3.1 Boolean sums

References

- E. I. Neciporuk, On a Boolean matrix, Systems Theory Res. 21 (1971), 236 – 239.
- Nicholas Pippenger, On another Boolean matrix, TCS 11 (1980), 49 – 56.
- Ingo Wegener, A new lower bound on the monotone network complexity of Boolean sums, Acta Informatica 13 (1980), 109 – 115.
- Kurt Mehlhorn, Some remarks on Boolean sums, Acta Informatica 12 (1979), 371 – 375.

A Boolean sum is the disjunction of a set of variables. We shall consider the monotone network complexity of sets of Boolean sums

$$f = (f_1, f_2, \dots, f_m) : \{0,1\}^n \to \{0,1\}^m$$

where

$$f_i = \bigvee_{j \in F_i} x_j \quad \text{and} \quad F_i \subseteq \{1, 2, \dots, n\}.$$

A set of Boolean sums is called (h,k)-disjoint if for all pairwise distinct $i_0, i_1, \dots, i_h$:

$$|F_{i_0} \cap F_{i_1} \cap \dots \cap F_{i_h}| \le k.$$

This means that any $h+1$ different Boolean sums have at most k variables in common.

Let $K_{s,t} = (A, B, E)$ denote the complete bipartite graph with $|A| = s$ and $|B| = t$.

We can represent a set $f: \{0,1\}^n \to \{0,1\}^m$ of Boolean sums by a bipartite graph

$$G_f = (X_n, F_m, E), \text{ where}$$

$X_n = \{x_1, x_2, \dots, x_n\}$, $F_m = \{f_1, f_2, \dots, f_m\}$ and $E = \{(x_j, f_i) \mid 1 \le j \le n,\ 1 \le i \le m \text{ and } j \in F_i\}$.

Then f is (h,k)-disjoint iff G_f does not contain $K_{k+1,h+1}$.

For an explicitly constructed set of Boolean sums in $M_{n,n}$, Neciporuk has proved an $\Omega(n^{3/2})$ lower bound for its monotone com= plexity in 1969. This was the first non-linear lower bound for an explicitly defined func= tion in $M_{n,m}$, $m \in O(n)$. The proof builts on the fact that $(1,1)$-disjoint Boolean sums have "nothing in common". Necipo= ruk uses a well known construction of a bipartite graph which contains $\Omega(n^{3/2})$ edges and no $K_{2,2}$.

T. Kővári, V.T. Sós, P. Turán, On a problem of K. Zarankiewicz, Colloq. Math 3 (1954), 50 – 57.

The well known problem of Zarankiewicz is the following. Let $G_2(n,n)$ denote a bipartite graph with n nodes in each colour class.

What is the maximal size $z(n,k)$ of a graph in $G_2(n,n)$ which does not contain a $K_{k,k}$?

Upper and lower bounds for $z(n,k)$ are known. For $k=2$ and $k=3$, these bounds are tight up a constant factor. For $k>3$, there is a non-constant gap between upper and lower bound such that the problem of Zarankiewicz is still open.

Some years later, Pippenger and Mehlhorn have generalized the approach of Neciporuk to Boolean sums which have "little in common" such that little can be gained by using conjunctions or overlap. We shall present Kurt Mehlhorn's proof of the following theorem.

Theorem 3.4

Let $f:\{0,1\}^n \to \{0,1\}^m$ be a (h,k)-disjoint set of Boolean sums. Then

$$C_{\Omega_m}(f) \geq \sum_{i=1}^{m} \frac{\frac{|F_i|}{k} - 1}{h \cdot \max\{1, h-2, k-2\}} .$$

The proof of Theorem 3.4 is based on two lemmas. The first lemma shows that using $\wedge$-gates can save at most the factor $\max\{h-2, k-2\}$.

