Approximate counting and NP search

Neil Thapen

Institute of Mathematics
Czech Academy of Sciences

Joint work with Leszek Kołodziejczyk

Outline

TFNP

Proofs and search problems

Approximate counting

Our result

NP search problems

An *NP search problem* is specified by

- a polynomial time relation $R(x, y)$ and a polynomial p

such that

- for every x there is a y such that $R(x, y)$ and $|y| < p(|x|)$.

NP search problems

An *NP search problem* is specified by

- a polynomial time relation $R(x, y)$ and a polynomial p

such that

- for every x there is a y such that $R(x, y)$ and $|y| < p(|x|)$.

It represents the computational task:

- given x, find y such that $R(x, y)$ and $|y| < p(|x|)$.

Examples

Usually we use the input x as a parameter describing a large object, rather than a code for a small object.

Examples

Usually we use the input x as a parameter describing a large object, rather than a code for a small object.

Pigeonhole principle (PPP)

We are given a parametrized family of polynomial time functions f_x such that $f_x < x$.
Input: x
Search task: find distinct $y, y' < x + 1$ such that $f_x(y) = f_x(y')$.

Examples

Usually we use the input x as a parameter describing a large object, rather than a code for a small object.

Pigeonhole principle (PPP)

We are given a parametrized family of polynomial time functions f_x such that $f_x < x$.
Input: x
Search task: find distinct $y, y' < x + 1$ such that $f_x(y) = f_x(y')$.

Weak pigeonhole principle (WPHP)

The setup is the same as for PPP.
Input: x
Search task: find distinct $y, y' < 2x$ such that $f_x(y) = f_x(y')$.

Examples

Finite Ramsey theorem (RAMSEY)

We are given a parametrized family of graphs G_n with vertices $[0, n)$, polynomial-time structure, and edges coloured red or blue.

Input: n

Search task: find vertices $v_1, \ldots, v_{(\log n)/2} \in [0, n)^{(\log n)/2}$ forming a monochrome clique.

Reducibility

Definition

Given search problems $R(x, y)$ and $Q(x', y')$, we say

R is *polynomial-time many-one reducible* to Q if there are polynomial time functions $f(x)$ and $g(x, y')$ such that

$$\forall x, y', \ Q(f(x), y') \to R(x, g(x, y')).$$

We write $R \leq Q$.

Reducibility

Definition

Given search problems $R(x, y)$ and $Q(x', y')$, we say

R is *polynomial-time many-one reducible* to Q if there are polynomial time functions $f(x)$ and $g(x, y')$ such that

$$\forall x, y', \ Q(f(x), y') \rightarrow R(x, g(x, y')).$$

We write $R \leq Q$.

If reducibility holds in both directions then R and Q are equivalent, written $R \equiv Q$.

TFNP

The name TFNP stands for *total functional NP*.

It is the class of all NP search problems.

TFNP

The name TFNP stands for *total functional NP*.

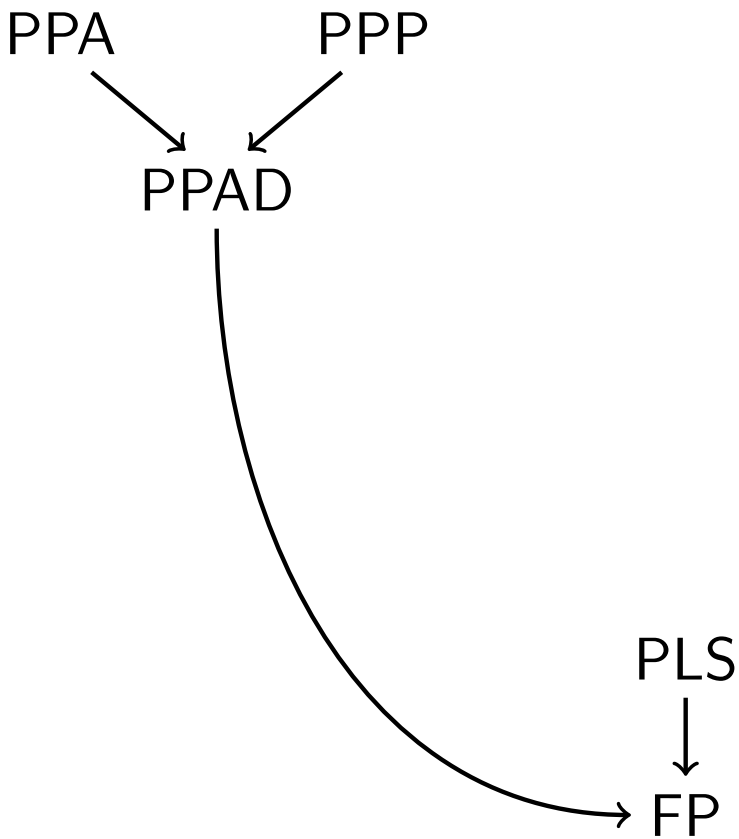
It is the class of all NP search problems.

