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:

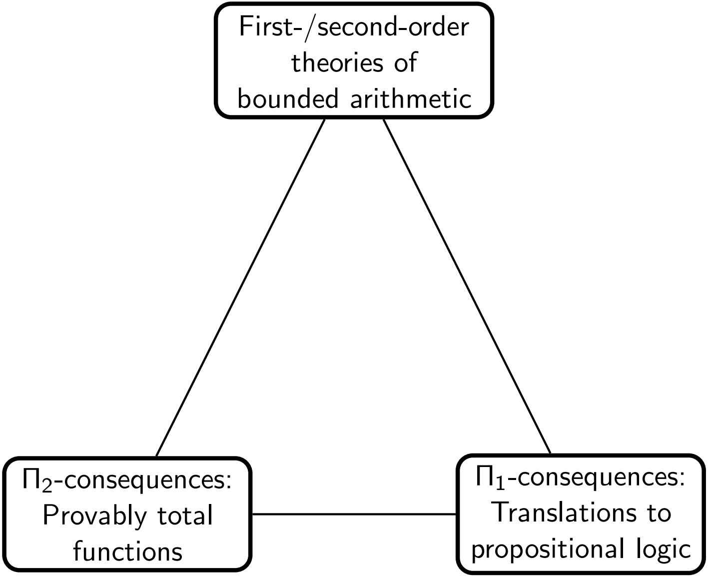
- A feasibly constructive proof that a function is total should provide a feasible method to compute it.
- A feasibly constructive proof of a universal statement should provide a feasible method to verify any given instance.

