On combinatorial principles and semi-algebraic proof systems

Ilario Bonacina

UPC Barcelona Tech

Haifa, July 31 2022 “Proof Complexity Workshop”

Talk based on a joint work with Maria Luisa Bonet (to appear LICS’22)

No algebra in this talk

- Logic based definitions for **static** semi-algebraic proof systems
- Natural combinatorial principles capturing the strength of those systems

Resolution (Res)

$F = C_1 \wedge \ldots \wedge C_m$ where C_j are clauses

Inference Rules

$$\frac{C \vee x \qquad C \vee \neg x}{C} \updownarrow \left\{ \begin{array}{ll} \dfrac{C \vee x \qquad C \vee \neg x}{C} & \text{(symmetric cut)} \\[2ex] \dfrac{C}{C \vee x \qquad C \vee \neg x} & \text{(symmetric weakening)} \end{array} \right.$$

$$\frac{C \vee \ell \vee \ell}{C \vee \ell} \text{ (idempotency)} \qquad\qquad \frac{}{x \vee \neg x} \text{ (excluded middle)}$$

Weighted Resolution

$F = \{(C_1, w_1),\ \ldots, (C_m, w_m)\}$ with w_i in a group, e.g. $\mathbb{Z}$, $\mathbb{F}_2, \ldots$

Substitution Rules

$$\frac{(C \vee x, w) \qquad (C \vee \neg x, w)}{(C, w)} \updownarrow \qquad\qquad \frac{(C, w_1 + w_2)}{(C, w_1) \qquad (C, w_2)} \updownarrow$$

$$\frac{(C \vee \ell \vee \ell, w)}{(C \vee \ell, w)} \text{ (idempotency)} \qquad\qquad \frac{}{(C, w) \qquad (C, -w)} \updownarrow$$

$$\frac{}{(x \vee \neg x, w)} \text{ (excluded middle)}$$

The definition works equally well for bounded depth-Frege.

(C_1, w_1) (C_2, w_2) $\dots$ (C_m, w_m)

$(C_m \vee y, w_m)$ $(C_m \vee \neg y, w_m)$

$(C \vee x, w)$ $(C \vee \neg x, w)$

(C, w) $(C, -w)$

(E, w) $(E, -w)$

…wait, but is this sound? $(\bot, 1)$

THM. The definitions we give for (unary) NS/SA/SOS correspond to systems p-equivalent to the usual definitions of (unary) NS/SA/SOS, when clauses are encoded using the **multiplicative** encoding.

$$\bigvee_{x \in Pos} x \vee \bigvee_{y \in Neg} \neg y \quad \longrightarrow \quad \left\{ \prod_{x \in Pos} \bar{x} \prod_{y \in Neg} y = 0 \right\}$$

$$\cup \left\{ x^2 = x,\ x + \bar{x} = 1,\ y^2 = y,\ y + \bar{y} = 1 \ :\ x \in Pos, y \in Neg \right\}$$

Sherali-Adams over $\mathbb{Z}$ ($SA_{\mathbb{Z}}$)

(C_1, w_1) (C_2, w_2) $\dots$ (C_m, w_m)

$(C_m \vee y, w_m)$ $(C_m \vee \neg y, w_m)$

$(C \vee x, w)$ $(C \vee \neg x, w)$

(C, w)

(C, w) $(C, -w)$ (E, w) $(E, -w)$

Only clauses with positive weights $(\perp, m)$ $m > 0$

Unary Sherali-Adams over $\mathbb{Z}$ ($uSA_{\mathbb{Z}}$)

(C_1, w_1) (C_2, w_2) $\ldots$ (C_m, w_m)

$(C_m \vee y, w_m)$ $(C_m \vee \neg y, w_m)$

$(C \vee x, w)$ $(C \vee \neg x, w)$

No instances of the rule $\dfrac{(C, w_1 + w_2)}{(C, w_1) \qquad (C, w_2)} \updownarrow$

And weights in $\{\pm 1\}$

(C, w)

