Summer School “Computational Complexity Theory”

Alexander Kulikov, Ivan Mihajlin, Alexander Smal, Dmitry Sokolov

August 2–6, 2021, Sirius

Contents

1 Commons

1.1 Asymptotic notation

We will be using standard big-O notation throughout the lecture notes. For two functions $f, g\colon \mathbb{Z}_{>0} \to \mathbb{R}_{>0}$,

- $f = O(g)$ (f grows no faster than g), if there exists a constant c such that $f(n) \le c \cdot g(n)$ for every n;
- $f = o(g)$ (f grows slower than g), if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$;
- $f = \Omega(g)$ (f grows no slower than g), if $g = O(f)$;
- $f = \omega(g)$ (f grows faster than g), if $g = o(f)$.

By a superlinear or a superpolynomial function we mean a function that grows faster than a linear or a polynomial function (that is, $w(n)$ and $n^{\omega(1)}$).

Problem 1.1 Give example of a function that grows faster than any polynomial (n^c for a constant c), but slower than any exponential function (c^n for a constant $c > 1$).

We also use a standard notation for a set of integers between 1 and n: $[n] := \{1, \dots, n\}$.

1.2 Boolean functions

By $B_{n,m}$ we denote the set of all *Boolean functions* $f\colon \{0,1\}^n \to \{0,1\}^m$ with n inputs and m outputs. By $B_n = B_{n,1}$ we denote the set of all *Boolean predicates* on n inputs. A function $f \in B_{n,m}$ is called *symmetric* if its value depends on the sum of input bits only (equivalently, if its value does not change under permuting the input values).

Problem 1.2 Find the size of $B_{n,m}$.

Below, we show a few examples of interesting Boolean functions. Some of them demonstrate, in particular, how a computational problem is converted into a sequence of Boolean functions. Note also that there is a natural one-to-one correspondence between infinite sequences $\{f_n \in B_n\}_{n=1}^{\infty}$ of Boolean functions and languages $L \subseteq \{0,1\}^*$: $f_n(x) = 1$ iff $|x| = n$ and $x \in L$.

- Symmetric functions:

  Parity: $\oplus_n(x_1, \dots, x_n) = x_1 \oplus \cdots \oplus x_n$.

  Majority: $\mathsf{MAJ}(x_1, \dots, x_n) = [x_1 + \cdots + x_n > n/2]$.

  Threshold: $\mathsf{THR}_n^k(x_1, \dots, x_n) = [x_1 + \cdots + x_n \ge k]$.

  OR: $\vee_n(x_1, \dots, x_n) = x_1 \vee \cdots \vee x_n = \mathsf{THR}_n^1(x_1, \dots, x_n)$.

  Counting: $\mathsf{MOD}_n^{m,r}(x_1, \dots, x_n) = [x_1 + \cdots + x_n \equiv r \bmod m]$.

  Sum: $\mathsf{SUM}_n \in B_{n,\ell}$, where $\ell = \lceil \log_2(n+1) \rceil$,

  $$\mathsf{SUM}_n(x_1, \dots, x_n) = (w_0, w_1, \dots, w_{\ell-1})\colon \sum_{i=1}^{n} x_i = \sum_{i=0}^{\ell-1} 2^i w_i\,.$$

- Functions corresponding to NP-complete problems:

  Hamiltonian cycle: $\mathsf{HAM}_n\colon \{0,1\}^{\binom{n}{2}} \to \{0,1\}$ outputs 1 iff the input graph (given by its adjacency matrix) contains a Hamiltonian path.

Clique: $\mathrm{CLIQUE}_n^k\colon \{0,1\}^{\binom{n}{2}} \to \{0,1\}$ outputs 1 iff the input graph contains a clique (that is, a complete subgraph) of size k.

- Other functions:

  Multiplexer: $M \in B_{n+k}$, where $n = 2^k$, is defined as follows:

  $$M(x_0, \dots, x_{k-1}, y_0, \dots, y_{n-1}) = y_\ell\,,$$

  where $0 \le \ell < n$ is an integer whose binary representation is $(x_0, \dots, x_{k-1})$. In other words, it takes a bit sequence y of length n and (the binary representation of) an index x and outputs the corresponding bit of y. (Also known as storage access function).

A *conjunctive normal form* (CNF) is a conjunction of clauses where a clause is a disjunction of variables and their negations:

$$(\neg x_2 \vee x_3) \wedge (x_1) \wedge (x_3 \vee \neg x_1 \vee x_2)\,.$$

Problem 1.3 Prove that for each $f \in B_n$, there exists a CNF computing f.

Problem 1.4 Give example of a function from B_n such that the size of any CNF computing it is $\Omega(2^n)$.

1.3 Satisfiability

The propositional satisfiability (SAT) problem is: given a Boolean formula in CNF, check whether it is possible to assign Boolean values to its variables so that the formula evaluates to True. For example, the formula:

$$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee x_2) \wedge (\neg x_2 \vee \neg x_3) \wedge (x_3 \vee \neg x_1)$$

is satisfiable: to satisfy it, one assigns $x_1 = 1$, $x_2 = 0$, $x_3 = 1$ (as usual, we use $0/1$ as shortcuts for False/True). However, by adding a clause $(\neg x_1 \vee x_2 \vee \neg x_3)$ to the formula, one makes it unsatisfiable.

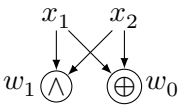
The state-of-the-art SAT-solvers are extremely efficient: they routinely solve instances from industry with millions of variables. It is easy to install and use them. Say, one can check the satisfiability of two formulas mentioned above using the following short `Python` code.

```
from pysat.solvers import Solver

clauses = [[1, -2, 3], [1, 2], [-2, -3], [3, -1]]
solver = Solver(bootstrap_with=clauses)
print(solver.solve())
print(solver.get_model())

solver.add_clause([-1, 2, -3])
print(solver.solve())
```

```
True
[1, -2, 3]
False
```

Problem 1.5 Try to come up with a relatively small CNF formula (say, with 500 variables) that is hard for the state-of-the-art SAT-solvers.

```python
def sum2(x1, x2):
    w0 = x1 ^ x2
    w1 = x1 * x2
    return w0, w1
```




```python
def sum3(x1, x2, x3):
    a = x1 ^ x2
    b = x2 ^ x3
    c = a | b
    w0 = a ^ x3
    w1 = c ^ w0
    return w0, w1
```

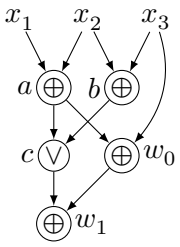


Figure 1: Optimal size straight line programs and circuits for SUM_2 and SUM_3. These two circuits are known as *half adder* and *full adder*.

1.4 Boolean circuits

A Boolean *straight line program* of size r for input variables $(x_1, \dots, x_n)$ is a sequence of r instructions where each instruction $g \leftarrow h \circ k$ applies a binary Boolean operation $\circ$ to two operands h, k each of which is either an input bit or the result of a previous instruction. If m instructions are designated as outputs, the straight line program computes a function $\{0,1\}^n \to \{0,1\}^m$ in a natural way. We denote the set of all such functions by $B_{n,m}$ and we let $B_n = B_{n,1}$. For a Boolean function $f\colon \{0,1\}^n \to \{0,1\}^m$, by $\mathsf{C}(f)$ we denote the minimum size of a straight line program computing f. A Boolean *circuit* shows a flow graph of a program.

Figure 1 gives an example for the $\mathsf{SUM}_n\colon \{0,1\}^n \to \{0,1\}^\ell$ function that computes the binary representation of the sum of n bits:

$$\mathsf{SUM}_n(x_1, \dots, x_n) = (w_0, w_1, \dots, w_{\ell-1})\colon \sum_{i=1}^{n} x_i = \sum_{i=0}^{\ell-1} 2^i w_i, \text{ where } \ell = \lceil \log_2(n+1) \rceil .$$

This function transforms n bits of weight 0 into ℓ bits of weights $(0, 1, \dots, \ell-1)$. The straight line programs are given in `Python` programming language. This makes particularly easy to verify their correctness. For example, the program for SUM_3 can be verified with just three lines of code:

```python
from itertools import product

for x1, x2, x3 in product(range(2), repeat=3):
    w0, w1 = sum3(x1, x2, x3)
    assert x1 + x2 + x3 == w0 + 2 * w1
```

Problem 1.6 Prove that for any function from B_n, there exists a circuit of size $O(2^n)$ computing it.

Problem 1.7 Prove that $\mathsf{C}(\oplus_n) = n - 1$.

Determining the exact value (as well as proving lower and upper bounds) of $\mathsf{C}(f)$ might be tricky. See Figure 2 for a toy example.

Problem 1.8 Design an algorithm that takes a circuit with n inputs and s gates and produces a CNF formula with $O(s+n)$ clauses that is equisatisfiable to the circuit in time $O(s+n)$.

1.5 Complexity classes

Let P be some computational problem. For our tasks, it will be enough to consider only "YES-or-NO" problems, *decision problems*. So, let's assume that every instance x of the problem P has answer either

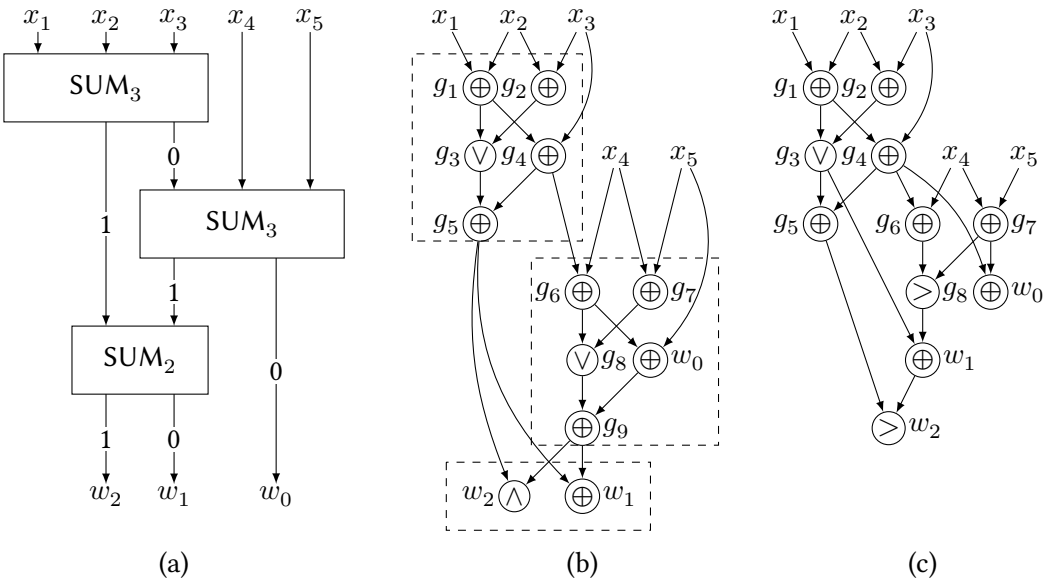


Figure 2: (a) A schematic circuit for SUM$_5$ composed out of two full adders and one half adder. (b) The corresponding circuit of size 12. (c) An improved circuit of size 11.

"YES" or "NO". To make it formal, we will associate P with the set of all "YES"-instances coded in binary. Now we can think of any computational problem P as a set of bit strings

$$P \subseteq \{0,1\}^*.$$

For example, the computational problem of checking that a graph is connected consists of encodings of all connected graphs.

Definition 1.1

We say that computational problem P is *computable in time* $t(n)$ if there is a Turing machine T such that:

- if $x \in P$, $T(x)$ outputs 1,
- otherwise, $T(x)$ outputs 0,
- the number of steps $T(x)$ does is at most $t(|x|)$.

Now we are ready to define the complexity class P.

Definition 1.2

Complexity class P is a set of all computational problems computable in polynomial time.

To define complexity class NP, we will need a notion of non-deterministic computation.

Definition 1.3

We say that computational problem P is *non-deterministically computable in time* $t(n)$ if there is a Turing machine T with two input tapes such that

- if $x \in P$, then there exists $y \in \{0,1\}^*$ such that $T(x,y)$ outputs 1,
- otherwise, $T(x,y)$ outputs 0 for all $y \in \{0,1\}^*$,
- the number of steps $T(x,y)$ does is at most $t(|x|)$.

Remark

Note that in $t(n)$ steps Turing machine can not read more then $t(n)$ bits of y. So, we can always assume that $|y| \leq t(|x|)$.

Now we are ready to define class NP.

Definition 1.5

Complexity class NP is a set of all computational problems non-deterministically computable in polynomial time.

Problem 1.9 Prove that P $\subseteq$ NP.

