Introduction to Bounded Arithmetic I
First- and Second-Order Theories

Sam Buss

Workshop on Proof Complexity
Special Semester on Complexity
St. Petersburg State University

May 15, 2016

Bounded arithmetic gives a rich perspective on and a different approach to fundamental questions in computational complexity from the point of view of mathematical logic.

It joins the study of

feasible computability and complexity

with questions about

provability and axiomatizability.

Bounded Arithmetic Theories S_2^i and T_2^i and more.

- Feasible fragments of Peano arithmetic, and Primitive Recursive Arithmetic. Formulated with restricted induction axioms.
- Have close connections to "feasible" complexity classes (e.g., P, polynomial time), and near-feasible complexity classes (e.g., the polynomial time hierarchy or PSPACE).
- Have close connections to propositional proof systems.
- Have close connections to open problems in computational complexity. (E.g., P versus NP, the polynomial time hierarchy, the existence of pseudorandom number generators, and the hardness of NP search problems).

Bounded arithmetic and bounded quantifiers

Language of bounded arithmetic includes:

$$0, \quad S, \quad +, \quad \cdot, \quad \leq, \quad |x|, \quad \lfloor \tfrac{1}{2}x \rfloor, \quad x\#y, \quad \mathrm{MSP}(x,i).$$

where

$$\begin{aligned} |x| \ &:= \ \text{length of binary representation of } x. \\ x\#y &:= 2^{|x|\cdot|y|}\text{; so } |x\#y| \ = \ |x|\cdot|y|+1. \\ \mathrm{MSP}(x,i) \ &:= \ \lfloor x/2^i \rfloor. \ (\text{“most significant part”}) \end{aligned}$$

Symbols for Peano Arithmetic plus:

- $x\#y$ gives polynomial growth rate functions.
- MSP gives simple sequence coding using binary representation.
- $|x|$ and $\lfloor \frac{1}{2}x \rfloor$ - facilitate “feasible” forms of induction.

Definition

- **Bounded Quantifier:** of the form $(\forall x{\le}t)$ or $(\exists x{\le}t)$.
- **Sharply Bounded Quantifier:** of the form $(\forall x{\le}|t|)$ or $(\exists x{\le}|t|)$.

Definition

A formula is **bounded** or **sharply bounded** provided all its quantifiers are bounded or sharply bounded (resp.).

Definition (Quantifier alternation classes)

$\Delta_0^b = \Sigma_0^b = \Pi_0^b$: Sharply bounded formulas

Σ_{i+1}^b: Closure of Π_i^b under existential bounded quantification and arbitrary sharply bounded quantification, modulo prenex operations.

Π_{i+1}^b is defined dually.

Connections with polynomial time and the polynomial time hierarchy:

- All terms $t(x)$ have polynomial growth rate: $|t(x)| = |x|^{O(1)}$.
- Sharply bounded formulas ($\Delta_0^b = \Sigma_0^b = \Pi_0^b$) are polynomial time predicates.
- Σ_1^b-formulas define exactly NP properties.
- Π_1^b-formulas define exactly coNP properties.
- Σ_i^b- and Π_i^b-formulas define exactly the predicates in the classes Σ_i^p and Π_i^p at the i-th level of the polynomial time hierarchy.

Why include $\#$ (smash)?

- Gives terms of polynomial growth rate; hence connections with the polynomial time hierarchy.
- Gives the growth rate needed for convenient arithmetization of metamathematics. (E.g., the operation of substitution requires polynomial growth rate.)
- Gives a **quantifier exchange property** (together with MSP)

$$(\forall x{<}|t|)(\exists y{<}s)A(x,y) \rightarrow (\exists w < t\#s)(\forall x{<}|t|)A(x,(w)_x)$$

for suitable Gödel decoding function $(\cdot)_x$

Axioms for bounded arithmetics:

BASIC: A set of open (quantifier-free) statements defining simple properties of the function symbols. For example,

$$x + (y + z) = (x + y) + z \qquad \mathrm{MSP}(x, S(i)) = \lfloor \tfrac{1}{2}\mathrm{MSP}(x, i) \rfloor.$$

Induction axioms: Letting A range over Φ-formulas,

Φ-IND: $$A(0) \land (\forall x)(A(x) \to A(x+1)) \to (\forall x)A(x).$$

Φ-PIND: $$A(0) \land (\forall x)(A(\lfloor \tfrac{1}{2}x \rfloor) \to A(x)) \to (\forall x)A(x).$$

Φ-LIND: $$A(0) \land (\forall x)(A(x) \to A(x+1)) \to (\forall x)A(|x|).$$

Φ-PIND and Φ-LIND are "polynomially feasible" versions of induction.

