Prime Factors in Models of PV_1

Ondřej Ježil
Department of Algebra, Charles University, Prague

September 8, 2025

Plan

1. Bounded arithmetic and primes
2. 'Every number has a prime factor' as a formal statement
3. The proof
4. Models of PV_1

Bounded arithmetic and complexity theory

- First order theories, weak subsystems of Peano Arithmetic.

Bounded arithmetic and complexity theory

- First order theories, weak subsystems of Peano Arithmetic.

Bounded arithmetic and complexity theory

- First order theories, weak subsystems of Peano Arithmetic.

$$\underbrace{\mathrm{PV}_1}_{\mathbf{P}\text{ induction}}$$

Bounded arithmetic and complexity theory

- First order theories, weak subsystems of Peano Arithmetic.

$$\underbrace{\mathrm{PV}_1}_{\mathbf{P}\text{ induction}} \subseteq \mathrm{PV}_1 + \underbrace{BB(\Sigma^b_0)}_{\text{sharply bounded choice}}$$

Bounded arithmetic and complexity theory

- First order theories, weak subsystems of Peano Arithmetic.

$$\underbrace{\mathrm{PV}_1}_{\textbf{P} \text{ induction}} \subseteq \mathrm{PV}_1 + \underbrace{BB(\Sigma^b_0)}_{\text{sharply bounded choice}} \subseteq \underbrace{S^1_2}_{\textbf{NP} \text{ length induction}}$$

Bounded arithmetic and complexity theory

- First order theories, weak subsystems of Peano Arithmetic.

$$\underbrace{\mathrm{PV}_1}_{\textbf{P} \text{ induction}} \subseteq \mathrm{PV}_1 + \underbrace{BB(\Sigma_0^b)}_{\text{sharply bounded choice}} \subseteq \underbrace{S_2^1}_{\textbf{NP} \text{ length induction}} \subseteq \cdots \subseteq \underbrace{T_2}_{\textbf{PH} \text{ induction}}$$

Bounded arithmetic and complexity theory

- First order theories, weak subsystems of Peano Arithmetic.

$$\underbrace{\mathrm{PV}_1}_{\textbf{P ind.}} \subseteq \mathrm{PV}_1+\underbrace{BB(\Sigma^b_0)}_{\text{sharply bounded choice}} \subseteq \underbrace{S^1_2}_{\textbf{NP ind.}} \subseteq \cdots \subseteq \underbrace{T_2}_{\textbf{PH ind.}}$$

- Already PV_1 and its mild extensions prove large number of results from complexity.

Bounded arithmetic and complexity theory

- First order theories, weak subsystems of Peano Arithmetic.

$$\underbrace{\mathrm{PV}_1}_{\textbf{P ind.}} \subseteq \mathrm{PV}_1+\underbrace{BB(\Sigma^b_0)}_{\text{sharply bounded choice}} \subseteq \underbrace{S^1_2}_{\textbf{NP ind.}} \subseteq \cdots \subseteq \underbrace{T_2}_{\textbf{PH ind.}}$$

- Already PV_1 and its mild extensions prove large number of results from complexity.
- Is PV_1 as strong as T_2?

Bounded arithmetic and complexity theory

- First order theories, weak subsystems of Peano Arithmetic.

$$\underbrace{\mathrm{PV}_1}_{\textbf{P ind.}} \subseteq \mathrm{PV}_1+\underbrace{BB(\Sigma^b_0)}_{\text{sharply bounded choice}} \subseteq \underbrace{S^1_2}_{\textbf{NP ind.}} \subseteq \cdots \subseteq \underbrace{T_2}_{\textbf{PH ind.}}$$

- Already PV_1 and its mild extensions prove large number of results from complexity.
- Is PV_1 as strong as T_2?

Bounded arithmetic and complexity theory

- First order theories, weak subsystems of Peano Arithmetic.

$$\underbrace{\mathrm{PV}_1}_{\textbf{P ind.}} \subseteq \mathrm{PV}_1+\underbrace{BB(\Sigma^b_0)}_{\text{sharply bounded choice}} \subseteq \underbrace{S^1_2}_{\textbf{NP ind.}} \subseteq \cdots \subseteq \underbrace{T_2}_{\textbf{PH ind.}}$$

- Already PV_1 and its mild extensions prove large number of results from complexity.
- Is PV_1 as strong as T_2?
  - Negative answer implies $\mathrm{PV}_1 \nvdash P = NP$,

Bounded arithmetic and complexity theory

- First order theories, weak subsystems of Peano Arithmetic.

$$\underbrace{\mathrm{PV}_1}_{\textbf{P ind.}} \subseteq \mathrm{PV}_1+\underbrace{BB(\Sigma^b_0)}_{\text{sharply bounded choice}} \subseteq \underbrace{S^1_2}_{\textbf{NP ind.}} \subseteq \cdots \subseteq \underbrace{T_2}_{\textbf{PH ind.}}$$

- Already PV_1 and its mild extensions prove large number of results from complexity.
- Is PV_1 as strong as T_2?
  - Negative answer implies $\mathrm{PV}_1 \nvdash P = NP$,

