Monotone Circuit Lower Bounds from Resolution

Ankit Garg *MSR*
Mika Göös *Harvard*
Pritish Kamath *MIT*
Dmitry Sokolov *KTH*

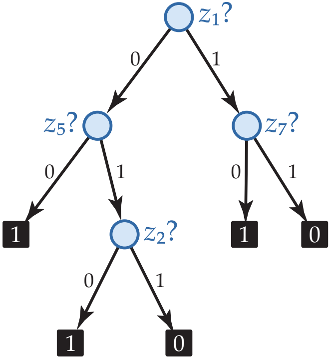
Background:

Query-to-communication lifting

(topic of my PhD thesis)

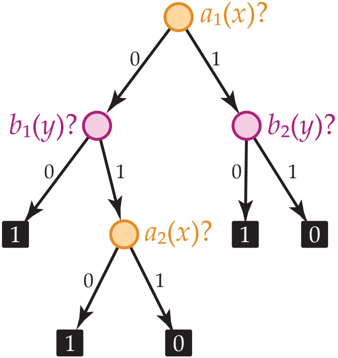
Query vs. Communication

$f(z)$



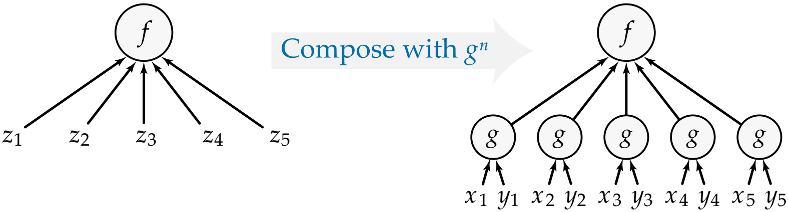
Decision trees

$F(x, y)$



Communication protocols

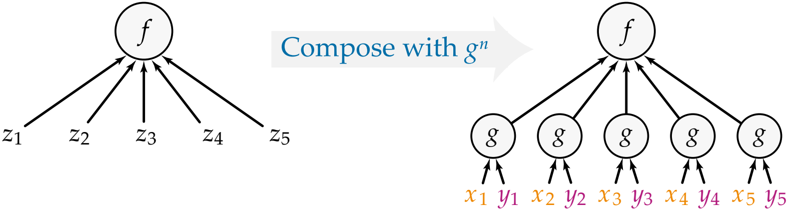
Composed functions $f \circ g^n$



Examples:

- Set-disjointness: $\textsc{Or} \circ \textsc{And}^n$
- Inner-product: $\textsc{Xor} \circ \textsc{And}^n$
- Equality: $\textsc{And} \circ \neg\textsc{Xor}^n$

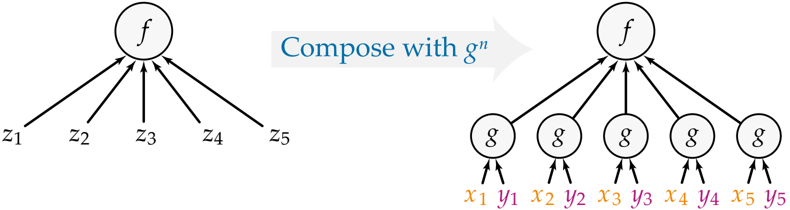
Composed functions $f \circ g^n$



In general: $g\colon \{0,1\}^m \times \{0,1\}^m \to \{0,1\}$ is a small gadget

- **Alice** holds $x \in (\{0,1\}^m)^n$
- **Bob** holds $y \in (\{0,1\}^m)^n$

Composed functions $f \circ g^n$



Lifting Theorem Template:

$$\mathsf{M}^{\mathsf{cc}}(f \circ g^n) \approx \mathsf{M}^{\mathsf{dt}}(f)$$

Composed functions $f \circ g^n$

M	Query	Communication	
P	deterministic	deterministic	[RM99, GPW15, dRNV16, HHL16]
BPP	randomised	randomised	[GPW17, AGJ⁺17]
NP	nondeterministic	nondeterministic	[GLM⁺15, G15]
many	poly degree	rank	[SZ09, She11, RS10, RPRC16]
many	conical junta deg.	nonnegative rank	[GLM⁺15, KMR17]
P^{NP}	decision list	rectangle overlay	[GKPW17]
	Sherali–Adams	LP complexity	[CLRS16, KMR17]
	sum-of-squares	SDP complexity	[LRS15]

Lifting Theorem Template:

$$\mathsf{M}^{\mathsf{cc}}(f \circ g^n) \approx \mathsf{M}^{\mathsf{dt}}(f)$$

Example: Classical vs. Quantum

[ABK16,ABB+16,GPW17]

$$\mathsf{BPP}^{\mathsf{dt}}(f) \;\geq\; \mathsf{BQP}^{\mathsf{dt}}(f)^{2.5}$$

$$\Downarrow$$

$$\mathsf{BPP}^{\mathsf{cc}}(f \circ g^n) \;\geq\; \mathsf{BQP}^{\mathsf{cc}}(f \circ g^n)^{2.5}$$