Lemma 3.1

Let $f: \{0,1\}^n \to \{0,1\}^m$ be a (h,k)-disjoint set of Boolean sums. Then

$$C_{\vee}(f) \leq \max\{1, h-2, k-2\}\, C_{\wedge,\vee}(f).$$

Proof:

Let β be an optimal Ω_m-network for f. Suppose that β contains

s $\vee$-gates and t $\wedge$-gates.

$$\Longrightarrow$$

$$C_{\Omega_m}(f) = s + t.$$

The idea is to eliminate successively the $\wedge$-gates using at most $k-1$ and $h-1$, respectively additional $\vee$-gates. Since the eliminated $\wedge$-gate is saved, we need for each elimination of an $\wedge$-gate at most $k-2$ and $h-2$, respectively additional gates.

More precisely, we construct a sequence $\beta_0, \beta_1, \ldots, \beta_t$ of monotone networks where β_i, $0 \leq i \leq t$ contains

$t-i$ $\wedge$-gates and $\leq s + \max\{1, h-1, k-1\} \cdot i$ $\vee$-gates.

Note that $\beta_0 = \beta$. Suppose that β_i, $0 \leq i < t$ is already constructed.

Construction of β_{i+1}:

Let v be a last $\wedge$-gate in topological order; i.e., between v and the output nodes there is no other $\wedge$-gate. Since β_i is acyclic, the gate v exists.

Let $res_{\beta_i}(v) = g$ and let g_1 and g_2 be the functions computed at the ingoing edges of v. Then the DNF-formula of $res_{\beta_i}(v)$ has the following form:

$$g = s_1 \vee s_2 \vee \ldots \vee s_p \vee t_1 \vee t_2 \vee \ldots \vee t_q$$

where each s_j is a variable and each t_j is a monomial of length at least two.

We distinguish two cases.

Case 1: $p \leq k$

By application of Theorem 3.2 with respect to $t_1, t_2, \ldots, t_q$, g can be replaced by

$$s_1 \vee s_2 \vee \ldots \vee s_p.$$

For doing this, we need

$$p-1 \leq k-1$$

additional $\vee$-gates, saving the $\wedge$-gate v.

The resulting network is B_{i+1}.

Case 2:

Let $f_1, f_2, \dots, f_\ell$ be the output functions which depend on $g = res_{B_i}(v)$.

By construction, between v and f_j, $1 \leq j \leq \ell$ there are only $\vee$-gates. Hence, we can write for $1 \leq j \leq \ell$

$$f_j = g \vee u_j.$$

Since f_j is a Boolean sum, u_j is not the constant 1.

Furthermore, for $1 \leq j \leq \ell$

$$\{s_1, s_2, \dots, s_p\} \subseteq F_j.$$

Since f is (h,k)-disjoint, $p > k$ implies

$$\ell \leq h.$$

Claim:

For $1 \leq j \leq \ell$ either $f_j = g_1 \vee u_j$ or $f_j = g_2 \vee u_j$

Proof of claim:

Note that $(g = g_1 \wedge g_2$ and $f_j = g \vee u_j)$

$\Longrightarrow$

$f_j \leq g_1 \vee u_j$ and $f_j \leq g_2 \vee u_j$.

Suppose that

$$f_j < g_1 \vee u_j \quad \text{and} \quad f_j < g_2 \vee u_j .$$

Then there are $\alpha_1, \alpha_2 \in \{0,1\}^n$ with

- $f_j(\alpha_1) = 0$ but $(g_1 \vee u_j)(\alpha_1) = 1$

and

- $f_j(\alpha_2) = 0$ but $(g_2 \vee u_j)(\alpha_2) = 1$.

