אוניברסיטת
אריאל
בשומרון

Computational Complexity

Lecturer: Edward A. Hirsch

https://edwardahirsch.github.io/edwardahirsch

edward.a.hirsch@gmail.com

January 7, 2026

Lecture plan (including the next lecture)

- Interactive protocols (continued).
  - Private coin: **IP** (continued).
  - Public coin: **MA**, **AM**.
  - Examples: Graph (non)isomorphism revisited, Set size estimation.
  - Where are **MA** and **AM**?
  - **IP** = **PSPACE**.

- The PCP Theorem and hardness of approximation.
  - Statement of the theorem.
  - Approximation algorithms.
  - Inapproximability as a corollary to the PCP Theorem.
  - Proof of (a part of) the theorem.
- Exercises.

Interactive Protocols
(continued)

Interactive proofs. Public coin: Arthur–Merlin proofs

~~Prover~~

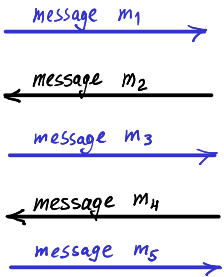
a magician

Merlin

~~Verifier~~

an ordinary person
(polynomial-time machine)

Arthur



V accepts or rejects this "proof"

Public coin ⇒ V is less powerful
(a private-coin V can send its random bits; a public-coin V cannot hide them).

One can assume that Arthur's messages are just the random bits
(Merlin can compute everything else himself) — this is how it is defined in the book.

Interactive proofs. Public coin: Arthur–Merlin proofs

Consider the problem $x \in^? L$, so x is the input. Merlin claims $x \in L$.
Denote (random variable!) $\mathrm{Out}\langle M, A\rangle(x)$ = the answer A gives after interacting with P on x

Definition

$L \in \mathbf{AM}[k] \iff$ there is a k-round public-coin protocol (the first message is Arthur's) with poly-time deterministic algorithm A s.t. $\forall x$

(Completeness) $x \in L \Rightarrow \exists M$ s.t. $\Pr\{\mathrm{Out}\langle M, A\rangle(x) = 1\} \geqslant \frac{3}{4}$ (some M convinces A).

(Soundness) $x \notin L \Rightarrow \forall M'$ only $\Pr\{\mathrm{Out}\langle M', A\rangle(x) = 1\} \leqslant \frac{1}{4}$ (nobody convinces A).

The probability is taken over a random string jointly viewed by M and A.

$\mathbf{AM} = \mathbf{AM}[2]$

Facts (we will not prove them):

- $\mathbf{AM}[O(1)] = \mathbf{AM}[2]$.
  This is why it's called **AM**: Arthur sends, then Merlin sends, that's it.
- $\mathbf{IP}[O(1)] = \mathbf{AM}[O(1)]$. We will show it for two rounds.
- One can assume perfect completeness
  (there is a Merlin that always persuades, i.e., prob. **1** instead of $\frac{3}{4}$).

Arthur–Merlin proofs via probabilistic reductions

Another view of **AM**: $L \in \mathbf{AM}$ if there is a randomized polynomial-time algorithm A s.t. $\forall x$

(Completeness) $x \in L \Rightarrow \quad \Pr\{\exists w : A(x, w) = 1\} \geqslant \frac{3}{4}$.

(Soundness) $\quad x \notin L \Rightarrow \quad \Pr\{\exists w : A(x, w) = 1\} \leqslant \frac{1}{4}$.

Arthur–Merlin proofs via probabilistic reductions

Another view of **AM**: $L \in \mathbf{AM}$ if there is a randomized polynomial-time algorithm A s.t. $\forall x$

(Completeness) $x \in L \Rightarrow \quad \Pr_r\{\exists w : A(x, w, r) = 1\} \geqslant \frac{3}{4}$.

(Soundness) $\quad x \notin L \Rightarrow \quad \Pr_r\{\exists w : A(x, w, r) = 1\} \leqslant \frac{1}{4}$.

