Adventures in Monotone Complexity and TFNP

Mika Göös
Institute for Advanced Study, Princeton, NJ, USA
mika@ias.edu

Pritish Kamath
Massachusetts Institute of Technology, Cambridge, MA, USA
pritish@mit.edu

Robert Robere
Simons Institute, Berkeley, CA, USA
robere@cs.toronto.edu

Dmitry Sokolov
KTH Royal Institute of Technology, Stockholm, Sweden
sokolov.dmt@gmail.com

Abstract

Separations: We introduce a monotone variant of XOR-SAT and show it has exponential monotone circuit complexity. Since XOR-SAT is in NC^2, this improves qualitatively on the monotone vs. non-monotone separation of Tardos (1988). We also show that monotone span programs over R can be exponentially more powerful than over finite fields. These results can be interpreted as separating subclasses of TFNP in communication complexity.

Characterizations: We show that the communication (resp. query) analogue of PPA (subclass of TFNP) captures span programs over F_2 (resp. Nullstellensatz degree over F_2). Previously, it was known that communication FP captures formulas (Karchmer–Wigderson, 1988) and that communication PLS captures circuits (Razborov, 1995).

2012 ACM Subject Classification Theory of computation → Communication complexity, Theory of computation → Circuit complexity, Theory of computation → Proof complexity

Keywords and phrases TFNP, Monotone Complexity, Communication Complexity, Proof Complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2019.38


Acknowledgements We thank Ankit Garg (who declined a co-authorship) for extensive discussions about monotone circuits. We also thank Thomas Watson and anonymous ITCS reviewers for comments.

1 Work done while at Harvard University; supported by the Michael O. Rabin Postdoctoral Fellowship.
2 Supported in parts by NSF grants CCF-1650733, CCF-1733808, and IIS-1741137.
3 Work done while at University of Toronto; supported by NSERC.
4 Supported by the Swedish Research Council grant 2016-00782, Knut and Alice Wallenberg Foundation grants KAW 2016.0066 and KAW 2016.0433.

© Mika Göös, Pritish Kamath, Robert Robere, and Dmitry Sokolov; licensed under Creative Commons License CC-BY
1 Our Results

We study the complexity of monotone boolean functions \( f: \{0, 1\}^n \to \{0, 1\} \), that is, functions satisfying \( f(x) \leq f(y) \) for every pair \( x \leq y \) (coordinate-wise). (An excellent introduction to monotone complexity is the textbook [36].) Our main results are new separations of monotone models of computation and characterizations of those models in the language of query/communication complexity. At the core of these results are two conceptual innovations.

1. We introduce a natural monotone encoding of the usual constraint satisfaction problem (Subsection 1.1). This definition unifies many other monotone functions considered in the literature.

2. We extend and make more explicit an intriguing connection between circuit complexity and total NP search problems (TFNP) via communication complexity. Several prior characterizations [37, 55] can be viewed in this light. This suggests a potentially useful organizational principle for circuit complexity measures; see Section 2 for our survey.

1.1 Monotone C-Sat

The basic conceptual insight in this work is a new simple definition: a monotone encoding of the usual constraint satisfaction problem (CSP). For any finite set of constraints \( C \), we introduce a monotone function \( C\text{-Sat} \). A general definition is given in Section 3, but for now, consider as an example the set \( C = 3\text{Xor} \) of all ternary parity constraints:

\[
3\text{Xor} := \{(v_1 \oplus v_2 \oplus v_3 = 0), (v_1 \oplus v_2 \oplus v_3 = 1)\}.
\]

We define \( 3\text{Xor-Sat}_n: \{0, 1\}^N \to \{0, 1\} \) over \( N := |C|n^3 = 2n^3 \) input bits as follows. An input \( x \in \{0, 1\}^N \) is interpreted as (the indicator vector of) a set of \( 3\text{Xor} \) constraints over \( n \) boolean variables \( v_1, \ldots, v_n \) (there are \( N \) possible constraints). We define \( 3\text{Xor-Sat}_n(x) := 1 \) iff the set \( x \) is unsatisfiable, that is, no boolean assignment to the \( v_i \) exists that satisfies all constraints in \( x \). This is indeed a monotone function: if we flip any bit of \( x \) from 0 to 1, this means we are adding a new constraint to the instance, thereby making it even harder to satisfy.

Prior work. Our \( C\text{-Sat} \) encoding generalizes several previously studied monotone functions.

(NL) Karchmer and Wigderson [37] (also [29, 49, 56] and textbooks [39, 36]) studied the \( \text{NL} \)-complete \textit{st-connectivity} problem. This is equivalent to a \( C\text{-Sat} \) problem with \( C \) consisting of a binary implication \( (v_1 \rightarrow v_2) \) and unit clauses \( (v_1) \) and \( (\neg v_1) \).

(P) Raz and McKenzie [52] (also [14, 15, 25, 18, 56, 48]) studied a certain \( \text{P} \)-complete \textit{generation} problem. In hindsight, this is simply \( \text{HORN-SAT} \), that is, \( C \) consists of \textit{Horn clauses}: clauses with at most one positive literal, such as \( (\neg v_1 \lor \neg v_2 \lor v_3) \).

(NP) Göös and Pitassi [25] and Oliveira [45, §3] (also [47, 48]) studied the \( \text{NP} \)-complete (dual of) \( \text{CNF-SAT} \) problem, where \( C \) consists of bounded-width clauses.

These prior works do not exhaust all interesting classes of \( C \), as is predicted by various classification theorems for CSPs [59, 20, 11, 63]. In this work, we focus on \textit{linear} constraints over finite fields \( \mathbb{F}_p \) (for example, \( 3\text{Xor-Sat} \) corresponding to \( \mathbb{F}_2 \)) and over the reals \( \mathbb{R} \).

1.2 Separations

First, we show that \( 3\text{Xor-Sat}_n \) cannot be computed efficiently with monotone circuits.

\begin{theorem}
3\text{Xor-Sat}_n \text{ requires monotone circuits of size } 2^{n^{\Omega(1)}}.
\end{theorem}
This theorem stands in contrast to the fact that there exist fast parallel (non-monotone) algorithms for linear algebra [44]. In particular, 3Xor-Sat is in NC^2. Consequently, our result improves qualitatively on the monotone vs. non-monotone separation of Tardos [61] who exhibited a monotone function in P (computed by solving a semidefinite program) with exponential monotone circuit complexity. For further comparison, another famous candidate problem to witness a monotone vs. non-monotone separation is the perfect matching function: it is in RNC^2 [40] while it is widely conjectured to have exponential monotone circuit complexity (a quasipolynomial lower bound was proved by Razborov [53]).

Span programs. The computational easiness of 3Xor-Sat_n can be stated differently: it can be computed by a linear-size monotone F_2-span program. Span programs are a model of computation introduced by Karchmer and Wigderson [38] (see also [36, §8] for exposition) with an extremely simple definition. An F-span program, where F is a field, is a matrix M ∈ F^{m×m′} each row of which is labeled by a literal, x_i or ¬x_i. We say that the program accepts an input x ∈ {0, 1}^n iff the rows of M whose labels are consistent with x (literals evaluating to true on x) span the all-1 row vector. The size of a span program is its number of rows m. A span program is monotone if all its literals are positive; in this case the program computes a monotone function.

A corollary of Theorem 1 is that monotone F_2-span programs cannot be simulated by monotone circuits without exponential blow-up in size. This improves on a separation of Babai, Gál, and Wigderson [3] who showed that monotone circuit complexity can be quasipolynomially larger than monotone F_2-span program size.

Furthermore, Theorem 1 holds more generally over any field F: an appropriately defined function 3Lin(F)-Sat_n (ternary F-linear constraints; see Section 3) is easy for monotone F-span programs, but exponentially hard for monotone circuits. No such separation, even superpolynomial, was previously known for fields of characteristic other than 2.

