You may answer in English or in French. If you answer in French, and you do not know the French equivalent of some English word that I am using, by all means do not invent your own translation: use the English word instead in that case.

Let me stress the value of rigor. In all arguments, you must: (a) stress the important idea; (b) unless stated otherwise, give explicit values for all required bounds (errors, probabilities, time and space usage, and so on); (c) give explicit references to the results you use, preferably by name (e.g., “the Immerman-Szelepcsényi theorem”).

By “polynomial”, I mean “univariate polynomial with coefficients in \( \mathbb{N} \), of degree at least 1, and whose value is strictly positive on every element of \( \mathbb{N} \)” unless stated otherwise.

1 More on MA

For every \( \epsilon \) such that \( 0 < \epsilon < 1/2 \), let \( \text{MA}_\epsilon \) be defined as \( \text{MA} \), except with error \( \epsilon \). In other terms, a language \( L \) is in \( \text{MA}_\epsilon \) if and only if there is a language \( D \in \text{P} \) and there are polynomials \( p \) and \( R \) such that for every input \( x \) (whose size I will always write as \( n \)):

- if \( x \in L \), then there is a \( y \) of size \( p(n) \) such that \( Pr_r[x\#y\#r \in D] \geq 1 - \epsilon \);
- if \( x \not\in L \), then for every \( y \) of size \( p(n) \), \( Pr_r[x\#y\#r \in D] \leq \epsilon \),

where \( r \) has size \( R(n) \).

One may use the following definition of \( \text{MA} \), which is a simplification of the definition given in the lecture notes. \( L \) is in \( \text{MA} \) if and only if for every polynomial \( q \), there is a language \( D \in \text{P} \) and there are polynomials \( p \) and \( R \) such that for every input \( x \) (whose size I will always write as \( n \)):

- if \( x \in L \), then there is a \( y \) of size \( p(n) \) such that \( Pr_r[x\#y\#r \in D] \geq 1 - 1/2^q(n) \);
• if $x \not\in L$, then for every $y$ of size $p(n)$, $\Pr_r[x\#y\#r \in D] \leq 1/2^{q(n)}$,

where $r$ has size $R(n)$. Let me draw your attention to the fact that, in this definition, the polynomials $p$ and $R$, as well as the language $D$, may depend on $q$.

We define the class $\text{MA}'$ by requiring that $p$ is independent of $q$. That is, $L$ is in $\text{MA}'$ if and only if there is a polynomial $p$ such that for every polynomial $q$, there is a language $D \in \text{P}$ and a polynomial $R$ such that [etc.]

**Question 1** Show that $\text{MA}_{1/3} \subseteq \text{MA}'$.

Let $L$ be in $\text{MA}_{1/3}$. Given $x \in L$, there is a $y$ of size $p(n)$ such that $\Pr_r[x\#y\#r \in D] \geq 2/3$. By Chernoff’s bound, as used in error reduction for $\text{BPP}$, letting $m \overset{\text{def}}{=} 36q(n)\log 2$, the probability that a majority of strings $x\#y\#r_i$ $(1 \leq i \leq m)$ is in $D$ is at least $1 - 1/2^{q(n)}$, where $r_1, \ldots, r_m$ are drawn uniformly, independently, at random.

If $x \not\in L$, then whatever $y$ of size $p(n)$ we pick, $\Pr_r[x\#y\#r \not\in D] \geq 2/3$, so the probability that a majority of strings $x\#y\#r_i$ $(1 \leq i \leq m)$ is in $D$ is at most $1/2^{q(n)}$.

Thus $L$ is in $\text{MA}'$: whatever polynomial $q$ we choose, $y$ still has size $p(n)$, and only the size of random strings increases from $R(n)$ to $36R(n)q(n)\log 2$.

**Question 2** Show that $\text{MA} \subseteq \text{MA}_{1/3} \subseteq \text{MA}' \subseteq \text{MA}$, hence all these classes are equal.

$\text{MA} \subseteq \text{MA}$, for any constant $\epsilon$ with $0 < \epsilon < 1$, since the error $1/2^{q(n)}$ is smaller than $\epsilon$ for, say, $q(n) \overset{\text{def}}{=} \lceil -\log_2 \epsilon \rceil + n$. (I am adding “+$n$” in order to satisfy the constraints that polynomials have to be of degree at least 1, see preliminary comments at the beginning of the exam.)

