Resolution and the binary encoding of combinatorial principles

Nicola Galesi

Sapienza, Rome

BIRS Workshop: Proof Complexity — January 21, 2020

Joint work with: Stefan Dantchev and Barnaby Martin

Resolution over s-DNF

A 2-DNF: $((v_1 \wedge \neg v_2) \vee (v_2 \wedge v_3) \vee (\neg v_1 \wedge v_3))$

	Resolution (= Res(1))	Res(2)
Main Rule	$\frac{C \vee x \qquad \neg x \vee D}{C \vee D}$	$\frac{C \vee (x \wedge y) \qquad (\neg x \vee \neg y) \vee D}{C \vee D}$
Refutations for	CNF	CNF

Proof Size for UNSAT CNF: minimal number of s-DNFs to derive the empty clause $\square$.

Resolution over s-DNF

1. The $\wedge$-*introduction rule* is

$$\frac{\mathcal{D}_1 \vee \bigwedge_{j \in J_1} l_j \quad \mathcal{D}_2 \vee \bigwedge_{j \in J_2} l_j}{\mathcal{D}_1 \vee \mathcal{D}_2 \vee \bigwedge_{j \in J_1 \cup J_2} l_j},$$

   provided that $|J_1 \cup J_2| \leq s$.

2. The *cut (or resolution) rule* is

$$\frac{\mathcal{D}_1 \vee \bigvee_{j \in J} l_j \quad \mathcal{D}_2 \vee \bigwedge_{j \in J} \neg l_j}{\mathcal{D}_1 \vee \mathcal{D}_2},$$

3. The two *weakening rules* are

$$\frac{\mathcal{D}}{\mathcal{D} \vee \bigwedge_{j \in J} l_j} \quad \text{and} \quad \frac{\mathcal{D} \vee \bigwedge_{j \in J_1 \cup J_2} l_j}{\mathcal{D} \vee \bigwedge_{j \in J_1} l_j},$$

   provided that $|J| \leq s$.

We turn a Res(s) proof upside-down, i.e. reverse the edges of the underlying graph and negate the s-DNF on the vertices, we get a special kind of restricted branching s-program whose nodes are labelled by s-CNFs and at each node some s-disjunction is queried.

1. Querying a new s-disjunction, and branching on the answer, which can be depicted as follows.

$$\begin{tikzcd} & \begin{array}{c}\mathcal{C} \\ ?\bigvee_{j\in J} l_j\end{array} \arrow[dl, "\top"'] \arrow[dr, "\bot"] & \\ \mathcal{C}\wedge\bigvee_{j\in J} l_j & & \mathcal{C}\wedge\bigwedge_{j\in J}\neg l_j \end{tikzcd} \tag{1}$$

2. Querying a known s-disjunction, and splitting it according to the answer:

$$\begin{tikzcd} & \begin{array}{c}\mathcal{C}\wedge\bigvee_{j\in J_1\cup J_2} l_j \\ ?\bigvee_{j\in J_1} l_j\end{array} \arrow[dl, "\top"'] \arrow[dr, "\bot"] & \\ \mathcal{C}\wedge\bigvee_{j\in J_1} l_j & & \mathcal{C}\wedge\bigvee_{j\in J_2} l_j \end{tikzcd} \tag{2}$$

3. There are two ways of forgetting information,

$$\begin{array}{c} \mathcal{C}_1 \wedge \mathcal{C}_2 \\ \downarrow \\ \mathcal{C}_1 \end{array} \qquad \text{and} \qquad \begin{array}{c} \mathcal{C} \wedge \bigvee_{j \in J_1} l_j \\ \downarrow \\ \mathcal{C} \wedge \bigvee_{j \in J_1 \cup J_2} l_j \end{array}, \tag{3}$$

k-clique principle

$G = (V, E)$. We want to define a formula

$\text{Clique}_k(G)$ satisfiable iff G contains a k-clique.

$x_{iv} \equiv$ "v is the i-th node in the clique"

$$\text{Clique}_k(G) = \begin{cases} \bigvee_{v \in V} x_{i,v} & i \in [k] & \text{a node in each position} \\ \neg x_{i,v} \vee \neg x_{i,u} & u \neq v \in V, i \in [k] & \text{no two nodes in one position} \\ \neg x_{i,u} \vee \neg x_{j,v} & (u,v) \notin E, i \neq j \in [k] & \text{"no-edges" are not in the clique} \end{cases}$$

Fact

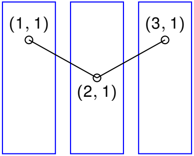
$\text{Clique}_k(G)$ *UNSAT iff G does not have a k-clique* *k*-Clique Principle: Simplified version

