Exercise 1 RP*

We define RP* as the class of all languages L for which there exists a probabilistic Turing machine M running in polynomial time, such that:

- If $x \in L$ then $Pr[M(x, r) \text{ reject}] < 1$
- If $x \notin L$ then $Pr[M(x, r) \text{ accept}] = 0$

Do you recognize this class?

Solution:
This is in fact NP.

- RP* \subseteq NP: For the same reason than RP \subseteq NP

- NP \subseteq RP*: Let us show that SAT \in RP*. Let M be the probabilistic Turing machine that, on a formula $\phi$ with $p$ free variable, and $r$ a random tape of bits (of length $\geq p$), evaluates $\phi$ on $r$. We have that $M$ runs in polynomial time. In addition, if we denote by $eval(\phi)$ the proportion of valuations that satisfy $\phi$, we have $Pr[M(\phi, r) = \top] = eval(\phi)$ and $Pr[M(\phi, r) = \bot] = 1 - eval(\phi)$. Therefore:
  - If $\phi \in \text{SAT}$, $eval(\phi) > 0$ and we have $Pr[M(\phi, r) = \bot] < 1$.
  - If $\phi \notin \text{SAT}$, $eval(\phi) = 0$ and $Pr[M(\phi, r) = \top] = 0$.

It follows that SAT \in RP*. As RP* is closed under logspace reduction, we have NP \subseteq RP*.

Exercise 2 BPP and oracle machines

Prove that P^{BPP} = BPP.

Solution:

- BPP \subseteq P^{BPP}: This is straightforward, since one can ask the oracle the answer.

- P^{BPP} \subseteq BPP: Let $L \in P^{BPP}$. By definition, there exists $B \in BPP$ and $M$ a TM (of execution time lower than a polynomial $p$) which decides $L$ by calling the oracle $B$. We know that for all polynomial $q$, there exists a probabilistic Turing machine $M_q$ running in polynomial time which decides $B$ with a two-sided error lower than $2^{-q(n)}$. Consider now the probabilistic TM $M'$ that executes $M$ and
will define the
know that, for all polynomial
polynomial time which recognizes
In any case, we have
•
1. Show that the language
where
a probabilistic Turing machine and
accepts the input
is
is
This problem is in
•
1. Let now
be the language of words
where
accepts
in at most
steps, for at least
of the possible random tapes of size
Is
? Is it in
1. Let
be the code of a non-deterministic Turing machine, an input of
and
a natural number, is
complete.
2. Let
be the language of words
where
designates the encoding of a probabilistic Turing machine and
a string on
’s alphabet such that
accepts
in at most
steps, for at least
of the possible random tapes of size

Exercise 3 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
is the code of a non-deterministic Turing machine, an input of
and
a natural number. Notice that the timeout
we set for the execution of
is lower than the length of
.
So the algorithm which simulates
on
is non-deterministic and runs in polynomial time. Then we can check that
\[ (M, x, 1^t) \in \{ (M, x, 1^t) \mid M \text{ accepts on input } x \text{ in time at most } t \} \]
Therefore, \( L \in \text{NP} \).

Exercise 4 NP and randomized classes
Show that if \( \text{NP} \subseteq \text{BPP} \) then \( \text{NP} = \text{RP} \).

Solution:
In any case, we have \( \text{RP} \subseteq \text{NP} \). Let’s now assume that \( \text{NP} \subseteq \text{BPP} \). So \( \text{SAT} \in \text{BPP} \). We know that, for all polynomial \( q \), we have \( M \) a probabilistic Turing machine running in polynomial time which recognizes \( \text{SAT} \), with an error lower than or equal to \( 2^{-\Omega(n)} \). We will define the \( M' \) a PTM which works as the following pseudocode:
Input: \( \phi \) a formulae with \( p \) free variables; \( r \) randoms ; \( r' \) randoms

\[ \psi := \phi; \]
for \( i < p \) do
  if \( M(\psi[x_i = \top], r_i) = \top \) then
    \( \psi := \psi[x_i = \top] \)
  else
    if \( M(\psi[x_i = \bot], r'_i) = \top \) then
      \( \psi := \psi[x_i = \bot] \)
    else
      return \( \bot \)
  end
end
return \( \psi \) is satisfied

Notice that \( p < |\phi| \). There is a at most 2\( p \) calls to \( M \). Hence, the running time of this algorithm is polynomial and total length of random word used is also polynomial.