If P = NP then every problem in TFNP can be solved in P.

But in fact we think there is a rich structure inside TFNP.

[Johnson, Papadimitriou, Yannakakis 88],
[Megiddo, Papadimitriou 91], [Papadimitriou 94]

TFNP



TFNP

Proofs and search problems

Approximate counting

Our result

A simple bounded arithmetic

Take $\mathcal{L}$ be a first-order language consisting of a name for every polynomial time function and relation.

Variables range over binary strings, which we identify with natural numbers.

A simple bounded arithmetic

Take $\mathcal{L}$ be a first-order language consisting of a name for every polynomial time function and relation.

Variables range over binary strings, which we identify with natural numbers.

Let BASE be the theory consisting of all true sentences of the form

$$\forall \bar{x} \varphi(\bar{x})$$

where φ is a quantifier-free $\mathcal{L}$-formula.

A simple bounded arithmetic

Take $\mathcal{L}$ be a first-order language consisting of a name for every polynomial time function and relation.

Variables range over binary strings, which we identify with natural numbers.

Let BASE be the theory consisting of all true sentences of the form

$$\forall \bar{x} \varphi(\bar{x})$$

where φ is a quantifier-free $\mathcal{L}$-formula.

All our theories will contain BASE. It follows that our theories will prove that all many-one reductions behave as they are supposed to.

A simple bounded arithmetic

Definition

A Σ_k^b formula is one of the form

$$\exists x_1 < t_1(\bar{z})\ \forall x_2 < t_2(x_1, \bar{z})\ \ \ldots\ \ \varphi(x_1, \ldots, x_k, \bar{z})$$

where there are k quantifier alternations,

$t_1, \ldots, t_k$ are $\mathcal{L}$-terms and φ is a quantifier-free $\mathcal{L}$-formula.

A simple bounded arithmetic

Definition

A Σ_k^b formula is one of the form

$$\exists x_1 < t_1(\bar{z})\ \forall x_2 < t_2(x_1, \bar{z})\ \ \ldots\ \ \varphi(x_1, \ldots, x_k, \bar{z})$$

where there are k quantifier alternations,

$t_1, \ldots, t_k$ are $\mathcal{L}$-terms and φ is a quantifier-free $\mathcal{L}$-formula.

Σ_1^b formulas express precisely the relations in NP.

Σ_k^b formulas express precisely the relations in Σ_k^p, the k-th level of the polynomial hierarchy.

A family of theories

Definition

Σ^b_k-IND is the theory consisting of BASE plus the induction scheme

$$\varphi(0) \wedge \forall x(\varphi(x) \to \varphi(x+1)) \to \forall x \varphi(x)$$

for all Σ^b_k formulas φ.

A family of theories

Definition

Σ^b_k-IND is the theory consisting of BASE plus the induction scheme

$$\varphi(0) \wedge \forall x(\varphi(x) \to \varphi(x+1)) \to \forall x \varphi(x)$$

for all Σ^b_k formulas φ.

Theorem [Buss 85]

Σ^b_k-IND proves:

"Every $P^{\Sigma^p_k}$ machine has a computation on every input."

For our purposes, Σ^b_k-IND is equivalent to this statement.

Search problems and theories

We can identify search problems in TFNP with true $\forall\Sigma^b_1$ sentences.

That is, true sentences of the form $\forall x \exists y < t(x)\, \varphi(x, y)$.

Search problems and theories

We can identify search problems in TFNP with true $\forall\Sigma^b_1$ sentences. That is, true sentences of the form $\forall x \exists y < t(x)\, \varphi(x, y)$.

The search problem is provably total in a theory T if T proves this sentence.

Write $\forall\Sigma^b_1(T)$ for the set of search problems provably total in T.

Relativized search problems

We now add an oracle tape to our machines and a symbol for an arbitrary oracle to our language.

Everything works as before, but we can now try to find oracles with respect to which we can show non-reducibility between search problems.

An old open problem

Problem

Is $\forall\Sigma^b_1(\Sigma^b_k\text{-IND})$ strictly bigger than $\forall\Sigma^b_1(\Sigma^b_j\text{-IND})$ if $k > j$?

This is open for $k \geq 2$.

An old open problem

Problem

Is $\forall\Sigma^b_1(\Sigma^b_k\text{-IND})$ strictly bigger than $\forall\Sigma^b_1(\Sigma^b_j\text{-IND})$ if $k > j$?

This is open for $k \geq 2$.