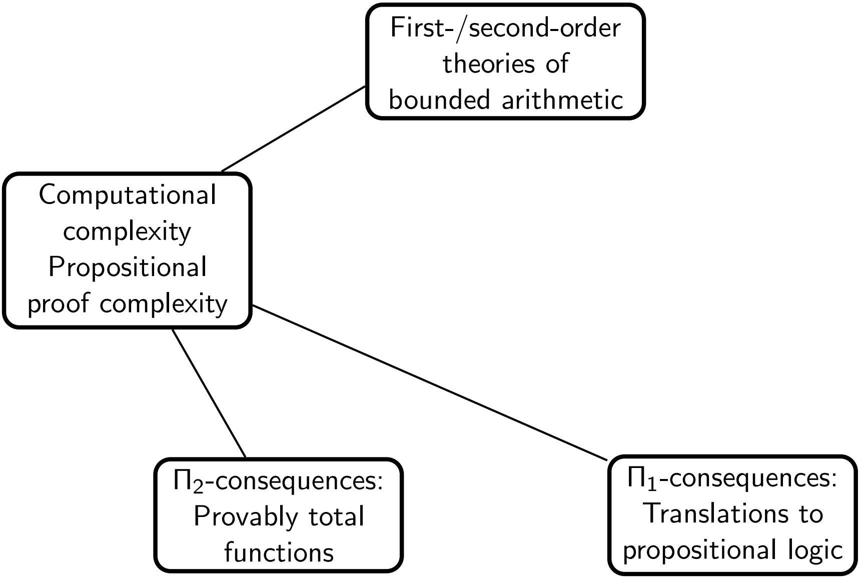
A quote

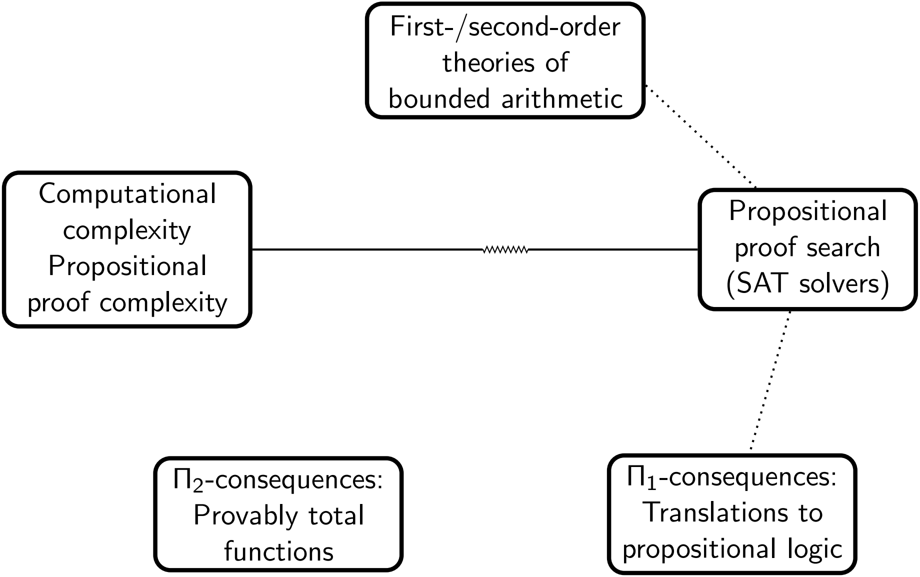
Cook, 1975, Feasibly constructive proofs and the propositional calculus

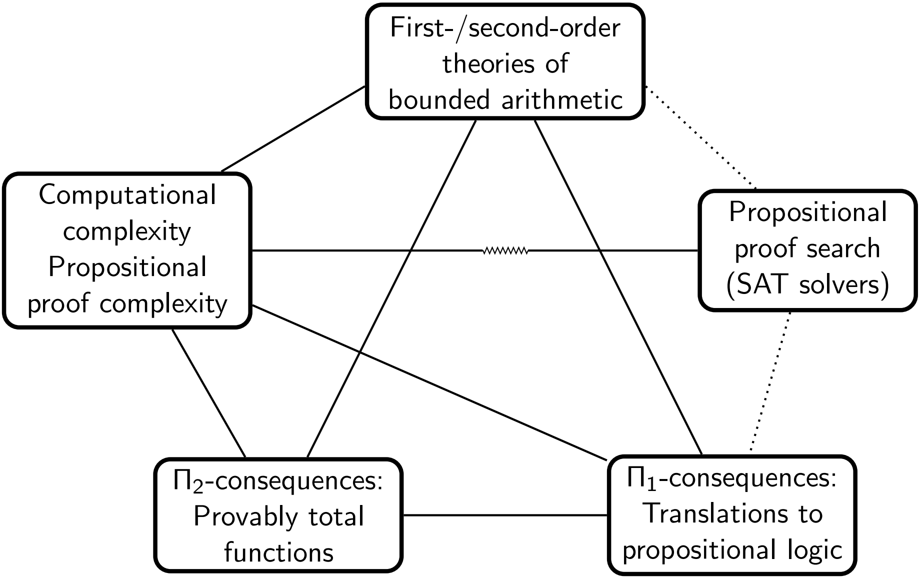
A constructive proof of, say, a statement $\forall x A$ *must provide an effective means of finding a proof of* A *for each value of* x*, but nothing is said about how long this proof is as a function of* x*. If the function is exponential or super exponential, then for short values of* x *the length 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
Π_2-consequences:
Provably total
functions
Π_1-consequences:
Translations to
propositional logic


First-/second-order theories of bounded arithmetic
Computational complexity Propositional proof complexity
Π_2-consequences: Provably total functions
Π_1-consequences: Translations to propositional logic


First-/second-order theories of bounded arithmetic
Computational complexity Propositional proof complexity
Propositional proof search (SAT solvers)
Π_2-consequences: Provably total functions
Π_1-consequences: Translations to propositional logic


First-/second-order theories of bounded arithmetic
Computational complexity Propositional proof complexity
Propositional proof search (SAT solvers)
Π_2-consequences: Provably total functions
Π_1-consequences: Translations to propositional logic

S^1_2, PV — Polynomial time — $e\mathcal{F}$ [B'85; C'76]

First-order theory S^1_2 of arithmetic:

- Terms have polynomial growth rate (smash, $\#$, is used).
- Bounded quantifiers $\forall x{\le}t$, $\exists x{\le}t$.
- Sharply bounded quantifiers $\forall x{\le}|t|$, $\exists x{\le}|t|$,

  bound x by *log* (or *length*) of t.
- Classes Σ^b_i and Π^b_i of formulas are defined by counting bounded quantifiers, ignoring sharply bounded quantifiers.
- Σ^b_1 formulas express exactly the NP predicates.

