If a monomial \( t' \) and a output function 
\[ y_{1}, y_{2}, \ldots, y_{m} \]
with
\[ t' t t_1, t' t t_2 \in IM(\overline{y}_{1}, \overline{y}_{2}, \ldots, \overline{y}_{m}) \]
but
\[ t' t \notin IM(\overline{y}_{1}, \overline{y}_{2}, \ldots, \overline{y}_{m}). \]

Then
\[ t' t t_2 \] contains a prime implicant \( (\overline{u_{1}}, \ldots, \overline{u_{m}}, e') \)
but
\[ t' t \] does not contain the prime implicant 
\( (\overline{u_{1}}, \ldots, \overline{u_{m}}, e'). \)

By construction, it holds \( e' = e \).
The same holds with respect to \( t' t t_1 \).

Altogether, we obtain
\[ (\overline{u_{1}}, \overline{u_{2}}, \ldots, \overline{u_{m}}, e) \]
is a submonomial of \( t' t t_1 \), and also a submonomial of 
\( t' t t_2 \) but not a submonomial of \( t' t \).

\[ \Rightarrow \]
\[ t_1 \] and \( t_2 \) have a variable in common.
But this contradicts the definition of \( A. \)

A monomial \( m \) is useful for an output of \( \overline{f} \) if \( m \) is a submonomial of a prime
Lemma 3.3 \implies

If the function \( g = \text{res}_P(w) \) includes several useful monomials of type \( e \), then we can replace \( g \) in \( P \) by \( \text{gvt} (t) \) where \( t \) is the common part of all useful monomials of type \( e \).

Goal:

Application of Lemma 3.3 without additional cost.

For doing this, we need the following properties:

1) \( v \)-gates are not counted.
2) All monomials of \( < m \) variables are given for free; i.e., are given as additional inputs of the network.

A monotone network fulfilling both properties is called an \(*\)-network. \( C_{\text{\text{*}}}^m \) is the associated complexity measure.

Given any \(*\)-network \( P \) for \( f^m_{\text{\text{*}}} \), we wish to transform \( P \) into a so-called standard \(*\)-network of the same complexity by applying Lemma 3.3.
For doing this, we consider the gates in $\beta$ in any topological order. Let $u$ be the current considered gate and $g := res_{\beta}(u)$.

For $1 \leq e \leq N$ let 

$$t_e := \begin{cases} 
0 & g \text{ contains at most one useful monomial of type } e \\
\text{common part of all useful monomials of type } e & \text{otherwise}
\end{cases}$$

Without additional cost, we replace $g$ by $g \cdot t_1 \cdot t_2 \cdots t_N$

$\Rightarrow$

altogether, we obtain a $\ast$-network $\beta'$ for $f_{MN}$ such that:

- all functions computed at the incoming and outgoing edges of the $n$-gates have at most one useful monomial of type $e$ as prime implicant.

**Theorem 3.7**

Let $m \geq 2$. Then

$$C_{EH}(f_{MN}) > C_{E}(f_{MN}) > C_{EH}^{\ast}(f_{MN}) > \frac{1}{2} N \cdot M^m$$
Corollary 3.1

For $n \geq 4$ let $\mu(n) := \lceil \log n \rceil$, $M(n) := 2$, 
$N(n) := \lfloor \frac{n}{2 \log n} \rfloor$ and $h_n := \frac{\mu(n)}{M(n)N(n)}$. Then the function $h_n$ depends on at most $n$ variables and lies at most $n$ output functions. Furthermore, $C_{2N}(h_n) = O(\frac{n^{2}}{\log n})$.

Proof: Exercise

Proof of Theorem 3.7:

Let $f_m$ be an optimal standard $*$-network for $f_{mn}$.

Goal:

For each $\ast$-gate $v \in B$, definition of a value function

$$ c_v : PIM(f_{mn}) \to [0,1] $$

such that

$$ c(v) := \sum_{1 \leq n_1, n_2, \ldots, n_m \leq M} \sum_{1 \leq e \leq N} c_v(n_1, n_2, \ldots, n_m, e) $$

Properties:

a) $c_v(n_1, n_2, \ldots, n_m, e)$ is an estimate of the
contribution of the \( v \)-gate \( v \) to the computation of the prime implicant \((h_{1}, \ldots, h_{m}, e)\).

b) Each gate contributes to all prime implicants at most the value one.

\[ \forall v \]

For an optimal \( \ast \)-network \( \beta \) there holds

\[
c(\beta) := \sum_{v \text{ \( v \)-gate in } \beta} c(v) \leq C_{w}^{\ast}(\beta) = C_{w}^{\ast}(f_{MN}^{m}).
\]

This means that \( c(\beta) \) is a lower bound for \( C_{w}^{\ast}(f_{MN}^{m}) \). The value function will have the following property:

**Claim 1.**

\[
c(\beta, h_{1}, h_{2}, \ldots, h_{m}, e) := \sum_{v \text{ \( v \)-gate in } \beta} c_{v}(h_{1}, h_{2}, \ldots, h_{m}, e) > \frac{1}{2}.
\]

Before defining the value function and proving the claim, we shall terminate the proof of the theorem.

