FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Lukáš Folwarczný

Graph communication protocols

Computer Science Institute of Charles University

Supervisor of the master thesis:	prof. RNDr. Pavel Pudlák, DrSc.
Study programme:	Computer Science
Study branch:	Theoretical Computer Science

Prague 2018

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In Prague, July 17, 2018 Lukáš Folwarczný

I wish to express my sincere gratitude to my supervisor Pavel Pudlák for his warmhearted guidance through the work on this thesis. Furthermore, I thank Ondřej Bouchala and Ivo Fukala for comments. Last but not least, I thank my family and friends for their precious support during my studies.

Title: Graph communication protocols

Author: Lukáš Folwarczný

Institute: Computer Science Institute of Charles University

Supervisor: prof. RNDr. Pavel Pudlák, DrSc., Institute of Mathematics of the Czech Academy of Sciences

Abstract: Graph communication protocols are a generalization of classical communication protocols to the case when the underlying graph is a directed acyclic graph. Motivated by potential applications in proof complexity, we study variants of graph communication protocols and relations between them.
The main result is a comparison of the strength of two types of protocols, protocols with equality and protocols with a conjunction of a constant number of inequalities. We prove that protocols of the first type are at least as strong as protocols of the second type in the following sense: For a Boolean function f, if there is a protocol with a conjunction of a constant number of inequalities of polynomial size solving f, then there is a protocol with equality of polynomial size solving f. We also introduce two new types of graph communication protocols, protocols with disjointness and protocols with non-disjointness, and prove that the first type is at least as strong as the previously considered protocols and that the second type is too strong to be useful for applications.

Keywords: communication complexity, Boolean functions, Boolean circuits

Contents

Introduction

Proving lower bounds is one of the principal aims of circuit complexity and proof complexity. Graph communication protocols come into play as a tool to translate the task of proving these lower bounds into the realm of communication complexity. This thesis studies variants of graph communication protocols and the relations between them.

Circuit complexity. Boolean circuits are a direct generalization of Boolean formulas where each intermediate result of the computation may be used repeatedly; a formal definition is given in Chapter 1. Circuit complexity is then the study of the computational power of circuits. With a special focus on lower bounds, this field is surveyed in the book by Jukna [Juk12].

Proof complexity. Propositional proof system, in the sense of Cook and Reckhow [CR79], is a sound and complete system for propositional tautologies with proofs verifiable in polynomial time. Examples of propositional proof systems include resolution, (extended) Frege systems, sequent calculus and cutting planes. Studying the strength of various propositional proof systems is the essence of proof complexity. A reference for this field is the book by Krajíček [Kra18].

It is worth noting that high-level ideas of complexity theory in general, including both circuit complexity and proof complexity, are explained in the book by Pudlák [Pud13].

Lower bounds. In circuit complexity, an (unconditional) lower bound for a certain class of circuits $\mathcal{C}$ and a function f is a theorem stating that the least size (or depth) of a circuit from $\mathcal{C}$ computing f is bounded from below by a certain function. In proof complexity, an (unconditional) lower bound is analogously a theorem stating that for a certain proof system P and a tautology ϕ the least length of a proof of ϕ in P is bounded from below by a certain function. A conditional lower bound is a theorem of the same kind, but proved under some assumption from complexity theory. Conditional lower bounds are not considered in this thesis.

Using a counting argument, Shannon [Sha49] proved that most functions require circuits of size at least $2^n/n$. Contrary to this fact, no explicit function has been proven to require superpolynomial circuits. Such lower bounds are known only for restricted classes of circuits. In particular, lower bounds for bounded depth circuits were proved independently by Ajtai [Ajt83] and Furst et al. [FSS84]. The lower bounds for monotone circuits, proved by Razborov [Raz85] and improved by Alon and Bopanna [AB87], are especially relevant for this thesis. An important recent result is the lower bound by Williams [Wil14] for ACC circuits; a general exposition of the relationship between SAT solving and lower bounds which lead to this result is given by Santhanam [San12].

In the context of proof complexity, the first strong results are an exponential lower bound for resolution by Haken [Hak85] (explained in the form of a game by Pudlák [Pud00]) and a superpolynomial lower bound for constant depth Frege systems by Ajtai [Ajt94] (an exponential lower bound was obtained by Krajíček [Kra94]). Nowadays, there are two or three general lower bound methods: the (random) restriction method (introduced in [FSS84]), feasible interpolation and also adversary argument is sometimes considered to be a general method.

Feasible interpolation. The feasible interpolation method was invented by Krajíček (idea formulated in [Kra94], applied in [Kra97]). In the basic setup, feasible interpolation reduces the task of proving a lower bound for a proof system P to proving a lower bound for Boolean circuits separating two NP sets. In the case of monotone feasible interpolation, lower bounds for monotone Boolean circuits are enough. However, lower bounds for different objects than monotone Boolean circuits may be used as well.

Motivation. Limits of monotone feasible interpolation by monotone Boolean circuits were already considered in the aforementioned paper by Krajíček [Kra97, Section 9]. However, general limits of monotone feasible interpolation are not known. The first result when monotone feasible interpolation was used with another computational model than Boolean circuits is due to Pudlák [Pud97]: He defined a generalization of monotone Boolean circuits called monotone real circuits (which were later proved by Rosenbloom [Ros97] to be strictly stronger than monotone Boolean circuits) and proved lower bounds for this model which lead, via monotone feasible interpolation, to lower bounds for the cutting planes proof system.

Studying graph communication protocols, considered by Krajíček [Kra18] to be "the primary objects" for feasible interpolation, is a way which could lead to superpolynomial lower bounds for proof systems for which no superpolynomial lower bounds are known. It could also lead to alternative proofs of known lower bounds; possibly for different tautologies.

An example of a proof system, belonging to the class of combined proof systems, for which no superpolynomial lower bound is known and for which monotone feasible interpolation via graph communication protocols could work is an extension of resolution called Res-lin or R(LIN) introduced by Itsykson and Sokolov [IS14] who also proved lower bounds for the tree-like version of this proof system. A different approach which could work for this proof system is randomized feasible interpolation due to Krajíček [Kra16].

Communication complexity. The field of communication complexity started in 1979 with the paper of Yao [Yao79]. The subject of this field is to study the complexity of problems when the input is distributed among several parties. In the basic two-party scenario, there are two parties with unlimited computational power, usually called Alice and Bob. Each of the parties receives its own part of the input; Alice $x \in \{0,1\}^n$ and Bob $y \in \{0,1\}^n$. The goal is to compute a given function $f(x,y)$ while exchanging the least number of bits between the parties. The computation is done using a specified protocol describing the acts of the parties. The applications of communication complexity include the analysis of data structures, streaming algorithms and of course proof complexity and circuit complexity. Details and a survey of the field is in the book by Kushilevitz and Nisan [KN97].

Karchmer-Wigderson game. Karchmer and Wigderson [KW88] proved the following theorem: For a Boolean function f, the minimum depth of a Boolean circuit computing f is equal to the communication complexity (that is the depth of a protocol) of a certain relation defined for f. There is also a monotone version of the relation for monotone circuits.

Graph communication protocols. In classical communication complexity, the measure of protocols is their depth and hence one can only consider protocols with the underlying graph being a tree. Graph communication protocols, that is protocols with the underlying graph being a directed acyclic graph, go back to Razborov [Raz95]. Razborov, inspired by Karchmer and Wigderson, used the size of certain protocols to characterize the size of circuits. See Pudlák [Pud10] or Sokolov [Sok17] for a survey of this topic. We give more details on the origin of graph communication protocols in Chapter 2.

Our contribution. We study several types of graph communication protocols. The main contribution is a comparison of the strength of two types of protocols, protocols with equality and protocols with a conjunction of a constant number of inequalities. We prove that protocols of the first type are at least as strong as protocols of the second type in the following sense: For a Boolean function f, if there is a protocol with a conjunction of a constant number of inequalities of polynomial size solving f, then there is a protocol with equality of polynomial size solving f. We also discuss the results, and we formulate and prove several simple claims illustrating the properties of the considered protocols.

Finally, we introduce two new types of graph communication protocols, protocols with disjointness and protocols with non-disjointness, and prove that the first type is at least as strong as the previously considered protocols and that the second type is too strong to be useful for applications.

Outline. Basic concepts and notations are introduced in Chapter 1. Definitions of protocols, their properties and statements of results are given in Chapter 2. The proofs of the two key lemmas are postponed to Chapter 3.

1. Preliminaries

Notations and basic concepts are introduced in this chapter.

Sets. We use the notation

$$\begin{aligned}[n] &:= \{1, 2, \ldots, n\};\\ [n]_0 &:= \{0, 1, \ldots, n\};\\ \omega &:= \{0, 1, \ldots\}.\end{aligned}$$

By $2^{<\omega}$, we denote the set of all finite subsets of ω.

Graphs. A *directed graph* is a pair (V, E) with $E \subseteq V \times V$. Elements of V are called *vertices*; elements of E are called *edges*. A *directed acyclic graph (DAG)* is a directed graph without a directed cycle, i.e. a sequence of edges in the form (v_1, v_2), $(v_2, v_3), \ldots, (v_{n-1}, v_n)$, (v_n, v_1). The *in-degree* (*out-degree*) of a vertex v is the number of $u \in V$ such that $(u, v) \in E$ ($(v, u) \in E$). Vertices with in-degree zero are called *sources*; vertices with out-degree zero are called *sinks*. A (directed) *tree* is a directed acyclic graph with one vertex of in-degree 0 and all other vertices of in-degree 1.

