UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

A HIERARCHY BETWEEN DISTRIBUTED COMMUNICATION MODELS COMBINING BROADCAST, CONGEST AND LOCAL ROUNDS

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN MATEMÁTICAS APLICADAS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MATEMÁTICO

PABLO VICENTE PAREDES HAZ

PROFESOR GUÍA:
IVÁN RAPAPORT ZIMERMANN
PROFESOR CO-GUÍA:
PEDRO MONTEALEGRE BARBA

COMISIÓN:
JOSÉ SOTO SAN MARTÍN

Este trabajo ha sido financiado por:
CMM ANID BASAL FB210005

SANTIAGO DE CHILE
2022

UNA JERARQUÍA ENTRE MODELOS DE COMUNICACIÓN DISTRIBUIDA COMBINANDO RONDAS BROADCAST, CONGEST Y LOCAL

En esta tesis, enmarcada en computación distribuida, se estudian diferentes modelos de comunicación distribuida construidos a partir de la combinación de rondas de modelos preexistentes tales como broadcast, congest y local. Especfícamente, buscamos construir un *lattice*, ordenado por inclusión, de los distintos lenguajes de grafos que se pueden decidir a través de combinaciones de los modelos anteriores.

Primero, se estudia si existen inclusiones entre los lenguajes que se pueden decidir en una ronda broadcast, una ronda local y una ronda congest. Segundo, se comparan los lenguajes que pueden ser decididos en todas las combinaciones de tamaño dos de los modelos anteriores. Tercero se estudian las combinaciones de tamaño tres de rondas broadcast y local. Cuarto, se estudian algunos modelos con un número constante de rondas broadcast y local.

Finalmente, se prueba una cota inferior de un problema de comunicación que se usa para hacer reducciones en las demostraciones anteriores.

A HIERARCHY BETWEEN DISTRIBUTED COMMUNICATION MODELS COMBINING BROADCAST, CONGEST AND LOCAL ROUNDS

In this thesis, framed in distributed computing, we study different distributed communication models constructed by combining rounds of already existing models such as broadcast, congest and local. Specifically, we seek to build a lattice, ordered by inclusion, of the distinct languages of graphs that can be decided by combinations of the previous models.

First, we study if there are inclusions between the languages that can be decided in one broadcast round, one local round and one congest round. Second, we compare the languages that can be decided in every 2-size combination of the previous models. Third, we study the 3-size combinations of broadcast and local rounds. Fourth, we study some models with a constant number of broadcast and local rounds.

Finally, we give a lower bound of a communication problem that we used for reductions in the previous proofs.

Para mi tati Fernando y abuela Ana,
Q.E.P.D.

Agradecimientos

Quiero agradecer, en primer lugar, a mi papá Ricardo, por ser el que, desde chico, potenció mis intereses por las matemáticas y me ha acompañado a lo largo de toda mi vida en mi desarrollo académico y, por supuesto, en mi desarrollo como persona. También, agradecimientos a mis hermanas Valentina y Antonia, y a mi hermano Pedro, con los que no hemos dejado de compartir el interés por el conocimiento y siempre han estado apoyando sin importar la distancia. Gracias también a la Érica que ha sido parte de esta familia y a todas las personas que me han acompañado desde que nací.

Segundo, a la gente que me acompañó en la universidad. Primero, al Nachín, Chama, Mati, Skill, JJ, Bryan por hacer de plan común un período lejos de ser solamente académico, sino también que de amistad y conexión. Después, a toda la gente que conocí en el DIM, en especial a mis amigos Mato, Barri, Rola, Rondrigo, Iván, Pingüino, Choco, Jipi. Sin ustedes no habría podido sobrevivir la carrera. También quería agradecer a las personas del Preu en Cuarentena, los cuales hicieron posible un proyecto del cual estoy tremendamente orgulloso.

Tercero, a toda la gente del Staff de Andrée Wolf, con la cual he crecido, pertenecido y compartido los valores que considero más importantes. Mención especial a la Marti, Luli y Vale por haber generado estos lazos de amistad tan fuertes este último año.

Finalmente, a mis profesores de la universidad y funcionarios, en especial a Eterin, Karen y Óscar, personas que siempre estuvieron atentas y me salvaron de muchísimos problemas. Agradezco a Pedro Montealegre e Iván Rapaport, mis profesores guía, los cuales hicieron tremenda pega de recibirme como alumno y acompañarme semana a semana con la mejor disposición para poder sacar esta tesis adelante.

Table of Content

Table Index

Illustrations Index

Introduction

Since the invention of modern computers (1945) until the 1980's, only centralized computing was considered, but the development of powerful microprocessors and high speed local area networks (LAN) with the potential of connecting thousands of computers with a transfer speed of millions of bits per second, led the path to distributed systems, such as World Wide Web.

Distributed computing models the situation where a big number of processors operate simultaneously sending messages to each other.

This thesis analyzes the relative power of distributed computing models for networks, all resulting from the combination of standard synchronous models such as LOCAL and $\mathsf{CONGEST}$ [1], as well as Broadcast Congested Clique (BCC) [2]. Each of these three models has its strengths and limitations.

In particular, $\mathsf{CONGEST}$ assumes the ability for each node to send a specific message to each of its neighbors at every round (even in a clique). However, the communication links have limited bandwidth. Specifically, at most $O(\log n)$ bits can be sent through any link during a round, in n-node networks. LOCAL assumes a link with unlimited bandwidth between any two neighboring nodes, but the information acquired by any node u after $t \geq 0$ rounds of communication is limited to the data available at nodes at distance at most t from u in the network. Finally, BCC supports all-to-all communications between the nodes, and thus does not suffer from the locality constraint of LOCAL and $\mathsf{CONGEST}$. However, at each round, each node is bounded to send a *same* $O(\log n)$-bit message to all the other nodes.

In this thesis, we investigate the power of models resulting from combining these three models, in order to take advantage of their positive aspects without suffering from their negative ones.

There are many existing communication networks that use a *combination* of multiple models to maximize cost-efficiency. For example, an organization can set up a hybrid Wide Area Network (WAN) by combining their own communication infrastructure with connections via Internet [3]. Also, the emerging of 5G allows smartphones to directly communicate through their wireless interface to other smartphones or smart devices, in contrast to the communication by cellular infrastructure.

For the sake of formally combining models, we focus on the standard framework of distributed *decision* problems on labeled graphs (see [4]). Such problems are defined by a collection $\mathcal{L}$ of pairs (G, ℓ), where $G = (V, E)$ is a graph, and $\ell : V \to \{0,1\}^*$ is a function assigning a label $\ell(u) \in \{0,1\}^*$ to every $u \in V$. Such a set $\mathcal{L}$ is called a *distributed language*. For instance, deciding whether a certain set U of nodes in a graph G forms a vertex cover can be modeled by the language

$$\texttt{vertex-cover} = \Big\{(G, \ell) : \forall \{u, v\} \in E(G),\ \ell(u) = 1 \vee \ell(v) = 1\Big\},$$

by labeling 1 all the vertices in U, and 0 all the other vertices. Similarly, deciding triangle-freeness can be modeled by the language $\texttt{triangle-freeness} = \{(G, \ell) : C_3 \not\preceq G\}$, where $H \preceq G$ denotes that H is a subgraph of G, and deciding whether a graph is planar can be captured by the language $\texttt{planarity} = \{(G, \ell) : G \text{ is planar}\}$. A distributed algorithm $\mathcal{A}$ *decides* $\mathcal{L}$ if every node running $\mathcal{A}$ eventually accepts or rejects, and the following condition is satisfied: for every labeled graph (G, ℓ),

$$(G, \ell) \in \mathcal{L} \iff \text{all nodes accept.}$$

That is, every node should accept in a yes-instance (i.e., an instance $(G, \ell) \in \mathcal{L}$), and, in a no-instance (i.e., an instance $(G, \ell) \notin \mathcal{L}$), at least one node must reject.

For every $t \geq 0$, let us denote by $\mathbf{L}^t$ the set of distributed languages $\mathcal{L}$ for which there is a t-round algorithm in the LOCAL model deciding $\mathcal{L}$. The sets $\mathbf{C}^t$ and $\mathbf{B}^t$ are defined similarly, for the $\mathsf{CONGEST}$ and BCC models, respectively. Also, we define $\mathbf{L}^* = \cup_{t\geq 0}\mathbf{L}^t$, $\mathbf{C}^* = \cup_{t\geq 0}\mathbf{C}^t$, and $\mathbf{B}^* = \cup_{t\geq 0}\mathbf{B}^t$. So, in particular, $\mathbf{L}^*$ is the class of distributed languages that can be decided in a constant number of rounds in the LOCAL model.

The three models under consideration, i.e., LOCAL, $\mathsf{CONGEST}$, and BCC exhibit very different behaviors with respect to decision problems. For instance, it is known [5, 6] that

$$\texttt{triangle-freeness} \in \mathbf{L} \setminus (\mathbf{B}^* \cup \mathbf{C}^*),$$

whenever one assumes, as we do in this thesis, that, for all models under consideration, every node is initially aware of the identifiers[1] of its neighbors. On the other hand, it is also known [7] that

$$\texttt{planarity} \in \mathbf{B} \setminus \mathbf{L}^*.$$

This means that while no LOCAL algorithms can decide planarity in a constant number of rounds, there is a 1-round BCC algorithm deciding planarity, and while no BCC algorithms can decide triangle-freeness in a constant number of rounds, there is a 1-round LOCAL algorithm deciding triangle-freeness. So, if one allows LOCAL algorithms to do just a single round of all-to-all communication, as in BCC, then both triangle-freeness and planarity can be solved in a constant number of rounds, hence increasing the computational power of LOCAL dramatically.

This observation led us to investigate scenarios such as the case in which the $\mathsf{CONGEST}$ model is enhanced by allowing nodes to perform few rounds in either LOCAL, or BCC. What would be the computing power of such a *hybrid* model? For answering this question, for a collection of non-negative integers $\alpha_1, \dots, \alpha_k$, $\beta_1, \dots, \beta_k$, and $\gamma_1, \dots, \gamma_k$, we define the set

$$\prod_{i=1}^{k} \mathbf{L}^{\alpha_i}\mathbf{B}^{\beta_i}\mathbf{C}^{\gamma_i}$$

or more explicitly,

$$\mathbf{L}^{\alpha_1}\mathbf{B}^{\beta_1}\mathbf{C}^{\gamma_1}\mathbf{L}^{\alpha_2}\mathbf{B}^{\beta_2}\mathbf{C}^{\gamma_2}\cdots\mathbf{L}^{\alpha_k}\mathbf{B}^{\beta_k}\mathbf{C}^{\gamma_k}$$

as the class of decision languages $\mathcal{L}$ which can be decided by an algorithm performing $\alpha_1 \geq 0$ rounds of LOCAL, followed by $\beta_1 \geq 0$ rounds of BCC, followed by $\gamma_1 \geq 0$ rounds of $\mathsf{CONGEST}$,

[1] In each of the models, every node u of a n-node network $G = (V, E)$ is supposed to be provided with an identifier $\mathsf{id}(u)$, where $\mathsf{id} : V \to [1, N]$ is one-to-one, and $N(n) = \mathrm{poly}(n)$, i.e., all identifiers can be stored on $O(\log n)$ bits in n-node networks. We also assume that all nodes are initially aware of the size n of the network, merely because this is the case in model BCC.

followed by $\alpha_2 \geq 0$ rounds of LOCAL, etc., up to $\gamma_k \geq 0$ rounds of CONGEST. For instance, we have

$$\{\mathsf{planarity}, \mathsf{triangle\text{-}freeness}\} \subseteq \mathbf{LB} \cap \mathbf{BL}.$$

However, how do compare **LB** and **BL**? And what about **CB** vs. **BC**, and **LC** vs. **CL**? These are the kinds of questions that we are studying in this thesis. In the long-term perspective, this line of research is motivated by the following question. Let $\mathcal{L}$ be a fixed distributed language, and let us assume that a round of LOCAL costs a (say, for acquiring high-throughput channels), that a round of BCC costs b (say, for benefiting of facilities supporting all-to-all communications), and that a round of CONGEST costs c. The goal is to minimize the total cost of an algorithm deciding $\mathcal{L}$ in a constant number of rounds, that is, to solve the following minimization problem:

$$\min_{\prod_{i=1}^{k} \mathbf{L}^{\alpha_i}\mathbf{B}^{\beta_i}\mathbf{C}^{\gamma_i} \ni \mathcal{L}} \left(a\sum_{i=1}^{k} \alpha_i + b\sum_{i=1}^{k} \beta_i + c\sum_{i=1}^{k} \gamma_i \right). \tag{0.1}$$

Note that, for $a = b = c = 1$, Eq. (0.1) corresponds to minimizing the number of rounds for deciding $\mathcal{L}$ when using a combination of the communication facilities provided by LOCAL, CONGEST, and BCC. For instance, deciding whether a graph is C_k-free can be achieved in $\lfloor \frac{k}{2} \rfloor$ rounds in LOCAL, that is, C_k-$\mathsf{freeness} \in \mathbf{L}^{\lfloor k/2 \rfloor}$. Eq. (0.1) is asking whether deciding C_k-$\mathsf{freeness}$ could be achieved at a lower cost by combining LOCAL, CONGEST, and BCC. For tackling Eq. (0.1), we need a better understanding of the fundamental effects resulting from combining these models.

Results

We provide a series of separation results between 2-round hybrid models. In particular, we show that **BC** and **CB** are incomparable. That is, there are languages in $\mathbf{BC} \setminus \mathbf{CB}$, and languages in $\mathbf{CB} \setminus \mathbf{BC}$. In fact, we show stronger separation results, by establishing that $\mathbf{BC} \setminus \mathbf{C}^*\mathbf{B} \neq \varnothing$, and $\mathbf{CB} \setminus \mathbf{BL}^* \neq \varnothing$. That is, in particular, there are languages that can be decided by a 2-round algorithm performing a single BCC round followed by one CONGEST round, which cannot be decided by any algorithm performing k CONGEST rounds followed by a single BCC round, for any $k \geq 1$.

On the positive side, it is shown that, for any non-negative integers $\alpha_1, \dots, \alpha_k, \beta_1, \dots, \beta_k$,

$$\prod_{i=1}^{k} \mathbf{L}^{\alpha_i}\mathbf{B}^{\beta_i} \subseteq \mathbf{L}^{\sum_{i=1}^{k} \alpha_i}\mathbf{B}^{\sum_{i=1}^{k} \beta_i}. \tag{0.2}$$

That is, if a language $\mathcal{L}$ can be decided by a t-round algorithm alternating LOCAL and BCC rounds, then $\mathcal{L}$ can be decided by a t-round algorithm performing all its LOCAL rounds first, and then all its BCC rounds. So, in particular $\mathbf{BL} \subseteq \mathbf{LB}$. We show that this separation is strict. A consequence of Eq. (0.2) is that the largest class of languages among all the ones considered in this thesis is $\mathbf{L}^*\mathbf{B}^*$, that is, languages that can be decided by algorithms performing k LOCAL rounds followed by k' BCC rounds, for some $k \geq 0$ and $k' \geq 0$. Thus, Eq. (0.1) should be studied for languages $\mathcal{L} \in \mathbf{L}^*\mathbf{B}^*$.

