Total Search Problems, Logic, and Proof Complexity

Robert Robere, School of Computer Science, McGill

Circuits, Communication, and Proofs, ICTS-Bengaluru.

August 13, 2026

Search Problems

Let $R \subseteq \{0,1\}^* \times \{0,1\}^*$ be an NP relation, so:

1. $R(x,y)$ is polynomial-time computable
2. R is polynomially bounded: there is $c > 0$ s.t.

$$R(x,y) \implies |y| \leq |x|^c$$

E.g. $R(x,y)$ true if y is a satisfying assignment for a formula encoded by x.

$\mathbf{Search}_R$

Input. $x \in \{0,1\}^*$

Feasible Solutions. $y \in \{0,1\}^{|x|^c}$ s.t. $(x, y) \in R$

Notation Alert.

We use R and $\mathbf{Search}_R$ interchangeably.

Example: Search-SAT

Input. CNF formula F

Feasible Solutions. Satisfying assignments for F.

Decision vs. Search

NP search problems are **often** close in complexity to the corresponding decision problem.

(SAT and Search-SAT related by a *self-reduction*.)

Example 2

This is not always true!

Let's say R is **total** if $\forall x \exists y R(x, y)$

Factoring

Input. A positive integer $n > 1$.

Feasible Solutions. Any prime factor of n.

Decision vs. Total Search

Total search problems correspond to *boring* decision problems.

"Does $n > 1$ have a prime factor?"

The answer is always **yes**.

Total Function NP

Define $\mathbf{TFNP} := \{\mathsf{Search}_R : R \text{ is a total NP relation}\}$

Ex. $\text{Factoring} \in \text{TFNP}$

Interesting Facts

- $\mathbf{FP} \subseteq \mathbf{TFNP} \subseteq F(\mathbf{NP}) \cap F(\mathbf{coNP})$
- Believed to *not* to have complete problems[1].

1. Pudlák. *On the complexity of finding falsifying assignments for Herbrand disjunctions.* ↩

Logical Phrasing

Our factoring search problem is a $\forall\exists$ *tautology*:

$$\forall x > 1 \exists y \leq x : \mathrm{IsPrime}(y) \wedge \mathrm{Divides}(x, y)$$

Note $R(x, y) = \mathrm{IsPrime}(y) \wedge \mathrm{Divides}(x, y)$ is polynomial-time computable.

Slogan:

"Every TFNP problem corresponds to a $\forall\Sigma_1^b$ tautology"

Σ_1^b refers to the existential quantifiers being *upper-bounded*:

$$\forall x > 1 \exists y \leq x : \mathrm{IsPrime}(y) \wedge \mathrm{Divides}(x, y)$$

Pigeon

Input: $n^{O(1)}$-size boolean circuit

$$C : \{0,1\}^n \to \{0,1\}^n$$

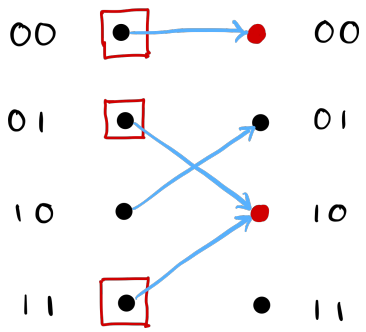
Feasible Solutions:

- Any $x : C(x) = 0^n$
- Any $x \neq y : C(x) = C(y)$

Tautology

Pigeonhole Principle

"Any $f : [N] \to [N-1]$ has a collision."



Pigeonhole Principle, Logically

Here is the logical form of the tautology:

$$\forall n, C : \text{IsCircuit}(C) \wedge |C| \leq n^c,$$

$$\exists x, y : C(x) = 0^n \vee ((x \neq y) \wedge C(x) = C(y))$$

Iteration

Input. $n^{O(1)}$-size circuits computing $S : \{0,1\}^n \to \{0,1\}^n$,

Feasible Solutions.

0^n if $S(0^n) = 0^n$

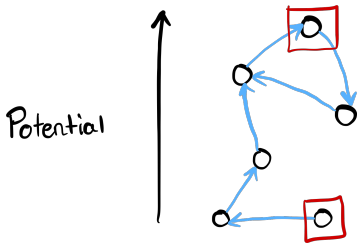
$x \in \{0,1\}^n : S(x) > x$ and $S(S(x)) \leq x$.