Boolean functions. The set $\{0, 1\}^n$ is the set of all n-bit sequences. For $x \in \{0, 1\}^n$, we denote the bits of x by $x_1, \ldots, x_n$. A (total) n-bit *Boolean function* is a function $f\colon \{0, 1\}^n \to \{0, 1\}$. A Boolean functions is *monotone* if $(\forall i \in [n])x_i \leq y_i$ implies $f(x) \leq f(y)$. A *partial* Boolean function is a function $f\colon S \to \{0, 1\}$ for some $S \subseteq \{0, 1\}^n$. A partial monotone Boolean function is a partial Boolean function which can be extended to some total monotone Boolean function. We use the notation $f^{-1}(b) = \{x \in \{0, 1\}^n \mid f(x) = b\}$ for $b \in \{0, 1\}$.

Boolean circuits. A *Boolean circuit* (in the de Morgan basis) is a directed acyclic graph $G = (V, E)$ with labeled vertices such that there are $2n + 2$ sources, labeled with $x_1, \ldots, x_n$, $\neg x_1, \ldots, \neg x_n$, 0, 1 and there is one sink; all non-source vetices have in-degree two and are labeled with $\land$ or $\lor$. The circuit computes an n-bit Boolean function in the obvious way: Sources are assigned the values of $x_1, \ldots, x_n$, $\neg x_1, \ldots, \neg x_n$, 0 and 1. Vertices labeled with $\land$ or $\lor$ are assigned the conjunction or disjunction of its predecessors. The output of the circuit is the value assigned to the sink. A Boolean circuit is *monotone* if there are only $n + 2$ sources labeled with $x_1, \ldots, x_n, 0$ and 1.

KW game. The task of the monotone Karchmer-Wigderson game for a partial monotone Boolean function f is, given $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$, to find an index i such that $x_i = 0 \land y_i = 1$. (Observe that there is always at least one such index i.)

2. Protocols

Graph communication protocols are introduced and defined in the beginning of this chapter. After the definitions we state the main theorems, discuss them and prove several related claims. Finally, we define two new types of protocols and prove two theorems concerning their strength.

2.1 General protocols

Karchmer and Wigderson [KW88] considered classical tree-like (i.e. the underlying graph is a tree) communication protocols and their depth to characterize the depth of circuits. Dag-like protocols (i.e. the underlying graph is a directed acyclic graph) considered in this thesis go back to Razborov [Raz95] who was considering the size of protocols to characterize the size of circuits. An explicit definition of these protocols was given by Krajíček [Kra97]. Our definition is a generalization of the definition by Hrubeš and Pudlák [HP18]. In comparison with [Kra97], only the monotone version is defined and the strategy function is omitted.

Definition 1. *Let $n \geq 1$ be a natural number. A* DAG communication protocol of degree d with the feasibility relation F *is a directed acyclic graph $G = (V, E)$ and a relation $F \subseteq \{0,1\}^n \times \{0,1\}^n \times V$, such that*

(i) G has one source v_0 (a node of in-degree zero) and the out-degree of every vertex is at most d,

(ii) for every sink ℓ (a node of out-degree zero), there exists an index i such that for every $x, y \in \{0,1\}^n$ it holds $(x, y, \ell) \in F$ iff $x_i = 0$ and $y_i = 1$.

Let f be a partial monotone Boolean function in n variables. We say that the protocol solves the monotone KW game for f *(or simply* solves f*), if for every $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$,*

(a) $(x, y, v_0) \in F$,

(b) for every $v \in V$ with $p \geq 1$ children $u_1, \ldots, u_p$, if $(x, y, v) \in F$ then there exists u_i with $(x, y, u_i) \in F$.

The size of a protocol is the number of vertices.

In the discussion, we write only protocols instead of DAG communication protocols. As in [HP18], we say that a vertex v is *feasible for x, y* if $(x, y, v) \in F$. By definition, the source is feasible for any $x \in f^{-1}(0)$, $y \in f^{-1}(1)$. The protocol solves the monotone KW game in the following sense: Given $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$ and any vertex v which is feasible for x, y (it is crucial that this holds for any feasible vertex, not just the source), we can find the solution to the KW game by traversing the graph via feasible vertices down to sinks which give us the solution.

Another general definition was given by Garg et al. [GGKS18, Section 2.1]. In their definition degree is fixed to 2 and general search problems are considered. If we restrict our definition to degree 2 and their definition to the monotone KW game, the definitions become equivalent.

2.2 Protocols with inequality and equality

The definition of what we call a DAG communication protocol with inequality was independently introduced by Hrubeš and Pudlák [HP18] and Sokolov [Sok17] (Sokolov considers only protocols of degree 2). Before that, Krajíček [Kra98] considered a different type of protocols with inequality.

Definition 2. *A* DAG communication protocol of degree d with inequality *is a DAG communication protocol of degree d with the feasibility relation F such that the relation F may be expressed as follows: For each vertex $v \in V$, there is a pair of functions $r_v^0, r_v^1 \colon \{0,1\}^n \to \mathbb{R}$ such that for all $x, y \in \{0,1\}^n$, it holds $(x, y, v) \in F$ iff $r_v^0(x) < r_v^1(y)$.*

Hrubeš and Pudlák [HP18] proved that the size of the minimum protocol of degree d with inequality solving f and the size of the minimum d-ary monotone real circuit computing f are equal (definition of monotone real circuits may be found in [Pud97] or [HP18]). That implies that the exponential lower bounds due to Pudlák [Pud97] for monotone real circuits hold also for protocols with inequality.

Replacing inequality with equality or a conjunction of inequalities, we obtain the following two definitions:

Definition 3. *A* DAG communication protocol of degree d with equality *is a DAG communication protocol of degree d with the feasibility relation F such that the relation F may be expressed as follows: For each vertex $v \in V$ there is a pair of functions $r_v^0, r_v^1 \colon \{0,1\}^n \to \mathbb{R}$ such that for all $x, y \in \{0,1\}^n$ it holds $(x, y, v) \in F$ iff $r_v^0(x) = r_v^1(y)$.*

Definition 4. *A* DAG communication protocol of degree d with a conjunction of c inequalities *is a DAG communication protocol of degree d with the feasibility relation F such that the relation F may be expressed as follows: For each vertex $v \in V$, there are c pairs of functions $r_v^{j,0}, r_v^{j,1} \colon \{0,1\}^n \to \mathbb{R}$ for $j \in [c]$. The functions satisfy for all $x, y \in \{0,1\}^n$ $(x, y, v) \in F$ iff $r_v^{1,0}(x) < r_v^{1,1}(y) \wedge \cdots \wedge r_v^{c,0}(x) < r_v^{c,1}(y)$.*

All three types of the protocols defined in this section are mentioned by Garg et al. [GGKS18] who name the protocols after the form of the feasible sets in the protocol. For a given v, the feasible set for v is the set of all pairs (x, y) such that $(x, y, v) \in F$. For protocols with inequality, the feasible sets are combinatorial triangles (definition may be found in [GGKS18]). For protocols with equality, the feasible sets are block-diagonal. And for protocols with a conjunction of c inequalities, the feasible sets are intersections of c combinatorial triangles.

2.3 Main theorems and discussion

We are ready to state our two main theorems:

Theorem 5. *Let P be a DAG communication protocol of degree d with inequality solving an n-bit partial monotone Boolean function f. If the size of P is s, then there exists a DAG communication protocol of degree 2 with equality solving f whose size is $\mathcal{O}(sn^{d+1})$.*

Theorem 6. *Let $c \geq 2$. Let P be a DAG communication protocol of degree d with a conjunction of c inequalities solving an n-bit partial monotone Boolean function f. If the size of P is s, then there exists a DAG communication protocol of degree 2 with equality solving f whose size is $\mathcal{O}(sc^d(c+d)(n+1)^{4c+2d-5})$.*

Both theorems are proved in two steps. The first step in the proof of Theorem 5 is a transformation of a protocol of degree d to a protocol of degree 2. This was already done by Hrubeš and Pudlák [HP18, Corollary 6 (ii)]:

Lemma 7. *Let P be a DAG communication protocol of degree d with inequality solving an n-bit partial monotone Boolean function f. If the size of P is s, then there exists a DAG communication protocol of degree 2 with inequality solving f whose size is $\mathcal{O}(sn^{d-2})$.*

In the first step of the proof of Theorem 6, we reduce the degree to 2 while increasing the number of inequalities to $2c + d - 3$:

Lemma 8. *Let P be a DAG communication protocol of degree d with a conjunction of c inequalities solving an n-bit partial monotone Boolean function f. If the size of P is s, then there exists a DAG communication protocol of degree 2 with a conjunction of $2c + d - 3$ inequalities solving f whose size is at most $s \cdot \max(c^d, d)$.*

It is then enough to prove the following lemma:

Lemma 9. *Let P be a DAG communication protocol of degree 2 with a conjunction of c inequalities solving an n-bit partial monotone Boolean function f. If the size of the protocol P is s, then there exists a DAG communication protocol of degree 2 with equality solving f whose size is $\mathcal{O}\left(sc(n+1)^{2c+1}\right)$.*

Lemma 7 and Lemma 9 for $c = 1$ together imply Theorem 5. Lemma 8 and Lemma 9 together imply Theorem 6.

