Complexité avancée - TD 10

Benjamin Bordais

December 8, 2021

We recall the definition of the Arthur-Merlin hierarchy.

Definition. *An* Arthur and Merlin triplet *is the data of* $(M, \mathcal{A}, D)$ *where* M *is a Merlin function, that is a function with the size of the output polynomial in the size of the input, possibly not computable, a randomized Turing machine* $\mathcal{A}$ *running in polynomial time and a language* $D \in \mathsf{P}$*. Then, for all* $w \in \{\mathsf{A}, \mathsf{M}\}^*$*, let us denote by* k *the number of times* A *appears in the word* w*. We consider the following algorithm induced by the word* w *(with* $n = |w|$ *and* $r_1, \ldots, r_k$ k *random tapes of size polynomial in* n*).*

$prot_w(M; x, r_1, \ldots, r_k)$:
$imp = x$
$i = 0$
for $j = 1, \ldots, n$:
if $w_j = \mathsf{A}$ ***then*** $(i = i+1,\ q_j = \mathcal{A}(imp, r_i);\ imp = imp \ \#\ r_i \ \#\ q_j)$
else $(y_j = M(imp);\ imp := imp \ \#\ y_j)$
accept if $(imp \in D)$, ***else reject***

We denote $prot[\mathcal{A}, M]_D(x, r_1, \ldots, r_k) = \top$ *if the previous algorithm accepts, otherwise* $prot[\mathcal{A}, M]_D(x, r_1, \ldots, r_k) = \bot$.

Now, $\mathsf{AM}[f]$ *for a proper function* f *denotes the class of languages* L *such that for any polynom* q*, there exists an Arthur and Merlin triplet* $(M, \mathcal{A}, D)$ *such that for any* x *of size* n*, letting* $w \in \{\mathsf{A}, \mathsf{M}\}^{f(n)}$*:*

1. *Completeness: if* $x \in L$ *then* $Pr[prot_w[\mathcal{A}, M]_D(x, r_1, \ldots, r_k) = \top] \geq 1 - 1/2^{q(n)}$
2. *Soundness: if* $x \notin L$ *then for any Merlin's function* M'*,* $Pr[prot_w[\mathcal{A}, M']_D(x, r_1, \ldots, r_k) = \bot] \geq 1 - 1/2^{q(n)}$

Exercise 1 (A quick exercise)**.** *Realize that* AM *and* MA *could be alternatively defined in the following way:*

- *A language* $L \in \mathbf{AM}$ *if and only if for all polynom* q *there exists a language* $D \in \mathsf{P}$ *and a polynom* p *such that:*
  - $x \in L \Rightarrow Pr_{r \in \{0,1\}^{p(|x|)}}[\exists y \in \{0,1\}^{p(|x|)},\ (x, r, y) \in D] \geq 1 - 1/2^{q(n)}$
  - $x \notin L \Rightarrow Pr_{r \in \{0,1\}^{p(|x|)}}[\exists y \in \{0,1\}^{p(|x|)},\ (x, r, y) \in D] \leq 1/2^{q(n)}$
- *A language* $L \in \mathbf{MA}$ *if and only if for all polynom* q *there exists a language* $D \in \mathsf{P}$ *and a polynom* p *such that:*
  - $x \in L \Rightarrow \exists y \in \{0,1\}^{p(|x|)},\ Pr_{r \in \{0,1\}^{p(|x|)}}[(x, r, y) \in D] \geq 1 - 1/2^{q(n)}$
  - $x \notin L \Rightarrow \forall y \in \{0,1\}^{p(|x|)},\ Pr_{r \in \{0,1\}^{p(|x|)}}[(x, r, y) \in D] \leq 1/2^{q(n)}$

Solution 1. *This follows directly from the definition of the Arthur-Merlin hierarchy with the use of majority voting.*

Exercise 2 (Arther-Merlin protocols)**.** *Prove the following statements, directly from definition of the Arthur-Merlin hierarchy:*

- $\mathbf{M} = \mathsf{NP}$*;*
- $\mathbf{A} = \mathsf{BPP}$*;*
- $\mathsf{NP}^{\mathsf{BPP}} \subseteq \mathbf{MA}$*;*
- $\mathbf{AM} \subseteq \mathsf{BPP}^{\mathsf{NP}}$*.*

Solution 2.

