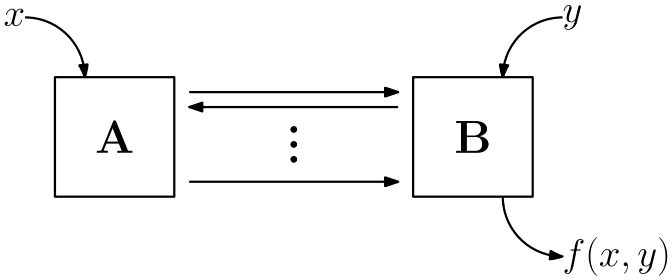
The communication complexity of functions with large outputs

Lila Fontes [1], Sophie Laplante [2], Mathieu Laurière [3], and Alexandre Nolin [4]

[1] Swarthmore College [2] Université Paris Cité [3] NYU Shanghai [4] CISPA

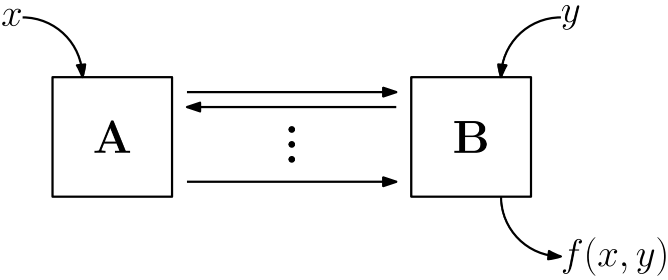
SIROCCO 2023

2-party communication complexity [Yao79]



$$x, y \in \{0,1\}^n, \qquad f(x,y) \in \{0,1\}^k$$

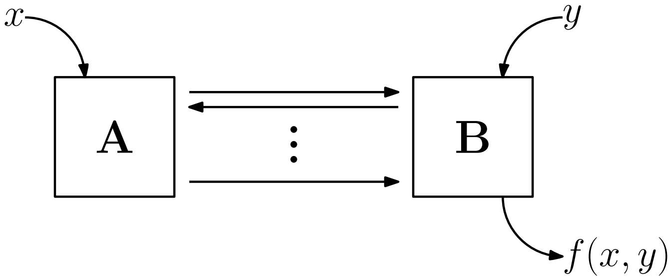
2-party communication complexity [Yao79]



$$x, y \in \{0,1\}^n, \qquad f(x,y) \in \{0,1\}^k$$

We only charge for the amount of communication.

2-party communication complexity [Yao79]



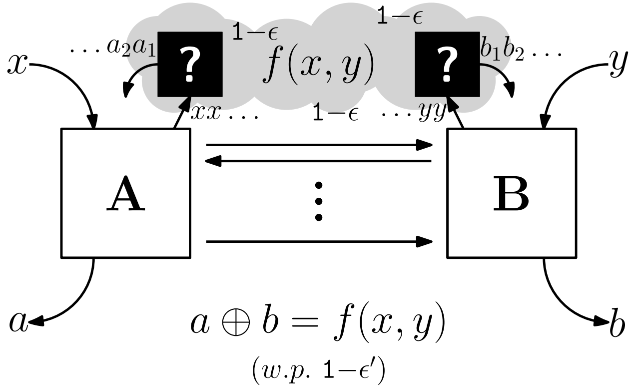
$$x, y \in \{0,1\}^n, \qquad f(x,y) \in \{0,1\}^k$$

We only charge for the amount of communication.

$$\underbrace{R_\epsilon(f)}_{\text{public-coin randomized complexity}} \leq \underbrace{D(f)}_{\text{deterministic complexity}} \in [0, n]$$

A riddle

Given a black box which computes a function f with error ϵ in the XOR model, how much do you need to communicate to compute f with error $\epsilon' < \epsilon$?



First remark: correctness of blackboxes?

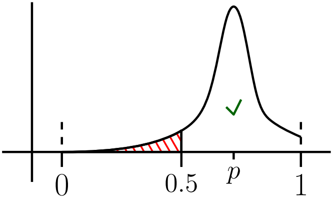
A single box / in expectation: correct w.p. $\geq 1 - \epsilon$.

Probability of a correct majority?

Probability of a correct constant fraction?

(Chernoff bound)

$$\Pr\left[\left|\frac{1}{t}\sum_{i=1}^{t} X_i - p\right| \geq \delta\right] \leq e^{-\frac{\delta^2 n}{2t(1-p)}}$$



First remark: correctness of blackboxes?

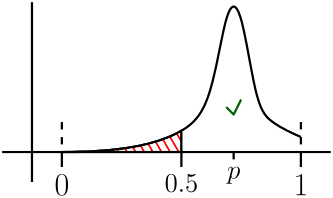
A single box / in expectation: correct w.p. $\geq 1 - \epsilon$.

Probability of a correct majority? $\geq 1 - \exp\left(-\Omega\left(\frac{t}{(1/2-\epsilon)^2}\right)\right)$

Probability of a correct constant fraction?

(Chernoff bound)

$$\Pr\left[\left|\frac{1}{t}\sum_{i=1}^{t} X_i - p\right| \geq \delta\right] \leq e^{-\frac{\delta^2 n}{2t(1-p)}}$$



First remark: correctness of blackboxes?

