Induction, search problems and approximate counting

Neil Thapen

Institute of Mathematics
Czech Academy of Sciences

Joint work with Leszek Kołodziejczyk

ASL Logic Colloquium 2019, Prague

Motivating question

How does the strength of a theory of arithmetic change, as we increase the amount of induction?

Motivating question

How does the strength of a theory of arithmetic change, as we increase the amount of induction?

We measure the **strength** of a theory by its set of consequences of some fixed low complexity, say Π_1 or Π_2.

Motivating question

How does the strength of a theory of arithmetic change, as we increase the amount of induction?

We measure the **strength** of a theory by its set of consequences of some fixed low complexity, say Π_1 or Π_2.

So we are asking: as we allow more complex formulas as induction hypotheses,

- are more Π_1 sentences provable?
- are more Π_2 sentences provable? Are more functions provably recursive?

Classical case - $I\Sigma_k$

The theory $I\Sigma_k$ consists of:

Classical case - $I\Sigma_k$

The theory $I\Sigma_k$ consists of:

A **base theory**, for which we could take

- the theory of discrete ordered rings, or
- elementary arithmetic, in a language with names for all Kalmar elementary functions

Classical case - $I\Sigma_k$

The theory $I\Sigma_k$ consists of:

A **base theory**, for which we could take

- the theory of discrete ordered rings, or
- elementary arithmetic, in a language with names for all Kalmar elementary functions

Plus the **induction scheme** for all Σ_k-formulas in the language.

Π_1 and Π_2 separations

We have the **separation** $\quad I\Sigma_k <_{\Pi_1} I\Sigma_{k+1}$

given by the Π_1 sentence $\mathrm{Con}(I\Sigma_k)$.

Π_1 and Π_2 separations

We have the **separation** $I\Sigma_k <_{\Pi_1} I\Sigma_{k+1}$

given by the Π_1 sentence $\mathrm{Con}(I\Sigma_k)$.

There are many interesting characterizations of Π_2 consequences. In particular the class $\mathcal{F}_k$ of **provably recursive** functions of $I\Sigma_k$ are those functions f for which

$$I\Sigma_k \vdash \forall x \exists! y \theta(x, y)$$

where θ is a Σ_1 formula for the graph of f.

We can show that $\mathcal{F}_{k+1}$ contains faster-growing functions than $\mathcal{F}_k$, and Π_2-separate the theories in this way.

Some complexity notation

- We identify numbers and the corresponding binary strings. We interpret a variable x as one or the other as convenient.
- We write $|x|$ for the binary length of x, approximately $\log_2(x)$. We often write $|n|^{O(1)}$ to mean "polynomial in $\log_2(n)$".
- FP is the class of polynomial time functions.
- The **polynomial hierarchy** consists of classes of relations P, Σ_1^p, Σ_2^p, etc. formed by putting bounded quantifiers in front of relations from P.
- We may write NP for Σ_1^p and coNP for Π_1^p.

Bounded arithmetic

As a **base theory** we take PV, which has

- a name for every FP function
- universal axioms fixing the basic relationships between them.

Bounded arithmetic

As a **base theory** we take PV, which has

- a name for every FP function
- universal axioms fixing the basic relationships between them.

(We could instead use Buss' theory BASIC, in the language $+$, $\cdot$, etc. With a bit of work, this gives the same hierarchy.)

Definition

- $T_2^k := \mathrm{PV} + \Sigma_k^b\text{-IND}$
- $T_2 := \bigcup_k T_2^k$ (this is equivalent to $I\Delta_0 + \Omega_1$)

Bounded arithmetic

As a **base theory** we take PV, which has

- a name for every FP function
- universal axioms fixing the basic relationships between them.

(We could instead use Buss' theory BASIC, in the language $+$, $\cdot$, etc. With a bit of work, this gives the same hierarchy.)

