Bounded Arithmetic II: Propositional Translations

Sam Buss

Caleidoscope Research School
Institute Henri Poincaré, Paris
June 17-18, 2019
Topics:

- Formal theories of weak fragments of Peano arithmetic
  - First- and second-order theories of bounded arithmetic
- $\forall \exists$ consequences: Provably total functions
  - Computational complexity characterizations
- $\forall$ consequences: Universal statements
  - Cook translation to propositional logic
  - Paris-Wilkie translation to propositional logic

Underlying philosophy:

- A feasibly constructive proof that a function is total should provide a feasible method to compute it.
- A feasibly constructive proof of a universal statement should provide a feasible method to verify any given instance.
Cook, 1975, Feasibly constructive proofs and the propositional calculus

A constructive proof of, say, a statement $\forall x A$ must provide an effective means of finding a proof of $A$ for each value of $x$, but nothing is said about how long this proof is as a function of $x$. If the function is exponential or super exponential, then for short values of $x$ the length of the proof of the instance of $A$ may exceed the number of electrons in the universe.

Introducing PV and the Cook translation
First-/second-order theories of bounded arithmetic

\( \Pi_2 \)-consequences:
Provably total functions

\( \Pi_1 \)-consequences:
Translations to propositional logic
First-/second-order theories of bounded arithmetic

Computational complexity
Propositional proof complexity

$\Pi_2$-consequences:
Provably total functions

$\Pi_1$-consequences:
Translations to propositional logic
First-/second-order theories of bounded arithmetic

Computational complexity
Propositional proof complexity

Π₂-consequences: Provably total functions

Π₁-consequences: Translations to propositional logic

Propositional proof search (SAT solvers)
First-/second-order theories of bounded arithmetic

Π₂-consequences: Provably total functions

Π₁-consequences: Translations to propositional logic

Computational complexity
Propositional proof complexity

Propositional proof search (SAT solvers)
First-order theory $S^1_2$ of arithmetic:

- Terms have polynomial growth rate (smash, $\#$, is used).
- Bounded quantifiers $\forall x \leq t$, $\exists x \leq t$.
- Sharply bounded quantifiers $\forall x \leq |t|$, $\exists x \leq |t|$, bound $x$ by $\log$ (or length) of $t$.
- Classes $\Sigma^b_i$ and $\Pi^b_i$ of formulas are defined by counting bounded quantifiers, ignoring sharply bounded quantifiers.
- $\Sigma^b_1$ formulas express exactly the NP predicates.
- $\Sigma^b_i$, $\Pi^b_i$ - express exactly the predicates at the $i$-th level of the polynomial time hierarchy.
- $S^1_2$ has polynomial induction $\text{PIND}$, equivalently length induction ($\text{LIND}$), for $\Sigma^b_1$ formulas $A$ (i.e., NP formulas):

$$A(0) \land (\forall x)(A(x) \rightarrow A(x+1)) \rightarrow (\forall x)A(|x|)$$
(1) **Provably total functions of $S^1_2$:**

- The $\forall\Sigma^b_1$-definable functions (aka: *provably total functions*) are precisely the polynomial time computable functions.
- $PV$: equational theory over polynomial time functions. [C’75]
- $S^1_2(PV)$ is conservative over both $S^1_2$ and $PV$.

(2) **Translation to propositional logic ("Cook translation")**

- Any polynomial identity ($\forall\Sigma^b_0$-property) provable in $PV / S^1_2$, has a natural translation to a family $F$ of propositional formulas. These formulas have polynomial size extended Frege ($eF$) proofs.

(3) $S^1_2$ proves the consistency of $eF$. Conversely, any propositional proof systems (p.p.s.) $S^1_2$ proves is consistent(provably) polynomially simulated by $eF$.

(4) Lines (formulas) in an $eF$ proof correspond to Boolean circuits. The circuit value problem is complete for $P$ (polynomial time).
Equational & First-order theories of bounded arithmetic

$\Sigma^1_2$-consequences: Provably total functions

$\Pi^1_1$-consequences: Translations to propositional logic

Polynomial time functions ($P$)

extended Frege ($eF$)
Proof lines are Boolean circuits (nonuniform $P$)

PV / $S^1_2$
The first-order theory $S^1_2$ proves:
$(\forall x, n)[“\text{The bits of } x \text{ do not code an incidence matrix of a bipartite graph on } [n + 1] \cup [n] \text{ violating the Pigeonhole Principle } PHP^{n+1}_n”]

Propositional translations $PHP^{n+1}_n$: $(n \geq 1)$

\[
\bigwedge_{i=0}^{n-1} \bigvee_{j=0}^{n-1} p_{i,j} \rightarrow \bigvee_{i=0}^{n-1} \bigvee_{j=0}^{n-1} \bigwedge_{i'=i+1}^{n} (p_{i,j} \land p_{i',j})
\]

The propositional variables $p_{i,j}$ correspond to the bits of the first-order variable $x$.