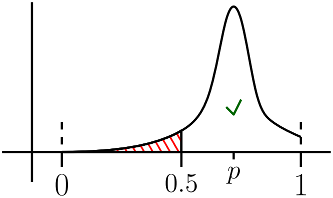
A single box / in expectation: correct w.p. $\geq 1 - \epsilon$.

Probability of a correct majority? $\geq 1 - \exp\left(-\Omega\left(\frac{t}{(1/2-\epsilon)^2}\right)\right)$

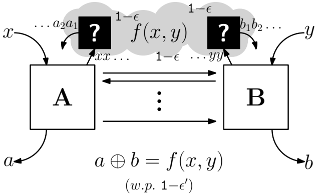
Probability of a correct constant fraction? $\geq 1 - \exp(-\Omega(t))$

(Chernoff bound)

$$\Pr\left[\left|\frac{1}{t}\sum_{i=1}^{t} X_i - p\right| \geq \delta\right] \leq e^{-\frac{\delta^2 n}{2t(1-p)}}$$



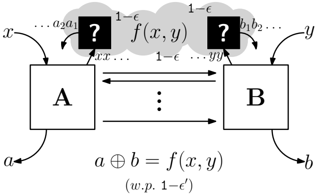
1st answer to the riddle



1. Use the black boxes $C_{\epsilon,\epsilon'} \in \Theta\left(\frac{\epsilon \cdot \ln\left(\frac{1}{\epsilon'}\right)}{\left(\frac{1}{2}-\epsilon\right)^2}\right)$ times, store results,
2. Alice sends all her a_i's to Bob,
3. Bob finds most common value $z \in \{0,1\}^k$ for $a_i \oplus b_i$.
4. Alice outputs the all-0 k-bit string, Bob outputs z.

Complexity: $C_{\epsilon,\epsilon'} \cdot k$.

1st answer to the riddle



1. Use the black boxes $C_{\epsilon,\epsilon'} \in \Theta\left(\frac{\epsilon \cdot \ln\left(\frac{1}{\epsilon'}\right)}{\left(\frac{1}{2}-\epsilon\right)^2}\right)$ times, store results,
2. Alice sends all her a_i's to Bob,
3. Bob finds most common value $z \in \{0,1\}^k$ for $a_i \oplus b_i$.
4. Alice outputs the all-0 k-bit string, Bob outputs z.

Complexity: $C_{\epsilon,\epsilon'} \cdot k$. ☹

Second remark: finding runs with equal output

Take two sets of ouputs of the blackboxes a_1, b_1 and a_2, b_2.

$$a_1 \oplus b_1 = a_2 \oplus b_2 \qquad \Longleftrightarrow \qquad a_1 \oplus a_2 = b_1 \oplus b_2$$

Second remark: finding runs with equal output

Take two sets of ouputs of the blackboxes a_1, b_1 and a_2, b_2.

$$\underbrace{a_1 \oplus b_1 = a_2 \oplus b_2}_{\text{runs have same output}} \qquad \Longleftrightarrow \qquad a_1 \oplus a_2 = b_1 \oplus b_2$$

Second remark: finding runs with equal output

Take two sets of ouputs of the blackboxes a_1, b_1 and a_2, b_2.

$$\underbrace{a_1 \oplus b_1 = a_2 \oplus b_2}_{\text{runs have same output}} \quad \Longleftrightarrow \quad \underbrace{a_1 \oplus a_2}_{\text{Alice's side}} = \underbrace{b_1 \oplus b_2}_{\text{Bob's side}}$$

Second remark: finding runs with equal output

Take two sets of ouputs of the blackboxes a_1, b_1 and a_2, b_2.

$$\underbrace{a_1 \oplus b_1 = a_2 \oplus b_2}_{\text{runs have same output}} \quad \Longleftrightarrow \quad \underbrace{a_1 \oplus a_2}_{\text{Alice's side}} = \underbrace{b_1 \oplus b_2}_{\text{Bob's side}}$$

Find runs which output the same thing with a protocol for Equality (costs $O(\log(1/\epsilon))$ for error ϵ)

Second remark: finding runs with equal output

Take two sets of ouputs of the blackboxes a_1, b_1 and a_2, b_2.

$$\underbrace{a_1 \oplus b_1 = a_2 \oplus b_2}_{\text{runs have same output}} \quad \Leftrightarrow \quad \underbrace{a_1 \oplus a_2}_{\text{Alice's side}} = \underbrace{b_1 \oplus b_2}_{\text{Bob's side}}$$

Find runs which output the same thing with a protocol for Equality (costs $O(\log(1/\epsilon))$ for error ϵ)

2nd answer to the riddle: $O\Big(C^2_{\epsilon,\epsilon'} \cdot \log(C^2_{\epsilon,\epsilon'}/\epsilon')\Big)$

Second remark: finding runs with equal output

Take two sets of ouputs of the blackboxes a_1, b_1 and a_2, b_2.