This brings us to our second theorem.

▶ Theorem 2. 3Lin(R)-Sat_n requires monotone F_p-span programs of size 2^{n^{Ω(1)}} for any prime p.

In other words: monotone R-span programs can be exponentially more powerful than monotone span programs over finite fields. This separation completes the picture for the relative powers of monotone span programs over distinct fields, since the remaining cases were exponentially separated by Pitassi and Robere [48].

Finally, our two results above yield a bonus result in proof complexity as a byproduct: the Nullstellensatz proof system (over any field) can be exponentially more powerful than the Cutting Planes proof system (see Subsection 4.2).

Techniques. The new lower bounds are applications of the lifting theorems for monotone circuits [23] and monotone span programs [48]. We show that, generically, if some unsatisfiable formula composed of C constraints is hard to refute for the Resolution (resp. Nullstellensatz) proof system, then the C-Sat problem is hard for monotone circuits (resp. span programs). Hence we can invoke (small modifications of) known Resolution and Nullstellensatz lower bounds [8, 10, 1]. The key conceptual innovation here is a reduction from unsatisfiable C-CSPs (or their lifted versions) to the monotone Karchmer–Wigderson game for C-Sat. This reduction is extremely slick, which we attribute to having finally found the “right” definition of C-Sat.
1.3 Characterizations

There are two famous “top-down” characterizations of circuit models (both monotone and non-monotone variants) using the language of communication complexity; these characterizations are naturally related to communication analogues of subclasses of TFNP.

(FP) Karchmer and Wigderson [37] showed that the logarithm of the (monotone) formula complexity of a (monotone) function $f: \{0,1\}^n \to \{0,1\}$ is equal, up to constant factors, to the communication complexity of the (monotone) Karchmer–Wigderson game:

| Search problem $KW(f)$ [resp. $KW^+(f)$] | input: a pair $(x, y) \in f^{-1}(1) \times f^{-1}(0)$ | output: an $i \in [n]$ with $x_i \neq y_i$ [resp. $x_i > y_i$] |

We summarize this by saying that the communication analogue of FP captures formulas. Here $FP \subseteq TFNP$ is the classical (Turing machine) class of total NP search problems efficiently solved by deterministic algorithms [42].

(PLS) Razborov [55] (see also [50, 60]) showed that the logarithm of the (monotone) circuit complexity of a function $f: \{0,1\}^n \to \{0,1\}$ is equal, up to constant factors, to the least cost of a PLS-protocol solving the $KW(f)$ (or $KW^+(f)$) search problem. Here a PLS-protocol (Definition 14 in Appendix A) is a natural communication analogue of $PLS \subseteq TFNP$ [35]. We summarize this by saying that the communication analogue of PLS captures circuits.

We contribute a third characterization of this type: the communication analogue of PPA captures $F_2$-span programs. The class PPA [46] is a well-known subclass of TFNP embodying the combinatorial principle “every graph with an odd degree vertex has another”. Informally, a search problem is in PPA if for every $n$-bit input $x$ we may describe implicitly an undirected graph $G_x = (V, E)$ (typically of size exponential in $n$; the edge relation is computed by a polynomial-size circuit) such that $G$ has degree at most 2, there is a distinguished degree-1 vertex $v^* \in V$, and every other degree-1 vertex $v \in V$ is associated with a feasible solution to the instance $x$ (that is, the solution can be efficiently computed from $v$).

Communication PPA. The communication analogue of PPA is defined canonically by letting the edge relation be computed by a (deterministic) communication protocol. Specifically, first fix a two-party search problem $S \subseteq S \times Y \times O$, that is, Alice gets $x \in S$, Bob gets $y \in Y$, and their goal is to find a feasible solution in $S(x, y) := \{o \in O : (x, y, o) \in S\}$. A PPA-protocol $\Pi$ solving $S$ consists of a vertex set $V$, a distinguished vertex $v^* \in V$, and for each vertex $v \in V$ there is an associated solution $o_v \in O$ and a protocol $\Pi_v$ (taking inputs from $S \times Y$). Given an input $(x, y)$, the protocols $\Pi_v$ implicitly describe a graph $G = G_{x,y}$.
on the vertex set $V$ as follows. The output of protocol $\Pi_v$ on input $(x, y)$ is interpreted as a subset $\Pi_v(x, y) \subseteq V$ of size at most 2. We define $\{u, v\} \in E(G)$ iff $u \in \Pi_v(x, y)$ and $v \in \Pi_u(x, y)$. The correctness requirements are:

(C1) if $\deg(v^*) \neq 1$, then $o_{v^*} \in S(x, y)$.

(C2) if $\deg(v) \neq 2$ for $v \neq v^*$, then $o_v \in S(x, y)$.

The cost of $\Pi$ is defined as $\log |V| + \max_v |\Pi_v|$ where $|\Pi_v|$ is the communication cost of $\Pi_v$.

Finally, define $\text{PPA}^{cc}(S)$ as the least cost of a PPA-protocol that solves $S$.

For a (monotone) function $f$, define $\text{SP}_f(f)$ (resp. $\text{mSP}_f(f)$) as the least size of a (monotone) $F$-span program computing $f$. Our characterization is in terms of $S = \text{KW}(f)$.

**Theorem 3.** For any boolean function $f$, we have $\log \text{SP}_f(f) = \Theta(\text{PPA}^{cc}(\text{KW}(f)))$. Furthermore, if $f$ is monotone, we have $\log \text{mSP}_f(f) = \Theta(\text{PPA}^{cc}(\text{KW}^+(f)))$.

**Query PPA.** Our second characterization concerns the Nullstellensatz proof system; see Section 3 for the standard definition. Span programs and Nullstellensatz are known to be connected via interpolation [51] and lifting [48]. Given our first characterization (Theorem 3), it is no surprise that a companion result should hold in query complexity: the query complexity analogue of PPA captures the degree of Nullstellensatz refutations over $\mathbb{F}_2$.

The query analogue of PPA is defined in the same way as the communication analogue, except we replace protocols by (deterministic) decision trees. In fact, query PPA was already studied by Beame et al. [6] who separated query analogues of different subclasses of TFNP.

To define it, first fix a search problem $S \subseteq \{0, 1\}^n \times \mathcal{O}$, that is, on input $x \in \{0, 1\}^n$ the goal is to find a feasible solution in $S(x) := \{o \in \mathcal{O} : (x, o) \in S\}$. A PPA-decision tree $T$ solving $S$ consists of a vertex set $V$, a distinguished vertex $v^* \in V$, and for each vertex $v \in V$ there is an associated solution $o_v \in \mathcal{O}$ and a decision tree $T_v$ (querying bits of an $n$-bit input). Given an input $x \in \{0, 1\}^n$, the decision trees $T_v$ implicitly describe a graph $G = G_x$ on the vertex set $V$ as follows. The output of $T_v$ on input $x$ is interpreted as a subset $T_v(x) \subseteq V$ of size at most 2. We then define $\{u, v\} \in E(G)$ iff $u \in T_v(x)$ and $v \in T_u(x)$. The correctness requirements are the same as before, 1 and 2. The cost of $T$ is defined as the maximum over all $v \in V$ and all inputs $x$ of the number of queries made by $T_v$ on input $x$. Finally, define $\text{PPA}^{dt}(S)$ as the least cost of a PPA-decision tree that solves $S$.

With any unsatisfiable $n$-variate boolean CSP $F$ one can associate a canonical search problem:

| CSP search problem $S(F)$ | input: an $n$-variate truth assignment $x \in \{0, 1\}^n$ output: constraint $C$ of $F$ falsified by $x$ (i.e., $C(x) = 0$) |

**Theorem 4.** The $\mathbb{F}_2$-Nullstellensatz degree of an $k$-CNF formula $F$ equals $\Theta(\text{PPA}^{dt}(S(F)))$.

