The communication complexity of addition, with applications

August 2014

Emanuele Viola

Northeastern University

- 2-player addition

Player P_1 gets integer $x_1 \in [-2^n , 2^n]$
P_2 x_2

How many communication bits to decide if $x_1 + x_2 > 0$ with error 1%?

Public-coin: A random string is shared

- 2-player addition $(x_1 + x_2 > 0?, \ x_i \in [-2^n , 2^n])$

[Nisan Safra '93] O(log n)

Idea: ?

- 2-player addition $(x_1 + x_2 > 0?,\ x_i \in [-2^n, 2^n])$

[Nisan Safra '93] $O(\log n)$

Idea: Compute $\max i : (x_1)_i \neq (x_2)_i$

Binary search on bits, run equality at each step.

Implementation:

Equality with $O(1)$ communication, error 1%

Binary search with noise; $O(\log n)$ steps still suffice

- 2-player addition $(x_1 + x_2 > 0?, \ x_i \in [-2^n, 2^n])$

[Smirnoff '88] $\Omega(\sqrt{\log n})$

[Nisan Safra '93] $O(\log n)$

This work: $\Omega(\log n)$

Corollary: $\Theta(\log n)$

- 2-player addition $(x_1 + x_2 > 0?,\ x_i \in [-2^n, 2^n])$

Proof of $\Omega(\log n)$ lower bound:

Hard distributions: $I \in [n]$ uniform $\qquad Y \in \{0,1\}^n$ uniform

$G = (G_1, G_2) = (Y_1\, Y_2 \ldots Y_n,\ Y_1\, Y_2 \ldots\ Y_I\quad 0\ 0 \ldots 0)$

$B = (B_1, B_2) = (Y_1\, Y_2 \ldots Y_n,\ Y_1\, Y_2 \ldots (1-Y_I)\ 0\ 0 \ldots 0)$

$G_1 \geq G_2$ always; $B_1 \geq B_2$ with probability ½

Claim: For every rectangle $R = R_1 \times R_2$ s. t. $\Pr[G \in R] \geq 1/n$

We have $\Pr[B \in R] \geq \Pr[G \in R] - 1/n^{0.3}$

Proof: Conditioned on $G_1 \in R_1$, $H(Y) \geq n - \log n$,

So Y_I has entropy $\geq 1 - \log(n)/n$, so $Y_I \approx$ uniform $\approx 1-Y_I$ ■

- 2-player addition $(x_1 + x_2 > 0?,\ x_i \in [-2^n, 2^n])$

- [Nisan Safra '93] O(log n) + [Newman '91]

  ➔ O(log n) communication, private-coin, not explicit

- This work: O(log n) communication, private-coin, explicit

  Proof: Use small-bias generator for equality

  Use space-bounded generator for binary search

■

Detour application I [Dutta Pandurangan Rajaraman Sun V.]

Problem: Two players, each holding a subset x_i of $[n]$.

Want ε-uniform element from symmetric difference $x_1 \oplus x_2$

Part of [DPRSV] proposal for spreading on dynamic networks

Claim: Explicit, private-coin protocol with ~$O(\log n/ \varepsilon)$ comm.

Proof:

?????

Detour application I [Dutta Pandurangan Rajaraman Sun V.]

Problem: Two players, each holding a subset x_i of $[n]$.
Want ε-uniform element from symmetric difference $x_1 \oplus x_2$

Part of [DPRSV] proposal for spreading on dynamic networks

Claim: Explicit, private-coin protocol with ~$O(\log n/ \varepsilon)$ comm.

Proof:
Players agree on uniform permutation π.
Run Nisan-Safra protocol on $\pi (x_1) \oplus \pi (x_2)$

π: pseudorandom generators for combinatorial rectangles
[Gopalan Meka Reingold Trevisan Vadhan]

■

Is multiplication harder than addition?

Cobham 1964

- 2-player multiplication

Player P_1 gets integer $x_1 \in [-2^n , 2^n]$

P_2 x_2

How many communication bits to decide if $x_1 \cdot x_2 > 2^{n/2}$ with error 1%?

