Nonconstructive Proofs of Existence for Provably Total Search Problems

Sam Buss

Workshop on Theoretical Computer Science
Higher School of Economics
Moscow
April 6, 2016

Total NP Search Problems — TFNP

Definition (Meggido-Papadimitriou'91; Papadimitriou'94)

A Total NP Search Problem (TFNP) is a polynomial time relation $R(x, y)$ so that R is

- *Total*: For all x, there exists y s.t. $R(x, y)$,
- *Honest (poly growth rate)*:
  If $R(x, y)$, then $|y| \leq p(|x|)$ for some polynomial p.

The TFNP Problem is:

Given an input x, output a y s.t. $R(x, y)$.

TFNP is intermediate between P (polynomial time) and NP (non-deterministic polynomial time).

Defn: FP is the set of polynomial time functions.

Thm: If $\mathrm{P} = \mathrm{NP}$, then TFNP problems are in FP.

Pf: This is immediate. Query the bits of a solution y. $\square$

Thm: If TFNP problems are in FP, then $\mathrm{NP} \cap \mathrm{coNP} = \mathrm{P}$.

Pf: If $(\exists y \leq s)A(x, y) \leftrightarrow (\forall y \leq t)B(x, y)$ is in $\mathrm{NP} \cap \mathrm{coNP}$, then $A(x, y) \vee \neg B(x, y)$ defines a TFNP predicate. $\square$

In particular, problems in $\mathrm{NP} \cap \mathrm{coNP}$ give rise to TFNP problems.

Open: Does TFNP contain $\mathrm{FP}^{\mathrm{NP}}$? More precisely, is $\mathrm{FP}^{\mathrm{NP}}$ polynomial time Turing reducible to TFNP?

The totality condition $\forall x\, \exists y\, R(x, y)$ is a **semantic** property. Thus TFNP is a *semantic* class, not a *syntactic* class. Correspondingly, we have two open open questions:

Open questions:

1. Is there an effective enumeration of the TFNP problems?
2. Does TFNP have a complete problem?

The semantic condition means that TFNP problems must come with a justification of the totality property. The two main frameworks for justifying totality are:

Complexity Theory: Giving combinatorial principles implying totality.

Bounded Arithmetic: Proving totality in formal theories.

This talk will survey both approaches, plus discuss recent results.

[Papadimitriou'94, ...]

1^{st} example:

Pigeonhole Principle, PIGEON **(PPP)**
Input: $x \in \mathbb{N}$ and injective $f : [x] \to [x-1]$ (purportedly)
Output: $a, b \in [x]$ s.t. either $f(a) \notin [x-1]$ or $f(a) = f(b)$.

The function f can be specified by either

a. A Boolean circuit (multiple output bits), or

b. An oracle.

Thus, the input size is polynomially bounded in $|x|$.

The function is exponential size, but is specified implicitly with a polynomial size description.

Let $R(x, y)$ and $Q(x, y)$ be TFNP problems.

Definition (Many-one reduction, $\preccurlyeq$)

A *(polynomial time) many-one reduction* from R to Q (denoted $R \preccurlyeq Q$) is a pair of polynomial time functions $f(x)$ and $g(x, y)$ so that, for all x, if y is a solution to $Q(f(x), y)$, then $g(x, y)$ is a solution to R, namely $R(x, g(x, y))$.

Definition (PPP)

PPP is the class of TFNP problems many-one reducible to PIGEON.

That is PPP is specified by the combinatorial principle:

PPP:

There is no injective map from $[x]$ *to* $[x-1]$.

More TFNP classes [Papadimitriou'94]:

PPA:

Any undirected graph with degrees ≤ 2 which has a vertex of degree 1 has another vertex of degree 1.

PPAD:

Any directed graph with in-/out-degrees ≤ 1 which has a vertex of total degree 1 has another vertex of total degree 1.

PPADS:

Any directed graph with in-/out-degrees ≤ 1 which has a source, also has a sink.

In all cases, the (exponential size) graph is given implicitly by a function f which computes the neighbors of a given vertex.
f is represented by either a circuit or an oracle.

Yet more:

Polynomial Local Search, PLS:
[Johnson, Papadimitriou, Yanakakis'88]

> *A directed graph with outdegree ≤ 1, and a nonnegative cost function which strictly decreases along directed edges, has a sink.*

Factoring:

> *Any integer ≥ 2 has a prime factor.*

SMITH:

> *An odd degree graph has an even number of Hamiltonian cycles.*

The latter two problems are particularly natural since they do not implicitly involve an exponential size graph.

NASH:

A two player game specified by payoff matrices, has a Nash equilibrium *(a mixed strategy which is local optimally for each player).*

P-LCP (Positive Linear Complementarity)

For an $n \times n$ matrix M and vector $\mathbf{q}$, there is either a solution $\mathbf{x}$, $\mathbf{y}$ *s.t.*

$$\mathbf{y} = M\mathbf{x} + \mathbf{q}, \qquad \mathbf{x}, \mathbf{y} \geq 0, \qquad \mathbf{x}^T\mathbf{y} = 0,$$

or a principal minor with determinant ≤ 0.

These also do not involve exponential size graphs.

Complete Problems (see "compendium" online)

The following are many-one complete for PPAD:

- 3D- and 2D-Sperner Lemma (Papadimitriou'94, Chen-Deng'06a]
- (2-player) NASH [Daskalakis-Goldberg-Papadimitriou'06, Chen-Deng'06b, Chen-Deng-Teng'09]
  For nonrelativized PPAD only, i.e., circuit specifications of graphs only!!
- (2D-)Brower Fixed Point [P'94; C-D'06a]

Open: Smith is in PPA. It is in PPAD? Is it PPA-complete?

Open: P-LCP is in PPAD. Is it PPAD-complete? [Adler-Verma'06/'11] conjecture no.

Factoring is many-one reducible to PPA via randomized reductions. [Jerábek'15, Buresh-Oppenheim'06]

For PPA, there are fewer known complete problems, apart from the canonical problem LEAF.

On non-orientable manifolds: SPERNER and TUCKER are PPA-complete. [Grigni'01; Friedl et al.'06; Deng et al.'ta]

[P'94] claimed TUCKER is PPAD-complete. But the argument only showed:

Thm: TUCKER is PPAD-hard.

This holds in the two dimensional case as well:

Thm: [Pálvölgi'09] 2-D TUCKER is PPAD-hard.