Definition

- $T_2^k := \mathrm{PV} + \Sigma_k^b\text{-IND}$
- $T_2 := \bigcup_k T_2^k$ (this is equivalent to $I\Delta_0 + \Omega_1$)

Here a Σ_k^b formula is one of the form

$$\exists y_1 < t_1(\bar{x})\ \forall y_2 < t_2(\bar{x}, y_1)\ \ \ldots\ \theta(\bar{x}, \bar{y})$$

where $t_1, \ldots, t_k$ are terms (FP functions) and θ is polynomial time.

Why study bounded arithmetic?

- General proof theoretic questions like - how little reasoning do we really need to prove something?
  (Such as the existence of infinitely many primes)

Why study bounded arithmetic?

- General proof theoretic questions like - how little reasoning do we really need to prove something?
  (Such as the existence of infinitely many primes)
- Natural framework to ask about the difficulty of doing complexity theory.
  Questions like: is $\mathsf{P} = \mathsf{NP}$, or $\mathsf{P} \neq \mathsf{NP}$, even consistent?

Why study bounded arithmetic?

- General proof theoretic questions like - how little reasoning do we really need to prove something?
  (Such as the existence of infinitely many primes)
- Natural framework to ask about the difficulty of doing complexity theory.
  Questions like: is $\mathsf{P} = \mathsf{NP}$, or $\mathsf{P} \neq \mathsf{NP}$, even consistent?
- By Buss' witnessing theorem, proofs have computational content – from a proof, we get an algorithm in the polynomial hierarchy

Why study bounded arithmetic?

- General proof theoretic questions like - how little reasoning do we really need to prove something?
  (Such as the existence of infinitely many primes)
- Natural framework to ask about the difficulty of doing complexity theory.
  Questions like: is $\mathsf{P} = \mathsf{NP}$, or $\mathsf{P} \neq \mathsf{NP}$, even consistent?
- By Buss' witnessing theorem, proofs have computational content – from a proof, we get an algorithm in the polynomial hierarchy
- Connections with propositional proof complexity

Back to our motivating question ...

Π_1 consequences

We would like to show: $T_2^k <_{\Pi_1} T_2^{k+1}$

Π_1 consequences

We would like to show: $T_2^k <_{\Pi_1} T_2^{k+1}$

Theorem [Paris-Wilkie]

$I\Delta_0+\exp$ does not prove $\mathrm{Con}(Q)$.

Π_1 consequences

We would like to show: $T_2^k <_{\Pi_1} T_2^{k+1}$

Theorem [Paris-Wilkie]

$I\Delta_0+\exp$ does not prove $\mathrm{Con}(Q)$.

Therefore in particular $T_2^{k+1} \nvdash Con(T_2^k)$.

So we cannot use consistency statements to separate our theories. Even bounded consistency cannot work. [Buss]

Unclear how to proceed.

Π_2 consequences

Question: Are there nice characterizations of the "provably recursive" functions $\mathcal{F}_k$ of T_2^k?

Π_2 consequences

Question: Are there nice characterizations of the "provably recursive" functions $\mathcal{F}_k$ of T_2^k?
Answer: Yes! Families of **TFNP search problems**.

Π_2 consequences

Question: Are there nice characterizations of the "provably recursive" functions $\mathcal{F}_k$ of T_2^k?
Answer: Yes! Families of **TFNP search problems**.

Question: Is $\mathcal{F}_{k+1}$ strictly bigger than $\mathcal{F}_k$, because it contains faster growing functions?

Π_2 consequences

Question: Are there nice characterizations of the "provably recursive" functions $\mathcal{F}_k$ of T_2^k?
Answer: Yes! Families of **TFNP search problems**.

Question: Is $\mathcal{F}_{k+1}$ strictly bigger than $\mathcal{F}_k$, because it contains faster growing functions?
Answer: No!

Π_2 consequences

Question: Are there nice characterizations of the "provably recursive" functions $\mathcal{F}_k$ of T_2^k?
Answer: Yes! Families of **TFNP search problems**.

