Randomized complexity classes

Today: the Arthur vs. Merlin hierarchy, and interactive proofs
Today

- Arthur vs. Merlin games
- Interactive proofs
- Various characterizations of AM
Arthur vs. Merlin games
Trading Group Theory for Randomness

László Babai

Dept. Algebra
Eötvös University
Budapest
Hungary H-1088

Dept. Computer Science
University of Chicago
1100 E 58th St.
Chicago, IL 60637

Abstract.

In a previous paper [8] we proved, using the elements of the theory of nilpotent groups, that none of the fundamental computational problems in matrix groups is known to be nontrivially hard. These problems were also shown to be unproven hypothesis concerning groups.

The aim of this paper is to replace most of the unproven group theory of [8] by elementary arguments. The result we prove is to a random variable $B$, the mentioned matrix groups belong to $\text{NP}$.

The problems we consider are membership of a matrix group given by a list of generators, from can be viewed as a sublinear-time random variable relative of the discrete logarithm problem, the $\text{NP}$ might be the lowest natural complexity class they move to.

We remark that the results remain valid for

1. Introduction

1.1. Randomness vs. mathematical intractability: a tradeoff


Arthur Merling Games: A Randomized Proof System, and a Hierarchy of Complexity Classes

LÁSZLÓ BABAI

Eötvös University, Budapest, Hungary and
University of Chicago, Chicago Illinois

AND

SHLOMO MORAN

Technion Haifa, Israel

Received June 24, 1986; revised August 3, 1987

One can view $\text{NP}$ as the complexity class that captures the notion of efficient provability by classical (formal) proofs. We consider a family of complexity classes with "just above $\text{NP}$" in every case in the hope of formulating the notion of efficient provability by overwhelming statistical evidence.

Such a concept should combine the nondeterministic nature of (classical) proofs and the statistical nature of conclusions via Monte Carlo algorithms such as a Solovay-Strassen style "proof" of primality. To accomplish this goal, two randomized interactive proof systems have recently been offered independently by S. Goldwasser, S. Micali, and C. Rackoff (GMR system) and [GMR] in "Proceedings, 17th ACM Symp. Theory of Comput., Providence, RI, 1985," 17pp. 291-304 and by L. Babai (Arthur-Merlin system) (in "Proceedings, 17th ACM Symp. Theory of Comput., Providence, RI, 1985," pp. 421-429), respectively. The proving power of the two systems has subsequently been shown by S. Goldwasser and M. Sipser to
Arthur vs. Merlin games

- Imagine we would like to decide whether $x \in L$
Imagine we would like to decide whether \( x \in L \).

We ask Arthur — a mere mortal, who lives only for polynomial time.
Arthur vs. Merlin games

- Imagine we would like to decide whether \( x \in L \)
- We ask Arthur — a mere mortal, who lives only for polynomial time
- Arthur can ask Merlin… a supernatural being able to give the answer to any problem (even non-computable)
Arthur vs. Merlin games

- Imagine we would like to decide whether $x \in L$
- We ask Arthur — a mere mortal, who lives only for polynomial time
- Arthur can ask Merlin… a supernatural being able to give the answer to any problem (even non-computable)

Arthur: http://lusile17.1.u.pic.centerblog.net/273f716e.jpg
Arthur vs. Merlin games

- Imagine we would like to decide whether $x \in L$
- We ask Arthur — a mere mortal, who lives only for polynomial time
- Arthur can ask Merlin… a supernatural being able to give the answer to any problem (even non-computable)
- but Arthur does not trust Merlin…

Arthur: http://lusile17.1.u.pic.centerblog.net/273f716e.jpg
Arthur vs. Merlin games

- Imagine we would like to decide whether \( x \in L \)
- We ask Arthur — a mere mortal, who lives only for polynomial time
- Arthur can ask Merlin for a proof \( y \) that \( x \) is in \( L \)
Arthur vs. Merlin games

- Imagine we would like to decide whether $x \in L$
- We ask Arthur — a mere mortal, who lives only for polynomial time
- Arthur can ask Merlin for a proof $y$ that $x$ is in $L$
- now Arthur can check Merlin’s proof… provided $y$ has polynomial size
Arthur vs. Merlin games

- **INPUT:** $x$
- Merlin answers $y$
- We check whether $(x, y) \in D$ (for some $D$ in $P$)

Arthur: http://lusile17.1.u.pic.centerblog.net/273f716e.jpg
Arthur vs. Merlin games

- **INPUT:** $x$
- Merlin answers $y$
- We check whether $(x,y) \in D$ (for some $D$ in $\mathbb{P}$)
- The languages decided this way are just those in $\mathbb{NP}$. 

Arthur: http://lusile17.l.u.pic.centerblog.net/273f716e.jpg
The class AM

- Now Arthur can also draw (uniform) random strings
- INPUT: $x$
- Arthur draws $r$ at random and computes a question
  \[ q \equiv A(x,r) \]
The class AM

- Now Arthur can also draw (uniform) **random strings**
- INPUT: $x$
- Arthur draws $r$ at random and computes a question
  
  \[ q \equiv A(x, r) \]
- … and sends $x#q#r$ to Merlin
The class AM

- Now Arthur can also draw (uniform) **random strings**
- **INPUT:** $x$
- Arthur draws $r$ at random and computes a question
  \[ q \equiv \mathcal{A}(x,r) \]
- … and sends $x#q#r$ to Merlin
- Merlin answers $y$

Arthur: [Image](http://lusile17.l.u.pic.centerblog.net/273f716e.jpg)
The class AM

- Now Arthur can also draw (uniform) random strings
- INPUT: $x$
- Arthur draws $r$ at random and computes a question $q \equiv A(x,r)$
- ... and sends $x\#q\#r$ to Merlin
- Merlin answers $y$
- We check whether $x\#q\#r\#y \in D$ (for some $D$ in P)

Arthur: http://lusile17.1.u.pic.centerblog.net/273f716e.jpg
The class AM

- Now Arthur can also draw (uniform) **random strings**
- **INPUT**: $x$
- Arthur draws $r$ at random and computes a question
  \[ q \equiv A(x, r) \]
- ... and sends $x\#q\#r$ to Merlin
- Merlin answers $y$
- We check whether
  \[ x\#q\#r\#y \in D \text{ (for some } D \text{ in } P) \]
- Acceptance condition: if $x \in L$ then succeeds with high prob.
  if $x \notin L$ then fails with high prob.
The class AM

- Now Arthur can also draw (uniform) random strings
- INPUT: $x$
- Arthur draws $r$ at random and computes a question $q \equiv A(x,r)$
- ... and sends $x#q#r$ to Merlin
- Merlin answers $y$
- We check whether $x#q#r#y \in D$ (for some $D$ in P)
- Acceptance condition: if $x \in L$ then succeeds with high prob.
  if $x \notin L$ then fails with high prob.

Arthur: http://lusile17.1.u.pic.centerblog.net/273f716e.jpg

... with a catch! (in fact, two)
The class AM, formally (1st try)

- $L$ is in AM iff there are:
  - a poly time Turing machine $A$
    (used by Arthur to compute questions $q \equiv A(x,r)$)
  - a function $M : \Sigma^* \rightarrow \Sigma^*$ producing poly size outputs
    (a Merlin map, not necessarily computable)
  - a poly time decidable language $D$
  such that:
- if $x \in L$ then $\Pr_r(x#q#r#y \in D) \geq 2/3$
- if $x \notin L$ then $\Pr_r(x#q#r#y \in D) \leq 1/3$
- where $q \equiv A(x,r)$, $y \equiv M(x#q#r)$

Then decide whether $x#q#r#y \in D$
The class AM, formally (1st try)

- $L$ is in AM iff there are:
  - a **poly time** Turing machine $A$ (used by Arthur to compute questions $q \leftarrow A(x,r)$)
  - a function $M: \Sigma^* \rightarrow \Sigma^*$ producing **poly size** outputs (a *Merlin map*, not necessarily computable)
  - a **poly time** decidable language $D$ such that:
    - if $x \in L$ then $\Pr_r(x \# q \# r \# y \in D) \geq 2/3$
    - if $x \notin L$ then $\Pr_r(x \# q \# r \# y \in D) \leq 1/3$
    - where $q \leftarrow A(x,r), y \leftarrow M(x \# q \# r)$

What *honest* Merlin plays, in order to make us accept when $x \in L$
The class $\text{AM}$, formally (1st try)

- $L$ is in $\text{AM}$ iff there are:
  - a poly time Turing machine $A$
    (used by Arthur to compute questions $q = A(x,r)$)
  - a function $M : \Sigma^* \to \Sigma^*$ producing poly size outputs
    (a Merlin map, not necessarily computable)
  - a poly time decidable language $D$ such that:
    - if $x \in L$ then $\Pr_r(x \# q \# r \# y \in D) \geq 2/3$
    - if $x \notin L$ then $\Pr_r(x \# q \# r \# y \in D) \leq 1/3$
    - where $q \neq A(x,r)$, $y \neq M(x \# q \# r)$

What **honest** Merlin plays, in order to make us accept when $x \in L$

First catch: when $x \notin L$, we should reject with high prob.
even if Merlin is **dishonest**, namely whatever $y$ it plays
The class AM, formally (2nd try)

- $L$ is in AM iff there are:
  - a poly time Turing machine $A$
  - a Merlin map $M : \Sigma^* \rightarrow \Sigma^*$ producing poly size outputs
  - a poly time decidable language $D$
  
  such that:

- if $x \in L$ then $\Pr_r(x \# q \# r \# y \in D) \geq 2/3$
  where $q \equiv A(x,r)$, $y \equiv M(x \# q \# r)$

Then decide whether $x \# q \# r \# y \in D$
The class AM, formally (2nd try)

- $L$ is in AM iff there are:
  - a poly time Turing machine $A$
  - a Merlin map $M : \Sigma^* \to \Sigma^*$ producing poly size outputs
  - a poly time decidable language $D$

  such that:

- if $x \in L$ then $\Pr_r(x \# q \# r \# y \in D) \geq 2/3$
  where $q \equiv A(x, r), y \equiv M(x \# q \# r)$