Do you know how to solve this?

- 2-player multiplication

Player P_1 gets integer $x_1 \in [-2^n , 2^n]$

P_2 x_2

How many communication bits to decide if $x_1 \cdot x_2 > 2^{n/2}$ with error 1%?

Corollary [V]: O(log n) communication

Proof:

Take logs

Results on logarithmic forms by Baker et al. imply that you can truncate after poly(n) digits.

Run protocol for addition. ∎

Outline

- Results for 2 players

- Results for k players

- Proof of O(log n) bound for k-player addition

- k-player addition

Player P_i gets $x_i \in [\,-2^n, 2^n\,]$, $i=1, \ldots, k$; (number-in-hand)

How much communication to decide $\sum_{i \leq k} x_i > 0$ with error 1%?

- From now on, public-coin model

  For simplicity, $k = O(1)$

- k-player addition $(\sum_i x_i > 0?,\ x_i \in [-2^n, 2^n],\ k = O(1))$

[Nisan '93] $O(\log^2 n)$

This work: $O(\log n)$

Corollary: $\Theta(\log n)$

- Degree-d polynomial-threshold function in n variables

  How much communication for number-on-forehead protocols among $k = d+1$ players?

  Corollaries to k-player addition:

  [Nisan '93] $O(\log^2 n)$

  This work: $O(\log n)$

- Application to the complexity of pseudorandom functions

Table 1: Pseudorandom functions $F : \{0,1\}^n \to \{0,1\}$ computable by circuits of size poly(n) and depth $O(1)$.

Complexity class	Security	Reference
TC^0	Secure against time $t = t(n)$ under assumptions in [NR04] against times poly$(n)t(n)$	[NR04, NRR02]
AC^0 with Mod m gates, any m; CC^0	Secure against time $n^{\lg^c n}$ under assumptions in [NR04] against time $2^{n^{\Omega(1)}}$ (circuit depth depends on c)	Theorem 11
AC^0 with Mod m gates, prime m	Breakable in time $n^{\lg^c n}$ (c depends on circuit depth)	[RR97, KL01]
AC^0 with $O(1)$ threshold gates and $O(1)$ symmetric gates (e.g. parity, majority)	Breakable in time poly(n)	Theorem 10
AC^0	Breakable in time poly(n)	[LMN93]

Claim: AC^0 with 1 threshold gate is breakable in poly(n) time

Note: Previously quasi-polynomial time was known.

Proof:
Hit AC^0 with a random restriction.

It collapses to a polynomial threshold function of degree O(1)

By previous fact, it has O(log n) communication (error 1%)

This means that the Babai-Nisan-Szegedy "norm" R
(see Chung Tetali, Raz, V Wigderson) is ≥ ?

Claim: AC^0 with 1 threshold gate is breakable in poly(n) time

Note: Previously quasi-polynomial time was known.

Proof:

Hit AC^0 with a random restriction.

It collapses to a polynomial threshold function of degree O(1)

By previous fact, it has O(log n) communication (error 1%)

This means that the Babai-Nisan-Szegedy “norm” R
(see Chung Tetali, Raz, V Wigderson) is ≥ 1/poly(n)

Whereas for a random function R is negligible

This difference can be detected in poynomial time. ∎

Outline

- Results for 2 players
- Results for k players
- Proof of O(log n) bound for k-player addition

- Recall k-player addition:

P_i gets integer $x_i \in [-2^n , 2^n]$

How much communication to decide $\sum_{i\leq k} x_i > 0$ with error 1%?

- Overview of ideas in our O(log n) protocol
  - We give O(1) protocol for k-player sum-equal, improving on Nisan's O(log n)
  - Using a recursion [Nisan] this gives O(log n log log n) protocol for k-player addition
  - We adapt [Nisan Safra] from k = 2 to k > 2 to obtain O(log n)

- k-player sum-equal

Player P_i gets integer $x_i \in [-2^n , 2^n]$

How much communication to decide $\sum_{i \leq k} x_i = 0$ with error 1%?

