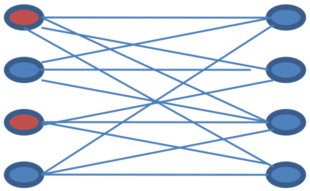
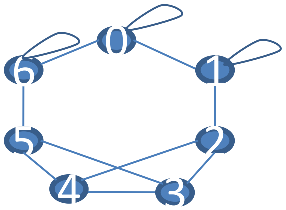


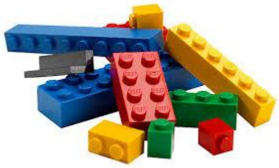


Power of reasoning over richer domains

Antonina Kolokolova (MUN)

NII Shonan meeting
“Logic and computational complexity”,
Sep 21, 2017

VNC^1



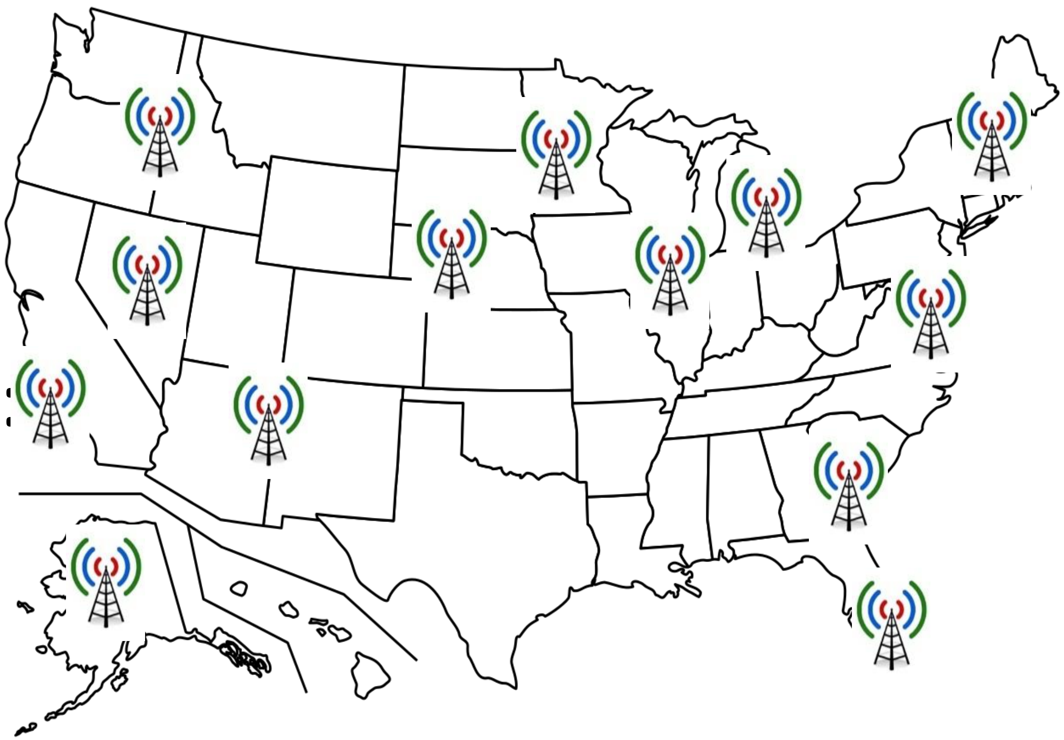
$$\begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{pmatrix}$$

Problem representations

Given a problem with its instance(s), how to state it to make it easier to solve?

Example: SAT vs. Integer LP

- FCC spectrum auction:
  - Essentially a colouring problem
  - ILP: poor
  - SAT: good



- TravelingSalesman:
  - ILP: good
  - SAT: poor

Problem representations

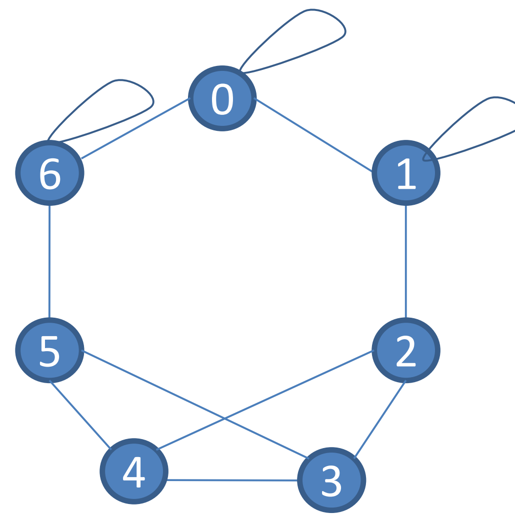
How to choose between propositional encoding, numerical encoding, their combination, something else?

Combinatorial vs. algebraic proofs

- Algebraic
  - Uses algebraic concepts
    - determinants, eigenvalues...
  - Relies on their properties for analysis

$$\begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{pmatrix}$$

- Combinatorial
  - Uses "simple to define" properties
  - Avoids algebra even in proofs
  - Algorithms of lower complexity!

