Computational Complexity; slides 13, HT 2022
Randomisation (continued)

Prof. Paul W. Goldberg (Dept. of Computer Science,
University of Oxford)

HT 2022

Reducing SAT to USAT with the aid of randomness

We give another example of a task where randomisation seems to be useful.
Also, interesting technique; illustration of probabilistic reasoning.

USAT: given a formula φ with at most 1 satisfying assignment, determine whether it is satisfiable. (U stands for "unique")

So, USAT is no harder than SAT, and in a sense it's also no easier.

Afterwards: a quick look at interactive proofs, another setting where randomisation is important

Reducing SAT to USAT with the aid of randomness

We reduce SAT to USAT.

Motivation: known algorithms for SAT take time $poly(n)2^n$. The "strong exponential time hypothesis" asserts that you *need* time proportional to 2^n.[1]
But: note Grover's algorithm, a quantum algorithm solving USAT in time $poly(n)2^{n/2}$. Reducing SAT to USAT means that on a quantum machine, SAT is also solved in time $poly(n)2^{n/2}$!

[1](non-strong) ETH: for 3SAT, 2^{kn} needed for some $k > 0$

Reducing SAT to USAT with the aid of randomness

We reduce SAT to USAT.

Motivation: known algorithms for SAT take time $poly(n)2^n$. The "strong exponential time hypothesis" asserts that you *need* time proportional to 2^n.[1]
But: note Grover's algorithm, a quantum algorithm solving USAT in time $poly(n)2^{n/2}$. Reducing SAT to USAT means that on a quantum machine, SAT is also solved in time $poly(n)2^{n/2}$!

Challenge: Given φ, construct ψ such that ψ has a unique satisfying assignment iff φ is satisfiable.

[1](non-strong) ETH: for 3SAT, 2^{kn} needed for some $k > 0$

Reducing SAT to USAT with the aid of randomness

We reduce SAT to USAT.

Motivation: known algorithms for SAT take time $poly(n)2^n$. The "strong exponential time hypothesis" asserts that you *need* time proportional to 2^n.[1]
But: note Grover's algorithm, a quantum algorithm solving USAT in time $poly(n)2^{n/2}$. Reducing SAT to USAT means that on a quantum machine, SAT is also solved in time $poly(n)2^{n/2}$!

Challenge: Given φ, construct ψ such that ψ has a unique satisfying assignment iff φ is satisfiable.

Idea: $\psi := \varphi \wedge \rho$, where ρ is some other formula over the same variables.

[1] (non-strong) ETH: for 3SAT, 2^{kn} needed for some $k > 0$

Reducing SAT to USAT

Challenge: Given φ, construct ψ such that ψ has a unique satisfying assignment iff φ is satisfiable.
Idea: $\psi := \varphi \wedge \rho$, where ρ is some other formula over the same variables.
Extension of the idea: $\psi_1 := \varphi \wedge \rho_1$, ... ,$\psi_k := \varphi \wedge \rho_k$; look for satisfying assignment of any of these...

Problem: Think of φ as having been chosen by an opponent. Given a choice of $\rho_1, \ldots, \rho_k$, he can pick a φ that fails for your choice. This is where randomness helps!

Reducing SAT to USAT

Challenge: Given φ, construct ψ such that ψ has a unique satisfying assignment iff φ is satisfiable.
Idea: $\psi := \varphi \wedge \rho$, where ρ is some other formula over the same variables.
Extension of the idea: $\psi_1 := \varphi \wedge \rho_1$, ... ,$\psi_k := \varphi \wedge \rho_k$; look for satisfying assignment of any of these...

Problem: Think of φ as having been chosen by an opponent. Given a choice of $\rho_1, \ldots, \rho_k$, he can pick a φ that fails for your choice. This is where randomness helps!