The easy direction of this characterization is that Nullstellensatz degree lower bounds $\text{PPA}^{dt}$. This fact was already observed and exploited by Beame et al. [6] to prove lower bounds for $\text{PPA}^{dt}$. Our contribution is to show the other (less trivial) direction.

Let us finally mention a related result in Turing machine complexity due to Belovs et al. [9]: a circuit-encoded version of Nullstellensatz is PPA-complete. Their proof is highly nontrivial whereas our characterizations admit relatively short proofs, owing partly to us working with simple nonuniform models of computation.
Adventures in Monotone Complexity and TFNP

Figure 1: The landscape of communication search problem classes (uncluttered by the usual ‘cc’ superscripts). A solid arrow $C_1 \rightarrow C_2$ denotes $C_1 \subseteq C_2$, and a dashed arrow $C_1 \rightarrow \neg C_2$ (in fact, an exponential separation). Some classes can characterize other models of computation (printed in blue). See Appendix A for definitions.

2 Survey: Communication TFNP

Given the results in Section 1, it is natural to examine other communication analogues of subclasses of TFNP. The goal in this section is to explain the current state of knowledge as summarized in Figure 1. The formal definitions of the communication classes appear in Appendix A.

TFNP. As is customary in structural communication complexity [2, 30, 27] we formally define $\text{TFNP}^{cc}$ (resp. $\text{PLS}^{cc}$, $\text{PPA}^{cc}$, etc.) as the class of all total two-party $n$-bit search problems that admit a nondeterministic protocol$^5$ (resp. $\text{PLS}$-protocol, $\text{PPA}$-protocol, etc.) of communication cost $\text{polylog}(n)$. For example, Karchmer–Wigderson games $\text{KW}(f)$ and $\text{KW}^+(f)$, for an $n$-bit boolean function $f$, have efficient nondeterministic protocols: guess a log $n$-bit coordinate $i \in [n]$ and check that $x_i \neq y_i$ or $x_i > y_i$. Hence these problems are in $\text{TFNP}^{cc}$. In fact, a converse holds: any total two-party search problem with nondeterministic complexity $c$ can be reduced to $\text{KW}^+(f)$ for some $2^c$-bit partial monotone function $f$, see [21, Lemma 2.3]. In summary, the study of total $\text{NP}$ search problems in communication complexity is equivalent to the study of monotone Karchmer–Wigderson games for partial monotone functions.

$^5$ That is, for any input $(x, y)$, every accepting computation of the nondeterministic protocol outputs a feasible solution $o \in \mathcal{O}$ for $(x, y)$. An alternative, more restrictive definition of $\text{TFNP}^{cc}$ (which is closer to how the classical class is defined) is to require that there is an efficient deterministic protocol that on input $(xo, y)$ decides whether $o \in \mathcal{O}$ is feasible for $(x, y)$. In this paper we stick with the stronger (and simpler) definition for convenience. All results hold equally well under the more restrictive definition.
Sometimes a partial function $f$ can be canonically extended into a total one $f'$ without increasing the complexity of $KW(f)$ (or $KW^+(f)$). This is possible whenever $KW(f)$ lies in a communication class that captures some associated model of computation. For example, if $KW(f)$ is solved by a deterministic protocol (resp. PLS-protocol, PPA-protocol) then the Karchmer–Wigderson connection can build us a corresponding formula (resp. circuit, $F_2$-span program) that computes some total extension $f'$ of $f$. Consequently, separating two communication classes that capture two monotone models is equivalent to separating the monotone models themselves.

**FP.** Raz and McKenzie [52] showed an exponential separation between monotone formula size and monotone circuit size. This can be rephrased as $PLS^cc \not\subseteq FP^cc$. Their technique is much more general: they develop a query-to-communication lifting theorem for deterministic protocols (see also [26] for exposition). By plugging in known query complexity lower bounds against randomized protocols have applications in proof complexity [34, 7, 33, 25] and algorithmic game theory [58, 5, 28, 57, 22, 4]. In particular, some of these works (for finding Nash equilibria) have introduced a communication analogue of the PPAD-complete problem, which we will continue to study in Subsection 4.2.

**PLS.** Razborov’s [54] famous monotone circuit lower bound for the clique/coloring problem (which is in $PPP^cc$) can be interpreted as an exponential separation $PPP^cc \not\subseteq PLS^cc$. We show a stronger separation $PPAD^cc \not\subseteq PLS^cc$ using the End-of-Line problem in Subsection 4.2. Note that this is even slightly stronger than Theorem 1, which only implies $PPA^cc \not\subseteq PLS^cc$.

**PPApD.** In light of our characterization of $PPA^cc$, we may interpret the inability of monotone $F_2$-span program to efficiently simulate monotone circuits [48] as a separation $PLS^cc \not\subseteq PPA^cc$. We show an incomparable separation $PPADS^cc \not\subseteq PPA^cc$ in Subsection 4.3.

In the other direction, prior work implies $PPA^cc \not\subseteq PPAD^cc$ as follows. Pitassi and Robere [48] exhibit a monotone $f$ (in hindsight, one can take $f := 3\text{Xor-Sat}_n$) computable with a small monotone $F_2$-span program (hence $KW^+(f) \in PPA^cc$) and such that $KW^+(f)$ has an exponentially large $\mathbb{R}$-partition number (see Section 3 for a definition); however, we observe that all problems in $PPAD^cc$ have a small $\mathbb{R}$-partition number (see Remark 9).

**PPP.** There are no lower bounds against $PPP^cc$ for an explicit problem in $TFNP^cc$. However, we can show non-constructively the existence of $KW(f) \in TFNP^cc$ such that $KW(f) \not\in PPP^cc$, which implies $PPP^cc \neq TFNP^cc$. Indeed, we argue in Remark 8 that every $S$ reduces to $KW^+(3\text{Cnf-Sat}_N)$ over $N := \exp(O(PPP^cc(S)))$ variables. Applying this to $S := KW(f)$ for an $n$-bit $f$, we conclude that $f$ is a (non-monotone) projection of $3\text{Cnf-Sat}_N$ for $N := \exp(O(PPP^cc(KW(f))))$. In particular, if $KW(f) \in PPP^cc$ (i.e., $PPP^cc(KW(f)) \leq \text{polylog}(n)$), then $f$ is in non-uniform quasipoly-size NP. Therefore $KW(f) \not\in PPP^cc$ for a random $f$.

**EoML, SoML, and comparator circuits.** One prominent circuit model that currently lacks a characterization via a $TFNP^cc$ subclass is comparator circuits [41, 16]. These circuits are composed only of comparator gates (taking two input bits and outputting them in sorted order) and input literals (positive literals in the monotone case).
We can show an upper bound better than \( \text{PLS}^{cc} \) for comparator circuits. Indeed, we introduce a new class \( \text{SoML} \) generalizing \( \text{EoML} \) \cite{32, 19} as follows. Recall that \( \text{EoML} \) is the class of problems reducible to \( \text{End-of-Metered-Line} \): we are given a directed graph of in/out-degree at most 1 with a distinguished source vertex \( v^* \) (in-degree 0), and moreover, each vertex is labeled with an integer “meter” that is strictly decreasing along directed paths; a solution is any sink or source distinct from \( v^* \). The complete problem defining \( \text{SoML} \) is \( \text{Sink-of-Metered-Line} \), which is the same as \( \text{End-of-Metered-Line} \) except only sinks count as solutions. It is not hard (left as an exercise) to adapt the characterization of circuits via \( \text{PLS}^{cc} \) \cite{55, 50, 60} to show that \( \text{KW}(f) \) is in \( \text{SoML}^{cc} \) if \( f \) is computed by a small comparator circuit. However, we suspect that the converse (SoML-protocol for \( \text{KW}(f) \) implies a comparator circuit) is false.