Put differently – do more TFNP problems arise from the axiom

every $P^{\Sigma^p_k}$ machine has a computation

than from the axiom

every $P^{\Sigma^p_j}$ machine has a computation?

The game induction principle

The *k-turn game induction principle* GI_k is an NP search problem.

It states a property of a sequence $G_0, \ldots, G_{x-1}$ of k-turn games.

The game induction principle

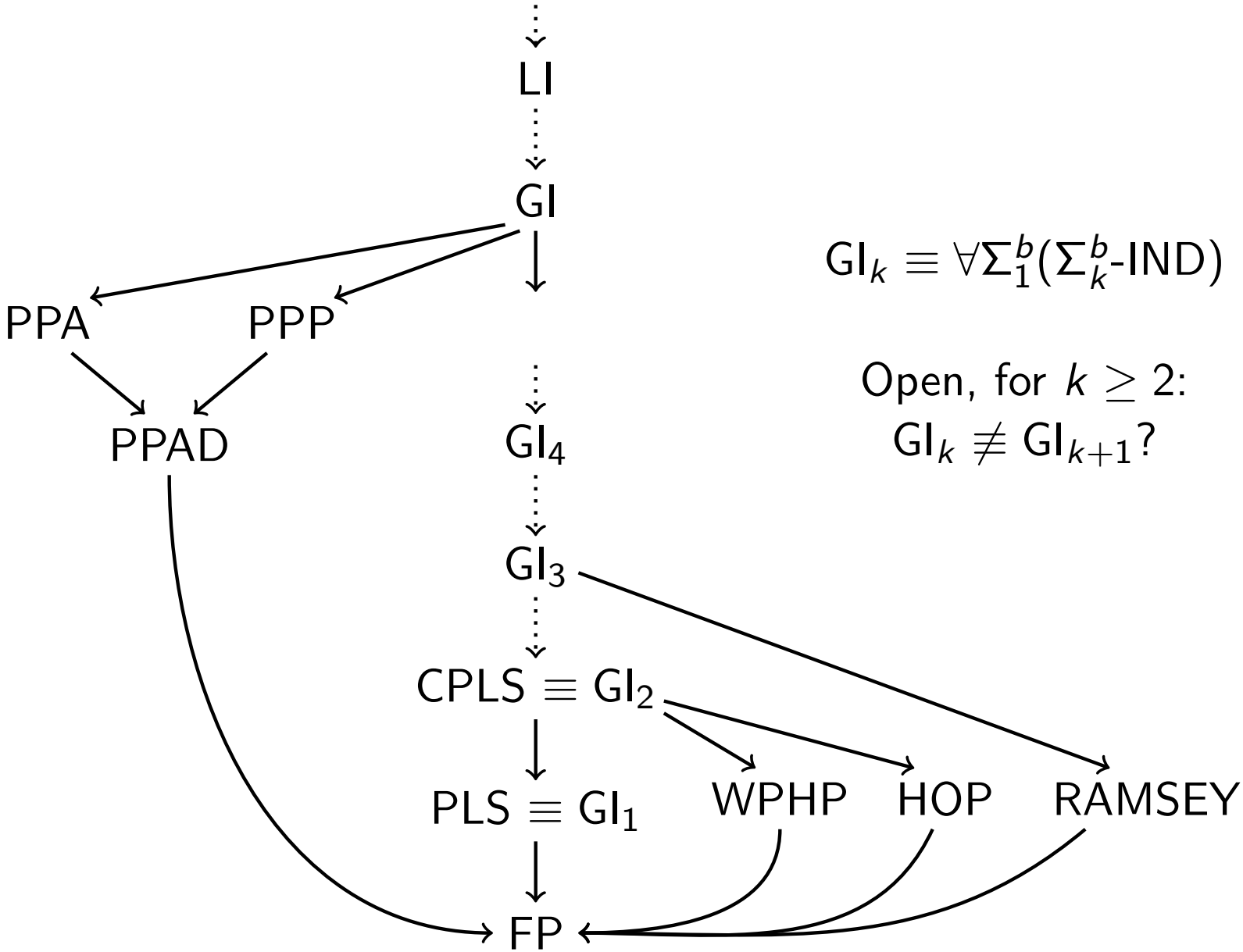
The *k-turn game induction principle* GI_k is an NP search problem.

It states a property of a sequence $G_0, \ldots, G_{x-1}$ of k-turn games.

Theorem [Skelley, NT 11]

For all k, $\mathsf{GI}_k \equiv \forall\Sigma^b_1(\Sigma^b_k\text{-IND})$.

TFNP



$GI_k \equiv \forall\Sigma_1^b(\Sigma_k^b\text{-IND})$

Open, for $k \geq 2$:
$GI_k \not\equiv GI_{k+1}$?

