TEXAS
The University of Texas at Austin

WHAT STARTS HERE CHANGES THE WORLD

Tighter Circuit Lower Bounds for MA/1 With Efficient PCPs

Joshua Cook
Joint work with Dana Moshkovitz

Trust Can’t Buy Time***

An Alternate Title

Untrusted Advice Vs Trusted Advice

	Deterministic	Randomized
No Advice	TIME[T]	BPTIME[T]
Untrusted, Adaptive	NTIME[T]	MATIME[T]
Trusted, Unadaptive	SIZE[T]*	BPTIME[T]/T
Untrusted, Unadaptive	ONTIME[T]	OMATIME[T]

* Is some gap between circuit size program size. Circuit size is more commonly studied, so used instead of TIME[T]/T

Expect New Resource To Help Solve Some Problems

Untrusted Advice Vs Trusted Advice

	Deterministic	Randomized
No Advice	TIME[T]	BPTIME[T]
Untrusted, Adaptive	NTIME[T]	MATIME[T]
Trusted, Unadaptive	SIZE[T]*	BPTIME[T]/T
Untrusted, Unadaptive	ONTIME[T]	OMATIME[T]

* Is some gap between circuit size program size. Circuit size is more commonly studied, so used instead of TIME[T]/T

Expect New Resource To Help Solve Some Problems

Suspect some problems can’t be sped up with these resources.

Untrusted Advice Vs Trusted Advice

	Deterministic	Randomized
No Advice	TIME[T]	BPTIME[T]
Untrusted, Adaptive	NTIME[T]	MATIME[T]
Trusted, Unadaptive	SIZE[T]*	BPTIME[T]/T
Untrusted, Unadaptive	ONTIME[T]	OMATIME[T]

* Is some gap between circuit size program size. Circuit size is more commonly studied, so used instead of TIME[T]/T

Expect New Resource To Help Solve Some Problems

Suspect some problems can't be sped up with these resources.

$$\mathrm{TIME}[n^4] \overset{?}{\subseteq} \mathrm{NTIME}[n]$$

Can All Statements Be Verified Faster than Computed?

Untrusted Advice Vs Trusted Advice

	Deterministic	Randomized
No Advice	TIME[T]	BPTIME[T]
Untrusted, Adaptive	NTIME[T]	MATIME[T]
Trusted, Unadaptive	SIZE[T]*	BPTIME[T]/T
Untrusted, Unadaptive	ONTIME[T]	OMATIME[T]

* Is some gap between circuit size program size. Circuit size is more commonly studied, so used instead of TIME[T]/T

Expect New Resource To Help Solve Some Problems

Suspect some problems can't be sped up with these resources.

$$\mathrm{TIME}[n^4] \overset{?}{\subseteq} \mathrm{NTIME}[n]$$

Can All Statements Be Verified Faster than Computed?

$$\mathrm{TIME}[n^4] \overset{?}{\subseteq} \mathrm{SIZE}[n]$$

Can fixed instance sizes be hard coded to faster, short programs?

Untrusted Advice Vs Trusted Advice

	Deterministic	Randomized
No Advice	TIME[T]	BPTIME[T]
Untrusted, Adaptive	NTIME[T]	MATIME[T]
Trusted, Unadaptive	SIZE[T]*	BPTIME[T]/T
Untrusted, Unadaptive	ONTIME[T]	OMATIME[T]

* Is some gap between circuit size program size. Circuit size is more commonly studied, so used instead of TIME[T]/T

Expect New Resource To Help Solve Some Problems

Suspect some problems can't be sped up with these resources.

$$\mathrm{TIME}[n^4] \overset{?}{\subseteq} \mathrm{NTIME}[n]$$

Can All Statements Be Verified Faster than Computed?

$$\mathrm{TIME}[n^4] \overset{?}{\subseteq} \mathrm{SIZE}[n]$$

Can fixed instance sizes be hard coded to faster, short programs?

$$\mathrm{NTIME}[n^4] \overset{?}{\subseteq} \mathrm{SIZE}[n]$$

Can any verifiable problem on fixed instance sizes be hard coded into a faster, short program.

Untrusted Advice Vs Trusted Advice

	Deterministic	Randomized
No Advice	TIME[T]	BPTIME[T]
Untrusted, Adaptive	NTIME[T]	MATIME[T]
Trusted, Unadaptive	SIZE[T]*	BPTIME[T]/T
Untrusted, Unadaptive	ONTIME[T]	OMATIME[T]

* Is some gap between circuit size program size. Circuit size is more commonly studied, so used instead of TIME[T]/T

Expect New Resource To Help Solve Some Problems

Suspect some problems can't be sped up with these resources.

$\mathrm{TIME}[n^4] \overset{?}{\subseteq} \mathrm{NTIME}[n]$

Can All Statements Be Verified Faster than Computed?

$\mathrm{TIME}[n^4] \overset{?}{\subseteq} \mathrm{SIZE}[n]$

Can fixed instance sizes be hard coded to faster, short programs?

$\mathrm{NTIME}[n^4] \overset{?}{\subseteq} \mathrm{SIZE}[n]$

Can any verifiable problem on fixed instance sizes be hard coded into a faster, short program.