Define a DAG on $\{0,1\}^n$ by adding an edge $(u, S(u))$ if $S(u) > u$.

Witnessing Theorem

Local Minima Principle.

"Every directed acyclic graph has a sink."



Other Examples:

Hash-Function Collision,

Inverting Permutations,

Finding Equilibria,

Fair Division,

Gradient Descent Minimization,

...

Reducibility

A search problem R is *mapping reducible* to S, denoted $R \leq_m S$, if there are polynomial-time algorithms I, O s.t.

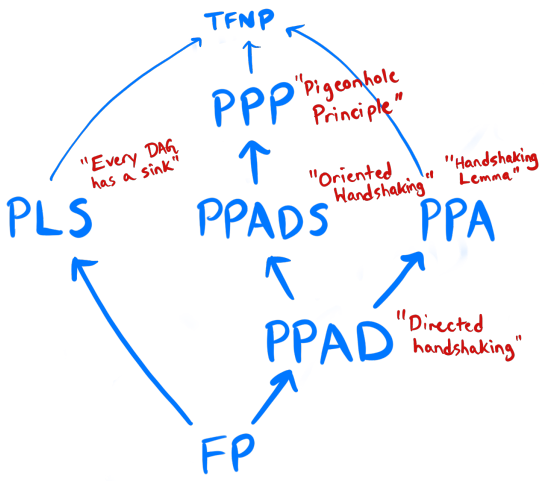
$$(I(x), y) \in S \implies (x, O(x, y)) \in R.$$

We can define **complexity classes**:

- $\mathrm{PPP} := \{R \in \mathrm{TFNP} : R \leq \mathrm{Pigeon}\}$
- $\mathrm{PLS} := \{R \in \mathrm{TFNP} : R \leq \mathrm{Iteration}\}$.

Prominent TFNP Subclasses

Depicted with corresponding $\forall\exists$-tautologies



TFNP and Provability

Proofs of Totality

Let R and S be total NP search problems.

Reducing $R \leq_m S$ is an algorithmic method of proving the theorem

$$S \text{ is total} \implies R \text{ is total.}$$

In the logical approach to TFNP, we get *converses* to this: *proofs of totality* imply *reductions*!

Two Paths:

1. Bounded Arithmetic
   - First order sub-theories of Peano Arithmetic, capturing "bounded" reasoning.
2. Proof Complexity
   - Proof systems for propositional tautologies.

Two Paths:

1. Bounded Arithmetic
   - First order sub-theories of Peano Arithmetic, capturing "bounded" reasoning.
2. Proof Complexity
   - Proof systems for propositional tautologies.

TFNP and Bounded Arithmetic

Start from a BASE theory capturing provability in $\mathbf{FP}$, e.g. PV, S^1_2.

A Σ^b_1 formula is one of the form

$$\psi(z) = \exists x \leq t(z) : \phi(z, x)$$

where $t(z)$ is a BASE term and ϕ is a formula over BASE.

Theorem. Σ^b_1 formulas are exactly the set of NP predicates.

$\therefore$ True $\forall\Sigma^b_1$ sentences correspond to total NP search problems!

Question

If T is a theory of arithmetic, what are the $\forall\Sigma^b_1$ consequences of T?

Question

If T is a theory of arithmetic, what are the $\forall\Sigma^b_1$ consequences of T?

Answer

A subclass of $\mathbf{TFNP}$!