- *Notice that for a language L:*

  $L \in \mathbf{M} \Leftrightarrow \exists D \in \mathsf{P}, \exists p \text{ poly}, L = \{x \mid \exists y, |y| < p(|x|) \wedge x\#y \in D\}$. This corresponds exactly to the certificate definition of NP .

- *Obvious, just have to write the two definitions:*

$\mathsf{BPP} \subseteq \mathbf{A}$ *: For $L \in \mathsf{BPP}$ with the machine $\mathcal{M}$ associated as in the definition of BPP, consider the language $D = \Sigma^*\#\Sigma^*\#\top \in \mathsf{P}$, and $\mathcal{A}$ which simulates the machine $\mathcal{M}$ and write the answer.*

$\mathbf{A} \subseteq \mathsf{BPP}$ *: We can just simulate $\mathcal{A}$ and check (in polynomial time) that it is in D.*

- *Let $L \in \mathsf{NP}^{\mathsf{BPP}}$, then there exists a polynom p and a language $L' \in \mathsf{P}^{\mathsf{BPP}}$ such that $L = \{x \mid \exists y,\ |y| \leq p(|x|),\ x\#y \in L'\}$. Moreover we know from the previous TD that $\mathsf{P}^{\mathsf{BPP}} = \mathsf{BPP}$, and from the previous answer that $\mathbf{A} = \mathsf{BPP}$. Therefore we have $L' \in \mathbf{A}$ such that $L = \{x \mid \exists y, x\#y \in L'\}$. That is, $L \in \mathbf{MA}$.*

- *Let L be in $\mathbf{AM}$, we have $(M, \mathcal{A}, D)$ given by the definition of $\mathbf{AM}$, with an error at $\varepsilon > 0$. Define A' the probabilistic Turung machine s.t. for an input x and a random word r, $\mathcal{A}'(x, r) = x\#r\#\mathcal{A}(x, r)$. Moreover, consider a polynom p bounding the size of the output of the Merlin functions considered (in particular M) and define $D' = \{x \mid \exists y,\ |y| \leq p(|x|),\ x\#y \in D\} \in \mathsf{NP}$ since $D \in \mathsf{P}$. Let M_o be the probabilistic oracle machine which simulates $\mathcal{A}'$ and call the oracle for the language D' on the answer, accepting with the BPP way. It follows that:*

  - *If $x \in L$, $Pr[M_o(x, r) = \top] = Pr[x\#r\#A(x, r) \in D'] = Pr[\exists y,\ |y| \leq p(|x|),\ x\#r\#A(x, r)\#y \in D] \geq 1 - \varepsilon$ (it's the definition of $\mathbf{AM}$)*
  - *If $x \notin L$, $Pr[M_o(x, r) = \bot] = Pr[x\#r\#A(x, r) \notin D'] = Pr[\forall y,\ |y| \leq p(|x|),\ x\#r\#A(x, r)\#y \notin D] = 1 - Pr[\exists y,\ |y| \leq p(|x|),\ x\#r\#A(x, r)\#y \in D] \geq 1 - \varepsilon$*

  Then $L \in \mathsf{BPP}^{\mathsf{NP}}$

Exercise 3 (The BP operator)**.** *We say that a language B reduces to language C under a randomized polynomial time reduction, denoted $B \leq_r C$, if there is a probabilistic polynomial-time Turing machine $\mathcal{M}$ such that for every x, $Pr[\mathcal{M}(x) \in C \Leftrightarrow x \in B] \geq \frac{2}{3}$.*

Recall the definition of $\mathsf{BP}\cdot\mathcal{C}$: $L \in \mathsf{BP}\cdot\mathcal{C}$ iff there exists a probabilistic Turing machine A running in polynomial time and a language $D \in \mathcal{C}$ s.t. for all input x:

