Jean Goubault-Larrecq

Randomized complexity classes

Today: **Sipser's coding lemmas**, and consequences

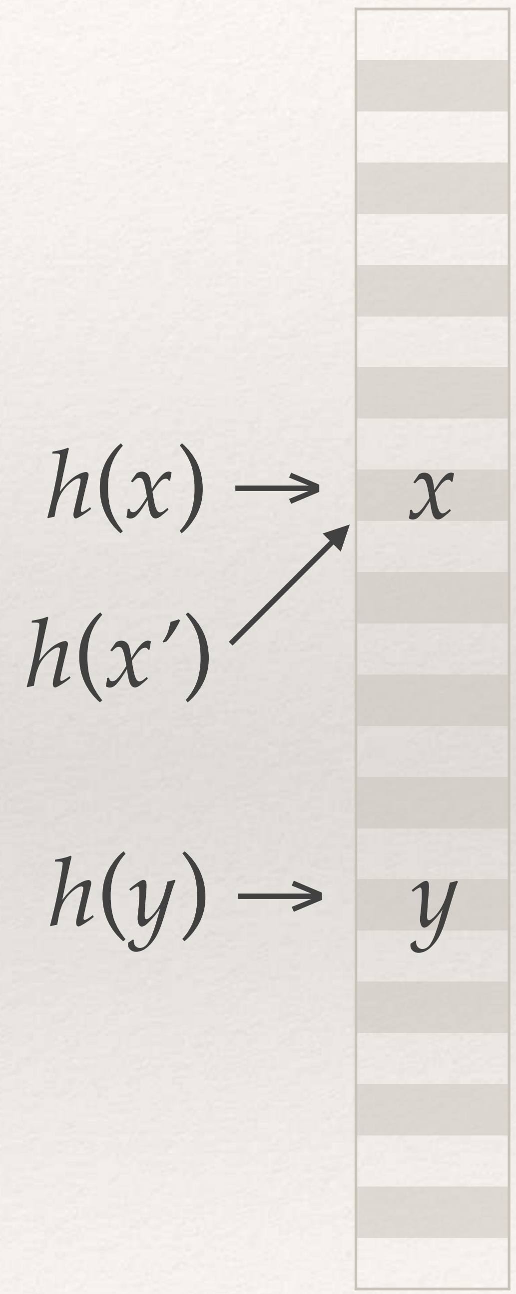
Today

- Sipser's coding lemmas
- **AM** is in the polynomial hierarchy
- The Goldwasser-Sipser theorem:
  public coins≡private coins
- The Boppana-Håstad-Zachos theorem:
  Graph Isomorphism is most certainly not **NP**-complete.

Sipser’s coding lemmas

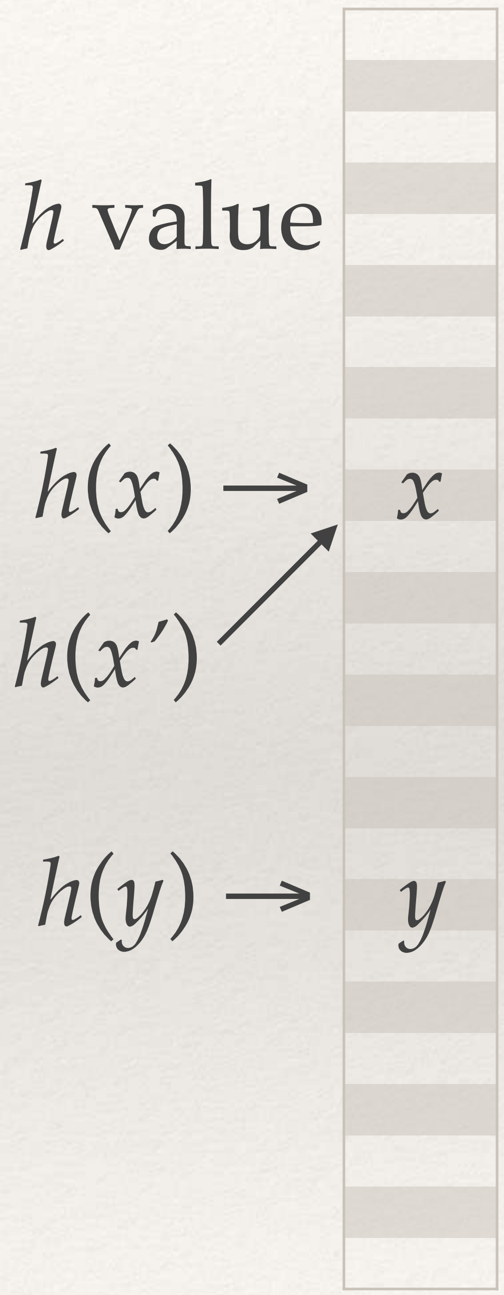
Hash tables

- Store elements of type σ (e.g. strings)
  In general, associative array $\sigma \rightarrow \tau$
- **Hash-table** = table of size N
  Hash function $h : \sigma \rightarrow [0, N\text{–}1]$
  Each datum x stored at position $h(x)$
- **Collision:** element x of σ
  such that there is an element $x' \neq x$ of σ
  with $h(x)=h(x')$



