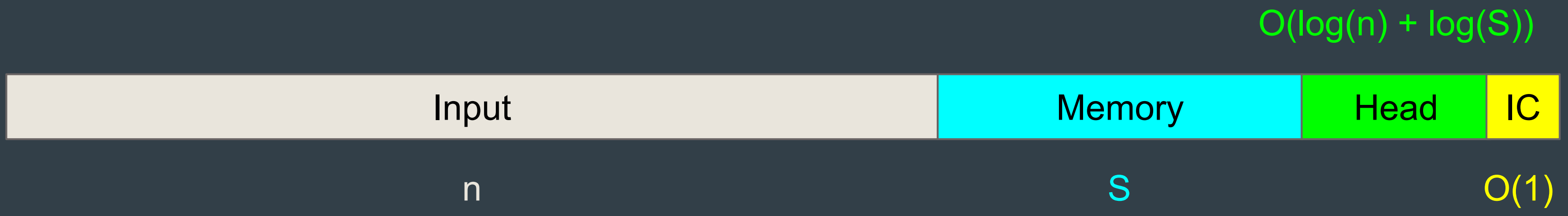
More Verifier Efficient Interactive Proofs For Bounded Space

Joshua Cook

Program To Run

- Deterministic Machine M in TISP[T, S]
  - Time T, Space S
- Think $S \ll n \ll T$
  - $S = n^{\alpha}$, $T = {}_{2}S^{\beta}$, for $\alpha, \beta > 1$



Arthur Doesn't Have Time!

Arthur wants to run M.

Doesn't have exponential time in S!

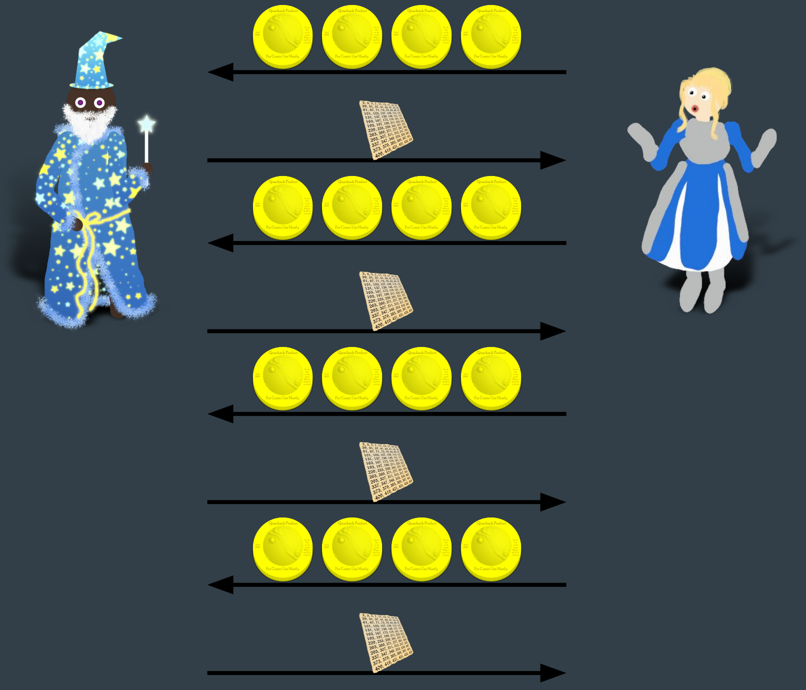
Merlin can help, but untrusted.

Has exponential time, but just $2^{O(S)}$.

Interactive Proofs (IPs)

Untrusted Merlin
(Prover P)
Randomized Arthur
(Verifier V)

Many Questions and
Answers.



Results

Interactive Time

L in ITIME[T_V, T_P]

Verifier time T_V, Prover time T_P

Completeness: If x in L,

then P convinces V with probability ⅔

Soundness: If x not in L,

then NO P' convinces V with probability ⅓

Main Result for TISP[T, S]

US: ITIME[$\tilde{O}(\log(T)S + n), 2^{O(S)}$]

(Previous Best, time not explicit prior)

Sha: ITIME[$\tilde{O}(\log(T)(S + n)), 2^{O(\log(T)(S + n))}$]

GKR: ITIME[$\tilde{O}(\log(T)S^2 + n), 2^{O(S)}$]