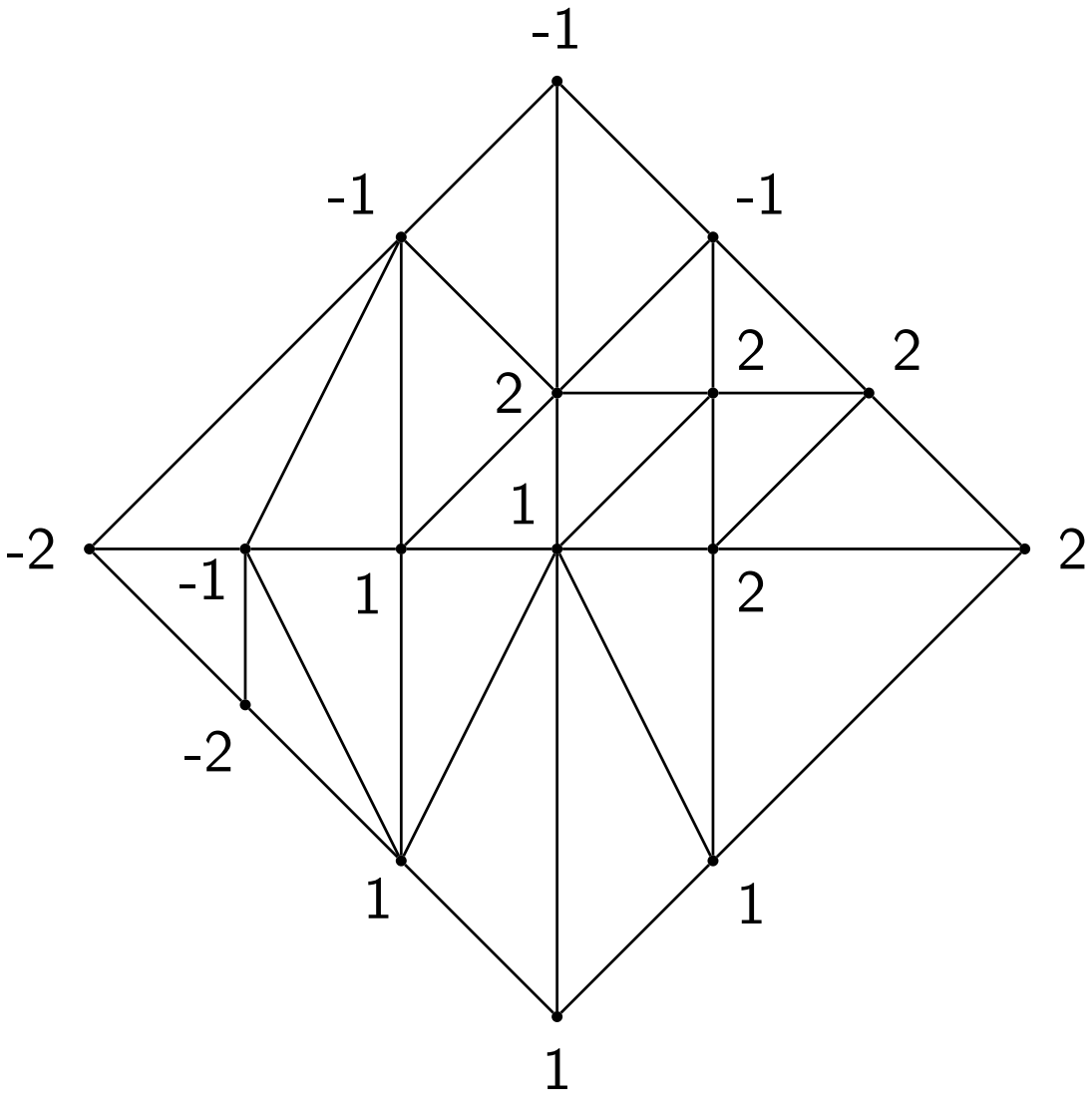
Tucker's Lemma: Let T be an antipodally symmetric triangulation of the unit ball B^n. Let λ map vertices of T to $\{\pm 1, \ldots, \pm n\}$ s.t. $\lambda(-v) = -\lambda(v)$ for boundary vertices v. Then T contains a 1-simplex (an edge) $\{v, w\}$ with $\lambda(v) = -\lambda(w)$.

The TUCKER search problem is many-one equivalent to the (discrete) BORSUK-ULAM problem.

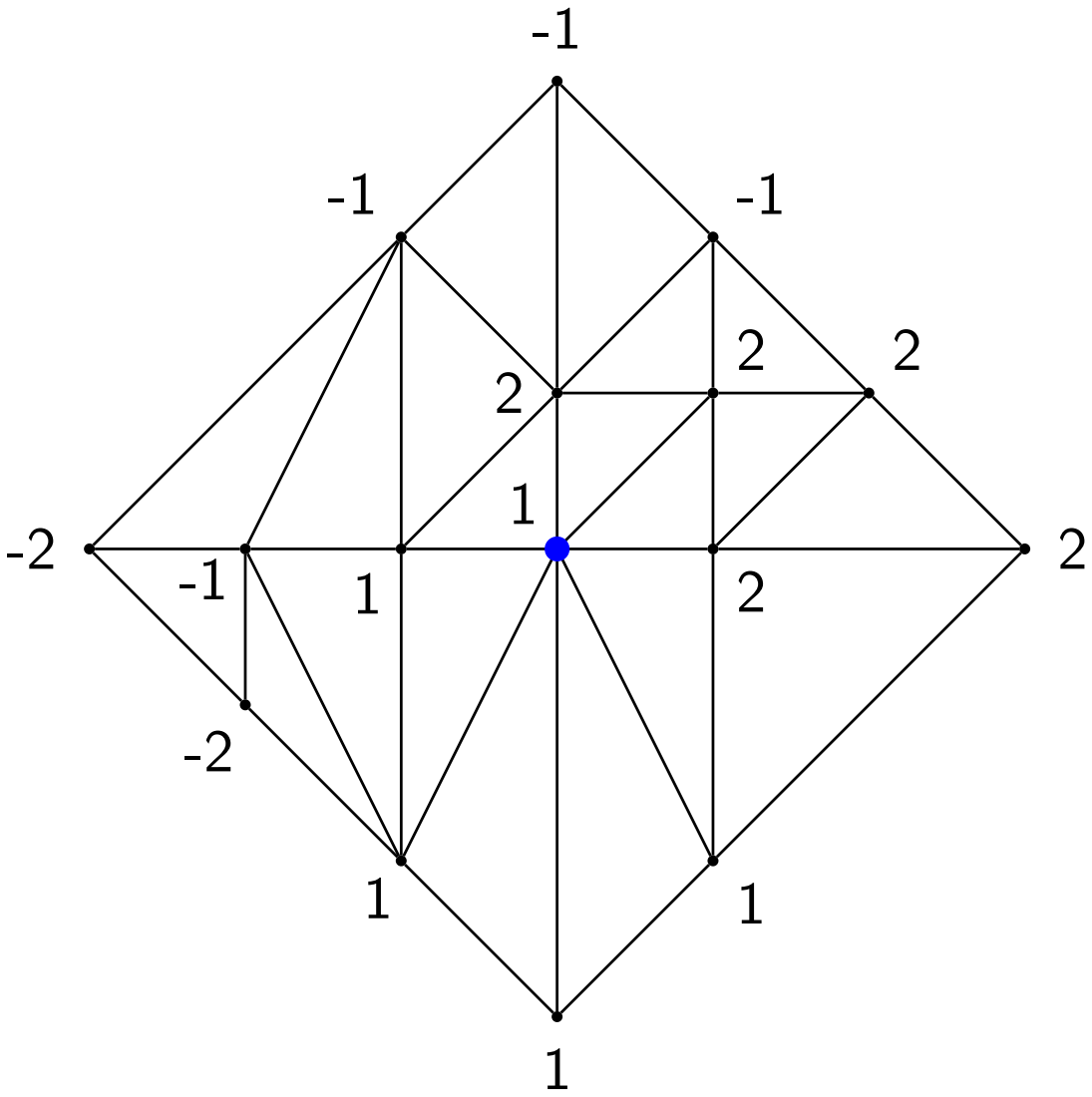
Theorem (Aisenberg-Bonet-B.)