Let

$$\alpha_1 = (a_1, a_2, \dots, a_n) \quad \text{and} \quad \alpha_2 = (b_1, b_2, \dots, b_n)$$

Consider

$$\alpha = (c_1, c_2, \dots, c_n) \text{ where } c_i := \max\{a_i, b_i\}, \quad 1 \leq i \leq n.$$

Since f_j is a Boolean sum and $f_j(\alpha_1) = f_j(\alpha_2) = 0$ there holds

$$f_j(\alpha) = 0.$$

Since $g_1 \vee u_j$ and $g_2 \vee u_j$ are monotone and

$$(g_1 \vee u_j)(\alpha_1) = (g_2 \vee u_j)(\alpha_2) = 1$$

there holds

$$(g_1 \vee u_j)(\alpha) = (g_2 \vee u_j)(\alpha) = 1 .$$

$\Rightarrow$

$u_j(\alpha) = 1$ or $\big(g_1(\alpha) = g_2(\alpha) = 1$ and hence, $g(\alpha) = 1 \big)$

In both cases, we obtain

$$f_j(\alpha) = (g \vee u_j)(\alpha) = 1,$$

a contradiction.

□

β_{i+1} is constructed from β_i in the following way:

(1) Replace g by the constant 0.

This eliminates the $\wedge$-gate v and at least one $\vee$-gate. After this replacement, the output node f_j, $1 \le j \le l$ computes the function u_j.

(2) For each f_j, $1 \le j \le l$ use one $\vee$-gate to compute $u_j \vee g_k$, $k \in \{1,2\}$ where

$$f_j = g_k \vee u_j.$$

This adds $l \le n$ $\vee$-gates. Since one $\vee$-gate is saved, we need at most

$$l-1 \le n-1$$

additional $\vee$-gates.

The resulting network is β_{i+1}.
This proves the lemma.

Now we shall prove a lower bound for $C_\vee\{f\}$.
All gates in an $\{\vee\}$-network computes the disjunction of some variables. We call such a gate <u>small</u> if the number of these variables is $\leq k$ and <u>large</u> otherwise.

<u>Lemma 3.2</u>
Let $f: \{0,1\}^n \to \{0,1\}^m$ be a (h,k)-disjoint set of Boolean sums. Then

$$C_\vee(f) \geq \sum_{i=1}^{m} \frac{1}{h}\left(\left\lceil \frac{|F_i|}{k} \right\rceil - 1\right).$$

<u>Proof</u>:

Let β be an optimal $\{\vee\}$-network for f.
Note that the input nodes of β are small.

Let v be a large gate in β. Since v computes the disjunction of $> k$ variables and f is (h,k)-disjoint, at most h output nodes f_i depend on v.

The large gates of β connects the small gates of β to the output nodes. Since each small gate computes the disjunction of at most k variables, the output node f_i is connected to at least

$$\left\lceil \frac{|F_i|}{k} \right\rceil$$

small gates.

For each gate v in β let $n(v)$ denote the number of outputs f_i which depend on v. If v is large then $n(v) \leq h$. Hence,

$$\sum_{\text{large } v \in \beta} n(v) \leq h \cdot |\{v \in \beta \mid v \text{ large}\}|.$$

Consider the set H_i of all large gates in β which are connected with the output node which computes f_i, $1 \leq i \leq m$.

As observed above, H_i has to connect at least $\left\lceil \frac{|F_i|}{k} \right\rceil$ different small nodes with the output node f_i. Hence, this subnet-work must contain a binary tree with $\left\lceil \frac{|F_i|}{k} \right\rceil$ leaves.

$\Longrightarrow$

$$\sum_{\text{large } v \in \beta} n(v) = \sum_{i=1}^{m} |H_i|$$

$$\geq \sum_{i=1}^{m} \left(\left\lceil \frac{|F_i|}{k} \right\rceil - 1 \right).$$

Hence we obtain

$$|\{v \in \beta \mid v \text{ large}\}| \geq \frac{1}{h} \sum_{i=1}^{m} \left(\left\lceil \frac{|F_i|}{k} \right\rceil - 1 \right).$$

This implies

$$C_v(f) \geq \frac{1}{h} \sum_{i=1}^{m} \left(\left\lceil \frac{|F_i|}{k} \right\rceil - 1 \right).$$

