Jean Goubault-Larrecq

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

László Babai

(STOC'1985)

Par Schmid, Renate — `https://opc.mfo.de/detail?photo_id=14372`, CC BY-SA 2.0 de, `https://commons.wikimedia.org/w/index.php?curid=18096981`

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 [BS] we proved, using the elements of the theory of *nilpotent groups*, that some of the *fundamental computational problems in matrix groups* be
These problems were also shown to belong
assuming an *unproven hypothesis* concerning
groups.

The aim of this paper is to replace most of
and unproven) group theory of [BS] by elem
binatorial arguments. The result we prove is
to a *random oracle B*, the mentioned matrix
lems belong to $(NP \cap coNP)^B$.

The problems we consider are *membership*
of a matrix group given by a list of generators.
lems can be viewed as multidimensional versio
relative of the discrete logarithm probl
$NP \cap coNP$ might be the lowest natural com
they may fit in.

We remark that the results remain valid f

1. Introduction

1.1. Randomness vs. mathematical intractability: a tradeoff

JOURNAL OF COMPUTER AND SYSTEM SCIENCES **36**, 254–276 (1988)

Arthur–Merlin 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 NP as the complexity class that captures the notion of efficient provability by classical (formal) proofs. We consider broader complexity classes (still "just above NP"), in the hope to formalize 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) (*in* "Proceedings, 17th ACM Symp. Theory of Comput., Providence, RI, 1985," pp. 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

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…

x

?

yes!

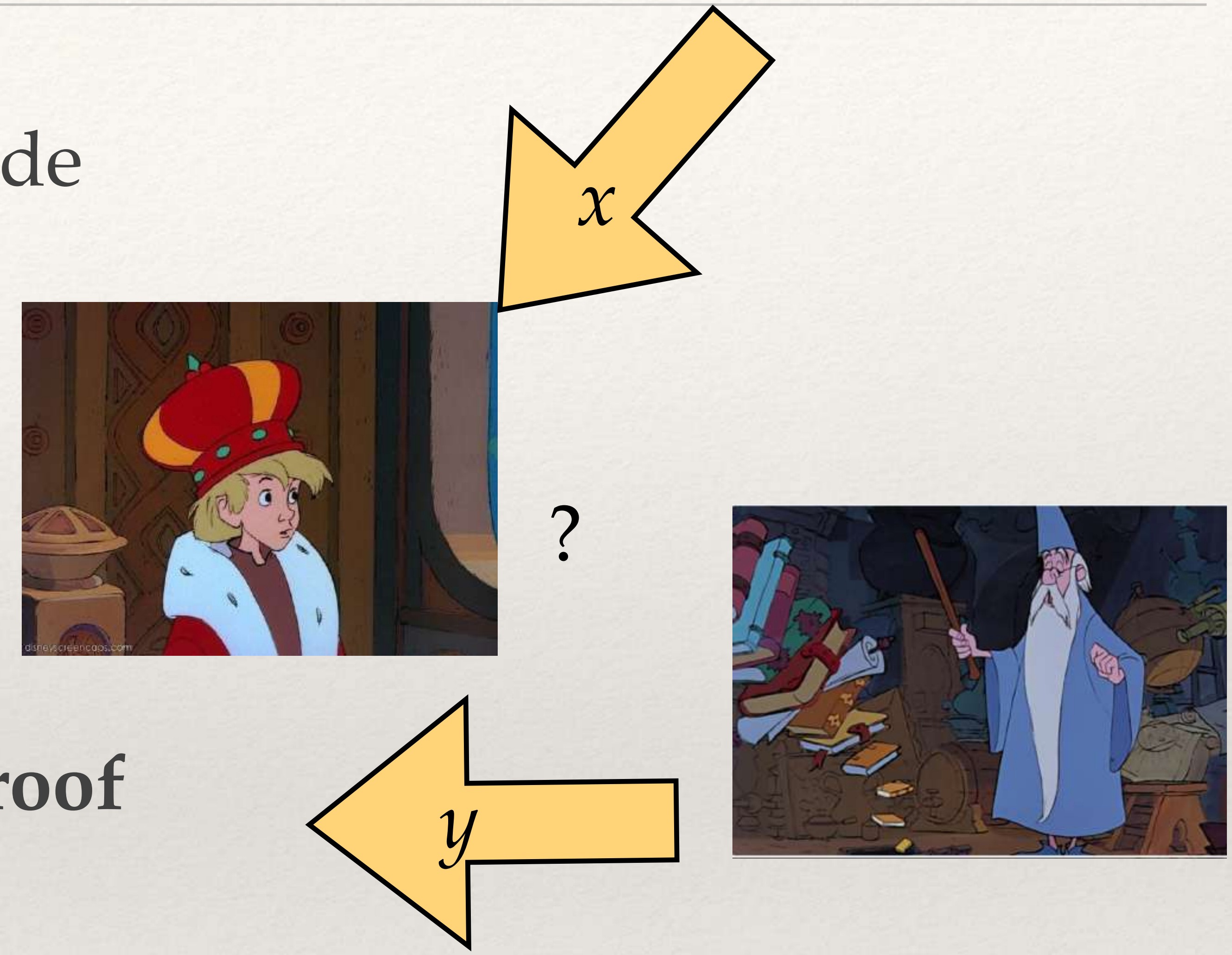
(trust me)

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

Merlin: `https://www.ecranlarge.com/uploads/image/001/011/merlin-l-enchanteur-photo-merlin-disney-1011190.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
- now Arthur can check Merlin's proof… provided y has polynomial size

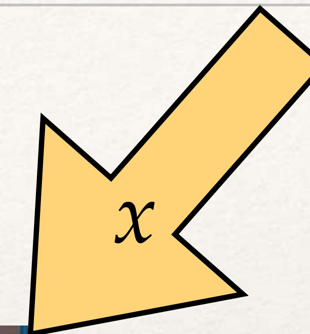


Arthur: `http://lusile17.l.u.pic.centerblog.net/273f716e.jpg`

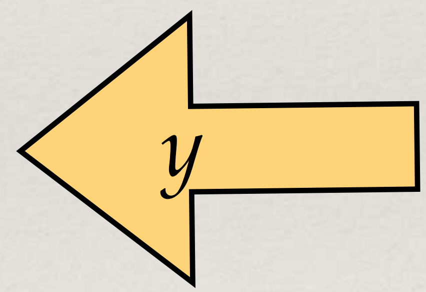
Merlin: `https://www.ecranlarge.com/uploads/image/001/011/merlin-l-enchanteur-photo-merlin-disney-1011190.jpg`

Arthur vs. Merlin games

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



?

