Some NP functions proof complexity and completeness

Sam Buss

Workshop on Logic and Computational Complexity
Shonan, Japan
September 20, 2017

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).

This talk will cover some examples (one classic and three recent), from the viewpoints of:

- Many-one Completeness within computational classes such as PPA.
- Proofs of totality in propositional logic.
- Proofs of totality in first-order and second-order theories, especially fragments of bounded arithmetic.
- Many-one Completeness for theories of bounded arithmetic.

Blind Monks
and
an Elephant

Hanabusa Itchō
(1652-1724)

Bounded Arithmetic
Propositional Proof Complexity
Complexity Classes (PPA, PPAD, etc.)

} — give three different viewpoints.

Part I: PHP example

Pigeonhole Principle (PHP): *a classic example*

For $m>n$, PHP_n^m states there is no injective function from $[m]$ to $[n]$.
PHP_n means PHP_n^{n+1}, the *pigeonhole principle*.
WPHP_n means $\mathrm{PHP}_{n/2}^n$, the *weak pigeonhole principle*.

As a search problem: For $n = 2^k$, PHP_n is the following: Given a function $f : \{0,1\}^k \to \{0,1\}^k$, find $x, x' \in \{0,1\}^k$ s.t.

- $f(x) = \vec{0}$ or
- $x \neq x'$ and $f(x) = f(x')$.

The function f is given either by a (poly-size) Boolean circuit C or by an oracle (defining the bit-graph) of f.
Open: Is there a poly time algorithm to solve this problem?

PHP_n is many-one complete for PPP. [Papadimtriou,'94]

(WPHP_n does not correspond to a common TFNP class.)

PHP in propositional logic:

Use propositional variables $P_{i,j}$ to mean $f : i \mapsto j$.

$$\mathrm{PHP}_n^m : \quad \bigwedge_{i\in[m]} \bigvee_{j\in[n]} P_{i,j} \quad \rightarrow \quad \bigvee_{i\neq i'\in[m]} \bigvee_{j\in[n]} (P_{i,j} \wedge P_{i',j})$$

Underlying search problem: Either find i falsifying the hypothesis, or i, i', j falsifying the conclusion.

Propositional proofs: PHP_n^m is a tautology, and thus has a proof in any complete propositional proof system.

Underlying proof complexity question: What are the shortest proofs of PHP_n^m?

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: $\dfrac{A \quad A \rightarrow B}{B}$

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: $\dfrac{A \quad A \rightarrow B}{B}$

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:

$$x \leftrightarrow A \qquad \text{[Tseitin'68]}$$

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

Intuition:
Frege (F) proofs reason using Boolean formulas;
Extended Frege (eF) proofs reason using Boolean circuits.

Theorem

- PHP_n *has poly size* eF *and* F *proofs.*
  [Cook-Reckhow'79, B'86]
- PHP_n *requires exponential size constant-depth* F *proofs.*
  [BIKPPW'92]
- WPHP_n *has quasipoly size, constant-depth* F *proofs.*
  [Paris-Wilkie-Woods'88, PW'85]

Bounded Arithmetic for NP Search Functions

An **NP Search Problem** of a formal theory R is a function defined by a polynomial time relation which is provably total by R.

Example: WPHP_n [Paris-Wilkie-Woods'88, Maciel-Pitassi-Woods'00/'02]

$$T_2^2 \vdash (\forall n)(\exists i, i' \leq n)[f(i) \geq \frac{n}{2} \vee (i \neq i' \wedge f(i) = f(i'))]$$

f may depend on n, so $f(n) = f(i, n)$.

All known results, for totality of NP Search functions in bounded arithmetic, use f given via an oracle; either as an uninterpreted function symbol, or via an oracle defining the bit-graph of f.

In other words, all known positive results work in the relativized setting.

First Order Theories [B'85]

S_2^1	Polynomial time functions [B'85]
T_2^1	The Class PLS [B-Krajíček'94]
T_2^2	The Class CPLS (Colored PLS) [Krajíček-Skelley-Thapen'07]
T_2^i	Unclear for $i > 2$, related to polynomial hierarchy
T_2	The union $\cup_i T_2^i$

Second Order Theories[B'85]

U_2^1	Related to polynomial space
V_2^1	Related to exponential time

Theorem

- *T_2^1 proves totality of PLS. [BK'94]*
- *T_2^2 proves totality of WPHP. [PWW'98, MPW'00]*
- *U_2^1 and V_2^1 prove totality of PHP and the common NP Search classes (PPA, PPAD, PPADS, PPP, Ramsey, Factoring, Bertram-Chebyshev, etc.)*
- *T_2 does not prove totality for PPP (PHP), PPA, PPAD, PPADS. [BIKPPW'92,PB'92]*

Part II: Kneser-Lovász Theorem

Kneser-Lovász Theorem (generalization of PHP):

Def'n: Fix $n > 1$ and $1 \leq k < n$. The (n, k)-Kneser graph has $\binom{n}{k}$ vertices: the k-subsets of $[n]$. The edges are the pairs

$$\{S, T\} \text{ such that } S \cap T = \emptyset,$$

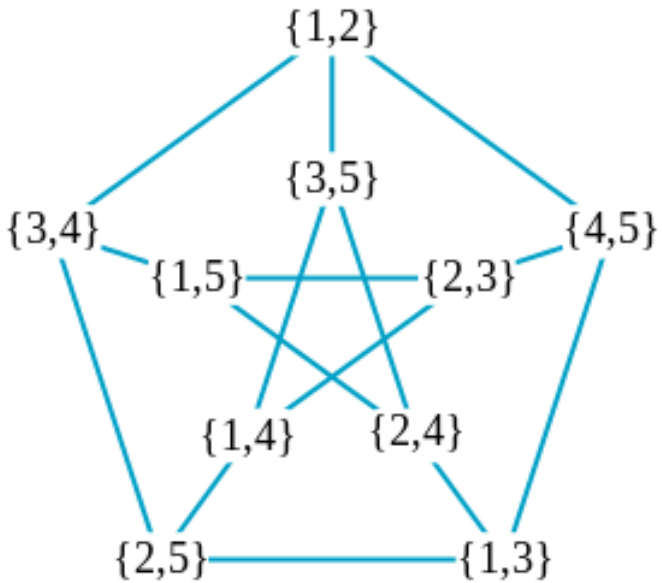
Kneser-Lovász Theorem: [Lovász'78] There is no coloring of the (n, k)-Kneser graph with $\leq n - 2k + 1$ colors.