(C, w) $(C, -w)$ (E, w) $(E, -w)$

Only clauses with positive weights $(\bot, 1)\ldots(\bot, 1)$

Nullstellensatz over $\mathbb{Z}$ ($NS_{\mathbb{Z}}$)

(C_1, w_1) (C_2, w_2) $\dots$ (C_m, w_m)

$(C_m \vee y, w_m)$ $(C_m \vee \neg y, w_m)$

$(C \vee x, w)$ $(C \vee \neg x, w)$

(C, w)

(C, w) $(C, -w)$ (E, w) $(E, -w)$

Only weakenings of initial clauses $(\perp, m)$ $m \neq 0$

Unary Nullstellensatz over $\mathbb{Z}$ ($uNS_{\mathbb{Z}}$)

(C_1, w_1) (C_2, w_2) $\dots$ (C_m, w_m)

$(C_m \vee y, w_m)$ $(C_m \vee \neg y, w_m)$

$(C \vee x, w)$ $(C \vee \neg x, w)$

(C, w)

No instances of the rule $\dfrac{(C, w_1 + w_2)}{(C, w_1) \qquad (C, w_2)} \updownarrow$

And weights in $\{\pm 1\}$

(C, w) $(C, -w)$ (E, w) $(E, -w)$

Only weakenings of initial clauses $(\bot, 1) \dots (\bot, 1)$

Nullstellensatz over $\mathbb{F}_p$ ($NS_{\mathbb{F}_p}$)

(C_1, w_1) (C_2, w_2) $\dots$ (C_m, w_m)

$(C_m \vee y, w_m)$ $(C_m \vee \neg y, w_m)$

$(C \vee x, w)$ $(C \vee \neg x, w)$

Weights in $\mathbb{F}_p$ and the sum also over $\mathbb{F}_p$

(C, w)

(C, w) $(C, -w)$ (E, w) $(E, -w)$

Only weakenings of initial clauses $(\perp, m)$ $m \neq 0$

Sum-of-Squares over $\mathbb{Z}$ ($SOS_{\mathbb{Z}}$)

(C_1, w_1) (C_2, w_2) $\ldots$ (C_m, w_m)

$(C_m \vee y, w_m)$ $(C_m \vee \neg y, w_m)$

$(C \vee x, w)$ $(C \vee \neg x, w)$

(C, w)

(C, w) $(C, -w)$ (E, w) $(E, -w)$

Partitioned into sets the form

$\{(C_i, w_i^2), (C_i \vee C_j, w_i w_j) \ : \ i \neq j \in I\}$

$(\perp, m)$ $m > 0$

Unary Sum-of-Squares over $\mathbb{Z}$ ($uSOS_{\mathbb{Z}}$)

$(C_1, w_1) \qquad (C_2, w_2) \qquad \dots \qquad (C_m, w_m)$

$(C_m \vee y, w_m) \qquad (C_m \vee \neg y, w_m)$

$(C \vee x, w) \qquad (C \vee \neg x, w)$

(C, w)

No instances of the rule $\dfrac{(C, w_1 + w_2)}{(C, w_1) \qquad (C, w_2)} \updownarrow$