$\mathrm{ONTIME}[n^4] \overset{?}{\subseteq} \mathrm{SIZE}[n]$

Can trusted programs always run faster than untrusted programs?

Santhanam: $\forall k>1$: $\mathrm{MATIME}[n^{O(k)}]/1 \not\subseteq \mathrm{SIZE}[O(n^k)]$

Non Explicit, Unbounded Polynomial

Santhanam: $\forall k>1$: MATIME$[n^{O(k)}]/1 \not\subseteq$ SIZE$[O(n^k)]$

Non Explicit, Unbounded Polynomial

Santhanam: $\forall k>1$: $MATIME[n^{O(k)}]/1 \not\subseteq SIZE[O(n^k)]$

Murray-Williams: $\forall k>1$: $MATIME[n^{ck^2}]/1 \not\subseteq SIZE[O(n^k)]$

Non Explicit, Unbounded Polynomial

Santhanam: $\forall k>1$: $\mathrm{MATIME}[n^{O(k)}]/1 \not\subseteq \mathrm{SIZE}[O(n^k)]$

$8 \gtrsim c \gtrsim 2$

Murray-Williams: $\forall k>1$: $\mathrm{MATIME}[n^{ck^2}]/1 \not\subseteq \mathrm{SIZE}[O(n^k)]$

Non Explicit, Unbounded Polynomial

Santhanam: $\forall k>1$: $\mathrm{MATIME}[n^{O(k)}]/1 \not\subseteq \mathrm{SIZE}[O(n^k)]$

$8 \gtrsim c \gtrsim 2$

Murray-Williams: $\forall k>1$: $\mathrm{MATIME}[n^{ck^2}]/1 \not\subseteq \mathrm{SIZE}[O(n^k)]$

Our result: $\exists a>1$: $\forall k<a$: $\mathrm{MATIME}[n^{k+o(1)}]/1 \not\subseteq \mathrm{SIZE}[O(n^k)]$

Non Explicit, Unbounded Polynomial

Santhanam: $\forall k>1$: $\quad MATIME[n^{O(k)}]/1 \not\subseteq SIZE[O(n^k)]$

$8 \gtrsim c \gtrsim 2$

Murray-Williams: $\forall k>1$: $\quad MATIME[n^{ck^2}]/1 \not\subseteq SIZE[O(n^k)]$

Non Explicit, but small

Our result: $\exists a>1$: $\forall k<a$: $\quad MATIME[n^{k+o(1)}]/1 \not\subseteq SIZE[O(n^k)]$

Non Explicit, Unbounded Polynomial

Santhanam: $\forall k>1$: $\quad$ $\mathrm{MATIME}[n^{O(k)}]/1 \not\subseteq \mathrm{SIZE}[O(n^k)]$

$8 \gtrsim c \gtrsim 2$

Murray-Williams: $\forall k>1$: $\quad$ $\mathrm{MATIME}[n^{ck^2}]/1 \not\subseteq \mathrm{SIZE}[O(n^k)]$

Non Explicit, but small

Our result: $\exists a>1$: $\forall k<a$: $\quad$ $\mathrm{MATIME}[n^{k+o(1)}]/1 \not\subseteq \mathrm{SIZE}[O(n^k)]$

$\forall k>1$: $\quad$ $\mathrm{MATIME}[n^{ak+o(1)}]/1 \not\subseteq \mathrm{SIZE}[O(n^k)]$

Win-Win if a is small

There exists randomized programs with one bit of trusted advice and a long, untrusted program advice that cannot be solved much faster with trusted advice.

Non Explicit, Unbounded Polynomial

Santhanam: $\forall k>1$: $\mathrm{OMATIME}[n^{O(k)}]/1 \not\subseteq \mathrm{BPTIME}[O(n^k)]/O(n^k)$

$8 \gtrsim c \gtrsim 2$

Murray-Williams: $\forall k>1$: $\mathrm{OMATIME}[n^{ck^2}]/1 \not\subseteq \mathrm{BPTIME}[O(n^k)]/O(n^k)$

Non Explicit, but small

Our result: $\exists a>1$: $\forall k<a$: $\mathrm{OMATIME}[n^{k+o(1)}]/1 \not\subseteq \mathrm{BPTIME}[O(n^k)]/O(n^k)$

$\forall k>1$: $\mathrm{OMATIME}[n^{ak+o(1)}]/1 \not\subseteq \mathrm{BPTIME}[O(n^k)]/O(n^k)$

Win-Win if a is small

Interactive Proofs (IPs)?

Untrusted Merlin
Randomized Arthur.

Many Questions and Answers.

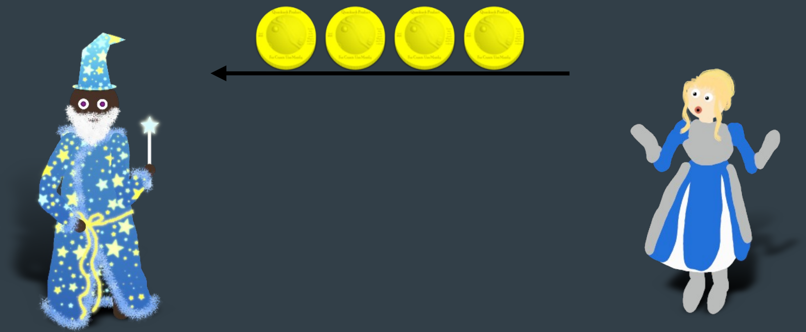
IVTIME[T]: Arthur time T.