Bounded arithmetic and complexity theory

- First order theories, weak subsystems of Peano Arithmetic.

$$\underbrace{\mathrm{PV}_1}_{\textbf{P ind.}} \subseteq \mathrm{PV}_1+\underbrace{BB(\Sigma^b_0)}_{\text{sharply bounded choice}} \subseteq \underbrace{S^1_2}_{\textbf{NP ind.}} \subseteq \cdots \subseteq \underbrace{T_2}_{\textbf{PH ind.}}$$

- Already PV_1 and its mild extensions prove large number of results from complexity.
- Is PV_1 as strong as T_2?
  - Negative answer implies $\mathrm{PV}_1 \nvdash P = NP$, a major open problem.

Bounded arithmetic and complexity theory

- First order theories, weak subsystems of Peano Arithmetic.

$$\underbrace{\mathrm{PV}_1}_{\textbf{P ind.}} \subseteq \mathrm{PV}_1+\underbrace{BB(\Sigma^b_0)}_{\text{sharply bounded choice}} \subseteq \underbrace{S^1_2}_{\textbf{NP ind.}} \subseteq \cdots \subseteq \underbrace{T_2}_{\textbf{PH ind.}}$$

- Already PV_1 and its mild extensions prove large number of results from complexity.
- Is PV_1 as strong as T_2?
  - Negative answer implies $\mathrm{PV}_1 \nvdash P = NP$, a major open problem.
- Mostly conditional results (using witnessing)

Bounded arithmetic and complexity theory

- First order theories, weak subsystems of Peano Arithmetic.

$$\underbrace{\mathrm{PV}_1}_{\textbf{P ind.}} \subseteq \mathrm{PV}_1+ \underbrace{BB(\Sigma^b_0)}_{\text{sharply bounded choice}} \subseteq \underbrace{S^1_2}_{\textbf{NP ind.}} \subseteq \cdots \subseteq \underbrace{T_2}_{\textbf{PH ind.}}$$

- Already PV_1 and its mild extensions prove large number of results from complexity.
- Is PV_1 as strong as T_2?
  - Negative answer implies $\mathrm{PV}_1 \nvdash P = NP$, a major open problem.
- Mostly conditional results (using witnessing)
- Some unconditional unprovability of complexity statements is known:

Bounded arithmetic and complexity theory

- First order theories, weak subsystems of Peano Arithmetic.

$$\underbrace{\mathrm{PV}_1}_{\textbf{P ind.}} \subseteq \mathrm{PV}_1+ \underbrace{BB(\Sigma^b_0)}_{\text{sharply bounded choice}} \subseteq \underbrace{S^1_2}_{\textbf{NP ind.}} \subseteq \cdots \subseteq \underbrace{T_2}_{\textbf{PH ind.}}$$

- Already PV_1 and its mild extensions prove large number of results from complexity.
- Is PV_1 as strong as T_2?
  - Negative answer implies $\mathrm{PV}_1 \nvdash P = NP$, a major open problem.
- Mostly conditional results (using witnessing)
- Some unconditional unprovability of complexity statements is known:
  - $\mathrm{PV}_1 \nvdash$ "$P \subseteq \mathrm{SIZE}[n^k]$" [Krajíček–Oliveira 2017]

Bounded arithmetic and complexity theory

- First order theories, weak subsystems of Peano Arithmetic.

$$\underbrace{\mathrm{PV}_1}_{\textbf{P ind.}} \subseteq \mathrm{PV}_1+\underbrace{BB(\Sigma^b_0)}_{\text{sharply bounded choice}} \subseteq \underbrace{S^1_2}_{\textbf{NP ind.}} \subseteq \cdots \subseteq \underbrace{T_2}_{\textbf{PH ind.}}$$

- Already PV_1 and its mild extensions prove large number of results from complexity.
- Is PV_1 as strong as T_2?
  - Negative answer implies $\mathrm{PV}_1 \nvdash P = NP$, a major open problem.
- Mostly conditional results (using witnessing)
- Some unconditional unprovability of complexity statements is known:
  - $\mathrm{PV}_1 \nvdash$ "$P \subseteq \mathrm{SIZE}[n^k]$" [Krajíček–Oliveira 2017]
  - $\mathrm{PV}_1 \nvdash$ "Strong NP vs coNP-type separation." [Pich–Santhanam 2021]

Primes in bounded arithmetic

- Prime numbers are some of the most studied finite objects in mathematics. They are also important for cryptography (RSA), coding and other combinatorial constructions.

Primes in bounded arithmetic

- Prime numbers are some of the most studied finite objects in mathematics. They are also important for cryptography (RSA), coding and other combinatorial constructions.
- What can we prove about them in bounded arithmetic?

Primes in bounded arithmetic

- Prime numbers are some of the most studied finite objects in mathematics. They are also important for cryptography (RSA), coding and other combinatorial constructions.
- What can we prove about them in bounded arithmetic?
- [Woods 1981]: $I\Delta_0 + PHP \vdash$ "There are infinitely many primes."

