’s input being $(x, r)$ and $(y, r)$. So, the random string $r$ is seen by both Alice and Bob, hence the name “public.”

The complexity measure of a function $f : X \times Y \to \{0, 1\}$ (w.r.t. protocols with public coins) can be defined similarly as in Definition 8.1 and denoted by $R^\text{pub}_0, R^\text{pub}_\epsilon(f), R^\text{pub}(f), R^\text{1,pub}_\epsilon(f)$.

We can show that $R^\text{1,pub}(\text{NEQ}) = O(1)$.

The following theorem states that public and private coins do not differ much complexity-wise.

**Theorem 8.3** Let $f : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$.

- $R_{\epsilon + \delta}(f) \leq R^\text{pub}_0(f) + O(\log n + \log(1/\delta))$, for every $\epsilon, \delta > 0$.
- $R_0(f) = O(R^\text{pub}_0(f) + \log n)$. 


3 Some examples

- $R^1(\text{EQ}) = \Theta(n)$.
- $R_0(\text{EQ}) = R_0(\text{NEQ}) = \Theta(n)$.
- $R(\text{EQ}) = R(\text{NEQ}) = \Theta(\log n)$.
- $R^1(\text{NEQ}) = \Theta(\log n)$.

Recall the function $\text{DISJ}_{k,n}$, where for every $x, y \subseteq \{1, \ldots, n\}$ such that $|x| = |y| = k$, $\text{DISJ}_{k,n}(x, y) = 1$ if and only if $x \cap y = \emptyset$.

- $R^{\text{pub}}(\text{DISJ}_{k,n}) = O(k)$.
- $R(\text{DISJ}_{k,n}) = O(k + \log n)$.

4 Relations between randomized, det. and non-det. protocols

Theorem 8.4 Let $f : X \times Y \rightarrow \{0, 1\}$ be a Boolean function.

- For every $0 \leq \epsilon < 1$, $R^1_\epsilon(f) \geq N^1(f)$.
- $R_0(f) \geq N(f)$.
- $R(f) = \Omega(D(f))$. More precisely,

$$D(f) \leq 2^{R_\epsilon(f)} \left( \log \left( \frac{1}{2} - \epsilon \right)^{-1} + R_\epsilon(f) \right).$$