Factoring and bounded arithmetic

Emil Jeřábek

jerabek@math.cas.cz

http://math.cas.cz/~jerabek/

Institute of Mathematics of the Czech Academy of Sciences, Prague

Symposium on 50 Years of Complexity Theory:
A Celebration of the Work of Stephen Cook
Toronto, May 2019

Outline

1. Bounded arithmetic and witnessing theorems
2. Search problems and classes
3. Quadratic reciprocity
4. Factoring and PPA

Bounded arithmetic and witnessing theorems

Proof complexity

Background: [Cook&Nguyen'10], [Krajíček'19]

The big picture: a loose 3-way correspondence between

- Propositional proof systems
- Complexity classes
- Theories of arithmetic

Proof complexity

Background: [Cook&Nguyen'10], [Krajíček'19]

The big picture: a loose 3-way correspondence between

- Propositional proof systems
  - proof systems P for classical propositional logic: a formula is a tautology $\iff$ it has a P-proof
  - P-proofs recognizable in polynomial time
  - main measure: length of proofs
    polynomial? exponential?
  - examples: resolution, sequent calculus, polynomial calculus, ...
- Complexity classes
- Theories of arithmetic

Proof complexity

Background: [Cook&Nguyen'10], [Krajíček'19]

The big picture: a loose 3-way correspondence between

- Propositional proof systems
- Complexity classes
  - depending on setup, language or search problem classes
  - AC^0, TC^0, NC^1, P, NP, Σ^p_k, PSPACE, ...
- Theories of arithmetic

Proof complexity

Background: [Cook&Nguyen'10], [Krajíček'19]

The big picture: a loose 3-way correspondence between

- Propositional proof systems
- Complexity classes
- Theories of arithmetic
  - weak first-order theories, valid in $\langle \mathbb{N}, +, \cdot \rangle$
  - one-sorted (numbers) or two-sorted (1: unary/index numbers, 2: sets $\approx$ strings $\approx$ binary numbers)
  - induction/comprehension for restricted class of formulas
  - bounded quantifiers: $\exists x < t\, \varphi(x)$, $\forall x < t\, \varphi(x)$

Theories vs. classes

Theory T corresponds to complexity class C:

- C-problems definable in $\mathbb{N}$ by formulas from Φ_C
- these definitions "provably total" in T (and "well behaved")
- Φ_C-definable problems provably total in T are in C
  - witnessing theorems
- T can "reason" with C-concepts
  - it proves induction/comprehension schemata for Φ_C

Witnessing theorems

Theorem template:

If T proves $\forall x\, \exists y\, \varphi(x, y)$, where $\varphi \in \Phi$

$\implies$ $\exists$ a function/search problem $f \in C$ s.t.

$$\mathbb{N} \vDash \forall x\, \varphi(x, f(x))$$

Prototypical example:

Theorem (Buss): If $S^1_2 \vdash \forall x\, \exists y\, \varphi(x, y)$, where $\varphi \in \Sigma^b_1$, then there is $f \in \mathrm{FP}$ s.t. $\mathbb{N} \vDash \forall x\, \varphi(x, f(x))$

- S^1_2: a theory corresponding to P
- Σ^b_1: $\approx$ NP

Applications of witnessing theorems

Common argument: T does not prove XYZ

- Assume it does
- By FGH witnessing theorem, there is $f \in C$ that computes UVW
- UVW is not computable in C because ...
  $\Longrightarrow$ contradiction

Typically, C is relativized or there are further assumptions

Constructive applications?

In principle, witnessing may work in the forward direction:

- Prove XYZ in T
- Infer the existence of a C-algorithm computing UVW

This is uncommon:

- Formalizing proofs in T is hard:
  to get anything done, we usually need to start with C-algorithms for the main steps, and then some ...
- Algorithms extracted by cut elimination from T-proofs tend to be crude and bloated
  $\Longrightarrow$ direct construction is usually more efficient
- More people work on algorithms than on bounded arithmetic $\Rightarrow$ someone would have already thought of it

Room to make a difference

Bounded arithmetic offers one advantage:
it supports (a little bit of) abstract reasoning

Illustration: two theories corresponding to P

- PV_1 (or VPV): bare-bones P-theory
  - FP-function symbols
  - defining equations, induction for P-predicates
- S^1_2 (or V^1):
  - $+$ a form of NP-induction
- Both: provably total NP-search problems $=$ FP
- S^1_2 proves: every integer has a prime factorization!