Problem 1.10 Show that complexity classes P and NP would not change if in definitions we used RAM machines instead of Turing machines.

Problem 1.11 Show that SAT $\in$ NP.

Problem 1.12 Suppose that you are given an algorithm for decision SAT that works in time $t(n)$, where n is the number of variables. Show how to find a satisfying assignment in time $O(n \cdot t(n))$.

1.6 Communication complexity

1.6.1 Communication games for functions

Definition 1.6

Let X, Y, and Z be non-empty finite sets, and f be a function, $f\colon X \times Y \to Z$. Two players, Alice and Bob, are playing *communication game for* f if Alice is given $x \in X$, Bob is given $y \in Y$, and their goal is to compute $z = f(x,y)$. By communicating to each other one bit at a time, Alice and Bob exchange information in order to compute $f(x,y)$. The game ends when both players know $f(x,y)$. The minimal number of messages that is enough to compute f on any pair of inputs defines *communication complexity of* f, denoted $\mathrm{CC}(f)$. The players know f in advance and can agree on some *communication protocol for* f (i.e., communication scheme).

Remark

Every communication protocol for f must be unambiguous, meaning that at every moment of communication both players know who sends the next message, and also know is the communication is over. In particular, every communication protocol uniquely identifies the player who sends the first message. The players can not use clocks, timers or other physical devices.

Let $X = Y = \{0,1,2\}$, $Z = \{0,1,2,3,4\}$, and $f(x,y) = x + y$.

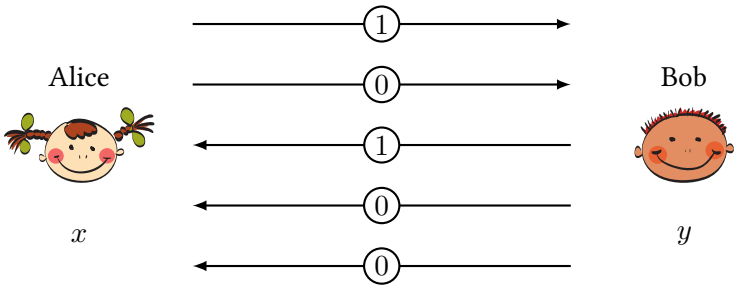


Figure 3: Communication of Alice and Bob.

Problem 1.13 Prove that $\mathrm{CC}(f) \leq 4$.

Problem 1.14 Prove that there is a protocol for f such that Alice and Bob send at most four bits, but for some pairs of inputs (x, y) the players send strictly less than four bits.

Problem 1.15 Prove that $\mathrm{CC}(f) = 4$.

Let $X = Y = \{0, \ldots, 2^k - 1\}$, $Z = \{0, \ldots, 2^{2k}\}$, and $f = x \cdot y$.

Problem 1.16 Prove that $\mathrm{CC}(f) \leq 2k$.

Problem 1.17 Let $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k$. Show that $\mathrm{CC}(f) \leq 2n$.

Problem 1.18 Let $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k$. Show that $\mathrm{CC}(f) \leq n + k$.

Problem 1.19 Let $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$, $f(x, y) = x$. Prove that for every protocol there is a pair of inputs (x, y) such that the players send at least n bits given x and y, respectively.

To prove lower bounds on the communication complexity we need to define the notion of communication protocol more formally.

Definition 1.8

Let X, Y, and Z be non-empty finite sets, and f be a function, $f\colon X \times Y \to Z$. *A communication protocol* Π for function f is an ordered rooted binary tree with the following labels and additional information.

- every internal node is labeled with either "A" or "B",
- every edge to a left child is labeled with "0", every edge to a right child is labeled with "1",
- every leaf is labeled with an element of Z.

For every internal node v labeled with "A" there is a function $A_v\colon X \to \{0,1\}$, and for every internal node u labeled with "B" there is a function $B_u\colon Y \to \{0,1\}$.

A value of the protocol Π on input (x, y), denoted $\Pi(x, y)$, is defined as a label of the final leaf of a root-to-leaf path $\pi(x, y)$, constructed using the following rules:

- first node of $\pi(x, y)$ is the root node,
- every next node of $\pi(x, y)$ is a child of the a previous one, moreover
  - every node v labeled with "A" is followed with a child connected by an edge with label $A_v(x)$,
  - every node u labeled with "B" is followed with a child connected by an edge with label $B_u(y)$,
- last node of $\pi(x, y)$ is a leaf.

The *depth* of the protocol is the depth of tree. The *size* of the protocol is the size of tree. The protocol Π is a *correct protocol* for f if for all $(x, y) \in X \times Y$, $\Pi(x, y) = f(x, y)$.

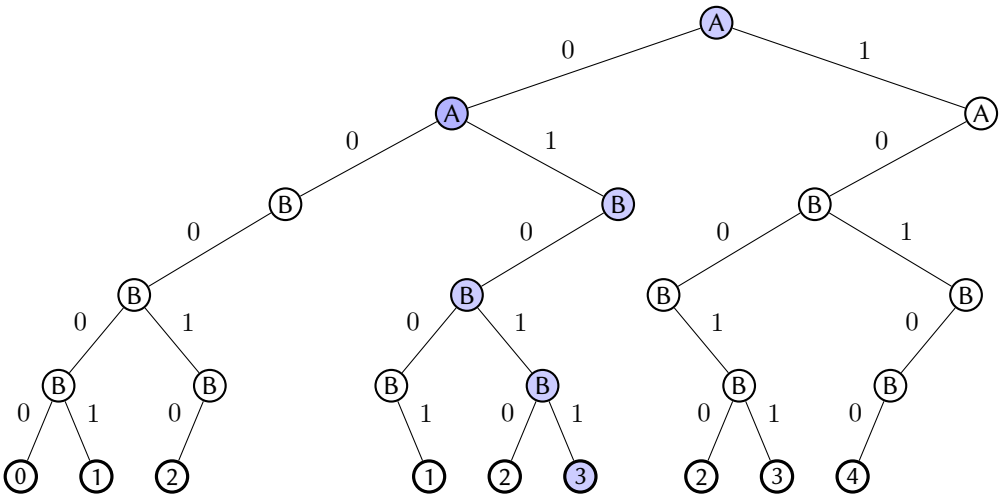


Figure 4: A simple communication protocol for $f\colon \{0,1,2\} \times \{0,1,2\} \to \{0,1,2,3,4\}$, such that $f(x, y) = x + y$: Alice sends x to Bob and Bob sends $x + y$ back. Blue nodes define $\pi(1, 2)$.

Definition 1.9

Communication complexity of function f is the depth of the shallowest communication protocol for f.

A communication protocol defines the communication of players on all pairs of inputs. Labels on internal nodes define who's turn is to send a message, labels on edges — the messages, labels on leafs — values of $f(x, y)$. The functions defined for internal nodes are the rules that the players use to choose which message to send. For every pair of inputs (x, y) there is a root-to-leaf path $\pi(x, y)$ defined by the rules above. The protocol is correct for f if for all pairs of inputs (x, y) the corresponding path $\pi(x, y)$ ends in a leaf labeled with $f(x, y)$.

Problem 1.20 Let Π be a communication protocol for $f\colon X \times Y \to Z$. Suppose that for two distinct pairs of inputs (x_1, y_1) and (x_2, y_2) the corresponding paths $\pi(x_1, y_1)$ and $\pi(x_2, y_2)$ end in the same leaf ℓ. Prove that $\pi(x_1, y_2)$ and $\pi(x_2, y_1)$ end in ℓ too.

Problem 1.21 Suppose that Eve is eavesdropping the communication of Alice and Bob. Show that if Eve knows the protocol then she can learn $f(x, y)$ even if she doesn't know x and y.

The equality function $\mathsf{EQ}_n\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is defined by

$$\mathsf{EQ}_n(x, y) = 1 \iff x = y.$$

Problem 1.22 Prove that $\mathrm{CC}(\mathsf{EQ}_n) = n + 1$.

The greater function $\mathsf{GT}_n\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ compares two n-bit integers:

$$\mathsf{GT}_n(x, y) = 1 \iff x > y.$$

Problem 1.23 Prove that $\mathrm{CC}(\mathsf{GT}_n) = n + 1$.

The disjointness function $\mathsf{DISJ}_n\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ checks that there is no $i \in [n]$ such that $x_i = y_i = 1$:

$$\mathsf{DISJ}_n(x, y) = 1 \iff \forall i \in [n], x_i \neq 1 \vee y_i \neq 1.$$

Problem 1.24 Prove that $\mathrm{CC}(\mathsf{DISJ}_n) \geq n$.

1.6.2 Communication games for relations

The communication game for functions can be generalized for relations.

Definition 1.10

Let X, Y, and Z be non-empty finite sets, and R be a relation, $R \subseteq X \times Y \times Z$. Two players, Alice and Bob, are playing *communication game for* R if Alice is given $x \in X$, Bob is given $y \in Y$, and their goal is to compute $z \in Z$ such that $(x, y, z) \in R$ (we assume that such z always exists). By communicating to each other one bit at a time, Alice and Bob exchange information in order to find z. The game ends when both players know the same z satisfying the condition. The minimal number of messages that is enough to find an appropriate z for any pair of inputs (x, y) defines *communication complexity of* R, denoted $\mathrm{CC}(R)$.

Definition 1.8 can be adapted for relations in a natural way.

1.6.3 Karchmer–Wigderson games

The seminal work of Karchmer and Wigderson [KW88] established a correspondence between De Morgan formulas for non-constant Boolean function f and communication protocols for the Karchmer–Wigderson game for f.

Definition 1.11

The Karchmer–Wigderson game for a function $f\colon \{0,1\}^n \to \{0,1\}$ is a particular case of a communication game for a relation KW_f:

$$\mathsf{KW}_f = \{(x,y,i) \mid x,y \in \{0,1\}^n, i \in [n], f(x) = 0, f(y) = 1, x_i \neq y_i\}.$$

A *monotone* version of the Karchmer–Wigderson game for a monotone function $f\colon \{0,1\}^n \to \{0,1\}$ is a particular case for a relation $\mathsf{KW}^{\mathsf{m}}_f$:

$$\mathsf{KW}^{\mathsf{m}}_f = \{(x,y,i) \mid x,y \in \{0,1\}^n, i \in [n], f(x) = 0, f(y) = 1, x_i = 0 \wedge y_i = 1\}.$$

In the Karchmer–Wigderson game for some function $f\colon \{0,1\}^n \to \{0,1\}$, Alice gets an input from preimage of 0, Bob gets an input from preimage of 1, and their goal is to find a position where their inputs differ. Note that such a position always exists. The monotone version is similar except that the goal is to find a position where Alice's bit is 0 and Bob's bit is 1.

Theorem 1.12 (Karchmer, Wigderson [KW88])

Let $f\colon \{0,1\}^n \to 1$ be a non-constant Boolean function.

- For every De Morgan formula ψ computing f there is a communication protocol Π for KW_f with the same underlying tree.
- For every communication protocol for KW_f there is a De Morgan formula ψ computing f with the same underlying tree.

This correspondence allows us to use communication complexity methods for proving lower bounds on formula depth and size.

Problem 1.25 Prove that $\mathrm{CC}(\mathsf{KW}_{\oplus_n}) \leq 2\lceil \log n \rceil$.

Problem 1.26 Prove that $\mathrm{CC}(\mathsf{KW}_{\vee_n}) = \lceil \log n \rceil$.

Problem 1.27 Let $n, k \in \mathbb{N}$ be such that k divides n. Suppose that we enforce the following restriction on the structure of the communication protocol for $\oplus_n$.

1. Alice sends first and she can send any numbers of bits.
2. Bob replies with any numbers of bits.
3. Alice sends at most $\lceil \log k \rceil$ bits.

Show that there is a communication protocol for $\oplus_n$ satisfying the restrictions with complexity at most $n/k + k + O(\log(n))$.

2 Circuit complexity

(We follow a recent excellent book by Jukna [Juk12] here. All the missing references can be found in the book.)

2.1 Lecture 1: Circuit lower bounds

2.1.1 Connection to algorithms

Theorem 2.1 (Pippenger, Fischer, 1979 [PF79])

For any Turing machine with running time $T(n)$ there exists a family of circuits $\{C_n\}_{n=1}^{\infty}$ for the same computational problem such that C_n has n inputs and size $O(T(n)\log T(n))$.

It follows that to prove that P $\neq$ NP it is enough to come up with a problem from NP of super-polynomial circuit size. (More formally, we would like to construct a family of functions $\{f_n\}_{n=1}^{\infty}$ such that $f_n \in B_n$, $\bigcup_{n=1}^{\infty} f_n^{-1}(1) \in$ NP and $\mathrm{C}(f) = \mathrm{superpoly}(n)$.)