Interactive Proofs (IPs)?

Untrusted Merlin
Randomized Arthur.

Many Questions and Answers.

IVTIME[T]: Arthur time T.

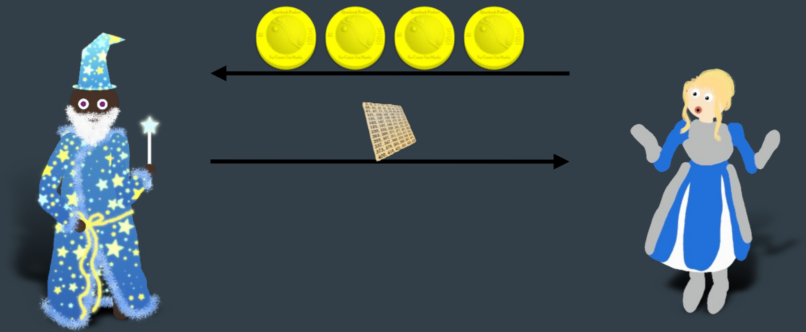


Interactive Proofs (IPs)?

Untrusted Merlin
Randomized Arthur.

Many Questions and Answers.

IVTIME[T]: Arthur time T.

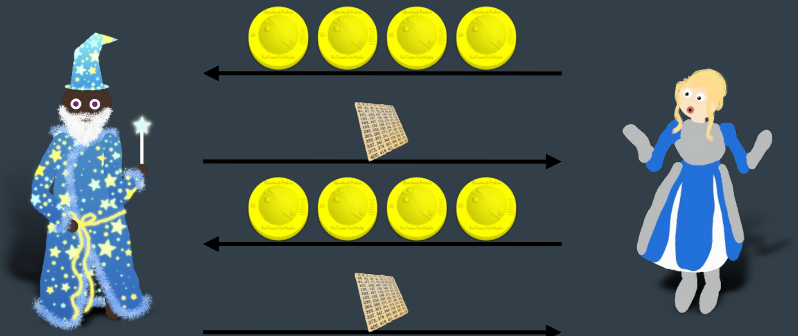


Interactive Proofs (IPs)?

Untrusted Merlin
Randomized Arthur.

Many Questions and Answers.

IVTIME[T]: Arthur time T.

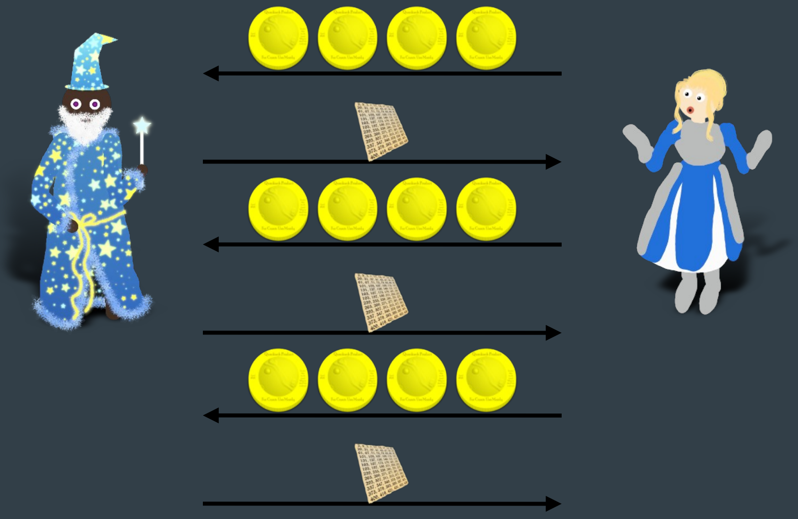


Interactive Proofs (IPs)?

Untrusted Merlin
Randomized Arthur.

Many Questions and Answers.

IVTIME[T]: Arthur time T.