- *if $x \in L$ then $Pr[A(x, r) \in D] \geq \frac{2}{3}$*

- *if* $x \notin L$ *then* $Pr[A(x,r) \notin D] \geq \frac{2}{3}$

1. *Show that* $\mathsf{BP} \cdot \mathcal{C} = \{L \mid L \leq_r L',\ \textit{for some } L' \in \mathcal{C}\}$
2. *Show that* $\mathsf{co}(\mathsf{BP} \cdot \mathcal{C}) = \mathsf{BP} \cdot \mathsf{co}(\mathcal{C})$ *and if* $\mathcal{C} \subseteq \mathcal{C}'$, *then* $\mathsf{BP} \cdot \mathcal{C} \subseteq \mathsf{BP} \cdot \mathcal{C}'$
3. *Show that* BPP *is closed under randomized polynomial time reduction.*
4. *Give a criterion on* $\mathcal{C}$ *so that:* $\mathsf{BP} \cdot (\mathsf{BP} \cdot \mathcal{C}) = \mathsf{BP} \cdot \mathcal{C}$.

The class $\mathsf{BP} \cdot \mathsf{NP}$

1. *Show that* $\mathsf{BP} \cdot \mathsf{P} = \mathsf{BPP}$
2. *Recall the proof that* $\mathsf{BP} \cdot \mathsf{NP} = \mathbf{AM}$
3. *Show that* $\mathsf{BP} \cdot \mathsf{NP} = \{L \mid L \leq_r \mathsf{SAT}\}$
4. *Show that* $\mathsf{BP} \cdot \mathsf{NP} \subseteq \Sigma_3^P$ *(with a direct proof)*
5. *Show that* $\mathsf{BP} \cdot \mathsf{NP} \subseteq \mathsf{NP}/poly$ *where* $\mathsf{NP}/poly$ *is defined analogously to* $\mathsf{P}/poly$ *(with advice string)*
6. *(Bonus) Show that if* $\overline{\mathsf{3SAT}} \leq_r \mathsf{3SAT}$ *then* PH *collapses to the third level.*

Solution 3. 1. *This comes directly from the definition of* $\mathsf{BP} \cdot \mathcal{C}$.

2. *For a language* L, *we have:*

$$\begin{aligned} L \in \mathsf{co}(\mathsf{BP} \cdot \mathcal{C}) &\Leftrightarrow \bar{L} \in \mathsf{BP} \cdot \mathcal{C} \\ &\Leftrightarrow \exists L' \in \mathcal{C},\ \bar{L} \leq_r L' \\ &\Leftrightarrow \exists L' \in \mathcal{C},\ \exists \textit{ a PTM } \mathcal{M},\ Pr_r[x \in \bar{L} \Leftrightarrow \mathcal{M}(x,r) \in L'] \geq 2/3 \\ &\Leftrightarrow \exists L' \in \mathcal{C},\ \exists \textit{ a PTM } \mathcal{M},\ Pr_r[x \in L \Leftrightarrow \mathcal{M}(x,r) \in \bar{L'}] \geq 2/3 \\ &\Leftrightarrow \exists L' \in \mathcal{C},\ L \leq_r \bar{L'} \\ &\Leftrightarrow \exists L'' \in \mathsf{co}\mathcal{C},\ L \leq_r L'' \\ &\Leftrightarrow L \in \mathsf{BP} \cdot \mathsf{co}(\mathcal{C}) \end{aligned}$$

The second fact is straightforward.