- if $x \notin L$ then $\forall$ Merlin map $M'$,
  $\Pr_r(x \# q \# r \# y \in D) \leq 1/3$
  where $q \equiv A(x, r), y \equiv M'(x \# q \# r)$
The class \( AM \), formally (2nd try)

- \( L \) is in \( AM \) iff there are:
  - a poly time Turing machine \( A \)
  - a Merlin map \( M : \Sigma^* \rightarrow \Sigma^* \) producing poly size outputs
  - a poly time decidable language \( D \) such that:
    - if \( x \in L \) then \( \Pr_r(x \# q \# r \# y \in D) \geq 2/3 \)
      where \( q \equiv A(x,r), y \equiv M(x \# q \# r) \)
    - if \( x \notin L \) then \( \forall \) Merlin map \( M' \),
      \( \Pr_r(x \# q \# r \# y \in D) \leq 1/3 \)
      where \( q \equiv A(x,r), y \equiv M'(x \# q \# r) \)

Second (more benign) catch: I do not know of any correct proof of error reduction in the literature; and I do not know of any simple one.
The class AM, formally (final)

- $L$ is in AM iff $\forall$ polynomial $n \mapsto g(n)$, there are:
  - a poly time Turing machine $A$
  - a Merlin map $M : \Sigma^* \rightarrow \Sigma^*$ producing poly size outputs
  - a poly time decidable language $D$ such that:

- if $x \in L$ then $\Pr_r(x\#q\#r\#y \in D) \geq 1 - 1/2^{g(n)}$
  where $q \triangleq A(x,r)$, $y \triangleq M(x\#q\#r)$

- if $x \not\in L$ then $\forall$ Merlin map $M'$,
  $\Pr_r(x\#q\#r\#y \in D) \leq 1/2^{g(n)}$
  where $q \triangleq A(x,r)$, $y \triangleq M'(x\#q\#r)$
The Arthur-Merlin hierarchy

- In general, for any word \( w \equiv a_1a_2\ldots a_k \in \{A, M\}^* \), there is a class \( w \) (note: boldface), of languages \( L \) such that \( \forall g, \exists A, M, D: \)
  - If \( x \in L \) then \( \Pr_r(\text{prot}_w(M; x, r) \text{ accepts}) \geq 1 - 1/2^g(n) \)
  - if \( x \not\in L \) then \( \forall M', \Pr_r(\text{prot}_w(M'; x, r) \text{ accepts}) \leq 1/2^g(n) \)

- \( \text{prot}_w(M; x, r_1r_2\ldots r_k): \)
  \( \text{inp} := x \)
  for \( j=1\ldots k: \)
    - if \( a_j=A \) then \( (q_j := A(\text{inp},r_j); \text{inp} := \text{inp}\#r_j\#q_j) \)
    - else \( (y_j := M(\text{inp}); \text{inp} := \text{inp}\#y_j) \)
  accept if \( \text{inp} \in D \), else reject
The Arthur-Merlin hierarchy

- In general, for any word $w \equiv a_1a_2...a_k \in \{A, M\}^*$, there is a class $w$ (note: boldface), of languages $L$ such that $\forall g, \exists A, M, D$:
  - If $x \in L$ then $\Pr_r(\text{prot}_w(M; x, r) \text{ accepts}) \geq 1 - 1/2^g(n)$
  - If $x \notin L$ then $\forall M', \Pr_r(\text{prot}_w(M'; x, r) \text{ accepts}) \leq 1/2^g(n)$

- $\text{prot}_w(M; x, r_1r_2...r_k)$:
  \[
  \begin{align*}
  \text{inp} &:= x \\
  \text{for } j=1...k: \\
  \quad \text{if } a_j=A \text{ then } (q_j := A(\text{inp}, r_j); \text{inp} := \text{inp}\#r_j\#q_j) \\
  \quad \text{else } (y_j := M(\text{inp}); \text{inp} := \text{inp}\#y_j) \\
  \text{accept if } \text{inp} \in D, \text{ else reject}
  \end{align*}
  \]

Arthur’s turn. « draw $r_j$ at random », compute question $q_j$, add both to history $\text{inp}$
The Arthur-Merlin hierarchy

- In general, for any word \( w \equiv a_1a_2\ldots a_k \in \{A, M\}^* \), there is a class \( w \) (note: boldface), of languages \( L \) such that \( \forall g, \exists A, M, D: \)
  - If \( x \in L \) then \( \Pr_r(\text{prot}_w(M; x, r) \text{ accepts}) \geq 1 - 1/2^g(n) \)
  - If \( x \notin L \) then \( \forall M', \Pr_r(\text{prot}_w(M'; x, r) \text{ accepts}) \leq 1/2^g(n) \)

\[
\text{prot}_w(M; x, r_1r_2\ldots r_k): \\
\text{inp} := x \\
\text{for } j = 1 \ldots k: \\
\quad \text{if } a_j = A \text{ then } (q_j := A(\text{inp}, r_j); \text{inp} := \text{inp}\# r_j\# q_j) \\
\quad \text{else } (y_j := M(\text{inp}); \text{inp} := \text{inp}\# y_j) \\
\text{accept if } \text{inp} \in D, \text{ else reject}
\]

Arthur’s turn. « draw \( r_j \) at random », compute question \( q_j \), add both to history \( \text{inp} \)

Merlin’s turn. find answer \( y_j \), add it to history \( \text{inp} \)
The Arthur-Merlin hierarchy: the low levels

- When $w = \varepsilon (k=0)$, $\varepsilon =$?
The Arthur-Merlin hierarchy: the low levels

- When $w=\varepsilon$ ($k=0$): $\varepsilon=\mathsf{P}$
- When $w=A$: $A=?$

In general, for any word $w = a_1a_2...a_k \in \{A, M\}^*$, there is a class $w$ (note: boldface), of languages $L$ such that $\forall g, \exists A,M,D$:

- If $x \in L$ then $\Pr_r(\text{prot}_w(M; x, r) \text{ accepts}) \geq 1 - 1/2^g(n)$
- If $x \notin L$ then $\forall M', \Pr_r(\text{prot}_w(M'; x, r) \text{ accepts}) \leq 1/2^g(n)$

```
{prot}_w(M; x, r_1r_2...r_k):
  inp := x
  for j=1...k:
    if a_j=A then (q_j := A(inp, r_j); inp := inp#r_j#q_j)
    else (y_j := M(inp); inp := inp#y_j)
  accept if inp \in D, else reject
```
The Arthur-Merlin hierarchy: the low levels

- When $w=\varepsilon$ ($k=0$): $\varepsilon=P$
- When $w=A$: $A=BPP$
- When $w=M$: $M=?$

In general, for any word $w = a_1a_2...a_k \in \{A, M\}^*$, there is a class $w$ (note: boldface), of languages $L$ such that $\forall g, \exists A, M, D$

- If $x \in L$ then $\Pr_{r}(prot_w(M; x, r) \text{ accepts}) \geq 1 - 1/2^{g(n)}$
- If $x \notin L$ then $\forall M', \Pr_{r}(prot_w(M'; x, r) \text{ accepts}) \leq 1/2^{g(n)}$

```
prom(M; x, r_1r_2...r_k):
  inp := x
  for $j=1...k$:
    if $a_j=A$ then ($q_j := A(inp, r_j); inp := inp#r_j#q_j$)
    else ($y_j := M(inp); inp := inp#y_j$)
  accept if $inp \in D$, else reject
```

Arthur’s turn. « draw $r_j$ at random », compute question $q_j$, add both to history $inp$

Merlin’s turn. find answer $y_j$, add it to history $inp$
The Arthur-Merlin hierarchy: the low levels

- When $w=\varepsilon (k=0)$: $\varepsilon=\text{P}$
- When $w=A$: $A=\text{BPP}$
- When $w=M$: $M=\text{NP}$
The Arthur-Merlin hierarchy: the low levels

- When $w=\varepsilon$ ($k=0$): $\varepsilon=P$
- When $w=A$: $A=BPP$
- When $w=M$: $M=NP$
- Then we have $MA$, $AM$, $AMAM = AM[2]$, $AM[3]$, $\ldots$, $AM[k]$, $\ldots$

- In general, for any word $w = a_1a_2\ldots a_k \in \{A, M\}^*$, there is a class $\omega$ (note: boldface), of languages $L$ such that $\forall g, \exists A,M,D$:
  - If $x \in L$ then $\Pr_{\omega}(\text{prot}_\omega(M; x, r) \text{ accepts}) \geq 1 - 1/2^g(n)$
  - If $x \notin L$ then $\forall M', \Pr_{\omega}(\text{prot}_\omega(M'; x, r) \text{ accepts}) \leq 1/2^g(n)$

Arthur’s turn. «draw $r$ at random», compute question $q_r$, add both to history $inp$

Merlin’s turn. find answer $y_r$, add it to history $inp$
Interactive proofs
Interactive proofs

The Knowledge Complexity of Interactive Proof-Systems

(Extended Abstract)

Shafl Goldwasser
MIT
Silvio Micali
MIT
Charles Rackoff
University of Toronto

1. Introduction

In the first part of the paper we introduce a new theorem-proving procedure, that is a new efficient method of communicating a proof. Any such method implies, directly or indirectly, a definition of proof. Our "proofs" are probabilistic in nature. On a long statement it is correct with probability at least and right with probability at least. In many cases the method is very high and the method is of communication. We prove a statement we only ask the question: second part of the question: We propose to classify languages according to the amount of additional knowledge that must be released for proving membership in them.

Of particular interest is the case where this additional knowledge is essentially 0 and we show that is possible to interactively prove that a number is quadratic residue. Moreover, we exhibit the price of adding information. The amount of communication is required to increase the amount of knowledge that must be released in order to prove that a number is quadratic residue.

By Weizmann Institute of Science - Weizmann Institute of Science, Public Domain, https://commons.wikimedia.org/w/index.php?curid=12112705

By Rguillou228 - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=74039300

https://alchetron.com/cdn/charles-rackoff-3aa39129-7251-4443-9d07-4e01fcfd9c-resize-750.jpg
Interactive proofs

- Note that in Arthur-Merlin games, Arthur must communicate not just $q$ but also its random bits $r$ to Merlin.
Interactive proofs

- Note that in Arthur-Merlin games, Arthur must communicate not just $q$ but also its random bits $r$ to Merlin.