This talk

The complexity of integer factoring

- [Buresh-Oppenheim'06] Factoring of integers N s.t.

$$N \equiv 1 \pmod 4, \qquad -1 \not\equiv \Box \pmod N,$$

  can be done in the class PPA

- [J'16] Factoring of general integers has a randomized reduction to PPA
  - [J'10] Prove the quadratic reciprocity theorem in a theory corresponding to PPA
  - Apply a witnessing theorem and an easy reduction

Search problems and classes

NP search problems

This talk: complexity of search problems, not languages

NP search problems:

- defined by a poly-time relation $R(x, y)$, $|y| \leq |x|^c$
- task: input $x \longmapsto$ output some y s.t. $R(x, y)$
- FNP $=$ class of all NP search problems
- TFNP $=$ subclass of total problems: $\forall x \, \exists y \, R(x, y)$

Examples

- Factoring: composite $N \in \mathbb{N} \longmapsto$ a proper factor of N
- FullFactoring: $N \in \mathbb{N} \longmapsto N = \prod_{i<k} P_i$, P_i primes
- ModSqRoot: $N, A \in \mathbb{N} \longmapsto B$ s.t. $B^2 \equiv A \pmod{N}$

PPA

PPA [Papadimitriou'94]:
Class of TFNP problems total due to a "parity argument"

Complete problem LEAF:

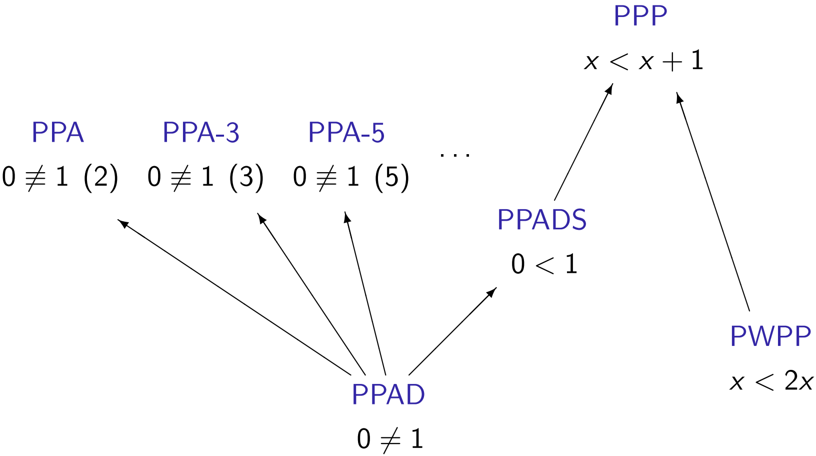
- Succinct graph G of degree ≤ 2, leaf vertex v_0
  $\longmapsto$ leaf vertex $v_1 \neq v_0$
- Total because $\sum_v \deg(v)$ is even

Alternative PPA-complete problem:

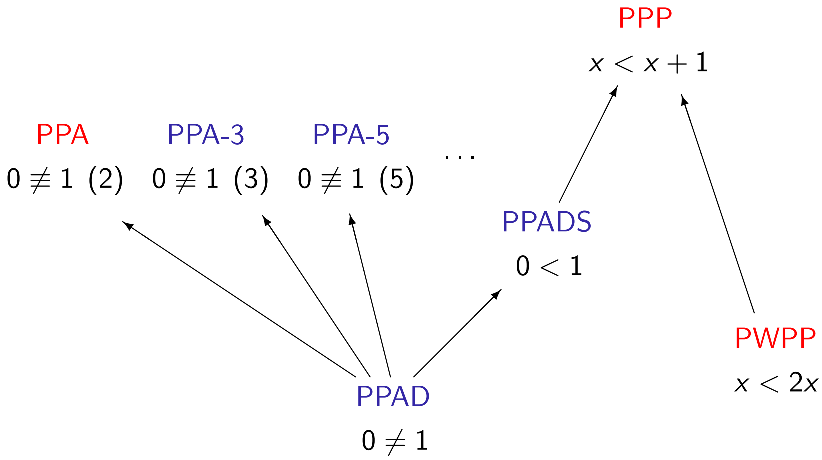
LONELY [BCEIP'98]

