# Complexité avancée
# TD 7

Cristina Sirangelo - LSV, ENS-Cachan

November 12, 2014

This TD refers to the following definitions of probabilistic Turing machine and class **RTIME**.

**Definition 0.1 (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 $\Sigma$, 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 0.2** **RTIME**$(f(n), f(n), accerr(n), rejerr(n))$ *is the class of all languages $L$ for which there exists a probabilistic Turing machine $M$ running in time $f(n)$ 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. Probabilistic Turing machines with and without stay-moves** Show that we get the same definition of the class **RTIME**$(f(n), f(n), accerr(n), rejerr(n))$ whether we allow the probabilistic Turing machine to have both right and stay-moves, or just right-moves, on the random tape.

**Exercise 2. Alternative definition of probabilistic TM** Show that the following is an alternative (equivalent) definition of probabilistic Turing machines and of the class **RTIME**$(f(n), f(n), accerr(n), rejerr(n))$.

A non-deterministic Turing machine $M$ with a fixed degree of non-determinism $K$ is a non-deterministic Turing machine which, in each configuration, has exactly $K$ possible non-deterministic choices (not necessarily all bringing to distinct configurations). A probabilistic Turing machine is a non-deterministic Turing machine $M$ with a fixed degree of non-determinism $K$, where we associate a probability $1/K$ to each non-deterministic choice. In other words, in each configuration the machine choses with equal probability which transition to follow, among the possible ones. The choices of the machine at any two different steps are assumed independent.

This implies that each run $R$ of $M$ can be assigned a probability $Pr[R]$ : the probability that the machine makes a sequence of choices that produces the run $R$.

Define the probability that $M$ accepts the input $x$ as

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

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

The class $\mathbf{RTIME}(f(n), f(n), accerr(n), rejerr(n))$ is defined as the class of languages $L$ having a probabilistic Turing machine $M$ (as defined here), 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 $Pr[M(x) = \top] \leq accerr(n)$.

**Exercise 3. BPP and PSPACE** Give a direct proof that $\mathbf{BPP} \subseteq \mathbf{PSPACE}$.

**Exercise 4. BPL** Let $\mathbf{BPL}$ be the class of languages $L$ having a probabilistic Turing machine running in polynomial time and logarithmic space (intended as working space, not including the random tape) such that :

if $x \in L$ then $Pr[M(x, r) = \bot] \leq \frac{1}{3}$,
if $x \notin L$ then $Pr[M(x, r) = \top] \leq \frac{1}{3}$

Show that $\mathbf{BPL} \subseteq \mathbf{P}$.

**Exercise 5. Expected running time** Given a probabilistic Turing Machine $M$, not necessarily halting on all inputs, let $T_M(x, r)$ be the random variable describing the running time of $M$ on input $x$ and random tape $r$. 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. Let $T_M(x, r) = +\infty$ if $M$ does not halt on $x, r$.

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

Consider the definitions of $\mathbf{RP}$ and $\mathbf{BPP}$ : 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 $\mathbf{RP}^E$ and $\mathbf{BPP}^E$ as $\mathbf{RP}$ and $\mathbf{BPP}$, but replacing the worst case running time with the expected running time.

Formally,

$\mathbf{RP}^E = \bigcup_{c \in \mathbb{N}} \mathbf{RTIME}^E(n^c, 0, 1/2)$ and

$\mathbf{BPP}^E = \bigcup_{c \in \mathbb{N}} \mathbf{RTIME}^E(n^c, 1/3, 1/3)$,

where $\mathbf{RTIME}^E(n^c, e_a, e_r)$ is the class of languages $L$ having a probabilistic Turing machine $M$ (which may not halt) such that, for each input $x$ of size $n$ :

– $Pr[M(x, r) \text{ does not halt }] = 0$ ;

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

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

**Exercise 6. BPP-completeness** Let $L$ be the language consisting of all $\langle M, x, 1^t\rangle$ – where $M$ is a probabilistic TM, $x$ is an input for $M$, and $t$ is a natural number – 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. Class** $?PP$ Define ?-probabilistic Turing machines as probabilistic Turing machines that halt on all inputs and, when halting on input $x$ and random tape content $r$, may have three different types of outcome : $M(x,r) = \top$ (accept), $M(x,r) = \perp$ (reject) and $M(x,r) =?$ (don't know).

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 $Pr[M(x,r) = \perp] = 0$
– if $x \notin L$ then $Pr[M(x,r) = \top] = 0$

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