Consistency of circuit lower bounds with bounded theories

Igor Carboni Oliveira

Department of Computer Science, University of Warwick.

Talk based on joint works with **Jan Bydžovský** (Vienna) and **Jan Krajíček** (Prague).

Theoretical Computer Science Seminar – University of Birmingham

This work was supported in part by a Royal Society University Research Fellowship.

Computational Complexity Theory

- Investigates limits and possibilities of algorithms and computations.

P vs BPP: Are *randomised* algorithms significantly faster than *deterministic* algorithms?

P vs NP: Is *finding* a solution harder than *verifying* a given solution?

Computational Complexity Theory

- Investigates limits and possibilities of algorithms and computations.

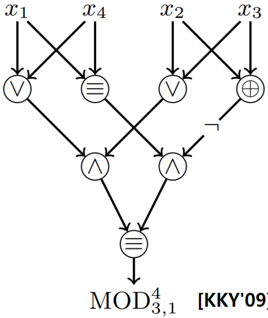
P vs BPP: Are *randomised* algorithms significantly faster than *deterministic* algorithms?

P vs NP: Is *finding* a solution harder than *verifying* a given solution?

- **Uniform computations**: single algorithm that works on all input lengths.

Boolean circuits and non-uniform computations

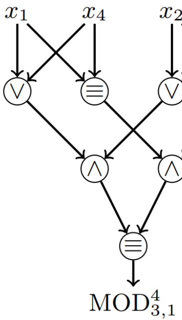
- A simple combinatorial model that captures computations:



[KKY'09]

Boolean circuits and non-uniform computations

- A simple combinatorial model that captures computations:



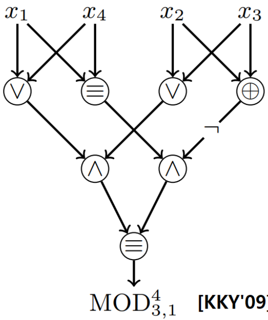
[KKY'09]

- **Non-uniform computations**:

Sequence $\{C_n\}_n$ of circuits, where C_n solves the problem on n-bit input instances.

Boolean circuits and non-uniform computations

- A simple combinatorial model that captures computations:



[KKY'09]

- **Non-uniform computations**:
Sequence $\{C_n\}_n$ of circuits, where C_n solves the problem on n-bit input instances.

- Algorithm running in time $T(n) \implies$ circuits C_n with $O(T(n) \cdot \log T(n))$ gates.

Circuit Complexity Theory

► Interested in circuit size (number of gates) required to compute $f\colon \{0,1\}^n \to \{0,1\}^m$.

[Shannon'49] Most functions $f\colon \{0,1\}^n \to \{0,1\}$ require circuits of size $\Omega(2^n/n)$.

► In connection to algorithms and complexity, we would like to understand the circuit size of "explicit" functions in P, NP, etc.

Research on restricted classes of circuits

► Much progress over the last 40 years in understanding limited classes of circuits, such as **small-depth** circuits with AND/OR/NOT gates.

– **Addition** of two n-bit numbers is *provably easier* than **Multiplication**.

– **DIST$_k$-CONNECTIVITY**(n) requires depth-d circuits of size $n^{k^{\Theta(1/d)}}$.

– The constant-depth circuit complexity of **k-CLIQUE** is precisely $n^{\Theta(k)}$.

Research on restricted classes of circuits

▶ Much progress over the last 40 years in understanding limited classes of circuits, such as **small-depth** circuits with AND/OR/NOT gates.

– **Addition** of two n-bit numbers is *provably easier* than **Multiplication**.

– **DIST$_k$-CONNECTIVITY**(n) requires depth-d circuits of size $n^{k^{\Theta(1/d)}}$.

– The constant-depth circuit complexity of k-**CLIQUE** is precisely $n^{\Theta(k)}$.

▶ However, many important algorithms produce circuits of **unbounded** depth.

Status of circuit lower bounds

- In this talk we will focus on **unrestricted** Boolean circuits.

- Best result for a problem in NP is a lower bound of $(3 + 1/86) \cdot n$ gates **[FGHK'16]**.

Status of circuit lower bounds

- In this talk we will focus on **unrestricted** Boolean circuits.
- Best result for a problem in NP is a lower bound of $(3 + 1/86) \cdot n$ gates **[FGHK'16]**.
- Proving a lower bound such as NP $\not\subseteq$ SIZE$[n^2]$ seems out of reach.
- Motivates the study of circuit lower bounds for classes believed to be larger than NP.

Frontiers

$\mathsf{ZPP}^{\mathsf{NP}} \not\subseteq \mathsf{SIZE}[n^k]$ [Kobler-Watanabe'90s]

$\mathsf{MA}/1 \not\subseteq \mathsf{SIZE}[n^k]$ [Santhanam'00s]

Frontiers

$\mathsf{ZPP}^{\mathsf{NP}} \not\subseteq \mathsf{SIZE}[n^k]$ [Kobler-Watanabe'90s]

$\mathsf{MA}/1 \not\subseteq \mathsf{SIZE}[n^k]$ [Santhanam'00s]

- **Frontier 1**: Lower bounds for **deterministic** class P^{NP}?

Frontiers

$\mathrm{ZPP}^{\mathrm{NP}} \not\subseteq \mathrm{SIZE}[n^k]$ [Kobler-Watanabe'90s]

$\mathrm{MA}/1 \not\subseteq \mathrm{SIZE}[n^k]$ [Santhanam'00s]

- **Frontier 1**: Lower bounds for **deterministic** class P^{NP}?

While we have lower bounds for larger classes, **there is an important issue**:

- **Frontier 2**: All results of the form $\omega(n)$ only hold on **infinitely many input lengths.**

a.e. versus i.o. results in algorithms and complexity

- **Mystery:** Existence of mathematical objects of certain sizes making computations easier only around corresponding input lengths.

a.e. versus i.o. results in algorithms and complexity

- **Mystery:** Existence of mathematical objects of certain sizes making computations easier only around corresponding input lengths.

- Issue **not** restricted to complexity lower bounds:

Sub-exponential time generation of canonical prime numbers [Oliveira-Santhamam'17].

The logical approach

- We discussed two frontiers in complexity theory:

1. Understand relation between P^{NP} and say $\mathsf{SIZE}[n^2]$.
2. Establish almost-everywhere circuit lower bounds.

- This work investigates these challenges from the **perspective of mathematical logic**.

Investigating complexity through logic

- Theories in the standard framework of first-order logic.

- Investigation of complexity results that can be established under certain axioms.

Example: Does theory T prove that SAT can be solved in polynomial time?

- **Complexity Theory** that considers **efficiency** and **difficulty of proving correctness**.

Bounded Arithmetics

- Fragments of Peano Arithmetic (PA).
- Intended model is $\mathbb{N}$, but numbers can encode binary strings and other objects.

Bounded Arithmetics

- Fragments of Peano Arithmetic (PA).
- Intended model is $\mathbb{N}$, but numbers can encode binary strings and other objects.