More lifting applications

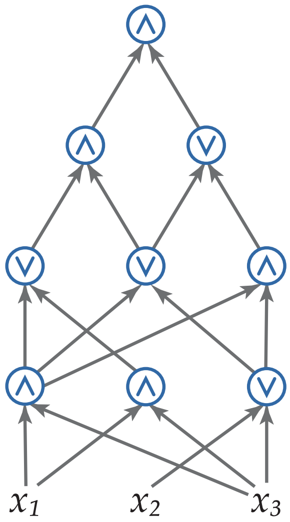
1. Monotone circuit complexity
2. Lower bounds in proof complexity
3. Multiparty set-disjointness
4. Communication vs. partition numbers
5. Clique vs. independent set
6. Alon–Saks–Seymour in graph theory
7. LP and SDP extension complexity
8. Learning theory (sign rank)
9. Approximate Nash equilibria

More lifting applications

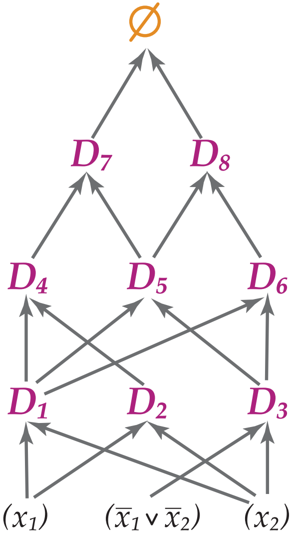
1. **Monotone circuit complexity**
2. **Lower bounds in proof complexity**
3. Multiparty set-disjointness
4. Communication vs. partition numbers
5. Clique vs. independent set
6. Alon–Saks–Seymour in graph theory
7. LP and SDP extension complexity
8. Learning theory (sign rank)
9. Approximate Nash equilibria

This work:

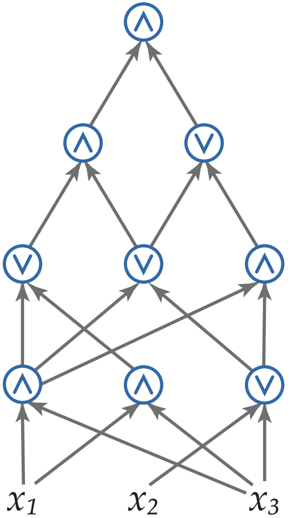
Monotone Circuit Lower Bounds
from Resolution



Monotone circuit



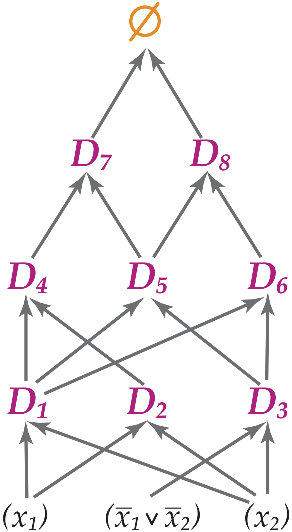
Resolution refutation



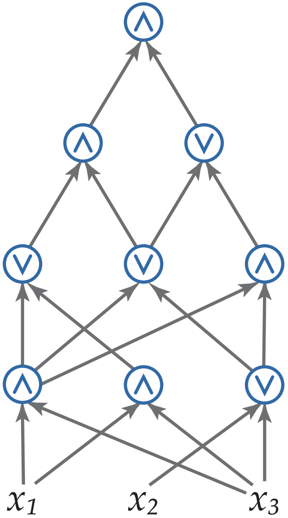
Monotone circuit

$\Longleftarrow$

Mon. feasible interpolation
[BPR97, Kra97]



Resolution refutation



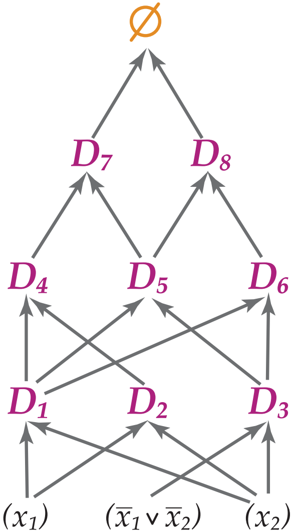
Monotone circuit

⟸

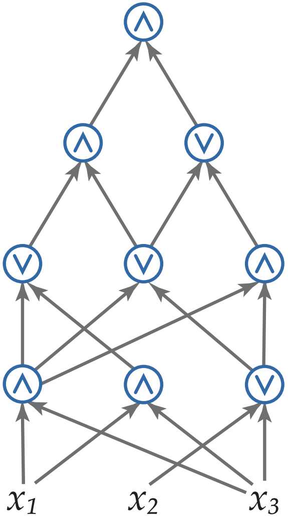
Mon. feasible interpolation [BPR97, Kra97]

⟹

This work



Resolution refutation

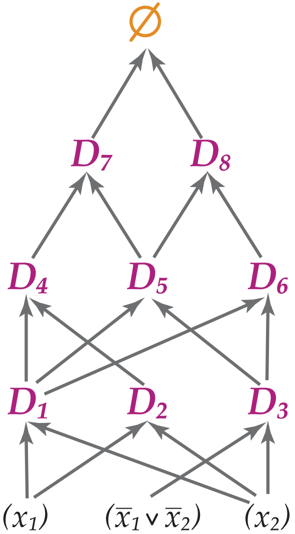


$\Longleftarrow$

Mon. feasible interpolation [BPR97, Kra97]

$\Longrightarrow$

This work



~~Monotone circuit~~
Dag-like protocol

~~Resolution refutation~~
Dag-like query model

▶ Monotone circuits

mKW search problem for monotone $f\colon \{0,1\}^n \to \{0,1\}$

- *input:* $(x, y) \in f^{-1}(1) \times f^{-1}(0)$
- *output:* coordinate $i \in [n]$ with $x_i = 1, y_i = 0$