The second inclusion is by **Question 1**.

The third inclusion is obvious: if $p$ can be chosen independently of $q$, then for each $q$ there is a $p$, ignoring its independence from $p$.

### 2 The class $S_2^p$

For every class of languages $\mathcal{C}$, we define a new class of languages $S_2^p \cdot \mathcal{C}$ as follows. A language $L$ is in $S_2^p \cdot \mathcal{C}$ if and only if there is a language $D$ in $\mathcal{C}$ and a polynomial $p$ such that, for every input $x$:

1. if $x \in L$, then there is a $y$ of size at most $p(n)$ such that for every $z$ of size at most $p(n)$, $x\#y\#z \in D$;
2. if \( x \not\in L \), then there is a \( z \) of size at most \( p(n) \) such that for every \( y \) of size at most \( p(n) \), \( x \# y \# z \not\in D \).

Note that the existence of \( y \) satisfying 1 and the existence of \( z \) satisfying 2 are mutually exclusive conditions.

It will be practical to call the words \( y \) above the proofs (or proof candidates) that \( x \in L \), while \( z \) is a counter-proof (candidate). Taking an analogy with trials and courtrooms, proofs are provided by the attorney general, counter-proofs are provided by the defense counsel, and \( L \) is in \( S_2 \cdot C \) if and only if every input word \( x \) is either a clear win for the attorney general (whatever counter-proofs are given by the defense) or a clear win for the defense (whatever proofs are put forward by the attorney general).

The class \( S_2^p \) is \( S_2 \cdot P \). It is pretty clear that \( NP \cup coNP \subseteq S_2^p \subseteq \Sigma_2^p \cap \Pi_2^p \).

The class \( S_2BP \cdot C \) is a probabilistic version of \( S_2 \cdot C \). \( L \) is in \( S_2BP \cdot C \) if and only if, for every polynomial \( q \), there is a language \( D \in C \) and there are polynomials \( p \) and \( R \) such that, for every input \( x \):

1. if \( x \in L \), then there is a \( y \) such that for every \( z \), \( \Pr_r[x \# y \# z \# r \in D] \geq 1 - 1/2^{p(n)+q(n)} \),

2. if \( x \not\in L \), then there is a \( z \) such that for every \( y \), \( \Pr_r[x \# y \# z \# r \in D] \leq 1/2^{p(n)+q(n)} \),

where \( y \) and \( z \) are of size \( p(n) \) and \( r \) has size \( R(n) \). Now either the defense counsel or the attorney general have a clear win again, but only with high probability.

**Question 3** Show that \( MA \subseteq S_2BP \cdot P \).

We use **Question 2**: it suffices to show that every language \( L \) in \( MA' \) is in \( S_2BP \cdot P \). By definition of \( MA' \), there is a polynomial \( p \) such that, for every polynomial \( q \), there is a language \( D \in P \) such that \( \{ etc. \} \), with error bound \( 1/2^{p(n)+q(n)} \). (Since \( q \) is arbitrary, we can use \( p + q \) instead.)

We take \( y \) of size \( p(n) \) as proofs, and we simply do not care about counter-proofs \( z \). Defining \( D' \) as \( \{ x \# y \# z \# r \mid x \# y \# r \in D \} \), we see that \( D' \) is still in \( P \), and using \( D' \) for \( D \) in the definition of \( S_2BP \cdot P \), \( L \) is then in \( S_2BP \cdot P \).

We note that, in particular, we can take \( p \) independent of \( q \), as in the definition of \( MA' \).

**Question 4** Let \( L \) be a language in \( S_2BP \cdot P \), and \( q, D, p, R \) be as in the definition of \( S_2BP \cdot P \). Let \( x \) be an arbitrary input, and \( Ryz \) defined \( \{ r \mid x \# y \# z \# r \in D \} \). With the aim of using Lautemann’s technique, show that if \( x \in L \) then \( \bigcap_z Ryz \) is huge for some \( y \), in the sense that \( \Pr_r[r \in \bigcap_z Ryz] \geq 1 - 1/2^{q(n)} \); and that if \( x \not\in L \) then \( \bigcup_y Ryz \) is small for some \( z \), in that \( \Pr_r[r \in \bigcup_y Ryz] \leq 1/2^{q(n)} \).
If $x \in L$, then let $y$ be the winning proof (against all counter-proofs $z$). Namely for every $z$, $\Pr_r[r \in R_{yz}] \geq 1 - 1/2^{p(n) + q(n)}$. Equivalently, for every $z$, $\Pr_r[r \notin R_{yz}] \leq 1/2^{p(n) + q(n)}$. By the sum bound, $\Pr_r[r \notin \bigcup_z R_{yz}] \leq 2^{p(n)/2^{p(n) + q(n)}} = 1/2^{q(n)}$.