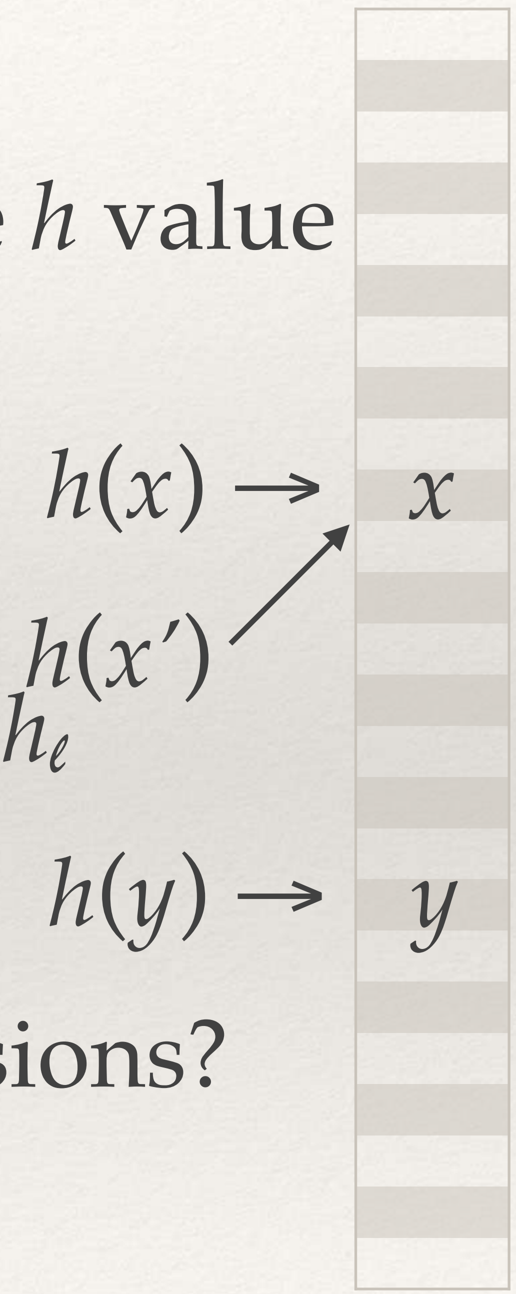
Collisions

- In practice, one avoids collisions by:
  — storing **lists** of data x with the same h value instead of just elements
  — **resizing** the table (increasing N) in case of collisions
- But how can we ensure that N is large enough so that there are **no** collisions? How do we choose h?



Collisions

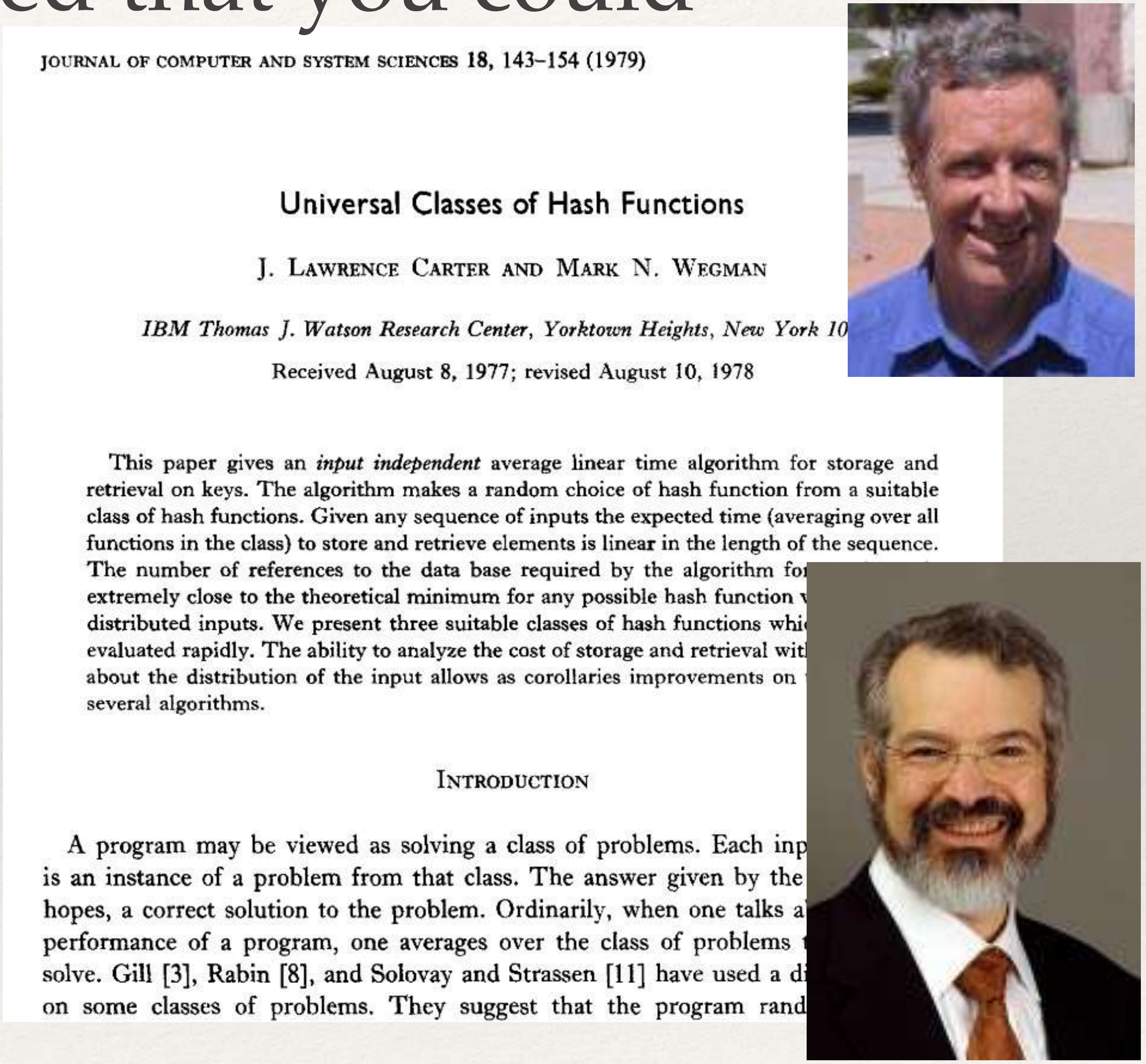
- In practice, one avoids collisions by:
  — storing **lists** of data x with the same h value instead of just elements
  — **resizing** the table (increasing N) in case of collisions
  — using **several** hash functions $h_1, \ldots, h_\ell$



- Still, how can we ensure that N is large enough so that there are **no** collisions?
  How do we choose $H \stackrel{\text{def}}{=} (h_1, \ldots, h_\ell)$?

Universal hash functions

- Carter and Wegman realized that you could **draw** $H \stackrel{\text{def}}{=} (h_1, \ldots, h_\ell)$ **at random** from certain so-called universal classes…
- and there are very simple such classes!



JOURNAL OF COMPUTER AND SYSTEM SCIENCES 18, 143–154 (1979)

Universal Classes of Hash Functions

J. LAWRENCE CARTER AND MARK N. WEGMAN

IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10

Received August 8, 1977; revised August 10, 1978

This paper gives an *input independent* average linear time algorithm for storage and retrieval on keys. The algorithm makes a random choice of hash function from a suitable class of hash functions. Given any sequence of inputs the expected time (averaging over all functions in the class) to store and retrieve elements is linear in the length of the sequence. The number of references to the data base required by the algorithm fo extremely close to the theoretical minimum for any possible hash function v distributed inputs. We present three suitable classes of hash functions whi evaluated rapidly. The ability to analyze the cost of storage and retrieval wit about the distribution of the input allows as corollaries improvements on several algorithms.