- k-player sum-equal ($\sum_{i \le k} x_i = 0?$, $x_i \in [-2^n, 2^n]$)

- [Nisan] Player P_i communicates $hash(x_i) = x_i \bmod p$

  Correctness by linearity: $\sum_i (x_i \bmod p) = (\sum_i x_i) \bmod p$

  Need $p = n^{\Omega(1)}$ ➔ $\Omega(\log n)$-bit hashes

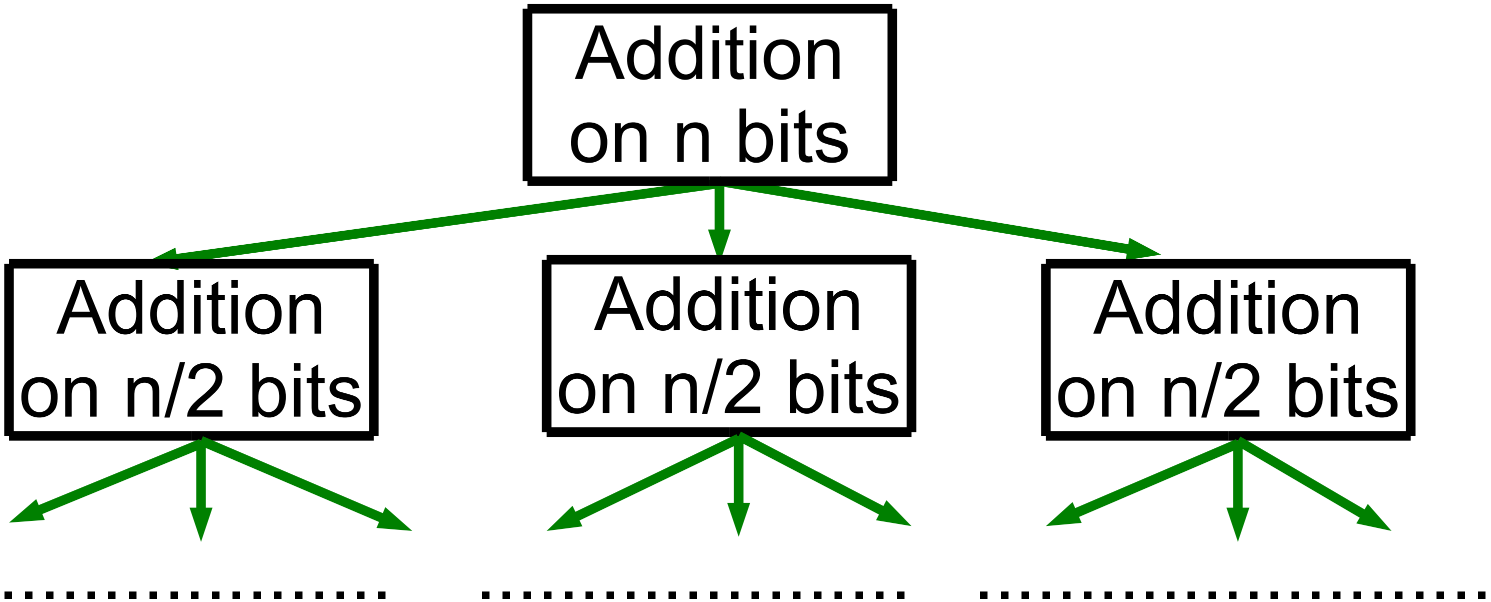
- This work: Use hash function analyzed by [Dietzfelbinger Hagerup Katajainen Penttonen]

  $hash(x_i)$ = "O(1) middle bits of $R \cdot x_i$, R random odd"