If $x \notin L$, then let $z$ be the winning counter-proof. For every $y$, $\Pr_r[r \in R_{yz}] \leq 1/2^{p(n) + q(n)}$. By the sum bound, $\Pr_r[r \in \bigcup_y R_{yz}] \leq 1/2^{q(n)}$.

**Question 5** With $L, q, D, p, R$ as above, show that, letting $m \overset{\text{def}}{=} R(n)+1$, and $m' \overset{\text{def}}{=} R(n)^2$:

(a) if $x \in L$, then: (A) there is a proof $y$ and there are random tapes $r_1, \ldots, r_m$ such that for all random tapes $s_1, \ldots, s_{m'}$, we have: for every $j \in \{1, \ldots, m'\}$, there is an $i \in \{1, \ldots, m\}$ such that for every counter-proof $z$, $x \#y\#z\#(r_i \oplus s_j) \in D$;

(b) if $x \notin L$, and if $n$ is large enough, then: (B) there is a counter-proof $z$ and there are random tapes $s_1, \ldots, s_{m'}$ such that for all random tapes $r_1, \ldots, r_m$, we have: there is a $j \in \{1, \ldots, m'\}$ such that for every $i \in \{1, \ldots, m\}$, for every proof $y$, $x \#y\#z\#(r_i \oplus s_j) \notin D$.

(Yes, in (B), we require “there is a $j$ such that for every $i$”, and not “for every $i$, there is a $j$”. By “large enough” I mean that there should be a constant $n_0$ such that (B) holds for every $n \geq n_0$.)

If $x \in L$, then for some $y$, $\Pr_r[r \in \bigcup_z R_{yz}] \geq 1 - 1/2^{q(n)}$ by Question 4. As with Lautemann’s technique, there are polynomially many random tapes $r_1, \ldots, r_m$ such that the sets $\bigcup_z R_{yz} \oplus r_i$ ($1 \leq i \leq m$) cover the whole set of possible random tapes. Explicitly, we find them by counting those that do not satisfying this property:

$$\Pr_{r_1,\ldots,r_m}[r \not\in \bigcup_i z R_{yz} \oplus r_i] \leq \sum_r \Pr_{r_1,\ldots,r_m}[r \notin \bigcup_i z R_{yz} \oplus r_i]$$

$$= \sum_r \prod_i \Pr_{r_i}[r \not\in z R_{yz} \oplus r_i]$$

$$\leq \sum_r \prod_i \frac{1}{2^{q(n)}} = \sum_r \frac{1}{2^{mq(n)}} = \frac{2^{R(n)}}{2^{(R(n)+1)q(n)}} < 1,$$

since $m = R(n) + 1$. The final inequality is due to the fact that $q(n) > 0$ (so $(R(n) + 1)q(n) \geq R(n) + 1$).

We obtain that if there is a $y$, and if there are $r_1, \ldots, r_m$ such that $\bigcup_i \bigcap_z R_{yz} \oplus r_i$ covers the whole set of random tapes. In particular, for every finite list of random tapes $s_1, \ldots, s_{m'}$, for every $j$, there is an $i$ such that $s_j$ is in $\bigcap_z R_{yz} \oplus r_i$; namely, for every $z$, $x \#y\#z\#(r_i \oplus s_j) \in D$.  

4
If \( x \notin L \), then for some \( z \), \( \Pr_r[r \in \bigcup_y R_{yz}] \leq 1/2^{q(n)} \) by Question 4. We might think of doing a similar calculation as above, but this would only lead us to a variant of (B) where we would say “for every \( i \), there is a \( j \)” instead of “there is a \( j \) such that for every \( i \)”.