**Claim 1 \( \Rightarrow \)**

\[
C_{w}^{\ast}(f_{MN}^{m}) \geq \sum_{e \in \mathbb{N}} \sum_{1 \leq h_{1}, \ldots, h_{m} \leq M \text{ \( e \)-seen}} c(\beta, h_{1}, h_{2}, \ldots, h_{m}, e) > \frac{1}{2} \cdot N \cdot M^{m}.
\]
Definition of the value function

Let $v$ be an $n$-gate in an optimal standard $x$-network $\beta$ for $f^m$. Let

$g', g''$ be the functions of the outgoing edges of $v$

$g := \text{res}_g(v)$.

Idea:

The value function $c_v$ assigns to the $n$-gate $v$ a positive value for the prime incident $t$ in $\text{PIM}(f^m)$ if

- $t \in \text{PIM}(g)$ and $t \notin \text{PIM}(g')$
- $t \in \text{PIM}(g')$ and $t \notin \text{PIM}(g'')$.

Question:

In these cases, which value $z_0$ should be chosen?

To define these values let

$i_1, i_2, \ldots, i_q \in \{1, 2, \ldots, N\}$ be those types such that

- $t \in \text{PIM}(f^m)$ of type $i$ with $t \in \text{PIM}(g)$ but $t \notin \text{PIM}(g')$.

Furthermore, let
\( j_0, j_1, \ldots, j_{n_1} \in \{1, 2, \ldots, N\} \) be these types such that

\[ I \in \text{PI}(\mathbf{f}_{MN}) \] of type \( j_0 \) with

\[ I \in \text{PI}(\mathbf{g}) \] but \( I \not\in \text{PI}(\mathbf{g}') \).

Then we define for \( I \in \text{PI}(\mathbf{f}_{MN}) \)

\[ c_v^I(t) := \begin{cases} \frac{1}{2q} & \text{if } I \in \text{PI}(\mathbf{g}) \text{ and } I \not\in \text{PI}(\mathbf{g}') \\ 0 & \text{otherwise} \end{cases} \]

\[ c_v''(t) := \begin{cases} \frac{1}{2q} & \text{if } I \in \text{PI}(\mathbf{g}) \text{ and } I \not\in \text{PI}(\mathbf{g}') \\ 0 & \text{otherwise} \end{cases} \]

Then, \( c_v(t) \) is defined by

\[ c_v(t) := c_v^I(t) + c_v''(t). \]

Then we obtain because of property (*)

\[ c_v^I(u) = \sum_{1 \leq j_1, \ldots, j_{n_1} \leq m} \sum_{l \in \mathbb{N}} c_v^I(j_1, \ldots, j_{n_1}, l) \]

\[ = q^I \cdot \frac{1}{2q} = \frac{1}{2} \]

Note that in an optimal standard \( \star \)-network, at

an \( \wedge \)-gate at most one prime implicant of each

type can have a value \( > 0 \).
\[ c''(v) = \sum_{1 \leq i_1, \ldots, i_m \leq M} \sum_{1 \leq t} c_i^n (f_{i_1}^{i_{i_1}} \ldots f_{i_m}^{i_{i_m}}, t) \]

\[ = q^n \cdot \frac{1}{2q^n} = \frac{1}{2}. \]

\[ \Rightarrow \]

\[ c'(v) = c'(v) + c''(v) = 1. \]

It remains to prove Claim 1.

**Proof of Claim 1:**

Consider the prime implicant \( t := (f_{i_1}, f_{i_2}, \ldots, f_{i_m}, t) \) and the corresponding output

\[ y_t := y_{i_1} y_{i_2} \ldots y_{i_m}. \]

Let \( \beta(t) \) be that subnetwork of \( \beta \) which contains the following gates and inputs:

- Gate \( v \) is contained in \( \beta(t) \) if there is a path \( P \) in \( \beta \) from \( v \) to the output \( y_t \) and \( t \) is a prime implicant of all functions computed on \( P \) (inclusive respect).

Additionally, the inputs of the gates in \( \beta(t) \) are contained in \( \beta(t) \) as well.

**Properties:**

1) For each input function \( g \) of \( \beta(t) \), holds \( t \in \Pi(g) \).
2) $t$ is prime implicant of each function $\text{res}_B(v)$ where $v$ is a gate in $B(t)$.

3) Let $v$ be a gate in $B(t)$ with both direct predecessors of $v$ are inputs of $B(t)$.

\[ \Rightarrow \]

$v$ is an $\vee$-gate (otherwise $t \in \text{pim}(\text{res}_B(v))$)

4) If an input of $B(t)$ is input of an $\vee$-gate of $B(t)$ then a proper shortening of $t$ is a prime implicant of that input.

Let $s_1, s_2, \ldots, s_D$ be those inputs of $B(t)$ which are input of an $\vee$-gate of $B(t)$.

Let $v(c_i)$ be an $\vee$-gate of $B(t)$ with input $s_i$.

Let

\[
c^*(v(c_i)) = \begin{cases} 
  c'(v(c_i)) & \text{if } s_i \text{ is the first input of } v(c_i) \\
  c''(v(c_i)) & \text{if } s_i \text{ is the second input of } v(c_i)
\end{cases}
\]

Properties 1 and 2 $\Rightarrow$ $c^*(v(c_i)) > 0$.

For $1 \leq i \leq D$ let $b_i := c^*(v(c_i))$.

W.l.o.g. we can assume $b_1 > b_2 > \ldots > b_D$.

Note $\frac{D}{2} \sum_{i=1}^{D} b_i > \frac{1}{2} \Rightarrow$ Claim 1.