Example: Theory $I\Delta_0$ [Parikh'71].

$I\Delta_0$ employs the language $\mathcal{L}_{PA} = \{0, 1, +, \cdot, <\}$.

14 axioms governing these symbols, such as:

1. $\forall x \;\; x + 0 = x$
2. $\forall x \forall y \;\; x + y = y + x$
3. $\forall x \;\; x = 0 \lor 0 < x$

$\dots$

Bounded formulas and bounded induction

Induction Axioms. $I\Delta_0$ also contains the induction principle

$$\psi(0) \wedge \forall x\,(\psi(x) \to \psi(x+1)) \to \forall x\,\psi(x)$$

for each **bounded formula** $\psi(x)$ (additional free variables are allowed in ψ).

Bounded formulas and bounded induction

Induction Axioms. $I\Delta_0$ also contains the induction principle

$$\psi(0) \land \forall x\,(\psi(x) \to \psi(x+1)) \to \forall x\,\psi(x)$$

for each **bounded formula** $\psi(x)$ (additional free variables are allowed in ψ).

A **bounded formula** only contains quantifiers of the form $\forall y \le t$ and $\exists y \le t$, where t is a term not containing y. Abbreviations for $\forall y\,(y \le t \to \ldots)$ and $\exists y\,(y \le t \land \ldots)$.

Bounded formulas and bounded induction

Induction Axioms. $I\Delta_0$ also contains the induction principle

$$\psi(0) \wedge \forall x\, (\psi(x) \rightarrow \psi(x+1)) \rightarrow \forall x\, \psi(x)$$

for each **bounded formula** $\psi(x)$ (additional free variables are allowed in ψ).

A **bounded formula** only contains quantifiers of the form $\forall y \leq t$ and $\exists y \leq t$, where t is a term not containing y. Abbreviations for $\forall y\, (y \leq t \rightarrow \ldots)$ and $\exists y\, (y \leq t \wedge \ldots)$.

- **This shifts the perspective from computability to complexity theory.**

Theories PV, S_2^1, and T_2^1

- [Cook'75] and [Buss'86] introduced theories more closely related to levels of PH:

Ex.: T_2^1 uses induction scheme for bounded formulas corresponding to NP-predicates.

Theories PV, S_2^1, and T_2^1

- [Cook'75] and [Buss'86] introduced theories more closely related to levels of PH:

Ex.: T_2^1 uses induction scheme for bounded formulas corresponding to NP-predicates.

- We will use language $\mathcal{L}_{PV}$ with function symbols for all p-time algorithms.

Theories PV, S_2^1, and T_2^1

- [Cook'75] and [Buss'86] introduced theories more closely related to levels of PH:

Ex.: T_2^1 uses induction scheme for bounded formulas corresponding to NP-predicates.

- We will use language $\mathcal{L}_{PV}$ with function symbols for all p-time algorithms.

This does not mean that the corresponding theories prove **correctness** of algorithms:
$T_2^1 \vdash \forall x \;\; f_{AKS}(x) = 1 \leftrightarrow$ "x is prime" ?

Theories PV, S_2^1, and T_2^1

- [Cook'75] and [Buss'86] introduced theories more closely related to levels of PH:

Ex.: T_2^1 uses induction scheme for bounded formulas corresponding to NP-predicates.

- We will use language $\mathcal{L}_{PV}$ with function symbols for all p-time algorithms.

This does not mean that the corresponding theories prove **correctness** of algorithms:
$T_2^1 \vdash \forall x \;\; f_{AKS}(x) = 1 \leftrightarrow$ "x is prime" ?

$$\text{PV} \approx \text{T}_2^0 \quad \subseteq \quad S_2^1 \quad \subseteq \quad T_2^1 \quad \subseteq \quad S_2^2 \quad \subseteq \quad T_2^2 \quad \subseteq \quad \ldots \subseteq \quad \bigcup_i T_2^i \approx I\Delta_0 + \Omega_1$$

More about correctness of algorithms and bounded arithmetic

- PV and S_2^1 can formalize several interesting algorithms (e.g. "Hungarian Method" for Bipartite Matching).

More about correctness of algorithms and bounded arithmetic

- PV and S^1_2 can formalize several interesting algorithms (e.g. "Hungarian Method" for Bipartite Matching).

- Suppose $S^1_2 \vdash$ **"Primality is in P"**, i.e., for some $\mathcal{L}_{\mathsf{PV}}$ function symbol g,

$S^1_2 \vdash \forall x\,(g(x) = 0 \leftrightarrow \exists y\,(1 < y < x \wedge y \,|\, x))$.
$S^1_2 \vdash \forall x\,(g(x) = 0 \rightarrow \exists y\,(1 < y < x \wedge y \,|\, x))$.
$S^1_2 \vdash \forall x\,(\neg g(x) = 0 \vee \exists y\,(1 < y < x \wedge y \,|\, x))$.
$S^1_2 \vdash \forall x\,\exists y\,(\neg g(x) = 0 \vee (1 < y < x \wedge y \,|\, x))$.

More about correctness of algorithms and bounded arithmetic

► PV and S^1_2 can formalize several interesting algorithms (e.g. “Hungarian Method” for Bipartite Matching).

► Suppose $S^1_2 \vdash$ **“Primality is in P”**, i.e., for some $\mathcal{L}_{\mathsf{PV}}$ function symbol g,

$$S^1_2 \vdash \forall x\,(g(x) = 0 \leftrightarrow \exists y\,(1 < y < x \wedge y \,|\, x)).$$
$$S^1_2 \vdash \forall x\,(g(x) = 0 \rightarrow \exists y\,(1 < y < x \wedge y \,|\, x)).$$
$$S^1_2 \vdash \forall x\,(\neg g(x) = 0 \vee \exists y\,(1 < y < x \wedge y \,|\, x)).$$
$$S^1_2 \vdash \forall x\,\exists y\,(\neg g(x) = 0 \vee (1 < y < x \wedge y \,|\, x)).$$

Now if $S^1_2 \vdash \forall x\,\exists y\,\varphi(x, y)$ for an open $\mathcal{L}_{\mathsf{PV}}$-formula φ, then by **Buss’ Witnessing Theorem**, $S^1_2 \vdash \forall x\,\varphi(x, h(x))$ for some $\mathcal{L}_{\mathsf{PV}}$ function symbol h.

More about correctness of algorithms and bounded arithmetic

▶ PV and S^1_2 can formalize several interesting algorithms (e.g. "Hungarian Method" for Bipartite Matching).