3. *Let* $B \in \mathsf{BPP}$, *we know that we can have* M_B *a PTM which decides* B *with an error lower than* $\frac{1}{12}$. *Let* $C \leq_r B$, *we have* M *a probabilistic polynomial-time Turing machine such that for every* x, $Pr[C(M(x,r)) = B(x)] \geq \frac{2}{3}$. *Let* M_C *be the PTM which simulates, for an input* x *and two random words* r *and* r', $M_B(M(x,r'),r)$. *Then:*

   - *If* $x \in C$, $P[M_C(x,r) = \bot] = P_{(r',r)}[M_B(M(x,r'),r) = \bot] \leq P_{r'}[C(M(x,r')) \neq B(x)] + P_r[M_B(y,r) = \bot \mid y \in B] \leq \frac{1}{3} + \frac{1}{12} \leq \frac{5}{12}$
   - *If* $x \notin C$, $P[M_C(x,r) = \top] = P_{(r',r)}[M_B(M(x,r'),r) = \top] \leq P_{r'}[C(M(x,r')) \neq B(x)] + P_r[M_B(y,r) = \top \mid y \notin B] \leq \frac{1}{3} + \frac{1}{12} \leq \frac{5}{12}$

   Therefore $C \in \mathsf{BPP}$

4. *Having* $\mathsf{BP} \cdot (\mathsf{BP} \cdot \mathcal{C}) = \mathsf{BP} \cdot \mathcal{C}$ *amounts to show that if* $L \leq_r L' \leq_r L''$ *then* $L \leq_r L''$ *for all* L, L', *and* L''. *To prove this result, one may want to compose both reductions, but by doing so, we would only obtain a probability of not making a mistake above* $2/3 \times 2/3 = 4/9 < 2/3$. *If we assume that* $\mathcal{C}$ *is democratic, then one can use also here majority voting to increase the threshold* $2/3$ *arbitrarily close to 1 while keeping a polynomial algorithm. Then, if the error rate of both reductions if less than* $1/10$, *then the error rate of the composition is less than* $1 - 9/10 * 9/10 = 19/100 \leq 1/3$.

1. *This is straightforward, for instance for the inclusion* $\supseteq$:

   Let $L \in \mathsf{BPP}$, *we have* A *a PTM given by the definition of* BPP, *we just simulate it with* A' *which won't accept or reject but will write* $\top$ *or* $\bot$. *Let* $D = \{\top\}$. *By definition of* BPP:

   - *If* $x \in L$, $P[A(x,r) \in D] = P[A'(x,r) = \top] \geq \frac{2}{3}$
   - *If* $x \notin L$, $P[A(x,r) \notin D] = P[A'(x,r) = \bot] \geq \frac{2}{3}$

   Therefore $\mathsf{BPP} \subseteq \mathsf{BP} \cdot \mathsf{P}$.

2. *The solution is in the course p.31, and it's the same construction that* $\mathsf{AM} \subseteq \mathsf{BPP}^{\mathsf{NP}}$.

3. *By definition, we have* $\mathsf{BP} \cdot \mathsf{NP} \supseteq \{L \mid L \leq_r \mathsf{SAT}\}$ *since* $\mathsf{SAT} \in \mathsf{NP}$. *Let us now prove that for every languages* L, L', L'', *we have that if* $L \leq_r L' \leq_l L''$ *then* $L \leq_r L''$. *Consider the probabilistic Turing machine* $\mathcal{M}$ *for the first reduction (i.e.* $Pr[\mathcal{M}(x) \in L' \Leftrightarrow x \in L] \geq \frac{2}{3}$*), and the Turing machine* $\mathcal{M}'$ *running logarithmic space for the second reduction (i.e.* $\mathcal{M}'(\mathcal{M}(x)) \in L'' \Leftrightarrow \mathcal{M}(x) \in L'$*). We consider the probabilistic Turing machine* $\mathcal{M}''$ *running in polynomial time that, on an input* x, *computes* $\mathcal{M}''(x) = \mathcal{M}'(\mathcal{M}(x))$ *(note that* $|\mathcal{M}(x)| \leq p(|x|)$ *for some polynom* p*). Then, we have* $\mathcal{M}''(x) \in L'' \Leftrightarrow \mathcal{M}'(\mathcal{M}(x)) \in L'' \Leftrightarrow \mathcal{M}(x) \in L'$. *Hence,* $Pr[\mathcal{M}''(x) \in L'' \Leftrightarrow x \in L] = Pr[\mathcal{M}(x) \in L' \Leftrightarrow x \in L] \geq \frac{2}{3}$. *It follows that for* $L \in \mathsf{BP} \cdot \mathsf{NP}$, *there exists* $L' \in \mathsf{NP}$ *such that* $L \leq_r L'$ *with* $L' \leq_l \mathsf{SAT}$ *since* SAT *is* NP*-complete. It follows that* $L \leq_r \mathsf{SAT}$ *and* $\mathsf{BP} \cdot \mathsf{NP} \subseteq \{L \mid L \leq_r \mathsf{SAT}\}$.