Theories vs TFNP Subclasses

See also [Buresh-Oppenheim-Morioka 04, Buss-Johnson 12, Müller 21] for related results:

Theory	$\forall\Sigma^b_1$ Consequences	Citation
PV, S^1_2	$\mathbf{FP}$	[Cook 75, Buss 86]
$BASE + \mathrm{COUNT}_2(PV)$	$\mathbf{PPA}$	[Jerabek 04]
$T^1_2 = BASE + \Sigma^b_1$-IND	$\mathbf{PLS}$	[Buss-Krajíček 94]
$T^2_2 = BASE + \Sigma^b_2$-IND	$\mathbf{CPLS}$	[Krajíćek-Skelley-Thapen 07]
$T^i_2 = BASE + \Sigma^b_i$-IND	$\mathbf{GI}_i$	[Beckmann-Buss 09, Skelley-Thapen 11]

Black-Box TFNP and Proof Complexity

Separating TFNP Subclasses

$\mathbf{FP} \subseteq \mathbf{TFNP} \subseteq \mathbf{NP}$, so any separation between $\mathbf{TFNP}$ classes above $\mathbf{FP}$ implies $\mathbf{P} \neq \mathbf{NP}$.

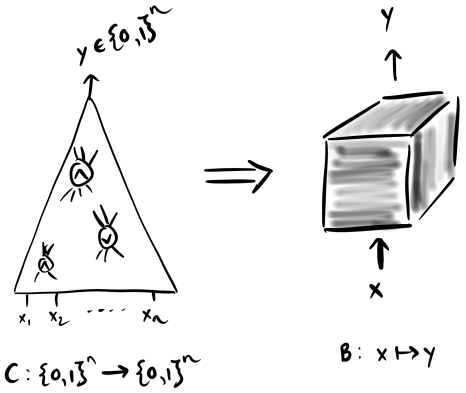
(Equivalently, we cannot separate the strength of the corresponding bounded arithmetic theories.)

What can we do?

Black-Box Setting

Essentially all known reductions in TFNP operate in the *black-box* setting.

Replace circuits with oracles, measure # of queries (i.e. *decision-tree complexity*).



Black-Box TFNP

$\mathbf{Pigeon}^{dt}$

Input: Function $f : [2^n] \to [2^n]$, **provided by oracle**.

Feasible Solutions:

- Any $x : f(x) = 0$
- Any $x \neq y : f(x) = f(y)$

Formalizing Black-Box TFNP

– A **query total search problem** is a sequence

$$R = \{R_n \subseteq \{0,1\}^N \times O_N^R\}_{N \in \mathbb{N}},$$

where O_N^R is polynomial size and each R_N is total.

– $R \in \mathrm{TFNP}^{dt}$ if $R_N(\cdot, o)$ can be computed by a $poly(\log N)$-depth decision tree for each $o \in O_N^R$.

Black-Box Queries and Proofs

There is a natural and close relationship between black-box query problems and propositional proofs.

Queries $\equiv$ Propositional Variables

Pigeons, Propositionally

Let $N = 2^n$.

Encode $f(x) \in [2^n] = [N]$ in binary using n propositional variables $f_{x,1}, f_{x,2}, \ldots, f_{x,n} \in \{0,1\}^n$.

Now, we can express the totality as a DNF:

$$\mathrm{Total}(Pigeon) := \bigvee_x [f(x) = 0] \vee \bigvee_{x \neq y} \bigvee_{k \in [N]} [f(x) = k] \wedge [f(y) = k]$$

This is a *propositional tautology*.

Refutations vs Proofs

Instead of the **DNF tautology**

$$\mathrm{Total}(Pigeon) := \bigvee_x [f(x) = 0] \vee \bigvee_{x \neq y} \bigvee_{k \in [N]} [f(x) = k] \wedge [f(y) = k]$$

we usually think of the **unsatisfiable CNF**

$$\neg\mathrm{Total}(Pigeon) := \bigwedge_x [f(x) \neq 0] \wedge \bigwedge_{x \neq y} \bigwedge_{k \in [N]} [f(x) \neq k] \vee [f(y) \neq k]$$

Canonical: False-Clause Search Problems

Let $F = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be an unsatisfiable CNF.

Let $\mathrm{Search}(F) \subseteq \{0,1\}^n \times [m]$ be the total relation

$$(x, i) \in S(F) \iff C_i(x) = 0.$$

"Given an assignment, find a false clause."

This is in **TFNP$^{\mathbf{dt}}$** if F has $poly(\log n)$ width.

Pigeon$^{\mathbf{dt}}$

Input: Function $f : [2^n] \to [2^n]$, **provided by oracle**.