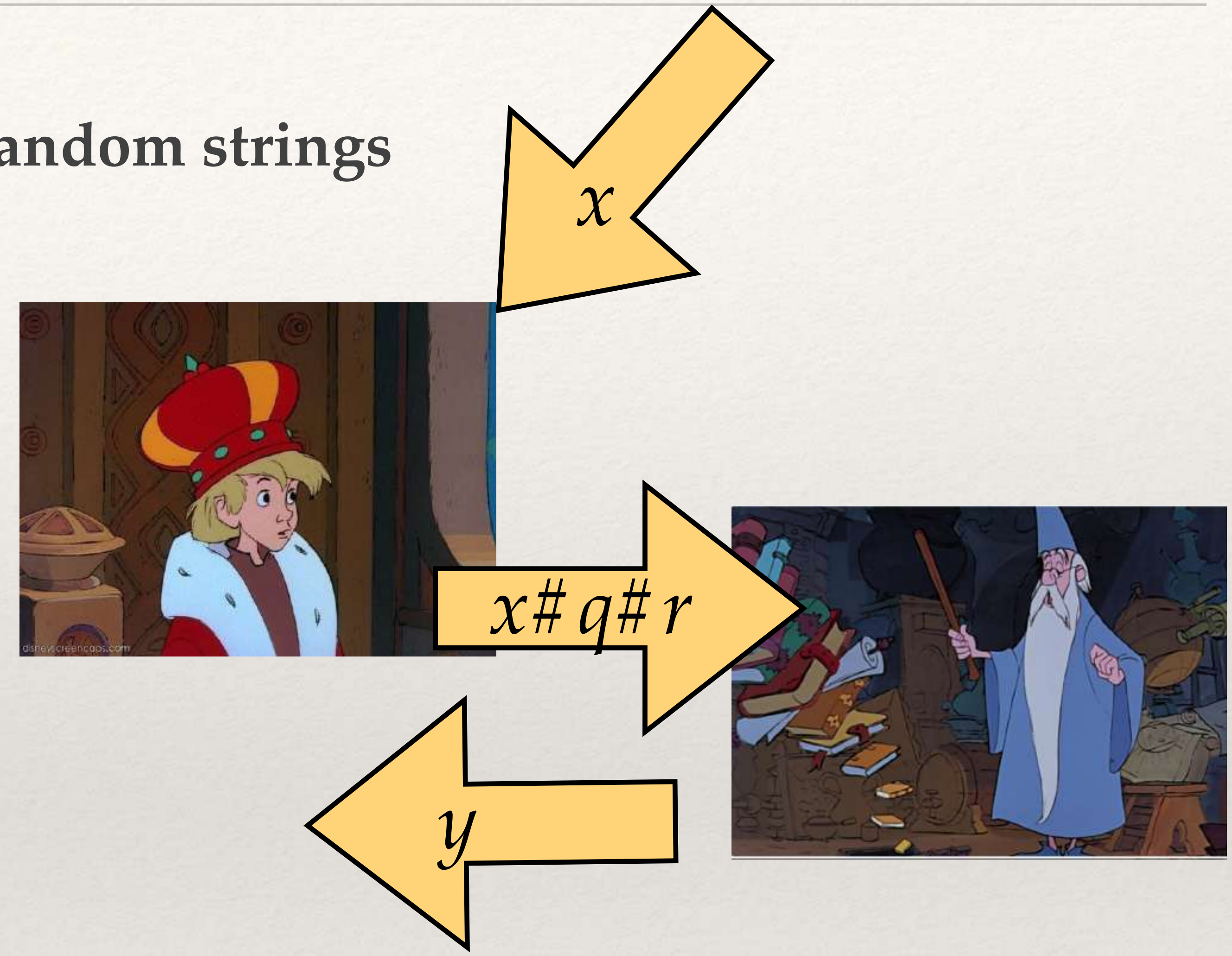


Arthur: `http://lusile17.l.u.pic.centerblog.net/273f716e.jpg`

Merlin: `https://www.ecranlarge.com/uploads/image/001/011/merlin-l-enchanteur-photo-merlin-disney-1011190.jpg`

The class **AM**

- Now Arthur can also draw (uniform) **random strings**
- INPUT: x
- Arthur draws r at random
  and computes a question
  $$q \stackrel{\text{def}}{=} \mathcal{A}(x,r)$$
- … 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.



… with a catch!
(in fact, two)

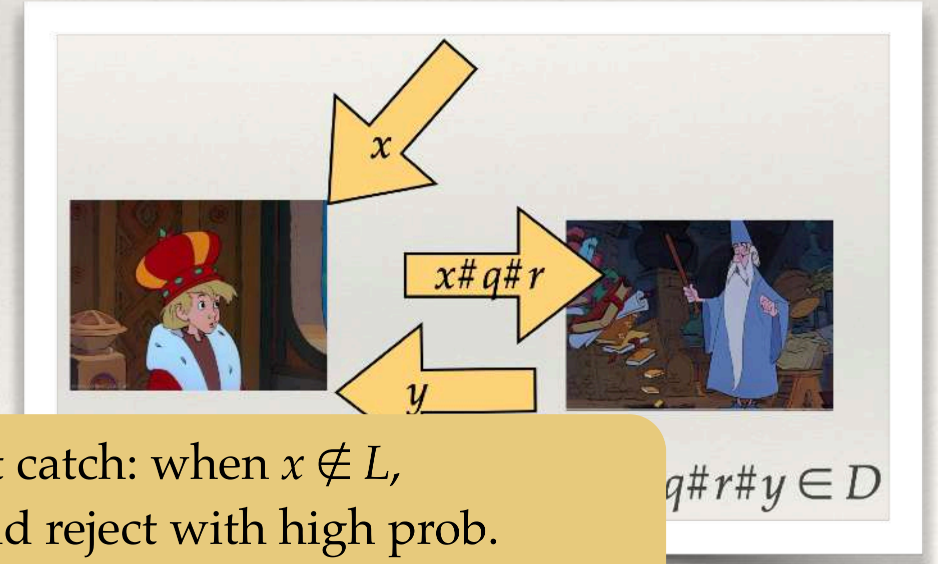
Arthur: `http://lusile17.l.u.pic.centerblog.net/273f716e.jpg`

Merlin: `https://www.ecranlarge.com/uploads/image/001/011/merlin-l-enchanteur-photo-merlin-disney-1011190.jpg`

The class **AM**, formally (1st try)

- L is in **AM** iff there are:
— a **poly time** Turing machine $\mathcal{A}$
(used by Arthur to compute questions $q \stackrel{\text{def}}{=} \mathcal{A}(x,r)$)
— 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 $q \stackrel{\text{def}}{=} \mathcal{A}(x,r)$, $y \stackrel{\text{def}}{=} M(x\#q\#r)$

