Formal Reasoning Using Low-Complexity Concepts

Lecture 1: Overview

Stephen Cook

Fall School, Prague

September, 2008

Reference:

Logical Foundations of Proof Complexity

Stephen Cook
Phuong Nguyen

To be published in the ASL Perspectives in Logic Series through Cambridge University Press

Almost-Complete draft (450 pages) now available on our web sites.

Comments and Corrections Appreciated

Two (related) aspects of Proof Complexity:

• Propositional Proof Complexity: Studies the lengths of proofs of tautologies in various proof systems.

• "Bounded Arithmetic": Studies the power of weak formal systems to prove theorems of interest in computer science.

Both are intimately related to mainstream complexity theory.

Here we mainly treat the second aspect, but we touch on the first.

Goals for **Mainstream Complexity Theory:**

(1) Classify computational problems according to complexity classes

(2) Separate (or collapse) complexity classes

Example Complexity Classes:

$$\mathbf{AC^0} \subset \mathbf{AC^0}(2) \subset \mathbf{TC^0} \subseteq \mathbf{NC}^1 \subseteq \mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{P} \subseteq \mathbf{NP}$$

Sad state of affairs concerning separation:
$\mathbf{AC^0}(6) = \mathbf{TC^0} = \ldots = \mathbf{P} = \mathbf{NP} = \mathbf{PH}$??

Analogous goals for Proof Complexity (Bounded Arithmetic):

(1) Classify theorems (of interest in computer science) according to the computational complexity of the concepts needed to prove them. ("Bounded Reverse Mathematics")

(2) Separate (or collapse) formal theories for various complexity classes.

(1) Classify theorems (of interest in computer science) according to the computational complexity of the concepts needed to prove them.

What does this mean?
Start with complexity class **P** (= polytime)

Call the associated formal theory **VP**.

We are mainly interested in Π_2 (and Π_1) theorems:

$\forall X \exists Y \varphi(X, Y)$ $\qquad$ (Y may be omitted)

where φ represents a polytime relation.

The proof must be *feasibly constructive*; i.e. it provides a polytime function $F(X)$ and a *correctness proof* of

$$\varphi(X, F(X))$$

The correctness proof must use only polytime concepts; e.g. induction on a polytime predicate.

Examples of theorems with proofs in **VP**

Kuratowski's Theorem
Hall's Theorem
Menger's Theorem
Extended Euclidean Algorithm
Linear Algebra
Hastad Switching Lemma (Razborov)

(Some may be provable with reasoning with complexity classes below **P**)

Conjecture: Fermat's Little Theorem is **not** provable in **VP**.

$$\forall X \forall A \exists D[(1 < A < X \wedge A^{X-1} \not\equiv 1 \bmod X) \rightarrow (1 < D < X \wedge D|X)]$$

If D can be found in polytime an efficient integer factoring algorithm would result.

Circuit Complexity Classes
Problems are specified by a (uniform) poly-size family $\langle C_n \rangle$ of Boolean circuits.
C_n solves problems with input length n.

$\mathbf{AC^0}$: bounded depth, unbounded fan-in $\wedge, \vee$.
(Log time hierarchy for Alternating TMs)
Contains binary $+$ but not parity or $\times$

$\mathbf{AC^0}$(2): allow unbounded fan-in parity gates.
Cannot count mod 3 [Raz 87],[Smo 87]

$\mathbf{AC^0}$(6): allow unbounded fan-in mod 6 gates.
Might be all of **PH**. (But can it count??)

$\mathbf{TC^0}$: allow threshold gates.
Contains binary $\times$

$\mathbf{NC^1}$: circuits must be trees
(Family of polysize Boolean formulas)

Reverse Math Question:
(1) Given a theorem, what is the least complexity class containing enough concepts to prove the theorem?
(Associate theory **VC** with complexity class **C**)

Examples of universal principles: $\forall X \varphi(X)$
(where now $\varphi(X)$ represents an $\mathbf{AC^0}$ relation.)

pigeonhole principle ($\mathbf{TC^0}$, not $\mathbf{AC^0}$)

planar st-connectivity principle (paths connecting diagonally opposite corners of a square must cross)
$\mathbf{AC^0}$ or $\mathbf{AC^0}(2)$

discrete Jordan curve theorem ($\mathbf{AC^0}$ or $\mathbf{AC^0}(2)$)

matrix identities ($AB = I \rightarrow BA = I$)
($\mathbf{P}$ – what about $\mathbf{NC}^2$?)

