# The communication complexity of addition

January 2013

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

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

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

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

Corollary: $\Theta(\log n)$

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

- [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

  Used in [Dutta Pandurangan Rajaraman Sun V.]

- 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 complexity of pseudorandom functions

Outline

- Overview of results

- 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 $\sum_{i \le k} x_i > 0$ 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 $\sum_{i \le k} x_i = 0$ 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

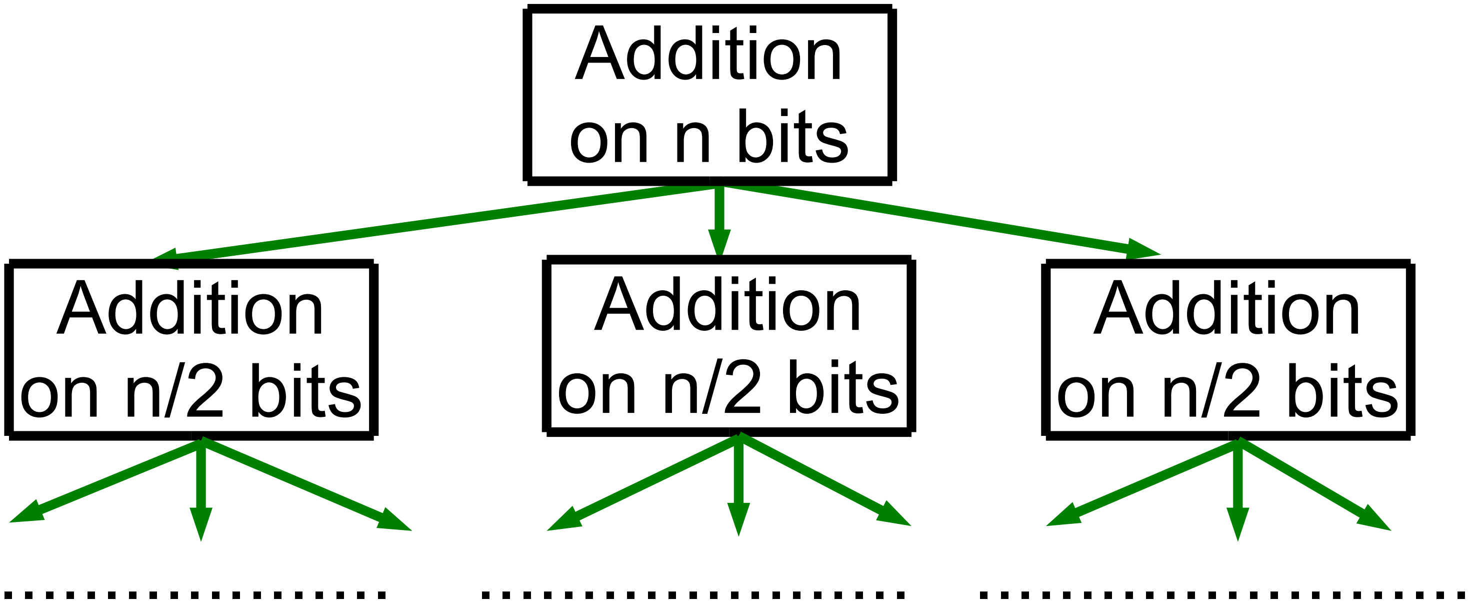
- This work: Use hash function analyzed by [Dietzfelbinger Hagerup Katajainen Penttonen]

  $hash(x_i)$ = "$O(1)$ 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

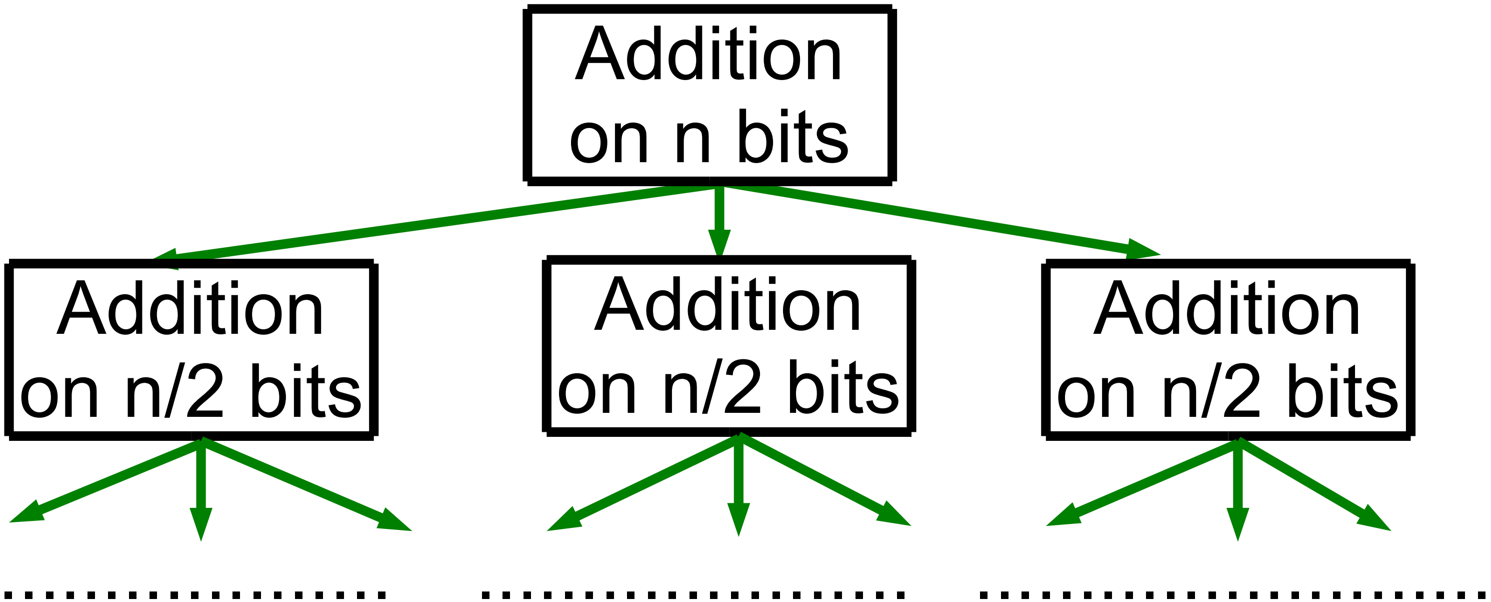
- [Nisan] Solving addition using sum-equal:




- 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.]

## 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