Interestingly, the separation results hold even for randomized protocols, which can err with probability at most $\epsilon \leq 1/5$. That is, in particular, there is a language $\mathcal{L} \in \mathbf{CB}$ (i.e., that can be decided by a deterministic 2-round algorithm) which cannot be decided with error probability at most $1/5$ by any randomized algorithm performing one BCC round first, followed by k LOCAL rounds, for any $k \geq 1$. All our results about 2-rounds hybrid models are summarized on Figure 0.1.

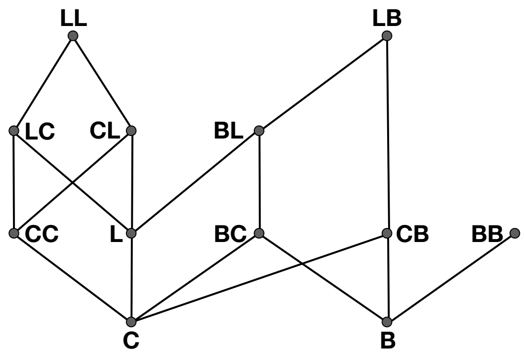

Figure 0.1: *The lattice of 2-round hybrid models. An edge between a set of languages $\mathbf{S}_1$ and a set $\mathbf{S}_2$, where $\mathbf{S}_1$ is at a level lower than $\mathbf{S}_2$, indicates that $\mathbf{S}_1 \subseteq \mathbf{S}_2$. In fact, all inclusions are strict. Transitive edges are not displayed. Two sets that are not connected by a monotone path are incomparable. For instance, **CB** and **BL** are incomparable, while **BC** $\subseteq$ **LB**.*

Then, we show separation problems for multiple rounds of one type of model. In particular, it is showed that, for every $k \in \mathbb{N}$, there is a language separating $k+1$ BCC rounds from k BCC rounds. The same result is proved with LOCAL rounds. Then, we show some separation languages for 3-round combinations of BCC and LOCAL rounds.

Finally, it is defined a two-party communication problem ($\texttt{XOR-index}$) in order to prove lower bounds by doing reductions to this problem, where Alice receives $x \in \{0,1\}^n$, $i \in [n]$ and Bob receives $y \in \{0,1\}^n$, $j \in [n]$. Then, the task is that, after one round of simultaneous communication, Alice outputs a boolean out_A and Bob outputs a boolean out_B such that $\text{out}_A \wedge \text{out}_B = x_j \oplus y_i$. It is showed that for $\varepsilon < 1/5$, $CC(\texttt{XOR-index}, \varepsilon) = \Omega(n)$.

Chapter 1

Preliminaries

In this chapter we give basic definitions, notations and lemmas that will be used throughout the thesis. We start with basic concepts. Then, we introduce the distributed models that will be the main focus in this thesis.

1.1. Basic Definitions

A *simple undirected graph*, from here on simply refered as a *graph*, is a pair of sets $G = (V, E)$ where V is a finite set, called *vertex set*, and $E \subseteq \binom{V}{2}$ is called *edge set*. Whenever we consider an edge $e \in E$, we shall refer it simply as $e = uv$, where $u, v \in V$ are the vertices joined by the edge. In that case, we say u and v are *adjacent* vertices. If several graphs are being considered, we will use the notation $V(G)$ and $E(G)$ for the vertex set and edge set of graph G.

For $U \subseteq V$ we denote $N(U)$ the set of adjacent vertices of U, this is $N(U) = \{v \in V \setminus U \mid \text{there exists } e \in E,\ e = uv\}$. We call $N(u)$ the set of *neighbors* of u. We denote $d(u) := |N(u)|$, the *degree* of u.

With the previous basic notions, we can define some particular type of graphs:

Definition 1.1 (Path) *A path is a non empty graph $P = (V, E)$ with:*

$$V = \{v_0, v_1, \ldots, v_k\}$$
$$E = \{v_0v_1, v_1v_2, \ldots, v_{k-1}v_k\}$$

Definition 1.2 (Cycle) *A cycle is a path as previously defined with the addition of the edge v_kv_0.*

Definition 1.3 (Induced Subgraph) *Let $G = (V, E)$ be a graph and $U \subseteq V$. The graph induced by U is the subgraph $G[U] = (U, E(U))$, where $E(U) = \{e \in \binom{U}{2} \mid e \in E\}$*

Definition 1.4 (Connected graph) *Let $G = (V, E)$ be a graph. We say that G is connected if for any pair of vertices $u, v \in G$ there exists a set of vertices P, such that the induced graph of P is a path with extremes in u and v.*

Definition 1.5 (Tree) *A tree is a connected graph without any cycles.*

1.2. Communication Models

Throughout this thesis we work with three different distributed models. We now define these models and their respective deterministic and probabilistic protocols.

Let G be a connected n vertex graph. Let $\ell : V(G) \longrightarrow \{0,1\}^*$ be an input function that assign labels to each node. Let $id : V \longrightarrow \{1, \ldots, \text{poly}(n)\}$ be a one-to-one function that gives an identification number to each node in the graph. Notice that $id(v)$ can be encoded using $O(\log n)$ bits for every $v \in V(G)$.

In a message-passing communication model, information is sent and received by vertices of G. In every round, each vertex can send a $O(b)$ size message to a determined set S of other vertices (neighbors, for example), with S and b depending on the model.

Every node in the graph is honest, this means that the information sent by a vertex is reliable and the connections are safe (the models are error free). Also, there is no limit in the computational power that each node has.

In each round of the algorithm, every node $v \in V$ can decide whether to stop or continue. An algorithm stops when all nodes decide to stop. The (round) complexity is the number of synchronous rounds required for the algorithm to stop.

The following models will be the ones used in this thesis.

1.2.1. LOCAL Model

Definition 1.6 *Let $G = (V, E)$ be a connected arbitrary graph. The* LOCAL *model is a synchronous message-passing communication model, where in each round, every vertex from V can send an unlimited size message to each of its neighbors given by E.*

In every LOCAL round, the node $v \in V$ sends all the information it knows about G to its neighbors. So, k local rounds will be equivalent to knowing the $k-$radius neighborhood that we will denote $N_k(v)$ for each $v \in V$.

The LOCAL model has been introduced in [8] at the beginning of the 1990s, when the celebrated $\Omega(\log^* n)$ lower bound on the number of rounds for computing a 3-coloring or a maximal independent set (MIS) in the n-node cycle was proved.

1.2.2. CONGEST Model

Definition 1.7 *Let $G = (V, E)$ be a connected arbitrary graph. The* CONGEST *model is a synchronous message-passing model where, in each round, every vertex $v \in V$ sends specific messages to its neighbors given by E, with the condition that the message size is $O(\log n)$, with $n = |V|$.*

The CONGEST model is a weaker variant of the LOCAL model in which the size of the messages exchanged at each round between neighbors is bounded to $O(\log n)$ bits, or B bits in the parametrized version of the model. This bound on the message size creates bottlenecks limiting the power of algorithms under this model. A fruitful line of research has established several non-trivial lower bounds on the round-complexity of CONGEST algorithms, by reduction from communication complexity problems (see for instance [9–13]). Nevertheless, several problems can still be solved in a constant number of rounds in CONGEST. This is for instance the case of computing a $(2 + \varepsilon)$-approximation of minimum vertex cover which can be done in $O(\log \Delta / \log \log \Delta)$ rounds [14] in graphs with maximum degree Δ. Also, *testing* (a weaker variant of decision, a la property-testing) the presence of specific subgraphs like small cliques or short cycles can be done in a constant number of rounds in CONGEST (see, e.g., [15–19]).

1.2.3. BCC Model

Definition 1.8 *Let $G = (V, E)$ be a connected arbitrary graph. The* BCC *model (broadcast congested clique), is another synchronous message-passing model where, in each round, every vertex $v \in V$ sends a unique message to all other nodes in the graph, with the condition that the message size is $O(\log n)$, with $n = |V|$.*

The *congested clique model* [2, 20] has first been introduced in its *unicast* version (UCC), where every node is allowed to send potentially different $O(\log n)$-bit messages to each of the other $n-1$ nodes at every round. In the UCC model, many natural problems can be solved in a constant number of rounds [21–23]. The *broadcast* variant of the congested clique, namely the BCC model, is weaker that the unicast variant, and lower bounds on the round-complexity of problems in the BCC model have been established, again by reduction to communication complexity problems. This is the case of problems such as detecting the presence of particular subgraphs [2], detecting planted cliques [24], or approximating the diameter of the network [25]. There are also positive results in what can be done using, for example, one BCC round [26]. An application of this model are typically global communications such as smartphones sending messages to another smartphones using cellular data.

1.2.4. Alternating models

In the present thesis we will study the hierarchy of all possible permutations of LOCAL BCC and CONGEST rounds. We denote the languages that can be accepted in a combination of these models as follows: $\mathbf{L}$, $\mathbf{B}$ and $\mathbf{C}$ are the set of languages that can be decided in one LOCAL, one BCC and one CONGEST round respectively. Then, the notation of combinations is simple, for example, $\mathbf{LB}^2\mathbf{C}$ is the set of languages that can be decided in one LOCAL round, followed by two BCC rounds, and finally followed by one CONGEST round.

1.3. Communication Complexity

We now define a tool that will be used throughout this thesis to prove some results by making reductions using communication problems.

Let $f : X \times Y \longrightarrow Z$ where in general we assume $X = Y = \{0,1\}^n$ and $Z = \{0,1\}$. There are two players, Alice and Bob (in some occasions we add a third player Charlie), such that Alice receives an input $x \in X$ and Bob receives $y \in Y$. In this thesis we focus on 2-way 1-round protocols, which means that each player sends only one message to the other player and they both see the messages at the same time. After seeing the respective message, each player outputs some $z \in Z$. Typically, one expect that both players output the same $z = f(x, y)$. But, as we are going to explain later, in this thesis we are going to consider another, less classical notion of correctness.

Definition 1.9 *The communication complexity of a communication problem P is the minimum size of bits exchanged between the two players in the worst case and it will be denoted as $CC(P)$. Furthermore, we denote $CC^1(P, \varepsilon)$ to the communication complexity of the best 2-way 1-round randomized protocol solving P with error probability at most ε.*

1.4. Disjointness

Several results in this thesis are proved by reduction to the communication complexity problem *set disjointness* (DISJ). Given two sets $x, y \subseteq [n]$ (usually represented as indicator vectors $x, y \in \{0,1\}^n$), the task is to decide whether $x \cap y = \varnothing$ (or equivalently whether $x_i \wedge y_i = 0$ for all $i \in [n]$). Formally, Alice receives x as input, and Bob receives y. The task is to compute

$$\mathrm{DISJ}(x,y) = \begin{cases} 1 & \text{if } x \cap y = \varnothing \\ 0 & \text{otherwise.} \end{cases}$$

The communication complexity of DISJ is high, as shown below.

Lemma 1.1 (Theorem 6.19 in [27]) *For every* $\varepsilon > 0$*,* $CC(\mathit{DISJ}, \frac{1}{2} - \varepsilon) = \Omega(\varepsilon^2 n)$*.*

Definition 1.10 *Let* $x, y, z \in \{0,1\}^n$ *.The 3-party number-on-forehead problem (3-NOF-DISJ) is a communication problem where the players Alice, Bob and Charlie receive the following inputs:*

- *Alice receives* y, z*.*
- *Bob receives* x, z*.*
- *Charlie receives* x, y*.*

All players must decide if there exists $i \in [n]$ *such that* $x_i = y_i = z_i$*.*

Lemma 1.2 (Result in [2]) $CC(\textit{3-NOF-DISJ}) = \Omega(n)$

Chapter 2

One Round

In this chapter we focus on what can be done in one round of each model.

Theorem 2.1 *There is a language that belongs to $\boldsymbol{L}$, but does not belong to $\boldsymbol{B}^*$.*

Proof. Lets consider the language `triangle-freeness` $= \{G \mid G$ does not contain C_3 as a subgraph$\}$.

It is easy to show that `triangle-freeness` can be accepted in one LOCAL round, since every node $v \in V$ knows if it belongs to a C_3 or not after one round, so v accepts if and only if v does not belong to a C_3.

In order to show that `triangle-freeness` belongs to $\mathbf{B}^*$, we make a reduction from 3-party number-on-forehead to `triangle-freeness`.

Let us assume, for the purpose of contradiction, that `triangle-freeness` can be decided in k BCC rounds, with k a positive integer.

Let $(x, y, z) \in \{0,1\}^{3m}$ be an instance of the 3-party number-on-forehead problem, where Alice receives (y, z), Bob receives (x, z) and Charlie receives (x, y). Without loss of generality, let $m = n^2/e^{\sqrt{\log n}}$. We know, by Lemma 0.20, that the communication complexity of 3-party number-on-forehead problem is $\Omega(m)$.

Alice, Bob and Charlie consider the graph $G = (V, E)$ with the following properties

1. G is a tripartite graph with $V = V_1 \sqcup V_2 \sqcup V_3$, where $|V_1| = |V_2| = n$ and $|V_3| = n/3$
2. G contains $n^2/e^{\sqrt{\log n}}$ triangles, and each edge of G belongs to exactly one triangle.

The existence of this graph is showed in [28].

Alice, Bob and Charlie enumerate each triangle in G from 1 to m in the same way. Denote E_{ij} the set of edges $e = uv$, with $u \in V_i$ and $v \in V_j$ with $i, j \in \{1, 2, 3\}$. Then, for each triangle i, there is an edge in E_{12}, E_{13} and E_{23}, since G is tripartite.

Alice, Bob and Charlie simulate the virtual graph $\tilde{G} \preceq G$ as it follows: $x \in \{0,1\}^m$ as the edges in E_{23}, this is, $x_i = 1$ if and only if the edge in E_{23} that belongs to the $i-$th triangle is in G. In the same way, they simulate y as the edges in E_{13} and z as the edges in E_{12}. By construction, we have that $\tilde{G} \in$ `triangle-freeness` if and only if (x, y, z) is a yes-instance of 3-party number-on-forehead problem.

Since Alice knows all the edges in $\tilde{G}$ that have an extremity in V_1, she can simulate the messages that vertices in V_1 will send in the first BCC round of the protocol Π deciding `triangle-freeness` (let M_1 be the union of messages broadcasted in the first round by vertices in V_1). Moreover, Bob can simulate in the same way the messages that vertices in V_2 will

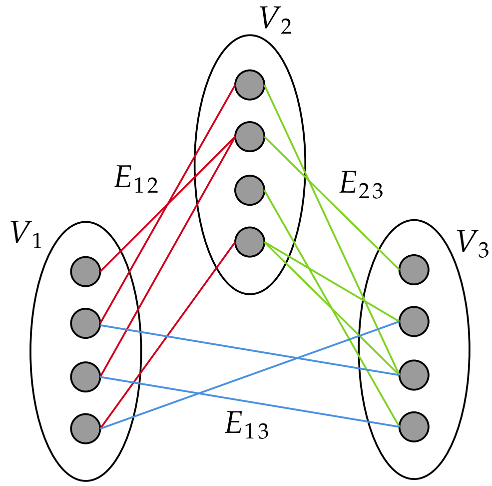

Figure 2.1: *Tripartite graph with* $V = V_1 \sqcup V_2 \sqcup V_3$ *and* $E = E_{12} \sqcup E_{23} \sqcup E_{13}$

send in the first BCC round, and Charlie can do the same with vertices in V_3 (M_2 and M_3 are defined similarly). Then, they can exchange M_1, M_2, M_3. Since every message from the first round is sent, Alice, Bob and Charlie can simulate the messages from the next round sent by V_1, V_2, V_3 respectively. Inductively, they exchange these messages until the k-th round in which Alice accepts if every vertex in V_1 accepts, Bob accepts if every vertex in V_2 accepts and Charlie accepts if every vertex in V_3 accepts.

