6. From monotone to non-monotone complexity

Razborov has proved an $n^{\Omega(\log n)}$ lower bound for the monotone complexity of the perfect matching function. Since a maximum matching of a graph can be computed in polynomial time, an Ω_0-network of polynomial size for the perfect matching function exists. Hence, the gap between monotone and non-monotone network complexity is at least $n^{\Omega(\log n)}$. In 1986, Éva Tardos has shown that this gap is indeed exponential. Hence, it is not always possible to obtain large lower bounds for the Ω_0-complexity from a large lower bound for the Ω_m-complexity of a monotone Boolean function. But this does not exclude the possibility that techniques developed for the proof of lower bounds for the monotone complexity can also be useful for the proof of lower bounds for the Ω_0-complexity of Boolean functions.

The first step will be the simplification of an arbitrary Ω_0-network without increasing its size more than a small constant factor. Given any Ω_0-network β, we can convert β to an equivalent Ω_0-network β' where all negations occur only at the input nodes. Moreover, $|\beta'| \leq 2|\beta|$. For doing this, we start at the output nodes and apply de Morgan rules for bringing the negations

to the input nodes. Since gates can be simultaneously negated and not negated, some gates have to be doubled. The resulting network is the so-called standard network where only input variables are negated.

Exercise:
Work out the transformation of an Ω_0-network into a standard network.

The standard complexity of a function $f \in B_{n,m}$ is the size of a smallest standard network which computes f. Note that the standard and the Ω_0-complexity of a function f differs at most by the factor two. Hence, for proving nonlinear lower bounds for the Ω_0-complexity of Boolean functions we can restrict us to the consideration of standard networks.

Next we shall extend the approximation method as done by Razborov. For doing this, we restrict us to standard networks.

30.06.

Given the two operations $\sqcup$ and $\sqcap$, we define the legitimate model $(S, \sqcup, \sqcap)$ to be the smallest subset of $P(\{0,1\}^n)$ such that

1) $A(0), A(1), A(x_1), \ldots, A(x_n), A(\neg x_1), \ldots, A(\neg x_n) \in S$
and
2) S is closed under the operations $\sqcup$ and $\sqcap$.

For $M, N \in S$ we do not need that $M \sqcap N \subseteq M \cap N$ or $M \sqcup N \subseteq M \cup N$. Hence, we define more generally

$$\delta^+_{\sqcup}(M,N) := (M \sqcup N) \setminus (M \cup N) \quad \text{and}$$
$$\delta^-_{\sqcup}(M,N) := (M \cup N) \setminus (M \sqcup N)$$

and

$$\delta^+_{\sqcap}(M,N) := (M \sqcap N) \setminus M \cap N \quad \text{and}$$
$$\delta^-_{\sqcap}(M,N) := (M \cap N) \setminus M \sqcap N.$$

For $f \in B_n$ and the legitimate model $(S, \sqcup, \sqcap)$, we define the <u>distance</u> $\rho(f,S)$ from f to S to be the minimal t such that there are $M \in S$ and triples $\langle op_1, M_1, N_1\rangle, \langle op_2, M_2, N_2\rangle, \ldots, \langle op_t, M_t, N_t\rangle$ with $op_i \in \{\sqcup, \sqcap\}$, $M_i, N_i \in S$, $1 \leq i \leq t$ such that

$$A(f) \subseteq M \cup \bigcup_{i=1}^{t} \delta^-_{op_i}(M_i, N_i) \quad \text{and}$$

$$M \subseteq A(f) \cup \bigcup_{i=1}^{t} \delta^+_{op_i}(M_i, N_i).$$

<u>Exercise:</u>
Prove that $\rho(f,S)$ is a lower bound for the standard complexity of f.

7. Formulas

An Ω-network such that each gate has outdegree at most one is called an Ω-formula. The formula size of a Boolean function $f \in B_{n,m}$ is the (network size +1) of the smallest Ω-formula computing f and is denoted by $L_\Omega(f)$.