Feasible Solutions:

- Any $x : f(x) = 0$
- $x \neq y : f(x) = f(y)$

$\mathrm{Search}(PHP)$

Input: Assignment to $f_{x,i}$ for $x \in [N], i \in [n]$

Feasible Solutions:

Any false clause of the Pigeonhole Principle CNF formula.

Canonicity of $\mathrm{Search}(F)$

If $R_N \subseteq \{0,1\}^N \times O_N^R$ is in $\mathbf{TFNP}^{dt}$, then define

$$\neg Total(R_N) := \bigwedge_{o \in O_N^R} \neg R_N(x, o).$$

This is a $poly(\log N)$ width CNF, since $R_N(x, o)$ is computed by a $poly(\log N)$-depth decision tree.

Black-Box Reducibility

- R_n is *depth-d reducible* to S_m, written $R_n \leq_d S_m$, if there is a mapping reduction (I, O) from R_n to S_m such that each output bit of I and O are computed by depth d decision trees
- $R \leq S$ if R is reducible to S by a $poly(\log N)$ depth reduction.

Examples.

PLS^{dt}

Define

$$\mathbf{PLS}^{dt} = \{R : R \leq \mathrm{Iteration}^{dt}\}$$

to be the class of all query total search problems reducible to Iteration.

Proof Complexity of $\mathbf{PLS}^{dt}$?

Remember each $R \in \mathbf{PLS}^{dt}$ corresponds to a sequence of $poly(\log N)$-width unsatisfiable CNFs:

$$\neg Total(R_N) := \bigwedge_{o \in O_N^R} \neg R(x, o).$$

Question. What is this family of CNFs?

Theorem[1].

Let $\mathbf{F} = \{F_n\}_n$ be a sequence of unsatisfiable CNFs on n variables, and recall that $S(\mathbf{F}) = \{S(F_n)\}_n$. Then

$S(\mathbf{F}) \in \mathbf{PLS}^{dt}$

if and only if

F_n has a $poly(\log n)$-width Resolution refutation.

(*Resolution.* $C \vee x, D \vee \neg x \vdash C \vee D$)

1. See [Kam20] and [BJ12] ↩

Theorem[1].

Let $\mathbf{F} = \{F_n\}_n$ be a sequence of unsatisfiable CNFs on n variables, and recall that $S(\mathbf{F}) = \{S(F_n)\}_n$. Then

$S(\mathbf{F}) \in \mathbf{PLS}^{dt}$

if and only if

F_n has $poly(\log n)$-width Resolution refutations.

$\therefore$ Proof complexity lower bounds imply TFNP separations!

1. See [Kam20] and [BJ12] ↩

Other Characterizations

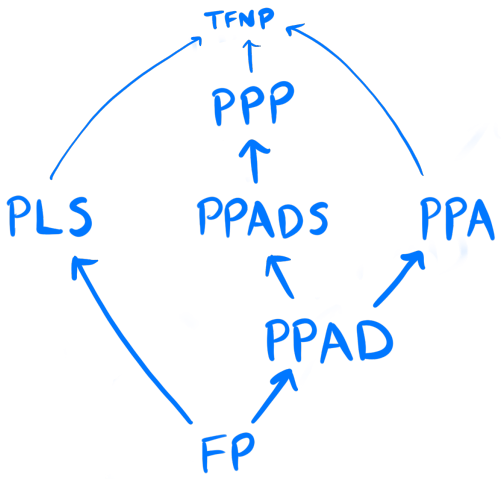
What other natural proof systems can be characterized?

Definition.

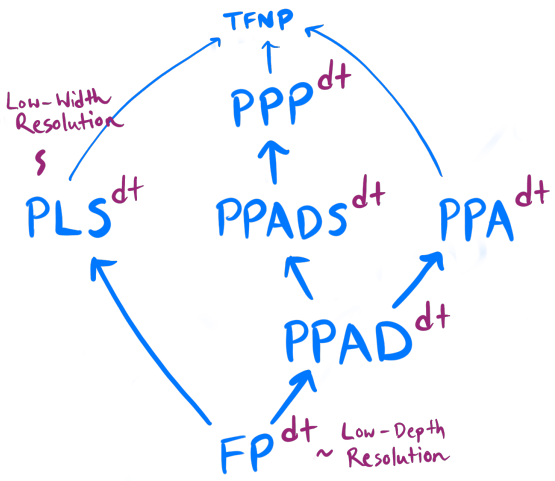
If P is a proof system and $\mathbf{C}$ is a subclass of $\mathbf{TFNP}$, write $\mathbf{C} \sim P$ if

$S(\mathbf{F}) \in \mathbf{C}^{dt}$ if and only if $P \vdash F$ efficiently.

