Complexité avancée - TD 6

Benjamin Bordais

November 18, 2020

We recall the definition of RP, coRP and BPP. A language L is in RP if there exists a Turing machine $\mathcal{M}$ running in polynomial time $p(n)$ on all input x such that $|x| = n$ and random tape r of size $p(n)$ such that:

- If $x \in L$, then $Pr_r[\mathcal{M}(x,r) = \top] \geq 1/2$;
- If $x \notin L$, then $Pr_r[\mathcal{M}(x,r) = \top] = 0$.

Similarly, a language L is in coRP if there exists such a Turing machine $\mathcal{M}$ ensuring:

- If $x \in L$, then $Pr_r[\mathcal{M}(x,r) = \top] = 1$;
- If $x \notin L$, then $Pr_r[\mathcal{M}(x,r) = \top] \leq 1/2$.

Finally, a language L is in BPP if there exists such a Turing machine $\mathcal{M}$ ensuring:

- If $x \in L$, then $Pr_r[\mathcal{M}(x,r) = \top] \geq 2/3$;
- If $x \notin L$, then $Pr_r[\mathcal{M}(x,r) = \top] \leq 1/3$.

Exercise 1 One-Minute Long Exercise

Between RP and coRP which language is "No-means-No" which language is "Yes-means-Yes"?

Solution:
RP is "Yes-means-Yes". Indeed, if $x \notin L$ then $P[\mathcal{M}(x,r) \text{ accepts}] = 0$. Therefore, if $P[\mathcal{M}(x,r) \text{ accepts}] \neq 0$ then $x \in L$. That is, if $\mathcal{M}(x,r)$ accepts, then $x \in L$.
Similarly, coRP is "No-means-No".

Exercise 2 Expected Running Time

Given a probabilistic Turing Machine M, not necessarily halting, let $T_M(x,r)$ be the random variable describing the running time of M on input x and random tape r (take $T_M(x,r) = +\infty$ if M does not halt on x,r). That is for all x, $Pr[T_M(x,r) = T]$ is the probability, taken over all possible (infinite) random tape contents, that M on input x halts after exactly T steps.

The expected running time of M on input x is the expectation $E[T_M(x,r)]$.

Consider the definitions of RP and BPP: here the Turing machines considered are required to halt in time at most n^c steps for some $c \geq 1$ on all inputs and for all possible random tape strings (worst case running time). Define RP^E and BPP^E as RP and BPP, but replacing the worst case running time with the expected running time.

Formally:

- $\mathsf{RP}^E = \bigcup_{c\in\mathbb{N}} \mathsf{RTIME}^E(n^c, 0, 1/2)$
- $\mathsf{BPP}^E = \bigcup_{c\in\mathbb{N}} \mathsf{RTIME}^E(n^c, 1/3, 2/3)$

where $\mathsf{RTIME}^E(n^c, p_{acc}, p_{rej})$ is the class of languages L for which there exists a probabilistic Turing machine M (which may not halt) such that, for each input x of size n:

- $Pr[T_M(x,r) = +\infty] = 0$;
- $E[T_M(x,r)] \leq |x|^c$;
- if $x \in L$ then $Pr[M(x,r)] = \top] \geq p_{rej}$;
- if $x \notin L$ then $Pr[M(x,r)] = \top] \leq p_{acc}$.

Show that $\mathsf{RP}^E = \mathsf{RP}$ and $\mathsf{BPP}^E = \mathsf{BPP}$.

Solution:

Idea: The idea is to put a timeout and, if the execution time runs out, do the right thing (here reject). That is:

- We consider a function K as timeout.
- Use Markov's inequality to have a boundary on the time when it exceeds the timeout.
- Set an appropriate value for K.
- Use error reduction.