Given an Ω-formula, we may also bound the input nodes by one by using copies of the input nodes. For functions in B_n, we obtain a tree where the root of the tree is the output node.

As shown at the beginning of the lecture, the network complexity $C_{\Omega_0}(f)$ corresponds to the sequential computation time for the function f. Similar relations can be derived between sequential space and the depth of a Boolean network. (see Ingo Wegeners book pp. 277 – 279). Hence, other major open problems of complexity theory like

$$DSPACE(\log n) \stackrel{?}{=} P$$

may be solved by proving a large lower bound for the depth needed to compute a function f in P.

The depth $D(\beta)$ of a network β is the length (number of gates) of a longest path from an input node to an output node in β.

① $D_\Omega(f)$ denotes the minimal depth of an Ω-network which computes the Boolean function f.
$D_{\Omega_0}(f)$ is the <u>depth complexity of f</u>.

As mentioned above, the proof of a lower bound for the depth complexity of a Boolean function f is an interesting problem. Next we shall show that to get a solution of this problem, it suffices to prove a lower bound for the formula size $L_{\Omega_0}(f)$ of f.

Given any Ω_0-formula β we can transform β into a standard formula β' where only input variables are negated as described above. Since the outdegree of each node in β is at most one, no node has to be doubled. Hence, the size and the depth of β' are the same as of β. Therefore we can restrict us to consider standard formulas for the derivation of lower bounds.

<u>Theorem 7.1</u>

For every Boolean function $f \in B_n$ there holds

$$\log L_{\Omega_0}(f) \leq D_{\Omega_0}(f) \leq 3 \cdot \log L_{\Omega_0}(f).$$

<u>Proof:</u>
Since a formula is a binary tree, any formula of depth d can have at most 2^d leaves. Hence,

$$D_{\Omega_0}(f) \geq \log L_{\Omega_0}(f).$$

For proving the second inequality, we can restrict us to the consideration of <u>monotone formulas</u> which are formulas over the basis $\Omega_m = \{\wedge, \vee\}$.

To see this let us consider any standard formula β for a function $f \in B_n$. If we replace in β each input node labeled by a negated variable $\neg x_i$, $i \in \{1, 2, \ldots, n\}$ by the new variable y_i then we obtain a monotone formula for a function $f' \in M_{2n}$. Obviously, $f(x) = f'(x, y)$. If we have any monotone formula for f' then we obtain a formula for f by replacing each variable y_i, $1 \leq i \leq n$ by the corresponding negated variable $\neg x_i$.

Hence, we only need to show that every monotone formula with n input nodes (i.e., n leaves) can be transformed into an equivalent monotone formula of depth at most $3 \cdot \log n$. We do this by induction of the number n of leaves

<u>$n = 2$:</u>

Since $2 < 3 \cdot \log 2 = 3$, the assertion is trivially true.

Let $n > 2$ and assume that the assertion holds true for all monotone formulas with fewer than n leaves.

<u>$n-1 \Rightarrow n$:</u>

Let β be any monotone formula with n leaves.

Starting in the output node of β, we visit always that direct predecessor of the current node which is root of the subformula with larger number of leaves until a node g is reached for which the following hold:

i) g is root of a subformula β' which contains $\geq \frac{m}{2}$ leaves.

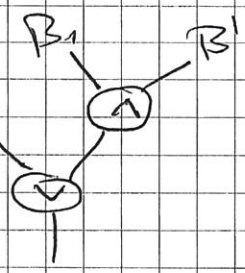
ii) The direct predecessors g_1 and g_2 of g are roots of subformulae which contain fewer than $\frac{m}{2}$ leaves.

Let β_0 and β_1 be those formulas which we obtain from β if we replace g by the constant 0 and 1, respectively.

Monotonicity $\Rightarrow$

$$\beta_1(a) = 0 \Rightarrow \beta_0(a) = 0.$$

Hence, the formulae

$\overline{\beta} =$ 

and β

are equivalent.