At the same time, circuits is a *non-uniform* computational model meaning that to solve a given computational problem one constructs an infinite sequence of circuits (a separate circuit for each input length). Due to this, circuits are strictly more powerful than algorithms.

Problem 2.1 Construct an undecidable (that is, algorithmically unsolvable) language $L \subseteq \{0,1\}^*$ such that $\mathrm{C}(L) \leq n$.

2.1.2 Maximum complexity

Theorem 2.2 (Shannon, 1949 [Sha49])

For almost all $f \in B_n$,

$$\mathrm{C}(f) = \Omega(2^n/n)\,.$$

(That is, the fraction of such function goes to 1 with n going to infinity.)

Theorem 2.3 (Muller, 1956 [Mul56]; Lupanov, 1958 [Lup59])

For any $f \in B_n$,

$$\mathrm{C}(f) = O(2^n/n)\,.$$

Problem 2.2 Prove that for any symmetric $f \in B_n$,

$$\mathrm{C}(f) \leq 5n + o(n)\,.$$

Problem 2.3 Let $n \leq s(n) \leq o(2^n/n)$. Prove that circuits of size $s+n$ are strictly more powerful than circuits of size s: there exists $f \in B_n$ such that $s < \mathrm{C}(f) \leq s+n$.

2.1.3 Lower bounds

As a warm-up, we prove an exact bound on the circuit size of the parity function $\oplus_n$ over the basis $U_2 = B_2 \setminus \{\oplus, \equiv\}$.

Theorem 2.4 (Schnorr, 1976 [Sch76])

$$\mathrm{C}_{U_2}(\oplus_n) = 3(n-1)\,.$$

Proving a $3n$ lower bound for the basis B_2 is harder (than for the basis U_2) as one cannot kill a $\oplus$-gate by assigning a constant to one of its inputs. Interestingly, it is possible to overcome this difficulty by using a more complicated function. Namely, $f \in B_n$ is called an *affine disperser for dimension* d if it is not constant on any affine subspace of $\mathbb{F}_2^n$ of dimension at least d. Explicit constructions of affine dispersers for sublinear dimension (that is, $d = o(n)$) are known. In the proof below, we will use the following property of an affine disperser f: let $S \subseteq \{0,1\}^n$ be a set of points of a Boolean hypercube satisfying a (consistent) system of at most $n - d$ affine equations (that is, equations of the form $\bigoplus_{i \in I} x_i = c$), then, f is not constant on S.

Theorem 2.5 (Demenkov, Kulikov, 2011 [DK11])

Let $f \in B_n$ be an affine disperser for dimension $d = o(n)$. Then,

$$\mathsf{C}(f) \geq 3n - o(n)\,.$$

Problem 2.4 Prove that

$$\mathsf{C}(\mathsf{THR}_n^2) \geq 2n - O(1)\,,$$

where $\mathsf{THR}_n^2(x_1, \ldots, x_n) = [x_1 + \cdots + x_n \geq 2]$.

Problem 2.5 Prove that

$$\mathsf{C}(\mathsf{THR}_n^2) \leq 2n + o(n)\,.$$

Problem 2.6 Prove that $\mathsf{C}(M) \leq 2n + o(n)$, where $M \in B_{n+\log_2 n}$ is the multiplexer function.

2.1.4 Overview of known lower bounds

An overview of known lower bounds, together with its main ideas and functions used, can be found in [DK11, Section 3]. The strongest known lower bound is $3.1n - o(n)$ [JL21], the record lower bound for a symmetric function is $2.5n - o(n)$ [Sto77].

2.1.5 Research problems

- Prove a $3.5n$ lower bound for an explicit Boolean function!
- Is there an affine disperser for sublinear dimension of linear circuit size? (All other functions with non-trivial lower bounds on circuit size can be computed by circuits of linear size.)
- Prove a $2.6n$ lower bound for a symmetric Boolean function!
- Prove a $4.4n$ upper bound for a symmetric Boolean function!
- Close the following gap!

$$2n - O(1) \leq \mathsf{C}(\mathsf{AND}, \mathsf{OR}, \mathsf{XOR}) \leq 2.5n - O(1)$$

2.2 Lecture 2: Formula lower bounds

A *formula* is a circuit whose graph is a tree. In other words, the out-degree of any gate is equal to one: the value of any gate cannot be reused. It turns out that for formulas we do have superlinear lower bounds.

We will consider two types of formulas: formulas over the full binary basis B_2 and formulas over $\{\wedge, \vee, \neg\}$ also known as de Morgan formulas. The *size* of a formula is the number of leaves in it (it is

equal to the number of internal gates plus one and it is usually more convenient to work with). By $\mathsf{L}(f)$ and $\mathsf{D}(f)$ we denote the minimum size and depth, respectively, of a formula computing f.

It is not difficult to show that for any de Morgan formula there exists an equivalent formula (computing the same function) of the same size where negations are applied to input variables only (to do this, one uses de Morgan laws). Due to this, in the following by a de Morgan formula we mean a formula with fan-in AND and OR gates with literals (variables and their negations) at the leaves.

2.2.1 Size versus depth

$$\mathsf{D}(f) = \Theta(\log \mathsf{L}(f)) . \tag{1}$$

The lower bound $\mathsf{D}(f) \geq \log_2 \mathsf{L}(f)$ is immediate, so below we focus on an upper bound. Note that nothing like (1) is known for circuits: the best known upper bound on the depth in terms of circuit size is $\mathsf{D}(f) = O(\mathsf{C}(f)/\log \mathsf{C}(f))$.

Theorem 2.6 (Spira, 1971 [Spi71])

For any $f \in B_n$,

$$\mathsf{D}(f) \leq 3 \log_2 \mathsf{L}(f) .$$

The best known constant (that can be plugged instead of 3 above) is 1.73 due to Khrapchenko.

2.2.2 Full basis formulas

In this section, by $\mathsf{L}(f)$ we mean the minimum size of a formula *over the full binary basis* B_2 computing f.

Theorem 2.7 (Nechiporuk, 1966)

Let $n = 2^k$ and let $f(z, y) \in B_{2n}$ be the following function: partition $z \in \{0,1\}^n$ into $k = \log_2 n$ blocks (of length $n/\log_2 n$), apply the parity function to each of the blocks, and denote the resulting sequence of $\log_2 n$ bits by x; then, the output of $f(z, y)$ is $M(x, y)$. Then,

$$\mathsf{L}(f) \geq n^{2-o(1)} .$$

Theorem 2.8 (Nechiporuk, 1966)

Let $f \in B_n$ and let $Y_1, Y_2, \ldots, Y_m$ be disjoint subsets of variables. By s_i denote the number of different subfunctions that one gets from f by assigning constants to all variables but Y_i. Then,

$$\mathsf{L}(f) \geq \frac{1}{4} \sum_{i=1}^{m} \log s_i . \tag{2}$$

Problem 2.7 Let $n = 2m \log m$. The element distinctness function takes n bits and treats them as m integers $0 \leq a_1, \ldots, a_m < m^2$. It outputs 1 iff a_i's are pairwise different. Prove that

$$\mathsf{L}(\mathsf{ED}) \geq n^{2-o(1)} .$$

Problem 2.8 Prove that

$$\mathsf{L}(\mathsf{CLIQUE}_n^3) = \Omega(n^3) .$$

Problem 2.9 Prove that Theorem 2.8 cannot give superquadratic lower bound: (2) is at most $O(n^2/\log n)$.

Problem 2.10 Prove that

$$\mathsf{L}(\mathsf{MOD}_n^{3,r}) = O(n^2)\,.$$

2.2.3 De Morgan formulas

In this section, by $\mathsf{L}(f)$ we mean the minimum size of a *de Morgan* formula computing f.

Theorem 2.9 (Subbotovskaya, 1961)

$$\mathsf{L}(\oplus_n) = \Omega(n^{1.5})\,.$$

Theorem 2.10 (Khrapchenko, 1971)

$$\mathsf{L}(\oplus_n) = \Theta(n^2)\,.$$

Problem 2.11 Prove that

$$\mathsf{L}(\mathsf{THR}_n^k) \geq k(n-k+1)\,.$$

(In particular, $\mathsf{L}(\mathsf{MAJ}_n) = \Omega(n^2)$.)

Theorem 2.11 (Andreev, 1987)

Let $f \in B_{2n}$ be a function from Theorem 2.7. Then

$$\mathsf{L}(f) \geq n^{2.5-o(1)}\,.$$

Problem 2.12 Prove that for any symmetric $f \in B_n$ there exists a de Morgan formula of logarithmic depth:

$$\mathsf{D}(f) = O(\log n)\,.$$

(This is a challenging problem. One way to solve it is to design an efficient protocol for the corresponding Karchmer–Wigderson game, see Definition 1.11. Another way is to design a logarithmic depth circuit for SUM_n.)

2.2.4 Overview of known lower bounds

For the full binary basis, the quadratic lower bounds by Nechiporuk remain unbeaten for 55 years already! For de Morgan formulas, the strongest lower bound $n^{3-o(1)}$ was proved by Håstad in 1991 [Hås98a]. An exposition of known lower bounds is given in Jukna's book [Juk12, Chapter 6].

2.2.5 Research problems

- Prove an $n^{2+\varepsilon}$ lower bound for formulas over B_2!
- Prove an $\omega(n \log n)$ lower bound for formulas over B_2 for a symmetric function!
- Prove an $n^{3+\varepsilon}$ lower bound for de Morgan formulas!
- Prove an $n^{2+\varepsilon}$ lower bound for de Morgan formulas for a symmetric function!

2.3 Lecture 3: Depth three circuit lower bounds

2.3.1 Connections to unrestricted circuits

Theorem 2.12 (Valiant, 1977 [Val77])

For any $\varepsilon > 0$, there exists $\delta(\varepsilon)$ such that: if $f \in B_n$ can be computed by a circuit of size εn and depth $\varepsilon \log_2 n$, then it can also be computed by a depth 3 circuit of the following form:

$$\mathsf{OR}_{2^{\frac{\delta n}{\log\log n}}} \circ \mathsf{AND} \circ \mathsf{OR}_{\sqrt{n}}.$$

Lemma 2.13 (Erdős, Graham, Szemeredi, 1975 [EGS75])

Let $G(V, E)$ be a graph of depth 2^k. One can remove $|E|/k$ edges from G to reduce its depth to at most 2^{k-1}.

2.3.2 A simple lower bound

2.3.3 Overview of known lower bounds

2.3.4 Research problems

3 Formula complexity

3.1 Block composition and KRW conjecture

Karchmer, Raz, and Wigderson [KRW95] suggested an approach for proving superpolynomial formula size lower bound for Boolean functions from class P. The suggested approach is to prove lower bounds on the formula depth of *the block-composition* of two arbitrary Boolean functions.

Definition 3.1

Let $f\colon \{0,1\}^m \to \{0,1\}$ and $g\colon \{0,1\}^n \to \{0,1\}$ be Boolean functions. *The block-composition* $f \diamond g\colon (\{0,1\}^n)^m \to \{0,1\}$ is defined by

$$(f \diamond g)(x_1, \dots, x_m) = f(g(x_1), \dots, g(x_m)),$$

where $x_1, \dots, x_m \in \{0,1\}^n$.

Let $\mathsf{D}(f)$ denote the minimal depth of De Morgan formula for function f. It is easy to show that $\mathsf{D}(f \diamond g) \le \mathsf{D}(f) + \mathsf{D}(g)$ by constructing a formula for $f \diamond g$ by substituting every variable in a formula for f with a copy of the formula for g. Karchmer, Raz, and Wigderson [KRW95] conjectured that this upper bound is roughly optimal.

Conjecture 3.2 (The KRW conjecture)

Let $f\colon \{0,1\}^m \to \{0,1\}$ and $g\colon \{0,1\}^n \to \{0,1\}$ be non-constant functions. Then

$$\mathsf{D}(f \diamond g) \approx \mathsf{D}(f) + \mathsf{D}(g).$$

If the conjecture is true then there is a polynomial-time computable function that does not have De Morgan formula of polynomial size, and hence $\mathsf{P} \not\subseteq \mathsf{NC}^1$. Consider the function $h\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, which interprets its first input as a truth table of a function $f\colon \{0,1\}^{\log n} \to \{0,1\}$

and computes the value of the block-composition of $\log n/\log\log n$ functions f on its second input:

$$h(f,x) = (\underbrace{f \diamond \cdots \diamond f}_{\log n/\log\log n})(x).$$

It is not hard to see that $h \in \mathsf{P}$. To show that $h \notin \mathsf{NC}^1$, let $\tilde{f}$ be a function with maximal depth complexity. By Shannon's counting argument $\tilde{f}$ has depth complexity roughly $\log n$. Assuming the KRW conjecture, the function $\tilde{f} \diamond \cdots \diamond \tilde{f}$ has depth complexity roughly $\log n \cdot (\log n/\log\log n) = \omega(\log n)$, and hence $\tilde{f}\diamond\cdots\diamond\tilde{f} \notin \mathsf{NC}^1$. Any formula for h must compute $\tilde{f}\diamond\cdots\diamond\tilde{f}$ if we hard-wire $f = \tilde{f}$ in it, so $h \notin \mathsf{NC}^1$. This argument is especially attractive since it does not seem to break any known meta mathematical barriers such as the concept of "natural proofs" by Razborov and Rudich [RR97] (the function h is very special, so the argument does not satisfy "largeness" property).

