communication complexity: a crash course

Rebecca Kempe

COMP 5901 Project Presentation

where we started

- complexity theory (in my view) is the study of computation under limitations
  - resource limitations: time, space
  - external "help": oracles, "advice"
  - choices: (non-)deterministic, probabilistic
- we have also seen many different models of computation (circuits, prover-verifier, etc.)

questions we like to ask

- under these limitations, is there an algorithm?
- under what limitations is there no longer an algorithm?
- what problems can we solve under these limitations?
- are these two sets of limitations fundamentally the same?

where we're going

- what happens when multiple machines with disjoint information need to work together?
- exploring communication as a resource
  - what is the minimum amount needed?
- some of the language/concepts will be familiar from discussing interactive proofs

this presentation comes in four parts

1. The two-party model
2. Multi-party models
3. Connections to distributed systems
4. Broadening our horizons

Part 1:

the two-party model

(deterministic case)

Alice and Bob want to compute a function

- Alice and Bob each have infinite power
- unfortunately, each only has half of the input
- Alice and Bob need to agree on an output
- the goal is to minimize how much they need to communicate

Alice and Bob need a protocol

- who speaks first? who speaks next?
- what do they say?
- when do they stop talking?
- what is the answer?

Each of these can only depend on the input or previously communicated information.

more formally...

- we have some function $f : X \times Y \to Z$
- for simplicity, $X = Y = \{0, 1\}^n$, $Z = \{0, 1\}$
- Alice and Bob want to compute $f(x, y)$, but Alice only knows x and Bob only knows y
- they exchange bits based on some protocol $\mathcal{P}$ that depends solely on f