Arthur–Merlin proofs via probabilistic reductions

Another view of **AM**: $L \in \mathbf{AM}$ if there is a randomized polynomial-time algorithm A s.t. $\forall x$

(Completeness) $x \in L \Rightarrow \quad \Pr_r\{\exists w : A(x, w, r) = 1\} \geqslant \frac{3}{4}$.

(Soundness) $\quad x \notin L \Rightarrow \quad \Pr_r\{\exists w : A(x, w, r) = 1\} \leqslant \frac{1}{4}$.

A language L is many-one reducible to a language S via a two-sided error probabilistic reduction (notation: $L = \mathbf{BP} \cdot S$) if $S \subseteq \bigcup_{n \in \mathbb{N}} \{0,1\}^n \times \{0,1\}^{p(n)}$ and

$$x \in L \quad \Rightarrow \quad \Pr_{r \in \{0,1\}^{p(|x|)}}\{(x, r) \in S\} \geqslant \frac{3}{4}$$

$$x \notin L \quad \Rightarrow \quad \Pr_{r \in \{0,1\}^{p(|x|)}}\{(x, r) \in S\} \leqslant \frac{1}{4}.$$

Arthur–Merlin proofs via probabilistic reductions

Another view of **AM**: $L \in \mathbf{AM}$ if there is a randomized polynomial-time algorithm A s.t. $\forall x$

(Completeness) $x \in L \Rightarrow \quad \Pr_r\{\exists w : A(x,w,r) = 1\} \geqslant \frac{3}{4}$.

(Soundness) $\quad x \notin L \Rightarrow \quad \Pr_r\{\exists w : A(x,w,r) = 1\} \leqslant \frac{1}{4}$.

A language L is many-one reducible to a language S via a two-sided error probabilistic reduction (notation: $L = \mathbf{BP} \cdot S$) if $S \subseteq \bigcup_{n \in \mathbb{N}} \{0,1\}^n \times \{0,1\}^{p(n)}$ and

$$x \in L \quad \Rightarrow \quad \Pr_{r \in \{0,1\}^{p(|x|)}}\{(x,r) \in S\} \geqslant \frac{3}{4}$$

$$x \notin L \quad \Rightarrow \quad \Pr_{r \in \{0,1\}^{p(|x|)}}\{(x,r) \in S\} \leqslant \frac{1}{4}.$$

Lemma

$\mathbf{AM} = \mathbf{BP} \cdot \mathbf{NP}$ (i.e., **AM** is the class of languages that are many-one reducible to **NP** via a two-sided error probabilistic reduction). In particular, $\mathbf{AM} \subseteq \mathbf{BPP}^{\mathbf{NP}}$.

Define $S := \{(x,r) : \exists w \ A(x,w,r) = 1\}$ ($\in \mathbf{NP}$). For every x,

$$\Pr_r\{(x,r) \in S\} = \Pr_r\{\exists w \ A(x,w,r) = 1\}.$$

An important example: Set lower bound protocol (with a quantifier!)

Consider set $S = \{y \in \{0,1\}^m : \exists w \ R(y,w) = 1\}$, where $R \in \mathbf{P}$.

Merlin: I claim that $|S| \geqslant K$.
Arthur: Prove it!

An important example: Set lower bound protocol (with a quantifier!)

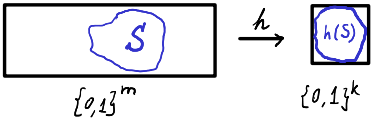
Consider set $S = \{y \in \{0,1\}^m : \exists w \ R(y,w) = 1\}$, where $R \in \mathbf{P}$.

Merlin: I claim that $|S| \geqslant K$.
Arthur: Prove it!

Arthur: Computes k s.t. $2^{k-2} < K \leqslant 2^{k-1}$.
Sends a random pairwise-independent hash-function $h\colon \{0,1\}^m \to \{0,1\}^k$.
Sends a random point $z \in \{0,1\}^k$.

Merlin: Sends $y \in h^{-1}(z) \cap S$.
Sends w s.t. $R(y,w) = 1$.

Let $p := K/2^k$ $(\in [\frac{1}{4}..\frac{1}{2}])$.



