Complexity of Randomization

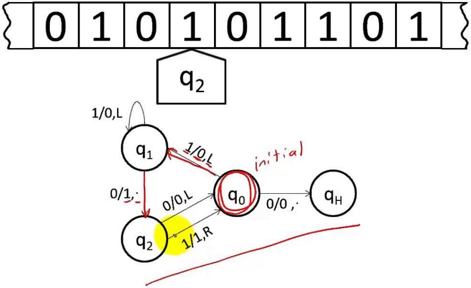
Ryan Anselm

June 22, 2022

Outline

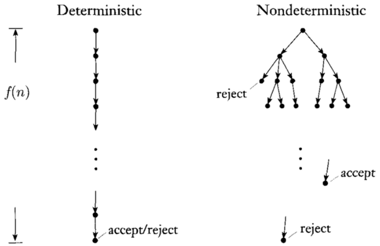
Background on Turing Machines

- (Deterministic) Turing machines (TMs) are a model of computation which use an infinite tape and a finite state automata.
- The transition function δ is a function of the current state of the automata and the current value being read on the tape that tells the TM what to do next. Given a standard TM, the computation it performs on an input string is a completely **deterministic** process.
- The set of input strings that a Turing machine will accept is its **language**, often denoted L.



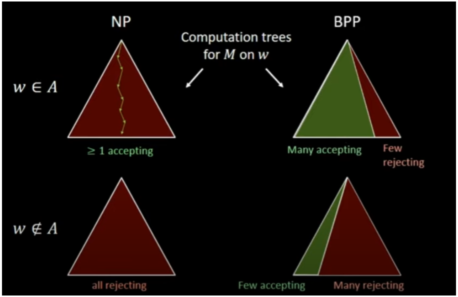
Probabilistic Turing Machines (PTMs) (Section 7.1)

- To formalize randomized computation, we define *Probabilistic Turing Machines (PTMs)*.
- **Def 7.1:** A *probabilistic Turing machine* is a Turing machine with two transition functions: δ_0 and δ_1. To execute PTM M on an input x, at each step we randomly choose with probability $\frac{1}{2}$ for each whether to apply δ_0 or δ_1.
- Choice of δ_0 or δ_1 is an independent random variable. M outputs 1 (accept) or 0 (reject).



PTMs cont.

- At each step of a PTM, each computation randomly branches into two paths with a 50/50 chance.
- For a computation of t steps, there are 2^t possible paths in the graph of all possible computations.
- $Pr[M(x) = 1]$ is equal to the fraction of branches leading to a 1 (accept) output.



BPP

- **Def 7.2:** For function $T : \mathbb{N} \to \mathbb{N}$ and $L \subseteq \{0,1\}^*$, a PTM M decides L in $T(n)$ if for every input $x \in \{0,1\}^*$, M halts in $T(|x|)$ steps regardless of its random steps, and $Pr[M(x) = L(x)] \geq \frac{2}{3}$.
- **BPTIME**(T(n)) = class of languages decided in $O(T(n))$ steps.
- **BPP** $= \cup_c$**BPTIME**(n^c).
- Essentially, **BPP** is the complexity class of problems that can be decided in polynomial time with sufficiently high probability of correctness (taken to be $\frac{2}{3}$ here).
- Choice of $\frac{2}{3}$ as constant is arbitrary, any fraction $> \frac{1}{2}$ would be equivalently strong. (Shown in Error Reduction Proof)

BPP cont.

- **P** $\subseteq$ **BPP** because if we choose $\delta_0 = \delta_1 = \delta$, the transition function becomes fixed.
- **(Alternative) Def 7.3:** a language L is in **BPP** if there exists a poly-time TM M and a polynomial $p : \mathbb{N} \rightarrow \mathbb{N}$ such that for every $x \in \{0,1\}^*$, $Pr_{r \in \{0,1\}^{p(|x|)}}[M(x, r) = L(x)] \geq 2/3$.
- All this is doing is making the sequence of randomized coin flips be an additional input to a standard Turing machine.
- This suggests that **BPP** $\subseteq$ **EXP** since it is possible to enumerate all possible random choices of a poly(n) PTM in $2^{poly(n)}$ time.
- **BPP = P**? Suspected to be true, but presently unproven.

Error reduction/amplification (Thm. 7.10)

We can show that we can replace $\frac{2}{3}$ with any constant greater than $\frac{1}{2}$ in the definition of **BPP**.

- **Thm. 7.10:** Suppose there exists a poly-time PTM M for language L such that for every input x and $c > 0$, $Pr[M(x) = L(x)] \geq \frac{1}{2} + |x|^{-c}$. Then, for every constant $d > 0$, there exists a poly-time PTM M' such that for every input x, $Pr[M'(x) = L(x)] \geq 1 - 2^{-|x|^d}$.
- This theorem shows there is a way to transform a poly-time PTM that has any tolerance greater than $1/2$ (since $1/2 + |x|^{-c}$ gets arbitrarily close to $1/2$ based on choice of c) to a new poly-time PTM with an arbitrarily high tolerance level (since $1 - 2^{-|x|^d}$ gets arbitrarily close to 1 based on choice of d).
- M' does the following: Repeat running $M(x)$ for $k = 8|x|^{2c+d}$ times, obtaining k outputs of 1 or 0. Output 1 if the majority of these runs output 1 and 0 otherwise.
- Repeating M k times is scaling it by a polynomial factor, so M' is still poly.

Error reduction/amplfication (Thm. 7.10) cont.