$$\underbrace{a_1 \oplus b_1 = a_2 \oplus b_2}_{\text{runs have same output}} \quad \Longleftrightarrow \quad \underbrace{a_1 \oplus a_2}_{\text{Alice's side}} = \underbrace{b_1 \oplus b_2}_{\text{Bob's side}}$$

Find runs which output the same thing with a protocol for Equality (costs $O(\log(1/\epsilon))$ for error ϵ)

2nd answer to the riddle: $O\Big(C^2_{\epsilon,\epsilon'} \cdot \log(C^2_{\epsilon,\epsilon'}/\epsilon')\Big)$

Second remark: finding runs with equal output

Take two sets of ouputs of the blackboxes a_1, b_1 and a_2, b_2.

$$\underbrace{a_1 \oplus b_1 = a_2 \oplus b_2}_{\text{runs have same output}} \quad \Longleftrightarrow \quad \underbrace{a_1 \oplus a_2}_{\text{Alice's side}} = \underbrace{b_1 \oplus b_2}_{\text{Bob's side}}$$

Find runs which output the same thing with a protocol for Equality (costs $O(\log(1/\epsilon))$ for error ϵ)

2nd answer to the riddle: $O\Big(C^2_{\epsilon,\epsilon'} \cdot \log(C^2_{\epsilon,\epsilon'}/\epsilon')\Big)$

No dependence on k, Alice and Bob oblivious to $f(x, y)$.

Third remark: batch equality

Theorem (Optimal batch equality [HPZZ'21, SIAM J COMP])

Solving t instances of Equality with error ϵ can be done in $O(t + \log(1/\epsilon))$ communication complexity.

Third remark: batch equality

Theorem (Optimal batch equality [HPZZ'21, SIAM J COMP])

Solving t instances of Equality with error ϵ can be done in $O(t + \log(1/\epsilon))$ communication complexity.

Intuitive idea: suppose a 1-bit hash gets computed for each instance. Costs t communication and half the non-equal instances are found in expectation. As the algorithm progresses, less and less remain to be found.

Third remark: batch equality

Theorem (Optimal batch equality [HPZZ'21, SIAM J COMP])

Solving t instances of Equality with error ϵ can be done in $O(t + \log(1/\epsilon))$ communication complexity.

Intuitive idea: suppose a 1-bit hash gets computed for each instance. Costs t communication and half the non-equal instances are found in expectation. As the algorithm progresses, less and less remain to be found.

3rd answer to the riddle: $O\Big(C^2_{\epsilon,\epsilon'} + \log(1/\epsilon')\Big)$

Third remark: batch equality

Theorem (Optimal batch equality [HPZZ'21, SIAM J COMP])

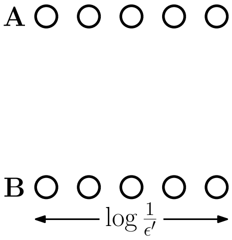
Solving t instances of Equality with error ϵ can be done in $O(t + \log(1/\epsilon))$ communication complexity.

Intuitive idea: suppose a 1-bit hash gets computed for each instance. Costs t communication and half the non-equal instances are found in expectation. As the algorithm progresses, less and less remain to be found.

3rd answer to the riddle: $O\Big(C^2_{\epsilon,\epsilon'} + \log(1/\epsilon')\Big)$ ☹

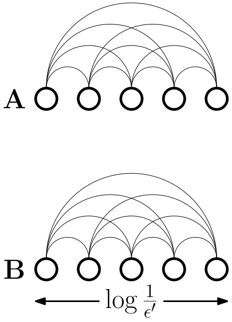
Fourth remark: eliminate most candidates fast

In a batch of $\Theta(\log(1/\epsilon'))$ runs, $> 1/3$ should be the correct output, w.p. $\geq 1 - \epsilon'$.



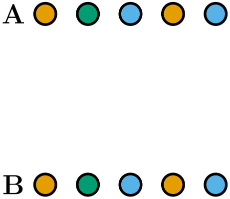
Fourth remark: eliminate most candidates fast

In a batch of $\Theta(\log(1/\epsilon'))$ runs, $> 1/3$ should be the correct output, w.p. $\geq 1 - \epsilon'$.



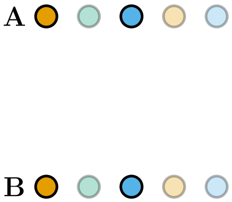
Fourth remark: eliminate most candidates fast

In a batch of $\Theta(\log(1/\epsilon'))$ runs, $> 1/3$ should be the correct output, w.p. $\geq 1 - \epsilon'$.



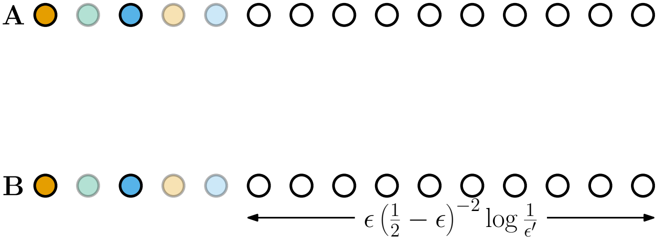
Fourth remark: eliminate most candidates fast

In a batch of $\Theta(\log(1/\epsilon'))$ runs, $> 1/3$ should be the correct output, w.p. $\geq 1 - \epsilon'$.