TFNP

Proofs and search problems

Approximate counting

Our result

Approximate counting

[Jeřábek 08, 09] introduced a theory that can carry out combinatorial arguments using approximate counting.

It is based on a formalization of Sipser's approximate counting with linear hash functions.

Approximate counting

[Jeřábek 08, 09] introduced a theory that can carry out combinatorial arguments using approximate counting.

It is based on a formalization of Sipser's approximate counting with linear hash functions.

We define it as

$$\mathrm{APC}_2 := \Sigma^b_1\text{-IND} + \mathrm{sWPHP}(P^{NP})$$

where $\mathrm{sWPHP}(P^{NP})$ asserts:

no P^{NP} function is a surjection $x \to 2x$, for any $x > 1$.

Approximate counting

[Jeřábek 08, 09] introduced a theory that can carry out combinatorial arguments using approximate counting.

It is based on a formalization of Sipser's approximate counting with linear hash functions.

We define it as

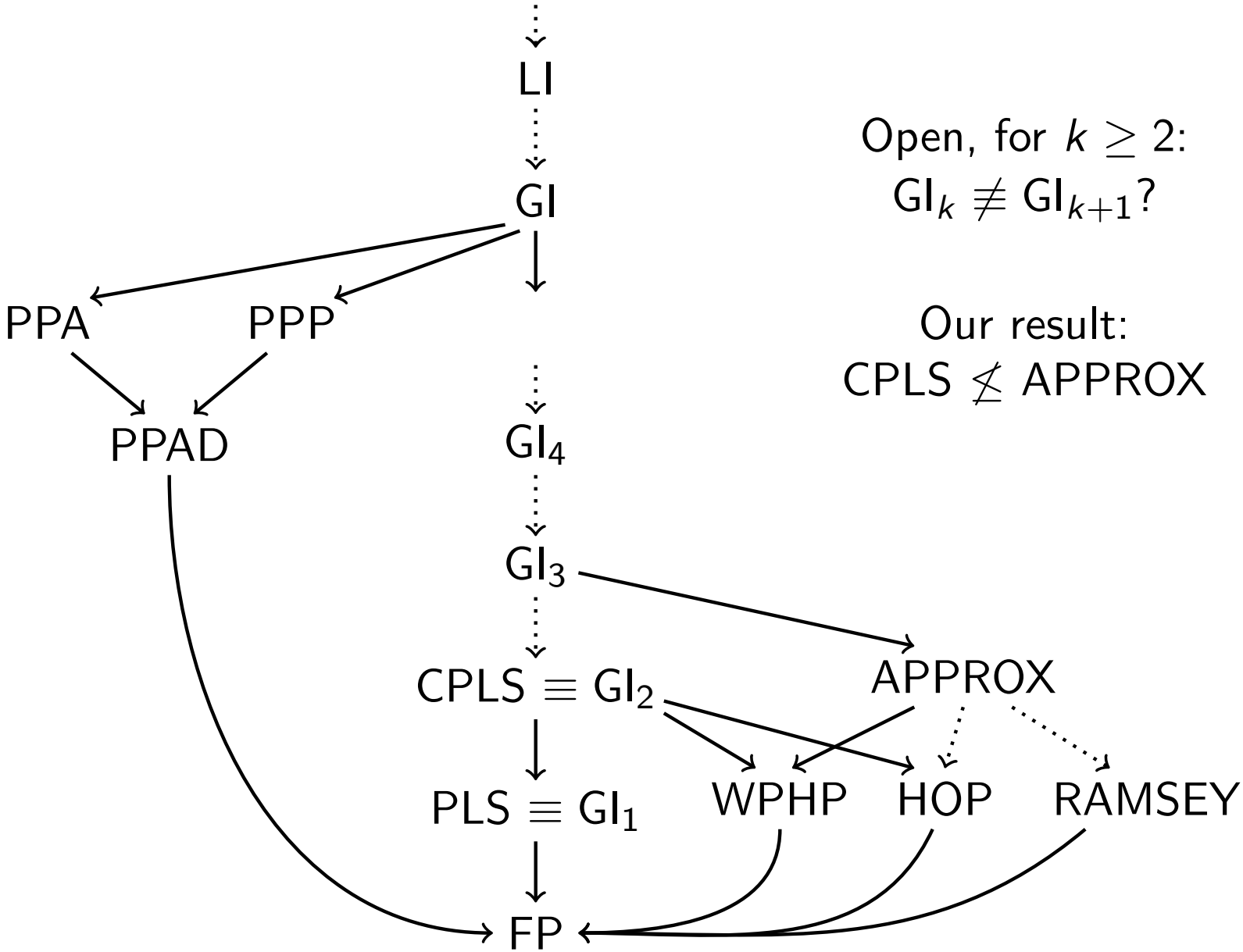
$$\mathrm{APC}_2 := \Sigma^b_1\text{-IND} + \mathrm{sWPHP}(P^{NP})$$

where $\mathrm{sWPHP}(P^{NP})$ asserts:

> no P^{NP} function is a surjection $x \to 2x$, for any $x > 1$.