It worth noting that the proof would work even assuming some weaker version of the KRW conjecture, like $\mathsf{D}(f \diamond g) \geq \mathsf{D}(f) + \varepsilon \cdot \mathsf{D}(g)$ or $\mathsf{D}(f \diamond g) \geq \varepsilon \cdot \mathsf{D}(f) + \mathsf{D}(g)$ for some $\varepsilon > 0$.

Problem 3.1 Show that the following versions of the KRW conjecture imply $\mathsf{P} \not\subseteq \mathsf{NC}^1$:

1. $\mathsf{D}(f \diamond g) \geq \mathsf{D}(f) + \varepsilon \cdot \mathsf{D}(g)$ for $\varepsilon > 0$,
2. $\mathsf{D}(f \diamond g) \geq \varepsilon \cdot \mathsf{D}(f) + \mathsf{D}(g)$ for $\varepsilon > 0$,
3. $\mathsf{L}(f \diamond g) \geq \mathsf{L}(f) \cdot \mathsf{L}(g)^{\varepsilon}$ for $\varepsilon > 0$,
4. $\mathsf{L}(f \diamond g) \geq \mathsf{L}(f)^{\varepsilon} \cdot \mathsf{L}(g)$ for $\varepsilon > 0$,
5. $\mathsf{D}(f \diamond g) \geq \varepsilon_1 \cdot \mathsf{D}(f) + \varepsilon_2 \cdot \mathsf{D}(g)$ for $\varepsilon_1, \varepsilon_2 > 0$ such that $\varepsilon_1 + \varepsilon_2 > 1$.

Problem 3.2 Prove that for any non-constant $f\colon \{0,1\}^n \to \{0,1\}$, $L(\vee_m \diamond f) = m \cdot \mathsf{L}(f)$.

3.2 KRW conjecture and communication complexity

Conjecture 3.2 can be reformulated in terms of communication complexity of the Karchmer–Wigderson game for the block-composition of two arbitrary Boolean functions. Let $\mathrm{CC}(R)$ denotes deterministic communication complexity of a relation R. For convenience, we also define *a block-composition for KW relations*, so that the following equality holds: $\mathsf{KW}_{f\diamond g} = \mathsf{KW}_f \diamond \mathsf{KW}_g$.

$$\mathsf{KW}_{f\diamond g} = \Big\{(X,Y,(i,j)) \mid X,Y \in \{0,1\}^{m\times n}, i \in [m], j \in [n], \\ f(g(X_1),\dots,g(X_m)) = 0, f(g(Y_1),\dots,g(Y_m)) = 1, X_{i,j} \neq Y_{i,j}.\Big\}$$

This leads to the following reformulation of the KRW conjecture.

Conjecture 3.3 (The KRW conjecture (reformulation))

Let $f\colon \{0,1\}^m \to \{0,1\}$ and $g\colon \{0,1\}^n \to \{0,1\}$ be non-constant functions. Then

$$\mathrm{CC}(\mathsf{KW}_f \diamond \mathsf{KW}_g) \approx \mathrm{CC}(\mathsf{KW}_f) + \mathrm{CC}(\mathsf{KW}_g).$$

The study of Karchmer–Wigderson games had already been shown to be a potent tool in the monotone setting — the monotone KW games were used to separate the monotone counterparts of classes NC^1 and NC^2 [KW88]. Therefore, there is a reason to believe that the communication complexity perspective might help to prove new lower bounds in the non-monotone setting.

3.3 Universal relations

In a series of works [Edm+01; HW90; Gav+17; DM18; KM18; Rez+20] several steps were taken towards proving the KRW conjecture. In the first two works [Edm+01; HW90] the authors proved the similar bound for the block-composition of two *universal relations*.

Definition 3.4

The universal relation of length n,

$$\mathsf{U}_n = \{(x, y, i) \mid x, y \in \{0,1\}^n, i \in [n], x_i \neq y_i\} \cup \{(x, x, \bot) \mid x \in \{0,1\}^n\}.$$

A communication problem for the universal relation is a generalization of the Karchmer–Wigderson games: Alice and Bob are given n-bit distinct strings and their goal is to find a coordinate $i \in [n]$ such that $x_i \neq y_i$. In contrast to KW games, in this game Alice and Bob can be given the same input string — in that case, they have to output a special symbol $\bot$ to indicate that the promise is broken. Intuitively, the universal relation is a more complex communication problem than KW game because the players do not have proof that their inputs are different.

Problem 3.3 Prove that $\mathrm{CC}(\mathsf{U}_n) \geq n + 1$.

Problem 3.4 Prove that $\mathrm{CC}(\mathsf{U}_n) \leq n + O(1)$.

Problem 3.5 Let $\mathsf{U}'_n = \{(x, y, i) \mid x, y \in \{0,1\}^n, i \in [n], x_i \neq y_i\}$. Show that $\mathrm{CC}(\mathsf{U}'_n) \leq \mathrm{CC}(\mathsf{U}_n) \leq \mathrm{CC}(\mathsf{U}'_n) + 2$.

For any non-constant $f\colon \{0,1\}^n \to \{0,1\}$, there is a natural reduction from KW_f to U_n: given inputs (x, y) for KW_f the players follow a protocol for U_n, the protocol outputs some i such that $x_i \neq y_i$, the players output i as it is a correct output for KW_f.

The block-composition of the universal relations generalizes the block-composition of KW games in the same manner.

Definition 3.5

The block-composition $\mathsf{U}_m \diamond \mathsf{U}_n$ of two universal relations is defined by

$$\begin{aligned}\mathsf{U}_m \diamond \mathsf{U}_n = &\left\{((a, X), (b, Y), (i, j)) \mid X_{i,j} \neq Y_{i,j}\right\} \\ &\cup \left\{((a, X), (b, Y), \bot) \mid \exists k \in [m]\colon a_k \neq b_k \wedge X_k = Y_k\right\} \\ &\cup \left\{((a, X), (b, Y), \bot) \mid a = b\right\},\end{aligned}$$

where $a, b \in \{0,1\}^m$, $X, Y \in \{0,1\}^{m \times n}$, $i \in [m]$, $j \in [n]$.

A similar reduction uses a protocol for the block-composition of the universal relations to solve the block-composition KW games. Thus, proving lower bounds for the universal relations seems to be a natural first step.

Theorem 3.6 (Proved in [Edm+01; HW90], improved in [KM18])

For any $n, m \in \mathbb{N}$,

$$\mathrm{CC}(\mathsf{U}_m \diamond \mathsf{U}_n) = m + n - O(\log m).$$

In particular, it is relatively easy to understand a special case of this result, proved in [HW90].

Theorem 3.7 (Proved in [HW90])

For any $n \in \mathbb{N}$,

$$\mathrm{CC}(\mathsf{U}_n \diamond \mathsf{U}_n) \geq 2n - 1.$$

In the subsequent works [Gav+17; KM18], the authors proved a lower bound on the block-composition of the Karchmer–Wigderson relation for an arbitrary function and the universal relation. This result is presented in terms of the number of leaves rather than formula depth.

Theorem 3.8 (Proved in [Gav+17], improved in [KM18])

For any $n, m \in \mathbb{N}$, and non-constant $f\colon \{0,1\}^m \to \{0,1\}$

$$\mathrm{CC}(f \diamond \mathsf{U}_n) = \log \mathsf{L}(f) + n - O(\log m).$$

Note that it is still not known whether the following conjecture is true.

Conjecture 3.9

For any $n, m \in \mathbb{N}$, and non-constant $f\colon \{0,1\}^m \to \{0,1\}$

$$\mathrm{CC}(f \diamond \mathsf{U}_n) = \mathrm{CC}(\mathsf{KW}_f) + n - O(\log m).$$

3.4 Latest results on KRW conjecture

In [DM18], the authors presented an alternative proof for the block-composition of an arbitrary function with the parity function in the framework of the Karchmer–Wigderson games (this result was originally proved in [Hås98b] using an entirely different approach). Their result gives an alternative proof of the cubic lower bound for Andreev's function [Hås98b].

Theorem 3.10 (Proved in [DM18])

Let $f\colon \{0,1\}^m \to \{0,1\}$ be a non-constant function. Then,

$$\mathsf{L}(f \diamond \oplus_n) \geq \frac{\mathsf{L}(f) \cdot \mathsf{L}(\oplus_n)}{2^{\tilde{O}(\sqrt{m + \log n})}}.$$

In the most recent paper [Rez+20] of the series, the authors extended the range of inner functions that can be handled in the monotone version of the KRW conjecture to all functions whose depth complexity can be lower bounded via query-to-communication lifting theorem. They also introduce an intermediate semi-monotone setting where only inner function is monotone and show a lower bound on the composition of the (non-monotone) universal relation with every monotone inner function for which a lower bound can be proved using a lifting theorem.

In the last section of [Edm+01], the authors introduced *the same function multiplexer communication game*, that is very similar to the Karchmer–Wigderson game for *the multiplexer function*.

Definition 3.11

The multiplexer function of size n is a function $\mathsf{M}_n\colon \{0,1\}^{2^n} \times \{0,1\}^n \to \{0,1\}$ with two arguments, such that $\mathsf{M}_n(f, x) = f_x$. It is convenient to interpret the string f as a truthtable of some function $f\colon \{0,1\}^n \to \{0,1\}$, so we can say that $\mathsf{M}_n(f, x) = f(x)$.

In the KW game for M_n, Alice gets a function $f\colon \{0,1\}^n \to \{0,1\}$ and $x \in \{0,1\}^n$, such that $f(x) = 0$, Bob gets a function $g\colon \{0,1\}^n \to \{0,1\}$ and $y \in \{0,1\}^n$, such that $g(y) = 1$. Their goal is to find a coordinate $i \in [2^n + n]$ such that $(f,x)_i \neq (g,y)_i$. The authors of [Edm+01] suggest to consider a version of this game where players are given the same function, i.e., $f = g$, so they only need to find the differing coordinate between x and y.

Definition 3.12

In *the same function multiplexer communication game (the multiplexer game)* MUX_n, Alice gets a function $f\colon \{0,1\}^n \to \{0,1\}$ and $x \in \{0,1\}^n$ such that $f(x) = 0$, Bob gets the same function $g\colon \{0,1\}^n \to \{0,1\}$ and $y \in \{0,1\}^n$ such that $g(y) = 1$. Their goal is to find a coordinate $i \in [n]$ such that $x_i \neq y_i$, or output $\perp$ if $f \neq g$ (if $x \neq y$ and $f \neq g$ then both outputs are possible).

The same function multiplexer communication game can be considered as a generalization of the Karchmer–Wigderson games for Boolean functions on n bits. Indeed, solving the KW game for any $g\colon \{0,1\}^n \to \{0,1\}$ can be reduced to the same function multiplexer game: Alice and Bob are given g and the corresponding x and y. Given that we already have a lower bound on $f \diamond \mathsf{U}_n$ [Gav+17; KM18], it looks natural to study the block-composition of the KW game for an arbitrary function and the same function multiplexer game. The detailed explanation how a lower bound on the block-composition of the KW game for an arbitrary function and the same function multiplexer might be used to separate P and NC^1, see [Mei20] for details (to the best of our knowledge, this result was independently proved by Russell Impagliazzo).

Remark 3.13

The KW game for M_n can also be considered as a generalization of KW games using the same reduction. On the other hand, it is unclear whether lower bounds on the block-composition with it implies any new results. Moreover, the following lower bound applies. Let $\mathrm{L}(f)$ denotes the minimal size of De Morgan formula computing f.

Theorem 3.14

For any $m, n \in \mathbb{N}$ with $n \geq 6 \log m$, and any non-constant function $f\colon \{0,1\}^m \to \{0,1\}$,

$$\mathrm{CC}(\mathsf{KW}_{f \diamond \mathsf{M}_n}) \geq \log L(f) + n - O(\log^* n).$$

4 Proof Complexity

4.1 Basic Definitions

Proof system for a language $L \subseteq \{0,1\}^*$ is a polynomial-time computable function $\Pi\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ such that:

- $x \in L$ iff there exists $y \in \{0,1\}^*$ such that $\Pi(x,y) = 1$;
- $x \notin L$ iff $\Pi(x,y) = 0$ for all $y \in \{0,1\}^*$.

