Bounded Arithmetic, Expanders, and Monotone Propositional Proofs

Sam Buss
joint work with Valentine Kabanets,
Antonina Kolokolova & Michal Koucký

Takeuti Symposium on Advances in Logic
Kobe, Japan
September 20, 2018

Talk overview

A. Bounded arithmetic theories are weak subtheories of Peano arithmetic with close connections to
   - Feasible complexity classes, e.g. P and NC^1.
   - Propositional proof complexity, via the Paris-Wilkie and the Cook translations.

   Moral: A proof in bounded arithmetic corresponds to a uniform family of propositional proofs.

B. Monotone propositional logic (MLK) is the propositional sequent calculus with no use of negation ($\neg$) permitted.
   LK is the usual propositional sequent calculus.
   Main theorem: MLK polynomially simulates LK.

C. This talk describes how to formalize, in VNC^1 — a theory of bounded arithmetic corresponding to NC^1, the construction of expander graphs. Using prior work [Arai; Cook-Morioka; Atserias-Galesi-Pudlák; Jeřábek], this proves the main theorem.

- The first Bounded Arithmetic theories ($I\Delta_0$, [Parikh'71, ...]) and (S_2^i, T_2^1, U_2^1, V_2^1 [B'85]) were for alternating linear time and for polynomial time (P), the polynomial hierarchy (PH), polynomial space and exponential time.
- Takeuti [90]: the RSUV isomorphism translates theories such as U_2^1 into theories for feasible classes below P.
- Clote-Takeuti [1992] achieved this for such several theories, including for alternating logarithmic time (Alogtime, or uniform NC^1), log space (L) and nondeterministic log space (NL). Especially, they defined the bounded arithmetic theory TNC for Alogtime.
- Arai [2000] developed an improved theory AID similar to TNC: he showed in addition that the theory AID has the Cook correspondence with propositional LK proofs.
- Cook-Morioka ['05], Cook-Nguyen['10] give newer versions, esp. VNC^1.

Def'n: The **propositional sequent calculus (LK)** is a propositional proof system whose proofs consist of sequents, with a finite set of valid inference forms, for example

$$\wedge\text{:right}\ \frac{\Gamma \longrightarrow \Delta, A \qquad \Gamma \longrightarrow \Delta, B}{\Gamma \longrightarrow \Delta, A \wedge B}$$

$$\text{Cut}\ \frac{\Gamma \longrightarrow \Delta, A \qquad A, \Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta}$$

Def'n: The **monotone sequent calculus (MLK)** is LK restricted to allow only monotone formulas to appear in sequents.

MLK proofs are allowed to be dag-like.

Main Theorem: LK proofs of monotone sequents can be simulated by polynomial size MLK proofs.

Technical talk outline

I. **Combinatorial construction of expander graphs**, avoiding algebraic concepts such as eigenvalues even in proofs of correctness.

II. This construction can be **carried out in** NC^1 (logarithmic depth Boolean circuits).

III. Combinatorial constructions are **provably correct** in the weak first-order theory VNC^1 corresponding to NC^1.

IV. **Application:** *Monotone* propositional logic (MLK) polynomially simulates non-monotone propositional logic (LK)

I. Construction of Expanders

Expander Graphs:

- Undirected graphs, allowing self-loops and multiple edges.
- Expander graphs are both **sparse** (usually constant degree) and **well connected**.
- A random walk on an expander graphs converges quickly
- Are used for pseudorandomness, e.g., for one-way functions, error-correcting codes, derandomization, etc.
- Are widely used in complexity theory, e.g.,
  - Reingold; Rozenman-Vadhan. USTCON in Logspace
  - Dinur: Combinatorial proof of PCP theorem
  - **Ajtai-Komlós-Szemerédi: AKS sorting networks**.