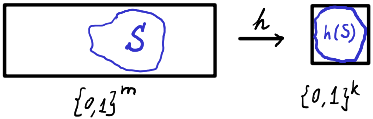
An important example: Set lower bound protocol (with a quantifier!)

Consider set $S = \{y \in \{0,1\}^m : \exists w\ R(y,w) = 1\}$, where $R \in \mathbf{P}$.

Merlin: I claim that $|S| \geqslant K$.
Arthur: Prove it!

Arthur: Computes k s.t. $2^{k-2} < K \leqslant 2^{k-1}$.
Sends a random pairwise-independent hash-function $h\colon \{0,1\}^m \to \{0,1\}^k$.
Sends a random point $z \in \{0,1\}^k$.

Merlin: Sends $y \in h^{-1}(z) \cap S$.
Sends w s.t. $R(y,w) = 1$.



Let $p := K/2^k$ $(\in [\frac{1}{4}..\frac{1}{2}])$.

Soundness:

$$\Pr_{h,z}\{h^{-1}(z) \cap S \neq \emptyset\} = \Pr_{h,z}\{z \in h(S)\} \leqslant \max_h \Pr_z\{z \in h(S)\} \leqslant |S| \cdot \frac{1}{2^k}.$$

If $|S| \leqslant K/2$, then Merlin succeeds with prob. $\leqslant \dfrac{|S|}{2^k} \leqslant \dfrac{K}{2 \cdot 2^k} \leqslant \dfrac{p}{2}$.

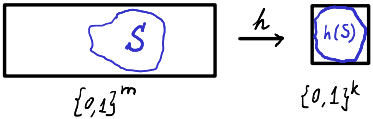
An important example: Set lower bound protocol (with a quantifier!)

Consider set $S = \{y \in \{0,1\}^m : \exists w\ R(y,w) = 1\}$, where $R \in \mathbf{P}$.

Merlin: I claim that $|S| \geqslant K$.
Arthur: Prove it!

Arthur: Computes k s.t. $2^{k-2} < K \leqslant 2^{k-1}$.
Sends a random pairwise-independent hash-function $h\colon \{0,1\}^m \to \{0,1\}^k$.
Sends a random point $z \in \{0,1\}^k$.

Merlin: Sends $y \in h^{-1}(z) \cap S$.
Sends w s.t. $R(y,w) = 1$.



Let $p := K/2^k$ $(\in [\frac{1}{4}..\frac{1}{2}])$.

Completeness: If $|S| \geqslant K\ldots$

$$\Pr_{h,z}\{h^{-1}(z) \cap S \neq \emptyset\} = \Pr_{h,z}\{\exists y \in S : h(y) = z\}$$

Estimate it even for every z! Inclusion-exclusion (for $|S| \leqslant 2^k$ — otherwise even better):

$$\geqslant \sum_{y \in S} \Pr_h\{h(y) = z\} - \frac{1}{2} \sum_{y \neq y' \in S} \Pr_h\{h(y) = z \wedge h(y') = z\} = \frac{|S|}{2^k} - \frac{1}{2} \cdot \frac{|S|^2}{2^{2k}}$$

$$= \frac{|S|}{2^k}\left(1 - \frac{|S|}{2^{k+1}}\right) \geqslant \frac{3}{4} \cdot \frac{|S|}{2^k} \geqslant \frac{3}{4}p.$$

An important example: Set lower bound protocol (with a quantifier!)

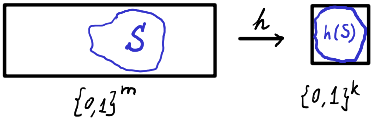
Consider set $S = \{y \in \{0,1\}^m : \exists w\ R(y,w) = 1\}$, where $R \in \mathbf{P}$.

Merlin: I claim that $|S| \geqslant K$.
Arthur: Prove it!

Arthur: Computes k s.t. $2^{k-2} < K \leqslant 2^{k-1}$.
Sends a random pairwise-independent hash-function $h\colon \{0,1\}^m \to \{0,1\}^k$.
Sends a random point $z \in \{0,1\}^k$.