We focus on UNSAT that is a language that consists of unsatifiable boolean formulas in CNF.

Resolution. The *width* of a clause C is the number of literals in C. A *CNF formula* is a conjunction of clauses and a *width-k CNF formula*, or simply a *k-CNF formula*, is a CNF formula where every clause

has width at most k. A *resolution derivation* from a CNF formula φ of a clause C is a sequence of clauses $(C_1, \dots, C_\tau)$ such that $C_\tau = C$ and, for each $i \in [\tau]$, C_i is either a clause of φ, or is some clause $C_j \vee D$ obtained by *weakening* a clause C_j, for some $j < i$, or is derived from C_j and $C_{j'}$, for some $j, j' < i$ by applying the *resolution rule*

$$\frac{B \vee x \qquad D \vee \neg x}{B \vee D}$$

where $C_j = B \vee x$, $C_{j'} = D \vee \neg x$, and $C_i = B \vee D$. The *size/length* of a resolution proof $(C_1, \dots, C_\tau)$ is τ and its *width* is the maximum width of any clause in the proof. A *resolution proof* (i.e. proof of unsatisfiability) of φ is a proof of the empty clause $\emptyset$ from it.

A resolution proof $(C_1, \dots, C_\tau)$ can also be viewed as a DAG, with nodes $[\tau]$ and, for all $i, j \in [\tau]$, a directed edge from j to i if C_j was used to derive C_i. If the DAG is a tree the proof is *tree-like*.

4.2 Proofs and Applications

Definition 4.1

An *unsatisfied clause search problem* Search_φ for an unsatisfiable CNF formula $\varphi := \bigwedge_{i \in I} C_i$ on n variables is defined as follows:

input: an n-variable assignment $z \in \{0,1\}^n$;

output: an element $i \in I$ such that clause C_i of φ is falsified by z.

Decision trees. Let $X := \{x_1, \dots, x_n\}$ be a set of propositional variables and $\mathcal{O}$ be a finite set. A *decision tree* is a tree. Every vertex of the tree is labeled by a variable from X, or by an element of the set $\mathcal{O}$ with respect to the following properties:

- if a vertex is labeled by $o \in \mathcal{O}$, then it is a leaf;
- if a vertex is labeled by a variable, then it has exactly two outgoing edges: one edge is labeled by 0 and the other one is labeled by 1.

Every decision tree T defines a function $f_T \colon \{0,1\}^n \to \mathcal{O}$. We assume that every input $z \in \{0,1\}^n$ induces a path from root to leaf in a natural way. If this path ends in a vertex with a label $o \in \mathcal{O}$ then we define $f_B(z) := o$.

We say that B is a *decision tree for the relation* $S \subseteq \{0,1\} \times \mathcal{O}$ iff f_B is consistent with S: namely if $f_B(z) = o$ then $(z, o) \in S$.

Theorem 4.2

Decision trees for Search_φ (aka DPLL algorithms for SAT problem) are equivalent to tree-like resolution proof of φ.

Problem 4.1 A *Horn clause* is a clause with at most one positive literal and a *Horn formula* is a CNF formulas that consists of Horn clauses. Let φ be a Horn formula of size m. Show that there is a decision tree for Search_φ of size $\mathsf{poly}(m)$.

Problem 4.2 Let $G := (V, E)$ be a directed acyclic graph. For each vertex $v \in V$ we introduce a propositional variable x_v and create a clause $(\neg x_v \vee \bigvee_{u:(u,v)\in E} x_u)$. Also we add unit clauses (x_w) for all sinks w. Show that there is a decision tree for the Search problem of size $\mathsf{poly}(|V|)$.

Problem 4.3 The Ordering Principle, denoted as OP, is a CNF formula defined on propositional variables $x_{u,v}$ for every two distinct $u, v \in [n]$, with the intended meaning that $x_{u,v}$ is true when u is smaller than v in the partial order. The clauses of OP are

$$\begin{aligned}
&\neg x_{u,v} \vee \neg x_{v,w} \vee x_{u,w} && \text{for any three distinct } u, v, w \in [n],\\
&x_{u,v} \vee x_{v,u} && \text{for any two distinct } u, v \in [n],\\
&\neg x_{u,v} \vee \neg x_{v,u} && \text{for any two distinct } u, v \in [n],\\
&\bigvee_{u:u\neq v} x_{u,v} && \text{for any } v \in [n].
\end{aligned}$$

Show that:

- any tree-like resolution proof of OP has size $2^{\Omega(n)}$;
- trere is a dag-like resolution proof of OP of size $\mathrm{poly}(n)$.

Monotone computations. Proof systems like resolution are weak enough and we can proof unconditional lower bounds for it. Surprisingly these lower bounds can be used for proving lower bounds on the much stronger computational models like monotone boolean formulas or monotone circuits.

At first we show a construction of a function that we associate with a CNF formula. Let start with some formula φ:

- $\varphi(x) := \bigvee C_j$;
- $g\colon \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}$ is some function that we call *gadget*;
- $\varphi \circ g := \varphi(g(y_1, z_1), g(y_2, z_2), g(y_3, z_3), \dots)$.

Let say that a formula ψ is a translation of the formula $\varphi \circ g$ into CNF, $\psi(y, z) := \bigvee_{j=1}^{m} D_j$. Let $Y := \{0,1\}^{nk}$ and $Z := \{0,1\}^{nk}$, define a partial monotone function $F_\psi\colon \{0,1\}^m \to \{0,1\}$ in the following way:

- with an assignment $a \in Y$ of y variables we associate a vector $w \in \{0,1\}^m$:
  - $w_i = 1$ iff there is an assignment $b \in Z$ to z variables such that $D_i(a_1, b_1, a_2, b_2, \dots) = 0$;

  and say that $F_\psi(w) := 1$.
- with an assignment $b \in Z$ of z variables we also associate a vector $w \in \{0,1\}^m$:
  - $w_i = 0$ iff there is an assignment $a \in Y$ to y variables such that $D_i(a_1, b_1, a_2, b_2, \dots) = 0$;

  and say that $F_\psi(w) := 0$.

In our examples we focus on $\mathsf{Ind}_m := \{0,1\}^{\log m} \times \{0,1\}^m \to \{0,1\}$ such that $\mathsf{Ind}_m(x, y) := y_x$.

Some example is given on fig. 5. Inner rectangles correspond to the sets of assignments that violate some clause D_i. Horizontal line is equivalent to some $a \in Y$ and corresponds to the first case, and vertical line is equivalent to some $b \in Z$ and corresponds to the second case.

Theorem 4.3 (Raz, McKenzie [RM99]; Göös, Pitassi, Watson [GPW15])

Resolution depth of φ is at least $d \Rightarrow \mathrm{CC}(\mathsf{Search}_{\varphi\circ\mathsf{Ind}_m}) \geq d \log n$, where $m := \mathrm{poly}(n)$.
In other words $\mathrm{CC}(\mathsf{Search}_{\varphi\circ\mathsf{Ind}_m}) \approx \mathrm{CC}(\mathsf{Ind}_m) \cdot \text{res-depth}(\varphi)$.

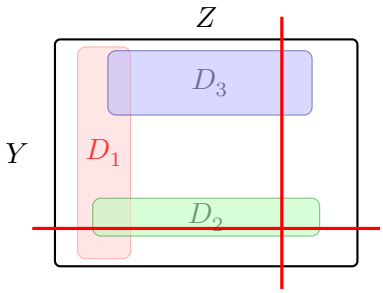


Figure 5: $F_\psi(1,1,0,\dots) := 1, \quad F_\psi(1,0,0,\dots) := 0$

Corollary: lower bound on monotone formulas 2^{n^ε}.

Problem 4.4 Show that $\mathrm{CC}(\mathsf{Search}_{\varphi\circ\mathsf{Ind}_m}) \le \mathrm{CC}(\mathsf{Ind}_m) \cdot \text{res-depth}(\varphi)$.

We can go a bit deeper and consider dag-like analog of communication games. However, we most intuitive definition is not extremely useful. But the characterization via "rectangles" give us the analog of the Karchmer–Wigderson result.

Definition 4.4 (Razborov [Raz95]; Pudlák [Pud10]; Sokolov [Sok17])

Let $X, Y \subseteq \{0,1\}^n$ be two sets. Let H be a directed acyclic graph and each vertex $v \in H$ is marked by some rectangle $R_v := X_v \times Y_v$. We say that it is *dag-like communication game* for a relation $S \subseteq X \times Y \times \mathcal{O}$ iff:

- inner nodes of H has out degree 2;
- $R_{\text{root}} = X \times Y$;
- if a, b are children of h then $R_h \subseteq R_a \cup R_b$;
- if h is a leaf then:
  - h is marked by some $o \in \mathcal{O}$;
  - $\forall (x,y) \in R_h$ the triple $(x,y,o) \in S$.

The *size* of the game is the size of the graph H.

Problem 4.5 Show that any classical communication protocol for some relation of size S is itself a dag-like communication protocol for the same relation.

Theorem 4.5 (Garg, Göös, Kamath, Sokolov [Gar+18])

Resolution size φ at least $S \Rightarrow$ size of *dag-like* protocols for $\mathsf{Search}_{\varphi\circ\mathsf{Ind}_m}$ at least $\Omega(S)$, where $m := \mathrm{poly}(n)$.

Corollary: lower bound on monotone circuits 2^{n^ε}.

Problem 4.6 Show that size of dag-like protocols for $\mathsf{Search}_{\varphi\circ\mathsf{Ind}_m}$ is at most size of resolution proof of $\varphi \circ \mathsf{Ind}_m$.

4.3 Tree-like Resolution Lower Bounds

Consider a game that is useful for proving tree-like lower bounds. There are two players: Prover and Delayer and unsatisfiable CNF formula φ. Delayer tries to convince Prover that formula is satisfiable, and Prover wants to catch the Prover.

Prover may ask a value of some variable x. A Delayer may choose an answer $\{0, 1\}$ or say "choose any" $(*)$ and in this case Prover chooses any preferred value. In the last case Prover give Delayer a coin. The game finishes if current partial assignment violates some clause of φ.

Theorem 4.6

If there is a tree-like resolution proof of φ of size s then Prover can wins the game with $\log s$ coins.

For proving lower bounds it will be enough to create the strategy for Delayer that helps him to get a lot of coins.

We start with the description of the hard formulas. The *Pigeonhole Principle* stating that there is no injective mapping of m pigeons into n holes if $m > n$ (PHP_n^m). This is one of the simplest, and yet most useful, combinatorial principles in mathematics, and it has been topic of extensive study in proof complexity. Since we wish to study unsatisfiable formulas, we encode the claim that there does in fact exist an injective mapping of pigeons to holes as a CNF formula consisting of *pigeon axioms*

$$P_i := \bigvee_{j \in [n]} x_{ij} \text{ for } i \in [m]$$

and *hole axioms*

$$H^k_{i,i'} := \neg x_{ij} \vee \neg x_{i'j} \text{ for } i, i' \in [m], j \in [n].$$

Let describe the strategy. In the beginning players have the empty partial assignment ρ and at each step they extend it for one variable. We say that hole j is *occupied* for a pigeon i iff $(x_{i,j} = 0) \in \rho$ or there is a pigeon k such that $(x_{k,j} = 1) \in \rho$. Suppose Prover asks a value of a variable $x_{i,j}$:

- if there is ℓ such that $x_{i,\ell} = 1$ then $\rho := \rho \cup \{x_{i,j} = 0\}$;
- if hole j is occupied for a pigeon i then $\rho := \rho \cup \{x_{i,\ell} = 0\}$;
- if there are more than $n/2$ holes are occupied for a pigeon i then $\rho := \rho \cup \{x_{i,\ell} = 1\}$;
- Prover may choose the value for $x_{i,j}$.

Theorem 4.7

In this strategy Delayer gets at least $n/2$ coins.

As a corollary we get the $2^{\Omega n}$ lower bound on the size of any tree-like resolution proof of PHP_n^m for any m.

Why is it not good enough?

1. It is only tree-like (modern sat solvers cannot be described by tree-like resolution).
2. the width of the formula is large (for the lower bounds on monotone computations we need formulas of small enough width).

4.4 Random Formulas

Definition 4.8

Let $\mathcal{F}(m, n, \Delta)$ denote the distribution of random Δ-CNF on n variables obtained by sampling m clauses (out of the $\binom{n}{\Delta}2^{\Delta}$ possible clauses) uniformly at random with replacement.