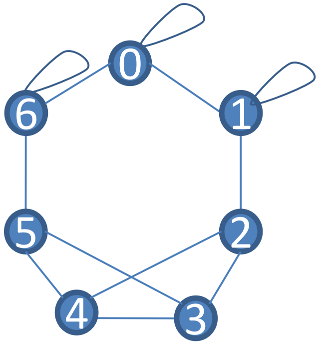


Problem representations

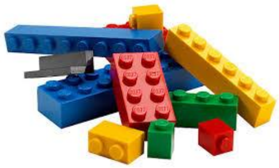
How to choose between algebraic
and combinatorial view,
e.g. of expander graphs?

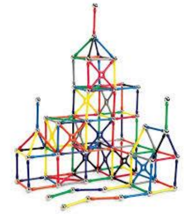
Some results

- Proof complexity of SMT:
  - Combining resolution with theories over underlying domain
    - Linear arithmetic, equality, uninterpreted functions with equality (EUF)...
    - Models satisfiability modulo theories solvers like resolution models SAT solvers
  - With EUF, can polynomially simulate Frege.



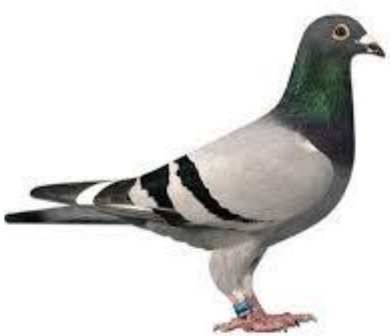
- Complexity of expander-based reasoning:
  - Can prove existence of expander graphs using purely combinatorial reasoning.
  - Corollary: monotone Frege is as powerful as non-monotone.

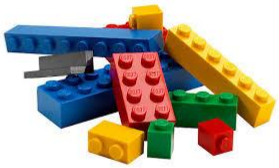





Proof complexity of Satisfiability Modulo Theories

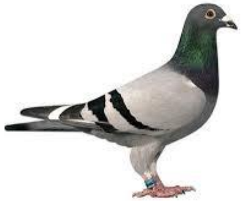
with Robert Robere and Vijay Ganesh





PigeonHolePrinciple

- PigeonHole Principle: there is no injective function from [n] to [n-1]



- PHP:

$$\wedge_{i \le n}(\vee_{j<n}\ p_{i,j}) \wedge \bigwedge_{i \neq k, j} (\neg\, p_{i,j} \vee \neg p_{k,j})$$

- =-PHP:

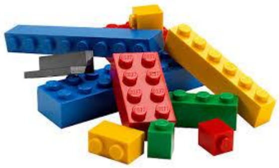
$$\bigwedge_{i \le \mathrm{n}} (\vee_{j<n} (p_i = h_j) \wedge \bigwedge_{i<k\le \mathrm{n}} (\, p_i \neq p_k)$$

- EUF-PHP:

$$\bigwedge_{x \in [n]} (\, f(x) \neq 0) \ \wedge \bigwedge_{x,y \in [n]} (\, x \neq y \rightarrow f(x) \neq f(y))$$

- LA-PHP:

$$\bigwedge_{i \le \mathrm{n}} (\Sigma_{j<n}\, x_{i,j} \ge 1) \ \wedge \bigwedge_{\substack{i,k \le n, \\ j<n}} (\, x_{i,j} + x_{k,j} \le 1)$$



PigeonHolePrinciple

- PigeonHole Principle: there is no injective function from [n] to [n-1]
- PHP:

$$\bigwedge_{i\le n}(\bigvee_{j<n}\ p_{i,j}) \wedge \bigwedge_{i\neq k,j}(\neg p_{i,j} \vee \neg p_{k,j})$$

- =-PHP:

$$\bigwedge_{i\le \mathrm{n}}(\bigvee_{j<n}(p_i = h_j)) \wedge \bigwedge_{i<k\le \mathrm{n}}(p_i \neq p_k)$$

- EUF-PHP:

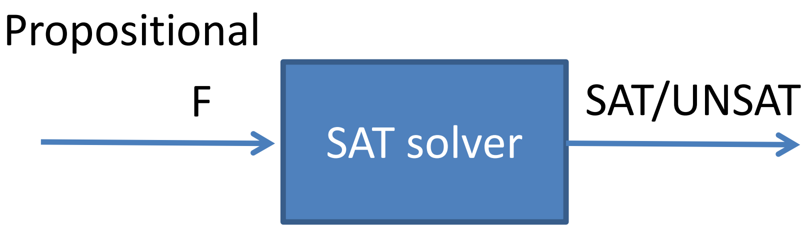
$$\bigwedge_{x\in[n]}(f(x)\neq 0) \ \wedge \bigwedge_{x,y\in[n]}(x\neq y \to f(x)\neq f(y))$$