Combining Lemma 3.1 and Lemma 3.2, Theorem 3.4 can be proved in the following way:

$$C_{\Omega_m}(f) \underset{\text{Le }3.1}{\geq} \frac{1}{\max\{1, h-2, k-2\}} \, C_v(f)$$

$$\underset{\text{Le }3.2}{\geq} \sum_{i=1}^{m} \frac{\frac{|F_i|}{k} - 1}{h \cdot \max\{1, h-2, k-2\}} .$$

$\blacksquare$

Nick Pippenger has given another proof for Theorem 3.4. It would be interesting to read Pippenger's proof and to compare both proofs with respect to the used properties of the function f.

Exercise

Read Pippenger's paper and compare both proofs.

In 1966, W. G. Brown has constructed a graph $G_2(n,n)$ which contains $\Omega(n^{5/3})$ edges and no $K_{3,3}$.

W. G. Brown, On graphs that do not contain a Thompson graph, Canad. Math. Bull. 9 (1966), 281 - 285.

Using this construction, we obtain an explicit (76)
(2,2)-disjoint set of Boolean sums
$f:\{0,1\}^n \to \{0,1\}^n$ with $\sum_{i=1}^{n} |F_i| = \Omega(n^{5/3})$.

Theorem 3.4 implies an $\Omega(n^{5/3})$ lower bound for the monotone network complexity of this function.

Further constructions

- A.E. Andreev, On a family of boolean matrices, Moscow Univ. Math. Bull. 41 (1986), 79-82.
- J. Kollár, L. Rónyai, T. Szabó, Norm-graphs and bipartite Turán numbers, Combinatorica 16 (1996), 399-406.

give sets of Boolean sums $f:\{0,1\}^n \to \{0,1\}^n$ with $\Omega(n^{2-\varepsilon})$ monotone network complexity for arbitrarily small $\varepsilon > 0$.

Especially, Kollár et al. have constructed a graph $G_2(n,n)$ which contains $\geq n^{2-\frac{1}{t}}$ edges and no $K_{t,(t!+1)}$ where $t \geq 2$ is an integer.

3.2 Boolean matrix multiplication

References

- Vaughan R. Pratt, The power of negative thinking in multiplying Boolean matrices, SIAM J. Comput. 4 (1974), 326-330.
- Mike S. Paterson, Complexity of monotone networks for Boolean matrix product, TCS 1 (1975), 13-20.
- Kurt Mehlhorn, Zvi Galil, Monotone switching circuits and Boolean matrix product, Computing 16 (1976), 99-111.

In 1974, Pratt has shown that each monotone network computing the product of two $n \times n$ Boolean matrices contains at least $\frac{1}{2}n^3$ $\wedge$-gates. Mehlhorn and Galil and Paterson have refined the method of Pratt and haved proved that the school-method is the unique optimal monotone network for Boolean matrix multiplication. We shall present the proof of Mehlhorn and Galil and Paterson.

Let $r, p, q \in \mathbb{N}$, A be an $(r \times p)$ and B be a $(p \times q)$ Boolean matrix. Then

$$C := A \cdot B$$

is an $(r \times q)$ Boolean matrix where

$$c_{ik} := \bigvee_{j=1}^{p} a_{ij} \wedge b_{jk}.$$

Theorem 3.5

Each monotone network which computes the product of an $(r \times p)$ Boolean matrix and a $(p \times q)$ Boolean matrix contains at least $r \cdot p \cdot q$ $\wedge$-gates and at least $r \cdot q(p-1)$ $\vee$-gates.

Proof:

Let β be an optimal monotone network computing the product matrix $C = A \cdot B$.

Goal:

The isolation of an $\wedge$-gate which computes $a_{ij} b_{jk}$ for all triples (i,j,k).

General method:

① Definition of predicates P on the nodes of β such that

- P holds for at least one output node of β and P does not hold for any in=put node of β.

