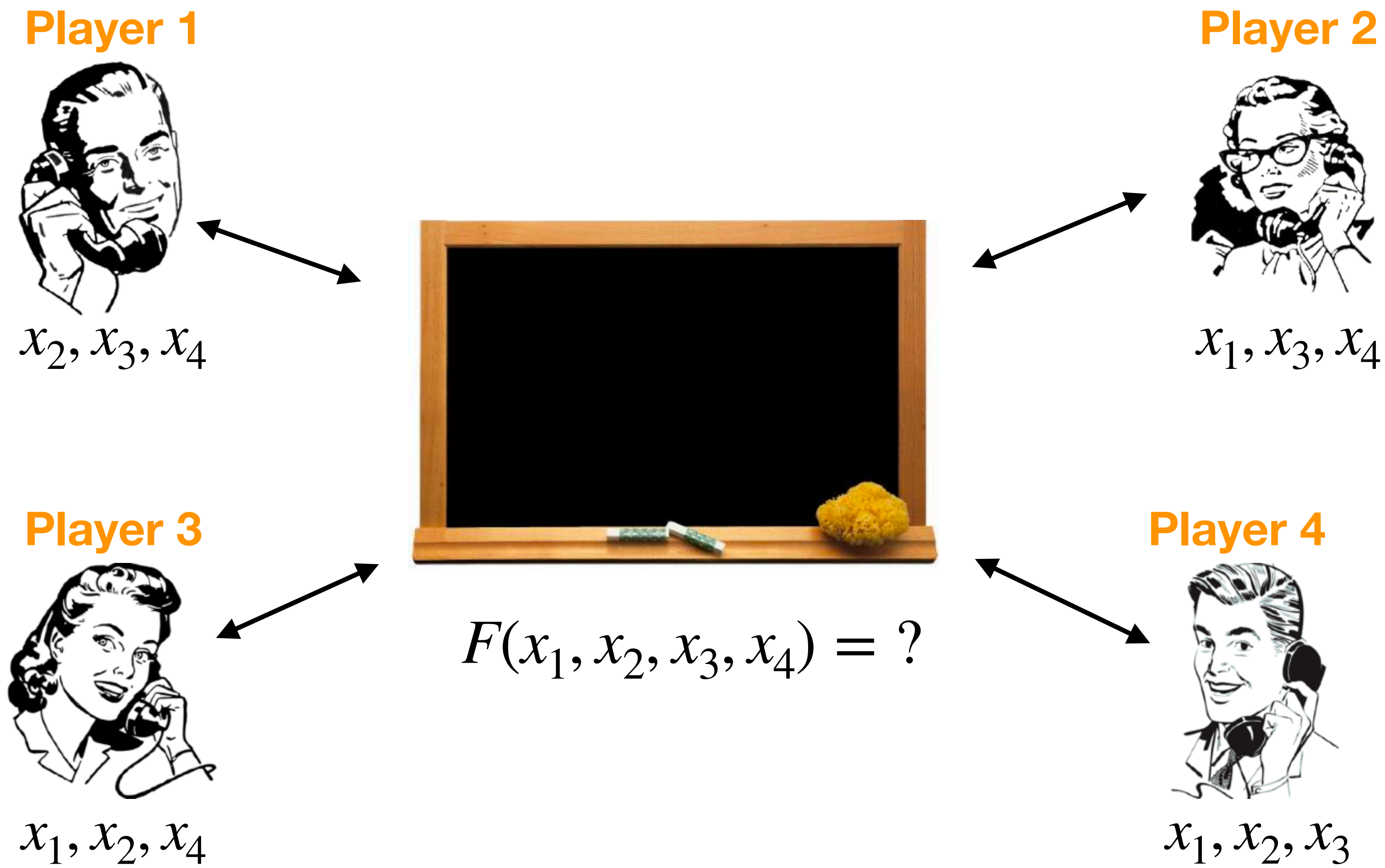
Simultaneous Multiparty Communication Protocols for Composed Functions

Yassine Hamoudi
IRIF, Université Paris Diderot, CNRS

MFCS 2018

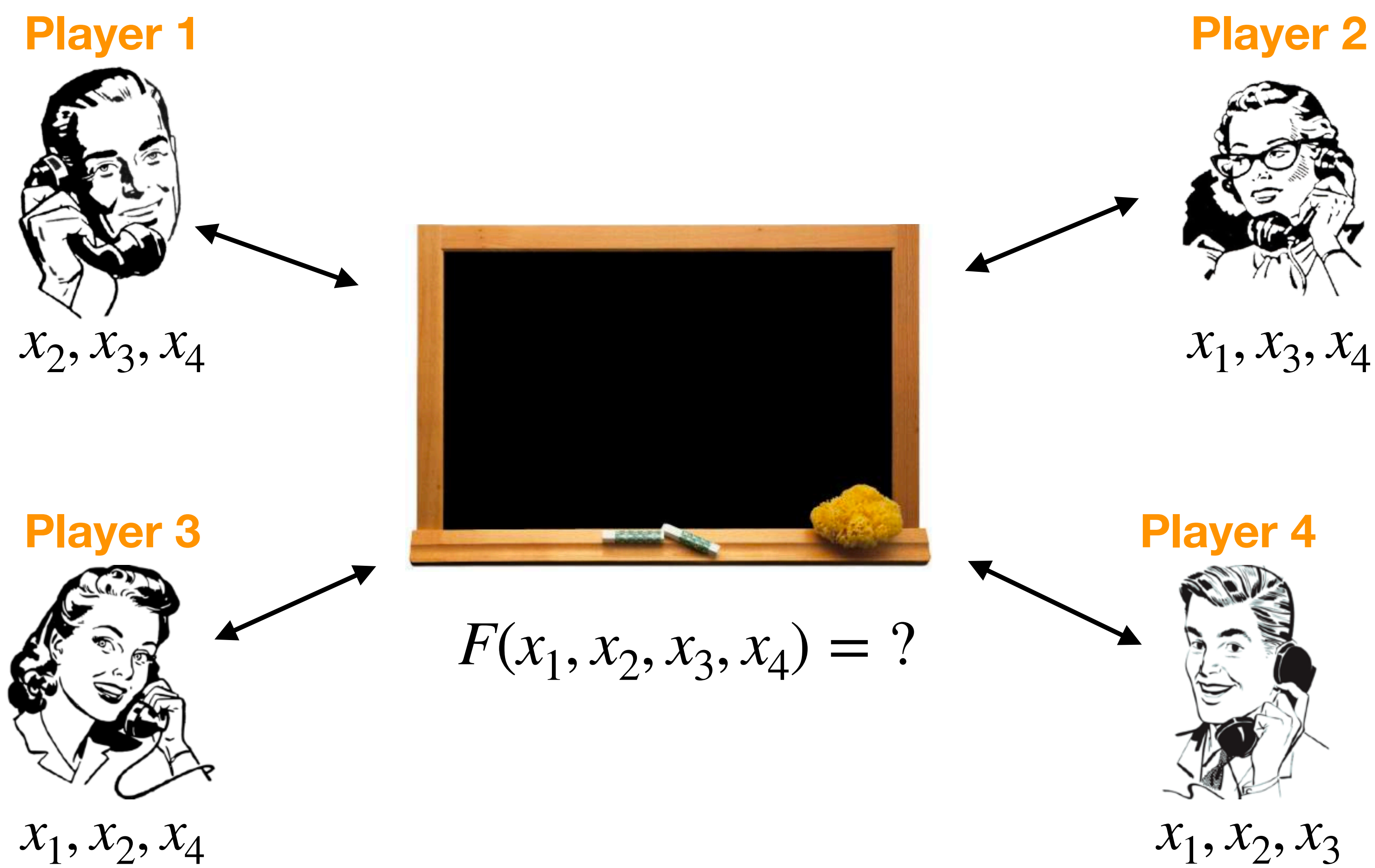
Number-On-Forehead model [Chandra, Furst, Lipton'83]

$$F : X_1 \times \cdots \times X_k \to \{0,1\}$$



Number-On-Forehead model [Chandra, Furst, Lipton'83]

$$F : X_1 \times \cdots \times X_k \to \{0,1\}$$



- **Player i doesn't know x_i (⇔ Number-On-Forehead)**
- **Communicate by broadcasting bits**
- **Players have unlimited computational power**

No randomness in this talk

Examples

$\mathbf{x_1, \ldots, x_n \in \{0,1\}^n}$

An always-O(n) protocol:

- Player 1 sends x_2
- Player 2 sends $F(x_1, \ldots, x_k)$

F is easy / protocol is efficient
$\Leftrightarrow$ communication cost $\log^{O(1)}(n)$

$\mathbf{x_1, \ldots, x_n \in \{0,1\}^n}$

An always-O(n) protocol:

- Player 1 sends x_2
- Player 2 sends $F(x_1, \ldots, x_k)$

F is easy / protocol is efficient
⇔ communication cost $\log^{O(1)}(n)$

Equality: $x_1 = \ldots = x_k$?

Two players: $\Omega(n)$

$k \geq 3$ players: $O(1)$

- Player 1 indicates if $x_2 = \ldots = x_k$
- Player 2 indicates if $x_1 = x_3$

Applications of the NOF model

- Branching programs, Ramsey theory [Chandra, Furst, Lipton'83]
- Quasi-random graphs [Chung, Tetali'93]
- Proof complexity [Beame, Pitassi, Segerlind'07]
- Circuit complexity [Håstad, Goldmann'91] [Razborov, Wigderson'93] [Beigel, Tarui'94]
- Data-structures for dynamic problems [Patrascu'10]

Applications of the NOF model

- Branching programs, Ramsey theory [Chandra, Furst, Lipton'83]
- Quasi-random graphs [Chung, Tetali'93]
- Proof complexity [Beame, Pitassi, Segerlind'07]
- Circuit complexity [Håstad, Goldmann'91] [Razborov, Wigderson'93] [Beigel, Tarui'94]
- Data-structures for dynamic problems [Patrascu'10]

F is hard to compute for
k ≥ log(n) players

[Håstad, Goldmann'91]
[Babai, Gál, Kimmel, Lokam'04]

$\Rightarrow$

F is not in **ACC^0**

communication cost: $(\log n)^{\omega(1)}$

Best lower bounds so far: $\tilde{\Omega}\left(\frac{n}{2^k}\right)$

polysize constant-depth circuits
with AND, OR, NOT, MOD$_m$ gates

Conjecture: MAJORITY $\notin$ ACC^0
Conjecture: NP $\not\subseteq$ ACC^0

Applications of the NOF model

- Branching programs, Ramsey theory [Chandra, Furst, Lipton'83]
- Quasi-random graphs [Chung, Tetali'93]
- Proof complexity [Beame, Pitassi, Segerlind'07]
- Circuit complexity [Håstad, Goldmann'91] [Razborov, Wigderson'93] [Beigel, Tarui'94]
- Data-structures for dynamic problems [Patrascu'10]

F is hard to compute for
$k \geq \log(n)$ players

[Håstad, Goldmann'91]
[Babai, Gál, Kimmel, Lokam'04]

$\Rightarrow$

F is not in **$\mathbf{ACC^0}$**

communication cost: $(\log n)^{\omega(1)}$

polysize constant-depth circuits
with AND, OR, NOT, MOD_m gates

Best lower bounds so far: $\tilde{\Omega}\left(\frac{n}{2^k}\right)$

Conjecture: MAJORITY $\not\in$ ACC^0
Conjecture: NP $\not\subseteq$ ACC^0

Log(n) barrier problem

Find a function that is hard to compute for **log(n)** or more players
in the Number-On-Forehead model.