We prove Lemma 8 and Lemma 9 in Chapter 3.

The rest of this section is devoted to a discussion of the two main theorems.

Theorem 5 and Theorem 6 are primarily intended for small values of the parameter d (e.g. constant). As mentioned above, there are exponential lower bounds for protocols of degree 2 with inequality and we expect that such lower bounds also exist for protocols with equality. The next claim shows that protocols with inequality of degree n can solve any function while having a small size and therefore we do not expect that the exponents in the theorems could be constant for large values of d.

Claim 10. *For every n-bit partial monotone Boolean function f, there is a DAG communication protocol of degree n with inequality solving f whose size is $n + 1$.*

Proof. We construct a protocol with a source v_0 which has n children $\ell_1, \ldots, \ell_n$. The functions of the source are $r^0_{v_0} \equiv 0$, $r^1_{v_0} \equiv 1$. The sink ℓ_i has functions $r^0_{\ell_i}(z) = r^1_{\ell_i}(z) = z_i$. The protocol is depicted in Fig. 2.1 and clearly solves f. □

We say that a type of protocols A is at least as strong as another type of protocols B if the following holds: Let f be an n-bit partial monotone Boolean

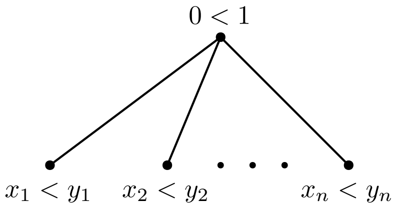

Figure 2.1: Proof of Claim 10

function. If there is a protocol of type B solving f whose size is polynomial in n, then there is a protocol of type A whose size is also polynomial in n.

Theorem 5 states that protocols of degree 2 with equality are at least as strong as protocols of a constant degree with inequality.

We consider the main contribution of Theorem 6 to be in the case $d = 2$ and $c = 2$. In this case, the theorem states that protocols of degree 2 with equality are at least as strong as protocols of degree 2 with a conjunction of two equalities. The following claim shows that there is a simple converse statement. Therefore the two types of protocols are equivalent in the sense of their strength.

Claim 11. *Let P be a DAG communication protocol of degree d with equality solving an n-bit partial monotone Boolean function f. If the size of P is s, then there exists a DAG communication protocol of degree d with a conjunction of two inequalities solving f whose size is s.*

Proof. We show how to express equality of two real values in the protocol by a conjunction of two inequalities. Let $G = (V, E)$ be the underlying graph of P. The constructed protocol P' with a conjunction of two inequalities has the same underlying graph. Consider a vertex $v \in V$ with functions $r_v^0, r_v^1 \colon \{0,1\}^n \to \mathbb{R}$ in P. Define

$$\varepsilon := \frac{1}{2} \min \left\{ |r_v^0(x) - r_v^1(y)| \,\middle|\, x, y \in \{0,1\}^n \wedge r_v^0(x) \neq r_v^1(y) \right\}.$$

(We set ε to an arbitrary positive value if the set is empty.)

We claim for all $x, y \in \{0,1\}^n$

$$r_v^0(x) \leq r_v^1(y) \Leftrightarrow \quad r_v^0(x) - \varepsilon < r_v^1(y), \tag{2.1}$$
$$r_v^0(x) \geq r_v^1(y) \Leftrightarrow -r_v^0(x) - \varepsilon < -r_v^1(y). \tag{2.2}$$

The implications from left to right of (2.1) and (2.2) are true because $\varepsilon > 0$. To prove the opposite implication in (2.1) we reorder the terms on the right as $r_v^0(x) - r_v^1(y) < \varepsilon$. It follows that $r_v^0(x) \leq r_v^1(y)$ from the definition of ε. Similarly for (2.2) we reorder the terms as $r_v^1(y) - r_v^0(x) < \varepsilon$ and obtain $r_v^0(x) \geq r_v^1(y)$.

Combining the two equivalences (2.1) and (2.2) we obtain

$$r_v^0(x) = r_v^1(y) \Leftrightarrow r_v^0(x) - \varepsilon < r_v^1(y) \wedge -r_v^0(x) - \varepsilon < -r_v^1(y). \tag{2.3}$$

It is therefore enough to set $q_v^0(x) := r_v^0(x) - \varepsilon$, $q_v^1(y) := r_v^1(y)$, $q_v^2(x) := -r_v^0(x) - \varepsilon$ and $q_v^3(y) := -r_v^1(y)$. We assign these functions to v as the functions $r_v^0, \dots, r_v^3$ in Definition 4. The protocol P' solves f because of (2.3). □

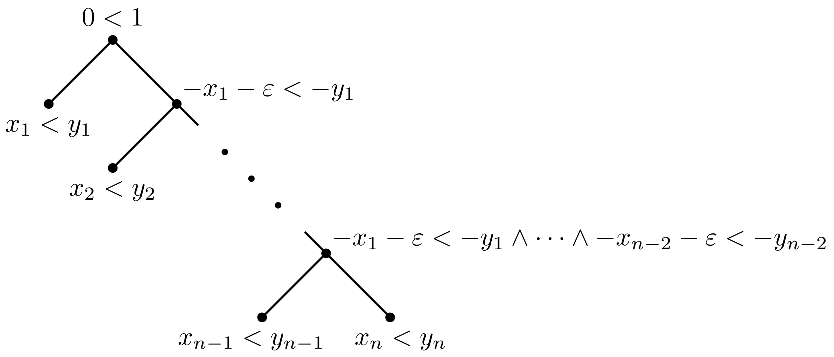
$0 < 1$
$-x_1 - \varepsilon < -y_1$
$x_1 < y_1$
$x_2 < y_2$
$-x_1 - \varepsilon < -y_1 \wedge \cdots \wedge -x_{n-2} - \varepsilon < -y_{n-2}$
$x_{n-1} < y_{n-1}$
$x_n < y_n$

Figure 2.2: Proof of Claim 12

Similarly to Claim 10 proving that protocols of degree n are already too strong, we can prove that protocols with a conjunction of $n-2$ inequalities are also too strong.

Claim 12. *For every n-bit partial monotone Boolean function f, there is a DAG communication protocol of degree 2 with a conjunction of $n-2$ inequalities solving f whose size is $2n-1$.*

Proof. We construct a protocol with non-sink vertices $v_0, \dots, v_{n-2}$ and sinks $\ell_1, \dots, \ell_n$. The sink ℓ_i has functions $r_{\ell_i}^{j,0}(z) = r_{\ell_i}^{j,1}(z) = z_i$ for every $j \in [c]$. The vertex v_i for $i \in [n-3]_0$ has children v_{i+1} and ℓ_{i+1}; the vertex v_{n-2} has children ℓ_{n-1} and ℓ_n.

This time, we define $\varepsilon := 1/2$ and use the fact, similar to the one in the proof of Claim 11, that for two bits a, b

$$a \geq b \Leftrightarrow -a - \varepsilon < -b.$$

For a vertex v_i, $i \in [n-2]_0$, we set the functions

$$\begin{aligned} r_{v_i}^{2j-2}(z) &= -z_j - \varepsilon && \text{for } j \in \{1, \dots, i\}; \\ r_{v_i}^{2j-1}(z) &= -z_j && \text{for } j \in \{1, \dots, i\}; \\ r^{2j-2} &\equiv 0 && \text{for } j \in \{i+1, \dots, c\}; \\ r^{2j-1} &\equiv 1 && \text{for } j \in \{i+1, \dots, c\}. \end{aligned}$$

See Fig. 2.2 for an illustration. We have constructed a protocol in the sense of Definition 4. It remains to verify that it solves an arbitrary monotone Boolean function f. The condition (a) of Definition 1 is satisfied. To prove the condition (b), consider the vertex v_i for $i \in [n-3]_0$. Assuming the vertex is feasible for $x \in f^{-1}(0), y \in f^{-1}(1)$, it holds $x_j \geq y_j$ for $j \in [i]$. There are two possibilities: Either $x_{i+1} < y_{i+1}$ and the son ℓ_{i+1} is feasible or $x_{i+1} \geq y_{i+1}$ and then the son v_{i+1} is feasible. Finally, consider the vertex v_{n-2} and assume that it is feasible for x, y. Then it holds $x_j \geq y_j$ for $j \in [n-2]$. Because of $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$, there is i such that $x_i < y_i$. It must hold that $i = n-1$ or $i = n$ and therefore at least one of the sons ℓ_{n-1} and ℓ_n is feasible. □

It is natural to ask whether a reduction of the degree as in Lemma 7 is also possible for protocols with equality or a conjunction of inequalities. We are not able to adapt the proof od Hrubeš and Pudlák [HP18] because their proof is based on the fact that protocols with inequality correspond to monotone real circuits. We do not have a computational model corresponding to the other types of protocols. However, a similar reduction of the degree for protocols with equality is a consequence of Theorem 6.