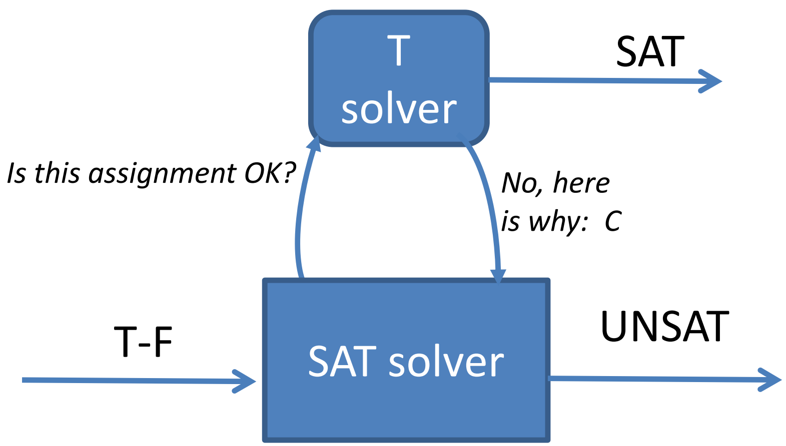
- LA-PHP:

$$\bigwedge_{i\le \mathrm{n}}(\Sigma_{j<n}\, x_{i,j} \ge 1) \ \wedge \bigwedge_{\substack{i,k\le n,\\ j<n}}(x_{i,j} + x_{k,j} \le 1)$$

- Propositional
- Theory of equality:
  - $(a = b \wedge b = c \to a = c)$
- Equality with uninterpreted functions (EUF)
  - equality axioms
  - Ackermann axioms: $(a = b \to f(a) = f(b))$
- Linear arithmetic

SAT vs. SMT





- PHP:

$$\bigwedge_{i\le n}\left(\bigvee_{j<n} p_{i,j}\right) \wedge \bigwedge_{i\neq k,j}(\neg p_{i,j} \vee \neg p_{k,j})$$

- =-PHP:

$$\bigwedge_{i\le n}(\bigvee_{j<n}\left(p_i = h_j\right) \wedge \bigwedge_{i<k\le n}(p_i \neq p_k)$$

- EUF-PHP:

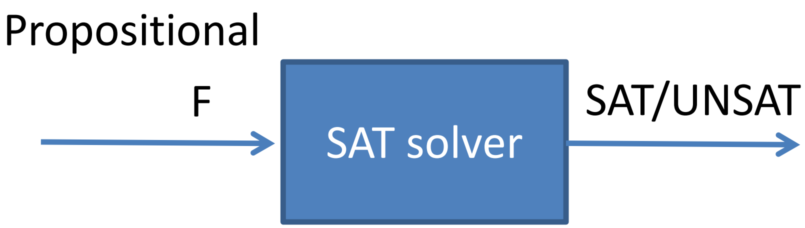
$$\bigwedge_{x\in[n]}(f(x)\neq 0) \wedge \bigwedge_{x,y\in[n]}(x \neq y \to f(x) \neq f(y))$$

- LA-PHP:

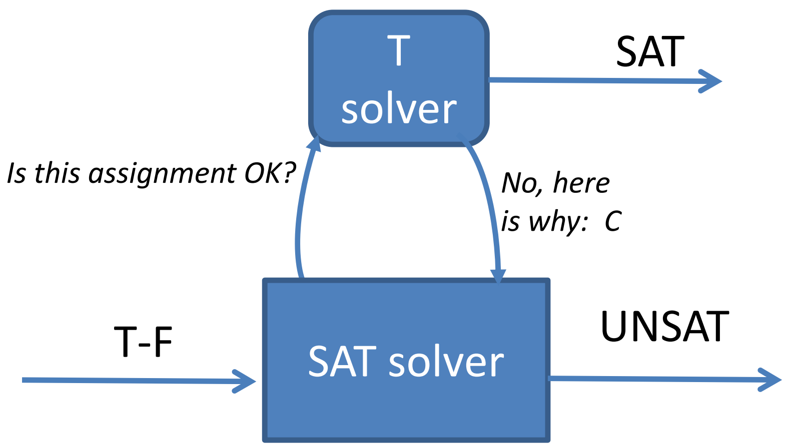
$$\bigwedge_{i\le n}(\Sigma_{j<n} x_{i,j} \ge 1) \wedge \bigwedge_{\substack{i,k\le n,\\ j<n}}(x_{i,j} + x_{k,j} \le 1)$$

For which theory T
can a SAT solver
with a T solver
simulate Extended Frege?

Res(T)



Resolution



Res(T)

Res(T):
Literals are atoms of the theory.
Rules of inference:

1. Resolution rule

$$\frac{(C \vee x) \quad (D \vee \neg x)}{(C \vee D)}$$

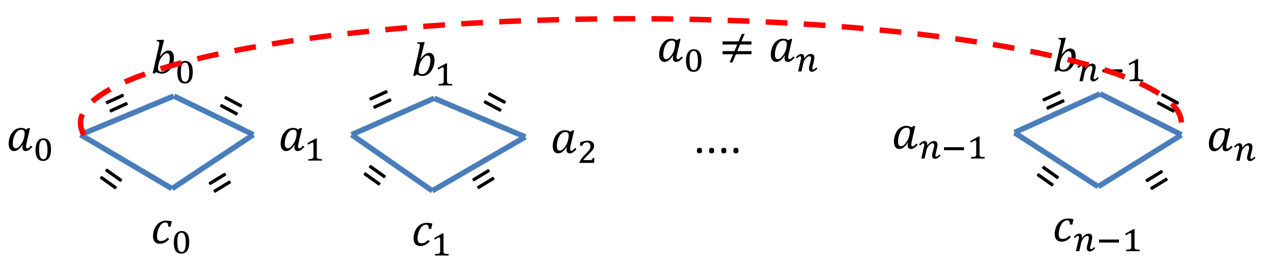
2. Clauses derivable from T:

- Eg: T is a theory of equality:
  - $(a \neq b \vee b \neq c \vee a = c)$
- Eg: T is linear arithmetic:
  - $(a \leq c \ \vee \ b \leq d \vee a + b > c + d)$

Can introduce new literals

New literals

- Theory solver has to be able to return a clause using literals not in the original formula:
  - if F contained a=b and b=c, T returns a clause $(a \neq b \vee b \neq c \vee$



Res(T) vs. SMT solvers

- CDCL (conflict-driven clause learning with restarts)
  - Repeat:
    - Assign some variables
    - Do unit clause propagation (set literals in unit clauses)
    - If there is an unsatisfied clause, backtrack and learn the conflict as a clause
    - Maybe restart, removing variable assignment, but keeping learned clauses

- CDCL(T):
  - Also check whether assignment makes sense for T
  - If not, learn a conflict clause.

- Resolution captures CDCL
  - Pipatsrisawat/Darwiche’11.

- Res(T) captures CDCL(T)
  - Generalizing Pipatsrisawat/Darwiche’11.

Power of Res(T)

- Res(Theory of Equality) is no more powerful than Resolution
  - Add all n^3 equality axioms to F, then solve.
- Res(LA) polynomially simulates R(lin)
- Resolution over Equality with Uninterpreted Functions theory, Res(EUF), can effectively p-simulate Frege.
  - Conjunctions of EUF atoms are decidable in $O(n \log n)$ time!
  - Using a variant of Union-Find algorithm.

Equality with uninterpreted functions theory (EUF)

- Signature:
  - uninterpreted function symbols of bounded arity
  - constants a, b, c...
- Terms: constants, and inductively $f(\bar{t})$ for functions.
- Atoms: equalities/disequalities over terms: $t_1 = t_2,\ t_1 \neq t_2$
- Formulas: conjunctions of atoms

$$(f(a) = b) \wedge (b = c) \wedge (g\big(f(a)\big) \neq c)$$

- Axioms:
  - Equality: $(a = b \wedge b = c \rightarrow a = c)$
  - Ackermann: $\bar{a} = \bar{b} \ \rightarrow\ f(\bar{a}) = f(\bar{b})$
- Can decide in near-linear time if a given EUF formula is satisfiable:
  - Downey-Sethi-Tarjan congruence closure (based on Union-Find)

Sequent calculus (LK)

- Equivalent to Frege.
  - Natural deduction

- Sequents: $A_1, \ldots, A_n \quad \longrightarrow \quad B_1, \ldots, B_m$
  - $A_1 \wedge \cdots \wedge A_n \rightarrow B_1 \vee \cdots \vee B_m$
  - Axioms $A \rightarrow A,\ 0 \rightarrow S,\ S \rightarrow 1.$
  - Rules for $\vee, \wedge, \neg$ and cut

$$\frac{F \rightarrow G, A \qquad A, F \rightarrow G}{F \rightarrow G}$$

$$\frac{F \rightarrow G, A}{\neg A, F \rightarrow G}$$

$$\frac{F \rightarrow G, A \qquad F \rightarrow G, B}{F \rightarrow G, A \wedge B}$$

$$\frac{A, B, F \rightarrow G}{A \wedge B, F \rightarrow G}$$

- Proof size: total number of symbols.

Res(EUF) simulates LK

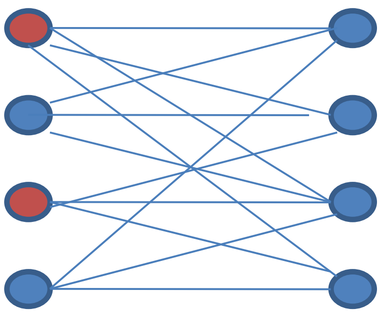
- Suppose there is an LK proof of $F \to 0$
  - An LK-refutation of F
- Add to F:
  - Two constants: $e_0 \neq e_1$
  - Definitions of N, O, A (and, or, not):
    - $N(e_0) = e_1, N(e_1) = e_0, O(e_1, e_0) = e_1, \ldots.$
  - Bounded variable range: $\wedge(x_i = e_0 \vee x_i = e_1)$

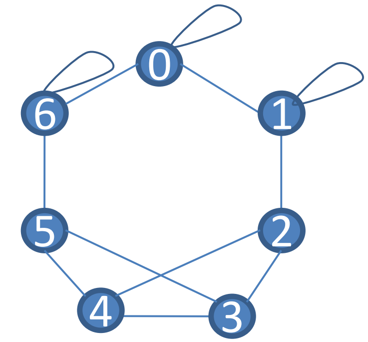
- Now simulate an LK proof by constructing terms for all formulas in the proof inductively
  - Prove that at each step of LK proof: $A_1 \ldots A_k \to B_1 \ldots B_\ell$
  - Either one of the A terms is e_0 or one of the B terms is e_1
    - Also for each subformula in proof so far, its term $= e_0$ or $= e_1$

Open problems

- Is it better to use SMT than propositionalize completely? If so, when?
  - Flatten:
    - replace nested terms by new variables
  - Bit blast:
    - represent each variable by $\log n$ bits.
  - add all relevant axioms explicitly.
- How to choose T given a problem and class of instances?
  - And how to choose T-representation?