Merlin: Sends $y \in h^{-1}(z) \cap S$.
Sends w s.t. $R(y,w) = 1$.

Let $p := K/2^k$ $(\in [\frac{1}{4}..\frac{1}{2}])$.



Overall:
If $|S| \geqslant K$, then $\Pr\{\text{Merlin succeeds}\} \geqslant \frac{3}{4}p$.
If $|S| \leqslant K/2$, then $\Pr\{\text{Merlin succeeds}\} \leqslant \frac{p}{2}$.

By repeating and using Chernoff's bound (like in **BPP** amplification) we get good probabilities.
(Set the threshold of the share of "yes" answers to, say, $\frac{5}{8}p$, because $\frac{p}{2} < \frac{5}{8}p < \frac{3}{4}p$.)

Converting private coin into public coin

Prover and Verifier look at a private-coin protocol.

(Completeness) $x \in L \Rightarrow \quad \mathrm{Out}\langle P, V\rangle(x) \geqslant \frac{3}{4}$.

(Soundness) $\quad x \notin L \Rightarrow \quad \mathrm{Out}\langle P, V\rangle(x) \leqslant \frac{1}{4}$.

~~Prover:~~ Merlin: I will persuade you that P succeeds whp.

That is, the set of random strings s.t. the message sent by Verifier have a good answer of Prover is large.

Converting private coin into public coin

Prover and Verifier look at a private-coin protocol.

(Completeness) $x \in L \Rightarrow \quad \mathrm{Out}\langle P, V\rangle(x) \geqslant \frac{3}{4}$.

(Soundness) $\quad x \notin L \Rightarrow \quad \mathrm{Out}\langle P, V\rangle(x) \leqslant \frac{1}{4}$.

~~Prover:~~ Merlin: I will persuade you that P succeeds whp.

That is, the set of random strings s.t. the message sent by Verifier have a good answer of Prover is large.

. . . and they engage in the public-coin set lower bound protocol for this set.

Why this set is in **NP**? The witness here is both verifier's randomness and Merlin's answer.

Converting private coin into public coin

Prover and Verifier look at a private-coin protocol.

(Completeness) $x \in L \Rightarrow \quad \mathrm{Out}\langle P, V\rangle(x) \geqslant \frac{3}{4}$.

(Soundness) $\quad x \notin L \Rightarrow \quad \mathrm{Out}\langle P, V\rangle(x) \leqslant \frac{1}{4}$.

~~Prover:~~ Merlin: I will persuade you that P succeeds whp.

That is, the set of random strings s.t. the message sent by Verifier have a good answer of Prover is large.

. . . and they engage in the public-coin set lower bound protocol for this set.

Why this set is in **NP**? The witness here is both verifier's randomness and Merlin's answer.

Arora–Barak exemplify it by a public-coin protocol for `GNI`.

There not just the set of random strings is large — the set of V's messages is larger in the case $x \in$ `GNI`, because they are generated from two graphs, not one.

Merlin–Arthur: Merlin speaks first

We consider only two-round protocols here (otherwise it is **AM**).

Definition

$L \in \mathbf{MA}$ if there is a randomized poly-time algorithm A s.t. $\forall x$

(Completeness) $x \in L \Rightarrow \exists w \quad \Pr\{A(x, w) = 1\} \geqslant \frac{3}{4}$ (once can convince A).

(Soundness) $x \notin L \Rightarrow \forall w \quad \Pr\{A(x, w) = 1\} \leqslant \frac{1}{4}$ (nobody convinces A).

Note that it's like **NP**, just verification is randomized!

Merlin–Arthur: Merlin speaks first

We consider only two-round protocols here (otherwise it is **AM**).

Definition

$L \in$ **MA** if there is a randomized poly-time algorithm A s.t. $\forall x$

(Completeness) $x \in L \Rightarrow \exists w \quad \Pr\{A(x, w) = 1\} \geqslant \frac{3}{4}$ (once can convince A).

