Space Complexity of Polynomial Calculus

Massimo Lauria

Sapienza – Università di Roma

LOGICAL APPROACHES TO BARRIERS IN COMPLEXITY II
CAMBRIDGE 2012

(joint work with Y. Filmus, J. Nordström, N.Thapen and N. Zewi)

“SAT has been solved”

—the daring practitioner—

“SAT is clearly hard”

—the savvy theorist—

“SAT solving can get better. . .”

—the relentless coder—

Next step is to study **memory requirements** of **algebraic** sat-solvers/theorem provers.

Modern SAT solvers are based on **Resolution**

but

algebraic reasoning can be beneficial.

- shorter proofs

- shorter proofs
- simple proof structure (e.g. Polynomial Calculus)

- shorter proofs
- simple proof structure (e.g. Polynomial Calculus)
- proof search (e.g. Buchberger)

- shorter proofs
- simple proof structure (e.g. Polynomial Calculus)
- proof search (e.g. Buchberger)
- implementations (e.g PolyBori)

. . . NOT ENOUGH TO SWITCH

. . . NOT ENOUGH TO SWITCH

- modern solvers are **very** optimized

. . . NOT ENOUGH TO SWITCH

- modern solvers are **very** optimized
- to change paradigm is bad for SAT races

. . . NOT ENOUGH TO SWITCH

- modern solvers are **very** optimized
- to change paradigm is bad for SAT races
- their main issue is **space**, not proof length

SPACE IN MODERN SAT-SOLVERS

- during a running, solvers **learn** clauses
- memory fills very quickly
- retaining the right clauses if fundamental

DOES ALGEBRA HELP?

- memory requirements
- relation between space and time

Our results

	width	variables	space	
PCR	n	n^2	$\Omega(n)$	**[Alekhnovich et al. 02]**
PC	3	n^2	$\Theta(n)$	pigeonhole principle
PC	$O(1)$	n	$O(n)$	any formula
PCR	$2\log n$	$n\log n$	$\Omega(n)$	bit-pigeonhole principle
PCR	4	n^2	$\Omega(n)$	xor-pigeonhole principle

Our results

	width	variables	space	
PCR	n	n^2	$\Omega(n)$	**[Alekhnovich et al. 02]**
PC	3	n^2	$\Theta(n)$	pigeonhole principle
PC	$O(1)$	n	$O(n)$	any formula
PCR	$2\log n$	$n\log n$	$\Omega(n)$	bit-pigeonhole principle
PCR	4	n^2	$\Omega(n)$	xor-pigeonhole principle

In this talk we sketch this proof!

Outline

1. Algebraic proof system PCR
2. Model space complexity in PCR refutations
3. We sketch the proof of a space lower bound for PCR

ALGEBRAIC PROOF SYSTEM

Proof Systems

Deterministic polynomial time $P(\cdot, \cdot)$

- if $F \in$ UNSAT then $P(F, \pi) = 1$ for some $\pi \in \{0,1\}^*$
- if $F \notin$ UNSAT then $P(F, \pi) = 0$ for all $\pi \in \{0,1\}^*$

Proof systems in complexity theory

There is P where any unsat formula has a "short" refutation in P

$$\Longleftrightarrow$$

NP=coNP

Cook-Reckhow program (1979)

Prove proof length lower bound for stronger and stronger system in order to prove NP$\neq$coNP

PROOF SYSTEMS AND SAT SOLVERS

The trace of "SAT-solver(F)=`unsat`" is a refutation for F.