Therefore, for \( \phi \) a formulae with \( p \) free variables, \( |\phi| = n \) and \( x = 2^{-q(n)} \):

- if \( \phi \notin L \) then \( Pr[M'(\phi, r) = \top] = 0 \) (since we check that the last \( \psi \) is satisfied, which implies that the valuation chosen satisfies \( \phi \)).
- if \( \phi \in L \) then \( Pr[M'(\phi, r) = \bot] \leq \sum_{i=0}^{2p-1}(1-x)^ix \) (it’s the probability that one simulation of \( M \) fails). That is, \( Pr[M'(\phi, r) = \bot] \leq \sum_{i=0}^{2p-1} x = 2p \cdot x = 2n \cdot 2^{-q(n)} \leq 2^{2n-q(n)} \).

So, with \( q(n) = 2n + 1 \) : if \( \phi \in L \) then \( Pr[M'(\phi, r) = \bot] \leq \frac{1}{2} \)

Then: \( \text{SAT} \in \text{RP} \).

Exercise 5 The PP class

The class \( \text{PP} \) is the class of languages \( L \) for which there exists a polynomial time probabilistic Turing machine \( M \) such that:

- if \( x \in L \) then \( Pr[M(x, r) \text{ accepts}] > \frac{1}{2} \)
- if \( x \notin L \) then \( Pr[M(x, r) \text{ accepts}] \leq \frac{1}{2} \)

Also define \( \text{PP}_< \) as the class of languages \( L \) for which there exists a polynomial time probabilistic Turing machine \( M \) such that:

- if \( x \in L \) then \( Pr[M(x, r) \text{ accepts}] > \frac{1}{2} \)
- if \( x \notin L \) then \( Pr[M(x, r) \text{ accepts}] < \frac{1}{2} \)

1. Show that \( \text{BPP} \subseteq \text{PP} \) and \( \text{NP} \subseteq \text{PP} \);
2. Show that \( \text{PP} = \text{PP}_< \) and that \( \text{PP} \) is closed under complement;
3. Consider the decision problem \( \text{MAJSAT} \):
   (a) Input: a boolean formula \( \phi \) on \( n \) variables
   (b) Output: the (strict) majority of the \( 2^n \) valuations satisfy \( \phi \).
Show that $\text{MAJSAT} \in \text{PP}$. In fact, $\text{MAJSAT}$ is PP-complete.

One may also consider the decision problem $\text{MAXSAT}$:

(a) Input: a boolean formula $\phi$ on $n$ variables, a number $K$

(b) Output: more than $K$ valuations satisfy $\phi$.

Show that $\text{MAXSAT}$ is also PP-complete (to prove that $\text{MAXSAT} \in \text{PP}$ one may reduce $\text{MAXSAT}$ to $\text{MAJSAT}$).

Solution:

1. • A language $L \in \text{BPP}$ is recognized by a PTM $M$ such that if $x \in L$ then $\Pr[M(x,r) \text{ accepts}] \geq \frac{2}{3}$, and if $x \notin L$ then $\Pr[M(x,r) \text{ accepts}] \leq \frac{1}{3}$. It follows that $L \in \text{PP}$.

• The class $\text{PP}$ is closed under logspace reduction. It suffice to show that $\text{SAT} \in \text{PP}$. Consider now a probabilistic Turing machine with an input that is a formula $\phi$. According to the first bit of the random tape, it either accepts or reads what remains of the random tape for a valuation and accepts if and only if it satisfies $\phi$. Then, if $\phi \in \text{SAT}$, we have $\Pr[M(x,r) \text{ accepts}] > \frac{1}{2}$, otherwise $\Pr[M(x,r) \text{ accepts}] = \frac{1}{2}$.