RRR: ITIME[$T^{o(1)}S^2 + n, T^{1+o(1)}S^{O(1)}$]

Us Vs Shamir

IP for SPACE[n^{α}] $T = 2^S$

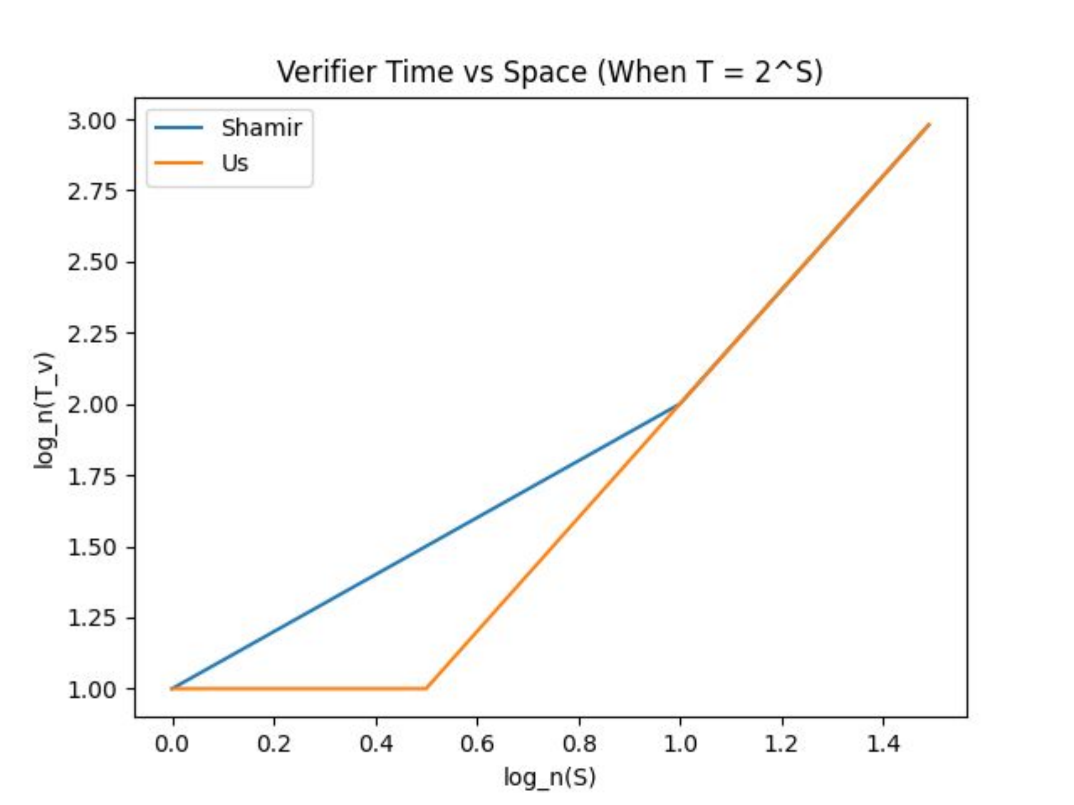
Verifier Time n^{β}

α vs β

Ours is better when $S < n$

Our prover is ALWAYS faster

$2^{O(S)}$ vs $2^{O(S^2)}$



IPs for Randomized Space

- Let L $\in$ BPSPACE[S]
- Standard: Saks Zhou, L $\in$ SPACE[$S^{3/2}$]:
  - Shamir, L has time S^3 verifier
- Us, Use Nisan's PRG with Our IP:
  - Reduction: space S, input length S^2
  - Our IP, L has time S^2 verifier
  - Match's deterministic IP

Nondeterministic Result

IP for NTISP[T, S]

US: ITIME[$\tilde{O}(\log(T)^2 S + n), 2^{O(S)}$]

Sha: ITIME[$\tilde{O}(\log(T)^2 (S + n)), 2^{O(\log(T)(S + n))}$]

GKR: ITIME[$\tilde{O}(\log(T) S^2 + n), 2^{O(S)}$]

RRR: NA

Us Vs GKR

IP for NTISP[$_2 n^{\alpha}$, n]