Induction hypothesis $\Rightarrow$

a) β_0 and β_1 can be replaced by equivalent

formulas $\tilde{\beta}_0$ and $\tilde{\beta}_1$ of depth $\leq 3 \cdot \log \frac{m}{2}$

b) β' can be replaced by an equivalent formula $\tilde{\beta}'$ of depth $\leq 1 + 3 \log \frac{m}{2}$

After these replacements, we get an equivalent formula $\tilde{\beta}$ of depth

$$\leq 2 + 1 + 3 \log \frac{m}{2}$$

$$= 3 \log m.$$

This proves the theorem. ∎

Our goal is now to prove lower bounds for the formula size of explicit defined Boolean functions.

7.1. Nechiporuk's method

Nechiporuk's method applies to the base B_2. Note that the number of leaves (input nodes) of a formula is equal one plus the size of the formula.

A <u>subfunction</u> of a Boolean function $f \in B_n$ on $Y \subseteq X_n$ is a function obtained from f by setting all the variables of $X_n \setminus Y$ to constants. Note that different settings can lead to different subfunctions.

The following theorem dues to Nechiporuk (1966).

Theorem 7.2

Let $f \in B_n$ and let $Y_1, Y_2, \ldots, Y_k$ be disjoint subsets of X_n. Let s_i be the number of distinct subfunctions of f on Y_i, $1 \le i \le k$. Then

$$L_{B_2}(f) \ge \frac{1}{4} \sum_{i=1}^{k} \log s_i .$$

Proof:

Let β be an optimal B_2-formula for f and let a_i be the number of leaves labeled by variables in Y_i, $1 \le i \le k$. It is sufficient to prove

$$a_i \ge \frac{1}{4} \log s_i .$$

Let T_i be that subtree of β consisting of all leaves labeled by some $x_j \in Y_i$ and consisting of all paths from these leaves to the output node of β.

Each node of T_i has in T_i indegree 0, 1 or 2. Let W_i be the set of nodes of indegree 2.

Since $|W_i| = a_i - 1$ it suffices to prove

$$|W_i| \ge \frac{1}{4}(\log s_i) - 1 .$$

Let P_i be the set of all paths in T_i starting from a leaf of T_i or a node in W_i and ending in a node in W_i or at the root of T_i and containing no node in W_i as inner node.

There are at most $|W_i| + 1$ different end-points of paths in P_i.

Construction $\Rightarrow$

At most two of these paths can end in the same node v.

These paths can be found by starting in v and going backwards until a node in W_i or a leaf is reached.

$\Rightarrow$

(*) $|P_i| \leq 2(|W_i| + 1)$.

Assignments to the variables in $X_n \setminus Y_i$ must lead to s_i different subformulas. Fix such an assignment α.

If we remove from β all gates that are evaluated to a constant 0 or 1 by α, then we obtain the tree T_i.

$\Rightarrow$

The subfunction $f(Y_i, \alpha)$ is computed by the gates of T_i whose fanin-1 gates correspond to fanin-2 gates of β with one of the input gates is replaced by a constant.

Consider any path $p \in P_i$. If h is the function computed at the first node of p (after the assignment α) then the function computed at the last

gate of P is $0, 1, h$ or $\neg h$.

$\Rightarrow$

Under different assignments at most

$$4^{|P_i|}$$

different subfunctions can be computed.

$\Rightarrow$

$$s_i \leq 4^{|P_i|}$$

Together with (*) this implies that

$$|W_i| \geq \frac{1}{4}(\log s_i) - 1.$$

This proves the theorem. ∎

Before we shall prove a lower bound for an explicit function using Nechiporuk's method, we shall answer the following question:

What is the largest possible lower bound which can be proved using Nechiporuk's method?