We shall prove $b_1 + b_2 + \ldots + b_D > \frac{1}{2}$. 
Choose \( w_i \in PIM(c_i) \) such that a proper prolongation \( w_i^* \) of \( w_i \) is contained in \( PIM(\text{res}(c_i)) \cap PIM(\text{pm}) \).

and the type of \( w_i^* \) is different to the types of \( w_1^*, w_2^*, \ldots, w_{i-1}^* \).

We can always choose \( w_i^* = t \). We distinguish two cases.

**Case 1:**

The choice of \( w_i \) according to the rules above is impossible for an \( i \leq D \).

\[ \Rightarrow \]

\( c^*(c_i) \) is positive for \( \leq (i-1) \) prime implicants.

Hence, by the definition of the value function

\[ b_i \geq (2c(i-1))^{-1} \]

Because of \( b_1 > b_2 > \ldots > b_D \), we obtain

\[ b_j \geq i b_i \geq i(2c(i-1))^{-1} > \frac{1}{2} \]

**Case 2:**

The choice of \( w_1, w_2, \ldots, w_D \) according to the rules above is possible.

Construction = \( w_i \in PIM(c_i) \) for \( 1 \leq i \leq D \).
$$\Rightarrow w_1 w_2 \ldots w_D \leq s_1 s_2 \ldots s_D$$

Construction of $R(t) \Rightarrow$

A row with $s_i(a) = 1$ for $1 \leq i \leq D$

there holds $y_t(a) = 1$

(this can easily be shown by induction)

$$\Rightarrow w_1 w_2 \ldots w_D \leq y_t.$$ 

All variables $w_i$ are of type $e_i$ and $e_1, e_2, \ldots, e_D$ are pairwise different.

Each $w_i$ is a proper shortening of $w_i^*$ and $\phi^*(\omega(q)) (w_i^*) > 0$.

$\Rightarrow$

$w_i$ contains $\leq m-1$ variables

$\Rightarrow$

$w_1 w_2 \ldots w_D \notin \text{IM}(y_t)$

a contradiction

Hence, Case 2 cannot occur.
3.4 The Boolean convolution

References


- Norbert Blum, An $\Omega (n^{4/3})$ lower bound on the monotone network complexity of the $n$th degree convolution, TCS 36 (1985), 58–69.


Let $A = \{a_0, a_1, \ldots, a_m\}$, $B = \{b_0, b_1, \ldots, b_n\}$ be two disjoint sets of $n$ variables. Then we define the $n$-th degree convolution $C_n$ as follows:

$C_n = (c_0, c_1, \ldots, c_{2n-1}) : \{0, 1\}^{2^n} \rightarrow \{0, 1\}^{2n-1}$, where
All sets of monotone functions considered so far (Boolean sums, Boolean matrix multiplication, generalized Boolean matrix multiplication) have disjunctive properties that Boolean convolution does not have.

To formalize this, we need some notations.

Let \( A = \{a_1, a_2, \ldots, a_n\} \) and \( B = \{b_1, b_2, \ldots, b_m\} \) be two disjoint sets of variables. A monotone function

\[
f = (f_1, f_2, \ldots, f_m): A \cup B \to \{0, 1\}^m
\]

is bilinear if each prime implicant of \( f \) consists of one variable from \( A \) and one variable from \( B \). Then, \( f \) is a set of bilinear forms.

Example:

Boolean matrix multiplication and also the Boolean convolution are bilinear.

We can extend this definition to multilinear forms (i.e., we have more than two disjoint sets of variables) in the obvious way.
A set \( f \) of disjoint bilinear forms is a semi-disjoint set of bilinear forms which also fulfills the following third property:

3) For \( 1 \leq k \leq m \), \( 0 \leq i, j \leq n-1 \),
\[ |PIM_{ij}(f) \cap PIM(f_k)| \leq 1. \]
Boolean matrix multiplication is disjoint.

Boolean convolution is not disjoint.

**Exercise**

a) Give a formal definition of multilinear forms. Extend the definitions of semi-disjointness and disjointness to multi-linear forms.

b) Show that the generalized Boolean matrix multiplication is a set of disjoint multilinear forms.

c) Show that Boolean convolution is not a set of disjoint bilinear forms.

Boolean sums are \((4, \leq)\)-disjoint. Boolean matrix multiplication is a set of disjoint bilinear forms. The generalized Boolean matrix multiplication is a set of disjoint multilinear forms. The Boolean convolution has not such disjointness property. The sets of variables upon which two functions \(f_1, f_2 \in C_n\) depend can be almost equal which is not the case for the sets of functions mentioned above.

As a consequence, the assumptions of Theorem 3.3 do not hold for the Boolean convolution such that this theorem cannot be applied.
Now, we shall prove a general lower bound for the monotone network complexity of semidisjoint bilinear forms.

**Theorem 3.8**

Let $f$ be a semidisjoint bilinear form. Let $T_i$ be the number of prime implicants which contain the variable $a_i$. Then

$$C_{a_i}(f) \geq \sum_{i=1}^{p} T_i^{1/2}$$

**Corollary 3.1**

The monotone network complexity of the Boolean $u$-th degree convolution is $\geq n^{3/2}$.

**Proof:**

This is a direct consequence of Theorem 3.8 since $T_i = n$ for $0 \leq i \leq u-1$.