TUCKER *and* 2-D TUCKER *are* PPA*-complete.*

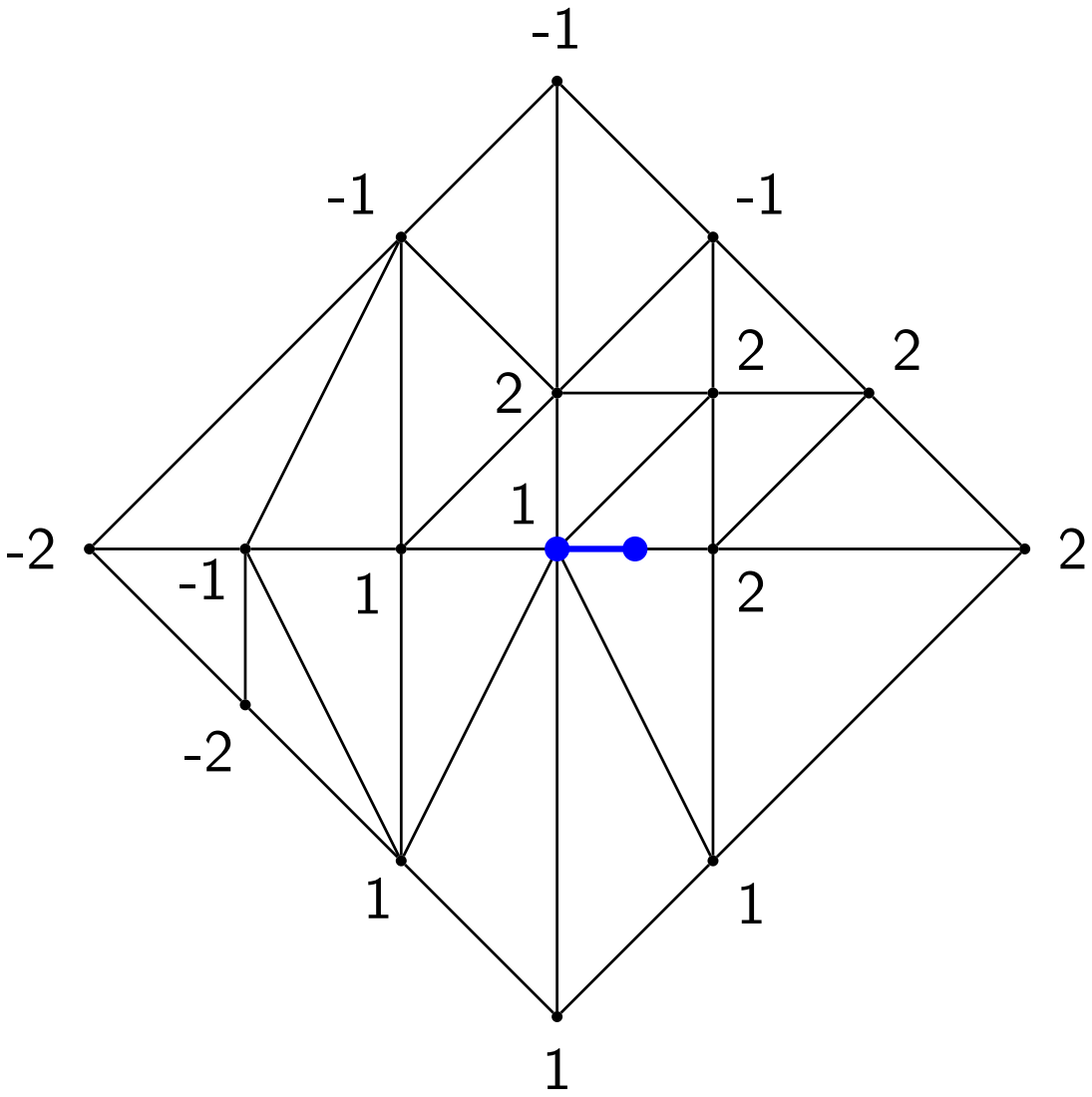
Example - 2D Tucker



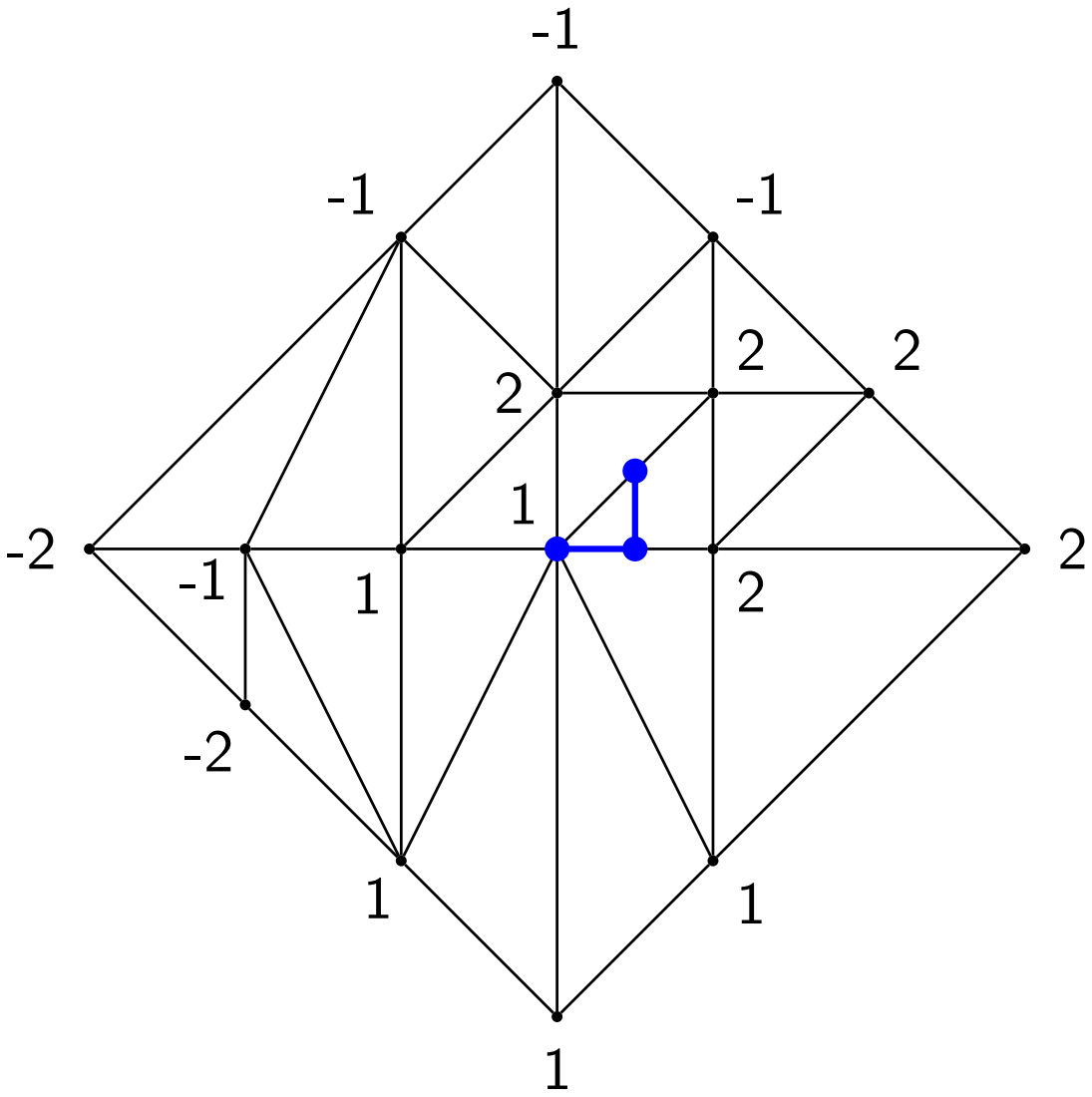
Example - 2D Tucker