Theorem 7.3

Let $f \in B_n$. The best lower bound which can be proved using Nechiporuk's method is not larger than

$$2 \cdot \frac{n^2}{\log n}.$$

Proof:

Let $Y_1, Y_2, \dots, Y_k$ be any partition of X_n and let

$$t_i := |Y_i|.$$

- There are 2^{n-t_i} different assignments of the variables in $X_n \setminus Y_i$.
- There are at most $2^{2^{t_i}}$ different subfunctions on Y_i.

$$\Rightarrow \quad s_i \leq \min\{2^{n-t_i}, 2^{2^{t_i}}\}$$

$$\Leftrightarrow \log s_i \leq \min\{n-t_i, 2^{t_i}\}.$$

W.l.o.g assume that only the first q sets $Y_1, Y_2, \dots, Y_q$ have larger size than $\log n$.

Since the sets $Y_1, Y_2, \dots, Y_k$ are pairwise disjoint there holds

$$q < \frac{n}{\log n}.$$

Furthermore

i) $$\sum_{i \leq i \leq q} \log s_i \leq \sum_{1 \leq i \leq q} (n - t_i) \leq \frac{n^2}{\log n}$$

and

ii) $$\sum_{q < i \leq k} \log s_i \leq \sum_{q < i \leq k} 2^{t_i}$$

Exercise:
Show that $\sum_{q<i\leq k} 2^{t_i}$ is maximized if as many as possible t_i have the largest possible value $\log n$.

Exercise $\Rightarrow$ $\sum_{q<i\leq k} 2^{t_i} \leq \frac{n^2}{\log n}$. ∎

Next we shall consider the <u>indirect storage access function</u> $ISA_n \in B_n$ introduced by Wolfgang Paul in 1875. Using Nechiporuk's method, Paul has proved an $\Omega(\frac{n^2}{\log n})$ lower bound for ISA_n which is defined as follows:

Let

$n = k + 2p$ where $p = 2^{2^\ell}$, $\ell \in \mathbb{N}$ and $k := \log p - \log\log p$.

Partition the input variables into

$$\{\underbrace{a_1, a_2, \ldots, a_k}_{a}\}, \{\underbrace{x_1, x_2, \ldots, x_p}_{x}\}, \{\underbrace{y_1, y_2, \ldots, y_p}_{y}\}$$

Denote by a the string

$$a_1 a_2 \ldots a_k$$

and by b the string

$$x_{(a)\log p+1}\, x_{(a)\log p+2} \ldots x_{(a)+1)\log p}$$

where

(a) is the binary number represented by $a+1$.

Then we define

$$ISA_n(a, x, y) := y_{(b)}.$$

Theorem 7.4

$$L_{B_2}(ISA_n) = \Omega\left(\frac{n^2}{\log n}\right).$$

Proof:

For $0 \le i \le \frac{P}{\log P} - 1$ let

$$S_i := \{x_{i \log P + 1}, x_{i \log P + 2}, \ldots, x_{(i+1) \log P}\}.$$

For fixed i, we replace the variables in a by constants such that for the resulting string a'

$$(a') = i.$$

Then y is the value table of the subfunctions on S_i.

All 2^P different value tables lead to different subfunctions on S_i.

$$\Rightarrow \quad S_i \ge 2^P \quad \text{for} \quad 0 \le i \le \frac{P}{\log P} - 1.$$

Theorem 7.2 $\Rightarrow$

$$L_{B_2}(ISA_n) \ge \frac{1}{4}\left(\frac{P}{\log P} \cdot P\right) = \frac{1}{4} \frac{P^2}{\log P}$$

$$= \Omega\left(\frac{n^2}{\log n}\right).$$

□

Since there is a B_2-formula of size $O(\frac{n^2}{\log n})$, the lower bound is tight up to a constant factor (see: John Savage, Models of Computation pp. 404 – 405.)

Nechiporuk's method has been applied to some Boolean functions (see Wegener p. 256).