(random) parity functions: let $x_1, \ldots, x_n$ be the variables of φ.
Let $\pi := \oplus_{x \in R}(x) \oplus b$ where each x_i is added to R with prob. $\frac{1}{2}$, and b is chosen to be TRUE/FALSE with equal probability $\frac{1}{2}$.

Think of R as standing for "relevant attributes"

Reducing SAT to USAT

Q: Why are random parity functions great?

A: Consider φ with set S of satisfying assignments. For random p.f. π, the expected number of satisfying assignments of $\varphi \wedge \pi$ is $\frac{1}{2}|S|$.

To see this, note that any satisfying assignment of φ gets eliminated with probability $\frac{1}{2}$.

Reducing SAT to USAT

Q: Why are random parity functions great?
A: Consider φ with set S of satisfying assignments. For random p.f. π, the expected number of satisfying assignments of $\varphi \wedge \pi$ is $\frac{1}{2}|S|$.

To see this, note that any satisfying assignment of φ gets eliminated with probability $\frac{1}{2}$.

Corollary: letting $\rho_k := \pi_1 \wedge \ldots \wedge \pi_k$ for independently randomly-chosen π_i, the expected number of satisfying assignments to $\varphi \wedge \rho_k$ is $|S|/2^k$.

Reducing SAT to USAT

Q: Why are random parity functions great?
A: Consider φ with set S of satisfying assignments. For random p.f. π, the expected number of satisfying assignments of $\varphi \wedge \pi$ is $\frac{1}{2}|S|$.

To see this, note that any satisfying assignment of φ gets eliminated with probability $\frac{1}{2}$.

Corollary: letting $\rho_k := \pi_1 \wedge \ldots \wedge \pi_k$ for independently randomly-chosen π_i, the expected number of satisfying assignments to $\varphi \wedge \rho_k$ is $|S|/2^k$.

This suggests the following approach:

- Generate ρ_k as above, for each $k = 1, 2, \ldots, n+1$.
- Search for a satisfying assignment to $\varphi \wedge \rho_k$.

Need to argue that for $k \approx \log_2 |S|$, we have reasonable chance of producing a formula with a *unique* s.a.

Pairwise independence of random p.f's:

Given $x \neq x' \in S$, and a random parity function π, we have:

$\Pr[x \text{ satisfies } \pi] = \frac{1}{2}$ $\Pr[x' \text{ satisfies } \pi] = \frac{1}{2}$

In addition:

$\Pr[x \text{ satisfies } \pi | x' \text{ satisfies } \pi] = \frac{1}{2}$

Proof:

For any x, $\pi(x) = v.x$ (or, $\neg v.x$) where v is characteristic vector of relevant attributes R of π.

($v.x$ denotes sum (XOR) of entries of x where corresponding entry of v is 1)

Let i be a bit position where $x'_i = 1$ and $x_i = 0$. i gets added to R with probability $\frac{1}{2}$, so value of $\pi(x')$ gets flipped with probability $\frac{1}{2}$.

similarly for conjunctions of random parity functions

Reducing SAT to USAT

For some k, we have $2^{k-2} \leq |S| \leq 2^{k-1}$.

Lemma: $\Pr[\text{there is unique } x \in S \text{ satisfying } \varphi \wedge \rho_k] \geq \frac{1}{8}$ (probability is w.r.t. random choice of ρ_k).

Proof: Let $p = 2^{-k}$ be the probability that $x \in S$ satisfies ρ_k.
Let N be the random variable consisting of the number of s.a.'s of $\varphi \wedge \rho_k$.
$E[N] = |S|p \in [\frac{1}{4}, \frac{1}{2}]$.

$$\Pr[N \geq 1] \geq \sum_{x \in S} \Pr[x \models \rho_k] - \sum_{x < x' \in S} \Pr[x \models \rho_k \wedge x' \models \rho_k] = |S|p - \binom{|S|}{2} p^2$$

By pairwise independence and union bound, we have $\Pr[N \geq 2] \leq \binom{|S|}{2} p^2$. So

$$\Pr[N = 1] = \Pr[N \geq 1] - \Pr[N \geq 2] \geq |S|p - 2\binom{|S|}{2} p^2 \geq |S|p - |S|^2 p^2 \geq \frac{1}{8}.$$

(where the last inequality uses $\frac{1}{4} \leq |S|p \leq \frac{1}{2}$.)