The theories S_2^i and T_2^i

Definition (Fragments of bounded arithmetic, B'85)

S_2^i: $\textsc{Basic} + \Sigma_i^b$-PIND.
T_2^i: $\textsc{Basic} + \Sigma_i^b$-IND.

$S_2 = \cup_i S_2^i$ and $T_2 = \cup_i T_2^i$.

Note: T_2 is essentially $I\Delta_0 + \Omega_1$. [Parikh'71, Wilkie-Paris'87]

Theorem (B'85, B'90)

(a) $S_2^1 \subseteq T_2^1 \preccurlyeq_{\forall\Sigma_2^b} S_2^2 \subseteq T_2^2 \preccurlyeq_{\forall\Sigma_3^b} S_2^3 \subseteq \cdots$
(b) *Thus,* $S_2 = T_2$.

Proof that $T^i_2 \supset S^i_2$:

Lemma

Σ^b_i-PIND follows from Σ^b_i-LIND (over $\textsc{Basic}$*, $i \geq 1$).*

Proof: (Sketch) To prove PIND for $A(x)$, (with c a free variable)

$$A(0) \wedge (\forall x)(A(\lfloor \tfrac{1}{2}x \rfloor) \rightarrow A(x)) \rightarrow A(c)$$

use LIND on $B(i) := A(t(i))$ for $t(i) := \mathrm{MSP}(c, |c| - i)$.
For this, note $B(0)$ and $B(|c|)$ are equivalent to $A(c)$ and $A(0)$.
Also, $t(i) = \lfloor \frac{1}{2} t(i{+}1) \rfloor$, so $(\forall i)(B(i) \rightarrow B(i{+}1))$ follows from $(\forall x)(A(\lfloor \frac{1}{2}x \rfloor) \rightarrow A(x))$. $\square$

Corollary

$S^i_2 \subset T^i_2$, for $i \geq 1$.

Provably total functions and Σ_i^b-definable functions

Definition

A function $f : \mathbb{N} \to \mathbb{N}$ is **provably total** in a theory R provided there is a formula $A_f(x, y)$ satisfying

- $A_f(x, y)$ defines the graph of $f(x) = y$
- R proves $(\forall x)(\exists! y) A_f(x, y)$
- A_f is polynomial time computable.

Definition

f is Σ_1^b**-definable by** R, provided there is a Σ_1^b-formula $A(x, y)$ such that

- $R \vdash (\forall x)(\exists y \leq t) A(x, y)$ for some term t.
- $R \vdash (\forall x, y, y')(A(x, y) \land A(x, y') \to y = y')$.
- $A(x, y)$ defines the graph of f.

"Σ_i^b-definable" is defined similarly, but allowing $A \in \Sigma_i^b$.

Theorem

Any Σ_1^b-definable function in S_2^i or T_2^i can be introduced conservatively into the language of the theory with its defining axiom, and be used freely in induction formulas.

Theorem (B'85)

S_2^1 can Σ_1^b-define every polynomial time function.

(The converse holds too.)

Hence, we can w.l.o.g. assume that all polynomial time functions are present in the languages of our theories of bounded arithmetic.

Similar definitions and results hold for predicates:

Definition

A predicate P is Δ_1^b-definable in R provided there are a Σ_1^b-formula A and a Π_1^b-formula B which are R-provably equivalent and which define the predicate P.

Theorem (B'85)

Every polynomial time predicate is Δ_1^b-definable by S_2^1.

(Again, a converse holds.)

Thus, every polynomial time predicate can be conservatively introduced to S_2^i or T_2^i with its defining axioms, and used freely in induction axioms.

Main Theorem for S^1_2

The converses of the last theorems also hold: S^1_2 can Σ^b_1-define *exactly* the polynomial time functions.

Theorem (Main Theorem for S^1_2, B'85)

Suppose f is Σ^b_1-defined by S^1_2. Then f is computable in polynomial time.
In fact, S^1_2 can prove f is computed by a polynomial time Turing machine.