Classical TFNP Subclasses
Each of our original TFNP subclasses...

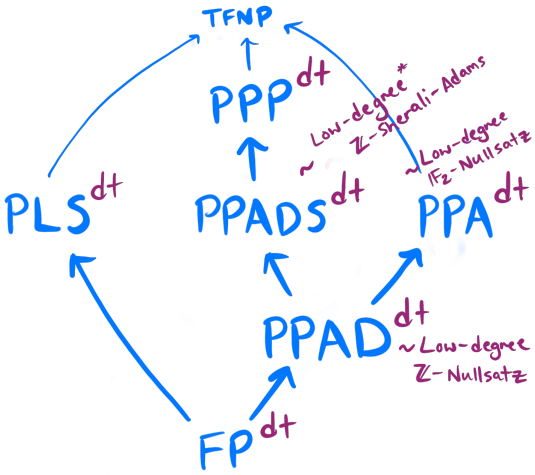


Boolean Systems
Fragments of Resolution proofs correspond to PLS and FP.



Algebraic Systems

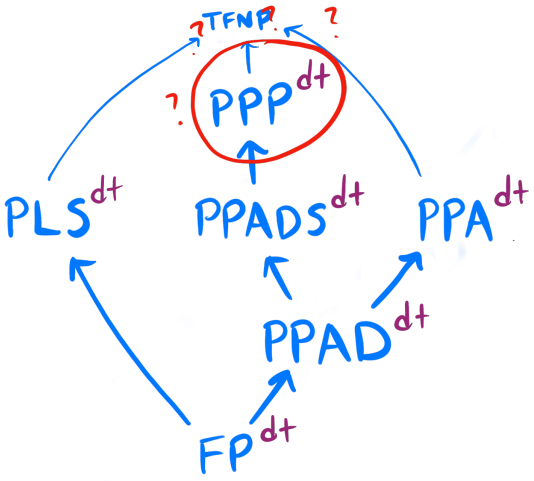
Algebraic proof systems correspond to PPAD, PPADS[1], and PPA[2]



1. [GHJMPRT22] Separations in Proof Complexity and TFNP. ↩
2. [GKRS19] Adventures in Monotone Complexity and TFNP ↩

Misbehaved Pigeons?

PPP is equivalent to no known "standard" proof system



When is $\mathbf{C} \sim P$ possible?

Buss, Fleming, and Impagliazzo[1] gave necessary and sufficient conditions:

1. Efficient P-proofs must be closed under *low-depth decision-tree substitutions.*
2. P must prove its own *reflection principle*, asserting that "If Π is a P-refutation of F, then F is unsatisfiable".

1. Buss, Fleming, Impagliazzo. *TFNP characterizations of proof systems and monotone circuits.* ↩

When is $\mathbf{C} \sim P$ possible?

Buss, Fleming, and Impagliazzo[1] gave necessary and sufficient conditions:

1. Efficient P-proofs must be closed under *low-depth decision-tree substitutions.*
2. P must prove its own soundness.

Examples

- Bounded-width Resolution
- Depth-$(d + 0.5)$ PK
- Bounded-degree Nullstellensatz
- Bounded-degree Polynomial Calculus
- Bounded-degree Sherali-Adams (unary coeffs.)
- Bounded-degree Sum-of-Squares (unary coeffs.)

1. Buss-Fleming-Impagliazzo. *TFNP characterizations of proof systems and monotone circuits.* ↩

When is $\mathbf{C} \sim P$ possible?