  Σ^b_i, Π^b_i - express exactly the predicates at the i-th level of the polynomial time hierarchy.
- S^1_2 has *polynomial induction* PIND, equivalently *length induction* (LIND), for Σ^b_1 formulas A (i.e., NP formulas):

$$A(0) \wedge (\forall x)(A(x) \rightarrow A(x{+}1)) \rightarrow (\forall x)A(|x|)$$

(1) Provably total functions of S^1_2:

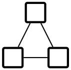
- The $\forall\Sigma^b_1$-definable functions (aka: *provably total functions*) are precisely the polynomial time computable functions.
- PV: equational theory over polynomial time functions. [C'75]
- $S^1_2(\mathrm{PV})$ is conservative over both S^1_2 and PV.

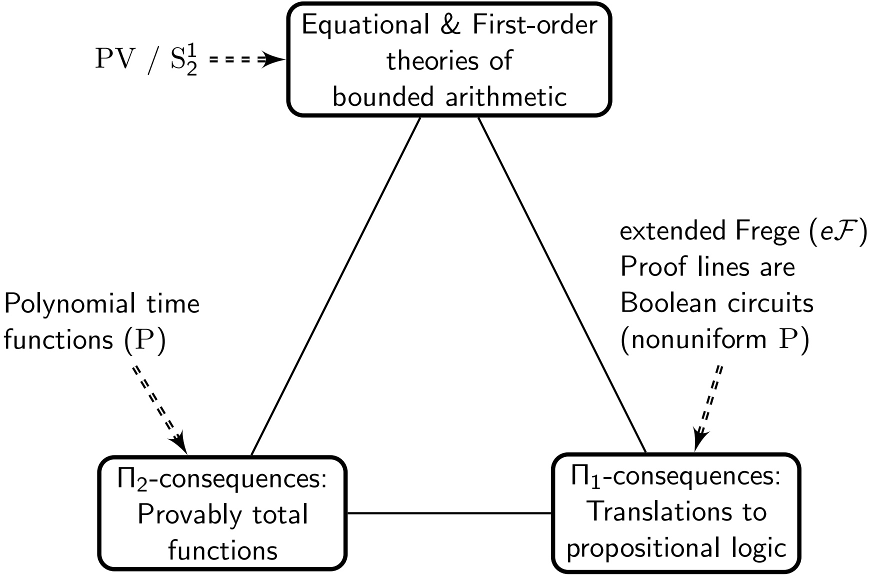
(2) Translation to propositional logic ("Cook translation")

- 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 ($e\mathcal{F}$) proofs.

(3) S^1_2 proves the consistency of $e\mathcal{F}$. Conversely, any propositional proof systems (p.p.s.) S^1_2 proves is consistent(provably) polynomially simulated by $e\mathcal{F}$.

(4) Lines (formulas) in an $e\mathcal{F}$ proof correspond to Boolean circuits. The circuit value problem is complete for P (polynomial time).




Equational & First-order theories of bounded arithmetic
PV / S_2^1
extended Frege ($e\mathcal{F}$)
Proof lines are Boolean circuits (nonuniform P)
Polynomial time functions (P)
Π_2-consequences: Provably total functions
Π_1-consequences: Translations to propositional logic

Example of Cook translation S^1_2, $e\mathcal{F}$, PHP.

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

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

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

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 $e\mathcal{F}$ proofs. [CR]

The Cook Translation from $S^1_2(\mathrm{PV})$ to $e\mathcal{F}$

[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 ($e\mathcal{F}$).

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

These results all lift to S^1_2 ...