▶ Proof systems

CNF search problem for unsatisfiable $F = \bigwedge_i D_i$

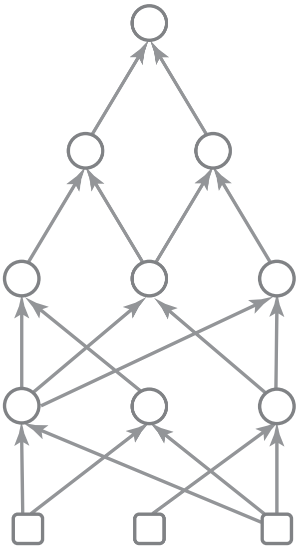
- *input:* truth assignment $z \in \{0,1\}^n$
- *output:* clause D_i such that $D_i(z) = 0$

Resolution refutation

Each dag node v is labeled with a disjunction $D_v\colon \{0,1\}^n \to \{0,1\}$

- *root r:* $D_r \equiv 0$ (constant 0)
- *node v with children u, u':*

  $D_v^{-1}(1) \supseteq D_u^{-1}(1) \cap D_{u'}^{-1}(1)$
- *leaf v:* D_v is an axiom

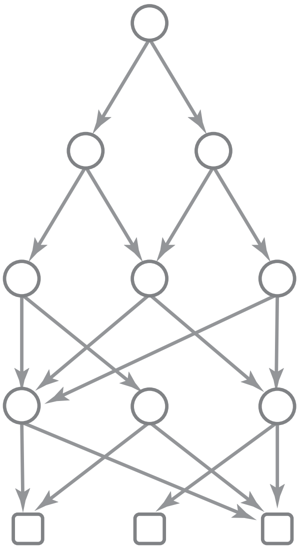


Top-down definition

Each dag node v is labeled with a conjunction $C_v\colon \{0,1\}^n \to \{0,1\}$

- *root r:* $C_r \equiv 1$ (constant 1)
- *node v with children u, u':*

  $C_v^{-1}(1) \subseteq C_u^{-1}(1) \cup C_{u'}^{-1}(1)$
- *leaf v:* Labeled with solution to **CNF search problem** valid for all $C_v^{-1}(1)$



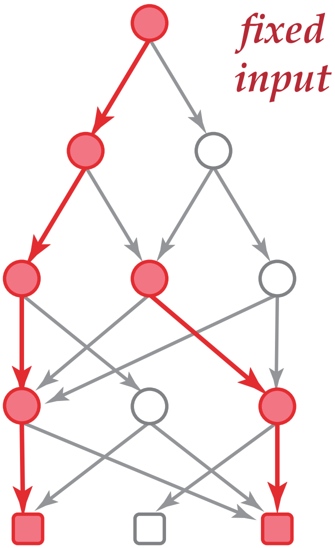
Top-down definition

Each dag node v is labeled with a conjunction $C_v \colon \{0,1\}^n \to \{0,1\}$

- *root r:* $C_r \equiv 1$ (constant 1)
- *node v with children u, u':*

$$\underbrace{C_v^{-1}(1)}_{\text{feasible set}} \subseteq C_u^{-1}(1) \cup C_{u'}^{-1}(1)$$

- *leaf v:* Labeled with solution to **CNF search problem** valid for all $C_v^{-1}(1)$



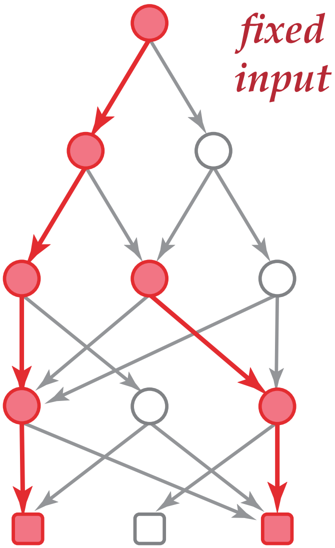
Monotone circuits

Let $f\colon \{0,1\}^n \to \{0,1\}$ be monotone

Each dag node v is labeled with a rectangle $R_v \subseteq f^{-1}(1) \times f^{-1}(0)$

- *root r:* $R_r = f^{-1}(1) \times f^{-1}(0)$
- *node v with children u, u':*

  $R_v \subseteq R_u \cup R_{u'}$
- *leaf v:* labeled with solution to **mKW search problem** valid for all R_v

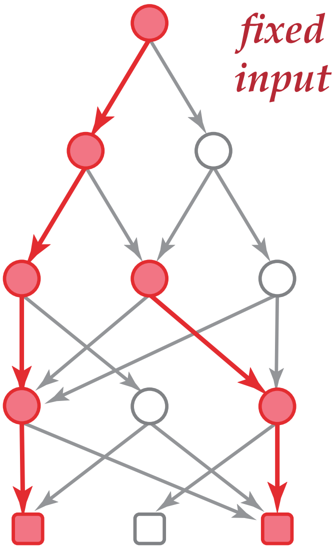


Abstract $\mathcal{F}$-dags

Let $S \subseteq \mathcal{I} \times \mathcal{O}$ be a search problem

Each dag node v is labeled with an $f_v : \mathcal{I} \to \{0,1\}$ from family $\mathcal{F}$