We write APPROX for $\forall\Sigma^b_1(\mathrm{APC}_2)$.

TFNP



Open, for $k \geq 2$:
$GI_k \not\equiv GI_{k+1}$?

Our result:
$CPLS \not\leq APPROX$

Ramsey theorem by precise counting (1)

G is the complete graph on $[0, n)$, with edges coloured red or blue.

Ramsey theorem by precise counting (1)

G is the complete graph on $[0, n)$, with edges coloured red or blue.

Choose an initial vertex $v_1 = 0$. Either at least half the edges touching v_1 are red, or at least half are blue.

Suppose more are red. Let $S(v_1, 0) = \{u \in [0, n) : (v_1, u) \text{ is red}\}$. We have $|S(v_1, 0)| \geq n/2$.

Ramsey theorem by precise counting (1)

G is the complete graph on $[0, n)$, with edges coloured red or blue.

Choose an initial vertex $v_1 = 0$. Either at least half the edges touching v_1 are red, or at least half are blue.

Suppose more are red. Let $S(v_1, 0) = \{u \in [0, n) : (v_1, u) \text{ is red}\}$.
We have $|S(v_1, 0)| \geq n/2$.

Let v_2 be a vertex in $S(v_1, 0)$. Over vertices $u \in S(v_1, 0)$, either at least half the edges (v_2, u) are red, or at least half are blue.

Suppose more are blue.
Let $S(v_1, v_2, 0, 1) = \{u \in S(v_1, 0) : (v_2, u) \text{ is blue}\}$.
Then $|S(v_1, v_2, 0, 1)| \geq n/4$.

...

Ramsey theorem by precise counting (2)

For $v_1, \ldots, v_k \in [0, n)^k$ and $\delta_1, \ldots, \delta_k \in \{0, 1\}^k$ define

$$S(\bar{v}, \bar{\delta}) := \{u \in [0, n) : \forall i \leq k, \quad \delta_i = 0 \text{ and } (v_i, u) \text{ is red} \\ \text{or } \delta_i = 1 \text{ and } (v_i, u) \text{ is blue}\}.$$

We inductively construct $v_1, \ldots, v_k$ and $\delta_1, \ldots, \delta_k$ such that

1. For all $i < k$, $v_{i+1} \in S(v_1, \ldots, v_i, \delta_1, \ldots, \delta_i)$
2. $|S(v_1, \ldots, v_k, \delta_1, \ldots, \delta_k)| \geq n/2^k$.

Ramsey theorem by precise counting (2)

For $v_1, \ldots, v_k \in [0, n)^k$ and $\delta_1, \ldots, \delta_k \in \{0, 1\}^k$ define

$$S(\bar{v}, \bar{\delta}) := \{u \in [0, n) : \forall i \leq k, \quad \delta_i = 0 \text{ and } (v_i, u) \text{ is red} \\ \text{or } \delta_i = 1 \text{ and } (v_i, u) \text{ is blue}\}.$$

We inductively construct $v_1, \ldots, v_k$ and $\delta_1, \ldots, \delta_k$ such that

1. For all $i < k$, $v_{i+1} \in S(v_1, \ldots, v_i, \delta_1, \ldots, \delta_i)$
2. $|S(v_1, \ldots, v_k, \delta_1, \ldots, \delta_k)| \geq n/2^k$.

By item 2, we can do this up to $k = \log n$.

Ramsey theorem by precise counting (2)

For $v_1, \ldots, v_k \in [0, n)^k$ and $\delta_1, \ldots, \delta_k \in \{0, 1\}^k$ define

$$S(\bar{v}, \bar{\delta}) := \{u \in [0, n) : \forall i \leq k, \quad \delta_i = 0 \text{ and } (v_i, u) \text{ is red} \\ \text{or } \delta_i = 1 \text{ and } (v_i, u) \text{ is blue}\}.$$

We inductively construct $v_1, \ldots, v_k$ and $\delta_1, \ldots, \delta_k$ such that

1. For all $i < k$, $v_{i+1} \in S(v_1, \ldots, v_i, \delta_1, \ldots, \delta_i)$
2. $|S(v_1, \ldots, v_k, \delta_1, \ldots, \delta_k)| \geq n/2^k$.

By item 2, we can do this up to $k = \log n$.

If half the bits of $\delta_1, \ldots, \delta_k$ are 0, then by item 1, $v_1, \ldots, v_k$ contains a red $k/2$-clique.