**Proof of Theorem 3.8:**

Let $f$ be an optimal $D_m$-network for $f$. Note that after replacing $a_0$ by 0, we obtain a sub-function of $f$ which is semidisjoint as well. Moreover, the values $T_i$, $i > 0$ do not change.
It suffices to prove that after setting $a_0$ to 0, at least $r_0^{1/2}$ gates have been eliminated. Let $s_0$ denote the number of functions $f_i$ with $f_i = a_0 b_j$ for any $j$.

Setting $a_0$ to 0 eliminates these $s_0$ $v$-gates where these outputs are computed. Since $f$ is semidisjoint, these gates cannot be used for the computation of other outputs.

**Claim**

Setting $a_0$ to 0 eliminates at least $(r_0 - s_0)^{1/2}$ $v$-gates.

Note that this claim implies that setting $a_0$ to 0 eliminates at least $r_0^{1/2}$ gates.

To prove the claim, we consider exactly those functions $f_k$ which depend on $a_0$ and which contain at least two prime implicants.

Let $P = v_0, v_1, \ldots, v_m$ be a path from $a_0$ to $f_k$.
The path $P$ contains at least one $v$-gate $v_e$ with

$$a_{i_0} j \in \text{resp } (v_e)$$

for an $i \neq 0$ and any $j \in \{0, 1, \ldots, n-1\}$.

Otherwise, because of the semi-disjointness of $f$, the function $f_k$ could not contain two prime implicants.

**Exercise**

Show that the property "$P$ contains no $v$-gate $v_e$ with $a_i j \in \text{resp } (v_e)$ for an $i \neq 0$ and some $j$" implies that $f_k$ does not contain two prime implicants.

The first such an $v$-gate on $P$ is called suitable for $P$. Let

$$\mathcal{P} := \{ P \mid P \text{ is a path from } a_0 \text{ to an output } f_k \text{ which depends on } a_0 \text{ and contains } \geq 2 \text{ prime implicants } \}.$$  

Let

$$V^* := \{ v \text{-gate } v \mid v \text{ is suitable for a path } P \in \mathcal{P} \}.$$  

Consider any $v \in V^*$. By construction, for each gate $w$ between $a_0$ and $v$, each prime implicant $p$ of $\text{resp } (w)$ has the following property:
p contains at least one of the following monomials as a submonomial:

- \(a_0\),
- the conjunction of two variables in \(A\),
- the conjunction of two variables in \(B\).

\(\implies\)

After setting \(a_0\) to 0 and an eventual application of Theorem 3.2, \(\text{res}_B(w)\) can be replaced by 0.

\(\impliedby\)

Setting \(a_0\) to 0 yields for each gate \(v \in V^*\) that one input can be replaced by 0.

\(\implies\)

After setting \(a_0\) to 0, each gate \(v \in V^*\) can be eliminated.

Claim: \(|V^*| > (r_0 - s_0)^{\frac{1}{2}}\)

To prove this claim, let 

\[ V^* = \{v_1, v_2, \ldots, v_q\} \]

Then there are

\(i_1, i_2, \ldots, i_q\) and \(j_1, j_2, \ldots, j_q\)

with \(\sigma_{i_k} b_{j_k} \leq \text{res}_B(v_e)\) for \(1 \leq e \leq q\).
After setting 
\[ a_{i_2}, b_{i_2} \to 1 \text{ for } 1 \leq i_2 \leq q, \]
all gates in \( V^* \) compute the constant 1.

\[ \Rightarrow \]
No output \( f_k \) of \( f \) with \( \geq 2 \) prime implicit

Consider any such an output \( f_k \) and
choose \( s \) such that
\[ a_0 b_s \in \text{PIM}(f_k). \]

After the assignment above, for the function \( f_k' \) computed at that output node,
the following holds:
\[ f_k' = 1 \text{ or } b_s \in \text{PIM}(f_k'). \]

In the second case
\[ a_{i_2} b_s \in \text{PIM}(f_k) \]
for an \( i_2 \in \{1, 2, \ldots, q \}. \)
Since \( i_2 \neq 0 \), this would contradict the

Since \( i_2 \neq 0 \), this would contradict the
semi-disjointness of \( f \). Hence,
\[ f_k'' = 1 \text{ and } a_{i_2} b_{i_k} \in \text{PIM}(f_k) \]
for some \( 1 \leq i_2, i_k \leq q. \)
This holds for all \( r_0 - s_0 \) output nodes depending on \( a_0 \) and having \( \geq 2 \) prime implicants.

Since \( f \) is semidisjoint, prime implicants of different output nodes are different.

At most \( q^2 \) distinct prime implicants can be constructed using \( q \) variables in \( A \) and \( q \) variables in \( B \).

\[ \Rightarrow \quad q^2 \geq r_0 - s_0 \]

\[ \Rightarrow \quad |V| \geq (r_0 - s_0)^{\frac{1}{2}} \]

Now, the assertion can be proved using induction.

Using Theorem 3.8, we have obtained an \( \Omega(n^{3/2}) \) lower bound for the number of \( \vee \)-gates in any monotone network which computes the \( n \)-th degree convolution. To get a lower bound for the number of \( \vee \)-gates, we need some other techniques.