(Soundness) $x \notin L \Rightarrow \forall w \quad \Pr\{A(x, w) = 1\} \leqslant \frac{1}{4}$ (nobody convinces A).

Note that it's like **NP**, just verification is randomized!

Example (circuit optimization):

Given a circuit C, say whether there exists an equivalent circuit of size $\leqslant k$.

Protocol:

Merlin sends a circuit D of size $\leqslant k$.
Arthur verifies the equivalence by PIT.

Merlin–Arthur: Merlin speaks first

We consider only two-round protocols here (otherwise it is **AM**).

Definition

$L \in \mathbf{MA}$ if there is a randomized poly-time algorithm A s.t. $\forall x$

(Completeness) $x \in L \Rightarrow \exists w \quad \Pr\{A(x,w)=1\} \geqslant \frac{3}{4}$ (once can convince A).

(Soundness) $x \notin L \Rightarrow \forall w \quad \Pr\{A(x,w)=1\} \leqslant \frac{1}{4}$ (nobody convinces A).

Note that it's like **NP**, just verification is randomized!

Lemma (perfect completeness for MA)

If we replace $\frac{3}{4}$ by 1, the definition is equivalent. In particular, $\mathbf{MA} \subseteq \Sigma_2^p$.

Sketch
Recall xor-cover
($\mathbf{BPP} \in \Sigma_2^p$):

- Let $|w| = p(n)$ for $|x| = n$.
- Arthur can repeat itself many times on the same w thus getting error $\leqslant 1/2^{n^2 p^2(n)}$.
- Merlin can send w with poly-many strings $t_1, \dots t_d$ to cover all random strings U by xor-copies of the set $Y_{x,w} = \{r : A(x,w,r)=1\}$.
- Arthur verifies $U = \bigcup_i Y_{x,w} \oplus t_i$ probabilistically.
- For $x \in L$, Arthur always accepts. Otherwise this $\bigcup \dots$ is tiny!

MA vs AM

$L \in \mathbf{MA}$
if there is a randomized polytime algorithm A s.t. $\forall x$

(Completeness) $x \in L \Rightarrow \exists w \quad \Pr\{A(x,w)=1\} \geqslant \frac{3}{4}$.

(Soundness) $x \notin L \Rightarrow \forall w \quad \Pr\{A(x,w)=1\} \leqslant \frac{1}{4}$.

$L \in \mathbf{AM}$
if there is a randomized polytime algorithm A s.t. $\forall x$

$x \in L \Rightarrow \quad \Pr\{\exists w : A(x,w)=1\} \geqslant \frac{3}{4}$.

$x \notin L \Rightarrow \quad \Pr\{\exists w : A(x,w)=1\} \leqslant \frac{1}{4}$.

Lemma

MA $\subseteq$ **AM**.

- Let $|w| = p(n)$ for $|x| = n$.
- Arthur can repeat itself many times on the same w thus getting error $\leqslant 1/2^{np(n)}$.
- Let Merlin send w even after Arthur thus turning MA into AM!
- (There are so few "bad" random strings that most of the good strings work for every Merlin's message, not just one.)

MA vs AM

$L \in$ **MA**
if there is a randomized polytime algorithm A s.t. $\forall x$

(Completeness) $x \in L \Rightarrow \exists w \quad \Pr\{A(x,w)=1\} \geqslant \frac{3}{4}$.
(Soundness) $x \notin L \Rightarrow \forall w \quad \Pr\{A(x,w)=1\} \leqslant \frac{1}{4}$.

$L \in$ **AM**
if there is a randomized polytime algorithm A s.t. $\forall x$

$x \in L \Rightarrow \quad \Pr\{\exists w : A(x,w)=1\} \geqslant \frac{3}{4}$.
$x \notin L \Rightarrow \quad \Pr\{\exists w : A(x,w)=1\} \leqslant \frac{1}{4}$.

Lemma

MA $\subseteq$ **AM**.

- Let $|w| = p(n)$ for $|x| = n$.
- Arthur can repeat itself many times on the same w thus getting error $\leqslant 1/2^{np(n)}$.
- Let Merlin send w even after Arthur thus turning MA into AM!
- (There are so few "bad" random strings that most of the good strings work for every Merlin's message, not just one.)