INTRODUCTION

A program may be viewed as solving a class of problems. Each inp is an instance of a problem from that class. The answer given by the hopes, a correct solution to the problem. Ordinarily, when one talks a performance of a program, one averages over the class of problems solve. Gill [3], Rabin [8], and Solovay and Strassen [11] have used a d on some classes of problems. They suggest that the program rand

https://cse.ucsd.edu/sites/cse.ucsd.edu/files/faculty/carter17-115x150.jpg
https://www.ithistory.org/sites/default/files/honor-roll/Mark%20N.%20Wegman.jpg

Linear hash functions

- Let $\Sigma=\{0,1\} = \mathbb{Z}/2\mathbb{Z}$
- A **linear** hash function $h : \Sigma^m \to \Sigma^{m'}$
  is just a linear map between vector spaces (over $\mathbb{Z}/2\mathbb{Z}$)
- … i.e., h is given by a **matrix of bits** $B = (b_{ij})_{i=1..m',\, j=1..m}$:
  $$h(x_1,\ldots,x_m) = (b_{i1}x_1 + \ldots + b_{im}x_m)_{i=1..m'}$$
- For computer geeks, each row $(b_{i1},\ldots,b_{im})$ is a **mask**
  and $b_{i1}x_1 + \ldots + b_{im}x_m$ is a **parity check**
  = exclusive or of the bits x_j at those positions j / $b_{ij}=1$

Linear hash functions

- A **linear** hash function $h : \Sigma^m \to \Sigma^{m'}$ is just a linear map between vector spaces (over $\mathbb{Z}/2\mathbb{Z}$)
- … i.e., h is given by a **matrix of bits** $B = (b_{ij})_{i=1..m',\, j=1..m}$:
  $h(x_1,\ldots,x_m) = (b_{i1}x_1 + \ldots + b_{im}x_m)_{i=1..m'}$

- It is easy to draw h at random uniformly: just draw mm' bits independently, uniformly, at random
- Let X be the set of data to be stored. Sipser realized that:
- **(Coding lemma I):** if X is sufficiently small, then drawing $H \stackrel{\text{def}}{=} (h_1, \ldots, h_\ell)$ at random, with **high probability** there will be no collision in X
- **(Coding lemma II):** if X is too large, then whichever $H \stackrel{\text{def}}{=} (h_1, \ldots, h_\ell)$ you take, there will **definitely** be a collision in X.

https://upload.wikimedia.org/wikipedia/commons/thumb/3/34/MIT-Science_Sipser_Michael.jpg/440px-MIT-Science_Sipser_Michael.jpg

The definition of collisions

- A **collision** x **for** $H \stackrel{\text{def}}{=} (h_1, \ldots, h_\ell) : \Sigma^m \to \Sigma^{m'}$ **in** X is a point:
  - in X
  - such that there are points $y_1, \ldots, y_\ell$,
    - all in X
    - all distinct from x
    - but $h_1(x) = h_1(y_1), \ldots, h_\ell(x) = h_\ell(y_\ell)$.
- If such an x exists, we say that X **has a collision for** H.

Sipser's coding lemma I (X small)

- **Lemma (3.15).** Let $X \subseteq \Sigma^m$. Assume $|X| \leq 2^{m'-1}$. Let $\ell \geq m'$.
  Then $\Pr_H(X \text{ has a collision for } H) \leq 1/2^{\ell-m'+1}$.

- Proof (1/5). We start by proving:
  Claim. For every **non-zero** $y \in \Sigma^m$,
  $\Pr_z(z \,.\, y=0)=\Pr_z(z \,.\, y=1)=1/2$.

A **collision** x **for** $H \stackrel{\text{def}}{=} (h_1, \ldots, h_\ell)$ **in** X is a point:
$: \Sigma^m \to \Sigma^{m'}$
— in X
— such that there are points $y_1, \ldots, y_\ell$,
  — all in X
  — all distinct from x
  — but $h_1(x)=h_1(y_1), \ldots, h_\ell(x)=h_\ell(y_\ell)$.

- Indeed, y has a non-zero coordinate y_i (hence $y_i=1$)
  Let $t \stackrel{\text{def}}{=} (0, \ldots, 0, 1, 0, \ldots, 0)$ with the only 1 at position i.
  Then $z \mapsto z \oplus t$ (flip bit i) is a **bijection**
  of $\{z \in \Sigma^m \mid z \,.\, y=0\}$ onto $\{z \in \Sigma^m \mid z \,.\, y=1\}$.

Sipser's coding lemma I (X small)

- **Lemma (3.15).** Let $X \subseteq \Sigma^m$. Assume $|X| \leq 2^{m'-1}$. Let $\ell \geq m'$.
  Then $\Pr_H(X$ has a collision for $H) \leq 1/2^{\ell-m'+1}$.

- Proof (2/5). Recap:
  Claim. For every **non-zero** $y \in \Sigma^m$,
  $\Pr_z(z \,.\, y=0)=\Pr_z(z \,.\, y=1)=1/2$.

A **collision** x **for** $H \stackrel{\text{def}}{=} (h_1, \ldots, h_\ell) : \Sigma^m \to \Sigma^{m'}$ **in** X is a point:
— in X
— such that there are points $y_1, \ldots, y_\ell$,
  — all in X
  — all distinct from x
  — but $h_1(x)=h_1(y_1), \ldots, h_\ell(x)=h_\ell(y_\ell)$.

- In particular, if $x \neq y_j$, then $\Pr_z(z \,.\, x=z \,.\, y_j) = 1/2$
  (take $y \stackrel{\text{def}}{=} x-y_j$).

Sipser's coding lemma I (X small)

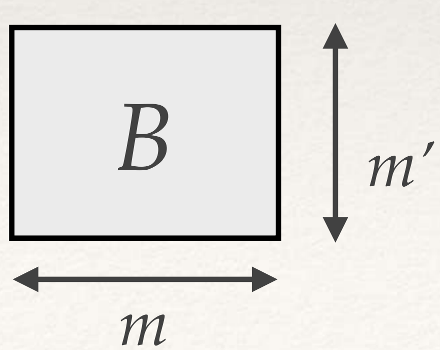
- **Lemma (3.15).** Let $X \subseteq \Sigma^m$. Assume $|X| \leq 2^{m'-1}$. Let $\ell \geq m'$.
  Then $\Pr_H(X$ has a collision for $H) \leq 1/2^{\ell-m'+1}$.