Primes in bounded arithmetic

- Prime numbers are some of the most studied finite objects in mathematics. They are also important for cryptography (RSA), coding and other combinatorial constructions.
- What can we prove about them in bounded arithmetic?
- [Woods 1981]: $I\Delta_0 + PHP \vdash$ "There are infinitely many primes."
- [Paris, Wilkie, Woods 1988]: $T_2 \vdash$ "There are infinitely many primes."

Primes in bounded arithmetic

- Prime numbers are some of the most studied finite objects in mathematics. They are also important for cryptography (RSA), coding and other combinatorial constructions.
- What can we prove about them in bounded arithmetic?
- [Woods 1981]: $I\Delta_0 + PHP \vdash$ "There are infinitely many primes."
- [Paris, Wilkie, Woods 1988]: $T_2 \vdash$ "There are infinitely many primes."

Primes in bounded arithmetic

- Prime numbers are some of the most studied finite objects in mathematics. They are also important for cryptography (RSA), coding and other combinatorial constructions.
- What can we prove about them in bounded arithmetic?
- [Woods 1981]: $I\Delta_0 + PHP \vdash$ "There are infinitely many primes."
- [Paris, Wilkie, Woods 1988]: $T_2 \vdash$ "There are infinitely many primes."

Theorem (Jeřábek 2003)

$S_2^1 \vdash$ *"Every number x has a prime divisor p."*

Primes in bounded arithmetic

- Prime numbers are some of the most studied finite objects in mathematics. They are also important for cryptography (RSA), coding and other combinatorial constructions.
- What can we prove about them in bounded arithmetic?
- [Woods 1981]: $I\Delta_0 + PHP \vdash$ "There are infinitely many primes."
- [Paris, Wilkie, Woods 1988]: $T_2 \vdash$ "There are infinitely many primes."

Theorem (Jeřábek 2003)

$S_2^1 \vdash$ *"Every number x has a prime divisor p."*

Proof.

By the Σ_1^b-LENGTH-MIN principle, take the number p of minimal length satifying

$$p > 1 \wedge p \mid x.$$

By the minimality, the divisors of this number p can be only p or 1. □

The existence of prime factors as a first order sentence

"Every number $x \geq 2$ has a prime divisor p."

The existence of prime factors as a first order sentence

"Every number $x \geq 2$ has a prime divisor p."

$$(\forall x \geq 2)(\exists p \leq x)(p \mid x \wedge \text{“}p \text{ is a prime”})$$

The existence of prime factors as a first order sentence

"Every number $x \geq 2$ has a prime divisor p."

$$(\forall x \geq 2)(\exists p \leq x)(p \mid x \wedge \text{“}p \text{ is a prime”})$$

$$(\forall x \geq 2)(\exists p \leq x)(p \mid x \wedge p > 1 \wedge (\forall z \leq p)(z \mid p \rightarrow (z = 1 \vee z = p)))$$

The existence of prime factors as a first order sentence

"Every number $x \geq 2$ has a prime divisor p."

$$(\forall x \geq 2)(\exists p \leq x)(p \mid x \wedge \text{“}p \text{ is a prime”})$$

$$(\forall x \geq 2)(\exists p \leq x)(p \mid x \wedge p > 1 \wedge (\forall z \leq p)(z \mid p \rightarrow (z = 1 \vee z = p)))$$

- This is a $\forall\exists\forall$ sentence (or $\forall\Sigma_2^b$) as we are using a **coNP** definition of primes.

The existence of prime factors as a first order sentence

"Every number $x \geq 2$ has a prime divisor p."

$$(\forall x \geq 2)(\exists p \leq x)(p \mid x \wedge \text{“}p \text{ is a prime”})$$

$$(\forall x \geq 2)(\exists p \leq x)(p \mid x \wedge p > 1 \wedge (\forall z \leq p)(z \mid p \rightarrow (z = 1 \vee z = p)))$$

- This is a $\forall\exists\forall$ sentence (or $\forall\Sigma_2^b$) as we are using a **coNP** definition of primes.
- What about other definitions of prime numbers?

The existence of prime factors as a first order sentence

"Every number $x \geq 2$ has a prime divisor p."

$$(\forall x \geq 2)(\exists p \leq x)(p \mid x \wedge \text{“}p \text{ is a prime”})$$

$$(\forall x \geq 2)(\exists p \leq x)(p \mid x \wedge p > 1 \wedge (\forall z \leq p)(z \mid p \rightarrow (z = 1 \vee z = p)))$$

- This is a $\forall\exists\forall$ sentence (or $\forall\Sigma^b_2$) as we are using a **coNP** definition of primes.
- What about other definitions of prime numbers?
- If S^1_2 were to prove a correctness of some **NP** definition of primes, we would get a p-time factorization algorithm.