- *root r:* $f_r \equiv 1$ (constant 1)
- *node v with children u, u':*
  $f_v^{-1}(1) \subseteq f_u^{-1}(1) \cup f_{u'}^{-1}(1)$
- *leaf v:* labeled with solution to S valid for all $f_v^{-1}(1)$



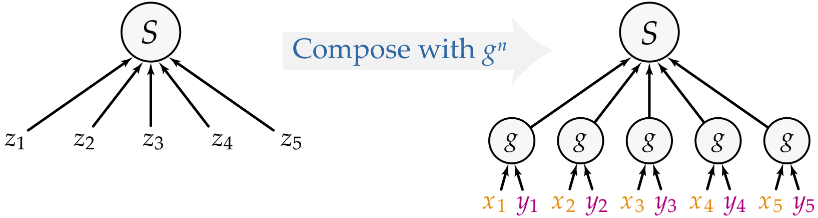
Summary

Model	**Family $\mathcal{F}$**	**Problem S**
Abstract $\mathcal{F}$-dags	$\mathcal{F}$	*any* S
Resolution	conjunctions	CNF search
Monotone circuit	rectangles	mKW search

Our result

Setup

- $S \subseteq \{0,1\}^n \times \mathcal{O}$ any query search problem
- $w(S)$ is the least **width** of conjunction-dag that solves S *(aka Resolution width)*
- $g\colon [m] \times \{0,1\}^m \to \{0,1\}$ where $m = n^{O(1)}$ is two-party index function: $g(x,y) = y_x$
- $S \circ g^n$ is **composed** search problem



Our result

Setup

- $S \subseteq \{0,1\}^n \times \mathcal{O}$ any query search problem
- $w(S)$ is the least **width** of conjunction-dag that solves S *(aka Resolution width)*
- $g\colon [m] \times \{0,1\}^m \to \{0,1\}$ where $m = n^{O(1)}$ is two-party index function: $g(x,y) = y_x$
- $S \circ g^n$ is **composed** search problem

Result

Rectangle-dag complexity of $S \circ g^n$ is

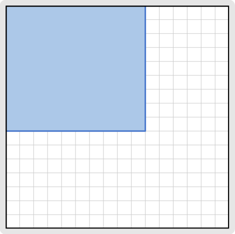
$$n^{\Theta(w(S))}$$

Bonus

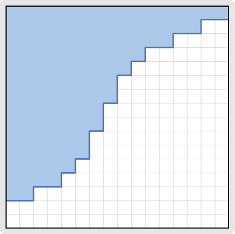
- **Triangle**-dags $\equiv$ Monotone **real** circuits [HC99, Pud97, HP17]

$\Uparrow$

- **LTF**-dags $\equiv$ Cutting Planes refutations



Rectangle



Triangle

Our result

Bonus

- **Triangle**-dags $\equiv$ Monotone **real** circuits [HC99, Pud97, HP17]

$\Uparrow$

- **LTF**-dags $\equiv$ Cutting Planes refutations

Result

Triangle-dag complexity of $S \circ g^n$ is

$$n^{\Theta(w(S))}$$

Upshot

1. Start with n-variable k-CNF F of Resolution width w
2. **Apply result:** $S_F \circ g^n$ has triangle-dag complexity $n^{\Theta(w)}$
3. Interpret $S_F \circ g^n$ as mKW/CNF search problem:

mKW: monotone function $f\colon \{0,1\}^{n^{O(k)}} \to \{0,1\}$
with monotone circuit complexity $n^{\Theta(w)}$

CNF: $n^{O(1)}$-variable $(k+O(1))$-CNF formula
with Cutting Planes complexity $n^{\Theta(w)}$

Previously: Clique [Pud97], random CNF [HP17, FPPR17]

Upshot

Jukna's 2012 textbook (Research Problem 19.17)

"It would be nice to have a lower bounds argument for cutting plane proofs ***explicitly*** showing what properties of contradictions do force long derivations."

mKW: monotone function $f\colon \{0,1\}^{n^{O(k)}} \to \{0,1\}$ with monotone circuit complexity $n^{\Theta(w)}$

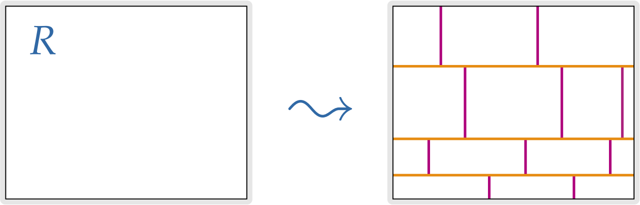
CNF: $n^{O(1)}$-variable $(k+O(1))$-CNF formula with Cutting Planes complexity $n^{\Theta(w)}$

Previously: Clique [Pud97], random CNF [HP17, FPPR17]

Tools from Prior Work

[GLMWZ15, GPW17]