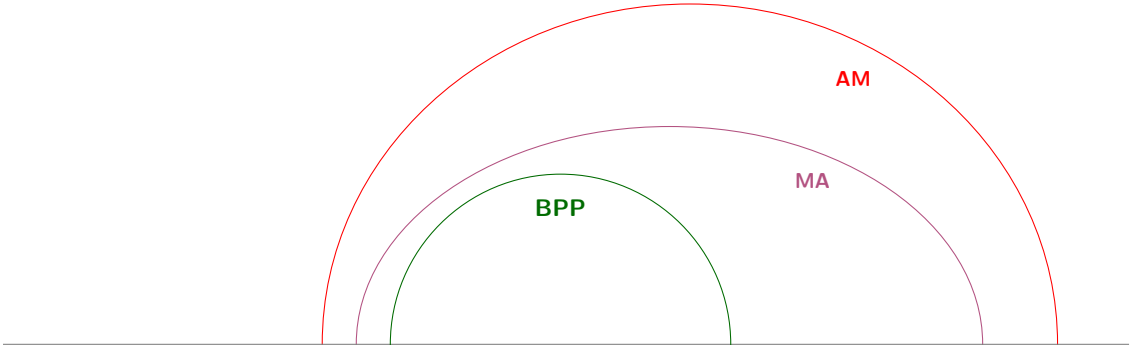
- Completeness: If $x \in L$, it only improves prob.
- Soundness:
  If $\exists w \ \Pr\{A(x,w)=1\} \leqslant \dfrac{1}{2^{np(n)}}$,
  then $\Pr\{\exists w : A(x,w)=1\} < 2^{p(n)} \times \dfrac{1}{2^{np(n)}} = \dfrac{1}{2^n}$.

An exercise

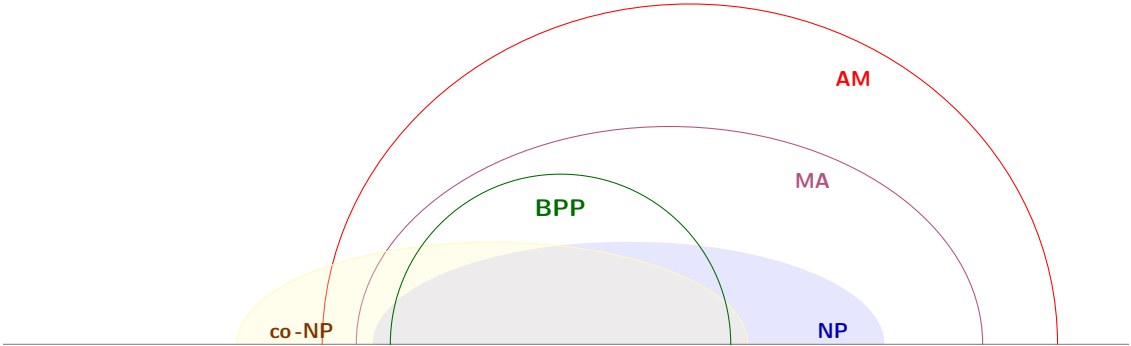
This slides was missing in the actual lecture.

Exercise: By analogy to $\mathbf{MA} \subseteq \mathbf{AM}$, show that $\mathbf{AMA} = \mathbf{AM}$, that is, the last random choice of Arthur can be made earlier.