The existence of prime factors as a first order sentence

"Every number $x \geq 2$ has a prime divisor p."

$$(\forall x \geq 2)(\exists p \leq x)(p \mid x \wedge \text{“}p \text{ is a prime”})$$

$$(\forall x \geq 2)(\exists p \leq x)(p \mid x \wedge p > 1 \wedge (\forall z \leq p)(z \mid p \rightarrow (z = 1 \vee z = p)))$$

- This is a $\forall\exists\forall$ sentence (or $\forall\Sigma^b_2$) as we are using a **coNP** definition of primes.
- What about other definitions of prime numbers?
- If S^1_2 were to prove a correctness of some **NP** definition of primes, we would get a p-time factorization algorithm.
- Other definitions of primes, like AKS, seem to be available only in much stronger theories than S^1_2.

The existence of prime factors as a first order sentence

"Every number $x \geq 2$ has a prime divisor p."

$$(\forall x \geq 2)(\exists p \leq x)(p \mid x \wedge \text{“}p \text{ is a prime”})$$

$$(\forall x \geq 2)(\exists p \leq x)(p \mid x \wedge p > 1 \wedge (\forall z \leq p)(z \mid p \rightarrow (z = 1 \vee z = p)))$$

- This is a $\forall\exists\forall$ sentence (or $\forall\Sigma^b_2$) as we are using a **coNP** definition of primes.
- What about other definitions of prime numbers?
- If S^1_2 were to prove a correctness of some **NP** definition of primes, we would get a p-time factorization algorithm.
- Other definitions of primes, like AKS, seem to be available only in much stronger theories than S^1_2.
- We will call this $\forall\exists\forall$ sentence PrimeFactor.

Provability of the existence of prime factors in PV_1

Problem

$$PV_1 \vdash \textsc{PrimeFactor}?$$

- In words, does PV_1 prove that every number has a prime factor?

Provability of the existence of prime factors in PV_1

Problem

$$PV_1 \vdash \textsc{PrimeFactor}?$$

- In words, does PV_1 prove that every number has a prime factor?

Provability of the existence of prime factors in PV_1

Problem

$$PV_1 \vdash \textsc{PrimeFactor}?$$

- In words, does PV_1 prove that every number has a prime factor?

Theorem (Main)

Assume that no family of Boolean circuits $\{C_n\}_{n\in\mathbb{N}}$ of polynomial size can factorize a constant fraction of products of two n-bit primes. Then,

$$PV_1 + BB(\Sigma_0^b) \nvdash \textsc{PrimeFactor}.$$

Provability of the existence of prime factors in PV_1

Problem

$$PV_1 \vdash \textsc{PrimeFactor}?$$

- In words, does PV_1 prove that every number has a prime factor?

Theorem (Main)

Assume that no family of Boolean circuits $\{C_n\}_{n\in\mathbb{N}}$ of polynomial size can factorize a constant fraction of products of two n-bit primes. Then,

$$PV_1 + BB(\Sigma_0^b) \nvdash \textsc{PrimeFactor}.$$

- $\textsc{PrimeFactor}$ is a $\forall\exists\forall$ sentence $\rightarrow$ KPT witnessing.

Proof sketch (for PV_1)

- Assume $\mathrm{PV}_1 \vdash P \subseteq \textsc{PrimeFactor}$, then by the KPT theorem, there is a polynomial-time student s and a number $c \geq 1$.

Proof sketch (for PV_1)

- Assume $PV_1 \vdash P \subseteq \text{PrimeFactor}$, then by the KPT theorem, there is a polynomial-time student s and a number $c \geq 1$.
  - The student gets some number x and always outputs some divisor y_i of x, $y_i > 1$.

Proof sketch (for PV_1)

- Assume $PV_1 \vdash P \subseteq \text{PrimeFactor}$, then by the KPT theorem, there is a polynomial-time student s and a number $c \geq 1$.
  - The student gets some number x and always outputs some divisor y_i of x, $y_i > 1$.
  - A second function, t, a teacher, tries to correct the student when it fails to give a prime divisor, by giving a proper divisor of y_i denoted z_i.

Proof sketch (for PV_1)

- Assume $PV_1 \vdash P \subseteq \text{PrimeFactor}$, then by the KPT theorem, there is a polynomial-time student s and a number $c \geq 1$.
  - The student gets some number x and always outputs some divisor y_i of x, $y_i > 1$.
  - A second function, t, a teacher, tries to correct the student when it fails to give a prime divisor, by giving a proper divisor of y_i denoted z_i.
  - After at most $c - 1$ corrections of the teacher, the student is guaranteed to output a prime.

Proof sketch (for PV_1)

- Assume $PV_1 \vdash P \subseteq \text{PrimeFactor}$, then by the KPT theorem, there is a polynomial-time student s and a number $c \geq 1$.
  - The student gets some number x and always outputs some divisor y_i of x, $y_i > 1$.
  - A second function, t, a teacher, tries to correct the student when it fails to give a prime divisor, by giving a proper divisor of y_i denoted z_i.
  - After at most $c - 1$ corrections of the teacher, the student is guaranteed to output a prime.
- **Goal:** Simulate the teacher and trick the student to do some non-trivial factorization.