Otherwise half the bits of $\delta_1, \ldots, \delta_k$ are 1, so by item 1, $v_1, \ldots, v_k$ contains a blue $k/2$-clique.

Ramsey theorem by approximate counting (1)

Suppose we can *approximately* count large, polynomial time sets.

We can write $|S| \succeq_\epsilon a$ to mean "the size of S is at least a with some small multiplicative error".

Ramsey theorem by approximate counting (1)

Suppose we can *approximately* count large, polynomial time sets.

We can write $|S| \succeq_\epsilon a$ to mean "the size of S is at least a with some small multiplicative error".

Choose an initial vertex $v_1 = 0$. Either at least one third of the edges touching v_1 are red, or at least one third are blue.

Suppose one third are red.
Let $S(v_1, 0) = \{u \in [0, n) : (v_1, u) \text{ is red}\}$.
We have $|S(v_1, 0)| \succeq_\epsilon n/3$.

Let v_2 be a vertex in $S(v_1, 0)$. Over vertices $u \in S(v_1, 0)$, either at least one third of the edges (v_2, u) are red, or at least one third are blue.

...

Ramsey theorem by approximate counting (2)

For $v_1, \ldots, v_k \in [0, n)^k$ and $\delta_1, \ldots, \delta_k \in \{0, 1\}^k$ define

$$S(\bar{v}, \bar{\delta}) := \{u \in [0, n) : \forall i \leq k, \quad \delta_i = 0 \text{ and } (v_i, u) \text{ is red} \\ \text{or } \delta_i = 1 \text{ and } (v_i, u) \text{ is blue}\}.$$

We inductively construct $v_1, \ldots, v_k$ and $\delta_1, \ldots, \delta_k$ such that

1. For all $i < k$, $v_{i+1} \in S(v_1, \ldots, v_i, \delta_1, \ldots, \delta_i)$
2. $|S(v_1, \ldots, v_k, \delta_1, \ldots, \delta_k)| \succeq_\epsilon n/3^k$.

By item 2, we can do this up to $k = \log_4 n$.

Ramsey theorem by approximate counting (2)

For $v_1, \ldots, v_k \in [0, n)^k$ and $\delta_1, \ldots, \delta_k \in \{0, 1\}^k$ define

$$S(\bar{v}, \bar{\delta}) := \{u \in [0, n) : \forall i \leq k, \quad \delta_i = 0 \text{ and } (v_i, u) \text{ is red} \\ \text{or } \delta_i = 1 \text{ and } (v_i, u) \text{ is blue}\}.$$

We inductively construct $v_1, \ldots, v_k$ and $\delta_1, \ldots, \delta_k$ such that

1. For all $i < k$, $v_{i+1} \in S(v_1, \ldots, v_i, \delta_1, \ldots, \delta_i)$
2. $|S(v_1, \ldots, v_k, \delta_1, \ldots, \delta_k)| \succeq_\epsilon n/3^k$.

By item 2, we can do this up to $k = \log_4 n$.

As before, we can now find a monochrome $k/2$-clique in $v_1, \ldots, v_k$, because we can count very small sets precisely.

Ramsey theorem by approximate counting (2)

For $v_1, \ldots, v_k \in [0, n)^k$ and $\delta_1, \ldots, \delta_k \in \{0, 1\}^k$ define

$$S(\bar{v}, \bar{\delta}) := \{u \in [0, n) : \forall i \leq k, \quad \delta_i = 0 \text{ and } (v_i, u) \text{ is red} \\ \text{or } \delta_i = 1 \text{ and } (v_i, u) \text{ is blue}\}.$$

We inductively construct $v_1, \ldots, v_k$ and $\delta_1, \ldots, \delta_k$ such that

1. For all $i < k$, $v_{i+1} \in S(v_1, \ldots, v_i, \delta_1, \ldots, \delta_i)$
2. $|S(v_1, \ldots, v_k, \delta_1, \ldots, \delta_k)| \succeq_\epsilon n/3^k$.

By item 2, we can do this up to $k = \log_4 n$.

As before, we can now find a monochrome $k/2$-clique in $v_1, \ldots, v_k$, because we can count very small sets precisely.

(In fact by a careful argument of this kind with smaller errors, we can still get a $(\log_2 n)/2$-clique.)

Ramsey theorem by approximate counting (3)

Can we do this argument in $\mathrm{APC}_2 = \Sigma^b_1\text{-IND}+\text{sWPHP}(P^{NP})$?

Ramsey theorem by approximate counting (3)

Can we do this argument in $\mathrm{APC}_2 = \Sigma^b_1$-IND+sWPHP($P^{NP}$)?