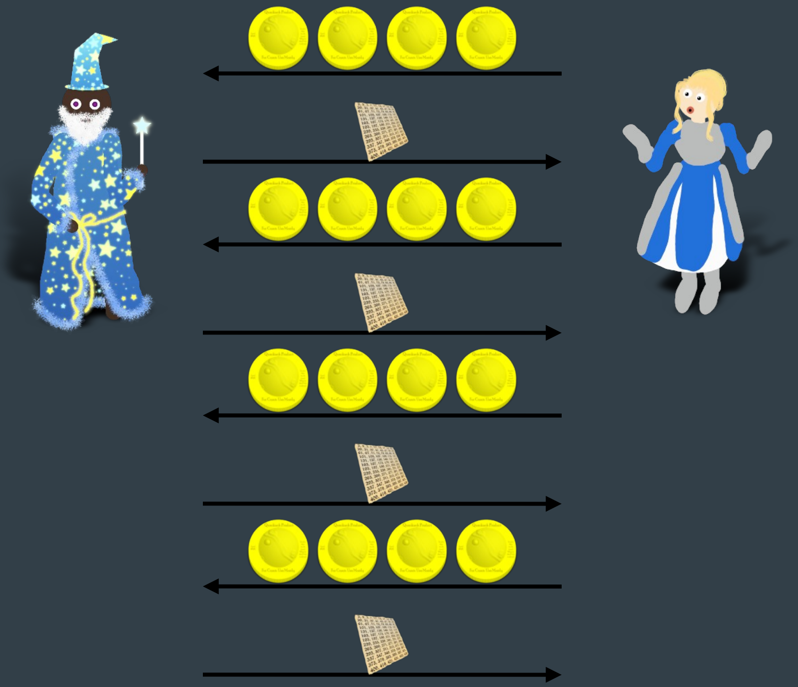


Interactive Proofs (IPs)?

Untrusted Merlin
Randomized Arthur.

Many Questions and Answers.

IVTIME[T]: Arthur time T.



How powerful is IP?

Shamir 92 proved IP = PSPACE!

$$\mathrm{SPACE}[n] \subseteq \mathrm{IVTIME}[n^2]$$

$$\mathrm{IVTIME}[n] \subseteq \mathrm{SPACE}[n]$$

Prover's for IP also small space!

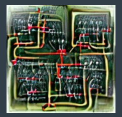
Circuit bounds for SPACE apply to IP!

Main Idea

Use a Circuit as Merlin in IP.

Merlin Gives a Circuit
Arthur Uses it to run IP

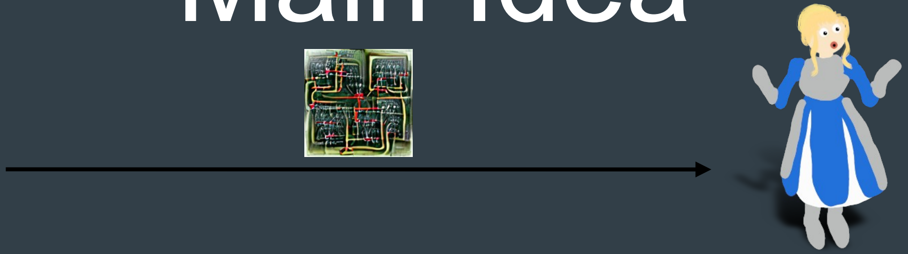
Main Idea

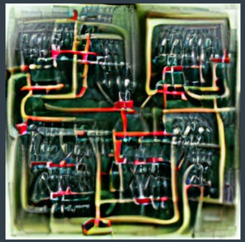


Use a Circuit as Merlin in IP.

Merlin Gives a Circuit
Arthur Uses it to run IP

Main Idea

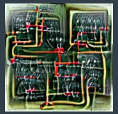


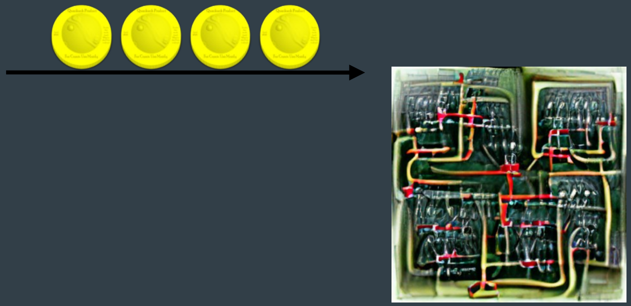


Use a Circuit as Merlin in IP.

Merlin Gives a Circuit
Arthur Uses it to run IP

Main Idea

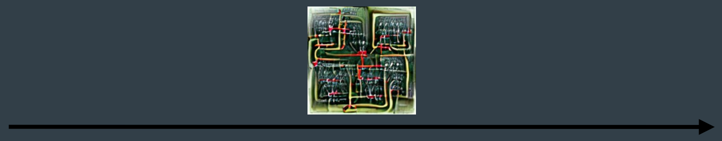


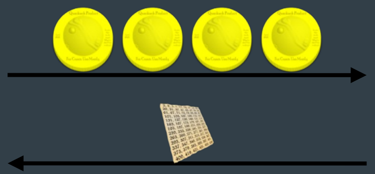


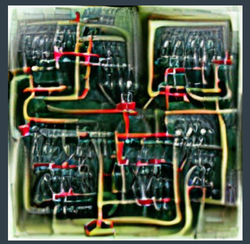
Use a Circuit as Merlin in IP.

Merlin Gives a Circuit
Arthur Uses it to run IP

Main Idea



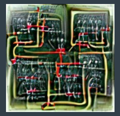


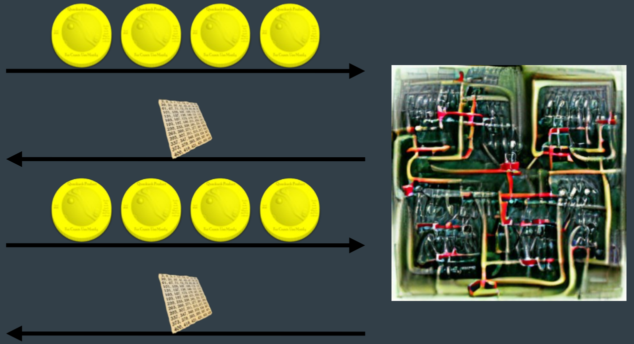


Use a Circuit as Merlin in IP.