Kneser-Lovász Theorem (generalization of PHP):

Def'n: Fix $n > 1$ and $1 \leq k < n$. The (n, k)-Kneser graph has $\binom{n}{k}$ vertices: the k-subsets of $[n]$. The edges are the pairs

$$\{S, T\} \text{ such that } S \cap T = \emptyset,$$

Kneser-Lovász Theorem: [Lovász'78] There is no coloring of the (n, k)-Kneser graph with $\leq n - 2k + 1$ colors.



The Petersen graph

$n = 5$

$k = 2$

$n-2k+1 = 2$

No 2 coloring

Kneser-Lovász Theorem (generalization of PHP):

Def'n: Fix $n > 1$ and $1 \leq k < n$. The (n, k)-Kneser graph has $\binom{n}{k}$ vertices: the k-subsets of $[n]$. The edges are the pairs

$$\{S, T\} \text{ such that } S \cap T = \emptyset,$$

Kneser-Lovász Theorem: [Lovász'78] There is no coloring of the (n, k)-Kneser graph with $\leq n - 2k + 1$ colors.

When $k = 1$, this is essentially the PHP^n_{n-1} pigeonhole principle.

Kneser-Lovász Theorem (generalization of PHP):

Def'n: Fix $n > 1$ and $1 \leq k < n$. The (n, k)-Kneser graph has $\binom{n}{k}$ vertices: the k-subsets of $[n]$. The edges are the pairs

$$\{S, T\} \text{ such that } S \cap T = \emptyset,$$

Kneser-Lovász Theorem: [Lovász'78] There is no coloring of the (n, k)-Kneser graph with $\leq n - 2k + 1$ colors.

Usual proof involves the octahedral Tucker lemma. This is a discrete analogue of the Borsuk-Ulam theorem about continuous functions on the n-sphere mapping some pair of antipodal points to the same value.

There is no known way to formalize these topology-based arguments with short propositional proofs, even in extended Frege systems.

Definition (Kneser-Lovász tautologies)

Let $n \geq 2k > 1$, and let $m = n - 2k + 1$ be the number of colors. For $S \in \binom{n}{k}$ and $i \in [m]$, the propositional variable $p_{S,i}$ has the intended meaning that vertex S of the Kneser graph is assigned the color i. The Kneser-Lovász principle is expressed propositionally by

$$\bigwedge_{S \in \binom{n}{k}} \bigvee_{i \in [m]} p_{S,i} \;\rightarrow\; \bigvee_{\substack{S,T \in \binom{n}{k} \\ S \cap T = \emptyset}} \bigvee_{i \in [m]} (p_{S,i} \wedge p_{T,i}) \,.$$

Definition (Kneser-Lovász tautologies)

Let $n \geq 2k > 1$, and let $m = n - 2k + 1$ be the number of colors. For $S \in \binom{n}{k}$ and $i \in [m]$, the propositional variable $p_{S,i}$ has the intended meaning that vertex S of the Kneser graph is assigned the color i. The Kneser-Lovász principle is expressed propositionally by

$$\bigwedge_{S \in \binom{n}{k}} \bigvee_{i \in [m]} p_{S,i} \;\rightarrow\; \bigvee_{\substack{S,T \in \binom{n}{k} \\ S \cap T = \emptyset}} \bigvee_{i \in [m]} (p_{S,i} \wedge p_{T,i}) \,.$$

Originally proposed as a tautology that might not have polynomial size (extended) Frege proofs. [CI'14]

A. Crãciun, G. Istrate; SAT'14

Definition (Kneser-Lovász tautologies)

Let $n \geq 2k > 1$, and let $m = n - 2k + 1$ be the number of colors. For $S \in \binom{n}{k}$ and $i \in [m]$, the propositional variable $p_{S,i}$ has the intended meaning that vertex S of the Kneser graph is assigned the color i. The Kneser-Lovász principle is expressed propositionally by

$$\bigwedge_{S \in \binom{n}{k}} \bigvee_{i \in [m]} p_{S,i} \;\rightarrow\; \bigvee_{\substack{S,T \in \binom{n}{k} \\ S \cap T = \emptyset}} \bigvee_{i \in [m]} (p_{S,i} \wedge p_{T,i}) \,.$$

Theorem [ABBCI'15]: Fix a value for k. The Kneser-Lovász Theorem has polynomial size extended Frege proofs, and quasipolynomial size Frege proofs.

J. Aisenberg, M.L. Bonet, B., A. Crãciun, G. Istrate; ICALP '15

The (quasi)polynomial size Frege and extended Frege proofs for the Kneser tautologies are based on new counting proofs.

The proof (omitted) is quite easy. It establishes a simplified form of Erdős-Ko-Rado, Hilton-Milner theorems on intersecting families of finite sets.

This avoids the usual arguments based on the topological Tucker Lemma, except for base cases.

What about the proof complexity of Tucker Lemma for propositional logic? ....