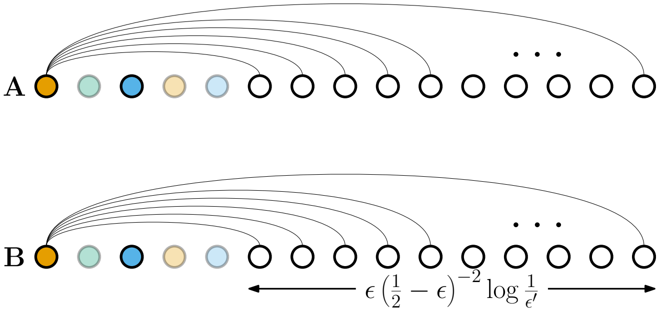
Fourth remark: eliminate most candidates fast

In a batch of $\Theta(\log(1/\epsilon'))$ runs, $> 1/3$ should be the correct output, w.p. $\geq 1 - \epsilon'$.



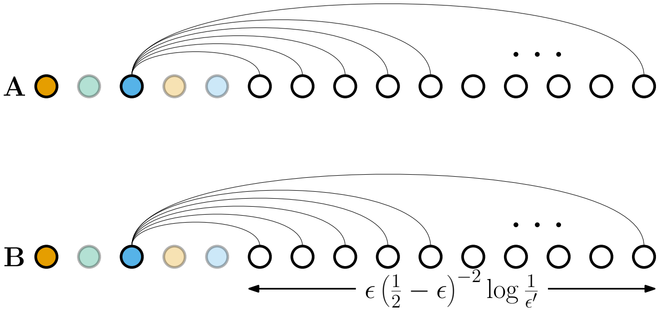
Fourth remark: eliminate most candidates fast

In a batch of $\Theta(\log(1/\epsilon'))$ runs, $> 1/3$ should be the correct output, w.p. $\geq 1 - \epsilon'$.



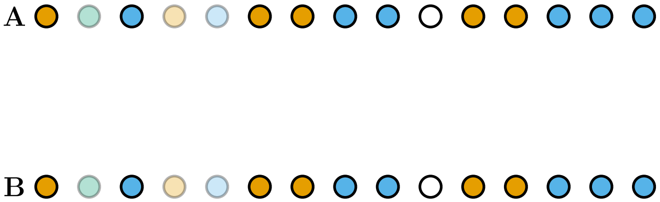
Fourth remark: eliminate most candidates fast

In a batch of $\Theta(\log(1/\epsilon'))$ runs, $> 1/3$ should be the correct output, w.p. $\geq 1 - \epsilon'$.



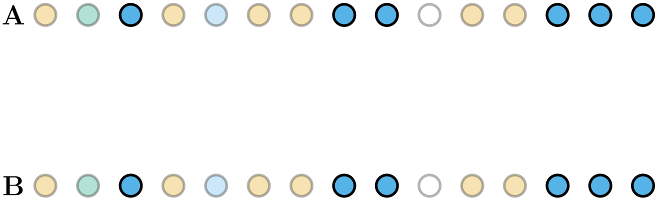
Fourth remark: eliminate most candidates fast

In a batch of $\Theta(\log(1/\epsilon'))$ runs, $> 1/3$ should be the correct output, w.p. $\geq 1 - \epsilon'$.



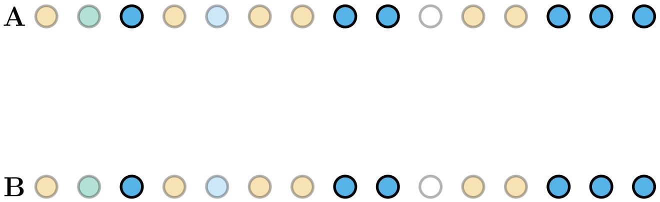
Fourth remark: eliminate most candidates fast

In a batch of $\Theta(\log(1/\epsilon'))$ runs, $> 1/3$ should be the correct output, w.p. $\geq 1 - \epsilon'$.



Fourth remark: eliminate most candidates fast

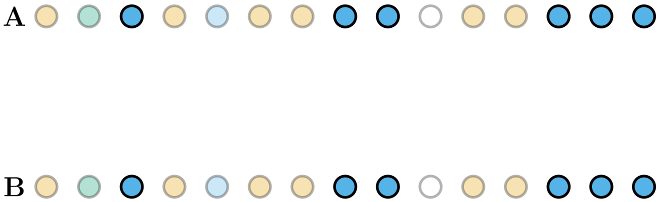
In a batch of $\Theta(\log(1/\epsilon'))$ runs, $> 1/3$ should be the correct output, w.p. $\geq 1 - \epsilon'$.



4th answer to the riddle: $O\big(C_{\epsilon,\epsilon'} + \log^2(1/\epsilon')\big)$

Fourth remark: eliminate most candidates fast

In a batch of $\Theta(\log(1/\epsilon'))$ runs, $> 1/3$ should be the correct output, w.p. $\geq 1 - \epsilon'$.