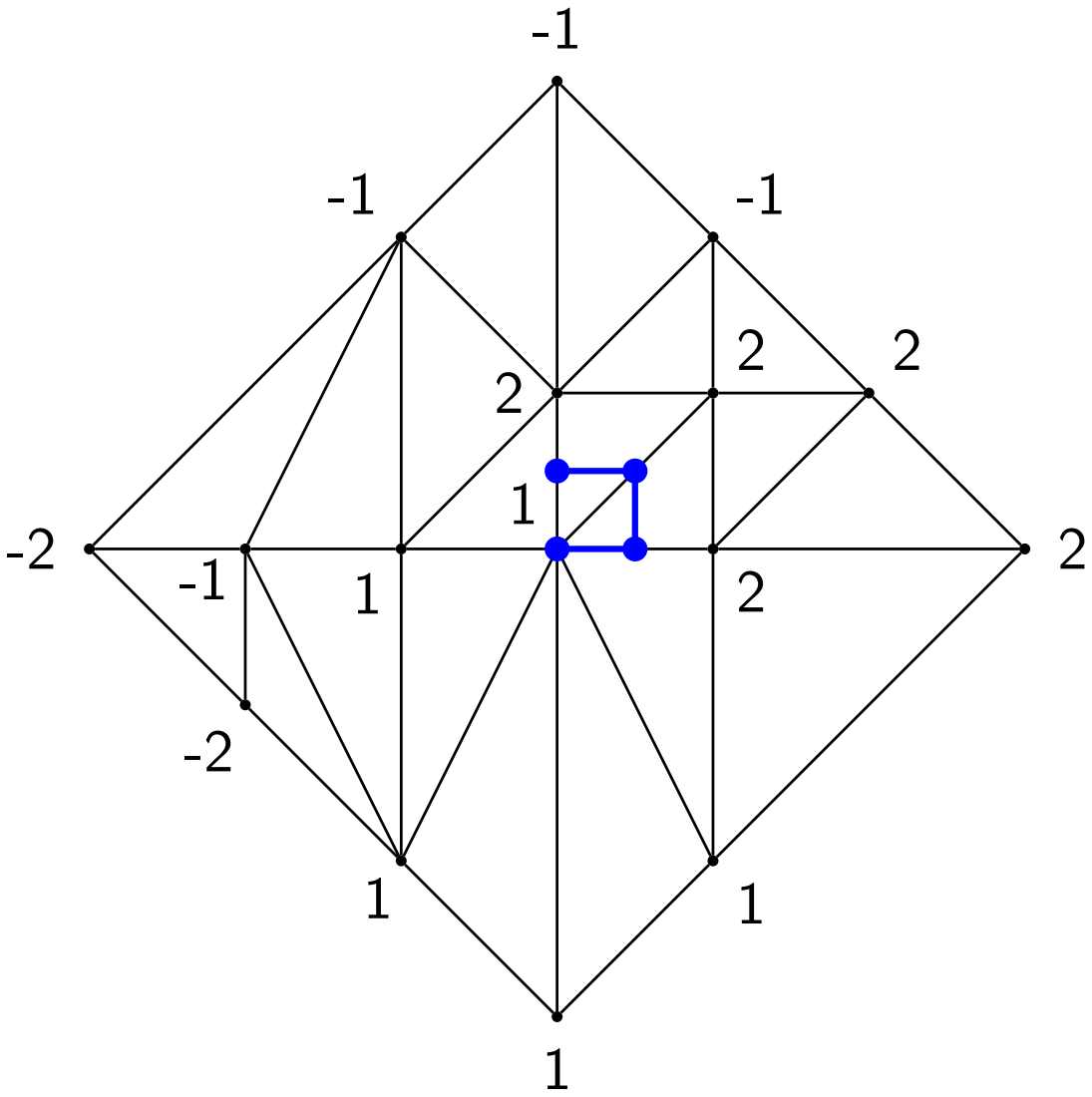
Example - 2D Tucker



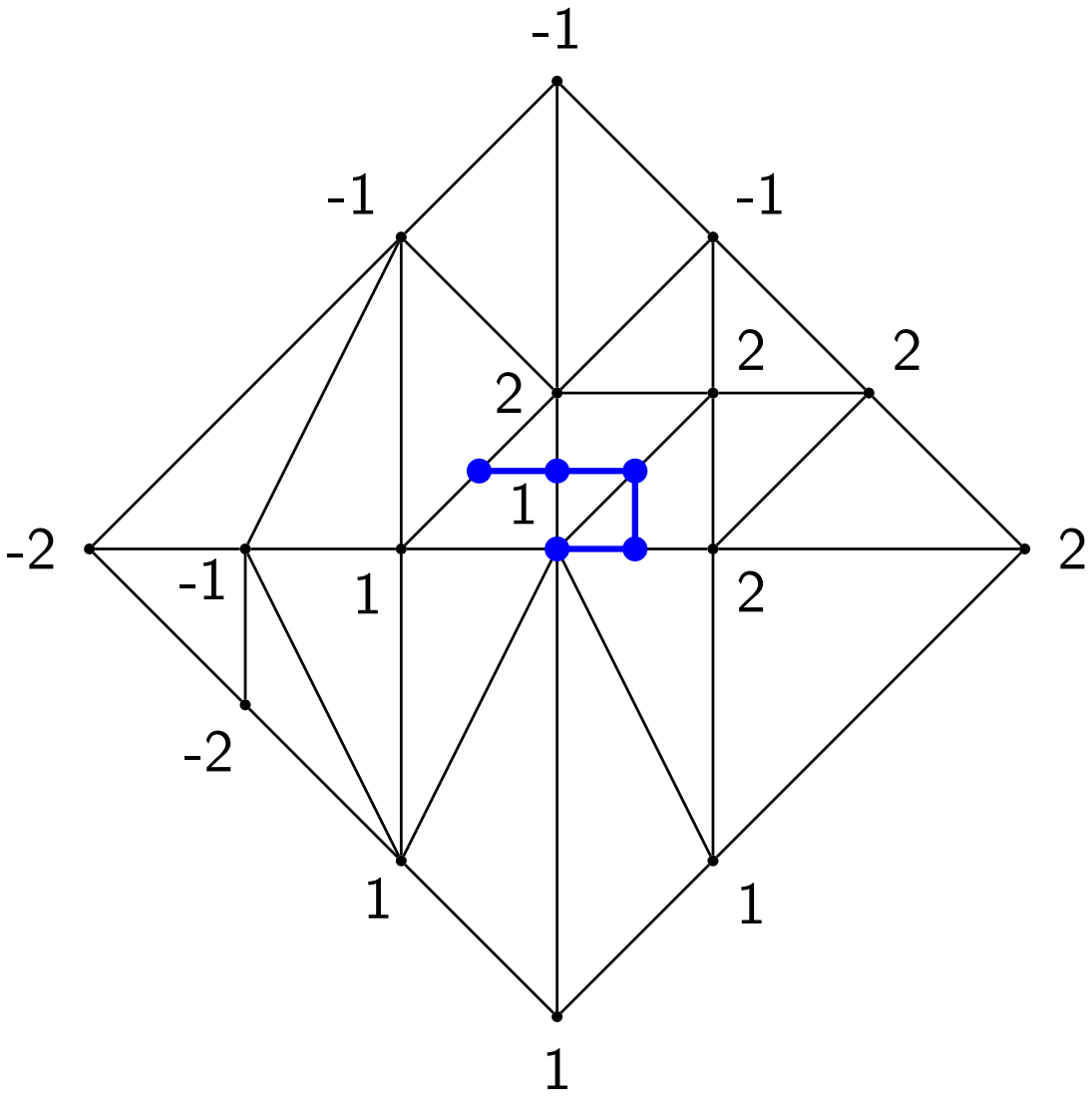
Example - 2D Tucker



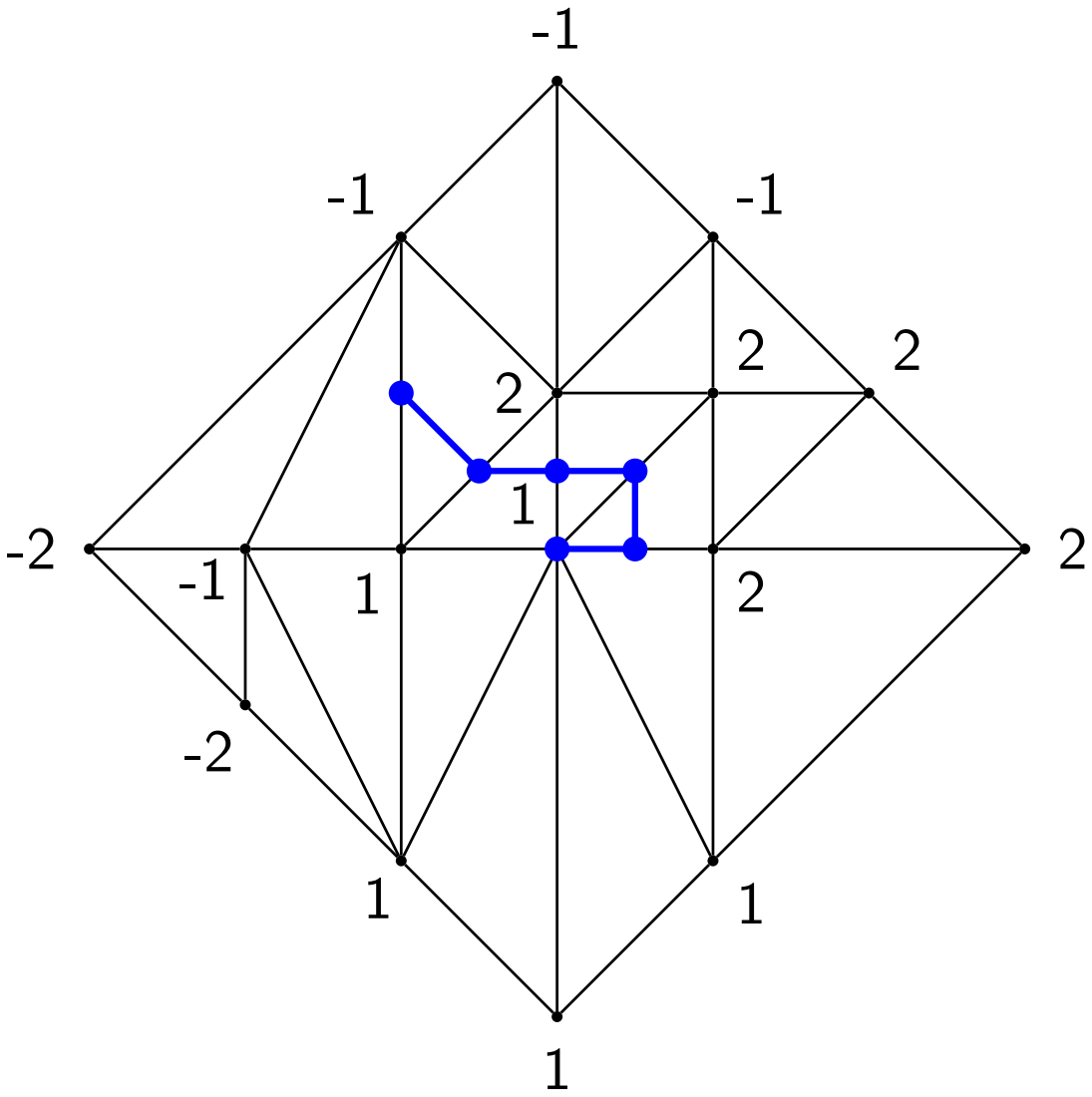