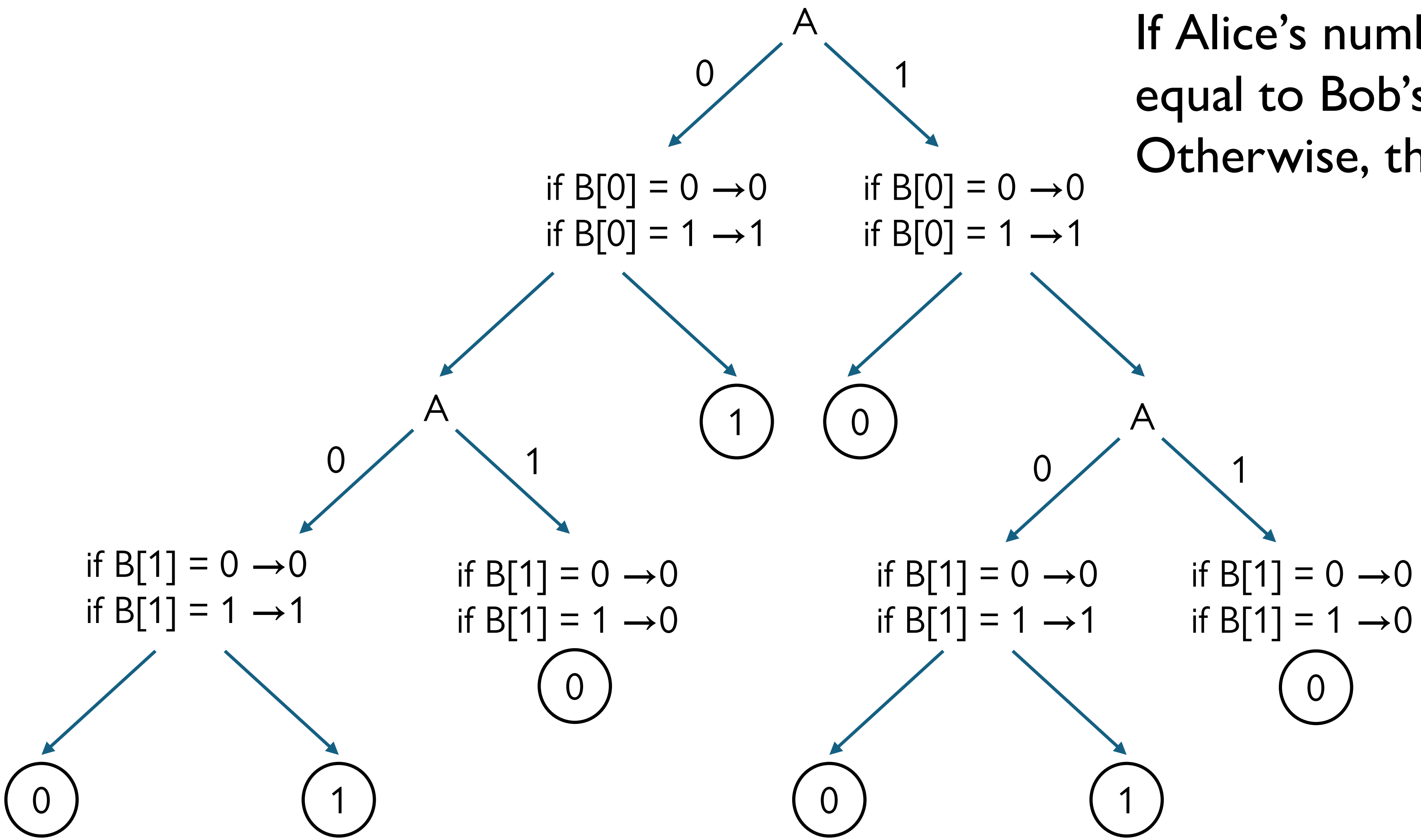
a protocol is a tree

definition: a protocol $\mathcal{P}$ is a binary tree where each internal node v is labelled either by a function $a_v : X \to \{0, 1\}$ or $b_v : Y \to \{0, 1\}$ and each leaf is labelled with an element $z \in Z$

an example

Let X, Y be sets of 2-bit strings. Alice and Bob need to figure out who has the bit that represents the higher number.

If Alice's number is higher than or equal to Bob's number, 0 is output. Otherwise, the output is 1.