DLL	$\longrightarrow$	tree-like resolution
Clause Learning	$\longrightarrow$	regular WRTL [BHJ '08]
CL + Restarts	$\longrightarrow$	resolution
CRYPTOMINISAT	$\longrightarrow$	fragments of PCR on $GF(2)$
POLYBORI	$\longrightarrow$	PC on $GF(2)$

Polynomial Calculus (PCR)

CNF formula	$\longrightarrow$	set of polynomials
SAT assignments	$\longrightarrow$	common roots
true	$\longrightarrow$	0
false	$\longrightarrow$	1
variable x	$\longrightarrow$	$x, \bar{x}$
$x \in \{\text{true}, \text{false}\}$	$\longrightarrow$	$x^2 - x$ $x + \bar{x} - 1$
$x \vee \neg y \vee \neg z \vee s \vee t$	$\longrightarrow$	$x \cdot \bar{y} \cdot \bar{z} \cdot s \cdot t$

Polynomial Calculus (PCR)

Linear combination

$$\frac{p \qquad q}{\alpha p + \beta q}$$

Multiplication

$$\frac{p}{xp}$$

$\boxed{F \vdash 1}$ iff $\boxed{F \in \text{UNSAT}}$

(SOUNDNESS) INFERENCE PRESERVES COMMON ROOTS

(COMPLETENESS) SIMULATES DECISION TREES

- PC defined in [CEI96] and PCR in [ABRW02];
- PCR strictly better than resolution in proof length;
- Size-Degree Trade-off [IPS99,GL10a];
- Exponential lower bounds on length are known [Raz98,AR03, BGIP01, BI10, IPS99, Raz98];
- Proof search is hard [GL10b] based on [AR08].

SPACE COMPLEXITY OF PCR

$$\cdots \to \begin{bmatrix} xz + yz \\ xz - 1 \\ \\ \end{bmatrix}$$

$$\cdots \rightarrow \begin{bmatrix} xz + yz \\ xz - 1 \\ \end{bmatrix} \rightarrow \begin{bmatrix} xz + yz \\ xz - 1 \\ 1 - yz \end{bmatrix}$$

- inference step from polynomials in memory

$$\cdots \rightarrow \begin{bmatrix} xz+yz \\ xz-1 \\ \end{bmatrix} \rightarrow \begin{bmatrix} xz+yz \\ xz-1 \\ 1-yz \end{bmatrix} \rightarrow \begin{bmatrix} xz+yz \\ \text{—} \\ 1-yz \end{bmatrix}$$

- inference step from polynomials in memory
- erasure of a polynomial

$$\cdots \rightarrow \begin{bmatrix} xz+yz \\ xz-1 \\ \end{bmatrix} \rightarrow \begin{bmatrix} xz+yz \\ xz-1 \\ 1-yz \end{bmatrix} \rightarrow \begin{bmatrix} xz+yz \\ \\ 1-yz \end{bmatrix} \rightarrow \begin{bmatrix} xz+yz \\ x^2-x \\ 1-yz \end{bmatrix} \cdots$$

- inference step from polynomials in memory
- erasure of a polynomial
- logical axiom/initial polynomial download

Space measure: #monomials in a configuration

Space measure: #monomials in a configuration

$$\begin{bmatrix} xz & + & yz \\ xz & - & 1 \\ 1 & - & yz \end{bmatrix}$$

(this configuration counts as space six)

Space measure: #monomials in a configuration

$$\begin{bmatrix} xz & + & yz \\ xz & - & 1 \\ 1 & - & yz \end{bmatrix}$$

(this configuration counts as space six)

Roads not taken

- $O(1)$ polynomials are always sufficient (#polynomials)
- too expensive compared to implementations (#symbols)

LITTLE IS KNOWN FOR PCR SPACE

- Lower bounds for wide CNFs [Alekhnovich et al. 2002]
- Length-Space trade-offs [Huynh, Nordström, 2012]
- Lower bounds for narrow CNFs [FLNTZ 2012]

Lower bound for narrow CNFs

BIT-PIGEONHOLE PRINCIPLE

Fix $n = 2^m$: there is no injective function $F : [n+1] \to \{0,1\}^m$.

BIT-PIGEONHOLE PRINCIPLE

Fix $n = 2^m$: there is no injective function $F : [n+1] \to \{0,1\}^m$.