Propositional Proof Systems

(Formulas built from $\wedge, \vee, \neg, x_1, x_2, ...,$,parentheses)

Definition: A prop proof system is a polytime function F from $\{0,1\}^*$ onto tautologies.
If $F(X) = A$ then X is a proof of A.
We say F is *poly-bounded* if every tautology of length n has a proof of length $n^{O(1)}$.

Easy Theorem: A poly-bounded prop proof system exists iff $\mathbf{NP} = \mathbf{coNP}$.

Frege Systems (Hilbert style systems)
Finitely many axiom schemes and rule schemes.
Must be sound and implicationally complete.
All Frege systems are essentially equivalent.
Gentzen's propositional **LK** is an example.

Embarrassing Fact: No nontrivial lower bounds known on proof lengths for Frege systems.

(So maybe Frege systems are poly-bounded??)

Hard tautologies from combinatorial principles
(proof systems are “nonuniform theories”)

Pigeonhole Principle: If $n+1$ pigeons are placed in n holes, some hole has at least 2 pigeons.

Atoms p_{ij} (pigeon i placed in hole j)
$1 \leq i \leq n+1,\ 1 \leq j \leq n$

$\neg\mathrm{PHP}_n^{n+1}$ is the conjunction of clauses:

$(p_{i1} \vee ... \vee p_{in})$ (pigeon i placed in some hole)
$1 \leq i \leq n+1$

$(\neg p_{ik} \vee \neg p_{jk})$ (pigeons i, j not both in hole k)
$1 \leq i < j \leq n+1$, $1 \leq k \leq n$

$\neg\mathrm{PHP}_n^{n+1}$ is unsatisfiable: $O(n^3)$ clauses

Theorem (Buss) PHP_n^{n+1} has polysize Frege proofs. [$\mathbf{NC}^1$ can count pigeons and holes.]

Theorem (Ajtai) PHP_n^{n+1} does not have polysize $\mathbf{AC}^0$-Frege proofs. [$\mathbf{AC}^0$ cannot count.]

Formal Theories for Polytime Reasoning

Traditional Method: Modify **PA** (Peano Arithmetic)

Variables $x, y, z, ...$ range over $\mathbb{N} = 0, 1, 2, ...$
Vocabulary $+, \times, 0, 1, =$
Axioms: Peano postulates, recursive definition of $+, \times$, Induction axiom for every formula $A(x)$

$$[A(0) \wedge \forall x(A(x) \rightarrow A(x+1))] \rightarrow \forall y A(y)$$

To get a theory for **P** we

- add new polytime function symbols and their defining axioms
- restrict induction

Two theories for polytime reasoning based on **PA** (Peano Arithmetic):

Example 1: **PV** [Cook 75] A universal theory with symbols for all polytime functions with axioms based on Cobham's Theorem. Induction becomes a derived result, via binary search.

Example 2: $\mathbf{S}_2^1$ [Buss 86] Add 3 new polytime function symbols and appropriate axioms, and replace the **PA** Induction Scheme by PIND scheme for $\mathbf{\Sigma}_1^b$ formulas

The two theories are equivalent for $\forall\mathbf{\Sigma}_1^b$ theorems. [Buss 86]

CLAIM: Theories based on **PA** are not appropriate for small complexity classes such as $\mathbf{AC^0}$ and $\mathbf{AC^0}(2)$ because $x \cdot y$ is not a function in these classes.

We base our theories on a **Two-Sorted** ("second-order") language $\mathcal{L}_A^2$ [Zambella 96]

NOTE: The natural inputs for Turing machines and circuits are finite strings.