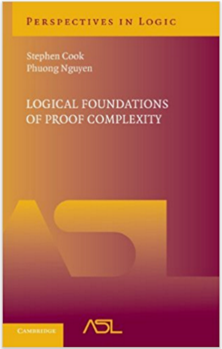
▶ Suppose $S^1_2 \vdash$ **"Primality is in P"**, i.e., for some $\mathcal{L}_{\mathsf{PV}}$ function symbol g,

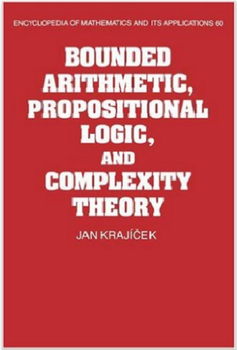
$S^1_2 \vdash \forall x\, (g(x) = 0 \leftrightarrow \exists y\, (1 < y < x \wedge y \,|\, x))$.
$S^1_2 \vdash \forall x\, (g(x) = 0 \rightarrow \exists y\, (1 < y < x \wedge y \,|\, x))$.
$S^1_2 \vdash \forall x\, (\neg g(x) = 0 \vee \exists y\, (1 < y < x \wedge y \,|\, x))$.
$S^1_2 \vdash \forall x\, \exists y\, (\neg g(x) = 0 \vee (1 < y < x \wedge y \,|\, x))$.

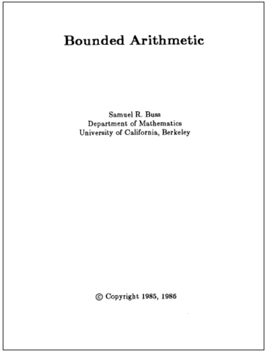
Now if $S^1_2 \vdash \forall x\, \exists y\, \varphi(x, y)$ for an open $\mathcal{L}_{\mathsf{PV}}$-formula φ, then by **Buss' Witnessing Theorem**, $S^1_2 \vdash \forall x\, \varphi(x, h(x))$ for some $\mathcal{L}_{\mathsf{PV}}$ function symbol h.

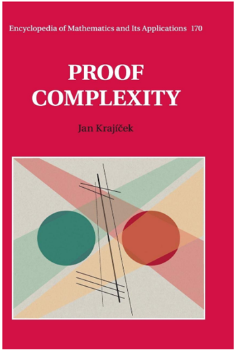
▶ This places **Integer-Factoring in P**, which contradicts cryptographic assumptions.

Resources I


Perspectives in Logic
Stephen Cook
Phuong Nguyen
LOGICAL FOUNDATIONS OF PROOF COMPLEXITY
CAMBRIDGE
ASL


ENCYCLOPEDIA OF MATHEMATICS AND ITS APPLICATIONS 60
BOUNDED ARITHMETIC, PROPOSITIONAL LOGIC, AND COMPLEXITY THEORY
JAN KRAJÍČEK


Bounded Arithmetic
Samuel R. Buss
Department of Mathematics
University of California, Berkeley
© Copyright 1985, 1986


Encyclopedia of Mathematics and Its Applications 170
PROOF COMPLEXITY
Jan Krajíček

Resources II

- **Some PhD Theses:**

Kerry Ojakian (CMU, 2004). *Combinatorics in Bounded Arithmetic.*

Emil Jerabek (Prague, 2005). *Weak Pigeonhole Principle and Randomized Computation.*

Dai Tri Man Le (Toronto, 2014). *Bounded Arithmetic and Formalizing Probabilistic Proofs.*

Jan Pich (Prague, 2014). *Complexity Theory in Feasible Mathematics.*

- **A recent work with pointers to several relevant references:**

Moritz Muller and Jan Pich (2019). *Feasibly constructive proofs of succinct weak circuit lower bounds.*

Formalizations in Bounded Arithmetic

- Many complexity results have been formalized in such theories.

Cook-Levin Theorem in PV [folklore].

PCP Theorem in PV [Pich'15].

Parity $\notin$ AC^0, k-Clique $\notin$ mSIZE$[n^{\sqrt{k}/1000}]$ in APC$^1 \subseteq T^2_2$ [Muller-Pich'19].

Formalizations in Bounded Arithmetic

- Many complexity results have been formalized in such theories.

Cook-Levin Theorem in PV [folklore].

PCP Theorem in PV [Pich'15].

Parity $\notin$ AC^0, k-Clique $\notin$ $\mathsf{mSIZE}[n^{\sqrt{k}/1000}]$ in $\mathsf{APC}^1 \subseteq T^2_2$ [Muller-Pich'19].

- Arguments often require ingenious modifications of original proofs:
  not clear how to manipulate probability spaces, real-valued functions, etc.

Formalizations in Bounded Arithmetic

- Many complexity results have been formalized in such theories.

Cook-Levin Theorem in PV [folklore].

PCP Theorem in PV [Pich'15].

Parity $\notin$ AC^0, k-Clique $\notin$ mSIZE$[n^{\sqrt{k}/1000}]$ in APC$^1 \subseteq T_2^2$ [Muller-Pich'19].

- Arguments often require ingenious modifications of original proofs:
  not clear how to manipulate probability spaces, real-valued functions, etc.

Rest of the talk: **Independence of complexity results** from bounded arithmetic.

Unprovability and circuit complexity

- Using $\mathcal{L}_{PV}$, we can refer to circuit complexity:

$$\exists y\ (\mathsf{Ckt}(y) \wedge \mathsf{Vars}(y) = n \wedge \mathsf{Size}(y) \leq 5n \wedge \forall x\ (|x| = n \rightarrow (\mathsf{Eval}(y, x) = 1 \leftrightarrow \mathsf{Parity}(x) = 1)))$$

n is the “feasibility” parameter (formally, the length of another variable N).

Unprovability and circuit complexity

- Using $\mathcal{L}_{PV}$, we can refer to circuit complexity:

$$\exists y\ (\mathsf{Ckt}(y) \wedge \mathsf{Vars}(y) = n \wedge \mathsf{Size}(y) \leq 5n \wedge \forall x\ (|x| = n \rightarrow (\mathsf{Eval}(y, x) = 1 \leftrightarrow \mathsf{Parity}(x) = 1)))$$

n is the "feasibility" parameter (formally, the length of another variable N).

- Sentences can express circuit size bounds of the form n^k for a given $\mathcal{L}_{PV}$-formula $\varphi(x)$.

Unprovability and circuit complexity

- Using $\mathcal{L}_{PV}$, we can refer to circuit complexity:

$$\exists y\ (\mathsf{Ckt}(y) \wedge \mathsf{Vars}(y) = n \wedge \mathsf{Size}(y) \leq 5n \wedge \forall x\ (|x| = n \rightarrow (\mathsf{Eval}(y,x) = 1 \leftrightarrow \mathsf{Parity}(x) = 1)))$$

n is the "feasibility" parameter (formally, the length of another variable N).

- Sentences can express circuit size bounds of the form n^k for a given $\mathcal{L}_{PV}$-formula $\varphi(x)$.

Two directions: unprovability of **LOWER** bounds and unprovability of **UPPER** bounds.

Unprovability of circuit **LOWER** bounds

- Initiated by Razborov in the nineties under a different formalization.

Motivation: Why are complexity lower bounds so difficult to prove?

Also: potential source of hard tautologies; self-referential arguments and implications.