And weights in $\{\pm 1\}$

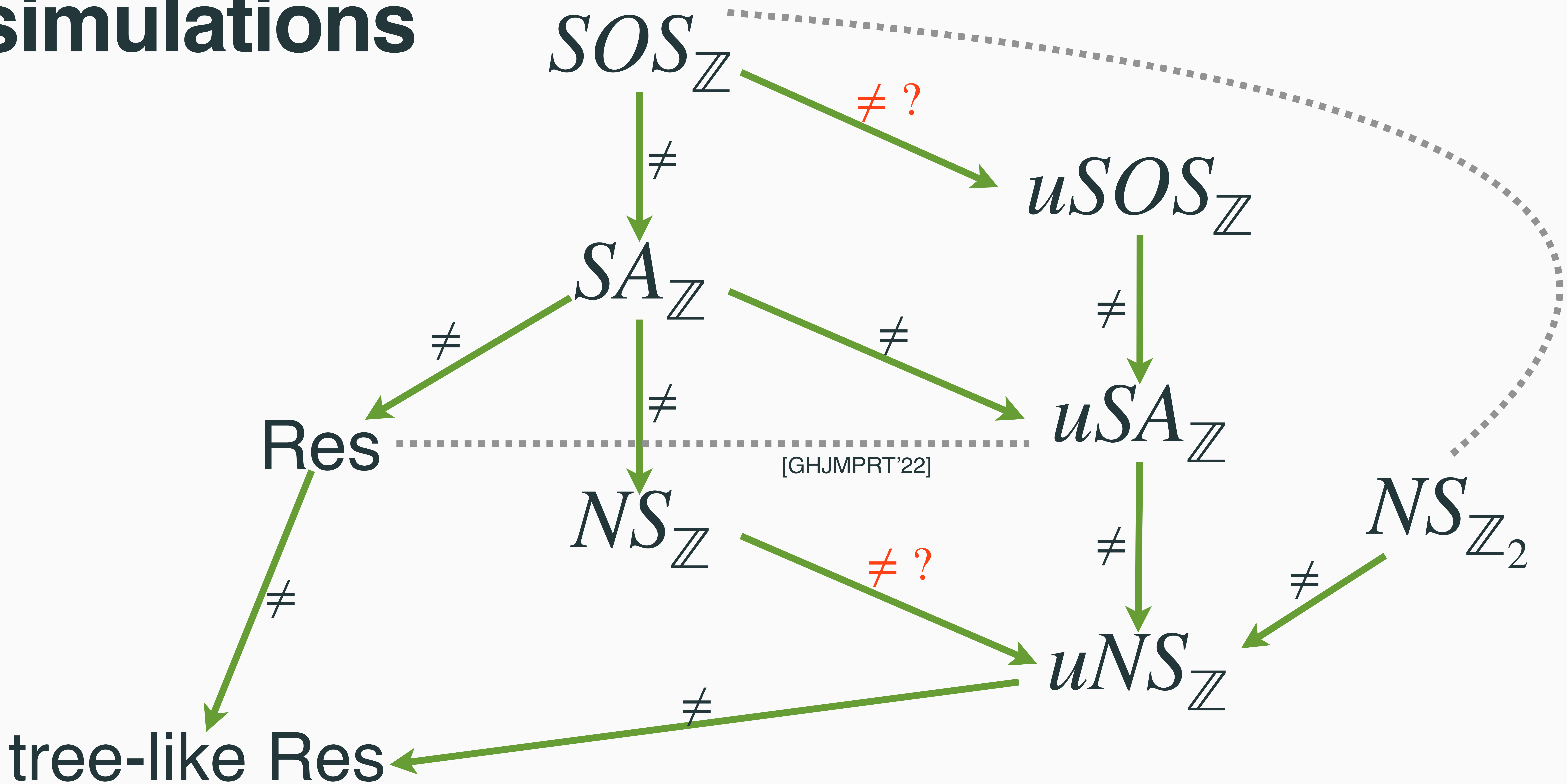
$(C, w) \qquad (C, -w) \qquad (E, w) \qquad (E, -w)$

Partitioned into sets the form

$\{(C_i, 1), (C_i \vee C_j, w_i w_j) \ : \ i \neq j \in I\}$

$(\bot, 1) \dots (\bot, 1)$

p-simulations



$A \longrightarrow B$ A p-simulates B

$A \cdots\cdots B$ A and B are incomparable

Combinatorial principles (Recap)