Buss, Fleming, and Impagliazzo[1] gave necessary and sufficient conditions:

1. Efficient P-proofs must be closed under *low-depth decision-tree substitutions.*
2. P must prove its own soundness.

Non-Examples

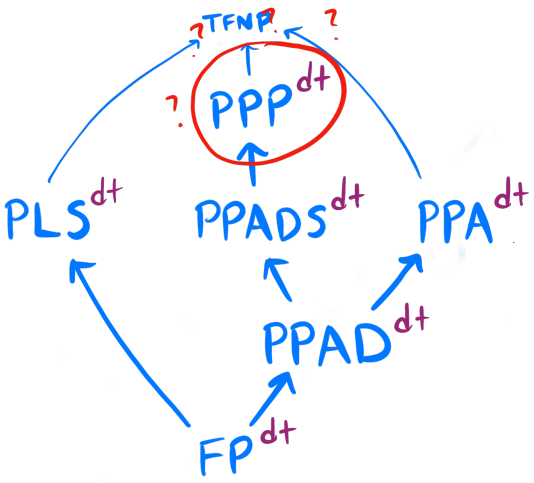
- Resolution (**1** fails if width is high).
- High-degree variants of algebraic/semi-algebraic systems (**1** seemingly fails).
- Cutting Planes (**1** fails, and also possibly **2**).

1. Buss-Fleming-Impagliazzo. *TFNP characterizations of proof systems and monotone circuits.* ↩

Misbehaved Pigeons?

PPP is equivalent to no known "standard" proof system

(But there is an "abstract" one...)



Where does this lead?

– Characterizations of *algebraic* and *semi-algebraic* proof systems in $\mathbf{TFNP}$.
– New *black-box separations* between $\mathbf{TFNP}$ subclasses.
– New proof systems (and novel characterizations of them).
– New separations in proof complexity.

Structural Proof Complexity

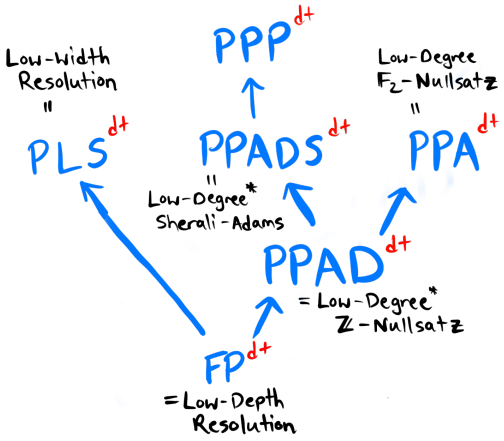
Structural relationships between $\mathbf{TFNP}^{dt}$ subclasses imply structural relationships between propositional proof systems.

Two Examples:

– Intersection Theorems
– Join Theorems

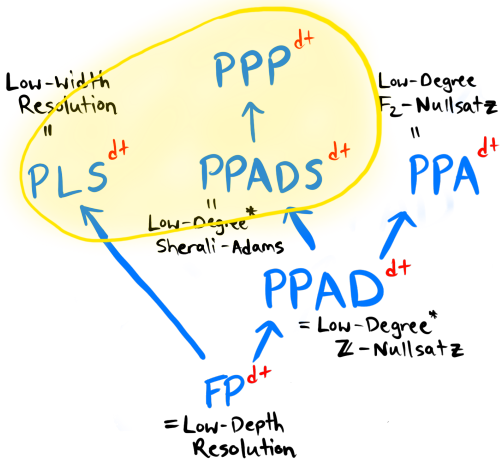
Separations in TFNP

All separations in black-box TFNP follow from separations between propositional proof systems.



Separations in TFNP

All separations in black-box TFNP follow from separations between propositional proof systems.



Sherali-Adams vs. Resolution

Bounded-degree Sherali-Adams can efficiently simulate bounded-degree Resolution, but the naive simulation uses coefficients that are exponentially-large in the width.

Question. ($\mathbf{PPADS}^{dt}$ vs. $\mathbf{PLS}^{dt}$)

Can **unary** Sherali-Adams simulate Resolution?

Theorem[1].

```
    "Any simulation of Resolution by Sherali-Adams requires large
coefficients or large degree".
```

Any degree $n^{o(1)}$ Sherali-Adams refutation of $\neg Total(\mathrm{Iteration}_n)$ requires coefficients with magnitude $\exp(n^{\Omega(1)})$.

