Mining the Surface: Proof Mining in the Bounded World

Amir Akbar Tabatabai

Czech Academy of Sciences

amir.akbar@gmail.com

July 3, 2018

A Proof Theoretical Dream

Let's begin with the following problem:

Π^0_1-Independence

Let $A \in \Pi^0_1$ be an arithmetical statement and T an arithmetical theory. How can we prove that A is unprovable in T?

A Proof Theoretical Dream

Let's begin with the following problem:

Π^0_1-Independence

Let $A \in \Pi^0_1$ be an arithmetical statement and T an arithmetical theory. How can we prove that A is unprovable in T?

There are some well-known methods to solve this problem, including:

A Proof Theoretical Dream

Let's begin with the following problem:

Π^0_1-Independence

Let $A \in \Pi^0_1$ be an arithmetical statement and T an arithmetical theory. How can we prove that A is unprovable in T?

There are some well-known methods to solve this problem, including:

- Using the second incompleteness theorem. In this case it is enough to reduce A to the consistency of T,

A Proof Theoretical Dream

Let's begin with the following problem:

Π^0_1-Independence

Let $A \in \Pi^0_1$ be an arithmetical statement and T an arithmetical theory. How can we prove that A is unprovable in T?

There are some well-known methods to solve this problem, including:

- Using the second incompleteness theorem. In this case it is enough to reduce A to the consistency of T,
- Using propositional proof complexity. If T is a bounded theory of arithmetic, it is possible to transform a proof of A in T to a sequence of short proofs of the propositional version of A, in a corresponding propositional calculus $\mathcal{P}_T$. Then proving a super-polynomial lower bound for $\mathcal{P}_T$ leads to the unprovability of A in T.

  Example: Pigeonhole principle is unprovable in $I\Delta_0(R)$.

A Proof Theoretical Dream

But there is no general method to prove the unprovability of a Π^0_1 statement. Compare the situation with Π^0_2 statements. Using ordinal analysis we have a characterization of all provably recursive functions by their growth rate captured by the proof theoretic ordinal of the theory. Hence:

A Proof Theoretical Dream

But there is no general method to prove the unprovability of a Π^0_1 statement. Compare the situation with Π^0_2 statements. Using ordinal analysis we have a characterization of all provably recursive functions by their growth rate captured by the proof theoretic ordinal of the theory. Hence:

To prove the unprovability of a Π^0_2 statement, it is enough to show that it defines a function with higher growth rate than what α_T captures.

A Proof Theoretical Dream

But there is no general method to prove the unprovability of a Π^0_1 statement. Compare the situation with Π^0_2 statements. Using ordinal analysis we have a characterization of all provably recursive functions by their growth rate captured by the proof theoretic ordinal of the theory.
Hence:

To prove the unprovability of a Π^0_2 statement, it is enough to show that it defines a function with higher growth rate than what α_T captures.

The reason why the Π^0_1 case is so complex is that the class Π^0_1 can be interpreted as a hierarchy of bounded formulas and hence its behavior depends on the open conjectures of the complexity theory.
For this matter, narrow down the problem to:

Total NP-Search Problems

Total NP-Search Problems

Let $B(x, y)$ be a p-time computable predicate, $A = \forall x \exists |y| \leq p(|x|) B(x, y)$ and T an arithmetical theory. How can we prove that A is unprovable in T?

Total NP-Search Problems

Total NP-Search Problems

Let $B(x, y)$ be a p-time computable predicate, $A = \forall x \exists |y| \leq p(|x|) B(x, y)$ and T an arithmetical theory. How can we prove that A is unprovable in T?

- The growth rate does not work anymore because the function is already bounded by the exponential function. Hence the height of the function does not work. But what about its width, namely the complexity of its most clever algorithm?

Total NP-Search Problems

Total NP-Search Problems

Let $B(x, y)$ be a p-time computable predicate, $A = \forall x \exists |y| \leq p(|x|) B(x, y)$ and T an arithmetical theory. How can we prove that A is unprovable in T?

- The growth rate does not work anymore because the function is already bounded by the exponential function. Hence the height of the function does not work. But what about its width, namely the complexity of its most clever algorithm?
- There is a trivial brute-force algorithm to find y based on searching all possible values below $2^{p(|x|)}$. The soundness of this algorithm is just based on the the fact that A is true. But we also know a T-proof of A.