$SA_{\mathbb{Z}}$ Weighted Pigeonhole Principle

$uSA_{\mathbb{Z}}$ Pigeonhole Principle

$NS_{\mathbb{Z}_2}$ Perfect Matching Principle

$NS_{\mathbb{Z}}$ Onto-Functional Weighted Pigeonhole Principle

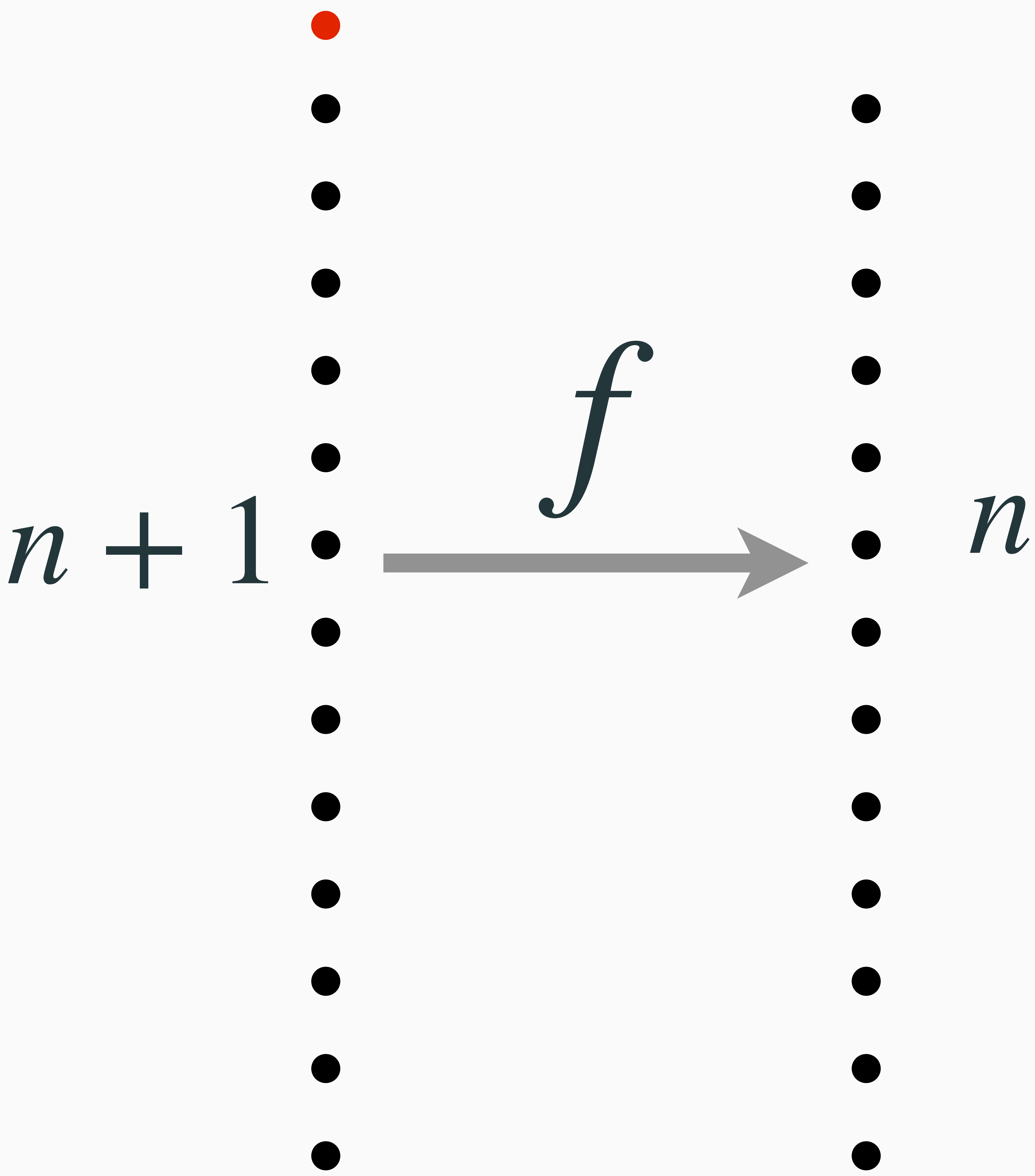
$uNS_{\mathbb{Z}}$ Onto-Functional Pigeonhole Principle

$SOS_{\mathbb{Z}}$

$uSOS_{\mathbb{Z}}$

Some other variations of Pigeonhole principle (**Work in progress**)