The corresponding theorem for predicates:

Theorem

The Δ^b_1-definable predicates of S^1_2 are precisely the predicates that are S^1_2-provably in P.

These show S^1_2 has proof-theoretic strength corresponding to polynomial time computation.

- The proof of the "Main Theorem for S^1_2" uses a "witnessing" argument.
- Applying cut elimination, there is a sequent calculus proof P of the sequent
$$\longrightarrow (\exists y \le t(c)) A(c, y).$$
in which every formula is Σ^b_1.
- The sequent calculus proof P can be read as a **computer program** for computing a y as a function c, together with a **proof of correctness** of the program.
- The program has polynomial runtime.
- The **PIND inferences** in the proof P correspond to **for-loops**.

The next three slides spell out a few more details...

Special subclasses of *prenex* formulas:

- **Strict Σ_i^b formulas (sΣ_i^b)**: Of the form

$$(\exists x_1 \le t_1)(\forall x_2 \le t_2) \cdots (Q x_i \le t_i) B(\vec{x}),$$

  where B is sharply bounded. (And subformulas of these.)

- **Sharply strict Σ_i^b formulas (ssΣ_i^b)**: Of the form

$$(\exists x_1 \le t_1)(\forall x_2 \le t_2) \cdots (Q x_i \le t_i)(\overline{Q} x_{i+1} \le |t_{i+1}|) B(\vec{x}),$$

  where B is quantifier free. (And subformulas of these.)

Proposition:

- Every Σ_i^b formula is equivalent to an $\mathrm{ss}\Sigma_i^b$ formula (provably in S_2^1).
- S_2^i may be equivalently formalized with $\mathrm{ss}\Sigma_i^b$-PIND ($i \ge 1$).

To prove the witnessing theorem, by free-cut elimination, it suffices to consider sequent calculus proofs in which every formula is an $\mathrm{ss}\Sigma_i^b$-formula.

Definition

Let $A(\vec{c})$ be $\mathrm{ss}\Sigma_i^b$. The predicate $Wit_A(\vec{c}, u)$ is defined so that

- If A is $(\exists x{\le}t)B(\vec{c}, x)$, $B \in \Delta_0^b$, then $Wit_A(\vec{c}, u)$ is the formula $u \le t \wedge B(\vec{c}, u)$.
- If A is in Δ_0^b, then $Wit_A(\vec{c}, u)$ is just $A(\vec{c})$.

We have immediately

Fact: $A(\vec{c}) \leftrightarrow (\exists u) Wit_A(\vec{c}, u)$.

Fact: Wit_A is a Δ_0^b-formula.

Theorem (Witnessing Lemma)

If $\Gamma \longrightarrow \Delta$ is an S^1_2-provable sequent of $\mathrm{ss}\Sigma^b_1$ formulas with free variables $\vec{c}$, then there is a function $f(\vec{c}, \vec{u})$ which is Σ^b_1-definable in S^1_2 and computable in polynomial time such that S^i_2 proves

$$\bigwedge_{\gamma_i \in \Gamma} Wit_{\gamma_i}(\vec{c}, u_i) \longrightarrow \bigvee_{\delta_j \in \Delta} Wit_{\delta_j}(\vec{c}, f(\vec{c}, \vec{u})).$$

The witnessing lemma is proved by induction on the number of lines in a free-cut free S^1_2-proof P of $\Gamma \longrightarrow \Delta$.

The Main Theorem for S^1_2 is an immediate corollary. □

Generalizations to $i > 1$.

Theorem (B'85)

Let $i \geq 1$. S_2^i can Σ_i^b-define every function which is polynomial time computable with an oracle from Σ_{i-1}^p.

Recall that for $i = 1$ this gave just the polynomial time functions.

Conversely:

Theorem (Main Theorem for S_2^i, B'85)

Let $i \geq 1$. Suppose f is Σ_i^b-defined by S_2^i. Then f is computable in $P^{\Sigma_{i-1}^p}$, that is, in polynomial time with an oracle for Σ_{i-1}^p.

Recall:

$$S^1_2 \subseteq T^1_2 \preccurlyeq_{\forall\Sigma^b_2} S^2_2 \subseteq T^2_2 \preccurlyeq_{\forall\Sigma^b_3} S^3_2 \subseteq \cdots$$

Theorem (B'90)

Let $i \geq 1$.