Cook translation yields:
The $PHP^{n+1}_n$ formulas have polynomial size $eF$ proofs. [CR]
[Cook’75] introduced an equational theory PV of polynomial time functions. And, characterized the logical strength of PV in terms of provability in extended Frege ($eF$).

- For a polynomial time identity $f(x) = g(x)$, define a family of propositional formulas $\left[ f = g \right]_n$.
- $\left[ f = g \right]_n$ expresses that $f(x) = g(x)$ for all $x$ with $|x| < n$.
- The variables in $\left[ f = g \right]_n$ are the bits $x_0, \ldots, x_{n-1}$ of $x$.
- If PV $\vdash f(x) = g(x)$, then the formulas $\left[ f = g \right]_n$ have polynomial size extended Frege proofs. [Cook’75]

These results all lift to $S_2^1$ ...
To describe the Cook translation for $S^1_2$:

- Suppose $A(x) \in \Sigma^b_0$ (sharply bounded) and $S^1_2 \vdash \forall x \ A(x)$.
- For $n > 0$, form $[A]_n$ as a polynomial size Boolean formula.
- $[A]_n$ has Boolean variables $x_0, \ldots, x_{n-1}$ representing the bits of $x$, where $|x| \leq n$.
- $[A]_n$ expresses that "$A(x)$ is true".

Rather than formally define $[A]$, we give an example (on the next slide).

Remark: A similar construction works if all polynomial time functions are added to the language and we work with $S^1_2(PV)$. In this case, $[f=g]_n$ needs to use extension variables to define the result of polynomial size circuit computing $f(x)$ and $g(x)$. 
Simple examples of $[[A(x)]]_n : [[(\forall a \leq |x|)(a-1 < x)]]_n$

For $x$ and a $n$-bit integers, with bits given by $x_i$’s and $a_i$’s:

$[[x=a]]_n := \bigwedge_{i=0}^{n-1} (x_i \leftrightarrow a_i)$.

$[[x<a]]_n := \bigvee_{i=0}^{n-1} \left( (a_i \land \neg x_i) \land \bigwedge_{j=i+1}^{n-1} (x_j \leftrightarrow a_j) \right)$.

$[[x \leq a]]_n := [[x < a]]_n \lor [[x = a]]_n$

$i$-th bit of $x - 1$: $(x-1)_i :\Leftrightarrow \left( x_i \leftrightarrow \bigvee_{j=0}^{i-1} x_j \right) \land [[x \neq 0]]_n$

$i$-th bit of $|x|$: $\bigvee_{j \leq n, (j)_i=1} \left( x_j \land \bigvee_{k=j+1}^{n} \neg x_k \right)$

$[[(\forall a \leq |x|)(a-1 < x)]]_n := \bigwedge_{a=0}^{n} \left( [[a \leq |x|]]_n \rightarrow [[a-1 \leq x]]_n \right)$.

The sharply bounded quantifier $(\forall a \leq |x|)$ becomes a conjunction. Each of the $n+1$ values for $a$ is “hardcoded” with constants for its bits.
Theorem (essentially [Cook’75])

If $S_2^1 \vdash (\forall x)A(x)$, where $A(x)$ is in $\Delta^b_0$ (or a polynomial time identity), then the tautologies $\prod^n A(x)$ have polynomial size extended Frege proofs.

Proof construction: Witnessing Lemma again. (Proof omitted.)

Theorem ([Cook’75])

- $S_2^1 \vdash \text{Con}(e\mathcal{F})$ (the consistency of $e\mathcal{F}$).
- For any propositional proof system $\mathcal{G}$, if $S_2^1 \vdash \text{Con}(\mathcal{G})$, then $e\mathcal{F}$ p-simulates $\mathcal{G}$.

That is, $e\mathcal{F}$ is the strongest propositional proof system whose consistency is provable by $S_2^1$. 
Generalizations to $S^i_2$ and $T^i_2$.

