Resolution and the binary encoding of combinatorial principles

Stefan Dantchev\textsuperscript{2} \hspace{1cm} Nicola Galesi\textsuperscript{1} \hspace{1cm} Barnaby Martin\textsuperscript{2}

\textsuperscript{1}Sapienza University Rome
\textsuperscript{2}Durham University

Conference on Computational Complexity — July 20, 2019
A 2-DNF: 

\[(\neg v_1 \land v_2) \lor (v_2 \land v_3) \lor (\neg v_1 \land v_3)\]

<table>
<thead>
<tr>
<th></th>
<th>Resolution (= Res(1))</th>
<th>Res(2)</th>
</tr>
</thead>
</table>
| **Main Rule** | \[
\frac{C \lor x}{C \lor D}
\frac{\neg x \lor D}{C \lor D}
\] | \[
\frac{C \lor (x \land y)}{C \lor D}
\frac{(\neg x \lor \neg y) \lor D}{C \lor D}
\] |
| **Refutations for** | CNF | CNF |

**Proof Size for UNSAT CNF**: minimal number of s-DNFs to derive the empty clause \(\Box\).
The \textit{\&-introduction rule} is

\[
\frac{\mathcal{D}_1 \lor \bigwedge_{j \in J_1} l_j \quad \mathcal{D}_2 \lor \bigwedge_{j \in J_2} l_j}{\mathcal{D}_1 \lor \mathcal{D}_2 \lor \bigwedge_{j \in J_1 \cup J_2} l_j},
\]

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

The \textit{cut (or resolution) rule} is

\[
\frac{\mathcal{D}_1 \lor \bigvee_{j \in J} l_j \quad \mathcal{D}_2 \lor \bigwedge_{j \in J} \neg l_j}{\mathcal{D}_1 \lor \mathcal{D}_2},
\]

The two \textit{weakening rules} are

\[
\frac{\mathcal{D}}{\mathcal{D} \lor \bigwedge_{j \in J} l_j}
\quad \text{and} \quad
\frac{\mathcal{D} \lor \bigwedge_{j \in J_1 \cup J_2} l_j}{\mathcal{D} \lor \bigwedge_{j \in J_1} l_j},
\]

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.

\[ C \land \bigvee_{j \in J} l_j \quad \text{T} \quad \bot \quad C \land \bigwedge_{j \in J} \neg l_j \]

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

\[ C \land \bigvee_{j \in J_1 \cup J_2} l_j \quad \text{T} \quad \bot \quad C \land \bigvee_{j \in J_2} l_j \]
There are two ways of forgetting information,

\[ C_1 \land C_2 \quad \text{and} \quad C \land \bigvee_{j \in J_1} l_j \]

\[ C_1 \quad \text{and} \quad C \land \bigvee_{j \in J_1 \cup J_2} l_j \]
\( G = (V, E) \). We want to define a formula \( \text{Clique}_k(G) \) satisfiable iff \( G \) contains a \( k \)-clique. 

\( x_{i,v} \equiv \text{"}v \text{ is the } i\text{-th node in the clique\text{"}} \)

\[
\text{Clique}_k(G) = \begin{cases} 
\bigvee_{v \in V} x_{i,v} & i \in [k] \\
\neg x_{i,v} \lor \neg x_{i,u} & u \neq v \in V, i \in [k] \\
\neg x_{i,u} \lor \neg x_{j,v} & (u, v) \notin E, i \neq j \in [k] 
\end{cases}
\]

- a node in each position
- no two nodes in one position
- "no-edges" are not in the clique

**Fact**

\( \text{Clique}_k(G) \text{ UNSAT iff } G \text{ does not have a } k\text{-clique} \)
**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

$$\text{Clique}_{k}^{n}(G) = \left\{ \begin{array}{l} 
\neg v_{i,a} \lor \neg v_{j,b} & ((i, a), (j, b)) \notin E \\
\lor_{a \in [n]} v_{i,a} & i \in [k]
\end{array} \right.$$  

**Fact**

$\text{Clique}_{k}^{n}(G) \text{ UNSAT iff } G \text{ does not have a } k\text{-clique}$
Motivations (Informal): Clique$_k^n(G)$ 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:**
  
  $$\omega^{a_j}_{i,j} = \begin{cases} 
  \omega_{i,j} & \text{if } a_j = 1 \\
  \neg \omega_{i,j} & \text{if } a_j = 0 
  \end{cases}$$

  $$\nu_{i,j} \equiv (\omega^{a_1}_{i,1} \land \ldots \land \omega^{a_{\log n}}_{i,\log n}), \text{ where } (j)_2 = \vec{a}$$

### Resolution and the binary encoding of combinatorial principles

$$\text{Bin-Clique}^n_k(G) = \bigwedge_{((i,a),(j,b)) \notin E} \left( (\omega^{1-a_1}_{i,1} \lor \ldots \lor \omega^{1-a_{\log n}}_{i,\log n}) \lor (\omega^{1-b_1}_{j,1} \lor \ldots \lor \omega^{1-b_{\log n}}_{j,\log n}) \right)$$
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.