Unprovability of circuit **LOWER** bounds

- Initiated by Razborov in the nineties under a different formalization.

Motivation: Why are complexity lower bounds so difficult to prove?

Also: potential source of hard tautologies; self-referential arguments and implications.

Example: Is it the case that $T^2_2 \nvdash k$-Clique $\notin$ SIZE$[n^{\sqrt{k}/100}]$?

Unprovability of circuit **LOWER** bounds

- Initiated by Razborov in the nineties under a different formalization.

Motivation: Why are complexity lower bounds so difficult to prove?

Also: potential source of hard tautologies; self-referential arguments and implications.

Example: Is it the case that $T^2_2 \nvdash k$-Clique $\notin$ SIZE$[n^{\sqrt{k}/100}]$?

- Extremely interesting, but not much is known in terms of **unconditional** unprovability results for strong theories such as PV.

Unprovability of circuit **UPPER** bounds

- We currently cannot rule out that SAT $\in$ SIZE$[10n]$. Can we at least show that easiness of SAT doesn't follow from certain axioms?

At least as interesting as previous direction:

Unprovability of circuit **UPPER** bounds

► We currently cannot rule out that SAT $\in$ SIZE$[10n]$. Can we at least show that easiness of SAT doesn't follow from certain axioms?

At least as interesting as previous direction:

1. **Necessary** before proving in the standard sense that SAT $\notin$ SIZE$[10n]$. Rules out algorithmic approaches in a principled way.

Unprovability of circuit **UPPER** bounds

► We currently cannot rule out that SAT $\in$ SIZE$[10n]$. Can we at least show that easiness of SAT doesn't follow from certain axioms?

At least as interesting as previous direction:

1. **Necessary** before proving in the standard sense that SAT $\notin$ SIZE$[10n]$. Rules out algorithmic approaches in a principled way.

2. **Formal evidence** that SAT is computationally hard:

– By Godel's completeness theorem, there is a model M of T where SAT is hard.
– M satisfies many known results in algorithms and complexity theory.

Unprovability of circuit **UPPER** bounds

► We currently cannot rule out that SAT $\in$ SIZE$[10n]$. Can we at least show that easiness of SAT doesn't follow from certain axioms?

At least as interesting as previous direction:

1. **Necessary** before proving in the standard sense that SAT $\notin$ SIZE$[10n]$. Rules out algorithmic approaches in a principled way.

2. **Formal evidence** that SAT is computationally hard:

   – By Godel's completeness theorem, there is a model M of T where SAT is hard.
   – M satisfies many known results in algorithms and complexity theory.

3. **Consistency of lower bounds:** Adding to T axiom stating that SAT is hard will never lead to a contradiction. We can develop a theory where circuit lower bounds exist.

Some works on unprovability of circuit upper bounds

- Cook-Krajicek, 2007: "*Consequences of the provability of* $\mathsf{NP} \subseteq \mathsf{P}/\mathsf{poly}$".

Initiated a systematic investigation. Conditional unprovability results.

Some works on unprovability of circuit upper bounds

- Cook-Krajicek, 2007: "*Consequences of the provability of* NP $\subseteq$ P/poly".

Initiated a systematic investigation. Conditional unprovability results.

- Krajicek-Oliveira, 2017: "*Unprovability of circuit upper bounds in Cook's theory PV*".

Established unconditionally that PV does not prove that P $\subseteq$ SIZE$[n^k]$.

Some works on unprovability of circuit upper bounds

- Cook-Krajicek, 2007: "*Consequences of the provability of* NP $\subseteq$ P$/$poly".

Initiated a systematic investigation. Conditional unprovability results.

- Krajicek-Oliveira, 2017: "*Unprovability of circuit upper bounds in Cook's theory PV*".

Established unconditionally that PV does not prove that P $\subseteq$ SIZE$[n^k]$.

- Bydzovsky-Muller, 2018: "*Polynomial time ultrapowers and the consistency of circuit lower bounds.*".

Model-theoretic proof of a slightly stronger statement.

Weaknesses of previous results

1. We would like to show unprovability results for theories believed to be stronger than PV.

Weaknesses of previous results

1. We would like to show unprovability results for theories believed to be stronger than PV.

2. Infinitely often versus almost everywhere results:

PV might still show that every $L \in \mathsf{P}$ is infinitely often in SIZE$[n^k]$.

Weaknesses of previous results

1. We would like to show unprovability results for theories believed to be stronger than PV.

2. Infinitely often versus almost everywhere results:

PV might still show that every $L \in \mathsf{P}$ is infinitely often in SIZE$[n^k]$.

- Recall issue mentioned earlier in the talk:

We lack techniques to show hardness with respect to every large enough input length.

This work

- T_2^1 and weaker theories cannot establish circuit upper bounds of the form n^k for classes contained in P^{NP}.

- Unprovability of infinitely often upper bounds, i.e., models where hardness holds almost everywhere.

- All results are **unconditional**.

Our main result

Theorem 1 (Informal): For each $k \geq 1$,

$$T^1_2 \quad \nvdash \quad \mathsf{P}^{\mathsf{NP}} \subseteq \text{i.o.SIZE}[n^k]$$

$$S^1_2 \quad \nvdash \quad \mathsf{NP} \subseteq \text{i.o.SIZE}[n^k]$$

$$\mathsf{PV} \quad \nvdash \quad \mathsf{P} \subseteq \text{i.o.SIZE}[n^k]$$

Our main result

Theorem 1 (Informal): For each $k \geq 1$,

$$T^1_2 \quad \nvdash \quad \mathsf{P}^{\mathsf{NP}} \subseteq \text{i.o.SIZE}[n^k]$$

$$S^1_2 \quad \nvdash \quad \mathsf{NP} \subseteq \text{i.o.SIZE}[n^k]$$

$$\mathsf{PV} \quad \nvdash \quad \mathsf{P} \subseteq \text{i.o.SIZE}[n^k]$$

Extensions. $\text{True}_1 \stackrel{\text{def}}{=} \forall\Sigma^b_1(\mathcal{L}_{PV})$-sentences true in $\mathbb{N}$ can be included in first item.

Example: $\forall x\, (\exists y\, (1 < y < x \wedge y \mid x) \leftrightarrow f_{\mathsf{AKS}}(x) = 0)$

$T^1_2 \cup \text{True}_1$ proves that Primes $\in \text{SIZE}[n^c]$ for some $c \in \mathbb{N}$, but not that $\mathsf{P}^{\mathsf{NP}} \subseteq \text{i.o.SIZE}[n^k]$.

A more precise statement

- $\mathcal{L}_{PV}$-formulas $\varphi(x)$ interpreted over $\mathbb{N}$ can define languages in P, NP, etc.