Work in **quantified propositional logic**, with Boolean quantifiers $(\forall q), (\exists q)$ ranging over $\{T, F\}$. Sequent calculus rules now include

$$
\frac{\Gamma \rightarrow \Delta, A(B)}{\Gamma \rightarrow \Delta, (\exists q)A(q)} \quad \frac{A(q), \Gamma \rightarrow \Delta}{(\exists q)A(q), \Gamma \rightarrow \Delta}
$$

where $B$ is any formula, and $q$ appears only as indicated. (Similar rules for $\forall$.)

- Let $G_i$ be the fragment in which only $\Sigma^B_i$-formulas may occur.
- $G_i$ proofs are *dag-like*.
- Let $G_i^*$ be $G_i$ restricted to use tree-like proofs.

**Theorem (Krajíček-Pudlák’90, Cook-Morioka’05)**

*Let $i \geq 1$. Analogously to $S^1_2$ and $e\mathcal{F}$,*

- $S^i_2$ corresponds to $G_i^*$.
- $T^i_2$ corresponds to $G_i$. 
Propositional proof systems ($\mathcal{F}, e\mathcal{F}, ...$)

**Frege proofs** ($\mathcal{F}$): Sequent calculus propositional system. Equivalent to a ‘textbook style’ proof system using modus ponens.

**Extended Frege proofs** ($e\mathcal{F}$): Frege systems augmented with extension rule allowing (iterated) introduction of new variables $x$ abbreviating formulas:

Extension axiom: $x \leftrightarrow \varphi$.

**$AC^0$-Frege, aka constant-depth Frege**: Frege proofs over $\land, \lor, \lnot$ with a constant bound on the number of alternations of $\land$’s and $\lor$’s. (Negations applied only to variables.)

**Quantified sequent calculus QBF** with $\forall p, \exists p$ Boolean quantifiers. $G_i$ is QBF restricted to $i$-levels of quantifiers.

Proof size = number of symbols in the proof.
(The purpose of extension is to reduce proof size.)
Open problems:

(1) Does the Frege system ($\mathcal{F}$) allow polynomial size proofs of tautologies? (Subexponential size?)

(2) Does the Frege system quasipolynomially simulate the extended Frege ($\text{eF}$) system?
   - No good combinatorial candidates for separation are known. [BBP, HT, B, AB, ...]

(3) QBF versus $\text{eF}$?
   - ($\text{eF}$ is equivalent to $\mathcal{G}_1^*$, i.e., tree-like $\mathcal{G}_1$).
Theories for polynomial space

- **PSA** - Equational theory for \( \text{PSPACE} \) functions [Dowd’78]
- **U_2^1** - Second-order theory for polynomial space [B’85]
- The \( \Sigma_{1,b}^1 \)-definable functions of **U_2^1** are precisely the \( \text{PSPACE} \) functions.
- **U_2^1(PSA)** is conservative over both **U_2^1** and PSA. [**]**
- \( \text{PSPACE} \) identities provable in **U_2^1** have natural translations to QBF formulas which have polynomial size QBF proofs.
VNC$^1$ - **Theory for NC$^1$.**

[Clote-Takeuti’92; Arai’00; Cook-Morioka’05; Cook-Nguyen’10]

- Cook translation to $\mathcal{F}$ proofs.

VL - **Theory for L.**

[Zambella’96, Perron’05, Cook-Nguyen’10]

- Cook translation to tree-like $\mathit{GL}^*$ for $\Sigma - \mathit{CNF}(2)$ formulas.

VNL - **Theory for NL.**

[Cook-Kolokolova’03, Perron’09, Cook-Nguyen’10]

- Cook translation is to a tree-like p.p.s. $\mathit{GNL}^*$ for $\Sigma$-Krom formulas.