Instead, (and fixing \( z \)), we estimate the probability over \( s_1, \ldots, s_m' \) that (B) would fail:

\[
\Pr_{s_1,\ldots,s_m'}[\exists r_1,\ldots,r_m, \forall j, \exists i, \exists y, x\#y\#z\#(r_i \oplus s_j) \in D]
\leq \sum_{r_1,\ldots,r_m} \prod_j \sum_i \Pr_{s_j}[s_j \in \bigcup_y R_{yz} \oplus r_j]
\leq \sum_{r_1,\ldots,r_m} \prod_j \sum_i \frac{1}{2^{q(n)}}
= 2^{mR(n)} \left( \frac{m}{2^{q(n)}} \right)^{m'}
\]

The latter is less than or equal to 1/2 if and only if \( mR(n) + m'(\log_2 m - q(n)) \leq -1 \), taking logs in base 2. When \( n \) tends to \(+\infty\), \( \log_2 m - q(n) = \log_2 R(n) - q(n) \) tends to \(-\infty\), since \( q \) has degree at least 1. Hence, for \( n \) large enough, \( mR(n) + m'\log_2 m - q(n) \leq -1 \) if and only if \( m' \geq (mR(n) + 1)/(q(n) - \log_2 m) = O(R(n)^2/q(n)) \). In particular, if \( m' \) is a \( O(R(n)^2) \), this is satisfied.

Hence, for \( n \) large enough, [there is a counter-proof \( z \)] and there are random tapes \( s_1, \ldots, s_m' \) such that for all \( r_1, \ldots, r_m \), there is a \( j \) such that for every \( i \), for every proof \( y \), \( x\#y\#z\#(r_i \oplus s_j) \in D \), as required.

**Question 6** Show that \( S_2^{BP} \cdot P \subseteq S_2 \cdot P \).

Let \( L \in S_2^{BP} \cdot P \), decided for every polynomial \( q \), by using \( D, p, R \) as above. We decide \( L \) in \( S_2^p \) by defining:

- proofs are tuples \((y, r_1, \ldots, r_m)\);
- counter-proofs are tuples \((z, s_1, \ldots, s_{m'})\);
- and we check that for every \( j \in \{1, \ldots, m'\} \), there is an \( i \in \{1, \ldots, m\} \) such that \( x\#y\#z\#(r_i \oplus s_j) \in D \), which can be done in polynomial time since \( m \) and \( m' \) are polynomial in \( n \) and \( D \) is in \( P \).

Indeed, if \( x \in L \), then we provide a proof \((y, r_1, \ldots, r_m)\) satisfying (A). Whatever counter-proof \((z, s_1, \ldots, s_{m'})\) is given, we have: (for all \( s_1, \ldots, s_{m'} \), in particular, those provided in the counter-proof) for every \( j \), there is an \( i \) such that (for every \( z \), in particular the one provided in
the counter-proof), $x \# y \# z \# (r_i \oplus s_j) \in D$. The argument is symmetric if $x \notin L$.

Technically, the size of proofs $(y, r_1, \ldots, r_m)$ is $O(p(n) + R(n)^2)$, and the size of counter-proofs $(z, s_1, \ldots, s_m')$ is $P(p(n) + R(n)^3)$, and those are different. We fix this by simply padding the proofs with blanks so as to reach the same size as the counter-proofs.

Of course, (B) only holds for $n$ large enough. For all values of $x$ such that $n$ is too small, we tabulate the answers.

**Question 7** Conclude that $\text{MA} \subseteq S_2^p$. What theorem(s) from the lecture notes does this improve upon?

We have $\text{MA} \subseteq S_2^{\text{BP}} \cdot P$ by Question 3 and $S_2^{\text{BP}} \cdot P \subseteq S_2 \cdot P = S_2^p$ by Question 6.

Since $S_2^p \subseteq \Sigma_2^p \cap \Pi_2^p$, and since $\text{BPP} \subseteq \text{MA}$, this improves upon the Sipser-Gács-Lautemann theorem $\text{BPP} \subseteq \Sigma_2^p \cap \Pi_2^p$.

This also improves the fact that $\text{MA} \subseteq \text{AM} \subseteq \Sigma_2^p \cap \Pi_2^p$, which is a consequence of theorems by Babai and by Theorem 3.19.