Corollary 13. *Let P be a DAG communication protocol of degree d with equality solving an n-bit partial monotone Boolean function f. If the size of P is s, then there exists a DAG communication protocol of degree 2 with equality solving f whose size is $\mathcal{O}(sd2^d(n+1)^{2d+3})$.*

Proof. We use Claim 11 to convert P into a protocol P' of degree d with a conjunction of two inequalities whose size is s. Using Theorem 6 for P', we obtain a protocol of degree 2 with equality solving f whose size is $\mathcal{O}(sd2^d(n+1)^{2d+3})$. □

2.4 Protocols with (non-)disjointness

We introduce two new types of protocols which have not been studied before.

Definition 14. *A* DAG communication protocol of degree d with disjointness *is a DAG communication protocol of degree d with the feasibility relation F such that the relation F may be expressed as follows: For each vertex $v \in V$, there is a pair of functions $S_v^0, S_v^1 \colon \{0,1\}^n \to 2^{<\omega}$ such that for all $x, y \in \{0,1\}^n$ it holds $(x, y, v) \in F$ iff $S_v^0(x) \cap S_v^1(y) = \emptyset$.*

Definition 15. *A* DAG communication protocol of degree d with non-disjointness *is a DAG communication protocol of degree d with the feasibility relation F such that the relation F may be expressed as follows: For each vertex $v \in V$, there is a pair of functions $S_v^0, S_v^1 \colon \{0,1\}^n \to 2^{<\omega}$ such that for all $x, y \in \{0,1\}^n$ it holds $(x, y, v) \in F$ iff $S_v^0(x) \cap S_v^1(y) \neq \emptyset$.*

We prove that protocols with disjointness are at least as strong as protocols with equality and that protocols with non-disjointness of small size solve any monotone Boolean function.

Theorem 16. *Let P be a DAG communication protocol of degree d with equality solving an n-bit partial monotone Boolean function f. If the size of P is s, then there exists a DAG communication protocol of degree d with disjointness solving f whose size is s.*

Proof. Let P be a protocol with the underlying graph $G = (V, E)$ and functions r_v^0, r_v^1 as in Definition 3. The constructed protocol P' has the same underlying graph. We show for each vertex v how to express inequality by disjointness. Let

$$R_v := \left\{ r_v^0(z) \,\middle|\, z \in \{0,1\}^n \right\} \cup \left\{ r_v^1(z) \,\middle|\, z \in \{0,1\}^n \right\}.$$

It naturally holds $|R_v| \leq 2^{n+1}$. We can therefore encode the values by $n+1$ bits. Formally, we fix an injective function $g_v \colon R_v \to \{0,1\}^{n+1}$. Because g_v is injective, for all x, y, we have

$$g_v(r_v^0(x)) = g_v(r_v^1(y)) \Leftrightarrow r_v^0(x) = r_v^1(y).$$

We fix x, y and set $a := g_v(r_v^0(x))$ and $b := g_v(r_v^1(y))$. For every vertex $v \in V$, we define

$$\begin{aligned}S_v^0(x) &:= \{i \mid i \in [n+1] \wedge a_i = 1\} \cup \{i+n+1 \mid i \in [n+1] \wedge a_i = 0\}\,,\\ S_v^1(y) &:= \{i \mid i \in [n+1] \wedge b_i = 0\} \cup \{i+n+1 \mid i \in [n+1] \wedge b_i = 1\}\,.\end{aligned}$$

We express disjointness as follows:

$$\begin{aligned}&S_v^0(x) \cap S_v^1(y) = \emptyset\\ \Leftrightarrow &(\forall i \in [n+1])(a_i = 0 \vee b_i = 1) \wedge (\forall i \in [n+1])(a_i = 1 \vee b_i = 0)\\ \Leftrightarrow &(\forall i \in [n+1]) a_i = b_i \Leftrightarrow r_v^0(x) = r_v^1(y)\end{aligned}$$

We concluded that $S_v^0(x) \cap S_v^1(y) = \emptyset \Leftrightarrow r_v^0(x) = r_v^1(y)$ which implies that P' solves f. □

Theorem 17. *For every n-bit partial monotone Boolean function f, there is a protocol of degree 2 with non-disjointness solving f whose size is $2n-1$.*

Proof. We use binary search to solve the KW game. The protocol will consist of vertices of the form $v(l, r)$ for some $l, r \in [n]$. For $v(l, r)$, we define the functions

$$\begin{aligned}S_{v(l,r)}^0(z) &:= \{i \in \{l, \ldots, r\} \mid z_i = 0\}\,,\\ S_{v(l,r)}^1(z) &:= \{i \in \{l, \ldots, r\} \mid z_i = 1\}\,.\end{aligned}$$

For every $x, y \in \{0,1\}^n$, it holds

$$S_{v(l,r)}^0(x) \cap S_{v(l,r)}^1(y) \neq \emptyset \Leftrightarrow (\exists i \in \{l, \ldots, r\})(x_i = 0 \wedge y_i = 1). \tag{2.4}$$

We define the underlying graph recursively. We add to the graph the vertex $v(1, n)$ as the root. For every vertex $v(l, r)$, in the graph we do the following: If $l = r$, then $v(l, r)$ is a sink. Otherwise, we add two sons

$$v\left(l, \left\lfloor \frac{l+r}{2} \right\rfloor\right) \quad \text{and} \quad v\left(\left\lfloor \frac{l+r}{2} \right\rfloor + 1, r\right).$$

See Fig. 2.3 depicting the case $n = 5$.

It is a protocol in the sense of Definition 1. It remains to show that it solves an arbitrary monotone Boolean function f. The source is feasible for $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$ because of (2.4) and monotonicity. Consider a vertex $v(l, r)$ with $l < r$. For its children $v(l, \left\lfloor \frac{l+r}{2} \right\rfloor)$ and $v(\left\lfloor \frac{l+r}{2} \right\rfloor + 1, r)$, it holds

$$\{l, \ldots, r\} = \left\{l, \ldots, \left\lfloor \frac{l+r}{2} \right\rfloor\right\} \dot\cup \left\{\left\lfloor \frac{l+r}{2} \right\rfloor + 1, \ldots, r\right\}.$$

Together with (2.4), it implies that if $v(l, r)$ is feasible for x, y, then at least one of the sons is feasible for x, y.

The size of the protocol is $2n-1$ because the underlying graph is a tree with n sinks where every non-sink vertex has out-degree 2. □

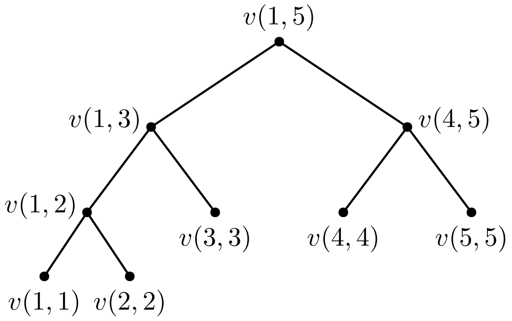

Figure 2.3: Proof of Theorem 17

3. Simulations

Lemma 8 and Lemma 9 are proven in this chapter.

3.1 Degree reduction

Lemma 8 as stated in Section 2.3:

Lemma 8. *Let P be a DAG communication protocol of degree d with a conjunction of c inequalities solving an n-bit partial monotone Boolean function f. If the size of P is s, then there exists a DAG communication protocol of degree 2 with a conjunction of $2c + d - 3$ inequalities solving f whose size is at most $s \cdot \max(c^d, d)$.*

Proof. Let $G = (V, E)$ be the underlying graph of P with functions $r_v^{j,0}, r_v^{j,1}$ as in Definition 4. Similarly to the proof of Claim 11, we define

$$\varepsilon := \frac{1}{2}\min\left\{\left|r_v^{j,0}(x) - r_v^{j,1}(y)\right| \,\middle|\, x, y \in \{0,1\}^n \land v \in V \land j \in [c] \land r_v^{j,0}(x) \neq r_v^{j,1}(y)\right\}.$$

(We again set ε to an arbitrary positive value if the set is empty.)

It then holds for every $x, y \in \{0,1\}^n$ and $v \in V$, $j \in [c]$

$$r_v^{j,0}(x) \geq r_v^{j,1}(y) \Leftrightarrow -r_v^{j,0}(x) - \varepsilon < -r_v^{j,1}(y).$$

We denote the functions in the constructed protocol by $q_v^{j,0}$ and $q_v^{j,1}$. For every vertex $v \in V$, we put v into the constructed protocol with functions $q_v^{j,0} = r_v^{j,0}$, $q_v^{j,1} = r_v^{j,1}$ for $j \in [c]$ and $q_v^{j,0} \equiv 0$, $q_v^{j,1} \equiv 1$ for $j > c$. For every non-sink vertex $v \in V$ with children $u_1, \dots, u_p$, we add a substructure connecting v with $u_1, \dots, u_p$. The idea of the substructure is to test sequentially all children and for each child test sequentially the inequalities: If the tested inequality is true, we test the next inequality. If the tested inequality is false, we test the next child.