Part III: Tucker's Lemma

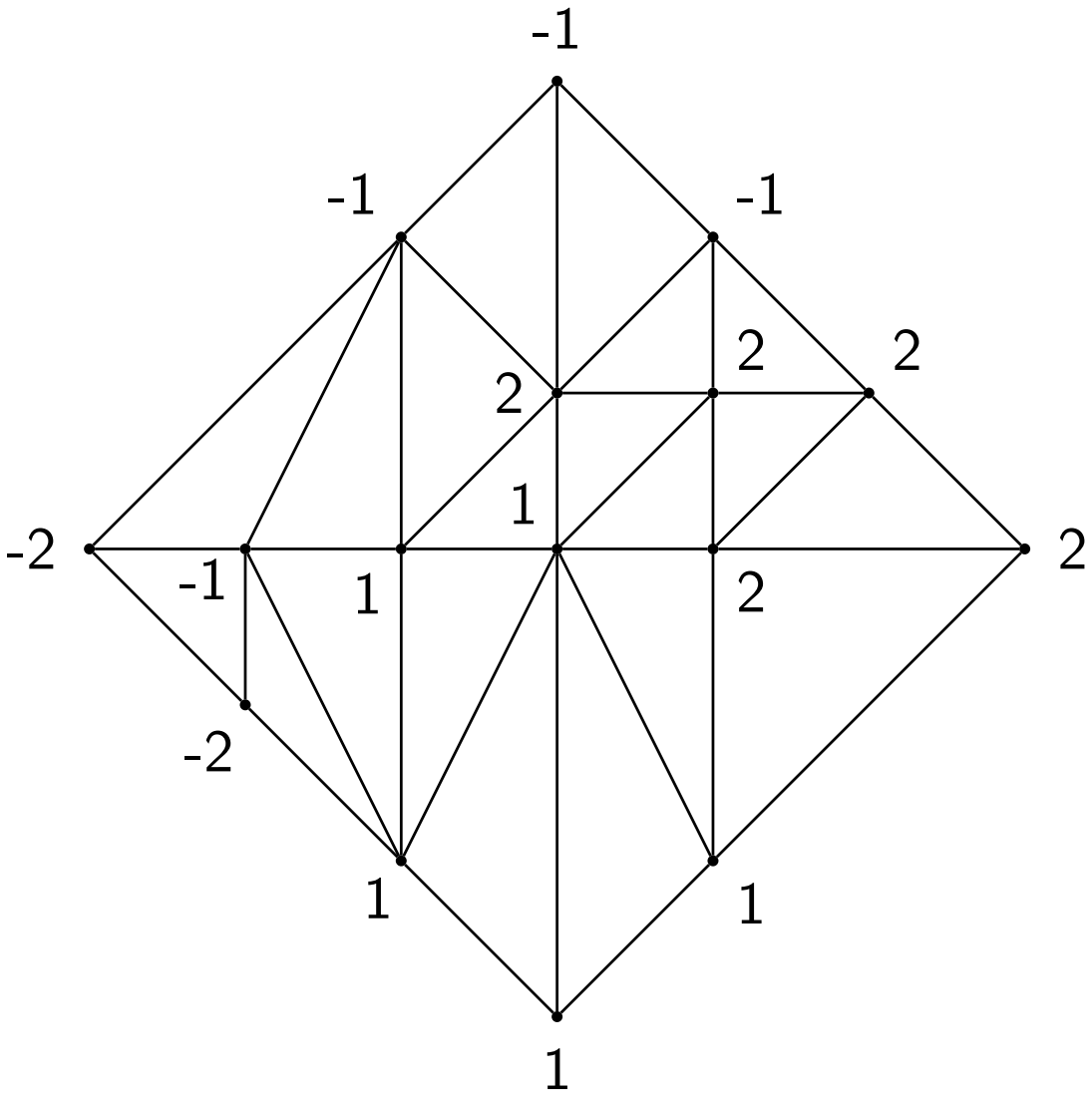
Tucker's Lemma

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 $\lambda(v) = -\lambda(w)$.

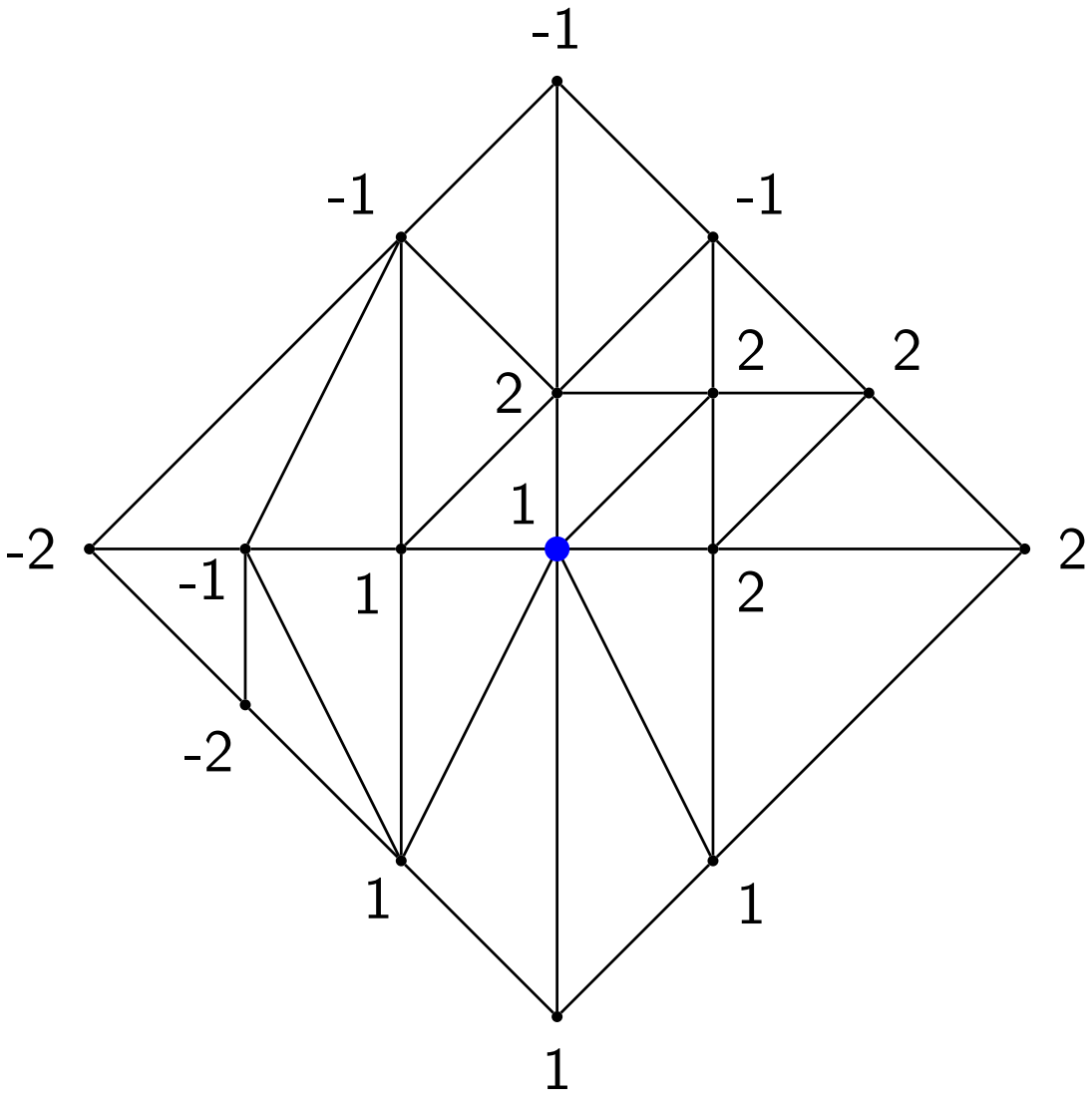
The TUCKER search problem is many-one equivalent to the (discrete) BORSUK-ULAM problem.