Since the size of every message that Alice, Bob and Charlie send is $O(n \log n)$ and they are solving the 3-party number-on-forehead problem as well, the number of rounds R that are needed in the protocol must satisfy $R \cdot n \log n = \Omega(n^2/e^{\sqrt{\log n}})$ which is not a constant number of rounds, hence a contradiction.

□

Theorem 2.2 *There is a language that belongs to* $\boldsymbol{B}$*, but does not belong to* $\boldsymbol{L}^*$.

Proof. Lets consider the language (At Most One Selected). This is, all the graphs in which every vertex is labeled with a zero, except for at most one vertex labeled with a one.

Notice that AMOS can be accepted in $\mathbf{B}$, since in the BCC round, every $v \in V$ sends $\ell(v)$. Then, all vertices know $|\ell^{-1}(\{1\})|$ and accepts if there is only one vertex labeled with a one. Otherwise, they reject.

Moreover, AMOS cannot be accepted in $\mathbf{L}^*$. For the purpose of contradiction, let us assume that AMOS can be decided in k LOCAL rounds, with k a positive integer. Let (G, ℓ) where G is a $4k$-path $v_1, v_2, \ldots, v_{4k}$ and $\ell(v_1) = \ell(v_{4k}) = 1$ and $\ell(v) = 0$ for all $v \in V(G) \setminus \{v_1, v_{4k}\}$. Let $(\tilde{G}, \tilde{\ell})$ be defined exactly as (G, ℓ) with the exception that $\ell(v_{4k}) = 0$. Hence, we have that (G, ℓ) does not belong to AMOS, whereas $(\tilde{G}, \tilde{\ell})$ does.

Note that the messages sent by v_2 to v_1 in the k rounds of the protocol are exactly the same since they does not depend on the value of $\ell(v_{4k})$. Then, we have that v_1 cannot distinguish a yes-instance from a no-instance, contradicting the determinism of the protocol.

□

Corolary 2.1 *There is a language that belongs to* $\boldsymbol{B}$ *but does not belong to* $\boldsymbol{C}^*$.

Theorem 2.3 *There is a language that belongs to* $\boldsymbol{C}$*, but does not belong to* $\boldsymbol{B}^*$. *This result* *holds even for randomized decision algorithms which may err with probability* $\frac{1}{2} - \varepsilon$*, for every* $\varepsilon > 0$*.*

Proof. Let us consider the distributed language

$$\textsf{disjointness-on-clique} = \{(K_n, \ell) \mid (\ell : V(K_n) \mapsto \{0,1\}^n) \wedge (\forall i \in [n], \exists v, \ell(v)_i = 0)\},$$

where K_n is the n-node clique, and $\ell(v)_i$ is the i-th entry of the vector $\ell(v)$.

We first show that $\textsf{disjointness-on-clique} \in \mathbf{C}$. Note first, that, in one round of $\textsf{CONGEST}$, the nodes can check whether they are in a clique. Indeed, recall that every node knows n, and therefore a node with degree less than $n-1$ rejects. Every node orders all nodes, including itself, according to their ID's, providing every node with a rank. Note that all nodes ranks the nodes the same. During the $\textsf{CONGEST}$ round, each node v sends $\ell(v)_i$ to the node with rank i (which could be itself). After the round of communication, the node v with rank i has the set $\{\ell(w)_i : w \in V(K_n)\}$. This node accepts if there exists $w \in V(K_n)$ such that $\ell(w)_i = 0$, and it rejects otherwise.

Let $k \in \mathbb{N}$. We show that $\textsf{disjointness-on-clique} \notin \mathbf{B}^k$. For establishing a contradiction, let us assume that there exists a k-round $\textsf{BCC}$ algorithm $\mathcal{A}$ deciding $\textsf{disjointness-on-clique}$ with error probability $\frac{1}{2} - \varepsilon$. We show how to use $\mathcal{A}$ for solving DISJ. Let $x, y \in \{0,1\}^n$ be an instance of DISJ. Alice and Bob consider the n-node clique K_n, with identifiers from 1 to n. Let $e = \{1, 2\}$ be the edge connecting the nodes with ID 1 and the node with ID 2. The two players consider the labeling ℓ such that $\ell(1) = x$, $\ell(2) = y$, and $\ell(v) = (1, 1, \ldots, 1)$ for every node v with $\mathrm{ID}(v) \geq 3$. Note that Alice does not know $\ell(2)$, and Bob does not know $\ell(1)$. By construction, we have that $\mathcal{A}$ accepts (K_n, ℓ) if and only if $\mathrm{DISJ}(x, y) = 1$. The two players simulate the k $\textsf{BCC}$ rounds of $\mathcal{A}$ as follows. At each round r, Alice sends to Bob the message $m_{1,r}$ broadcasted by the node with ID 1, and Bob sends to Alice the message $m_{2,r}$ broadcasted by the node with ID 2. With this information, Alice and Bob can simulate $\mathcal{A}$, tell each other whether one of the nodes they simulate rejects, and then compute $\mathrm{DISJ}(x, y)$. This protocol for DISJ has communication complexity $O(k \log n)$, a contradiction with Lemma 1.1. $\square$

We do not show that there is a language in $\mathbf{L} \setminus \mathbf{C}^*$, but we separate one $\textsf{LOCAL}$ round from two $\textsf{CONGEST}$ rounds with the same language that separates the $\textsf{LOCAL}$ round with $\mathbf{BB}, \mathbf{BC}$ and $\mathbf{CB}$.

Theorem 2.4 *There is a language that belongs to* $\boldsymbol{L}$ *but does not belong to* $\boldsymbol{BB} \cup \boldsymbol{BC} \cup \boldsymbol{CB} \cup \boldsymbol{CC}$*. This result holds even for randomized decision algorithms which may err with probability* $\frac{1}{2} - \varepsilon$*, for every* $\epsilon > 0$*.*

Proof. Let us consider the following distributed language

$$\begin{aligned}\textsf{disjointness-on-edge} = &\{(PC_{2n}, \ell) \mid (n > 2) \wedge (\ell(u_1) \in \{0,1\}^n) \wedge (\ell(v_1) \in \{0,1\}^n)\\ &\wedge (\mathrm{DISJ}(\ell(u_1), \ell(v_1)) = 1) \wedge (\forall w \notin \{u_1, v_1\}, \ell(w) = \perp)\},\end{aligned}$$

where PC_{2n} is the path or cycle $(u_n, \ldots, u_1, v_1, \ldots v_n)$ with $2n$ nodes. A simple $\textsf{LOCAL}$ algorithm guarantees that $\textsf{disjointness-on-edge} \in \mathbf{L}$, that is, every node of degree > 2 rejects, and u_1 and v_1 exchange their values and accept if and only if $\mathrm{DISJ}(\ell(u_1), \ell(v_1)) = 1$.

Let $\mathbf{S} = \mathbf{BB} \cup \mathbf{BC} \cup \mathbf{CB} \cup \mathbf{CC}$. We now show that $\textsf{disjointness-on-edge} \notin \mathbf{S}$. For establishing a contradiction, let us assume that there exists a 2-round algorithm $\mathcal{A}$ mixing

CONGEST and BCC for deciding disjointness-on-clique with error probability $\frac{1}{2} - \varepsilon$. We show how to use this algorithm to compute DISJ. Let (x, y) be an instance of DISJ. Alice and Bob construct the instance (P, ℓ) of disjointness-on-edge where $\ell(u_1) = x$ and $\ell(v_1) = y$. By construction $\text{DISJ}(x, y) = 1$ if and only if $(P, \ell) \in$ disjointness-on-edge. Of course, Alice does not know $\ell(v_1)$, and Bob does not know $\ell(u_1)$. All messages communicated in the first round of $\mathcal{A}$ by all nodes different from u_1 and v_1 do not depend on (x, y), and can thus be simulated by the players without any communication. Alice and Bob generate and exchange the messages that u_1 and v_1 communicate in the first round of $\mathcal{A}$. If the first round is a CONGEST round, note that each of the two nodes may generate two messages, one for each of their two neighbors. For the second rounds, Alice and Bob have all the information sufficient to compute what messages will be generate by the nodes, excepted for nodes u_1 and v_1, respectively. So Alice and Bob exchange these messages. Alice accepts if all nodes $u_1, \ldots, u_n$ accept, and Bob accept if all node $v_1, \ldots, v_n$ accept. Then they exchange their decision. This protocol computes DISJ with error probability $\frac{1}{2} - \varepsilon$. This is a contradiction with Lemma 1.1 as only $O(\log n)$ bits were exchanged by the two players. □

Chapter 3

Two Rounds

In this chapter we focus on the hierarchy between all combinations of size two that can be done with the three models considered.

Theorem 3.1 *There is a language that belongs to $\boldsymbol{BL}$, but does not belong to $\boldsymbol{B}^* \cup \boldsymbol{L}^*$*

Proof. Lets consider the language `triangle-on-max-degree-freeness` as it follows:

$$\texttt{TOMDF} = \{G \mid \forall v \in V \text{ such that } v \in C_3 \preceq G, \deg(v) < \Delta(G)\}$$

This is, TOMDF is the set of graphs G such that, for every triangle T in G, all nodes in T have a degree smaller than the maximum degree of G.

Note that TOMDF is in $\mathbf{BL}$, since in the BCC round every vertex sends its degree, so every vertex learns the value of $\Delta(G)$. Then, in the LOCAL round, each vertex knows if they belong to a triangle, so it rejects if its degree is the same as $\Delta(G)$. Otherwise, it accepts.

Moreover, TOMDF cannot be decided in $\mathbf{L}^*$. Let $k > 0$ be an integer. Lets consider a $n/2$ size path $\{v_1, v_2, \ldots v_n\}$ in a tree graph T with $n >> k$. Notice that v_2 will not receive the information of $\deg(v_{n-1})$, so it cannot know if $\deg(v) = \Delta(G)$ in k LOCAL rounds. Then, it cannot distinguish an instance where it has smaller degree than $\Delta(G)$, from an instance where there is a node with larger degree. Hence, there is a positive probability that a vertex rejects a yes-instance. It remains to show that TOMDF cannot be decided in $\mathbf{B}^*$.

Let us assume, for the purpose of contradiction, that there exists $k \geq 0$, such that TOMDF can be decided by an algorithm $\mathcal{A}$ performing k BCC rounds, i.e., $\texttt{TOMDF} \in \mathbf{B}^k$. We can use $\mathcal{A}$ to decide `triangle-freeness` in $k+1$ BCC rounds.

Let G be an arbitrary graph. In the first BCC round, every node $v \in V$ broadcasts its identifier $\mathsf{id}(v)$ and its degree $\deg(v)$, and hence learns the maximum degree Δ of G. Then every node simulates $\mathcal{A}$ on the virtual graph G' on $n' = n(\Delta+1) - \sum_v d(v)$ nodes obtained from G by adding a set S_v of $\Delta - d(v)$ pending vertices to each vertex v of G. Every node $v \in V$ simulates $\mathcal{A}$ in G' by simulating its execution on v and on all the nodes in S_v. Specifically, after the first BCC round, v knows the set of ID's used in G, and thus the rank of its ID in this set. Therefore, it can compute the set I composed of the smallest $n' - n$ positive integers that are not used as ID's in G. Furthermore, it can assign ID's to its $\Delta - d(v)$ pending virtual neighbors in G', using its rank and the degrees of all the nodes with lower rank in G, so that

1. the ID of each virtual node is unique in G', and
2. every node of G knows the ID's assigned to the pending virtual neighbors of every other

node in G.

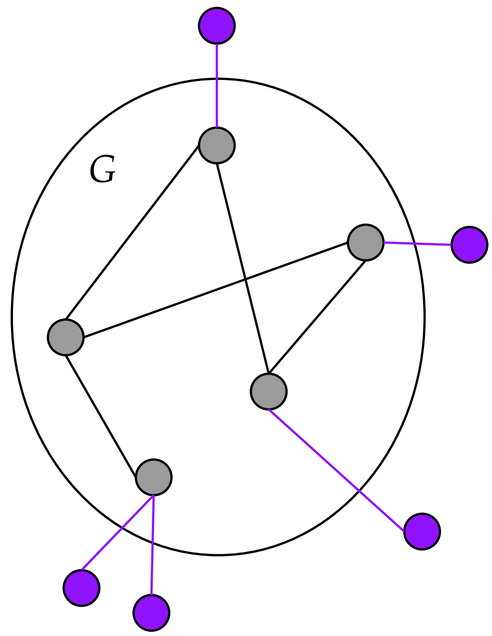

Figure 3.1: *Example of virtual graph G' simulated by the 5-node graph G*

It follows that each node v does not need to simulate the messages broadcasted in $\mathcal{A}$ by the nodes in S_v. In fact, every node v can simulate the behavior of all the virtual nodes in $S = \cup_{u \in V(G')} S_u$ at each round of $\mathcal{A}$. As a consequence, the simulation of $\mathcal{A}$ in G' does not yield any overhead on the number of bits to be broadcasted by each (real) node v running $\mathcal{A}$.

After the k BCC rounds of $\mathcal{A}$ in G' have been simulated, every node v accepts (on G) if itself and all the nodes in S_v accept in $\mathcal{A}$ on G'. Now, by construction, $G' \in \mathtt{TOMDF}$ if and only if G is triangle-free. Since $\mathcal{A}$ decides $\mathtt{TOMDF}$, we get that `triangle-freeness` is decided in $\mathbf{B}^{k+1}$, a contradiction.

$\square$

Theorem 3.2 *There is a language that belongs to $\boldsymbol{LL}$ but does not belong to $\boldsymbol{BB} \cup \boldsymbol{LB} \cup \boldsymbol{CC}$. This result holds even for randomized decision algorithms which may err with probability $\frac{1}{2} - \varepsilon$, for every $\epsilon > 0$.*

Proof. Let us define the distributed language `disjointness-on-path` of pairs (P, ℓ) where $P = u_n, \ldots, u_1, v_1, \ldots v_n$ is a path of length $2n$ ($n > 2$), and $\ell : V(P) \to \{0,1\}^*$ satisfies that $\ell(w) = \perp$ if $w \neq \{u_2, v_2\}$ and $DISJ(\ell(u_2), \ell(v_2)) = 1$. In words, `disjointness-on-path` is the language of paths that have a yes-instance of disjointness in two nodes at distance 2. Trivially `disjointness-on-path` $\in \mathbf{L}^2$: in a protocol every node except v_1 and u_1 accept. Nodes v_1 and u_1 learn the values of $\ell(v_2)$ and $\ell(u_2)$ and accept if and only if $DISJ(\ell(u_2), \ell(v_2)) = 1$.

