Streaming Algorithms

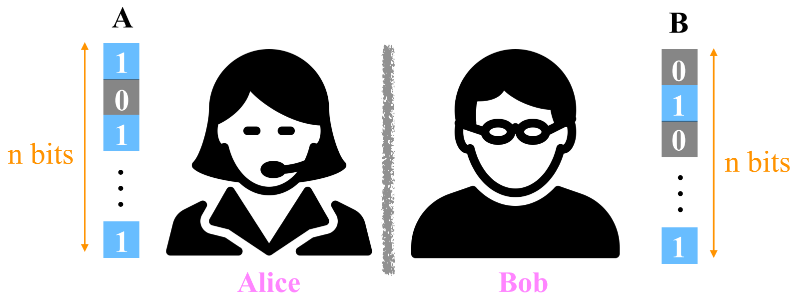
Meng-Tsung Tsai

05/18/2018

References

- "Communication Complexity," Kushilevitz

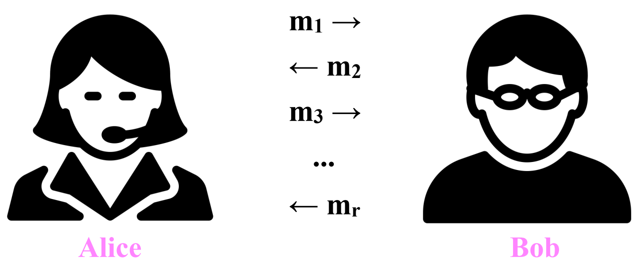
Computing f: $\{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$



How many bits are needed to transfer between Alice and Bob, 2-way and multiple-round, so that f(A, B) can be computed?

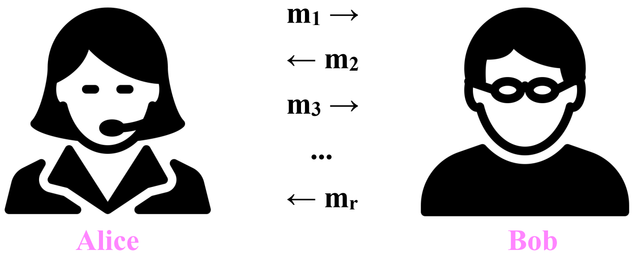
Computing f: $\{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$

$m_1\$m_2\$\ldots\$m_r\$\$$ is called a transcript



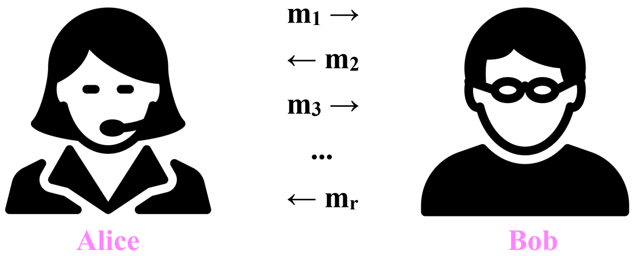
Given a protocol P and an input (x, y), the number of sent bits is $s_P(x, y) = \sum_i |m_i|$.

Computing f: $\{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$



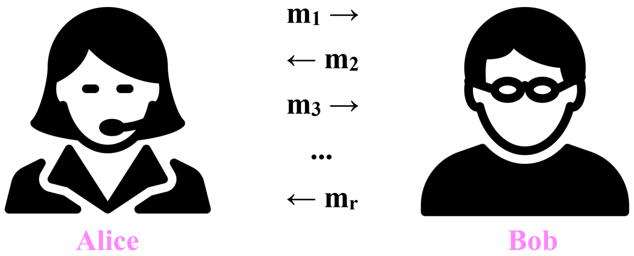
The complexity measure of protocol P is
$D(P) = \max_{(x,y)} s_P(x, y)$.

Computing f: $\{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$



The deterministic communication complexity of f
is $D(f) = \min_P D(P)$
among those P's that solve f.