The Main Theorem (informal)

Following Kreisel, let us ask the following question:

Proof Mining

Does a T-proof of A lead to a more clever algorithm than the blind brute-force? Is it possible to measure this sort of "cleverness" by ordinals?

The Main Theorem (informal)

Following Kreisel, let us ask the following question:

Proof Mining

Does a T-proof of A lead to a more clever algorithm than the blind brute-force? Is it possible to measure this sort of "cleverness" by ordinals?

The answer is Yes!

The Main Theorem (informal)

Let $B(x, y)$ be a p-time computable predicate, p a polynomial, $A = \forall x \exists |y| \leq p(|x|) B(x, y)$ and T an arithmetical theory with the proof theoretic ordinal α. Then TFAE:

- $T \vdash A$
- There exists $\beta \prec \alpha$ and a sequence of the length β of *poly-time verifiable* computational steps beginning by zero and ending in y such that $B(x, y)$.

It is not that obvious!

We mentioned that A is solvable using a brute-force algorithm of checking all possibilities for y. It takes finite (i.e., $2^{p(|x|)}$) many steps, far less than any infinite ordinal. Where does the problem lie?

It is not that obvious!

We mentioned that A is solvable using a brute-force algorithm of checking all possibilities for y. It takes finite (i.e., $2^{p(|x|)}$) many steps, far less than any infinite ordinal. Where does the problem lie?

Poly-time Verifiablity

A step is called poly-time verifiable if it represents a poly-time algorithm whose soundness is provable via poly-time reasoning in PV.

It is not that obvious!

We mentioned that A is solvable using a brute-force algorithm of checking all possibilities for y. It takes finite (i.e., $2^{p(|x|)}$) many steps, far less than any infinite ordinal. Where does the problem lie?

Poly-time Verifiablity

A step is called poly-time verifiable if it represents a poly-time algorithm whose soundness is provable via poly-time reasoning in PV.

The brute-force algorithm is not local:

The Algorithm: *Check all y's in order till reaching a y such that $B(x, y)$. Then keep that y till the end of the process and then return it.*

However, to ensure that this y works, you have to know the existence of a $y \leq 2^{p(|x|)} B(x, y)$ already, and this is not necessarily provable in PV.

Formalization: A Poly-time Ordinal Representation

Definition

Let α be an ordinal with a primitive recursive representation. Then we say

$$\mathbb{A} = (A, \prec_A, +_A, \cdot_A, -_A, d_A(\cdot,\cdot), 0_A, 1_A)$$

is a polytime representation of the ordinal α when A and $\prec_A$ are polytime relations, $+_A, \cdot_A, -_A, d_A(\cdot,\cdot)$ are polytime functions and constants $0_A, 1_A$ such that:

(*i*) The structure $\mathbb{A} = (A, \prec_A, +_A, \cdot_A, -_A, d_A(\cdot,\cdot), 0_A, 1_A)$ is isomorphic to $\mathfrak{A} = (\alpha, \prec_\alpha, +_\alpha, \cdot_\alpha, -_\alpha, d_A(\cdot,\cdot), 0_\alpha, 1_\alpha)$ where $-_\alpha, d_A(\cdot,\cdot)$ are subtraction and division from right, i.e. for $\beta \preceq \alpha$ we have $\alpha - \beta = \gamma$ where $\beta + \gamma = \alpha$ and otherwise, $\alpha - \beta = 0$. For division, if $\beta \neq 0$, by $d(\alpha, \beta)$ we mean the unique γ where $\alpha = \beta\gamma + \delta$ and $\delta \prec \beta$.

Formalization: A Poly-time Ordinal Representation

Definition

(*ii*) PV proves the axioms of discrete ordered semi-rings for the structure $\mathbb{A}$ without the commutativity of addition and the axioms which state that $\prec_A$ preserves under left addition and left multiplication by a non-zero element.

(*iii*) PRA proves that $\mathbb{A}$ is equivalent to the primitive recursive representation of $\mathfrak{A}$.

Ordinal Flows

Define $\forall_1$ as the class of all universal formulas which is inductively defined as the least set that includes p-time predicates and is closed under conjunction, disjunction, implication with p-time precedent and universal quantifiers.