$\Rightarrow$ There exists a gate v in β such that

- P does not hold for any ingoing edge of v but P holds for $res_\beta(v)$.

Let $I(P)$ denote the set of these gates.

$\rightsquigarrow$

1) Identification of $r \cdot q$ $\wedge$-gates for the products

$$a_{i1} b_{1k}, \quad 1 \le i \le r, \quad 1 \le k \le q$$

and elimination of these gates by setting

$$\begin{aligned} a_{i1} &\text{ to } 1, \quad 1 \le i \le r \\ b_{1k} &\text{ to } 0, \quad 1 \le k \le q. \end{aligned}$$

The resulting monotone network computes the $(r, p-1, q)$-matrix product C' where

$$c'_{ik} = \bigvee_{j=2}^{p} a_{ij} \wedge b_{jk}.$$

2) Application of induction.

If we consider a gate v then

- h denotes always $res_\beta(v)$ and
- h_1, h_2 denote the input functions of v.

For $1 \le i \le r$, $1 \le k \le q$ we define the predicate P_{ik} by

$$P_{ik}(v) \iff a_{i1} b_{1k} \le h \wedge a_{i1} \not\le h \wedge b_{1k} \not\le h.$$

This implies that $a_{i1} b_{1k}$ is a prime implikant of h.

The input nodes of β does not fulfill P_{ik} but the output function c_{ik} fulfills P_{ik}.

$$\Longrightarrow \qquad I(P_{ik}) \neq \emptyset.$$

Consider $v \in I(P_{ik})$. First we shall show that v is an $\wedge$-gate.

Suppose that v is an $\vee$-gate. Then

$a_{i1} b_{1k} \le h$ implies

$$a_{i1} b_{1k} \le h_1 \text{ or } a_{i1} b_{1k} \le h_2.$$

W.l.o.g., we can assume

$$a_{i1} b_{1k} \le h_1.$$

Now $\neg P_{ik}(h_1)$ implies

$$a_{i1} \le h_1 \text{ or } b_{1k} \le h_1.$$

$$\Longrightarrow \qquad \neg P_{ik}(h), \text{ a contradiction.}$$

Hence, v is an $\wedge$-gate.

Since $a_{i1} b_{1k} \leq h$ there hold

$$a_{i1} b_{1k} \leq h_1 \text{ and } a_{i1} b_{1k} \leq h_2.$$

Furthermore,

$$\neg P_{ik}(h_1) \text{ and } \neg P_{ik}(h_2) \Rightarrow$$

Either

$$a_{i1} \leq h_1 \text{ and } b_{1k} \leq h_2$$

or

$$a_{i1} \leq h_2 \text{ and } b_{1k} \leq h_1.$$

Note that $a_{i1} \leq h_1 \wedge a_{i1} \leq h_2 \Rightarrow a_{i1} \leq h$ and hence, $\neg P_{ik}(h)$. Analogously, we can exclude $b_{1k} \leq h_1 \wedge b_{1k} \leq h_2$.

Altogether, we have found for each pair (i,k) an $\wedge$-gate v with $P_{ik}(v)$ holds.

Now we shall prove that these $\wedge$-gates are pairwise distinct. For doing this, suppose

$$v \in I(P_{i_1k_1}) \cap I(P_{i_2k_2}) \text{ with } (i_1,k_1) \neq (i_2,k_2)$$

Up to symmetrie, one of the following two situations occurs.

1) $a_{i_11}, a_{i_21} \leq h_1$ and $b_{1k_1}, b_{1k_2} \leq h_2$

2) $a_{i_11}, b_{1k_2} \leq h_1$ and $b_{1k_1}, a_{i_21} \leq h_2$.

As we shall see, we can apply in both situations Theorem 3.3 to construct a better network than β. This contradicts the optimality of β.