Applications of the NOF model

- Branching programs, Ramsey theory [Chandra, Furst, Lipton'83]
- Quasi-random graphs [Chung, Tetali'93]
- Proof complexity [Beame, Pitassi, Segerlind'07]
- Circuit complexity [Håstad, Goldmann'91] [Razborov, Wigderson'93] [Beigel, Tarui'94]
- Data-structures for dynamic problems [Patrascu'10]

F is hard to compute for $k \geq \log(n)$ players

communication cost: $(\log n)^{\omega(1)}$

Best lower bounds so far: $\tilde{\Omega}\left(\frac{n}{2^k}\right)$

[Håstad, Goldmann'91]
[Babai, Gál, Kimmel, Lokam'04]

$\Rightarrow$

even in the ***simultaneous*** *NOF model*

F is not in **ACC^0**

polysize constant-depth circuits with AND, OR, NOT, MOD_m gates

Conjecture: MAJORITY $\notin ACC^0$
Conjecture: NP $\not\subseteq ACC^0$

Log(n) barrier problem

Find a function that is hard to compute for **log(n)** or more players in the ***simultaneous*** Number-On-Forehead model.

Simultaneous Number-On-Forehead model

Player 1

x_2, x_3, x_4

Player 2

x_1, x_3, x_4

Player 3

x_1, x_2, x_4

Player 4

x_1, x_2, x_3

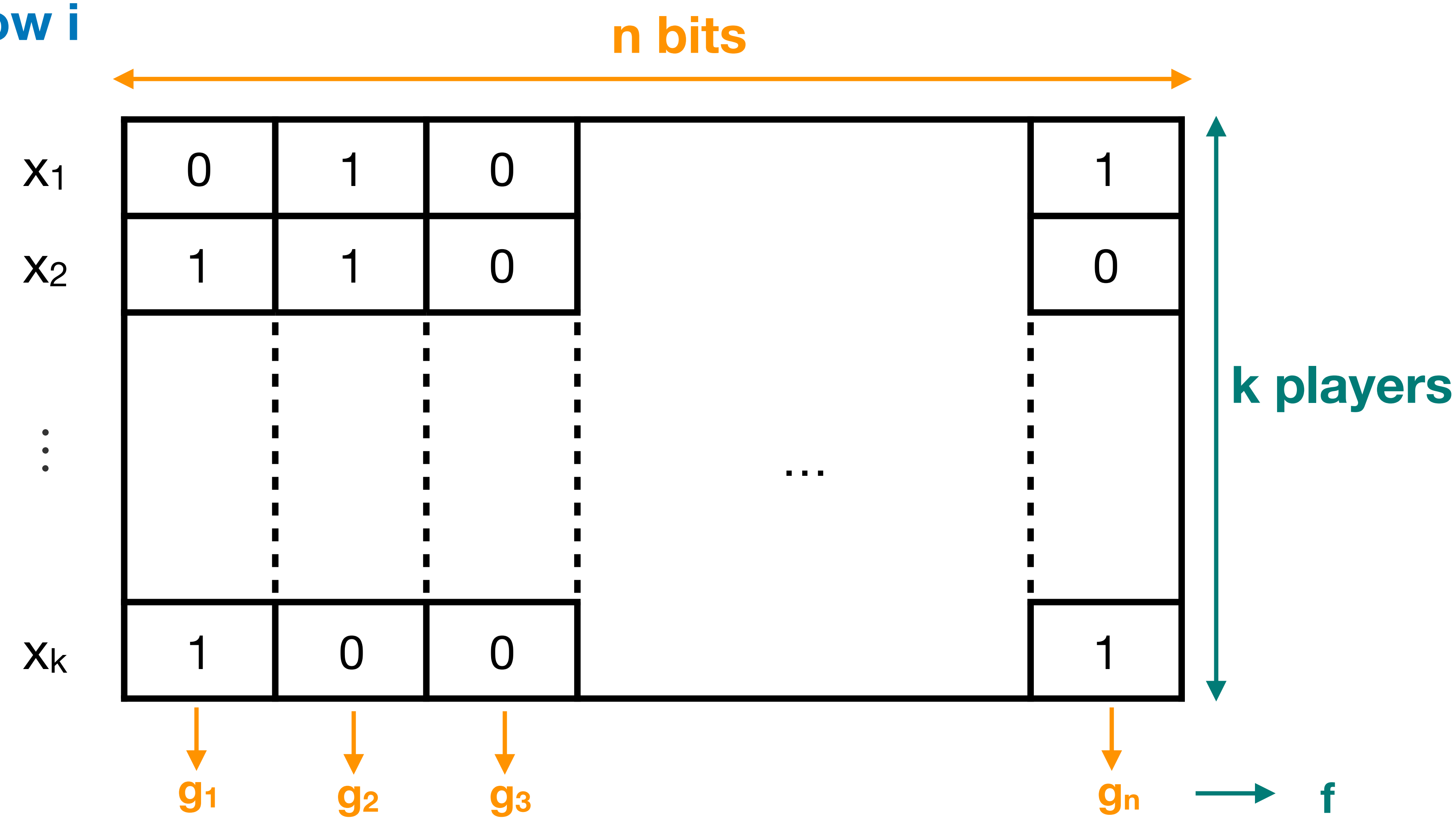
Referee

$F(x_1, x_2, x_3, x_4) = ?$

One-way communication to a referee, no interactions

Candidates to break the log(n) barrier: the composed functions

- **Input: $x_1, \ldots, x_k \in \{0,1\}^n$**
- **Player i doesn't see row i**



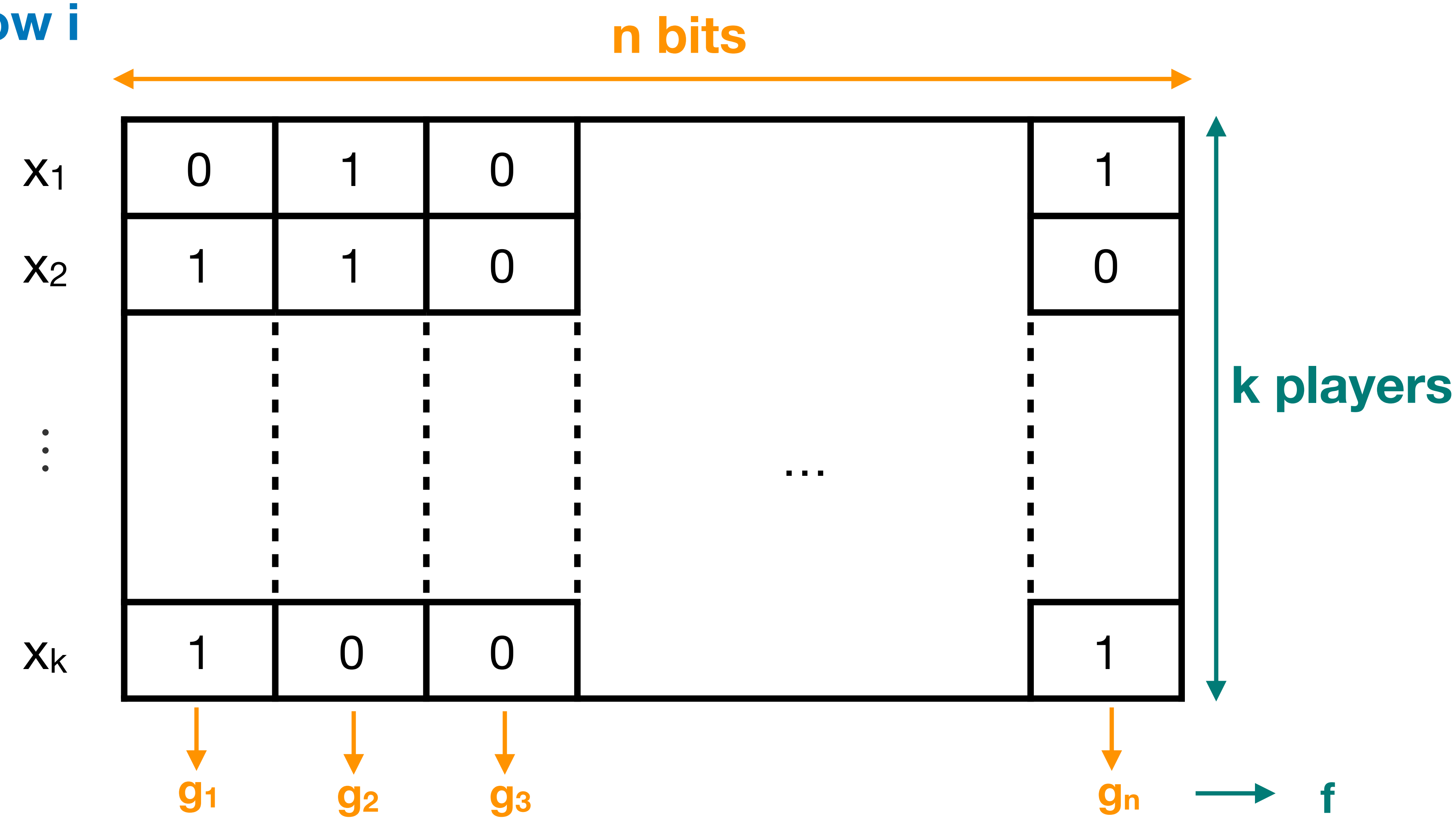
Composed function:

$\mathbf{f} \circ (\mathbf{g_1}, \ldots, \mathbf{g_n})$ where $\mathbf{f} : \{\mathbf{0}, \mathbf{1}\}^{\mathbf{n}} \to \{\mathbf{0}, \mathbf{1}\}$

and $\mathbf{g_j} : \{\mathbf{0}, \mathbf{1}\}^{\mathbf{k}} \to \{\mathbf{0}, \mathbf{1}\}$

Candidates to break the log(n) barrier: the composed functions

- **Input: $x_1, \ldots, x_k \in \{0,1\}^n$**
- **Player i doesn't see row i**



Composed function:

$\mathbf{f} \circ (\mathbf{g_1}, \ldots, \mathbf{g_n})$ where $\mathbf{f} : \{\mathbf{0}, \mathbf{1}\}^{\mathbf{n}} \to \{\mathbf{0}, \mathbf{1}\}$

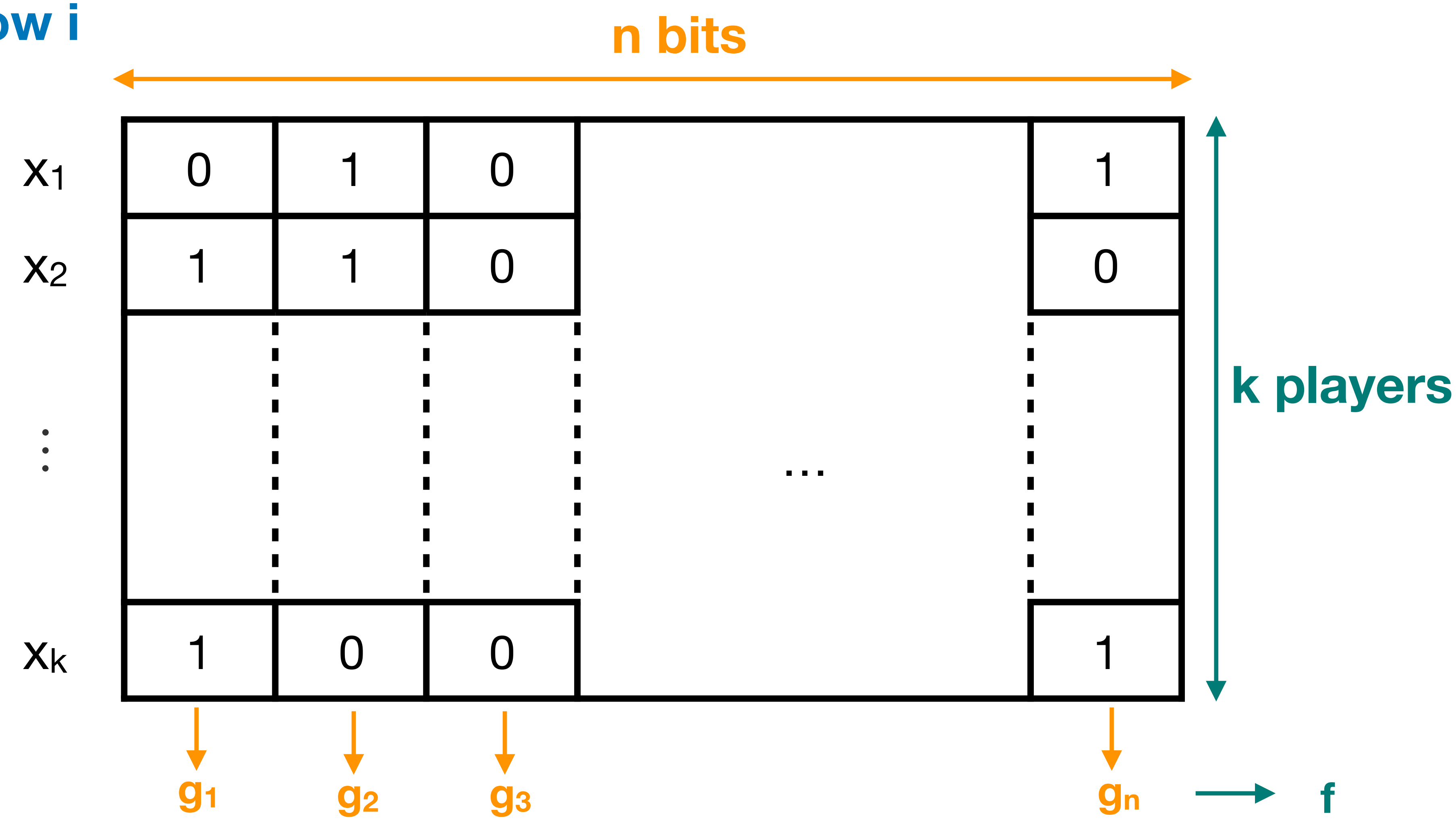
and $\mathbf{g_j} : \{\mathbf{0}, \mathbf{1}\}^{\mathbf{k}} \to \{\mathbf{0}, \mathbf{1}\}$

Examples:

- Generalized Inner Product: $MOD_2 \circ (AND, \ldots, AND)$
- Disjointness: $OR \circ (AND, \ldots, AND)$
- Majority of Majority: $MAJ \circ (MAJ, \ldots, MAJ)$

Candidates to break the log(n) barrier: the composed functions

- **Input: $x_1, \ldots, x_k \in \{0,1\}^n$**
- **Player i doesn't see row i**



Composed function:

$\mathbf{f} \circ (\mathbf{g_1}, \ldots, \mathbf{g_n})$ where $\mathbf{f} : \{\mathbf{0}, \mathbf{1}\}^{\mathbf{n}} \to \{\mathbf{0}, \mathbf{1}\}$

and $\mathbf{g_j} : \{\mathbf{0}, \mathbf{1}\}^{\mathbf{k}} \to \{\mathbf{0}, \mathbf{1}\}$

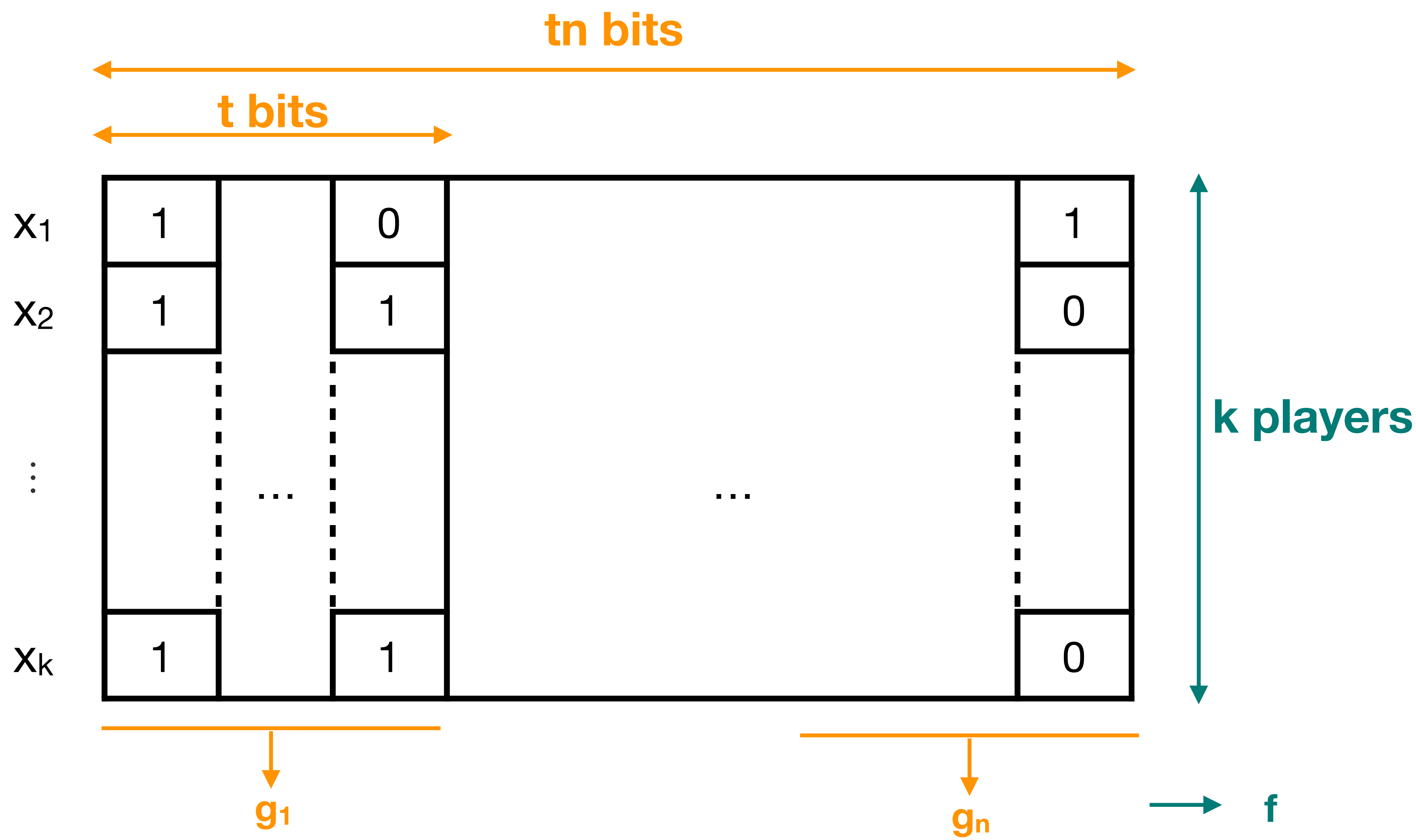
Examples:

- Generalized Inner Product: $MOD_2 \circ (AND,\ldots,AND)$
- Disjointness: $OR \circ (AND,\ldots,AND)$
- Majority of Majority: $MAJ \circ (MAJ,\ldots,MAJ)$