4. *We proceed very similarly to the proof that* $\mathsf{BPP} \subseteq \Sigma_2^p$. *That is, consider* $L \in \mathsf{BP} \cdot \mathsf{NP}$ *decided with error* $1/2^n$ *in polytime* $p(n)$ *by a probabilistic Turing machine* $\mathcal{M}$. *For* $x \in \Sigma^*$, *we denote* $R_x = \{r \in \{0,1\}^{p(n)} \mid \mathcal{M}(x,r)$ *accepts*$\}$. *We use the facts:*

   - *If* $R_x \geq (1 - 1/2^n) \cdot 2^{p(n)}$, *then there exists* $t_0, \ldots t_{p(n)/n}$ *such that* $R \oplus t_0, \ldots, R \oplus t_{p(n)/n}$ *covers* $\{0,1\}^{p(n)}$;
   - *If* $R_x \leq (1/2^n) \cdot 2^{p(n)}$, *then for all* $t_0, \ldots t_{p(n)/n}$ *such that* $R \oplus t_0, \ldots, R \oplus t_{p(n)/n}$ *does not cover* $\{0,1\}^{p(n)}$.

   We get that for $L \in \mathsf{BP} \cdot \mathsf{NP}$, *we have* $\mathcal{M}$ *a non-deterministic Turing machine running in polynomial time and* q *a poly. such that:* $x \in L \Leftrightarrow \exists t_0 ... t_{q(n)/n}\ \forall r \in \{0,1\}^{q(n)} \bigvee_{i \leq q(n)/n} \mathcal{M}(x, r \oplus t_i)$. *The only difference wth the case* BPP *is that the Turing machine* $\mathcal{M}$ *is non-deterministic. Therefore, we get that* $L \in \Sigma_3^P$.

   In fact $\mathsf{BP} \cdot \Sigma_i^P \subseteq \Sigma_{i+2}^P$ *for all* $i \geq 0$.

5. *Similar to the proof that* $\mathsf{BPP} \subseteq \mathsf{P}/poly$

Exercise 4 (The PP class)**.** *The class* 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* $Pr[M(x,r)\ accepts\ ] > \frac{1}{2}$
- *if* $x \notin L$ *then* $Pr[M(x,r)\ accepts\ ] \leq \frac{1}{2}$

1. *Consider the decision problem* MAXSAT*:*

   (a) *Input: a boolean formula* ϕ *on* n *variables, a number* K

   (b) *Output: more than* K *valuations satisfy* ϕ*.*

   Show that MAXSAT $\in$ PP*. In fact,* MAXSAT *is* PP*-complete.*

   One may also consider the decision problem MAJSAT*:*

   (a) *Input: a boolean formula* ϕ *on* n *variables*

   (b) *Output: the (strict) majority of the* 2^n *valuations satisfy* ϕ*.*

   Show that MAJSAT *is also* PP*-complete.*

2. *Show that* $\mathbf{MA} \subseteq$ PP*.*

Solution 4. *1. Let us first show that* MAXSAT *is in* PP*. First, if* $K \geq 2^{n-1}$*, we reject with probability* $1 - 2^{n-1}/K$ *and otherwise draw a valuation at random and accept if this valuation accepts. In that case, the probability of acceptance is at least* $1/2$ *if and only if ther are at least* $K/2^n$ *satisfying valuations. If* $K < 2^{n-1}$ *we proceed analogously with* $2^n - K$ *instead of* K *(up to reversing the roles of accept and reject).*