Fix a non-sink vertex $v \in V$ with children $u_1, \dots, u_p$. If $p \leq 2$, we just connect v with its children. For the rest of the discussion, assume $p \geq 3$. We add all vertices of the form $v((i_1, \dots, i_{p'}), k)$ with $p' \in [p-2]_0$, $i_1, \dots, i_{p'} \in [c]$, $k \in [c-1]_0$. The meaning of the vertex $v((i_1, \dots, i_{p'}), k)$ is that the vertex v is feasible in the original protocol, the child $u_{p'+1}$ is being tested and the first k inequalities of this child are true. For $j \in [p']$ the inequality $r_{u_j}^{i_j,0}(x) < r_{u_j}^{i_j,1}(y)$ is the first inequality of u_j which is false. We denote $w = v((i_1, \dots, i_{p'}), k)$. When $p' = 0$ and $k = 0$, w is just another name for v. It follows from the description of w that it has the following functions

$$\begin{array}{rclcrcll}
q_w^{j,0} &=& r_v^{j,0} & \text{and} & q_w^{j,1} &=& r_v^{j,1} & \text{for } j \in [c]; \\
q_w^{c+j,0} &=& -r_{u_j}^{i_j,0} - \varepsilon & \text{and} & q_w^{c+j,1} &=& -r_{u_j}^{i_j,1} & \text{for } j \in [p']; \\
q_w^{c+p'+j,0} &=& r_{u_{p'+1}}^{j,0} & \text{and} & q_w^{c+p'+j,1} &=& r_{u_{p'+1}}^{j,1} & \text{for } j \in [k]; \\
q_w^{j,0} &\equiv& 0 & \text{and} & q_w^{j,1} &\equiv& 1 & \text{for } j > c + p' + k.
\end{array}$$

We describe the children of w:

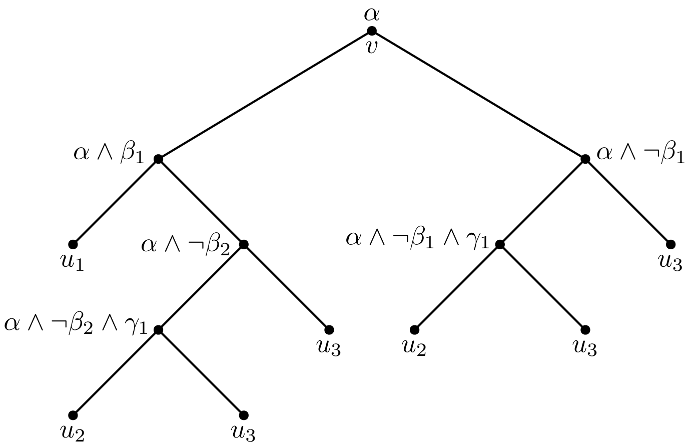
α
v
$\alpha \wedge \beta_1$
$\alpha \wedge \neg\beta_1$
u_1
$\alpha \wedge \neg\beta_2$
$\alpha \wedge \neg\beta_1 \wedge \gamma_1$
u_3
$\alpha \wedge \neg\beta_2 \wedge \gamma_1$
u_3
u_2
u_3
u_2
u_3

Figure 3.1: The substructure in the proof of Lemma 8 for $c = 2$ and $p = 3$

- If $p' < p - 2$, the first child of w is $v((i_1, \dots, i_{p'}, k+1), 0)$. This corresponds to the case when the inequality $r^{k+1,0}_{u_{p'+1}}(x) < r^{k+1,1}_{u_{p'+1}}(y)$ is false and the next child will be tested.
- If $p' = p - 2$, the first child of w is u_p. Here, $u_{p'+1}$ is the penultimate child and if it is not feasible (i.e. the inequality $r^{k+1,0}_{u_{p'+1}}(x) < r^{k+1,1}_{u_{p'+1}}(y)$ is false), the last child is automatically feasible.
- If $k < c - 1$, the second child of w is $v((i_1, \dots, i_{p'}), k + 1)$.
- If $k = c - 1$, the second child of w is $u_{p'+1}$.

See Fig. 3.1 for the substructure in the case $c = 2$ and $p = 3$. The formulas in the figure are

$$\begin{aligned}\alpha &\equiv r^{1,0}_v(x) < r^{1,1}_v(y) \wedge r^{2,0}_v(x) < r^{2,1}_v(y),\\ \beta_j &\equiv r^{j,0}_{u_1}(x) < r^{j,1}_{u_1}(y),\\ \gamma_j &\equiv r^{j,0}_{u_2}(x) < r^{j,1}_{u_2}(y).\end{aligned}$$

This concludes the description of the protocol and it remains to verify that it is a valid protocol solving f. By definition, each vertex has out-degree of at most two. Each substructure is acyclic and has a single source v and sinks $u_1, \dots, u_p$. Hence the whole underlying graph of the protocol is acyclic and has a single source v_0. The sinks of the new protocol are the sinks from the original protocol. Therefore the protocol satisfies the conditions (i) and (ii) of Definition 1.

The condition (a) of Definition 1 is satisfied because the constructed protocol has the same source with the same functions as the original protocol. It remains to prove the condition (b).

Consider a vertex w and assume it is feasible for $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$. The inequality

$$r^{k+1,0}_{u_{p'}+1}(x) < r^{k+1,1}_{u_{p'}+1}(y) \tag{3.1}$$

is either true or false.

- If (3.1) is false and $p' < p - 2$, then the first child $v((i_1, \dots, i_{p'}, k + 1), 0)$ is feasible because each of its inequalities is either contained among the

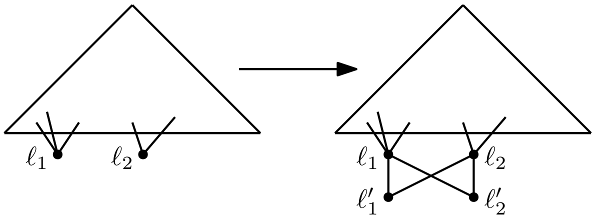

Figure 3.2: Modification of the graph in the proof of Lemma 9

inequalities of w or the inequality is $-r^{k+1,0}_{u_{p'}+1}(x) - \varepsilon < -r^{k+1,1}_{u_{p'}+1}(y)$ which is equivalent to (3.1) being false.

- If (3.1) is false and $p' = p - 2$, then none of the children $u_1, \dots, u_{p-1}$ is feasible. Because w contains inequalities $r^{j,0}_v(x) < r^{j,1}_v(y)$ for $j \in [c]$, the vertex v in the original protocol is feasible and hence at least one of the children u_1, $\dots$, u_p is feasible. Therefore the child u_p is feasible in the original protocol and also in the constructed protocol.

- If (3.1) is true and $k < c - 1$, then the child $v((i_1, \dots, i_{p'}), k + 1)$ is feasible because each of its inequalities is either contained among the inequalities of w or the inequality is (3.1).

- If (3.1) is true and $k = c - 1$, then $u_{p'+1}$ is feasible for the same reasons as in the previous case.

For each vertex $v \in V$, we added at most $\sum_{p'=0}^{p-2} c^{p'+1} \leq \max(c^p, p)$ vertices. Therefore the size of the protocol is at most $s \cdot \max(c^d, d)$. □

3.2 Protocols with a conjunction of inequalities

Lemma 9 as stated in Section 2.3:

Lemma 9. *Let P be a DAG communication protocol of degree 2 with a conjunction of c inequalities solving an n-bit partial monotone Boolean function f. If the size of the protocol P is s, then there exists a DAG communication protocol of degree 2 with equality solving f whose size is $\mathcal{O}\left(sc(n+1)^{2c+1}\right)$.*

Proof. Let $G = (V, E)$ be the underlying graph of P with functions $r^{j,0}_v$ and $r^{j,1}_v$ as in Definition 4. To reduce the number of cases to consider, we modify the underlying graph. To each vertex with exactly one child, we attach some sink. After this change, the size of the protocol is the same and the protocol still solves the function f. If there is only one sink, there is a trivial protocol solving f: The protocol of size one consisting only of this sink. We can therefore assume that there are at least two sinks. To reduce the number of cases even further, we modify the graph in the following way: For every sink $\ell_1 \in V$, we add a new sink ℓ_1' with the same functions $r^{j,0}_{\ell_1}$, $r^{j,1}_{\ell_1}$. The children of ℓ_1 are ℓ_1' and ℓ_2' for some other original sink ℓ_2. The modified protocol still solves f, but now every vertex has either two non-sink children or two sink children. The size of the modified protocol is at most $2s$. See Fig. 3.2.

Suppose w.l.o.g. that $r_v^{j,0}(z), r_v^{j,1}(z) \in \{0, \dots, 2^{n+1} - 1\}$ for every non-sink vertex $v \in V$, $z \in \{0,1\}^n$ and $j \in [c]$. (There are at most 2^{n+1} values of $r_v^{j,0}(z)$ and $r_v^{j,1}(z)$ for fixed j and v; the only thing that matters is the relative order of the values.) We consider $r_v^{j,0}(z)$ and $r_v^{j,1}(z)$ to be $(n+1)$-bit numbers and denote by $r_v^{j,0}(z)[i]$ and $r_v^{j,1}(z)[i]$ the i-th most significant bit; that is $r_v^{j,0}(z) = \sum_{i=1}^{n+1} 2^{n+1-i} r_v^{j,0}(z)[i]$.