Question: Is $\mathcal{F}_{k+1}$ strictly bigger than $\mathcal{F}_k$, because it contains faster growing functions?
Answer: No!

Parikh's theorem

Every provably total function with a bounded graph, in a bounded theory, is bounded.

In this case, for every k, if $f \in \mathcal{F}_k$ then $|f(x)|$ is polynomial in $|x|$.
We cannot separate Π_2 consequences by growth rates.

$\forall\Sigma_1^b$ consequences

Because of Parikh's theorem, instead of Π_2 consequences we consider the class $\forall\Sigma_1^b$, that is, sentences of the form

$$\forall x \exists y < t(x)\, \theta(x, y)$$

where $t \in \mathrm{FP}$ and θ is polynomial time.

We would like to show: $T_2^k <_{\forall\Sigma_1^b} T_2^{k+1}$

We would like to use the characterizations $\mathcal{F}_k$ of their provably recursive functions.

$\forall\Sigma_1^b$ consequences

Because of Parikh's theorem, instead of Π_2 consequences we consider the class $\forall\Sigma_1^b$, that is, sentences of the form

$$\forall x \exists y < t(x)\, \theta(x, y)$$

where $t \in \mathrm{FP}$ and θ is polynomial time.

We would like to show: $T_2^k <_{\forall\Sigma_1^b} T_2^{k+1}$

We would like to use the characterizations $\mathcal{F}_k$ of their provably recursive functions.

Problem

If $\mathrm{P} = \mathrm{NP}$, then $\mathcal{F}_k = \mathrm{FP}$ for all k, as for any true sentence of the form $\forall x \exists y < t\, \varphi(x, y)$, given x we can find a y in polynomial time.

Relativization

So to have a hope of showing separations, we consider **relativized** theories.

Relativization

So to have a hope of showing separations, we consider **relativized** theories.

We add a symbol α for an undefined relation, called an oracle.

We replace PV with PV(α), which talks about polynomial time machines with an oracle tape for α.

Relativization

So to have a hope of showing separations, we consider **relativized** theories.

We add a symbol α for an undefined relation, called an oracle.

We replace PV with PV(α), which talks about polynomial time machines with an oracle tape for α.

We use PV(α) instead of PV in all our definitions. So for example

$$T_2^k(\alpha) := \mathrm{PV}(\alpha) + \Sigma_k^b(\alpha)\text{-IND}.$$

Relativization

So to have a hope of showing separations, we consider **relativized** theories.

We add a symbol α for an undefined relation, called an oracle.

We replace PV with PV(α), which talks about polynomial time machines with an oracle tape for α.

We use PV(α) instead of PV in all our definitions. So for example

$$T_2^k(\alpha) := \mathrm{PV}(\alpha) + \Sigma_k^b(\alpha)\text{-IND}.$$

Everything in the rest of the talk is relativized.
For clarity we will not write the (α) in the names of theories etc.

Note that we are still studying induction, just in a slightly bigger, more flexible language.

Known relativized separations

Theorem

In the relativized setting, $T_2^k < T_2^{k+1}$

But the complexity of the separating sentence depends on k.
It is $\forall\Sigma_{k+1}^b$ [Buss-Krajíček]

Known relativized separations

Theorem

In the relativized setting, $T_2^k < T_2^{k+1}$

But the complexity of the separating sentence depends on k.
It is $\forall\Sigma^b_{k+1}$ [Buss-Krajíček]

For $\forall\Sigma^b_1$ consequences, we have

Theorem

In the relativized setting, $T_2^0 <_{\forall\Sigma^b_1} T_2^1 <_{\forall\Sigma^b_1} T_2^2$

Known relativized separations

Theorem

In the relativized setting, $T_2^k < T_2^{k+1}$

But the complexity of the separating sentence depends on k.
It is $\forall\Sigma^b_{k+1}$ [Buss-Krajíček]

For $\forall\Sigma^b_1$ consequences, we have

Theorem

In the relativized setting, $T_2^0 <_{\forall\Sigma^b_1} T_2^1 <_{\forall\Sigma^b_1} T_2^2$

No higher $\forall\Sigma^b_1$ separations are known. Conceivably $T_2^2 =_{\forall\Sigma^b_1} T_2$.