Definition of expander graph $G = (V, E)$, of constant degree d

For $U, \overline{U}$ a proper partition of the vertices V, let

$$\textit{edge-exp}_G(U) \;:=\; \frac{|E(U,\overline{U})|}{d \cdot \min(|U|, |\overline{U}|)}.$$

$E(U, \overline{U})$ is the set edges between E and $\overline{E}$.
The **edge expansion** of G is $\min_U(\textit{edge-exp}_G(U))$.

G is an **expander graph** if it has $\Omega(1)$ edge expansion.

—

Edge expansion can be lower bounded in terms of the spectral gap (second largest eigenvalue λ_2) of the adjacency matrix.

—

Our work requires instead *combinatorial* constructions and proofs.

Classical (non)construction: [Pinkser'73]
A randomly chosen degree d graph is an expander.

Iterative Constructions:
Start with finite size expander graph(s). Then iteratively use:

- Powering (to increase expansion).
- Zig-zag product or replacement product (to reduce the degree).
- Tensoring (to increase the size of the graph).
- Adding self-loops (helps maintain edge expansion).

Original construction [Reingold-Vadhan-Wigderson'02]

- Used Zig-zag product, proof based on spectral gap.
- [Alon, Schwartz, Shapira'08] used Replacement product with combinatorial argument.

Our arguments for powering will use also Mihail's combinatorial proof of mixing times from edge expansion [1989].

<u>The explicit construction:</u> (Similar to the prior constructions)
Starts with constants c, d and two "small" (fixed) graphs
- a $2d$-regular G_0 with edge expansion $\geq \epsilon = 1/1296$
- a d-regular H on $(2(4d)^2)^c$ vertices with edge expansion $\geq 1/3$.

Iterate: $G_{i+1} = [\bigcirc((\bigcirc G_i) \otimes (\bigcirc G_i))]^c \circ H$.

- Add self-loops ($\bigcirc$) to double the degree
- Tensor ($\otimes$) with itself
- Add self-loops
- Power to constant c
- Replace each vertex with H (replacement product, $\circ$)

Theorem: Each G_i is degree $2d$ and has edge expansion $\geq \epsilon$.
The size of G_{i+1} is greater than (size of $G_i)^2$, (size *squares*)
$|G_i| = (|G_0|D_0)^{2^i}/D_0 > 2^{2^i}$, where $D_0 = (2(4d)^2)^c$.

Graph operations in more detail: $G = (V, E)$ of degree D.

Adding self-loops: $\bigcirc G$.

Add D self-loops to every vertex.

Vertex set remains the same. Degree doubles to $2D$.

Tensoring with itself: $G \otimes G$.

"Crossproduct of G with itself".

Vertex set is $V \times V$.

Degree squares to become D^2.

Raise to power c: G^c.

Paths of length c in G are edges of G^c.
Vertex set is unchanged. Degree becomes D^c.

Graph operations in more detail: $G = (V, E)$ of degree D and $H = (V', E')$ of size $|V'| = D$ and degree d.

Replacement product: $G \circ H$.

Replace each G-vertex $v \in V$ with a copy H_v of H.

Thus vertex set is $V \times V'$.

An edge $e = (v_1, v_2)$ in G becomes

d parallel edges between vertices of H_{v_1} and H_{v_2}.

If v_2 is i-th-neighbor of v_1 in G, it uses i-th vertex of H_{v_1}.

Degree becomes $2d$.

Rotation map: For the replacement product, it is necessary to order the edges reaching each vertex. The **rotation map** of a graph G computes from $v \in V$ and $i < D$: the i-th neighbor w of v, and the index j such that v is the j-th neighbor of w.

Main Theorem 1: The rotation map of G_i is uniformly computable from i, j, v in

- Polynomial time.
- Alternating linear time.

Proof (a) Straightforward unwinding of construction gives the polynomial time algorithm.
(b) Alternating linear time: G_{i+1}'s rotation map is computed from G_i's rotation map in *constant* alternation *linear* time.
Only a *single* recursive call to G_i is needed.
E.g. for powering, nondeterministically guess the path of length c.
Then universally verify correctness of each step in the path.
Since the size of G_{i+1} is $>$ square of size G_i, the "linear time" is decreasing by factor of two with each recursive call. So the overall running time is linear (but not constant alternation). $\square$.