Proof sketch (for PV_1)

- Assume $PV_1 \vdash P \subseteq \text{PrimeFactor}$, then by the KPT theorem, there is a polynomial-time student s and a number $c \geq 1$.
  - The student gets some number x and always outputs some divisor y_i of x, $y_i > 1$.
  - A second function, t, a teacher, tries to correct the student when it fails to give a prime divisor, by giving a proper divisor of y_i denoted z_i.
  - After at most $c - 1$ corrections of the teacher, the student is guaranteed to output a prime.
- **Goal:** Simulate the teacher and trick the student to do some non-trivial factorization.
- **Naive attempt:** Give the student a product of c many primes, we hope that it can non-trivially factorize at least one pair.

$$x = p_1 p_2 p_3 p_4$$

Proof sketch (for PV_1)

- Assume $PV_1 \vdash P \subseteq \text{PrimeFactor}$, then by the KPT theorem, there is a polynomial-time student s and a number $c \geq 1$.
  - The student gets some number x and always outputs some divisor y_i of x, $y_i > 1$.
  - A second function, t, a teacher, tries to correct the student when it fails to give a prime divisor, by giving a proper divisor of y_i denoted z_i.
  - After at most $c - 1$ corrections of the teacher, the student is guaranteed to output a prime.
- **Goal:** Simulate the teacher and trick the student to do some non-trivial factorization.
- **Naive attempt:** Give the student a product of c many primes, we hope that it can non-trivially factorize at least one pair.

$$x = p_1 p_2 p_3 p_4$$

Proof sketch (for PV_1)

- Assume $PV_1 \vdash P \subseteq \text{PrimeFactor}$, then by the KPT theorem, there is a polynomial-time student s and a number $c \geq 1$.
  - The student gets some number x and always outputs some divisor y_i of x, $y_i > 1$.
  - A second function, t, a teacher, tries to correct the student when it fails to give a prime divisor, by giving a proper divisor of y_i denoted z_i.
  - After at most $c - 1$ corrections of the teacher, the student is guaranteed to output a prime.
- **Goal:** Simulate the teacher and trick the student to do some non-trivial factorization.
- **Naive attempt:** Give the student a product of c many primes, we hope that it can non-trivially factorize at least one pair.

$$x = p_1 p_2 p_3 p_4 \rightarrow s : y_1 = p_1 p_2 p_3 p_4$$

Proof sketch (for PV_1)

- Assume $PV_1 \vdash P \subseteq \text{PrimeFactor}$, then by the KPT theorem, there is a polynomial-time student s and a number $c \geq 1$.
  - The student gets some number x and always outputs some divisor y_i of x, $y_i > 1$.
  - A second function, t, a teacher, tries to correct the student when it fails to give a prime divisor, by giving a proper divisor of y_i denoted z_i.
  - After at most $c - 1$ corrections of the teacher, the student is guaranteed to output a prime.
- **Goal:** Simulate the teacher and trick the student to do some non-trivial factorization.
- **Naive attempt:** Give the student a product of c many primes, we hope that it can non-trivially factorize at least one pair.

$$x = p_1 p_2 p_3 p_4 \to s : y_1 = p_1 p_2 p_3 p_4 \to t : z_1 = p_1 p_2 p_3$$

Proof sketch (for PV_1)

- Assume $PV_1 \vdash P \subseteq \text{PrimeFactor}$, then by the KPT theorem, there is a polynomial-time student s and a number $c \geq 1$.
  - The student gets some number x and always outputs some divisor y_i of x, $y_i > 1$.
  - A second function, t, a teacher, tries to correct the student when it fails to give a prime divisor, by giving a proper divisor of y_i denoted z_i.
  - After at most $c - 1$ corrections of the teacher, the student is guaranteed to output a prime.
- **Goal:** Simulate the teacher and trick the student to do some non-trivial factorization.
- **Naive attempt:** Give the student a product of c many primes, we hope that it can non-trivially factorize at least one pair.

$$x = p_1p_2p_3p_4 \to s : y_1 = p_1p_2p_3p_4 \to t : z_1 = p_1p_2p_3 \to s : y_2 = p_1p_2p_3?$$

Proof sketch (for PV_1)

- Assume $PV_1 \vdash P \subseteq \text{PrimeFactor}$, then by the KPT theorem, there is a polynomial-time student s and a number $c \geq 1$.
  - The student gets some number x and always outputs some divisor y_i of x, $y_i > 1$.
  - A second function, t, a teacher, tries to correct the student when it fails to give a prime divisor, by giving a proper divisor of y_i denoted z_i.
  - After at most $c - 1$ corrections of the teacher, the student is guaranteed to output a prime.
- **Goal:** Simulate the teacher and trick the student to do some non-trivial factorization.
- **Naive attempt:** Give the student a product of c many primes, we hope that it can non-trivially factorize at least one pair.

$$x = p_1p_2p_3p_4 \to s : y_1 = p_1p_2p_3p_4 \to t : z_1 = p_1p_2p_3 \to s : y_2 = p_1p_2p_3?$$

But the student can easily reply: $y_2 = p_4$

Proof sketch II

- Both the teacher and the student can easily compute all divisions and gcd's.