A **collision** x **for** $H \stackrel{\text{def}}{=} (h_1, \ldots, h_\ell) : \Sigma^m \to \Sigma^{m'}$ **in** X is a point:
— in X
— such that there are points $y_1, \ldots, y_\ell$,
— all in X
— all distinct from x
— but $h_1(x)=h_1(y_1), \ldots, h_\ell(x)=h_\ell(y_\ell)$.

- Proof (3/5). Recap:
  If $x \neq y_j$, then $\Pr_z(z \,.\, x = z \,.\, y_j) = 1/2$

h is given by a **matrix of bits** $B = (b_{ij})_{i=1..m', j=1..m}$:
$h(x_1,\ldots,x_m) = (b_{i1}x_1 + \ldots + b_{im}x_m)_{i=1..m'}$

- Hence $\Pr_h(h(x)=h(y_j))$
  $= \Pr_{B\ (m' \times m)\ \text{matrix}}(B.x = B.y_j)$
  $= \Pr_{B\ (m' \times m)\ \text{matrix}}(\forall \text{row } z \text{ of } B,\ z \,.\, x = z \,.\, y_j) = 1/2^{m'}$



Sipser's coding lemma I (X small)

- **Lemma (3.15).** Let $X \subseteq \Sigma^m$. Assume $|X| \leq 2^{m'-1}$. Let $\ell \geq m'$.
  Then $\Pr_H(X$ has a collision for $H) \leq 1/2^{\ell-m'+1}$.

A **collision** x **for** $H \stackrel{\text{def}}{=} (h_1, \ldots, h_\ell)$ **in** X is a point:
$: \Sigma^m \to \Sigma^{m'}$
— in X
— such that there are points $y_1, \ldots, y_\ell$,
  — all in X
  — all distinct from x
  — but $h_1(x)=h_1(y_1), \ldots, h_\ell(x)=h_\ell(y_\ell)$.

- Proof (4/5).
  $\Pr_h(h(x)=h(y_j)) = 1/2^{m'}$ (recap).
- $\Pr_H(x$ is a collision for H in $X)$
  $= \Pr_H(\exists y_1, \ldots, y_\ell \in X-\{x\}, \wedge_{j=1}^{\ell} h_j(x)=h_j(y_j))$
- $\leq \Sigma_{y1, \ldots, y_\ell \in X-\{x\}} \Pi_{j=1}^{\ell} \Pr_h(h(x)=h(y_j))$ (sum bound+independence)
- $\leq (|X|-1)^\ell / 2^{\ell m'} < 1/2^\ell$

Sipser's coding lemma I (X small)

- **Lemma (3.15).** Let $X \subseteq \Sigma^m$. Assume $|X| \leq 2^{m'-1}$. Let $\ell \geq m'$.
  Then $\Pr_H(X$ has a collision for $H) \leq 1/2^{\ell-m'+1}$.

- Proof (5/5). Recap:
  $\Pr_H(x$ is a collision for H in $X) < 1/2^\ell$

A **collision** x **for** $H \stackrel{\text{def}}{=} (h_1, \ldots, h_\ell)$ **in** X is a point:
$: \Sigma^m \to \Sigma^{m'}$
— in X
— such that there are points $y_1, \ldots, y_\ell$,
  — all in X
  — all distinct from x
  — but $h_1(x) = h_1(y_1), \ldots, h_\ell(x) = h_\ell(y_\ell)$.

- $\Pr_H(X$ has a collision for $H)$
  $= \Pr_H(\exists x \in X, x$ is a collision for H in $X)$

- $\leq |X|/2^\ell \leq 1/2^{\ell-m'+1}$. $\square$

Sipser's coding lemma II (X large)

- **Lemma (3.16).** Let $X \subseteq \Sigma^m$. Assume $|X| > \ell.2^{m'}$.
  Then X (definitely) has a collision for H.
- Proof. A collision x for H in X is a point in X /
  $\forall j\ (1 \le j \le \ell),\ \exists y\ (=y_j) \in X-\{x\},\ h_j(x)=h_j(y)$
- Hence, if X has no collision for H, then
  for each $x \in X$,
  there is a $j\ (1 \le j \le \ell)$ / $\forall y \in X-\{x\},\ h_j(x) \neq h_j(y)$
- For each $x \in X$, let $\kappa(x) \stackrel{\text{def}}{=}$ the least such j.
- Then $x \in X \mapsto (j, h_j(x))$ where $j=\kappa(x)$ is **injective**
  … since otherwise $\exists y \in X-\{x\},\ j=\kappa(x)(=\kappa(y))$ and $h_j(x)=h_j(y)$
- Hence card $X \le \ell$.card $\Sigma^{m'} = \ell.2^{m'}$. □

A **collision** x **for** $H \stackrel{\text{def}}{=} (h_1, \ldots, h_\ell)$ **in** X is a point:
$: \Sigma^m \to \Sigma^{m'}$
— in X
— such that there are points $y_1, \ldots, y_\ell$,
— all in X
— all distinct from x
— but $h_1(x)=h_1(y_1), \ldots, h_\ell(x)=h_\ell(y_\ell)$.

Large or small?

- Assume you are given a set R, which is either large or small.
  Let the **gap** be size(large)/size(small).
  We have two techniques to decide whether R is large or small.
- **Lautemann:** (as used in Babai's theorem, Lemma 3.11)
  — R large $\Rightarrow \forall r_1, \ldots, r_k, \exists r', r' \oplus r_i \in R$
  — R small $\Rightarrow \Pr_{r1, \ldots, rk}(\exists r', r' \oplus r_i \in R)$ small
- **Sipser:**
  — R large $\Rightarrow \forall H, \exists$collision for H in R
  — R small $\Rightarrow \Pr_H(\exists$collision for H in $R)$ small

No error if R large

Requires gap $(1\text{–}1/2^n)/(1/2^n) \sim 2^n$

No error if R large

Only needs gap $\ell.2^{m'}/2^{m'-1} = 2\ell = \text{poly}(n)$

Lautemann or Sipser?