Pigeonhole Principle



PHP_n^{n+1}: f is total and injective

$x_{i1} \vee \cdots \vee x_{in}$ f.a. $i \in [n+1]$

$\neg x_{ij} \vee \neg x_{i'j}$ f.a. $j \in [n]$ & $i \neq i' \in [n+1]$

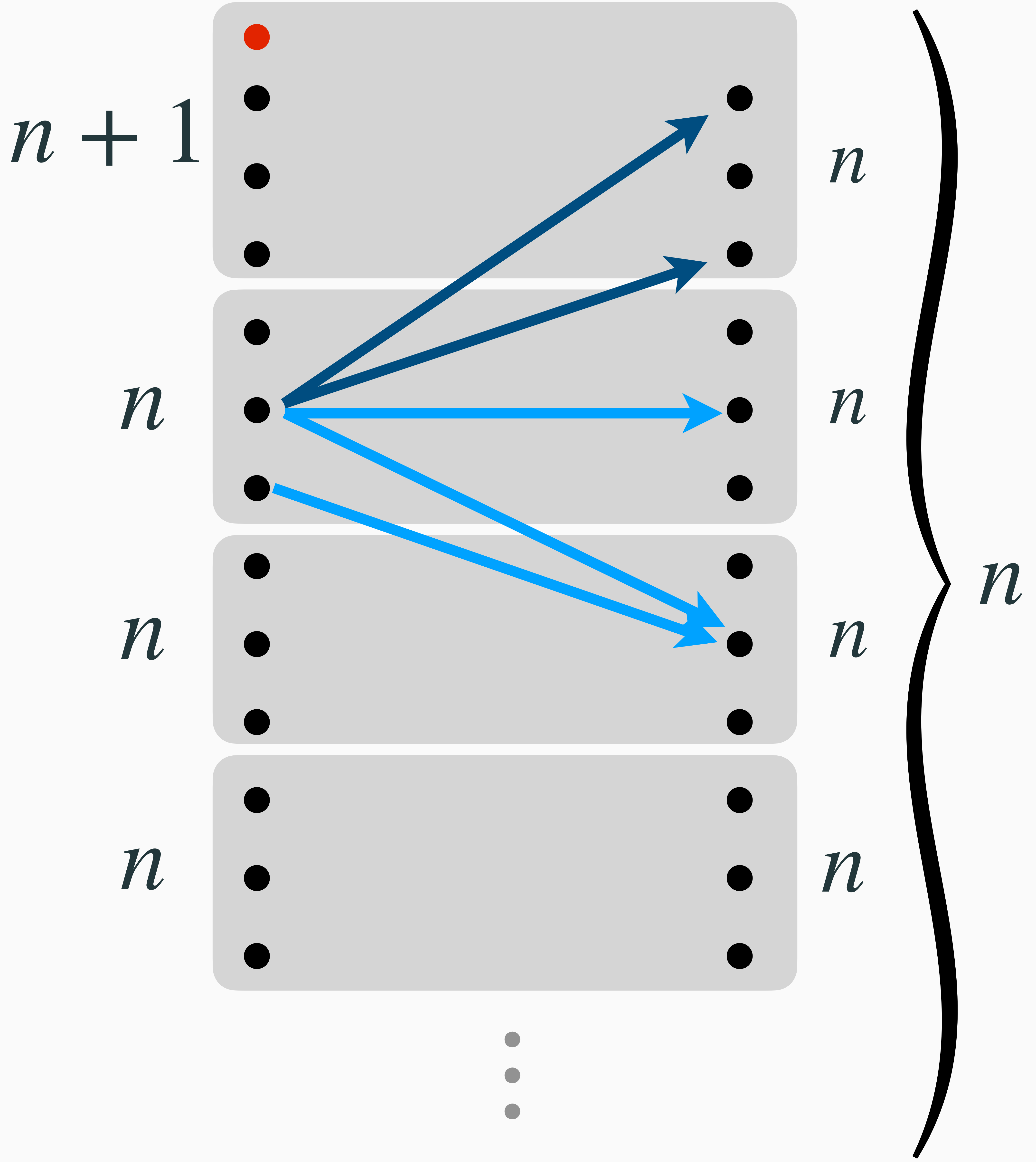
$PHP(G)$ is PHP_n^{n+1} where

$G \subseteq K_{n+1,n}$ and $x_{ij} =$ "False" for

every $(i,j) \notin E(G)$

THM. $PHP(G)$ is easy to refute in $uSA_{\mathbb{Z}}$

Weighted PHP (*wt*PHP)