- The sentence $\mathsf{UB}_k^{\text{i.o.}}(\varphi)$ expresses that the corresponding n-bit boolean functions are computed infinitely often by circuits of size n^k:

$$\forall 1^{(\ell)}\, \exists 1^{(n)}(n \geq \ell)\, \exists C_n(|C_n| \leq n^k)\, \forall x(|x| = n),\ \varphi(x) \equiv (C_n(x) = 1)$$

Theorem

For any of the following pairs of an $\mathcal{L}_{PV}$-theory T and a uniform complexity class $\mathcal{C}$:

(a) $T = T_2^1$ *and* $\mathcal{C} = \mathsf{P}^{\mathsf{NP}}$,

(b) $T = S_2^1$ *and* $\mathcal{C} = \mathsf{NP}$,

(c) $T = \mathsf{PV}$ *and* $\mathcal{C} = \mathsf{P}$,

there is an $\mathcal{L}_{PV}$-formula $\varphi(x)$ defining a language $L \in \mathcal{C}$ such that T does not prove the sentence $\mathsf{UB}_k^{\text{i.o.}}(\varphi)$.

High-level ideas

- Two approaches (forget the "i.o." condition for now):

$$T_2^1 \quad \nvdash \quad \mathsf{P}^{\mathsf{NP}} \subseteq \text{i.o.SIZE}[n^k]$$
$$S_2^1 \quad \nvdash \quad \mathsf{NP} \subseteq \text{i.o.SIZE}[n^k]$$

Main ingredient is the use of **"logical" Karp-Lipton theorems**.

$$\mathsf{PV} \quad \nvdash \quad \mathsf{P} \subseteq \text{i.o.SIZE}[n^k]$$

Extract from (non-uniform) circuit upper bound proofs a **"uniform construction"**.

Approach 1: “Logical” Karp-Lipton theorems

- A few unconditional circuit lower bounds in complexity theory use KL theorems. For instance, $\mathsf{ZPP}^{\mathsf{NP}} \not\subseteq \mathsf{SIZE}[n^k]$ can be derived from:

[Kobler-Watanabe’98] If $\mathsf{NP} \subseteq \mathsf{SIZE}[\text{poly}]$ then $\mathsf{PH} \subseteq \mathsf{ZPP}^{\mathsf{NP}}$.

Approach 1: "Logical" Karp-Lipton theorems

- A few unconditional circuit lower bounds in complexity theory use KL theorems. For instance, $\mathsf{ZPP}^{\mathsf{NP}} \not\subseteq \mathsf{SIZE}[n^k]$ can be derived from:

[Kobler-Watanabe'98] If $\mathsf{NP} \subseteq \mathsf{SIZE}[\mathrm{poly}]$ then $\mathsf{PH} \subseteq \mathsf{ZPP}^{\mathsf{NP}}$.

- Stronger collapses provide better lower bounds. It is not known how to collapse to P^{NP}.

Better KL theorems in fact necessary in this case [Chen-McKay-Murray-Williams'19].

Approach 1: "Logical" Karp-Lipton theorems

▶ A few unconditional circuit lower bounds in complexity theory use KL theorems. For instance, $\mathsf{ZPP}^{\mathsf{NP}} \not\subseteq \mathsf{SIZE}[n^k]$ can be derived from:

[Kobler-Watanabe'98] If $\mathsf{NP} \subseteq \mathsf{SIZE}[\mathsf{poly}]$ then $\mathsf{PH} \subseteq \mathsf{ZPP}^{\mathsf{NP}}$.

▶ Stronger collapses provide better lower bounds. It is not known how to collapse to P^{NP}.

Better KL theorems in fact necessary in this case [Chen-McKay-Murray-Williams'19].

[Cook-Krajicek'07] If $\mathsf{NP} \subseteq \mathsf{SIZE}[\mathsf{poly}]$ and **this is provable in a theory** $T \in \{\mathsf{PV}, S^1_2, T^1_2\}$, then PH collapses to a complexity class $\mathcal{C}_T \subseteq \mathsf{P}^{\mathsf{NP}}$.

Approach 2: A “bridge” between uniform and non-uniform circuits

If PV $\vdash$ P $\subseteq$ SIZE$[n^k]$, try to extract from PV-proof a “uniform” circuit family for each $L \in$ P.

This would contradict known separation P $\not\subseteq$ P-unifom-SIZE$[n^k]$ [Santhanam-Williams’13].

Approach 2: A "bridge" between uniform and non-uniform circuits

If PV $\vdash$ P $\subseteq$ SIZE$[n^k]$, try to extract from PV-proof a "uniform" circuit family for each $L \in$ P.

This would contradict known separation P $\not\subseteq$ P-unifom-SIZE$[n^k]$ [Santhanam-Williams'13].

- This doesn't quite work, but is the main intuition behind [Krajicek-Oliveira'17].

Approach 2: A “bridge” between uniform and non-uniform circuits

If PV $\vdash$ P $\subseteq$ SIZE$[n^k]$, try to extract from PV-proof a “uniform” circuit family for each $L \in$ P.

This would contradict known separation P $\not\subseteq$ P-unifom-SIZE$[n^k]$ [Santhanam-Williams’13].

- This doesn’t quite work, but is the main intuition behind [Krajicek-Oliveira’17].

- Theorem 1 (c) strengthens Krajicek-Oliveira to rule out PV $\vdash$ P $\subseteq$ i.o.SIZE$[n^k]$.

Approach 2: A “bridge” between uniform and non-uniform circuits

If PV $\vdash$ P $\subseteq$ SIZE$[n^k]$, try to extract from PV-proof a “uniform” circuit family for each $L \in$ P.

This would contradict known separation P $\not\subseteq$ P-unifom-SIZE$[n^k]$ [Santhanam-Williams’13].

- This doesn’t quite work, but is the main intuition behind [Krajicek-Oliveira’17].

- Theorem 1 (c) strengthens Krajicek-Oliveira to rule out PV $\vdash$ P $\subseteq$ i.o.SIZE$[n^k]$.

Complications appear because Santhanam-Williams doesn’t provide a.e. lower bounds.

The unprovability result of Krajicek-Oliveira’17

The sentence $\mathsf{UB}_{k,c}(h)$ expresses that function symbol h admits circuits of size $\leq cn^k$.

Theorem. For every $k \geq 1$ there is a PV function symbol h such that for no constant $c \geq 1$ PV proves the sentence $\mathsf{UB}_{k,c}(h)$.

The unprovability result of Krajicek-Oliveira'17

The sentence $\mathsf{UB}_{k,c}(h)$ expresses that function symbol h admits circuits of size $\leq cn^k$.

Theorem. For every $k \geq 1$ there is a PV function symbol h such that for no constant $c \geq 1$ PV proves the sentence $\mathsf{UB}_{k,c}(h)$.

Remark. $\mathsf{UB}_{k,c}(h)$ is a $\forall\exists\forall$-sentence in $\mathcal{L}_{\mathsf{PV}}$, and can be written as:

$$\mathsf{UB}_{k,c}(h) \quad \equiv \quad \forall z\, \exists C\, \forall x\, \phi_h(z, C, x), \text{ where } \phi_h \text{ is quantifier-free.}$$

The basic idea

- Logic/Provability as a bridge between <u>non-uniform</u> and <u>uniform</u> computations.