- G formed from k blocks V_b of n nodes each: $G = (\bigcup_{b\in[k]} V_b, E)$
- Variables $v_{i,q}$ with $i \in [k], a \in [n]$, with clauses

$$\mathrm{Clique}^{\mathrm{n}}_{\mathrm{k}}(G) = \begin{cases} \neg v_{i,a} \vee \neg v_{j,b} & ((i,a),(j,b)) \notin E \\ \bigvee_{a\in[n]} v_{i,a} & i \in [k] \end{cases}$$

Fact

$\mathrm{Clique}^{\mathrm{n}}_{\mathrm{k}}(G)$ *UNSAT iff G does not have a k-clique*



$$\mathrm{Clique}^{n}_{k}(G) = \begin{cases} x_{1,1} \\ x_{2,1} \\ x_{3,1} \\ (\neg x_{1,1} \vee \neg x_{3,1}) \end{cases}$$

Motivations(Informal): Clique^{n}_{k} captures the proof strength of adding to a proof system the ability to count up to k. [1,2]

[1]=[Beyersorff Galesi Lauria Razborov 12]
[2]=[Dantchev Martin Szeider 11]

k-Clique Principle (Binary Version)

- (Bit-)Variables: $\omega_{i,j}$, for $i \in [k], j \in [\log n]$
- Notation:

$$\omega_{i,j}^{a_j} = \begin{cases} \omega_{i,j} & \text{if } a_j = 1 \\ \neg\omega_{i,j} & \text{if } a_j = 0 \end{cases}$$

$$v_{i,j} \equiv (\omega_{i,1}^{a_1} \wedge \ldots \wedge \omega_{i,\log n}^{a_{\log n}}), \text{ where } (j)_2 = \vec{a}$$

$$\text{Bin-Clique}_k^n(G) = \bigwedge_{((i,a),(j,b)) \notin E} \left((\omega_{i,1}^{1-a_1} \vee \ldots \vee \omega_{i,\log n}^{1-a_{\log n}}) \vee (\omega_{j,1}^{1-b_1} \vee \ldots \vee \omega_{j,\log n}^{1-b_{\log n}}) \right)$$

Pigeonhole principle (Binary Version)

- (Bit-)Variables: $\omega_{i,j}$, for $i \in [m]$, $j \in [\log n]$,
- Notation:

$$\omega_{i,j}^{h_j} = \begin{cases} \omega_{i,j} & \text{if } h_j = 1 \\ \neg\omega_{i,j} & \text{if } h_j = 0 \end{cases}$$

ω_{ij} encodes that $i \mapsto h$ and j-th bit of h is h_j.

$$p_{ih} \equiv (\omega_{i1}^{h_1} \wedge \ldots \wedge \omega_{i \log n}^{h_{\log n}})$$

two distinct pigeons i and i' cannot go into the same hole h, i.e. with the same binary representation

PHP_n^m : *Unary encoding*	
$\bigvee_{j=1}^{n} p_{i,j}$	$i \in [m]$
$\overline{p}_{i,j} \vee \overline{p}_{i',j}$	$i, \neq i' \in [m], j \in [n]$

Bin-PHP_n^m : *Binary encoding*
$\bigvee_{j=1}^{\log n} \omega_{i,j}^{1-h_j} \vee \bigvee_{j=1}^{\log n} \omega_{i',j}^{1-h_j}$
$i \neq i' \in [m], h \in [n]$

- preserve the combinatorial hardness of the unary principle;
- are less exposed to details of the encoding when attacked with a lower bound technique;
- give significative lower bounds.

Example: Formula width

Size-Width tradeoffs for Res: $Size(F \vdash) \geq e^{\Omega(\frac{(w(F\vdash)-w(F))^2}{|Vars(F)|})}$.
Space-Width relation for Res:
$Space(F \vdash) \geq w(F \vdash) - w(F) + 1$

$$w(\text{PHP}) = n \textit{ while } w(\text{Bin-PHP}) = 2\log n$$
$$|Vars(\text{PHP})| = mn \textit{ while } |Vars(\text{Bin-PHP})| = m\log n$$

Fact

Res(1) *proofs of* $\mathrm{Clique}^{\mathrm{n}}_{\mathrm{k}}(G) \longmapsto$ Res($\log n$) *proofs of* $\text{Bin-Clique}^{\mathrm{n}}_{\mathrm{k}}(G)$.

$$v_{i,a} \equiv (\omega_{i,1}^{a_1} \wedge \ldots \wedge \omega_{i,\log n}^{a_{\log n}})$$

Fact

Res(1) *proofs of* $\mathrm{PHP}^m_n \longmapsto$ Res($\log n$) *proofs of* Bin-PHP^m_n

$$p_{ih} \equiv (\omega_{i1}^{h_1} \wedge \ldots \wedge \omega_{i\log n}^{h_{\log n}})$$

Known results for k-Clique Principles in Res

- For any G there are $O(n^k)$ proofs in tree-Res (brute force)
- If G is the $(k-1)$-partite graph: $\mathrm{Clique}^n_k(G)$ has Read Once-Res refutations of size $O(2^k n^2)$ [1,2]
- Difficult to find G's without a k-clique making hard to refute $\mathrm{Clique}^n_k(G)$.