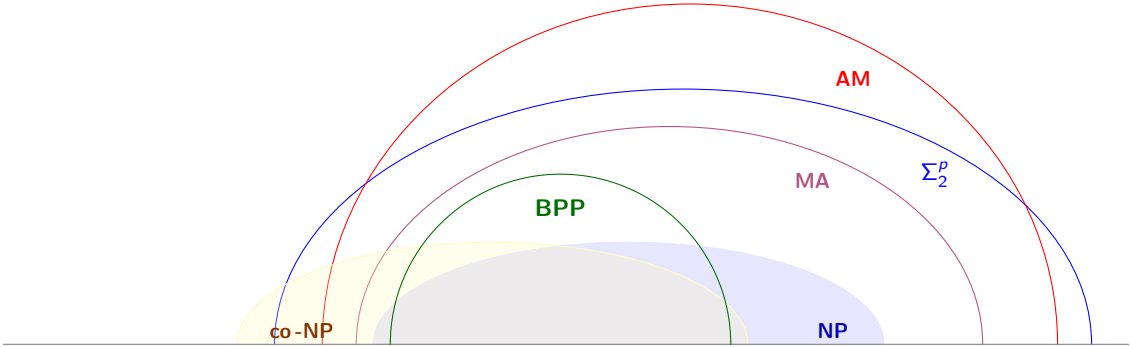
Public-coin classes vs PH



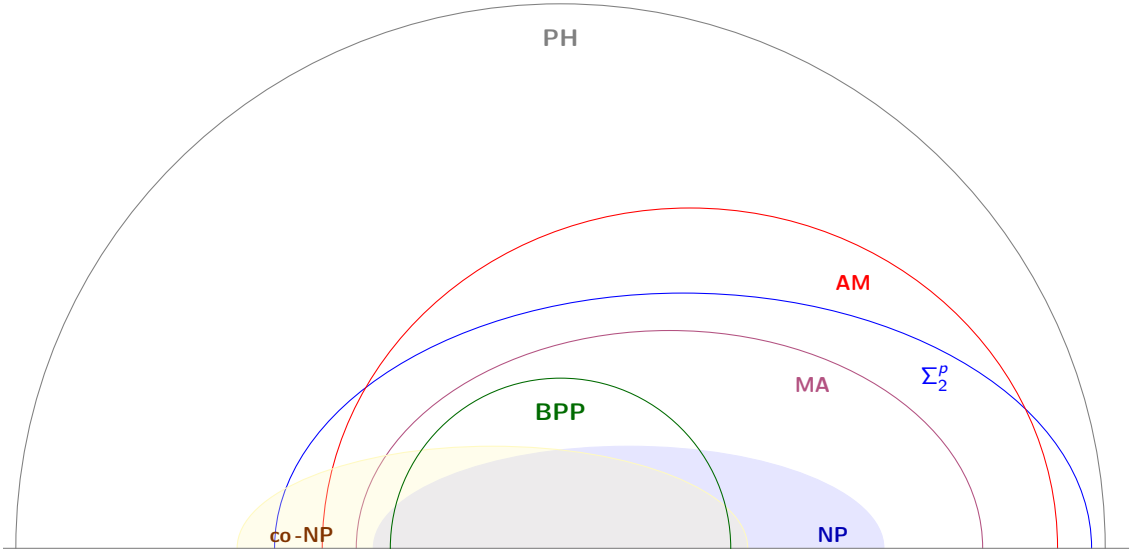
Public-coin classes vs PH



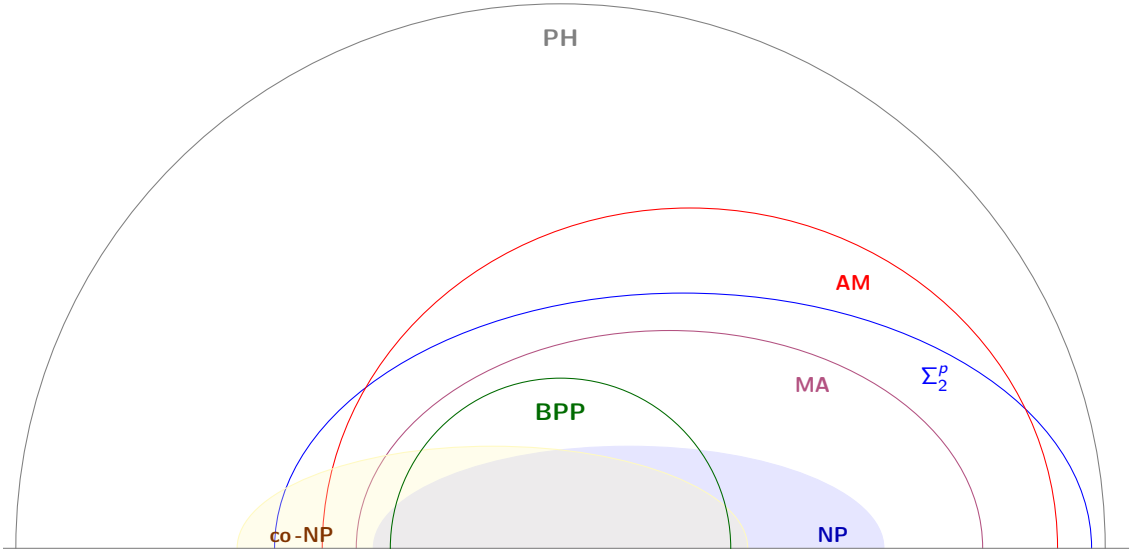
Public-coin classes vs PH



Public-coin classes vs PH



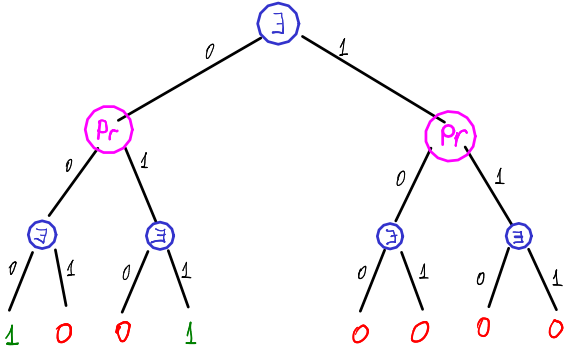
Public-coin classes vs **PH**



. . . and **IP** is above all that because (spoiler!) we will show that **IP** = **PSPACE**.

IP = PSPACE

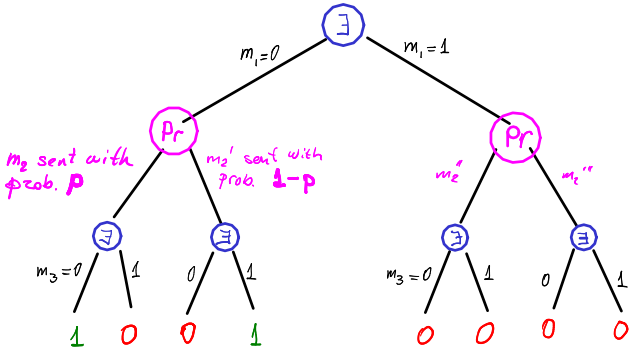
IP $\subseteq$ **PSPACE**:



This is wrong! The coin in this tree is public — yet we wanted a private one.

IP = PSPACE

IP $\subseteq$ **PSPACE**:



Split over the messages (not coins) and count the probabilities.

Now one can design a recursive (thus **PSPACE**) deterministic algorithm for computing the probability of acceptance: If the pr. of acceptance in two substrees are p_1 and p_2, then

- For a "$\exists$" node, the tree evaluates as $\max(p_1, p_2)$ (Merlin selects the best option!).
- For a "Pr" node, the tree evaluates as $pp_1 + (1-p)p_2$ (Arthur selects a subtree at random).

IP = PSPACE

PSPACE $\subseteq$ **IP**:
It suffices to design an **IP** protocol for a **PSPACE**-complete problem (QBF).
A sketch has been shown on the whiteboard.

IP = PSPACE

Corollary

1. An equivalent definition of **IP** is with perfect completeness (one can change $\frac{3}{4}$ to 1 in the definition).
2. $\mathbf{IP} = \mathbf{AM}[\mathrm{poly}(n)]$.
3. $\mathbf{IP} = \mathbf{co}\text{-}\mathbf{IP}$.

1 and 2 are straightforward properties of our protocol for QBF.
Thus one convert any protocol first to a **PSPACE** algorithm (thus to QBF) and then to 1 and/or 2.

3 is due to $\mathbf{PSPACE} = \mathbf{co}\text{-}\mathbf{PSPACE}$.