Proof sketch II

- Both the teacher and the student can easily compute all divisions and gcd's.
- **Idea:** Give the student a product of 2^c many primes, and when it claims some part of its factorization is a prime divisor, get rid of half of its primes!

Proof sketch II

- Both the teacher and the student can easily compute all divisions and gcd's.
- **Idea:** Give the student a product of 2^c many primes, and when it claims some part of its factorization is a prime divisor, get rid of half of its primes!

Proof sketch II

- Both the teacher and the student can easily compute all divisions and gcd's.
- **Idea:** Give the student a product of 2^c many primes, and when it claims some part of its factorization is a prime divisor, get rid of half of its primes!

$$x = p_1 \dots p_8$$

Proof sketch II

- Both the teacher and the student can easily compute all divisions and gcd's.
- **Idea:** Give the student a product of 2^c many primes, and when it claims some part of its factorization is a prime divisor, get rid of half of its primes!

$$x = p_1 \dots p_8 \to s : y_1 = p_1 \dots p_8$$

Proof sketch II

- Both the teacher and the student can easily compute all divisions and gcd's.
- **Idea:** Give the student a product of 2^c many primes, and when it claims some part of its factorization is a prime divisor, get rid of half of its primes!

$$x = p_1 \dots p_8 \to s : y_1 = p_1 \dots p_8 \to t : z_1 = p_1 p_2 p_3 p_4 \to$$

Proof sketch II

- Both the teacher and the student can easily compute all divisions and gcd's.
- **Idea:** Give the student a product of 2^c many primes, and when it claims some part of its factorization is a prime divisor, get rid of half of its primes!

$$x = p_1 \dots p_8 \to s : y_1 = p_1 \dots p_8 \to t : z_1 = p_1 p_2 p_3 p_4 \to$$

The student 'has' to answer with $p_1 \dots p_4$ or $p_5 \dots p_8$!

Proof sketch II

- Both the teacher and the student can easily compute all divisions and gcd's.
- **Idea:** Give the student a product of 2^c many primes, and when it claims some part of its factorization is a prime divisor, get rid of half of its primes!

$$x = p_1 \dots p_8 \to s : y_1 = p_1 \dots p_8 \to t : z_1 = p_1 p_2 p_3 p_4 \to$$

The student 'has' to answer with $p_1 \dots p_4$ or $p_5 \dots p_8$!

- The numbers which do not result in non-trivial factorization are called obvious at a given round.

Proof sketch II

- Both the teacher and the student can easily compute all divisions and gcd's.
- **Idea:** Give the student a product of 2^c many primes, and when it claims some part of its factorization is a prime divisor, get rid of half of its primes!

$$x = p_1 \dots p_8 \to s : y_1 = p_1 \dots p_8 \to t : z_1 = p_1 p_2 p_3 p_4 \to$$

The student 'has' to answer with $p_1 \dots p_4$ or $p_5 \dots p_8$!

- The numbers which do not result in non-trivial factorization are called obvious at a given round.
- Their prime factorizations are closed under unions, intersections and complements, they form **a field of sets**.

Proof sketch III

- **By two lemmas about fields of sets:**

Proof sketch III

- **By two lemmas about fields of sets:**
  - By the general properties of fields of sets, if the student outputs a non-obvious answer y_i, there will be two primes p_i and p_j such that $\gcd(y_i, p_i p_j)$ is already either p_i or p_j.

Proof sketch III

- **By two lemmas about fields of sets:**
  - By the general properties of fields of sets, if the student outputs a non-obvious answer y_i, there will be two primes p_i and p_j such that $\gcd(y_i, p_i p_j)$ is already either p_i or p_j.
  - The number of prime factors of any obvious number decreases by a factor of 2 after every correction. We start with 2^c primes and there will be $c - 1$ corrections: The student has to answer non-obviously at some point!

Proof sketch III

- **By two lemmas about fields of sets:**
  - By the general properties of fields of sets, if the student outputs a non-obvious answer y_i, there will be two primes p_i and p_j such that $\gcd(y_i, p_i p_j)$ is already either p_i or p_j.
  - The number of prime factors of any obvious number decreases by a factor of 2 after every correction. We start with 2^c primes and there will be $c - 1$ corrections: The student has to answer non-obviously at some point!
- We get a uniform algorithm which (uniformly randomly) gets $(2^c - 2)$-primes and a product pq of two primes and factorizes pq with constant probability.

Proof sketch III

- **By two lemmas about fields of sets:**
  - By the general properties of fields of sets, if the student outputs a non-obvious answer y_i, there will be two primes p_i and p_j such that $\gcd(y_i, p_i p_j)$ is already either p_i or p_j.
  - The number of prime factors of any obvious number decreases by a factor of 2 after every correction. We start with 2^c primes and there will be $c - 1$ corrections: The student has to answer non-obviously at some point!
- We get a uniform algorithm which (uniformly randomly) gets $(2^c - 2)$-primes and a product pq of two primes and factorizes pq with constant probability.