Examples of $\forall\Sigma_1^b$ separating principles

The following are provable in T_2^2 but not T_2^1:

Examples of $\forall\Sigma_1^b$ separating principles

The following are provable in T_2^2 but not T_2^1:

- **The injective weak pigeonhole principle iWPHP**
  For all n, if α determines a map f from $2n$ to n (by defining the bits of each $f(x)$), then there exist $x < x' < 2n$ such that $f(x) = f(x')$.

Examples of $\forall\Sigma_1^b$ separating principles

The following are provable in T_2^2 but not T_2^1:

- **The injective weak pigeonhole principle iWPHP**
  For all n, if α determines a map f from $2n$ to n (by defining the bits of each $f(x)$), then there exist $x < x' < 2n$ such that $f(x) = f(x')$.
- **The herbrandized ordering principle HOP**
  The ordering principle is a $\forall\Sigma_2^b$ sentence asserting that if α determines a total ordering on $[0, n)$, then it has a least element. HOP is a $\forall\Sigma_1^b$ version of this.

Examples of $\forall\Sigma_1^b$ separating principles

The following are provable in T_2^2 but not T_2^1:

- **The injective weak pigeonhole principle iWPHP**
  For all n, if α determines a map f from $2n$ to n (by defining the bits of each $f(x)$), then there exist $x < x' < 2n$ such that $f(x) = f(x')$.
- **The herbrandized ordering principle HOP**
  The ordering principle is a $\forall\Sigma_2^b$ sentence asserting that if α determines a total ordering on $[0, n)$, then it has a least element. HOP is a $\forall\Sigma_1^b$ version of this.

The following is provable in T_2^3 but not T_2^1:

- **The finite Ramsey theorem RAM**
  For all n, if α determines a graph on $[0, n)$, then it has a homogeneous set of size at least $\log n/2$.

Approximate counting

Theorem

The principles iWPHP, HOP and RAM are all provably in Jeřábek's theory APC_2 of **approximate counting**.

$$\mathrm{APC}_2 := T^1_2 + \mathrm{sWPHP}(\mathrm{FP}^{\mathrm{NP}}).$$

Here $\mathrm{sWPHP}(\mathrm{FP}^{\mathrm{NP}})$ is a scheme asserting that no $\mathrm{FP}^{\mathrm{NP}}$ function is a surjection from n to $2n$.

Approximate counting

Theorem

The principles iWPHP, HOP and RAM are all provably in Jeřábek's theory APC_2 of **approximate counting**.

$$\mathrm{APC}_2 := T^1_2 + \mathrm{sWPHP}(\mathrm{FP}^{\mathrm{NP}}).$$

Here $\mathrm{sWPHP}(\mathrm{FP}^{\mathrm{NP}})$ is a scheme asserting that no $\mathrm{FP}^{\mathrm{NP}}$ function is a surjection from n to $2n$.

In APC_2 we can write an expression for the approximate size of a bounded Σ^b_1 set, up to a multiplicative error, and use this expression in inductions.

In this way APC_2 can directly formalize many standard counting arguments in finite combinatorics.

Our question

It seems hard to show $T_2^2 <_{\forall\Sigma_1^b} T_2$.

Our question

It seems hard to show $T_2^2 <_{\forall\Sigma_1^b} T_2$. Instead we ask:

Question [Buss-Kołodziejczyk-T]

Is $\mathrm{APC}_2 <_{\forall\Sigma_1^b} T_2$?

Our question

It seems hard to show $T_2^2 <_{\forall\Sigma_1^b} T_2$. Instead we ask:

Question [Buss-Kołodziejczyk-T]

Is $\mathrm{APC}_2 <_{\forall\Sigma_1^b} T_2$?