- In interactive proofs, Arthur only gives out $q$, and may therefore keep $q$ secret (but is not forced too).
The class AM

- $L$ is in AM iff $\forall$ polynomial $n \mapsto g(n)$, there are:
  - a poly time Turing machine $\mathcal{A}$
  - a Merlin map $M : \Sigma^* \rightarrow \Sigma^*$ producing poly size outputs
  - a poly time decidable language $D$ such that:

- if $x \in L$ then $\Pr_r(x \# q \# r \# y \in D) \geq 1 - 1/2^g(n)$
  where $q \overset{\text{def}}{=} \mathcal{A}(x, r)$, $y \overset{\text{def}}{=} M(x \# q \# r)$

- if $x \notin L$ then $\forall$ Merlin map $M'$,
  $\Pr_r(x \# q \# r \# y \in D) \leq 1/2^g(n)$
  where $q \overset{\text{def}}{=} \mathcal{A}(x, r)$, $y \overset{\text{def}}{=} M'(x \# q \# r)$
The class $\mathsf{AM} \not\subseteq \text{IP}[1]$

- $L$ is in $\text{IP}[1]$ iff $\forall$ polynomial $n \mapsto g(n)$, there are:
  - a poly time Turing machine $A$
  - a Merlin map $M : \Sigma^* \to \Sigma^*$ producing poly size outputs
  - a poly time decidable language $D$ such that:

- if $x \in L$ then $\Pr_r(x \# q \# r \# y \in D) \geq 1 - 1/2^g(n)$
  where $q \stackrel{\text{def}}{=} A(x, r)$, $y \stackrel{\text{def}}{=} M(x \# q \# r)$

- if $x \notin L$ then $\forall$ Merlin map $M'$,
  $\Pr_r(x \# q \# r \# y \in D) \leq 1/2^g(n)$
  where $q \stackrel{\text{def}}{=} A(x, r)$, $y \stackrel{\text{def}}{=} M'(x \# q \# r)$
The class $\text{AM} \neq \text{IP}[1]$

- $L$ is in $\text{IP}[1]$ iff $\forall$ polynomial $n \rightarrow g(n)$, there are:
  - a poly time Turing machine $A$
  - a Merlin map $M : \Sigma^* \rightarrow \Sigma^*$ producing poly size outputs
  - a poly time decidable language $D$ such that:

  - if $x \in L$ then $\Pr_r(x \# q \# r \# y \in D) \geq 1 - \frac{1}{2^{g(n)}}$
    where $q \equiv A(x, r)$, $y \equiv M(x \# q \# r)$
  
  - if $x \notin L$ then $\forall$ Merlin map $M'$,
    $\Pr_r(x \# q \# r \# y \in D) \leq \frac{1}{2^{g(n)}}$
    where $q \equiv A(x, r)$, $y \equiv M'(x \# q \# r)$

Note that $r$ still takes part in the final decision (and in Arthur’s computations, of course).
Example: Graph Isomorphism

- Let $V = \{1, \ldots, N\}$ set of vertices,
  
  $G_N \stackrel{\text{def}}{=} \text{directed graphs on } V,$

  $S_N \stackrel{\text{def}}{=} \text{group of permutations of } V.$
Example: Graph Isomorphism

- Let \( V = \{1, \ldots, N\} \) set of vertices,
  \( G_N \overset{\text{def}}{=} \text{directed graphs on } V, \)
  \( S_N \overset{\text{def}}{=} \text{group of permutations of } V. \)

- \( S_N \text{ acts on } G_N \) by: \( \forall \pi \in S_N, \forall G=(V,E) \in G_N, \)
  \( \pi.G \overset{\text{def}}{=} (V, \{(\pi(u), \pi(v)) \mid (u, v) \in E\}) \)
Example: Graph Isomorphism

- Let $V = \{1, \ldots, N\}$ set of vertices,
  $G_N \overset{\text{def}}{=} \text{directed graphs on } V$,
  $S_N \overset{\text{def}}{=} \text{group of permutations of } V$.

- $S_N$ acts on $G_N$ by: $\forall \pi \in S_N$, $\forall G=(V,E) \in G_N$,
  $\pi.G \overset{\text{def}}{=} (V, \{(\pi(u), \pi(v)) \mid (u, v) \in E\})$

- Two graphs
  $G_1=(V, E_1), G_2=(V, E_2)$ (with the same $V$)
  are isomorphic ($G_1 \equiv G_2$) iff $\exists \pi \in S_N$, $\pi.G_1=G_2$. 
Example: Graph Isomorphism

- **Graph isomorphism:**
  
  INPUT: 2 graphs $G_1=(V, E_1)$, $G_2=(V, E_2)$ (with the same $V$)
  
  QUESTION: are $G_1$, $G_2$ isomorphic?

  $S_N$ acts on $G_N$ by: $\forall \pi \in S_N$, $\forall G=(V,E) \in S_N$, $\pi.G \equiv (V, \{(\pi(u), \pi(v)) \mid (u, v) \in E\}$

  Two graphs $G_1=(V, E_1)$, $G_2=(V, E_2)$ (with the same $V$) are **isomorphic** iff $\exists \pi \in S_N$, $\pi.G_1=G_2$. 


Example: Graph Isomorphism

- **Graph isomorphism:**
  
  INPUT: 2 graphs $G_1=(V, E_1), G_2=(V, E_2)$ (with the same $V$)
  
  QUESTION: are $G_1, G_2$ isomorphic?

- Clearly in NP

$S_N$ acts on $G_N$ by: $\forall \pi \in S_N, \forall G=(V,E) \in S_N,
\pi.G \equiv (V, \{(\pi(u), \pi(v)) \mid (u, v) \in E\}$

Two graphs $G_1=(V, E_1), G_2=(V, E_2)$ (with the same $V$) are **isomorphic** iff $\exists \pi \in S_N, \pi.G_1=G_2$. 
Example: Graph Isomorphism

- **Graph isomorphism:**
  - INPUT: 2 graphs $G_1=(V, E_1)$, $G_2=(V, E_2)$ (with the same $V$)
  - QUESTION: are $G_1$, $G_2$ isomorphic?

- Clearly in **NP**
- Not known to be in **P**, nor **NP**-complete…
Example: Graph Isomorphism

- **Graph isomorphism:**
  INPUT: 2 graphs $G_1=(V, E_1)$, $G_2=(V, E_2)$ (with the same $V$)
  QUESTION: are $G_1$, $G_2$ isomorphic?

- Clearly in **NP**

- Not known to be in **P**, nor **NP**-complete...

- We will show, using results on **MA**, **AM**, **IP**[1], etc. that it is **not NP-complete** (unless **PH** collapses)

  (This is only the beginning: Babai gave a super polynomial time algo for GI in 2015; you need to understand first everything in the course to have a hope of understanding it!)
Example: Graph Non-Isomorphism

- $\text{GNI} \overset{\text{def}}{=} \text{complement of GI}: \text{in coNP, not known to be in P or coNP-complete}$
Example: Graph Non-Isomorphism

- $\text{GNI} \overset{\text{def}}{=} \text{complement of GI}$: in $\text{coNP}$, not known to be in $\text{P}$ or $\text{coNP}$-complete

- Prop. $\text{GNI}$ is in $\text{IP}[1]$. 

**GI**

INPUT: 2 graphs $G_1=(V, E_1), G_2=(V, E_2)$ (with the same $V$)

QUESTION: are $G_1, G_2$ isomorphic?
Example: Graph Non-Isomorphism

- **GNI** $\equiv$ complement of **GI**: in **coNP**, not known to be in **P** or **coNP**-complete

- **Prop.** **GNI** is in **IP**[1].

- **Algorithm.**
  — Arthur draws $i \in \{1,2\}$, $\pi \in S_N$ at random uniformly, sends $q \equiv \pi.G_i$
  — Merlin answers $j \in \{1,2\}$
  — We accept if $i=j$, reject otherwise.
Prop. GNI is in $\text{IP}[1]$. 

Proof. 
— If $(G_1, G_2) \in \text{GNI}$, there is a unique $j \in \{1,2\}$ such that $G_j \equiv \pi.G_i$, (viz., $i$) 
Merlin plays that $j$, forcing acceptance (always).
Prop. GNI is in IP[1].

Proof.
— If \((G_1, G_2) \notin \text{GNI}\),
  
  then \(G_1 \equiv G_2 \equiv \pi.G_i\) \(\text{(viz., } i\text{)}\)
  
  and Merlin has no information about \(i\)

Whatever Merlin plays, \(\Pr(\text{acceptance}) = 1/2\).
GNI is in $\text{IP}[1]$ (2/3)

- **Prop. GNI is in $\text{IP}[1]$.**

- **Proof.**
  - If $(G_1, G_2) \notin \text{GNI}$, then $G_1 \equiv G_2 \equiv \pi.G_i$, (viz., $i$)
  - and Merlin has **no information** about $i$
  - Whatever Merlin plays, $\Pr(\text{acceptance})=1/2$.

That is in fact irrelevant to the proof. But that shows that GNI has a **zero-knowledge** proof!
Prop. GNI is in $\text{IP}[1]$. 

Error too big (1/2).

$\Rightarrow$ Repeat experiments (à la $\text{RP}$), but in parallel.

— Arthur draws $i \in \{1,2\}$, $\pi \in S_N$ at random uniformly, sends $q \equiv \pi \cdot G_i$
— Merlin answers $j \in \{1,2\}$
— We accept if $i=j$, reject otherwise.
Prop. GNI is in $\text{IP}[1]$. 

$\text{Error too big (1/2).}$

$\Rightarrow$ Repeat experiments (à la $\text{RP}$), but in parallel.