Since the graph G_i is exponentially bigger than the size of the inputs to the rotation map function, we get:

Corollary As a function of G_i, there is an alternating logarithmic time algorithm (an NC^1 algorithm) to compute the edge relation on G_i.

Key constructive justification of edge-expansion:

Lemma If U is a set of vertices of G_{i+1} with $\textit{edge-exp}_{G_{i+1}}(U) < \epsilon$, then there exists a set U' of vertices of G_i such that $\textit{edge-exp}_{G_i}(U') < \epsilon$.

Proof idea: There is an NC^1 algorithm to compute membership in U' in terms of U. The correctness is provable by purely combinatorial means without recourse to algebraic concepts such as eigenvalues.

Technical tools needed:

- Representing graphs and rotation maps.
- Definition of the expansion of a set U (as a rational).
- Summing sequences of rationals (common denominator).
- Arithmetic manipulations of these sequences.
- Cauchy-Schwartz inequality.

All of these can be done in the bounded arithmetic theory VNC^1...

III. Formalizability in bounded arithmetic VNC^1.

Notation: VNC^1 is a second-order theory of bounded arithmetic [Cook-Morioka'05], [Cook-Nguyen'10]; the first versions were defined by [Clote-Takeuti'92], [Arai'00].

VNC^1 corresponds in proof-theoretic strength to NC^1.

Its provably total functions are precisely the NC^1-functions.

$\underline{\mathrm{VNC}^1}$ First-order objects code (small) integers.
Second-order objects code strings, graphs, sequences, etc.
Σ_0^B is the set of formulas with no second order quantifiers.

<u>Axioms of VNC^1</u> include: BASIC axioms (purely universal).

Σ_0^B-Comprehension (and hence Σ_0^B-induction).

Σ_0^B-**Tree Recursion axiom**: The value of a balanced Boolean formula (a tree) with Σ_0^B functions for gates is well-defined, and defines a function encoded by a second order object.
The depth of the tree is given by a first-order object.

The proofs of edge expansion for G_i are based on combinatorial constructions, counting, and summations of series. VNC^1 can formalize all these arguments and can also prove the correctness of the NC^1 algorithm for the graphs G_i.

Main Theorem 2: The theory VNC^1 can prove the existence of the expander graphs G_i, as encoded by second-order objects, and can prove their expansion properties by using the constructions of Main Theorem 1, and the above Lemma.

More details on formalization in VNC^1.

First-order objects (integers, pairs of integers, etc.) encode small numbers, e.g., indices of vertices or edges.

Second-order objects encode sets, e.g., sets of vertices, or sets of edges (i.e., graphs).

$\textit{edge-exp}_G(U)$ is definable in VNC^1 using counting, which is known to be definable in VNC^1.

Theorem:

$$\mathrm{VNC}^1 \vdash \forall i\, \exists G{=}(V,E), |G|{=}i \;\&\; \forall U{\subset}V\; \textit{edge-exp}_G(U){>}1/1296.$$

Proof uses the "parameter-free Π_1^B-LLIND", a new logarithmic-length induction principle for Σ_1^B (NP) properties, which is justified by the "squaring" growth rate of the expander construction.

VNC^1 proves parameter-free Π_1^B-LLIND:

Theorem: Suppose $\theta(X)$ is a Σ_0^B-formula containing only X free. and let $\psi(a)$ be $(\exists X{\le}a)\theta(X)$. Also suppose VNC^1 proves

$$(\forall a)(\psi(a) \to \psi(\sqrt{a})). \tag{1}$$

Then VNC^1 proves $\psi(a) \to \psi(1)$, and thus also proves $\theta(Y) \to (\exists X{\le}\, 1)\theta(X))$.