Let $\mathbf{S} = (\mathbf{BB} \cup \mathbf{LB} \cup \mathbf{CC})$. We now show that `disjointness-on-path` $\notin \mathbf{S}$. By contradiction, let us assume that there exists an $1/2 - \epsilon$-error algorithm $\mathcal{A}$ in $\mathbf{S}$ solving `disjointness-on-path`. We show how to define a two-player, $1/2 - \epsilon$-error protocol Π for DISJ. Let (x, y) be an instance of DISJ and consider the instance (P, ℓ^*) of `disjointness-on-path` where $\ell^*(u_2) = x$ and $\ell^*(v_2) = y$. Clearly, $(P, \ell^*) \in$ `disjointness-on-path` if and only if $(x, y) \in$ DISJ .

First, let us suppose that $\mathcal{A}$ is a protocol consisting in two BCC rounds. In this case Π consists in two rounds of communication. Initially, using x Alice simulates the first round of

$\mathcal{A}$ in every node of P except v_2 obtaining messages $\{m(w) : w \in V(P) \setminus \{v_2\}\}$. Similarly, using y Bob simulates the first round of $\mathcal{A}$ on every node of P except u_2, obtaining messages $\{m(w) : w \in V(P) \setminus \{u_2\}\}$. Then, in the first round of Π, Alice and Bob interchange $m(u_2)$ and $m(v_2)$, in order to obtain the pack of messages $M = \{m(w) : w \in V(P)\}$ that every node receives in the first BCC round of $\mathcal{A}$. The second round is very similar: using x and M Alice simulates $\mathcal{A}$ on every node except v_2, obtaining the pack of messages communicated in the second round of $\mathcal{A}$ except for the message of v_2. At the same time using y and M Bob simulates $\mathcal{A}$ on every node except u_2, obtaining the pack of messages communicated in the second round of $\mathcal{A}$ except for the message of u_2. Then, in the second round of Π Alice and Bob interchange the second messages of u_2 and v_2. Finally, Alice simulates the output of every node except v_2 and accept if every node accepts. Bob simulates the output of every node except u_2 and accept if every node accepts. We deduce that Π is an ϵ-error protocol for `disjointness`. Nevertheless, the number of bits communicated in the execution of Π corresponds to the two messages broadcasted by u_2 and v_2, which is $O(\log n)$. This contradicts Lemma 1.1. We deduce that `disjointness-on-path` does not belong to **BB**.

Now, let us suppose that $\mathcal{A}$ is a protocol consisting in a LOCAL round followed by a BCC round. In this case Π consists in just one round of communication. Initially, using x Alice simulates first round of $\mathcal{A}$ obtaining that nodes u_1, u_2 and u_3 learn the value of x, and all other nodes have no information of x. Then, Alice simulates $\mathcal{A}$ to generate the messages that u_1, u_2 and u_3 communicate in the BCC round and sends these messages to Bob. Similarly, Bob simulate the rounds of $\mathcal{A}$ and communicates the messages that v_1, v_2 and v_3 communicate in the BCC round and then sends such messages to Alice. After the communication round of Π, Alice and Bob generate the information communicated by every vertex of the graph except $u_1, u_2, u_3, v_1, v_2, v_3$. Since these nodes have no information of x or y, this simulation can be done without sending any further messages between Alice and Bob. Finally, Alice simulates the output of every node except v_1, v_2, v_3 and accept if all accept. Bob simulates the output of every node except u_1, u_2, u_3 and accept if all accept. We deduce that Π is an ϵ-error protocol for `disjointness`. Nevertheless, the number of bits communicated in the execution of Π corresponds to the messages broadcasted by $u_1, u_2, u_3, v_1 v_2$ and v_3, which is $\mathcal{O}(\log n)$. This contradicts Lemma 1.1. We deduce that `disjointness-on-path` does not belong to **LB**.

Finally, let us suppose that $\mathcal{A}$ is a protocol consisting in two CONGEST rounds. In this case Π consists in just one round of communication. Observe that all messages communicated in the first round of $\mathcal{A}$ by nodes different than u_2, v_2 do not depend on (x, y) and can be simulated by the players without any communication. Then, protocol Π consists in Alice and Bob generating and interchanging the messages that u_1 and v_1 communicate in the second round of $\mathcal{A}$. Then Alice accept if every node $u_1, \dots, u_n$ accepts, and Bob accept if every node $v_1, \dots, v_n$ accept. By the correctness of $\mathcal{A}$, with probability $1 - \epsilon$, every node accepts if and only if (P, ℓ^*) is a yes-instance of `disjointness-on-path`. We deduce that Π is an ϵ-error protocol for `disjointness`. Nevertheless, the number of bits communicated in the execution of Π corresponds to the messages interchanged by v_1 and u_1, which are of size $O(\log n)$ in total. This contradicts Lemma 1.1. We deduce that `disjointness-on-path` does not belong to **CC**. □

Theorem 3.3 *There is a language that belongs to **CL** but does not belong to **LC**. This result holds even for randomized decision algorithms which may err with probability $\frac{1}{2} - \varepsilon$, for every $\varepsilon > 0$.*

Proof. Let us consider the graph $S_n = (V = (V_1, V_2), E)$ in which there exists $u^*v^* \in E$ such that : $u^* \in V_1$, $S_n^1 = S_n(V_1) \setminus \{v^*\}$ is a star graph with n leaves rooted in u^*, $v^* \in V_2$ and $S_n^2 = S_n(V_2) \setminus \{u^*\}$ is a star with n leaves rooted in v^*. Let us consider the following distributed language:

$$\begin{aligned}\texttt{DISJ-edge-star} = \{S_n = (V, E),\ & \ell : V \mapsto \{0,1\} \times [n], \\ & \ell(u) = (b, i) \text{ for some } u \in V_1 \iff x_i = b, \\ & \ell(v) = (b', i') \text{ for some } v \in V_2 \iff y_{i'} = b', \\ & \neg \bigvee_{i \in [n]} x_i \wedge y_i = 1, x, y \in \{0,1\}^n\}\end{aligned}$$

First, observe that $\texttt{DISJ} - \texttt{edge} - \texttt{star} \ \in \mathbf{CL}$. In fact, a protocol π in the hybrid $\mathsf{CONGEST} + \mathsf{LOCAL}$ model for $\texttt{DISJ} - \texttt{edge} - \texttt{star}$ can be described as: in the first round of communication all the nodes in the leaves of each star send its input. More precisely, each $v \in V_1$ and $u \in V_2$ send x_i and y_i respectively for some $i \in [n]$. Observe that after the first round of communication u^*, v^* are able to recover x and y from the messages sent by their neighbors. If the inputs of the leaves are not correct in the sense that each index i given in the input is different, they reject. Then, in the second round of communication, the node u^* sends a message containing x to v^* and v^* sends a message containing y to v^*. Finally the nodes in the leaves accept and u^*, v^* compute $\text{DISJ}(x, y) = \neg \bigvee_{i \in [n]} x_i \wedge y_i$ and accept if and only if $\text{DISJ}(x, y) = 1$

Now, we are going to show that $\texttt{DISJ} - \texttt{edge} - \texttt{star} \ \notin \mathbf{LC}$. By contradiction, let us assume that there exists a $\frac{1}{2} - \varepsilon$ protocol π in the hybrid $\mathsf{LOCAL}+ \mathsf{CONGEST}$ model for $\texttt{DISJ} - \texttt{edge} - \texttt{star}$. We consider an instance (x, y) of the set disjointness problem $\texttt{DISJ}$. Let $n = |x| = |y|$. We are going to describe a $\frac{1}{2} - \varepsilon$ protocol π' for $\texttt{DISJ}$.. Let us consider the instance of $\texttt{DISJ} - \texttt{edge} - \texttt{star}$ (S_n, ℓ) in which ℓ assigns x_i to each leaf in S_n^1 and y_i to each leaf in S_n^2. Observe that (x, y) is a yes instance of $\texttt{DISJ}$ if and only if (S_n, ℓ) is a yes instance for $\texttt{DISJ} - \texttt{edge} - \texttt{star}$. Let us define $S_n^A = S_n^1$ and $S_n^B = S_n^2$ i.e. we consider the graph induced by each of the star graphs in S_n. We say that Alice and Bob have S_n^A and S_n^B respectively. Since the roots u^* and v^* of S_n^1 and S_n^2 respectively have an empty input, Alice and Bob can simulate the LOCAL round of π. Then, Alice and Bob simulate the messages sent by the nodes during the $\mathsf{CONGEST}$ round of π. Observe that, since v^* is not in S_n^A and u^* is not in S_n^B, Alice cannot simulate he message m_{v^*,u^*} sent by v^* to u^* and Bob cannot simulate he message m_{u^*,v^*} sent by u^* to v^*. However, since Alice and Bob can simulate the local round, Alice can simulate m_{u^*,v^*} and Bob can simulate m_{v^*,u^*}. Thus, Alice sends a message m_A containing m_{u^*,v^*} to Bob and Bob sends a message m_B containing m_{v^*,u^*} to Alice. Finally, both players can simulate π and thus, they compute $\text{DISJ}(x, y)$. However, the cost of the protocol π' is $O(\log n)$ because the size of m_A and m_B is $O(\log n)$ which is a contradiction. □

Theorem 3.4 *There is a language that belongs to* ***LC*** *but does not belong to* ***CL****. This result holds even for randomized decision algorithms which may err with probability* $\frac{1}{2} - \varepsilon$*, for every* $\varepsilon > 0$.

Proof. Consider the language $\texttt{special-disjointness}$ defined by the pairs (G, ℓ) such that: (1) G is defined from a path $P = v_1, v_2, v_3, v_4$, a clique K_n, and two more vertices u_1, u_2. Node v_4 is adjacent to an arbitrary node of the clique, and v_1 has two pending vertices u_1 and u_2.

(2) $\ell : V(G) \to (\{0,1\}^*, [4])$ is a function such that:

- $\ell(w) = (\perp, \perp)$ for every $w \in V(K_n)$,
- $\ell(v_1) = (\perp, 1)$, $\ell(v_2) = (\perp, 2)$, $\ell(v_3) = (\perp, 3)$,
- $\ell(v_4) = (b, 4)$, with $b \in \{0,1\}$
- $\ell(u_1) = (x, \perp)$ and $\ell(u_2) = (y, \perp)$ with $x, y \in \{0,1\}^n$, and
- $\forall i \in [n]$, $x_i y_i = 0$ if and only if $b = 1$. In words, x, y are a yes-instance of DISJ if and only if the input of b is 1.

We first show that `special-disjointness` is in **LC**. The protocol has two verification algorithms, that are evaluated in parallel. We say that a node accepts if it accepts in both algorithms. The first algorithm, that we call *topology verification* consists in each node v sending its degree and the second coordinate of $\ell(v)$. Then,

- If v has degree $n-1$, then v accepts if and only if $\ell(v) = (\perp, \perp)$ and it has $n-2$ neighbors of degree $n-1$ and one neighbor of degree n.
- If v has degree n, then v accepts if and only if $\ell(v) = (\perp, \perp)$ it has $n-1$ neighbors of degree $n-1$ and one neighbor w of degree 2 such that $\ell(w)_2 = 4$.
- If v has degree 2 and $\ell(v)_2 = 4$, then v accepts if and only if one of the neighbors of v has degree n and the other neighbor has degree 2 and has 3 in its second component.
- If v has degree 2 and $\ell(v)_2 = \{2,3\}$, then v accepts if and only one neighbor has $\ell(v)_2 - 1$ and the other $\ell(v)_2 + 1$ in their second components.
- If v has degree 3, then v accepts if and only if $d(v)_2 = 1$, one neighbor has 2 in its second components, and the other two has $\perp$ in their second component.
- If v has degree 1, then v accepts if and only if $d(v)_2 = \perp$, and its neighbor has 1 in its second component.

Observe that all nodes accept in the topology verification algorithm if and only if G satisfies the properties of the language. Clearly the topology verification algorithm belongs to $\mathbf{CL} \cap \mathbf{LC}$, since it is only needed one LOCAL round to verify the previous properties. Then, it does not use bits in the CONGEST round.

In the following part let us assume that every node accepts in the topology verification algorithm. The second verification algorithm, called *input verification* is used to verify the conditions on ℓ, specially the last condition. In the input verification algorithm, every node with a degree different than 1, 2 or 3 immediately accepts. For the other nodes v, the algorithm is the following:

- If v has degree 1 (i.e. $v = u_1$ or $v = u_2$), then in the first round v communicates $\ell(v)_1$ to its neighbor, then accepts.
- If v has degree 3 (i.e. $v = v_1$), then in the first round v does not communicate anything. In the second round v receives x and y from two of neighbors. If x and y are not of the same length n, then v rejects. Otherwise, it verifies whether $x_i y_i = 0$ for every $i \in [n]$. If the answer is affirmative, it communicates a bit 1 to the other neighbor, and otherwise it communicates a bit 0.

- If v is such that $\ell(v)_2 = 4$ (i.e. $v = v_4$), then v communicates $b = \ell(v)_1$ to its neighbors in the first round and then accept.
- If v is such that $\ell(v)_2 = 3$ (i.e. $v = v_3$), then v sends nothing in the first round, and receives b from one neighbor. In the second round, he communicates b to the other neighbor and accept.
- If v is such that $\ell(v)_2 = 2$ (i.e. $v = v_2$), then v sends nothing in the first two rounds, but receives two bits in the second round from two different neighbors. Then v accept if the two bits are equal.

In simple words, the input verification algorithm consists in communicating x and y to v_1, then v_1 checks whether x and y are disjoint and communicates the answer to v_2. At the same time, the bit b is communicated to v_2 in two communication rounds. The first round can be done in $\mathbf{L}$ as we do not have bandwidth restrictions. The second can be done in $\mathbf{C}$ as at most one bit is communicated per edge. We deduce `special-disjointness` $\in \mathbf{LC}$.

We now show that `special-disjointness` $\notin \mathbf{CL}$. Let $\mathcal{A}$ be an $1/2 - \varepsilon$-error $\mathbf{CL}$ algorithm for `special-disjointness`. We show that $\mathcal{A}$ can be transformed into a two-player protocol Π for `disjointness`. Observe first that given a yes-instance of `special-disjointness`, we have that if we change b for $1-b$ on the input of node v_4, we obtain a no-instance. However, the nodes in a distance greater than 2 from v_4 can not see this difference. Therefore, as $\epsilon < 1/2$, we have that all vertices in a distance greater than 2 (in particular u_1, u_2 and v_1) from v_4 necessarily accept in $\mathcal{A}$ independently on the values of x and y. Following a similar reasoning, we obtain that all nodes at distance greater than 2 from u_1 and u_2 (in particular v_3, v_4 and all the nodes in the clique) must accept independently of the value of b. Therefore, the only node that rejects in the illegal inputs is v_2.