In the constructed procotol with equality, we assume that all outputs of functions at non-sink vertices are finite sequences of bits. (To do that, we fix a bijection between binary sequences and a subset of real numbers. We may for example identify the sequence $b_1 b_2 \dots b_\ell$ with the integer with binary expansion $1 b_1 b_2 \dots b_\ell$.)

In every non-sink vertex v of the constructed protocol, both functions r_v^0 and r_v^1 will have the same fixed number of output bits. Testing the equality of $r_v^0(x) = b_1 b_2 \dots b_\ell$ and $r_v^1(x) = c_1 c_2 \dots c_\ell$ is then equivalent to testing whether the conjunction

$$b_1 = c_1 \wedge \dots \wedge b_\ell = c_\ell$$

is true. Conversely, every conjunction of equalities in the form

$$f_1(x) = g_1(y) \wedge \dots \wedge f_\ell(x) = g_\ell(y),$$

where $f_i, g_i \colon \{0,1\}^n \to \{0,1\}$, can be expressed by setting the functions $r_v^0(x) := f_1(x) f_2(x) \dots f_\ell(x)$ and $r_v^1(y) := g_1(y) g_2(y) \dots g_\ell(y)$. The important thing is that for each equality the left-hand side is a function of x and the right-hand side is a function of y.

To streamline the description of the protocol, we do not explicitly describe the functions $r_v^0(x)$ and $r_v^1(y)$ for each vertex v. Instead, we label vertices with conjunctions of equalities such that the left-hand side of each equality is a function of x and the right-hand side of each equality is a function of y. The actual functions can be deduced using the method from the previous paragraph.

The key idea of the simulation is that an inequality can be expressed as one of $n+1$ equalities. Let a, b be two $(n+1)$-bit numbers with the binary expansions $a = a_1 a_2 \dots a_{n+1}$ and $b = b_1 b_2 \dots b_{n+1}$. The observation we use is that

$$a < b \Leftrightarrow \exists i \in [n+1] \Big(a_1 = b_1 \wedge \dots \wedge a_{i-1} = b_{i-1} \wedge a_i = 0 \wedge 1 = b_i \Big).$$

The left-hand side of each equality is a function of a and the right-hand side is a function of b. Using the method from the previous paragraph for converting a conjunction of equalities into a single equality, we obtain

$$a < b \Leftrightarrow \exists i \in [n+1] (a_1 a_2 \dots a_{i-1} a_i 1 = b_1 b_2 \dots b_{i-1} 0 b_i).$$

For each non-sink vertex $v \in V$, $i \in [n+1]_0$ and $j \in [c]$, we define the following conjunction of equalities

$$\phi_v^{j,(i,=)} \equiv \bigwedge_{k=1}^{i} \left(r_v^{j,0}(x)[k] = r_v^{j,1}(y)[k] \right).$$

And for each non-sink vertex $v \in V$, $i \in [n+1]$ and $j \in [c]$, we define the following conjunctions of equalities

$$\begin{aligned} \phi_v^{j,(i,\neq)} &\equiv \phi_v^{j,(i-1,=)} \wedge r_v^{j,0}(x)[i] = 1 - r_v^{j,1}(y)[i], \\ \phi_v^{j,(i,<)} &\equiv \phi_v^{j,(i-1,=)} \wedge r_v^{j,0}(x)[i] = 0 \wedge 1 = r_v^{j,1}(y)[i], \\ \phi_v^{j,(i,>)} &\equiv \phi_v^{j,(i-1,=)} \wedge r_v^{j,0}(x)[i] = 1 \wedge 0 = r_v^{j,1}(y)[i]. \end{aligned}$$

The formula $\phi_v^{j,(i,=)}$ means that the first i bits of $r_v^{j,0}(x)$ and $r_v^{j,1}(y)$ are equal. The formula $\phi_v^{j,(i,\neq)}$ means that the first $i-1$ bits of $r_v^{j,0}(x)$ and $r_v^{j,1}(y)$ are equal and the i-th bit differs. The meaning of formulas $\phi_v^{j,(i,<)}$ and $\phi_v^{j,(i,>)}$ is analogous.

Similarly to expressing inequality as one of $n+1$ equalities, we can express the conjunction of c inequalities as one of $(n+1)^c$ equalities

$$\bigwedge_{j=1}^{c}\left(r_v^{j,0}(x) < r_v^{j,1}(y)\right) \Leftrightarrow (\exists(i_1,\dots,i_c)\in[n+1]^c)\bigwedge_{j=1}^{c}\phi_v^{j,(i_j,<)}.$$

For each non-sink vertex v and $I=(i_1,\dots,i_c)\in[n+1]^c$, we define

$$\Phi_v^I \equiv \bigwedge_{j=1}^{c}\phi_v^{j,(i_j,<)}.$$

For each non-sink vertex v and $G=(g_1,\dots,g_{c'})$ where it holds $c'\leq c$ and $g_j\in[n+1]_0\times\{=,\neq,<,>\}$, we define

$$\Psi_v^G \equiv \bigwedge_{j=1}^{c'}\phi_v^{j,g_j}.$$

Now, we construct the protocol P' with equality. Let the underlying graph of the new protocol P' be $G'=(V',E')$. For each sink $\ell\in V$ with the index i from Definition 1 (ii), we put ℓ into V' with the functions $r_\ell^0(z)=z_i+1$ and $r_\ell^1(z)=z_i$. For each non-sink vertex $v\in V$, we add $(n+1)^c$ vertices $v(I)$ for $I\in[n+1]^c$. Each $v(I)$ is labeled with the conjunction Φ_v^I.

Let u_1 and u_2 be the children of a non-sink vertex $v\in V$. If $v(I)$ is feasible for $x\in f^{-1}(0)$ and $y\in f^{-1}(1)$, then Φ_v^I implies that v is feasible for x and y in the original protocol. That in turn implies that u_1 or u_2 is feasible for x and y. From that it follows that at least one of the vertices in V' corresponding to u_1 and u_2 (i.e. $u_1(I)$, $u_2(I)$ for $I\in[n+1]^c$ if u_1 and u_2 are non-sink vertices or u_1 and u_2 if u_1 and u_2 are sinks) is feasible. To simulate the original protocol, we add a substructure connecting vertices $v(I)$ for $I\in[n+1]^c$ with the vertices corresponding to the children u_1 and u_2.

Let v_0 be the source of G. Because v_0 is feasible for $x\in f^{-1}(0)$ and $y\in f^{-1}(1)$, one of the vertices $v_0(I)$ for $I\in[n+1]^c$ is feasible for x and y in the constructed protocol. We add a new source t labeled with an empty conjunction and a substructure connecting t with the vertices $v_0(I)$ for $I\in[n+1]^c$.

Before precisely describing the substructures, we explain the high-level idea. To this end, consider a non-sink vertex $v\in V$ with two non-sink children u_1 and u_2. To connect $v(I)$ with $u_1(I')$ and $u_2(I'')$ for $I',I''\in[n+1]^c$, we will be adding vertices of the form $v(I,G_1)$ and $v(I,(j,a,b),G_2)$. Each vertex $v(I,G_1)$ is labeled with $\Phi_v^I\wedge\Psi_{u_1}^{G_1}$. Each vertex $v(I,(j,a,b),G_2)$ is labeled with $\Phi_v^I\wedge\phi_{u_1}^{j,(a,b)}\wedge\Psi_{u_2}^{G_2}$. If the vertex $v(I,G_1)$ is feasible, it means that v is feasible in the original protocol and we test whether u_1 is also feasible. The value G_1 encodes a partial information about the inequalities $r_{u_1}^{j,0}(x) < r_{u_1}^{j,1}(y)$ for $j\in[c]$. Traversing the graph via feasible vertices corresponds to extending the partial information G_1, inequality by inequality and each inequality bit by bit. Either we find out that all inequalities of u_1 are true and thus find $I'\in[n+1]^c$ such that $u_1(I')$ is feasible, or at least one of the inequalities of u_1 is false. In the case when at least one of the inequalities of

u_1 is false, in particular $\phi_{u_1}^{j,(a,b)}$ holds for $(a,b) \in [n+1] \times \{>\}$ or $(a,b) = (n+1, =)$, we move to a vertex $v(I, (j,a,b), G_2)$ for $G_2 = ((0, =))$. In general, if the vertex $v(I, (j,a,b), G_2)$ is feasible, then the vertex v is feasible in the original protocol, but the vertex u_1 is not feasible because the j-th inequality does not hold. Then the vertex u_2 is feasible in the original protocol and hence there is an I'' such that $u_2(I'')$ is feasible. We extend the partial information G_2, inequality by inequality, bit by bit until we find such I''.

We describe the substructure for a non-sink vertex $v \in V$. If v has two sink children ℓ_1 and ℓ_2, the substructure is trivial: For each $I \in [n+1]^c$, we attach ℓ_1 and ℓ_2 as children to $v(I)$. Let us assume that v has two non-sink children u_1 and u_2.

We denote $G_1 = (g_1^1, \dots, g_{c_1}^1)$ and $G_2 = (g_1^2, \dots, g_{c_2}^2)$.