$\text{—Arthur draws } g(n) \text{ bits } i_1, \ldots, i_{g(n)}$
  and $g(n)$ permutations $\pi_1, \ldots, \pi_{g(n)}$,
  sends $(\pi_1.G_{i_1}, \ldots, \pi_{g(n)}.G_{i_{g(n)})}$

$\text{—Merlin replies } j_1, \ldots, j_{g(n)}$
$\text{—We accept if } i_1=j_1 \text{ and } \ldots \text{ and } i_{g(n)}=j_{g(n)}, \text{ reject otherwise.}$
Prop. GNI is in IP[1].

Error too big (1/2).
⇒ Repeat experiments (à la RP), but in parallel.

— Arthur draws \( i \in \{1,2\} \), \( \pi \in S_N \) at random uniformly, sends \( q \equiv \pi.G_i \)
— Merlin answers \( j \in \{1,2\} \)
— We accept if \( i=j \), reject otherwise.

— Arthur draws \( g(n) \) bits \( i_1, \ldots, i_{g(n)} \)
and \( g(n) \) permutations \( \pi_1, \ldots, \pi_{g(n)} \),
sends \( (\pi_1.G_{i_1}, \ldots, \pi_{g(n)}.G_{i_{g(n)}}) \)
— Merlin replies \( j_1, \ldots, j_{g(n)} \)
— We accept if \( i_1=j_1 \) and … and \( i_{g(n)}=j_{g(n)} \), reject otherwise.

Error \( 1/2g(n) \) now (and still no error if \( (G_1, G_2) \in \text{GNI} \)).
GNI is in AM

- We will see later that GNI is in AM.
GNI is in AM

- We will see later that GNI is in AM.
- This is a better result, since AM \subseteq IP[1] (Any AM game can be simulated as an IP[1] game where Arthur sends both q and r as its question!)
GNI is in AM

- We will see later that GNI is in AM.
- This is a better result, since AM ⊆ IP[1]
  (Any AM game can be simulated as an IP[1] game
   where Arthur sends both q and r as its question!)
- In fact, AM = IP[1]... but this is a pretty hard result,
  due to Goldwasser and Sipser.
GNI is in AM

- We will see later that GNI is in AM.
- This is a better result, since AM ⊆ IP[1] (Any AM game can be simulated as an IP[1] game where Arthur sends both q and r as its question!)
- In fact, AM = IP[1]… but this is a pretty hard result, due to Goldwasser and Sipser.
- Meanwhile, let us return to the study of MA, AM, etc.
Other equivalent definitions of AM

1. BP·NP
The **BP·** operator

- **Generalizing BPP.**
  For any class \( C \), the class **BP·** \( C \):

- **A language** \( L \) **is in** **BP·** \( C \) **iff**
  there is a language \( D \) in \( C \), and a poly time TM \( M \) such that for every input \( x \) (of size \( n \)):
  — if \( x \in L \) then \( \Pr_r [M(x,r) \in D] \geq 2/3 \)
  — if \( x \not\in L \) then \( \Pr_r [M(x,r) \in D] \leq 1/3 \).
The \textbf{BP} · operator

\begin{itemize}
  \item Generalizing \textbf{BPP}.
    For any class \( C \), the class \( \text{BP} \cdot C \):
  \item A language \( L \) is in \( \text{BP} \cdot C \) iff there is a language \( D \) in \( C \), and a poly time TM \( M \) such that for every input \( x \) (of size \( n \)):
    \begin{itemize}
      \item if \( x \in L \) then \( \Pr_r [M(x,r) \in D] \geq 2/3 \)
      \item if \( x \notin L \) then \( \Pr_r [M(x,r) \in D] \leq 1/3 \).
    \end{itemize}
  \item In particular, \( \text{BP} \cdot \text{P} = \text{BPP} \).
\end{itemize}
Error reduction: democracy

- As for \textbf{BPP}, we can reduce the error in \textbf{BP} \cdot C from 1/3 to 1/2^{g(n)} for any polynomial g(n)

A language $L$ is in $\textbf{BP} \cdot C$ iff there is a language $D$ in $C$, and a poly time TM such that for every input $x$ (of size $n$):

- if $x \in L$ then $\Pr_r [M(x,r) \in D] \geq 2/3$
- if $x \notin L$ then $\Pr_r [M(x,r) \in D] \leq 1/3$. 
Error reduction: democracy

- As for BPP, we can reduce the error in $\text{BP} \cdot C$ from $1/3$ to $1/2^{g(n)}$ for any polynomial $g(n)$.

- ... provided that $C$ is democratic
  (non-standard name; obtained through a vote in class a few years ago)

A language $L$ is in $\text{BP} \cdot C$ iff there is a language $D$ in $C$, and a poly time TM $M$ such that for every input $x$ (of size $n$):
- if $x \in L$ then $\Pr_r [M(x,r) \in D] \geq 2/3$
- if $x \notin L$ then $\Pr_r [M(x,r) \in D] \leq 1/3$. 
As for BPP, we can reduce the error in $\text{BP} \cdot C$ from $1/3$ to $1/2^{g(n)}$ for any polynomial $g(n)$ provided that $C$ is democratic (non-standard name; obtained through a vote in class a few years ago).

Defn. $C$ is democratic iff for every $L \in C$,
$$\{w_1 \# \ldots \# w_k \mid \text{a majority of words } w_i \text{ is in } L\} \text{ is in } C.$$
Error reduction through democracy

- Let $L \in \text{BP} \cdot C$, with $D$ as here

A language $L$ is in $\text{BP} \cdot C$ iff there is a language $D$ in $C$, and a poly time $TM$ such that for every input $x$ (of size $n$):
- if $x \in L$ then $\Pr_r [M(x,r) \in D] \geq 2/3$
- if $x \notin L$ then $\Pr_r [M(x,r) \in D] \leq 1/3$.

Defn. $C$ is democratic iff for every $L \in C$, $\{w_1 \# \ldots \# w_k \mid \text{a majority of words } w_i \text{ is in } L\}$ is in $C$. 
Error reduction through democracy

- Let $L \in \text{BP} \cdot C$, with $D$ as here →
- Let $D' \equiv \{w_1#...#w_k \mid$ a majority of words $w_i$ is in $D\}$
  $D'$ is again in $C$

A language $L$ is in $\text{BP} \cdot C$ iff there is a language $D$ in $C$, and a poly time TM $M$, such that for every input $x$ (of size $n$):
  - if $x \in L$ then $\Pr_r [M(x,r) \in D] \geq 2/3$
  - if $x \notin L$ then $\Pr_r [M(x,r) \in D] \leq 1/3$.

Defn. $C$ is democratic iff for every $L \in C$,
$\{w_1#...#w_k \mid$ a majority of words $w_i$ is in $L\}$ is in $C$.

Application to voting (4/4)

- Assume that $\Pr_r [M(x,r) \text{ errs}] \leq 1/3$,
  how large should $N$ be so that the probability $P$ that more than $1/2$ of $N$ votes $M(x,r)$
  is $\leq 1/2^{g(n)}$?
- Answer: at least $36 \cdot g(n) \cdot \log 2$
- Proof. $\exp (\frac{-N}{36}) \leq 1/2^{g(n)}$ iff $-N/36 \leq -g(n) \log 2$

Note: if $g(n)$ is polynomial, this is polynomial, too

Application to voting (3/4)

- Assume that $\Pr_r [M(x,r) \text{ errs}] \leq 1/3$.
  - Assume that $\Pr_r [M(x,r) \text{ errs}] \leq 1/3$.
  - Assume that $\Pr_r [M(x,r) \text{ errs}] \leq 1/3$.
- Assume that $\Pr_r [M(x,r) \text{ errs}] \leq 1/3$.
Let \( L \in \text{BP} \cdot C \), with \( D \) as here →

Let \( D' \equiv \{ w_1 \# \cdots \# w_k \mid \) a majority of words \( w_i \) is in \( D \} \)

\( D' \) is again in \( C \)

It suffices to decide whether \( M(x, r_1) \# \cdots \# M(x, r_{36g(n)\log 2}) \in D' \)

... doable in \( C \) if \( C \) closed under poly time reductions

Defn. \( C \) is democratic iff for every \( L \in C \),
\( \{ w_1 \# \cdots \# w_k \mid \) a majority of words \( w_i \) is in \( L \} \) is in \( C \).

A language \( L \) is in \( \text{BP} \cdot C \) iff there is a language \( D \) in \( C \), and a poly time TM \( M \) such that for every input \( x \) (of size \( n \)):
- if \( x \in L \) then \( \Pr_r [M(x, r) \in D] \geq 2/3 \)
- if \( x \notin L \) then \( \Pr_r [M(x, r) \in D] \leq 1/3 \).

Application to voting (4/4)

- Assume that \( \Pr_r (M(x, r) \text{ errs}) \leq 1/3 \), how large should \( N \) be so that the probability \( P \) that more than \( 1/2 \) of \( N \) votes \( M(x, r) \) is \( \leq 1/2^{g(n)} \)?
- Answer: at least \( 36g(n) \log 2 \)

Proof. \( \exp(-N/36) \leq 1/2^{g(n)} \) iff \( -N/36 \leq -g(n) \log 2 \)

The only magical formula you'll need to remember for error reduction by majority voting

Note: if \( g(n) \) is polynomial, this is polynomial, too
Let \( L \in \text{BP} \cdot C \), with \( D \) as here →

Let \( D' \equiv \{ w_1\# \ldots \# w_k \mid \) a majority of words \( w_i \) is in \( D \}\)

\( D' \) is again in \( C \)

It suffices to decide whether \( M(x,r_1)\# \ldots \# M(x,r_{36g(n)\log 2}) \in D' \)

... doable in \( C \) if \( C \) closed under poly time reductions

Then error is \( \leq 1/2^{g(n)} \) (Chernoff!)

A language \( L \) is in \( \text{BP} \cdot C \) iff
there is a language \( D \) in \( C \), and a poly time TM \( M \)
such that for every input \( x \) (of size \( n \)):
— if \( x \in L \) then \( \Pr_r [M(x,r) \in D] \geq 2/3 \)
— if \( x \notin L \) then \( \Pr_r [M(x,r) \in D] \leq 1/3 \).

Defn. \( C \) is democratic iff for every \( L \in C \),
\( \{ w_1\# \ldots \# w_k \mid \) a majority of words \( w_i \) is in \( L \}\) is in \( C \).