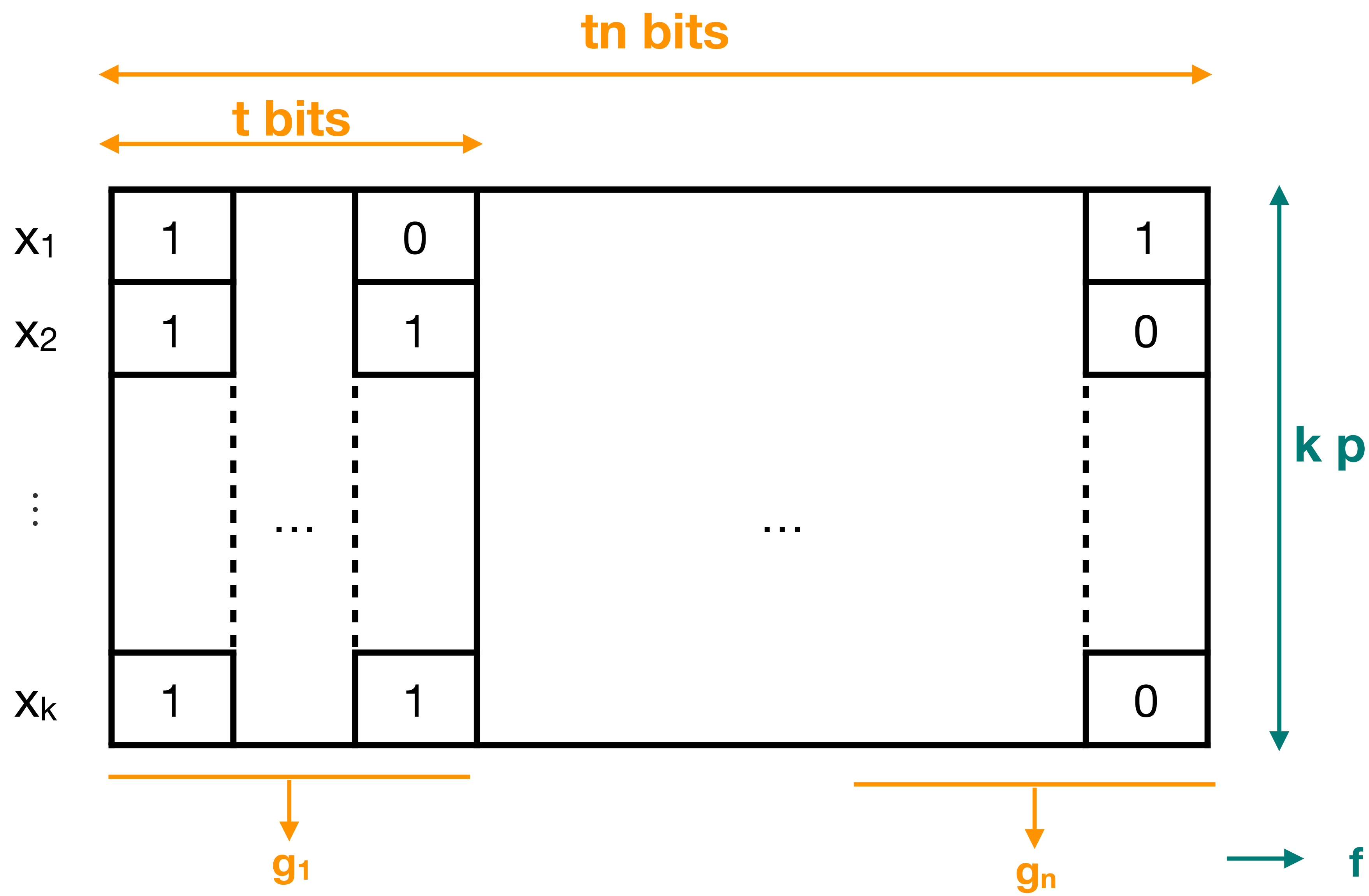
There is an efficient **simultaneous** protocol for $f \circ (g_1,\ldots,g_n)$ when **f is symmetric** and $k \geq \Omega(\log n)$.

[Grolmusz'94]
[Babai, Gál, Kimmel, Lokam'04]
[Ada, Chattopadhyay, Fawzi, Nguyen'15]

Block-composed functions



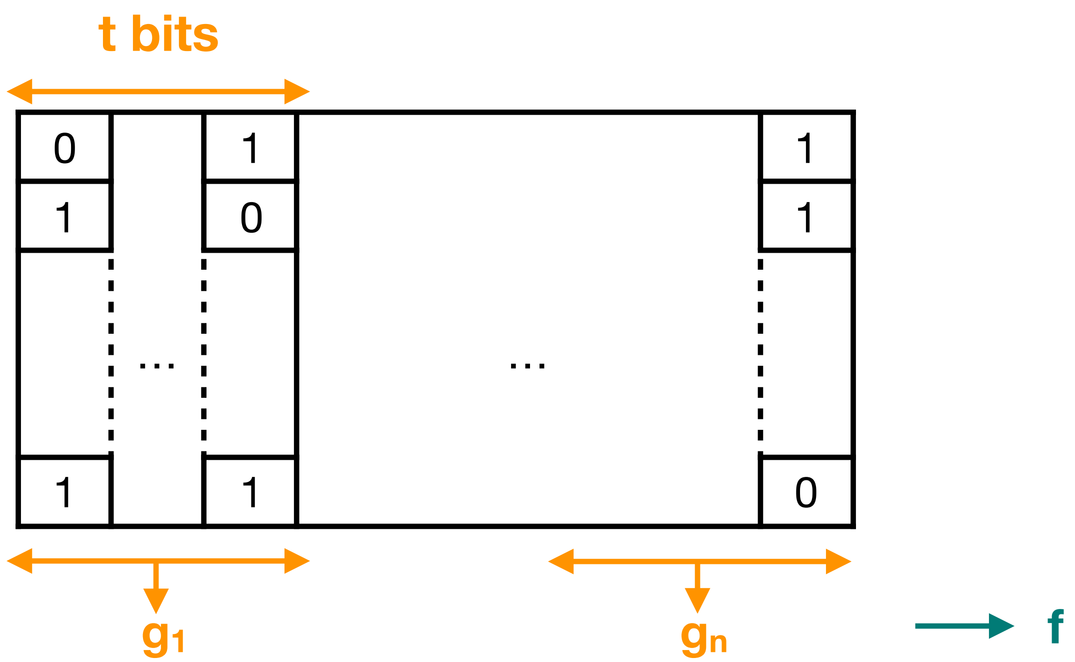
Block-composed functions



Conjecture [Babai et. al.'04] The **simultaneous** communication cost of **MAJ ∘ (MAJ,…,MAJ)** is $(\log n)^{\omega(1)}$ for $t \geq \sqrt{n}$ and $k \geq \Omega(\log n)$.

Unknown even for t = 2

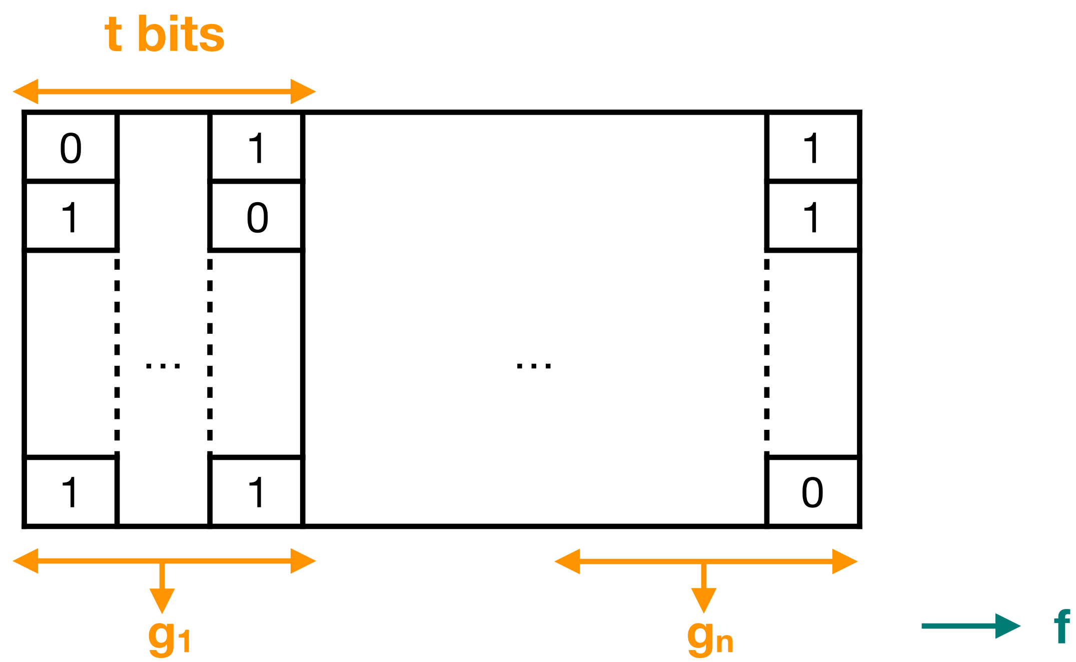
Our result



Theorem: If **t is constant**, there is an efficient **simultaneous** protocol for $f \circ (g_1,\ldots,g_n)$ when f, $g_1, \ldots, g_n$ are **symmetric** and $k \geq \Omega(\log n)$.

MAJ ∘ (MAJ,…,MAJ) cannot break the log n barrier for constant t

Our result

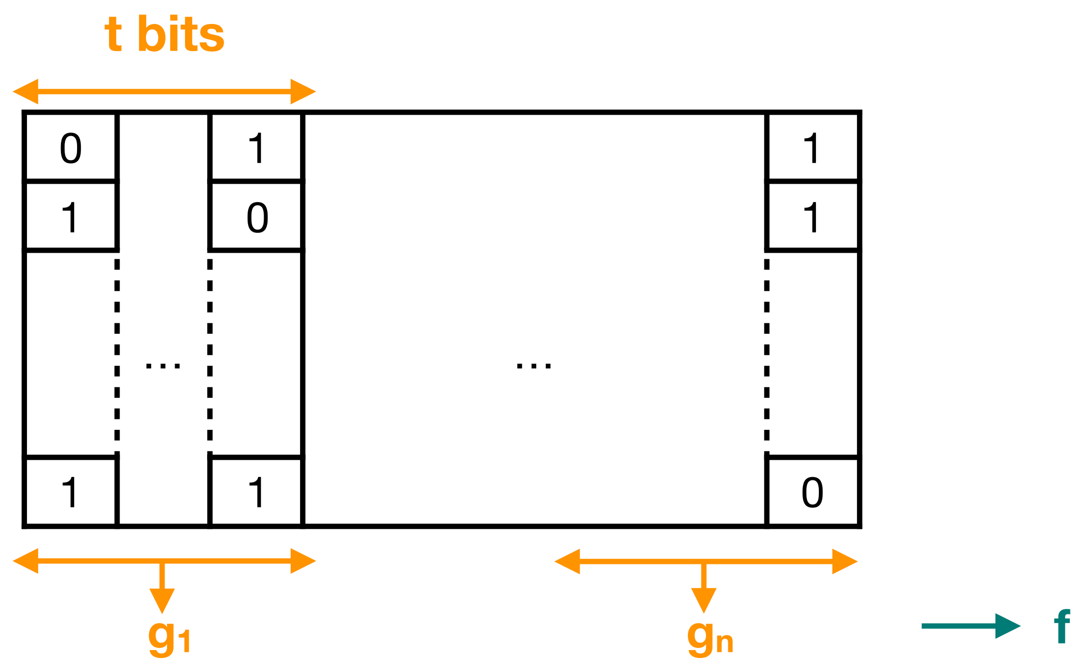


Theorem: If **t is constant**, there is an efficient **simultaneous** protocol for $f \circ (g_1,\ldots,g_n)$ when f, $g_1, \ldots, g_n$ are **symmetric** and $k \geq \Omega(\log n)$.