Verifier Time n^{β}

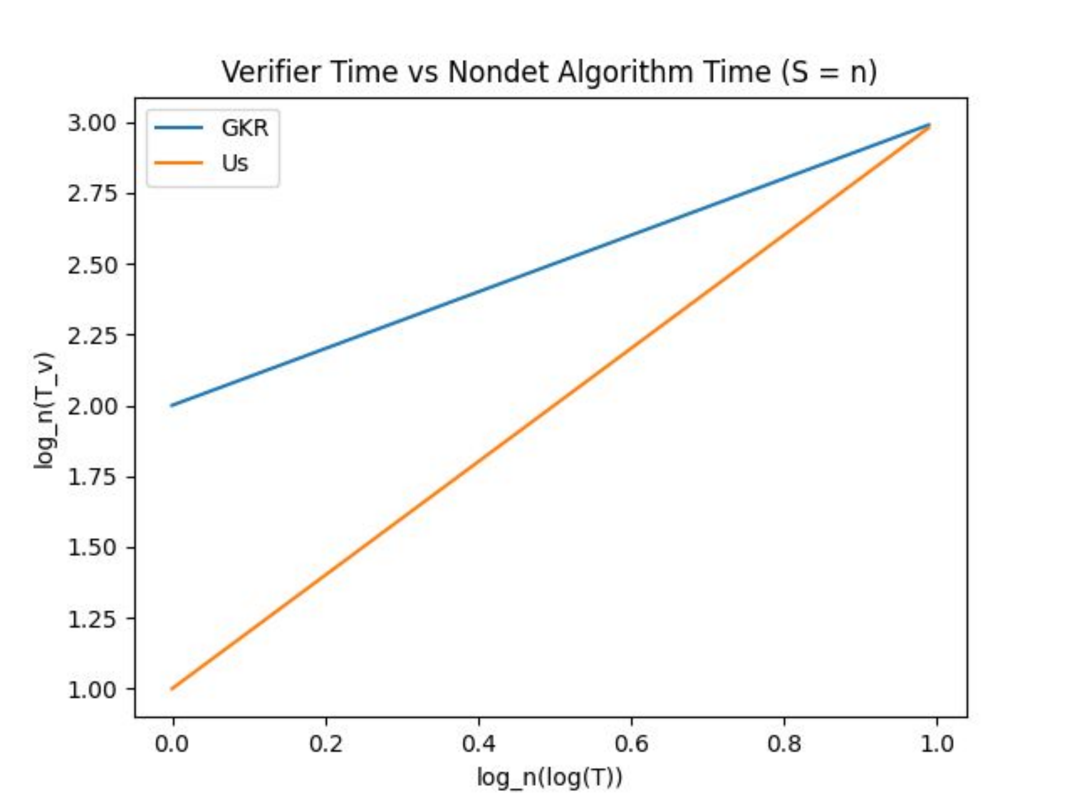
α vs β

Ours is better

When $T \ll 2^S$

Deterministic Algorithm

Both Prover $2^{O(S)}$



Proof

Proof Outline

Us

- Space to Matrix
  - Simpler reduction
- Matrix Sum Check
  - Simpler
- Arithmetize Multitape
  - Allows $S < n$

Shamir

- Space to QBF
  - Needs conditioning
- QBF Sum Check
  - Requires Specific Format Reduction
- Arithmetize Single

Why Not Single Tape TM?