Proof sketch III

- **By two lemmas about fields of sets:**
  - By the general properties of fields of sets, if the student outputs a non-obvious answer y_i, there will be two primes p_i and p_j such that $\gcd(y_i, p_i p_j)$ is already either p_i or p_j.
  - The number of prime factors of any obvious number decreases by a factor of 2 after every correction. We start with 2^c primes and there will be $c - 1$ corrections: The student has to answer non-obviously at some point!
- We get a uniform algorithm which (uniformly randomly) gets $(2^c - 2)$-primes and a product pq of two primes and factorizes pq with constant probability. (All of these primes, including p and q are independently uniformly randomly sampled from the same set.)

Proof sketch III

- **By two lemmas about fields of sets:**
  - By the general properties of fields of sets, if the student outputs a non-obvious answer y_i, there will be two primes p_i and p_j such that $\gcd(y_i, p_i p_j)$ is already either p_i or p_j.
  - The number of prime factors of any obvious number decreases by a factor of 2 after every correction. We start with 2^c primes and there will be $c - 1$ corrections: The student has to answer non-obviously at some point!
- We get a uniform algorithm which (uniformly randomly) gets $(2^c - 2)$-primes and a product pq of two primes and factorizes pq with constant probability. (All of these primes, including p and q are independently uniformly randomly sampled from the same set.)
- We conclude by using the averaging argument to convert this probabilistic algorithm into a sequence of circuits. □

Plausibility of the assumption

- The assumption that no non-uniform polynomial time algorithm should be able to factorize a constant fraction of products of n-bit primes seems widely accepted in cryptography.

Plausibility of the assumption

- The assumption that no non-uniform polynomial time algorithm should be able to factorize a constant fraction of products of n-bit primes seems widely accepted in cryptography.
- The Main Theorem works even if we restrict the set of n-bit primes to some subset of 'harder to factorize primes', such as safe primes and strong primes.

Plausibility of the assumption

- The assumption that no non-uniform polynomial time algorithm should be able to factorize a constant fraction of products of n-bit primes seems widely accepted in cryptography.
- The Main Theorem works even if we restrict the set of n-bit primes to some subset of 'harder to factorize primes', such as safe primes and strong primes.
- This can be interpreted as follows: If

$$\mathrm{PV}_1 + BB(\Sigma_0^b) \vdash \textsc{PrimeFactor},$$

then a constant fraction of all RSA moduli are not safe.

Algebraic interpretation of the Main Theorem

- By the completeness theorem, the assumption of the Main Theorem implies the existence of a model $M \models \mathrm{PV}_1$ (possibly extended by $BB(\Sigma_0^b)$) such that there is a number $m \in M$ whose every proper divisor has a proper divisor:

$$\cdots \mid m_4 \mid m_3 \mid m_2 \mid m_1 \mid m \in M.$$

Algebraic interpretation of the Main Theorem

- By the completeness theorem, the assumption of the Main Theorem implies the existence of a model $M \models \mathrm{PV}_1$ (possibly extended by $BB(\Sigma_0^b)$) such that there is a number $m \in M$ whose every proper divisor has a proper divisor:

$$\cdots \mid m_4 \mid m_3 \mid m_2 \mid m_1 \mid m \in M.$$

Algebraic interpretation of the Main Theorem

- By the completeness theorem, the assumption of the Main Theorem implies the existence of a model $M \models \mathrm{PV}_1$ (possibly extended by $BB(\Sigma_0^b)$) such that there is a number $m \in M$ whose every proper divisor has a proper divisor:

$$\cdots \mid m_4 \mid m_3 \mid m_2 \mid m_1 \mid m \in M.$$

Therefore, m does not have a prime factorization in M,

Algebraic interpretation of the Main Theorem

- By the completeness theorem, the assumption of the Main Theorem implies the existence of a model $M \models \mathrm{PV}_1$ (possibly extended by $BB(\Sigma_0^b)$) such that there is a number $m \in M$ whose every proper divisor has a proper divisor:

$$\cdots \mid m_4 \mid m_3 \mid m_2 \mid m_1 \mid m \in M.$$

  Therefore, m does not have a prime factorization in M,as none of its divisors is a prime!
- Provably, this cannot happen in S_2^1, this conditional result gives us an algebraic property we can see in some models of PV_1 and but in no models of S_2^1.

Algebraic interpretation of the Main Theorem

- By the completeness theorem, the assumption of the Main Theorem implies the existence of a model $M \models \mathrm{PV}_1$ (possibly extended by $BB(\Sigma_0^b)$) such that there is a number $m \in M$ whose every proper divisor has a proper divisor:

$$\cdots \mid m_4 \mid m_3 \mid m_2 \mid m_1 \mid m \in M.$$

  Therefore, m does not have a prime factorization in M,as none of its divisors is a prime!
- Provably, this cannot happen in S_2^1, this conditional result gives us an algebraic property we can see in some models of PV_1 and but in no models of S_2^1.
- In algebra, the integral domains with the property that every non-invertible element has some irreducible factor are called Furstenberg domains.