Application to voting (4/4)

Assume that \( \Pr_r (M(x,r) \text{ errs}) \leq 1/3 \),
how large should \( N \) be so that the probability \( P \) that more than \( 1/2 \)
of \( N \) votes \( M(x,r) \) all err \( \leq 1/2^{g(n)\log 2} \)?

\[ \text{Answer: at least } 36 \; g(n) \; \text{ og } 2 \]

Proof. \( \exp(-N/36) \leq 1/2^{g(n)} \) iff
\( -N/36 \leq -g(n) \log 2 \)

Note: if \( g(n) \) is polynomial, this is polynomial, too
We have proved:

**Thm.** Let $C$ be democratic and closed under poly time reductions, and $g(n)$ be a polynomial. Then $\text{BP} \cdot C$, is also the class of languages $L$ such that [...]:

— if $x \in L$ then $\Pr_r [M(x,r) \in D] \geq 1 - \frac{1}{2^g(n)}$
— if $x \notin L$ then $\Pr_r [M(x,r) \in D] \leq \frac{1}{2^g(n)}$.

**Defn.** $C$ is **democratic** iff for every $L \in C$, \{w_1\ldots#w_k | a majority of words $w_i$ is in $L$\} is in $C$.

A language $L$ is in $\text{BP} \cdot C$ iff there is a language $D$ in $C$, and a poly time TM $M$ such that for every input $x$ (of size $n$):

— if $x \in L$ then $\Pr_r [M(x,r) \in D] \geq \frac{2}{3}$
— if $x \notin L$ then $\Pr_r [M(x,r) \in D] \leq \frac{1}{3}$.

**Application to voting (4/4)**

- Assume that $\Pr_r[M(x,r) \text{ errs}] \leq \frac{1}{3}$, how large should $N$ be so that the probability $P$ that more than $1/2$ of $N$ votes $M(x,r)$ is $\leq 1/2^g(n)$?
- Answer: at least $36 \cdot g(n) \log 2$
- Proof: $\exp(-N/36) \leq 1/2^{g(n)}$ iff $-N/36 \leq -g(n) \log 2$
Error reduction through democracy

- We have proved:

**Thm.** Let $C$ be democratic and closed under poly time reductions, and $g(n)$ be a polynomial. Then $\text{BP} \cdot C$, is also the class of languages $L$ such that [...]:
  - if $x \in L$ then $\Pr_r [M(x,r) \in D] \geq 1 - 1/2g(n)$
  - if $x \notin L$ then $\Pr_r [M(x,r) \in D] \leq 1/2g(n)$.

**Defn.** $C$ is **democratic** iff for every $L \in C$, \{\text{majority of words $w_i$ is in $L$}\} is in $C$.

A language $L$ is in $\text{BP} \cdot C$ iff there is a language $D$ in $C$, and a poly time TM $M$, such that for every input $x$ (of size $n$):
  - if $x \in L$ then $\Pr_r [M(x,r) \in D] \geq 2/3$
  - if $x \notin L$ then $\Pr_r [M(x,r) \in D] \leq 1/3$.

Application to voting (4/4)

- Assume that $\Pr_r([M(x,r) \text{ errs}] \leq 1/3$, *how large* should $N$ be so that the probability $P$ that more than $1/2$ of $N$ votes $M(x,r)$ is $\leq 1/2g(n)$?

Answer: at least $36 g(n) \log 2$

- Proof. $\exp(-N/36) \leq 1/2g(n)$ iff $-N/36 \leq -g(n) \log 2$
Examples of democratic classes

Fact. P is democratic. (Easy.)

Defn. C is democratic iff for every $L \in C$, 
\[ \{w_1 \# \cdots \# w_k \mid \text{a majority of words } w_i \text{ is in } L \} \text{ is in } C. \]
Examples of democratic classes

- **Fact.** P is democratic. (Easy.)
- **Prop.** NP is democratic.

**Defn.** C is democratic iff for every $L \in C$,

\[ \{w_1 \# \ldots \# w_k \mid \text{a majority of words } w_i \text{ is in } L \} \text{ is in } C. \]
Examples of democratic classes

- Fact. \( P \) is democratic. (Easy.)
- Prop. \( NP \) is democratic.
- No, we cannot check whether each \( w_i \) is in \( L \), because if that check fails, then the whole computation fails.

Defn. \( C \) is democratic \iff \( \text{for every } L \in C, \{ w_1 \# \ldots \# w_k \mid \text{a majority of words } w_i \text{ is in } L \} \text{ is in } C. \)
Examples of democratic classes

- **Fact.** P is democratic. (Easy.)
- **Prop.** NP is democratic.
- No, we cannot check whether each $w_i$ is in $L$, because if that check fails, then the whole computation fails.
- Instead, we **guess** a subset $I$ of indices of $\geq k/2$ elements, and we check that $\forall i \in I, w_i$ is in $L$.  

**Defn.** $C$ is democratic iff for every $L \in C$, 
\[ \{w_1 \# \ldots \# w_k \mid \text{a majority of words } w_i \text{ is in } L\} \in C. \]
BP · NP has error reduction

Thm. $L \in \text{BP} \cdot \text{NP}$ iff $\forall \text{poly } g$, $\exists D \in \text{NP}$, poly time TM $M$ /

— if $x \in L$ then

$$\Pr_r [M(x,r) \in D] \geq 1 - 1/2^g(n)$$

— if $x \notin L$ then

$$\Pr_r [M(x,r) \in D] \leq 1/2^g(n).$$
Thm (Prop. 3.5). $\text{AM} = \text{BP} \cdot \text{NP}$.
Thm (Prop. 3.5). AM = BP·NP.

Proof (1/4). Let $L \in AM$, as here:

- If $x \in L$ then $\Pr_r(x\#q\#r#y \in D) \geq 1 - 1/2^{\varepsilon(n)}$
  where $q = A(x,r)$, $y = M(x\#q\#r)$
- If $x \notin L$ then $\forall$ Mermin map $M'$,
  $\Pr_r(x\#q\#r#y \in D) \leq 1/2^{\varepsilon(n)}$
  where $q = A(x,r)$, $y = M'(x\#q\#r)$
\textbf{AM = BP \cdot NP}

- Thm (Prop. 3.5). AM = BP \cdot NP.

- Proof (1/4). Let $L \in \text{AM}$, as here:

- Let $D' = \{x\#r \mid \exists y, x\#q\#r\#y \in D, \text{ where } q \overset{\text{def}}{=} A(x,r)\}$: $D'$ is in NP.
Thm (Prop. 3.5). \( AM = BP \cdot NP \).

Proof (1/4). Let \( L \in AM \), as here:

Let \( D' \triangleq \{ x \# r \mid \exists y, x \# q \# r \# y \in D, \text{ where } q \triangleq A(x,r) \} \):

\( D' \) is in \( NP \).

If \( x \in L \), \( \Pr_r(x \# r \in D') \)

\[ = \Pr_r(\exists y, x \# q \# r \# y \in D, \text{ where } q \triangleq A(x,r)) \]

\[ \geq \Pr_r(x \# q \# r \# y \in D, \text{ where } q \triangleq A(x,r), y \triangleq M(x \# q \# r)) \]

\[ \geq 1 - 1/2^{g(n)} \]
Thm (Prop. 3.5). \( \text{AM} = \text{BP} \cdot \text{NP} \).

Proof (1/4). Let \( L \in \text{AM} \), as here:

Let \( D' \equiv \{ x \# r \mid \exists y, x \# q \# r \# y \in D, \text{where} q \equiv A(x,r) \} \):

\( D' \) is in \( \text{NP} \).

If \( x \in L \), \( \Pr_r(x \# r \in D') \)
\( = \Pr_r(\exists y, x \# q \# r \# y \in D, \text{where} q \equiv A(x,r)) \)
\( \geq \Pr_r(x \# q \# r \# y \in D, \text{where} q \equiv A(x,r), y \equiv M(x \# q \# r)) \)
\( \geq 1 - 1/2^g(n) \)

because \( \exists y, P(y) \) is implied by \( P(M(x \# q \# r)) \)
Thm (Prop. 3.5). AM = BP·NP.

Proof (2/4). Let \( L \in AM \), as here:

Let \( D' \defeq \{ x\#r \mid \exists y, x\#q\#r\#y \in D, \text{ where } q \defeq A(x,r) \} \): \( D' \) is in NP.
Thm (Prop. 3.5). AM = BP·NP.

Proof (2/4). Let \( L \in AM \), as here:

Let \( D' \overset{\text{def}}{=} \{ x \# r \mid \exists y, x \# q \# r \# y \in D, \text{ where } q \overset{\text{def}}{=} A(x, r) \} \): \( D' \) is in NP.

- if \( x \in L \) then \( \Pr_r(x \# q \# r \# y \in D) \geq 1 - 1/2^{s(n)} \)
  where \( q \overset{\text{def}}{=} A(x, r), y \overset{\text{def}}{=} M(x \# q \# r) \)
- if \( x \notin L \) then \( \forall \text{Merlin map } M' \),
  \( \Pr_r(x \# q \# r \# y \in D) \leq 1/2^{s(n)} \)
  where \( q \overset{\text{def}}{=} A(x, r), y \overset{\text{def}}{=} M'(x \# q \# r) \)
\[ \text{AM = BP} \cdot \text{NP} \]

- **Thm (Prop. 3.5).** AM = BP \cdot NP.

- **Proof (2/4).** Let \( L \in \text{AM} \), as here:

  - Let \( D' \triangleq \{ x \# r \mid \exists y, x \# q \# r \# y \in D, \text{ where } q \triangleq A(x,r) \} \): \( D' \) is in NP.

  - If \( x \not\in L \), then let \( M'(x \# q \# r) \triangleq \text{best of Merlin's responses, i.e., some } y \text{ such that } x \# q \# r \# y \in D \text{ if one exists} \)
Thm (Prop. 3.5). \( \text{AM} = \text{BP} \cdot \text{NP} \).