We start with an optimal monotone network \( \beta_0 \) computing \( C_n \). We have no knowledge about the structure of \( \beta_0 \). To get knowledge, we
transform the network into a normal form network $B$, which computes a number of subfunctions of $C_u$. For doing this, we split each output of a gate into several parts. We do this in such a manner that after the transformation the following normal form property holds:

- On every path $P$ leading from an input node $u$ with $\text{op}(u) = b_r \in B$ to an output node there exists a node $w$ such that:
  
  a) the direct successor of $w$ on the path $P$ is an $\land$-gate or the output node and
  
  b) $\exists b_s \in B$, $b_s \neq b_r$ and $\exists A_s \subset A$, $|A_s| > 2 \cdot 9$ such that

  $$b_s \land \left( \bigvee_{a_j \in A_s} a_j \right) \leq \text{res}_B(w).$$

The normal form transformation enlarges the number of $\land$-gates at most by the factor 4. During the transformation, we count some $\land$-gates. After the termination of the normal form transformation, we have counted $L \frac{A}{2} \left( \frac{n^2}{9} + n \right)$ $\land$-gates in $B_0$ and we are done or at least $\frac{n^2}{2}$ products $a_i b_j$ are still computed at the output gates in $B_1$, which compute the subfunction $C_{i,j}$ of $C_{i,j}$.
\[ a_i \text{ such that at least } q \text{ products } a_i b_j \text{ are computed at those output nodes.} \]

Now we first set \( a_2 \) to 1 and then we set successively all \( q \) \( b_2 \)'s to 1. We prove that after every fixing of a \( b_2 \), at least \( \frac{1}{2} \) of \( \land \)-gates are eliminated.

\[ \implies \]

In total, \( \geq \left\lfloor \frac{1}{2} q^2 \right\rfloor \) \( \land \)-gates are eliminated.

To see this let us consider the computation graph which computes the product \( a_i b_e \) at the output node for \( \overline{C} \). It

On every path \( P \) from the input node \( u \) with \( \text{op}(u) = b_2 \) to the output node \( q \) computing \( \overline{C} \) consider the node \( u \) of the normal form property.

Assume there are less than \( \frac{1}{2} q \) \( \land \)-gates.

\[ \implies \]

\( m < q \) pairwise distinct such nodes \( w \) are in the computation graph.

Normal form property \( \implies \)

\[ \exists b_{s_1}, b_{s_2}, \ldots, b_{s_m} \in B, \ b_{s_j} \neq b_e, \ 1 \leq j \leq m \]

and
\[ \exists A_1, A_2, \ldots, A_m \subseteq A, \forall j \geq 2q, 1 \leq j \leq m \exists \]

such that

\[ a_i \bigwedge_{j=1}^{m} \left( b_{s_j} \bigwedge_{a_k \in A_j} (a_k \leq \forall) \right) \leq \text{res}_{B_1}(g). \]

Definition of n-th degree convolution \implies

\[ \forall b_s \in B \text{ there exists at most one } a_t \in A \text{ with} \]

\[ a_t b_s \leq C_{ite}. \]

Hence, for \( 1 \leq j \leq m \), it holds that

\[ \exists a_j \in A_j \text{ such that } a_j \bigwedge_{r=1}^{m} b_{s_r} \leq C_{ite}. \]

and hence,

\[ a_i \bigwedge_{j=1}^{m} a_j \bigwedge_{r=1}^{m} b_{s_r} \leq C_{ite}. \]

By construction,

\[ a_i \bigwedge_{j=1}^{m} a_j \bigwedge_{r=1}^{m} b_{s_r} \leq \text{res}_{B_1}(g). \]

This contradicts that \( C_{ite} = \text{res}_{B_1}(g) \) is a subfunction of \( C_{ite} \).

\[ \implies \exists \frac{1}{2} q \wedge \text{-gates exist in the computation graph.} \]

By setting \( a_i \) and \( b_k \) to \( 1 \), all these \( \wedge \)-gates are eliminated.
Now we shall give the lower bound proof.

The construction of $B_1$

Let $B_0$ be a monotone network computing $C_n$ with minimal number of $\land$-gates. Let $0 < \eta < \frac{1}{2} n$.

Beginning at the input nodes of $B_0$, we shall construct $B_1$ successively. In each step, we consider a node $u$ in $B_0$, the direct predecessors of which were constructed in $B_1$ before.

$u \rightarrow$ small network $S_u$ with output nodes $u'$ and $u''$.

The input nodes of $S_u$ are the output nodes of $S_v$ and $S_w$ where $v$ and $w$ are the direct predecessors of $u$ in $B_0$.

For $0 \leq \ell \leq 2n - 2$, the node in $B_0$ which computes $C_{\ell}^{\land}$ is denoted by $C_{\ell}^{\land}$.

An $\land$-gate $g$ with $\text{pred}(g) = \{ g_1, g_2 \}$ is called a $(\ast)$-type $\land$-gate if

$$\text{op}(g_1) \in B \text{ and } \text{res}_{B_0}(g_2) = \bigvee_{a_j \in A'} a_j$$

where $\emptyset \neq A \subseteq A'$.

The network $B_1$ is constructed such that the following holds:
i) $\text{res}_{B_1}(u') \lor \text{res}_{B_1}(u'') \leq \text{res}_{B_0}(u')$.

ii) If $\exists b_\theta \in B, A_\theta \subseteq A$ maximal, $A_\theta \neq \emptyset$ such that $b_\theta \land (\lor_{\theta \in A_\theta} a_\theta) \leq \text{res}_{B_1}(u')$ then $|A_\theta| \geq 2q$.

iii) On every path $p$ leading from a node $u$ with $\text{op}(u) = b_r \in B$ to an $n$-gate $n$ which is not an $(\times)$-type gate or to the node $u''$ there exists a node $w$ with $\exists b_\theta \in B, b_\theta \neq b_r$ and $\exists A_\theta \subseteq A, |A_\theta| \geq 2q$ such that $b_\theta \land (\lor_{\theta \in A_\theta} a_\theta) \leq \text{res}_{B_1}(w)$.