- Sipser will have a real advantage over Lautemann only later, when we show that **GNI** is in **AM**, not just in **IP**[1]
- … and in principle when we show the Goldwasser-Sipser theorem (later)
- For now, we will use Sipser to show that **AM** $\subseteq \Pi^p_2$ and Lautemann would be just as practical here
- We start by showing that for every language $L \in$ **AM**, we can require **perfect soundness** (=no error if $x \in L$).

$$\mathbf{AM} \subseteq \Pi^p_2$$

AM with perfect soundness (1/4)

- Let L be in **AM**. For some $D \in \mathbf{P}$,
  — if $x \in L$ then $(\mathrm{E}r, \exists y, x\#r\#y \in D) \geq 1\text{–}1/2^n$ (« large »)
  — if $x \notin L$ then $(\mathrm{E}r, \exists y, x\#r\#y \in D) \leq 1/2^n$ (« small »)
- Let $R \stackrel{\text{def}}{=} \{r \in \Sigma^m \mid \exists y, x\#r\#y \in D\}$ (either large or small)
- Arthur draws $H \stackrel{\text{def}}{=} (h_1, \ldots, h_\ell)$ at random uniformly ($mm'\ell$ bits)
- Merlin answers a (claimed) collision r in R
- We check that this is a collision.
  — if $x \in L$ then $\forall H, \exists$collision for H in R (**perfect soundness**!)
  — if $x \notin L$ then $\Pr_H(\exists\text{collision for } H \text{ in } R) \leq 1/2^{\ell-m'+1}$

$m=q(n)$ (poly, given), but we should determine m', ℓ

Can we really do this in polynomial time?

AM with perfect soundness (2/4)

- Let L be in **AM**. For some $D \in \mathbf{P}$,
  — if $x \in L$ then $(\mathrm{E}r, \exists y, x\#r\#y \in D) \geq 1-1/2^n$ (« large »)
  — if $x \notin L$ then $(\mathrm{E}r, \exists y, x\#r\#y \in D) \leq 1/2^n$ (« small »)
- Let $R \stackrel{\text{def}}{=} \{r \in \Sigma^m \mid \exists y, x\#r\#y \in D\}$ (either large or small)
- Arthur draws $H \stackrel{\text{def}}{=} (h_1, \ldots, h_\ell)$ at random uniformly ($mm'\ell$ bits
- Merlin answers a (claimed) collision r in R
- We check that this is a collision.
  — if $x \in L$ then $\forall H$, $\exists$collision for H in R (**perfect soundne**
  — if $x \notin L$ then $\Pr_H(\exists\text{collision for } H \text{ in } R) \leq 1/2^{\ell-m'+1}$

- $m=q(n)$ (assume $m \geq n$): determine m', ℓ
- Use Sipser I and II:

  Lemma (3.15). Let $X \subseteq \Sigma^m$. Assume $|X| \leq 2^{m'-1}$. Let $\ell \geq m'$.
  Then $\Pr_H(X \text{ has a collision for } H) \leq 1/2^{\ell-m'+1}$.

  Lemma (3.16). Let $X \subseteq \Sigma^m$. Assume $|X| > \ell.2^{m'}$, where $\ell \geq m'$.
  Then X (definitely) has a collision for H.

- To apply Sipser I, need $|R| \leq 2^{m'-1}$ if $x \notin L$ $\Rightarrow$ e.g., require $m'-1 \geq m-n$ (1)
- To apply Sipser II, need $|R| > \ell.2^{m'}$ if $x \in L$ $\Rightarrow$ e.g., require $m'+\log_2 \ell < m-1$ (2)
- We wish error to be $\leq 1/2^{g(n)}$ $\Rightarrow$ require $\ell-m'+1 \geq g(n)$ (3)
- Other constraints: $\ell \geq m'$ (4), both polynomial in n.

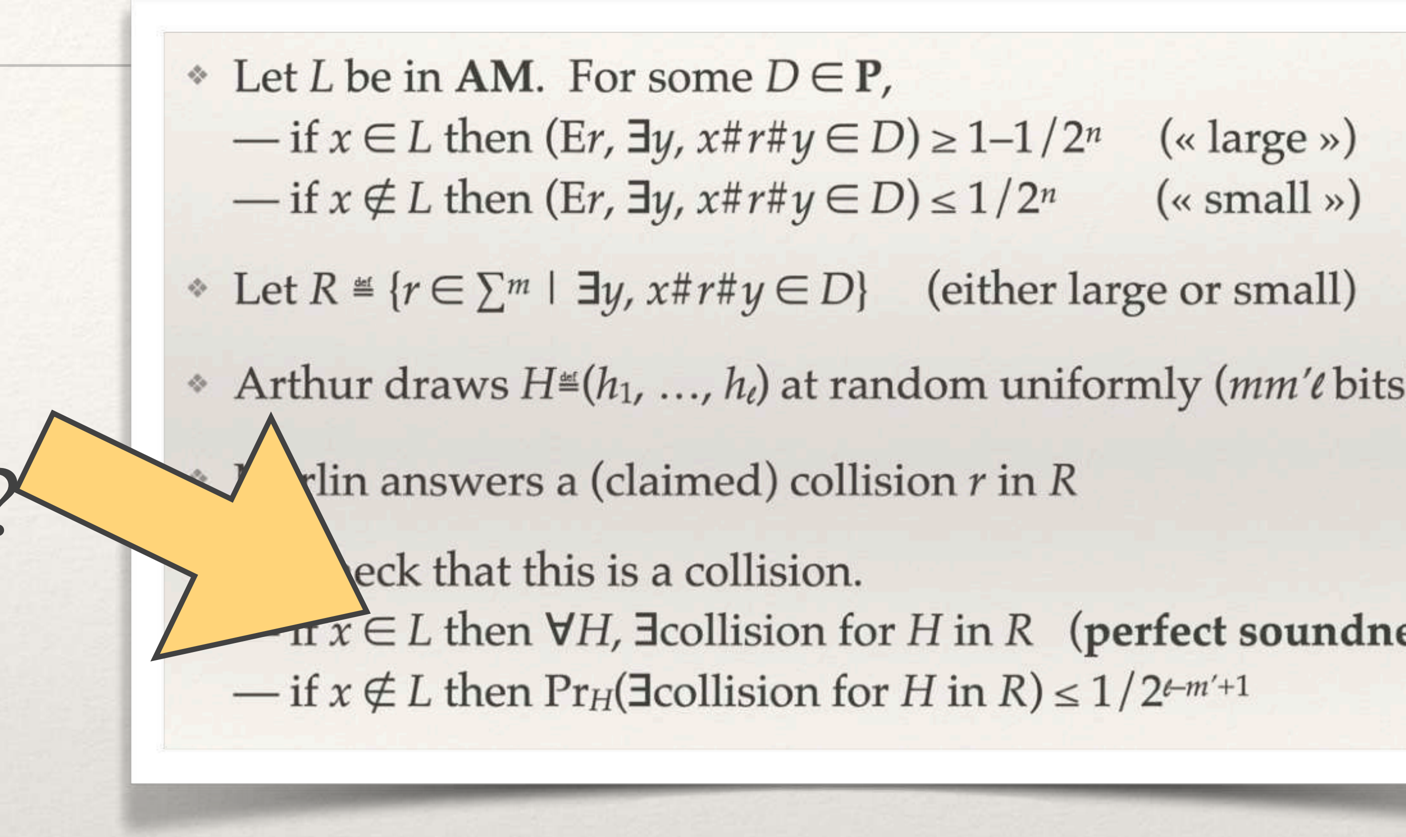
E.g., $m' \stackrel{\text{def}}{=} m-n+1$ (for (1)), $\ell \stackrel{\text{def}}{=} m'+g(n)$ (for (3), (4))
((2) OK for n large enough, otherwise tabulate)