Merlin Gives a Circuit
Arthur Uses it to run IP

Main Idea

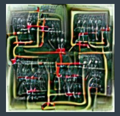


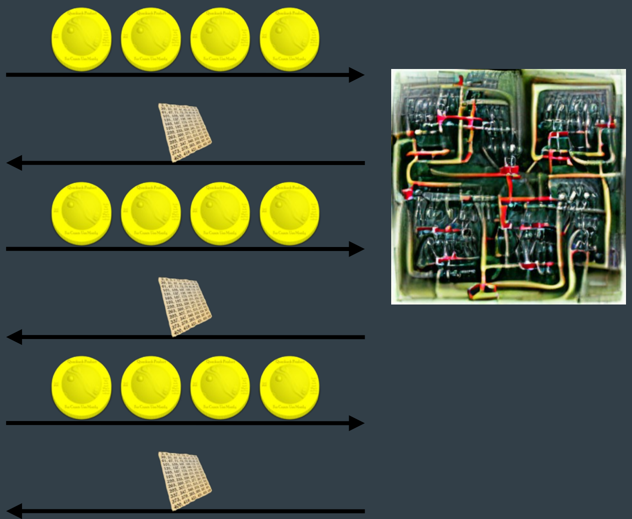


Use a Circuit as Merlin in IP.

Merlin Gives a Circuit
Arthur Uses it to run IP

Main Idea

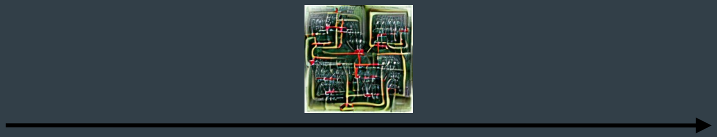


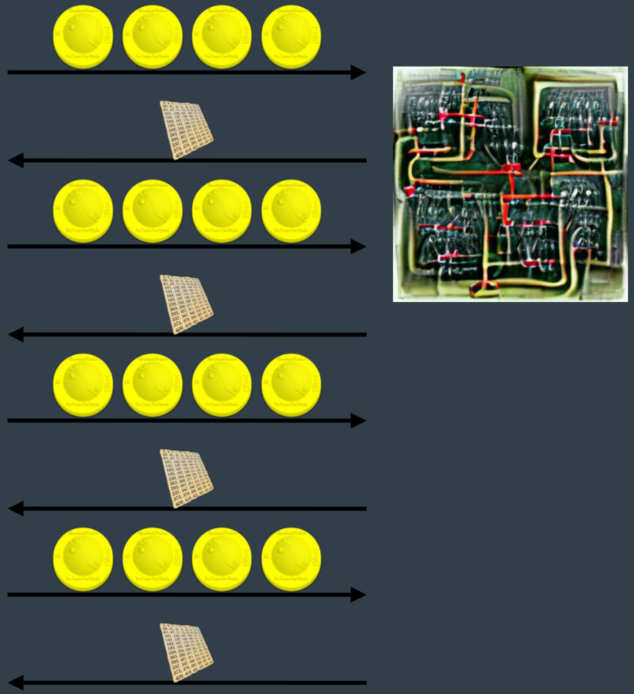


Use a Circuit as Merlin in IP.

Merlin Gives a Circuit
Arthur Uses it to run IP

Main Idea

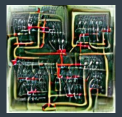


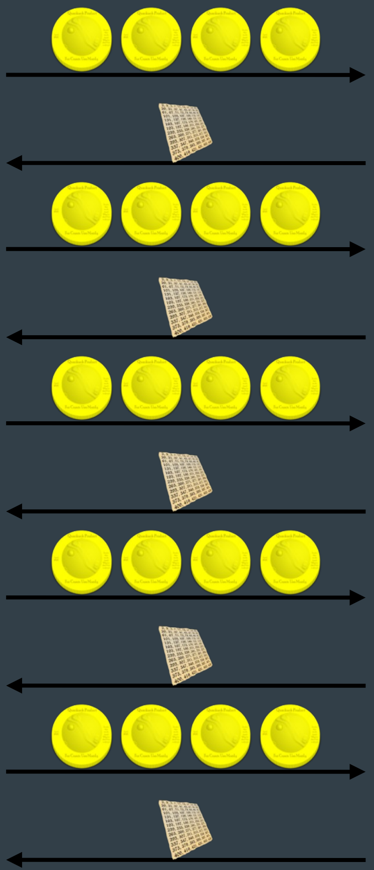


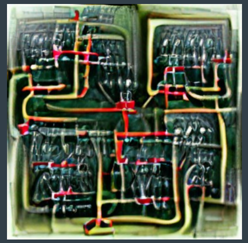
Use a Circuit as Merlin in IP.