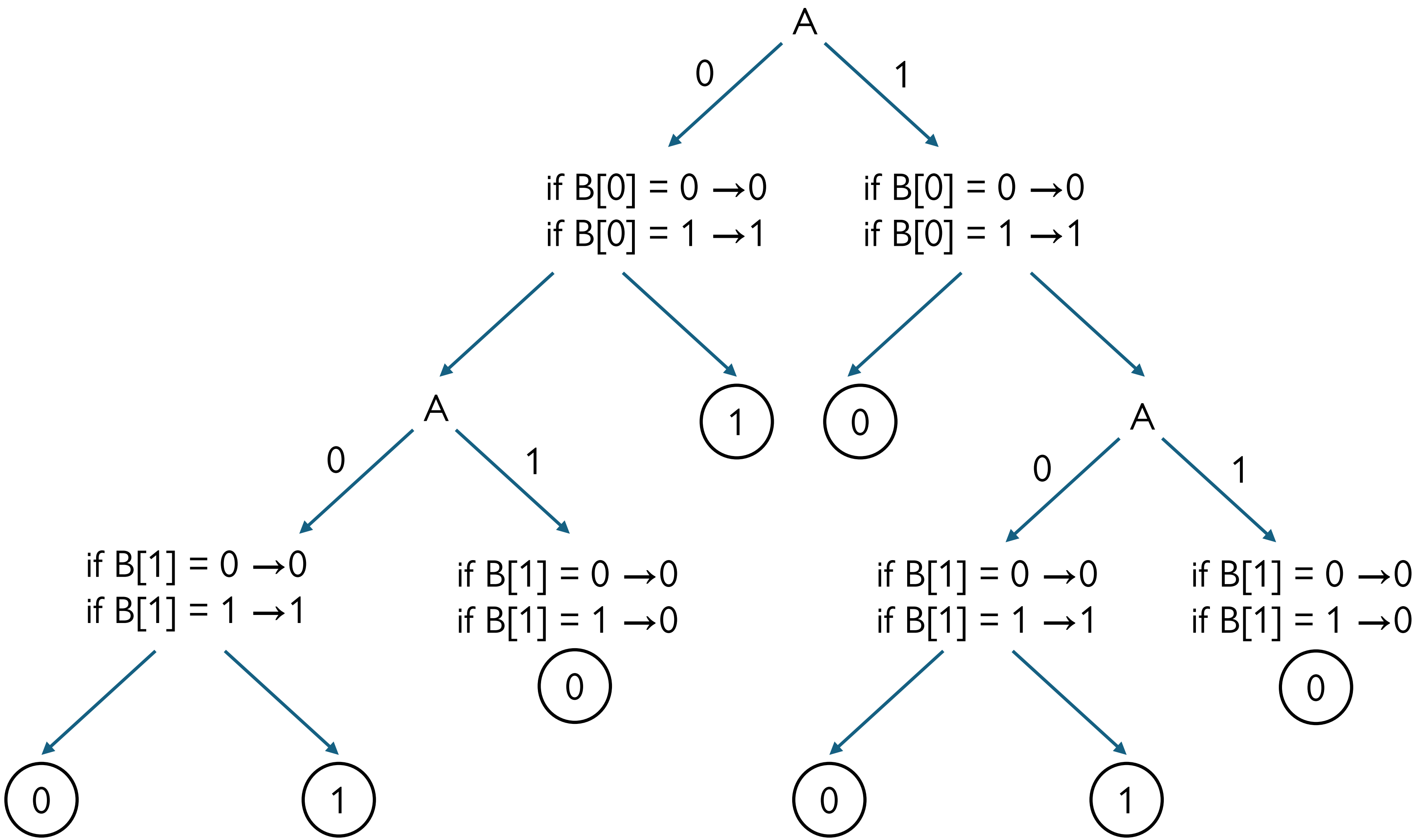


protocol costs

definition: the cost of a protocol $\mathcal{P}$ on an input (x, y) is the length of the path taken on the input, and the cost of the protocol $\mathcal{P}$ is the height of the tree.

an example

The cost of this protocol on the input (01, 10) is 2. The overall cost of this protocol is 4.



communication complexity

definition: the (deterministic) communication complexity of a function $f : X \times Y \to Z$, denoted $D(f)$, is the minimum cost of $\mathcal{P}$ over all protocols $\mathcal{P}$ that compute f.

by the way… (an aside)

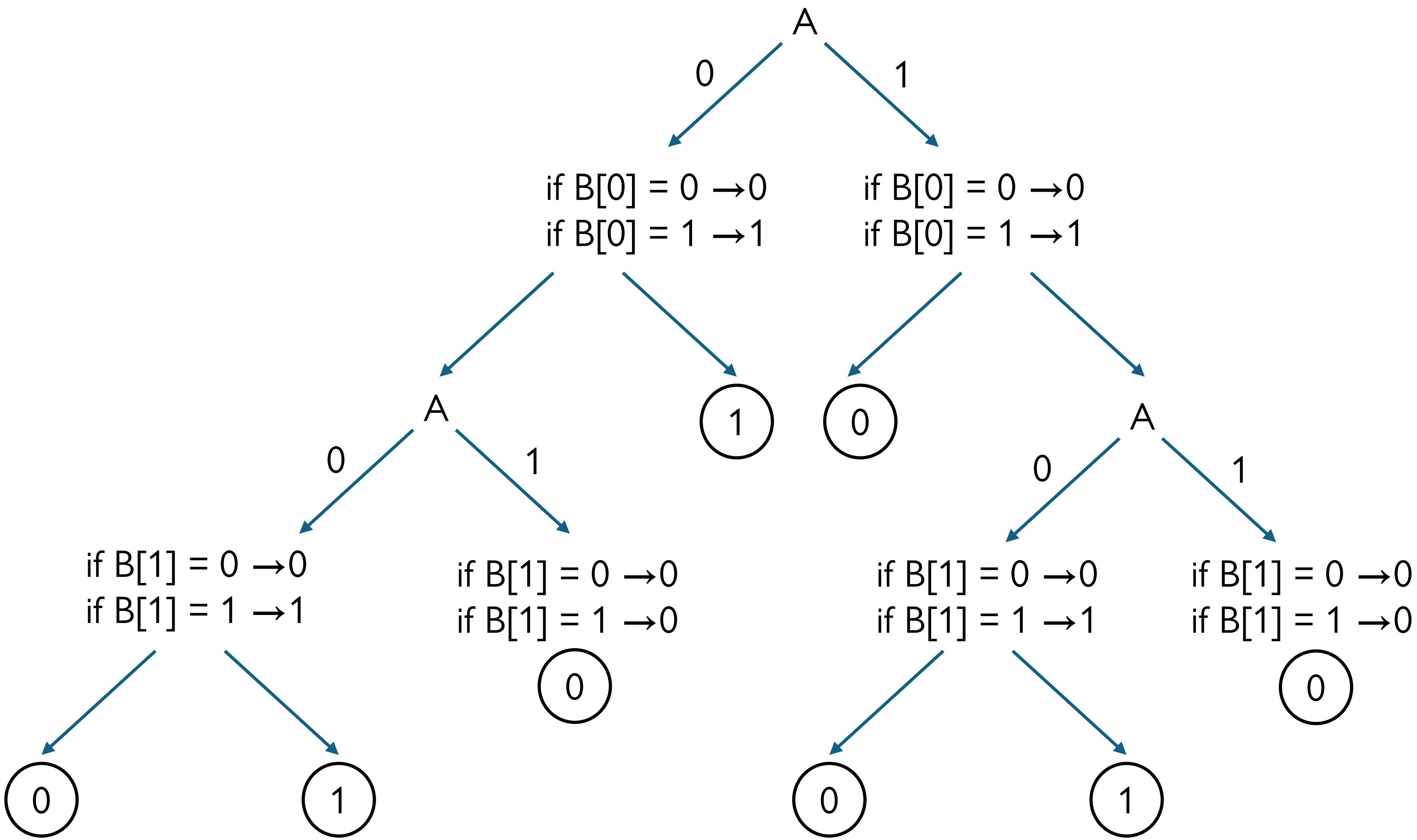
- this is clearly a non-uniform model of computation – different input sizes require different protocols
  - we can think of specific problems as having "protocol families"
- non-deterministic and randomized versions also exist (public and private coins as well)