A probabilistic Turing machine that checks that a valuation read on the random tape satisfies the formula decides MAJSAT *for* PP*. Then,* MAJSAT *can be reduced to* 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,* MAXSAT *is also* PP*-hard. Let us now show that* MAXSAT $\in$ PP*. To do so, let us reduce* MAXSAT *to* MAJSAT*. Consider an instance* (ϕ, i) *of* 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* ϕ*. Then, we consider the formula* ψ *as:*

$$\begin{aligned}\psi = &(x_1 \wedge \ldots \wedge x_{r_1}) \\ &\vee (\neg x_1 \wedge \ldots \wedge \neg x_{r_1} \wedge x_{r_1+1} \wedge \ldots \wedge x_{r_2}) \\ &\vee \cdots \\ &\vee (\neg x_1 \wedge \ldots \wedge \neg x_{r_{k-1}} \wedge x_{r_{k-1}+1} \wedge \ldots \wedge x_{r_k})\end{aligned}$$

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

2. *Consider the characterization* $\mathbf{MA}$ *of exercise 1 with a bound equal to* $2/3$*. Let* $L \in \mathbf{MA}$ *and the corresponding Arthur-Merlin* $(M, \mathcal{A}, D)$*. Here, once* y *is fixed (which the result of the Merlin map* M *whose size is bounded by the polynom* p*), we can repeat the experience – that is, iterate* $36 \cdot q(|x|) \cdot \log(2)$ *calls to the Arthur probabilistic Turing machine – and use majority voting and the Chernoff bound to*

have the error rate below $1/2^{q(|x|)}$ *for all polynom* q*. This new probabilistic Turing machine works in polynomial time, specifically* $36 \cdot q(|x|) \cdot \log(2) \cdot p(|x|)$*, which is also the length of the random tape used by this new Turing machine on an input* (x, y) *of size* $|x| + p(|x|)$*. Let* $q = p + 2$*,* $u = 36 \cdot q \cdot \log(2)$*, and* $D'_x = \{(r_1, \ldots, r_{u(|x|)}, y) \mid |y| = p(|x|) \wedge \forall i \wedge |r_i| = p(|x|) \wedge$ *the majority of the* r_i *ensures* $(x, r_i, y) \in D\}$*. Note that deciding if* $(r, y) \in D'_x$ *can be done in polynomial time. Then,*

- $x \in L \Rightarrow \exists y \in \{0,1\}^{p(|x|)},\ Pr_{r \in \{0,1\}^{u(|x|) \cdot p(|x|)}}[(r, y) \in D'_x] \geq 1 - 1/2^{p(|x|)+2}$
- $x \notin L \Rightarrow \forall y \in \{0,1\}^{p(|x|)},\ Pr_{r \in \{0,1\}^{u(|x|) \cdot p(|x|)}}[(r, y) \in D'_x] \leq 1/2^{p(|x|)+2}$

Let us now consider the size of the set D'_x*. If* $x \in L$*, we have:*

$$\sum_{y \in \{0,1\}^{p(|x|)}} \sum_{r \in \{0,1\}^{u(|x|) \cdot p(|x|)}} [(r, y) \in D'_x] \geq 2^{u(|x|) \cdot p(|x|)} \cdot (1 - 1/2^{p(|x|)+2}) \geq 2^{u(|x|) \cdot p(|x|) - 1}$$

If $x \notin L$*, we have:*

$$\sum_{y \in \{0,1\}^{p(|x|)}} \sum_{r \in \{0,1\}^{u(|x|) \cdot p(|x|)}} [(r, y) \in D'_x] \leq 2^{p(|x|)} \cdot 2^{u(|x|) \cdot p(|x|)} \cdot (1/2^{p(|x|)+2}) = 2^{u(|x|) \cdot p(|x|) - 2}$$

Therefore, we have $x \in L \Leftrightarrow |D'_x| \geq 2^{u(|x|) \cdot p(|x|) - 1}$*.*