Merlin Gives a Circuit
Arthur Uses it to run IP

Main Idea







Use a Circuit as Merlin in IP.

Merlin Gives a Circuit
Arthur Uses it to run IP

Santhanam's Proof: Lower Bound From IP=PSPACE

Santhanam's Proof: Lower Bound From IP=PSPACE

PSPACE

$\subseteq$?

P/poly

Santhanam's Proof: Lower Bound From IP=PSPACE

PSPACE $\subseteq$? P/poly

YES

- PSPACE $\not\subset$ SIZE[n^k] (PSPACE can search outside SIZE[n^k]).
- PSPACE=MA (MA guesses prover circuit for IP).

Santhanam's Proof: Lower Bound From IP=PSPACE

PSPACE $\subseteq$? P/poly

YES

- PSPACE $\not\subset$ SIZE[n^k] (PSPACE can search outside SIZE[n^k]).
- PSPACE=MA (MA guesses prover circuit for IP).

NO

- PSPACE-Complete L not in P/poly.
- Suppose L circuit size T>poly(n).
- Pad so T just above n^k (advice ensures padding right).
- MA guesses prover circuit.

Santhanam's Proof: Lower Bound From IP=PSPACE

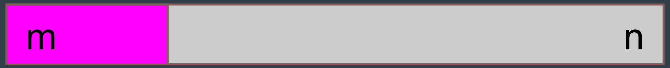
PSPACE $\subseteq$? P/poly

YES

- PSPACE $\not\subset$ SIZE[n^k] (PSPACE can search outside SIZE[n^k]).
- PSPACE=MA (MA guesses prover circuit for IP).

NO

- PSPACE-Complete L not in P/poly.
- Suppose L circuit size T>poly(n).
- Pad so T just above n^k (advice ensures padding right).
- MA guesses prover circuit.



$$T(m) \sim n^k$$

Santhanam's Proof: Lower Bound From IP=PSPACE

PSPACE $\subseteq$? P/poly

YES

- PSPACE $\not\subset$ SIZE[n^k] (PSPACE can search outside SIZE[n^k]).
- PSPACE=MA (MA guesses prover circuit for IP).

NO

- PSPACE-Complete L not in P/poly.
- Suppose L circuit size T>poly(n).
- Pad so T just above n^k (advice ensures padding right).
- MA guesses prover circuit.

- **To simulate verifier-prover interaction need time polynomially larger than prover circuit size.**

m n

$T(m) \sim n^k$

Santhanam's Proof: Lower Bound From IP=PSPACE

PSPACE $\subseteq$? P/poly

YES

- PSPACE $\not\subset$ SIZE[n^k] (PSPACE can search outside SIZE[n^k]).
- PSPACE=MA (MA guesses prover circuit for IP).

NO

- PSPACE-Complete L not in P/poly.
- Suppose L circuit size T>poly(n).
- Pad so T just above n^k (advice ensures padding right).
- MA guesses prover circuit.

- **To simulate verifier-prover interaction need time polynomially larger than prover circuit size.**
- Idea: Use PCP to minimize verifier time, queries, interaction.

m n

$$T(m) \sim n^k$$

New PCP Theorem

For Time-Space[T,S] there is PCP verifier with:

1. Verifier time O~(n+**logT**).
2. Prover space O~(S+n).
3. Queries O(logn + loglogT).
4. Answer size O(loglogT).

New PCP Theorem

Think of $T=2^n$ and $S=n$

For Time-Space[T,S] there is PCP verifier with:

1. Verifier time O~(n+**logT**).
2. Prover space O~(S+n).
3. Queries O(logn + loglogT).
4. Answer size O(loglogT).

New PCP Theorem

Think of $T=2^n$ and $S=n$

For Time-Space[T,S] there is PCP verifier with:

As opposed to polylogT [BGHSV05,…]

1. Verifier time $\tilde{O}(n+\log T)$.
2. Prover space $\tilde{O}(S+n)$.
3. Queries $O(\log n + \log\log T)$.
4. Answer size $O(\log\log T)$.

New PCP Theorem

Think of $T=2^n$ and $S=n$

For Time-Space[T,S] there is PCP verifier with:

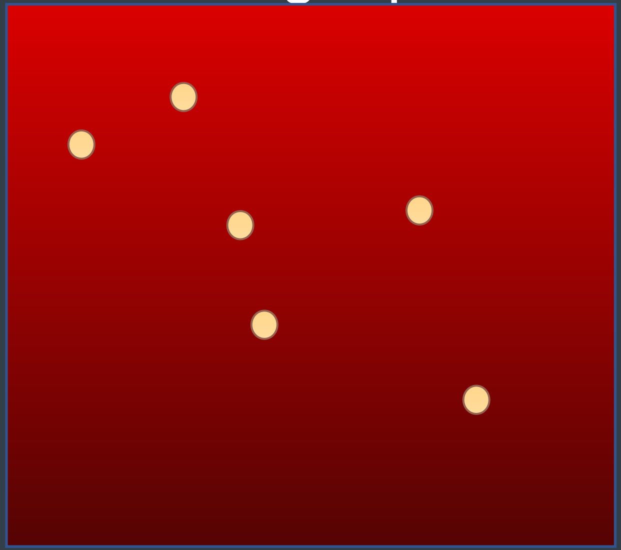
As opposed to polylogT [BGHSV05,…]

1. Verifier time $\tilde{O}(n+\mathbf{logT})$.
2. Prover space $\tilde{O}(S+n)$.
3. Queries $O(\log n + \log\log T)$.
4. Answer size $O(\log\log T)$.