Work in progress: New p.p.s.’s $\mathit{eLDT}$ and $\mathit{eLNDT}$ for branching programs and nondeterministic branching programs as Cook translations for VL and VNL. [B-Das-Knop, following Cook]
<table>
<thead>
<tr>
<th>Formal Theory</th>
<th>Propositional Proof System</th>
<th>Total Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV, $S_2^1$, VPV</td>
<td>$e\mathcal{F}$, $G_1^*$</td>
<td>P</td>
</tr>
<tr>
<td>$T_2^1$, $S_2^2$</td>
<td>$G_1$, $G_2^*$</td>
<td>$\leq_{1-1}(PLS)$</td>
</tr>
<tr>
<td>$T_2^2$, $S_2^3$</td>
<td>$G_2$, $G_3^*$</td>
<td>$\leq_{1-1}(CPLS)$</td>
</tr>
<tr>
<td>$T_2^i$, $S_2^{i+1}$</td>
<td>$G_i$, $G_i^{*}$</td>
<td>$\leq_{1-1}(LLi_i)$</td>
</tr>
<tr>
<td>PSA, $U_2^1$, $W_1^1$, $V_2^1$</td>
<td>QBF</td>
<td>$Pspace^{**}$</td>
</tr>
<tr>
<td></td>
<td>**</td>
<td>EXPTIME</td>
</tr>
<tr>
<td>VNC$^1$</td>
<td>Frege ($\mathcal{F}$)</td>
<td>ALogTime</td>
</tr>
<tr>
<td>VL</td>
<td>GL$^*$</td>
<td>L</td>
</tr>
<tr>
<td>VNL</td>
<td>GNL$^*$</td>
<td>NL</td>
</tr>
</tbody>
</table>

PV, PSA - equational theories.
$S_2^i$, $T_2^i$ - first order
$U_2^1$, $V_2^1$, VNC$^1$, VL, VNL, VPV - second order
<table>
<thead>
<tr>
<th>Formal Theory</th>
<th>Propositional Proof System</th>
<th>Total Functions</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV, $S_2^1$, VPV</td>
<td>$eF$, $G_1^*$</td>
<td>$P$</td>
<td>[C, B, CN]</td>
</tr>
<tr>
<td>$T_2^1$, $S_2^2$</td>
<td>$G_1$, $G_2^*$</td>
<td>$\leq_{1-1}(PLS)$</td>
<td>[B, KP, KT, BK]</td>
</tr>
<tr>
<td>$T_2^2$, $S_2^3$</td>
<td>$G_2$, $G_3^*$</td>
<td>$\leq_{1-1}(CPLS)$</td>
<td>[B, KP, KT, KST]</td>
</tr>
<tr>
<td>$T_i^1$, $S_i^{i+1}$</td>
<td>$G_i$, $G_i^*$</td>
<td>$\leq_{1-1}(LLI_i)$</td>
<td>[B, KP, KT, KNT]</td>
</tr>
<tr>
<td>PSA, $U_2^1$, $W_1^1$</td>
<td>QBF</td>
<td>$Pspace^{**}$</td>
<td>[D, B, S]</td>
</tr>
<tr>
<td>$V_2^1$</td>
<td></td>
<td>$EXPTIME$</td>
<td>[B]</td>
</tr>
<tr>
<td>$VNC^1$</td>
<td>Frege ($F$)</td>
<td>$ALogTime$</td>
<td>[CT, A; CM, CN]</td>
</tr>
<tr>
<td>VL</td>
<td>$GL^*$</td>
<td>$L$</td>
<td>[Z, P, CN]</td>
</tr>
<tr>
<td>VNL</td>
<td>$GNL^*$</td>
<td>$NL$</td>
<td>[CK, P, CN]</td>
</tr>
</tbody>
</table>

Using Cook translation to propositional proof systems (p.p.s.’s)

$F$, $eF$ - Frege and extended Frege.

$G_i$, QBF - quantified propositional logics.

Starred (*) propositional proof systems are tree-like.
<table>
<thead>
<tr>
<th>Formal Theory</th>
<th>Propositional Proof System</th>
<th>Total Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV, S₁, VPV</td>
<td>eF, G₁*</td>
<td>P</td>
</tr>
<tr>
<td>T₁, S₂</td>
<td>G₁, G₂*</td>
<td>≤₁⁻¹(PLS)</td>
</tr>
<tr>
<td>T₂, S₃</td>
<td>G₂, G₃*</td>
<td>≤₁⁻¹(CPLS)</td>
</tr>
<tr>
<td>Tᵢ, Sᵢ⁺¹</td>
<td>Gᵢ, Gᵢ⁺¹</td>
<td>≤₁⁻¹(LLIᵢ)</td>
</tr>
<tr>
<td>PSA, U₁</td>
<td>QBF</td>
<td>Pspace**</td>
</tr>
<tr>
<td>W₁, V₁</td>
<td></td>
<td>EXPTIME</td>
</tr>
<tr>
<td>V₂</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| VNC¹         | Frege (F)                | ALogTIME        |
| VL           | GL*                      | L               |
| VNL          | GNL*                     | NL              |

PLS = Polynomial local search [JPY]
CPLS = “Colored” PLS [ST]
LLI = Linear local improvement
Pause

Next: Paris-Wilkie translation
**Paris-Wilkie translation:** is a second kind of translation to propositional logic.