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

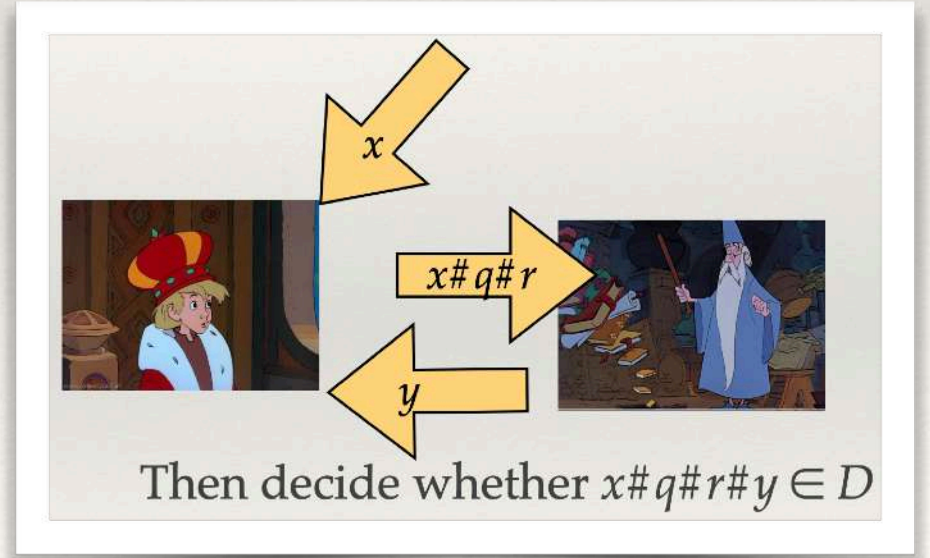
The class **AM**, formally (2nd try)

- L is in **AM** iff there are:
  — a **poly time** Turing machine $\mathcal{A}$
  — 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 $q \overset{\text{def}}{=} \mathcal{A}(x,r)$, $y \overset{\text{def}}{=} M(x\#q\#r)$

- if $x \notin L$ then $\forall$Merlin map M',
  $\Pr_r(x\#q\#r\#y \in D) \leq 1/3$
  where $q \overset{\text{def}}{=} \mathcal{A}(x,r)$, $y \overset{\text{def}}{=} M'(x\#q\#r)$

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.

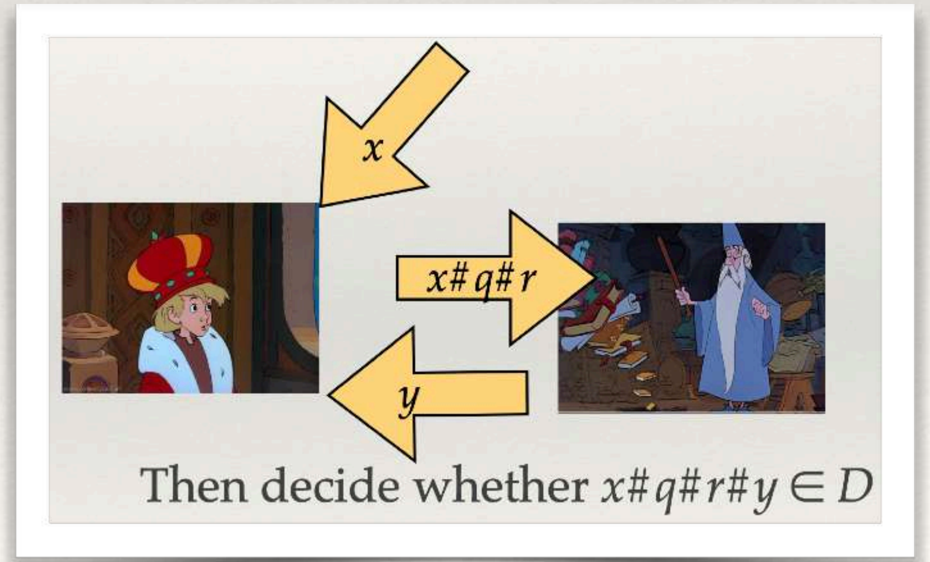


The class **AM**, formally (final)

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

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

- if $x \notin L$ then $\forall$**Merlin map** M',
  $$\Pr_r(x\#q\#r\#y \in D) \leq 1/2^{g(n)}$$
  where $q \stackrel{\text{def}}{=} \mathcal{A}(x,r)$, $y \stackrel{\text{def}}{=} \boldsymbol{M'}(x\#q\#r)$



The Arthur-Merlin hierarchy

- In general, for any word $w \stackrel{\text{def}}{=} a_1 a_2 \ldots a_k \in \{\texttt{A}, \texttt{M}\}^*$,
  there is a class **w** (note: boldface), of languages L such that $\forall g$, $\exists \mathcal{A}, M, D$:

- If $x \in L$ then $\Pr_{\underline{r}}(\mathrm{prot}_w(M; x, \underline{r})$ accepts$) \geq 1 - 1/2^{g(n)}$

- if $x \notin L$ then $\forall \boldsymbol{M'}$, $\Pr_{\underline{r}}(\mathrm{prot}_w(\boldsymbol{M'}; x, \underline{r})$ accepts$) \leq 1/2^{g(n)}$

- $\mathrm{prot}_w(M; x, r_1 r_2 \ldots r_k)$:
  $inp := x$
  `for` j=1…k:
  `if` a_j=`A` `then` $(q_j := \mathcal{A}(inp, r_j);\ inp := inp\#r_j\#q_j)$
  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=ε (k=0), ε=?

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

- $\mathrm{prot}_w(M; x, r_1r_2\ldots r_k)$:
  $inp := x$
  for j=1…k:
      if a_j=**A** **then** $(q_j := \mathcal{A}(inp, r_j);\ inp := inp\#r_j\#q_j)$
      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=ε (k=0): **ε=P**
- When w=`A`: **A**=?

- In general, for any word $w \triangleq a_1 a_2 \ldots a_k \in \{\texttt{A}, \texttt{M}\}^*$, there is a class $\boldsymbol{w}$ (note: boldface), of languages L such that $\forall g, \exists \mathcal{A}, M, D$:
- If $x \in L$ then $\Pr_{\underline{r}}(\mathrm{prot}_w(M; x, \underline{r}) \text{ accepts}) \geq 1-1/2^{g(n)}$
- if $x \notin L$ then $\forall M'$, $\Pr_{\underline{r}}(\mathrm{prot}_w(M'; x, \underline{r}) \text{ accepts}) \leq 1/2^{g(n)}$
- prot$_w$(M; x, $r_1 r_2 \ldots r_k$):
  $inp := x$
  `for` j=1…k:
    `if` a_j=`A` `then` ($q_j := \mathcal{A}(inp, r_j)$; $inp := inp\#r_j\#q_j$)
    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=ε (k=0): **ε=P**
- When w=`A`: **A=BPP**
- When w=`M`: **M=?**