See example next slides ....

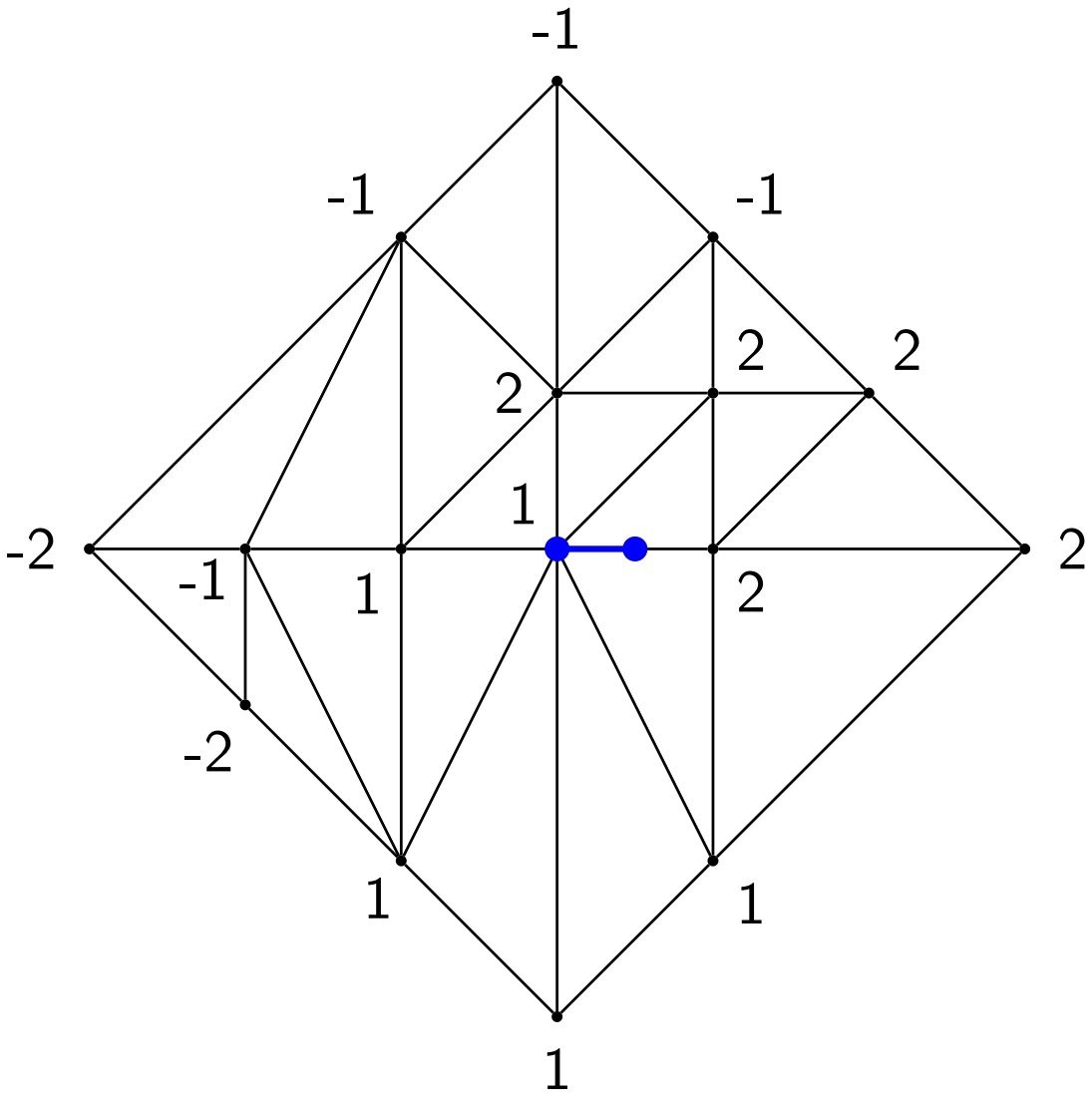
Example - 2D Tucker



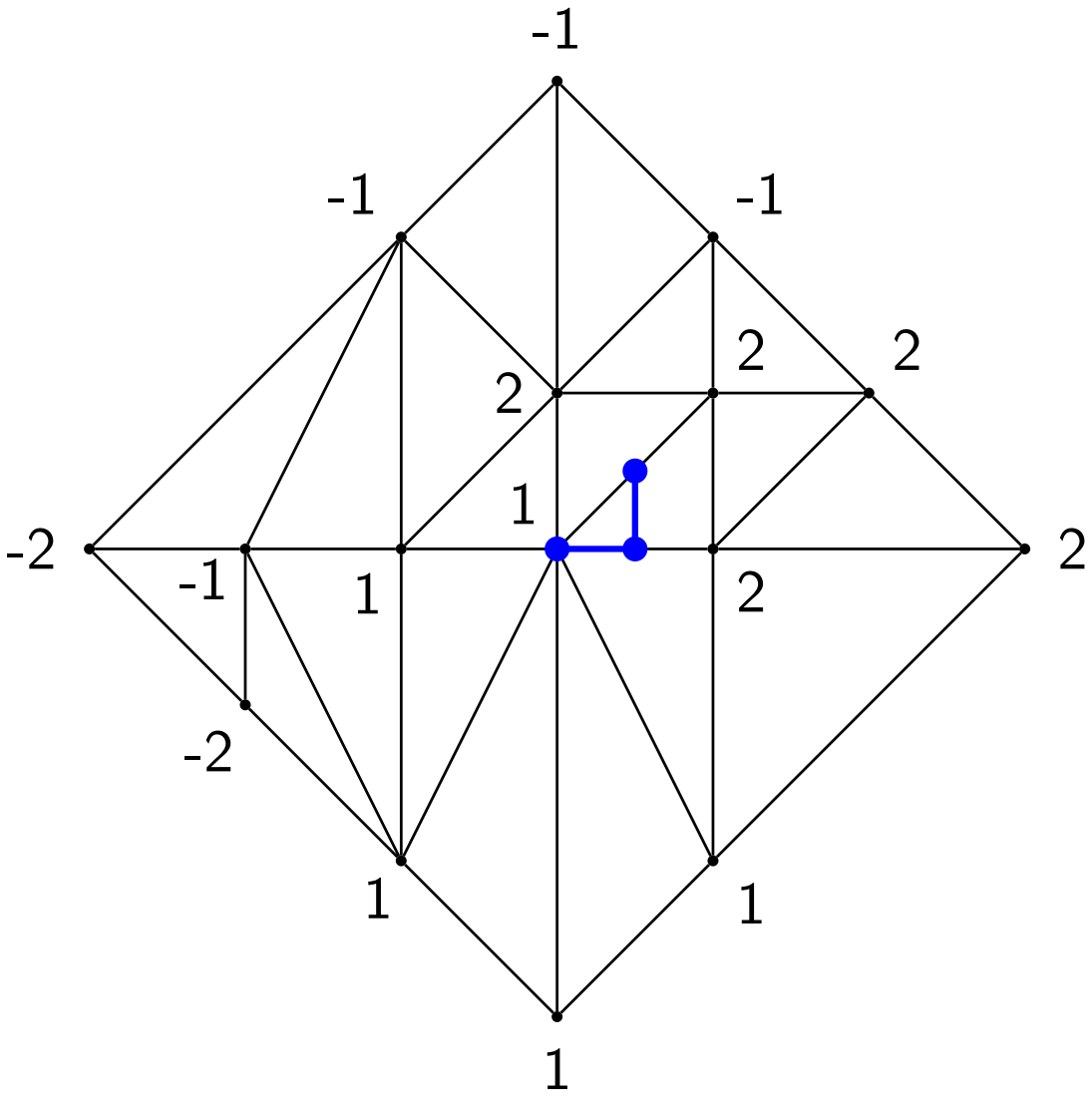
Example - 2D Tucker



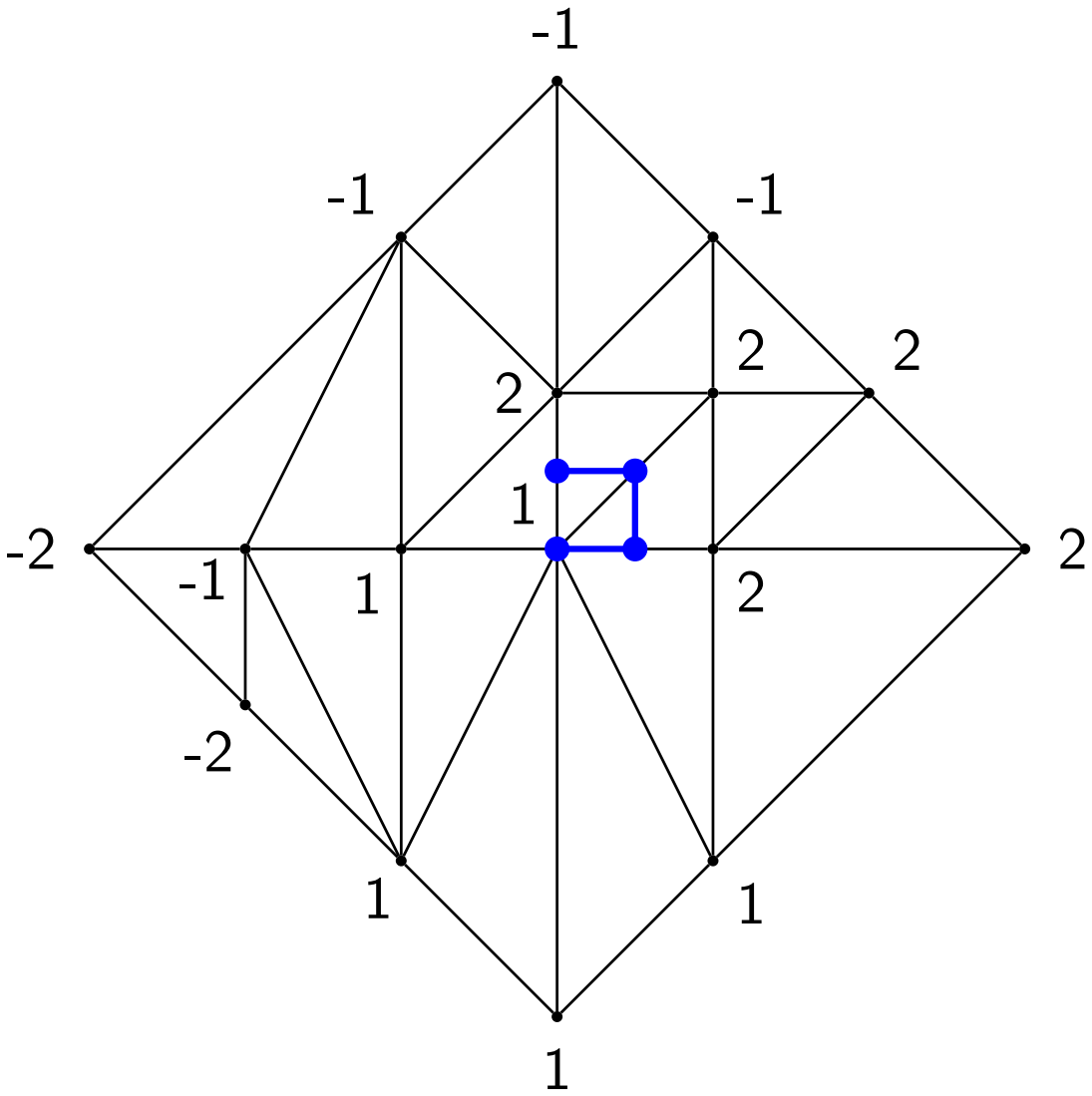
Example - 2D Tucker



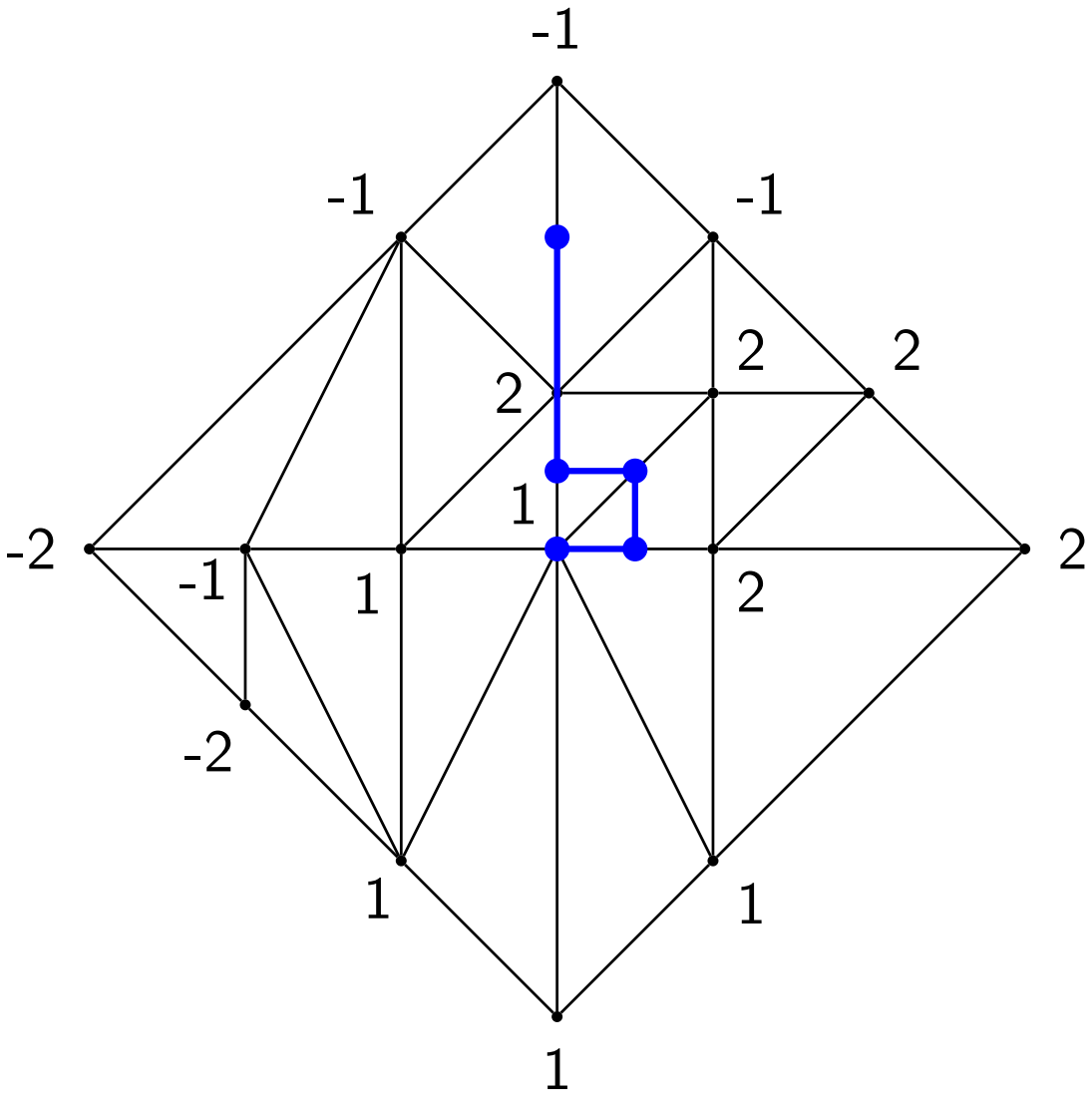
Example - 2D Tucker



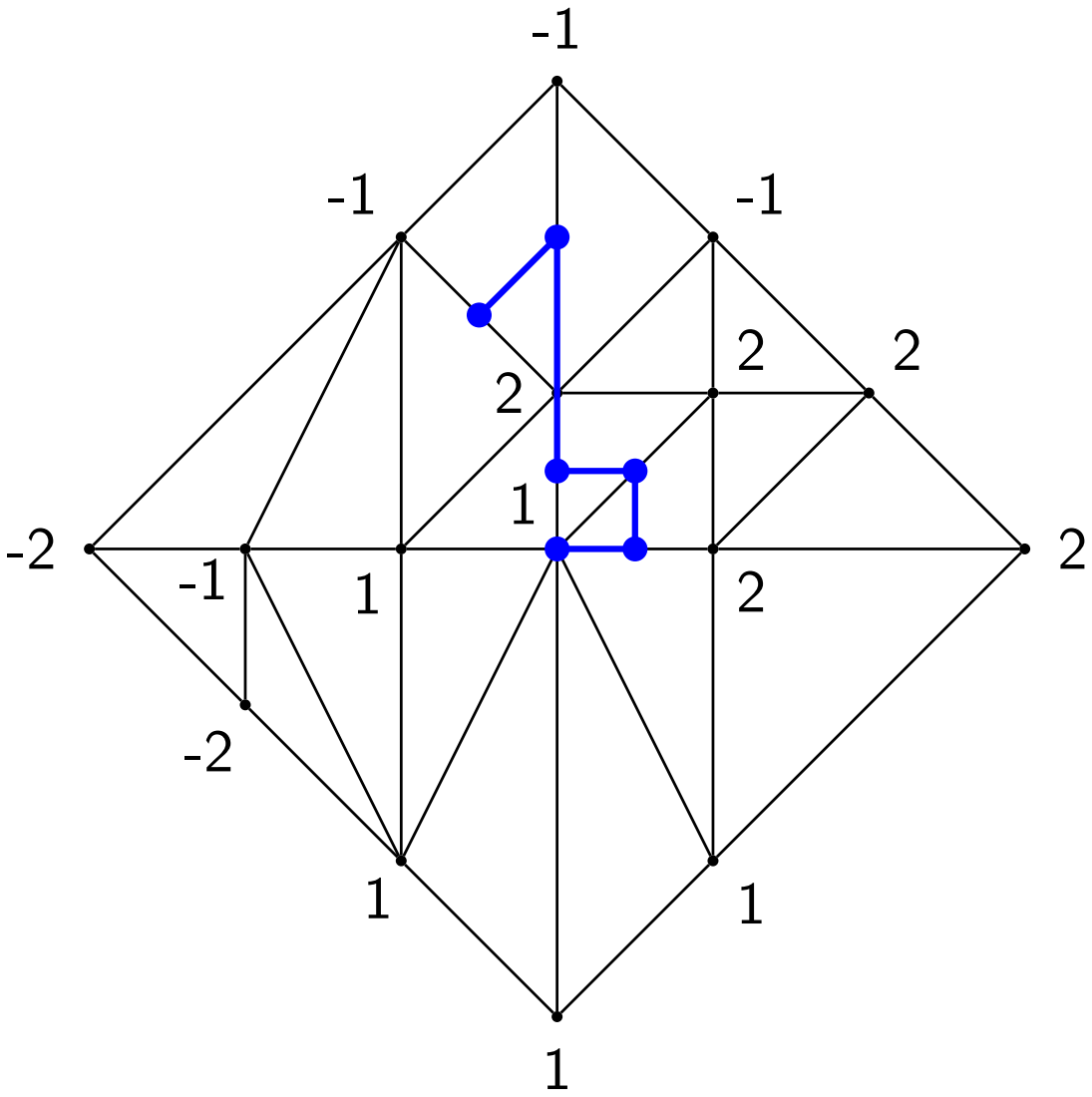
Example - 2D Tucker



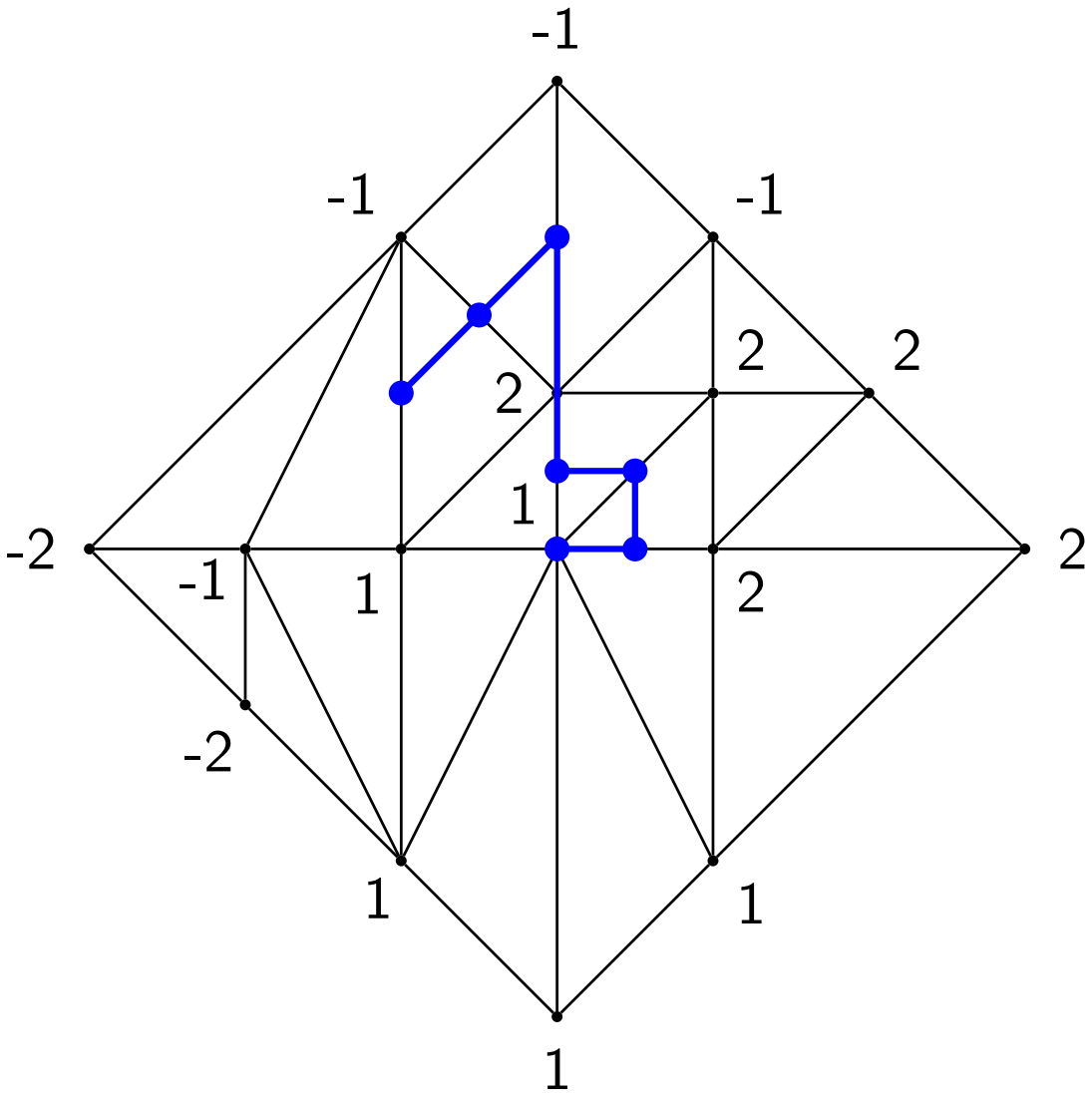
Example - 2D Tucker



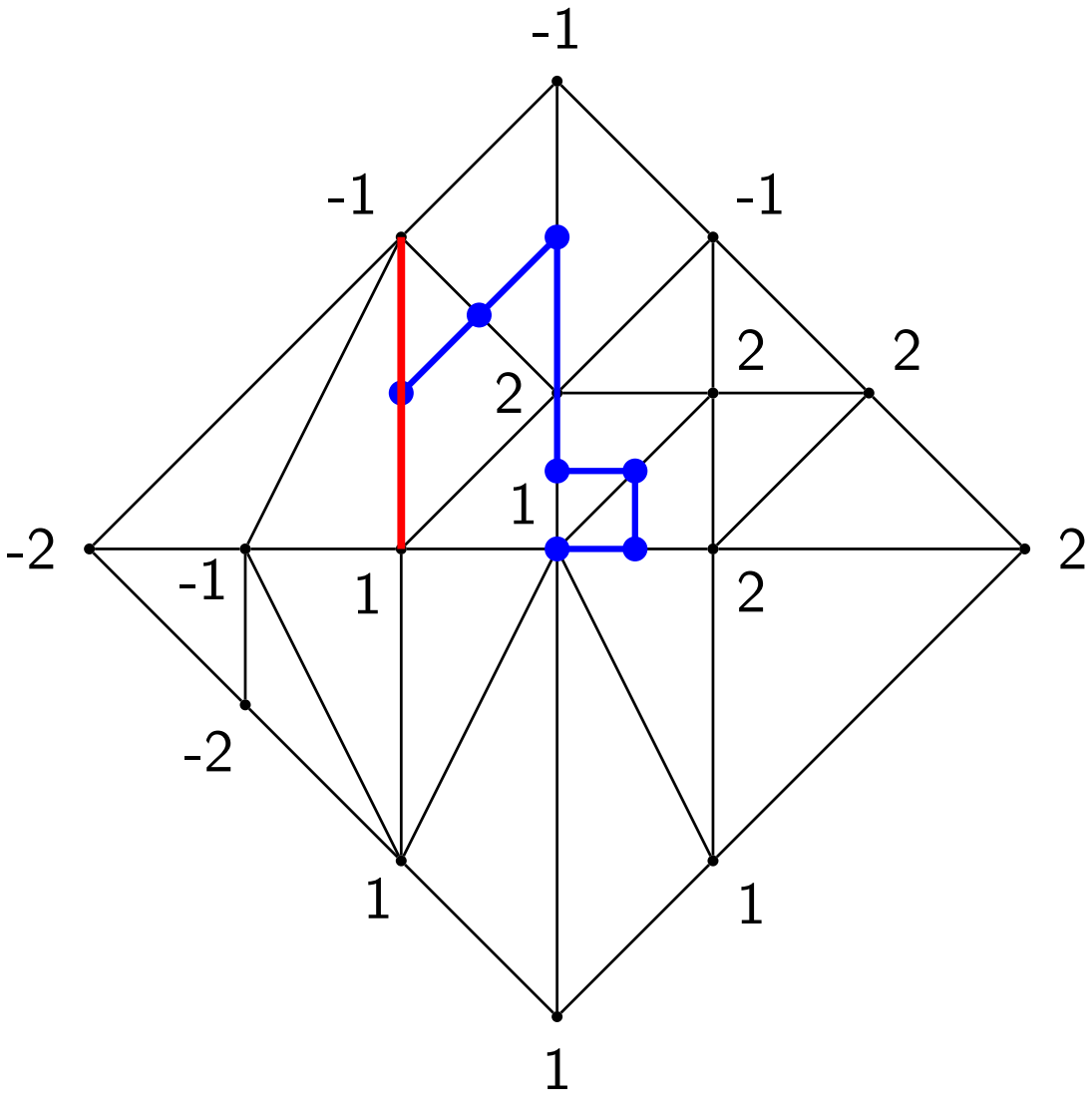
Example - 2D Tucker



Example - 2D Tucker



Example - 2D Tucker