- Succinct partial matching on $\{0,1\}^n \smallsetminus \{0^n\}$
  $\longmapsto$ unmatched vertex

TFNP classes by counting arguments



TFNP classes by counting arguments



FACTORING has randomized reductions to PPA and PWPP

A theory for PPA

Recall: theory S^1_2 corresponds to FP

Axiom $Count_2(f)$: variant of LONELY

$$\forall a, p\, \exists x\, \bigl(f(x,{}_p) > 2a \lor f(f(x,{}_p),{}_p) \ne x \lor f(x,{}_p) = x\bigr)$$

"If $f(-,p)$ is an involution on $[2a+1]$, it has a fixpoint"

Theory $TPPA = S^1_2 + Count_2(PV)$ corresponds to PPA:

Theorem: If $TPPA \vdash \forall x\, \exists y\, \varphi(x,y)$, where $\varphi \in \Sigma^b_1$, then the search problem $x \longmapsto y$ s.t. $\varphi(x,y)$ is in PPA

Relies on [Buss&Johnson'12]: $\mathrm{FP}^{\mathrm{PPA}} = \mathrm{PPA}$

Quadratic reciprocity

Legendre symbol

$a \in \mathbb{Z}$, p odd prime:

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{if } a \equiv 0 \pmod{p} \\ 1 & \text{if } a \text{ quadratic residue} \pmod{p} \\ -1 & \text{if } a \text{ quadratic nonresidue} \pmod{p} \end{cases}$$

$a \mapsto \left(\frac{a}{p}\right)$ is a Dirichlet character:
p-periodic and completely multiplicative

$$\left(\frac{a}{p}\right)\left(\frac{b}{p}\right) = \left(\frac{ab}{p}\right)$$

Euler's criterion:

$$\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \pmod{p}$$

Quadratic reciprocity theorem

Notation: $p^* = (-1)^{\frac{p-1}{2}} p \implies p^* = \pm p$ and $p^* \equiv 1 \ (4)$

Theorem [Gauß 1801]:

- (quadratic reciprocity) If p, q are odd primes then

$$\left(\frac{q}{p}\right) = \left(\frac{p^*}{q}\right)$$

- (supplementary laws)

$$\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}} = \begin{cases} 1 & p \equiv 1 \pmod 4 \\ -1 & p \equiv -1 \pmod 4 \end{cases}$$

$$\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}} = \begin{cases} 1 & p \equiv \pm 1 \pmod 8 \\ -1 & p \equiv \pm 3 \pmod 8 \end{cases}$$

Reciprocity laws

- quadratic reciprocity conjectured by L. Euler
- incomplete proof attempt by A.-M. Legendre
- C. F. Gauß: gives 8 proofs of "aureum theorema"
  over 240 proofs published by now (cf. [Lemmermeyer'00])
- generalized and more abstract reciprocity laws
  central topic in algebraic number theory ever since
  (Hilbert r., Artin r., Langlands program, . . .)

Jacobi symbol

$a \in \mathbb{Z}$, $n > 0$ odd, $n = \prod_{i<k} p_i^{e_i}$ prime factorization:

$$\left(\frac{a}{n}\right) = \prod_{i<k} \left(\frac{a}{p_i}\right)^{e_i}$$

- $a \mapsto \left(\frac{a}{n}\right)$ still n-periodic, completely multiplicative
- ~~Euler's criterion,~~ $\cancel{\left(\frac{a}{n}\right) = 1 \implies a \equiv \square \pmod{n}}$
- $\left(\frac{a}{n}\right) \neq 0 \iff \gcd(a, n) = 1$
- quadratic reciprocity: $n, m > 0$ odd $\implies$

$$\left(\frac{m}{n}\right) = \left(\frac{n^*}{m}\right)$$

- supplementary laws

Efficient computation of $\left(\frac{A}{N}\right)$

Definition of the Jacobi symbol $\left(\frac{A}{N}\right)$ requires factorization of N

Reciprocity $\implies$ polynomial-time algorithm ($\approx$ gcd):