Interactive proofs

- an important application of randomisation in context of computational complexity

NP problems as "one-round interrogation":

skeptic: show me a solution
prover: ⟨solution⟩

skeptic can easily *check* prover's solution.
prover is "all-powerful".

A problem $\mathcal{X}$ is in NP if there's a poly-time TM (the skeptic), and a function (the prover) that can convince the skeptic...

Can an extension of above protocol "capture" other complexity classes?

Interactive proofs

- General idea: multi-round interaction

c.f. mathematician with new theorem, tries to convince colleagues...

Idea for definition: A problem belongs to IP if there's a communication protocol with a function $\mathcal{P}$ (the prover) and a poly-time computable function $\mathcal{V}$ (the verifier) such that:

- for problem-instance $\mathcal{I}$ of size n, allow poly(n) rounds of interaction (sequence of questions/challenges). Let's limit messages to polynomial length.
- $\mathcal{P}$ and $\mathcal{V}$'s messages may depend on previous interaction
- $\mathcal{V}$ ends up accepting iff $\mathcal{I}$ is a yes-instance...

Interactive proofs

- General idea: multi-round interaction

c.f. mathematician with new theorem, tries to convince colleagues...

Idea for definition: A problem belongs to IP if there's a communication protocol with a function $\mathcal{P}$ (the prover) and a poly-time computable function $\mathcal{V}$ (the verifier) such that:

- for problem-instance $\mathcal{I}$ of size n, allow poly(n) rounds of interaction (sequence of questions/challenges). Let's limit messages to polynomial length.
- $\mathcal{P}$ and $\mathcal{V}$'s messages may depend on previous interaction
- $\mathcal{V}$ ends up accepting iff $\mathcal{I}$ is a yes-instance...

But: consider *deterministic* verifier. Prover can supply all answers "upfront": no need to interact.

The Complexity Class IP

Definition. A decision problem $\mathcal{L}$ belongs to the complexity class IP if there is

- a communication protocol $\mathcal{C}$ and
- a randomised polynomial-time bounded algorithm $\mathcal{V}$ (the verifier)

with the property that

1. there is a function $\mathcal{P}$ (the prover) such that if $w \in \mathcal{L}$

$$\Pr[\ \mathcal{P} \text{ persuades } \mathcal{V} \text{ to accept } w] \geq \frac{2}{3}$$

2. for all "prover" functions $\mathcal{P}'$, if $w \notin \mathcal{L}$

$$\Pr[\ \mathcal{P}' \text{ persuades } \mathcal{V} \text{ to accept } w] \leq \frac{1}{3}$$

$\mathcal{L}$ belongs to IP$[k]$ if at most k communication rounds are necessary.

Graph-Non-Isomorphism in IP

Recall. An isomorphism between two graphs H and G is a function $f : V(H) \to V(G)$ such that

1. f is a bijection between $V(H)$ and $V(G)$ and
2. for all $u, v \in V(H)$: $\{u, v\} \in E(H) \iff \{f(v), f(u)\} \in E(G)$.

Graph isomorphism has no known poly-time algorithm

Graph isomorphism is easily seen to be in NP but unlikely to be NP-complete, has subexponential algorithm

It's also known that if GI is NP-complete, then $\Sigma_2^P = \Pi_2^P$, thus PH collapses

Graph-Non-Isomorphism in IP

(c.f. coke vs pepsi taste test)

Input. Graphs G_1 and G_2.

Communication.