- In general, for any word $w \doteq a_1a_2\ldots a_k \in \{\texttt{A}, \texttt{M}\}^*$, there is a class **$w$** (note: boldface), of languages L such that $\forall g$, $\exists \mathcal{A}, M, D$:
- If $x \in L$ then $\Pr_{\underline{r}}(\text{prot}_w(M; x, \underline{r}) \text{ accepts}) \geq 1 - 1/2^{g(n)}$
- if $x \notin L$ then $\forall M'$, $\Pr_{\underline{r}}(\text{prot}_w(M'; x, \underline{r}) \text{ accepts}) \leq 1/2^{g(n)}$
- prot$_w$(M; x, $r_1r_2\ldots r_k$):
  inp := x
  `for` j=1…k:
  `if` a_j=`A` `then` (q_j := $\mathcal{A}$(*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 $w=\varepsilon$ ($k=0$): $\boldsymbol{\varepsilon}=\mathbf{P}$
- When $w=\mathtt{A}$: $\mathbf{A}=\mathbf{BPP}$
- When $w=\mathtt{M}$: $\mathbf{M}=\mathbf{NP}$
- Then we have **MA**, **AM**,
  AMAM = **AM**[2], **AM**[3], …,
  AM[k], …

- In general, for any word $w \triangleq a_1a_2\ldots a_k \in \{\mathtt{A}, \mathtt{M}\}^*$, there is a class $\boldsymbol{w}$ (note: boldface), of languages L such that $\forall g, \exists \mathcal{A}, M, D$:
- If $x \in L$ then $\Pr_{\underline{r}}(\mathrm{prot}_w(M; x, \underline{r})\text{ accepts}) \geq 1-1/2^{g(n)}$
- if $x \notin L$ then $\forall M'$, $\Pr_{\underline{r}}(\mathrm{prot}_w(M'; x, \underline{r})\text{ accepts}) \leq 1/2^{g(n)}$
- $\mathrm{prot}_w(M; x, r_1r_2\ldots r_k)$:

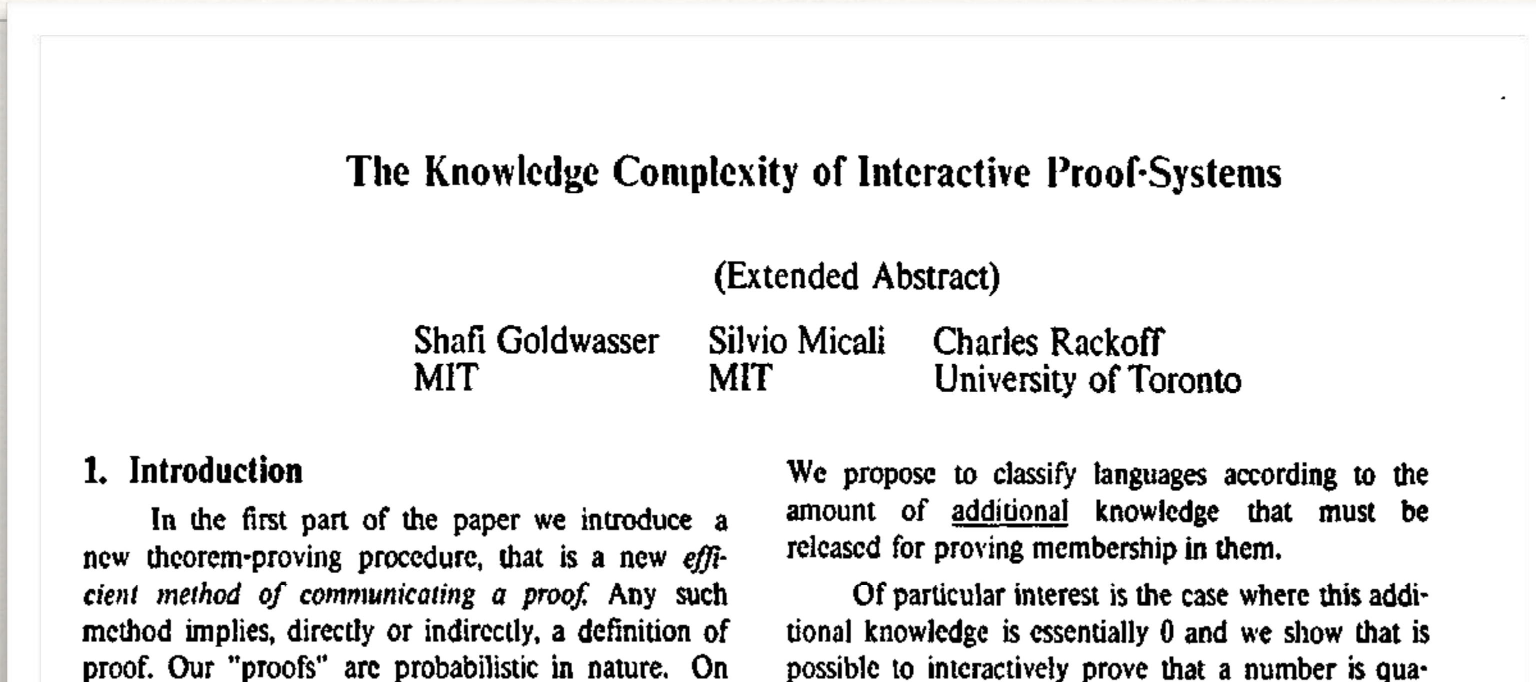
```
inp := x
for j=1...k:
    if a_j=A then (q_j := 𝒜(inp,r_j); inp := inp#r_j#q_j)
    else (y_j := M(inp); inp := inp#y_j)
accept if inp ∈ D, else reject
```

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*

Interactive proofs

Interactive proofs



The Knowledge Complexity of Interactive Proof-Systems

(Extended Abstract)

Shafi 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

We propose to classify languages according to the amount of <u>additional</u> 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 qua-

(STOC'1985 aussi!)

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-3aa39129-7251-4443-9d07-4e01fcfdc9c-resize-750.jpeg

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 r **secret** (but is not forced too).