\[
\text{PHP}_n^m : \textit{Unary encoding} \quad \begin{align*}
\bigvee_{j=1}^{n} v_{i,j} & \quad i \in [m] \\
\overline{v_{i,j}} \lor \overline{v_{i',j}} & \quad i, \neq i' \in [m], j \in [n]
\end{align*}
\]

\[
\text{Bin-PHP}_n^m : \textit{Binary encoding} \quad \begin{align*}
\bigvee_{j=1}^{\log n} \neg\omega_{i,j} \lor \bigvee_{j=1}^{\log n} \neg\omega_{i',j} & \quad i \neq i' \in [m]
\end{align*}
\]

Size-Width tradeoffs for Res: \( \text{Size}(F \vdash) \geq e^{\Omega\left(\frac{(w(F \vdash) - w(F))^2}{\text{Vars}(F)}\right)} \)

Space-Width Relations for Res: \( \text{Space}(F \vdash) \geq w(F \vdash) - w(F) + 1 \)

\( w(\text{PHP}) = n \) while \( w(\text{Bin-PHP}) = 2\log n \)
Fact
Res(1) proofs of $\text{Clique}_k^n(G) \iff \text{Res}(\log n)$ proofs of $\text{Bin-Clique}_k^n(G)$.

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

Fact
Res(1) proofs of $\text{PHP}_m^m$ $\iff$ Res($\log n$) proofs of $\text{Bin-PHP}_m^m$
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: \( \text{Clique}^n_k(G) \) has Reg-Res refutations of size \( O(2^k n^2) \) \([1]\)
- Difficult to find \( G \)'s without a \( k \)-clique making hard to refute \( \text{Clique}^n_k(G) \).

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

<table>
<thead>
<tr>
<th>( G \sim \mathcal{G}(n, p) )</th>
<th>tree-Res</th>
<th>Reg-Res</th>
<th>Res(1)</th>
<th>Res(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Clique}^n_k(G) )</td>
<td>( \Omega(n^k) )[1]</td>
<td>( \Omega(n^k) )[2]</td>
<td>Open - ( \Omega(2^k) ) [4]</td>
<td>Open</td>
</tr>
<tr>
<td>( \text{Bin-Clique}^n_k(G) )</td>
<td>-</td>
<td>-</td>
<td>( \Omega(n^k) )[3]</td>
<td>( \Omega(n^k) ), ( s = o(\log \log n) )</td>
</tr>
</tbody>
</table>

[1] = [Beyersdorff Galesi Lauria 13 ]
[2] = [Atserias Bonacina de Rezende Lauria Nördstrom Razborov 18]
[3] = [Lauria Pudlák Rödl Thapen 17 ]
[4] = [Pang 19, ECCC]
$\varepsilon, \delta > 0$. Any refutation of Bin-PHP$_n^m$ in Res(s) for $s \leq 2+\varepsilon\sqrt{\log n}$ is of size $2^{\Omega(n^{1-\delta})}$.

Theorem

There are tree-Res(1) refutations of Bin-PHP$_n^m$ of size $2^{\Theta(n)}$. 
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 ∼ \mathcal{G}(n, p) [2,3,4]

4. Induction on s.
   - Base Case: known hardness on Res(1) [4].

[1]=Segerlind Buss Impagliazzo 04
[3]=Atserias Bonacina de Rezende Lauria Nördstrom Razborov 18
[4]=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 $\mathcal{D}$.

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

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

Put $\pi$ 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

\[ \begin{array}{c}
\vdots \\
C \\
? \bigvee_{j \in [s]} l_j \\
C \wedge \bigvee_{j \in [s]} l_j
\end{array} \quad \begin{array}{c}
1 \\
0 \\
C \wedge \bigwedge_{j \in [s]} \neg l_j
\end{array} \]
Let \( cv(C) < d \), witnessed by variable set \( \{v_1, \ldots, v_d\} \).
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 $T_1 = (\ell^1_1, \ldots, \ell^s_1), \ldots, T_r = (\ell^1_r, \ldots, \ell^s_r)$ such that the $\bigwedge T_i$’s are terms of $F$. 
A \textit{s-restriction} assigns $\lfloor \frac{\log n}{2^{s+1}} \rfloor$ bit-variables $\omega_{i,j}$ in each block $i \in [k]$.

**Fact**

\textit{if} $\sigma$ \textit{and} $\tau$ \textit{are (disjoint) s-restrictions, then} $\sigma \tau$ \textit{is a} $(s - 1)$-\textit{restriction}

A \textit{random s-restriction} for Bin-Clique$^n_k(G)$ 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

\[ G = \left( \bigcup_{b \in [k]} V_b, E \right) \] and \( 0 < \alpha < 1 \). \( U \) is \( \alpha \)-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 $\sigma$ is \textit{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).