MAJ ∘ (MAJ,…,MAJ) cannot break the log n barrier for constant t

	Block-width t	Model	Conditions
[Ada, Chattopadhyay, Fawzi, Nguyen'15]	1	simultaneous	f symmetric
[Chattopadhyay, Saks'14]	log log n	non-simultaneous	f symmetric
[Chattopadhyay, Saks'14]	log n	non-simultaneous	f, $g_1, \ldots, g_n$ symmetric
Our result	constant	simultaneous	f, $g_1, \ldots, g_n$ symmetric

Our result



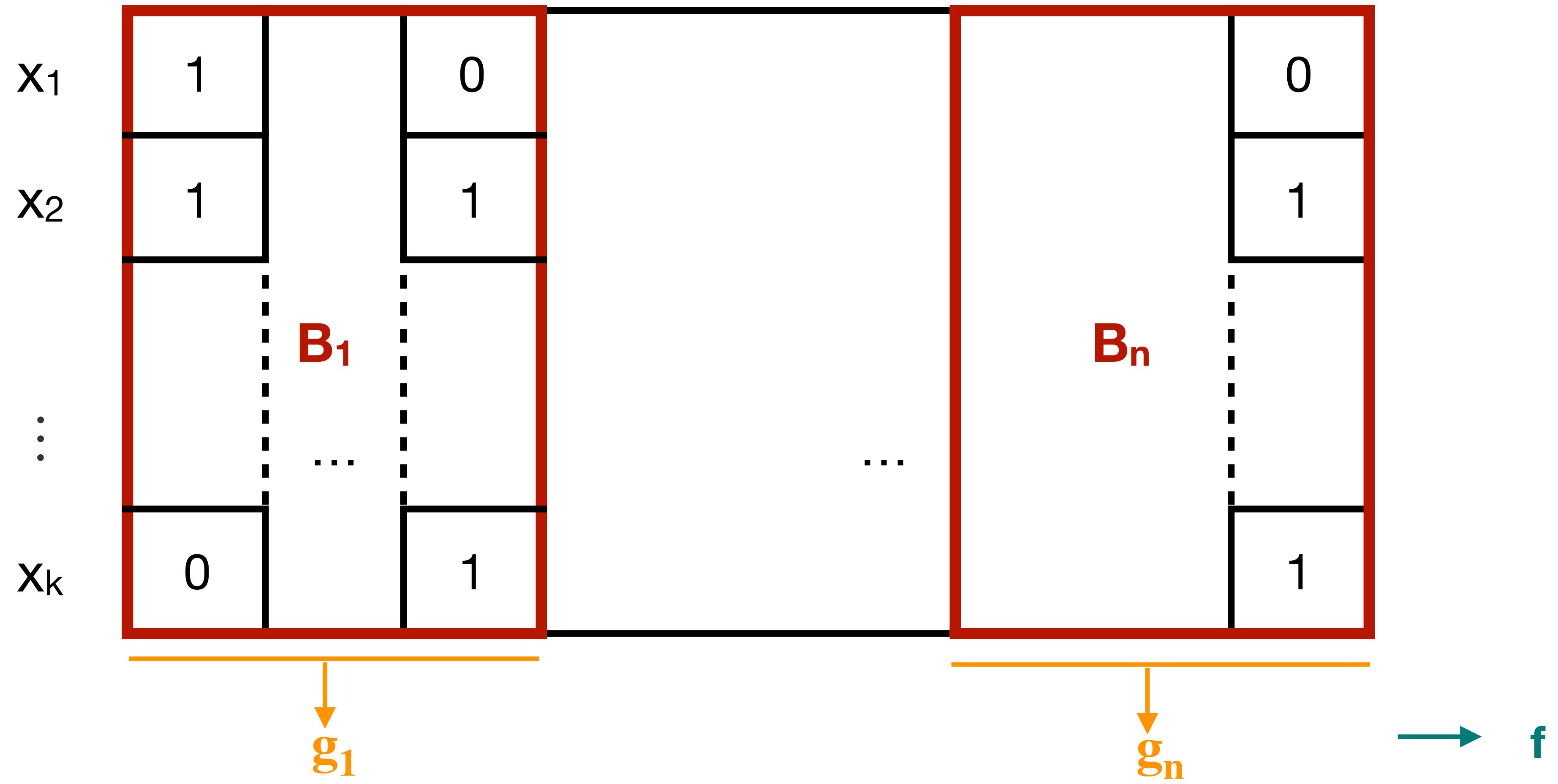
Theorem: If **t is constant**, there is an efficient **simultaneous** protocol for $f \circ (g_1,\ldots,g_n)$ when f, $g_1, \ldots, g_n$ are **symmetric** and $k \geq \Omega(\log n)$.

MAJ ○ (MAJ,…,MAJ) cannot break the log n barrier for constant t

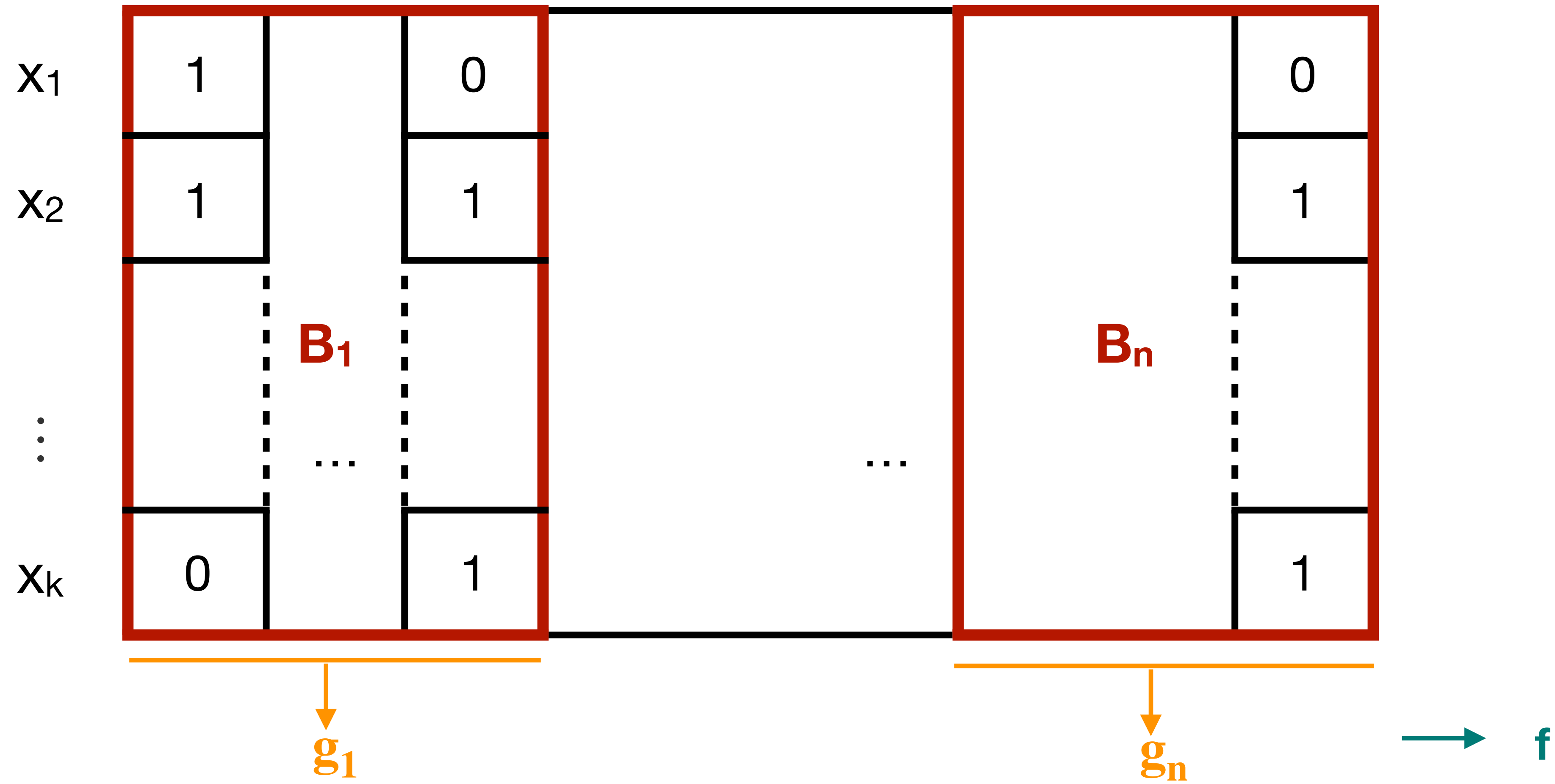
Roadmap (when $k = \Theta(\log n)$)**:**

1. Reduce to the case of equal inner functions $g = g_1 = \ldots = g_n$
2. Simultaneous protocol for $f \circ (g,\ldots,g)$ with a generalization of [Babai et. al.'04]