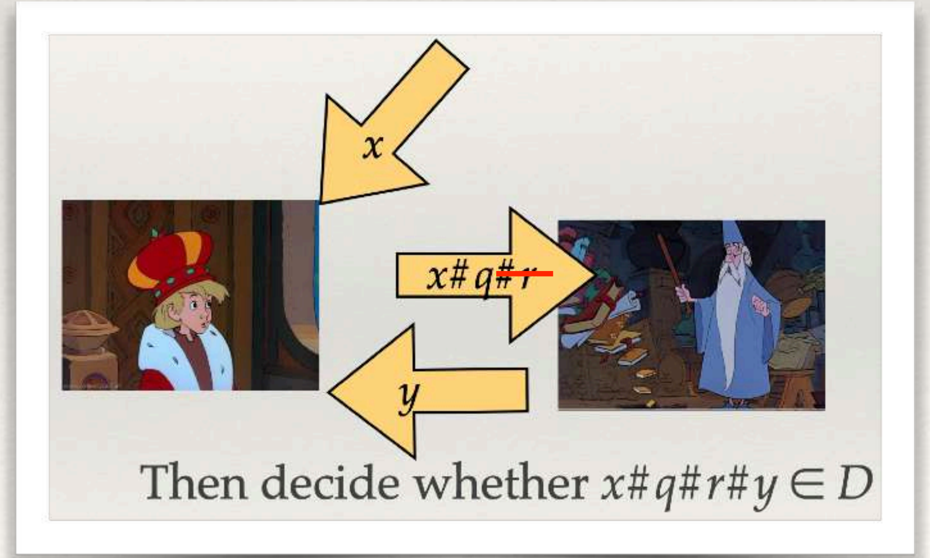
The class ~~AM~~ IP[1]

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

Note that r still takes part
in the final decision
(and in Arthur's computations, of course)

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

- if $x \notin L$ then $\forall$**Merlin map** M',
  $$\Pr_r(x\#q\#r\#y \in D) \leq 1/2^{g(n)}$$
  where $q \stackrel{\text{def}}{=} \mathcal{A}(x,r)$, $y \stackrel{\text{def}}{=} M'(x\#q\cancel{\#r})$



Example: Graph Isomorphism

- Let $V = \{1, \ldots, N\}$ set of vertices,
  $\boldsymbol{G}_N \stackrel{\text{def}}{=}$ directed graphs on V,
  $\mathbf{S}_N \stackrel{\text{def}}{=}$ group of permutations of V.

- $\mathbf{S}_N$ acts on $\boldsymbol{G}_N$ by: $\forall \pi \in \mathbf{S}_N$, $\forall G=(V,E) \in \boldsymbol{G}_N$,
  $\pi.G \stackrel{\text{def}}{=} (V, \{(\pi(u), \pi(v)) \mid (u, v) \in E\}$

- Two graphs
  $G_1=(V, E_1)$, $G_2=(V, E_2)$ (with the same V)
  are **isomorphic** ($G_1 \equiv G_2$) iff $\exists\, \pi \in \mathbf{S}_N$, $\pi.G_1=G_2$.

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: $\forall \pi \in \mathbf{S}_N$, $\forall G=(V,E) \in \mathbf{S}_N$,
$\pi.G \stackrel{\text{def}}{=} (V, \{(\pi(u), \pi(v)) \mid (u, v) \in E\}$

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

- 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

- **GNI** $\stackrel{\text{def}}{=}$ complement of **GI**: in **coNP**,
  not known to be in **P** or **coNP**-complete
- **Prop. GNI is in IP**[1].
- *Algorithm.*
  — Arthur draws $i \in \{1,2\}$, $\pi \in \mathbf{S}_N$ at random uniformly,
  sends $q \stackrel{\text{def}}{=} \pi.G_i$
  — Merlin answers $j \in \{1,2\}$
  — We accept if $i=j$, reject otherwise.

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?

GNI is in IP[1] (1/3)

— Arthur draws $i \in \{1,2\}$, $\pi \in \mathbf{S}_N$ at random uniformly, sends $q \stackrel{\text{def}}{=} \pi.G_i$
— Merlin answers $j \in \{1,2\}$
— We accept if i=j, reject otherwise.

- **Prop. GNI is in IP**[1].
- *Proof.*
  — If $(G_1, G_2) \in$ **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).

GNI is in IP[1] (2/3)

— Arthur draws $i \in \{1,2\}$, $\pi \in \mathbf{S}_N$ at random uniformly, sends $q \stackrel{\text{def}}{=} \pi.G_i$
— Merlin answers $j \in \{1,2\}$
— We accept if $i=j$, reject otherwise.

- **Prop. GNI** is in **IP**[1].
- *Proof.*
  — If $(G_1, G_2) \notin$ **GNI**,
  then $G_1 \equiv G_2 \equiv \pi.G_i$, (viz., i)
  and Merlin has **no information** about i
  Whatever Merlin plays, Pr(acceptance)=1/2.

That is in fact irrelevant to the proof.
But that shows that **GNI** has a
zero-knowledge proof!

GNI is in IP[1] (3/3)

> — Arthur draws $i \in \{1,2\}$, $\pi \in \mathbf{S}_N$ at random uniformly,
> sends $q \stackrel{\text{def}}{=} \pi.G_i$
> — Merlin answers $j \in \{1,2\}$
> — We accept if $i=j$, reject otherwise.

- **Prop. GNI** is in **IP**[1].
- Error too big (1/2).
  ⇒ Repeat experiments (à la **RP**), but **in parallel**.
- —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 $1/2^{g(n)}$ now (and still no error if $(G_1, G_2) \in$ **GNI**).

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!)
- 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 $\mathbf{BP} \cdot C$:

A language L is in **BPP** if and only if there is a **polynomial-time** TM $\mathcal{M}$ such that for every input x (of size n):

if $x \in L$ then $\Pr_r [\mathcal{M}(x,r) \text{ accepts}] \geq 2/3$

if $x \notin L$ then $\Pr_r [\mathcal{M}(x,r) \text{ accepts}] \leq 1/3$.

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

- In particular, $\mathbf{BP} \cdot \mathbf{P} = \mathbf{BPP}$.

Error reduction: democracy

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

A language L is in **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 $\Pr_r\,[\mathcal{M}(x,r) \in D] \geq 2/3$
— if $x \notin L$ then $\Pr_r\,[\mathcal{M}(x,r) \in D] \leq 1/3$.

- … 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 $L \in \mathbf{BP} \cdot C$, with D as here $\rightarrow$

> A language L is in $\mathbf{BP} \cdot C$ 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 $\Pr_r\,[\mathcal{M}(x,r) \in D] \geq 2/3$
> — if $x \notin L$ then $\Pr_r\,[\mathcal{M}(x,r) \in D] \leq 1/3$.