2.1 Open problems

In query complexity, the relative complexities of TFNP subclasses are almost completely understood \cite{6, 12, 43}. In communication complexity, by contrast, there are huge gaps in our understanding as can be gleaned from Figure 1. For example:

1. There are no lower bounds against classes \( \text{PPADS}^{cc} \) and \( \text{PPA}^{cc} \) for an explicit problem in \( \text{TFNP}^{cc} \). For starters, show \( \text{PLS}^{cc} \not\subset \text{PPADS}^{cc} \) or \( \text{PPA}^{cc} \not\subset \text{PPADS}^{cc} \).

2. Find computational models captured by \( \text{EoML}^{cc} \), \( \text{SoML}^{cc} \), \( \text{PPAD}^{cc} \), \( \text{PPADS}^{cc} \), \( \text{PPA}^{cc} \).

3. Query-to-communication lifting theorems are known for \( \text{FP} \) \cite{52}, \( \text{PLS} \) \cite{23}, \( \text{PPA} \) \cite{48}. Prove more. (This is one way to attack Question 1 if proved for \( \text{PPADS} \).)

4. Prove more separations. For example, can our result \( \text{PPADS}^{cc} \not\subset \text{PPA}^{cc} \) be strengthened to \( \text{SoML}^{cc} \not\subset \text{PPA}^{cc} \)? This is closely related to whether monotone comparator circuits can be more powerful than monotone \( F_2 \)-span programs (no separation is currently known).

3 Preliminaries

\( \mathcal{C} \)-Sat. Fix an alphabet \( \Sigma \) (potentially infinite, e.g., \( \Sigma = \mathbb{R} \)). Let \( \mathcal{C} \) be a finite set of \( k \)-ary predicates over \( \Sigma \), that is, each \( C \in \mathcal{C} \) is a function \( C : \Sigma^k \to \{0, 1\} \). We define a monotone function \( \mathcal{C} \)-\( \text{Sat}_n : \{0, 1\}^N \to \{0, 1\} \) over \( N = |\mathcal{C}|n^k \) input bits as follows. An input \( x \in \{0, 1\}^N \) is interpreted as a \( \mathcal{C} \)-CSP instance, that is, \( x \) is (the indicator vector of) a set of \( \mathcal{C} \)-constraints, each applied to a \( k \)-tuple of variables from \( v_1, \ldots, v_n \). We define \( \mathcal{C} \)-\( \text{Sat}_n(x) := 1 \) iff the \( \mathcal{C} \)-CSP \( x \) is unsatisfiable: no assignment \( v \in \Sigma^n \) exists such that \( C(v) = 1 \) for all \( C \in x \).

For a field \( F \), we define \( k\text{Lin}(F) \) as the set of all \( F \)-linear equations of the form

\[
\sum_{i \in [k]} a_i v_i = a_0, \quad \text{where } a_i \in \{0, \pm1\}.
\]

In particular, we recover \( 3\text{Xor-Sat}_n \) defined in Section 1 essentially as \( 3\text{Lin}(F_2) \)-\( \text{Sat}_n \). We could have allowed the \( a_i \) to range over \( F \) when \( F \) is finite, but we stick with the above convention as it ensures that the set \( k\text{Lin}(\mathbb{R}) \) is always finite.

Boolean alphabets. We assume henceforth that all alphabets \( \Sigma \) contain distinguished elements \( 0 \) and \( 1 \). We define \( \mathcal{C}_{\text{bool}} \) to be the constraint set obtained from \( \mathcal{C} \) by restricting each \( C \in \mathcal{C} \) to the boolean domain \( \{0, 1\}^k \subseteq \Sigma^k \). Moreover, if \( F \) is a \( \mathcal{C} \)-CSP, we write \( F_{\text{bool}} \) for the \( \mathcal{C}_{\text{bool}} \)-CSP obtained by restricting the constraints of \( F \) to boolean domains. Consequently, any \( S(F_{\text{bool}}) \) associated with a \( \mathcal{C} \)-CSP \( F \) is a boolean search problem.
Algebraic partitions. We say that a subset $A \subseteq X \times Y$ is monochromatic for a two-party search problem $S \subseteq X \times Y \times O$ if there is some $o \in O$ such that $o \in S(x,y)$ for all $(x,y) \in A$. Moreover, if $M \in \mathbb{F}^{X \times Y}$ is a matrix, we say $M$ is monochromatic if the support of $M$ is monochromatic. For any field $\mathbb{F}$, an $F$-partition of a search problem $S$ is a set $\mathcal{M}$ of rank-1 matrices $M \in \mathbb{F}^{X \times Y}$ such that $\sum_{M \in \mathcal{M}} M = 1$ and each $M \in \mathcal{M}$ is monochromatic for $S$. The size of the partition is $|\mathcal{M}|$. The $F$-partition number $\chi_{\mathbb{F}}(S)$ is the least size of an $\mathbb{F}$-partition of $S$. In the following characterization, recall that we use $\text{SP}_{\mathbb{F}}$ and $m\text{SP}_{\mathbb{F}}$ to denote (monotone) span program complexity.

\textbf{Theorem 5 ([21]).} For any boolean function $f$ and any field $\mathbb{F}$, $\text{SP}_{\mathbb{F}}(f) = \chi_{\mathbb{F}}(\text{KW}(f))$. Furthermore, if $f$ is monotone then $m\text{SP}_{\mathbb{F}}(f) = \chi_{\mathbb{F}}(\text{KW}^+(f))$.

Nullstellensatz. Let $P := \{p_1 = 0, p_2 = 0, \ldots, p_m = 0\}$ be an unsatisfiable system of polynomial equations in $\mathbb{F}[z_1, z_2, \ldots, z_n]$ for a field $\mathbb{F}$. An $F$-Nullstellensatz refutation of $P$ is a sequence of polynomials $q_1, q_2, \ldots, q_m \in \mathbb{F}[z_1, z_2, \ldots, z_n]$ such that $\sum_{i=1}^m q_i p_i = 1$ where the equality is syntactic. The degree of the refutation is $\max_i \deg(q_i p_i)$. The $F$-Nullstellensatz degree of $P$, denoted $\text{NS}_{\mathbb{F}}(P)$, is the least degree of an $F$-Nullstellensatz refutation of $P$.

Moreover, if $F$ is a $k$-CNF formula (or a boolean $k$-CSP), we often tacitly think of it as a polynomial system $P_F$ by using the standard encoding (e.g., $(z_1 \lor \neg z_2) \implies (1 - z_1) z_2 = 0$) and also including the boolean axioms $z^2 - z = 0$ in $P_F$ if we are working over $\mathbb{F} \neq \mathbb{F}_2$.

Lifting theorems. Let $S \subseteq \{0,1\}^n \times O$ be a boolean search problem and $g: X \times Y \to \{0,1\}$ a two-party function, usually called a gadget. The composed search problem $S \circ g^n \subseteq X^n \times Y^m \times O$ is defined as follows: Alice holds $x \in X^n$, Bob holds $y \in Y^m$, and their goal is to find an $o \in S(z)$ where $z := g^n(x,y) = (g(x_1, y_1), \ldots, g(x_n, y_n))$. We focus on the usual index gadget $\text{Ind}_m: [m] \times \{0,1\}^m \to \{0,1\}$ given by $\text{Ind}_m(x,y) := y_x$.

The main results of [23, 48] can be summarized as follows (we define more terms below).