For each:

- two pigeons a and b
- hole $s \in \{0,1\}^m$

$$\underbrace{(f_{a,1} \neq s_1) \vee \cdots \vee (f_{a,m} \neq s_m)}_{F(a) \neq s} \bigvee \underbrace{(f_{b,1} \neq s_1) \vee \cdots \vee (f_{b,m} \neq s_m)}_{F(b) \neq s}$$

EXAMPLE

$$F(1) \neq \langle 1101 \rangle \text{ or } F(3) \neq \langle 1101 \rangle$$

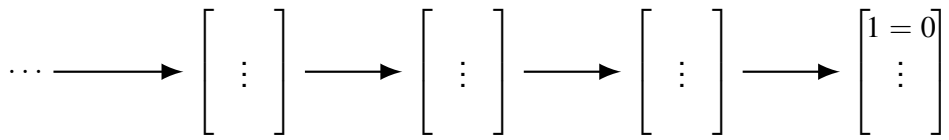
translates to

$$(\neg f_{1,1} \vee \neg f_{1,2} \vee f_{1,3} \vee \neg f_{1,4}) \bigvee (\neg f_{3,1} \vee \neg f_{3,2} \vee f_{3,3} \vee \neg f_{3,4})$$

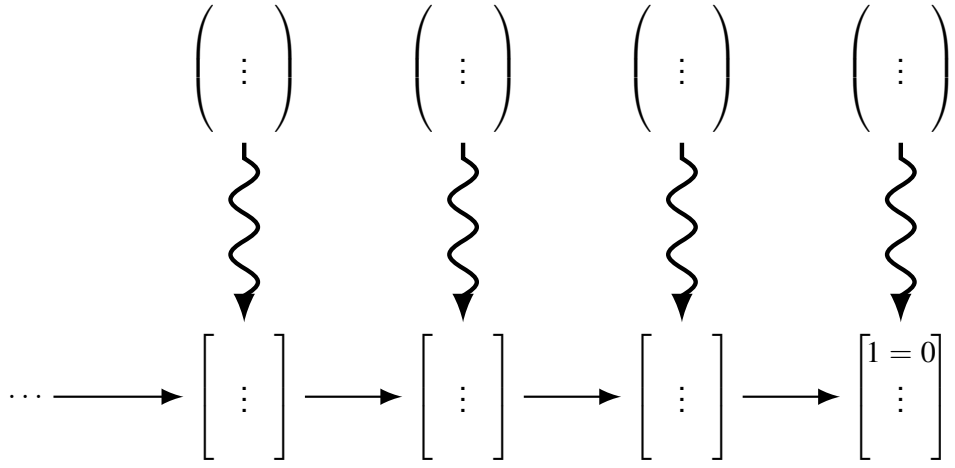
THEOREM

Any PCR refutation of the “Bit” pigeohole principle has a configuration with at least $n/8$ monomials.