1. *S^{i+1}_2 is $\forall\exists^b_{i+1}$-conservative over T^i_2.*
2. *In particular, T^i_2 can Σ^b_{i+1} define precisely the functions in $P^{\Sigma^b_i}$.*

Proof idea:

- First show that T^i_2 can Σ^b_{i+1} define the functions in $P^{\Sigma^b_i}$.
- Second, show that T^i_2 can prove (each instance of) the Witnessing Lemma for S^{i+1}_2.

So far, we have characterized the Σ_1^b-definable functions of only S_2^1. For T_2^i and S_2^{i+1}, we have characterized only the Σ_{i+1}^b-definable functions.
We'll address this for T_2^1 next.

One extra theorem for Part I.a:

Theorem (Krajíček-Pudlák-Takeuti'91, B'95, Zambella'96)

If $T_2^i = S_2^{i+1}$, then the polynomial time hierarchy collapses (provably) — to $\Sigma_{i+1}^p/poly$ and to $B(\Sigma_{i+2}^b)$.

Pause

Part I.b: T^1_2 and PLS

Polynomial Local Search (PLS)

Inspired by Dantzig's algorithm and other local search algorithms:

Definition (JPY'88.)

A PLS problem consists of polynomial time functions: $N(x,s)$, $i(x)$, and $c(x,s)$, polynomial time predicate $F(x,s)$, and polynomial bound $b(x) \le 2^{|x|^{O(1)}}$ such that

0. $\forall x(F(x,s) \to s \le b(x))$.
1. $\forall x(F(x,i(x)))$.
2. $\forall x(N(x,s) = s \vee c(x,N(x,s)) < c(x,s))$.
3. $\forall x(F(x,s) \to F(x,N(x,s)))$.

The input is x.
A **solution** is a point s such that $F(x,s)$ and $N(x,s) = s$.

Thus, a solution is a local minimum.

Polynomial Local Search (PLS) — and more generally, any Σ_1^b-definable function of a theory of bounded arithmetic — are special kinds of TFNP, Total NP Search, Problems:

Definition (Poljak-Turzík-Pudlák'82, JPY'88, Papadimitriou'94)

TFNP, the class of Total NP Functions is the set of polynomial time relations $R(x, y)$ such that $R(x, y)$ implies $|y| = |x|^{O(1)}$ and such that R is *total*, i.e., for all x, there exists y s.t. $R(x, y)$.

T^1_2 and PLS [B-Krajíček'94]

A *Polynomial Local Search* PLS is formalized in S^1_2 provided its feasible set, initial point function, neighborhood function, and cost function are Σ^b_1-defined (as polynomial time functions).

Theorem

T^1_2 can prove that any (formalized) PLS problem is total.

Proof: By Σ^b_1-minimization, T^1_2 can prove there is a minimum cost value c_0 satisfying

$$(\exists s \leq b(x))(F(x,s) \wedge c(x,s) = c_0).$$

Choosing s that realizes the cost c_0 gives either a solution to the PLS problem or a place where the PLS conditions are violated. $\square$

Open: Can T^1_2 witness PLS problems using single-valued Σ^b_1-definable functions?

Theorem (B-Krajíček'94)

If $A \in \Sigma_1^b$ and $T_2^1 \vdash (\forall x)(\exists y)A(x,y)$, then there is a PLS *problem R such that T_2^1 proves*

$$(\forall x)(\forall y)(R(x,y) \to A(x,(y)_1)).$$

If $A \in \Delta_1^b$, then can replace "$(y)_1$" with just "y".

This gives an exact complexity characterization of the $\forall\Sigma_1^b$-definable functions of T_2^1, in terms of PLS-computability. Namely:

Theorem

The Σ_1^b-definable (multi)functions of T_2^1 are precisely the projections of PLS functions.

Theorem (Witnessing Lemma)

If $\Gamma \longrightarrow \Delta$ *is a* T^1_2*-provable sequent of* $\mathrm{ss}\Sigma^b_1$ *formulas with free variables* $\vec{c}$*, then there is a* PLS *problem* $R(\langle \vec{c}, \vec{u} \rangle, v)$ *so that* T^1_2 *proves*

$$Wit_\Gamma(\vec{c}, \vec{u}) \wedge R(\langle \vec{c}, \vec{u} \rangle, v) \to Wit_\Delta(\vec{c}, v).$$

Proof idea: Use a free-cut free T^1_2-proof, proceed by induction on number of inferences in the proof. Arguments are similar to to what was used to prove the witnessing lemma for S^i_2 ($i = 1$ case). Most cases just require closure of PLS under polynomial time operations. However, induction (Σ^b_1-IND inference) now requires exponentially long iteration: this is handled via the exponentially many possible cost values. $\square$

The Theorem generalizes to $i > 1$ for T^i_2 with $\mathrm{PLS}^{\Sigma^b_{i-1}}$. We later discuss further improvements on this.