Situation 1:

Suppose $i_1 \neq i_2$. Then for the input function h_1 and

$$t = 1 \ , \ t_1 = a_{i_1 1} \text{ and } t_2 = a_{i_2 1} \ ,$$

the assumptions of Theorem 3.3 are fulfilled. Hence, h_1 can be replaced by $h_1 \vee 1$.

$\Rightarrow$

One input of v gets to be constant such that the gate v can be eliminated.

This contradicts the optimality of β.

If $i_1 = i_2$ then $k_1 \neq k_2$. Analogously, this case can be excluded as well.

Situation 2:

By application of Theorem 3.3, we can replace both inputs of v by 1.

Exercise:

Show that in Situation 2, both inputs of v can be fixed at 1 without changing the function computed by β.

This contradicts the optimality of β as well.

Altogether, we have proved that the sets $I(P_{ik})$ are pairwise disjoint.

$\Longrightarrow$

$r \cdot q$ different $\wedge$-gates are isolated.

Now we shall consider the $\vee$-gates in β. Analogous to the consideration of the $\wedge$-gates, we define for $1 \leq i \leq r$, $1 \leq k \leq q$ a predicate Q_{ik} in the following way:

$Q_{ik}(v) \Longleftrightarrow a_{i1} b_{1k} \leq h \leq A_i \vee b_{1k}$ and $h \nleq b_{1k}$

where $A_i := \bigvee_{j \neq 1} a_{ij}$.

The input nodes does not fulfill Q_{ik}. The output node c_{ik} fulfills Q_{ik} since

- $a_{i1} b_{1k} \in PIM(c_{ik}) \Rightarrow a_{i1} b_{1k} \leq c_{ik}$.
- Each prime implicant of c_{ik} contains the variable b_{1k} or a variable in $\{a_{i2}, a_{i3}, \ldots, a_{ip}\}$. Hence,

  $$c_{ik}(\alpha) = 1 \Rightarrow (A_i \vee b_{1k})\alpha = 1.$$

  $\Longrightarrow$

  $$c_{ik} \leq A_i \vee b_{1k}.$$

- $c_{ik} \nleq b_{1k}$ is obvious.

Hence, $I(Q_{ik}) \neq \phi$ if $p \geq 2$.

Exercise:

Prove the following two assertions:

a) If $v \in I(Q_{ik})$ then v is an $\vee$-gate and either $h_1 \leq b_{1k}$ or $h_2 \leq b_{1k}$

b) The sets $I(Q_{ik})$ are pairwise disjoint.

Altogether, we have achieved the following:

1) For every pair (i,k), we have located an $\wedge$-gate in β with one of its input functions has prime implicant a_{i1}. For different pairs, we have identified different $\wedge$-gates.

$\Longrightarrow$

Fixing a_{i1} at 1 for $1 \leq i \leq r$ eliminates $r \cdot q$ $\wedge$-gates in β.

2) If $p > 1$ then we have located an $\vee$-gate for all pairs (i,k) where one of its input functions contains only prime implicants containing the variable b_{1k}. For different pairs, we have identified different $\vee$-gates.

$\Longrightarrow$

Fixing b_{1k} at 0 for $1 \leq k \leq q$ eliminates $r \cdot q$ $\vee$-gates in β.

Exercise

Show that fixing a_{i1} at 1 and b_{1k} at 0 for $1 \le i \le r$ and $1 \le k \le q$ transforms β into an optimal monotone network for the functions

$$c'_{ik} := \bigvee_{j=2}^{p} a_{ij} b_{jk}.$$

Altogether, applying induction, the assertion of the theorem is proved. ∎

3.3 A generalized Boolean matrix product

References

- Ingo Wegener, Switching functions whose monotone complexity is nearly quadratic, TCS 9 (1979), 83-97.
- Ingo Wegener, Boolean functions whose monotone complexity is of size $\frac{n^2}{\log n}$, TCS 21 (1982), 213-224.