To describe the Cook translation for S^1_2:

- Suppose $A(x) \in \Sigma^{\mathrm{b}}_0$ (sharply bounded) and $\mathrm{S}^1_2 \vdash \forall x\, A(x)$.
- 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 $\mathrm{S}^1_2(\mathrm{PV})$. 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 $[\![A(x)]\!]_n$: $[\![(\forall a{\le}|x|)(a{-}1 < x)]\!]_n$

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

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

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

$$[\![x{\le}a]\!]_n := [\![x < a]\!]_n \vee [\![x = a]\!]_n$$

i-th bit of $x-1$: $\quad (x{-}1)_i \ :\Leftrightarrow \ \Big(x_i \leftrightarrow \bigvee_{j=0}^{i-1} x_j\Big) \wedge [\![x{\ne}0]\!]_n$

i-th bit of $|x|$: $\quad \bigvee_{j\le n,(j)_i=1}\Big(x_j \wedge \bigvee_{k=j+1}^{n} \neg x_k\Big)$

$$[\![(\forall a{\le}|x|)(a{-}1 < x)]\!]_n \ := \ \bigwedge_{a=0}^{n}\Big([\![a{\le}|x|]\!]_n \to [\![a{-}1{\le}x]\!]_n\Big).$$

The sharply bounded quantifier $(\forall a{\le}|x|)$ becomes a conjunction. Each of the $n{+}1$ values for a is "hardcoded" with constants for its bits.

Theorem (essentially [Cook'75])

If $\mathrm{S}^1_2 \vdash (\forall x)A(x)$, where $A(x)$ is in Δ^{b}_0 (or a polynomial time identity), then the tautologies $[\![A(x)]\!]_n$ have polynomial size extended Frege proofs.

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

Theorem ([Cook'75])

- $\mathrm{S}^1_2 \vdash Con(e\mathcal{F})$ *(the consistency of* $e\mathcal{F}$*).*
- *For any propositional proof system* $\mathcal{G}$*, if* $\mathrm{S}^1_2 \vdash Con(\mathcal{G})$*, 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^1_2.

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

$$\frac{\Gamma \longrightarrow \Delta, A(B)}{\Gamma \longrightarrow \Delta, (\exists q)A(q)} \qquad \frac{A(q), \Gamma \longrightarrow \Delta}{(\exists q)A(q), \Gamma \longrightarrow \Delta}$$

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

- Let G_i be the fragment in which only Σ^{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: $\quad x \leftrightarrow \varphi.$

$\mathrm{AC}^{\mathbf{0}}$**-Frege, aka constant-depth Frege:** Frege proofs over $\wedge, \vee, \neg$ with a constant bound on the number of alternations of $\wedge$'s and $\vee$'s. (Negations applied only to variables.)

Quantified sequent calculus $\mathbf{QBF}$ 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 ($e\mathcal{F}$) system?

- No good combinatorial candidates for separation are known. [BBP,HT,B,AB,...]

(3) QBF versus $e\mathcal{F}$?

- ($e\mathcal{F}$ is equivalent to G_1^*, i.e., tree-like G_1).

Theories for polynomial space

- PSA - Equational theory for PSPACE functions [Dowd'78]
- U^1_2 - Second-order theory for polynomial space [B'85]
- The $\Sigma^{1,b}_1$-definable functions of U^1_2 are precisely the PSPACE functions.
- $\mathrm{U}^1_2(\mathrm{PSA})$ is conservative over both U^1_2 and PSA. [**]
- PSPACE identities provable in U^1_2 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 GL^* for $\Sigma - \mathrm{CNF}(2)$ formulas.

VNL - **Theory for** NL**.**
[Cook-Kolokolova'03, Perron'09, Cook-Nguyen'10]

- Cook translation is to a tree-like p.p.s. GNL^* for Σ-Krom formulas.

<u>Work in progress:</u> New p.p.s.'s eLDT and eLNDT for branching programs and nondeterministic branching programs as Cook translations for VL and VNL. [B-Das-Knop, following Cook]