### 3 What would happen if polynomial space languages had polynomial circuits?

**Question 8** By looking back at the Shamir(-Shen) protocol, show that if Merlin can convince Arthur that the input QBF formula is true, it can do so in polynomial space. In other words, show that one can implement the Merlin function $M$ in the definition of the $\text{IP}$ protocol for QBF by a polynomial space Turing machine. Beware that the obvious way of computing the polynomial that Merlin should play in order to play honestly would take exponential space, because it involves computing polynomials of polynomial size on polynomially many variables. Do not bother to give me a full-fledged proof: give the idea, and explain why this gets around the difficulty that I just mentioned.

There are several solutions. Each one rests on the fact that evaluating a polynomial $F_k(X_1, \ldots, X_k) \overset{\text{def}}{=} Q_{k+1}X_{k+1}, \ldots, Q_mX_m, G(X_1, \ldots, X_k, X_{k+1}, \ldots, X_n)$ (at round $k$) on values $X_1 := v_1, \ldots, X_k := v_k \pmod{p}$ can be done in polynomial space. The honest polynomial $P_k(X_k)$ at round $k$ is $F_k(v_1, \ldots, v_{k-1}, X_k)$, for some (random) values $v_1, \ldots, v_{k-1}$, with $X_k$ remaining free.
One possibility is to evaluate $P_k(v)$ for $d+1$ values $v \mod p$, where $d$ is the (polynomial) bound on the max degree of all polynomials involved. We then compute $P_k(X_k)$ by Lagrange interpolation.

Another possibility is to guess a degree $\leq d$ polynomial $P(X_k)$ in one variable, and to check that it coincides with the honest polynomial $P_k(X_k)$, by checking that $P(v) = P_k(v)$ for all values $v \mod p$. (We can also refine this by only checking it for $d+1$ distinct values instead of all of them, but this improvement is not necessary.) This also requires polynomial space only. Non-determinism can be removed by appealing to Savitch’s theorem.

**Question 9** If $\text{PSPACE} \subseteq \text{P}/\text{poly}$, by the previous question, we can modify the Shamir(-Shen) protocol for QBF by replacing Merlin by a polynomial family of circuits. Iron out the details and show that, if $\text{PSPACE} \subseteq \text{P}/\text{poly}$, then QBF is in $\text{MA}$. I don’t need a full-fledged formal proof, but I want an explanation of the technical difficulties you must solve, and how you solve them.

Merlin needs to compute polynomial $P_k$ at each round (of polynomial size), but circuits only compute bits. We simply reconstruct $P_k$ one bit at a time. In other words, an honest Merlin can compute $P_k$ in polynomial space, given the current history $h$ of the interaction it has with Arthur. Hence it can also compute the $j$th bit of $P_k$ in polynomial space as a function of $h$ and $j$. If $\text{PSPACE} \subseteq \text{P}/\text{poly}$, then there are polynomial-sized circuits $C_n$ that do exactly this.

The maximum size of $h$ we need, hence also of $(h,j)$, is bounded by a polynomial $p(n)$.

The $\text{MA}$ protocol for QBF then requests the circuits $C_m$ for $m = 0, 1, \ldots, p(n)$ from Merlin. Arthur then simulates the whole Shamir(-Shen) protocol by calling the relevant circuits in order to simulate the computation of $P_k$ by Merlin at each round.

If the original QBF formula is true, then Merlin will make Arthur accept (always) by playing the circuits for its honest strategy in the Shamir(-Shen) protocol.

If it is false, then whatever circuits Merlin provides, Arthur will accept with low probability, as in the Shamir(-Shen) protocol.

**Question 10** Conclude that $\text{PSPACE} \subseteq \text{P}/\text{poly}$ implies $\text{PSPACE} = \text{MA}$, and that the polynomial-time hierarchy then collapses at level 2.
MA is closed under polynomial-time reductions, since they can be incorporated in Arthur’s computation. Since QBF is PSPACE-complete, and because of the previous questions, PSPACE ⊆ P/poly implies PSPACE ⊆ MA.

We have MA ⊆ S^P_2 (by Question 7) ⊆ Σ_2 ∩ Π_2 ⊆ PH ⊆ PSPACE (we can also use Babai’s theorem and Theorem 3.19 in order to obtain MA ⊆ Π^P_2 ⊆ PH, and PH ⊆ PSPACE was obtained during Philippe Schnoebelen’s lectures), so this implies that PSPACE = MA and PH = Σ^P_2.