4th answer to the riddle: $O\big(C_{\epsilon,\epsilon'} + \log^2(1/\epsilon')\big)$

Last optimization: largest component in random graphs

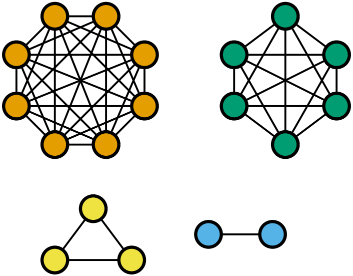
Lemma (Variation of a lemma in [ER'60])

- *$G(n, p)$: graph with n vertices, each edge picked w.p. p*
- *$L_1(G)$: size of the largest connected component of G.*
- *$\alpha \in [0, 1]$ and $c \in \mathbb{R}^+$*

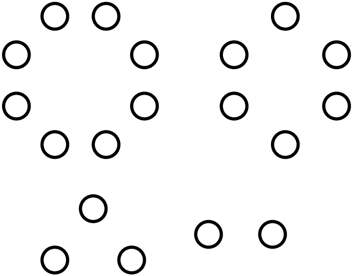
$$\Pr[L_1(G(n, c/n) < (1-\alpha)n] \leq e^{\left(\ln(2) - \frac{\alpha}{2}\left(1-\frac{\alpha}{2}\right)c\right)n}$$

In particular, goes to 0 exponentially fast with n if $\alpha c > 4\ln(2)$.

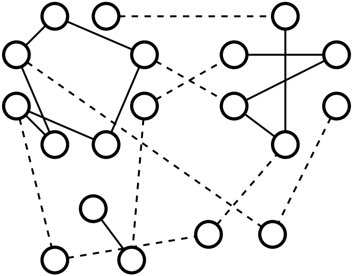
Last optimization: largest component in random graphs



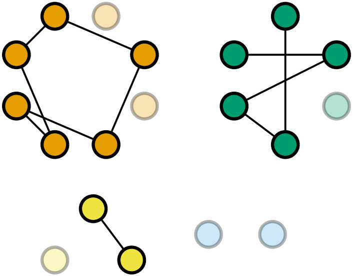
Last optimization: largest component in random graphs



Last optimization: largest component in random graphs



Last optimization: largest component in random graphs



Last optimization: largest component in random graphs

Lemma (Variation of a lemma in [ER'60])

- *$G(n, p)$: graph with n vertices, each edge picked w.p. p*
- *$L_1(G)$: size of the largest connected component of G.*
- *$\alpha \in [0, 1]$ and $c \in \mathbb{R}^+$*

$$\Pr[L_1(G(n, c/n) < (1 - \alpha)n] \leq e^{\left(\ln(2) - \frac{\alpha}{2}\left(1 - \frac{\alpha}{2}\right)c\right)n}$$

In particular, goes to 0 exponentially fast with n if $\alpha c > 4\ln(2)$.

last answer to the riddle:

Last optimization: largest component in random graphs

Lemma (Variation of a lemma in [ER'60])

- $G(n,p)$*: graph with n vertices, each edge picked w.p. p*
- $L_1(G)$*: size of the largest connected component of* G.
- $\alpha \in [0,1]$ *and* $c \in \mathbb{R}^+$

$$\Pr[L_1(G(n,c/n) < (1-\alpha)n] \leq e^{\left(\ln(2) - \frac{\alpha}{2}\left(1-\frac{\alpha}{2}\right)c\right)n}$$

In particular, goes to 0 *exponentially fast with n if* $\alpha c > 4\ln(2)$.

last answer to the riddle: $O\left(C_{\epsilon,\epsilon'}\right) = O\left(\frac{\epsilon \cdot \ln\left(\frac{1}{\epsilon'}\right)}{\left(\frac{1}{2}-\epsilon\right)^2}\right)$ ☺

Why the riddle?

$$C_{\epsilon,\epsilon'} \in \Theta\left(\frac{\epsilon \cdot \ln\left(\frac{1}{\epsilon'}\right)}{\left(\frac{1}{2} - \epsilon\right)^2}\right)$$

Theorem (Usual error reduction [folklore, KN'97])

Let $\epsilon > \epsilon' > 0$ *and* $\mathcal{M} \in \{\mathrm{open}, \mathrm{loc}, \mathrm{B}, \mathrm{A}\}$, *then:*

$$R^{\mathcal{M}}_{\epsilon'}(f) \leq C_{\epsilon,\epsilon'} \cdot R^{\mathcal{M}}_{\epsilon}(f).$$

Why the riddle?

$$C_{\epsilon,\epsilon'} \in \Theta\left(\frac{\epsilon \cdot \ln\left(\frac{1}{\epsilon'}\right)}{\left(\frac{1}{2}-\epsilon\right)^2}\right)$$

Theorem (Usual error reduction [folklore, KN'97])

Let $\epsilon > \epsilon' > 0$ *and* $\mathcal{M} \in \{\mathrm{open}, \mathrm{loc}, \mathrm{B}, \mathrm{A}\}$*, then:*

