Algebrization

Jesse Comer & Tanner Duve

June 2023

S. Aaronson & A. Wigderson. *Algebrization: A New Barrier in Complexity Theory*, 2008

Barriers in Complexity Theory

This talk is about the *difficulty* of resolving many open complexity-theoretic problems.

Some important complexity-theoretic statements and proofs are known to *relativize*.

Algebrization is a generalization of the notion of relativization.

We will use two running examples:

$$\mathbf{PSPACE} \subseteq \mathbf{IP}$$

and

$$\mathbf{NP} \subseteq \mathbf{P}.$$

Diagonalization

Recall:

Definition (Diagonalization)

Diagonalization is any technique that relies solely on the following properties of TMs:

1. (Encodings) The existence of an effective representation of TMs by strings.
2. (Simulation) The ability of one TM to simulate another without much overhead in running time or space.

Examples:

- the undecidability of the halting problem
- the time hierarchy theorems.

Relativization

Let $\mathcal{C}$, $\mathcal{D}$ denote arbitrary complexity classes.

A containment $\mathcal{C} \subseteq \mathcal{D}$ *relativizes* if, for all oracles A, we have that $\mathcal{C}^A \subseteq \mathcal{D}^A$.

A separation $\mathcal{C} \not\subseteq \mathcal{D}$ *relativizes* if, for all oracles A, we have that $\mathcal{C}^A \not\subseteq \mathcal{D}^A$.

A *proof technique* of a complexity-theoretic statement relativizes if it is still valid (with only small changes) when the classes are taken relative to an arbitrary oracle A.

All proofs using only diagonalization are relativizing.

Recall: $\mathbf{P}$ vs $\mathbf{NP}$ does not relativize

Theorem (Baker, Gill, Solovay, 1975)

There exist oracles A and B such that $\mathbf{P}^A = \mathbf{NP}^A$ and $\mathbf{P}^B \neq \mathbf{NP}^B$.

It follows that any proof of $\mathbf{P} = \mathbf{NP}$ or $\mathbf{P} \subsetneq \mathbf{NP}$ must use non-relativizing techniques.

In particular, diagonalization alone cannot resolve $\mathbf{P}$ vs $\mathbf{NP}$.

Do we have any non-relativizing techniques?

It is known that $\mathbf{PSPACE} \subseteq \mathbf{IP}$ is not a relativizing result:

Theorem (Chang et. al., 1988)

There exist oracles A and B such that $\mathbf{IP}^A = \mathbf{PSPACE}^A$ and $\mathbf{IP}^B \neq \mathbf{PSPACE}^B$.

In fact:

Theorem (Fortnow, Sipser, 1988)

There exist an oracle A such that $\mathbf{coNP}^A \not\subseteq \mathbf{IP}^A$.

Yet...

Theorem (Shamir, 1992)

$\mathbf{IP} = \mathbf{PSPACE}$.

PSPACE $\subseteq$ IP

Could the techniques in the proof of the **IP** theorem help us resolve **P** vs **NP**?

Theorem

PSPACE $\subseteq$ **IP**.

Proof.

Since TQBF is **PSPACE**-complete, it suffices to give an interactive protocol for TQBF. Let $\varphi := \forall x_1 \exists x_2 \forall x_3 \ldots \exists x_n \psi(x_1, x_2, \ldots, x_n)$ be an input to TQBF.

Using arithmetization and linearization operators on φ, we obtain the expression

$$\forall x_1 L_1 \exists x_2 L_1 L_2 \forall x_3 L_1 L_2 L_3 \ldots \exists x_n L_1 L_2 \ldots L_n p_\varphi(X_1, \ldots, X_n),$$

which the prover must convince the verifier is nonzero.

PSPACE $\subseteq$ IP

Theorem

PSPACE $\subseteq$ **IP**.

Proof.

Each round:

- P sends a univariate polynomial $S(X_i)$, claiming that $\mathcal{O}S(X_i) = C_i$ for some $C_i \in \mathbb{F}$, where $\mathcal{O} \in \{\exists X_i, \forall X_i, L_{X_i}\}$;
- V checks $S(0) + S(1)$, $S(0) \cdot S(1)$, or $a_1 S(0) + (1 - a_1)S(1)$, depending on $\mathcal{O}$;
- V sends a $r_i \in_R \mathbb{F}$ and requests that P show that $S(r_i)$ evaluates correctly.