Definition

Let $A(\vec{x})$, $B(\vec{x})$ and $H(\delta, \vec{x})$ be some formulas in $\forall_1$. A tuple (H, β) where $\beta \prec \alpha$ is called an α-flow if

(*i*) $\mathrm{PV} \vdash A(\vec{x}) \leftrightarrow H(0, \vec{x})$.

(*ii*) $\mathrm{PV} \vdash \forall\, 1 \preceq \delta \prec \beta\ [\forall \gamma \prec \delta\ H(\gamma, \vec{x}) \rightarrow \forall \gamma \prec \delta + 1\ H(\gamma, \vec{x})]$.

(*iii*) $\mathrm{PV} \vdash H(\beta, \vec{x}) \leftrightarrow B(\vec{x})$.

We denote the existence of an α-flow from A to B by $A \rhd_\alpha B$ and we abbreviate $\bigwedge \Gamma \rhd_\alpha \bigvee \Delta$ by $\Gamma \rhd_\alpha \Delta$. Moreover, when it is clear from the context, we omit the subscript α everywhere.

The Main Lemma

The following lemma establishes a high-level calculus for the ordinal flows:

The Main Lemma

- (Conjunction left.) If $\Gamma, A \rhd \Delta$ then $\Gamma, A \wedge B \rhd \Delta$ and $\Gamma, B \wedge A \rhd \Delta$.
- (Conjunction right.) If $\Gamma \rhd \Delta, A$ and $\Gamma \rhd \Delta, B$ then $\Gamma \rhd \Delta, A \wedge B$.
- (Disjunction left.) If $\Gamma, A \rhd \Delta$ and $\Gamma, B \rhd \Delta$ then $\Gamma, A \vee B \rhd \Delta$.
- (Disjunction right.) If $\Gamma \rhd \Delta, A$ then $\Gamma \rhd \Delta, A \vee B$ and $\Gamma \rhd \Delta, B \vee A$.
- (Cut.) If $\Gamma \rhd \Delta, A$ and $\Gamma', A \rhd \Delta'$ then $\Gamma, \Gamma' \rhd \Delta, \Delta'$.
- (Contraction left.) If $\Gamma, A, A \rhd \Delta$ then $\Gamma, A \rhd \Delta$.
- (Contraction right.) If $\Gamma \rhd \Delta, A, A$ then $\Gamma \rhd \Delta, A$.
- (Universal left.) If $\Gamma, A(t) \rhd \Delta$ then $\Gamma, \forall y\ A(y) \rhd \Delta$.
- (Universal right.) If $\Gamma \rhd \Delta, A(y)$ then $\Gamma \rhd \Delta, \forall y\ A(y)$.
- (Induction.) If $\Gamma, \forall \gamma \prec \delta\ A(\gamma) \rhd \Delta, \forall \gamma \prec \delta + 1\ A(\gamma)$ then $\Gamma \rhd \Delta, A(\beta)$.

The Main Theorem (formal)

The Main Theorem (A.A.)

Let T be a theory of arithmetic, α_T be its proof theoretic ordinal with a polynomial time representation and $\Gamma \cup \Delta \subseteq \forall_1$. Then $T \vdash \Gamma \Rightarrow \Delta$ iff $\Gamma \rhd_{\alpha_T} \Delta$.

Proof.

The completeness part is easy. For the soundness part, use continuous cut elimination technique to transform the proof in T to a proof in $\mathrm{PRA} + \bigcup_{\beta \prec \alpha} \forall_1 - \mathrm{TI}(\prec_\beta)$. Then interpret this theory in a theory consisting of just p-time functions and transfinite induction on formulas in $\forall_1$. Finally use induction on the length of the last proof and the main lemma to prove that if $\Gamma \Rightarrow \Delta$ is provable, then $\Gamma \rhd \Delta$. $\square$

The Main Theorem (formal)

Paraphrasing the main theorem, we have:

Theorem (A.A.)