Now let (x, y) be an instance of DISJ. In protocol Π, Alice and Bob assume that they play the role of u_1 and u_2 in an instance of `special-disjointness` where the identifiers are chosen arbitrarily, and where $\ell(u_1)_1 = x$ and $\ell(u_2)_1 = y$ and $\ell(v_4)_2 = 1$. Alice simulates the CONGEST round of $\mathcal{A}$ for node u_1, generating a message m_A. Similarly, Bob simulates the CONGEST round of $\mathcal{A}$ for node u_2, generating the message m_B. Then Alice and Bob interchange m_A and m_B. Using that information, Alice and Bob can simulate $\mathcal{A}$ the message that node v_1 sends to v_2 in the first and second round. Notice that the message that v_1 sends to v_2 in the first round does not depend on x and y, while the message sent by v_1 in the second round only depends on m_A and m_B. On the other hand, Alice and Bob can simulate the two rounds of v_3 and v_4, as their messages do not depend on x and y. Then, Alice and Bob obtain the two messages received by v_2 from his neighbors v_1 and v_3. Using that information, the nodes can simulate the output of v_2 in $\mathcal{A}$ and accept if and only if v_2 accepts. From the correctness of $\mathcal{A}$ and the discussion of previous paragraph, we deduce that Π an ε-error protocol for DISJ. Nevertheless, in Π only $O(\log n)$ bits were communicated in total, which is a contradiction with Lemma 1.1. We deduce that `special-disjointness` $\notin \mathbf{CL}$. □

For the hybrid model combining BCC and LOCAL rounds, we first prove that $\mathbf{BL} \subseteq \mathbf{LB}$ showing a stronger result and then we show a separating problem to prove $\mathbf{LB} \neq \mathbf{BL}$.

Theorem 3.5 *Let $k \geq 1$ be an integer, and let $\alpha_1, \dots, \alpha_k$ and $\beta_1, \dots, \beta_k$ be non-negative integers. We have $\prod_{i=1}^{k} \boldsymbol{L}^{\alpha_i} \boldsymbol{B}^{\beta_i} \subseteq \boldsymbol{L}^{\sum_{i=1}^{k} \alpha_i} \boldsymbol{B}^{\sum_{i=1}^{k} \beta_i}$.*

Proof. Let $\mathcal{L} \in \prod_{i=1}^{k} \mathbf{L}^{\alpha_i} \mathbf{B}^{\beta_i}$, and let $\mathcal{A}$ be a distributed algorithm deciding $\mathcal{L}$ in the corresponding hybrid model combining LOCAL and BCC. Let us consider the maximum integer

$t < \sum_{i=1}^{k}(\alpha_i + \beta_i)$ such that $\mathcal{A}$ performs BCC at round t, and LOCAL at round $t+1$. (If no such t exist, then $\mathcal{A}$ is already in the desired form.) We transform $\mathcal{A}$ into $\mathcal{A}'$ performing the same as $\mathcal{A}$, excepted that rounds t and $t+1$ are switched.

Specifically, let us consider a run of $\mathcal{A}$ for an instance (G, ℓ). Let B_u be the message broadcasted by u at round t of $\mathcal{A}$, and, for every neighbor v of u, let $L_{u,v}$ be the message sent by u to v at round $t+1$ of $\mathcal{A}$. To define $\mathcal{A}'$, let S_u be the state of every node u at the beginning of round t of $\mathcal{A}$, and let $N_G(u)$ be the set of neighbors of u in G. In $\mathcal{A}'$, every node u sends its state S_u to all its neighbors at round t, using LOCAL. At round $t+1$ of $\mathcal{A}'$, every node u broadcasts B_u to all nodes, using BCC (this is doable, as u was able to produce B_u based on S_u at round t). Finally, before completing round $t+1$, every node u uses the collection $\{S_v : v \in N_G(u)\}$ and the collection $\{B_w : w \in V(G)\}$ to compute the messages $L_{v,u}$ for all $v \in N_G(u)$, by simulating what would have done every such neighbor v at round t of $\mathcal{A}$. Indeed, $L_{v,u}$ depend solely on S_v and $\{B_w : w \in V(G)\}$. It follows that, at the end of round $t+1$ of $\mathcal{A}'$, every node u can compute its state after $t+1$ rounds of $\mathcal{A}$. By repeating the same switch operation until no LOCAL rounds occur after a BCC round, we eventually obtain an algorithm deciding $\mathcal{L}$ and establishing that $\mathcal{L} \in \mathbf{L}^{\sum_{i=1}^{k}\alpha_i}\mathbf{B}^{\sum_{i=1}^{k}\beta_i}$. $\square$

Theorem 3.6 *There is a language that belongs to* $\boldsymbol{LB}$*, but does not belong to* $\boldsymbol{BL}^*$

Proof. Let $k \in \mathbb{N}$. We will construct a language that can be decided in $\mathbf{LB}$ but cannot be decided in $\mathbf{BL}^k$.

Lets consider the following language:

$$\begin{aligned} 0,2\text{-}\mathtt{colored\text{-}triangles} = \{(G,\ell) \mid \; & \ell : V \longrightarrow \{r,b,k\}, \\ & G \text{ has no } C_3 \text{ as a subgraph } \vee \\ & (G[\ell^{-1}(\{r\})] \text{ has at least one } C_3 \text{ as a subgraph } \wedge \\ & G[\ell^{-1}(\{b\})] \text{ has at least one } C_3 \text{ as a subgraph})\} \end{aligned}$$

This is, the set of graphs that does not contain any triangles, or contains at least one triangle of each color (red and blue), where a red (resp. blue) triangle is a triangle whose vertices are all asigned with the color red (resp. blue).

It is clear that 0,2-$\mathtt{colored\text{-}triangles}$ can be decided in $\mathbf{LB}$, since in the local round a vertex $v \in V$ knows if it belongs to a monochromatic C_3 or not. If it belongs to two or more triangles, it rejects immediately. If it belongs to only one triangle, in the BCC round sends the color of the triangle and the ID's of the other vertices in the triangle. If it does not belong to a triangle, then it sends nothing in the BCC round. Finally, every node knows how many triangles of each color are in G, so they accept if the property of the language is satisfied, and it rejects if it is not.

On the other hand, lets suppose for the purpose of contradiction, that there is a protocol Π that can decide 0,2-$\mathtt{colored\text{-}triangles}$ in $\mathbf{BL}$. Let $\mathcal{F}$ be the family of the pairs $(G, \ell) = ((V, E), \ell)$ with $V = V_r \sqcup V_k \sqcup V_b$ and $E = E_r \sqcup E_k \sqcup E_b$ such that there exists $v_{kr}, v_{kb} \in V_k$ that satisfy the following properties:

- $G_k = (V_k, E_k)$ is a $10k$-node-path such that v_{kr} and v_{kb} are the extremities.
- $e = uv \in E_r \Longrightarrow u, v \in V_r \cup \{v_{kr}\}$
- $e = uv \in E_b \Longrightarrow u, v \in V_b \cup \{v_{kb}\}$

- $e = uv \in E_k \implies u, v \in V_k$
- $v \in V_i \implies \ell(v) = i, \ \forall i \in \{r, b, k\}$

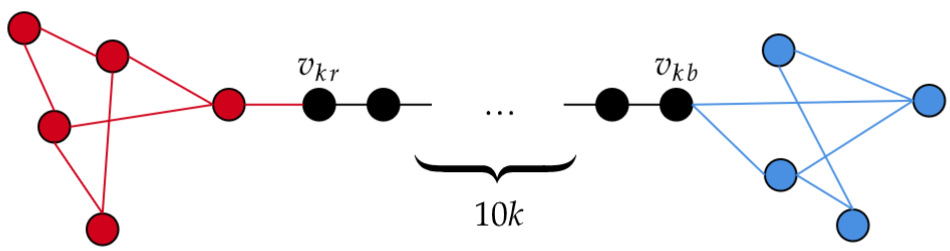

Figure 3.2: *Graph G in $\mathcal{F}$ with sets V_r and E_b in red, V_b and E_b in blue and the path (V_k, E_k) in black*

Let $(G, \ell) \in \mathcal{F}$. Notice that in order to decide if $(G, \ell) \in$ 0,2-colored-triangles, without loss of generality, a vertex $v \in V_r$ has to know if there is a C_3 in G_b or not in the BCC round, since in the LOCAL round, no information of G_b is sent to vertices in G_r. Then, as we already know, triangle-freeness cannot be solved in **B**, so there must exist two configurations $G_{b_{yes}}$ and $G_{b_{no}}$, (the first one with a C_3, and the second one without), that send the same exact messages in the BCC round.

Since in the k LOCAL rounds no new information is sent from G_r to G_b nor from G_b to G_r since the size of the path, v has to decide if it accepts or rejects based on the information sent by vertices in G_b in the BCC round.

So, if we consider G_1 with $G_r = G_{r_{yes}}$ and $G_b = G_{b_{yes}}$, and G_2 with $G_r = G_{r_{yes}}$ and $G_b = G_{b_{no}}$, we notice that G_1 is a yes-instance of 0,2-colored-triangles, whereas G_2 is a no-instance. Then, since v receives the same messages of G_b in both instances, v cannot decide in a deterministic way if the graph belongs or not to the language. Hence, 0,2-colored-triangles cannot be decided in $\mathbf{BL}^k$.

□

We now prove that $\mathbf{CB} \setminus \mathbf{BC} \neq \varnothing$ showing a stronger result.

Theorem 3.7 *There is a language that belongs to **CB**, but does not belong to $\mathbf{BL}^*$. This result holds even for a randomized algorithm with error $\varepsilon < 1/5$.*

Proof. Let us consider the distributed language denoted one-marked-edge defined as

$$\texttt{one-marked-edge} = \Big\{(G, \ell) : \big(\ell : V(G) \to \{0,1\}\big) \\ \wedge \Big(\Big|\Big\{\{u,v\} \in E(G) : \ell(u) = \ell(v) = 1\Big\}\Big| = 1\Big)\Big\}.$$

In words, the language corresponds to the graphs G with a potential mark on each node, satisfying that exactly one edge of G has its two endpoints marked. We have one-marked-edge $\in$ **CB**. Indeed, a simple algorithm consists, for each node, to learn which of its neighbors are marked, in one CONGEST round, and to broadcast its number of marked incident edges, in one BCC round. The nodes reject if the total sum of marked edges is different from 2 (i.e., exactly two nodes are incident to a unique marked edge). They accept otherwise.

We now prove that `one-marked-edge` $\notin \mathbf{BL}^*$. We show that this result holds even for a randomized algorithm which may err with probability $\epsilon < 1/5$. For the purpose of contradiction, let us assume that, for some $k \geq 0$, there exists an ϵ-error ($\epsilon < 1/5$) algorithm $\mathcal{A}$ solving `one-marked-edge` using one BCC round followed by k consecutive LOCAL rounds. We show how to use $\mathcal{A}$ for designing an ϵ-error 1-round protocol Π solving `XOR-index` by communicating only $\mathcal{O}(\sqrt{m})$ bits on m-bit instances, contradicting the fact that `XOR-index` has communication complexity $\Omega(m)$.

Let $(x, i) \in \{0,1\}^m \times [m]$ and $(y, j) \in \{0,1\}^m \times [m]$ be an instance of `XOR-index`. Without loss of generality, we assume that $m = \binom{n}{2}$ for some $n \in \mathbb{N}$. Let us consider a graph G on $2n + 4k$ nodes, composed of two disjoint copies of a clique of size n, plus a path P of $4k$ nodes. Let us denote by G^A and G^B the two cliques. The IDs assigned to the nodes of G^A are picked in $[n]$, while the IDs assigned to the nodes of G^B are picked in $[2n] \setminus [n]$. One extremity of P is connected to all nodes in G^A, and the other extremity of P is connected all nodes in G^B. Let us denote by P^A the $2k$ nodes of P closest to G^A, and by P^B the $2k$ nodes of P closest to G^B. These nodes are assigned IDs $2n+1, \ldots, 2n+4k$, consecutively, starting from the extremity of P connected to G^A.

We enumerate the $m = \binom{n}{2}$ edges in G^A and G^B from 1 to m. Then, in Π, the players interpret their input vectors x and y as indicators of the edges of G^A and G^B respectively. We denote by G_{xy} the subgraph of G such that, for every $r \in [m]$, the r-th edge e of G^A (resp., G^B) is in G_{xy} if and only if $x_r = 1$ (resp., $y_r = 1$). Also, all edges incident to nodes of P are in G_{xy}. Let $\{u_A^i, v_A^i\}$ be the endpoints of the i-th edge of G_A, and let $\{u_B^j, v_B^j\}$ represent the endpoints of the j-th edge of G_B. (These edges may or may not be in G_{xy} depending on the values of x_j and y_i.) We define $\ell_{ij} : V(G) \to \{0,1\}$ as the marking function such that $\ell_{ij}(w) = 1$ if and only if $w \in \{u_A^i, v_A^i, u_B^j, v_B^j\}$. By construction, we have that $(G_{xy}, \ell_{ij}) \in$ `one-marked-edge` if and only if $((x,i),(y,j))$ is a yes-instance of `XOR-index`, i.e., $x_j \neq y_i$. We say that Alice owns all nodes in $V(G^A) \cup V(P^A)$, and Bob owns all nodes in $V(G^B) \cup V(P^B)$. Observe that the edges of G_{xy} incident to nodes owned by Alice depend only on x, while the edges of G_{xy} incident to nodes owned by Bob only depend on y.

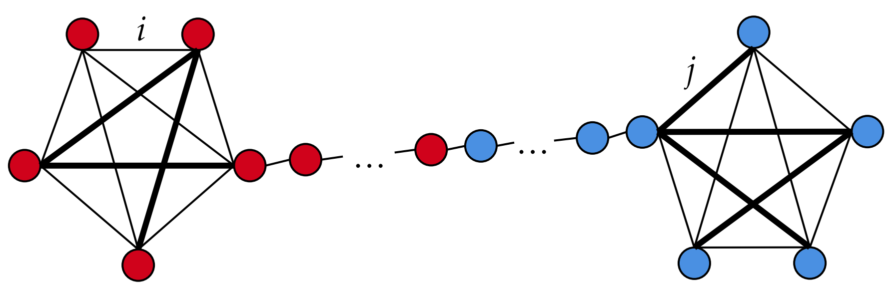

Figure 3.3: *Graph simulated by Alice and Bob where Alice owns the red nodes and Bob owns the blue nodes. $E(G_{xy})$ are represented as the bold edges. Notice that in this example $y_i = 0$ and $x_j = 1$.*

We are now ready to describe Π. First, Alice and Bob simulate the BCC round of algorithm $\mathcal{A}$ on all the nodes of G_{xy} owned by them, respectively, considering that *no vertices are marked*. This simulation results in each player constructing a set of $n + 2k$ messages, one for each node of the clique owned by the player, plus one message for each of the $2k$ nodes in the sub-path owned by the player. We denote by M_0^A and M_0^B the set of messages produced by Alice and Bob, respectively. Next, the players repeat the same procedure, but considering now that *all vertices are marked*, from which it results sets of messages denoted by M_1^A and M_1^B, respectively. Finally, Alice sends the pair (M_0^A, M_1^A) to Bob, as well as her input index i. Similarly, Bob sends the pair (M_0^B, M_1^B) to Alice, as well as his input index j. Observe that the size of these messages is $\mathcal{O}((n+k)\log n)$ bits.