Step 1: equal inner functions

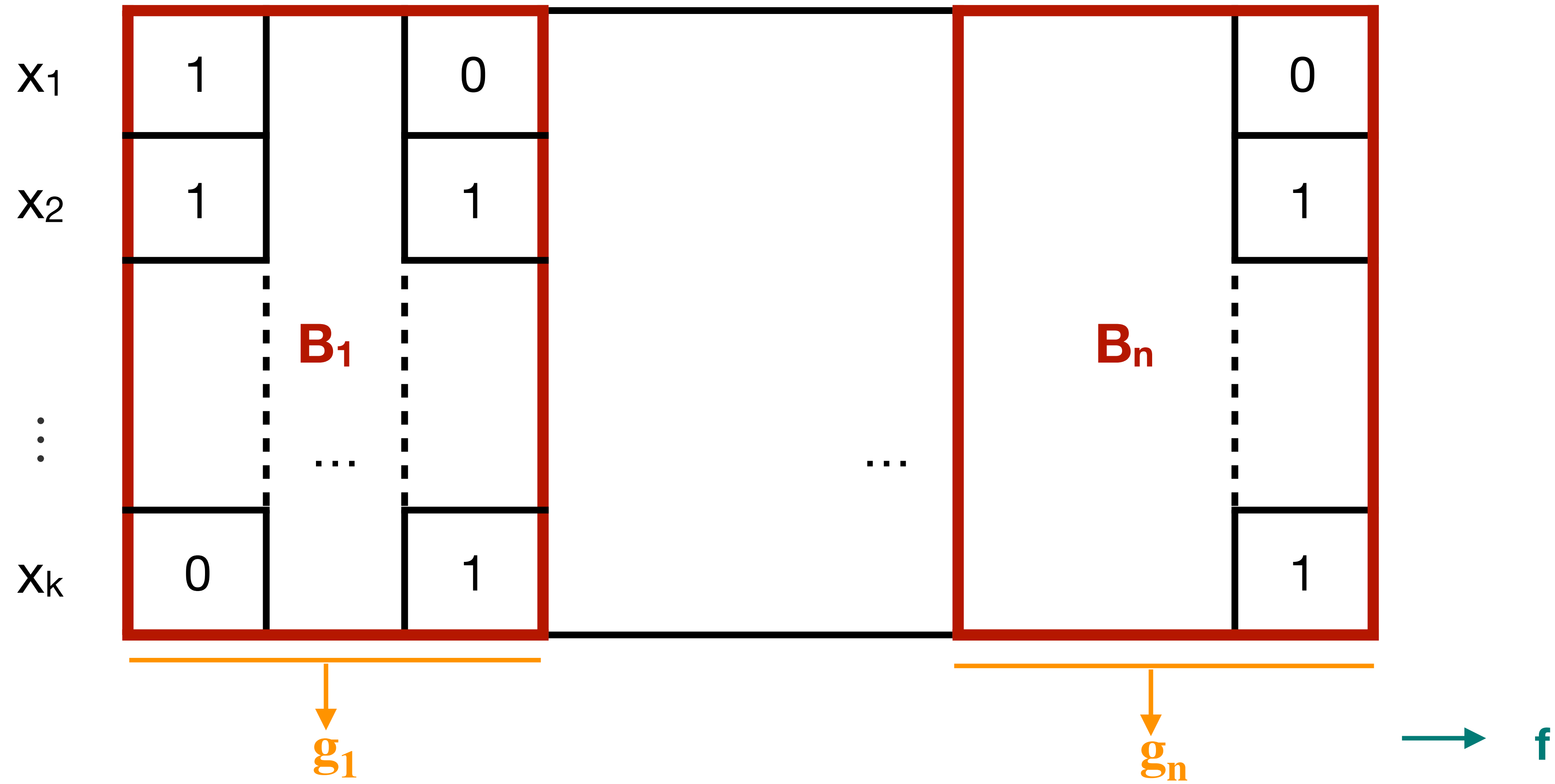


Step 1: equal inner functions



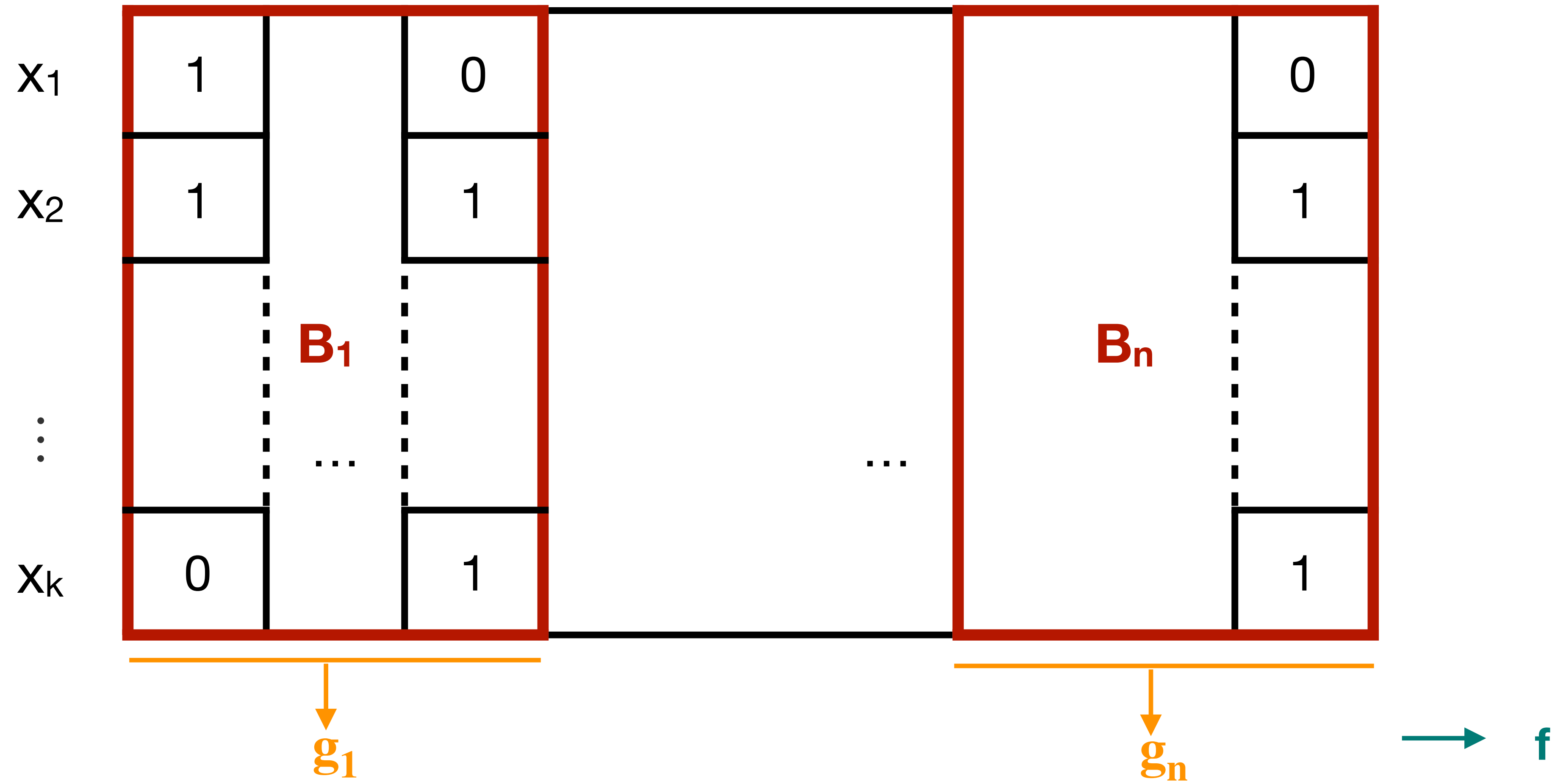
1. Each $\mathbf{g_j}$ is decomposed in a basis of symmetric functions: $\mathbf{g_j(x) = \sum_a c_a(g_j) \cdot m_a(x)}$

Step 1: equal inner functions



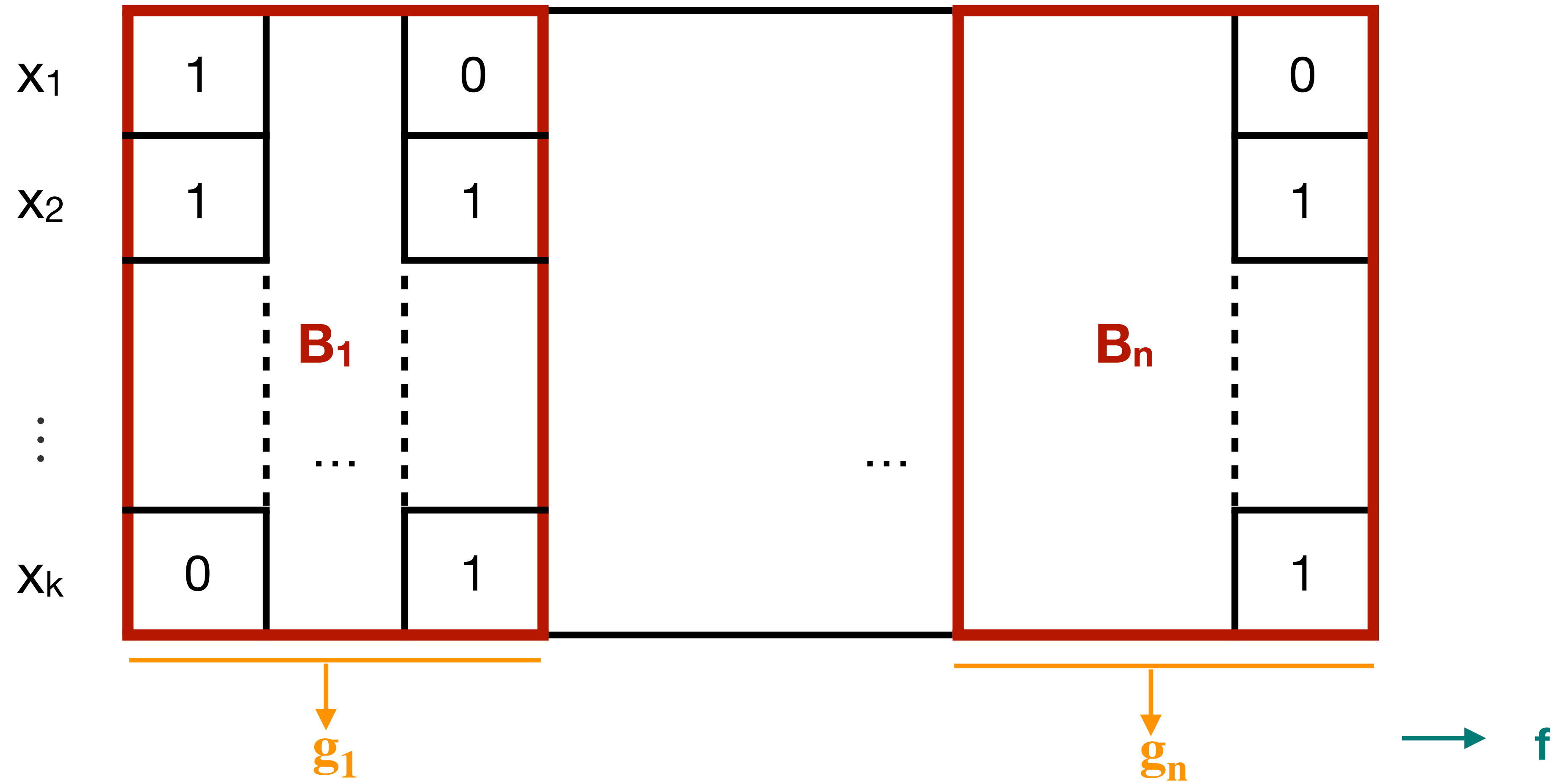
1. Each $\mathbf{g_j}$ is decomposed in a basis of symmetric functions: $\mathbf{g_j(x) = \sum_a c_a(g_j) \cdot m_a(x)}$
2. For each basis element m_a, define the matrix M_a where each B_j is repeated $\mathbf{c_a(g_j)}$ times.