In the final round, the verifier has a univariate polynomial $S(X_1)$, which P claims is equal to $C_1 \in \mathbb{F}$. The verifier directly evaluates $S(r_1)$ for some $r_1 \in_R \mathbb{F}$. $\square$

Why exactly doesn't this proof relativize?

Why does the proof of **PSPACE** $\subseteq$ **IP** not relativize?

Claim (False!)

$\mathbf{PSPACE}^A \subseteq \mathbf{IP}^A$ *for any oracle A.*

Proof.

TQBF, in general, is not complete for $\mathbf{PSPACE}^A$.

We now must consider TQBF^A, in which the underlying formula might contain gates calling the oracle A; this problem is $\mathbf{PSPACE}^A$-complete for any oracle A.

Let $\varphi := \forall x_1 \exists x_2 \forall x_3 \ldots \exists x_n \psi(x_1, x_2, \ldots, x_n)$ be an input to TQBF^A.

Q: How do we arithmetize a formula with A gates?

A: Extension polynomials!

Extension Polynomials and Oracles (over Finite Fields)

Definition (Extension Polynomials)

Let $A_m : \{0,1\}^m \to \{0,1\}$ be a Boolean function, and let $\mathbb{F}$ be a finite field. Then an *extension polynomial* of A_m over $\mathbb{F}$ is a polynomial $\tilde{A}_{m,\mathbb{F}} : \mathbb{F}^m \to \mathbb{F}$ such that $\tilde{A}_{m,\mathbb{F}}(x) = A_m(x)$ whenever $x \in \{0,1\}^m$.

Definition (Oracles)

An *oracle* A is a collection $(A_m)_{m \in \mathbb{Z}^+}$, where $A_m : \{0,1\}^m \to \{0,1\}$.

Extension Polynomials and Oracles (over Finite Fields)

Definition (Extension Oracles)

An *extension oracle* $\tilde{A}$ of an oracle A is a collection of polynomials $\tilde{A}_{m,\mathbb{F}} : \mathbb{F}^m \to \mathbb{F}$, one for each $m \in \mathbb{Z}^+$ and finite field $\mathbb{F}$ such that

1. $\tilde{A}_{m,\mathbb{F}}$ is an extension of A_m for all m and $\mathbb{F}$, and
2. there exists some c such that $mdeg(\tilde{A}_{m,\mathbb{F}}) \leq c$ for all m and $\mathbb{F}$,

where $mdeg(p)$ (the *multidegree* of p) denotes the maximum degree of any x_i.

Given a complexity class $\mathcal{C}$, we write $\mathcal{C}^{\tilde{A}}$ for the class of languages decidable by a $\mathcal{C}$ machine that can query $\tilde{A}_{m,\mathbb{F}}$ for any integer m and finite field $\mathbb{F}$.

Let's return to our attempt to relativize the proof that **PSPACE** $\subseteq$ **IP**.

Why does the proof of **PSPACE** $\subseteq$ **IP** not relativize?

Claim (False!)

PSPACE$^A \subseteq$ **IP**^{A} *for any oracle A.*

Proof.

Let $\varphi := \forall x_1 \exists x_2 \forall x_3 \ldots \exists x_n \psi(x_1, x_2, \ldots, x_n)$ be an input to TQBF^A. When arithmetizing this formula, we replace an A gate with m inputs with $\tilde{A}_{m,\mathbb{F}}$.
The interactive steps and checks all work exactly the same.

In the final step, V needs to directly evaluate the polynomial over a randomly chosen field element. This may not be possible:

He has access to A_m, but he needs access to $\tilde{A}_{m,\mathbb{F}}$.

Theorem

PSPACE$^A \subseteq$ **IP**$^{\tilde{A}}$ *for any oracle A, and any finite field extension $\tilde{A}$ of A.*

Algebrization

Theorem

$\mathbf{PSPACE}^A \subseteq \mathbf{IP}^{\tilde{A}}$ *for any oracle* A*, and any finite field extension* $\tilde{A}$ *of* A*.*

Let's turn this into a definition:

Definition

We say the complexity class inclusion $\mathcal{C} \subseteq \mathcal{D}$ *algebrizes* if $\mathcal{C}^A \subseteq \mathcal{D}^{\tilde{A}}$ for all oracles A and all finite field extensions $\tilde{A}$ of A.

Definition

We say the separation $\mathcal{C} \not\subseteq \mathcal{D}$ *algebrizes* if $\mathcal{C}^{\tilde{A}} \not\subseteq D^A$ for all $A, \tilde{A}$.