Rectangles ↔ conjunctions



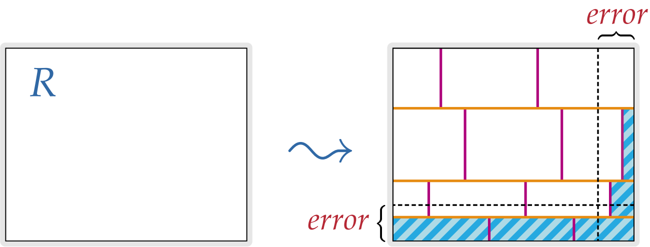
Large rectangle $R \subseteq [m]^n \times \{0,1\}^{mn}$ in the domain of $S \circ g^n$ can be partitioned into subrectangles

$$R = \bigcup_i R^i$$

such that $g^n(R^i) =$ **large subcube** in the domain of S

R of density 2^{-d} $\implies$ codimension-d subcubes

Rectangles $\leftrightarrow$ conjunctions



Large rectangle $R \subseteq [m]^n \times \{0,1\}^{mn}$ in the domain of $S \circ g^n$ can be partitioned into subrectangles

$$R = \bigcup_i R^i$$

such that $g^n(R^i) =$ **large subcube** in the domain of S

R of density 2^{-d} $\implies$ codimension-d subcubes

Game-Theoretic Characterisation of Resolution Width

[Pud00, AD08]

Explorer vs Adversary

Let $S \subseteq \{0,1\}^n \times \mathcal{O}$ be a search problem

- Game state is $\rho \in \{0,1,*\}^n$, initially $\rho = *^n$
- In each round *Explorer* makes a move

  Query: *Explorer* chooses $i \in [n]$
  Adversary responds $b \in \{0,1\}$
  Update $\rho_i \leftarrow b$

  Forget: *Explorer* chooses $i \in [n]$
  Update $\rho_i \leftarrow *$

- Game ends when solution to S can be deduced for ρ

Explorer vs Adversary

Let $S \subseteq \{0,1\}^n \times \mathcal{O}$ be a search problem

- Game state is $\rho \in \{0,1,*\}^n$, initially $\rho = *^n$
- In each round *Explorer* makes a move

Query: *Explorer* chooses $i \in [n]$
Adversary responds $b \in \{0,1\}$
Update $\rho_i \leftarrow b$

Forget: *Explorer* chooses $i \in [n]$
Update $\rho_i \leftarrow *$

- Game ends when solution to S can be deduced for ρ

$w(S)$ = least w such that *Explorer* has a strategy that maintains ρ of width $\leq w$

Proof outline:

Given size-2^d rectangle-dag for $S \circ g^n$
extract width-d Explorer-strategy for S

Proof outline

1. For each node v of rectangle-dag, partition $R_v = \bigcup_i R_v^i$ where each subrectangle is $\underbrace{\textbf{$\rho$-like}}$ for $|\rho| \le d$

$$g^n(R_v^i) = \text{ strings consistent with } \rho$$

Proof outline

1. For each node v of rectangle-dag, partition $R_v = \bigcup_i R_v^i$ where each subrectangle is $\underbrace{\textbf{$\rho$-like}}$ for $|\rho| \leq d$

   $g^n(R_v^i) =$ strings consistent with ρ

2. Extract width-d Explorer-strategy by walking down the rectangle-dag, starting at root

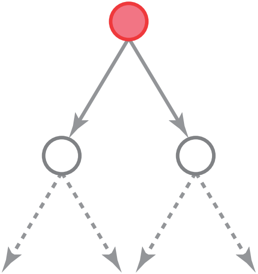
Invariant

At node v: Game state ρ, maintain ρ-like $R' \subseteq R_v$

1. Root

$R_{\text{root}} \;=\;$ domain of g^n

which is $*^n$-like

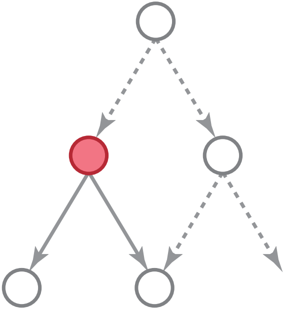


Invariant

At node v: Game state ρ, maintain ρ-like $R' \subseteq R_v$

2. Internal node

Crux!



Invariant

At node v: Game state ρ, maintain ρ-like $R' \subseteq R_v$

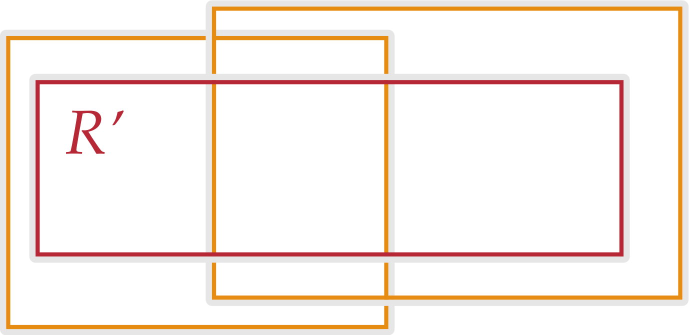
$$\rho = 0\ 1\ 1\ 0\ *\ *\ *\ *\ *\ *\ *\ *$$

R'

Invariant

At node v: Game state ρ, maintain ρ-like $R' \subseteq R_v$

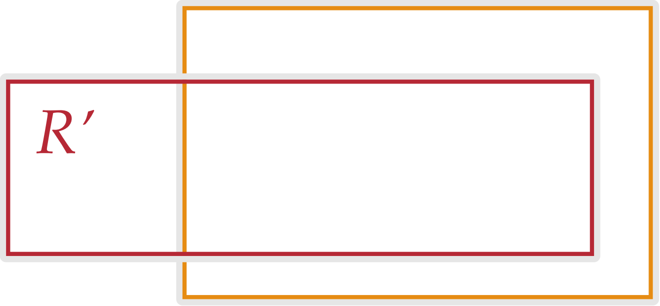
$$\rho = \mathbf{0\ 1\ 1\ 0} * * * * * * * *$$