AM with perfect soundness (3/4)

- Now m, m', ℓ=poly(n)
  Can we check that r is a collision in R in poly time?

- No: just checking that r is in R is an **NP** problem…

- Instead…



- Let L be in **AM**. For some $D \in$ **P**,
  — if $x \in L$ then (Er, $\exists y$, $x\#r\#y \in D$) $\geq 1-1/2^n$ (« large »)
  — if $x \notin L$ then (Er, $\exists y$, $x\#r\#y \in D$) $\leq 1/2^n$ (« small »)
- Let $R \triangleq \{r \in \Sigma^m \mid \exists y, x\#r\#y \in D\}$ (either large or small)
- Arthur draws $H \triangleq (h_1, \ldots, h_\ell)$ at random uniformly ($mm'\ell$ bits
- rlin answers a (claimed) collision r in R
- eck that this is a collision.
  if $x \in L$ then $\forall H$, $\exists$collision for H in R (**perfect soundne**
  — if $x \notin L$ then $\Pr_H$($\exists$collision for H in R) $\leq 1/2^{\ell-m'+1}$

AM with perfect soundness (4/4)

- Let L be in **AM**. For some $D \in \mathbf{P}$,
  — if $x \in L$ then $(\mathrm{E}r, \exists y, x\#r\#y \in D) \geq 1-1/2^n$ (« large »)
  — if $x \notin L$ then $(\mathrm{E}r, \exists y, x\#r\#y \in D) \leq 1/2^n$ (« small »)
- Let $R \stackrel{\text{def}}{=} \{r \in \Sigma^m \mid \exists y, x\#r\#y \in D\}$ (either large or small)
- Arthur draws $H \stackrel{\text{def}}{=} (h_1, \ldots, h_\ell)$ at random uniformly ($mm'\ell$ bits
- ~~Merlin answers a (claimed) collision r in R~~
- We check that this is a collision.
  — if $x \in L$ then $\forall H$, $\exists$collision for H in R (**perfect soundne**
  — if $x \notin L$ then $\Pr_H(\exists\text{collision for } H \text{ in } R) \leq 1/2^{\ell-m'+1}$

- We require Merlin to give us:
  - a claimed collision r
  - a **proof** y that $r \in R$
    (i.e., Merlin claims that $x\#r\#y \in D$)
  - points $r_1, \ldots, r_\ell$
  - **proofs** y_j that each r_j is in R
- And we check that $x\#r\#y \in D$,
  $x\#r_j\#y_j \in D$ for each $j=1..\ell$,
  $r \neq r_j$ and $h_j(r)=h_j(r_j)$ for each $j=1..\ell$. □

in poly time!

AM with perfect soundness (4/4)

- We have proved:

Prop (3.18). Every $L \in \mathbf{AM}$ can be decided with an **AM** game with perfect soundness (no error if $x \in L$)

- Hence:

Thm (3.19). $\mathbf{AM} \subseteq \Pi\mathrm{P}_2$.

- Proof. $x \in L$ iff $\forall H$, $\exists$collision for H in R (with proofs!)
  and proofs of collisions can be checked in poly time. □

- Let L be in **AM**. For some $D \in \mathbf{P}$,
  — if $x \in L$ then $(\mathrm{E}r, \exists y, x\#r\#y \in D) \geq 1\text{–}1/2^n$ (« large »)
  — if $x \notin L$ then $(\mathrm{E}r, \exists y, x\#r\#y \in D) \leq 1/2^n$ (« small »)
- Let $R \triangleq \{r \in \Sigma^m \mid \exists y, x\#r\#y \in D\}$ (either large or small)
- Arthur draws $H \triangleq (h_1, \ldots, h_\ell)$ at random uniformly ($mm'\ell$ bits
- ~~Merlin answers a (claimed) collision r in R~~
- We check that this is a collision.
  — if $x \in L$ then $\forall H$, $\exists$collision for H in R (**perfect soundne**
  — if $x \notin L$ then $\mathrm{Pr}_H(\exists\text{collision for } H \text{ in } R) \leq 1/2^{\ell - m' + 1}$

Graph Non-Isomorphism is in **AM**

Reminder: Graph Non-Isomorphism

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

> **GI**
> INPUT: 2 graphs $G_1=(V, E_1)$, $G_2=(V, E_2)$ (with the same V)
> $V = \{1, \ldots, N\}$
> QUESTION: are G_1, G_2 isomorphic?

- **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.

Note: it is crucial here that i remains secret!
This is not an **AM** game

GNI is in AM

- **Idea:** (that fails, but we will fix this later)
  Let $X_i \stackrel{\text{def}}{=} \{\text{graphs } G \text{ on } V \text{ such that } G \equiv G_i\}$,
  $X \stackrel{\text{def}}{=} X_1 \cup X_2$
- Imagine $|X_1| \approx |X_2| \approx K$
- If $(G_1, G_2) \in$ **GNI**,
  i.e. if $G_1 \not\equiv G_2$ then $|X| \approx 2K$ (X is large)
- Otherwise $|X| \approx K$ (X is small)
- We test which is the case using Sipser
  (All random bits in H are public $\Rightarrow$ in **AM**.)