If PV $\vdash$ $\text{UB}_{k,c}(h)$ using a proof π (sequence of symbols), extract from π computational information about sequence C_n of circuits computing h.

The basic idea

- Logic/Provability as a bridge between non-uniform and uniform computations.

If PV $\vdash$ UB$_{k,c}(h)$ using a proof π (sequence of symbols), extract from π computational information about sequence C_n of circuits computing h.

- Since PV is sound, provability of a sentence implies that the sentence is true in the usual sense (in $\mathbb{N}$).

The basic idea

- Logic/Provability as a bridge between non-uniform and uniform computations.

If PV $\vdash$ UB$_{k,c}(h)$ using a proof π (sequence of symbols), extract from π computational information about sequence C_n of circuits computing h.

- Since PV is sound, provability of a sentence implies that the sentence is true in the usual sense (in $\mathbb{N}$).

- Perhaps contradict known (unconditional) lower bounds in uniform circuit complexity ?

The basic idea

- Logic/Provability as a bridge between non-uniform and uniform computations.

If PV $\vdash$ $\text{UB}_{k,c}(h)$ using a proof π (sequence of symbols), extract from π computational information about sequence C_n of circuits computing h.

- Since PV is sound, provability of a sentence implies that the sentence is true in the usual sense (in $\mathbb{N}$).

- Perhaps contradict known (unconditional) lower bounds in uniform circuit complexity ?

(We will later explain an important issue with this idea, and how it can be fixed.)

The uniform lower bound

R. Santhanam and R. Williams, "*On uniformity and circuit lower bounds*", 2014.

Theorem. For every $k \geq 1$, there is $L \in$ P such that $L \notin$ P-uniform-SIZE(n^k).

The uniform lower bound

R. Santhanam and R. Williams, "*On uniformity and circuit lower bounds*", 2014.

Theorem. For every $k \geq 1$, there is $L \in$ P such that $L \notin$ P-uniform-SIZE(n^k).

Why is this result so special?

$L \in$ DTIME(n^ℓ), but P-uniform generating algorithm can run in time n^{2^ℓ}, $n^{2^{2^{\ell \cdot k}}}$, etc.

The uniform lower bound

R. Santhanam and R. Williams, "*On uniformity and circuit lower bounds*", 2014.

Theorem. For every $k \geq 1$, there is $L \in$ P such that $L \notin$ P-uniform-SIZE(n^k).

Why is this result so special?

$L \in$ DTIME(n^ℓ), but P-uniform generating algorithm can run in time n^{2^ℓ}, $n^{2^{2^{\ell \cdot k}}}$, etc.

► Proof is a clever win-win argument by contradiction (non-constructive), and relies on a time hierarchy theorem with advice.

The uniform lower bound

R. Santhanam and R. Williams, "*On uniformity and circuit lower bounds*", 2014.

Theorem. For every $k \geq 1$, there is $L \in \mathsf{P}$ such that $L \notin$ P-uniform-SIZE(n^k).

Why is this result so special?

$L \in$ DTIME(n^ℓ), but P-uniform generating algorithm can run in time n^{2^ℓ}, $n^{2^{2^{\ell \cdot k}}}$, etc.

▶ Proof is a clever win-win argument by contradiction (non-constructive), and relies on a time hierarchy theorem with advice.

▶ **Our Approach.** From a PV-proof of UB$_{k,c}(h)$, we try to extract a poly-time generating algorithm. We can't control its p-time bound, but this is okay with the theorem above!

The KPT Witnessing Theorem

J. Krajíček, P. Pudlák, and G. Takeuti: "*Bounded arithmetic and the polynomial hierarchy*", 1991.

Theorem. Assume T is a universal theory with vocabulary $\mathcal{L}$, ϕ is a quantifier-free $\mathcal{L}$-formula, and

$$T \vdash \forall z \exists C \forall x \;\; \phi(z, C, x) \; .$$

Then there exist a constant $d \geq 1$ and a finite sequence $t_1, \ldots, t_d$ of $\mathcal{L}$-terms such that

$$T \vdash \phi(z, t_1(z), x_1) \;\vee\; \phi(z, t_2(z, x_1), x_2) \;\vee\; \ldots \;\vee\; \phi(z, t_d(z, x_1, \ldots, x_{d-1}), x_d).$$

The KPT Witnessing Theorem

J. Krajíček, P. Pudlák, and G. Takeuti: "*Bounded arithmetic and the polynomial hierarchy*", 1991.

Theorem. Assume T is a universal theory with vocabulary $\mathcal{L}$, ϕ is a quantifier-free $\mathcal{L}$-formula, and

$$T \vdash \forall z \exists C \forall x \;\; \phi(z, C, x) \; .$$

Then there exist a constant $d \geq 1$ and a finite sequence $t_1, \ldots, t_d$ of $\mathcal{L}$-terms such that

$$T \vdash \phi(z, t_1(z), x_1) \;\vee\; \phi(z, t_2(z, x_1), x_2) \;\vee\; \ldots \;\vee\; \phi(z, t_d(z, x_1, \ldots, x_{d-1}), x_d).$$

- The result can be established using proof theory or model theory.

Applying the KPT Theorem to PV and $\mathsf{UB}_{k,c}(f)$

- Fix $k \geq 1$, and assume that for every $f \in \mathcal{L}_{\mathsf{PV}}$ we have $c \geq 1$ such that

$$\mathsf{PV} \vdash \mathsf{UB}_{k,c}(f) \qquad \text{Recall that this is } \forall z\, \exists C\, \forall x\, \phi_f(z, C, x).$$

Applying the KPT Theorem to PV and $\text{UB}_{k,c}(f)$

- Fix $k \geq 1$, and assume that for every $f \in \mathcal{L}_{\text{PV}}$ we have $c \geq 1$ such that

$$\text{PV} \vdash \text{UB}_{k,c}(f) \qquad \text{Recall that this is } \forall z\, \exists C\, \forall x\, \phi_f(z, C, x).$$

- Assume we get $d = 1$ after applying the KPT statement, i.e.,

$$\text{PV} \vdash \phi_f(z, t_1^f(z), x_1)\ , \qquad \text{where } t_1^f(z) \text{ is an } \mathcal{L}_{\text{PV}}\text{-term.}$$

Applying the KPT Theorem to PV and $\mathsf{UB}_{k,c}(f)$

- Fix $k \geq 1$, and assume that for every $f \in \mathcal{L}_{\mathsf{PV}}$ we have $c \geq 1$ such that

$$\mathsf{PV} \vdash \mathsf{UB}_{k,c}(f) \qquad \text{Recall that this is } \forall z\, \exists C\, \forall x\, \phi_f(z, C, x).$$

- Assume we get $d = 1$ after applying the KPT statement, i.e.,

$$\mathsf{PV} \vdash \phi_f(z, t_1^f(z), x_1)\ , \qquad \text{where } t_1^f(z) \text{ is an } \mathcal{L}_{\mathsf{PV}}\text{-term.}$$

- Then, by the soundness of PV, if we set z to be some n-bit integer $1^{(n)}$,

$$\mathbb{N} \models \forall x_1\, \phi_f(1^{(n)}, t_1^f(1^{(n)}), x_1).$$

Applying the KPT Theorem to PV and $\mathrm{UB}_{k,c}(f)$

- Fix $k \geq 1$, and assume that <u>for every</u> $f \in \mathcal{L}_{\mathrm{PV}}$ we have $c \geq 1$ such that

  $\mathrm{PV} \vdash \mathrm{UB}_{k,c}(f)$ Recall that this is $\forall z \, \exists C \, \forall x \, \phi_f(z, C, x)$.