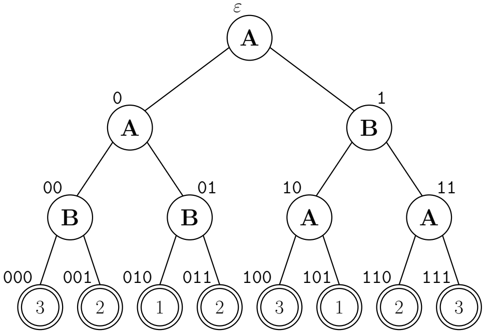
$$R^{\mathcal{M}}_{\epsilon'}(f) \leq C_{\epsilon,\epsilon'} \cdot R^{\mathcal{M}}_{\epsilon}(f).$$

Theorem (XOR error reduction)

Let $\epsilon > \epsilon' > 0$*, then:*

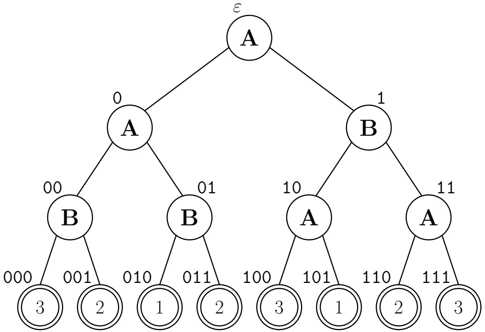
$$R^{\mathrm{xor}}_{\epsilon'}(f) \leq C_{\epsilon,\epsilon'} \cdot (R^{\mathrm{xor}}_{\epsilon}(f)) + O(C_{\epsilon,\epsilon'}).$$

Communication complexity: protocol tree



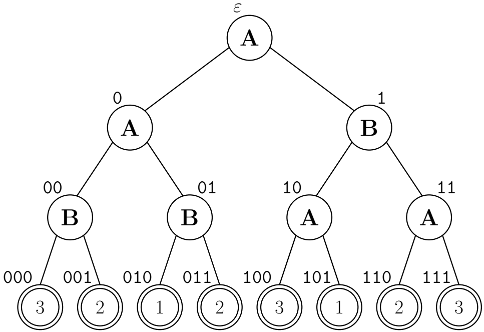
- Nodes are partitioned between Alice and Bob.

Communication complexity: protocol tree



- Nodes are partitioned between Alice and Bob.
- A node's owner decides whether to go left or right from there.

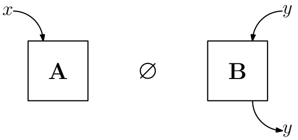
Communication complexity: protocol tree



- Nodes are partitioned between Alice and Bob.
- A node's owner decides whether to go left or right from there.
- The process is unambiguous.

An ambiguity in the model.

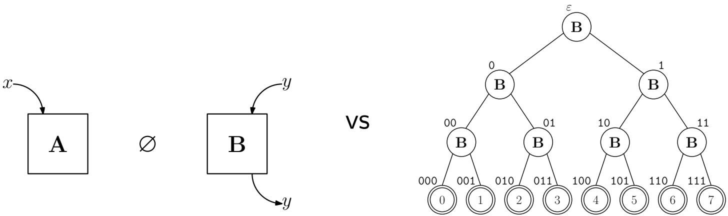
Consider the function $id_B(x, y) = y$.



vs

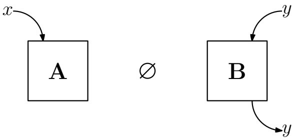
An ambiguity in the model.

Consider the function $id_B(x, y) = y$.

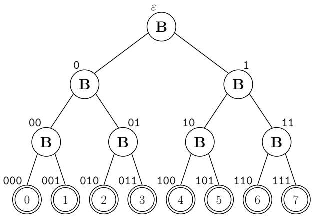


An ambiguity in the model.

Consider the function $id_B(x, y) = y$.



vs



Who outputs the result matters.

Nothing new

The observation that 'who outputs' matters is nothing new.

- Sending a message [Shannon'48]
- NBA problem [Orlitsky'90]
- Compression to information [BR'14, BBCR'13, BMY'15, Sherstov'18, BK'18]

Nothing new

The observation that 'who outputs' matters is nothing new.

- Sending a message [Shannon'48]
- NBA problem [Orlitsky'90]
- Compression to information [BR'14, BBCR'13, BMY'15, Sherstov'18, BK'18]

However...

- ...never systematically studied?

Adapted tree definition

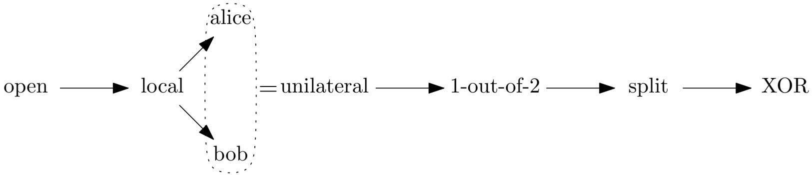
Leaves are now labeled by an output mechanism:

- It may be an output
- It may be a function of one of the player's input (if one player outputs)
- It may be two functions of the player's inputs (in which case the two players output something)

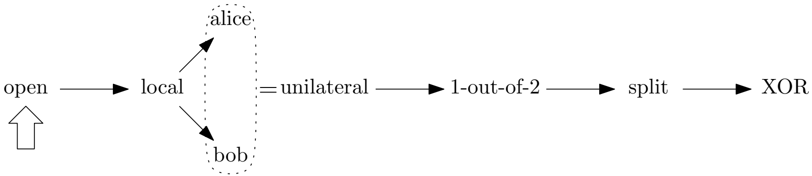
We define different models of communication complexity, with the measures:

- $D^{\mathcal{M}}(f) =$ deterministic communication complexity of f in model $\mathcal{M}$.
- $R_{\epsilon}^{\mathcal{M}}(f) =$ randomized communication complexity of f in model $\mathcal{M}$ with error $\leq \epsilon$.