There is no difference between RP and BPP here, except for the boundary K. More formally:

1. - $\mathsf{RP} \subseteq \mathsf{RP}^E$: Obvious, because if a TM halts in a polynomial time it halts with a average polynomial time. $\square$
   - $\mathsf{RP}^E \subseteq \mathsf{RP}$:
     Given a Turing machine $\mathcal{M}$ such as in the definition of RP^E for a language L in RP^E. By definition, there exists c s.t. $E[T_M(x,r)] < |x|^c$. For $K \in \mathbb{R}[X]$ a polynom, we define $\mathcal{M}_K$ a TM which executes M on x and rejects if the number of steps taken exceeds $K(|x|)$. Then:
     - If $x \notin L$, $Pr[\mathcal{M}_K(x,r) = \top] \leq Pr[\mathcal{M}(x,r) = \top] = 0$
     - If $x \in L$, $Pr[M_K(x,r) = \bot] \leq Pr[M(x,r) = \bot] + Pr[T_M(x,r) \geq K(|x|)]$

     By Markov's inequality, $Pr[T_M(x,r) \geq K(|x|)] \leq= \frac{E(T_M(x,r))}{K(|x|)} = \frac{|x|^c}{K(|x|)}$.
     If we set $K(n)$ to $4 \cdot n^c$ (for instance), we have $Pr[M_K(x,r) = \top] = 1 - Pr[M_K(x,r) = \bot] \geq 1 - (\frac{1}{2} + \frac{1}{4}) \geq \frac{1}{4}$. It follows that $L \in \mathsf{RP}(1/4) = \mathsf{RP}$.
2. $\mathsf{BPP}^E = \mathsf{BPP}$: it's exactly the same proof.
   - $\mathsf{BPP} \subseteq \mathsf{BPP}^E$: similarly.

- $\mathsf{BPP}^E \subseteq \mathsf{BPP}$:
  Given a Turing machine $\mathcal{M}$ such as in the definition of BPP^E for a language L in BPP^E. By definition, there exists c s.t. $E[T_M(x,r)] < |x|^c$. For $K \in \mathbb{R}[X]$ a polynom, we define $\mathcal{M}_K$ a TM which executes M on x and rejects if the number of steps taken exceeds $K(|x|)$. Then:
  - If $x \notin L$, $Pr[\mathcal{M}_K(x,r) = \top] \leq Pr[\mathcal{M}(x,r) = \top] \leq 1/3$
  - If $x \in L$, $Pr[M_K(x,r) = \bot] \leq Pr[M(x,r) = \bot] + Pr[T_M(x,r) \geq K(|x|)]$

  By Markov's inequality, $Pr[T_M(x,r) \geq K(|x|)] \leq \frac{E(T_M(x,r))}{K(|x|)} = \frac{|x|^c}{K(|x|)}$.
  If we set $K(n)$ to $12 \cdot n^c$ (for instance), we have $Pr[M_K(x,r) = \top] = 1 - Pr[M_K(x,r) = \bot] \geq 1 - (\frac{1}{3} + \frac{1}{12}) \geq \frac{7}{12}$. Hence, in both cases, the probability of error is lower than or equal to 5/12. That is, $L \in \mathsf{BPP}(5/12) = \mathsf{BPP}$.

Exercise 3 BPP and PSPACE

- Argue that $\mathsf{BPP}(1/2) = \{$ all languages $\}$ and $\mathsf{BPP} = \mathsf{coBPP}$.
- Give a direct proof that $\mathsf{BPP} \subseteq \mathsf{PSPACE}$.