Formal Theory	Propositional Proof System	Total Functions	
PV, S^1_2, VPV	$e\mathcal{F}$, G^*_1	P	[C, B, CN]
T^1_2, S^2_2	G_1, G^*_2	$\le_{1\text{-}1}(\mathrm{PLS})$	[B, KP, KT, BK]
T^2_2, S^3_2	G_2, G^*_3	$\le_{1\text{-}1}(\mathrm{CPLS})$	[B, KP, KT, KST]
T^i_2, S^{i+1}_2	G_i, G^*_{i+1}	$\le_{1\text{-}1}(\mathrm{LLI}_i)$	[B, KP, KT, KNT]
PSA, U^1_2, W^1_1	QBF	*Pspace***	[D, B, S]
V^1_2	**	$\mathrm{EXPTIME}$	[B]
VNC^1	Frege ($\mathcal{F}$)	ALOGTIME	[CT, A; CM, CN]
VL	GL^*	L	[Z, P, CN]
VNL	GNL^*	NL	[CK, P, CN]

PV, PSA - equational theories.
S^i_2, T^i_2 - first order
$\mathrm{U}^1_2, \mathrm{V}^1_2$, $\mathrm{VNC}^1, \mathrm{VL}, \mathrm{VNL}, \mathrm{VPV}$ - second order

Formal Theory	Propositional Proof System	Total Functions	
PV, S^1_2, VPV	$e\mathcal{F}$, G^*_1	P	[C, B, CN]
T^1_2, S^2_2	G_1, G^*_2	$\leq_{1\text{-}1}(\mathrm{PLS})$	[B, KP, KT, BK]
T^2_2, S^3_2	G_2, G^*_3	$\leq_{1\text{-}1}(\mathrm{CPLS})$	[B, KP, KT, KST]
T^i_2, S^{i+1}_2	G_i, G^*_{i+1}	$\leq_{1\text{-}1}(\mathrm{LLI}_i)$	[B, KP, KT, KNT]
PSA, U^1_2, W^1_1	QBF	*Pspace***	[D, B, S]
V^1_2	**	$\mathrm{EXPTIME}$	[B]
VNC^1	Frege ($\mathcal{F}$)	$\mathrm{A}\textsc{LogTime}$	[CT, A; CM, CN]
VL	GL^*	L	[Z, P, CN]
VNL	GNL^*	NL	[CK, P, CN]

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

$\mathcal{F}, e\mathcal{F}$ - Frege and extended Frege.

G_i, QBF - quantified propositional logics.

Starred (*) propositional proof systems are tree-like.

Formal Theory	Propositional Proof System	Total Functions	
PV, S^1_2, VPV	$e\mathcal{F}$, G^*_1	P	[C, B, CN]
T^1_2, S^2_2	G_1, G^*_2	$\le_{1\text{-}1}$(PLS)	[B, KP, KT, BK]
T^2_2, S^3_2	G_2, G^*_3	$\le_{1\text{-}1}$(CPLS)	[B, KP, KT, KST]
T^i_2, S^{i+1}_2	G_i, G^*_{i+1}	$\le_{1\text{-}1}$(LLI_i)	[B, KP, KT, KNT]
PSA, U^1_2, W^1_1	QBF	*Pspace***	[D, B, S]
V^1_2	**	EXPTIME	[B]
VNC^1	Frege ($\mathcal{F}$)	ALogTime	[CT, A; CM, CN]
VL	GL^*	L	[Z, P, CN]
VNL	GNL^*	NL	[CK, P, CN]

PLS = Polynomial local search [JPY]

CPLS = "Colored" PLS [ST]

LLI = Linear local improvement

Pause

Next: Paris-Wilkie translation

Second order arithmetic & Paris-Wilkie translations

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, α), essentially oracles.
- Propositional variables now represent values of the second order objects α.
  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, ...].

  α denotes an uninterpreted second-order object (a predicate, or oracle),

  and $\#$ is the polynomial growth rate function $x\#y = 2^{|x|\cdot|y|}$

Example of Paris-Wilkie translation

Let T be the theory $I\Delta_0$ or $I\Delta_0(\#)$.

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

$$(\forall x{\le}a)(\exists y{<}a)\alpha(x,y) \to (\exists x{<}x'{\le}a)(\exists y{<}a)(\alpha(x,y)\wedge\alpha(x',y))$$

then PHP_n^{n+1} has polynomial (quasipolynomial, resp) size AC^0-Frege proofs.