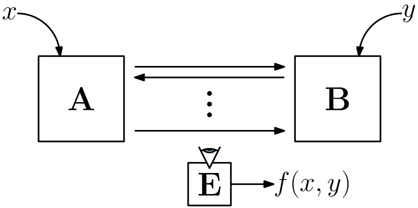
Models



Models

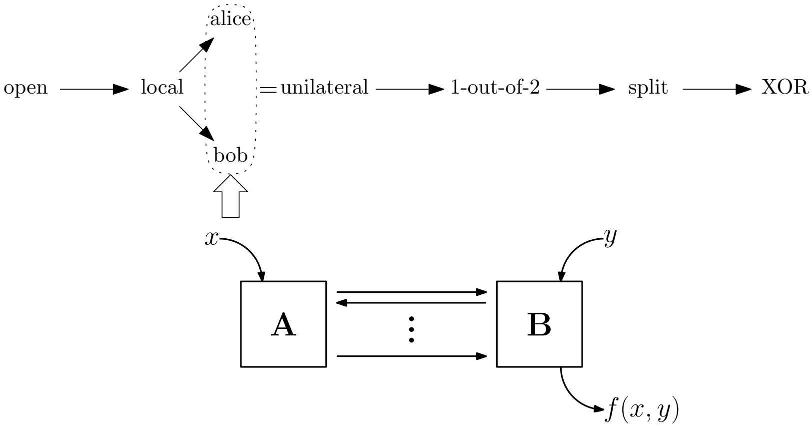


alice
open
local
=unilateral
1-out-of-2
split
XOR
bob

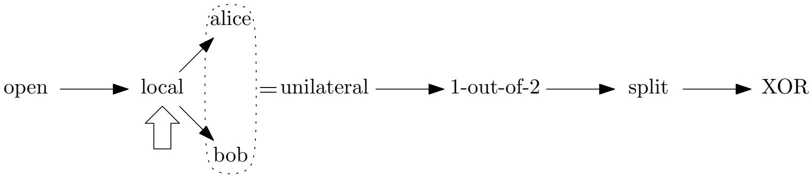


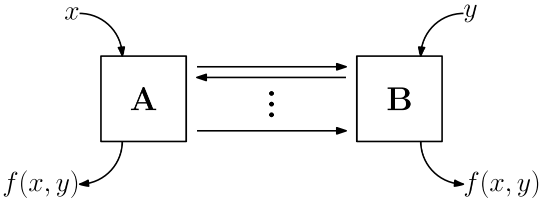
x
y
A
B
E
$f(x, y)$

Models

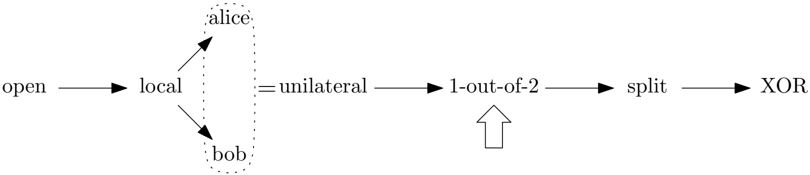


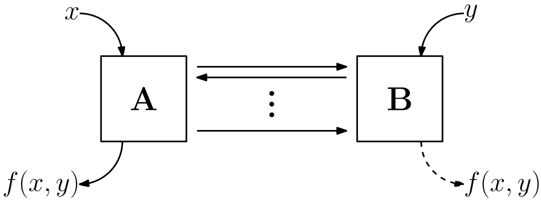
Models


alice
open
local
=unilateral
1-out-of-2
split
XOR
bob

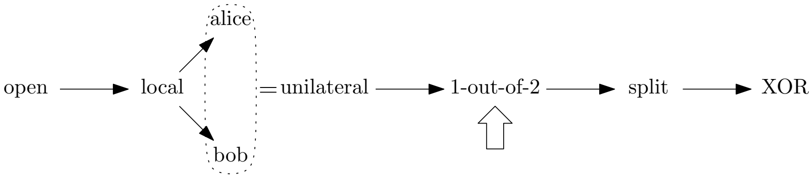

x
y
A
B
$f(x, y)$
$f(x, y)$

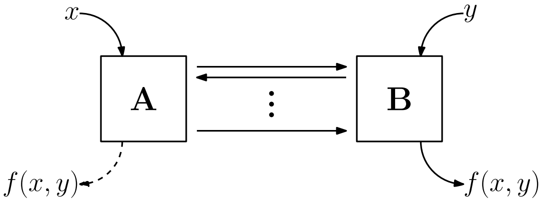
Models


open
local
alice
bob
=unilateral
1-out-of-2
split
XOR

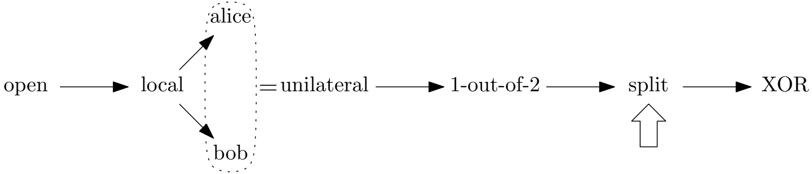

x
y
A
B
$f(x, y)$
$f(x, y)$

Models

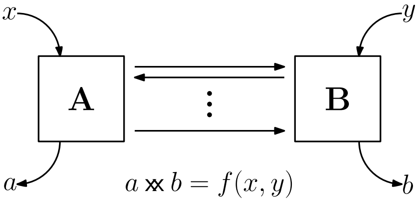

open
local
alice
bob
=unilateral
1-out-of-2
split
XOR


x
y
A
B
$f(x,y)$
$f(x,y)$