- Assume we get $d = 1$ after applying the KPT statement, i.e.,

  $\mathrm{PV} \vdash \phi_f(z, t_1^f(z), x_1)$, where $t_1^f(z)$ is an $\mathcal{L}_{\mathrm{PV}}$-term.

- Then, by the soundness of PV, if we set z to be some n-bit integer $1^{(n)}$,

$$\mathbb{N} \models \forall x_1 \, \phi_f(1^{(n)}, t_1^f(1^{(n)}), x_1).$$

- Now $t_1^f(1^{(n)})$, a term in PV, corresponds in $\mathbb{N}$ to a poly-time computation.
  The assumption that we get this for all $f \in \mathcal{L}_{\mathrm{PV}}$ contradicts Santhanam-Williams.

Applying the KPT Theorem to PV and $\mathsf{UB}_{k,c}(f)$

- Fix $k \geq 1$, and assume that for every $f \in \mathcal{L}_{\mathsf{PV}}$ we have $c \geq 1$ such that

$$\mathsf{PV} \vdash \mathsf{UB}_{k,c}(f) \qquad \text{Recall that this is } \forall z\, \exists C\, \forall x\, \phi_f(z, C, x).$$

- Assume we get $d = 1$ after applying the KPT statement, i.e.,

$$\mathsf{PV} \vdash \phi_f(z, t_1^f(z), x_1)\ , \qquad \text{where } t_1^f(z) \text{ is an } \mathcal{L}_{\mathsf{PV}}\text{-term.}$$

- Then, by the soundness of PV, if we set z to be some n-bit integer $1^{(n)}$,

$$\mathbb{N} \models \forall x_1\, \phi_f(1^{(n)}, t_1^f(1^{(n)}), x_1).$$

- Now $t_1^f(1^{(n)})$, a term in PV, corresponds in $\mathbb{N}$ to a poly-time computation.
  The assumption that we get this for all $f \in \mathcal{L}_{\mathsf{PV}}$ contradicts Santhanam-Williams. $\square$

The general case

- If $d > 1$, we obtain from $\mathsf{PV} \vdash \mathsf{UB}_{k,c}(f)$ the more general scenario:

$$\mathbb{N} \models \phi(z, t_1(z), x_1) \ \lor \ \phi(z, t_2(z, x_1), x_2) \ \lor \ \ldots \ \lor \ \phi(z, t_d(z, x_1, \ldots, x_{d-1}), x_d).$$

The general case

- If $d > 1$, we obtain from PV $\vdash$ UB$_{k,c}(f)$ the more general scenario:

$$\mathbb{N} \models \phi(z, t_1(z), x_1) \ \vee \ \phi(z, t_2(z, x_1), x_2) \ \vee \ \ldots \ \vee \ \phi(z, t_d(z, x_1, \ldots, x_{d-1}), x_d).$$

Either $t_1(1^{(n)})$ outputs a correct circuit for f, or

There is a counter-example $a_1 \in \{0,1\}^n$, and $t_2(1^{(n)}, a_1)$ outputs a correct circuit, or

$\ldots$

The general case

- If $d > 1$, we obtain from PV $\vdash$ $\mathsf{UB}_{k,c}(f)$ the more general scenario:

$$\mathbb{N} \models \phi(z, t_1(z), x_1) \ \vee \ \phi(z, t_2(z, x_1), x_2) \ \vee \ \ldots \ \vee \ \phi(z, t_d(z, x_1, \ldots, x_{d-1}), x_d).$$

Either $t_1(1^{(n)})$ outputs a correct circuit for f, or
There is a counter-example $a_1 \in \{0,1\}^n$, and $t_2(1^{(n)}, a_1)$ outputs a correct circuit, or
…

- Due to the counter-examples, we can only show that $f \in$ [P-uniform / $O(n)$]-SIZE(n^k).

The general case

- If $d > 1$, we obtain from PV $\vdash$ UB$_{k,c}(f)$ the more general scenario:

$$\mathbb{N} \models \phi(z, t_1(z), x_1) \ \vee \ \phi(z, t_2(z, x_1), x_2) \ \vee \ \ldots \ \vee \ \phi(z, t_d(z, x_1, \ldots, x_{d-1}), x_d).$$

Either $t_1(1^{(n)})$ outputs a correct circuit for f, or

There is a counter-example $a_1 \in \{0,1\}^n$, and $t_2(1^{(n)}, a_1)$ outputs a correct circuit, or

$\ldots$

- Due to the counter-examples, we can only show that $f \in$ [P-uniform / $O(n)$]-SIZE(n^k).

- **Contradiction? A difficulty is the lack of super-linear non-uniform lower bounds!**

How to establish an unconditional unprovability result?

- Apply KPT to a specific $\mathsf{UB}_{k,c}(g)$, obtaining a disjunction of $\leq d$ formulas, $d \in \mathbb{N}$.
  (We will eliminate one by one in d stages, until we get a contradiction.)

How to establish an unconditional unprovability result?

- Apply KPT to a specific UB$_{k,c}(g)$, obtaining a disjunction of $\leq d$ formulas, $d \in \mathbb{N}$.
  (We will eliminate one by one in d stages, until we get a contradiction.)

- To handle each elimination step, we formalize the SW win-win argument inside PV.

How to establish an unconditional unprovability result?

- Apply KPT to a specific $\mathrm{UB}_{k,c}(g)$, obtaining a disjunction of $\leq d$ formulas, $d \in \mathbb{N}$.
  (We will eliminate one by one in d stages, until we get a contradiction.)

- To handle each elimination step, we formalize the SW win-win argument inside PV.

The following ideas are crucial:

- Using the constructivity of PV and Herbrand's Theorem, it can be shown that the counter-examples that previously caused difficulties can be provably witnessed in PV.

How to establish an unconditional unprovability result?

- Apply KPT to a specific $\mathrm{UB}_{k,c}(g)$, obtaining a disjunction of $\leq d$ formulas, $d \in \mathbb{N}$.
  (We will eliminate one by one in d stages, until we get a contradiction.)

- To handle each elimination step, we formalize the SW win-win argument inside PV.