$r \leftarrow 1$
while $A \neq 0$ **do**:
  if $A < 0$ **then**:
    $A \leftarrow -A$
    $r \leftarrow -r$ **if** $N \equiv -1 \pmod 4$
  while A is even **do**:
    $A \leftarrow A/2$
    $r \leftarrow -r$ **if** $N \equiv \pm 3 \pmod 8$
  swap A and N
  $A \leftarrow -A$ **if** $A \equiv -1 \pmod 4$
  reduce A modulo N so that $|A| < N/2$
if $N > 1$ **then** output 0 **else** output r

Efficient computation of $\left(\frac{A}{N}\right)$

Definition of the Jacobi symbol $\left(\frac{A}{N}\right)$ requires factorization of N

Reciprocity $\implies$ polynomial-time algorithm ($\approx$ gcd):

$r \leftarrow 1$
while $A \neq 0$ **do**:
  if $A < 0$ **then**:
    $A \leftarrow -A$
    $r \leftarrow -r$ **if** $N \equiv -1 \pmod 4$
  while A is even **do**:
    $A \leftarrow A/2$
    $r \leftarrow -r$ **if** $N \equiv \pm 3 \pmod 8$
  swap A and N
  $A \leftarrow -A$ **if** $A \equiv -1 \pmod 4$
  reduce A modulo N so that $|A| < N/2$
if $N > 1$ **then** output 0 **else** output r

Efficient computation of $\left(\frac{A}{N}\right)$

Definition of the Jacobi symbol $\left(\frac{A}{N}\right)$ requires factorization of N

Reciprocity $\implies$ polynomial-time algorithm ($\approx$ gcd):

> $r \leftarrow 1$
> **while** $A \neq 0$ **do**:
>   **if** $A < 0$ **then**:
> $A \leftarrow -A$
> $r \leftarrow -r$ **if** $N \equiv -1 \pmod 4$
>   **while** A is even **do**:
> $A \leftarrow A/2$
> $r \leftarrow -r$ **if** $N \equiv \pm 3 \pmod 8$
>   swap A and N
> $A \leftarrow -A$ **if** $A \equiv -1 \pmod 4$
>   reduce A modulo N so that $|A| < N/2$
> **if** $N > 1$ **then** output 0 **else** output r

Efficient computation of $\left(\frac{A}{N}\right)$

Definition of the Jacobi symbol $\left(\frac{A}{N}\right)$ requires factorization of N

Reciprocity $\implies$ polynomial-time algorithm ($\approx$ gcd):

$r \leftarrow 1$
while $A \neq 0$ **do**:
  if $A < 0$ **then**:
    $A \leftarrow -A$
    $r \leftarrow -r$ **if** $N \equiv -1 \pmod 4$
  while A is even **do**:
    $A \leftarrow A/2$
    $r \leftarrow -r$ **if** $N \equiv \pm 3 \pmod 8$
  swap A and N
  $A \leftarrow -A$ **if** $A \equiv -1 \pmod 4$
  reduce A modulo N so that $|A| < N/2$
if $N > 1$ **then** output 0 **else** output r

Gauß's lemma in *TPPA*

Several proofs of QRT (Gauß 3&5, Eisenstein 3, ...) based on

Gauß's lemma: p odd prime, $P^+ = \{1, \dots, \frac{p-1}{2}\} \implies$

$$\left(\frac{a}{p}\right) = (-1)^{\#\{m \in P^+ : (ma \bmod p) \notin P^+\}}$$

- Only the parity of the set matters
- Key insight: One can witness the parity by explicit poly-time involutions!

Related work:

- [H-B'84], [Zag'90]: Fermat's 2-□ theorem by involutions
- [BI'91]: Supplementary laws by mod 8 counting principles

Quadratic reciprocity in *TPPA*

Theorem [J'10]: *TPPA* proves

- Quadratic reciprocity theorem + supplementary laws, for Legendre and Jacobi symbols
- Multiplicativity of Legendre and Jacobi symbols

Corollary:
TPPA proves soundness of the poly-time algorithm

There is a *PV*-function $J(a, n)$ s.t.