Expected answer: YES

- This would push the boundary where can prove separations solidly up above T_2^1.
- We know $T_2^1 < \mathrm{APC}_2 \leq T_2^3$, so APC_2 is not that far from T_2^2 in strength.

Our question

It seems hard to show $T_2^2 <_{\forall\Sigma_1^b} T_2$. Instead we ask:

Question [Buss-Kołodziejczyk-T]

Is $\mathrm{APC}_2 <_{\forall\Sigma_1^b} T_2$?

Expected answer: YES

- This would push the boundary where can prove separations solidly up above T_2^1.
- We know $T_2^1 < \mathrm{APC}_2 \leq T_2^3$, so APC_2 is not that far from T_2^2 in strength.

Less expected answer: NO

- On the other hand, approximate counting is very useful; perhaps it can do everything.
- In fact, if we add a parity quantifier to the language, then T_2 collapses to APC_2 [Buss-Kołodziejczyk-Zdanowski].

Our result

Theorem

Yes, $\mathrm{APC}_2 <_{\forall\Sigma_1^b} T_2$.

Our result

Theorem

Yes, $\mathrm{APC}_2 <_{\forall\Sigma^b_1} T_2$.

We show that a certain $\forall\Sigma^b_1$ sentence CPLS, which **is** provable in T^2_2, **is not** provable in APC_2.

The principle CPLS is a kind of skolemized Σ^b_2-induction scheme.

The proof

Recall $\mathrm{APC}_2 := T^1_2 + \mathrm{sWPHP}(\mathrm{FP}^{\mathrm{NP}})$.

How do we show unprovability of a $\forall\Sigma^b_1$ sentence in APC_2?

The proof

Recall $\mathrm{APC}_2 := T^1_2 + \mathrm{sWPHP}(\mathrm{FP}^{\mathrm{NP}})$.

How do we show unprovability of a $\forall\Sigma^b_1$ sentence in APC_2?

By carefully constructing an oracle.

The proof

Recall $\mathrm{APC}_2 := T^1_2 + \mathrm{sWPHP}(\mathrm{FP}^{\mathrm{NP}})$.

How do we show unprovability of a $\forall\Sigma^b_1$ sentence in APC_2?

By carefully constructing an oracle.

We start with a simpler example:

Theorem [Krajicek]

$T^1_2 \nvdash \mathrm{iWPHP}$.

The Prover-Adversary Game

Let $\exists y < t\, \theta(n, y)$ be a Σ_1^b statement about an oracle α. For example, θ might say that y is a witness to iWPHP. Note that it only every requires $|n|^{O(1)}$ bits of α to make $\theta(n, y)$ true or false.

The Prover-Adversary Game

Let $\exists y < t\, \theta(n, y)$ be a Σ_1^b statement about an oracle α. For example, θ might say that y is a witness to iWPHP. Note that it only every requires $|n|^{O(1)}$ bits of α to make $\theta(n, y)$ true or false.

The **Adversary** claims to know an oracle α for which there is no witness $y < t$ such that $\theta(n, y)$ is true in α.

The Prover-Adversary Game

Let $\exists y < t\, \theta(n, y)$ be a Σ_1^b statement about an oracle α. For example, θ might say that y is a witness to iWPHP. Note that it only every requires $|n|^{O(1)}$ bits of α to make $\theta(n, y)$ true or false.

The **Adversary** claims to know an oracle α for which there is no witness $y < t$ such that $\theta(n, y)$ is true in α.

At each turn the **Prover** may ask the Adversary for the value of bit of α; or he may forget a bit, to save memory. If so, the Adversary can give the bit a different values next time it is asked.

The Prover-Adversary Game

Let $\exists y < t\, \theta(n, y)$ be a Σ_1^b statement about an oracle α. For example, θ might say that y is a witness to iWPHP. Note that it only every requires $|n|^{O(1)}$ bits of α to make $\theta(n, y)$ true or false.