Application: The hypothesis $(\forall a)(\psi(a) \to \psi(\sqrt{a}))$ will express a version of

$$(\exists U{\subset}G_i)\mathit{edge\text{-}exp}_U \le 1/1296 \to (\exists U{\subset}G_{i-1})\mathit{edge\text{-}exp}_U \le 1/1296.$$

The conclusion $\psi(a) \to \psi(1)$ will express a version of

$$(\forall U{\subset}G_i)\mathit{edge\text{-}exp}_U > 1/1296.$$

IV. Monotone propositional logic (sequent calculus)

Monotone Boolean Function: Let $0 < 1$, i.e. "False" $<$ "True". A Boolean function $f(\vec{x})$ is *monotone* provided that whenever $\vec{x} \le \vec{y}$, we have $f(\vec{x}) \le f(\vec{y})$.

Monotone Boolean Formula: A propositional formula over the basis $\land$ and $\lor$.

Sequent: $A_1, \ldots, A_k \longrightarrow B_1, \ldots, B_\ell$ means

$$A_1 \land A_2 \land \cdots \land A_k \to B_1 \lor B_2 \lor \cdots \lor B_\ell$$

Example: Pigeonhole principle tautologies: PHP_n

$$\bigwedge_{i=0}^{n} \bigvee_{j=0}^{n-1} x_{i,j} \quad \to \quad \bigvee_{0 \le i_1 < i_2 \le n} \bigvee_{j=0}^{n-1} (x_{i_1,j} \land x_{i_2,j}).$$

Def'n: The **propositional sequent calculus (LK)** is a propositional proof system whose proofs consist of sequents, with a finite set of valid inference forms, for example

$$\wedge\text{:right}\ \frac{\Gamma \longrightarrow \Delta, A \qquad \Gamma \longrightarrow \Delta, B}{\Gamma \longrightarrow \Delta, A \wedge B}$$

$$\text{Cut}\ \frac{\Gamma \longrightarrow \Delta, A \qquad A, \Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta}$$

Def'n: The **monotone sequent calculus (MLK)** is LK restricted to allow only monotone formulas to appear in sequents.

MLK proofs are allowed to be dag-like.

Theorem: [Atserias-Galesi-Galvalda'01; Aterias-Galesi-Pudlák'02] For monotone sequent tautologies, MLK quasipolynomially simulates LK.

Proof idea: Restrict to "slices" where a fixed number of inputs are true. Then simulate $\neg x$ using threshold formulas.
The properties of the threshold formulas must be proved; and the natural recursively-defined threshold formulas that admit such proofs are quasipolynomial size.

Theorem: [B '86] The PHP_n tautologies have polynomial size LK proofs.

Corollary: MLK has quasipolynomial size proofs of the pigeonhole tautologies PHP_n.

Theorem: [Jeřábek '11] If VNC^1 can prove the existence of expander graphs, then MLK polynomially simulates LK.

Proof idea: Working in a slightly stronger system VNC^1_*, the AKS sorting networks can be constructed from expander graphs, and their correctness proved. VNC^1_* corresponds to logspace uniform NC^1-computability, so the AKS sorting networks can serve as logspace uniform polynomial size threshold circuits. Thus, MLK polynomially simulates LK (logspace uniformly).

As a corollary:

Main Theorem 3: MLK polynomially simulates LK.

Corollary. (Example) MLK has polynomial size proofs of the PHP_n tautologies.

Corollary. Propositional LJ (intuitionistic logic) polynomially simulates LK w.r.t. monotone sequents. ([Jeřábek '09])

Open Questions

- Can expanders be formalized also in VTC^0, the system of bounded arithmetic corresponding to TC^0?
  "TC_0" $=$ "constant depth threshold circuits."
- Are there U_{E^*}-uniform sorting networks? Can this be done with a modification of the AKS construction with our NC^1-expanders?.
- Can tree-like MLK polynomially simulate MLK (equivalently, simulate LK on monotone sequents)?
- Can USTCON$\in$LogSpace [Reingold'08] be formalized in VL or VLV, systems of bounded arithmetic corresponding to LogSpace?

Thank You!