Complexité avancée - TD 7

Simon Halfon

November 2, 2016

This exercise sheet refers to the following definitions of probabilistic Turing machine:

Definition 1 (Probabilistic Turing machine) *A probabilistic Turing machine running in time $f(n)$ (and space $g(n)$) is a deterministic Turing machine having, besides the input and working tapes, a read-only extra tape of alphabet Σ, called the* random tape. *At each transition, the Turing machine moves right on the random tape. For every input x of size n and every content of size $f(n)$ of the random tape, the machine halts in at most $f(n)$ steps (and using space at most $g(n)$).*

Definition 2 $\mathsf{RTIME}(f(n), l(n), accerr(n), rejerr(n))$ *is the class of all languages L for which there exists a probabilistic Turing machine M running in time $f(n)$ and reading $l(n)$ bit in input such that:*

- *If $x \in L$ then $Pr[M(x,r) = \bot] \leq rejerr(n)$*
- *If $x \notin L$ then $Pr[M(x,r) = \top] \leq accerr(n)$*

where the probability is obtained by considering the random tape content r uniformly distributed over all possible $\Sigma^{f(n)}$ contents of size $f(n)$.

Exercise 1: 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 machine is required to halt in time at most n^c 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:

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

where $RT^E(n^c, e_a, e_r)$ 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:

- $E[T_M(x,r)] \leq n^c$;
- if $x \in L$ then $Pr[M(x,r)] = \bot] \leq e_r$;
- if $x \notin L$ then $Pr[M(x,r)] = \top] \leq e_a$.

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

Exercise 2: Alternative definition of probabilistic TM

1. Show that the notion of probabilistic Turing machines defined below is equivalent to the one given in introduction.

   A PTM is a non-deterministic Turing machine M with a fixed degree of non-determinism $K \geq 2$ (*i.e.* in each configuration, the machine has exactly K possible choices, not necessarily all bringing to distinct configurations).

   We associate a probability $1/K$ to each non-deterministic choice. In other words, in each configuration the machine chooses with equal probability which transition to follow, among the possible ones. The choices of the machine at any two different steps are assumed independent.

   To a run R of M, we assign the probability $Pr[R]$ that the machine makes a sequence of choices that produces the run R. The probability that M accepts the input x is given by:

   $$Pr[M(x) = \top] = \sum_{R \text{ accepting run of } M \text{ on } x} Pr[R]$$

   and the probability that it rejects x is $Pr[M(x) = \bot] = 1 - Pr[M(x) = \top]$.

   The class $\mathsf{RTIME}'(f(n), f(n), accerr(n), rejerr(n))$ is defined as the class of languages L for which there exists a PTM M running in time $f(n)$, such that: if $x \in L$ then $Pr[M(x) = \bot] \leq rejerr(n)$, and if $x \notin L$ then $Pr[M(x) = \top] \leq accerr(n)$.

2. In the lecture, we have restricted our attention to probabilistic Turing machines with random-tape alphabet $\Sigma = \{0, 1\}$. Show that in the simple case of RP, this is not a restriction. Only describe the equivalence between a 2-symbol alphabet and a 3-symbol alphabet: $\mathsf{RP}^{\{0,1\}} = \mathsf{RP}^{\{0,1,2\}}$.

3. One can also show that assuming uniform probability is not a restriction. More precisely, consider the following third variant of probabilistic Turing machines:

   Given $\rho \in]0, 1[$, a ρ-PTM is a PTM of degree of non-determinism 2, but such that the first choice is made with probability ρ and the second one with probability $1 - \rho$. Assuming ρ is computable in polynomial time, that is one can compute the i-th bit of ρ in time i^k for some constant k, show that:

   (a) A ρ-PTM can be simulated by a PTM in expected time $O(1)$.
   Hint: pick a number in]0,1[at random

   (b) Conversely, a PTM can be simulated by a ρ-PTM in expected time $O(\frac{1}{\rho(1-\rho)})$
   Hint: pick pairs (r_1, r_2) of random bits until $r_1 \neq r_2$

   Curiosity: if ρ is not computable, then a ρ-PTM can compute indecidable languages.

Exercise 3: BPP and PSPACE

Give a direct proof that $\mathsf{BPP} \subseteq \mathsf{PSPACE}$.

Exercise 4: Probabilistic Logarithmic Space

Propose a definition RSPACE.

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

Show that:

1. For any $0 < \varepsilon < 1$, $\mathsf{RL} = \cup_{k \in \mathbb{N}} \mathsf{RSPACE}(k \cdot log(n), \infty, 0, \varepsilon)$
2. $\mathsf{RL} \subseteq \mathsf{NL}$
3. $\mathsf{RL} \subseteq \mathsf{RP}$

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 $\texttt{REACH} \in \mathsf{RL}$. The reachability problem for undirected graph has since been proved to be in L, using derandomization techniques.

Exercise 5: Dunno machines

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. $\bot$, resp. ?) if the computation of M on x with random tape r accepts (resp. rejects, resp. ends in the dunno state).

Define the probabilistic complexity class ?PP as follows:

$L \in$?PP iff 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 $Pr[M(x, r) = \bot] = 0$
- if $x \notin L$ then $Pr[M(x, r) = \top] = 0$

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

Exercise 6: BPP-completeness

1. Show that the language $L = \{(M, x, 1^t) \mid M \text{ accepts on input } x \text{ in time at most } t\}$, where M is the code of a non-deterministic Turing machine, x an input of M and t a natural number, is NP-complete.
2. Let L be the language of words $(M, x, 1^t)$ where M designates the encoding of a probabilistic Turing machine and x a string on $M's$ alphabet such that M accepts x in at most t steps, for at least $2/3$ of the possible random tapes of size t.

   Is L BPP-hard? Is it in BPP ?

Exercise 7: NP and randomized classes

Show that if $\mathsf{NP} \subseteq \mathsf{BPP}$ then $\mathsf{NP} = \mathsf{RP}$.