- The Paris-Wilkie translation applies to first-order theories with second-order predicates (free variables, $\alpha$), essentially oracles.
- Propositional variables now represent values of the second order objects $\alpha$.
- In contrast, the Cook translation uses variables for the bits of first-order objects (the function’s inputs).
- Paris-Wilkie translations are most commonly applied to fragments of $I\Delta_0(\#, \alpha)$. [P, PW, ...].
  - $\alpha$ denotes an uninterpreted second-order object (a predicate, or oracle),
  - and $\#$ is the polynomial growth rate function $x\#y = 2^{\left|x\right|\cdot\left|y\right|}$
Let $T$ be the theory $I\Delta_0$ or $I\Delta_0(\#)$.

**Thm:** [PW] If $T(\alpha)$ proves the pigeonhole principle

$$(\forall x \leq a)(\exists y < a)\alpha(x, y) \rightarrow (\exists x < x' \leq a)(\exists y < a)(\alpha(x, y) \land \alpha(x', y))$$

then $\text{PHP}_{n+1}^n$ has polynomial (quasipolynomial, resp) size $AC^0$-Frege proofs.

Recall $\text{PHP}_{n+1}^n$:

$$\bigwedge_{i=0}^{n} \bigvee_{j=0}^{n-1} p_{i,j} \rightarrow \bigvee_{i=0}^{n-1} \bigvee_{i'=i+1}^{n} \bigvee_{j=0}^{n-1} (p_{i,j} \land p_{i',j})$$

Propositional variables $p_{i,j}$ correspond to truth values of $\alpha(x, y)$. 

[Example of Paris-Wilkie translation]
On the other hand, [A,BPI,KPW],

**Thm:** $\text{PHP}^{n+1}_n$ requires exponential size $\text{AC}^0$-Frege proofs.

*Proof idea:* apply a Hastad-style switching lemma, to reduce to a proof in which all formulas are decision trees.

**Corollary:** Neither $I\Delta_0$ nor $I\Delta_0(\#)$ proves the pigeonhole principle.

But, [PWW,MPW], ...

**Thm:** $I\Delta_0(\#)$ proves the weak pigeonhole principle (replacing “$\exists y < a$” with “$\exists y < a/2$”).

**Corollary:** The propositional weak pigeonhole principle $\text{PHP}^{2n}_n$ has quasipolynomial size $\text{AC}^0$-Frege proofs.
A hierarchy of fragments of $I\Delta_0(\#)$: [B]

- $T^i_2$ - induction for $\Sigma^b_i$ predicates (the $i$-th level of the polynomial time hierarchy).
- $S^i_2$ - length induction for $\Sigma^b_i$ predicates.
- $S^1_2 \subseteq T^1_2 \preceq_{\forall \Sigma^b_2} S^2_2 \subseteq T^2_2 \preceq_{\forall \Sigma^b_3} S^3_2 \subseteq T^3_2 \preceq_{\forall \Sigma^b_4} \cdots$

**Thm:** [KPT]

- If $T^i_2 = S^{i+1}_2$, then the polynomial time hierarchy collapses.
- In fact, if $T^i_2 \preceq_{\forall \Sigma^b_{i+2}} S^{i+1}_2$, then the polynomial time hierarchy collapses.
- $T^i_2(\alpha) \neq S^{i+1}_2(\alpha)$; i.e., relative to an oracle.
\[ S^1_2(\alpha) \subseteq T^1_2(\alpha) \preceq_{\forall \Sigma^b_2(\alpha)} S^2_2(\alpha) \subseteq T^2_2(\alpha) \preceq_{\forall \Sigma^b_3(\alpha)} \cdots \]

<table>
<thead>
<tr>
<th>Formal Propositional Total</th>
<th>Paris-Wilkie translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory Proof System [K]</td>
<td>Total Functions</td>
</tr>
<tr>
<td>( T^1_2(\alpha), S^2_2(\alpha) ) **</td>
<td>( \leq_{1-1}(\text{PLS}(\alpha)) )</td>
</tr>
<tr>
<td>( T^2_2(\alpha), S^3_2(\alpha) ) res(log)</td>
<td>( \leq_{1-1}(\text{CPLS}(\alpha)) )</td>
</tr>
<tr>
<td>( T^i_2(\alpha), S^{i+1}_2(\alpha) ) depth ((i - \frac{3}{2}))-Frege</td>
<td>( \leq_{1-1}(\text{LLI}_i(\alpha)) )</td>
</tr>
</tbody>
</table>

Depth \((n + \frac{1}{2})\)-Frege means LK proofs with formulas having at most \(n+1\) alternations, the bottom level having only logarithmic fanin. \( \text{res}(\log) = \text{depth} \frac{1}{2}\)-Frege.

