The art of reduction
(or, Depth through Breadth)

Avi Wigderson
Institute for Advanced Study
Princeton
How difficult are the problems we want to solve?

Problem A ?  Problem B ?

Reduction: An efficient algorithm for A implies an efficient algorithm for B

If A is easy then B is easy

If B is hard then A is hard
Constraint Satisfaction I

AI, Operating Systems, Compilers, Verification, Scheduling, ...

3-SAT: Find a Boolean assignment to the $x_i$ which satisfies all constraints
$(x_1 \lor x_3 \lor x_7) (x_1 \lor x_2 \lor x_4) (x_5 \lor x_3 \lor x_6) ...$

3-COLORING: Find a vertex coloring using which satisfies all constraints
Classic reductions

Thm[Cook, Levin '71] 3-SAT is NP-complete

Thm[Karp 72]: 3-COLORING is NP-complete

Proof: If 3-COLORING easy then 3-SAT easy

formula $\rightarrow$ graph

$satisfying \leftrightarrow legal$

assignment $\leftrightarrow$ coloring

Claim: In every legal 3-coloring, $z = x \lor y$
NP - completeness

Papadimitriou: “The largest intellectual export of CS to Math & Science.”

“2000 titles containing ‘NP-complete’ in Physics and Chemistry alone”

“We use it to argue computational difficulty. They to argue structural nastiness”

Most NP-completeness reductions are “gadget” reductions.
**Constraint Satisfaction II**

AI, Programming languages, Compilers, Verification, Scheduling, ...

**3-SAT:** Find a Boolean assignment to the $x_i$ which satisfies **most** constraints

$(x_1 \lor x_3 \lor \overline{x_7}) (x_1 \lor x_2 \lor x_4) (x_5 \lor x_3 \lor x_6) \ldots$

Trivial to satisfy 7/8 of all constraints! (simply choose a random assignment)

How much better can we do efficiently?
Hardness of approximation

Thm[Hastad 01] Satisfying $7/8 + \varepsilon$ fraction of all constraints in 3-SAT is NP-hard


Nontrivial approx $\rightarrow$ Interactive proofs $\rightarrow$ exact solution of 3-SAT
3-SAT is $7/8+\varepsilon$ hard

Influences in Boolean functions

Distributed Computing

Decade

3-SAT is .99 hard

2 Decades

NP
3-SAT is hard
How to minimize players’ influence

Public Information Model [BenOr-Linial]:
Joint random coin flipping / voting
Every good player flips a coin, then combine using function $f$.
Which $f$ is best?

Thm [Kahn—Kalai—Linial]: For every Boolean function on $n$ bits, some player has influence $> (\log n)/n$
(uses Functional & Harmonic Analysis)
3-SAT is \( \frac{7}{8} + \epsilon \) hard

3-SAT is .99 hard

IP = Interactive Proofs
2IP = 2-prover Interactive Proofs
PCP = Probabilistically Checkable Proofs

Optimization
Approx Algorithms
Arithmetization
Program Checking

Cryptography
Zero-knowledge

PCP = NP
2IP = NEXP

NP = PSPACE

3-SAT hard
IP = (Probabilistic) Interactive Proofs [Babai, Goldwasser-Micali-Rackoff]

$L \in \text{NP}$

Prover \( x \) Verifier

Accept \((x,w)\) \( \text{iff } x \in L \)

$L \in \text{IP}$

Prover \( q_1 \) $\cdots$ \( q_r \) Verifier (probabilistic) \( a_1 \) $\cdots$ \( a_r \)

Accept \((x,q,a)\) \( \text{whp iff } x \in L \)
ZK IP: Zero-Knowledge Interactive Proofs
[Goldwasser-Micali-Rackoff]

\[ \text{Accept } (x, q, a) \text{ whp iff } x \in L \]

\( L \in \text{ZK IP} \)

Thm [Goldreich-Micali-Wigderson]: If 1-way functions exist, \( \text{NP} \subseteq \text{ZK IP} \)
(Every proof can be made into a ZK proof!)

Thm [Ostrovsky-Wigderson]: \( \text{ZK} \leftrightarrow \text{1WF} \)
2IP: 2-Prover Interactive Proofs

Verif. accept \((x, q, a, p, b)\) whp iff \(x \in L\)

Theorem [BGKW] : \(NP \subseteq ZK \ 2IP\)
What is the power of Randomness and Interaction in Proofs?

Trivial inclusions:
- \( IP \subseteq \text{PSPACE} \)
- \( 2IP \subseteq \text{NEXP} \)

Few nontrivial examples:
- Graph non isomorphism

......

Few years of stalemate
Meanwhile, in a different galaxy...

Self testing & correcting programs
[Blum-Kannan, Blum-Luby-Rubinfeld]

Given a program $P$ for a function $g : \mathbb{F}^n \rightarrow \mathbb{F}$ and input $x$, compute $g(x)$ correctly when $P$ errs on $\leq$ (unknown) $1/8$ of $\mathbb{F}^n$.