The following ideas are crucial:

- Using the constructivity of PV and Herbrand's Theorem, it can be shown that the counter-examples that previously caused difficulties can be provably witnessed in PV.

- In the SW win-win analysis, the second case only needs a non-uniform assumption. This allows us to move from d to $d-1$ (our result is about non-uniform upper bounds!).

How to establish an unconditional unprovability result?

- Apply KPT to a specific $\mathsf{UB}_{k,c}(g)$, obtaining a disjunction of $\leq d$ formulas, $d \in \mathbb{N}$.
  (We will eliminate one by one in d stages, until we get a contradiction.)

- To handle each elimination step, we formalize the SW win-win argument inside PV.

The following ideas are crucial:

- Using the constructivity of PV and Herbrand's Theorem, it can be shown that the counter-examples that previously caused difficulties can be provably witnessed in PV.

- In the SW win-win analysis, the second case only needs a non-uniform assumption. This allows us to move from d to $d - 1$ (our result is about non-uniform upper bounds!).

Check the papers for more details!

Bounded theories and a.e. vs i.o. circuit bounds

Parikh's Theorem. Let $A(\vec{x}, y)$ be a bounded formula.

$$\text{If } \; I\Delta_0 \vdash \forall \vec{x}\, \exists \boldsymbol{y}\, A(\vec{x}, y) \; \text{ then } \; I\Delta_0 \vdash \forall \vec{x}\, \exists \boldsymbol{y} \leq \boldsymbol{t}(\vec{\boldsymbol{x}})\, A(\vec{x}, y).$$

Bounded theories and a.e. vs i.o. circuit bounds

Parikh's Theorem. Let $A(\vec{x}, y)$ be a bounded formula.

$$\text{If } \; I\Delta_0 \vdash \forall \vec{x}\, \exists \boldsymbol{y}\, A(\vec{x}, y) \; \text{ then } \; I\Delta_0 \vdash \forall \vec{x}\, \exists \boldsymbol{y} \leq \boldsymbol{t}(\vec{\boldsymbol{x}})\, A(\vec{x}, y).$$

- We use similar results to "tame" i.o. upper bounds in bounded arithmetic.

Example: If $T^1_2 \vdash \mathsf{SAT} \in \text{i.o.}\mathsf{SIZE}[n^k]$ then $T^1_2 \vdash \mathsf{SAT} \in \mathsf{SIZE}[n^{k'}]$.

Bounded theories and a.e. vs i.o. circuit bounds

Parikh's Theorem. Let $A(\vec{x}, y)$ be a bounded formula.

If $I\Delta_0 \vdash \forall \vec{x}\, \exists \boldsymbol{y}\, A(\vec{x}, y)$ then $I\Delta_0 \vdash \forall \vec{x}\, \exists \boldsymbol{y} \leq \boldsymbol{t}(\vec{\boldsymbol{x}})\, A(\vec{x}, y)$.

- We use similar results to "tame" i.o. upper bounds in bounded arithmetic.

Example: If $T_2^1 \vdash \text{SAT} \in \text{i.o.SIZE}[n^k]$ then $T_2^1 \vdash \text{SAT} \in \text{SIZE}[n^{k'}]$.

- Not every language is paddable, and more delicate arguments are needed.

Concluding Remarks: Logic and P vs NP

- A major question is to establish the unprovability of P = NP:

For a function symbol $f \in \mathcal{L}_{PV}$, consider the universal sentence

$$\varphi_{\mathsf{P}=\mathsf{NP}}(f) \stackrel{\text{def}}{=} \forall x \, \forall y \; \psi_{\mathsf{SAT}}(x, y) \rightarrow \psi_{\mathsf{SAT}}(x, f(x))$$

Concluding Remarks: Logic and P vs NP

- A major question is to establish the unprovability of P = NP:

For a function symbol $f \in \mathcal{L}_{PV}$, consider the universal sentence

$$\varphi_{\mathsf{P}=\mathsf{NP}}(f) \stackrel{\text{def}}{=} \forall x \, \forall y \; \psi_{\mathsf{SAT}}(x,y) \rightarrow \psi_{\mathsf{SAT}}(x, f(x))$$

Conjecture. For no function symbol f in $\mathcal{L}_{PV}$ theory PV proves the sentence $\varphi_{\mathsf{P}=\mathsf{NP}}(f)$.

Concluding Remarks: Logic and P vs NP

- A major question is to establish the unprovability of P = NP:

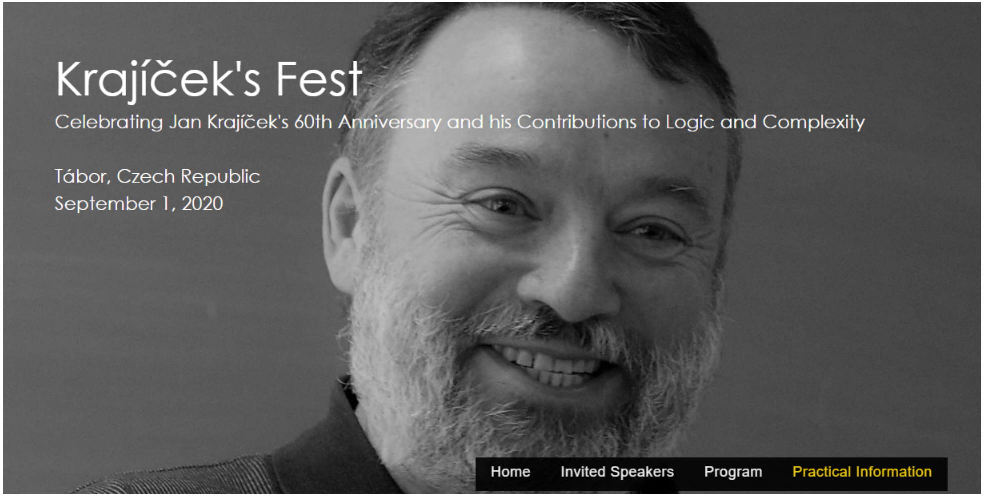
For a function symbol $f \in \mathcal{L}_{PV}$, consider the universal sentence

$$\varphi_{\mathsf{P}=\mathsf{NP}}(f) \stackrel{\text{def}}{=} \forall x\, \forall y\; \psi_{\mathsf{SAT}}(x, y) \rightarrow \psi_{\mathsf{SAT}}(x, f(x))$$

Conjecture. For no function symbol f in $\mathcal{L}_{PV}$ theory PV proves the sentence $\varphi_{\mathsf{P}=\mathsf{NP}}(f)$.

- Reduces to the study of unprovability of circuit **lower** bounds (Theorem 2 in our work).
- Motivates **both** research directions (**unprovability of upper and lower bounds**).

Thank you


Krajíček's Fest
Celebrating Jan Krajíček's 60th Anniversary and his Contributions to Logic and Complexity
Tábor, Czech Republic
September 1, 2020
Home
Invited Speakers
Program
Practical Information

Complexity Theory with a Human Face

1-4 September 2020, Tábor, Czech Republic