Proof (2/4). Let \( L \in \text{AM} \), as here:

- Let \( D' \overset{\text{def}}{=} \{ x \# r \mid \exists y, x \# q \# r \# y \in D, \text{ where } q \overset{\text{def}}{=} A(x, r) \} \): \( D' \) is in \( \text{NP} \).
  - If \( x \notin L \), then let \( M'(x \# q \# r) \overset{\text{def}}{=} \text{best of Merlin's responses}, \) i.e., some \( y \) such that \( x \# q \# r \# y \in D \) if one exists
  - Then \( \Pr_r(x \# r \in D') = \Pr_r(\exists y, x \# q \# r \# y \in D, \text{ where } q \overset{\text{def}}{=} A(x, r)) \leq \Pr_r(x \# q \# r \# M'(x \# q \# r) \in D, \text{ where } q \overset{\text{def}}{=} A(x, r)) \leq 1/2^{g(n)} \)

- If \( x \in L \) then \( \Pr_r(x \# q \# r \# y \in D) \geq 1 - 1/2^{g(n)} \) where \( q \overset{\text{def}}{=} A(x, r), y \overset{\text{def}}{=} M(x \# q \# r) \)
- If \( x \notin L \) then \( \forall \text{Merlin map } M', \) \( \Pr_r(x \# q \# r \# y \in D) \leq 1/2^{g(n)} \) where \( q \overset{\text{def}}{=} A(x, r), y \overset{\text{def}}{=} M'(x \# q \# r) \)
AM = BP·NP

- **Thm (Prop. 3.5).** AM = BP·NP.

- **Proof (2/4).** Let $L \in AM$, as here:

  - Let $D' \equiv \{x \# r \mid \exists y, x \# q \# r \# y \in D, \text{ where } q \equiv A(x,r)\}$: $D'$ is in NP.

  - If $x \notin L$, then let $M'(x \# q \# r) \equiv \text{best}$ of Merlin's responses, i.e., some $y$ such that $x \# q \# r \# y \in D$ if one exists

  - Then $\Pr_r(x \# r \in D')$
    
    \[= \Pr_r(\exists y, x \# q \# r \# y \in D, \text{ where } q \equiv A(x,r))\]
    
    \[\leq \Pr_r(x \# q \# r \# M'(x \# q \# r) \in D, \text{ where } q \equiv A(x,r))\]
    
    \[\leq 1/2^{g(n)}\]

  because $M'$ is best:

  \[(\exists y, x \# q \# r \# y \in D) \Rightarrow x \# q \# r \# M'(x \# q \# r) \in D\]
AM = BP·NP

- Thm (Prop. 3.5). AM = BP·NP.

- Proof (3/4). Let $L \in \text{BP} \cdot \text{NP}$, as here:
  Let $D = \{ q \mid \exists y, q#y \in D' \}$, with $D' \in \text{P}$,
  $D'' = \{ x#q#r#y \mid q#y \in D' \}$: in P.
Thm (Prop. 3.5). $\text{AM} = \text{BP} \cdot \text{NP}$.

Proof (3/4). Let $L \in \text{BP} \cdot \text{NP}$, as here:

Let $D \equiv \{q \mid \exists y, q\#y \in D'\}$, with $D' \in \text{P}$,

$D'' \equiv \{x\#q\#r\#y \mid q\#y \in D'\}$: in $\text{P}$.

Let $A(x, r) \equiv M(x, r)$, and $M(x\#q\#r) \equiv$ some $y$ such that $q\#y \in D'$ if one exists,
Thm (Prop. 3.5). AM = BP·NP.

Proof (3/4). Let \( L \in \text{BP} \cdot \text{NP} \), as here:

Let \( D \equiv \{ q \mid \exists y, q\#y \in D' \} \), with \( D' \in \text{P} \),
\[
D'' \equiv \{ x\#q\#r\#y \mid q\#y \in D' \}: \text{in } \text{P}.
\]

Let \( A(x,r) \equiv M(x,r) \), and \( M(x\#q\#r) \equiv \text{some } y \text{ such that } q\#y \in D' \text{ if one exists}, \)

If \( x \in L \), \( \Pr_r(x\#q\#r\#y \in D'', \text{where } q \equiv A(x,r), y \equiv M(x\#q\#r)) \)
\[
= \Pr_r(q\#y \in D', \text{where } q \equiv A(x,r), y \equiv M(x\#q\#r))
\geq \Pr_r(\exists y, q\#y \in D', \text{where } q \equiv A(x,r))
\geq \Pr_r(\text{M}(x,r) \in D) \geq 1 - 1/2^g(n)
\]
Thm (Prop. 3.5). $\text{AM} = \text{BP} \cdot \text{NP}$.

Proof (3/4). Let $L \in \text{BP} \cdot \text{NP}$, as here:

Let $D \equiv \{q \mid \exists y, q\#y \in D'\}$, with $D' \in \text{P}$,

$D'' \equiv \{x\#q\#r\#y \mid q\#y \in D'\}$: in $\text{P}$.

Let $A(x,r) \not\equiv M(x,r)$, and $M(x\#q\#r) \not\equiv$ some $y$ such that $q\#y \in D'$ if one exists,

If $x \in L$, $\Pr_r(x\#q\#r\#y \in D'')$, where $q \not\equiv A(x,r)$, $y \not\equiv M(x\#q\#r)$

$= \Pr_r(q\#y \in D'$, where $q \not\equiv A(x,r)$, $y \not\equiv M(x\#q\#r)$

$\geq \Pr_r(\exists y, q\#y \in D'$, where $q \not\equiv A(x,r))$

$= \Pr_r(M(x,r) \in D) \geq 1 - 1/2^g(n)$

because $M$ is best:

$(\exists y, q\#y \in D') \Rightarrow q\#M(x\#q\#r) \in D'$
Thm (Prop. 3.5). \( \text{AM} = \text{BP} \cdot \text{NP} \).

Proof (4/4). Let \( L \in \text{BP} \cdot \text{NP} \), as here:
Let \( D \equiv \{ q \mid \exists y, q \# y \in D' \} \), with \( D' \in \text{P} \),
\( D'' \equiv \{ x \# q \# r \# y \mid q \# y \in D' \} \): in \( \text{P} \).

Let \( A(x,r) \equiv M(x,r) \)
**AM = BP·NP**

- **Thm (Prop. 3.5).** AM = BP·NP.

- **Proof (4/4).** Let $L \in \text{BP} \cdot \text{NP}$, as here:
  Let $D \triangleq \{ q \mid \exists y, \, q\#y \in D' \}$, with $D' \in \text{P}$,
  $D'' \triangleq \{ x\#q\#r\#y \mid q\#y \in D' \}$: in \text{P}.

- Let $A(x,r) \triangleq M(x,r)$

- If $x \notin L$, for any $M'$, $\Pr_r(x\#q\#r\#y \in D'', \text{where } q \triangleq A(x,r), \, y \triangleq M'(x\#q\#r))$
  
  
  $\leq \Pr_r(\exists y, \, q\#y \in D', \text{where } q \triangleq A(x,r))$

  $\leq \Pr_r(M(x,r) \in D) \leq 1/2^{g(n)}$  $\Box$

- **Thm.** $L \in \text{BP} \cdot \text{NP}$ iff $\forall \text{poly } g,$
  $\exists D \in \text{NP},$ poly time TM $M$ /
  — if $x \in L$ then
    $\Pr_r[M(x,r) \in D] \geq 1-1/2^{g(n)}$
  — if $x \notin L$ then
    $\Pr_r[M(x,r) \in D] \leq 1/2^{g(n)}.$
Thm (Prop. 3.5).  \( \text{AM} = \text{BP} \cdot \text{NP} \).

Proof (4/4).  Let \( L \in \text{BP} \cdot \text{NP} \), as here:
Let \( D \triangleq \{ q \mid \exists y, q#y \in D' \} \), with \( D' \in \text{P} \),
\( D'' \triangleq \{ x#q#r#y \mid q#y \in D' \} \): in \( \text{P} \).

Let \( A(x,r) \triangleq M(x,r) \)

If \( x \notin L \), for any \( M' \),
\[ \Pr_r(x#q#r#y \in D'', \text{ where } q \triangleq A(x,r), y \triangleq M'(x#q#r)) \]
\[ = \Pr_r(q#y \in D', \text{ where } q \triangleq A(x,r), y \triangleq M'(x#q#r)) \]
\[ \leq \Pr_r(\exists y, q#y \in D', \text{ where } q \triangleq A(x,r)) \]
\[ = \Pr_r(M(x,r) \in D) \leq 1/2^g(n) \quad \square \]

because \( \exists y, P(y) \) is implied by \( P(M'(x#q#r)) \)
Other equivalent definitions of AM

2. Extended quantifiers
Let us say that Arthur is **lazy** if Arthur does not bother to compute any question: \( A(x,r) = \varepsilon \)

\[
\text{prot}_{\text{lazy}}(M; x, r_1 r_2 \ldots r_k):
\]
\[
inp := x
\]
\[
\text{for } j=1 \ldots k:
\]
\[
\text{if } a_j = A \text{ then } (q_j := A(inp, r_j); inp := inp\#r_j\#q_j)
\]
\[
\text{else } (y_j := M(inp); inp := inp\#y_j)
\]
\[
\text{accept if } inp \in D, \text{ else reject}
\]
Lazy Arthur

- Let us say that Arthur is lazy if Arthur does not bother to compute any question: $A(x,r) = \varepsilon$

- **Prop (Lemma 3.8).** For every word $w \in \{A, M\}^*$, the class $w_{\text{lazy}}$ when Arthur is constrained to be lazy is equal to the class $w$.