The **Adversary** claims to know an oracle α for which there is no witness $y < t$ such that $\theta(n, y)$ is true in α.

At each turn the **Prover** may ask the Adversary for the value of bit of α; or he may forget a bit, to save memory. If so, the Adversary can give the bit a different values next time it is asked.

Prover wins when his memory β makes $\theta(n, y)$ true for some $y < t$.

The Prover-Adversary Game

Let $\exists y < t\, \theta(n, y)$ be a Σ_1^b statement about an oracle α. For example, θ might say that y is a witness to iWPHP. Note that it only every requires $|n|^{O(1)}$ bits of α to make $\theta(n, y)$ true or false.

The **Adversary** claims to know an oracle α for which there is no witness $y < t$ such that $\theta(n, y)$ is true in α.

At each turn the **Prover** may ask the Adversary for the value of bit of α; or he may forget a bit, to save memory. If so, the Adversary can give the bit a different values next time it is asked.

Prover wins when his memory β makes $\theta(n, y)$ true for some $y < t$.

Theorem [Buss-Krajíček]

If $T_2^1 \vdash \forall n \exists y < t\, \theta(n, y)$, then there is a winning strategy for the Prover in which he never needs to remember more than $|n|^{O(1)}$ bits.

The Prover-Adversary Game for iWPHP

The principle iWPHP is about an oracle α describing a function mapping "pigeons" $< 2n$ to "holes" $< n$.

The Prover-Adversary Game for iWPHP

The principle iWPHP is about an oracle α describing a function mapping "pigeons" $< 2n$ to "holes" $< n$.

So to show $T_2^1 \nvdash$ iWPHP it is enough to give a strategy for the Adversary which works against a Prover who can remember facts about only a very small number of pigeons.

The Prover-Adversary Game for iWPHP

The principle iWPHP is about an oracle α describing a function mapping "pigeons" $< 2n$ to "holes" $< n$.

So to show $T^1_2 \nvdash$ iWPHP it is enough to give a strategy for the Adversary which works against a Prover who can remember facts about only a very small number of pigeons.

But this is easy: the Adversary maintains a partial injection ρ of pigeons to holes which extends the Prover's current memory, and doesn't set any pigeons the Prover doesn't know about.

The Prover-Adversary Game for iWPHP

The principle iWPHP is about an oracle α describing a function mapping "pigeons" $< 2n$ to "holes" $< n$.

So to show $T^1_2 \nvdash$ iWPHP it is enough to give a strategy for the Adversary which works against a Prover who can remember facts about only a very small number of pigeons.

But this is easy: the Adversary maintains a partial injection ρ of pigeons to holes which extends the Prover's current memory, and doesn't set any pigeons the Prover doesn't know about.

Since the the size of $\mathrm{dom}(\rho)$ is always much smaller than n, when the Prover asks about a new pigeon it is easy to find a hole to map it to.

Back to CPLS

We can use a similar argument to show $T_2^1 \not\vdash \mathrm{CPLS}$.

Back to CPLS

We can use a similar argument to show $T_2^1 \not\vdash \text{CPLS}$.

CPLS has a similar size parameter n. We fix n.

Back to CPLS

We can use a similar argument to show $T_2^1 \not\vdash$ CPLS.

CPLS has a similar size parameter n. We fix n.

In the Adversary strategy, instead of using partial injections ρ we define a family H of **legal restrictions** ρ satisfying

- Each $\rho \in H$ is a partially-defined oracle which does not contain a witness to CPLS
- . . . some other nice properties.

Back to CPLS

We can use a similar argument to show $T_2^1 \nvdash$ CPLS.

CPLS has a similar size parameter n. We fix n.

In the Adversary strategy, instead of using partial injections ρ we define a family H of **legal restrictions** ρ satisfying

- Each $\rho \in H$ is a partially-defined oracle which does not contain a witness to CPLS
- . . . some other nice properties.