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? $V = \{1, \ldots, N\}$

The main problem is this:
$|X_i|$ can vary wildly,
— from $N!/2$ (if G_i is a chain)
— to 1 (if G_i is a complete graph)

Oops, gap is only 2:
not enough for Sipser, but we will see later how to increase it.

Building sets of uniform size

- X_i is the **orbit** of G_i under the group action of $\mathbf{S}_N$ on $\mathbf{G}_N$
- Let $\phi_i \stackrel{\text{def}}{=} \{\pi \in \mathbf{S}_N \mid \pi.G_i = G_i\}$
  stabilizer of G_i
- The **orbit-stabilizer thm:**
  | orbit | . | stabilizer |
  = order of the group $\mathbf{S}_N$
  I.e., $|X_i \times \phi_i| = N!$
- … independently of i.

GI

INPUT: 2 graphs $G_1=(V, E_1)$, $G_2=(V, E_2)$ (with the same V)
$V = \{1, \ldots, N\}$
QUESTION: are G_1, G_2 isomorphic?

- Let $V = \{1, \ldots, N\}$ set of vertices,
  $\mathbf{G}_N \stackrel{\text{def}}{=}$ directed graphs on V,
  $\mathbf{S}_N \stackrel{\text{def}}{=}$ group of permutations of V.
- $\mathbf{S}_N$ acts on $\mathbf{G}_N$ 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** ($G_1 \equiv G_2$) iff $\exists\, \pi \in \mathbf{S}_N$, $\pi.G_1=G_2$.

Building sets of uniform size

- Hence let $X'_i \overset{\text{def}}{=} X_i \times \phi_i = \{(G,\pi) \mid G \equiv G_i \text{ et } \pi.G_i = G_i\}$, $X \overset{\text{def}}{=} X'_1 \cup X'_2$
- If $(G_1, G_2) \in$ **GNI**, i.e. if $G_1 \not\equiv G_2$ then $|X| = 2N!$
- Otherwise $|X| = N!$
- Good… but gap is still only 2.

GI

INPUT: 2 graphs $G_1=(V, E_1)$, $G_2=(V, E_2)$ (with the same V) $V = \{1, \ldots, N\}$
QUESTION: are G_1, G_2 isomorphic?

- X_i is the **orbit** of G_i under the group action of $\mathbf{S}_N$ on $\mathbf{G}_N$
- Let $\phi_i \overset{\text{def}}{=} \{\pi \in \mathbf{S}_N \mid \pi.G_i = G_i\}$ **stabilizer** of G_i
- The **orbit-stabilizer thm:** $|X_i \times \phi_i| = N!$

The power trick (repeating experiments virtually)

- Hence let $X'_i \stackrel{\text{def}}{=} X_i \times \phi_i = \{(G,\pi) \mid G \equiv G_i \text{ et } \pi.G_i = G_i\}$,
  $X \stackrel{\text{def}}{=} (X'_1 \cup X'_2)^k$
- If $(G_1, G_2) \in$ **GNI**, i.e.
  if $G_1 \not\equiv G_2$ then $|X| = (2N!)^k$
- Otherwise $|X| = (N!)^k$
- Gap is now 2^k.
  Now take k so large that this exceeds 2ℓ.

GI

INPUT: 2 graphs $G_1=(V, E_1)$, $G_2=(V, E_2)$ (with the same V) $V = \{1, \ldots, N\}$
QUESTION: are G_1, G_2 isomorphic?

- X_i is the **orbit** of G_i under the group action of $\mathbf{S}_N$ on G_N
- Let $\phi_i \stackrel{\text{def}}{=} \{\pi \in \mathbf{S}_N \mid \pi.\, G_i = G_i\}$ **stabilizer** of G_i
- The **orbit-stabilizer thm:** $|X_i \times \phi_i| = N!$

GNI is in AM

Let $X'_i \stackrel{\text{def}}{=} X_i \times \phi_i = \{(G,\pi) \mid G \equiv G_i \text{ et } \pi.G_i = G_i\}$,
$X \stackrel{\text{def}}{=} (X'_1 \cup X'_2)^k$

- Arthur draws $H \stackrel{\text{def}}{=} (h_1, \ldots, h_\ell)$ at random uniformly ($mm'\ell$ bits)
- Merlin answers a (claimed) collision x in X (with proofs!)
- We check (the proofs) that this is a collision.
  — if $(G_1, G_2) \in$ **GNI** then $\forall H, \exists$collision for H in X
  — if $(G_1, G_2) \notin$ **GNI** then

$$\Pr_H(\exists\text{collision for } H \text{ in } R) \le 1/2^{\ell - m' + 1}$$

We need to tune m, m', ℓ and k so that the error is $\le 1/2^{g(n)}$

Determining m, m', ℓ, and k

Note: size(graph)=N^2 (adjacency matrix)
size(input)=n=$2N^2$

- $m=k \times$ (size(graph)+size(permutation))
  $= O(k(N^2+N \log N)) = O(kN^2) = O(kn)$

Let $X'_i \stackrel{\text{def}}{=} X_i \times \phi_i = \{(G,\pi) \mid G \equiv G_i \text{ et } \pi.G_i = G_i\}$,
$X \stackrel{\text{def}}{=} (X'_1 \cup X'_2)^k$

— if $(G_1, G_2) \in$ **GNI** then $\forall H$, $\exists$collision for H in X
— if $(G_1, G_2) \notin$ **GNI** then
$Pr_H(\exists\text{collision for } H \text{ in } R) \leq 1/2^{\ell-m'+1}$

- Use Sipser I and II:

Lemma (3.15). Let $X \subseteq \Sigma^m$. Assume $|X| \leq 2^{m'-1}$. Let $\ell \geq m'$.
Then $Pr_H(X \text{ has a collision for } H) \leq 1/2^{\ell-m'+1}$.

Lemma (3.16). Let $X \subseteq \Sigma^m$. Assume $|X| > \ell.2^{m'}$, where $\ell \geq m'$.
Then X (definitely) has a collision for H.