Sample application: \( T^2_2 \vdash \text{PHP}^2_n \). Hence, the bit-graph weak PHP has \( \text{res}(\log) \) refutations of quasipolynomial size. Likewise, any sparse instance of the weak PHP. [MPW]
Open problem:

(4) Do the theories $T^i_2(\alpha)$ have distinct (increasing) $\forall \Sigma^b_0(\alpha)$-consequences?
   - Note this would not have any (known) computational complexity implications.

(5) For $i \geq 1$, does depth $i$-Frege quasipolynomially simulate depth $(i+1)$-Frege with respect to refuting sets of clauses?
   - Note that this is the nonuniform version of Question (4).

For (5): Best results to-date are a superpolynomial separation, based on upper and lower bounds for the pigeonhole principle. [IK]

Hastad switching lemma gives exponential separation of expressibility in depth $i$ versus depth $i+1$. (!)

(5) asks: Does this extra expressiveness allow shorter proofs?
Pause
It is also interesting to study the $\forall \Sigma^b_1$-consequences of the theories $T^i_2$. These define a subset of the TFNP problems:

**Definition:** [MP, P] 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. $R(x, y)$,
- **Polynomial 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. $R(x, y)$.

Note the solution $y$ may not be unique!
**TFNP classes** need to come with a proof of totality, usually either a combinatorial principle or a formal proof.

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

**Parity principle (PPAD) [P]**
  Input: A directed graph \( G \) with in- and out-degrees \( \leq 1 \), and a vertex \( v \) of total degree 1.
  Output: Another vertex \( v' \) of total degree 1.

**Polynomial Local Search (PLS) [JPY]**
  Input: A directed graph with out-degree \( \leq 1 \), and a nonnegative cost function which strictly decreases along directed edges.
  Output: A sink vertex.
Proofs in bounded arithmetic also establish TFNP problems:

PLS - same as before

CPLS - PLS with a Herbrandized coNP ($\Pi_1^b$) accepting condition.

**RAMSEY**
- Input: an undirected graph on $n$ nodes.
- Output: a clique or co-clique of size $\frac{1}{2} \log n$.

But, now the inputs are coded with a second-order object $\alpha$.
The output is a first-order object.

**Thm.** The PLS function is provably total in $T_2^1(\alpha)$, and is many-one complete for the provably total relations of $T_2^1(\alpha)$. [BK]

**Thm.** The same holds for CPLS and $T_2^2(\alpha)$. [KST]

**Thm.** $T_2^3(\alpha)$ proves the totality of RAMSEY. [P]

See also: Game Induction [ST], Local Improvement [KNT, BB], ...
Open problems:

(6) Do the $\forall \Sigma^b_1(\alpha)$ consequences (or, the provably total functions) of $T^i_2$ form a proper hierarchy (for $i = 2, 3, 4, \ldots$)?

(7) Does $T^2_2(\alpha)$ prove the totality of RAMSEY?

The $T^3_2(\alpha)$ proof of RAMSEY is essentially a refinement of the usual inductive combinatorial proof of the Ramsey theorem (via a reduction to the pigeonhole principle). It appears that proving RAMSEY in $T^2_2(\alpha)$ would require a new method proof for Ramsey’s theorem.

See also related results and questions for the theory of approximate counting, APC$^2$. [J,KT]
TFNP problems for stronger theories:

**Consistency search** problem for Frege proofs: [BB]

Input: A (purported) Frege proof of \( \bot \).
Output: A local error in the proof.

Also introduced as the **Wrong proof** search problem [GP].

**Thm.**

- The Frege Consistency Search problem is provable in \( U^1_2(\alpha) \)
  and many-one complete for its provably total functions. [BB]
- The same holds for extended Frege and \( V^1_2(\alpha) \). [K, BB]

Here the input is coded by a second-order object; i.e., algorithms
have *oracle* access to the Frege “proof” and seek a local error.

The “standard” TFNP problems are all included in the Consistency
Search/Wrong Proof search classes for all these theories. [BB, GP]
Finis
Finis

Thank you!