\textbf{Theorem 6.} Let $k \geq 1$ be a constant and let $m = m(n) := n^C$ for a large enough constant $C \geq 1$. Then for any an unsatisfiable boolean $n$-variate $k$-CSP $F$,

\begin{align*}
\text{[23]:} & \quad \text{PLS}^{\text{st}}(S(F) \circ \text{Ind}_m^n) = \text{PLS}^{\text{st}}(S(F)) \cdot \Theta(\log n), \\
\text{[48]:} & \quad \text{PPA}^{\text{st}}(S(F) \circ \text{Ind}_m^n) = \text{PPA}^{\text{st}}(S(F)) \cdot \Theta(\log n), \\
\text{[48]:} & \quad \log \chi_{\mathbb{F}}(S(F) \circ \text{Ind}_m^n) = \text{NS}_{\mathbb{F}}(F) \cdot \Theta(\log n), \quad \forall \mathbb{F} \in \{\mathbb{F}_p, \mathbb{R}\}.
\end{align*}

For aesthetic reasons, we have used $\text{PLS}^{\text{st}}(S(F))$ here to denote the Resolution width of $F$ (introduced in [10]), which is how the result of [23] was originally stated. (But one can check that the query analogue of PLS, obtained by replacing protocols with decision trees in Definition 14, is indeed equivalent to Resolution width.) We also could not resist incorporating our new characterizations of $\text{PPA}^{\text{cc}}$ and $\text{PPA}^{\text{st}}$ to interpret the result of [48] specialized to $\mathbb{F}_2$.

4 Proofs of Separations

In this section, we show lower bounds for $\mathcal{C}$-SAT against monotone circuits (Theorem 1) and monotone span programs (Theorem 2), plus some bonus results ($\text{PPAD}^{\text{cc}} \not\subseteq \text{PLS}^{\text{cc}}$, $\text{PPADS}^{\text{cc}} \not\subseteq \text{PPA}^{\text{cc}}$, Nullstellensatz degree vs. Cutting Planes).
4.1 Reduction

The key to our lower bounds is a new reduction. We show that a lifted version of \( S(F_{\text{bool}}) \), where \( F \) is an unsatisfiable \( \mathcal{C} \)-CSP, reduces to the monotone Karchmer–Wigderson game for \( \mathcal{C} \)-Sat. Note that we require \( F \) to be unsatisfiable over its original alphabet \( \Sigma \), but the reduction is from the booleanized (and hence easier-to-refute) version of \( F \).

**Lemma 7.** Let \( F \) be an unsatisfiable \( \mathcal{C} \)-CSP. Then \( S(F_{\text{bool}}) \circ \text{IND}^n_m \) reduces to \( KW^+(\mathcal{C} \text{-Sat}_{mn}) \).

**Proof.** Suppose the \( \mathcal{C} \)-CSP \( F \) consists of \( k \)-ary constraints \( C_1, \ldots, C_t \) applied to variables \( z_1, \ldots, z_n \). We reduce \( S(F_{\text{bool}}) \circ \text{IND}^n_m \subseteq [m]^n \times \{(0,1)^m\}^n \times [t] \) to the problem \( KW^+(f) \subseteq f^{-1}(1) \times f^{-1}(0) \times [N] \) where \( f := \mathcal{C} \text{-Sat}_{mn} \) over \( N := |\mathcal{C}|(mn)^k \) input bits. The two parties compute locally as follows.

**Alice:** Given \((x_1, \ldots, x_n) \in [m]^n\), Alice constructs a \( \mathcal{C} \)-CSP over variables \( \{v_{i,j} : (i,j) \in [n] \times [m]\} \) that is obtained from \( F \) by renaming its variables \( z_1, \ldots, z_t \) to \( v_{1,x_1}, \ldots, v_{n,x_n} \) (in this order). Since \( F \) was unsatisfiable, so is Alice’s variable-renamed version of it.

Thus, when interpreted as an indicator vector of constraints, Alice has constructed a 1-input of \( \mathcal{C} \text{-Sat}_{mn} \).

**Bob:** Given \( y \in \{(0,1)^m\}^n \), Bob constructs a \( \mathcal{C} \)-CSP over variables \( \{v_{i,j} : (i,j) \in [n] \times [m]\} \) as follows. We view \( y \) naturally as a boolean assignment to the variables \( v_{i,j} \). Bob includes in his \( \mathcal{C} \)-CSP instance all possible \( \mathcal{C} \)-constraints \( C \) applied to the \( v_{i,j} \) such that \( C \) is satisfied under the assignment \( y \) (i.e., \( C(y) = 1 \)). This is clearly a satisfiable \( \mathcal{C} \)-CSP instance, as the assignment \( y \) satisfies all Bob’s constraints. Thus, when interpreted as an indicator vector of constraints, Bob has constructed a 0-input of \( \mathcal{C} \text{-Sat}_{mn} \).

It remains to argue that any solution to \( KW^+(\mathcal{C} \text{-Sat}_{mn}) \) gives rise to a solution to \( S(F_{\text{bool}}) \circ \text{IND}^n_m \). Indeed, a solution to \( KW^+(\mathcal{C} \text{-Sat}_{mn}) \) corresponds to a \( \mathcal{C} \)-constraint \( C \) that is present in Alice’s \( \mathcal{C} \)-CSP but not in Bob’s. By Bob’s construction, such a \( C \) must be violated by the assignment \( y \) (i.e., \( C(y) = 0 \)). Since all Alice’s constraints involve only variables \( v_{1,x_1}, \ldots, v_{n,x_n} \), the constraint \( C \) must in fact be violated by the partial assignment to the said variables, which is \( z = \text{IND}^n_m(x,y) \). Thus the constraint of \( F \) from which \( C \) was obtained via renaming is a solution to \( S(F_{\text{bool}}) \circ \text{IND}^n_m \).

**Remark 8.** (Generic reduction to CNF-Sat). We claim that any problem \( S \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{O} \) that lies in one of the known subclasses of TFNP\(^{\mathsf{cc}}\) (as listed in Section 2) reduces efficiently to \( KW^+(k\text{CNF-Sat}_{mn}) \) for constant \( k \) (one can even take \( k = 3 \) by standard reductions). Let us sketch the argument for \( S \in \text{PPP}^{\mathsf{cc}} \); after all, better reductions are known for PLS\(^{\mathsf{cc}}\) and PPA\(^{\mathsf{cc}}\), namely to HORNSAT and 3XOR-SAT.

**Proof Sketch.** Let \( \Pi := (V, \{v^s\}, o_n, \Pi_c) \) be a PPL-protocol solving \( S \) of cost \( c := \text{PPP}^{\mathsf{cc}}(S) \). We may assume wlog that all the \( \Pi_c \) have constant communication cost \( k \leq O(1) \) by embedding the protocol trees of the \( \Pi_c \) as part of the implicitly described bipartite graph. In particular, we view each \( \Pi_c \) as a function \( X \times Y \rightarrow \{0,1\}^k \) where the output is interpreted according to some fixed map \( \{0,1\}^k \rightarrow V \). Consider a set of \( n := k|V| \) (where \( |V| \leq 2^k \)) boolean variables \( \{z_{v,i} : (v,i) \in V \times [k]\} \) with the intuitive interpretation that \( z_{v,i} \) is the \( i \)-th output bit of \( \Pi_c \). We may encode the correctness conditions for \( \Pi \) as an unsatisfiable \( 2k \)-CNF formula \( F \) over the \( z_{v,i} \) that has, for each \( \{v,u\} \in \binom{V}{2} \), clauses requiring that the outputs of \( \Pi_c \) and \( \Pi_u \) (as encoded by the \( z_{v,i} \)) should point to distinct vertices. Finally, we note that computing the \( i \)-th output bit (\( \Pi_{v,i} \)) reduces to a large enough constant-size index gadget \( \text{IND}_{O(k)} \) (which embeds any two-party function of communication complexity \( k \leq O(1) \)). Therefore \( S \) naturally reduces to \( S(F) \circ \text{IND}_{O(1)} \), which by Lemma 7 reduces to \( KW^+(2k\text{CNF-Sat}_{O(n)}) \), as desired.
4.2 Monotone circuit lower bounds