7.2 Krapchenko's method

Nechiporuk's method works for all (binary) bases. If the base is B_2 then the parity function

$$f(x) := x_1 \oplus x_2 \oplus \dots \oplus x_n$$

is a simple function.

For an input $a \in f^{-1}(c)$, $c \in \{0,1\}$, all neighbours; i.e., all inputs in which differ from a at exactly one position, are elements of $f^{-1}(\bar{c})$.

For the functions $g(x) = x_i$ computed at the input nodes, each vector $a \in g^{-1}(c)$ has exactly one neighbour in $g^{-1}(\bar{c})$.

As observed above, if the base is Ω_0 then we can restrict us to consider standard formulas. We shall present Krapchenko's method for standard formulas. We do not present the

proof due to Krapchenko but a simpler proof due to Mike Paterson. First, we need a definition.

A <u>formal complexity measure</u> μ is a function

$$\mu : B_n \to \mathbb{N}$$

such that

i) $\mu(x_i) = 1$ for $1 \leq i \leq n$,
ii) $\mu(f) = \mu(\neg f)$ for $f \in B_n$, and
iii) $\mu(f \vee g) \leq \mu(f) + \mu(g)$ for $f, g \in B_n$.

Note that by deMorgan's rules there holds

$$\mu(f \wedge g) \leq \mu(f) + \mu(g).$$

Furthermore, L_{Ω_0} is a formal complexity measure. Since $L_{\Omega_0}(f \vee g) = L_{\Omega_0}(f) + L_{\Omega_0}(g)$ it follows that L_{Ω_0} is the largest formal complexity measure. This is formally proved by the following Lemma.

<u>Lemma 7.1</u>

$L_{\Omega_0}(f) \geq \mu(f)$ for all $f \in B_n$ and all formal complexity measures μ.

<u>Proof</u>:

We prove the assertion by induction on $\ell = L_{\Omega_0}(f)$.

$\ell = 1$:

Then $f = x_i$ or $f = \overline{x}_i$. By the definition of formal complexity measure there holds also

$$\mu(f) = 1.$$

Let $\ell = L_{\Omega_0}(f) > 1$ and assume that the assertion holds for all functions $f' \in B_n$ with $L_\Omega(f') < \ell$ and all complexity measures μ.

Let β be an optimal standard formula for f. Let v be the last gate of β and g, h be the input functions of v. Then

$$f = g \; op(v) \; h$$

where $op(v) \in \{\wedge, \vee\}$.

The subformulas for g and h are optimal, too.

Induction hypothesis $\Rightarrow$

$$\begin{aligned} L_{\Omega_0}(f) &= L_{\Omega_0}(g) + L_{\Omega_0}(h) \\ &\geq \mu(g) + \mu(h) \\ &\geq \mu(f). \end{aligned}$$

■

We look for a formal complexity measure K such that for many difficult functions f $K(f)$ is large and can be estimated easily.

Let $H(A,B)$ denote the set of neighbours $(a,b) \in A \times B$. Let

$$K_{AB} := \frac{|H(A,B)|^2}{|A| \cdot |B|} \quad \text{and}$$

$$K(f) := \max \{ K_{AB} \mid A \subseteq f^{-1}(1),\ B \subseteq f^{-1}(0) \}.$$

Theorem 7.5

$$L_{\Sigma_0}(f) \geq K(f) \quad \text{for all } f \in B_n.$$

Proof:

Lemma 7.1 $\Rightarrow$

It suffices to prove that the Khrapchenko measure K is a formal complexity measure.