We say for u_1 that:

The j-th inequality is $\begin{cases} \textit{confirmed} & \text{if } g_j^1 \in [n+1] \times \{<\}; \\ \textit{active} & \text{if } g_j^1 \in [n]_0 \times \{=\} \cup [n+1] \times \{\neq\}. \end{cases}$

Similarly for u_2, the j-th inequality is $\begin{cases} \textit{confirmed} & \text{if } g_j^2 \in [n+1] \times \{<\}; \\ \textit{active} & \text{if } g_j^2 \in [n]_0 \times \{=\}. \end{cases}$

For each $I \in [n+1]^c$, we add into V' all vertices $v(I, G_1)$ such that $c_1 \in [c]$, the first $c_1 - 1$ inequalities of u_1 are confirmed and the c_1-th inequality is active. Furthermore, we add into V' all vertices $v(I, (j,a,b), G_2)$ such that $j \in [c]$, $(a,b) \in [n+1] \times \{>\}$ or $(a,b) = (n+1, =)$, $c_2 \in [c]$, the first $c_2 - 1$ inequalities of u_2 are confirmed and the c_2-th inequality is active. We use $v(I, ((0, =)))$ as an alternative name for $v(I)$. (It is consistent to do that as for $G_1 = ((0, =))$ the formula $\Psi_{u_1}^{G_1}$ is just an empty conjunction.) We also denote $g_j^k = (a_j^k, b_j^k)$ for $k \in \{1,2\}$ and $j \in [c_k]$.

We describe the children of the vertices:

Case 1 Vertex $v(I, G_1)$. We are testing the child u_1.

Case 1.1 $b_{c_1}^1 =$ '$=$' The first child of $v(I, G_1)$ is

$$v(I, (g_1^1, \dots, g_{c_1-1}^1, (a_{c_1}^1 + 1, \neq))).$$

If $a_{c_1}^1 < n$, the second child of $v(I, G_1)$ is

$$v(I, (g_1^1, \dots, g_{c_1-1}^1, (a_{c_1}^1 + 1, =))).$$

If $a_{c_1}^1 = n$, the second child of $v(I, G_1)$ is

$$v(I, (c_1, n+1, =), ((0, =))).$$

Case 1.2 $b_{c_1}^1 =$ '$\neq$' The first child of $v(I, G_1)$ is

$$v(I, (c_1, a_{c_1}^1, >), ((0, =))).$$

If $c_1 < c$, the second child of $v(I, G_1)$ is

$$v(I, (g_1^1, \dots, g_{c_1-1}^1, (a_{c_1}^1, <), (0, =))).$$

If $c_1 = c$, the second child of $v(I, G_1)$ is

$$u_1((a_1^1, \dots, a_c^1)).$$

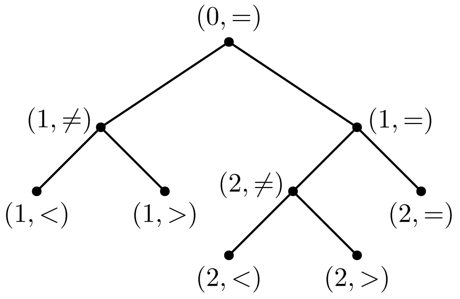

Figure 3.3: Case 1 in the proof of Lemma 9

Case 2 Vertex $v(I,(j,a,b),G_2)$. We know that the child u_1 is not feasible and we are extending G_2 to find $I'' \in [n+1]^c$ such that $u_2(I'')$ is feasible. It holds by definition that $b_{c_2} =$ '$=$'.

Case 2.1 $c_2 < c$ The first child of $v(I,(j,a,b),G_2)$ is

$$v(I,(j,a,b),(g^2_1,\ldots,g^2_{c_2-1},(a^2_{c_2}+1,<),(0,=))).$$

If $a^2_{c_2} < n-1$, the second child of $v(I,(j,a,b),G_2)$ is

$$v(I,(j,a,b),(g^2_1,\ldots,g^2_{c_2-1},(a^2_{c_2}+1,=))).$$

If $a^2_{c_2} = n-1$, the second child of $v(I,(j,a,b),G_2)$ is

$$v(I,(j,a,b),(g^2_1,\ldots,g^2_{c_2-1},(n+1,<),(0,=))).$$

Case 2.2 $c_2 = c$ The first child of $v(I,(j,a,b),G_2)$ is

$$u_2((a^2_1,\ldots,a^2_c)).$$

If $a^2_c < n-1$, the second child of $v(I,(j,a,b),G_2)$ is

$$v(I,(j,a,b),(g^2_1,\ldots,g^2_{c-1},(a^2_c+1,=))).$$

If $a^2_c = n-1$, the second child of $v(I,(j,a,b),G_2)$ is

$$u_2((a^2_1,\ldots,a^2_{c-1},n+1)).$$

Fig. 3.3 shows Case 1 for one inequality and $n = 1$; Fig. 3.4 shows Case 2 for one inequality.

We describe the substructure connecting the new source t with the vertices $v_0(I)$ for $I \in [n+1]^c$. The substructure is almost identical to Case 2. We will be adding vertices of the form $t(g_1,\ldots,g_{c'})$ labeled with $\Psi^{(g_1,\ldots,g_{c'})}_{v_0}$. As in Case 2, we say that:

$$\text{The } j\text{-th inequality of } v_0 \text{ is } \begin{cases} \textit{confirmed} & \text{if } g_j \in [n+1]\times\{<\}; \\ \textit{active} & \text{if } g_j \in [n]_0\times\{=\}. \end{cases}$$

We use $t((0,=))$ as an alternative name for the vertex t (it is consistent to do that as $\Psi^{((0,=))}_{v_0}$ is an empty conjunction). We denote $(a_j,b_j) = g_j$ for $j \in [c']$. We add all vertices $t(g_1,\ldots,g_{c'})$ such that $c' \in [c]$, the first $c'-1$ inequalities are confirmed and the c'-th inequality is active.

We describe the children. For the active inequality, it holds $b_j =$ '$=$'.

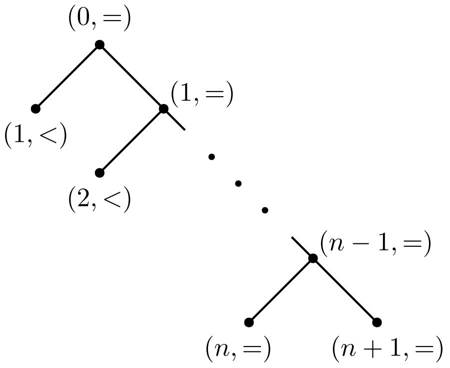

Figure 3.4: Case 2 in the proof of Lemma 9

Case 0.1 $c' < c$ The active inequality is not the last one. The first child of the vertex $t(g_1, \ldots, g_{c'})$ is

$$t(g_1, \ldots, g_{c'-1}, (a_{c'} + 1, <), (0, =)).$$

If $a_{c'} < n - 1$, the second child of $t(g_1, \ldots, g_{c'})$ is

$$t(g_1, \ldots, g_{c'-1}, (a_{c'} + 1, =)).$$

If $a_{c'} = n - 1$, the second child of $t(g_1, \ldots, g_{c'})$ is

$$t(g_1, \ldots, g_{c'-1}, (n + 1, <), (0, =)).$$

Case 0.2 $c' = c$ The active inequality is the last one. The first child of the vertex $t(g_1, \ldots, g_c)$ is

$$v_0((a_1, \ldots, a_{c-1}, a_c + 1)).$$

If $a_c < n - 1$, the second child of $t(g_1, \ldots, g_c)$ is

$$t(g_1, \ldots, g_{c-1}, (a_c + 1, =)).$$

If $a_c = n - 1$, the second child of $t(g_1, \ldots, g_c)$ is

$$v_0((a_1, \ldots, a_{c-1}, n + 1)).$$

The description of the protocol P' is complete. We verify that it is a valid protocol and that it solves the function f.

Each vertex has by definition out-degree 2. The substructure for the source is acyclic and has one source t and sinks $v_0(I)$ for $I \in [n+1]^c$. The substructure for a non-sink vertex $v \in V$ with two non-sink children u_1 and u_2 is also acyclic; it has sources $v(I)$ for $I \in [n+1]^c$ and sinks $u_1(I')$ and $u_2(I'')$ for $I', I'' \in [n+1]^c$. The substructure for a non-sink vertex $v \in V$ with two sink children ℓ_1 and ℓ_2 is acyclic, has sources $v(I)$ for $I \in [n+1]^c$ and sinks ℓ_1 and ℓ_2. Therefore the whole protocol is acyclic and has one source t and its sinks are the vertices corresponding to the sinks in the original protocol. Hence the protocol satisfies the conditions (i) and (ii) of Definition 1.

The condition (a) is also satisfied as the source t is labeled with an empty conjunction. It remains to verify the condition (b). The verification is based on the following implications.

For every non-sink vertex $v \in V$, $j \in [c]$, $i \in [n]_0$, we have

$$\phi_v^{j,(i,=)} \to \left(\phi_v^{j,(i+1,=)} \vee \phi_v^{j,(i+1,\neq)}\right). \tag{3.2}$$

For every non-sink vertex $v \in V$, $j \in [c]$, $i \in [n+1]$, we have

$$\phi_v^{j,(i,\neq)} \to \left(\phi_v^{j,(i,<)} \vee \phi_v^{j,(i,>)}\right). \tag{3.3}$$

For every non-sink vertex $v \in V$, $j \in [c]$, $i \in [n-2]_0$, assuming v is feasible in the original protocol for $x \in f^{-1}(0)$, $y \in f^{-1}(1)$, we have

$$\phi_v^{j,(i,=)} \to \left(\phi_v^{j,(i+1,<)} \vee \phi_v^{j,(i+1,=)}\right). \tag{3.4}$$

If v is feasible, then the inequality $r_v^{j,0}(x) < r_v^{j,1}(y)$ is true. The implication (3.4) follows.