the set of outputs is a matrix

- many techniques for proving lower bounds involve combinatorial or algebraic analysis of communication matrices
- to every function $f : X \times Y \to Z$, we associate a $|X|$ by $|Y|$ matrix, where rows are indexed by elements of X, columns are indexed by elements of Y, and each (x, y) entry is simply $f(x, y)$

an example

If Alice's number is higher than or equal to Bob's number, 0 is output. Otherwise, the output is 1.



an example

If Alice's number is higher than or equal to Bob's number, 0 is output. Otherwise, the output is 1.

	00	01	10	11
00	0	1	1	1
01	0	0	1	1
10	0	0	0	1
11	0	0	0	0

monochromatic rectangles

- a combinatorial rectangle in $X \times Y$ is a subset $R \subseteq X \times Y$ such that $R = A \times B$ for some $A \subseteq X$ and $B \subseteq Y$
- a subset $R \subseteq X \times Y$ such that f is fixed on R is called f-monochromatic
- rectangles **do not** need to be contiguous in our drawings of them

monochromatic rectangles

	000	001	010	011	100	101	110	111
000	0	1	1	0	1	0	0	0
001	1	0	0	0	0	0	0	1
010	1	0	0	0	1	0	0	0
011	0	0	1	0	0	0	0	1
100	1	0	0	0	1	0	0	1
101	1	1	1	0	0	0	1	1
110	0	0	0	0	1	0	0	1
111	0	1	1	0	1	1	0	1