Step 1: equal inner functions



1. Each $\mathbf{g_j}$ is decomposed in a basis of symmetric functions: $\mathbf{g_j(x) = \sum_a c_a(g_j) \cdot m_a(x)}$
2. For each basis element m_a, define the matrix M_a where each B_j is repeated $\mathbf{c_a(g_j)}$ times.
3. For each m_a, compute **SUM○(m_a,…,m_a)** on M_a.

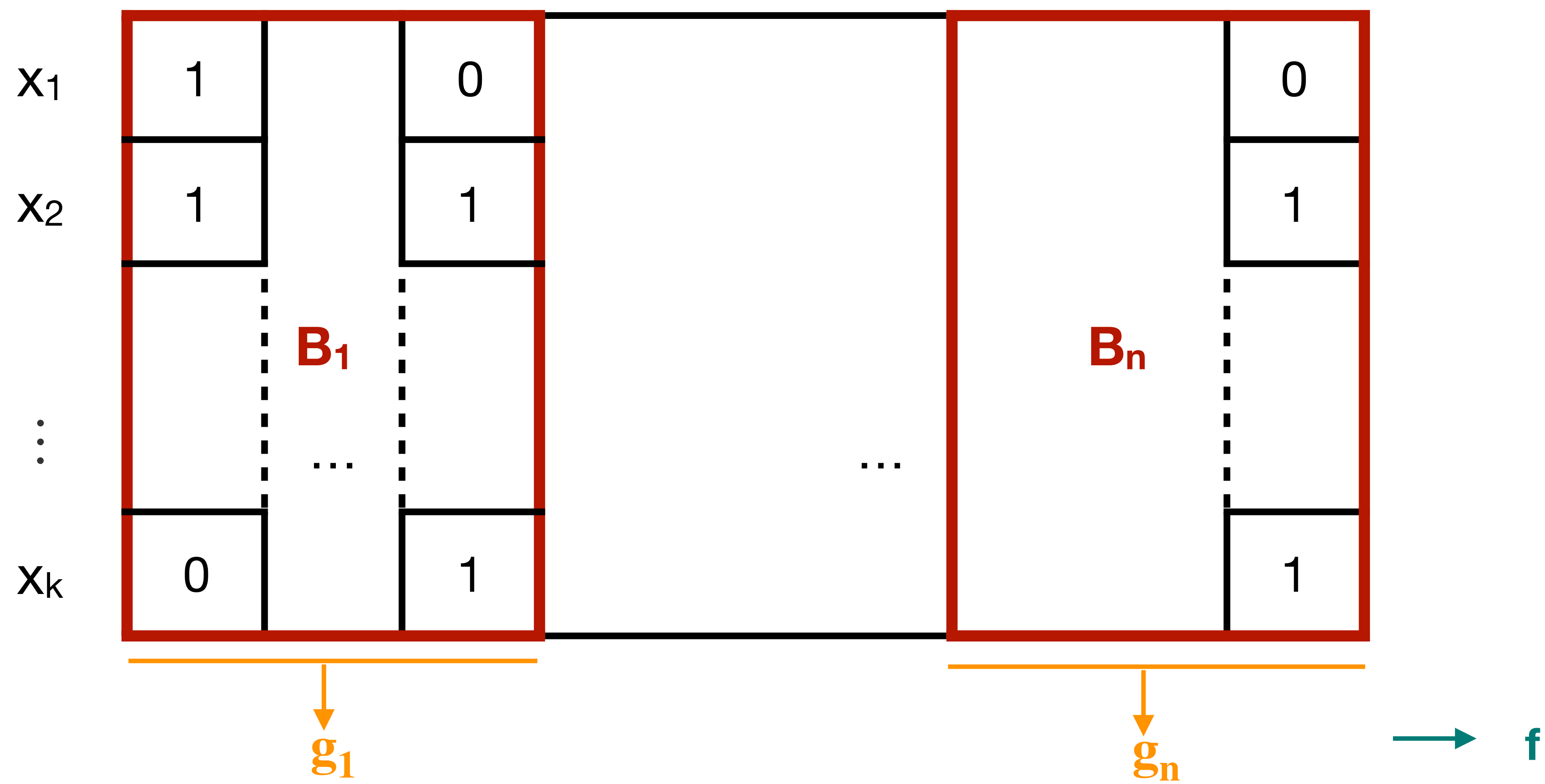
Step 1: equal inner functions



1. Each $\mathbf{g_j}$ is decomposed in a basis of symmetric functions: $\mathbf{g_j(x) = \sum_a c_a(g_j) \cdot m_a(x)}$
2. For each basis element m_a, define the matrix M_a where each B_j is repeated $\mathbf{c_a(g_j)}$ times.
3. For each m_a, compute $\mathbf{SUM \circ (m_a, \ldots, m_a)}$ on M_a.

$$\sum_a \mathrm{SUM} \circ (m_a, \ldots, m_a)(M_a) = \sum_a \sum_j c_a(g_j) \cdot m_a(B_j) = \sum_j g_j(B_j)$$

Step 1: equal inner functions



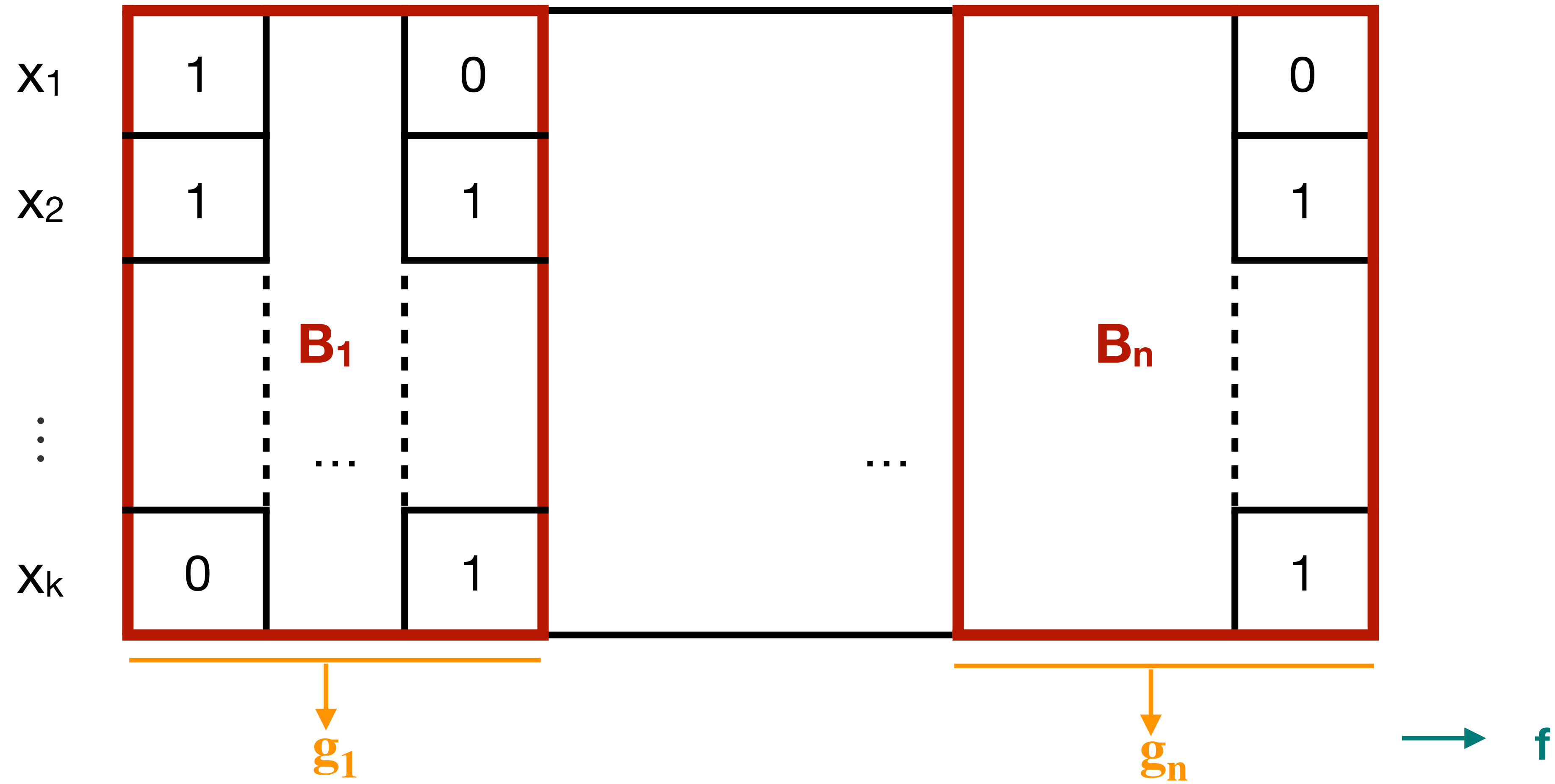
1. Each $\mathbf{g_j}$ is decomposed in a basis of symmetric functions: $\mathbf{g_j(x) = \sum_a c_a(g_j) \cdot m_a(x)}$