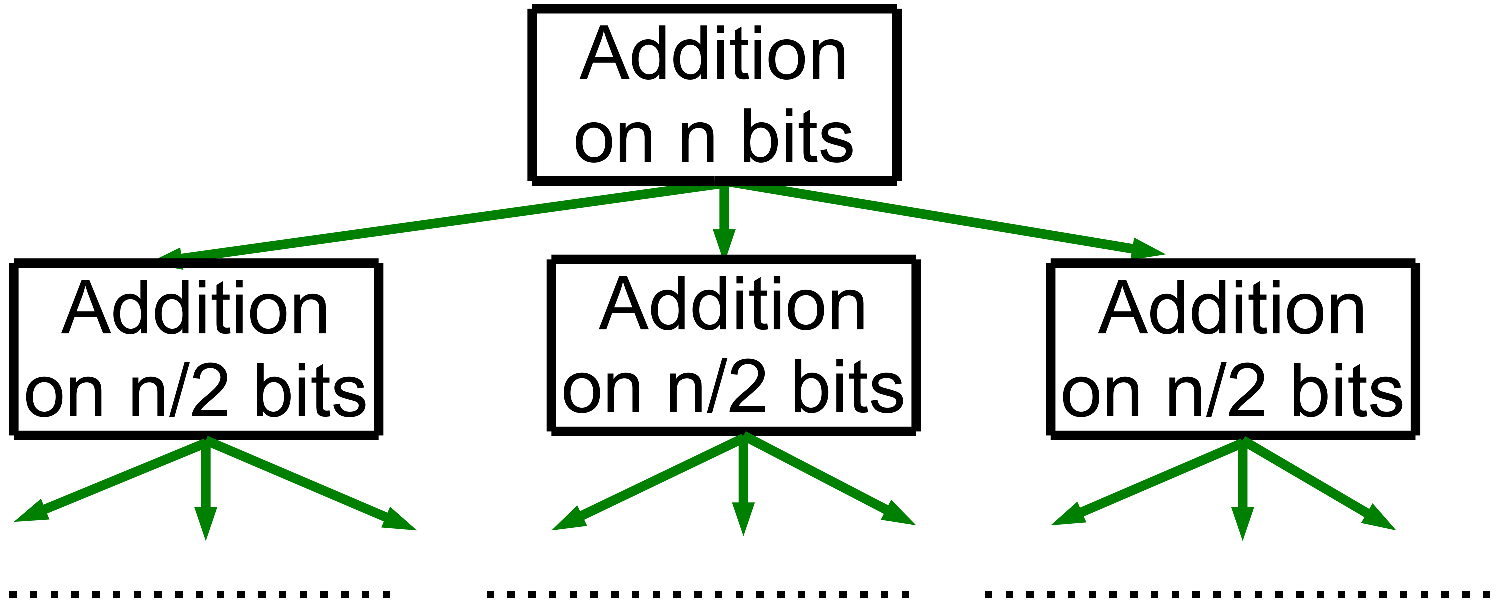
  Almost linear: $\sum_{i \le k} hash(x_i) = hash(\sum_{i \le k} x_i)$ +/- k

  O(1)-bit hashes

- [Nisan] Solving addition using sum-equal:



- At each node solve O(1) sum-equal, to determine if sum of lower halves matters or not.
- Depth of tree = O(log n)
- Naively, for total error 1% need to solve each sum-equal with error ≤ 1/log n ➔ O(log n log log n) protocol



- [Nisan Safra] obtain O(log n) for k=2 players using binary search with noise
- Exploits geometry not present for k > 2
- We show how to use binary search with noise for any k: write sum-equal questions along a path as single question

Summary

2-player addition: $\Theta(\log n)$, improves Smirnoff's '88 $\Omega(\sqrt{\log n})$

k-player addition: $\Theta(\log n)$, improves Nisan's '93 $O(\log^2 n)$

Useful for polynomal-threshold functions,

complexity of pseudorandom functions,

[Dutta Pandurangan Rajaraman Sun V.]

multiplication

Open problems

- For large number k of players:

  We show sum-equal mod p is $\Theta(k \log k)$

  Over integers only know $O(k \log k)$, $\Omega(k)$

- Recall for 2-player addition we gave $O(\log n)$ protocol private-coin and explicit

  Not known for $k > 2$ players.

  One approach would be to derandomize the hash function