Holmgren-Rothblum`18 could give O~(n+logT) verifier time, but O(logT) queries

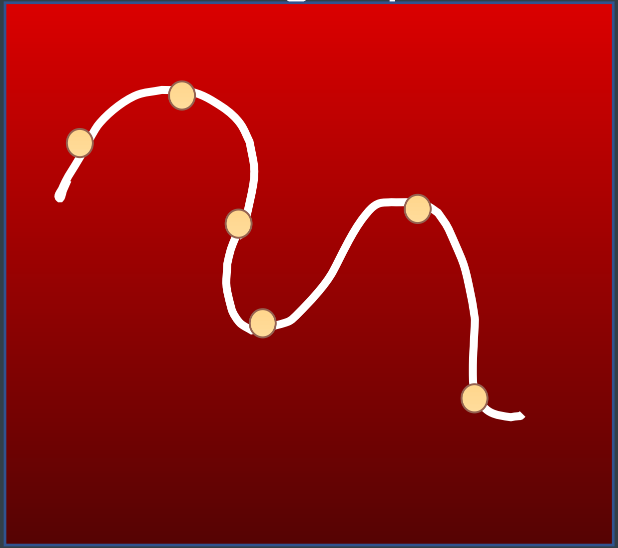
What Goes Into New PCP: Ultra-Efficient Query Reduction

- "Aggregation Through Curves": How to evaluate an m-variate low degree polynomial on k points using a prover?



What Goes Into New PCP: Ultra-Efficient Query Reduction

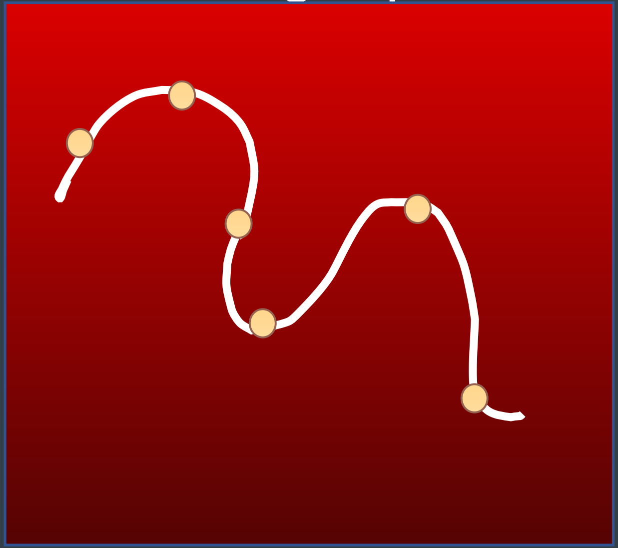
- "Aggregation Through Curves": How to evaluate an m-variate low degree polynomial on k points using a prover?



What Goes Into New PCP: Ultra-Efficient Query Reduction

"Aggregation Through Curves": How to evaluate an m-variate low degree polynomial on k points using a prover?

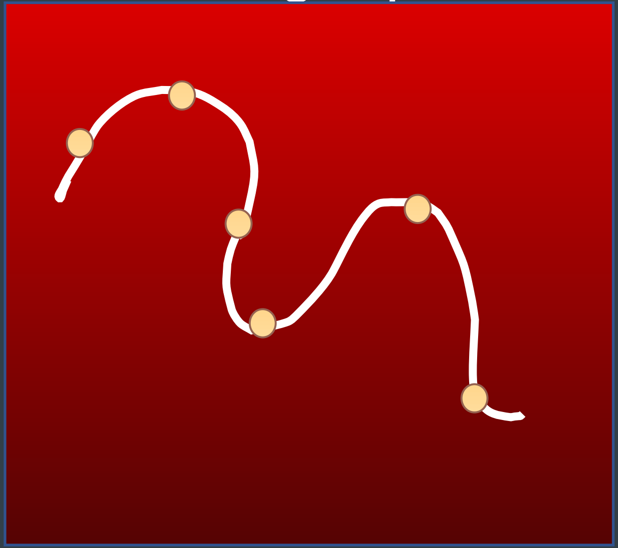
1. Pass degree-k curve through k points and random point.
2. Ask prover for the restriction of polynomial to curve.
3. Check restriction on random point.



What Goes Into New PCP: Ultra-Efficient Query Reduction

"Aggregation Through Curves": How to evaluate an m-variate low degree polynomial on k points using a prover?

1. Pass degree-k curve through k points and random point.
2. Ask prover for the restriction of polynomial to curve.
3. Check restriction on random point.