**Remark**

Property i) means that the output nodes of $S_u$ compute only subfunctions of $\text{res}_{B_0}(u)$. Property iii) ensures that, after the construction of $B_1$, the normal form property introduced above holds.

Now we shall construct $S_u$. We distinguish three cases.

**Case 1:** $u$ is an input node of $B_0$.

Then $S_u$ consists of the nodes
\[
\begin{cases}
    u' \text{ with } \text{op}(u') = \text{op}(u) \\
    u'' \text{ with } \text{op}(u'') = 0
    & \text{if } \text{op}(u) \in B
    \\
    u' \text{ with } \text{op}(u') = 0 \\
    u'' \text{ with } \text{op}(u'') = \text{op}(u)
    & \text{if } \text{op}(u) \in A
\end{cases}
\]

Obviously, conditions i), ii) and iii) hold after this construction

Case 2: \( u \) is an \( v \)-gate with \( \text{pred}(u) = \{v, w\} \).

Then \( \delta_u \) is constructed by

\[ v' \quad \circ \quad w' \quad \text{and} \quad v'' \quad \circ \quad w'' \]

\( v' \quad \circ \quad v \quad \circ \quad w' \quad \text{and} \quad v'' \quad \circ \quad v' \quad \circ \quad w'' \)

For this construction, we need no \( v \)-gate. Since properties i), ii) and iii) hold for \( \delta_v \) and \( \delta_w \), these properties also hold for \( \delta_u \).

Case 3: \( u \) is an \( v \)-gate with \( \text{pred}(u) = \{v, w\} \).

We have to realize \( v'w' \), \( v'w'' \), \( v''w' \) and \( v''w'' \).

Step 1: Realization of \( v''w'' \)

Construct

\[ v'' \quad \circ \quad w'' \]

\[ v'' \quad \circ \quad \text{Op} \quad w'' \]

\[ v'' \quad \circ \quad \text{Op} \quad v'' \quad \circ \quad w'' \]
For the realization of the other three products, we must make sure that the properties ii) and iii) are not destroyed after the construction.

Hence, for \( v' \) and \( w' \), respectively, we distinguish two cases according to whether the following property is fulfilled or not.

We say for a node \( g \in \{ v', w' \} \) that \( g \) is bipotent if

\[ \exists b_s \in B, a_s \in A \text{ and } b_r \in B, b_r \neq b_s, A_r \subseteq A \text{ such that} \]

\[ b_s \land \left( \bigvee_{a_j \in A_s} a_j \right) \lor b_r \land \left( \bigvee_{a_j \in A_r} a_j \right) \leq \text{res}_{\beta_s}(g) \]

\textbf{Remark:}

Since property ii) holds for \( \delta_v \) and \( \delta_w \),

\[ |A_s| \geq 2q \text{ and } |A_r| \geq 2q. \]

If a node \( g \in \{ v', w' \} \) is not bipotent then either \( \text{res}_{\beta_s}(g) = 0 \) or \( \text{res}_{\beta_s}(g) \leq b_r \land \left( \bigvee_{a_j \in A_r} a_j \right) \) for \( b_r \in B \) and \( A_r \subseteq A \).

\textbf{Step 2: Realization of } v'w'.

a) \( v' \) and \( w' \) are bipotent.

Construct \( v \leftarrow u_2 \rightarrow w' \).
b) At least one of \( v \) and \( w' \) is not bipotent.

\[ \Rightarrow \]

There is at most one \( b_s \in B \) such that

\[ b_s \wedge \left( \bigvee_{a_j \in A_s} \alpha_j \right) \leq \text{res}_{\beta_s}(v'w') \quad \text{for} \quad A_s \subseteq A. \]

If no such \( b_s \in B \), \( A_s \subseteq A \) exist then by the structure of \( C_u \) and Theorem 3.2, we can replace \( v'w' \) by \( 0 \) without changing the functions which are computed and we need not realize the product \( v'w' \).

Let \( b_s \in B \), \( A_s \subseteq A \) maximal such that

\[ b_s \wedge \left( \bigvee_{a_j \in A_s} \alpha_j \right) \leq \text{res}_{\beta_s}(v'w'). \]

We distinguish two cases.

i) \( |A_s| \geq 2q \)

Then we construct

\[ \bigcirc \xrightarrow{a} \] \[ \begin{array}{c} \text{where } \alpha \text{ is a network which computes} \\ \bigvee_{a_j \in A_s} \alpha_j \text{ using only } (|A_s|-1) \text{ } \vee \text{-gates.} \end{array} \]

ii) \( |A_s| < 2q \)

Then we do not realize the product \( v'w' \). By the structure of \( C_u \), we destroy the contr...
Step 3: Realization of $v''w''$

- If $v'$ is bipotent:
  - if $v'$ is not bipotent and
  - if $B \subseteq B$, $A_5 \subseteq A$ maximal
    - with $|A_5| \geq 2q$, such that
      - $\lambda_{A_5} \left( V_{-a_j} \right) \notin \text{res}_{A_5} \left( v''w'' \right)$
    - otherwise

Step 4: Realization of $v''w'$

Analogous to that for $v''w''$. Produce result in $u''_4$ or $u''_3$.