Algebraic interpretation of the Main Theorem

- By the completeness theorem, the assumption of the Main Theorem implies the existence of a model $M \models \mathrm{PV}_1$ (possibly extended by $BB(\Sigma_0^b)$) such that there is a number $m \in M$ whose every proper divisor has a proper divisor:

$$\cdots \mid m_4 \mid m_3 \mid m_2 \mid m_1 \mid m \in M.$$

  Therefore, m does not have a prime factorization in M,as none of its divisors is a prime!
- Provably, this cannot happen in S_2^1, this conditional result gives us an algebraic property we can see in some models of PV_1 and but in no models of S_2^1.
- In algebra, the integral domains with the property that every non-invertible element has some irreducible factor are called Furstenberg domains.
- The Main Theorem can be interpreted as: Under our assumptions, there are models of PV_1 which are not positive parts of Furstenberg domains.

Related work

Theorem (KPT 1991)

Assume that $\mathbf{NP} \not\subseteq \mathbf{P}/poly$. *Then* $PV_1 \neq S^1_2$.

Related work

Theorem (KPT 1991)

Assume that $\mathbf{NP} \not\subseteq \mathbf{P}/poly$. *Then* $PV_1 \neq S^1_2$.

Theorem (Cook, Thapen 2006)

If $PV_1 \vdash BB(\Sigma^b_0)$, *then there is a probabilistic polynomial time algorithm factoring every product of odd primes.*

Related work

Theorem (KPT 1991)

Assume that $\mathbf{NP} \not\subseteq \mathbf{P}/poly$. *Then* $PV_1 \neq S^1_2$.

Theorem (Cook, Thapen 2006)

If $PV_1 \vdash BB(\Sigma^b_0)$, *then there is a probabilistic polynomial time algorithm factoring every product of odd primes.*

- Thus, under the assumption of the Main Theorem we get:

  PV_1

Related work

Theorem (KPT 1991)

Assume that $\mathbf{NP} \not\subseteq \mathbf{P}/poly$. *Then* $PV_1 \neq S^1_2$.

Theorem (Cook, Thapen 2006)

If $PV_1 \vdash BB(\Sigma^b_0)$, *then there is a probabilistic polynomial time algorithm factoring every product of odd primes.*

- Thus, under the assumption of the Main Theorem we get:

  PV_1

Related work

Theorem (KPT 1991)

Assume that $\mathbf{NP} \not\subseteq \mathbf{P}/poly$. *Then* $PV_1 \neq S^1_2$.

Theorem (Cook, Thapen 2006)

If $PV_1 \vdash BB(\Sigma^b_0)$, *then there is a probabilistic polynomial time algorithm factoring every product of odd primes.*

- Thus, under the assumption of the Main Theorem we get:

$$\mathrm{PV}_1 <_{[CT06]} \mathrm{PV}_1 + BB(\Sigma^b_0)$$

Related work

Theorem (KPT 1991)

Assume that $\mathbf{NP} \not\subseteq \mathbf{P}/poly$. *Then* $PV_1 \neq S^1_2$.

Theorem (Cook, Thapen 2006)

If $PV_1 \vdash BB(\Sigma^b_0)$, *then there is a probabilistic polynomial time algorithm factoring every product of odd primes.*

- Thus, under the assumption of the Main Theorem we get:

$$\mathrm{PV}_1 <_{[CT06]} \mathrm{PV}_1 + BB(\Sigma^b_0) <_{\text{Main Theorem}} S^1_2$$

Related work

Theorem (KPT 1991)

Assume that $\mathbf{NP} \not\subseteq \mathbf{P}/poly$. *Then* $PV_1 \neq S^1_2$.

Theorem (Cook, Thapen 2006)

If $PV_1 \vdash BB(\Sigma^b_0)$, *then there is a probabilistic polynomial time algorithm factoring every product of odd primes.*

- Thus, under the assumption of the Main Theorem we get:

$$\mathrm{PV}_1 <_{[CT06]} \mathrm{PV}_1 + BB(\Sigma^b_0) <_{\text{Main Theorem}} S^1_2 \leq T^1_2$$

Related work

Theorem (KPT 1991)

Assume that $\mathbf{NP} \not\subseteq \mathbf{P}/poly$. *Then* $PV_1 \neq S^1_2$.

Theorem (Cook, Thapen 2006)

If $PV_1 \vdash BB(\Sigma^b_0)$, *then there is a probabilistic polynomial time algorithm factoring every product of odd primes.*

- Thus, under the assumption of the Main Theorem we get:

$$\mathrm{PV}_1 <_{[CT06]} \mathrm{PV}_1 + BB(\Sigma^b_0) <_{\text{Main Theorem}} S^1_2 \leq T^1_2$$

Problem

Does the same assumption also separate S^1_2 *from* T^1_2*? Can we find similar assumptions which separate many theories in a row in Buss' hierarchy?*

$\exists M \models \mathrm{PV}_1 : M \models$ ‘Thank you for your attention.’