Let Y be the Boolean matrix product of the matrix X_1 and the transposed matrix X_2. Then we have $y_{ij} = 1$ iff the i-th row of X_1 and the j-th row of X_2 have a common one.

In 1978, Wegener has generalized this to the "direct product" of m $M \times N$-matrices $X_1, X_2, \ldots, X_m$. For each choice of one row of every matrix, the corresponding output is one iff the chosen rows have a common one.

To formalize this, let x^i_{he} denote the element of matrix M_i at position (h, e). Then we say that x^i_{he} is a variable of <u>type e</u>. This means that the type of a variable is its position in the corresponding row.

For $1 \leq h_1, h_2, \ldots, h_m \leq M$ let

$$y_{h_1 h_2 \ldots h_m} := \bigvee_{1 \leq e \leq N} x^1_{h_1 e} x^2_{h_2 e} \cdots x^m_{h_m e}.$$

We say that the prime implicant

$$(h_1, h_2, \ldots, h_m, e) := x^1_{h_1 e} x^2_{h_2 e} \cdots x^m_{h_m e}$$

is of <u>type e</u>.

<u>The generalized Boolean matrix product</u> f^m_{MN}, $m, M \geq 2$ is defined as follows:

$$f^m_{MN} : \{0,1\}^{mMN} \longrightarrow \{0,1\}^{M^m}$$

where

$$y_{h_1 h_2 \ldots h_m} := \bigvee_{1 \leq e \leq N} x^1_{h_1 e} x^2_{h_2 e} \cdots x^m_{h_m e}.$$

First, we shall prove an upper bound for the monotone network complexity of f^m_{MN}.

<u>Theorem 3.6</u>

$$C_{\Omega_m}(f^m_{MN}) \le N \cdot M^m (2 + (M-1)^{-1}) \le 3NM^m.$$

<u>Proof</u>:

Assume that all monomials $(h_1, h_2, \ldots, h_{i-1}, l)$ are computed. Then $N \cdot M^i$ additional $\wedge$-gates suffice to compute all monomials $(h_1, \ldots, h_i, l)$

$\Rightarrow$

$$C_\wedge(f^m_{MN}) \le N \cdot \sum_{2 \le i \le m} M^i$$

$$\le N(M^{m+1} - 1)(M-1)^{-1}$$

$$\le NM^m(1 + (M-1)^{-1}).$$

Afterwards, each output can be computed using $N-1$ $\vee$-gates. ∎

Our goal is to prove a $\frac{1}{2} NM^m$ lower bound for the number of $\wedge$-gates in a monotone network which computes f^m_{MN}.

First, we shall investigate the structure of optimal monotone networks computing f^m_{MN}.

Lemma 3.3

Let g be a function which is computed in a monotone network for f^m_{MN}. Let $(i_1, i_2, \ldots, i_m, \ell), (j_1, j_2, \ldots, j_m, \ell) \in PIM(g)$ for some $\ell \in \{1, 2, \ldots, N\}$ and $i_k, j_k \in \{1, 2, \ldots, M\}$, $1 \le k \le m$. Let $A := \{ k \mid i_k = j_k \}$ and let

$t := \bigwedge_{k \in A} x^k_{i_k \ell}$. Then g can be replaced

by $h := g \vee t$ and the resulting network still computes f^m_{MN}.

Proof:

Let

$$t_1 := \bigwedge_{k \notin A} x^k_{i_k \ell} \quad \text{and} \quad t_2 := \bigwedge_{k \notin A} x^k_{j_k \ell}.$$

Definition of $A \Rightarrow$

t_1 and t_2 have no common variable.

Now we shall prove that the assumptions of Theorem 3.3 are fulfilled with respect to the chosen t, t_1 and t_2.

By construction

$$tt_1, tt_2 \in PIM(g) \subseteq IM(g).$$

To prove the second assumption, we assume that this assumption does not hold; i.e.,