Example - 2D Tucker



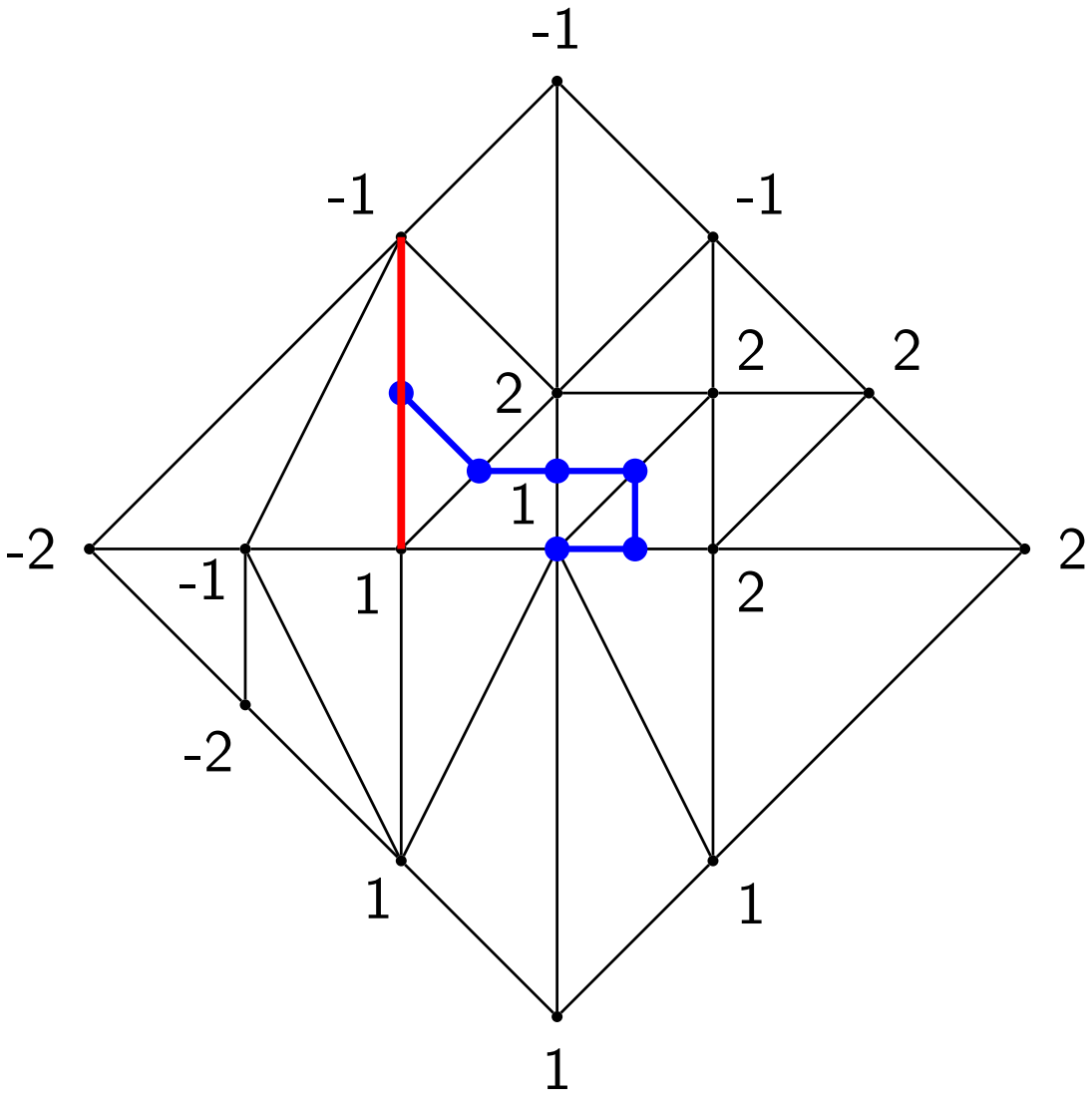
Example - 2D Tucker



Example - 2D Tucker



Example - 2D Tucker



k-Truncated Tucker Lemma [ABBCI'15]. Let $k > 2$. The k-Truncated Tucker Lemma is defined similarly to the Tucker lemma, except λ is only defined on low-dimensional subspaces. (Details omitted.)