For which theory T would Res(T)
effectively p-simulate Extended Frege?





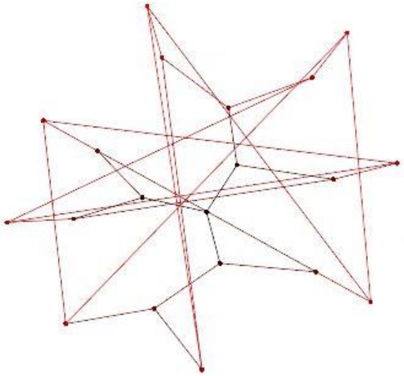
Complexity of Expander-Based Reasoning and the Power of Monotone Proofs

with Sam Buss, Valentine Kabanets
and Michal Koucky

VNC^1

$$\begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{pmatrix}$$

Expander graphs



- Graphs which are both
  - sparse (usually constant degree)
  - and well connected (log length path between any two points.

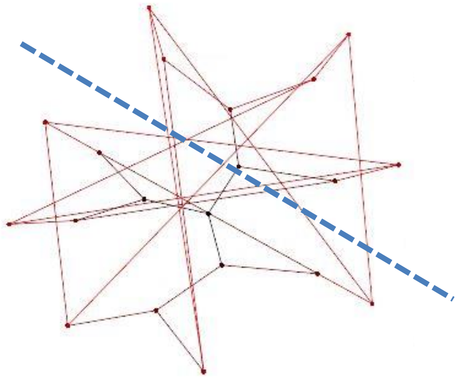
- Expander graphs are pseudorandom objects. A random graph is an expander with high probability.

- Random walk on an expander converges fast.

Uses of expanders

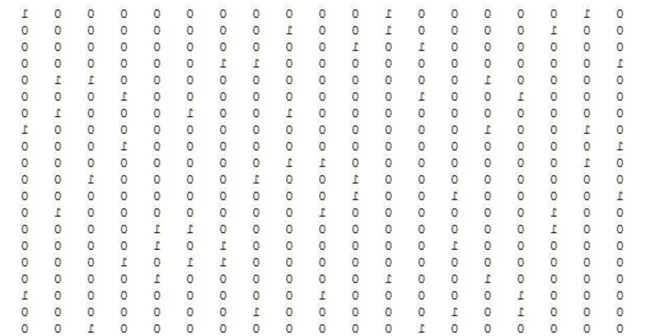
- As pseudorandom objects
  - One-way functions of Goldreich'2000
  - Cryptographic hash functions: Charles/Goren/Lauter...
  - Error-correcting codes, derandomization...
- In complexity theory
  - Reingold and Rozenman/Vadhan: USTCON in LogSpace
  - Dinur: combinatorial proof of the PCP theorem
  - Ajtai/Komlos/Szemeredi: AKS sorting networks

Combinatorial definition of expanders



- d-regular undirected (multi)graphs
- Edge expansion:
  - min fraction of edges crossing a cut (normalized by smaller side size).
  - $h(G) = \min_{\left\{\emptyset \neq U, |U| \leq \frac{n}{2}\right\}} \frac{|E(U,U^c)|}{|U|}$
- Expander: h(G) is constant.

Algebraic definition of expanders



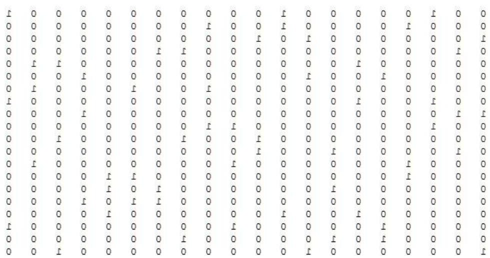
- Spectral gap: d-λ_2,
  - d is the degree of G
  - λ_2 is the second largest eigenvalue of adjacency matrix M_G of G.

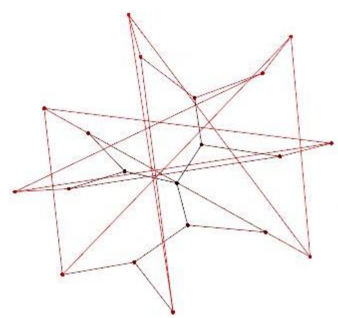
- Expander (λ-expander):
  - A graph that has a constant spectral gap.