Known Lower Bounds: $(G \sim \mathcal{G}(n,p),\ p = n^{-(1+\epsilon)\frac{2}{k-1}})$

$G \sim \mathcal{G}(n,p)$	tree-Res	Reg-Res	Res(1)	Res(s)
$\mathrm{Clique}^n_k(G)$	$\Omega(n^k)$[1]	$\Omega(n^k)$[2]	Open - $\Omega(2^k)$ [4]	Open
$\mathrm{Bin\text{-}Clique}^n_k(G)$	–	–	$\Omega(n^k)$[3]	$\Omega(n^k)$, $s = o(\sqrt{\log\log n})$

[1] = [Beyersdorff Galesi Lauria 13]
[2] = [Atserias Bonacina de Rezende Lauria Nördstrom Razborov 18]
[3] = [Lauria Pudlák Rödl Thapen 17]
[4] = [Pang 19]

Results for Bin-PHP$_n^m$

Theorem

$\delta > 0$. *Any refutation of* Bin-PHP$_n^m$ *in* Res(s) *for* $s \leq \sqrt{\log n}$ *is of size* $2^{\Omega(n^{1-\delta})}$.

Theorem

*There are tree-*Res(1) *refutations of* Bin-PHP$_n^m$ *of size* $2^{\Theta(n)}$.

Lower Bound Proof (for Bin-Clique$^n_k(G)$)

Main Tools(for Binary Principles):

1. *Covering Number* on s-DNFs [1]
   - Res(s) proofs with small CN efficiently simulated in Res($s-1$)
   - *Bottlenecks*
2. *(Random) restrictions* for binary principles
3. *Hardness properties* of Bin-Clique$^n_k(G)$, when $G \sim \mathcal{G}(n, p)$ [2]
4. Induction on s.
   - Base Case: known hardness on Res(1) [3].

[1]=[Segerlind Buss Impagliazzo 04]
[2]=[Beyersdorff Galesi Lauria 13]
[3]=[Lauria Pudlák Rödl Thapen 17]

A *covering set* for a s-DNF $\mathcal{F}$ is a set of literals L such that each term of $\mathcal{F}$ has at least a literal in L.

The *covering number* $cv(\mathcal{F})$ of a s-DNF $\mathcal{F}$ is the minimal size of a covering set for $\mathcal{D}$.

$$CN(\pi) = \max_{\mathcal{F} \in \pi} c(\mathcal{F})$$

Small covering number vs simulations

Lemma (Simulation Lemma)

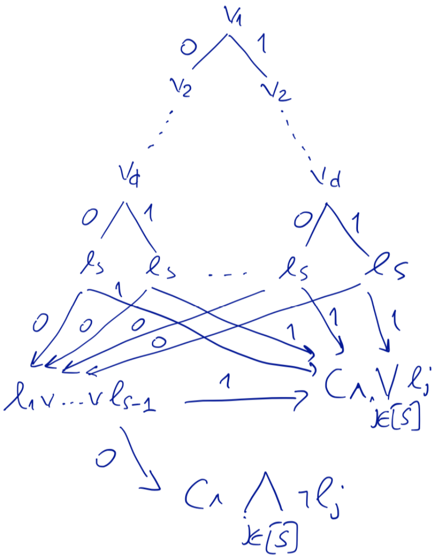
If F has a refutation π in $\mathrm{Res}(s)$ with $CN(\pi) < d$, then F has a $\mathrm{Res}(s-1)$ refutation of size at most $2^{d+2}N$.

Put π upside-down. Get a restricted branching s-program whose nodes are labelled by s-CNFs and at each node some s-disjunction $\bigvee_{j\in[s]} l_j$ is queried.

Example

$$
\begin{array}{ccccc}
 & & \vdots & & \\
 & & \mathcal{C} & & \\
 & & ?\bigvee_{j\in[s]} l_j & & \\
 & 1 \swarrow & & \searrow 0 & \\
\mathcal{C}\wedge\bigvee_{j\in[s]} l_j & & & & \mathcal{C}\wedge\bigwedge_{j\in[s]}\neg l_j
\end{array}
\tag{4}
$$

Let $cv(\mathcal{C}) < d$, witnessed by variable set $\{v_1, \ldots, v_d\}$.