1. $\mathcal{V}$ randomly chooses $i \in \{1, 2\}$, randomly permutes vertices of G_i to obtain new graph H isomorphic to G_i.
2. $\mathcal{V}$ sends H to $\mathcal{P}$
3. $\mathcal{P}$ identifies the graph G_j to which H is isomorphic, and sends j back.
4. $\mathcal{V}$ accepts if $i = j$.

Repeat (in parallel or sequentially) until $\mathcal{V}$ "reasonably convinced".

(c.f. coke vs pepsi taste test)

Input. Graphs G_1 and G_2.

Communication.

1. $\mathcal{V}$ randomly chooses $i \in \{1, 2\}$, randomly permutes vertices of G_i to obtain new graph H isomorphic to G_i.
2. $\mathcal{V}$ sends H to $\mathcal{P}$
3. $\mathcal{P}$ identifies the graph G_j to which H is isomorphic, and sends j back.
4. $\mathcal{V}$ accepts if $i = j$.

Repeat (in parallel or sequentially) until $\mathcal{V}$ "reasonably convinced".

Theorem. IP = PSPACE

(See Sipser, Theorem 10.29)
Arora/Barak: IP=PSPACE (Chapter 8.3)

Zero-Knowledge Proofs

Applications.

1. Secure authentication. convince someone you know some password etc without revealing it
2. Auctions.
   - Several companies place bids for items/frequencies/mining rights ...
   - They place their bids simultaneously.
   - After the bidding process, each company wants to be convinced that the winner really bid more than itself.
   - The winner doesn't want to reveal their bid.

Next: graph isomorphism. Standard IP has prover reveal the isomorphism: let's disallow that!

A Zero-Knowledge Proof for Graph Isomorphism

Given: Two graphs G_1, G_2

Prover's secret: An isomorphism π between G_1, G_2

Prover wants to prove to Verifier that $G_1 \cong G_2$ without revealing π.

A Zero-Knowledge Proof for Graph Isomorphism

Given: Two graphs G_1, G_2

Prover's secret: An isomorphism π between G_1, G_2

Prover wants to prove to Verifier that $G_1 \cong G_2$ without revealing π.

Communication protocol.

1. $\mathcal{P}$ randomly selects $i \in \{1, 2\}$ and computes a random permutation of $|V(G_i)|$ generating a graph $H \cong G_i$
2. $\mathcal{P}$ sends H to $\mathcal{V}$ and keeps the isomorphism $f : H \cong G_i$.
3. $\mathcal{V}$ randomly selects $j \in \{1, 2\}$ and sends j back to $\mathcal{P}$.
4. $\mathcal{P}$ computes an isomorphism π_j (either f or $\pi \circ f$) between G_j and H, and sends it to $\mathcal{V}$.
5. $\mathcal{V}$ accepts if $H = \pi_j(G_j)$

A Zero-Knowledge Proof for Graph Isomorphism

Given: Two graphs G_1, G_2

Prover's secret: An isomorphism π between G_1, G_2

Prover wants to prove to Verifier that $G_1 \cong G_2$ without revealing π.

Communication protocol.

1. $\mathcal{P}$ randomly selects $i \in \{1,2\}$ and computes a random permutation of $|V(G_i)|$ generating a graph $H \cong G_i$
2. $\mathcal{P}$ sends H to $\mathcal{V}$ and keeps the isomorphism $f : H \cong G_i$.
3. $\mathcal{V}$ randomly selects $j \in \{1,2\}$ and sends j back to $\mathcal{P}$.
4. $\mathcal{P}$ computes an isomorphism π_j (either f or $\pi \circ f$) between G_j and H, and sends it to $\mathcal{V}$.
5. $\mathcal{V}$ accepts if $H = \pi_j(G_j)$

- If $G_1 \cong G_2$ then $\mathcal{P}$ can always convince $\mathcal{V}$.
- Otherwise, $\mathcal{P}$ fails with probability $\frac{1}{2}$, which again can be amplified.
- The computation can be done efficiently.