- Let $D' \stackrel{\text{def}}{=} \{w_1\# \ldots \# w_k \mid$ a majority of words w_i is in $D\}$

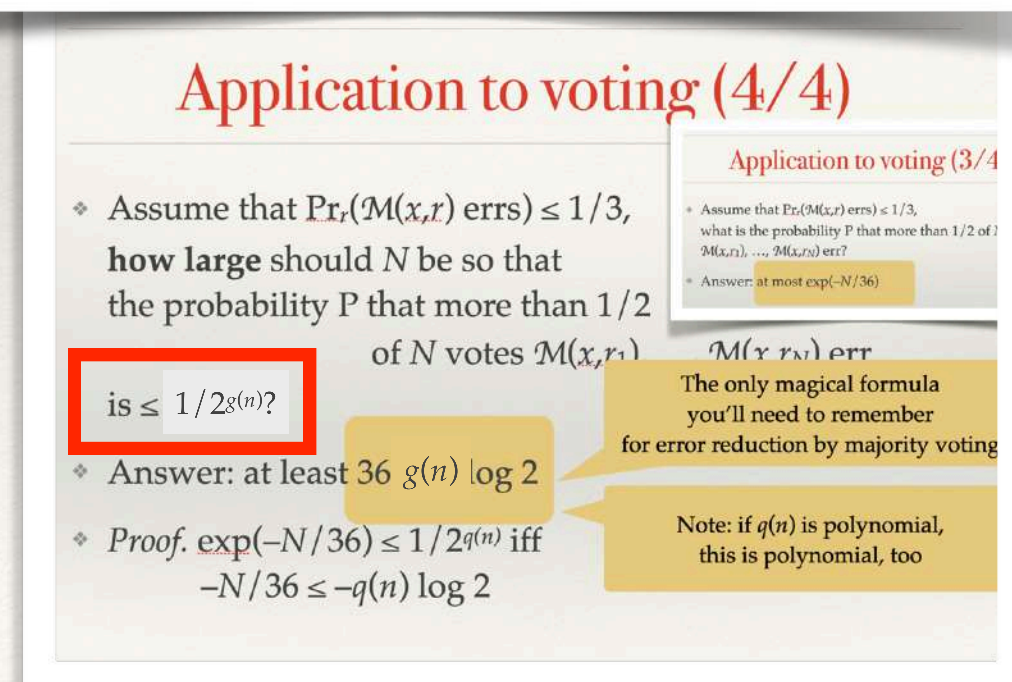
  D' is again in C

> **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.

- It suffices to decide whether

  $$\underbrace{\mathcal{M}(x,r_1)\#\ldots\#\mathcal{M}(x,r_{36g(n)\log 2})}_{\text{(in poly-time)}} \in D'$$

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



Error reduction through democracy

- We have proved:

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

A language L is in **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 $\Pr_r\,[\mathcal{M}(x,r) \in D] \geq 2/3$
— if $x \notin L$ then $\Pr_r\,[\mathcal{M}(x,r) \in D] \leq 1/3$.

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 $\Pr_r(\mathcal{M}(x,r)$ errs$) \leq 1/3$,
how large should N be so that
the probability P that more than $1/2$
of N votes $\mathcal{M}(x,r_1)$ ... $\mathcal{M}(x,r_N)$ err
is $\leq 1/2^{g(n)}$?
- Answer: at least $36\ g(n) \log 2$
- *Proof.* $\exp(-N/36) \leq 1/2^{q(n)}$ iff
$-N/36 \leq -q(n) \log 2$

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

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

Application to voting (3/4)

- Assume that $\Pr_r(\mathcal{M}(x,r)$ errs$) \leq 1/3$,
what is the probability P that more than 1/2 of
$\mathcal{M}(x,r_1)$, ..., $\mathcal{M}(x,r_N)$ err?
- Answer: at most $\exp(-N/36)$

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

- 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. $\square$

$\mathbf{BP} \cdot \mathbf{NP}$ has error reduction

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

AM = BP·NP

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

- **Thm (Prop. 3.5). AM = BP·NP.**
- Proof (1/4). Let $L \in$ **AM**, as here:
- Let $D' \stackrel{\text{def}}{=} \{x\#r \mid \exists y, x\#q\#r\#y \in D$, where $q \stackrel{\text{def}}{=} \mathcal{A}(x,r)\}$:

  D' is in **NP**.

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

    $= \Pr_r(\exists y, x\#q\#r\#y \in D$, where $q \stackrel{\text{def}}{=} \mathcal{A}(x,r))$

    $\geq \Pr_r(x\#q\#r\#y \in D$, where $q \stackrel{\text{def}}{=} \mathcal{A}(x,r)$, $y \stackrel{\text{def}}{=} M(x\#q\#r))$

    $\geq 1 - 1/2^{g(n)}$

because $\exists y, P(y)$ is implied
by $P(M(x\#q\#r))$

AM = BP·NP

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

- **Thm (Prop. 3.5). AM = BP·NP.**
- Proof (2/4). Let $L \in$ **AM**, as here:
- Let $D' \stackrel{\text{def}}{=} \{x\#r \mid \exists y, x\#q\#r\#y \in D$, where $q \stackrel{\text{def}}{=} \mathcal{A}(x,r)\}$: D' is in **NP**.
  - If $x \notin L$, then let $M'(x\#q\#r) \stackrel{\text{def}}{=}$ **best** of Merlin's responses, i.e., some y such that $x\#q\#r\#y \in D$ if one exists
  - Then $\Pr_r(x\#r \in D')$
    $= \Pr_r(\exists y, x\#q\#r\#y \in D$, where $q \stackrel{\text{def}}{=} \mathcal{A}(x,r))$
    $\leq \Pr_r(x\#q\#r\#M'(x\#q\#r) \in D$, where $q \stackrel{\text{def}}{=} \mathcal{A}(x,r))$
    $\leq 1/2^{g(n)}$

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. $L \in \mathbf{BP} \cdot \mathbf{NP}$ iff $\forall$poly g,
$\exists D \in \mathbf{NP}$, poly time TM $\mathcal{M}$ /
— if $x \in L$ then
$\Pr_r[\mathcal{M}(x,r) \in D] \geq 1\text{–}1/2^{g(n)}$
— if $x \notin L$ then
$\Pr_r[\mathcal{M}(x,r) \in D] \leq 1/2^{g(n)}$.

- **Thm (Prop. 3.5). AM = BP·NP.**
- Proof (3/4). Let $L \in \mathbf{BP{\cdot}NP}$, as here:
  Let $D \stackrel{\text{def}}{=} \{q \mid \exists y, q\#y \in D'\}$, with $D' \in \mathbf{P}$,
  $D'' \stackrel{\text{def}}{=} \{x\#q\#r\#y \mid q\#y \in D'\}$: in $\mathbf{P}$.
- Let $\mathcal{A}(x,r) \stackrel{\text{def}}{=} \mathcal{M}(x,r)$, and $M(x\#q\#r) \stackrel{\text{def}}{=}$ some y such that $q\#y \in D'$ if one exists,
- If $x \in L$, $\Pr_r(x\#q\#r\#y \in D''$, where $q \stackrel{\text{def}}{=} \mathcal{A}(x,r)$, $y \stackrel{\text{def}}{=} M(x\#q\#r))$

$$= \Pr_r(q\#y \in D', \text{ where } q \stackrel{\text{def}}{=} \mathcal{A}(x,r),\ y \stackrel{\text{def}}{=} M(x\#q\#r))$$