- Define random variable X_i as whether $M(x) = L(x)$ on the ith run of M. We have that $\mathbb{E}[X_i] = Pr[X_i = 1] = p$ where $p = \frac{1}{2} + |x|^{-c}$ in the worst case.
- We can use the *Chernoff Bound* from probability theory to analyze this: $Pr[|\sum_{i=1}^{k} X_i - pk| > \delta pk] < e^{-\frac{\delta^2}{4}pk}$ (proof of this statement not provided).
- If we set $\delta = |x|^{-c}/2$, for the RHS of the Chernoff Bound we get $e^{-\frac{1}{4|x|^{2c}}\frac{1}{2}8|x|^{2c+d}} = e^{-|x|^d} \leq 2^{-|x|^d}$
- To interpret the LHS of the Chernoff bound, note that $\mathbb{E}[\sum_{i=1}^{k} X_i] = \sum_{i=1}^{k} \mathbb{E}[X_i] = kp$ so Chernoff describes an upper bound on the amount the actual number of successes deviates from the expected number of successes.
- The Chernoff bound shows that if we pick $k = 8|x|^{2c+d}$, the probability of M' outputting an incorrect result is at most $2^{-|x|^d}$, so the probability of a correct result is $\geq 1 - 2^{-|x|^d}$ (as arbitrarily close to 1 as we like).

One-sided error (Section 7.3)

- **BPP** allows for both false positives and false negatives to occur. (i.e. outputting 0 when $x \in L$ and 1 when $x \notin L$)
- Many probabilistic algorithms have one-sided error (i.e. if x not in L, they will never output 1, but may output 0 when x in L). This is better than BPP, and is captured by the class **RP**.

RP and coRP

- **Def 7.6: RTIME**(T(n)) contains all L such that there exists a PTM that runs in $T(n)$ such that
  $x \in L \implies Pr[M(x) = 1] \geq \frac{2}{3}, \ x \notin L \implies Pr[M(x) = 0] = 1$
- **RP** $= \cup_{c>0}$**RTIME**(T(n^c)).
- **RP** $\subseteq$ **NP** because any accepting branch of the computation is a "certificate" showing that an input is in the language
- **coRP** is the class of one-sided error on the other side.
  (**coRP** $= \{L | \bar{L} \in$ **RP**$\}$).
- **RP** only has false negative errors, **coRP** only has false positive errors.

ZPP

- Define random variable $T_{M,x}$ to be the running time of PTM M on input x.
- "Zero-sided error": a PTM that never makes errors.
- **Def 7.7: ZTIME**(T(n)) contains all languages L for which there is a machine M which has an expected running time of $O(T(n))$, and such that for all inputs x, $M(x) = L(x)$. (The machine returns a result that is always correct)
- **ZPP** $= \cup_c$**ZTIME**$(\mathrm{T}(n^c))$.
- **ZPP** is the class of all algorithms which have an *expected* running time that is polynomial time. (i.e. $\mathbb{E}[T_{M,x}]$ is polynomial in $|x|$).
- The running time may be longer than $poly(|x|)$ in some computation paths, but the overall expected value must be $O(poly(|x|))$.

Theorem 7.8: ZPP = RP ∩ coRP

Proof:

1. Show $RP \cap coRP \subseteq ZPP$
   - Let $L \in RP \cap coRP$
   - Then L is recognized by PTMs M_1 and M_2 where M_1 matches RP constraints (if output is 1, guaranteed $x \in L$) and M_2 matches $coRP$ constraints (if output is 0, guaranteed $x \notin L$).
   - Construct a PTM M which is always right and has polynomial expected runtime as follows:
   - Run M_1. If it outputs 1, M outputs 1. Run M_2. If it outputs 0, M outputs 0. If M_1 gives 0 and M_2 gives 1, loop back and repeat this process again.
   - In each loop, $Pr[M_1(x) = 0 \wedge M_2(x) = 1] \leq 1/2$. (If $x \notin L$, M_1 always outputs 0 and M_2 outputs 1 $\frac{1}{3}$ of the time; If $x \in L$, M_1 outputs 0 $\frac{1}{3}$ of the time and M_2 always outputs 1)
   - The times needed to repeat the loop follows a geometric random variable. $\mathbb{E}[\text{runtime of M}] \leq 2(\text{runtime of } M_1 + \text{runtime of } M_2)$.

ZPP = RP ∩ coRP cont.

2. Show $ZPP \subseteq RP \cap coRP$

- Fix $L \in ZPP$, let M be a PTM accepting L that satisfies ZPP constraints (i.e. $\mathbb{E}[\text{runtime of M on x}] \leq \text{p}(|x|)$ for some polynomial p of $|x|$).
- To show $L \in RP$: Create a PTM M' which on an input x runs M for $3p(|x|)$ steps.
- If M' outputs something in $3p(|x|)$ steps, return that output.
- If M' runs more than $3p(|x|)$ steps, halt and output 0.
- We can use Markov's inequality: $Pr(X \geq a) \leq \frac{\mathbb{E}[X]}{a}$ with $a = 3\mathbb{E}[X]$ to observe that $Pr(X \geq 3\mathbb{E}[X]) \leq \frac{1}{3}$.
- If $x \notin L$ then M' is guaranteed to output 0. If $x \in L$, then M' outputs 0 with probability $\leq \frac{1}{3}$ (take X as a random variable representing the running time of M on x), so outputs 1 with probability $\geq \frac{2}{3}$
- $\implies L \in RP$. To show $L \in coRP$, have M' output 1 instead after $3p(|x|)$ steps.

ZPP $=$ RP $\cap$ coRP

- We have shown $RP \cap coRP \subseteq ZPP$
- We have also shown $ZPP \subseteq RP \cap coRP$
- Hence, $ZPP = RP \cap coRP$ $\blacksquare$!