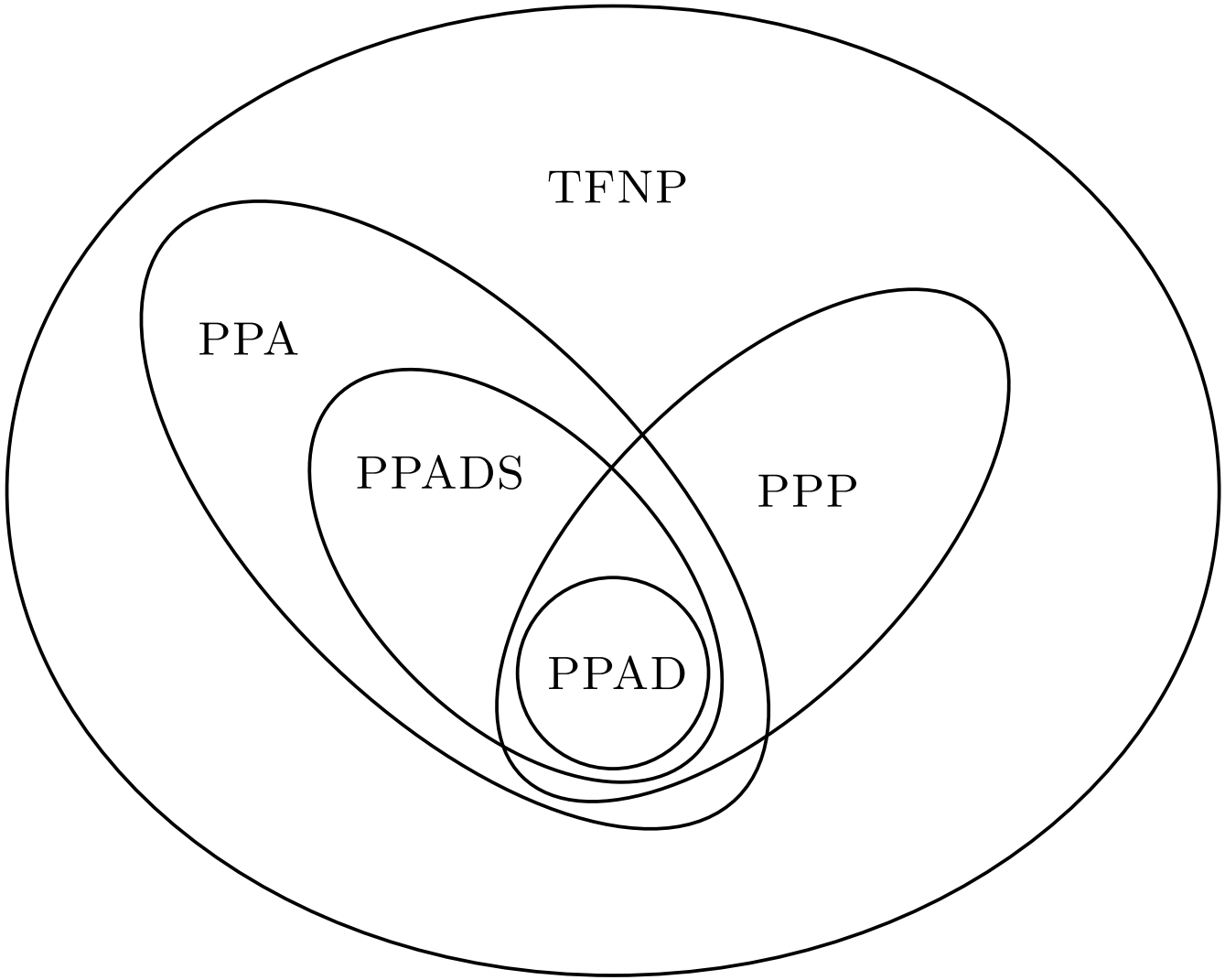
For $k > 0$, the k-Truncated Tucker lemma is natural and can be expressed as a TFNP problem that does not require encoding an exponentially large graph.

Theorem (ABBCI'15; Aisenberg'16)

k-Truncated Tucker $\preccurlyeq_{\text{many-one}}$ *$(k+1)$-Truncated Tucker.*
k-Truncated Tucker Lemma $\Rightarrow$ *k-Kneser-Lovász Theorem.*

Open Question: Do the k-Truncated Tucker search problems form a proper hierarchy of TFNP problems?

More Open Questions: Is the Octahedral Tucker Lemma PPA-complete? (The Octahedral Tucker uses the first barycentric triangulation and is in PPA [Pálvölgi'09].)



Subclasses of TFNP. In the oracle setting, the shown inclusions are proper [BCEIP 1998].

II. Bounded Arithmetic Approach to TFNP

Bounded Arithmetic theories.
Weak fragments of Peano arithmetic.

Induction restricted to bounded (Δ_0) formulas.

Hierarchy of first-order theories [B'85]:

$$S_2^1 \subseteq T_2^1 \preccurlyeq S_2^2 \subseteq T_2^2 \preccurlyeq S_2^3 \subseteq \cdots$$

Hierarchy of second-order theories:

$$U_2^1 \subseteq V_2^1 \subseteq \cdots$$

TFNP problems can be defined using provability in a theory of bounded arithmetic to establish totality.

First Witnessing Theorems for Bounded Arithmetic [B'85]

Theories	Graph Definability	Computational Complexity
S_2^1	Σ_1^b (TFNP)	P
T_2^1 or S_2^2	Σ_2^b	P^{NP}
T_2^2 or S_2^3	Σ_{i+1}^b	$\mathrm{P}^{\Sigma_{i+1}^b}$
U_2^1	$\Sigma_1^{1,b}$	PSPACE
V_2^1	$\Sigma_1^{1,b}$	$\mathrm{EXPTIME}$

Def'n: We identify the Σ_1^b-definable functions of a theory R as the TFNP functions which are definable in R.

Thus: The TFNP problems definable in S_2^1 are precisely the polynomial time functions.

For other classes, the definable functions listed in the table are not in TFNP, since their graph is not in P (nor in NP).

Theorem (B-Krajíček'94)

The TFNP *problems which are definable in* T^1_2 *(equivalently,* S^2_2*) are precisely the* PLS *(Polynomial Local Search) problems.*

Equivalently, PLS is many-one complete for the Σ^b_1-definable functions of T^1_2 (or, S^2_2).

More PLS-based TFNP problems:

Colored PLS: [Krajíček-Skelley-Thapen'07]. Herbrandized PLS search problems with coNP definable set of feasible solutions.

Π^p_k**-PLS:** [Beckmann-B.'09/'10]. Herbrandized PLS search problems with Π^p_{k-1} definable set of feasible solutions.

Theorem. [KST'07, BB'09/'10]

1. Colored PLS is many-one complete for the TFNP problems of T^2_2.
2. Π^p_k-PLS is many-one complete for the TFNP problems of T^k_2.

Weak Pigeonhole and Ramsey

Weak Pigeonhole (WPHP)

There is no injective map from $[2x]$ *to* $[x]$.