- Sipser I: need $|X| \leq 2^{m'-1}$ if $(G_1, G_2) \notin$ **GNI** $\Rightarrow$ require $m'-1 \geq k.\log_2(N!)$ (1)
- Sipser II: need $|X| > \ell.2^{m'}$ if $(G_1, G_2) \in$ **GNI** $\Rightarrow$ require $m'+\log_2 \ell < k(1+\log_2(N!))$ (2)
- We wish error to be $\leq 1/2^{g(n)}$ $\Rightarrow$ require $\ell-m'+1 \geq g(n)$ (3)
- Other constraints: $\ell \geq m'$ (4), all of m, m', ℓ, k polynomial in n.

E.g., $k \stackrel{\text{def}}{=} N$, $m' \stackrel{\text{def}}{=} 1+k.\log_2(N!)$ (for (1)), $\ell \stackrel{\text{def}}{=} m'+g(n)$ (for (3), (4))
((2) OK for n large enough, otherwise tabulate)

Checking (proofs of) collisions

Let $X'_i \stackrel{\text{def}}{=} X_i \times \phi_i = \{(G,\pi) \mid G \equiv G_i \text{ et } \pi.G_i = G_i\}$,
$X \stackrel{\text{def}}{=} (X'_1 \cup X'_2)^k$

- Explicitly, Merlin sends:
  — an element x
  — a **proof** that x is in X
  — elements $x_1, \ldots, x_\ell$
  — proofs that each x_j is in X
- Then we will check the proofs
  \+ $x \neq x_j$ and $h_j(x) = h_j(x_j)$ for each $j=1..\ell$.
- A **proof** that $x \stackrel{\text{def}}{=} ((G'_1,\pi_1),\ldots,(G'_k,\pi_k))$ is in X is:
  — for each $i=1..k$, a permutation π'_i / $\pi'_i.G'_i = G_1$ or G_2
  — Checking it means checking $\pi'_i.G'_i = G_1$ or G_2,
  and also $\pi_i.G'_i = G'_i$, for each i.

GNI is in AM… and more

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

- We have proved:

  Prop (3.17). GNI is in AM (not just **IP**[1]).

- In fact:

  Thm (3.25; Goldwasser-Sipser).
  For every $k \geq 1$, $\mathbf{IP}[k] \subseteq \mathbf{AM}[k+1]$.

- I will omit the proof, see the lecture notes.

- So $\mathbf{AM} \subseteq \mathbf{IP}[1] \subseteq \ldots \subseteq \mathbf{IP}[k] \subseteq \mathbf{AM}[k+1] = \mathbf{AM}$

- **Corl.** For every $k \geq 1$, $\mathbf{IP}[k] = \mathbf{AM}[k] = \mathbf{AM}$. (!)

https://upload.wikimedia.org/wikipedia/commons/thumb/3/34/MIT-Science_Sipser_Michael.jpg/440px-MIT-Science_Sipser_Michael.jpg

DOES co-NP HAVE SHORT INTERACTIVE PROOFS?

Ravı B. BOPPANA * and Johan HASTAD **

Department of Mathematıcs and Laboratory for Computer Scıence, Massachusetts Instıtute of Technology, 545 Technology Sq
Cambrıdge, MA 02139, U S A

Stathıs ZACHOS

Department of Computer and Informatıon Scıence, Brooklyn College of The Cıty Unıversıty of New York, Brooklyn, NY 11
U S A

Communıcated by Davıd Grıes
Receıved 10 July 1986
Revısed 15 September 1986 and 24 December 1986

Babaı (1985) and Goldwasser, Mıcalı and Rackoff (1985) ıntroduced two probabılıstıc extensıons of the complexıty
NP The two complexıty classes, denoted AM[Q] and IP[Q] respectıvely, are defıned usıng randomızed ınteractıve pr
between a prover and a verıfıer Goldwasser and Sıpser (1986) proved that the two classes are equal We prove that ı
complexıty class co-NP ıs contaıned ın IP[k] for some constant k (ı e , ıf every language ın co-NP has a short ıntera
proof), then the polynomıal-tıme hıerarchy collapses to the second level As a corollary, we show that ıf the G
Isomorphısm problem ıs NP-complete, then the polynomıal-tıme hıerarchy collapses

The Boppana-Håstad-Zachos theorem

https://math.mit.edu/images/profile/boppana_ravi.png

https://www.ae-info.org/attach/User/Hastad_Johan/scaled-0x200_haastad_johan_small_ae.jpg

https://alchetron.com/cdn/stathis-zachos-44e8a09d-57b0-4dd6-8d38-c8273e19ee3-resize-750.jp

The Boppana-Håstad-Zachos theorem

- **Thm (3.20).** If **coNP** $\subseteq$ **AM** then **PH** collapses at level 2.
- *Proof.* Let $L \in \Sigma^p_2$. We will show that L is in Π^p_2.
  $L=\{x \mid \exists y, (x,y) \in L'\}$, for some $L' \in$ **coNP**.
- Hence $L' \in$ **AM**. There is a D in **P** such that:
  — if $(x,y) \in L'$ then $(\mathrm{E}r, \exists z, x\#y\#r\#z \in D)$ large
  — if $(x,y) \notin L'$ then $(\mathrm{E}r, \exists z, x\#y\#r\#z \in D)$ small
- — if $x \in L$ then $(\exists y, \mathrm{E}r, \exists z, x\#y\#r\#z \in D)$ large
  — if $x \notin L$ then $(\exists y, \mathrm{E}r, \exists z, x\#y\#r\#z \in D)$ small
- Hence $L \in$ **MAM** = **AM** (Babai) $\subseteq \Pi^p_2$. $\square$

The BHZ theorem, and Graph Isomorphism

- **Corl (3.21).** If **GI** is **NP**-complete
  then **PH** collapses at level 2.
- *Proof.* **AM** is **closed under poly time reductions**.
- Remember that **GNI** is in **AM**, as we have just shown.
- Hence if **GI** is **NP**-complete,
  then **GNI** is **coNP**-complete,
  hence **coNP** $\subseteq$ **AM**.
  Now apply the previous theorem. □

Graph Isomorphism

- **Corl (3.21).** If **GI** is **NP**-complete then **PH** collapses at level 2.
- Remember that **GI** is not known to be in **P**, and not known to be **NP**-complete.
- The BHZ theorem shows that the latter is unlikely.
- Note: Babai gave a super polynomial time algo for GI in 2015 (still does not solve the question, but what a progress!); builds on a lot of things, including BHZ.

Next time…

IP and PSPACE

- **IP** and **AM** with **polynomially many rounds** (the classes **IP** and **ABPP**)
- Shamir's theorem: **ABPP=IP=PSPACE** (!)