After the communication, Alice and Bob decide their outputs as follows. First, each player extracts from M_1^A the messages produced by u_A^j and v_A^j, and extract from M_1^B the messages produced by u_B^i and v_B^i. Then, they extract from M_0^A and M_0^B the messages of every other node. Let us call M the resulting set of messages. Observe that M corresponds exactly to the set of messages communicated during the BCC round of $\mathcal{A}$ on input (G_{xy}, ℓ_{ij}). Then, Alice and Bob simulate the k LOCAL rounds of $\mathcal{A}$ on all the vertices they own. This is possible as the nodes of P are not marked, for every instance of `XOR-index`. Each player accepts if all the nodes owned by this player accept. Since $(G_{xy}, \ell_{ij}) \in$ `one-marked-edge` if and only if $((x,i),(y,j))$ is a yes-instance of `XOR-index`, we get that Π is an ϵ-error protocol solving `XOR-index` on inputs of size m by communicating only $O((n+k)\log n) = O(\sqrt{m})$ bits, which is a contradiction with Theorem 5.1. □

Theorem 3.8 *There is a language that belongs to* ***BC****, but it does not belong to* ***CB****. This result holds even for randomized algorithms, which may err with probability ϵ, for every $\epsilon <$ $1/5$.*

Proof. For every $n \geq 2$, let us consider the path P_{2n+1}, i.e., the path with $2n+1$ nodes, denoted consecutively $a_1, \ldots, a_n, c, b_n, \ldots, b_1$. Let $x \in \{0,1\}^n$, $y \in \{0,1\}^n$, $i \in [n]$, and $j \in [n]$. We define the labeling $\ell_{x,y,i,j}$ of the nodes of P_n as follows:

$$\ell_{x,y,i,j}(a_1) = i, \;\; \ell_{x,y,i,j}(a_n) = x, \;\; \ell_{x,y,i,j}(b_n) = y, \;\; \ell_{x,y,i,j}(b_1) = j,$$

and, for every $v \notin \{a_1, a_n, b_1, b_n\}$, $\ell_{x,y,i,j}(v) = \perp$. We define the distributed language

$$\texttt{XOR-index-path} = \{(P_{2n+1}, \ell_{x,y,i,j}) : (n \geq 2) \wedge (x, y \in \{0,1\}^n) \wedge (i, j \in [n]) \wedge (x_j \neq y_i)\}.$$

First, we show that `XOR-index-path` $\in$ **BC**. During the BCC round, every node broadcasts its ID, and the IDs of its neighbors (a node with more than two neighbors simply rejects). Also, degree-1 nodes broadcasts their labels. Note that the $2n+1$ nodes can then check whether they are vertices of the path P_{2n+1}, and, if this is not the case, they reject. Let i and j be the labels broadcasted by the two extremities of the path. Based on the information broadcasted by all the nodes, each of the two nodes a_n and b_n adjacent to the middle node c of the path knows which of the two labels i or j correspond to the index broadcasted by its farthest extremity in the path, b_1 and a_1, respectively. Thus, during the $\mathsf{CONGEST}$ round, a_n and b_n can send the bits x_j and y_i to the center c of the path, which checks whether $x_j \neq y_i$, and accepts or rejects accordingly.

Now, we show that `XOR-index-path` $\notin$ **CB**. Let us assume for the purpose of contradiction that there exists a 2-round algorithm $\mathcal{A}$ deciding `XOR-index-path` by performing one $\mathsf{CONGEST}$ round followed by one BCC rounds. To solve an instance $((x,i),(y,j))$ of `XOR-Index`, Alice and Bob simulate $\mathcal{A}$ on the path P_{2n+1} with consecutive IDs $1, \ldots, 2n+1$. Specifically, Alice simulates the $n+1$ nodes $a_1, \ldots, a_n, c$, while Bob simulates the $n+1$ nodes $b_1, \ldots, b_n, c$, with the nodes labeled with $\ell_{x,y,i,j}$. For simulating the $\mathsf{CONGEST}$ round, Alice sends to Bob the message m_{a_n} sent from a_n to c during that round, and Bob sends to Alice the message m_{b_n} sent from b_n to c during that round. The BCC round is actually simulated simultaneously. More precisely, Alice and Bob can both construct the messages broadcasted by all nodes $a_3, \dots, a_{n-2}$ and $b_3, \dots, b_{n-2}$, merely because they know their IDs and their labels (equal to $\perp$), and they can therefore infer the messages these nodes receive during the $\mathsf{CONGEST}$ round. So, these messages do not need to be communicated between the players. Moreover, Alice knows a priori what messages $m'_{a_1}, m'_{a_2}, m'_{a_{n-1}}$, and m'_{a_n} are to be broadcasted by a_1, a_2, a_{n-1} and a_n during the BCC round, and can send them to Bob. Symmetrically, Bob knows a priori what messages $m'_{b_1}, m'_{b_2}, m'_{b_{n-1}}$, and m'_{b_n} are to be broadcasted by b_1, b_2, b_{n-1} and b_n during the BCC round, and can send them to Alice. As for node c, thanks to the messages m_{a_n} and m_{b_n} sent by Alice to Bob, and by Bob to Alice, respectively, both players can construct the message to be sent by c during the BCC round. So, in total, for simulating $\mathcal{A}$, Alice (resp., Bob) just needs to send the messages $m_{a_n}, m'_{a_1}, m'_{a_2}, m'_{a_{n-1}}, m'_{a_n}$ to Bob (resp., the messages $m_{b_n}, m'_{b_1}, m'_{b_2}, m'_{b_{n-1}}, m'_{b_n}$ to Alice), which consumes $O(\log n)$ bits of communication in total. Each player accepts if all the nodes he or she simulates accept, and rejects otherwise. Alice and Bob are thus able to solve `XOR-index` by exchanging $O(\log n)$ bits only, which contradicts Theorem 5.1. $\square$

Chapter 4

Other Results

4.1. Three Rounds

We now proceed to show languages that separate the already known inclusions between BCC and LOCAL rounds of Theorem 3.5.

4.1.1. Two LOCAL rounds and one BCC round

Theorem 4.1 *There is a language that belongs to* ***LBL****, but does not belong to* ***BLL****.*

Proof. Since `0,2-colored-triangles` can be decided in **LB**, it is clear that it can also be decided in **LBL** using the protocol of **LB** and doing nothing in the last LOCAL round.

On the other hand, from Theorem 3.6 we showed that `0,2-colored-triangles` cannot be decided in **BLL**.

□

Theorem 4.2 *There is a language that belongs to* ***LLB****, but does not belong to* ***LBL****.*

Proof. Consider the following language

$$\begin{aligned}\texttt{0,2-colored-hexagons} = \{(G,\ell) \mid\ & \ell : V \longrightarrow \{r,b,k\},\\ & G \text{ has no } C_6 \text{ as a subgraph} \vee \\ & (G[\ell^{-1}(\{r\})] \text{ has at least one } C_6 \text{ as a subgraph} \wedge \\ & G[\ell^{-1}(\{b\})] \text{ has at least one } C_6 \text{ as a subgraph})\}\end{aligned}$$

We notice that `0,2-colored-hexagons` can be decided in **LLB**, since in the first two LOCAL rounds, every vertex knows if it belongs to a C_6 and the color of the vertices belonging to it. Then, in the BCC round, if a vertex belongs to more than one monochromatic C_6, it rejects immediately, since there will be more than one C_6 of one color. If a vertex belongs to one monochromatic C_6 sends its color. Therefore, after the BCC round, every vertex will know if there is at least one hexagon of color r and b, or no hexagons at all.

On the other hand, lets suppose by contradiction that `0,2-colored-hexagons` is decidable in **LBL**. We will use the family $\mathcal{F}$ of Theorem 3.6.

According to [29], the BCYCLE$_k$[r] problem which consist of detecting a $k-$cycle in the broadcast congested clique model, with the initial condition that every node $v \in V$ knows

$N_r(v)$ takes $\Omega(n^{1/\lfloor k/2 \rfloor})$ bits per node in one round if $r \leq k/3$, which in this case holds since $r = 2$ (after the first LOCAL round) and $k = 6$. Then, there must exist two configurations $G_1, G_2 \in \mathcal{F}$ such that $G_b(G_1) = G_{b_{yes}}$ and $G_b(G_2) = G_{b_{no}}$, (the first one with a C_6, and the second one without), whose vertices send the same messages in the BCC round.

Then, as in the final LOCAL round no information is sent from G_r to G_b nor from G_b to G_r, vertices in G_r have to decide if they accept or reject based on the information sent by vertices in G_b in the BCC round.

So, if we consider G_1, G_2 with $G_r(G_1) = G_r(G_2) = G_{r_{yes}}$, $G_b(G_1) = G_{b_{yes}}$ and $G_b(G_2) = G_{b_{no}}$, we notice the same messages will be sent in the BCC round for the two graphs, and $G_1 \in$ `0,2-colored-hexagons`, whereas $G_2 \notin$ `0,2-colored-hexagons`. Then, `0,2-colored-hexagons` cannot be decided in **LBL**.

□

4.1.2. Two BCC rounds and one LOCAL round

Theorem 4.3 *There is a language that belongs to **BLB**, but does not belong to **BBL**.*

Proof. Consider the language `0,2-colored-triangles`. Since `0,2-colored-triangles` can be decided in **LB**, it can clearly be decided in **BLB** by doing nothing in the first BCC round, and then copying the protocol to decide in **LB**.

On the other hand, from Theorem 3.6 we showed that 0,2-colored-triangles cannot be decided in **BBL**.

□

4.1.3. Three rounds with different number of BCC / LOCAL rounds

Theorem 4.4 *There is a language that belongs to **LLB**, but does not belong to **BBL**.*

Proof. Consider the language `0,2-triangle-freeness`. Since `0,2-colored-triangles` can be decided in **LB**, it can clearly be decided in **LLB** by doing nothing in the first LOCAL round, and then copying the protocol to decide in **LB**.

On the other hand, from Theorem 3.6 we showed that 0,2-colored-triangles cannot be decided in **BBL**.

□

Theorem 4.5 *There is a problem that belongs to **BBL**, but does not belong to **LLB**.*

For this theorem, we do not show a language that separates these two models. Instead, we show a separating problem where nodes in a graph G are given inputs and after they communicate with the respective models, they must output a specific string defined in the problem. So, we naturally extend the set **BBL** and **LLB** and say that a problem can belong to these sets.

Proof. Consider the problem P in which the task is that all nodes must output $N(v_0)$, where v_0 is the node with highest ID between the nodes with the highest degree.

Notice that $P \in$ **BBL**, since in the first BCC round every $v \in V$ sends $(\mathrm{ID}(v),\ \deg(v))$. Then, after the first round, v_0 will be known by every node so in the second BCC round, nodes belonging to $N(v_0)$ send their ID and the rest do nothing.

On the other hand, suppose $P \in$ **LLB**. Notice that it is equivalent to assume that $P \in$ **B** with the condition that every node v knows $N_3(v)$. Let P' be P with the condition that every node knows their 3-radius neighborhood.

Let Δ be the protocol that solves P' in one BCC round. Now, our goal is to use Δ as a sub-routine to reconstruct the whole graph G.

Let G' be the virtual graph G with $2n$ extra isolated nodes.

Let $v \in V$, v emulates being neighbor with the $2n$ extra nodes in G'. We denote G'_v to the graph G with the extra edges that v simulates. Then, v sends $m_v(G') \cup m_v(G'_v)$, where $m_v(G)$ denotes the message sent by v in the protocol Δ with the graph G. Hence, an arbitrary node $u \in V$ has the following information about v:

$$m_v(G') \cup m_v(G'_v) \cup \bigcup_{i \in \{n+1,\ldots,3n\}} m_i(G'_v)$$

where $i \in \{n+1, \ldots, 3n\}$ are the ID's of the extra nodes. The two first messages are sent by v in the BCC round, meanwhile the messages sent by the virtual nodes can be simulated by u since it knows their ID's and neighbors.

So, u can learn the neighbors of v decoding the following messages:

$$\bigcup_{r \in V \setminus \{v\}} m_r(G') \cup m_v(G'_v) \cup \bigcup_{i \in \{n+1,\ldots,3n\}} m_i(G'_v)$$

since v is the node with highest degree in G'_v.

As this works for all $u, v \in V$, every node can learn the neighborhood of all nodes in the graph, with which they can reconstruct the graph G. Since it is needed at least $\Omega(n^2)$ bits to reconstruct a graph and in one BCC round it is used in total $O(n \log n)$ bits, we get our contradiction. $\square$

4.2. Multiple Rounds Of One Model

In this chapter we prove two results. First, a separation language between $\mathbf{L}^k$ and $\mathbf{L}^{k+1}$, and then, a strong result that will let us conclude that there exist a separation language between $\mathbf{B}^k$ and $\mathbf{B}^{k+1}$.

Theorem 4.6 *Let k be a positive integer. There is a language that belongs to $\boldsymbol{L}^{k+1}$, but does not belong to $\boldsymbol{L}^k$*

Proof. Lets define the language k-$\mathsf{cycle\text{-}freeness} = \{(G, \ell) \mid C_k \text{ not a subgraph of } G\}$ for $k \in \mathbb{N}$.

Notice that in l local rounds, a node v can decide if it belongs or not to a C_{2l+1} at most. So, $2k+3$-$\mathsf{cycle\text{-}freeness}$ can be decided in $k+1$ LOCAL rounds, but cannot be decided in k LOCAL rounds.

$\square$

Now, we prove that there is a language that can be decided in k BCC rounds, but it cannot be decided in any finite combination of LOCAL, $\mathsf{CONGEST}$, and BCC rounds that contains $k-1$ BCC rounds.

Theorem 4.7 *Let $k > 1$. For every set $\boldsymbol{S} = \prod_{i=1}^{p} \boldsymbol{L}^{\alpha_i} \boldsymbol{B}^{\beta_i} \boldsymbol{C}^{\gamma_i}$ such that $\sum_{i=1}^{p} \beta_i = k-1$, it holds that $\boldsymbol{B}^k \setminus \mathbf{S} \neq \varnothing$. This result holds even for randomized decision algorithms which may err with probability $\frac{1}{2} - \varepsilon$, for every $\varepsilon > 0$.*

In order to prove this result, we consider the communication complexity known as `pointer chasing`. Let $S_A \cup S_B = [n]$ be a partition of $[n]$ such that $|S_A| = |S_B|$, and a function $f : S \to S$ such that $f(S_A) \subseteq S_B$ and $f(S_B) \subseteq S_A$. Call $f_A = f|_{S_A}$ and $f_B = f|_{S_B}$. In problem k-`pointer chasing` Alice receives as input S_A together with f_A and Bob receives as inputs S_B and f_B. The task is to compute the parity of the number of 1's in the binary representation of $f^k(0)$ (k compositions of f).

Let us denote $C^k(f)$ the communication complexity of function f restricted to k-round protocols. In [30] is shown the following result.