"number" variables $x, y, z ...$ (range over $\mathbb{N}$)

"string" variables $X, Y, Z ...$
range over **finite** subsets of $\mathbb{N}$
(arbitrary subsets of $\mathbb{N}$ for analysis)

Language $\mathcal{L}_A^2 = [0, 1, +, \cdot, |\ |; \in, \leq, =_1, =_2]$

Standard model $\mathbb{N}_2 = \langle \mathbb{N}, \text{finite}(\mathbb{N}) \rangle$

$0, 1, +, \cdot, \leq, =$ usual meaning over $\mathbb{N}$

$$|X| = \begin{cases} 1 + \sup(X) & \text{if } X \neq \emptyset \\ 0 & \text{if } X = \emptyset \end{cases}$$

$y \in X$ (set membership) (Write $X(y)$)

number terms $s, t, u ...$ defined as usual
only string terms are variables $X, Y, Z, ...$

Notation: $X(t) \equiv t \in X$, t a term

Definitions: Σ_0^B formula: All number quantifiers bounded.
No string quantifiers. (Free string variables allowed.)

Σ_1^B formula has the form

$$\exists Y_1 \le t_1 ... \exists Y_k \le t_k \ \varphi$$

$k \ge 0$, φ is Σ_0^B.

$\exists X \le t \ \varphi$ stands for $\exists X(|X| \le t \wedge \varphi)$, where t does not involve X.

Σ_1^1 is the class of formulas

$$\exists Y_1 ... \exists Y_k \varphi$$

$k \ge 0$, φ is Σ_0^B (no bounds on the string quantifiers).

Σ_i^B formulas begin with at most i blocks of bounded string quantifiers $\exists\forall\exists...$ followed by a Σ_0^B formula.

Note: Σ_i^B corresponds to **strict** $\Sigma_i^{1,b}$.

Two-Sorted Complexity Classes

In general, number inputs $x, y, z...$ are presented in unary.
String inputs $X, Y, Z, ...$ are presented as bit strings.

Definition A relation $R(\vec{x}, \vec{X})$ is in $\mathbf{AC^0}$ iff some ATM (alternating Turing machine) accepts R in time $O(\log n)$ with a constant number of alternations. [Similarly for two-sorted **P**]

Representation Theorem [BIS,I,Wrathall]
(a) The Σ_0^B formulas $\varphi(\vec{x}, \vec{X})$ represent precisely the relations $R(\vec{x}, \vec{X})$ in $\mathbf{AC^0}$.

(b) The Σ_1^B formulas represent precisely the **NP** relations.

(c) The Σ_i^B formulas, $i \geq 1$, represent precisely the Σ_i^p relations.

Function Classes and Bit Graphs

Definition If **C** is a class of relations, then the function class **FC** contains

(a) All p-bounded number-valued functions $f(\vec{x}, \vec{X})$ s.t. its graph

$$G_f(y, \vec{x}, \vec{X}) \equiv (y = f(\vec{x}, \vec{X}))$$

is in **C**.

(b) All p-bounded string-valued functions $F(\vec{x}, \vec{X})$ such that its bit graph

$$B_F(i, \vec{x}, \vec{X}) \equiv F(\vec{x}, \vec{X})(i)$$

is in **C**.

p-bounded means for some polynomial $q(\vec{x}, \vec{X})$:
$f(\vec{x}, \vec{X}) \leq q(\vec{x}, |\vec{X}|)$
$|F(\vec{x}, \vec{X})| \leq q(\vec{x}, |\vec{X}|)$

All functions in $\mathbf{FAC^0}$ must have graphs (or bit graphs) representable by Σ_0^B formulas

Example: $Plus(X,Y) = X + Y$ (binary $+$)
$Plus \in \mathbf{FAC^0}$

$$Plus(X,Y)(i) \equiv X(i) \oplus Y(i) \oplus Carry(X,Y,i)$$

$$\begin{aligned} Carry(i,X,Y) \;\; \equiv \;\; & \exists j < i[X(j) \wedge Y(j) \wedge \\ & \forall k < i(j < k \supset (X(k) \vee Y(k))] \end{aligned}$$

NON-Examples:
$X \cdot Y$ (binary multiplication) NOT in $\mathbf{FAC^0}$.
$Parity(X) \equiv X$ has an odd number of ones.
$Parity \notin \mathbf{AC^0}$ (Ajtai, FSS)
$Parity(X)$ NOT representable by a Σ_0^B formula.

Hierarchy of Theories $\mathbf{V^0} \subset \mathbf{V}^1 \subseteq \mathbf{V}^2 \subseteq ...$
All have underlying vocabulary $\mathcal{L}_A^2$
For $i \geq 1$, $\mathbf{V}^i$ is "RSUV" isomorphic to $\mathbf{S}_2^i$.