Pause

Part I.c: Second order theories U_2^1 and V_2^1

We now consider theories of bounded arithmetic formulated in a second-order language.

- Second-order variables $X, Y, Z, \ldots$ or $\alpha, \beta, \gamma, \ldots$. These range over sets of integers.
- Viewed computationally, such an X can be viewed as an oracle.
- Notation: $t \in X$ is usually written as $X(t)$.
- Second-order variables implicitly have polynomial bounds on their members. This corresponds to the fact that there is a polynomial upper bound on the size of oracle queries to X.

Relativized versions of S_2^i and T_2^i

Definition ($\Sigma_i^b(\alpha)$ and $\Pi_i^b(\alpha)$)

$\Sigma_i^b(\alpha)$ and $\Pi_i^b(\alpha)$ are defined exactly like Σ_i^b and Π_i^b but now allowing atomic formulas $\alpha(t)$.

Definition

- $S_2^i(\alpha)$ is: $\textsc{Basic} + \Sigma_i^b(\alpha)$-PIND.
- $T_2^i(\alpha)$ is: $\textsc{Basic} + \Sigma_i^b(\alpha)$-IND.

$S_2(\alpha) = T_2(\alpha) = \cup_i T_2^i(\alpha)$.

Theorem

- *The $\Sigma_1^b(\alpha)$-definable functions of $S_2^1(\alpha)$ are precisely the functions in P^α (so α is an oracle).*
- *The $\Sigma_1^b(\alpha)$-definable functions of $T_2^1(\alpha)$ are precisely the projections of PLS^α functions.*

A Hierarchy of Second-Order Formulas.

Definition

- The $\Sigma_0^{1,b} = \Pi_0^{1,b}$ formulas are the formulas with bounded first order quantifiers, but no unbounded quantifiers and no second-order quantifiers.
- (For $i \geq 0$.) The class of $\Sigma_{i+1}^{1,b}$ contains the formulas of the form $(\exists \vec{X})A(\vec{X})$ for A in $\Pi_i^{1,b}$. We also close under conjunction and disjunction.
- The class of $\Pi_{i+1}^{1,b}$-formulas is defined dually.

Informally: We count second-order quantifiers, disregard first-order quantifiers, and disallow unbounded quantifiers.

Remark: The $\Sigma_1^{1,b}$-formulas define exactly the predicates in $\mathrm{NEXPTIME}$ (nondeterministic exponential time).

The theories U_2^1 and V_2^1

Definition

- U_2^1 is $\text{Basic} + \Sigma_0^{1,b}$-CA $+ \Sigma_1^{1,b}$-PIND.
- V_2^1 is $\text{Basic} + \Sigma_0^{1,b}$-CA $+ \Sigma_1^{1,b}$-IND.

where $\Sigma_0^{1,b}$-CA (Comprehension on bounded formulas) is

$$(\exists\alpha)[(\forall x)(\alpha(x) \leftrightarrow A(x, \vec{y}, \vec{\beta}))],$$

for all $\Sigma_0^{1,b}$-formulas $A(x, \vec{y}, \vec{\beta})$.

Theorem (B'85)

The $\Sigma_1^{1,b}$-definable functions

- *of U_2^1 are precisely the* PSPACE*-functions,*
- *of V_2^1 are precisely the* EXPTIME*-functions.*

Summary of theories above

Theory	Axioms	Definable functions	Definability type
S_2^1	Σ_1^b-PIND	Poly. time (P)	Σ_1^b-definable
T_2^1	Σ_1^b-IND	Poly. Local Search (PLS)	Σ_1^b-definable
U_2^1	$\Sigma_1^{1,b}$-PIND	PSPACE	$\Sigma_1^{1,b}$-definable
V_2^1	$\Sigma_1^{1,b}$-IND	$\mathrm{EXPTIME}$	$\Sigma_1^{1,b}$-definable
S_2^i	Σ_i^b-PIND	$\mathrm{P}^{\Sigma_{i-1}^b}$	Σ_i^b-definable
T_2^i	Σ_i^b-IND	$\mathrm{PLS}^{\Sigma_{i-1}^b}$	Σ_i^b-definable

S_2^{i+1} and T_2^i have the same Σ_i^b-definable functions and the same Σ_{i+1}^b-definable functions.