$$\geq \Pr_r(\exists y,\ q\#y \in D', \text{ where } q \stackrel{\text{def}}{=} \mathcal{A}(x,r))$$

$$= \Pr_r(\mathcal{M}(x,r) \in D) \geq 1\text{– } 1/2^{g(n)}$$

because M is best:
$(\exists y,\ q\#y \in D') \Rightarrow q\#M(x\#q\#r) \in D'$

AM = BP·NP

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

- **Thm (Prop. 3.5). AM = BP·NP.**
- Proof (4/4). Let $L \in \mathbf{BP{\cdot}NP}$, as here:
  Let $D \stackrel{\text{def}}{=} \{q \mid \exists y,\ q\#y \in D'\}$, with $D' \in \mathbf{P}$,
  $D'' \stackrel{\text{def}}{=} \{x\#q\#r\#y \mid q\#y \in D'\}$: in $\mathbf{P}$.
- Let $\mathcal{A}(x,r) \stackrel{\text{def}}{=} \mathcal{M}(x,r)$
- If $x \notin L$, for any M', $\Pr_r(x\#q\#r\#y \in D''$, where $q \stackrel{\text{def}}{=} \mathcal{A}(x,r)$, $y \stackrel{\text{def}}{=} M'(x\#q\#r))$

$$= \Pr_r(q\#y \in D', \text{ where } q \stackrel{\text{def}}{=} \mathcal{A}(x,r),\ y \stackrel{\text{def}}{=} M'(x\#q\#r))$$

$$\leq \Pr_r(\exists y,\ q\#y \in D', \text{ where } q \stackrel{\text{def}}{=} \mathcal{A}(x,r))$$

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

because $\exists y, P(y)$ is implied by $P(M'(x\#q\#r))$

Other equivalent definitions of **AM**
2. Extended quantifiers

Lazy Arthur

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

- **Prop (Lemma 3.8).** For every word $w \in \{\texttt{A}, \texttt{M}\}^*$, the class $\mathbf{w}_{\text{lazy}}$ 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. □

- $\text{prot}_{w\text{lazy}}(M; x, r_1 r_2 \ldots r_k)$:
  $inp := x$
  `for` j=1…k:
  `if` a_j=`A` `then` ($\cancel{q_j := \mathcal{A}(inp,r_j)}$); $inp := inp\#r_j\cancel{\#q_j}$)
  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):

$$\mathsf{E} r \in R, F(r) \quad \stackrel{\text{def}}{=} \quad \sum_{r \in R} F(r) \ / \ \text{card } R$$

- **Merlin** (maximize):

$$\exists y \in Y, F(y) \quad \stackrel{\text{def}}{=} \quad \max_{y \in Y} F(y)$$

(Note: if F takes its values in $\{0,1\}$, this is really the existential quantifier…)

A small catch

- **Arthur** (expectation):
  $$\mathrm{E} r \in R, F(r) \quad \stackrel{\text{def}}{=} \quad \sum_{r \in R} F(r) \;/\; \text{card } R$$
- **Merlin** (maximize):
  $$\exists y \in Y, F(y) \quad \stackrel{\text{def}}{=} \quad \max_{y \in Y} F(y)$$

- 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** $y \in Y$, $F(y) \leq a$.

« Skolemization »

- **Arthur** (expectation):
  $\mathsf{E}r \in R,\ F(r) \quad \stackrel{\text{def}}{=} \quad \sum_{r \in R} F(r) \ /\ \text{card } R$
- **Merlin** (maximize):
  $\exists y \in Y,\ F(y) \quad \stackrel{\text{def}}{=} \quad \max_{y \in Y} F(y)$

- **Prop.** $\mathsf{E}r \in R,\ \exists y \in Y,\ F(r,y)$
  $= \exists f : R \to Y,\ \mathsf{E}r \in R,\ F(r,f(r))$

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

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

- $$\ldots \leq \max_{f : R \to Y} \mathsf{E}r \in R,\ F(r,f(r))$$

« Skolemization »

- **Arthur** (expectation):

  $\mathsf{E} r \in R,\ F(r) \quad \overset{\text{def}}{=} \quad \sum_{r \in R} F(r)\ /\ \mathrm{card}\ R$

- **Merlin** (maximize):

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

- **Prop.** $\mathsf{E} r \in R,\ \exists y \in Y,\ F(r,y)$

  $= \exists f : R \to Y,\ \mathsf{E} r \in R,\ F(r,f(r))$

- Proof (2/2).

  For every f, $F(r,f(r)) \leq \max_{y \in Y} F(r,y)$

- Take expectations:

  $$\mathsf{E} r \in R,\ F(r,f(r)) \leq \mathsf{E} r \in R,\ \max_{y \in Y} F(r,y)$$

- Now take max over f. $\square$

« Skolemization »: an example

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

- **Arthur** (expectation):
  $\mathrm{E}r \in R, F(r) \quad \stackrel{\text{def}}{=} \quad \sum_{r \in R} F(r)\ /\ \mathrm{card}\ R$
- **Merlin** (maximize):
  $\exists y \in Y, F(y) \quad \stackrel{\text{def}}{=} \quad \max_{y \in Y} F(y)$

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

Expectations and probabilities

- Arthur (expectation):
  $$\mathrm{E}r \in R,\ F(r) \quad \stackrel{\text{def}}{=} \quad \textstyle\sum_{r \in R} F(r) \ /\ \mathrm{card}\ R$$
- Merlin (maximize):
  $$\exists y \in Y,\ F(y) \quad \stackrel{\text{def}}{=} \quad \max_{y \in Y} F(y)$$

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

- 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)
- Also, $\mathrm{E}r, F(r) \quad = \quad \Pr_r(F(r)=1)$

(« expectation of a predicate = its probability of occurring »)

A-M as E-∃ formulae

- **Arthur** (expectation):
  $\mathsf{E}r \in R,\ F(r) \ \stackrel{\text{def}}{=}\ \sum_{r \in R} F(r)\ /\ \text{card } R$
- **Merlin** (maximize):
  $\exists y \in Y,\ F(y) \ \stackrel{\text{def}}{=}\ \max_{y \in Y} F(y)$