Lemma 4.9 (Chvál, Szemerédi [CS88])

For any $\Delta \geq 3$ whp $\varphi \sim \mathcal{F}(m, n, \Delta)$ is unsatisfiable if $m \geq \ln 2 \cdot 2^{\Delta} n$.

With a random formula φ we associate a bipartite dependency graph (L, R, E) in a natural way:

- L corresponds to clauses of φ;
- R corresponds to variables of φ;
- $(u, v) \in E$ iff a variable x_v appears in a clause C_u (maybe with a negation).

Theorem 4.10

Let Δ be big enough constant and $m = \mathcal{O}(n)$ then whp any resolution proof of $\varphi \sim \mathcal{F}(m, n, \Delta)$ should have size $\exp(n)$.

In fact $\Delta = 3$ and $m \ll n^{\Delta/2}$ is sufficient for this result.

The following Lemma gives us the key property that we want to use for proving lower bounds, it is a modification of well-known result for random graphs (see [Vad12]). For definitions see the next section.

Lemma 4.11

If $m = \mathcal{O}(n)$, $\Delta > 11$ and $\varphi \sim \mathcal{F}(m, n, \Delta)$ then whp G_{φ} is an $(r, \Delta, 5)$-boundary expander where $r = \Omega(\frac{n}{\Delta})$.

Problem 4.7 Let φ be a formula on n variables. Show that size of any resolution proof of $\varphi \circ \oplus_2$ is $2^{\Omega(w)}$ where w is the least resolution width of φ.

4.4.1 Expander Graphs

We use the following notation: $\mathrm{N}_G(S)$ is the set of neighbours of the set of vertices S in the graph G, $\partial_G(S)$ is the set of unique neighbours of the set of vertices S in the graph G. We omit the index G if the graph is evident from the context.

A bipartite graph $G := (L, R, E)$ is an (r, Δ, c)-*expander* if all vertices $u \in L$ have degree at most Δ and for all sets $S \subseteq L$, $|S| \leq r$, it holds that $|\mathrm{N}(S)| \geq c \cdot |S|$. Similarly, $G := (L, R, E)$ is an (r, Δ, c)-*boundary expander* if all vertices $u \in L$ have degree at most Δ and for all sets $S \subseteq L$, $|S| \leq r$, it holds that $|\partial S| \geq c \cdot |S|$. In this context, a simple but useful observation is that

$$|\mathrm{N}(S)| \leq |\partial S| + \frac{\Delta|S| - |\partial S|}{2} = \frac{\Delta|S| + |\partial S|}{2},$$

since all non-unique neighbours have at least two incident edges. This implies that if a graph G is an $(r, \Delta, (1-\varepsilon)\Delta)$-expander then it is also an $(r, \Delta, (1-2\varepsilon)\Delta)$-boundary expander.

Some useful properties of expander graphs are given in Appendix A. Also your can find there some proofs of the following lemmas.

Let $G := (L, R, E)$ denote a bipartite graph. Consider a *closure* operation that seems to have originated in [AR03; Ale+04].

Definition 4.12

For vertex sets $S \subseteq L, U \subseteq R$ we say that the set S is (U, r, ν)-*contained* if $|S| \leq r$ and $|\partial S \setminus U| < \nu|S|$. For any set $J \subseteq R$ let $S := \mathrm{Cl}^{r,\nu}(J)$ denote an arbitrary but fixed set of maximal size such that S is (J, r, ν)-contained.

Lemma 4.13

Suppose that G is an (r, Δ, c)-boundary expander and that $J \subseteq R$ has size $|J| \leq \Delta r$. Then $|\mathrm{Cl}^{r,\nu}(J)| < \frac{|J|}{c-\nu}$.

Suppose $J \subseteq R$ is not too large. Then Lemma 4.13 shows that the closure of J is not much larger. Thus, after removing the closure and its neighbourhood from the graph, we are still left with a decent expander. The following lemma makes this intuition precise.

Lemma 4.14

Let $J \subseteq R$ be such that $|J| \leq \Delta r$ and $|\mathrm{Cl}^{r,\nu}(J)| \leq \frac{r}{2}$ and let $G' := G \setminus (\mathrm{Cl}^{r,\nu}(J) \cup J \cup \mathrm{N}(\mathrm{Cl}^{r,\nu}(J)))$. Then any set S of vertices from the left side of G', with size $|S| \leq \frac{r}{2}$, satisfies that $|\partial_{G'} S| \geq \nu|S|$.

4.4.2 Delayer Strategy

Let describe the strategy.

Algorithm 1 $G := \{L, R, E\}$ is a dependency graph

$\rho := \emptyset$
while we can do **do**
    Prover query a variable $x \in R$
    If x is assigned by ρ then answer this value and go to next iteration
    Prover may chose the value x (wlog $x = b$)
    Remove x from R
    $B := \max\{S \subseteq L \mid |B| \leq r, |\partial_G(B)| \leq 3|B|\}$ ▷ Note: $3 < 5$
    Pick an assignment ν on $\mathrm{N}(B)$ that satisfy all constraints from the set B
    Remove B from L and $\mathrm{N}(B)$ from R
    $\rho := \rho \cup \{x = b\} \cup \nu$

Theorem 4.15

If dependency graph of φ is $(r, \Delta, 5)$-expander then Delayer gets at least $\Omega\left(\frac{r}{\Delta}\right)$ coins.

In fact this strategy give us the lower bound on the resolution width.

Theorem 4.16

If dependency graph of φ is $(r, \Delta, 5)$-expander then any resolution proof have width at least $\Omega\left(\frac{r}{\Delta}\right)$.

4.4.3 Restriction Argument

We start with the lower bound on the CNF representation of parity function on n inputs. For the sake of contradiction assume that we have CNF of size at most $\left(\frac{8}{7}\right)^{n/2}$.

1. In any CNF should be a clause that contains all n variables.
2. Hit our CNF by random restriction ρ:
   - choose $n/2$ variables with probability uniformly at random;
   - for each chosen variable choose a value uniformly at random.
3. Each clause of width $w \geq n/4$ will be set to 1 by ρ with probability at least:

$$\left(1 - \frac{w}{2n}\right)^{n/2} \geq \left(\frac{7}{8}\right)^{n/2}.$$

   Hence if CNF was small then there is an assignment that maps all clauses of width at least $n/4$ to 1.
4. After application of ρ parity is still parity on $n/2$ variables. And it is a contradiction.

For proving resolution lower bounds we want to use similar arguments. Let $\pi := (D_1, \dots, D_\ell)$ be a Resolution proof of some formula φ and H is a set of clauses of width at least w_0. For the sake of contradiction assume that π has small size and apply the following algorithm.

1. If π is small then H is small.
2. Pick the most frequent literal y in H. Note that it is contained in at least w_0/n fraction of clauses.
3. Set y to 1 in π. This operation set to 1 all clauses that contain y. Hence we can erase them from the proof.
4. After this assignment $\pi \restriction (y = 1)$ is still a proof of a restricted formula.
5. Repeat while we have clauses of large width.

If H is small we kill all clauses of large width in a few iterations. To achieve a contradiction we want to show that if we peek a "perfect" formula φ then even after a restriction described above any resolution proof should have width large than w_0.

Here is the algorithm that reduces a proof size into a proof of small width.

Algorithm 2 π is a resolution proof of our fomula, r, ε are a parameters

$O_1 := \emptyset$ ▷ Set of *active* input bits
$G_1 := G$ ▷ $G_i = (L_i, R_i, E_i)$
$i := 1$
$\rho_1 := \emptyset$
$H := \{$clauses in π of width at least $\varepsilon n\}$
while H is not empty **do**
    Pick the most frequent literal y. It correspond to a variable $x \in R_i$
    Pick an value b that maps y to 1
    $O_{i+1} := O_i \cup \{x\}$
    $G'_{i+1} := G_i \setminus \{x\}$
    $B_i := \max\{B \subseteq L'_{i+1} \mid |B| \leq r/2, |\partial_{G'_{i+1}}(B)| \leq 3|B|\}$ ▷ Note: $3 < 5$
    Pick an assignment ν_i on $\mathrm{N}_{G'_{i+1}}(B_i)$ that satisfy all constraints from the set B_i
    $G_{i+1} := G'_{i+1} \setminus (B_i \cup \mathrm{N}_{G'_{i+1}}(B_i))$
    $\rho_{i+1} := \rho_i \cup \{x = b\} \cup \nu_i$
    Hit all elements in H by ρ_{i+1}
    $i := i + 1$
return ρ_i

To conclude the lower bound on the size of resolution proofs we need to show that after application of ρ_i to a random formula whp it remains hard in terms of width.

Lemma 4.17 (Informal)

For all i the graph G_i is an $(r, \Delta, 3)$-expander.

And arguments that we used for tree-like will give us the desired width lower bound.

4.5 Polynomial Calculus Lower Bounds

A set I of polynomials is an *ideal* if I is closed under linear combination and multiplication by any polynomial from $\mathbb{F}[X]$. Given a set of polynomials $S := \{g_1, \dots, g_k\}$, the *ideal generated by* S is defined as the smallest ideal $I := \langle S \rangle$ that contains the set S. A *literal* is either a variable x or its negation $\bar{x}$. In PCR — and contrary to the classic boolean setting — the value 0 is interpreted as *true* and the value 1 as *false*. Note that we treat x and $\neg x$ as two distinct variables. We refer to set of variables by X — which contains both positive and negative variables — and we work over the polynomial ring $\mathbb{F}[X]$.

Polynomial Calculus. By analogy with Resolution we can define *Polynomial Calculus* ($\mathrm{PCR}^{\mathbb{F}}$) proof system. The $\mathrm{PCR}^{\mathbb{F}}$ proof system contains the following axioms:

- **boolean axioms**: $x^2 - x$ for all variables x;
- **complementary axioms**: $x + \bar{x} - 1$ for all variables x.

And the following derivation rules:

- **linear combination**: $\frac{p \quad q}{\alpha p + \beta q}$ for any $\alpha, \beta \in \mathbb{F}, p, q \in \mathbb{F}[X]$;
- **multiplication**: $\frac{p}{xp}$ for any $p \in \mathbb{F}[X]$.

The $\mathrm{PCR}^{\mathbb{F}}$-proof of unsatisfiability of a system of polynomial equations $\mathcal{F} := \{f_i = 0\}_{i=1}^m$ is a sequence of polynomials $(p_1, p_2, p_3, \dots, p_\ell)$ such that each element is either a polynomial from $\mathcal{F}$, an axiom, or obtained from previous by using the derivation rules, and $p_\ell = 1$.

Lets consider an example $\mathbb{F} := \mathbb{R}$:

$$\begin{cases} x+y+z-2=0 \\ xy=0 \\ xz=0 \\ yz=0 \end{cases} \qquad \dfrac{\dfrac{\dfrac{x+y+z-2}{x^2+xy+yz-2x} \quad \dfrac{xy \quad yz}{xy+yz}}{x^2-x \qquad x^2-2x} \qquad \dfrac{\vdots}{y+z}}{\dfrac{x \qquad\qquad\qquad\qquad y+z}{x+y+z} \qquad\qquad x+y+z-2}{1}$$

The *size* of a PCR refutation is the total number of non-zero monomials (counted with repetition) that appear in the derivation when all polynomials are expanded out as linear combinations of monomials. The *degree* of a PCR refutation is the maximal degree of a non-zero monomial that appears in the derivation.

4.5.1 Polynomials and Reduction Over Ideals

In this section we care only about values of polynomials on boolean cube, hence we consider only ideals that contain polynomials $x^2 - x$ for all $x \in X$.

Let I be an ideal. We say that set $V_I := \{a \in \mathbb{F}^n \mid \forall p \in I, p(a) = 0\}$ is a *variety* of ideal I. Note that since our ideals contain polynomials $x^2 - x$ then varieties of these ideals are subsets of boolean cube.

Problem 4.8 $p \in I$ iff $\forall a \in V_I, p(a) = 0$.

Two polynomials p, q are said to be *equivalent modulo an ideal* I, written $p \sim_I q$ if $p - q \in I$. This relation is an equivalence relation. For any polynomial p we fix a special representative of the equivalence class $[p]$ that we call the *reduction of* p *modulo* I and write as $R_I(p)$. If an ideal I is generated by a set of polynomials S, we abuse notation slightly and write $R_S(p)$ for $R_I(p)$.

To define the representative, we fix any order $\prec$ on the polynomials that respects inclusion:

1. monomial $m_1 \prec m_2$ whenever m_1 is a submonomial of m_2;
2. extend this to a total order on monomials arbitrarily;
3. finally, extend this order to polynomials by comparing the largest monomials (under $\prec$) that appear in the polynomials.