Proposition 4.1 *For every $k \geq 1$,*

- $C^k(k\text{--}\texttt{pointer chasing}) = O(\log n)$*, and*
- $C^{k-1}(k\text{--}\texttt{pointer chasing}) = O(n - k\log n)$*. This result holds even for randomized decision algorithms which may err with probability $\frac{1}{2} - \varepsilon$, for every $\varepsilon > 0$.*

Proof. Let $\ell \geq 1$. We define a language that belongs to $\mathbf{B}^k$ but does not belong to $\mathbf{L}^*\mathbf{B}^{k-1}$. Let us consider the language k-`pointer-chasing-on-long-path` (k-`PCLP`) as the set of paths of length n, where the two endpoints of the path have yes-instance k-`pointer-chasing`. Formally,

$$k\text{-}\mathsf{PCLP} = \Big\{(G,\ell) : \Big(\ell : V(G) \to \{0,1\}^* \cup \{\perp\}\Big) \;\wedge\; \Big(G \text{ is a path with endpoints } u, v\Big)$$
$$\wedge \Big(\ell(w) = \perp \;\wedge\; (\ell(u), \ell(v)) \in k - \texttt{pointer} - \texttt{chasing}\Big)\Big\}.$$

We have that k-`PCLP` belongs to $\mathbf{B}^k$. Indeed, a simple algorithm consists in all vertices communicating its degree in the first round, and, at the same time, the two endpoints of the path communicating the successive evaluations of f. Once $f^k(0)$ is computed, every node accepts if

- all except two nodes have degree 2, and the two remaining nodes have degree 1 (hence the graph is a path)
- the number of 1 bits in the binary representation of $f^k(0)$ is 1.

We now show that k-`PCLP` $\notin \mathbf{L}^*\mathbf{B}^{k-1}$. By contradiction, let us suppose that for some $t \geq 0$ there exists an $1/2 - \varepsilon$-error $\mathbf{L}^t\mathbf{B}^{k-1}$ algorithm $\mathcal{A}$ for k-`PCLP`. Given an instance (f_A, f_B) of k-`pointer chasing`, we define an instance (P_n, ℓ^*) of k-`PCLP` as follows. First, consider on an $2n$-node ($n > t$) path together endpoints u and v. Second, assign $\ell(w) = \perp$ to every node except the endpoints. Third, assign $\ell(u) = f_A$ and $\ell(v) = f_B$. We call P^u_d and P^v_d the set of nodes at distance at most d from u and v, respectively. Observe that $(P_n, \ell^*) \in k$-`PCLP` if and only if $(f_A, f_B) \in k$-`pointer-chasing`.

Now consider the following ε-error two-player $k-1$-round algorithm Π for ℓ-`pointer chasing`. Alice and Bob virtually construct the input (P_n, ℓ^*). We say that Alice owns the nodes in P^u_t and Bob owns the nodes P^v_t. All nodes that are not owned by Alice or Bob are called *remaining nodes*. The nodes simulate the t LOCAL rounds of $\mathcal{A}$ on the nodes they own and on the remaining nodes. Notice that the players can simulate these rounds without any communication, since the information of the endpoints are at distance $2n > 2t$. Then, the players perform $k-1$ rounds of communication. On the i-th round, Alice and Bob simulate the i-th BCC round of $\mathcal{A}$ on all the nodes they own, generating a packages of messages M_A^i and M_B^i, respectively. Then, they communicate M_A^i and M_B^i to each other. Finally, each player simulate the i-th BCC round of $\mathcal{A}$ on the remaining nodes generating a package of messages M_R^i. Observe that these latter messages can be generated as they depend only on the messages sent on the previous rounds, and not on the inputs of f_A and f_B. Finally, Alice and Bob have each the packages of messages $M^1, \dots, M^{k-1}$ corresponding to the $k-1$ BCC rounds of $\mathcal{A}$. Using that information the players can simulate the output of all nodes they own, as well as the output of the remaining nodes. The players then accept in Π if and only if all the nodes they own and the remaining nodes accept in $\mathcal{A}$. By the correctness of $\mathcal{A}$, we obtain that with probability $1/2 - \epsilon$, all the nodes in $\mathcal{A}$ accept if and only if Alice and Bob accept in Π. We deduce that Π is an $k-1$-round, ε-error protocol for k-`pointer-chasing`. However, in protocol Alice and Bob communicate $O(\ell t \log n) = O(\log n)$ bits, which contradicts Proposition 4.1. We deduce that k-PCLP $\notin \mathbf{L}^*\mathbf{B}^{k-1}$.

Finally, notice that from Theorem 3.5 and the fact that $\mathbf{C} \subseteq \mathbf{L}$, we have that all problems solvable in $\mathbf{S} = \prod_{i=1}^k \mathbf{L}^{\alpha_i}\mathbf{B}^{\beta_i}\mathbf{C}^{\gamma_i}$ can be solved in by an algorithm in $\mathbf{L}^*\mathbf{B}$. $\square$

Chapter 5

XOR-INDEX

Definition 5.1 (XOR-INDEX) *Consider the communication problem where Alice receives as input $x \in \{0,1\}^n$ and $i \in [n]$, whereas Bob receives $y \in \{0,1\}^n$ and $j \in [n]$. The task is that Alice outputs a boolean out_A and Bob outputs a boolean out_B such that $out_A \wedge out_B = x_i \oplus y_j$.*

We focus on 2−way 1−round protocols, that is, each player sends only one message to the other player, both players send their messages simultaneously (it cannot depend on the other player's message), and each player must decide his or her output upon reception of the message sent by the other player.

For every 2−player communication problem P, and for every $\epsilon > 0$, let us denote by $CC^1(P, \epsilon)$ the communication complexity of the best 2−way 1−round randomized protocol solving P with error probability at most ϵ.

Theorem 5.1 *For every non-negative $\epsilon < 1/5$, $CC^1(\texttt{XOR-index}, \epsilon) = \Omega(n)$ bits.*

Proof. Let $0 \leq \epsilon < 1/5$, and let Π randomized protocol solving XOR-index with error probability at most ϵ, where Alice communicates k_A bits to Bob, and Bob communicates k_B bits to Alice. Without loss of generality, we can assume that, in Π, Alice (resp., Bob) sends explicitly the value of i (resp., j) to Bob (resp., Alice). Indeed, this merely increases the communication complexity of Π by an additive factor $O(\log n)$, which has no consequence, as we shall show that $k_A + k_B = \Omega(n)$.

Let us consider the probabilistic distribution over the inputs of Alice and Bob, where x and y are drawn uniformly at random from $\{0,1\}^n$, and i and j are drawn uniformly at random from $[n]$. Let us denote X and I the random variables equal to the inputs of Alice, and Y and J the random variables equal to the inputs of Bob. Let M_A (resp., M_B) be the random variable equal to the message sent by Alice (resp., Bob) in Π on input (X, I) (resp., (Y, J)). Note that M_A and M_B have values in $\Omega_A = \{0,1\}^{k_A}$ and $\Omega_B = \{0,1\}^{k_B}$, respectively, of respective size 2^{k_A} and 2^{k_B}.

Let us fix $i, j \in [n]$, $m_A \in \Omega_A$, and $m_B \in \Omega_B$. Let $\mathcal{E}^A_{m_A,j}$ be the event corresponding to Bob receiving $J = j$ as input, and Alice sending $M_A = m_A$ to Bob in the communication round. Similarly, let $\mathcal{E}^B_{m_B,i}$ be the event corresponding to Alice receiving $I = i$ as input, and Bob sending $M_B = m_B$ to Alice in the communication round. For $a, b \in \{0,1\}$, we set:

$$p(a, m_A, j) = \Pr\Big[X_J = a \mid \mathcal{E}^A_{m_A,j}\Big], \text{ and } q(b, m_B, i) = \Pr\Big[Y_I = b \mid \mathcal{E}^B_{m_B,i}\Big],$$

and

$$p(a, j) = \Pr[X_J = a \mid J = j], \text{ and } q(b, i) = \Pr[Y_I = b \mid I = i].$$

Observe that $p(a,j) = q(b,i) = 1/2$. Let a^* and b^* be the most probable values of X_j given (m_A, j), and of Y_i given (m_B, i), respectively. Formally,

$$a^* = \text{argmax}_{a\in\{0,1\}} p(a, m_A, j), \quad \text{and} \quad b^* = \text{argmax}_{b\in\{0,1\}} q(b, m_B, i).$$

Observe that $p(a^*, m_A, j) \geq 1/2$ and $q(b^*, m_A, j) \geq 1/2$. We first establish the following technical lemma.

Lemma 5.1 *Let $\mathcal{F}$ the the event that* Π *fails. We have*

$$\Pr\Big[\mathcal{F} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\Big] \geq \Pr\Big[a^* \neq X_J \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\Big] \cdot \Pr\Big[b^* \neq Y_I \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\Big].$$

Proof. Without loss of generality, we assume that, in Π, after having communicated the pair (m_A, i), Alice computes b^*, and decides her output as follows. If $b^* \neq x_j$, then Alice accepts with some fixed probability p_A, and if $b^* = x_j$ then Alice accepts with some fixed probability q_A. The probabilities p_A and q_A determines the actions of Alice. Similarly, we can assume that, after having communicated (m_B, j), Bob computes a^*, and decides as follows. If $a^* \neq y_i$ then he accepts with some fixed probability p_B, and if $a^* = y_i$ then he accepts with some fixed probability q_B. Note that, in the case where the players do not take in account the value of a^* and b^*, then one can simply choose $p_A = q_A$ and $p_B = q_B$. Let us denote

$$R_A = \Pr\Big[a^* = X_J \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\Big], \quad \text{and} \quad R_B = \Pr\Big[b^* = Y_I \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\Big].$$

Observe that

$$\Pr\Big[\overline{\mathcal{F}} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\Big] = {}^1\!/\!_2 \Pr\Big[\overline{\mathcal{F}} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}, X_J \neq Y_I\Big] + {}^1\!/\!_2 \Pr\Big[\overline{\mathcal{F}} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}, X_J = Y_I\Big].$$

Now, conditioned on $X_J \neq Y_I$, the event $\overline{\mathcal{F}}$ corresponds to the event when Alice accepts and Bob accepts. Observe that, conditioned on $\mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}$, these two latter events are independent. Moreover, conditioned on $X_J \neq Y_I$, the event $a^* \neq Y_I$ is equal to the event $a^* = X_J$. Similarly, conditioned on $X_J \neq Y_I$, the event $b^* \neq X_J$ is equal to the event $b^* = X_I$. It follows that

$$\begin{cases} \Pr\Big[\text{Alice accepts} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}, X_J \neq Y_I\Big] = R_B\, p_A + (1 - R_B)\, q_A; \\ \Pr\Big[\text{Bob accepts} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}, X_J \neq Y_I\Big] = R_A\, p_B + (1 - R_A)\, q_B. \end{cases}$$

This implies that

$$\begin{aligned} &\Pr[\overline{\mathcal{F}} | \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}, X_J \neq Y_I] \\ &\quad = \Pr\Big[\text{Alice accepts and Bob accepts } \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}, X_J \neq Y_I\Big] \\ &\quad = \Pr\Big[\text{Alice accepts} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}, X_J \neq Y_I\Big] \cdot \Pr\Big[\text{Bob accepts} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}, X_J \neq Y_I\Big] \\ &\quad = \Big(R_B\, p_A + (1 - R_B)\, q_A\Big) \cdot \Big(R_A\, p_B + (1 - R_A)\, q_B\Big). \end{aligned}$$

let us now consider the case when conditioning on $X_J = Y_I$. In this case, the event $\overline{\mathcal{F}}$ corresponds to the complement of the event when Alice accepts and Bob accepts. Observe that, conditioned on $X_J = Y_I$, the event $a^* \neq Y_I$ is equal to the event $a^* \neq X_J$, and the event

$b^* \neq X_J$ is equal to the event $b^* \neq Y_I$. It follows that

$$\begin{cases} \Pr\Big[\text{Alice accepts} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}, X_J = Y_I\Big] = (1-R_B)\,p_A + R_B\,q_A; \\ \Pr\Big[\text{Bob accepts} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}, X_J = Y_I\Big] = (1-R_A)\,p_B + R_A\,q_B/ \end{cases}$$

This implies that

$$\begin{aligned} &\Pr[\overline{\mathcal{F}} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}, X_J = Y_I)] \\ &\quad = 1 - \Pr\Big[\text{Alice accepts and Bob accepts } \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}, X_J = Y_I\Big] \\ &\quad = 1 - \Pr\Big[\text{Alice accepts} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}, X_J = Y_I\Big] \cdot \Pr\Big[\text{Bob accepts} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}, X_J = Y_I\Big] \\ &\quad = 1 - \Big((1-R_B)\,p_A + R_B\,q_A\Big) \cdot \Big((1-R_A)\,p_B + R_A\,q_B\Big). \end{aligned}$$

Therefore, by combining the two cases, we get that

$$\begin{aligned} &\Pr[\overline{\mathcal{F}} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}] \\ &\quad = \frac{1}{2}\Big(R_A\,(p_A+q_A)\,(p_B-q_B) + R_B\,(p_A-q_A)\,(p_B+q_B) + 1 - p_A\,p_B + q_A\,q_B\Big). \end{aligned}$$

Conditioned to the events $\mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}$, the best protocol Π corresponds to the one that picks the values of p_A, q_A, p_B, q_B that maximize the previous quantity, restricted to the fact that p_A, q_A, p_B, q_B, R_A and R_B must be values in $[0,1]$, and that R_A and R_B must be at least $1/2$. The maximum can be found using the Karush-Kuhn-Tucker (KKT) conditions [31]. In fact, as the restrictions are affine linear functions, the optimal value is one solution of the following system of equations:

$$\begin{array}{rcl} (R_A+R_B-1)p_B - (R_A-R_B)q_B - 2\mu_1 + 2\mu_5 & = & 0 \\ (R_A+R_B-1)p_A + (R_A-R_B)q_A - 2\mu_2 + 2\mu_6 & = & 0 \\ (R_A-R_B)p_B - (R_A+R_B-1)q_B - 2\mu_3 + 2\mu_7 & = & 0 \\ -(R_A-R_B)p_A - (R_A+R_B-1)q_A - 2\mu_4 + 2\mu_8 & = & 0 \\ \mu_1(p_A-1) & = & 0 \\ \mu_2(p_B-1) & = & 0 \\ \mu_3(q_A-1) & = & 0 \\ \mu_4(q_B-1) & = & 0 \\ -\mu_5 p_A & = & 0 \\ -\mu_6 p_B & = & 0 \\ -\mu_7 q_A & = & 0 \\ -\mu_8 q_B & = & 0 \end{array}$$

The set of all solutions to this system is given in Table 5.1, together with the corresponding evaluation of $\Pr\Big[\overline{\mathcal{F}} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\Big]$. It follows from Table 5.1 that the value of $\Pr\Big[\overline{\mathcal{F}} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\Big]$ is upper bounded by $R_A + R_B - R_A R_B$. Indeed, assuming, without loss of generality, that $R_A \geq R_B$, we have:

$$
\begin{aligned}
\frac{1}{2}(1-(R_A+R_B)) \leq 1-R_A \leq 1-\frac{R_A+R_B}{2} &\leq 1-R_B \leq \frac{1}{2} \\
&\leq \min\{R_B, \frac{1}{2}(1+R_A-R_B)\} \leq \max\{R_B, \frac{1}{2}(1+R_A-R_B)\} \\
&\leq \frac{R_A+R_B}{2} \leq R_A \leq 1-(1-R_A)(1-R_B).
\end{aligned}
$$

Finally, observe that

$$
(1-R_A)(1-R_B) = \Pr\Big[a^* \neq X_J \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\Big] \cdot \Pr\Big[b^* \neq Y_I \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\Big],
$$

from which we get that

$$
\Pr\Big[\mathcal{F} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\Big] \geq \Pr\Big[a^* \neq X_J | \mathcal{E}^A_{m_A,j}\Big] \cdot \Pr\Big[b^* \neq Y_I \mathcal{E}^B_{m_B,i}\Big],
$$

as claimed.