Invariant

At node v: Game state ρ, maintain ρ-like $R' \subseteq R_v$

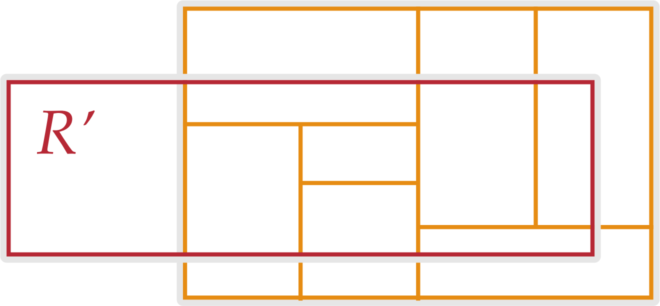
$$\rho = \mathbf{0\ 1\ 1\ 0} * * * * * * * *$$



Invariant

At node v: Game state ρ, maintain ρ-like $R' \subseteq R_v$

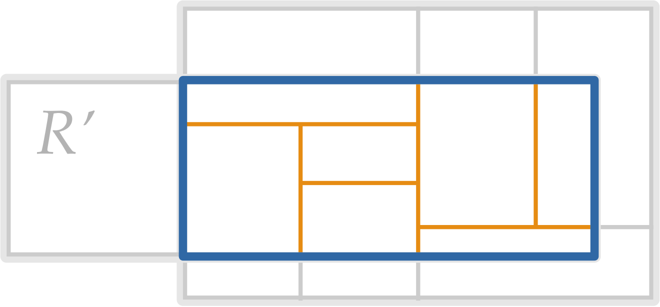
$\rho = \mathbf{0\ 1\ 1\ 0} * * * * * * * *$



Invariant

At node v: Game state ρ, maintain ρ-like $R' \subseteq R_v$

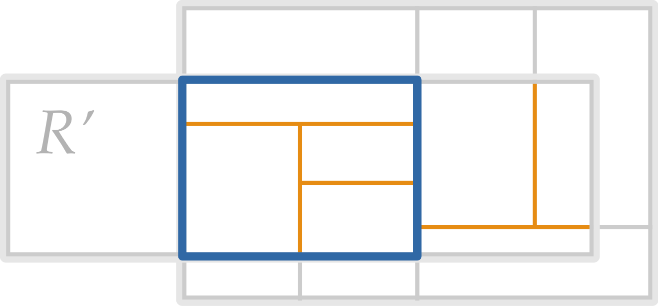
$\rho = \mathbf{0}\ \mathbf{1}\ \mathbf{1}\ \mathbf{0}\ *\ *\ *\ \mathbf{?}\ *\ *\ *\ *$



Invariant

At node v: Game state ρ, maintain ρ-like $R' \subseteq R_v$

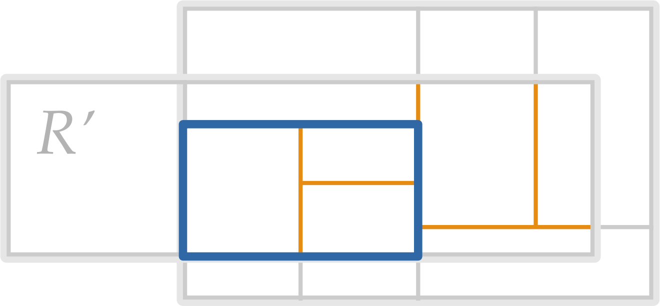
$\rho = 0\ 1\ 1\ 0\ *\ *\ *\ 1\ *\ *\ ?\ *$



Invariant

At node v: Game state ρ, maintain ρ-like $R' \subseteq R_v$

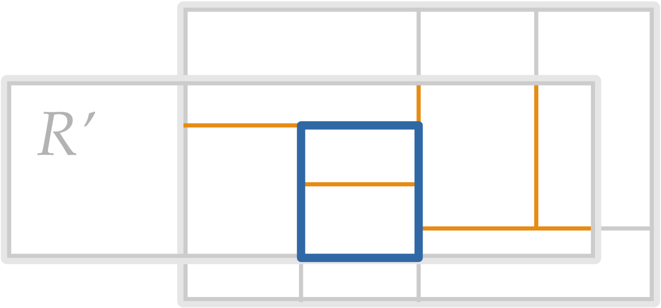
$$\rho = 0\ 1\ 1\ 0\ *\ *\ *\ 1\ ?\ *\ 0\ *$$



Invariant

At node v: Game state ρ, maintain ρ-like $R' \subseteq R_v$

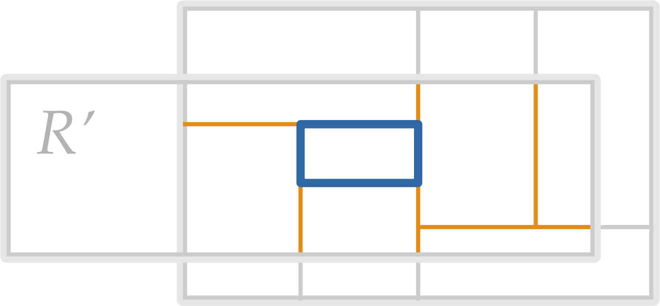
$\rho = 0\ 1\ 1\ 0\ *\ *\ *\ 1\ 0\ ?\ 0\ *$