$R_I(p)$ is then defined as $\min(\{q \in [p]\})$.

Problem 4.9 $\forall a \in V_I, p(a) = (R_I(p))(a)$.

Problem 4.10 Check the following properties of R_I:

- $R_I(\alpha p + \beta q) = \alpha R_I(p) + \beta R_I(q)$, where $\alpha, \beta \in \mathbb{F}$;
- $R_I(xp) = R_I(xR_I(p))$, where x is a variable.

4.5.2 Lower Bounds on Polynomial Calculus and R Operator

The restriction part works pretty well also for Polynomial Calculus and we can reduce the question about size of the proof to a question about the degree of the proof. So we need some machinery for proving lower bounds on the degree.

Let $\mathcal{F}$ be a system of polynomial equations. Everything that we can derive in PCR lives in the ideal $\mathcal{F}$. Hence:

- $R_{\mathcal{F}}$ maps to 0 everything that we can derive in PCR,
- moreover if $\mathcal{F}$ is satisfiable then $R_{\mathcal{F}}(1) = 1$

that give us the certificate that there is no PCR proofs of $\mathcal{F}$ (of any degree). If $\mathcal{F}$ is unsatisfiable we want to mimic this operator up to degree d.

Theorem 4.18

Let $\mathcal{F}$ be a system of polynomial equations. For any d if there is a linear operator R such that:

- $R(f) = 0$ for all axioms f;
- $R(1) = 1$;
- $R(xt) = R(xR(t))$ for all terms t of degree at most $d-1$ and all variables x,

then there is no PCR proof of $\mathcal{F}$ of degree d.

4.5.3 Separation with Resolution

Consider an undirected bipartite graph $G := (U, V, E)$ where $|U| = n+1$ and $|V| = n$. The *Perfect Matching Principle* (aka *functional onto-Pigeonhole Principle*) PMP_G on variables $x_{u,v}$ where $(u,v) \in E$ consists of the following clauses:

- for all $w \in U \cup V$: $\bigvee_{u \in \mathrm{N}(w)} x_{w,u}$;
- for all $w \in U \cup V$ and all u, v such that $(w,u), (w,v) \in E$: $\neg x_{w,u} \vee x_{w,v}$.

Problem 4.11 Let G be a constant degree graph. Show that for any field $\mathbb{F}$ there is $\mathrm{PCR}^{\mathbb{F}}$-proof of PMP_G of size $\mathrm{poly}(n)$.

Harder version of previous problem.

Problem 4.12 Let G be an arbitrary graph. Show that for any field $\mathbb{F}$ there is $\mathrm{PCR}^{\mathbb{F}}$-proof of PMP_G of size $\mathrm{poly}(n)$.

The solution of the following problem is known in the literature, but maybe it is good problem to think about. The problem is tricky since it does not fit into presented framework.

Problem 4.13 Show that there is a graph G such that any resolution proof of PMP_G has size 2^{n^ε} for some ε (in fact $\varepsilon = 1$).

4.6 Research Problems

Disclaimer! The following research problems have some history of success and failures, consult with your advisor for the right choice.

Problem 4.14 *Weak* PHP *and resolution.* Let $m = 2^n$. What is the size of the optimal resolution proof of PHP_n^m?

Problem 4.15 *Weak* PHP *and* $\mathrm{PCR}^{\mathbb{F}}$. What is the size of the optimal $\mathrm{PCR}^{\mathbb{F}}$ proof of $\mathsf{PHP}_n^{n^3}$?

A *linear clause* is a disjunction of linear equalities $\bigvee_{i=1}^{k} (f_i = \alpha_i)$, where f_i is a linear form and $\alpha_i \in \mathbb{F}_2$. Equivalently we may rewrite a linear clause as the negation of a system of linear equalities $\neg \bigwedge_{i=1}^{n} (f_i = 1 + \alpha_i)$. A *linear CNF formula* is a conjunction of linear clauses. We say that propositional formula φ is *semantically implied* by the set of formulas $\psi_1, \psi_2, \ldots, \psi_k$ if every assignment that satisfies ψ_i for all $i \in [k]$ also satisfies φ.

We define a proof system Res($\oplus$) that can be used to prove that a linear CNF formula is unsatisfiable. This system has two rules:

- *the weakening rule*: allows to derive from a linear clause C any linear clause D such that C semantically implies D;
- *the resolution rule*: allows to derive from linear clauses $(f = 0) \vee D$ and $(f = 1) \vee D'$ the linear clause $D \vee D'$.

A derivation of a linear clause C from a linear CNF φ in the Res($\oplus$) system is a sequence of linear clauses that ends with C and every clause is either a clause of φ or it may be obtained from previous clauses by a derivation rule. The proof of the unsatisfiability of a linear CNF is a derivation of the empty clause (contradiction).

Problem 4.16 Res($\oplus$). Give any superpolynomial lower bound on the size of Res($\oplus$) proofs.

Consider some set of polynomial equations and inequalities $\mathcal{F} := \{p_1 = 0, p_2 = 0, \dots, p_m = 0; r_1 \geq 0, r_2 \geq 0, \dots, r_\ell > 0\}$ over the real field and over variables $x_1, \dots, x_n$, and we require that $\mathcal{F}$ include, for each $i \in [n]$, axioms $x_i^2 - x_i = 0$, $\bar{x}_i^2 - \bar{x}_i = 0$, $x_i + \bar{x}_i - 1 = 0$, and also the axiom $1 \geq 0$.

A Sum-of-squares (SOS) proof of $r \geq 0$ from $\mathcal{F}$ is a set of polynomials $\{q_1, \dots, q_m; s_1, \dots, s_\ell) \in \mathbb{R}[x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n]$ such that

$$\sum_{j \in [m]} q_j p_j + \sum_{j \in [\ell]} s_j r_j = r,$$

where each s_j is a positive linear combination of squared polynomials, that is, s_j can be written as $s_j = \sum_i \alpha_{j,i} q_{j,i}^2$ for some $\alpha_{j,i}$'s that are positive real numbers and $q_{j,i}$'s that are polynomials. Under the assumption that all polynomial equations and inequalities in $\mathcal{F}$ are satisfied, the summands $q_i p_j$ are equal to zero and the summands $s_j r_j$ are nonnegative; hence, $r \geq 0$.

Problem 4.17 *Non-automatizability.* Let $\mathcal{F}$ encodes a satisfiability of some CNF formula φ. How hard to algorithmically find the shortest (or approximately shortest) SOS proof of this system?

Problem 4.18 SOS *size/degree.* Is there any CNF formula such that has a short SOS proof, but any SOS proof of it has a large degree?

Problem 4.19 *Dag-like communication.* Show lower bounds for dag-like protocols for KW^{m} where instead of rectangles we can use more complicated structures.

5 Fine-grained complexity

5.1 Central problems and conjectures

Definition 5.1 (Fine-grained reduction)

(T_1, T_2) fine-grained reduction, where T_1, T_2 are function $N \to N$ from problem P to Q is an A algorithm that solves problem A given oracle to problem B such that:

- Time complexity of A on inputs of size n is $O(T_1(n)^{1-\alpha}$ for some α.
- For every $\delta > 0$ There exists an $\epsilon > 0$ such that for any computational path of A on inputs of size n sizes of the oracle calls $S_1, \dots, S_k$ satisfy the following condition:

$$\sum_{i=1}^{k} T_2(S_i)^{1-\delta} \leq T_1(n)^{1-\epsilon}$$

Definition 5.2 (3-SUM)

Given lists A, B, C of n integers. Is there $a \in A, b \in B, c \in C$ such that $a + b + c = 0$?

Definition 5.3 (All pairs shortest path)

Given a weighted graph G on n vertices. Return an $n \times n$ matrix M, such that $M[a, b]$ is equal to the weight of the lightest path from vertex a to b.

Definition 5.4 (k-SAT)

Given a k-CNF formula ϕ on n variables. Is there an assignment to the variables that satisfies ϕ?

Definition 5.5 (Orthogonal vectors problem)

Given two sets of size n of vectors $A, B \in \{0, 1\}^m$. Are there vectors $a \in A, b \in B$ such that $\sum_{i \in [m]} a_i b_i = 0$.

Conjecture 5.6 (3-SUM conjecture)

There is no $n^{2-\varepsilon}$ algorithm for 3-SUM for any ϵ.

Conjecture 5.7 (APSP conjecture)

There is no $n^{3-\varepsilon}$ algorithm for APSP for any ϵ.

Conjecture 5.8 (Exponential time hypothesis)

There is no algorithm that solves k-SAT in time $2^{o(n)}$ for any k.

Conjecture 5.9 (Strong exponential time hypothesis)

There is no $\varepsilon > 0$ such that k-SAT can be solved in time $2^{(1-\varepsilon)n}$ for any k.

Conjecture 5.10 (Nondeterministic strong exponential time hypothesis)

There is no $\varepsilon > 0$ such that k-SAT can be solved in time $2^{(1-\varepsilon)n}$ co-nondeterministically for any k.

Conjecture 5.11 (Orthogonal vectors conjecture)

There is no $n^{2-\varepsilon}$ algorithm for OV with $m = O(\log n)$ for any ϵ.

5.2 Problems and lower bounds

Definition 5.12 (3 points on a line)

Given n points on 2 dimensional plane. Is there a line that goes through 3 given points?

Definition 5.13 (Min-plus matrix product)

Given $n \times n$ matrices A, B. Return an $n \times n$ matrix C, such that:

$$\forall i, j \in [n]: C[i, j] = \min_{k \in [n]} A[i, k] + B[k, j]$$

Definition 5.14 (Negative-weight triangle)

Given a weighted graph G. Is there a triangle with negative weight.

Definition 5.15 (Zero-weight triangle)

Given a weighted graph G. Is there a triangle with weight exactly 0?

Definition 5.16 (k-Dominating set)

Given a graph G. Is there a dominating set of size k?

Definition 5.17 (3-coloring)

Given a graph G. Is chromatic number of G at most 3?

5.3 Lower bounds

In the following table we have a list of problems with a time complexity and list of conjectures that would be false if algorithm with such complexity existed for such problem:

3-SUM	$n^{2-\varepsilon}$	3-SUM conj.
APSP	$n^{3-\varepsilon}$	APSP conj.
k-SAT	$2^{(1-\varepsilon)n}$	SETH
k-SAT	$2^{o(n)}$	ETH
OV	$n^{2-\varepsilon}$	SETH, OV conj.
Zero weight triangle	$n^{3-\varepsilon}$	3-SUM, APSP
3-points on a line	$n^{2-\varepsilon}$	3-SUM
k-Dominating set	$n^{k-\varepsilon}$	SETH
3-Coloring	$2^{o(n)}$	ETH

5.4 Other theorems

Theorem 5.18

If there is $(2^n, n^2)$ fine-grained reduction from k-SAT to 3-SUM then NSETH is false.

Theorem 5.19

If there is $(2^n, n^3)$ fine-grained reduction from k-SAT to APSP then NSETH is false.

Theorem 5.20

If ETH is false then E^{NP} is not computable by linear size circuits.

5.5 Problems

Problem 5.1 Show that if 3-SUM, where all integers are bounded by n^4 can be solved by a randomized algorithm with complexity $n^{2-\varepsilon}$, then 3-SUM also can be solved in time $n^{2-\varepsilon+o(1)}$.

Problem 5.2 Show that Min-plus product of matrices where all the integers in matrices are non-negative and no more than t could be solved in time $O(t^2 n^\omega)$.

Problem 5.3 Show that Min-plus product of matrices where all the integers in matrices are non-negative and no more than t could be solved in time $O(tn^\omega)$.

Problem 5.4 Prove that Zero-weight triangle does not have an $n^{3-\varepsilon}$ algorithm under 3-SUM conjecture by providing a fine-grained reduction from Convolution 3-SUM to Zero-weight triangle.

Problem 5.5 Show that k-Domination sets can be solved in time $n^{k-1+o(1)}$ on graphs with average degree $d = o(n)$. Show that under SETH there is no algorithm for this problem with time complexity $n^{k-1-\varepsilon}$.

Problem 5.6 Show that under NSETH there is no $2^n, n^2$ fine-grained reduction from k-SAT to Max flow.

Problem 5.7 Show that under NSETH there is no $(2^n, n^3)$ fine-grained reduction from k-SAT to Min-plus matrix product.

Problem 5.8 Show that if there is a co-nondetermenistic algorithm for k-SAT for all k with running time $2^{o(n)}$ then E^{NP} is not computable by linear size circuits.