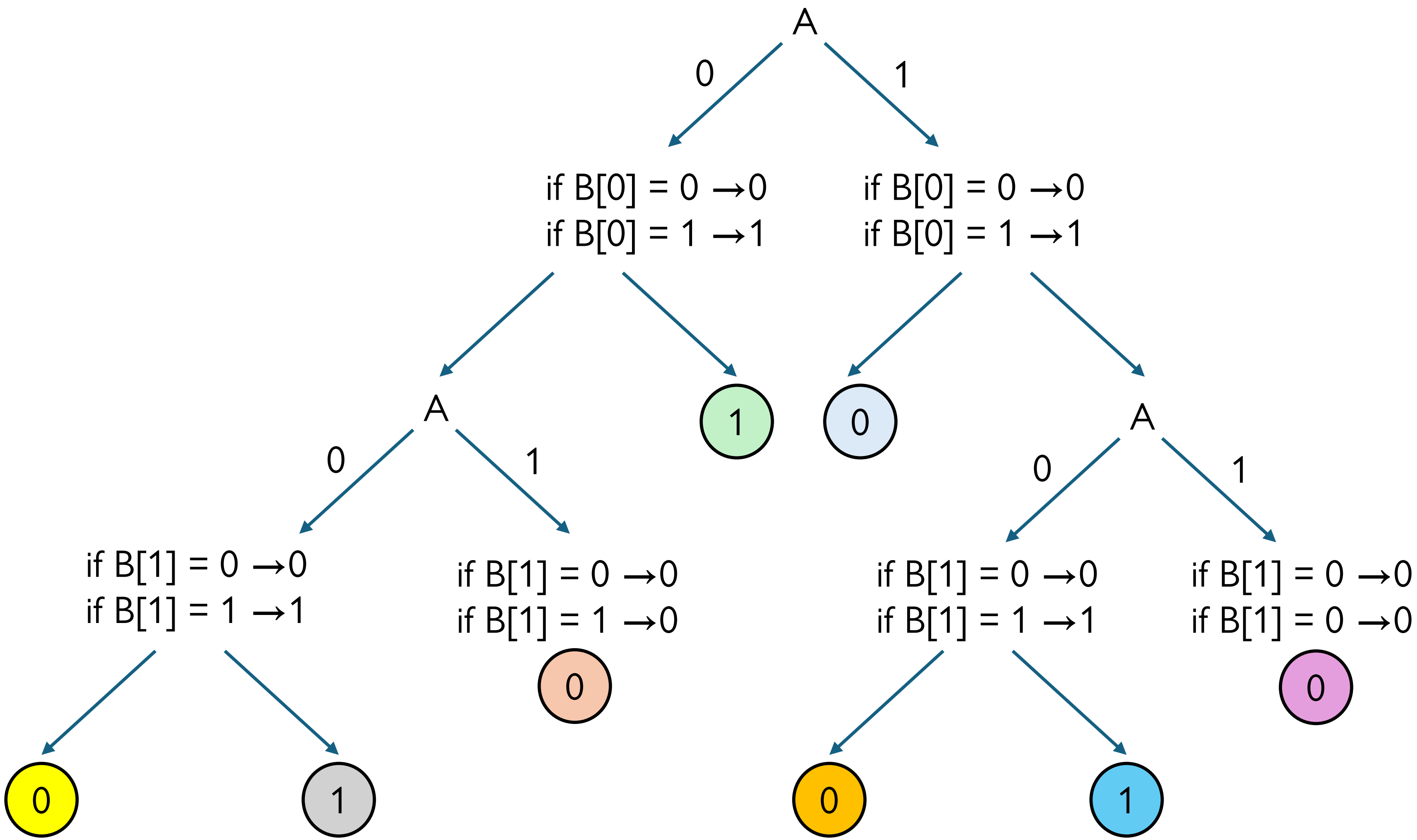
(Kushilevitz and Nisan, 1997)

a leaf is a rectangle

- fact: any protocol $\mathcal{P}$ for a function f induces a partition of $X \times Y$ into f-monochromatic rectangles, and the number of rectangles is the number of leaves of $\mathcal{P}$
- proof: for every node v, the set of inputs that reach v is a rectangle, and the fact that the rectangle is monochromatic follows by definition of a protocol.

an example

If Alice's number is higher than or equal to Bob's number, 0 is output. Otherwise, the output is 1.



If Alice's number is higher than or equal to Bob's number, 0 is output. Otherwise, the output is 1.

an example

	00	01	10	11
00	0	1	1	1
01	0	0	1	1
10	0	0	0	1
11	0	0	0	0

a lower bound is a bound on rectangles

- corollary: if every possible partition of $X \times Y$ into f-monochromatic rectangles requires at least t rectangles, then $D(f) \geq \log_2 t$
- proof: every possible protocol would then have at least t leaves, so the depth of the tree would be at least $\log_2 t$

a lower bound is a bound on rectangles

- fooling set method

  if there exists a large enough subset of input pairs that cannot be in the same monochromatic rectangle, then there are many monochromatic rectangles

- rank lower bounds

  this involves calculating the rank of the communication matrix; $D(f) \geq \log_2 \operatorname{rank}(f)$