Let $B(x, y)$ be a p-time computable predicate, p be a polynomial, $A = \forall x \exists |y| \leq p(|x|) B(x, y)$ and T an arithmetical theory with the proof theoretic ordinal α. Then TFAE:

1. $T \vdash A$,
2. There exists $\beta \prec \alpha$, polytime computable functions f, g, h and a polytime computable predicate $G(u, z)$ such that:
   - $\mathrm{PV} \vdash G(\beta, 0)$,
   - $\mathrm{PV} \vdash 0 \prec \gamma \preceq \beta \rightarrow g(\gamma, z) \prec \gamma$,
   - $\mathrm{PV} \vdash G(\gamma, z) \rightarrow G(g(\gamma, z), f(\gamma, z))$,
   - $\mathrm{PV} \vdash G(0, z) \rightarrow [|h(z)| \leq p(|x|) \wedge B(x, h(z))]$.

The Main Theorem (formal)

Proof.

Since $T \vdash \forall y(|y| \leq p(|x|) \rightarrow \neg B(x,y)) \Rightarrow \bot$ We have:

(*i*) $\mathrm{PV} \vdash [\forall y(|y| \leq p(|x|) \rightarrow \neg B(x,y))] \leftrightarrow H(0,\vec{x})$.

(*ii*) $\mathrm{PV} \vdash \forall\, 1 \preceq \delta \prec \theta\ [\forall \gamma \prec \delta\ H(\gamma,\vec{x}) \rightarrow \forall \gamma \prec \delta + 1\ H(\gamma,\vec{x})]$.

(*iii*) $\mathrm{PV} \vdash H(\theta,\vec{x}) \leftrightarrow \bot$.

Since $H \in \forall_1$ it is in the form $\forall z K(u,z)$ where K is a p-time predicate. W.l.o.g define $K(\theta+1,z) = \bot$. Using the Herbrand's theorem and some minor tricks we have some polytime functions f, g and h such that:

- $\mathrm{PV} \vdash \neg K(\theta+1,0)$,
- $\mathrm{PV} \vdash 0 \prec \gamma \preceq \theta + 1 \rightarrow g(\gamma,z) \prec \gamma$,
- $\mathrm{PV} \vdash \neg K(\gamma,z) \rightarrow \neg K(g(\gamma,z), f(\gamma,z))$,
- $\mathrm{PV} \vdash \neg K(0,z) \rightarrow [|h(z)| \leq p(|x|) \wedge B(x,h(z))]$.

Finally define $\beta = \theta + 1$ and $G = \neg K$. □

Applications I

Applying the theorem to PA we have:

Corollary (A. Beckmann)

Let $B(x, y)$ be a p-time computable predicate, p be a polynomial and $A = \forall x \exists |y| \leq p(|x|) B(x, y)$. Then TFAE:

1. $\mathrm{PA} \vdash A$,
2. There exists $\beta \prec \epsilon_0$, polytime computable functions f, g, h and a polytime computable predicate $G(u, z)$ such that:
   - $\mathrm{PV} \vdash G(\beta, 0)$,
   - $\mathrm{PV} \vdash 0 \prec \gamma \preceq \beta \to g(\gamma, z) \prec \gamma$,
   - $\mathrm{PV} \vdash G(\gamma, z) \to G(g(\gamma, z), f(\gamma, z))$,
   - $\mathrm{PV} \vdash G(0, z) \to [|h(z)| \leq p(|x|) \wedge B(x, h(z))]$.

Applications II

Applying the theorem to the extensions of PA we have:

Corollary (A.A.)

Let $B(x, y)$ be a p-time computable predicate, p be a polynomial, $A = \forall x \exists |y| \leq p(|x|) B(x, y)$ and α is closed under the operation $\beta \mapsto \omega^\beta$. Then TFAE:

1. $\mathrm{PA} + \bigcup_{\beta \prec \alpha} \mathrm{TI}(\beta) \vdash A$,
2. There exists $\beta \prec \alpha$, polytime computable functions f, g, h and a polytime computable predicate $G(u, z)$ such that:
   - $\mathrm{PV} \vdash G(\beta, 0)$,
   - $\mathrm{PV} \vdash 0 \prec \gamma \preceq \beta \to g(\gamma, z) \prec \gamma$,
   - $\mathrm{PV} \vdash G(\gamma, z) \to G(g(\gamma, z), f(\gamma, z))$,
   - $\mathrm{PV} \vdash G(0, z) \to [|h(z)| \leq p(|x|) \wedge B(x, h(z))]$.

Thank you for your attention!