By induction on k up to roughly $\log n$, we showed that:

There exist sequences $v_1, \ldots, v_k$ and $\delta_1, \ldots, \delta_k$ such that

1. For all $i < k$, $v_{i+1} \in S(v_1, \ldots, v_i, \delta_1, \ldots, \delta_i)$
2. $|S(v_1, \ldots, v_k, \delta_1, \ldots, \delta_k)| \succeq_\epsilon n/3^k$.

Membership in S is polynomial time. By the precise definition of the counting relation $\succeq_\epsilon$, item 2 is Σ^b_2.

Ramsey theorem by approximate counting (3)

Can we do this argument in $\mathrm{APC}_2 = \Sigma^b_1$-IND+sWPHP($P^{NP}$)?

By induction on k up to roughly $\log n$, we showed that:

There exist sequences $v_1, \ldots, v_k$ and $\delta_1, \ldots, \delta_k$ such that

1. For all $i < k$, $v_{i+1} \in S(v_1, \ldots, v_i, \delta_1, \ldots, \delta_i)$
2. $|S(v_1, \ldots, v_k, \delta_1, \ldots, \delta_k)| \succeq_\epsilon n/3^k$.

Membership in S is polynomial time. By the precise definition of the counting relation $\succeq_\epsilon$, item 2 is Σ^b_2.

Therefore this is an induction with $\log n$ many steps, with a Σ^b_2 inductive hypothesis. This can be done in Σ^b_1-IND.

We need some form of WPHP for cardinalities to behave.

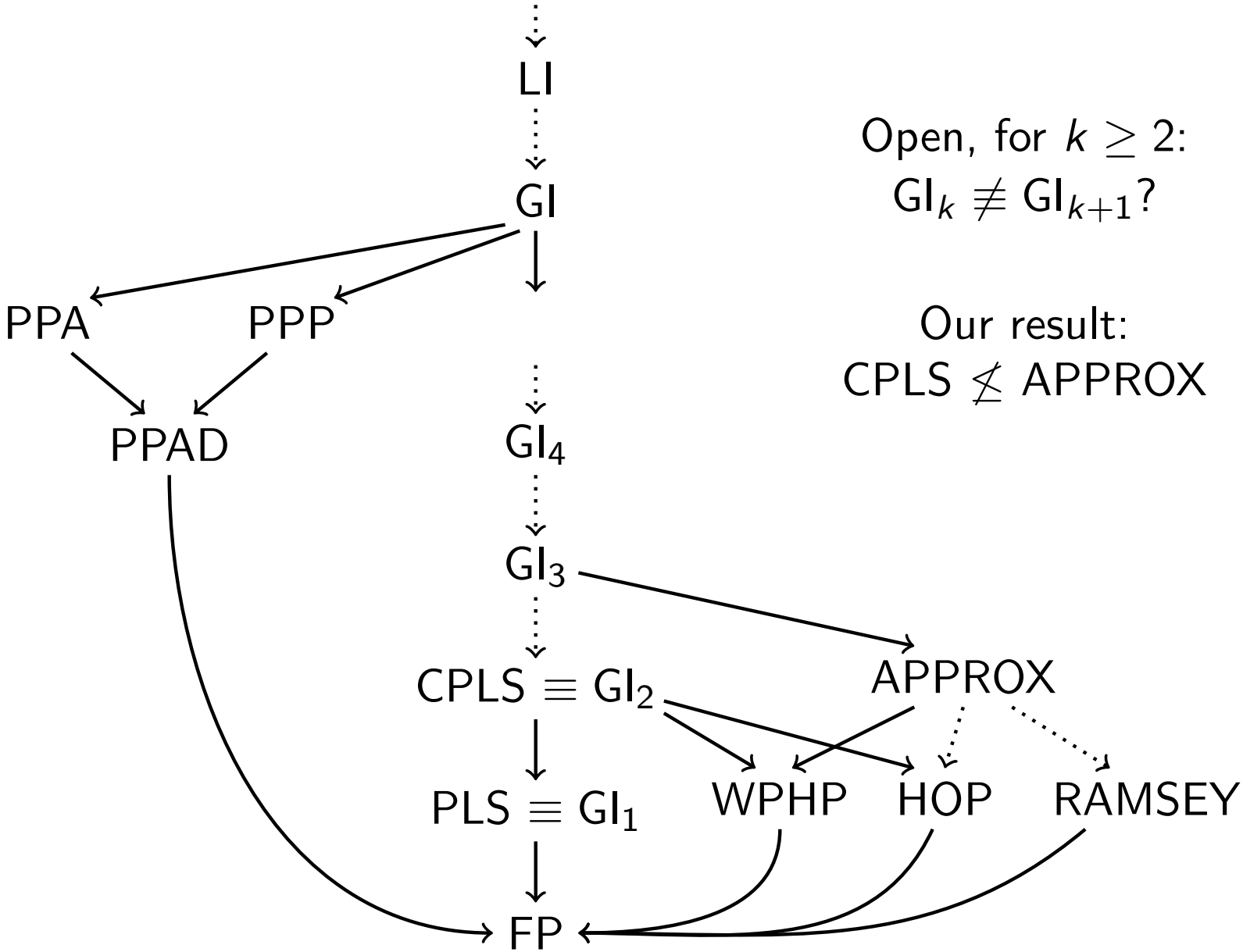
TFNP

Proofs and search problems

Approximate counting

Our result

TFNP



Open, for $k \geq 2$:
$GI_k \not\equiv GI_{k+1}$?