References

[Ale+04] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. "Pseudorandom Generators in Propositional Proof Complexity." In: *SIAM J. Comput.* 34.1 (2004), pp. 67–88. DOI: 10.1137/S0097539701389944. URL: https://doi.org/10.1137/S0097539701389944.

[AR03] Michael Alekhnovich and Alexander A. Razborov. "Lower Bounds for Polynomial Calculus: Non-Binomial Case." In: *Proceedings of the Steklov Institute of Mathematics* 242 (2003). Available at http://people.cs.uchicago.edu/~razborov/files/misha.pdf. Preliminary version in *FOCS '01*., pp. 18–35.

[CS88] Vašek Chvátal and Endre Szemerédi. "Many Hard Examples for Resolution." In: *J. ACM* 35.4 (Oct. 1988), pp. 759–768. ISSN: 0004-5411. DOI: 10.1145/48014.48016. URL: http://doi.acm.org/10.1145/48014.48016.

[DK11] Evgeny Demenkov and Alexander S. Kulikov. "An Elementary Proof of a 3n - o(n) Lower Bound on the Circuit Complexity of Affine Dispersers." In: *Mathematical Foundations of Computer Science 2011 - 36th International Symposium, MFCS 2011, Warsaw, Poland, August 22-26, 2011. Proceedings*. Ed. by Filip Murlak and Piotr Sankowski. Vol. 6907. Lecture Notes in Computer Science. Springer, 2011, pp. 256–265. DOI: 10.1007/978-3-642-22993-0_25. URL: https://doi.org/10.1007/978-3-642-22993-0_25.

[DM18] Irit Dinur and Or Meir. "Toward the KRW Composition Conjecture: Cubic Formula Lower Bounds via Communication Complexity." In: *Comput. Complex.* 27.3 (2018), pp. 375–462. DOI: 10.1007/s00037-017-0159-x. URL: https://doi.org/10.1007/s00037-017-0159-x.

[Edm+01] Jeff Edmonds, Russell Impagliazzo, Steven Rudich, and Jirí Sgall. "Communication complexity towards lower bounds on circuit depth." In: *Comput. Complex.* 10.3 (2001), pp. 210–246. DOI: 10.1007/s00037-001-8195-x. URL: https://doi.org/10.1007/s00037-001-8195-x.

[EGS75] Paul Erdős, Ronald L. Graham, and Endre Szemeredi. *On Sparse Graphs with Dense Long Paths*. Tech. rep. Stanford, CA, USA, 1975.

[Gar+18] Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. "Monotone Circuit Lower Bounds from Resolution." In: *Proceedings of the 50th Symposium on Theory of Computing (STOC)*. ACM, 2018, pp. 902–911. DOI: 10.1145/3188745.3188838.

[Gav+17] Dmitry Gavinsky, Or Meir, Omri Weinstein, and Avi Wigderson. "Toward Better Formula Lower Bounds: The Composition of a Function and a Universal Relation." In: *SIAM J. Comput.* 46.1 (2017), pp. 114–131. DOI: 10.1137/15M1018319. URL: https://doi.org/10.1137/15M1018319.

[GMT09] Konstantinos Georgiou, Avner Magen, and Madhur Tulsiani. "Optimal Sherali-Adams Gaps from Pairwise Independence." In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Ed. by Irit Dinur, Klaus Jansen, Joseph Naor, and José Rolim. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 125–139. ISBN: 978-3-642-03685-9.

[GPW15] Mika Göös, Toniann Pitassi, and Thomas Watson. "Deterministic Communication vs. Partition Number." In: *Proceedings of the 56th Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2015, pp. 1077–1088. DOI: 10.1109/FOCS.2015.70.

[Hås98a] Johan Håstad. "The Shrinkage Exponent of de Morgan Formulas is 2." In: *SIAM J. Comput.* 27.1 (1998), pp. 48–64. DOI: 10.1137/S0097539794261556. URL: https://doi.org/10.1137/S0097539794261556.

[Hås98b] Johan Håstad. "The Shrinkage Exponent of de Morgan Formulas is 2." In: *SIAM J. Comput.* 27.1 (1998), pp. 48–64. DOI: 10.1137/S0097539794261556. URL: https://doi.org/10.1137/S0097539794261556.

[HW90] Johan Håstad and Avi Wigderson. "Composition of the Universal Relation." In: *Advances In Computational Complexity Theory, Proceedings of a DIMACS Workshop, New Jersey, USA, December 3-7, 1990*. Ed. by Jin-Yi Cai. Vol. 13. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. DIMACS/AMS, 1990, pp. 119–134. URL: http://dimacs.rutgers.edu/Volumes/Vol13.html.

[JL21] Tianqi Yang Jiatu Li. $3.1n - o(n)$ *Circuit Lower Bounds for Explicit Functions*. TR21-023. ECCC, 2021. URL: https://eccc.weizmann.ac.il/report/2021/023/.

[Juk12] Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*. Vol. 27. Algorithms and combinatorics. Springer, 2012. ISBN: 978-3-642-24507-7. DOI: 10.1007/978-3-642-24508-4. URL: https://doi.org/10.1007/978-3-642-24508-4.

[KM18] Sajin Koroth and Or Meir. "Improved Composition Theorems for Functions and Relations." In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018)*. Ed. by Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer. Vol. 116. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, 48:1–48:18. ISBN: 978-3-95977-085-9. DOI: 10.4230/LIPIcs.APPROX-RANDOM.2018.48. URL: http://drops.dagstuhl.de/opus/volltexte/2018/9452.

[KRW95] Mauricio Karchmer, Ran Raz, and Avi Wigderson. "Super-Logarithmic Depth Lower Bounds Via the Direct Sum in Communication Complexity." In: *Computational Complexity* 5.3/4 (1995), pp. 191–204. DOI: 10.1007/BF01206317. URL: https://doi.org/10.1007/BF01206317.

[KW88] Mauricio Karchmer and Avi Wigderson. "Monotone Circuits for Connectivity Require Super-logarithmic Depth." In: *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*. Ed. by Janos Simon. ACM, 1988, pp. 539–550. ISBN: 0-89791-264-0. DOI: 10.1145/62212.62265. URL: http://doi.acm.org/10.1145/62212.62265.

[Lup59] Oleg Lupanov. "A method of circuit synthesis." In: *Izvestiya VUZov, Radiofizika* 1 (1959), pp. 120–140.

[Mei20] Or Meir. "Toward Better Depth Lower Bounds: Two Results on the Multiplexor Relation." In: *Comput. Complex.* 29.1 (2020), p. 4. DOI: 10.1007/s00037-020-00194-8. URL: https://doi.org/10.1007/s00037-020-00194-8.

[Mul56] David E. Muller. "Complexity in Electronic Switching Circuits." In: *IRE Transactions on Electronic Computers* EC-5 (1956), pp. 15–19.

[PF79] Nicholas Pippenger and Michael J. Fischer. "Relations Among Complexity Measures." In: *J. ACM* 26.2 (1979), pp. 361–381. DOI: 10.1145/322123.322138. URL: https://doi.org/10.1145/322123.322138.

[Pud10] Pavel Pudlák. "On extracting computations from propositional proofs (a survey)." In: *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010, December 15-18, 2010, Chennai, India*. 2010, pp. 30–41. DOI: 10.4230/LIPIcs.FSTTCS.2010.30. URL: http://dx.doi.org/10.4230/LIPIcs.FSTTCS.2010.30.

[Raz95] A. A. Razborov. "Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic." In: *Izvestiya RAN. Ser. Mat.* (1 1995), pp. 201–224.

[Rez+20] Susanna F. de Rezende, Or Meir, Jakob Nordström, Toniann Pitassi, and Robert Robere. "KRW Composition Theorems via Lifting." In: *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*. IEEE, 2020, pp. 43–49. DOI: 10.1109/FOCS46700.2020.00013. URL: https://doi.org/10.1109/FOCS46700.2020.00013.

[RM99] Ran Raz and Pierre McKenzie. "Separation of the Monotone NC Hierarchy." In: *Combinatorica* 19.3 (1999), pp. 403–435. DOI: 10.1007/s004930050062.

[RR97] Alexander A. Razborov and Steven Rudich. "Natural proofs." In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 24–35.

[Sch76] Claus-Peter Schnorr. "The Combinational Complexity of Equivalence." In: *Theor. Comput. Sci.* 1.4 (1976), pp. 289–295. DOI: 10.1016/0304-3975(76)90073-6. URL: http://dx.doi.org/10.1016/0304-3975(76)90073-6.

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

[Sok17] Dmitry Sokolov. "Dag-Like Communication and Its Applications." In: *Proceedings of the 12th Computer Science Symposium in Russia (CSR)*. Springer, 2017, pp. 294–307. ISBN: 978-3-319-58747-9. DOI: 10.1007/978-3-319-58747-9_26.

[Spi71] P. M. Spira. "On time-hardware complexity tradeoffs for Boolean functions." In: *Proc. 4th Hawaii Symp. on System Sciences*. 1971, pp. 525–527.

[Sto77] Larry J. Stockmeyer. "On the Combinational Complexity of Certain Symmetric Boolean Functions." In: *Math. Syst. Theory* 10 (1977), pp. 323–336. DOI: 10.1007/BF01683282. URL: https://doi.org/10.1007/BF01683282.

[Vad12] Salil P. Vadhan. *Pseudorandomness*. Hanover, MA, USA: Now Publishers Inc., 2012. ISBN: 1601985940, 9781601985941.

[Val77] Leslie G. Valiant. "Graph-Theoretic Arguments in Low-Level Complexity." In: *Mathematical Foundations of Computer Science 1977, 6th Symposium, Tatranska Lomnica, Czechoslovakia, September 5-9, 1977, Proceedings*. Ed. by Jozef Gruska. Vol. 53. Lecture Notes in Computer Science. Springer, 1977, pp. 162–176. DOI: 10.1007/3-540-08353-7_135. URL: https://doi.org/10.1007/3-540-08353-7_135.

A More on Expanders

The next proposition is well known in the literature. In this form it was used in [GMT09].

Proposition A.1

If $G := (L, R, E)$ is an (r, Δ, c)-boundary expander then for any set $S \subseteq L$ of size $k \leq r$ there is an enumeration $v_1, v_2, \ldots, v_k \in S$ and a sequence $R_1, \ldots, R_k \subseteq \mathrm{N}(S)$ such that:

- $R_i = \mathrm{N}(v_i) \setminus \left(\bigcup_{j=1}^{i-1} \mathrm{N}(v_j) \right)$;
- $|R_i| \geq c$.

In particular, there is a matching on the set S.

Proof. We create this sequence in reversed order. Since $|S| \leq r$ it holds that $|\partial S| \geq c|S|$ and there is a vertex $v_k \in S$ such that $|\partial S \cap \mathrm{N}(v_k)| \geq c$. Let $R_k := |\partial S \cap \mathrm{N}(v_k)|$, and repeat the process on $S \setminus \{v_k\}$. □

Lemma A.2

Suppose that G is an (r, Δ, c)-boundary expander and that $J \subseteq R$ has size $|J| \leq \Delta r$. Then $|\mathrm{Cl}^{r,\nu}(J)| < \frac{|J|}{c-\nu}$.

Proof. By definition we have that $|\partial \mathrm{Cl}^{r,\nu}(J) \setminus J| < \nu |\mathrm{Cl}^{r,\nu}(J)|$. Since $|\mathrm{Cl}^{r,\nu}(J)| \leq r$ by definition, the expansion property of the graph guarantees that $c|\mathrm{Cl}^{r,\nu}(J)| - |J| \leq |\partial \mathrm{Cl}^{r,\nu}(J) \setminus J|$. The conclusion follows. □

Lemma A.3

Let $J \subseteq R$ be such that $|J| \leq \Delta r$ and $|\mathrm{Cl}^{r,\nu}(J)| \leq \frac{r}{2}$ and let $G' := G \setminus (\mathrm{Cl}^{r,\nu}(J) \cup J \cup \mathrm{N}(\mathrm{Cl}^{r,\nu}(J)))$. Then any set S of vertices from the left side of G', with size $|S| \leq \frac{r}{2}$, satisfies that $|\partial_{G'} S| \geq \nu |S|$.

Proof. Suppose the set $S \subseteq L(G')$ violates the boundary expansion guarantee. Observe that $\mathrm{Cl}^{r,\nu}(J)$ and S are both sets of size at most $\frac{r}{2}$. Furthermore, the set $(\mathrm{Cl}^{r,\nu}(J) \cup S)$ is (J, r, ν)-contained in the graph G. As $\mathrm{Cl}^{r,\nu}(J)$ is a (J, r, ν)-contained set of maximal cardinality, this leads to a contradiction. □