TFNP classes PPA and PPAD (Parity Principles) [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.

In both 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.

Thm: $\textsc{Tucker}$ is in PPA. [P'94]

For PPA, there were only a few known complete problems other than 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.

Theorem (Aisenberg-Bonet-B.)

TUCKER and 2-D TUCKER are PPA-complete.

Recently, Deng, Feng and Kulkarni proved that also the Octahedral Tucker Lemma is PPA complete.

Part IV: Frege (In)Consistency Search

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

Recall: Frege proofs are textbook proof systems with finitely many axiom schemes and with modus ponens as the only rule of inference.

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 $i_1, \ldots, i_\ell \leq x$ 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 following 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.'17]

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

Recall that U_2^1 has proof complexity corresponding to polynomial space.

Corollary.

All problems from PPP, PPA, PPAD, PPADS, PLS, Ramsey, Factoring, Bertram-Chebyshev, etc., are many-one reducible to the Frege consistency search problem.

A couple related results, but using (conjecturally) stronger propositional proof systems.

Theorem. [Beckmann-B.'17; Krajíček'16]

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

Recall that V_2^1 has proof complexity corresponding to exponential time.

Theorem. [Papadimitriou-Goldberg'??]

PPA, PPPAD, PPADS, PPP, PLS are reducible to the consistency search problem for a very strong propositional logic (quantifiers, and function definitions by extension). ("WRONG PROOF").

(The Papadimitriou-Goldberg proof system is (conjecturally) stronger than even extended Frege.

In conclusion, a selected open problem:

SMITH:

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

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

The SMITH problem particularly natural since it does not implicitly involve an exponential size graph.

Thank you!