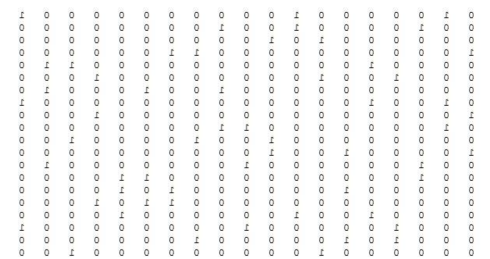
Combinatorial vs. algebraic

- Cheeger inequality:

$$\frac{d-\lambda_2}{2} \leq h(G) \leq \sqrt{2d(d-\lambda_2)}$$







  - So constant spectral gap ~ constant expansion
  - Most proofs use algebraic definition
    - Some loss in parameters in combinatorial setting
  - Combinatorial definition allows lower complexity algorithms

Formalizing "combinatorial"

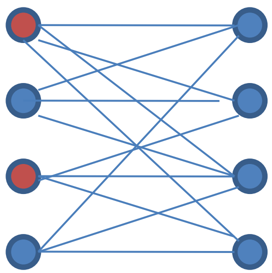
- Take a system of reasoning which cannot define algebraic objects
  - No eigenvalues, determinants, etc
  - E.g., a system based on polynomial-size formulas (NC^1-reasoning)

- Proofs in this system are combinatorial (unless algebra $\in NC^1$)
  - Combinatorial proofs of correctness of algorithms or existence of combinatorial objects.
  - Not known to prove AB=I => BA=I

Our results

- We give an NC^1 proof of existence of expander graphs of arbitrary size.
  - Includes a combinatorial analysis of a fully explicit expander construction.
  - And its formalization in an NC^1 theory
- Corollary: monotone proofs are as powerful as non-monotone.
  - Monotone LK polynomially simulates LK.
  - By adding the last piece to [Atserias-Galesi-Gavalda'01, Atserias-Galesi-Pudlak'02, Jerabek'11]

Expander constructions

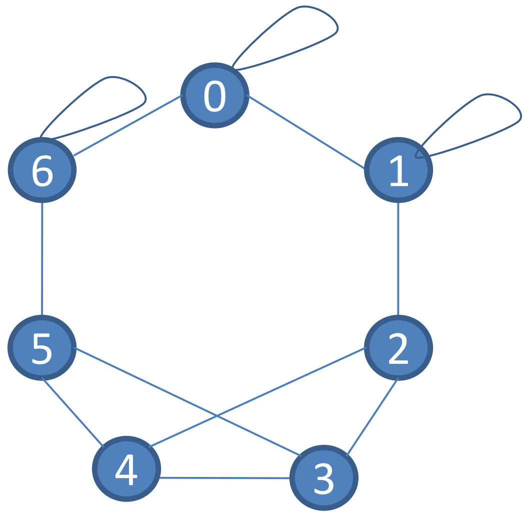


Example 1: Margulis, Gabber/Galil bipartite expanders.

(x,y) -> (x,y), (x,x+y), (x,x+y+1), (x+y,y), (x+y+1,y)

Example 2: (from Hoory/Linial/Wigderson)

Z_p: for every $v \neq 0$, connect
v to v-1, v+1 and v^{-1}.
For v=0, connect v to 0,1 and p-1.



Iterative constructions

- Start with constant-size expanders. Obtain a large size graph by repeatedly applying:
  - Powering (to increase expansion)
  - Zig-zag or replacement product (to reduce degree)
  - Tensoring (to grow quickly).

- Originally by [Reingold-Vadhan-Wigderson'02]
  - Zig-zag product. Proof uses spectral gap.
  - [Alon, Schwartz, Shapira'08] Replacement product with its combinatorial analysis.

- Explicit: given vertex v and $i \in [1 \ldots d]$, produce (w, j) such that w is the i^{th} neighbour of v, and v is the j^{th} neighbour of w in resulting graph.
  - In time O(log |G|).

Our variant of the construction

- Start with 2d-regular G_0 with $\mathrm{h}(\mathrm{G}_0) = \epsilon = 1/1296$ and d-regular H, $h(H) = 1/3$.

- Apply the following $\sim\log n$ times:
  1. Add self-loops to double the degree; tensor with itself
  2. Add self-loops again and power to a constant c
  3. Replace each vertex with H.
- Each G_i has $h(G_i) = \epsilon$ and size $>$ squared.

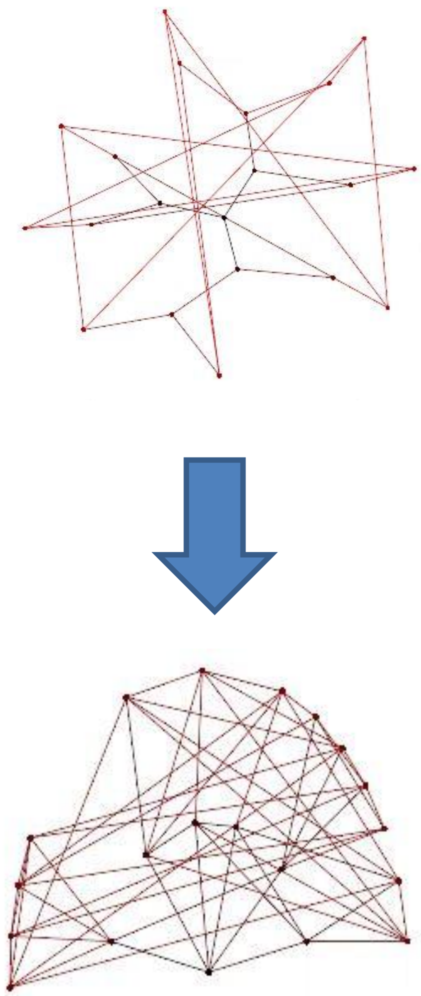
- Fully explicit: NC^1 algorithm to compute k^{th} neighbour w of v in the final G, and its edge index j from (v,k)