PSPACE $\subseteq$ IP does not *relativize*, but it does *algebrize*.

Deterministic Query Complexity

Let $A : \{0,1\}^n \to \{0,1\}$ be a Boolean function.

We can view A as a length $N = 2^n$ string encoding its truth table.

For example, the *MAJORITY* function on 3-bit inputs has the following truth table:

Input 1	Input 2	Input 3	Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

and would be represented by the string 00010111 of length $N = 2^3 = 8$.

Deterministic Query Complexity

Let $N = 2^n$. We can view a Boolean function $f : \{0,1\}^N \to \{0,1\}$ as computing a property of functions $A : \{0,1\}^n \to \{0,1\}$.

Suppose we can compute f by querying the input A at various points $x \in \{0,1\}^n$.

The *deterministic query complexity* of f (notation: $D(f)$) is the minimum number of queries made by any deterministic algorithm that evaluates f on every input.

Deterministic Query Complexity

Definition (Deterministic Query Complexity)

Let $f : \{0,1\}^N \to \{0,1\}$ be a Boolean function, and let $\mathcal{M}$ be the set of deterministic algorithms M such that M^A outputs $f(A)$ for every oracle $A : \{0,1\}^n \to \{0,1\}$.

Then the *deterministic query complexity of f* is defined as

$$D(f) := \min_{m \in \mathcal{M}} \max_{A} T_M(A),$$

where $T_M(A)$ is the number of queries to A made by M^A.

Lower bounds are proven via *adversary arguments*.

Deterministic Query Complexity Example: OR

Consider $OR : \{0,1\}^N \to \{0,1\}$, where

$$OR(A) = 1 \text{ if and only if } A(x) = 1 \text{ for some } x \in \{0,1\}.$$

Proposition

$D(OR) = 2^n$.

Proof.

Suppose some algorithm M makes only $k < 2^n$ queries in the worst case.

Then it makes at most k queries $x_1, \ldots, x_k \in \{0,1\}^n$ on the all-zeroes function $A : \{0,1\}^n \to \{0,1\}$.

We can choose $B : \{0,1\}^n \to \{0,1\}$ so that it agrees with A on $x_1, \ldots x_k$, but has $B(y) = 1$ for some $y \in \{0,1\}^n$ such that $y \neq x_i$ for all $i \leq k$. $\square$

Algebraic Query Complexity

Definition (Deterministic Algebraic Query Complexity)

Let $f : \{0,1\}^N \to \{0,1\}$ be a Boolean function, $\mathbb{F}$ a finite field, and $c \in \mathbb{Z}^+$.

Let $\mathcal{M}$ be the set of deterministic algorithms M such that $M^{\tilde{A}}$ outputs $f(A)$ for every oracle $A : \{0,1\}^n \to \{0,1\}$ and finite field extension $\tilde{A} : \mathbb{F}^n \to \mathbb{F}$ of A with $mdeg(\tilde{A}) \leq c$.

Then the *deterministic algebraic query complexity of f over $\mathbb{F}$* is defined as

$$\tilde{D}_{\mathbb{F},c}(f) := \min_{m \in \mathcal{M}} \max_{A,\tilde{A}: mdeg(\tilde{A}) \leq c} T_M(\tilde{A}),$$

where $T_M(\tilde{A})$ is the number of queries to $\tilde{A}$ made by $M^{\tilde{A}}$.

To prove lower bounds, we construct *adversary polynomials*.

Facts about Multilinear Polynomials

First, we state some facts regarding multilinear polynomials.

Let $z = z_1 \ldots z_n \in \{0,1\}^n$. Define

$$\delta_z(x) = \prod_{i \leq n : z_i = 1} x_i \prod_{i \leq n : z_i = 0} (1 - x_i).$$

Then for any multilinear polynomial $m : \mathbb{F}^n \to \mathbb{F}$, we can write m as follows:

$$m(x) = \sum_{z \in \{0,1\}^n} m_z \delta_z(x).$$

Furthermore, every Boolean function $A : \{0,1\}^n \to \{0,1\}$ has a unique multilinear extension over a field $\mathbb{F}$.

Lower Bounds for Algebraic Query Complexity

Lemma

Let $\mathbb{F}$ be a field and let $y_1, \ldots, y_t$ be points in $\mathbb{F}^n$. Then there exists a multilinear polynomial $m : \mathbb{F}^n \to \mathbb{F}$ such that

1. $m(y_i) = 0$ *for all* $i \in [t]$, *and*
2. $m(z) = 1$ *for at least* $2^n - t$ *Boolean points* z.