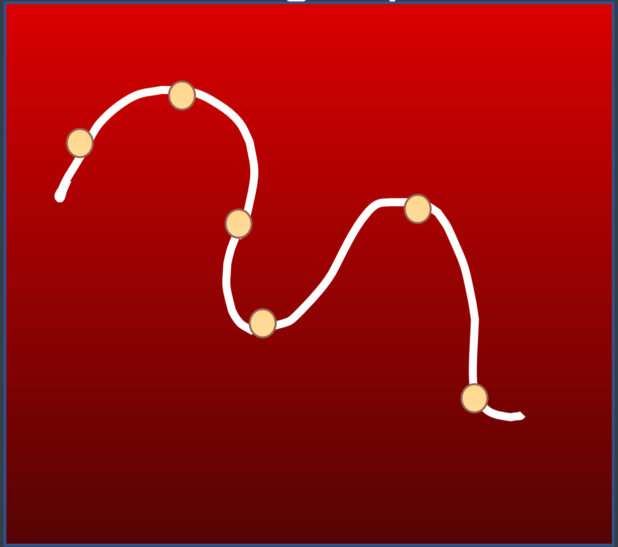


Time to compute curve ~km, instead of ~$k+m$.

What Goes Into New PCP: Ultra-Efficient Query Reduction

"Aggregation Through Curves": How to evaluate an m-variate low degree polynomial on k points using a prover?

1. Pass degree-k curve through k points and random point.
2. Ask prover for the restriction of polynomial to curve.
3. Check restriction on random point.



Time to compute curve ~km, instead of ~k+m.

Idea: need linear transformation of k points in time ~k+m. Possible for related points.

For Which k Prove $MATIME[n^{k+o(1)}]/1 \not\subset SIZE[n^k]$?

Have three cases:

1. $PSPACE \not\subset P/poly$
2. $SPACE[n] \subseteq SIZE[n^{1+o(1)}]$
3. **$\exists a>1$**: $SPACE[n] \subseteq SIZE[n^{a+o(1)}] - SIZE[n^{a-o(1)}]$

For Which k Prove $MATIME[n^{k+o(1)}]/1 \not\subset SIZE[n^k]$?

Have three cases:

All k. Santhanam's IP works, part of input running IP on shrinks very quickly, poly overhead shrinks.

1. $PSPACE \not\subset P/poly$
2. $SPACE[n] \subseteq SIZE[n^{1+o(1)}]$
3. $\exists$ **a>1**: $SPACE[n] \subseteq SIZE[n^{a+o(1)}] - SIZE[n^{a-o(1)}]$

For Which k Prove $\mathrm{MATIME}[n^{k+o(1)}]/1 \not\subset \mathrm{SIZE}[n^k]$?

Have three cases:

1. $\mathrm{PSPACE} \not\subset \mathrm{P/poly}$
2. $\mathrm{SPACE}[n] \subseteq \mathrm{SIZE}[n^{1+o(1)}]$
3. $\exists a>1$: $\mathrm{SPACE}[n] \subseteq \mathrm{SIZE}[n^{a+o(1)}] - \mathrm{SIZE}[n^{a-o(1)}]$

All k. Santhanam’s IP works, part of input running IP on shrinks very quickly, poly overhead shrinks.

All k. Space ~ Size. From our PCP Space ~ Prover Space ~ Prover size.

For Which k Prove MATIME[$n^{k+o(1)}$]/1 $\not\subset$ SIZE[n^k]?

Have three cases:

1. PSPACE $\not\subset$ P/poly
2. SPACE[n] $\subseteq$ SIZE[$n^{1+o(1)}$]
3. $\exists$ **a>1**: SPACE[n] $\subseteq$ SIZE[$n^{a+o(1)}$] - SIZE[$n^{a-o(1)}$]

All k. Santhanam's IP works, part of input running IP on shrinks very quickly, poly overhead shrinks.

All k. Space ~ Size. From our PCP Space ~ Prover Space ~ Prover size.

k < a. For k = a, Space[n] $\not\subset$ Size [n^a], but Prover Space[n] ~ Size [$n^{a+o(1)}$]. So OMA time is about Size [$n^{a+o(1)}$]. Pad inputs for k < a.

For k > a, need something stronger than Space[n] for hard problem. Space hardness might stall, may need Space[n^k], but then prover requires Space [n^k], may need Size[n^{ka}].

Citations

Sanjeev Arora and Shmuel Safra. “Probabilistic Checking of Proofs: A New Characterization of NP”. JACM 1998.

L. Babai, L. Fortnow, and C. Lund. “Nondeterministic exponential time has two-prover interactive protocols”. FOCS 1990.

Joshua Cook, Dana Moshkovitz. “Tighter MA/1 Circuit Lower Bounds From Verifier Efficient PCPs for PSPACE”. 2022.

Cody Murray and Ryan Williams. “Circuit Lower Bounds for Non-deterministic Quasi-Polytime: An Easy Witness Lemma for NP and NQP”. STOC 2018.

Rahul Santhanam. “Circuit Lower Bounds for Merlin-Arthur Classes”. STOC ’07.

Adi Shamir. “IP = PSPACE”. JACM 1992.

Ryan Williams. “Non-uniform ACC Circuit Lower Bounds”. CCC 2011.