Pause

Introduction to Bounded Arithmetic II
Translations to Propositional Logic

Sam Buss

Workshop on Proof Complexity
Special Semester on Complexity
St. Petersburg State University

May 15, 2016

Next topics:

- Translations from bounded arithmetic to propositional logic
  - "Cook-style" translations.
  - "Paris-Wilkie style" translations.

Frege proofs

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

Connectives: $\land$, $\lor$, $\neg$, and $\to$.

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

Axioms: Finite set of axiom schemes, e.g.: $A \land B \to A$

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: $\land$, $\lor$, $\neg$, and $\to$.

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

Axioms: Finite set of axiom schemes, e.g.: $A \land B \to A$

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 (quasi)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 functions computed by a Boolean circuits can require exponential size to express with 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

Propositional logic can also be formalized in the sequent calculus. For Γ, Δ sets of formulas, the sequent $\Gamma \longrightarrow \Delta$ means the same as $\bigwedge \Gamma \rightarrow \bigvee \Delta$.

Axioms and Rules of Inference: $A \longrightarrow A$ (axiom)

$$\neg\text{:left}\ \frac{\Gamma \longrightarrow \Delta, A}{\neg A, \Gamma \longrightarrow \Delta} \qquad \neg\text{:right}\ \frac{A, \Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, \neg A}$$

$$\wedge\text{:left}\ \frac{A, B, \Gamma \longrightarrow \Delta}{A \wedge B, \Gamma \longrightarrow \Delta} \qquad \wedge\text{:right}\ \frac{\Gamma \longrightarrow \Delta, A \qquad \Gamma \longrightarrow \Delta, B}{\Gamma \longrightarrow \Delta, A \wedge B}$$

and similar rules for $\vee$, $\wedge$, $\rightarrow$, and $\leftrightarrow$.

$$\text{Cut}\ \frac{\Gamma \longrightarrow \Delta, A \qquad A, \Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta} \qquad \frac{\Gamma \longrightarrow \Delta}{\Gamma' \longrightarrow \Delta'} \text{ where } \Gamma' \supseteq \Gamma,\ \Delta' \supseteq \Delta$$

Example: Proof of $A \to A$.

$$\to\!:\!\text{right}\ \frac{A \longrightarrow A}{\longrightarrow (A \to A)}$$

Example: $A \land B \to B \land A$

$$\wedge\!:\!\text{right}\ \dfrac{\wedge\!:\!\text{left}\ \dfrac{\text{Structural}\ \dfrac{B \longrightarrow B}{A, B \longrightarrow B}}{A \land B \longrightarrow B} \qquad \dfrac{\dfrac{A \longrightarrow A}{A, B \longrightarrow A}}{A \land B \longrightarrow A}}{A \land B \longrightarrow B \land A}$$

The Cook Translation from $S^1_2(PV)$ 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 $A(x) \in \Sigma^b_0$ (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 $S^1_2(\mathrm{PV})$. 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 $x-1$: $\qquad (x{-}1)_i \ :\Leftrightarrow \ \Big(x_i \leftrightarrow \bigvee_{j=0}^{i-1} x_j\Big) \wedge [\![x{\neq}0]\!]_n$

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

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

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 $S^1_2 \vdash (\forall x)A(x)$, 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])

- *$S^1_2 \vdash Con(e\mathcal{F})$ (the consistency of $e\mathcal{F}$).*
- *For any propositional proof system $\mathcal{G}$, if $S^1_2 \vdash Con(\mathcal{G})$, 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_2^i and T_2^i.

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_i^B-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_2^1 and $e\mathcal{F}$,

- *S_2^i corresponds to G_i^*.*
- *T_2^i corresponds to G_i.*

Pause

Part II.c: The Paris-Wilkie translations

The Paris-Wilkie ['85] translation transforms proofs in $T_2^k(\alpha)$ to constant-depth propositional sequent calculus (LK) proofs.