Proof Sketch

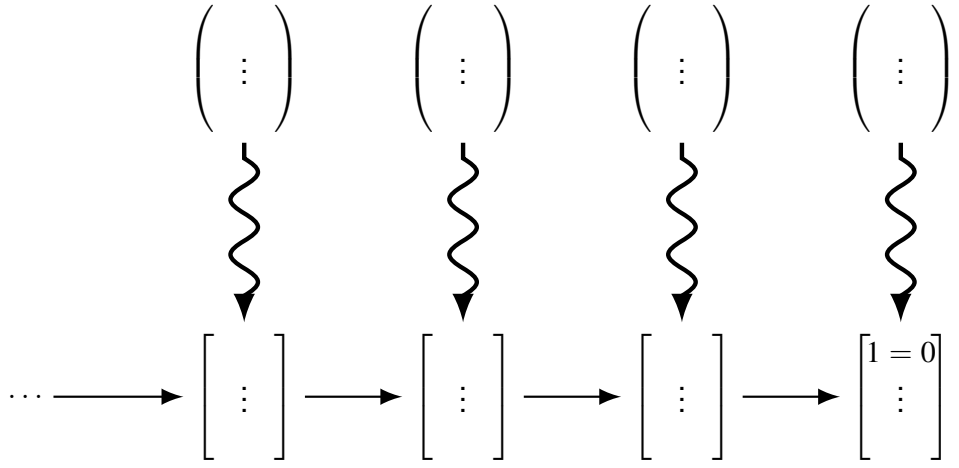


Proof Sketch



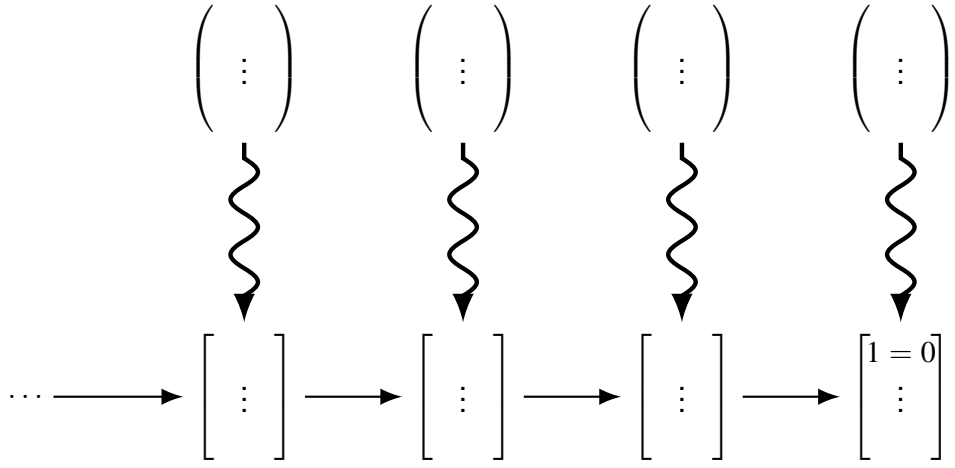
1. a parallel sequence of $(\ldots)$ such that $(\ldots) \rightsquigarrow [\ldots]$;

Proof Sketch



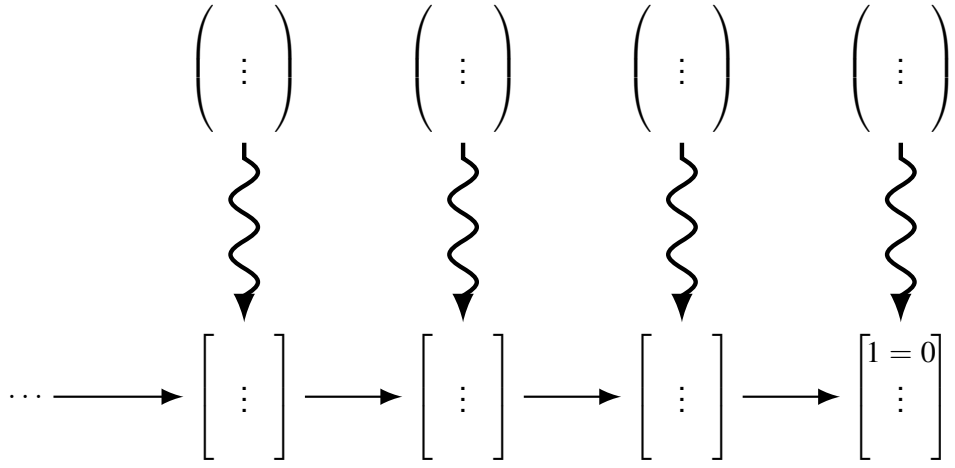
1. a parallel sequence of $(\ldots)$ such that $(\ldots) \rightsquigarrow [\ldots]$;
2. size of $(\ldots)$ if at most twice the size of $[\ldots]$; **(assuming monomial space $\leq n/8$:)**

Proof Sketch



1. a parallel sequence of $(\ldots)$ such that $(\ldots) \rightsquigarrow [\ldots]$;
2. size of $(\ldots)$ if at most twice the size of $[\ldots]$; **(assuming monomial space $\leq n/8$:)**
3. $(\ldots)$ of size $\leq n/4$ are all satisfiable;