If in the construction above
- $u''_j$, $j \in \{1, 2, 3\}$ ($u''_j$, $j \in \{1, 2, 3, 4, 3\}$, resp.)
  - do not exist then construct
    - $u''_j$ with $\text{op}(u''_j) = 0$ ($u''_j$ with $\text{op}(u''_j) = 0$, resp.)

Realization of $u''_j$ and $u''_3$:

$$u'' = \bigvee_{j \in \{1, 2, 3\}} u''_j$$

$$u''_3 = \bigvee_{j \in \{1, 2, 3, 4, 3\}} u''_j$$
For the construction of $\delta u$, we need at most four $\wedge$-gates.

**Exercise**

Prove that after the construction of $\delta u$, the properties i), ii) and iii) are fulfilled for $\delta u$.

If we destroy the computation of some prime implicants of $C_u$, then, by construction, for the number $N$ of these prime implicants we have

$$N < t \cdot 2^q,$$

where $(4 - t)$ $\wedge$-gates are used for the realization of the four products $v'w'$, $v'w''$, $v''w'$ and $v''w''$.

The following lemma characterizes the network $B_u$.

**Lemma 3.3**

In $B_u$, the following properties are fulfilled:

1. For all node $u \in B_o$ and the output nodes $u'$, $u''$ of $\delta u$, the following holds:
   a) $\text{res}_{B_u}(u') \lor \text{res}_{B_u}(u'') \leq \text{res}_{B_o}(u)$.
   b) If $\exists b_s \in B$, $A_s \subseteq A$ maximal, $A_s \neq \emptyset$ such that $b_s \wedge (v \lor g_j) \leq \text{res}_{B_u}(u')$ then $|A_s| > 2^q$.

2. For $0 \leq k \leq 2u - 2$, the output node $c_k$ computes 0.
(3) On every path $P$ leading from a node $li$ with $\text{op}(li) = b_r \in B$ to an $\lor$-gate $g_i$ which is not a $(x)$-type gate or to $c_i''$, $i \in \{0,1,\ldots,2n-2\}$, there exists a node $w$ such that

$$\exists b_s \in B, b_s = b_r \text{ and } \exists A_s \subseteq A, \|A_s\| \geq 2q$$

such that

$$b_s \land (\lor_{a_j}, q_j \in A_s) \leq \text{res}_{b_s}(w).$$

(4) $0 \leq L_\lambda(\beta_1) \leq 4 C_{\lambda}^{\lor}(C_u) - m$ where $L_\lambda(\beta_1)$ is the number of $\lor$-gates in $\beta_1$ and at most $m \cdot 2q$ prime implicants of $C_u$ have been destroyed.

Proof:

From the construction of $\beta_1$, (3) and (3) follow directly. Assertion (2) follows from (3) by and the structure of $c_i''$, $0 \leq i \leq 2n-2$. As observed above, for each $\lor$-gate which is not used for the construction of $c_i''$, at most $2q$ prime implicants have been destroyed. Hence, assertion (4) follows.

Using the network $\beta_1$, we shall prove the following theorem.

**Theorem 3.8**

$$C_{\lambda}^{\lor}(C_u) \geq L_\lambda \frac{A}{B} \min \left\{ \frac{n^2}{q} - n, q^2 3^f \right\}.$$
Setting \( q = n^{\frac{n}{3}} \), we obtain the following corollary.

**Corollary 3.2**

\[
\mathcal{C}_{\overline{C_2}}(C_n) \geq \left\lceil \frac{n}{8} \left( n^{\frac{n}{3}} - n \right) \right\rceil.
\]

**Proof:**

If \( m \) in (4) of Lemma 3.3 \( m \geq \frac{n}{2} \left( \frac{n_2}{q} - n \right) \)
then the lower bound is proved. Otherwise, at least \( \log q \) prime implicants of \( C_n \) remain.

\[ \Rightarrow \]

\( \exists a \in E A, \exists \overline{B} \in B \) with \( |\overline{B}| = q \) such that

\[ a \cdot b \leq \text{res}_{\overline{B}}(C_{i\cdots}) \quad \forall b \in \overline{B}. \]

Let \( \overline{\overline{B}} = \{ b_1, b_2, \ldots, b_q \} \).

We first fix \( a \) to 1 and eliminate all superfluous gates.

**Observation**

Fixing \( a \) to 1 does not destroy the normal form property since, if \( a \in E A \), the set \( A \) grows into the whole set \( A \) after fixing \( a \) to 1.

Then successively we set each \( b \in \overline{\overline{B}} \) to 1

and eliminate all superfluous gates.

Since fixing an input variable to 1 does not affect the property that one function implies another function, during this process the normal form property is not
destroyed.

Now we prove that in each step in which we set a $b_j \in B$ to 1, at least $\frac{1}{2} q^2$ $v$-gates are eliminated.

$\Rightarrow$

After the termination of this process, we have eliminated at least $\frac{1}{2} q^2$ $v$-gates and the theorem is proved.

Assume, we have constructed the monotone function $\beta_2$ from $\beta_1$ by setting

$$ a_1, b_{e_1}, b_{e_2}, \ldots, b_{e_{r-1}}, 1 \leq r < q $$

to 1.