- Propositional variables x_i in the LK-proofs correspond to values $\alpha(i)$ of the second-order α.
- Bounded quantifiers in the $T_2^k(\alpha)$ proof become conjunctions or disjunctions.
- The depth of the propositional formulas is $\approx k$ (including small fan-in gates at the bottom).

Example of the PW translation of PHP

The statement that $\alpha(x,y)$ does not violate the pigeonhole principle can be expressed as:

$$\mathrm{PHP}^{\alpha}(a) \ := \ (\forall x{\le}a)(\exists y{<}a)(\alpha(x,y)) \rightarrow (\exists x{\le}a)(\exists x'{<}x)(\exists y{<}a)[\alpha(x,y) \wedge \alpha(x',y)].$$

For a fixed integer $n \in \mathbb{N}$, $[\![PHP^{\alpha}(a)]\!]_n$ is the propositional formula (also denoted PHP^{n+1}_n):

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

General principles for the translation:

- First order values are set to constants, and evaluated to a fixed value. There are no Boolean variables for bits of first-order objects.
- The Boolean values $\alpha(\cdots)$ become propositional variables.

Depth and Σ'-depth of LK formulas and proofs

The *depth* of a formula is the maximum nesting depth of blocks of $\wedge$'s and $\vee$'s. Literals have depth 0.

For the Paris-Wilkie translation from bounded arithmetic formulas to propositional logic, a better notion is Σ'-depth which allows small fanin at the bottom for free:

Definition

Let S be a proof size parameter (size upper bound). The formulas that have Σ'-depth d *with respect to* S are inductively defined as follows:

- If φ has size $\le \log S$, then φ has Σ'-depth 0.
- If each φ_i has Σ'-depth d, then $\bigvee_{i\in\mathcal{I}}\varphi_i$ and $\bigwedge_{i\in\mathcal{I}}\varphi_i$ have Σ'-depth $(d+1)$.

Σ'-depth d is often called "*depth* $d+\frac{1}{2}$".

Definition

Let S be a size parameter. An LK-proof P is a Σ'-depth d proof of size S provided:

- P has $\leq S$ symbols,
- Every formula in P has Σ'-depth $\leq d$, w.r.t. S.

Σ'-depth d proofs are particularly useful for translating $\mathrm{ss}\Sigma_d^b$ formulas to propositional logic. The inner, sharply bounded quantifiers correspond to the bottom level of small fanin gates.

Definitions similar to Σ'-depth given by: [K'94] of Σ-depth; [BB'03] of Θ-depth.

Theorem (Paris-Wilkie translation)

Suppose $i \geq 2$ and

- $A(a, \alpha)$ *is a* $\Sigma^b_{i-2}(\alpha)$ *formula,*
- $T^i_2(\alpha) \vdash \forall a\, A(a, \alpha)$.

Then

- *There are quasipolynomial size LK proofs* P_n *of the propositional translations* $[\![A(a, \alpha)]\!]_n$, *such that*
- P_n *consists sequents of formulas of depth* Σ'*-depth* $\leq i-2$.

Proof. (Proof omitted.) A direct translation of the $T^i_2(\alpha)$ proof gives LK proofs with all formulas Σ'-depth $\leq i$.
Exploiting special properties of these proofs, using constructions of [Razborov'94] and [Krajíček'94] (see also [Beckmann-B'05]) reduces the Σ'-depth by 2. □.

There is no injective map from $[a+1]$ to $\lfloor a/2 \rfloor$.

$$\mathrm{WPHP}^{\alpha}(a) \;:=$$
$$(\forall x{\le}a)(\exists y{<}\lfloor \tfrac{1}{2}a \rfloor)(\alpha(x,y)) \\ \quad\quad \rightarrow (\exists x{\le}a)(\exists x'{<}x)(\exists y{<}\lfloor \tfrac{1}{2}a \rfloor)[\alpha(x,y) \wedge \alpha(x',y)]$$

For a fixed integer $n \in \mathbb{N}$, $[\![WPHP^{\alpha}(a)]\!]_n$ is the propositional formula (also denoted $\mathrm{WPHP}^{n+1}_{n/2}$):

$$\bigwedge_{i=0}^{n} \bigvee_{j=0}^{n/2-1} x_{i,j} \quad \rightarrow \quad \bigvee_{i=0}^{n} \bigvee_{i'=0}^{i-1} \bigvee_{j=0}^{n/2-1} \left(x_{i,j} \wedge x_{i',j} \right).$$

Theorem (Paris-Wilkie-Woods'88, Maciel-Pitassi-Woods'00)

- $T^2_2(\alpha) \vdash \forall a\, \mathrm{WPHP}^\alpha(a)$
- *The tautologies* $\mathrm{WPHP}^{n+1}_{n/2}$ *have polynomial size LK proofs of* Σ'*-depth 0.*

Theorem (Krajíček)

$T^1_2(\alpha)$ *(hence* $S^2_2(\alpha)$*) does not prove* $\forall a\, \mathrm{WPHP}^\alpha(a)$.