- **Proof.** See lecture notes. Idea: Merlin is so powerful he can reconstruct Arthur’s questions without Arthur’s help. □

```plaintext
prot_{w_{\text{lazy}}}(M; x, r_1r_2\ldots r_k):
inp := x
for j=1\ldots k:
   if $a_j=A$ then ($q_j := A(inp,r_j); inp := inp#r_j#q_j$)
   else ($y_j := M(inp); inp := inp#y_j$)
accept if $inp \in D$, else reject
```
A logical approach

- Model both Arthur and Merlin as quantifiers (over $r, y$)
- ... for « predicates » with values in $[0, 1]$ over finite sets
- **Arthur** (expectation):
  \[ E_{r \in R, F(r)} = \frac{\sum_{r \in R} F(r)}{\text{card } R} \]
- **Merlin** (maximize):
  \[ \exists y \in Y, F(y) = \max_{y \in Y} F(y) \]
  (Note: if $F$ takes its values in $\{0,1\}$, this is really the existential quantifier...
A small catch

- The notations $\mathbb{E}$, $\exists$ are practical, e.g.:
  - $(\exists y \in Y, F(y)) \geq a$ iff there is a $y \in Y$ such that $F(y)) \geq a$
A small catch

- The notations $E$, $\exists$ are practical, e.g.:

- $(\exists y \in Y, F(y)) \geq a$ iff there is a $y \in Y$ such that $F(y)) \geq a$.

- But beware that $(\exists y \in Y, F(y)) \leq a$ iff for every $y \in Y$, $F(y) \leq a$. 

<table>
<thead>
<tr>
<th><strong>Arthur</strong> (expectation):</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E r \in R, F(r)$</td>
</tr>
<tr>
<td>$\equiv \sum_{r \in R} F(r) / \text{card } R$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Merlin</strong> (maximize):</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\exists y \in Y, F(y)$</td>
</tr>
<tr>
<td>$\equiv \max_{y \in Y} F(y)$</td>
</tr>
</tbody>
</table>
Prop. $E_r \in R, \exists y \in Y, F(r, y)$
$= \exists f : R \rightarrow Y, E_r \in R, F(r, f(r))$

- **Arthur** (expectation):
  $E_r \in R, F(r) \; \# \; \sum_{r \in R} F(r) / \text{card } R$

- **Merlin** (maximize):
  $\exists y \in Y, F(y) \; \# \; \max_{y \in Y} F(y)$
Prop. \( E_r \in R, \exists y \in Y, F(r,y) \)
\[ = \exists f : R \rightarrow Y, \ E_r \in R, \ F(r,f(r)) \]

Proof (1/2).
Let \( f(r) \triangleq \text{best } y \), viz. some \( y \) that maximizes \( F(r,y) \)
Then \( \exists y \in Y, F(r,y) = F(r,f(r)) \)
Prop. \( \mathbb{E}r \in R, \exists y \in Y, F(r, y) \)  
\[= \exists f : R \to Y, \mathbb{E}r \in R, F(r, f(r)) \]

Proof (1/2).  
Let \( f(r) \overset{\text{def}}{=} \text{best } y\), viz. some \( y \) that maximizes \( F(r, y) \)  
Then \( \exists y \in Y, F(r, y) = F(r, f(r)) \)

Take expectations:  
\( \mathbb{E}r \in R, \exists y \in Y, F(r, y) = \mathbb{E}r \in R, F(r, f(r)) \)
Prop. $\exists r \in R, \exists y \in Y, F(r, y) = \exists f : R \to Y, \mathbb{E}r \in R, F(r, f(r))$

Proof (1/2).
Let $f(r) \triangleq \text{best } y$, viz. some $y$ that maximizes $F(r, y)$
Then $\exists y \in Y, F(r, y) = F(r, f(r))$

Take expectations:
$\mathbb{E}r \in R, \exists y \in Y, F(r, y) = \mathbb{E}r \in R, F(r, f(r))$

$\ldots \leq \max_{f : R \to Y} \mathbb{E}r \in R, F(r, f(r))$
Prop. \( \forall r \in R, \exists y \in Y, F(r,y) \]
\[= \exists f : R \to Y, \forall r \in R, F(r,f(r)) \]

Proof (2/2).
For every \( f \), \( F(r,f(r)) \leq \max_{y \in Y} F(r,y) \)
Prop. \( E_r \in R, \exists y \in Y, F(r, y) \)
\[= \exists f : R \rightarrow Y, E_r \in R, F(r, f(r)) \]

Proof (2/2).
For every \( f \), \( F(r, f(r)) \leq \max_{y \in Y} F(r, y) \)

Take expectations:
\[ E_r \in R, F(r, f(r)) \leq E_r \in R, \max_{y \in Y} F(r, y) \]
Prop. \( \forall r \in R, \exists y \in Y, F(r,y) \)

\[ = \exists f : R \rightarrow Y, \forall r \in R, F(r,f(r)) \]

Proof (2/2).
For every \( f \), \( F(r,f(r)) \leq \max_{y \in Y} F(r,y) \)

Take expectations:
\[ \forall r \in R, F(r,f(r)) \leq \forall r \in R, \max_{y \in Y} F(r,y) \]

Now take max over \( f \). \( \square \)
« Skolemization »: an example

- $E_{r_1}, \exists y_1, E_{r_2}, \exists y_2, F(x, r_1, y_1, r_2, y_2)$
Skolemization: an example

- $E r_1, \exists y_1, E r_2, \exists y_2, F(x, r_1, y_1, r_2, y_2)$
- $= \exists f_1, E r_1, E r_2, \exists y_2, F(x, r_1, y_1, r_2, y_2)$
  where $y_1 \overset{\text{def}}{=} f_1(r_1)$

Arthur (expectation):
$E r \in R, F(r) \overset{\text{def}}{=} \sum_{r \in R} F(r) \ /	ext{card} \ r$

Merlin (maximize):
$\exists y \in Y, F(y) \overset{\text{def}}{=} \max_{y \in Y} F(y)$

Prop. $E r \in R, \exists y \in Y, F(r, y)$
$= \exists f : R \rightarrow Y, E r \in R, F(r, f(r))$
« Skolemization »: an example

- $E r_1, \exists y_1, E r_2, \exists y_2, F(x, r_1, y_1, r_2, y_2)$
  - $= \exists f_1, E r_1, E r_2, \exists y_2, F(x, r_1, y_1, r_2, y_2)$
    where $y_1 \stackrel{\text{def}}{=} f_1(r_1)$
  - $= \exists f_1, f_2, E r_1, r_2, F(x, r_1, y_1, r_2, y_2)$
    where $y_1 \stackrel{\text{def}}{=} f_1(r_1), y_2 \stackrel{\text{def}}{=} f_2(r_1, r_2)$

Arthur (expectation):
$E r \in R, F(r) \equiv \sum_{r \in R} F(r) / \text{card } R$

Merlin (maximize):
$\exists y \in Y, F(y) \equiv \max_{y \in Y} F(y)$

Prop.
$E r \in R, \exists y \in Y, F(r, y) \equiv \exists f : R \rightarrow Y, E r \in R, F(r, f(r))$
Let $F$ be $\{0,1\}$-valued (not $[0,1]$) i.e., a predicate

- **Arthur** (expectation):
  
  $$E_r \in R, F(r) \equiv \sum_{r \in R} F(r) / \text{card } R$$

- **Merlin** (maximize):
  
  $$\exists y \in Y, F(y) \equiv \max_{y \in Y} F(y)$$

**Prop.** $E_r \in R, \exists y \in Y, F(r,y) = \exists f \colon R \to Y, E_r \in R, F(r,f(r))$
Let $F$ be $\{0,1\}$-valued (not $[0,1]$) i.e., a **predicate**

Recall that $\exists y \in Y, F(y)$ (=max) is then the existential quantifier (... and is therefore $\{0,1\}$-valued)
Let $F$ be \{0,1\}-valued (not [0,1]), i.e., a **predicate**

Recall that $\exists y \in Y, F(y)$ (=max) is then the existential quantifier (… and is therefore \{0,1\}-valued)

Also, $\mathbb{E}_r, F(r) = \Pr_r(F(r)=1)$

(« expectation of a predicate = its probability of occurring »)
Prop (3.10). \( L \in \text{AMAM} \) iff for every polynomial \( g(n) \), there is a poly time predicate \( P \) /
— if \( x \in L \), then \( G(x) \geq 1 - 1/2^{g(n)} \)
— if \( x \notin L \) then \( G(x) \leq 1/2^{g(n)} \)

where \( G(x) \equiv \exists r_1, \exists y_1, \exists r_2, \exists y_2, P(x, r_1, y_1, r_2, y_2) \)

(I will let you generalize to other classes of the A-M hierarchy)
Prop (3.10). \( L \in AMAM \) iff 
for every polynomial \( g(n) \), 
there is a poly time predicate \( P \) /
— if \( x \in L \), then \( G(x) \geq 1 - 1/2g(n) \)
— if \( x \notin L \) then \( G(x) \leq 1/2g(n) \)

where \( G(x) = E r_1, \exists y_1, E r_2, \exists y_2, P(x, r_1, y_1, r_2, y_2) \)

Proof (1/5). \( G(x) = \exists f_1, f_2, E r_1, r_2, P(x, r_1, y_1, r_2, y_2) \)

where \( y_1 \triangleq f_1(r_1), y_2 \triangleq f_2(r_1, r_2) \) « skolemization »
Prop (3.10). $L \in \text{AMAM}$ iff
for every polynomial $g(n)$,
there is a poly time predicate $P$ /
— if $x \in L$, then $G(x) \geq 1 - 1/2^{g(n)}$
— if $x \notin L$ then $G(x) \leq 1/2^{g(n)}$
where $G(x) \equiv E_{r_1}, \exists y_1, E_{r_2}, \exists y_2, P(x, r_1, y_1, r_2, y_2)$

Proof (1/5). $G(x) = \exists f_1, f_2, E_{r_1}, r_2, P(x, r_1, y_1, r_2, y_2)$
where $y_1 \equiv f_1(r_1), y_2 \equiv f_2(r_1, r_2)$

Hence $G(x) = \exists f_1, f_2, P_{r_1, r_2}(x \# r_1 \# y_1 \# r_2 \# y_2 \in D)$
where $D \equiv \{x \# r_1 \# y_1 \# r_2 \# y_2 \mid P(x, r_1, y_1, r_2, y_2) = 1\}$ (note: $D \in \mathbb{P}$)
A-M as E-∃ formulae

- Proof (2/5). If $L \in \text{AMAM}$ with Merlin map $M$ and a lazy Arthur,

- if $x \in L$ then let $f_1(r_1) \overset{\text{def}}{=} M(x\#r_1)$ (in short, $y_1$)