For every non-sink vertex $v \in V$, $j \in [c]$, assuming again v is feasible in the original protocol for $x \in f^{-1}(0)$, $y \in f^{-1}(1)$, we have

$$\phi_v^{j,(n-1,=)} \to \left(\phi_v^{j,(n,<)} \vee \phi_v^{j,(n+1,<)}\right). \tag{3.5}$$

Because the inequality $r_v^{j,0}(x) < r_v^{j,1}(y)$ is true, the formula $\phi_v^{j,(n+1,=)}$ cannot be true. The implication (3.5) follows.

The general principle we implicitly use in the verification is that if for any formulas $\alpha_0, \alpha_1, \alpha_2$ it holds

$$\alpha_0 \to (\alpha_1 \vee \alpha_2),$$

then it also holds

$$\beta \wedge \alpha_0 \to ((\gamma \wedge \alpha_1) \vee (\delta \wedge \alpha_2)),$$

where

$$\beta \equiv \bigwedge_{\xi \in S} \xi \qquad \gamma \equiv \bigwedge_{\xi \in S'} \xi \qquad \delta \equiv \bigwedge_{\xi \in S''} \xi$$

for some sets of equalities S, S', S'' satisfying $S' \subseteq S$ and $S'' \subseteq S$.

We verify the condition for vertices added in the above cases:

Case 0 The vertex $t(g_1, \ldots, g_{c'})$ is feasible. The vertex v_0 is feasible. If it holds $a_{c'} < n-1$, the feasibility of one of the children (in both Case 0.1 and Case 0.2) follows from (3.4). If $a_{c'} = n-1$, the feasibility of one of the children follows from (3.5).

Case 1 The vertex $v(I, G_1)$ is feasible.

Case 1.1 $b_{c_1}^1 =$ '$=$' The feasibility of one of the children follows from (3.2) for u_1.

Case 1.2 $b_{c_1}^1 =$ '$\neq$' The feasibility of one of the children follows from (3.3) for u_1.

Case 2 The vertex $v(I, (j, a, b), G_2)$ is feasible. The formula Φ_v^I implies that v is feasible in the original protocol. The formula $\phi_{u_1}^{j,(a,b)}$ implies that u_1 is not feasible. Therefore u_2 is feasible. If $a_{c_2}^2 < n-1$, the feasibility of one of the children (in both Case 2.1 and Case 2.2) follows from (3.4) for u_2. If $a_{c_2}^2 = n-1$, the feasibility of one of the children follows from (3.5) for u_2.

Finally, we estimate the size of the protocol. For each sink $\ell \in V$, we add just one vertex to V'. For each non-sink vertex $v \in V$, we add vertices of the form $v(I, G_1)$ and $v(I, (j, a, b), G_2)$. There are $(n+1)^c$ choices for I. There are at most

$$\sum_{c_1=1}^{c} ((n+1)^{c_1-1} \cdot 2(n+1)) = \frac{2(n+1)((n+1)^c - 1)}{n} = \mathcal{O}((n+1)^c)$$

choices for G_1 because $g_1^1, \ldots, g_{c_1-1}^1 \in [n+1] \times \{<\}$, $g_{c_1}^1 \in [n]_0 \times \{=\} \cup [n+1] \times \{\neq\}$. There are $c(n+2)$ choices for (j, a, b) because $j \in [c]$ and $(a, b) \in [n+1] \times \{>\}$ or $(a, b) = (n+1, =)$. There are

$$\sum_{c_2=1}^{c} ((n+1)^{c_2-1} \cdot n) = (n+1)^c - 1$$

choices for G_2 because $g_1^2, \ldots, g_{c_2-1}^2 \in [n+1] \times \{<\}$, $g_{c_2}^2 \in [n-1]_0 \times \{=\}$. Finally, we add at most

$$\sum_{c'=1}^{c} ((n+1)^{c'-1} \cdot n) = (n+1)^c - 1$$

vertices of the form $t(g_1, \ldots, g_{c'})$ because $g_1, \ldots, g_{c'-1} \in [n+1] \times \{<\}$ and $g_{c'} \in [n-1]_0$.

After the initial transformations, it holds that $|V| \leq 2s$. The total number of vertices in V' is then at most

$$2s(n+1)^c \left(\mathcal{O}((n+1)^c) + c(n+2)((n+1)^c - 1)\right) + (n+1)^c - 1.$$

The size of the protocol is therefore $\mathcal{O}(sc(n+1)^{2c+1})$. □

Conclusion

We proved several relations between different types of protocols. From our perspective, the most important corollaries of our theorems are: (1) protocols of degree 2 with equality are at least as strong as protocols of degree 2 with inequality; (2) protocols of degree 2 with equality have the same strength as protocols of degree 2 with a conjunction of two inequalities. Furthermore, we defined protocols with disjointness and proved that they are at least as strong as protocols with equality.

There are exponential lower bounds for protocols with inequality. The key question is whether the possibly stronger protocols considered in this thesis may be applied to obtain lower bounds in proof complexity. To answer this question, it is necessary to find out whether lower bounds for protocols with equality, or even for protocols with disjointness, are possible.

Bibliography

[AB87] Noga Alon and Ravi B. Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7(1):1–22, 1987.

[Ajt83] Miklós Ajtai. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1 – 48, 1983.

[Ajt94] Miklós Ajtai. The complexity of the pigeonhole principle. *Combinatorica*, 14(4):417–433, 1994.

[CR79] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979.

[FSS84] Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

[GGKS18] Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 902–911. ACM, 2018.

[Hak85] Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.

[HP18] Pavel Hrubeš and Pavel Pudlák. A note on monotone real circuits. *Information Processing Letters*, 131:15–19, 2018.

[IS14] Dmitry Itsykson and Dmitry Sokolov. Lower bounds for splittings by linear combinations. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part II*, volume 8635 of *Lecture Notes in Computer Science*, pages 372–383. Springer, 2014.

[Juk12] Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*, volume 27 of *Algorithms and combinatorics*. Springer, 2012.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

[Kra94] Jan Krajíček. Lower bounds to the size of constant-depth propositional proofs. *The Journal of Symbolic Logic*, 59(1):73–86, 1994.

[Kra97] Jan Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *The Journal of Symbolic Logic*, 62(2):457–486, 1997.

[Kra98] Jan Krajíček. Interpolation by a game. *Mathematical Logic Quarterly*, 44:450–458, 1998.

[Kra16] Jan Krajíček. Randomized feasible interpolation and monotone circuits with a local oracle. *CoRR*, abs/1611.08680, 2016.

[Kra18] Jan Krajíček. *Proof complexity.* Cambridge University Press, 2018. In preparation.

[KW88] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 539–550. ACM, 1988.

[Pud97] Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *The Journal of Symbolic Logic*, 62(3):981–998, 1997.

[Pud00] Pavel Pudlák. Proofs as games. *The American Mathematical Monthly*, 107(6):541–550, 2000.

[Pud10] Pavel Pudlák. On extracting computations from propositional proofs (a survey). In Kamal Lodaya and Meena Mahajan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010, December 15-18, 2010, Chennai, India*, volume 8 of *LIPIcs*, pages 30–41. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010.

[Pud13] Pavel Pudlák. *Logical Foundations of Mathematics and Computational Complexity - A Gentle Introduction.* Springer monographs in mathematics. Springer, 2013.

[Raz85] Alexander A. Razborov. Lower bounds on the monotone complexity of some Boolean functions. *Doklady Akademii Nauk SSSR*, 285:798–801, 1985.

[Raz95] Alexander A. Razborov. Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izv. Ross. Akad. Nauk Ser. Mat.*, 59(1):201–224, 1995.

[Ros97] Arnold Rosenbloom. Monotone real circuits are more powerful than monotone boolean circuits. *Information Processing Letters*, 61(3):161–164, 1997.

[San12] Rahul Santhanam. Ironic complicity: Satisfiability algorithms and circuit lower bounds. *Bulletin of the EATCS*, 106:31–52, 2012.

[Sha49] Claude E. Shannon. The synthesis of two-terminal switching circuits. *Bell Systems Technical Journal*, 28:59–98, 1949.

[Sok17] Dmitry Sokolov. Dag-like communication and its applications. In Pascal Weil, editor, *Computer Science - Theory and Applications - 12th International Computer Science Symposium in Russia, CSR 2017, Kazan, Russia, June 8-12, 2017, Proceedings*, volume 10304 of *Lecture Notes in Computer Science*, pages 294–307. Springer, 2017.

[Wil14] Ryan Williams. Nonuniform ACC circuit lower bounds. *Journal of the ACM*, 61(1):2:1–2:32, 2014.

[Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In Michael J. Fischer, Richard A. DeMillo, Nancy A. Lynch, Walter A. Burkhard, and Alfred V. Aho, editors, *Proceedings of the 11h Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA*, pages 209–213. ACM, 1979.