Proof idea: If $S^2_2(\alpha)$ did prove $\mathrm{WPHP}^\alpha(a)$, then there would be a P^{NP} algorithm which, given size parameter a, and oracle access to α, finds a place where α fails to be a violation of the pigeonhole principle. (I.e., finds values x, x', y).
Such an algorithm can be fooled by an adversary: For each NP query, the adversary extends a partial 1-1 function so as to give the NP query the answer "Yes" if this is possible.
At the end the algorithm does not have enough information to produce the needed values x, x', y. □

The fact that we can prove the independence of $\mathrm{WPHP}^{\alpha}(a)$ from $T^1_2(\alpha)$, but not $T^2_2(\alpha)$, is typical.

Indeed, $T^2_2(\alpha)$ represents a *complexity barrier*: we lack good independence results for $T^2_2(\alpha)$.
Some comments / open questions:

- We know $T^2_2(\alpha) \neq T_2(\alpha)$. This follows from the existence of an oracle separating the polynomial time hierarchy.
- But do $T^2_2(\alpha)$ and $T^2(\alpha)$ have the same $\forall\Sigma^b_0(\alpha)$-consequences? The same $\forall\Sigma^b_1(\alpha)$-consequences?
- Similar comments apply to Jeřábek's ['09] bounded arithmetic theory APC_2 of approximate counting.
- The Ramsey theorem is known to be provable in $T^3_2(\alpha)$ [Pudlák'91]. Is it provable in $T^2_2(\alpha)$?

Comments/questions continued:

- In the non-uniform setting: are there sets of sequents of Σ'-depth 0, which have size S LK refutations of Σ'-depth 1, but which do not have quasipolynomial size (in S) LK refutations of Σ'-depth 0?
- More generally, are there super-quasipolynomial separations of depth k LK proofs from depth $k+1$ LK proofs with respect to refutating sets of clauses?
- [Razborov '95] shows that it is possible to extract natural proofs from $T^1_2(\alpha)$ proofs. Is this possible for $T^2_2(\alpha)$ proofs?

Pause

Introduction to Bounded Arithmetic III
Provable Total NP Search Problems

Sam Buss

Workshop on Proof Complexity
Special Semester on Complexity
St. Petersburg State University

May 15, 2016

Total NP Search Problems — TFNP

Definition (Poljak-Turzík-Pudlák'82; 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 for R 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).

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 $R(x,g(x,y))$ holds.

TFNP Problems from Complexity Theory

[Papadimitriou'94] identified a large number of TFNP problems:

1^{st} example:

Pigeonhole Principle, PIGEON **(PPP)**
Input: $x \in \mathbb{N}$ and injective $f : [x] \to [x-1]$ (purportedly)
Output: $a \neq b \in [x]$ 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 or via an oracle.

PPP:

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

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.*

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.*

and more ...

Σ_1^b-definable functions of bounded arithmetic give rise to TFNP problems. The first good example was the Σ_1^b-definability of PLS in T_2^1. The ones listed above are Σ_1^b-definable in U_2^1.

More examples include:

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

Π_k^p-PLS: [Beckmann-B.'09/'10]. Herbrandized PLS search problems with Π_{k-1}^p 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_k^p-PLS is many-one complete for the TFNP problems of T_2^k.

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.-Kołodziejczyk-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^k_2.

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

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 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.

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

$\perp$

Search Problem: Find the mistake in the proof!

Theorem. [Beckmann-B.'??]

The Frege consistency search problem is many-one complete for the TFNP problems of U^1_2.

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

The extended Frege consistency search problem is many-one complete for the TFNP problems of V^1_2.

Recall that U^1_2 and V^1_2 have proof complexity corresponding to polynomial space and exponential time.

the end

Thank you!