Proof Sketch



1. a parallel sequence of $(\ldots)$ such that $(\ldots) \rightsquigarrow [\ldots]$;
2. size of $(\ldots)$ if at most twice the size of $[\ldots]$; **(assuming monomial space $\leq n/8$:)**
3. $(\ldots)$ of size $\leq n/4$ are all satisfiable;
4. contradiction since $(\ldots) \rightsquigarrow [1 = 0]$

SPECIAL CONFIGURATIONS

(2-CNFs where no pigeon is mentioned twice)

$$\begin{pmatrix} \neg f_{1,3} & \vee & f_{4,2} \\ f_{5,1} & \vee & f_{7,3} \\ & \vdots & \\ \neg f_{6,5} & \vee & \neg f_{2,3} \end{pmatrix}$$

A satisfying assignment:

- satisfies the 2-CNFs
- no collision on the **occurring** pigeons

Observation

Any special configuration with at most $n/4$ clauses is satisfiable.

Proof.

$$\begin{pmatrix} \neg f_{1,3} & \vee & f_{4,2} \\ f_{5,1} & \vee & f_{7,3} \\ & \vdots & \\ \neg f_{6,5} & \vee & \neg f_{2,3} \end{pmatrix}$$

1. at most $n/2$ pigeons;
2. at least $n/2+1$ free holes per pigeon;
3. at least one free hole per satisfied literal.

□

Input: $(M_0, M_1, \ldots, M_l)$ with $|M_i| \leq n/8$

Output: $(S_0, S_1, \ldots, S_l)$ such that $|S_i| \leq 2|M_i|$ and S_i implies M_i

Input: $(M_0, M_1, \ldots, M_l)$ with $|M_i| \leq n/8$

Output: $(S_0, S_1, \ldots, S_l)$ such that $|S_i| \leq 2|M_i|$ and S_i implies M_i

$S_0 := \emptyset$ [Initial configuration]

$S_{i+1} := S_i$ [Inference]

$S_{i+1} := S_i$ [Logical axioms]

Input: $(M_0, M_1, \ldots, M_l)$ with $|M_i| \leq n/8$

Output: $(S_0, S_1, \ldots, S_l)$ such that $|S_i| \leq 2|M_i|$ and S_i implies M_i

$S_0 := \emptyset$ [Initial configuration]

$S_{i+1} := S_i$ [Inference]

$S_{i+1} := S_i$ [Logical axioms]

[Download of $F(a) \neq s \vee F(b) \neq s$]

$S_{i+1} := S_i$ $[a, b \in S_i]$

$S_{i+1} := S_i \cup \{f_{a,1} \vee f_{b,1}\}$ $[a, b \notin S_i]$

$S_{i+1} := S_i \cup \{f_{a,1} \vee f_{c,1}\}$ for some $c \notin S_i$ $[a \notin S_i, b \in S_i]$

[Erasure step] is the hard case

How may clauses in 2-CNF influence the value of a monomial?

≤ 2 space complexity preserved;

> 2 weak influence, most clauses can be removed.

Proof recap

Assuming monomial space $\leq n/8$:

- a corresponding sequence of small 2-CNFs;
- all such small 2-CNFs are “satisfiable”;
- last memory configuration is satisfiable. **(contradiction)**

WRAPPING UP

- PCR is a candidate model of future SAT solvers;
- we need to study space to discover if it is the case;
- we have sketched a space lower bounds for $2 \log n$-CNFs.

Things I want you to work on

THINGS I WANT YOU TO WORK ON

- A linear PCR space lower bounds for (random) 3-CNFs
- space bounds for other proof systems (e.g. cutting planes)
- trading space for time

THINGS I WANT YOU TO WORK ON

- A linear PCR space lower bounds for (random) 3-CNFs
- space bounds for other proof systems (e.g. cutting planes)
- trading space for time

- theoretical space bounds vs memory usage
- improve PCR implementations

Thank you

Y. Filmus

J.Nordström

N. Taphen

N. Zewi