Computing f: $\{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$



$D(f) \leq n+1$.

f-monochromatic rectangle

Bob's input

Alice's input	(0, 0, ..., 0)	(1, 0, ..., 0)	...	(1, 1, ..., 1)
(0, 0, ..., 0)	1	1	...	0
(1, 0, ..., 0)	0	1	...	1
...	...	...	...	...
(1, 1, ..., 1)	0	1	...	1

The possible inputs form an 2^n by 2^n matrix.

f-monochromatic rectangle

Bob's input

Alice's input

	(0, 0, ..., 0)	(1, 0, ..., 0)	...	(1, 1, ..., 1)
(0, 0, ..., 0)	1	1	...	0
(1, 0, ..., 0)	0	1	...	1
...	...	...	...	...
(1, 1, ..., 1)	0	1	...	1

Let S be a subset of possible inputs, i.e. $S \subseteq \{0,1\}^n \times \{0,1\}^n$.
If the associated f values are not the same, then
the inputs in S cannot share the same transcript.

f-monochromatic rectangle

Bob's input

Alice's input

	(0, 0, ..., 0)	(1, 0, ..., 0)	...	(1, 1, ..., 1)
(0, 0, ..., 0)	1	1	...	0
(1, 0, ..., 0)	0	1	...	1
...	...	...	...	...
(1, 1, ..., 1)	0	1	...	1

If S is not a rectangle $A \times B$, i.e. $A \subseteq \{0, 1\}^n$ and $B \subseteq \{0, 1\}^n$,
the inputs in S cannot share the same transcript.

f-monochromatic rectangle

Bob's input

Alice's input

	(0, 0, ..., 0)	(1, 0, ..., 0)	...	(1, 1, ..., 1)
(0, 0, ..., 0)	1	1	...	0
(1, 0, ..., 0)	0	1	...	1
...	...	...	...	...
(1, 1, ..., 1)	0	1	...	1

Let C* be the minimum number of f-monochromatic
rectangles into which one can partition all possible inputs.

f-monochromatic rectangle

Bob's input

Alice's input

	(0, 0, ..., 0)	(1, 0, ..., 0)	...	(1, 1, ..., 1)
(0, 0, ..., 0)	1	1	...	0
(1, 0, ..., 0)	0	1	...	1
...	...	...	...	...
(1, 1, ..., 1)	0	1	...	1

Then $D(f) \geq \log_2 C^*$.

The fooling set method

Let $\{(x_1, y_1), (x_2, y_2), \ldots, (x_\ell, y_\ell)\}$ be a subset of possible inputs. We call it a fooling set of size ℓ if for some $b \in \{0, 1\}$

- $f(x_i, y_i) = b$ for all $1 \le i \le \ell$
- $f(x_i, y_j) \ne b$ or $f(x_j, y_i) \ne b$ for all $1 \le i < j \le \ell$

If f has a fooling set of size ℓ, then $C^* \ge \ell$. (Why?)

Disjointness

Given $x, y \in \{0, 1\}^n$, $f(x, y) = 1$ iff there exists no index i so that $x_i = y_i = 1$.

DISJ has a fooling set of size $C(n, n/2)$, and hence $D(\text{DISJ}) = \Omega(n)$.

Equality

Given $x, y \in \{0, 1\}^n$, $f(x, y) = 1$ iff $x = y$.

EQ has a fooling set of size 2^n, and hence $D(\text{EQ}) = \Omega(n)$.

Triangle-freeness

Given an n-node undirected simple graph G, testing whether G is triangle-free requires $\Omega(n^2/p)$ bits for any p-pass deterministic algorithms.

Idea: reduce DISJ(n^2) to Triangle-freeness

Construct a graph G of 3n nodes, 1x, 2x, ..., nx, 1y, 2y, ..., ny, 1z, 2z, ..., nz. For Alice's array of length n^2, if (i*n+j)-th bit is 1, then link an edge connecting 1i and 3j, or do nothing otherwise. For Bob's array of length n^2, if (i*n+j)-th bit is 1, then link an edge connecting 2i and 3j, or do nothing otherwise. Finally, link an edge to connect 1k and 2k for each k in {1, 2, ..., n}.