Part 2:

multi-party models

(mostly definitions)

k players want to compute a *k*-argument function

- this is a function $f(x_1, x_2, \ldots, x_k)$
- each player has infinite computational power
- unfortunately, each player only has *some* of the input (there *may* be overlap)
- the goal is to minimize how much the players need to communicate

different types of models

- “number on forehead” models
- “number in hand” models
- blackboard (BB) model
- message passing (MP) model
- simultaneous message passing (SMP) model

decreasing in power

the blackboard model

- we can imagine that there is a shared blackboard every player can see, and each time a player wants to send information, they write it on the blackboard
- messages are exchanged according to a fixed protocol, and the cost of the protocol is the number of bits written in the worst case

the blackboard model

- “number in hand” (NIH) version: each of the *k* players only knows one of the *k* arguments
- “number on forehead” (NOF) version: the i_{th} player knows all of the arguments except x_i
- two-party techniques can be used under the NIH model
- the NOF model requires new techniques

the message passing model

- each player sends messages to one other player at a time
  - clique topology: messages are sent directly to players
  - hub topology: messages are sent to a coordinator that forwards the messages
- more relevant to distributed computing, but somewhat less studied than the BB model

MP model challenges

- each player only observes part of the communication transcript
- trying to analyze a k-dimensional communication matrix could be challenging
- current techniques for proving lower bounds involve reductions to 2-party problems
  - symmetrization and composition

the simultaneous message passing model

- there is a coordinator, and each player can only send one message to the coordinator
- after receiving all of the messages, the coordinator should output the answer
- it is expected that all players know the number of players k, the input size n, etc.

Part 3:

connections to distributed systems

(very briefly)

a bit of history

- communication complexity was founded in 1979 in a paper by Andrew Yao
- "Some Complexity Questions related to Distributive Computing"
- motivating question: what complications arise when more than one processor is used to solve a problem

where is the message passing model actually used?

- this is the canonical model in distributed computing (and also in distributed systems)
- this is a good abstraction for:
  - sensor networks
  - network routers
  - distributed databases
  - and so on…

Part 4:

broadening our horizons

(complexity classes; information complexity)

comm. complexity classes

- we can define analogs of the traditional complexity classes in this setting

$$P^{CC}, RP^{CC}, BPP^{CC}, NP^{CC}, PP^{CC}$$

- we know almost everything!

$$P^{CC} \subsetneq RP^{CC} \subsetneq BPP^{CC} \subsetneq NP^{CC}$$

information complexity

- a separate but related notion that examines what amount of information needs to be exchanged in order to solve a problem
- the information complexity of a problem is a lower bound on its comm. complexity
- this is defined pretty differently (more analytic than combinatorial in nature)

References

Arora, Sanjeev, and Boaz Barak. 2009. *Computational Complexity: A Modern Approach*. 1st ed. USA: Cambridge University Press.

Braverman, Mark. 2022. "Communication and information complexity." In *Proceedings of the 2022 International Congress of Mathematicians*, edited by Dmitry Beliaev and Stanislav Smirnov, 1:284–320. https://doi.org/10.4171/ICM2022/208.

Huang, Zengfeng, and Ke Yi. 2014. "The Communication Complexity of Distributed epsilon-Approximations." In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, 591–600. https://doi.org/10.1109/FOCS.2014.69.

Karchmer, Mauricio. 1989. *Communication Complexity: A New Approach to Circuit Depth*. 1st ed. USA: The MIT Press.

Kushilevitz, Eyal, and Noam Nisan. 1997. *Communication Complexity*. 1st ed. USA: Cambridge University Press.

O'Donnell, Ryan. 2020a. "Basics of Communication Complexity." https://www.youtube.com/watch?v=mQQ36cDnmR8.

———. 2020b. "Deterministic Communication Complexity." https://www.youtube.com/watch?v=zFHWmmThdT .

Pitassi, Toniann. 2014. "Communication Complexity: Applications and New Directions: Lecture 1." https://www.cs.toronto.edu/ toni/Courses/CommComplexity2014/Lectures/lecture1.pdf.

Yao, Andrew Chi-Chih. 1979. "Some Complexity Questions Related to Distributive Computing (Preliminary Report)." In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, 209–13. STOC '79. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/800135.804414.

Zhang, Qin. 2016. "Lower Bound Techniques for Multiparty Communication Complexity." https://www.youtube.com/watch?v=qNTEZNEGAJ8.