Claim

After setting $b_{e_r}$ to 1, at least $\frac{1}{2} q$ more $v$-gates can be eliminated.

Proof of claim

Since $a_1 b_{e_1} b_{e_2} \ldots b_{e_{r-1}} \neq \text{res}_{\beta_1} (c_{i+e_r})$ and $a_1 b_{e_r} \leq \text{res}_{\beta_1} (c_{i+e_r})$ the following hold:

i) $\text{res}_{\beta_2} (c_{i+e_r}) \neq 1$ and

ii) $b_{e_r} \leq \text{res}_{\beta_2} (c_{i+e_r})$.

Let $v$ be the node in $\beta_2$ with $\text{op}(v) = b_{e_r}$. 

We consider all paths $P_1, P_2, \ldots, P_5$ with
a) start node \( h \) and end node \( c_{i+1}^n \) and
b) \( b_{i,t} \leq \text{res}_{B_2}(v) \) for all nodes \( v \) on \( P_j \), \( 1 \leq j \leq s \)

Since \( b_{i,t} \leq \text{res}_{B_2}(c_{i+1}^n) \), at least one such a path exists.

Obviously, setting \( b_{i,t} \) to 1 eliminates all \( \wedge \)-gates on the paths \( P_1, P_2, \ldots, P_s \).

If on these paths \( \frac{1}{2} q \) \( \wedge \)-gates exist, we are done. Assume that less than \( \frac{1}{2} q \) \( \wedge \)-gates exist.

**Property b1 \( \Rightarrow \)**

All \( \wedge \)-gates on these paths are not \((\wedge)\)-type-\( \wedge \)-gates.

**Name** \( b_0 \) from property \( \Rightarrow \)

For every path leading from the node \( h \) to the first \( \wedge \)-gate \( g \) on a path \( P_j \), \( 1 \leq j \leq s \) (or to \( c_{i+1}^n \) if no \( \wedge \)-gate on \( P_j \) exists) there is a node \( w \) such that

\[ b_{t,j} \in B, b_{t,j} \neq b_{i,t}, \exists A_j \in A \text{ with } |A_j| \geq q \]

such that

\[ b_{t,j} \wedge ( \bigvee_{A_j \in A} a_d ) \leq \text{res}_{B_2}(w) \]

and \( g \in \text{Succ}(w) \) (\( c_{i+1}^n \) \( \text{Succ}(w) \), resp.)
\[ \bigvee_{j \in \mathbb{N}} \left( b_{t_j} \land \left( \bigvee_{a_d \in A_j} a_d \right) \right) \leq \text{res}_{\mathbb{P}_2} (C^*_{\text{iter}}). \]

and if no \( \land \)-gate on \( P_j \) exists then
\[ b_{t_j} \land \left( \bigvee_{a_d \in A_j} a_d \right) \leq \text{res}_{\mathbb{P}_2} (C^*_{\text{iter}}). \]

Assume that for all \( P_j \), \( 1 \leq j \leq s \), an \( \land \)-gate on \( P_j \) exists. (Otherwise, the same proof with \( s = 1 \) works).

Since less than \( \frac{1}{2} q \) \( \land \)-gates exist on \( P_1, P_2, \ldots, P_s \), less than \( q \) of the \( b_{t_j} \), \( 1 \leq j \leq s \) can be pairwise distinct. Let
\[ B' := \{ b_{t_1}, b_{t_2}, \ldots, b_{t_s}, b_{e_1}, b_{e_2}, \ldots, b_{e_{r-1}} \}. \]

Note that \( b_{e_r} \notin B' \). Then we have
\[ a_c \land \bigwedge_{b_j \in B'} b_j \land \left( \bigvee_{a_d \in A_j} a_d \right) \leq \text{res}_{\mathbb{P}_2} (C^*_{\text{iter}}). \]

Since \( \forall b_j \in B' \) there exists at most one \( a_p \in A_j \) with \( a_p \land b_j \leq C_{\text{iter}} \)
( namely \( p = (i + e_r) - j \) if \( (i + e_r) - j \geq 0 \)),
\[ |B'| < 2q \text{ and } |A_j| > 2q, \quad 1 \leq j \leq s, \]
the following holds:
\[ \exists a_{p_3} \in A_j \text{ such that } a_{p_3} \land \bigwedge_{b_j \in B'} b_j \leq C_{\text{iter}}, \]
and hence,
\[ \alpha_i \land \beta_d \land \bigwedge_{j=1}^{s} \alpha_j \land \beta_1 \]

But by construction
\[ \alpha_i \land \beta_d \land \bigwedge_{j=1}^{s} \alpha_j \leq \text{res}_{p_n}(C'') \]

and by Lemma 33, Property (1.a), \( \text{res}_{p_n}(C'') \) is a subfunction of \( C_{iter} \), a contradiction.

\[ \Rightarrow \]

On \( P_1, P_2, \ldots, P_s \) at least \( \frac{1}{2} q \) \( v \)-gates exist.

Hence, the claim and therefore Theorem 3.9 is proved.

Aitrinlick and Sergeev have improved the lower bound for the number of \( v \)-gates to \( \Omega \left( \frac{n^2}{\log^6 n} \right) \).

They use the fact that the Boolean convolution can be reduced to Boolean cyclic convolution which can be reduced to certain Boolean sums which are related to circulant matrices (see also Julka, pp 386-380).