Proof.

Suppose

$$m(x) = \sum_{z \in \{0,1\}^n} m_z \delta_z(x).$$

Then requirement (1) corresponds to t linear equations over $\mathbb{F}$ in the variables m_z.

Then we can choose a solution to this system of equations with $2^n - t$ of the m_z's set to 1. Hence $m(z) = 1$ for at least $2^n - t$ values. $\square$

Lower Bounds for Algebraic Query Complexity

Lemma (Adversary Lemma)

Let $\mathbb{F}$ be a field and let $y_1, \ldots, y_t$ be points in $\mathbb{F}^n$. Then for at least $2^n - t$ Boolean points $w \in \{0,1\}^n$, there exists a multiquadratic extension polynomial $p : \mathbb{F}^n \to \mathbb{F}$ such that

1. $p(y_i) = 0$ *for all* $i \in [t]$,
2. $p(w) = 1$, *and*
3. $p(z) = 0$ *for all Boolean points* $z \neq w$.

Proof.

Let $m : \mathbb{F}^n \to \mathbb{F}$ be the multilinear polynomial from the preceding lemma, and let $w \in \{0,1\}^n$ be such that $m(w) = 1$.

Then $p(x) = m(x)\delta_w(x)$ satisfies (1)-(3). $\square$

Lower Bounds for Algebraic Query Complexity

Lemma (Generalized Adversary Lemma)

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function, and let $\mathcal{F}$ be a collection of fields (possibly with multiplicity).

For every $\mathbb{F} \in \mathcal{F}$, let $\mathcal{Y}_{\mathbb{F}} \subseteq \mathbb{F}^n$, and $p_{\mathbb{F}} : \mathbb{F}^n \to \mathbb{F}$ be a multiquadratic polynomial over $\mathbb{F}$ extending f.

Then there exists $B \subseteq \{0,1\}^n$ with $|B| \leq \sum_{\mathbb{F}\in\mathcal{C}} |\mathcal{Y}_{\mathcal{F}}|$ such that, for all Boolean function $f' : \{0,1\}^n \to \{0,1\}$ that agree with f on B, there exist multiquadratic polynomials $p'_{\mathbb{F}} : \mathbb{F}^n \to \mathbb{F}$ (one for each $\mathbb{F} \in \mathcal{F}$) such that

(i) *$p'_{\mathbb{F}}$ extends f', and*

(ii) *$p'_{\mathbb{F}}(y) = p_{\mathbb{F}}(y)$ for all $y \in \mathcal{Y}_{\mathbb{F}}$.*

$\mathbf{NP} \not\subseteq \mathbf{P}$ does not algebrize

Theorem

There exist $A, \tilde{A}$ such that $\mathbf{NP}^{\tilde{A}} \subseteq \mathbf{P}^{A}$.

Proof.

Let A be any **PSPACE**-complete language, and $\tilde{A}$ its unique multilinear extension.

The multilinear extension of a **PSPACE** language can be computed in **PSPACE** (Babai, Fortnow, Lund, 1991).

Thus

$$\mathbf{NP}^{\tilde{A}} = \mathbf{NP}^{\mathbf{PSPACE}} \subseteq \mathbf{NPSPACE} = \mathbf{PSPACE} \subseteq \mathbf{P}^{\mathbf{PSPACE}} = \mathbf{P}^{A}.$$

□

$\mathbf{NP} \subseteq \mathbf{P}$ does not algebrize

Theorem

There exist $A, \tilde{A}$ such that $\mathbf{NP}^A \not\subseteq \mathbf{P}^{\tilde{A}}$.

Proof.

The proof is analogous to the BGS lazy diagonalization construction.

We construct the oracle A and its extension oracle $\tilde{A}$ recursively.

At each stage of the construction, we fix some A_m functions, as well as $\tilde{A}_{m,\mathbb{F}}$ for *every* finite field $\mathbb{F}$.

The key difference from the BGS proof is that, when we simulate a machine M_i and it rejects an input 1^n, we will use the generalized adversary lemma to choose an appropriate extension oracle.

$\mathbf{NP} \subseteq \mathbf{P}$ does not algebrize

Proof.

Clearly, the following language is in $\mathbf{NP}^A$ for all oracles A:

$$L = \{1^n \mid \exists w \in \{0,1\}^n \text{ s.t. } A_n(w) = 1\}.$$

We want to choose A and $\tilde{A}$ so that $L \notin \mathbf{P}^{\tilde{A}}$.