Single tape TM require $S > n$

Concern, need $\tilde{O}(n + S)$ time arithmetization

- Show for multitape TM, paper uses RAM

RAM more efficient, only constant factor

Reduction To Matrix

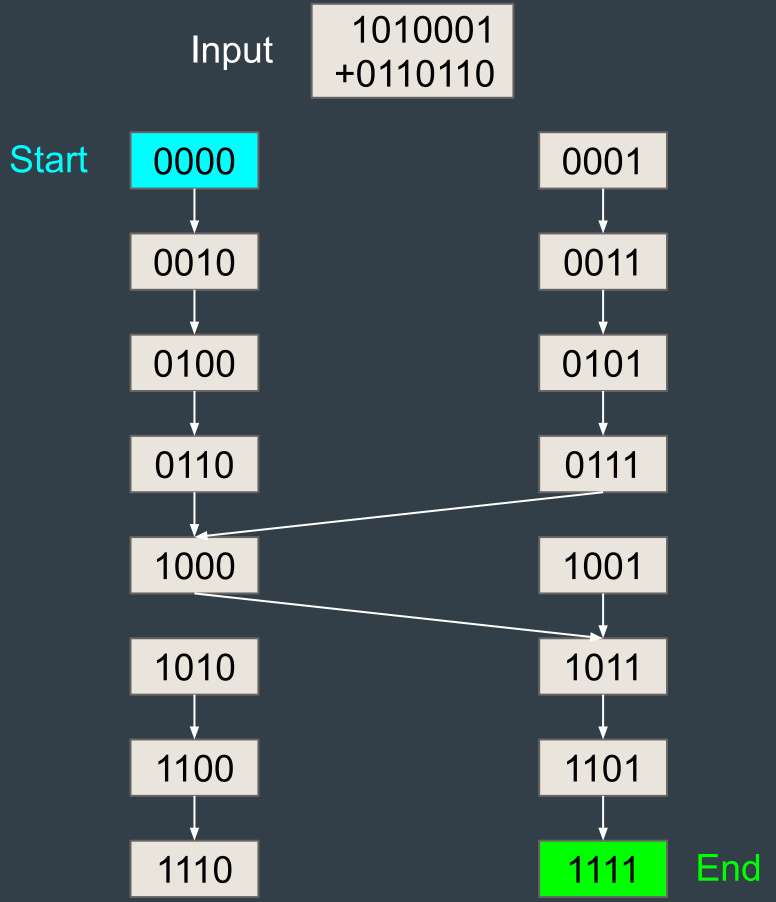
Computation Graph

View space S program as
2^S state graph, G
Edges are state transitions

Graph is a function of
Input, Program

Accepts IFF there is a length T path from start to end.

Edges are fast to compute



Adjacency Matrix

Represent G as an adjacency, M

Algorithm accepts in time T iff

$$M^T_{\text{start, end}} = 1$$

By repeated squaring,

$$M^T = M^{2^t}$$

For t = log(T)

Run matrix sum check log(T) times

		1													
			1												
				1											
					1										
						1									
							1								
								1							
								1							
										1					
										1					
											1				
												1			
													1		
														1	

Matrix Sum Check

Sum Check (LFKN)

Given: individual degree d polynomial, $p: \mathbb{F}^s \to \mathbb{F}$,
and $\alpha \in \mathbb{F}$

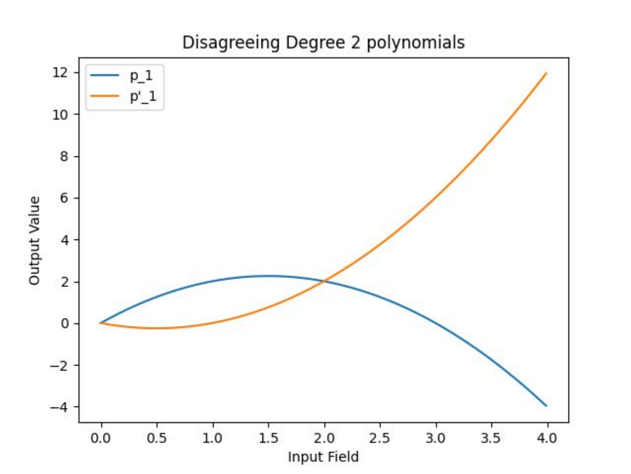
Reduce claim: $\alpha = \sum_{a \in \{0,1\}^s} p(a)$

To new claim: $\square = p(b)$
some $\square \in \mathbb{F}$, $b \in \mathbb{F}^s$

Sum Check Protocol

- Ask for $p_1(x) = \sum_{a \in \{0,1\}^{S-1}} p(x, a)$
- Check if $\alpha = p_1(0) + p_1(1)$
- Set b_1 randomly
- Ask for $p_2(x) = \sum_{a \in \{0,1\}^{S-2}} p(b_1, x, a)$
- Check if $p_1(b_1) = p_2(0) + p_2(1)$
- ...