Invariant

At node v: Game state ρ, maintain ρ-like $R' \subseteq R_v$

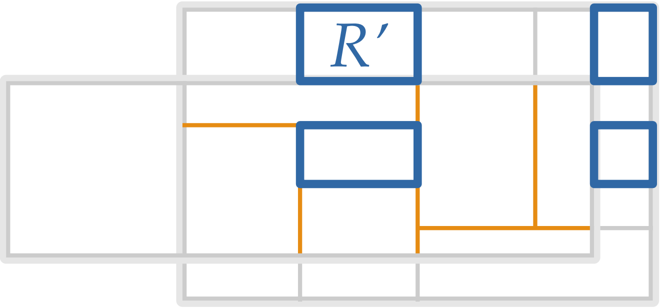
$\rho = \mathbf{0\ 1\ 1\ 0} * * * \mathbf{1\ 0\ 1\ 0} *$



Invariant

At node v: Game state ρ, maintain ρ-like $R' \subseteq R_v$

$$\rho = * * * * * * * 1 0 1 0 *$$



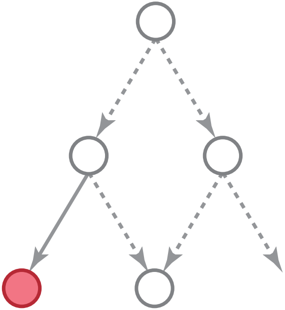
Invariant

At node v: Game state ρ, maintain ρ-like $R' \subseteq R_v$

3. Leaf

Leaf labeled with $o \in \mathcal{O}$
also valid for ρ

Game ends!



Invariant

At node v: Game state ρ, maintain ρ-like $R' \subseteq R_v$

3.2 Simplified proof

To explain the basic idea, we first give a simplified version of the proof: We assume that all rectangles R involved in Π—call them the *original* rectangles—can be partitioned *errorlessly* into ρ-structured subrectangles for ρ of width $O(d)$. That is, invoking the 2-round partitioning scheme for each original R, we assume that

(⋆) *Assumption:* All subrectangles R^i in the resulting partition $R = \bigcup_i R^i$ satisfy the "structured" case of Lemma 5 for $k := 2d \log n$.

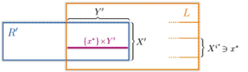
In Section 3.3 we remove this assumption by explaining how the proof can be modified to work in the presence of some error rows/columns.

Overview. We extract a width-$O(d)$ Explorer-strategy for S by walking down the rectangle-dag Π, starting at the root. For each original rectangle R that is reached in the walk, we maintain a ρ-structured subrectangle $R' \subseteq R$ chosen from the 2-round partition of R. Note that ρ will have width $O(d)$ by our choice of k. The intention is that ρ will record the current state of the game. There are three issues to address: (1) Why is the starting condition of the game met? (2) How do we take a step from a node of Π to one of its children? (3) Why are we done once we reach a leaf?

(1) Root case. At start, the root of Π is associated with the original rectangle $R = [m]^n \times \{0,1\}^{mn}$ comprising the whole domain. The 2-round partition of R is trivial: it contains a single part, the $*^n$-structured R itself. Hence we simply maintain the $*^n$-structured $R \subseteq R$, which meets the starting condition for the game.

(2) Internal step. This is the crux of the argument: Supposing the game has reached state $\rho_{R'}$ and we are maintaining some $\rho_{R'}$-structured subrectangle $R' \subseteq R$ associated with an internal node v, we want to move to some $\rho_{L'}$-structured subrectangle $L' \subseteq L$ associated with a child of v. Moreover, we must keep the width of the game state at most $O(d)$ during this move.

Let the two original rectangles associated with the children of v be L and L^*. Because $R' \subseteq L \cup L^*$ at least one of L and L^*, say L, covers at least half of R. That is, the rectangle $X' \times Y' := R' \cap L$ has density $\geq 1/2$ inside R'. Since R' satisfies the "structured" case in Lemma 5 we know that $X' \times Y' = R' \cap L$ is still $\rho_{R'}$-structured. By Lemma 4 there exists some $x^* \in X'$ such that $\{x^*\} \times Y'$ is $\rho_{R'}$-like. Let the partition of L according to the 2-round scheme be $L = \bigcup_{i,z} X^i \times Y^{i,z}$. Let i^* be the unique index such that $x^* \in X^{i^*}$. Recall that X^{i^*} is associated with some subset of blocks $I^* \subseteq [n]$ such that all parts of the form $X^{i^*} \times Y^{i^*,z}$ are ρ-structured with fix $\rho = I^*$. In particular, we have $|I^*| \leq O(d)$.



As Explorer, we now query the input bits in coordinates $J := I^* \setminus \text{fix}\,\rho_{R'}$ (in any order) obtaining some response string $z_J \in \{0,1\}^J$ from the Adversary. As a result, the state of the game becomes the extension of $\rho_{R'}$ by z_J, call it ρ^*, which has width $|\text{fix}\,\rho^*| = |\text{fix}\,\rho_{R'} \cup J| \leq O(d)$.