Our result:
$CPLS \not\leq APPROX$

A complexity-theoretic characterization of APPROX

Theorem

The search problems $R(x, y)$ reducible to APPROX are precisely those that can be solved in the following way.

A complexity-theoretic characterization of APPROX

Theorem

The search problems $R(x, y)$ reducible to APPROX are precisely those that can be solved in the following way.

Fix: a term $t(x)$, P^{NP} functions f_x and g_x,
and a search problem $Q(x', y')$ in PLS.

On input x:

A complexity-theoretic characterization of APPROX

Theorem

The search problems $R(x, y)$ reducible to APPROX are precisely those that can be solved in the following way.

Fix: a term $t(x)$, P^{NP} functions f_x and g_x,
and a search problem $Q(x', y')$ in PLS.

On input x:

- First solve the instance (f_x, g_x, t) of the *retraction WPHP*.

  That is: find a witness w that f_x is not a surjection $t \to 2t$ with inverse g_x.

A complexity-theoretic characterization of APPROX

Theorem

The search problems $R(x, y)$ reducible to APPROX are precisely those that can be solved in the following way.

Fix: a term $t(x)$, P^{NP} functions f_x and g_x,
and a search problem $Q(x', y')$ in PLS.

On input x:

- First solve the instance (f_x, g_x, t) of the *retraction WPHP*.

  That is: find a witness w that f_x is not a surjection $t \to 2t$ with inverse g_x.
- Then solve Q on w to find y' such that $Q(w, y')$.

A complexity-theoretic characterization of APPROX

Theorem

The search problems $R(x, y)$ reducible to APPROX are precisely those that can be solved in the following way.

Fix: a term $t(x)$, P^{NP} functions f_x and g_x,
and a search problem $Q(x', y')$ in PLS.

On input x:

- First solve the instance (f_x, g_x, t) of the *retraction WPHP*.

  That is: find a witness w that f_x is not a surjection $t \to 2t$ with inverse g_x.
- Then solve Q on w to find y' such that $Q(w, y')$.

Then $R(x, y')$.

Proof that CPLS $\not\leq$ APPROX (1)

We want to find an oracle α describing an instance of CPLS which cannot be solved in this way.

Our approach is: find a partial oracle ρ such that

- ρ "forces" some witness to the retraction WPHP, but
- it is still difficult to search for a witness to CPLS in oracles extending ρ.

Proof that CPLS $\not\leq$ APPROX (2)

We define a *legal* partial oracle to be one which does not contain a witness to CPLS (plus some other nice properties).

Proof that CPLS $\not\leq$ APPROX (2)

We define a *legal* partial oracle to be one which does not contain a witness to CPLS (plus some other nice properties).

Lemma [Pudlák, NT 16]

There is a distribution on partial oracles ρ, such that given any NP query $\exists v \theta(u, v)$, for a random ρ with high probability either

- there is a witness to the query in ρ, or
- there is no witness to the query in any legal $\sigma \supseteq \rho$.

In either case we say the query is *fixed* by ρ.

Proof that CPLS $\not\leq$ APPROX (3)

Lemma

Using the same distribution, for every P^{NP} machine M and input u, for a random ρ with high probability there is a computation of M on u in which every NP query made by M is fixed by ρ.

Say that such a computation is *fixed* by ρ.

Proof that CPLS $\not\leq$ APPROX (3)

Lemma

Using the same distribution, for every P^{NP} machine M and input u, for a random ρ with high probability there is a computation of M on u in which every NP query made by M is fixed by ρ.

Say that such a computation is *fixed* by ρ.

Corollary

There is a partial oracle ρ which fixes computations of M on almost all inputs u.

Proof that CPLS $\not\leq$ APPROX (3)

Lemma

Using the same distribution, for every P^{NP} machine M and input u, for a random ρ with high probability there is a computation of M on u in which every NP query made by M is fixed by ρ.

Say that such a computation is *fixed* by ρ.

Corollary

There is a partial oracle ρ which fixes computations of M on almost all inputs u.

This guarantees that a good-enough witness to the retraction WPHP exists over ρ. The condition on legal extensions makes it difficult to find a witness to CPLS over ρ, as required. □