Corollary.

$\mathbf{PPADS}^{dt} \not\subseteq \mathbf{PLS}^{dt}$.

$\mathbf{PPP}^{dt} \not\subseteq \mathbf{PLS}^{dt}$[2]

1. Göös, Hollender, Jain, Maystre, Pires, Robere, Tao. *Separations in TFNP and Proof Complexity.* ↩
2. Independently observed by Bonacina and Thapen. ↩

More Black-Box Results

- Separations between classes and their Turing Closures, e.g. $\mathbf{PPP}^{dt} \neq \mathbf{FP}^{\mathbf{PPP}^{dt}}$ [1]
- Separations between all possible *intersections* of classes[2].
- Separations between Ramsey and Pigeon, among other classes [3].
- Separations between "dense" TFNP problems[4]
- Relationships with provability of lower bounds [5]
- ...

1. Fleming-Grosser-Pitassi-Robere 24 ↩
2. Hubacek-Khaniki-Thapen 24 and Li-Pires-Robere 24 ↩
3. Jain, Li, Robere, Xun 24 ↩
4. Li 24 ↩
5. Li-Li-Ren 25 ↩

Structural Proof Complexity

$\mathrm{Either}(R, S)$

Input. An input to R and an input to S.

Output. A solution to either of the instances.

Fact. $Q \leq_m \mathrm{Either}(R, S)$ iff

$$Q \leq_m R \text{ and } Q \leq_m S.$$

In other words, $\mathrm{Either}(R, S)$ is complete for $\mathbf{R} \cap \mathbf{S}$.

Intersections in Complexity

The intersection of two complexity classes is not typically **syntactic**.

E.g. $\mathbf{NP} \cap \mathbf{coNP}$ is not widely believed to have complete problems.

Are the $\mathrm{Either}(R, S)$ classes **natural**?

Theorem[1].

$$\mathbf{CLS} = \mathbf{PLS} \cap \mathbf{PPAD},$$

where $\mathbf{CLS}$ is the class of problems reducible to the "totality of gradient descent".

1. Fearnley, Goldberg, Hollender, Savani. The complexity of gradient descent: CLS = PPAD ∩ PLS. ↩

Intersection Theorems

Opened the floodgates for these collapses[1].

Theorem[2]. $\mathbf{SOPL} = \mathbf{PLS} \cap \mathbf{PPADS}$, and $\mathbf{EOPL} = \mathbf{PLS} \cap \mathbf{PPAD}$.

Later work[3][4] characterized all intersections among classical $\mathbf{TFNP}$ classes.

1. See also Ishizuka. *The complexity of the parity argument with potential.* ↩
2. Göös, Hollender, Jain, Maystre, Pires, Robere, Tao. *Further collapses in TFNP.* ↩
3. Hubáček, Khaniki, Thapen. *TFNP intersections through the lens of feasible disjunction.* ↩
4. Li, Pires, Robere. *Intersection classes in TFNP and proof complexity.* ↩

Intersection Proof Systems

This raises the possibility of defining *intersection proof systems*.

Definition. A proof system P is the *intersection* of proof systems Q and R if P has an efficient refutation of formula F if and only if both Q and R have efficient refutations of formula F.

Example[1].

$\mathbf{PLS} \sim$ Narrow Resolution.

$\mathbf{PPADS} \sim$ Unary Sherali-Adams.

$\mathbf{PLS} \cap \mathbf{PPADS} = \mathbf{SOPL} \sim$ Reversible Resolution

Lines are clauses.

Two symmetric rules, applied with *substitution*:

1. **Resolution.** $C \vee x, C \vee \neg x \vdash C$
2. **Weakening.** $C \vdash C \vee x, C \vee \neg x$.