2. For each basis element m_a, define the matrix M_a where each B_j is repeated $\mathbf{c_a(g_j)}$ times.

3. For each m_a, compute **SUM○(m_a,…,m_a)** on M_a.

$$\sum_a \mathbf{SUM} \circ (\mathbf{m_a}, \ldots, \mathbf{m_a})(\mathbf{M_a}) = \sum_a \sum_j \mathbf{c_a(g_j) \cdot m_a(B_j)} = \sum_j \mathbf{g_j(B_j)}$$

Enough to compute $\mathbf{f} \circ (\mathbf{g_1}, \ldots, \mathbf{g_n})$ since f is **symmetric.**



1. Each $\mathbf{g_j}$ is decomposed in a basis of symmetric functions: $\mathbf{g_j(x) = \sum_a c_a(g_j) \cdot m_a(x)} \mod \mathbf{p}$

   for prime p ∈ (n,2n)

2. For each basis element m_a, define the matrix M_a where each B_j is repeated $\mathbf{c_a(g_j)}$ times.

   Size ≤ $k \times n^2$

3. For each m_a, compute **SUM○(m_a,…,m_a)** on M_a.

$$\sum_a \mathbf{SUM} \circ (\mathbf{m_a}, \ldots, \mathbf{m_a})(\mathbf{M_a}) = \sum_a \sum_j \mathbf{c_a(g_j) \cdot m_a(B_j)} = \sum_j \mathbf{g_j(B_j)}$$

Enough to compute $\mathbf{f} \circ (\mathbf{g_1}, \ldots, \mathbf{g_n})$ since f is **symmetric.**

Step 3: equation solving protocol

[Babai, Gál, Kimmel, Lokam'04] Protocol for **t = 1**:

x_1	0	1	1	1	1	1	1
x_2	1	1	1	0	0	1	0
x_3	1	0	1	1	0	0	0
x_4	0	1	1	0	1	0	0
x_5	1	1	1	0	1	1	0

$y_0 = 0$
$y_1 = 1$
$y_2 = 1$
$y_3 = 3$
$y_4 = 1$
$y_5 = 1$

- For a k×n matrix M, define $y(M) = (y_0,\ldots,y_k)$ where **y_i = #columns with exactly i 1's.**

→ Knowing $y(M)$ is enough to compute $f\circ(g,\ldots,g)$ when **f and g** are **symmetric**.

Step 3: equation solving protocol

[Babai, Gál, Kimmel, Lokam’04] Protocol for **$t = 1$**:

x_1	0	1	1	1	1	1	1
x_2	1	1	1	0	0	1	0
x_3	1	0	1	1	0	0	0
x_4	0	1	1	0	1	0	0
x_5	1	1	1	0	1	1	0

M_1

$y_0 = 0$
$y_1 = 1$
$y_2 = 1$
$y_3 = 3$
$y_4 = 1$
$y_5 = 1$

- For a k×n matrix M, define $y(M) = (y_0,\ldots,y_k)$ where **y_i = #columns with exactly i 1's.**

  → Knowing y(M) is enough to compute $f\circ(g,\ldots,g)$ when **f and g** are **symmetric**.

- For each $1 \le i \le k$, **Player i sends $y(M_i)$** where M_i is the submatrix seen by Player i.

  → This will be the only communication part of our protocol.

Step 3: equation solving protocol

[Babai, Gál, Kimmel, Lokam’04] Protocol for **t = 1**:

x_1	0	1	1	1	1	1	1
x_2	1	1	1	0	0	1	0
x_3	1	0	1	1	0	0	0
x_4	0	1	1	0	1	0	0
x_5	1	1	1	0	1	1	0

M_1

$y_0 = 0$
$y_1 = 1$
$y_2 = 1$
$y_3 = 3$
$y_4 = 1$
$y_5 = 1$

- For a k×n matrix M, define $y(M) = (y_0,\dots,y_k)$ where **y_i = #columns with exactly i 1's.**

  → Knowing y(M) is enough to compute $f\circ(g,\dots,g)$ when **f and g** are **symmetric**.

- For each $1 \le i \le k$, **Player i sends $y(M_i)$** where M_i is the submatrix seen by Player i.

  → This will be the only communication part of our protocol.

- Using $y(M_1), \dots, y(M_k)$, one can define an equation whose **only integral solution** is y(M).

  → The referee computes y(M) and then $f\circ(g,\dots,g)$.

Step 3: equation solving protocol

[Babai, Gál, Kimmel, Lokam'04] Protocol for **t = 1**:

x_1	0	1	1	1	1	1	1
x_2	1	1	1	0	0	1	0
x_3	1	0	1	1	0	0	0
x_4	0	1	1	0	1	0	0
x_5	1	1	1	0	1	1	0

$\mathbf{M_1}$

$y_0 = 0$
$y_1 = 1$
$y_2 = 1$
$y_3 = 3$
$y_4 = 1$
$y_5 = 1$

We generalize this protocol to $t > 1$, and show that the corresponding equation admits exactly one integral solution when $k \geq \Omega(\log n)$.

Conclusion

Our result: MAJ ○ (MAJ,…,MAJ) cannot break the log n barrier for any **constant t**

(in fact, any symmetric $f \circ (g_1,\ldots,g_n)$)

Future directions:

- Efficient simultaneous protocol for non-constant t and/or non-symmetric $g_1,\ldots,g_n$
- Strong lower bound for $k \geq \log n$ players

  → only general method known: discrepancy method and its variants

  → [Podolskii, Sherstov'17]: first $\omega(1)$ lower bound when $k \geq \log n$ for explicit function

arXiv: 1710.01969

Step 0: decreasing the number of players to k = Θ(log n)

Lemma: If $g : Y_1,\ldots,Y_k \to Y$ is symmetric then

$$g'(y_1,\ldots,y_{k'}) = g(\underbrace{y_1,\ldots,y_{k'}}_{\text{variables}},\ \underbrace{y_{k'+1},\ldots,y_k}_{\text{any fixed values}})$$

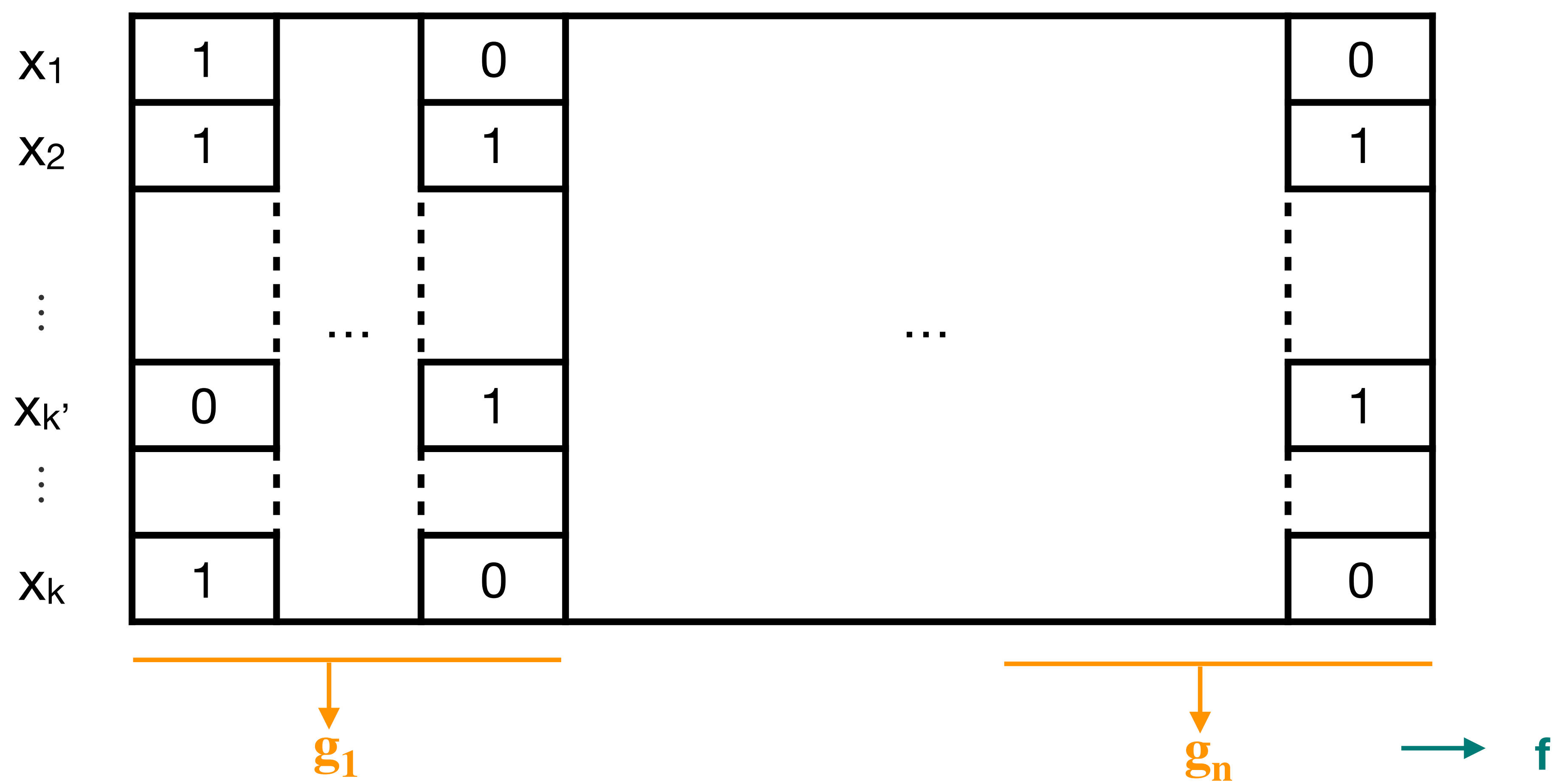
is symmetric.

Step 0: decreasing the number of players to k = Θ(log n)

Lemma: If $g : Y_1,\dots,Y_k \to Y$ is symmetric then

$$g'(y_1,\dots,y_{k'}) = g(\underbrace{y_1,\dots,y_{k'}}_{\text{variables}},\ \underbrace{y_{k'+1},\dots,y_k}_{\text{any fixed values}})$$

is symmetric.



Step 0: decreasing the number of players to k = Θ(log n)

Lemma: If $g : Y_1,\dots,Y_k \to Y$ is symmetric then

$$g'(y_1,\dots,y_{k'}) = g(\underbrace{y_1,\dots,y_{k'}}_{\text{variables}},\ \underbrace{y_{k'+1},\dots,y_k}_{\text{any fixed values}})$$

is symmetric.