We can find a winning strategy for the Adversary in which she always maintains $\rho \in H$ extending the Prover's current memory.

Proof idea

We want to show $T_2^1 + \mathrm{sWPHP}(\mathrm{FP}^{\mathrm{NP}}) \nvdash \mathrm{CPLS}$.

We know how to deal with T_2^1. How about sWPHP?

Proof idea

We want to show $T_2^1 + \mathrm{sWPHP}(\mathrm{FP}^{\mathrm{NP}}) \nvdash \mathrm{CPLS}$.

We know how to deal with T_2^1. How about sWPHP?

Simplification. By some logical tricks, we can replace $\mathrm{sWPHP}(\mathrm{FP}^{\mathrm{NP}})$ with a formula $\mathrm{iWPHP}(F, n)$ asserting

$$F \text{ is not an injection } 2n \to n$$

for some $F \in \mathrm{FP}^{\mathrm{NP}}$. (This is not quite true, but close.)

Proof idea

We want to show $T_2^1 + \mathrm{sWPHP}(\mathrm{FP}^{\mathrm{NP}}) \nvdash \mathrm{CPLS}$.

We know how to deal with T_2^1. How about sWPHP?

Simplification. By some logical tricks, we can replace $\mathrm{sWPHP}(\mathrm{FP}^{\mathrm{NP}})$ with a formula $\mathrm{iWPHP}(F, n)$ asserting

$$F \text{ is not an injection } 2n \to n$$

for some $F \in \mathrm{FP}^{\mathrm{NP}}$. (This is not quite true, but close.)

Goal now

Find a legal restriction ρ which "forces" $\mathrm{iWPHP}(F, n)$ to be true. Then do the Prover-Adversary argument where the Adversary now uses only legal restrictions extending ρ.

Key lemma 1

Definition

An NP query $\exists z < t\, \theta(z)$ is **fixed** by a legal restriction ρ if either

- $\theta(z)$ is defined and true in ρ for some $z < t$, or
- $\theta(z)$ is not defined and true in any legal $\sigma \supseteq \rho$, for any $z < t$.

We say it is fixed respectively to YES or NO.

Key lemma 1

Definition

An NP query $\exists z < t\, \theta(z)$ is **fixed** by a legal restriction ρ if either

- $\theta(z)$ is defined and true in ρ for some $z < t$, or
- $\theta(z)$ is not defined and true in any legal $\sigma \supseteq \rho$, for any $z < t$.

We say it is fixed respectively to YES or NO.

[Pudlak-T] define a distribution $\mathcal{R}$ over legal restrictions (for CPLS) satisfying the following.

Fixing Lemma

For any NP query Q, $\Pr_{\rho \in \mathcal{R}}[\rho \text{ does not fix } Q\ ] < n^{-1/7}$.

Key lemma 1

Definition

An NP query $\exists z < t\, \theta(z)$ is **fixed** by a legal restriction ρ if either

- $\theta(z)$ is defined and true in ρ for some $z < t$, or
- $\theta(z)$ is not defined and true in any legal $\sigma \supseteq \rho$, for any $z < t$.

We say it is fixed respectively to YES or NO.

[Pudlak-T] define a distribution $\mathcal{R}$ over legal restrictions (for CPLS) satisfying the following.

Fixing Lemma

For any NP query Q, $\Pr_{\rho \in \mathcal{R}}[\rho \text{ does not fix } Q\ ] < n^{-1/7}$.

This is a weaker, but more widely applicable, version of Hastad's switching lemma for DNFs.

Key lemma 2

Recall that we are dealing with $\mathrm{FP}^{\mathrm{NP}}$ computations,

These run for $|n|^{O(1)}$ steps. At each step, they make an NP query and get a YES/NO answer.

Key lemma 2

Recall that we are dealing with $\mathrm{FP}^{\mathrm{NP}}$ computations,

These run for $|n|^{O(1)}$ steps. At each step, they make an NP query and get a YES/NO answer.