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

\begin{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 $G \sim G(n, p)$. With high probability both properties hold:

1. all $\alpha$-transversal sets $U$ are $\beta$-extendible;
2. $G$ does not have a $k$-clique.
\end{lemma}

[1]=[Beyersdorff Galesi Lauria 13]
Idea of the proof

Property (Clique\((G, s, k))\):

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

Theorem

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

By Extension Lemma there exists a \(G \sim G(n, n^{-(2(1+\epsilon)/(k-1)})\) with the extension properties.

Lemma

Clique\((G, 1, k)\) holds. (use [1])

[1]=[Lauria Pudlák Rödl Thapen 17]
Steps of the proof

Lemma

\[ \text{Clique}(G, s - 1, k) \Rightarrow \text{Clique}(G, s, k). \]

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

- Since \( \neg \text{Clique}(G, s, k) \), then \( \exists \) a \( s \)-restriction \( \rho \) and \( \pi \) a proof of Bin-Clique\(_k^\mathbb{Z}(G)\rceil_\rho \), such that \( |\pi| < L(s) \).
- Let \( c = c(s) \) be the bottleneck number and \( r = cs \)
- \( \sigma \) be a \( s \)-random restriction on Bin-Clique\(_k^\mathbb{Z}(G)\rceil_\rho \).
- \( \Pr[\text{bottleneck } F \text{ survives in } \pi\rceil_\sigma] \leq e^{-\frac{r}{p(s)}} \). Use Independence Property.
- \( \Pr[\text{CN}(\pi\rceil_\sigma) \geq c] < 1 \). Union bound.
- Define \( \tau = \sigma\rho \) and apply Simulation Lemma to \( \pi\rceil_\sigma \). We get a (s-1)-restriction \( \tau \) and a \( \leq L(s)2^{c+2} \) size proof in Res\((s - 1)\) of Bin-Clique\(_k^\mathbb{Z}(G)\rceil_\tau \). If \( L(s)2^{c+2} < L(s - 1) \), this is \( \neg \text{Clique}(G, s - 1, k) \).
- knowing \( p(s) \), define \( d(s) \) 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\(_m^n\)  

<table>
<thead>
<tr>
<th></th>
<th>tree-Res</th>
<th>Res(s), (m \leq 2n)</th>
<th>Res(s), (m &gt; 2n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bin-PHP(_m^n)</td>
<td>(2^{\Theta(n)})</td>
<td>(2^{\Omega(n^{1-\delta})} ) ( (s \leq \log^{\frac{1}{2+\epsilon}} n))</td>
<td></td>
</tr>
<tr>
<td>PHP(_m^n)</td>
<td>(2^{\Theta(n \log n)}) [3,4]</td>
<td>(2^{\Omega\left(\frac{n}{\log \log n}\right)} ) ( (s \leq \sqrt{\log n})) [2]</td>
<td>[1]</td>
</tr>
</tbody>
</table>

**A form of optimality of the lower bound:** [5] Proved an upper bound of \(O(2^{\sqrt{n \log n}})\) in Res for PHP\(_m^n\), 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

**OP**

<table>
<thead>
<tr>
<th><strong>Unary encoding</strong></th>
<th><strong>Binary encoding</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>( \overline{v}_{x,x} )</td>
<td>( x \in [n] )</td>
</tr>
<tr>
<td>( \overline{v}<em>{x,y} \lor \overline{v}</em>{y,z} \lor v_{x,z} )</td>
<td>( x, y, z \in [n] )</td>
</tr>
<tr>
<td>( \lor_{i \in [n]} v_{x,i} )</td>
<td>( x \in [n] )</td>
</tr>
</tbody>
</table>

**Bin-OP**

<table>
<thead>
<tr>
<th><strong>Binary encoding</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>( \overline{v}_{x,x} )</td>
</tr>
<tr>
<td>( \overline{v}<em>{x,y} \lor \overline{v}</em>{y,z} \lor v_{x,z} )</td>
</tr>
<tr>
<td>( \lor_{i \in [\log n]} \omega_{x,i}^{1-a_i} \lor v_{x,a} )</td>
</tr>
</tbody>
</table>

**Lemma**

**Bin-OP** and **Bin-LOP** have polynomial size 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 \( \Pi_2 \)-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 a the binary versions of a large family of formulas (those having clauses \( v_{i,j} \oplus v_{j,i} \), implying a comparisons among all pair of variables). **LOP** is included here.

- Comparisons of binary encodings with other compact encodings: unary functional encodings where i.e. clauses of the form \( v_{i,1} \lor \ldots \lor v_{i,n} \) replaced with \( v_{i,1} + \ldots + v_{i,n} = 1 \).
Conclusions

We prove lower bounds for $\text{Res}(s)$ for binary principles without using any form of the Switching Lemma.

Ad hoc random restrictions and an inductive argument allow to lift hardness results for Resolution.

- Binary versions of combinatorial principles might be useful benchmark/starting-point for trying lower bounds in stronger proof systems.
Thanks for your attention!!!