We have to show that the three properties of the definition of formal complexity measure hold for the Khrapchenko measure K.

i) $K(x_i) = 1$: follows from the observation that all neighbours (a,b) with $a \in x_i^{-1}(1)$ and $b \in x_i^{-1}(0)$ differ exactly in the i-th component. Hence,

$$|H(A,B)|^2 \leq |A| \cdot |B| \quad \text{for all } A \subseteq f^{-1}(1), B \subseteq f^{-1}(0)$$

We obtain equality if we choose

$A = \{(0,0,\dots,0,1,0,\dots,0)\}$ ith unit vector

$B = \{(0,0,\dots,0)\}$ zero vector

ii) Note that the definition of $k(f)$ is symmetric with respect to $f^{-1}(1)$ and $f^{-1}(0)$. Hence,

$$k(f) = k(-f).$$

iii) We have to show that

$$k(f \vee g) \leq k(f) + k(g).$$

We choose $A \subseteq (f \vee g)^{-1}(1)$ and $B \subseteq (f \vee g)^{-1}(0)$ in such a way that

$$k(f \vee g) = k_{AB}.$$

Then

$$B \subseteq f^{-1}(0) \text{ and } B \subseteq g^{-1}(0)$$

We partition the set A into disjoint sets

$$A_f \subseteq f^{-1}(1) \quad \text{and} \quad A_g \subseteq g^{-1}(1).$$

Then $H(A,B)$ is the disjoint union of $H(A_f,B)$ and $H(A_g,B)$. Let

$$a_g := |A_g|, \quad a_f := |A_g|, \quad a := |A|, \quad b := |B|$$

$$h_f := |H(A_f,B)|, \quad h_g := |H(A_g,B)| \text{ and } h := |H(A,B)|$$

Then

$$k(f \vee g) = \frac{h^2}{a \cdot b} \quad \text{and}$$

$$k(f) + k(g) \geq \frac{h_f^2}{a_f \cdot b} + \frac{h_g^2}{a_g \cdot b}.$$

It suffices to prove

$$h^2 \cdot a^{-1} b^{-1} \leq h_f^2 a_f^{-1} b^{-1} + h_g^2 a_g^{-1} b^{-1}$$

$$\Leftrightarrow \quad h^2 a^{-1} \leq h_f^2 a_f^{-1} + h_g^2 a_g^{-1}$$

Note that $h = h_f + h_g$ and $a = a_f + a_g$.

Hence,

$$\Leftrightarrow \quad (h_f + h_g)^2 a_f a_g \leq h_f^2 a_g (a_f + a_g) + h_g^2 a_f (a_f + a_g).$$

$$\Leftrightarrow \quad (h_f^2 + 2 h_f h_g + h_g^2) a_f a_g \leq h_f^2 a_g^2 + h_f^2 a_g a_f + h_g^2 a_f^2 + h_g^2 a_g a_f$$

$$\Leftrightarrow \quad 0 \leq h_f^2 a_g^2 - 2 h_f h_g a_f a_g + h_g^2 a_f^2$$

$$= (h_f a_g - h_g a_f)^2 \qquad \checkmark$$

∎

Next we shall show that the largest lower bound which can be proved using Knopp-Chenko's method is n^2.

Theorem 7.6

$$K(f) \leq n^2 \quad \text{for all } f \in B_n.$$

Proof:

Note that each vector has at most n neighbours. Hence,

$$|H(A,B)| \leq \min\{n|A|, n|B|\}.$$

$$\Rightarrow |H(A,B)|^2 \leq n^2 \cdot |A| \cdot |B|.$$

Finally, we apply Khrapchenko's method to the parity function.

Theorem 7.7

Let $f_n(x) := x_1 \oplus x_2 \oplus \dots \oplus x_n$ be the parity function. Then $L_{\Omega_0}(f_n) \geq n^2$.

Proof:

If $A = f_n^{-1}(1)$ and $B = f_n^{-1}(0)$ then

$$|A| = |B| = 2^{n-1}.$$

Each neighbour of $a \in A$ is in B and vice versa.

$\Rightarrow$

$$|H(A,B)| = n \cdot 2^{n-1}$$

Hence by Theorem 7.5

$$L_{\Omega_0}(f_n) \geq n^2.$$

Exercise:

Let $n = 2^k$, $k \in \mathbb{N}$. Prove that

$$L_{\Omega_0}(f_n) = n^2.$$