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

- **Prop (3.10).** $L \in$ **AMAM** iff
  for every polynomial $g(n)$,
  there is a poly time predicate P /
  — if $x \in L$, then $G(x) \geq 1 - 1/2^{g(n)}$
  — if $x \notin L$ then $G(x) \leq 1/2^{g(n)}$
  where $G(x) \stackrel{\text{def}}{=} \mathsf{E}r_1,\ \exists y_1,\ \mathsf{E}r_2,\ \exists y_2,\ P(x,r_1,y_1,r_2,y_2)$
  A M A M
- Proof (1/5). $G(x) = \exists f_1, f_2,\ \mathsf{E}r_1, r_2,\ P(x,r_1,y_1,r_2,y_2)$
  where $y_1 \stackrel{\text{def}}{=} f_1(r_1)$, $y_2 \stackrel{\text{def}}{=} f_2(r_1,r_2)$ « skolemization »
- Hence $G(x) = \exists f_1, f_2,\ \Pr_{r_1, r_2}(x\#r_1\#y_1\#r_2\#y_2 \in D)$
  where $D \stackrel{\text{def}}{=} \{x\#r_1\#y_1\#r_2\#y_2 \mid P(x,r_1,y_1,r_2,y_2)=1\}$ (note: $D \in \mathbf{P}$)

(I will let you generalize to other classes of the A-M hierarchy)

A-M as E-∃ formulae

Prop (3.10). $L \in$ **AMAM** iff
for every polynomial $g(n)$,
there is a poly time predicate P /
— if $x \in L$, then $G(x) \geq 1-1/2^{g(n)}$
— if $x \notin L$ then $G(x) \leq 1/2^{g(n)}$
where $G(x) \stackrel{\text{def}}{=} \exists f_1, f_2, \Pr_{r1,\, r2}(x\#r_1\#y_1\#r_2\#y_2 \in D)$
and $y_1 \stackrel{\text{def}}{=} f_1(r_1)$, $y_2 \stackrel{\text{def}}{=} f_2(r_1,r_2)$

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

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

- Then $G(x) = \exists f_1, f_2, \Pr_{r1,\, r2}(x\#r_1\#y_1\#r_2\#y_2 \in D)$
  $\geq \Pr_{r1,r2}(x\#r_1\#y_1\#r_2\#y_2 \in D)$
  $\geq 1-1/2^{g(n)}$

A-M as E-∃ formulae

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

Prop (3.10). $L \in$ **AMAM** iff
for every polynomial $g(n)$,
there is a poly time predicate P /
— if $x \in L$, then $G(x) \geq 1\text{–}1/2^{g(n)}$
— if $x \notin L$ then $G(x) \leq 1/2^{g(n)}$
where $G(x) \stackrel{\text{def}}{=} \exists f_1, f_2, \Pr_{r1,\, r2}(x\#r_1\#y_1\#r_2\#y_2 \in D)$
and $y_1 \stackrel{\text{def}}{=} f_1(r_1)$, $y_2 \stackrel{\text{def}}{=} f_2(r_1,r_2)$

- if $x \notin L$ then for all maps f_1, f_2,
  let $M'(x\#r_1) \stackrel{\text{def}}{=} f_1(r_1)$, $M'(x\#r_1\#y_1\#r_2) \stackrel{\text{def}}{=} f_2(r_1,r_2)$
  and M' of anything else be arbitrary (e.g., ε)

- Then $G(x) \leq \Pr_r(x\#r_1\#y_1\#r_2\#y_2 \in D) \leq 1/2^{g(n)}$
  where $y_1 \stackrel{\text{def}}{=} M'(x\#r_1)$, $y_2 \stackrel{\text{def}}{=} M'(x\#r_1\#y_1\#r_2)$

A-M as E-∃ formulae

Prop (3.10). $L \in$ **AMAM** iff
for every polynomial $g(n)$,
there is a poly time predicate P /
— if $x \in L$, then $G(x) \geq 1\text{–}1/2^{g(n)}$
— if $x \notin L$ then $G(x) \leq 1/2^{g(n)}$
where $G(x) \stackrel{\text{def}}{=} \exists f_1, f_2, \Pr_{r1,\, r2}(x\#r_1\#y_1\#r_2\#y_2 \in D)$
and $y_1 \stackrel{\text{def}}{=} f_1(r_1),\ y_2 \stackrel{\text{def}}{=} f_2(r_1,r_2)$

- Proof (4/5). If L is as here →
- for each $x \in L$ there are maps f_1, f_2 such that
$$\Pr_{r1,\, r2}(x\#r_1\#y_1\#r_2\#y_2 \in D) \geq 1\text{–}1/2^{g(n)}$$
$$\text{where } y_1 \stackrel{\text{def}}{=} f_1(r_1),\ y_2 \stackrel{\text{def}}{=} f_2(r_1,r_2)$$
- Let $M(x\#r_1) \stackrel{\text{def}}{=} f_1(r_1)$, $M(x\#r_1\#y_1\#r_2) \stackrel{\text{def}}{=} f_1(r_1,r_2)$, else arbitrary
- If $x \in L$ then $\Pr_r(x\#r_1\#y_1\#r_2\#y_2 \in D)\ \ (y_1 \stackrel{\text{def}}{=} M(x\#r_1),\ y_2 \stackrel{\text{def}}{=} M(x\#r_1\#y_1\#r_2))$
$$= \Pr_{r1,\, r2}(x\#r_1\#y_1\#r_2\#y_2 \in D)\ (y_1 \stackrel{\text{def}}{=} f_1(r_1),\ y_2 \stackrel{\text{def}}{=} f_2(r_1,r_2))$$
$$\geq 1\text{–}1/2^{g(n)}$$

A-M as E-∃ formulae

Prop (3.10). $L \in$ **AMAM** iff
for every polynomial $g(n)$,
there is a poly time predicate P /
— if $x \in L$, then $G(x) \geq 1\text{–}1/2^{g(n)}$
— if $x \notin L$ then $G(x) \leq 1/2^{g(n)}$
where $G(x) \stackrel{\text{def}}{=} \exists f_1, f_2, \Pr_{r1,\, r2}(x\#r_1\#y_1\#r_2\#y_2 \in D)$
and $y_1 \stackrel{\text{def}}{=} f_1(r_1)$, $y_2 \stackrel{\text{def}}{=} f_2(r_1,r_2)$

- Proof (5/5). If L is as here $\rightarrow$
- If $x \notin L$ then
  for every Merlin map M',

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

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

- Then $\Pr_r(x\#r_1\#y_1\#r_2\#y_2 \in D)$ $(y_1 \stackrel{\text{def}}{=} M'(x\#r_1),\, y_2 \stackrel{\text{def}}{=} M'(x\#r_1\#y_1\#r_2))$
  $= \Pr_{r1,\, r2}(x\#r_1\#y_1\#r_2\#y_2 \in D)$ $(y_1 \stackrel{\text{def}}{=} f_1(r_1),\, y_2 \stackrel{\text{def}}{=} f_2(r_1,r_2))$
  $\leq G(x) \leq 1/2^{g(n)}$. $\square$

Next time…

The Arthur-Merlin hierarchy collapses!

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

AM (All other classes **w** equal to **AM**)

$\cup\!|$

MA

$\subseteq \quad \supseteq$

NP **BPP**

$\supseteq \quad \subseteq$

P