Ramsey

A graph G *on* $[x]$ *has either a clique or an independent set of size* $\frac{1}{2}\log x$.

No completeness results are known for these problems:

Theorem

a. WPHP *is provable/definable as a* TFNP *problem in* T^2_2. [Paris-Wilkie-Woods'88, Maciel-Pitassi-Woods'00/'02]

b. RAMSEY *is provable/definable as a* TFNP *problem in* T^3_2. [Pudlák'91, see also Jerábek'09]

Herbrandized Ordering Principle (HOP)

A linear ordering $\prec$ on $[x]$ cannot have a total immediate predecessor function.

k-round Game Induction Principle (GI_k)

A winning strategy for two player k-round game is preserved under iterations of many-one reductions between games.

Theorem: [B.-Kolodziejczyk-Thapen'14] HOP is provable in T_2^2.

It is unlikely HOP is many-one complete for the TFNP problems of T_2^2.

Theorem: [Skelly-Thapen'11] GI_k is many-one complete for the TFNP problems of T_2^k.

[Pudlák-Thapen'12]: Similar results for k-round max/min games, and a related Nash equilibrium principle.

Local Improvement Principles

k-round Local Improvement Principle LI_k

> *Labels on a directed acyclic graph on $[x]$ can be consistently updated in a well-founded manner for k-rounds.*

LI (no subscript) allows $k = x$ (exponentially many rounds)

LLI - graph is a line.

RLI - graph is a rectangle.

Theory	Many-One Complete	
T_2^k or S_2^{k+1}	LI_k	[KNT'11]
V_2^1	LI	[KNT'11]
V_2^1	$\mathrm{LI}_{\log}$, LI with $O(\log n)$ rounds	[BB'14]
U_2^1	LLI, Linear LI	[BB'14]
U_2^1	$\mathrm{LLI}_{\log}$	[KNT'11]
V_2^1	RLI, Rectangular LI	[KNT'11]
V_2^1	$\mathrm{RLI}_{\log}$	[BB'14]
U_2^1	RLI_1	[BB'14]

Frege proofs

Frege proofs are the usual "textbook" proof systems for propositional logic, using modus ponens as their only rule of inference.

Connectives: $\wedge$, $\vee$, $\neg$, and $\rightarrow$.

Modus ponens (MP): $\dfrac{A \quad A \rightarrow B}{B}$

Axioms: Finite set of axiom schemes, e.g.: $A \wedge B \rightarrow A$

Defn: Proof *size* is the number of symbols in the proof.

Frege proofs and Extended Frege proofs

Frege proofs are the usual "textbook" proof systems for propositional logic, using modus ponens as their only rule of inference.

Connectives: $\wedge$, $\vee$, $\neg$, and $\rightarrow$.

Modus ponens (MP): $\dfrac{A \quad A \rightarrow B}{B}$

Axioms: Finite set of axiom schemes, e.g.: $A \wedge B \rightarrow A$

Extended Frege proofs allow also the *extension axiom*, which lets a new variable x abbreviate a formula A:

$$x \leftrightarrow A$$

Defn: Proof *size* is still the number of symbols in the proof.

Open Question

Do Frege proofs polynomially simulate extended Frege proofs? That is, can every extended Frege proof of size n be transformed into a Frege proof of size $p(n))$ or $2^{p(\log n)}$, for some polynomial p?

Intuition: Extended Frege proofs can reason about Boolean circuits, Frege proofs about Boolean formulas.

It is generally conjectured that Boolean circuits can require exponential size to express as Boolean formulas.

By analogy, it is generally conjected Frege proofs can require exponential size to simulate extended Frege proofs.

Example of a Frege proof of $A \to A$:

$A \to (B \to A)$	Axiom
$(A \to (B \to A)) \to (A \to (B \to A) \to A) \to (A \to A)$	Axiom
$(A \to (B \to A) \to A) \to (A \to A)$	M.P. 1,2
$(A \to (B \to A) \to A)$	Axiom
$A \to A$	M.P. 3,4

Example of a Frege "proof" of a contradiction:

$A \to (\neg A \to A)$	Axiom
$(A \to (\neg A \to A)) \to (A \to (\neg A \to A) \to A) \to A$	Axiom
$(A \to (\neg A \to A) \to A) \to A$	M.P. 1,2
$(A \to (\neg A \to A) \to A)$	Axiom
A	M.P. 3,4

$\vdots$

as above, interchanging A and $\neg A$

$\vdots$

$\neg A$

obtain a contradiction

$\bot$

Search Problem: Find the mistake in the proof!

Frege proof consistency as a total NP search problem

Code an (exponentially long) Frege proof P with an oracle X. The value $X(i)$ gives the i-th symbol of P.

Search problem: Show that X does not code a valid Frege proof of a contradiction.

Frege Consistency Search Problem - *Informal*

Input: Second-order X and first-order x.
Output: A set of values $i_1, \ldots, i_\ell$ so that the values $X(i_1), \ldots, X(i_\ell)$ show X does not code a valid Frege proof of a contradiction.

Since the Frege proof is exponentially long, it may contain exponentially long formulas.

However, ℓ should be polynomially bounded by $|x|$: Frege proofs need to be carefully encoded to allow this.

Frege proofs encoded by oracle $X(i)$ contain:

- Fully parenthesized formulas, terminated by commas.
- Each parenthesis has a pointer to its matching parenthesis.
- Each comma has the type of inference for the previous formula, plus pointers to the formulas used as hypotheses.

This allows any syntactic error in the Frege proof to be identified by constantly many positions $i_1, \ldots, i_\ell$ in X.

Theorem. [Beckmann-B.'??]

The Frege consistency search problem is many-one complete for the TFNP problems of U_2^1.

Theorem. [Beckmann-B.'??; Krajíček'??]

The extended Frege consistency search problem is many-one complete for the TFNP problems of V_2^1.

Recall that U_2^1 and V_2^1 have proof complexity corresponding to polynomial space and exponential time.

Open Questions / Future Problems

1. Better alignment between the complexity and the bounded arithmetic approaches to TFNP. E.g.
   a. WPHP as a natural TFNP complexity class?
   b. Bounded arithmetic theories corresponding to PPA or PPAD or PPADS or PPP?
      Already have: T^2_2 corresponds to PLS.
2. Proof systems whose consistency search problem is TFNP complete for the theories T^k_2.
   (Conjecture: Res(log) for T^1_2.)
3. RLI_2 and 1-Truncated Tucker have polynomial size extended Frege proofs. Do they have (quasi)polynomial size Frege proofs? Are definable as TFNP problems in U^1_2?

Thank you!