Powering: $M_{G'} = \mathrm{M}_{\mathrm{G}}^{\mathrm{k}}$

- Easy with eigenvalues: $\lambda_2 \rightarrow \lambda_2^k$

- Combinatorially, let h(G)=ϵ.
  - First, add d self-loops to G.
  - Using [Mihail'89] mixing lemma
    - and mixing -> expansion
  - Get $h(G') = \frac{1}{2}(1-\left(1-\frac{\epsilon^2}{4}\right))^{k/2}$

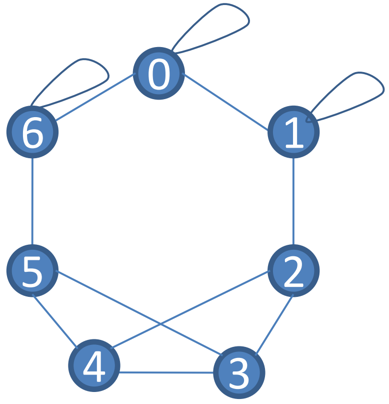
Proof : Cauchy-Schwartz and sums.



Mihail'89 mixing lemma

- A random walk on an expander converges to uniform distribution exponentially fast.

- *More precisely, let*
  - *G be a d-regular graph with edge expansion* ϵ.
    - *Add d self-loops to each vertex of G to obtain G'*
  - *A be a normalized adjacency matrix of G'*
  - *π be any distribution on vertices of G'*
    - *u the uniform distribution on vertices of G'*



- Then $\left|\left|A^k\pi - u\right|\right|^2 \leq \left(1 - \left(\frac{\epsilon^2}{4}\right)\right)^k \left|\left|\pi - u\right|\right|^2$

Constructiveness

- For formalization, need an NC^1 algorithm:
  - Given a non-expanding set U' in G'
  - Produce a non-expanding set U in G.
- From [Mihail'89] proof:
  - Sort vertices in decreasing $\pi - u$ order
  - If some U' in G' is non-expanding, then so is a set of first k vertices in G for some k. Test which one.
    - Both sorting and testing are in NC^1

Formalizing “combinatorial”

- Bounded arithmetic:
  - Theories ~ complexity classes.
  - For a class C, a theory V-C can reason about C-definable concepts (numbers and strings)

- Eigenvalues, determinants, etc are not known to be computable in NC^1

- So proofs in V-NC^1 are “combinatorial” in a strict sense.
  - If V-NC^1 cannot prove existence of eigenvalues
  - Then it cannot formalize proofs relying on eigenvalues, even in disguise.

Bounded arithmetic theories

- V^0 : first-order reasoning
- VTC^0: V^0 + “exists numones(y,X)=z”
- VNC^1: V^0 + “exists an evaluation of a Boolean formula”
  - Not known to prove AB=I -> BA = I
  - Uniform version of Frege/Sequent Calculus LK
- $V^1 \approx S_2^1$…

Formalizations

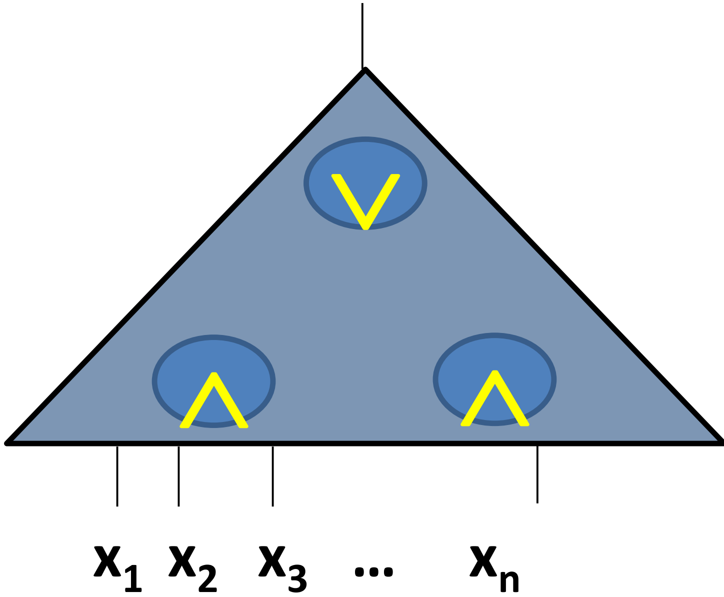
- Approximate counting, randomized computation, PRGs (Jerabek), PCPs (Pich), Toda's theorem in higher complexity theories.

- Assuming existence of expanders, correctness of AKS sorting networks is provable in a (slightly non-uniform version of) VNC^1 (Jerabek)

- Our result: Theory VNC^1 proves existence of expanders of arbitrary size.
  - Thus, NC^1 reasoning is enough to prove correctness of AKS sorting networks.

Complexity in monotone

- Monotone functions:
  - $\forall x, y,\ \ x \subseteq y \Rightarrow f(x) \leq f(y)$
  - Majority, Threshold, Clique...
- Monotone circuits:
  - AND, OR gates.
  - $\text{Clique}_{k,n}$ requires monotone circuits of size $\geq 2^{\epsilon\sqrt{k}}$ for some ϵ.