1. Göös, Hollender, Jain, Maystre, Pires, Robere, Tao. *Separations in TFNP and Proof Complexity.* ↩

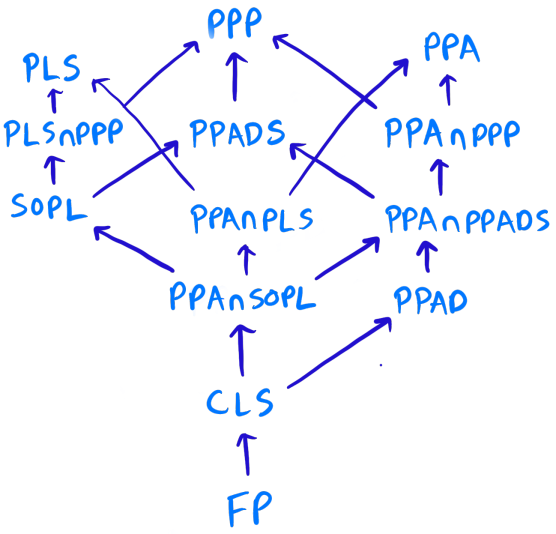
Intersections

All non-trivial intersections in black-box setting

All have proof systems.

Some *join* theorems also possible (not depicted)

See Yaroslav's Talk next week!



Summary

- $\mathbf{TFNP}$ is an "algorithmic counterpart" to proof complexity and bounded arithmetic.
- Separations between proof systems (or bounded arithmetic theories) are often equivalent to separations between $\mathbf{TFNP}$ subclasses.
- There are benefits to studying *both* sides of the theory: $\mathbf{TFNP}$ gives new structural characterizations of proof systems, and proof systems separate $\mathbf{TFNP}$ subclasses.

Structural Aspects

The structural aspects of $\mathbf{TFNP}$ are (to me) particularly attractive.

They give *organizational principles* for many existing proof systems, and give systematic methods to generate *new* proof systems.

It has revealed *new structural phenomena* (intersection and join theorems) in proof complexity.

Next Steps?

Progress on boundary questions?

What about the bounded-depth Frege hierarchy with respect to a CNF formula, or proving lower bounds for Res($\oplus$) or $AC^0[2]$-Frege?

Strong Proof Systems?

Are there interesting combinatorial principles capturing Frege or extended Frege?

Related to the long-standing open problem of finding hard tautologies for these systems.

There are some principles ("Local Improvement Principles"[1]) but they are hard to work with.

See Noah's talk next week

1. Kołodziejczyk, Nguyen, Thapen. *The provably total NP search problems of weak second order bounded arithmetic.* ↩

Outlier TFNP Classes

Understand the relationships between "outlier" classes, like Factoring, Ramsey, etc.

We can also attempt to understand *very weak* TFNP principles.

Many of these are related to various infinite games on graphs (e.g. parity games). Are there more proof systems lurking down there?

Improve our Weaknesses

- Currently study deterministic decision-tree reductions, which is somewhat restricted.
- Many results should still hold under stronger reductions (e.g. Turing reductions[1][2]), some separations are now known[3].
- Randomized reductions? Or distributional hardness? See Morgan's talk next week!

1. Buss, Johnson. Propositional Proofs and reductions between NP search problems. 2012 ↩
2. Müller. Typical forcing, NP search problems, and an extension of a theorem of Riis. 2021 ↩
3. Fleming, Grosser, Pitassi, Robere. Black-box PPP is not Turing-Closed. 2024 ↩

Stronger Oracles? Randomized Reductions?

- What about, e.g., capturing semi-algebraic systems like Cutting Planes that are outside the theory?
- Can we strengthen the oracles used[1]?
- A lot of work going *beyond* TFNP to higher in the polynomial hierarchy (e.g. $\mathrm{TF}\Sigma_2$), with links to meta-complexity.
  See Deniz talk next week!

1. Thapen. Fitting large complexity classes into TFNP ↩

Thank You!

$$\mathrm{Total}(\mathrm{Pigeon}) = \bigvee_{x} [f_x = 0^n] \vee \bigvee_{x \neq x'} \bigvee_{k} [f_x = k] \wedge [f_{x'} = k].$$

$$\neg\mathrm{Total}(\mathrm{Pigeon}) = \bigwedge_{x} [y_x \neq 0^n] \wedge \bigwedge_{x \neq x'} [y_x \neq y_{x'}].$$

Intersection Classes and Related Results