$$TPPA \vdash n > 0 \text{ odd} \implies J(a, n) = \left(\frac{a}{n}\right)$$

Factoring and PPA

Witnessing argument

Legendre symbol special case of Jacobi symbol
$\Longrightarrow$ the poly-time algorithm applies to it:

$$TPPA \vdash J(a,n) = 1 \wedge n \text{ prime} \implies a \equiv \Box \pmod{n}$$

Reformulate as a $\forall\Sigma^b_1$ statement:

$$TPPA \vdash J(a,n) = 1 \implies \exists b \left(b \text{ factor of } n \text{ or } a \equiv b^2 \pmod{n} \right)$$

$\Longrightarrow$ can apply the witnessing theorem for $TPPA$

Witnessing argument (cont'd)

Corollary

The problem FACROOT is in PPA:

- given A and odd $N > 0$ s.t. $\left(\frac{A}{N}\right) = 1$, find
  - a proper factor of N, or
  - a square root of A modulo N

Note:

- direct construction possible, but complicated (messy dynamic programming)
- [Buresh-Oppenheim'06]: the special case $A = -1$

Randomized reduction

Lemma: $N > 0$ odd, not a prime power $\Longrightarrow$ with prob. $\geq \frac{1}{4}$:
a random A is a quadratic nonresidue mod N s.t. $\left(\frac{A}{N}\right) = 1$

Corollary:

- $\textsc{Factoring}$ has a randomized poly-time reduction to $\textsc{FacRoot} \in \mathsf{PPA}$
- $\textsc{FullFactoring}$ has a randomized poly-time reduction to $\mathsf{FP}^{\textsc{FacRoot}} \subseteq \mathsf{PPA}$

Derandomization: $\textsc{FullFactoring} \in \mathsf{PPA}$
assuming a generalized/extended Riemann hypothesis
(for quadratic Dirichlet L-functions)

Unconditional deterministic results

Theorem: The following problems are in PPA:

- $N, A \in \mathbb{N} \longmapsto B$ s.t. $B^2 \equiv A \pmod{N}$ if it exists
- N odd, not a perfect square $\longmapsto A$ s.t. $\left(\frac{A}{N}\right) = -1$
- $N \geq 3 \longmapsto$ quadratic nonresidue $A \in (\mathbb{Z}/N\mathbb{Z})^\times$

Definition: N is M-strongly composite if N is a product of two quadratic nonresidues modulo M

Factoring of M-strongly composite numbers is in PPA for

- M constant, or
- $M = (\log N)^c!_{\longleftarrow \text{factorial, not exclamation}}$

Open problems

- Full unconditional derandomization:
  is FACTORING in PPA?
- Other classes:
  does FACTORING reduce to PPA-3 or PPAD?

Thank you for attention!

References

- P. Beame, S. Cook, J. Edmonds, R. Impagliazzo, T. Pitassi: The relative complexity of NP search problems, J. Comput. System Sci. 57 (1998), 3–19
- A. Berarducci, B. Intrigila: Combinatorial principles in elementary number theory, Ann. Pure Appl. Logic 55 (1991), 35–50
- J. Buresh-Oppenheim: On the **TFNP** complexity of factoring, `http://www.cs.toronto.edu/~bureshop/factor.pdf` (2006)
- S. R. Buss, A. S. Johnson: Propositional proofs and reductions between NP search problems, Ann. Pure Appl. Logic 163 (2012), 1163–1182
- S. A. Cook, P. Nguyen: Logical foundations of proof complexity, Cambridge Univ. Press (2010)
- C. F. Gauß: Disquisitiones arithmeticae, Fleischer, Leipzig (1801)
- R. Heath-Brown: Fermat's two squares theorem, Invariant 11 (1984), 3–5

References

- E.J.: Abelian groups and quadratic residues in weak arithmetic, Math. Logic Quart. 56 (2010), 262–278
- E.J.: Integer factoring and modular square roots, J. Comput. System Sci. 82 (2016), 380–394
- J. Krajíček: Proof complexity, Cambridge Univ. Press (2019)
- F. Lemmermeyer: Reciprocity laws, Springer (2000)
- C. H. Papadimitriou: On the complexity of the parity argument and other inefficient proofs of existence, J. Comput. System Sci. 48 (1994), 498–532
- D. Zagier: A one-sentence proof that every prime $p \equiv 1 \pmod 4$ is a sum of two squares, Amer. Math. Monthly 97 (1990), 144