Two arrays intersect iff G has a triangle.

Exercise 1

Given an n-node undirected simple graph G, outputting the girth of G requires $\Omega(n^2/p)$ bits for any p-pass deterministic algorithms.

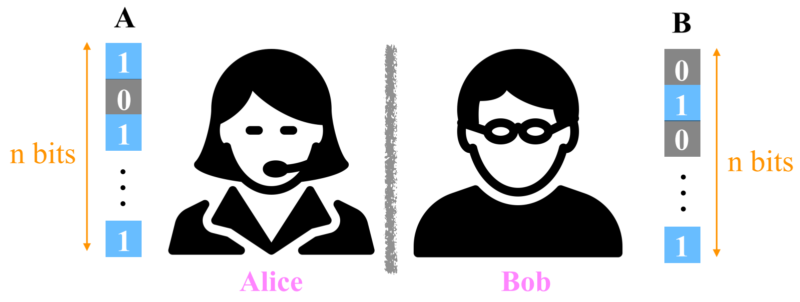
Exercise 2

Given a point set P of n points on a plane, output the extreme points on the convex hull of P in clockwise order requires $\Omega(n/p)$ bits for any p-pass deterministic algorithm.

Randomized Protocol (Public Coins)

Computing f: $\{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$

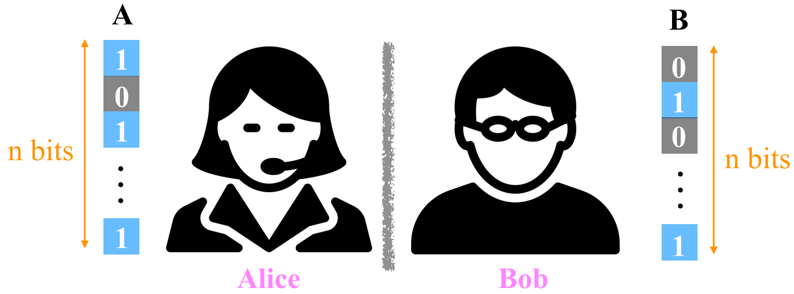
a public random string: 000101110111



How many bits are needed to transfer between Alice and Bob, 2-way and multiple-round, so that f(A, B) can be computed, given the access of a public random string?

Computing f: $\{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$

a public random string: 000101110111



We say f is computed by a randomized protocol if for any input the output is correct with a constant probability > 1/2, say 2/3 or 3/4. (The constant doesn't matter.)

Equality

The randomized communication complexity $R(EQ) = O(\log n)$.

Idea: random checksum

Interpret Alice's array and Bob's array to be a polynomial of x with degree < n. Let them be p(x) and q(x).

If we randomly pick a value r from [0, ..., U-1] for some prime U, the probability that $p(r) = q(r) \pmod U$ is $< n/|U|$.

Here is a separation between the deterministic complexity and randomized complexity. $D(EQ) = \Omega(n)$ but $R(EQ) = O(\log n)$.

Exercise 3

Greater-than: Given $x, y \in \{0, 1\}^n$, $f(x, y) = 1$ iff $x > y$.

Show that $D(GT) = \Omega(n)$ and $R(GT) = O(\log^2 n)$.

Known Results

$D(DISJ) = \Omega(n)$, $R(DISJ) = \Omega(n)$

$D(EQ) = \Omega(n)$, $R(EQ) = O(\log n)$

$D(GT) = \Omega(n)$, $R(GT) = O(\log n)$

$D(IP) = \Omega(n)$, $R(IP) = \Omega(n)$ // inner product of two arrays mod 2

$D^{1\text{-way}}(Index) = \Omega(n)$, $R^{1\text{-way}}(Index) = \Omega(n)$

Part of Written Assignment #3

Give a problem on n-node simple graphs, and prove that any 1-pass streaming algorithm has a space lower bound of $\Omega(k)$ where $k = n^c$ for some constant $c > 0$.