  $f_2(r_1,r_2) \overset{\text{def}}{=} M(x\#r_1\#f_1(r_1)\#r_2)$ (y_2)

Prop (3.10). $L \in \text{AMAM}$ iff for every polynomial $g(n)$,
for every polynomial $g(n)$,
— if $x \in L$, then $G(x) \geq 1 - 1/2^{g(n)}$
— if $x \not\in L$ then $G(x) \leq 1/2^{g(n)}$

where $G(x) \equiv \exists f_1, f_2, P_{r_1, r_2}(x\#r_1\#y_1\#r_2\#y_2 \in D)$
and $y_1 \equiv f_1(r_1), y_2 \equiv f_2(r_1, r_2)$
A-M as E-∃ formulae

- Proof (2/5). If \( L \in \text{AMAM} \) with Merlin map \( M \) and a lazy Arthur,

- if \( x \in L \) then let \( f_1(r_1) \overset{\text{def}}{=} M(x \# r_1) \) (in short, \( y_1 \))
  \[
  f_2(r_1,r_2) \overset{\text{def}}{=} M(x \# r_1 \# f_1(r_1) \# r_2) \quad (y_2)
  \]

- Then \( G(x) = \exists f_1, f_2, \text{Pr}_{r_1, r_2}(x \# r_1 \# y_1 \# r_2 \# y_2 \in D) \)
  \[
  \geq \text{Pr}_{r_1, r_2}(x \# r_1 \# y_1 \# r_2 \# y_2 \in D)
  \geq 1 - 1/2^g(n)
  \]

Prop (3.10). \( L \in \text{AMAM} \) iff for every polynomial \( g(n) \),
- if \( x \in L \), then \( G(x) \geq 1 - 1/2^g(n) \)
- if \( x \notin L \), then \( G(x) \leq 1/2^g(n) \)
  where \( G(x) \equiv \exists f_1, f_2, \text{Pr}_{r_1, r_2}(x \# r_1 \# y_1 \# r_2 \# y_2 \in D) \)
  and \( y_1 \equiv f_1(r_1), y_2 \equiv f_2(r_1, r_2) \)
A-M as E-∃ formulae

- Proof (3/5). If $L \in \text{AMAM}$ with Merlin map $M$ and a lazy Arthur,

- if $x \notin L$ then for all maps $f_1, f_2$,
    
    let $M'(x \# r_1) \equiv f_1(r_1)$, $M'(x \# r_1 \# y_1 \# r_2) \equiv f_2(r_1, r_2)$
    
    and $M'$ of anything else be arbitrary (e.g., $\varepsilon$)

Prop (3.10). $L \in \text{AMAM}$ iff for every polynomial $g(n)$,
there is a poly time predicate $P /$
- if $x \in L$, then $G(x) \geq 1 - 1/2g(n)$
- if $x \notin L$ then $G(x) \leq 1/2g(n)$
where $G(x) \equiv \exists f_1, f_2, \Pr_{r_1, r_2}(x \# r_1 \# y_1 \# r_2 \# y_2 \in D)$
and $y_1 = f_1(r_1), y_2 = f_2(r_1, r_2)$
Proof (3/5). If \( L \in \text{AMAM} \) with Merlin map \( M \) and a lazy Arthur,

if \( x \not\in L \) then for all maps \( f_1, f_2, \)
let \( M'(x \# r_1) \defeq f_1(r_1), \ M'(x \# r_1 \# y_1 \# r_2) \defeq f_2(r_1, r_2) \)
and \( M' \) of anything else be arbitrary (e.g., \( \varepsilon \))

Then \( G(x) \leq \Pr_r(x \# r_1 \# y_1 \# r_2 \# y_2 \in D) \leq 1/2^{g(n)} \)
where \( y_1 \defeq M'(x \# r_1), \ y_2 \defeq M'(x \# r_1 \# y_1 \# r_2) \)
Proof (4/5). If \( L \) is as here \( \rightarrow \)

for each \( x \in L \) there are maps \( f_1, f_2 \)
such that

\[
\Pr_{r_1, r_2}(x \# r_1 \# y_1 \# r_2 \# y_2 \in D) \geq 1 - 1/2^g(n)
\]

where \( y_1 \equiv f_1(r_1), \ y_2 \equiv f_2(r_1, r_2) \)

Prop (3.10). \( L \in \text{AMAM} \) iff
for every polynomial \( g(n) \),
there is a poly time predicate \( P \)
— if \( x \in L \), then \( G(x) \geq 1 - 1/2^g(n) \)
— if \( x \notin L \), then \( G(x) \leq 1/2^g(n) \)

where \( G(x) \equiv \exists f_1, f_2, \Pr_{r_1, r_2}(x \# r_1 \# y_1 \# r_2 \# y_2 \in D) \)
and \( y_1 \equiv f_1(r_1), \ y_2 \equiv f_2(r_1, r_2) \)
A-M as E-∃ formulae

Proof (4/5). If $L$ is as here →

- for each $x \in L$ there are maps $f_1, f_2$ such that
  \[ \Pr_{r_1, r_2}(x \# r_1 \# y_1 \# r_2 \# y_2 \in D) \geq 1 - 1/2^{g(n)} \]
  where $y_1 \equiv f_1(r_1)$, $y_2 \equiv f_2(r_1, r_2)$

- Let $M(x \# r_1) \equiv f_1(r_1)$, $M(x \# r_1 \# y_1 \# r_2) \equiv f_1(r_1, r_2)$, else arbitrary

Prop (3.10). $L \in \text{AMAM}$ iff
for every polynomial $g(n)$,
there is a poly time predicate $P$
— if $x \in L$, then $G(x) \geq 1 - 1/2^{g(n)}$
— if $x \notin L$ then $G(x) \leq 1/2^{g(n)}$

where $G(x) \equiv \exists f_1, f_2, \Pr_{r_1, r_2}(x \# r_1 \# y_1 \# r_2 \# y_2 \in D)$
and $y_1 \equiv f_1(r_1), y_2 \equiv f_2(r_1, r_2)$
Proof (4/5). If $L$ is as here →

- for each $x \in L$ there are maps $f_1, f_2$ such that
  \[
  \Pr_{r_1, r_2}(x \# r_1 \# y_1 \# r_2 \# y_2 \in D) \geq 1 - 1/2g(n)
  \]
  where $y_1 \equiv f_1(r_1), y_2 \equiv f_2(r_1, r_2)$

- Let $M(x \# r_1) \equiv f_1(r_1), M(x \# r_1 \# y_1 \# r_2) \equiv f_1(r_1, r_2)$, else arbitrary

- If $x \in L$ then $\Pr_r(x \# r_1 \# y_1 \# r_2 \# y_2 \in D)$ ($y_1 \equiv M(x \# r_1), y_2 \equiv M(x \# r_1 \# y_1 \# r_2)$)
  \[
  = \Pr_{r_1, r_2}(x \# r_1 \# y_1 \# r_2 \# y_2 \in D) (y_1 \equiv f_1(r_1), y_2 \equiv f_2(r_1, r_2))
  \]
  \[
  \geq 1 - 1/2g(n)
  \]

Prop (3.10). $L \in \text{AMAM}$ iff
for every polynomial $g(n)$,
there is a poly time predicate $P /$
— if $x \in L$, then $G(x) \geq 1 - 1/2g(n)$
— if $x \notin L$ then $G(x) \leq 1/2g(n)$
where $G(x) \equiv \exists f_1, f_2, Pr_{r_1, r_2}(x \# r_1 \# y_1 \# r_2 \# y_2 \in D)$
  and $y_1 \equiv f_1(r_1), y_2 \equiv f_2(r_1, r_2)$
Proof (5/5). If $L$ is as here →

If $x \notin L$ then

for every Merlin map $M'$,

let $f_1(r_1) \overset{\text{def}}{=} M'(x \# r_1)$ (in short, $y_1$)

$f_2(r_1, r_2) \overset{\text{def}}{=} M'(x \# r_1 \# f_1(r_1) \# r_2)$ (y_2)

Prop (3.10). $L \in \text{AMAM}$ iff

for every polynomial $g(n)$,

there is a poly time predicate $P$

— if $x \in L$, then $G(x) \geq 1 - 1/2g(n)$
— if $x \notin L$ then $G(x) \leq 1/2g(n)$

where $G(x) \equiv \exists f_1, f_2, P_{r_1, r_2}(x \# r_1 \# y_1 \# r_2 \# y_2 \in D)$

and $y_1 \equiv f_1(r_1), y_2 \equiv f_2(r_1, r_2)$
Proof (5/5). If $L$ is as here →

If $x \notin L$ then

for every Merlin map $M'$,

let $f_1(r_1) \overset{\text{def}}{=} M'(x\#r_1)$ (in short, $y_1$)

$f_2(r_1, r_2) \overset{\text{def}}{=} M'(x\#r_1\#f_1(r_1)\#r_2)$ (y_2)

Then $\Pr_r(x\#r_1\#y_1\#r_2\#y_2 \in D)$ ($y_1 \overset{\text{def}}{=} M'(x\#r_1)$, $y_2 \overset{\text{def}}{=} M'(x\#r_1\#y_1\#r_2)$)

$= \Pr_{r_1, r_2}(x\#r_1\#y_1\#r_2\#y_2 \in D)$ ($y_1 \overset{\text{def}}{=} f_1(r_1)$, $y_2 \overset{\text{def}}{=} f_2(r_1, r_2)$)

$\leq G(x) \leq 1/2^g(n)$. □
Next time...
The Arthur-Merlin hierarchy collapses!

- We will see that the whole Arthur-Merlin hierarchy looks like this!

\[
\begin{align*}
P & \subseteq \text{BPP} \\
\text{NP} & \subseteq \text{MA} \\
\text{AM} & \subseteq \text{P} \\
\text{(All other classes} & \text{equal to AM)}
\end{align*}
\]
The Arthur-Merlin hierarchy collapses!

- We will see that the whole Arthur-Merlin hierarchy looks like this!

\[ \vdash \subseteq \ P \subseteq \ BPP \subseteq \ NP \subseteq \ MA \subseteq \ AM \] (All other classes \( w \) equal to \( AM \))