□

We now show that, whenever the messages sent by Alice and Bob are too small, the distributions of a^* and of b^* is not far from the uniform. We make use of some basic definitions and tools on information complexity, and we refer to [27] for more details. Let (Ω, μ) be a discrete probability space. Given a random variable X we denote by $p_X : \Omega \mapsto \mathbb{R}$ the discrete density function of X, i.e., $p_X(\omega) = \Pr[X = \omega]$. We denote by $\mathbb{H} : \Omega \mapsto \mathbb{R}^+$ the entropy function, defined as $\mathbb{H}(X) = \sum_{\omega \in \Omega} p_X(\omega) \frac{1}{\log p_X(\omega)}$. Recall that, given two random variables X, Y on Ω, the entropy of X conditioned to Y is

$$
\mathbb{H}(X \mid Y) = \mathbb{E}_{p_Y(y)}(H(X \mid Y = y)).
$$

Moreover, let μ and ν be two probability measures on Ω. The total variation distance between μ and ν is defined as $|u - v|_{\text{TV}} = \sup_{E \subseteq \Omega} |\mu(E) - \nu(E)|$. It is known that $|u - v|_{\text{TV}} = \frac{1}{2} \sum_{\omega \in \Omega} |\mu(\omega) - \nu(\omega)|$. In addition, the Kullback-Liebler divergence between μ and ν is defined as $\mathbb{D}(\mu || \nu) = \sum_{\omega \in \Omega} \mu(\omega) \log \frac{\mu(\omega)}{\nu(\omega)}$. Given two random variables X and Y, their mutual information is defined as $\mathbb{I}(X; Y) = \mathbb{D}(p_{X,Y} || p_X p_Y)$. It is known that

$$
\mathbb{I}(X; Y) = \mathbb{H}(X) - \mathbb{H}(X \mid Y) = \mathbb{H}(Y) - \mathbb{H}(Y \mid X) = \mathbb{I}(Y; X).
$$

Finally, the mutual information of X, Y conditioned on a random variable Z is defined as the function $\mathbb{I}(X; Y \mid Z) = \mathbb{E}_{p_Z(z)}[\mathbb{I}(X; Y \mid Z = z)]$. Having all these notions at hand, we shall use some lemmas stated in the preliminaries section.

We observe that:

$$
\begin{cases}
\mathbb{I}(X_J; M_A \mid J) & = & \frac{1}{n} \sum_{j \in [n]} \mathbb{I}(X_j; M_A) & \leq & \frac{\mathbb{I}(X; M_A)}{n} & \leq & \frac{\mathbb{H}(M_A)}{n} & \leq & \frac{k_A}{n} \\
\mathbb{I}(Y_I; M_B \mid I) & = & \frac{1}{n} \sum_{i \in [n]} \mathbb{I}(Y_i; M_B) & \leq & \frac{\mathbb{I}(Y; M_B)}{n} & \leq & \frac{\mathbb{H}(M_B)}{n} & \leq & \frac{k_B}{n}.
\end{cases}
$$

By Pinsker's inequality, it follows that:

$$
\begin{cases}
\mathbb{E}_{(m_A, j)} \left(\| p(\cdot, m_A, j) - p(\cdot, j) \| \right) & \leq & \sqrt{\frac{k_A}{n}} \\
\mathbb{E}_{(m_B, i)} \left(\| q(\cdot, m_B, i) - q(\cdot, i) \| \right) & \leq & \sqrt{\frac{k_B}{n}}
\end{cases}
$$

These latter bounds imply that

$$\begin{cases} \mathbb{E}_{(m_A,j)}\left(p(a^*, m_A, j)\right) & \leq & \frac{1}{2} + \sqrt{\frac{k_A}{n}} \\ \mathbb{E}_{(m_B,i)}\left(q(b^*, m_B, i)\right) & \leq & \frac{1}{2} + \sqrt{\frac{k_B}{n}} \end{cases}$$

Now, from Lemma 5.1, we have that

$$\begin{aligned} \Pr\Big[\mathcal{F} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\Big] &\geq p(1-a^*, m_A, j)\cdot q(1-b^*, m_B, i) \\ &= \Big(1 - p(a^*, m_A, j)\Big)\cdot\Big(1 - q(b^*, m_B, i)\Big). \end{aligned}$$

As a consequence, we have

$$\begin{aligned} \Pr[\mathcal{F}] &= \mathbb{E}_{m_A,m_B,i,j}(\Pr\Big[\mathcal{F} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\Big]) \\ &= \sum_{m_A,m_B,i,j} \Pr\Big[\mathcal{F} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\Big]\cdot\Pr\Big[\mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\Big] \\ &\geq \sum_{m_A,m_B,i,j}\Big(1 - p(a^*_{(m_A,j)}, m_A, j)\Big)\Big(1 - q(b^*_{(m_B,i)}, m_B, i)\Big)\Pr\Big[\mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\Big] \\ &= \sum_{m_A,m_B,i,j}\Big(1 - p(a^*_{(m_A,j)}, m_A, j)\Big)\Big(1 - q(b^*_{(m_B,i)}, m_B, i)\Big)\Pr\Big[\mathcal{E}^A_{m_A,j}\Big]\cdot\Pr\Big[\mathcal{E}^B_{m_B,i}\Big] \\ &= \sum_{m_A,j}\Big(1 - p(a^*_{(m_A,j)}, m_A, j)\Big)\Pr\Big[\mathcal{E}^A_{m_A,j}\Big]\cdot\sum_{m_B,i}\Big(1 - q(b^*_{(m_B,i)}, m_B, i)\Big)\Pr\Big[\mathcal{E}^B_{m_B,i}\Big] \\ &= \Big(1 - \mathbb{E}_{(m_A,j)}\Big(p(a^*_{(m_a,j)}, m_A, j)\Big)\Big)\cdot\Big(1 - \mathbb{E}_{(m_B,i)}\Big(q(b^*_{(m_B,i)}, m_B, i)\Big)\Big) \\ &\geq \left(\frac{1}{2} - \sqrt{\frac{k_A}{n}}\right)\cdot\left(\frac{1}{2} - \sqrt{\frac{k_B}{n}}\right). \end{aligned}$$

Since $\Pr[\mathcal{F}] \leq \varepsilon$, we must have $(\frac{1}{2} - \sqrt{\frac{k_A}{n}})\cdot(\frac{1}{2} - \sqrt{\frac{k_B}{n}}) \leq \varepsilon \leq 1/5$, implying that $k_A = \Omega(n)$ or $k_B = \Omega(n)$.

$\square$

Table 5.1: Solutions of the equations given by the KKT conditions in the proof of Lemma 5.1, and the corresponding value of $\Pr\left[\mathcal{F} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\right]$.

p_A	p_B	q_A	q_B	$\Pr\left[\mathcal{F} \mid \mathcal{E}^A_{m_A,j}, \mathcal{E}^B_{m_B,i}\right]$
1	1	1	1	$1/2$
0	0	0	0	$1/2$
0	1	0	$\frac{R_A-R_B}{R_A+R_B-1}$	$1/2$
0	$\frac{R_A+R_B-1}{R_A-R_B}$	0	1	$1/2$
0	1	1	1	$1-R_B$
1	0	$\frac{R_B-R_A}{R_A+R_B-1}$	0	$1/2$
$\frac{R_A+R_B-1}{R_B-R_A}$	0	1	0	$1/2$
1	0	1	1	$1-R_A$
0	0	1	1	$1 - 1/2\,R_A - 1/2\,R_B$
0	1	0	$\frac{R_A+R_B-1}{R_A-R_B}$	$1/2$
0	$\frac{R_A-R_B}{R_A+R_B-1}$	0	1	$1/2$
1	1	0	1	R_B
0	1	0	1	$1/2$
1	0	0	1	$1/2 - 1/2\,R_A + 1/2\,R_B$
0	0	0	1	$1/2$
1	0	$\frac{R_A+R_B-1}{R_B-R_A}$	0	$1/2$
$\frac{R_B-R_A}{R_A+R_B-1}$	0	1	0	$1/2$
1	1	1	0	R_A
0	1	1	0	$1/2 + 1/2\,R_A - 1/2\,R_B$
1	0	1	0	$1/2$
0	0	1	0	$1/2$
1	1	0	0	$1/2\,R_A + 1/2\,R_B$
0	1	0	0	$1/2$
1	0	0	0	$1/2$

Chapter 6

Conclusions

In this thesis, first, we compared the three one round models, concluding that there is no inclusion between the languages that can be decided in one local round and one broadcast round, extending this result to a constant number of rounds.

Second, we compared every two round combination between the three models and we conclude that there is an inclusion between some of them and no inclusion between the rest. Moreover, we showed some languages that separate these combinations, showing that the inclusions proved before were strict.

Third, we successfully showed that every language resulting of the combination between broadcast and local rounds are comparable in terms of inclusion. Furthermore, we showed some languages that separate some of the three round combinations of broadcast and local rounds. Also, we showed a language that separate k rounds of broadcast rounds and $k+1$ broadcast rounds.

A possible continuation of this work is to complete all the comparisons between three round models combining the three models. Another one is to show a general language that can separate every combination of broadcast and local rounds, making the respective inclusions strict.

Bibliography

[1] Peleg, D., Distributed Computing: A Locality-Sensitive Approach. SIAM, 2000.

[2] Drucker, A., Kuhn, F., y Oshman, R., "On the power of the congested clique model," en Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing, pp. 367–376, 2014.

[3] A. Tell, W. Babalola, G. K. y Chinta, K., "Sd-wan: A modern hybrid-wan to enable digital transformation for businesses," 2018.

[4] Feuilloley, L. y Fraigniaud, P., "Survey of distributed decision," Bull. EATCS, vol. 119, 2016.

[5] Drucker, A., Kuhn, F., y Oshman, R., "On the power of the congested clique model," en ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014 (Halldórsson, M. M. y Dolev, S., eds.), pp. 367–376, ACM, 2014, doi: 10.1145/2611462.2611493.

[6] Izumi, T. y Le Gall, F., "Triangle finding and listing in congest networks," en Proceedings of the ACM Symposium on Principles of Distributed Computing, pp. 381–389, 2017.

[7] Becker, F., Kosowski, A., Matamala, M., Nisse, N., Rapaport, I., Suchan, K., y Todinca, I., "Allowing each node to communicate only once in a distributed system: shared whiteboard models," Distributed Comput., vol. 28, no. 3, pp. 189–200, 2015.

[8] Linial, N., "Locality in distributed graph algorithms," SIAM J. Comput., vol. 21, no. 1, pp. 193–201, 1992.

[9] Abboud, A., Censor-Hillel, K., Khoury, S., y Paz, A., "Smaller cuts, higher lower bounds," ACM Transactions on Algorithms (TALG), vol. 17, no. 4, pp. 1–40, 2021.

[10] Artur, C. y Konrad, C., "Detecting cliques in congest networks," Distributed Computing, vol. 33, no. 6, pp. 533–543, 2020.

[11] Elkin, M., "An unconditional lower bound on the time-approximation trade-off for the distributed minimum spanning tree problem," SIAM Journal on Computing, vol. 36, no. 2, pp. 433–456, 2006.

[12] Peleg, D. y Rubinovich, V., "A near-tight lower bound on the time complexity of distributed minimum-weight spanning tree construction," SIAM Journal on Computing, vol. 30, no. 5, pp. 1427–1442, 2000.

[13] Sarma, A. D., Holzer, S., Kor, L., Korman, A., Nanongkai, D., Pandurangan, G., Peleg, D., y Wattenhofer, R., "Distributed verification and hardness of distributed approximation," SIAM Journal on Computing, vol. 41, no. 5, pp. 1235–1265, 2012.

[14] Bar-Yehuda, R., Censor-Hillel, K., y Schwartzman, G., “A distributed $(2+\varepsilon)$-approximation for vertex cover in $O(\log\Delta/\varepsilon \log\log\Delta)$ rounds,” Journal of the ACM, vol. 64, no. 3, pp. 1–11, 2017.

[15] Censor-Hillel, K., Fischer, E., Schwartzman, G., y Vasudev, Y., “Fast distributed algorithms for testing graph properties,” Distributed Comput., vol. 32, no. 1, pp. 41–57, 2019.

[16] Even, G., Fischer, O., Fraigniaud, P., Gonen, T., Levi, R., Medina, M., Montealegre, P., Olivetti, D., Oshman, R., Rapaport, I., y Todinca, I., “Three notes on distributed property testing,” en 31st International Symposium on Distributed Computing (DISC), vol. 91 de LIPIcs, pp. 15:1–15:30, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[17] Fraigniaud, P. y Olivetti, D., “Distributed detection of cycles,” ACM Trans. Parallel Comput., vol. 6, no. 3, pp. 12:1–12:20, 2019.

[18] Fraigniaud, P., Rapaport, I., Salo, V., y Todinca, I., “Distributed testing of excluded subgraphs,” en 30th International Symposium on Distributed Computing (DISC), vol. 9888 de LNCS, pp. 342–356, Springer, 2016.

[19] Levi, R., Medina, M., y Ron, D., “Property testing of planarity in the CONGEST model,” Distributed Comput., vol. 34, no. 1, pp. 15–32, 2021.

[20] Lotker, Z., Pavlov, E., Patt-Shamir, B., y Peleg, D., “Mst construction in o (log log n) communication rounds,” en 15th ACM sSmposium on Parallel Algorithms and Architectures (SPAA), pp. 94–100, 2003.

[21] Chang, Y.-J., Fischer, M., Ghaffari, M., Uitto, J., y Zheng, Y., “The complexity of $(\Delta+1)$ coloring in congested clique, massively parallel computation, and centralized local computation,” en ACM Symposium on Principles of Distributed Computing (PODC), pp. 471–480, 2019.

[22] Jurdziński, T. y Nowicki, K., “Mst in $O(1)$ rounds of congested clique,” en 29th ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 2620–2632, 2018.

[23] Lenzen, C., “Optimal deterministic routing and sorting on the congested clique,” en ACM Symposium on Principles of Distributed Computing (PODC), pp. 42–50, 2013.

[24] Chen, L. y Grossman, O., “Broadcast congested clique: Planted cliques and pseudorandom generators,” en Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, pp. 248–255, 2019.

[25] Holzer, S. y Pinsker, N., “Approximation of distances and shortest paths in the broadcast congest clique,” en 19th International Conference On Principles Of Distributed Systems (OPODIS), 2016.

[26] Florent Becker, Pedro Montealegre, I. R. I. T., “The role of randomness in the broadcast congested clique model,” 2020.

[27] Rao, A. y Yehudayoff, A., Communication Complexity: and Applications. Cambridge University Press, 2020.

[28] Ruzsa, I. Z. y Szemerédi, E., “Triple systems with no six points carrying three triangles,” 1976.

[29] Florent Becker, Pedro Montealegre, I. R. y Todinca, I., “The impact of locality on the

detection of cycles in the broadcast congested clique model," 2018.

[30] Nisan, N. y Widgerson, A., "Rounds in communication complexity revisited," en Proceedings of the twenty-third annual ACM symposium on Theory of computing, pp. 419–429, 1991.

[31] Sundaram, R. K. *et al.*, A first course in optimization theory. Cambridge university press, 1996.