Recall PHP_n^{n+1}:

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

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

On the other hand, [A,BPI,KPW],

Thm: PHP_n^{n+1} requires exponential size 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 PHP_n^{2n} has quasipolynomial size AC^0-Frege proofs.

A hierarchy of fragments of $I\Delta_0(\#)$: [B]

- T^i_2 - induction for Σ^{b}_i predicates (the i-th level of the polynomial time hierarchy).
- S^i_2 - length induction for Σ^{b}_i predicates.
- $\mathrm{S}^1_2 \subseteq \mathrm{T}^1_2 \preccurlyeq_{\forall\Sigma^{\mathrm{b}}_2} \mathrm{S}^2_2 \subseteq \mathrm{T}^2_2 \preccurlyeq_{\forall\Sigma^{\mathrm{b}}_3} \mathrm{S}^3_2 \subseteq \mathrm{T}^3_2 \preccurlyeq_{\forall\Sigma^{\mathrm{b}}_4} \cdots$

Thm: [KPT]

- If $\mathrm{T}^i_2 = \mathrm{S}^{i+1}_2$, then the polynomial time hierarchy collapses.
- In fact, if $\mathrm{T}^i_2 \preccurlyeq_{\forall\Sigma^{\mathrm{b}}_{i+2}} \mathrm{S}^{i+1}_2$, then the polynomial time hierarchy collapses.
- $\mathrm{T}^i_2(\alpha) \neq \mathrm{S}^{i+1}_2(\alpha)$; i.e., relative to an oracle.

$$\mathrm{S}^1_2(\alpha) \subseteq \mathrm{T}^1_2(\alpha) \preccurlyeq_{\forall\Sigma^b_2(\alpha)} \mathrm{S}^2_2(\alpha) \subseteq \mathrm{T}^2_2(\alpha) \preccurlyeq_{\forall\Sigma^b_3(\alpha)} \cdots$$

Formal Theory	Paris-Wilkie translation Propositional Proof System [K]	Total Functions
$\mathrm{T}^1_2(\alpha)$, $\mathrm{S}^2_2(\alpha)$	**	$\le_{1\text{-}1}(\mathrm{PLS}(\alpha))$
$\mathrm{T}^2_2(\alpha)$, $\mathrm{S}^3_2(\alpha)$	**res(log)**	$\le_{1\text{-}1}(\mathrm{CPLS}(\alpha))$
$\mathrm{T}^i_2(\alpha)$, $\mathrm{S}^{i+1}_2(\alpha)$	**depth $(i-\frac{3}{2})$-Frege**	$\le_{1\text{-}1}(\mathrm{LLI}_i(\alpha))$

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

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

Open problem:

(4) Do the theories $\mathrm{T}^i_2(\alpha)$ have distinct (increasing) $\forall\Sigma^{\mathrm{b}}_0(\alpha)$-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

TFNP, Provably total functions

It is also interesting to study the $\forall\Sigma_1^b$-consequences of the theories T_2^i. 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 ≤ 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 ≤ 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
$\mathbf{CPLS}$ - PLS with a Herbrandized coNP (Π_1^{b}) accepting condition.

$\mathbf{RAMSEY}$
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 α.
The output is a first-order object.

Thm. The PLS function is provably total in $\mathrm{T}_2^1(\alpha)$, and is many-one complete for the provably total relations of $\mathrm{T}_2^1(\alpha)$. [BK]

Thm. The same holds for CPLS and $\mathrm{T}_2^2(\alpha)$. [KST]

Thm. $\mathrm{T}_2^3(\alpha)$ proves the totality of RAMSEY. [P]

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

Open problems:

(6) Do the $\forall\Sigma_1^b(\alpha)$ consequences (or, the provably total functions) of T_2^i form a proper hierarchy (for $i = 2, 3, 4, \ldots$)?

(7) Does $T_2^2(\alpha)$ prove the totality of RAMSEY?

The $T_2^3(\alpha)$ 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 $\perp$.
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 $\mathrm{U}^1_2(\alpha)$ and many-one complete for its provably total functions. [BB]
- The same holds for extended Frege and $\mathrm{V}^1_2(\alpha)$. [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!