Fix an enumeration $M_1, M_2, \ldots$ of all $\mathbf{DTIME}(n^{\log(n)})$ oracle machines. Define

$$M_i(n) = \begin{cases} 1 & \text{if } M_i \text{ accepts } 1^n \\ 0 & \text{otherwise.} \end{cases} \quad \text{and} \quad L(n) = \begin{cases} 1 & \text{if } 1^n \in L \\ 0 & \text{otherwise.} \end{cases}$$

We want to ensure that for each $i \in \mathbb{Z}^+$, there's some $n \in \mathbb{Z}^+$ such that

$$M_i(n) \neq L(n).$$

$\mathbf{NP} \subseteq \mathbf{P}$ does not algebrize

Proof.

The construction of $\tilde{A}$ proceeds in stages:

- Assume for each $j < i$ that
  1. $L(j)$ is fixed, and
  2. there's some n_j such that $M_j(n_j) \neq L(n_j)$.
- Let S_j be the set of indices n such that some $\tilde{A}_{n,\mathbb{F}}$ is queried by M_j on input 1^{n_j}.
- Let $T_i := \bigcup_{j<i} S_j$.
- For each $n \in T_i$, we consider $\tilde{A}_{n,\mathbb{F}}$ to be fixed.
- Let n_i be least such that $n_i \notin T_i$ and $2^{n_i} > n_i^{\log(n_i)}$.
- Simulate M_i on 1^{n_i}. If M_i queries some $\tilde{A}_{n,\mathbb{F}}(y)$, then
  1. If $n \in T_i$, return consistently;
  2. Otherwise, return 0.

$\mathbf{NP} \subseteq \mathbf{P}$ does not algebrize

Proof.

Let S_i denote the set of n such that M_i queried some $\tilde{A}_{n,\mathbb{F}}$.

For all $m \in S_i \setminus T_i$, other than n_i, and all $\mathbb{F}$, set $\tilde{A}_{n,\mathbb{F}}$ to the constant 0 polynomial.

For n_i, we distinguish cases, depending on whether or not M_i accepted 1^{n_i}.

If M_i accepted 1^{n_i}, then we set $\tilde{A}_{n_i,\mathbb{F}}$ to the constant 0 polynomial for all $\mathbb{F}$ (and hence $L(n_i) = 0$).

$\mathbf{NP} \subseteq \mathbf{P}$ does not algebrize

Proof.

If M_i rejected 1^{n_i}, then let $\mathcal{Y}_{\mathbb{F}} = \{y \in \mathbb{F}^{n_i} \mid M_i \text{ queried } \tilde{A}_{n_i,\mathbb{F}(y)}\}$.

We must have that $\sum_{\mathbb{F}} |\mathcal{Y}_{\mathbb{F}}| \leq n_i^{\log(n_i)}$.

By the generalized adversary lemma, there exists $w \in \{0,1\}^{n_i}$ such that for all $\mathbb{F}$, we can choose a multiquadratic polynomial $\tilde{A}_{n_i,\mathbb{F}} : \mathbb{F}^{n_i} \to \mathbb{F}$ such that

(i) $\tilde{A}_{n_i,\mathbb{F}}(y) = 0$ for all $y \in \mathcal{Y}_{\mathbb{F}}$,

(ii) $\tilde{A}_{n_i,\mathbb{F}}(w) = 1$, and

(iii) $\tilde{A}_{n_i,\mathbb{F}}(z) = 0$ for all Boolean $z \neq w$.

In particular, the answers to queries to $\tilde{A}_{n_i,\mathbb{F}}$ are consistent with all queries that have been make so far, but $\tilde{A}_{n_i,\mathbb{F}}(w) = 1$ for some $w \in \{0,1\}^n$.

Thus $L(n_i) = 1$. □

Concluding remarks

Some additional non-algebrizing statements:

1. $\mathbf{PSPACE} \not\subseteq \mathbf{P}$,
2. $\mathbf{NP} \subseteq \mathbf{BPP}$,
3. $\mathbf{NP} \subseteq \mathbf{P}/\mathbf{poly}$,
4. $\mathbf{NEXP} \not\subseteq \mathbf{P}/\mathbf{poly}$,
5. $\mathbf{EXP}^{\mathbf{NP}} \not\subseteq \mathbf{P}/\mathbf{poly}$,
6. and more....

Questions?