Xor-Sat. The easiest result to prove is Theorem 1: an exponential monotone circuit lower bound for $3\text{Xor-Sat}_n$. By the characterization of [55] it suffices to show

$$\text{PLS}^c(\text{KW}^+(3\text{Xor-Sat}_n)) \geq n^{\Omega(1)}. \quad (1)$$

Urquhart [62] exhibited unsatisfiable $n$-variable $3\text{Xor-CSPs}$ $F$ (aka Tseitin formulas) requiring linear Resolution width, that is, $\text{PLS}^c(S(F)) \geq \Omega(n)$ in our notation. Hence Theorem 6 implies that $\text{PLS}^c(S(F) \circ \text{Ind}_{n^m}) \geq \Omega(n)$ for some $m = n^{O(1)}$. By the reduction in Lemma 7, we get that $\text{PLS}^c(\text{KW}^+(3\text{Xor-Sat}_{nm})) \geq \Omega(n)$. (Note that 3Xor has a boolean alphabet, so $F = F_{\text{bool}}$.) This yields the claim (1) by reparameterizing the number of variables.

Lin(F)-Sat. More generally, we can prove a similar lower bound over any field $F \in \{F_p, \mathbb{R}\}$:

$$\text{PLS}^c(\text{KW}^+(3\text{Lin}(F)-\text{Sat}_n)) \geq n^{\Omega(1)}. \quad (2)$$

Fix such an $F$ henceforth. This time we start with a $k\text{Lin}(F)$-CSP introduced in [13] for $F = F_p$ (aka mod-$p$ Tseitin formulas), but the definition generalizes to any field. The CSP is constructed based on a given directed graph $G = (V, E)$ that is regular: $\text{in-deg}(v) = \text{out-deg}(v) = k/2$ for all $v \in V$. Fix also a distinguished vertex $v^* \in V$. Then $F = F_{G,F}$ is defined as the following $k\text{Lin}(F)$-CSP over variables $\{z_e : e \in E\}$:

$$\forall v \in V : \sum_{(v,u) \in E} z_{(v,u)} - \sum_{(u,v) \in E} z_{(u,v)} = 1_{v^*}(v), \quad (F_{G,F})$$

where $1_{v^*}(v^*) = 1$ and $1_{v^*}(v) = 0$ for $v \neq v^*$. This system is unsatisfiable because the sum over $v \in V$ of the RHS equals 1 whereas the sum of the LHS equals 0 (each variable appears once with a positive sign, once with a negative sign).

We claim that the booleanized $k$-CSP $F_{\text{bool}}$ (more precisely, its natural $k$-CNF encoding) has linear Resolution width, that is, $\text{PLS}^c(S(F_{\text{bool}})) \geq \Omega(n)$ in our notation. Indeed, the constraints of $F_{\text{bool}}$ are $k/2$-robust in the sense that if a partial assignment $\rho \in \{0, 1, \ast\}^k$ fixes the value of a constraint of $F_{\text{bool}}$, then $\rho$ must set more than $k/2$ variables. Alekhnovich et al. [1, Theorem 3.1] show that if $k$ is a large enough constant, there exist regular expander graphs $G$ such that $F_{\text{bool}}$ (or any $k$-CSP with $\Omega(k)$-robust constraints) has Resolution width $\Omega(n)$, as desired.

Combining the above with the lifting theorem in Theorem 6 and the reduction in Lemma 7 yields $\text{PLS}^c(k\text{Lin}(F)-\text{Sat}_n) \geq n^{\Omega(1)}$ for large enough $k$. Finally, we can reduce the arity from $k$ to 3 by a standard trick. For example, given the linear constraint $a_1 v_1 + a_2 v_2 + a_3 v_3 + a_4 v_4 = a_0$ we can introduce a new auxiliary variable $u$ and two equations $a_1 v_1 + a_2 v_2 + u = 0$ and $-u + a_3 v_3 + a_4 v_4 = a_0$. In general, we replace each equation on $k > 3$ variables with a collection of $k - 3$ equations by introducing $k - 3$ auxiliary variables to create an equisatisfiable instance. This shows that $k\text{Lin}(F)$-Sat$_n$ reduces to (i.e., is a monotone projection of) $3\text{Lin}(F)$-Sat$_{kn}$, which concludes the proof of (2).

PPAD$^c \not\subseteq$ PLSc via End-of-Line. Consider the R-linear system $F = F_{G,R}$ defined above. We observe that $S(F_{\text{bool}})$ is in fact equivalent to (a query version of) the PPAD-complete End-of-Line problem. In the End-of-Line problem, we are given a directed graph of in/out-degree at most 1 and a distinguished source vertex $v^*$ (in-degree 0); the goal is to find a sink or a source distinct from $v^*$ (cf. Definition 15). On the other hand, in $S(F_{\text{bool}})$
we are given a boolean assignment \( z \in \{0, 1\}^E \), which can be interpreted as (the indicator vector of) a subset of edges defining a (spanning) subgraph \( G_z \) of \( G \); the goal is to find a vertex \( v \in V \) such that either

1. \( v = v^* \) and out-deg\((v) \neq \text{in-deg}(v) + 1 \) in \( G_z \); or
2. \( v \neq v^* \) and out-deg\((v) \neq \text{in-deg}(v) \) in \( G_z \).

The only essential difference between \( S(F_{\text{bool}}) \) and END-OF-LINE is that the graph \( G_z \) can have in/out-degree a large constant \( k/2 \) rather than 1. But there is a standard reduction between the two problems [46]: we may locally interpret a vertex \( v \in V(G_z) \) with out-deg\((v) = \text{in-deg}(v) = \ell \) as \( \ell \) distinct vertices of in/out-degree 1. This reduction also shows that the lifted problem \( S(F_{\text{bool}}) \circ \text{IND}_m \) for \( m = n^{O(1)} \) admits a \( O(\log n) \)-cost PPAD-protocol, and is thus in PPAD\(^{cc} \). By contrast, we proved above that this problem is not in PLSC\(^c \) for appropriate \( G \).

**Remark 9 (Algebraic partitions for PPAD\(^{cc} \)).** We claim that every problem \( S \in \text{PPAD}^{cc} \) admits a small \( Z \)-partition, and hence a small \( F \)-partition over any field \( F \). More precisely, we argue that \( \log_v \chi_Z(S) \leq O(\text{PPAD}^{cc}(S)) \). Indeed, let \( \Pi = (V, v^*, \alpha_v, \Pi_v) \) be an optimal PPAD-protocol for \( S \). We define a \( Z \)-partition \( M \) by describing it as a nondeterministic protocol for \( S \) whose accepting computations output weights in \( Z \) (interpreted as values of the entries of an \( M \in M \)): On input \((x, y)\), guess a vertex \( v \in V \); if \( v \) is a sink in \( G_{x, y} \), accept with weight 1; if \( v \) is a source distinct from \( v^* \), accept with weight \(-1\); otherwise reject (i.e., weight \( 0 \)). This protocol accepts with overall weight \#(sinks) \(-\#(\text{non-distinguished sources}) = 1 \) on every input \((x, y)\), as desired.

A similar argument yields an analogous query complexity bound \( \text{NS}_Z(F) \leq O(\text{PPAD}^{R}(S(F))) \) where \( \text{PPAD}^{R}(S) \) is the least cost of a PPAD-decision tree (Definition 15) solving \( S \).