Bottlenecks in Res(s)

A c-bottleneck in a Res(s) proof is a s-DNF F whose $cv(F) \geq c$. $c(s)$ is the *bottleneck number* at Res(s).

Fact (Independence)

If $c = rs$, $r \geq 1$ and $cv(F) \geq c$, then in F it is always possible to find r pairwise disjoint s-tuples of literals $T_1 = (\ell_1^1, \ldots, \ell_1^s), \ldots, T_r = (\ell_r^1, \ldots, \ell_r^s)$ such that the $\bigwedge T_i$'s are terms of F.

Restrictions

A *s-restriction* assigns $\lfloor \frac{\log n}{2^{s+1}} \rfloor$ bit-variables $\omega_{i,j}$ in each block $i \in [k]$.

Fact

if σ and τ are (disjoint) s-restrictions, then $\sigma\tau$ is a $(s-1)$-restriction

A *random s-restriction* for $\text{Bin-Clique}^{\text{n}}_{\text{k}}(G)$ is an s-restriction obtained by choosing independently in each block i, $\lfloor \frac{\log n}{2^{s+1}} \rfloor$ variables among $\omega_{i,1}, \ldots, \omega_{i,\log n}$, and setting these uniformly at random to 0 or 1.

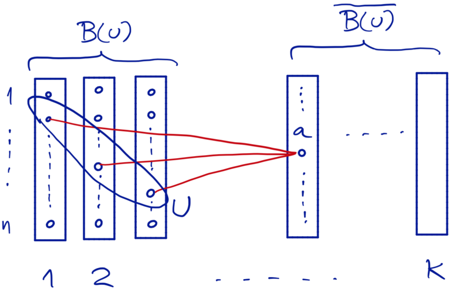
Hardness Properties

$G = (\bigcup_{b \in [k]} V_b, E)$ and $0 < \alpha < 1$. U is *α-transversal* if:

1. $|U| \leq \alpha k$, and
2. for all $b \in [k]$, $|V_b \cap U| \leq 1$.

Let $B(U) \subseteq [k]$ be the set of blocks mentioned in U, and $\overline{B(U)} = [k] \setminus B(U)$.

U is *extendible* in a block $b \in \overline{B(U)}$ if there exists a vertex $a \in V_b$ which is a *common neighbour of all nodes in U*.



A restriction σ is *consistent* with $v = (i, a)$ if for all $j \in [\log n]$, $\sigma(\omega_{i,j})$ is either a_j or not assigned (i.e. assigns the right bit or can do it in the future)

Definition

Let $0 < \alpha, \beta < 1$. A α-transversal U is β-extendible, if for all β-restriction σ, there is a node v^b in each block $b \in \overline{B(U)}$, such that σ is consistent with v^b.

Lemma (Extension Lemma, similar to [1])

Let $0 < \epsilon < 1$, let $k \leq \log n$. Let $1 > \alpha > 0$ and $1 > \beta > 0$ such that $1 - \beta > \alpha(2 + \epsilon)$. Let $G \sim \mathcal{G}(n, p)$. With high probability both properties hold:

1. *all α-transversal sets U are β-extendible;*
2. *G does not have a k-clique.*

[1]=[Beyersodrff Galesi Lauria 13]

Idea of the proof

Property (Clique(G, s, k))

For any s-restriction ρ, there are no Res(s) refutations of $\text{Bin-Clique}^{\mathrm{n}}_{\mathrm{k}}(G){\restriction}_{\rho}$ *of size less than* $n^{\frac{\delta(k-1)}{\mathrm{d}(s)}}$.

Theorem

If $\text{Clique}(G, s, k)$ *holds, then there are no* $\text{Res}(s)$ *proofs of* $\text{Bin-Clique}^{\mathrm{n}}_{\mathrm{k}}(G)$ *with size* $n^{\frac{\delta(k-1)}{\mathrm{d}(s)}}$.

Theorem

Let $1 < s = o(\sqrt{\log\log n})$. *There exists a graph G such that* $\text{Res}(s)$ *refutations of* $\text{Bin-Clique}^{\mathrm{n}}_{\mathrm{k}}(G)$ *are* $n^{\Omega(k)}$.

By Extension Lemma there exists a $G \sim \mathcal{G}(n, p)$ with the extension properties.