Sum Check Idea (Schwartz-Zippel)



If $\alpha \neq \sum_{a \in \{0,1\}^s} p(a)$, then p_1 is incorrect.

p_1 is degree d, equal to true $p_1 \leq$ d places

$\Pr[\text{agree at } b_1] \leq d / |\mathbb{F}|$

Sum Check Performance

There exists an IP with verifier V, prover P:

Completeness: If $\alpha=\sum_{a\in\{0,1\}^s} p(a)$, with P, V gives $\square\in\mathbb{F}$ and $b\in\mathbb{F}^S$ s.t. $\square=p(b)$

Soundness: If $\alpha\neq\sum_{a\in\{0,1\}^s} p(a)$, for any P', V gives $\square=p(b)$ with probability $< Sd\,/\,|\mathbb{F}|$

Time: Verifier $Sd\ \tilde{O}(\log(|\mathbb{F}|))$ Prover $2^{O(S)}\ \tilde{O}(\log(|\mathbb{F}|))$

Matrix Multilinear Extension

For 2^Sx2^S matrix M containing elements of $\mathbb{F}$

Let **M** : $\mathbb{F}^S \times \mathbb{F}^S \rightarrow \mathbb{F}$ be s.t.

- **M** is multilinear (individual degree 1)
- For any a, c $\in \{0,1\}^S$, **M**(a, c) = $M_{a,c}$

Matrix Sum Check (Thaler)

By definition $M^2_{a,c} = \sum_{b \in \{0,1\}^s} M_{a,b} M_{b,c}$

Also have $\mathbf{M}^2(a,c) = \sum_{b \in \{0,1\}^s} \mathbf{M}(a,b)\mathbf{M}(b,c)$

For claim $\alpha = \mathbf{M}^2(a,c)$, let $p(b) = \mathbf{M}(a,b)\mathbf{M}(b,c)$

Sum check reduces to □ $= \mathbf{M}(a,b)\mathbf{M}(b,c)$

Product Reduction

Reduce claim: □ = p(a)p(b)

To new claim: α' = p(c)

- Let ψ: $\mathbb{F} \to \mathbb{F}^S$ be line s.t. $\psi(0) = a$, $\psi(1) = b$

  $$\psi(x) = (1-x)\,a + x\,b$$

- Ask for degree S polynomial $q(x) \stackrel{\text{def}}{=} p(\psi(x))$
- Check if □ = q(0)q(1)
- For random z, set $\alpha' = q(z)$, $c = \psi(z)$

  $$\alpha' = q(z) = p(\psi(z)) = p(c)$$

Repeated Square Rooting

For start a, end b:

Verifier given claim $M^T_{a,b}=1$, or $\mathbf{M^T}(a,b) = 1$

Reduce to claim $\mathbf{M^{2^{t-1}}}_{(a',b')}=\alpha'$, $\mathbf{M^{2^{t-2}}}_{(a'',b'')}=\alpha''$...

After log(T) times, have claim $\mathbf{M}(a^*,b^*) = \alpha^*$

Uses S log(T) operations over $\mathbb{F}$

Arithmetization

Calculate **M**, multilinear extension

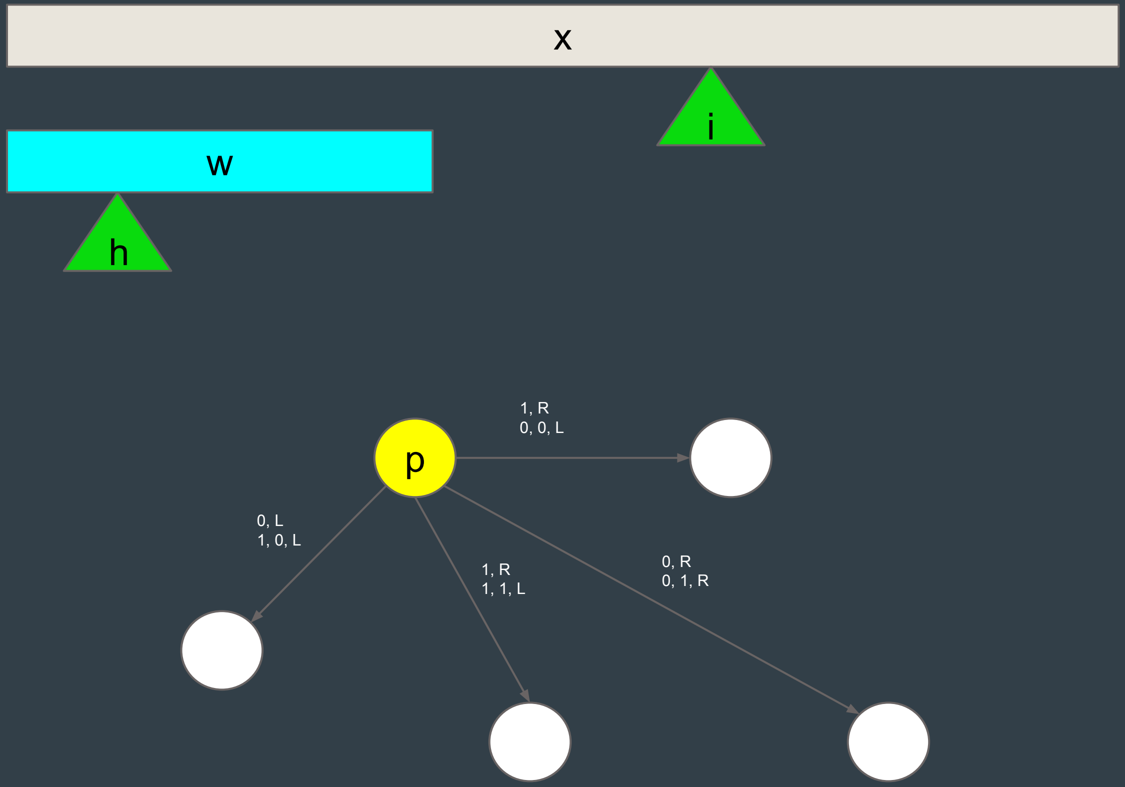
From program definition, $M_{a,b}$ simple.