**Nullstellensatz vs. Cutting Planes.** By the above remark, \( F_{\text{bool}} \) for \( F = F_{G, \mathbb{F}} \) admits a low-degree – in fact, constant-degree – Nullstellensatz refutation over any field \( \mathbb{F} \). Nullstellensatz degree behaves well with respect to compositions: if we compose \( F_{\text{bool}} \) with a gadget \( \text{IND}_m \), \( m = n^{O(1)} \) (see, e.g., [23, §8] how this can be done), the Nullstellensatz degree can only increase by the query complexity of the gadget, which is \( O(\log n) \) for \( \text{IND}_m \). This gives us an \( n^{O(1)} \)-variate boolean \( k \)-CSP \( F' := F_{\text{bool}} \circ \text{IND}_m \) (where \( k \) is constant [23, §8]) such that \( \text{NS}_\mathbb{F}(F') \leq O(\log n) \). On the other hand, we can invoke the strong version of the main result of [23]: if \( F \) has Resolution width \( w \), then \( F \circ \text{IND}_m \) requires Cutting Planes refutations of length \( n^{O(w)} \). In summary, \( F' \) witnesses that \( F \)-Nullstellensatz can be exponentially more powerful than log of Cutting Planes length.

### 4.3 Monotone span program lower bounds

Let us prove Theorem 2: \( 3\text{Lin}(\mathbb{R})\)-SAT\(_n \) requires exponential-size monotone \( \mathbb{F}_p \)-span programs, that is,

\[
\chi_{\mathbb{F}_p}(\text{KW}^+(3\text{Lin}(\mathbb{R})\text{-SAT}_n)) \geq n^{\Omega(1)}.
\]

(3)

Using Theorem 6 and Lemma 7 similarly as in Subsection 4.2, it suffices to show that \( \text{NS}_{\mathbb{R}}(F_{\text{bool}}) \geq n^{\Omega(1)} \), for some unsatisfiable \( k\text{Lin}(\mathbb{R})\)-CSP \( F \) where \( k \) is a constant. To this end, we consider an \( \mathbb{R} \)-linear system \( F = F_{G, U, \mathbb{R}} \) that generalizes \( F_{G, \mathbb{R}} \) defined above:

\[
\forall v \in V : \sum_{(v, u) \in E} z_{(v, u)} - \sum_{(u, v) \in E} z_{(u, v)} = \text{\textbf{1}}_U(v),
\]

\((F_{G, U, \mathbb{R}})\)
where \(1_U: V \to \{0,1\}\) is the indicator function for \(U \subseteq V\). This is unsatisfiable as long as \(U \neq \emptyset\). Combinatorially, the boolean search problem \(S(F_{\text{bool}})\) can be interpreted as an end-of-\(\ell\)-lines problem for \(\ell := |U|\): given a graph with distinguished source vertices \(U\), find a sink or a source not in \(U\). It is important to have many distinguished sources, \(|U| \geq n^{\Omega(n)}\), as otherwise \(S(F_{\text{bool}})\) is in \(\text{PPAD}^\text{DT}\) [31] and hence \(F_{\text{bool}}\) has too low an \(F_p\)-Nullstellensatz degree (by Remark 9).

**Nullstellensatz lower bound.** To show \(\text{NS}(F_{\text{bool}}) \geq n^{\Omega(1)}\) for an appropriate \(F = F_{G,U,R}\), we adapt a result of Beame and Riis [8]. They proved a Nullstellensatz lower bound for a related bijective pigeonhole principle \(P_n\) whose underlying graph has unbounded degree; we obtain a bounded-degree version of their result by a reduction.

> **Lemma 10** ([8, §8]). Fix a prime \(p\). The following system of polynomial equations over variables \(\{x_{ij} : (i,j) \in D \times R\}\), where \(|D| = n\) and \(|R| = n - n^{\Omega(1)}\), requires \(F_p\)-Nullstellensatz degree \(n^{\Omega(1)}\):

\[
\begin{align*}
(i) & \quad \forall i \in D: \sum_{j \in R} x_{ij} = 1 \quad \text{"each pigeon occupies a hole"}, \\
(ii) & \quad \forall j \in R: \sum_{i \in D} x_{ij} = 1 \quad \text{"each hole houses a pigeon"}, \\
(iii) & \quad \forall i \in D, \{j,j'\} \in \binom{R}{2}: x_{ij} x_{ij'} = 0 \quad \text{"no pigeon occupies two holes"}, \\
(iv) & \quad \forall j \in R, \{i,i'\} \in \binom{D}{2}: x_{ij} x_{ij'} = 0 \quad \text{"no hole houses two pigeons"}. \\
\end{align*}
\]

We construct a natural bounded-degree version \(G\) of the complete bipartite graph \(D \times R\) and show that each constraint of \(F_{\text{bool}}\) for \(F = F_{G,U,R}\) is a low-degree \(F_p\)-Nullstellensatz consequence of \(P_n\). Hence, if \(F_{\text{bool}}\) admits a low-degree \(F_p\)-Nullstellensatz proof, so does \(P_n\) (see, e.g., [13, Lemma 1] for composing proofs), which contradicts Lemma 10.

The directed graph \(G = (V,E)\) is obtained from the complete bipartite graph \(D \times R\) as illustrated in Figure 2 (for \(|D| = 4\) and \(|R| = 3\)). Specifically, each vertex of degree \(d\) in \(D \times R\) is replaced with a binary tree of height \(\log d\). The result is a layered graph with the first and last layers identified with \(D\) and \(R\), respectively. We also add a “feedback” edge from each vertex in \(R\) to a vertex in \(D\) according to some arbitrary injection \(R \to D\) (dashed edges in Figure 2). The vertices in \(D\) not incident to feedback edges will form the set \(U\) (singleton in Figure 2).

This defines a boolean 3-CSP \(F_{\text{bool}}\) for \(F = F_{G,U,R}\) over variables \(\{z_e : e \in E\}\). In order to reduce \(P_n\) to \(F_{\text{bool}}\), we define an affine map between the variables \(x_{ij}\) of \(P_n\) and \(z_e\) of \(F_{\text{bool}}\). Namely, for a feedback edge \(e\) we set \(z_e := 1\), and for every other \(e = (v,u)\) we set

\[
z_{(v,u)} := \sum_{i \in D_v, j \in R_u} x_{ij},
\]

where \(D_v := \{i \in D : v\) is reachable from \(i\) without using feedback edges\},

\(R_u := \{j \in R : j\) is reachable from \(u\) without using feedback edges\}.

Note in particular that this map naturally identifies the edge-variables \(z_e\) in the middle of \(G\) (yellow edges) with the variables \(x_{ij}\) of \(P_n\). The other variables \(z_e\) are simply affinely dependent on the middle edge-layer. We then show that from the equations of \(P_n\) we can derive each constraint of \(F_{\text{bool}}\). Recall that the constraint for \(v \in V\) requires that the out-flow \(\sum_{(v,u) \in E} z_{(v,u)}\) equals the in-flow \(\sum_{(u,v) \in E} z_{(u,v)}\) (plus 1 iff \(v \in U\)).

\[v \notin D \cup R:\] Suppose \(v\) is on the left side of \(G\) (right side is handled similarly) so that

\[
z_{(v,u)} = \sum_{j \in R_u} x_{ij}
\]

for some fixed \(i \in D\). The out-flow is

\[
\sum_{(v,u) \in E} z_{(v,u)} = \sum_{(v,u) \in E} \sum_{j \in R_u} x_{ij} = \sum_{j \in R_u} x_{ij}.
\]
On the other hand, \( v \) has a unique incoming edge \( (u^*, v) \) so the in-flow is \( \sum_{(u,v) \in E} z(u,v) = z(u^*, v) = \sum_{j \in R_v} x_{ij} \), which equals (4).