**Easy case** $g$ is linear: $g(w+z)=g(w)+g(z)$

Given $x$, Pick $y$ at random, and compute

$$\forall x \text{ Prob}_y [P(x+y)-P(y) \neq g(x)] \leq 1/4$$

Other functions?
- The Permanent [BF,L]
- Low degree polynomials
Determinant & Permanent

$X = (x_{ij})$ is an $n \times n$ matrix over $F$

Two polynomials of degree $n$:

**Determinant**: $d(X) = \sum_{\sigma \in S_n} \text{sgn}(\sigma) \prod_{i \in [n]} X_{i\sigma(i)}$

**Permanent**: $g(X) = \sum_{\sigma \in S_n} \prod_{i \in [n]} X_{i\sigma(i)}$

**Thm[Gauss]** Determinant is easy

**Thm[Valiant]** Permanent is hard
The Permanent is self-correctable
[Beaver-Feigenbaum, Lipton]

Given $X$, pick $Y$ at random, and compute
$P(X+1Y), P(X+2Y), \ldots, P(X+(n+1)Y)$

WHP

$g(X+1Y), g(X+2Y), \ldots, g(X+(n+1)Y)$

But $g(X+uY)=h(u)$ is a polynomial of degree $n$.
Interpolate to get $h(0)=g(X)$

$g(X) = \sum_{\sigma \in S_n} \prod_{i \in [n]} X_{i \sigma(i)}$

$P \text{ errs on } \leq 1/(8n)$
Hardness amplification

\[ X = (X_{ij}) \text{ matrix,} \quad \text{Per}(X) = \sum_{\sigma \in S_n} \prod_{i \in [n]} X_{i\sigma(i)} \]

If the Permanent can be efficiently computed for most inputs, then it can for all inputs!

If the Permanent is hard in the worst-case, then it is also hard on average

Worst-case \(\rightarrow\) Average case reduction
Works for any low degree polynomial.

Arithmetization: Boolean functions \(\rightarrow\) polynomials
Thm [Nisan]  Permanent $\in 2IP$

Thm [Lund-Fortnow-Karloff-Nisan]  coNP $\subseteq IP$

Thm [Shamir]  IP = PSPACE

Thm [Babai-Fortnow-Lund]  2IP = NEXP

Thm [Arora-Safra, Arora-Lund-Motwani-Sudan-Szegedy]  PCP = NP
Conceptual meaning

Thm [Lund-Fortnow-Karloff-Nisan] $\text{coNP} \subseteq \text{IP}$
- Tautologies have short, efficient proofs
- $\forall$ can be replaced by $\exists$ (and prob interaction)

Thm [Shamir] $\text{IP} = \text{PSPACE}$
- Optimal strategies in games have efficient pfs
- Optimal strategy of one player can simply be a random strategy

Thm [Babai-Fortnow-Lund] $\text{2IP} = \text{NEXP}$

Thm [Feige-Goldwasser-Lovasz-Safra-Szegedy] $\text{2IP}=\text{NEXP} \rightarrow \text{CLIQUE}$ is hard to approximate
PCP: Probabilistically Checkable Proofs

Prover \(\xrightarrow{w} \) Verifier

Wiles \(\xrightarrow{W} \) Referee

Accept \((x, w_i, w_j, w_k)\) iff \(x \in L\) WHP

NP verifier: Can read all of \(w\)
PCP verifier: Can (randomly) access only a constant number of bits in \(w\)

PCP Thm [AS, ALMSS]: \(NP = PCP\)

Proofs \(\rightarrow\) robust proofs reduction

PCP Thm [Dinur '06]: gap amplification
3-SAT is $7/8 + \varepsilon$ hard

3-SAT is $.99$ hard

3-SAT is $.995$ hard

3-SAT is $1 - \frac{1}{n}$ hard

3-SAT is $1 - \frac{2}{n}$ hard

3-SAT is $1 - \frac{1}{n}$ hard

3-SAT is $1 - \frac{4}{n}$ hard

Inapprox
Gap amplification

3-SAT is hard

IP

NP 3-SAT hard

SL=L

Space Complexity

SL=L : Graph Connectivity $\in$ LogSpace

Combinatorial

Algebraic

PCP

2IP

IP
Getting out of mazes / Navigating unknown terrains (without map & memory)

*n*-vertex maze/graph Only a local view (logspace)

Thm [Aleliunas-Karp-Lipton-Lovasz-Rackoff ’80]:
A random walk will visit every vertex in $n^2$ steps (with probability >99%)

Thm [Reingold ’05]: SL=L:
A deterministic walk, computable in Logspace, will visit every vertex.
Uses ZigZag expanders
[Reingold-Vadhan-Wigderson]
Dinur’s proof of PCP thm ‘06
log n phases:
Inapprox gap
Amplification Phase:
Graph powering and composition
(of the graph of constraints)

3-SAT is hard

3-SAT is 1-1/n hard

3-SAT is 1-2/n hard

3-SAT is 1-4/n hard

3-SAT is .995 hard

3-SAT is .99 hard

Connected graphs

λ < 1-1/n

λ < 1-2/n

λ < 1-4/n

λ < .995

Expanders

λ < .99

Reingold’s proof of SL=L ’05
log n phases:
Spectral gap amplification Phase:
Graph powering and composition

Zigzag product

Expander graphs
Comm Networks
Error correction
Complexity, Math ...
Conclusions & Open Problems
Computational definitions and reductions give new insights to millennia-old concepts, e.g. Learning, Proof, Randomness, Knowledge, Secret

Theorem(s): If FACTORING is hard then
- Learning simple functions is impossible
- Natural Proofs of hardness are impossible
- Randomness does not speed-up algorithms
- Zero-Knowledge proofs are possible
- Cryptography & E-commerce are possible
Open Problems

Can we base cryptography on $P \neq NP$?
Is 3-SAT hard on average?

Thm [Feigenbaum-Fortnow, Bogdanov-Trevisan]: Not with Black-Box reductions, unless...
The polynomial time hierarchy collapses.

Explore non Black-Box reductions!
What can be gleaned from program code?