Lemma

$\text{Clique}(G, 1, k)$ *holds. (use [1])*

Steps of the proof

Lemma

Clique$(G, s-1, k) \Rightarrow$ Clique(G, s, k) *as long as* $s = o(\sqrt{\log\log n})$.

We prove that $\neg$Clique$(G, s, k) \Rightarrow \neg$Clique$(G, s-1, k)$. Let $L(s) = n^{\frac{\delta(k-1)}{\mathrm{d}(s)}}$.

1. Since $\neg$Clique(G, s, k), then $\exists$ a s-restriction ρ and π a proof of Bin-Clique$^{\mathrm{n}}_{\mathrm{k}}(G)\restriction_\rho$, such that $|\pi| < L(s)$.
2. Let $c = c(s)$ be the bottleneck number and $r = cs$
3. σ be a s-random restriction on Bin-Clique$^{\mathrm{n}}_{\mathrm{k}}(G)\restriction_\rho$.
4. Pr[bottleneck F survives in $\pi\restriction_\sigma$] $\leq e^{-\frac{r}{\mathrm{p}(s)}}$. Use *Independence Property*.
5. Pr[$CN(\pi\restriction_\sigma) \geq c$] < 1. *Union bound*.
6. Define $\tau = \sigma\rho$ and apply *Simulation Lemma* to $\pi\restriction_\sigma$. We get a (s-1)-restriction τ and a $\leq L(s)2^{c+2}$ size proof in $Res(s-1)$ of Bin-Clique$^{\mathrm{n}}_{\mathrm{k}}(G)\restriction_\tau$. If $L(s)2^{c+2} < L(s-1)$, this is $\neg$Clique$(G, s-1, k)$.
7. knowing $\mathrm{p}(s)$, define $\mathrm{d}(s)$ and $c(s)$ in such a way to force $L(s)2^{c+2} < L(s-1)$ and union bound to work.

The case of Bin-PHP$_n^m$

	tree-Res	Res(s), $m \leq 2n$	Res(s), $m > 2n$
Bin-PHP$_n^m$	$2^{\Theta(n)}$	$2^{\Omega(n^{1-\delta})}$ $(s = o(\sqrt{\log n}))$	$2^{\Omega(n^{1-\delta})}$ $(s = o(\sqrt{\log n}))$
PHP$_n^m$	$2^{\Theta(n \log n)}$ [3,4]	$2^{\Omega(\frac{n}{\log \log n})}(s \leq \sqrt{\log n})$ [2]	[1,...]

A form of optimality of the lower bound: [5] Proved an upper bound of $O(2^{\sqrt{n \log n}})$ in Res for PHP$_n^m$, when $m \geq 2^{\sqrt{n \log n}}$. Use the fact that size S proof in Res(1) for PHP implies size S proof in Res$(\log n)$ for Bin-PHP.

[1]=[Razborov 02] (Survey: "Proof Complexity of PHP")
[2]=[Segerlind Buss Impagliazzo 03]
[3]=[Beyersdorff Galesi Lauria 10]
[4]=[Dantchev Riis 01]
[5]=[Buss Pitassi 97]

Other Results for Binary Principles

$\mathrm{OP_n}$: *Unary encoding*	
$\overline{v}_{x,x}$	$x \in [n]$
$\overline{v}_{x,y} \vee \overline{v}_{y,z} \vee v_{x,z}$	$x, y, z \in [n]$
$\bigvee_{i \in [n]} v_{x,i}$	$x \in [n]$

$\mathrm{Bin\text{-}OP_n}$: *Binary encoding*	
$\overline{\nu}_{x,x}$	$x \in [n]$
$\overline{\nu}_{x,y} \vee \overline{\nu}_{y,z} \vee \nu_{x,z}$	$x, y, z \in [n]$
$\bigvee_{i \in [\log n]} \omega_{x,i}^{1-a_i} \vee \nu_{x,a}$	$x, a \in [n]$

Lemma

$\mathrm{Bin\text{-}OP_n}$ *and* $\mathrm{Bin\text{-}LOP_n}$ *have polynomial size* $\mathrm{Res}(1)$ *proofs.*

- Res proof complexity of binary version of propositional version of principles which are expressible as first order formulae with no finite model in Π_2-form, i.e. as $\forall \vec{x} \exists \vec{w} \varphi(\vec{x}, \vec{w})$ (Riis approach).
- Relations between different forms of binary encodings.
- Complexity of proofs in Res of the binary versions of a large family of formulas (those having clauses $v_{i,j} \oplus v_{j,i}$, implying a comparisons among all pairs of variables). *LOP* is included here.

Further Development in Sherali-Adams

[Dantchev Ghani Martin 19]. Similar approach for Sherali-Adams.