Solution:

- For an arbitrary language L, we can consider the randomized Turing machine that accepts an input with probability 1/2 regardless of that input. Furthermore, from a probabilistic Turing machine such that $L \in \mathsf{BPP}$, we can swap the accept and reject so that we also have $\bar{L}$ in BPP.
- Consider $\mathcal{M}$ a PTM for a language L in BPP. By definition, we have $c \in \mathbb{N}$, such that $T_M(x,r) \leq |x|^c$, for all r of length lower than $|x|^c$. Let x be a word and $n = |x|$. There are $Max(x) = |\Sigma|^{n^c}$ different r to test if r is written on the finite alphabet Σ. We use the following pseudocode:

```
Simulation(x):
        let nacc = 0
        let nrej = 0
        for r = 0 to Max(x) - 1 do
                res = Execute M(x,r)
                if (res)
                then nacc ++
                else nrej ++
                end if
        endfor
        return (nacc > nrej)
```

  The values r,$nacc$ and $nrej$ have a (bit) length lower than n^c. Moreover, by definition, executing $M(x,r)$ takes polynomial time, so a fortiori, also polynomial space. It follows that $L \in \mathsf{PSPACE}$.

Exercise 4 Probabilistic Logarithmic Space

Propose a definition of $\mathsf{RSPACE}(f(n), p_{acc}, p_{rej})$.
Consider $\mathsf{RL} = \bigcup_{k \in \mathbb{N}} \mathsf{RSPACE}(k \cdot log(n), 0, 1/2)$ the class of languages that can be decided in probabilistic logarithmic space (the machine does not necessarily halt).

Show that:

1. Consider L in RL and $\mathcal{M}$ a probabilistic Turing machine which decides L. If $x \notin L$, then $\forall r$, $M(x, r) \neq \top$
2. $\mathsf{RL} \subseteq \mathsf{NL}$
3. $\mathsf{RL} \subseteq \mathsf{RP}$

Solution:

Idea: What's difficult here is the possibility that the Machine doesn't stop. Moreover (for the same reason) we don't have a boundary on our random word so we have a probability on an infinite set (as in ZPP). For the definition we won't use the usual definition by contraposition because there is not two but three cases.

Proof: We define $\mathsf{RSPACE}(p(n), p_{acc}, p_{rej})$ as the class of all languages L such that there is a randomized Turing machine M, working in space $p(n)$, that terminates with probability 1, and such that:

- If $x \in L$, then $Pr[M(x, r) = \top] \geq p_{rej}$
- If $x \notin L$, then $Pr[M(x, r) = \bot] \leq p_{acc}$

There is no bound in the working tape.

1. Consider $x \notin L$ and assume that there exists r (on a finite alphabet Σ) such that $M(x, r) = \top$. Let $n \in \mathbb{N}$ be number of steps taken by the execution $M(x, r)$. Then for all infinite words w, we have $M(x, r_{\leq n} \cdot w) = \top$. It follows that $Pr[M(x, r) = \top] \geq 1/|\Sigma|^n > 0$. Hence the contradiction since $Pr[M(x, r) = \bot] < 1$.
2. $\mathsf{RL} \subseteq \mathsf{NL}$: Given $L \in \mathsf{RL}$, $\mathcal{M}$ a randomized Turing Machine which decides L, we build the non deterministic Turing machine $\mathcal{M}'$ which follows the execution of the machine $\mathcal{M}$ and, when a random bit is required, guesses it. In that case:
   - if $x \in L$ then $Pr[\mathcal{M}(x, r) = \top] \geq \frac{1}{2}$, so $(\exists r, \mathcal{M}(x, r) = \top)$ then $\mathcal{M}'(x) = \top$
   - if $x \notin L$ then $(\forall r, \mathcal{M}(x, r) = \bot)$ hence $\mathcal{M}'(x) = \bot$

   Therefore: $\mathcal{M}'$ recognize L. Moreover, the resulting NL machine runs in space $k \cdot \log(n)$ for some k, but may fail to terminate. As in the lectures, we know that any run of more than $a^{k \cdot \log(n)}$ steps (where a is the alphabet size) will visit the same configuration twice. So we can stop any run when it exceeds that number of steps, and reject. This requires a counter of size $k \cdot \log(n)$. Then $L \in \mathsf{NL}$.
3. $\mathsf{RL} \subseteq \mathsf{NL} \subseteq \mathsf{P} \subseteq \mathsf{RP}$.