Monotone proof complexity?

Monotone sequent calculus (MLK)

- Monotone version of LK [Buss-Pudlak'95]

- Sequents: $A_1, \ldots, A_n \longrightarrow B_1, \ldots, B_m$
  - all A_i, B_j are formulas over ∧,∨.
  - Axioms $A \rightarrow A, 0 \rightarrow S,\ S \rightarrow 1.$
  -
  - Rules for ∨,∧ and cut
    - No rule for ¬

$$\frac{F \rightarrow G, A \qquad F \rightarrow G, B}{F \rightarrow G, A \wedge B} \qquad \frac{A, B, F \rightarrow G}{A \wedge B, F \rightarrow G} \qquad \frac{F \rightarrow G, A \qquad A, F \rightarrow G}{F \rightarrow G}$$

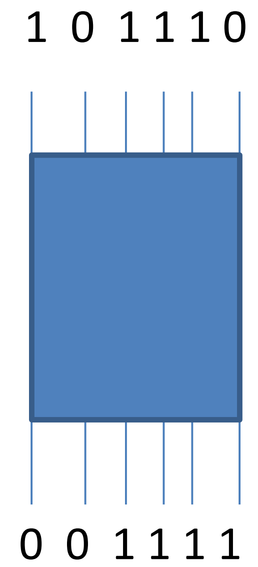
- Non-uniform version of VNC^1
- Polynomial-size proofs of PHP

MLK polynomially simulates LK

- [Atserias-Galesi-Gavalda'01, Atserias-Galesi-Pudlak'02]:
  - Simulate $\neg x$ using threshold formulas:
    - if k 1s in the input, and still k 1s with x_i replaced by 0, then x_i = 0
    - Slice functions idea.
  - Recursive definition of thresholds gives quasipolysize proofs.

  - Monotone NC^1 threshold functions?
    - AKS sorting networks

AKS sorting networks



- Sorting network:
  - n inputs, n outputs (Boolean)
  - Outputs input bits in sorted order

- [Ajtai-Komlos-Szemeredi'83]
  - Monotone log-depth sorting networks
  - Based on expanders

AKS sorting networks

- [Jerabek'11] Properties of AKS sorting networks are in (slightly non-uniform) VNC^1. So
  - if VNC^1 proves that expanders exist,
  - get polysize proofs for properties of thresholds
  - and polynomial simulation of LK by MLK.

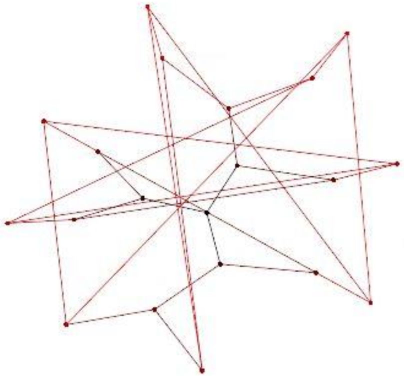
- Here: VNC^1 proves that expanders exist.

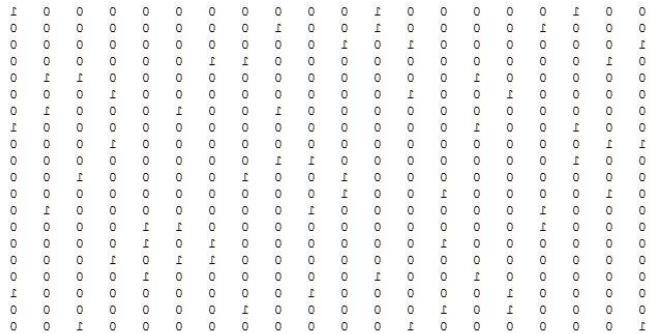
Open problems

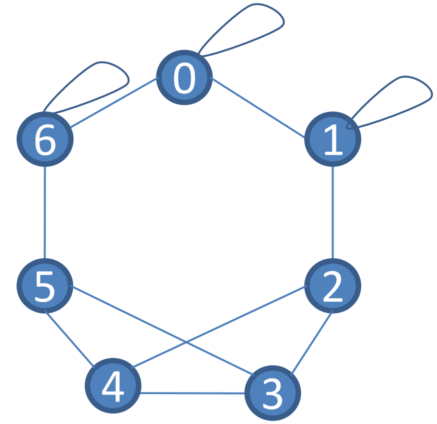
- Can existence of expanders be proven in VTC^0?

- Complexity of USTCONN $\in$ L ?
  - Our analysis needs both initial graphs to be expanders.

- Proof complexity of other results that now rely on algebra?

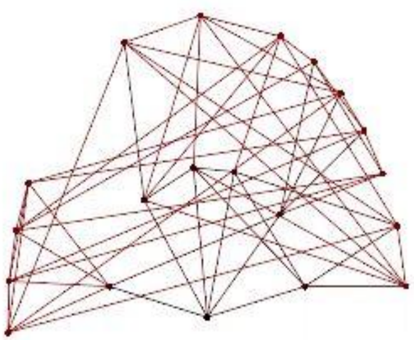






$$\begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{pmatrix}$$

Thank you!



VNC^1