Theory $\mathbf{V^0}$ for class $\mathbf{AC^0}$
(All our theories extend $\mathbf{V^0}$)
2-BASIC Axioms:

B1. $x + 1 \neq 0$
B2. $x + 1 = y + 1 \supset x = y$
B3. $x + 0 = x$
B4. $x + (y + 1) = (x + y) + 1$
B5. $x \cdot 0 = 0$
B6. $x \cdot (y + 1) = (x \cdot y) + x$
B7. $(x \leq y \wedge y \leq x) \supset x = y$
B8. $x \leq x + y$
B9. $0 \leq x$
B10. $x \leq y \vee y \leq x$
B11. $x \leq y \leftrightarrow x < y + 1$
B12. $x \neq 0 \supset \exists y \leq x(y + 1 = x)$
L1. $X(y) \supset y < |X|$
L2. $y + 1 = |X| \supset X(y)$
SE. $[|X| = |Y| \wedge \forall i < |X|(X(i) \leftrightarrow Y(i))] \supset X = Y$

Also $\mathbf{V^0}$ needs $\mathbf{\Sigma_0^B}$-COMP (Comprehension)

$$\exists Z \leq y \forall j < y[Z(j) \leftrightarrow \varphi(j, \vec{x}, \vec{X})]$$

where $\varphi(j, \vec{x}, \vec{X})$ is a $\mathbf{\Sigma_0^B}$ formula without Z.

$\mathbf{V^0}$ proves the following
(because $|X| = 1+$ largest element of X...)

X-MIN

$$0 < |X| \supset \exists x < |X|(X(x) \wedge \forall y < x \ \neg X(y))$$

and X-IND

$$[X(0) \wedge \forall y < z(X(y) \supset X(y+1))] \supset X(z)$$

Therefore $\mathbf{V^0}$ proves (using Σ_0^B-COMP)

Σ_0^B-IND: $\quad [\varphi(0) \wedge \forall x(\varphi(x) \supset \varphi(x+1))] \supset \forall z \varphi(z)$

and

Σ_0^B-MIN: $\quad \exists x \varphi(x) \supset \exists x[\varphi(x) \wedge \neg \exists y(y < x \wedge \varphi(y))]$

where $\varphi(x)$ is any Σ_0^B-formula (with parameters).

Propositional Translations of Σ_0^B-formulas
See [C 75, PW 87]

For each $n \in \mathbb{N}$, $\varphi(X)[n]$ is propositional formula expressing $\varphi(X)$ when $|X| = n$.

The propositional variables of $\varphi(X)[n]$ are $p_0^X, \ldots, p_{n-1}^X$

Example: $Pal(X)$ says "X is a palindrome".

$$\forall y < |X|(X(y) \leftrightarrow X(|X| \dot{-} y \dot{-} 1))$$

Then $Pal(X)[4]$ is

$$(p_0^X \leftrightarrow p_3^X) \wedge (p_1^X \leftrightarrow p_2^X) \wedge (p_2^X \leftrightarrow p_1^X) \wedge (p_3^X \leftrightarrow p_0^X)$$

Theorem: (i) If $\varphi(X)$ is true then $\langle \varphi(X)[n] \rangle$ is a poly-size family of tautologies.
(ii) If $\mathbf{V^0} \vdash \varphi(X)$ then $\langle \varphi(X)[n] \rangle$ has polysize $\mathbf{AC^0}$-Frege proofs.

A **two-dimensional array** is represented by a string X. Define

$$X(i,j) = X(\langle i,j \rangle)$$

where $\langle i,j \rangle$ is a standard pairing function.

Example: $PHP(y,X)$ (Pigeonhole Principle)
This is a Σ_0^B formula.
Think $X(i,j)$ means pigeon $i \longrightarrow$ hole j.

$$\begin{array}{l} \forall i \leq y \exists j < y X(i,j) \supset \\ \exists i \leq y \exists j \leq y \exists k < y (i < j \wedge X(i,k) \wedge X(j,k)) \end{array}$$

The propositional translation
$PHP(n,X)[\langle n+1, n \rangle]$ is very close to the Pigeonhole tautologies PHP_n^{n+1}

Since these tautologies do not have polysize $\mathbf{AC^0}$-Frege proofs (Ajtai) it follows that $\mathbf{V^0}$ does not prove $PHP(y,X)$.