Consider now the following randomized polynomial time algorithm on an input $|x|$*: according to the first random bit, either it accepts with probability* $1 - 1/2^{p(|x|)+1}$ *(for instance by reading* $p(|x|) + 1$ *bits and rejecting iff all are 0s) or it chooses randomly (and uniformly) an instance* (r, y) *of* D'_x *and accepts if* $(r, y) \in D'_x$*. Then, if we denote by* p_x *the probability of accepting, it ensures:*

$$p_x = \frac{1}{2} \cdot (1 - \frac{1}{2^{p(|x|)+1}}) + \frac{1}{2} \cdot \frac{|D'_x|}{2^{p(|x|)+u(|x|) \cdot p(|x|)}}$$

Hence:

$$x \in L \Leftrightarrow \frac{|D'_x|}{2^{p(|x|)+u(|x|) \cdot p(|x|)}} \geq \frac{1}{2^{p(|x|)+1}} \Leftrightarrow p_x \geq \frac{1}{2}$$

It follows that $L \in \mathsf{PP}$*.*

Exercise 5 (A little come back to P and RP)**.** *We define a random language* A *by setting that each word* $x \in \{0,1\}^*$ *is in* A *with probability* $1/2$*. Show that almost surely (on the probabilistic choice on the language* A*) we have* $\mathsf{P}^A = \mathsf{RP}^A$*.*

Hint: Fix an $\epsilon > 0$ *and an enumeration* $(M_i)_{i \in \mathbb{N}}$ *of probabilistic Turing machine running in polynomial time with an oracle. Exhibit deterministic polynomial time Turing machines* $(N_i)_{i \in \mathbb{N}}$ *and consider the probability (over the random language considered) that there is one* i *such that* M_i *and* N_i *does not coincide*

Solution 5. *For all languages* A*, we have* $\mathsf{P}^A \subseteq \mathsf{RP}^A$*. Now, consider the other inclusion. Let* $(M_i)_{i \in \mathbb{N}}$ *be an enumeration of probabilistic Turing machine running in polynomial time with an oracle. Fix an* $\epsilon > 0$*. Let us denote by* $(M'_i)_{i \in \mathbb{N}}$ *the Turing machine that executes* $i + 2 \cdot n + \log \epsilon$ *(with* n *the size of the input) time the machine* M_i*, to get the probability of error* $\leq \epsilon \cdot 2^{-i-2n}$ *(if the machine* M_i *has a behavior as in* RP*) and let us denote by* t'_i *its execution time. Now, we consider the deterministic machine* N_i^A *that simulates the execution of the machine* $M_i'^A$ *by replacing random bits read by the call to the oracle machine on (arbitrary) words of size bigger that* t'_i *(therefore, which are not used by* $M_i'^A$*). Since* A *is random, so are the bits from the oracle and they also are*

independent. Note that all the machines N_i run in polynomial time. Then, if we consider M_i^A with the acceptance condition of type RP^A and which then recognizes the language $L(M_i^A)$, for all x of size n we have:

$$Pr_A[N_i^A(x) \neq [x \in L(M_i^A)]] \leq \epsilon \cdot 2^{-i-2n}$$

By summing over all such x:

$$Pr_A[\exists x \in \{0,1\}^n,\ N_i^A(x) \neq [x \in L(M_i^A)]] \leq \epsilon \cdot 2^{-i-n}$$

Now, we sum over all such n:

$$Pr_A[\exists x \in \{0,1\}^*,\ N_i^A(x) \neq [x \in L(M_i^A)]] \leq 2 \cdot \epsilon \cdot 2^{-i}$$

Finally, we sum over all such i:

$$Pr_A[\exists i,\ \exists x \in \{0,1\}^*,\ N_i^A(x) \neq [x \in L(M_i^A)]] \leq 4 \cdot \epsilon$$

Therefore, we have $Pr_A[\mathsf{RP}^A \subsetneq \mathsf{P}^A] \leq 4\epsilon$. This holds for all $\epsilon > 0$. That is, almost surely, P^A and RP^A coincide.