For \( v \in D \): (Case \( v \in R \) is handled similarly). The in-flow equals 1 (either \( v \in U \) so that we have the \(+1\) term from \( 1_U(v) \); or \( v \notin U \) and the value of a feedback-edge variable gives \(+1\)).

The out-flow equals \( \sum_{j \in R_v} x_{ij} = \sum_{j \in R_v} x_{ij}^2 = 1 \) by (4), \( R_v = R \), and (ii).

Finally, we can verify the boolean axioms \( z_e^2 = z_e \). This holds trivially for feedback edges \( e \).

Let \( e = (v, u) \) be an edge in the left side of \( G \) (right side is similar) so that \( z_e = \sum_{j \in R_v} x_{ij} \) for some fixed \( i \in D \). We have \( z_e^2 = (\sum_{j \in R_v} x_{ij})^2 = \sum_{j \in R_v} x_{ij}^2 = \sum_{j \in R_v} x_{ij} = z_e \) by (iii) and the boolean axioms for \( P_n \).

This concludes the reduction and hence the proof of (3).

**PPADS\(^{cc} \notin PPA\(^{cc} \)** via End-of-\( \ell \)-Lines. It is straightforward to check that \( F_{\text{bool}} \) for \( F = F_{G,U,R} \) is in the query class \( \text{PPADS}\(^{dt} \) (Definition 16). In particular, in the \( \text{PPADS} \) decision tree, we can define the distinguished vertex \( v^* \) as being associated with any vertex from \( U \). Similarly, the lifted problem \( S' := S(F_{\text{bool}}) \circ \text{IND}\(^m\)_n \) for \( m = n^{\Theta(1)} \) is in the communication class \( \text{PPADS}\(^{cc} \). By contrast, we just proved that \( \chi_{\text{P}_n}(S') \geq n^{\Omega(1)} \), which implies that \( S' \notin \text{PPA}\(^{cc} \).

## 5 Proofs of Characterizations

Due to space constraints, the proofs of Theorem 3 and Theorem 4 are omitted from this extended abstract. See the full version [24] for complete proofs.

### References


Adventures in Monotone Complexity and TFNP


Adventures in Monotone Complexity and TFNP


A Appendix: TFNP Class Definitions

For each TFNP subclass there is a canonical definition of its communication or query analogue: we simply let communication protocols or decision trees (rather than circuits) implicitly define the objects that appear in the original Turing machine definition. Each communication class $C^{cc}$ (resp. query class $C^{dt}$) is defined via a $C$-protocol (resp. $C$-decision tree) that solves a two-party search problem $S \subseteq \{0,1\}^{n/2} \times \{0,1\}^{n/2} \times \mathcal{O}$ (resp. $S \subseteq \{0,1\}^{n} \times \mathcal{O}$). The class $C^{cc}$ (resp. $C^{dt}$) is then defined as the set of all $n$-bit search problems $S$ that admit a $\text{polylog}(n)$-cost $C$-protocol (resp. $C$-decision tree). We only define the communication analogues below with the understanding that a query version can be obtained by replacing mentions of a protocol $\Pi(x,y)$ by a decision tree $T_v(x)$; the cost of a $C$-decision tree is defined as $\max_{v,x} \#(\text{queries made by } T_v(x))$. In what follows, sink means out-degree 0, and source means in-degree 0.

Definition 11. (FP)

*Syntax:* $\Pi$ is a (deterministic) protocol outputting values in $\mathcal{O}$.

*Object:* $n/a$

*Correctness:* $\Pi(x,y) \in S(x,y)$.

*Cost:* $|\Pi| := \text{communication cost of } \Pi$. 
Definition 12. (EoML)
Syntax: $V$ is a vertex set with a distinguished vertex $v^* \in V$. For each $v \in V$: $o_v \in \mathcal{O}$ and $
abla \Pi_v$ is a protocol outputting a tuple $(s_v(x, y), p_v(x, y), \ell_v(x, y)) \in V \times \mathbb{Z}$. 
Object: Digraph $G_{x,y} = (V, E)$ where $(v, u) \in E$ iff $s_v(x, y) = u$ and $p_v(x, y) = v$. 
Correctness: If $\ell_v(x, y) > \ell_u(x, y)$. 
Cost: $\log |V| + \max_v |\Pi_v|$. 

Definition 13. (SoML)
Syntax: Same as in Definition 12. 
Object: Same as in Definition 12. 
Correctness: If $v^*$ is a sink or non-source in $G_{x,y}$, then $o_v \in S(x, y)$. 
Cost: $\log |V| + \max_v |\Pi_v|$. 

Definition 14. (PLS)
Syntax: $V$ is a vertex set. For each $v \in V$: $o_v \in \mathcal{O}$ and $\Pi_v$ is a protocol outputting a pair $(s_v(x, y), \ell_v(x, y)) \in V \times \mathbb{Z}$. 
Object: Digraph $G_{x,y} = (V, E)$ where $(v, u) \in E$ iff $s_v(x, y) = u$ and $\ell_v(x, y) > \ell_u(x, y)$. 
Correctness: If $v$ is a sink in $G_{x,y}$, then $o_v \in S(x, y)$. 
Cost: $\log |V| + \max_v |\Pi_v|$. 

Definition 15. (PPAD)
Syntax: $V$ is a vertex set with a distinguished vertex $v^* \in V$. For each $v \in V$: $o_v \in \mathcal{O}$ and $\Pi_v$ is a protocol outputting a pair $(s_v(x, y), p_v(x, y)) \in V \times V$. 
Object: Digraph $G_{x,y} = (V, E)$ where $(v, u) \in E$ iff $s_v(x, y) = u$ and $p_v(x, y) = v$. 
Correctness: If $v^*$ is a sink or non-source in $G_{x,y}$, then $o_v \in S(x, y)$. 
Cost: $\log |V| + \max_v |\Pi_v|$. 

Definition 16. (PPADS)
Syntax: Same as in Definition 15. 
Object: Same as in Definition 15. 
Correctness: If $v^*$ is a sink or non-source in $G_{x,y}$, then $o_v \in S(x, y)$. 
Cost: $\log |V| + \max_v |\Pi_v|$. 

Definition 17. (PPA)
Syntax: $V$ is a vertex set with a distinguished vertex $v^* \in V$. For each $v \in V$: $o_v \in \mathcal{O}$ and $\Pi_v$ is a protocol outputting a subset $\Pi_v(x, y) \subseteq V$ of size at most 2. 
Object: Undirected graph $G_{x,y} = (V, E)$ where $\{v, u\} \in E$ iff $v \in \Pi_u(x, y)$ and $u \in \Pi_v(x, y)$. 
Correctness: If $v^*$ has degree $\neq 1$ in $G_{x,y}$, then $o_v \in S(x, y)$. 
Cost: $\log |V| + \max_v |\Pi_v|$. 

Definition 18. (PPP)
Syntax: $V$ is a vertex set with a distinguished vertex $v^* \in V$. For each unordered pair $\{v, u\} \in \binom{V}{2}$: $o_{\{v, u\}} \in \mathcal{O}$. For each $v \in V$: $\Pi_v$ is a protocol outputting values in $V - v^*$. 
Object: Bipartite graph $G_{x,y} = (V \times (V - v^*), E)$ where $(v, w) \in E$ iff $\Pi_v(x, y) = w$. 
Correctness: If $(v, w)$ and $(u, w)$, $v \neq u$, are edges in $G_{x,y}$, then $o_{\{v, u\}} \in S(x, y)$. 
Cost: $\log |V| + \max_v |\Pi_v|$. 

M. Göös, P. Kamath, R. Robere, and D. Sokolov 38:19