Note that there is some $y^* \in Y'$ (and hence some $(x^*, y^*) \in R' \cap L$) such that $G(x^*, y^*)$ is consistent with ρ^*; indeed, the whole row $\{x^*\} \times Y'$ is $\rho_{R'}$-like and ρ^* extends $\rho_{R'}$. In the partition of L, let $L' := X^{i^*} \times Y^{i^*,z^*}$ be the unique part such that $(x^*, y^*) \in L'$. Note that L' is $\rho_{L'}$-like for some $\rho_{L'}$ that is consistent with $G(x^*, y^*)$ and fix $\rho_{L'} = I^*$. Hence ρ^* extends $\rho_{L'}$. As Explorer, we now forget all queried bits in ρ^* except those queried in $\rho_{L'}$.

We have recovered our invariant: the game state is $\rho_{L'}$ and we maintain a $\rho_{L'}$-structured subrectangle L' of an original rectangle L. Moreover, the width of the game state remained $O(d)$.

(3) Leaf case. Suppose the game state is ρ and we are maintaining an associated ρ-structured subrectangle $R' \subseteq R$ corresponding to a *leaf* node. The leaf node is labeled with some solution $o \in \mathcal{O}$ satisfying $R' \subseteq (S \circ G)^{-1}(o)$, that is, $G(R') \subseteq S^{-1}(o)$. But $G(R') = C_\rho^{-1}(1)$ by Lemma 3 so that $C_\rho^{-1}(1) \subseteq S^{-1}(o)$. Therefore the game ends. This concludes the (simplified) proof.

3.3 Accounting for error

Next, we explain how to get rid of the assumption (⋆) by accounting for the rows and columns that are classified as error in Lemma 5 for $k := 2d \log n$. The partitioning of Π's rectangles is done more carefully: We sort all original rectangles in *reverse topological order* $R_1, R_2, \ldots, R_{n^d}$ from leaves to root, that is, if R_i is a descendant of R_j then R_i comes before R_j in the order. Then we process the rectangles in this order:

Initialize cumulative error sets $X^*_{\text{err}} = Y^*_{\text{err}} := \emptyset$. *Iterate for* $i = 1, 2, \ldots, n^d$ rounds:

1. Remove from R_i the rows/columns $X^*_{\text{err}}, Y^*_{\text{err}}$. That is, update
$$R_i \leftarrow R_i \setminus (X^*_{\text{err}} \times \{0,1\}^{mn} \cup [m]^n \times Y^*_{\text{err}}).$$
2. Run the 2-round partitioning scheme for R_i. Output all resulting subrectangles that satisfy the "structured" case of Lemma 5 for $k := 2d \log n$. (All non-structured subrectangles are omitted). Call the resulting error rows/columns X_{err} and Y_{err}.
3. Update $X^*_{\text{err}} \leftarrow X^*_{\text{err}} \cup X_{\text{err}}$ and $Y^*_{\text{err}} \leftarrow Y^*_{\text{err}} \cup Y_{\text{err}}$.

In words, an original rectangle R_i is processed only after all of its descendants are partitioned. Each descendant may contribute some error rows/columns, accumulated into sets $X^*_{\text{err}}, Y^*_{\text{err}}$, which are deleted from R_i before it is partitioned. The partitioning of R_i will in turn contribute its error rows/columns to its ancestors.

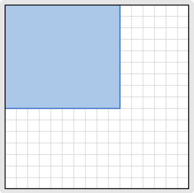
We may now repeat the proof of Section 3.2 *using only the structured subrectangles output by the above process.* We highlight two key properties that allow the proof to go through verbatim.

- First, the cumulative error at the end of the process is tiny: $X^*_{\text{err}}, Y^*_{\text{err}}$ have density at most $n^d \cdot n^{-2d} \leq 1/4$ by a union bound over all rounds. In particular, the root rectangle R_{n^d} (with errors removed) still has density $\geq 1/2$ inside $[m]^n \times \{0,1\}^{mn}$ and so it is $*^n$-structured. This allows us to meet the starting condition for the game.
- Second, by construction, the cumulative error sets grow as we walk from leaves towards the root. This means that our error handling does not interfere with the internal step: each structured subrectangle R' of an original rectangle R is covered by the structured subrectangles of R's children.

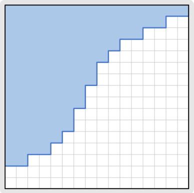
Open problems

Open problems

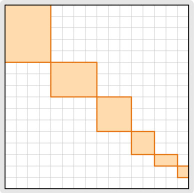
Q1. Lifting for dags over *intersections-of-k-triangles*
(Resolution over Cutting Planes)



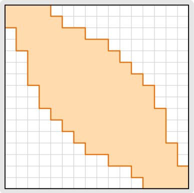
Rectangle



Triangle



Block-diagonal



Intersection of 2 triangles

Open problems

Q1. Lifting for dags over *intersections-of-k-triangles*
(Resolution over Cutting Planes)

Q2. Lifting for *nondeterministic* NOF protocols
(Towards dag-like LBs for semi-algebraic proof systems)

Q3. Superlinear depth for small monotone circuits?
(Razborov'16: "A New Kind of Tradeoff")

Open problems

Q1. Lifting for dags over *intersections-of-k-triangles*
(Resolution over Cutting Planes)

Q2. Lifting for *nondeterministic* NOF protocols
(Towards dag-like LBs for semi-algebraic proof systems)

Q3. Superlinear depth for small monotone circuits?
(Razborov'16: "A New Kind of Tradeoff")

Cheers!