2. Trivially, we have $\text{PP}_e \subseteq \text{PP}$. Now, consider $L \in \text{PP}$ and its associated Turing machine $M$ running in polynomial time $p$. Without loss of generality, we assume that the alphabet of the random tape is of size 2, hence the probability of a random word for $M$ on an input $x$ such that $|x| = n$ is $2^{-p(n)}$. Therefore, if $x \in L$ then $\Pr[M(x,r) \text{ accepts}] \geq \frac{1}{2} + \frac{1}{2^p(n)}$. Now, we construct another Turing machine $M'$ that runs $M$ on an input. If $M$ would reject, $M'$ rejects too, and if $M$ would accept then $M'$ rejects with probability $\frac{1}{2^p(n)}$ (for instance, by reading a word in the random tape of length $p(n)$ and accepting only if there are only $0$s). Then:

- if $x \in L$: $\Pr[M(x,r) \text{ accepts}] \geq \left(\frac{1}{2} + \frac{1}{2^p(n)}\right) \cdot \left(1 - \frac{1}{2^p(n)}\right) = \frac{1}{2} + \frac{1}{2^p(n)+1} - \frac{1}{2^{2p(n)}} > \frac{1}{2}$
- if $x \notin L$: $\Pr[M(x,r) \text{ accepts}] \leq \left(\frac{1}{2}\right) \cdot \left(1 - \frac{1}{2^p(n)}\right) < \frac{1}{2}$

That is, $L \in \text{PP}_e$. The stability under complement then follows by inverting the accepting and rejecting states.

3. A probabilistic Turing machine that checks that a valuation read on the random tape satisfies the formula decides $\text{MAJSAT}$ for $\text{PP}$. Then, $\text{MAJSAT}$ can be reduced to $\text{MAXSAT}$ in logarithmic (as one has to write on the output tape the number $2^{n-1} + 1$ in binary, which consists in a 1, $n - 2$ 0s and then a 1). Therefore, $\text{MAXSAT}$ is also PP-hard. Let us now show that $\text{MAXSAT} \in \text{PP}$. To do so, let us reduce $\text{MAXSAT}$ to $\text{MAJSAT}$. Consider an instance $(\phi, i)$ of $\text{MAXSAT}$ with $0 \leq r_1 < r_2 < \ldots < r_k \leq n$ such that $2^n - i = 2^{n-r_1} + \ldots + 2^{n-r_k}$ (the values $n - r_j$ refers to the 1s in the binary decomposition of $2^n - i$). Let us denote $x_1, \ldots, x_n$ the variables of $\phi$. Then, we consider the formula $\psi$ as:

$$\psi = (x_1 \land \ldots \land x_{r_1}) \lor (\neg x_1 \land \ldots \land \neg x_{r_1} \land x_{r_1+1} \land \ldots \land x_{r_2}) \lor \ldots \lor (\neg x_1 \land \ldots \land \neg x_{r_{k-1}} \land x_{r_{k-1}+1} \land \ldots \land x_{r_k})$$
We can see there are exactly $2^{n-r_j}$ valuations satisfying the $j$-th line of $\psi$. With the negation at beginning of the lines, no valuation satisfies two lines of $\psi$. Therefore, there are exactly $2^{n-r_1} + \ldots + 2^{n-r_k} = 2^n - i$ valuations satisfying $\psi$. Consider now a fresh variable $y$ and the formula: $\phi' = (y \land \phi) \lor \neg (y \land \psi)$. Then, we have $\phi'$ computable in polynomial time from $\phi$ and $\phi$ is satisfied by more than $i$ valuations if and only if $\phi'$ is satisfied by more than half of valuations, i.e. $\phi \in \text{MAXSAT} \iff \phi' \in \text{MAJSAT}$. 