How to calculate **M**?

Sum over every edge in program, simple formula can calculate easily.

Two Tape TM

Program has two tapes, input and working, Λ program transitions



Input x,
Initial state a = (p, i, h, w)
Final state b = (p’, i’, h’, w’)

p, p’ TM program states,
i, i’ input heads
h, h’ working space heads
w, w’ working space contents

Transition Function

$\sum_{\lambda \in \Lambda} u(\lambda, p)\, v(\lambda, p')\, \mathrm{Inp}(\lambda, x, i, i')\, \mathrm{Wrk}(\lambda, h, h', w, w')$

u λ is from state p v λ is to state p'

Inp x at i has symbol in λ, i' is i+1 or i-1 from λ

Wrk w at h from λ, h' is h+1 or h-1 from λ,
w' at h from λ, w' = w elsewhere

Use different symbols! Calculate extensions separately!

Closer Look: Wrk(λ, h, h’, w, w’)

$$\sum_{i \in [S]} eq(i, h)\, eq(i+D(\lambda), h')\, bef(i, w, w')\, aft(i, w, w')\, eq(us(\lambda), w_i)\, eq(vs(\lambda), w'_i)$$

eq	checks equality	D	1 for R, -1 for L
bef	equality before i	aft	equality after i
us	starting symbol	vs	ending symbol

Use different symbols! Calculate extensions separately!

Calculate **Wrk** Efficiently

- **eq**$(w_i, w'_i) = w_i w'_i + (1 - w'_i)(1 - w'_i)$
- **bef**$(i+1, w, w')$ = **bef**(i, w, w') **eq**(w_i, w'_i)
- **bef**(i, w, w') can be calculated for each i in O(S) operations. **aft** similarly
- Similarly, eq(i, h) for each i with O(S) ops.
- Only O(S) operations in Wrk

Finishing up Arithmetization

- **Inp** similarly calculated in O(n) operations
- Total **M** only takes O(n + S) operations.

Prover Time

Entire M can be constructed in time ~ 2^{2S}

Each M^k for $k = 2^i$ in time ~ $\log(T)2^{\omega S}$

Any **M^k**(a, b) calculated in time ~ 2^{2S}

Open Problems

- Remove log(T) factor from verifier time
- Do nondeterministic algorithms have same verifier time as deterministic?
- Same verifier time, poly(T) time prover?
- Gives quadratic gap interactive hierarchy
  - Fine grain interactive hierarchy?

Citations

- Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. “Algebraic methods for interactive proof systems”. FOCS 1990.
- Adi Shamir. “IP = PSPACE”. JACM 1992.
- Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. “Delegating computation: Interactive proofs for muggles”. JACM 2015.
- Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. “Constant-round interactive proofs for delegating computation”. STOC 2016.
- Michael Saks and Shiyu Zhou. “BpH space(s)⊆dspace(s3/2)”. J. Comput. Syst. Sci. 1999.
- Noam Nisan. “Pseudorandom generators for space-bounded computations”. STOC 1990.
- Justin Thaler. “Time-Optimal Interactive Proofs for Circuit Evaluation”. CRYPTO 2013.