Definition

A computation w of such a machine is **fixed** by a legal restriction ρ if every query in w is fixed by ρ (to the answer given in w).

Key lemma 2

Recall that we are dealing with $\mathrm{FP}^{\mathrm{NP}}$ computations,

These run for $|n|^{O(1)}$ steps. At each step, they make an NP query and get a YES/NO answer.

Definition

A computation w of such a machine is **fixed** by a legal restriction ρ if every query in w is fixed by ρ (to the answer given in w).

In [Kołodziejczyk-T] we extend the fixing lemma slightly to show, for the same distribution $\mathcal{R}$:

Fixing Lemma for Computations

For any such $\mathrm{FP}^{\mathrm{NP}}$ machine query M,

$\Pr_{\rho \in \mathcal{R}}[\rho$ does not fix some computation of $M\ ] < n^{-1/6}$.

Forcing a collision

Recall that we have a single function $F : 2n \to n$ computed by a machine $M \in \mathrm{FP}^{\mathrm{NP}}$. We want to "force" iWPHP(F, n) to true.

Forcing a collision

Recall that we have a single function $F : 2n \to n$ computed by a machine $M \in \mathrm{FP}^{\mathrm{NP}}$. We want to "force" iWPHP(F, n) to true.

Consider $2n$ copies of M, running on inputs $x = 0, 1, \ldots, 2n - 1$.

By the fixing lemma for computations and an averaging argument, if we take a random $\rho \in \mathcal{R}$, then with high probability ρ fixes a computation of M simultaneously for at least $2/3$ of these inputs.

Forcing a collision

Recall that we have a single function $F : 2n \to n$ computed by a machine $M \in \mathrm{FP}^{\mathrm{NP}}$. We want to "force" iWPHP(F, n) to true.

Consider $2n$ copies of M, running on inputs $x = 0, 1, \ldots, 2n - 1$.

By the fixing lemma for computations and an averaging argument, if we take a random $\rho \in \mathcal{R}$, then with high probability ρ fixes a computation of M simultaneously for at least $2/3$ of these inputs.

That is, **as long as we only work with legal restrictions extending** ρ, there are $x_1, \ldots, x_{4n/3} < 2n$ such that we can treat each $F(x_i)$ as though it takes a fixed value $y_i < n$.

By the standard pigeonhole principle, we can find $x_i \neq x_j$ such that $y_i = y_j$. This is our "forced" collision.

Proof summary

We want to show "$T_2^1 + \mathrm{iWPHP}(F, n) \nvdash \mathrm{CPLS}$".

Proof summary

We want to show "$T_2^1 + \text{iWPHP}(F, n) \nvdash \text{CPLS}$".

We use the Prover-Adversary game. We fix suitable n, x_i, x_j, y.

The Adversary claims to know an oracle α such that in α

- CPLS is false
- $F(x_i) = y$ and $F(x_j) = y$.

Proof summary

We want to show "$T_2^1 + \text{iWPHP}(F, n) \nvdash \text{CPLS}$".

We use the Prover-Adversary game. We fix suitable n, x_i, x_j, y.

The Adversary claims to know an oracle α such that in α

- CPLS is false
- $F(x_i) = y$ and $F(x_j) = y$.

These are both coNP claims. (In the second case, the coNP assertion is that the NO replies in the computations are correct.)

Proof summary

We want to show "$T_2^1 + \text{iWPHP}(F, n) \nvdash \text{CPLS}$".

We use the Prover-Adversary game. We fix suitable n, x_i, x_j, y.

The Adversary claims to know an oracle α such that in α

- CPLS is false
- $F(x_i) = y$ and $F(x_j) = y$.

These are both coNP claims. (In the second case, the coNP assertion is that the NO replies in the computations are correct.)

By the definition of legal restrictions and the choice of ρ, the Adversary has a strategy which sticks to legal restrictions extending ρ and which guarantees that the Prover is never able to witness that either claim is false.

It follows that there is no such proof. □