- Pigeons fly to holes in the same group or in some adjacent group.
- If a pigeon flies to the lower group it must fly twice.
- Holes can accept at most 1 pigeon coming from the same group or the larger group.
- Holes can accept at most 2 pigeons coming from the lower group.

THM. $wtPHP(G)$ is easy to refute in $SA_{\mathbb{Z}}$

$$
\begin{tikzcd}
\text{Tree-like Res}(d)+PHP(G) \arrow[d] & \text{Tree-like Res}(d)+wtPHP(G) \arrow[l] \arrow[d] \\
\text{degree-}d\ uSA_{\mathbb{Z}} \arrow[d] & \text{degree-}d\ SA_{\mathbb{Z}} \arrow[l] \arrow[d] \\
\text{width-}d\ \text{Res}+PHP(G) & \text{width-}d\ \text{Res}+wtPHP(G) \arrow[l]
\end{tikzcd}
$$

The graphs G can be taken of degree at most 3 and the height of the Res(d) derivations is 5.

Res(d) $+PHP$

$F = C_1 \wedge \ldots \wedge C_m$ where C_j are d-DNF

Each π_j is a Res(d)-derivation from F of a d-DNF D_i' and all together the $D_1', \ldots, D_\ell'$ are a substitution instance of PHP_n^{n+1}

THM. Analogous p-simulations for:

- $NS_{\mathbb{Z}}$ but with **onto-functional** versions of $PHP(G)$ and $wtPHP(G)$
- $NS_{\mathbb{F}_2}$ but with MOD_2 principle [IS'06]
- depth-d versions of NS/SA
- uSOS/SOS (new combinatorial principles, **work in progress**)

The argument in all those cases is essentially the same.

Proof Idea: Generalize the p-simulation of DRMaxSAT by bounded-depth Frege + PHP from [BBIM-SM'18].

Depth-c Frege$+PHP(G)$

$\downarrow$

$uSOS_{\mathbb{Z}}$ where all the squares are only allowed to have at most $O(\log n)$ negative monomials

Depth-d version of Sherali-Adams

$SA_{\mathbb{Z}}^{(d)}$ is defined as $SA_{\mathbb{Z}}$ but instead of using weighted resolution uses weighted depth-d Frege and the same soundness condition.

THM. $SA_{\mathbb{Z}}^{(d)}$ is p-equivalent to circular depth-d Frege.

THM. $uSA_{\mathbb{Z}}^{(d)}$ is strictly stronger than depth-d Frege, at least for $d = o(\log \log n)$.

Proof. Use hardness of PHP in depth-d Frege

THM. MOD_2 is hard to refute in $uSA_{\mathbb{Z}}^{(d)}$, at least for $d = o(\log \log n)$.

Proof. Use hardness of MOD_2 in depth-d Frege $+PHP$ [Aj'90, BP'96]

Open problems

Is MOD_2 hard for depth-d Frege $+$ $wtPHP$? (E.g. for constant d)

A **yes** would imply MOD_2 is hard for $SA^{(d)}_{\mathbb{Z}}$ (and circular depth-d Frege)

Is $wtPHP$ hard for depth-d Frege $+$ PHP? (E.g. for constant d)

A **yes** would imply $uSA^{(d)}_{\mathbb{Z}}$ does not p-simulate $SA_{\mathbb{Z}}$

Does $uSOS_{\mathbb{Z}}$ p-simulate Resolution?

Find some family of combinatorial principles Φ s.t. depth-d Frege + Φ p-simulates Cutting Planes. (e.g. is $\Phi = PHP + MOD_p$ enough?)