We actually have that $\mathsf{RL} = \mathsf{NL}$: one can prove that a random walk on an undirected graph solves the reachability problem with high probability, and one can adapt this idea to directed graph, proving $\texttt{REACH} \in \mathsf{RL}$.

Exercise 5 Dunno Machine

Define a ?-probabilistic Turing machine as a probabilistic Turing machine that halts on all inputs but with three final states: an accepting state, a rejecting state and a dunno state. Given x an input and r a random tape content, we note $M(x,r) = \top$ (resp. $\perp$, resp. ?) if the computation of M on x with random tape r accepts (resp. rejects, resp. ends in the dunno state).

We define the probabilistic complexity class ?PP as follows:

$L \in ?\mathsf{PP}$ if and only if there exists a ?-probabilistic Turing machine M working in (worst case) time $p(n)$, with random tape size $p(n)$ (for some polynomial p) and such that:

- for all x, $Pr[M(x,r) = ?] \leq \frac{1}{2}$
- if $x \in L$ then $Pr[M(x,r) = \perp] = 0$
- if $x \notin L$ then $Pr[M(x,r) = \top] = 0$

How does this class relate to the classical probabilistic complexity classes?

Solution: The answer is ZPP, the idea is that the Dunno state can be rejected or accepted to simulate respectively RP or coRP. Moreover if you have the two machines (for RP and coRP) and if you ask to both of them the same question, you have a good probability to be sure of the answer, whatever it is, else you return a Dunno state.

- $? - \mathsf{PP} \subseteq \mathsf{RP}$:

  Given $\mathcal{M}$ a ?-PTM for a language L in $? - \mathsf{PP}$.

  We define $\mathcal{M}'$ which simulates M but rejects if the answer is ?. Therefore :

  - if $x \in L$ then $Pr[\mathcal{M}'(x,r) = \perp] \leq \frac{1}{2}$
  - if $x \notin L$ then $Pr[\mathcal{M}'(x,r) = \top] = 0$

  So $L \subseteq \mathsf{RP}$.

- $? - \mathsf{PP} \subseteq \mathsf{coRP}$:

  Given $\mathcal{M}$ a ?-PTM for a language L in $? - \mathsf{PP}$.

  We define $\mathcal{M}'$ which simulates M but accepts if the answer is ?. Therefore :

  - if $x \in L$ then $Pr[\mathcal{M}'(x,r) = \perp] = 0$
  - if $x \notin L$ then $Pr[\mathcal{M}'(x,r) = \top] \leq \frac{1}{2}$

  So $L \subseteq \mathsf{coRP}$.

- $\mathsf{ZPP} \subseteq ? - \mathsf{PP}$

  Consider $\mathcal{M}_1$ (for RP) and $\mathcal{M}_2$ (for $\mathsf{co} - RP$) for a language $L \in \mathsf{ZPP}$. We define $\mathcal{M}$ which simulates $\mathcal{M}_1$ and M_2 and accepts if $\mathcal{M}_1$ does, rejects if $\mathcal{M}_2$ does and ends in a dunno state otherwise.

  Therefore:

  - if $x \in L$ then $Pr[\mathcal{M}(x,r) = ?] \leq Pr[M_1(x,r) = \perp] \leq \frac{1}{2}$
  - if $x \notin L$ then $Pr[\mathcal{M}(x,r) = ?] \leq Pr[M_2(x,r) = \top] \leq \frac{1}{2}$

So, for all x $Pr[\mathcal{M}(x,r) = ?] \leq \frac{1}{2}$

Moreover:

- if $x \in L$ then $Pr[\mathcal{M}(x,r) = \bot] = Pr[\mathcal{M}_2(x,r) = \bot] = 0$
- if $x \notin L$ then $Pr[\mathcal{M}(x,r) = \top] = Pr[\mathcal{M}_1(x,r) = \top] = 0$