Complexité avancée - TD 9

Benjamin Bordais

December 09, 2020

We recall the definition of the Arthur-Merlin hierarchy.

Definition 1 *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, ..., r_k) :
imp = x
i = 0
for j = 1, ..., n:
        if w_j = A then (i = i+1, q_j = A(imp, r_i); imp = imp # r_i # q_j)
        else (y_j = M(imp); imp := imp # y_j)
accept if (imp ∈ D), else reject
```

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

Recall the definition of the Arthur-Merlin hierarchy: $\mathsf{AM}[f]$ *for a proper function* f *denotes the class of languages* L *such that 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* $Pr[prot_w[\mathcal{A}, M]_D(x, r_1, \ldots, r_k) = \top] \geq 2/3$
2. *Soundness: if* $x \notin L$ *then for any Merlin's function* M'*,* $Pr[prot_w[\mathcal{A}, M']_D(x, r_1, \ldots, r_k) = \bot] \geq 2/3$

Exercise 1 NP and BPP

- if $\mathsf{P} = \mathsf{NP}$ then $\mathsf{BPP} = \mathsf{P}$.
- if $\mathsf{NP} \subseteq \mathsf{BPP}$ then $\mathsf{AM} = \mathsf{MA}$ (you may use the fact (or even prove!) that $\mathsf{BPP}^{\mathsf{BPP}} = \mathsf{BPP}$).

Exercise 2 AM with perfect soundness

Define AM_{ps} as AM with perfect soundness, that is, in the case $x \notin L$, for all Merlin's function, the probability to reject is equal to 1. Show that $\mathsf{AM}_{ps} = \mathcal{C} \subseteq \mathsf{AM}$, where $\mathcal{C}$ is a known complexity class.

Exercise 3 BPP-completeness? – A follow up

Recall the exercise from TD07:

1. Show that the language $L_{\mathsf{NP}} = \{(M, x, 1^t) \mid M \text{ accepts on input } x \text{ in time at most } t\}$, 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 now L_{BPP} 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} BPP-hard? Is it in BPP ?

It is straightforward to prove that L_{BPP} is BPP-hard, however, it is not known if it is in BPP. To this day, no BPP-complete problem is known. This can be circumvent with promise problem. However, promise problems also ensure counter intuitive properties.

Definition 2 *A promise problem L is a pair $(L_{yes}, L_{no}) \subseteq (\{0,1\}^*)^2$ such that $L_{yes} \cap L_{no} = \emptyset$. The set $L_{yes} \cup L_{no}$ is called the* promise*.*

Definition 3 *A promise problem $L = (L_{yes}, L_{no})$ is Karp-reducible to the promise problem $L' = (L'_{yes}, L'_{no})$ if there exists a polynomial time computable function $f : \{0,1\}^* \to \{0,1\}^*$ such that:*

- *if $x \in L_{yes}$, then $f(x) \in L'_{yes}$;*
- *if $x \in L_{no}$, then $f(x) \in L'_{no}$.*

Definition 4 *A promise problem $L = (L_{yes}, L_{no})$ is Cook-reducible to the promise problem $L' = (L'_{yes}, L'_{no})$ if there exists polynomial time Turing machine $M^{L'}$ with an oracle in L' such that:*

- *if $x \in L_{yes}$, then $M^{L'}(x) = \top$;*
- *if $x \in L_{no}$, then $M^{L'}(x) = \bot$.*

Note that the correctness of the answer of the oracle is only guaranteed if the query is in the promise of the language L'.

We can now define the alternative to BPP with promise problems.

Definition 5 *Let BPP_{prm} be the set of promise problems $L = (L_{yes}, L_{no})$ such that there exists a probabilistic Turing machine M running in polynomial time such that:*

- *if $x \in L_{yes}$, then $Pr_r[M(x, r) = \top] \geq 2/3$;*
- *if $x \in L_{no}$, then $Pr_r[M(x, r) = \top] \leq 1/3$.*

1. Exhibit a BPP_{prm}-complete problem (for Karp reductions).
2. Define analogously to BPP_{prm} the classes NP_{prm} and coNP_{prm}.
3. Give a NP_{prm}-complete problem (for Karp-reduction).
4. Prove that if L is Karp-reducible to L' and L' is Cook-reducible to L'' then L is Cook reducible to L''.

5. Prove that the following problem xSAT is in $\mathsf{NP}_{prm} \cap \mathsf{coNP}_{prm}$ and is NP_{prm}-hard for Cook reductions:
    - $L_{yes} = \{(\varphi_1, \varphi_2) \mid \varphi_1 \in \mathsf{SAT},\ \varphi_2 \notin \mathsf{SAT}\}$
    - $L_{no} = \{(\varphi_1, \varphi_2) \mid \varphi_1 \notin \mathsf{SAT},\ \varphi_2 \in \mathsf{SAT}\}$

Exercise 4 The PP class

This is the same exercise as last week. The only new question is question 4.

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

Also define $\mathsf{PP}_<$ 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 $\mathsf{BPP} \subseteq \mathsf{PP}$ and $\mathsf{NP} \subseteq \mathsf{PP}$;

2. Show that $\mathsf{PP} = \mathsf{PP}_<$ and that PP is closed under complement;

3. 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 $\mathsf{MAJSAT} \in \mathsf{PP}$. In fact, MAJSAT is PP-complete.

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

4. The class $\#\mathsf{P}$ is the class of **functions** $f : \Sigma^* \to \mathbb{N}$ for which there exists a relation $R \subseteq \Sigma^* \times \Sigma^*$ and a polynomial p such that:

    (a) for every $x, y \in \Sigma^*$, $R(x,y)$ implies $|y| < p(|x|)$
    (b) $R \in \mathsf{P}$
    (c) for every x, $f(x) = |\{y \mid R(x,y)\}|$

    The function $f(x)$ counts the number of words y such that $(x,y) \in R$. In fact $\#\mathsf{P}$ is as powerful as PP, however this class cannot be compared directly since $\#\mathsf{P}$ is a class of functions. In fact, we need to use oracle machines. Specifically, for a function $f \in \#\mathsf{P}$, a Turing machine can use as oracle the function f that, when a word u is written on the oracle tape, writes in constant time $f(u)$ in binary in that oracle tape. Then, $\mathsf{P}^{\#\mathsf{P}} = \cup_{f \in \#\mathsf{P}} \mathsf{P}^f$. The class P^{PP} is defined as usual.

    Prove that: $\mathsf{P}^{\mathsf{PP}} = \mathsf{P}^{\#\mathsf{P}}$.

5. Show that **MA** $\subseteq$ 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 $(N_i)_{i\in\mathbb{N}}$ such that the probability (over the random language considered) that there is one i such that M_i and N_i does not coincide is lower than $C \cdot \epsilon$ for a constant C. You may use the language A as a random bit generator.*