Models

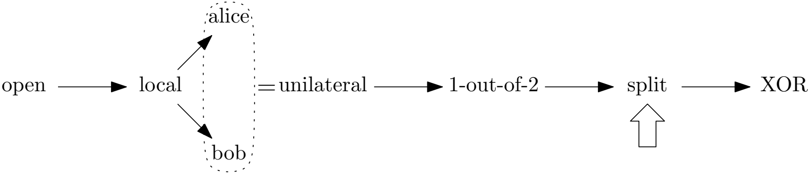


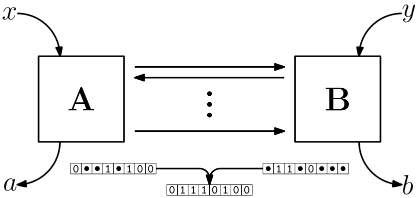
alice
open
local
=unilateral
1-out-of-2
split
XOR
bob



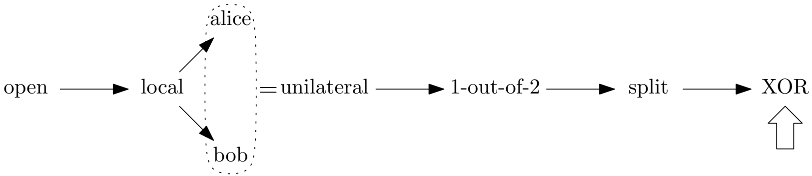
x
y
A
B
a
b
a ⨳ b = f(x, y)

Models

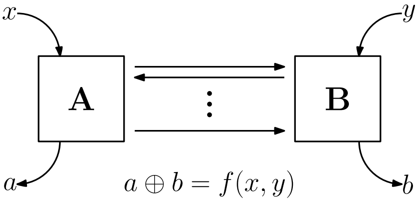

open
local
alice
bob
= unilateral
1-out-of-2
split
XOR


x
y
A
B
a
b
0 • • 1 • 1 0 0
• 1 1 • 0 • • •
0 1 1 1 0 1 0 0

Models

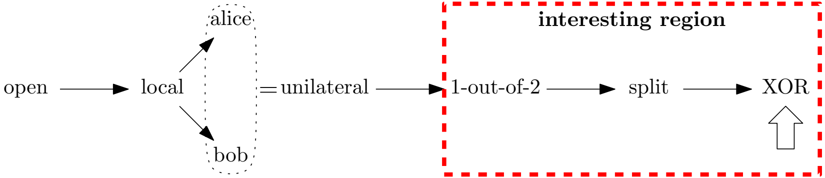


open
local
alice
bob
=unilateral
1-out-of-2
split
XOR

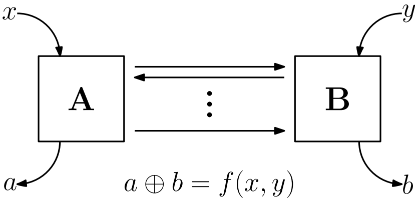


x
y
A
B
a
b
$a \oplus b = f(x, y)$

Models

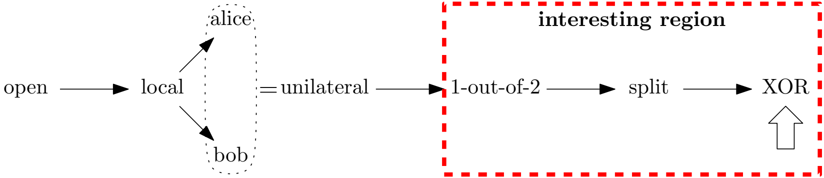


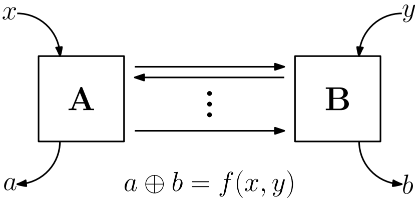
alice
open
local
=unilateral
bob
interesting region
1-out-of-2
split
XOR



x
y
A
B
a
b
$a \oplus b = f(x, y)$

Models





Thanks!