# On the Multiplicative Complexity of Symmetric Boolean Functions

Luís Brandão, Çağdaş Çalık, Meltem Sönmez Turan, René Peralta

National Institute of Standards and Technology (Gaithersburg, MD, USA)

The 3rd International Workshop on
Boolean Functions and their Applications (BFA)
June 19, 2018 (Loen, Norway)

Contact email: circuit_complexity@nist.gov

# Outline

# Outline

# Boolean functions and circuits

**We focus on Boolean functions (i.e., predicates)**

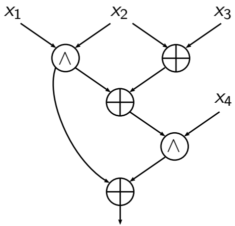
- $f : \{0,1\}^n \to \{0,1\}$ with $n$ bits of input and **1** bit of output.
- $\mathcal{B}_n$: set of $(2^{2^n})$ Boolean functions with $n$ input bits.

# Boolean functions and circuits

**We focus on Boolean functions (i.e., predicates)**

- $f : \{0,1\}^n \to \{0,1\}$ with $n$ bits of input and **1** bit of output.
- $\mathcal{B}_n$: set of $(2^{2^n})$ Boolean functions with $n$ input bits.

Boolean circuit: A combination of logic gates to compute functions.
(A directed acyclic graph of gates, with inputs as sources, and with outputs as sinks.)




Example gates (fanin 2)

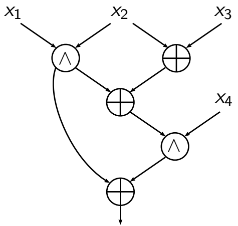
| input bits | output bits | |
|---|---|---|
| | AND ($\wedge$) | XOR ($\oplus$) |
| 00 | 0 | 0 |
| 01 | 0 | 1 |
| 10 | 0 | 1 |
| 11 | 1 | 0 |

# Boolean functions and circuits

**We focus on Boolean functions (i.e., predicates)**

- $f : \{0,1\}^n \to \{0,1\}$ with $n$ bits of input and **1** bit of output.
- $\mathcal{B}_n$: set of $(2^{2^n})$ Boolean functions with $n$ input bits.

Boolean circuit: A combination of logic gates to compute functions.
(A directed acyclic graph of gates, with inputs as sources, and with outputs as sinks.)




Example gates (fanin 2)

| input bits | output bits | |
|---|---|---|
| | AND ($\wedge$) | XOR ($\oplus$) |
| 00 | 0 | 0 |
| 01 | 0 | 1 |
| 10 | 0 | 1 |
| 11 | 1 | 0 |

- For nonlinear gates, we focus on AND gates with fanin 2.
- For linear gates, we focus on XOR gates with arbitrary fanin.

# Multiplicative complexity (MC)

**$c_\wedge(f)$: MC of a function $f$**

- min # nonlinear gates needed to implement $f$ by a Boolean circuit

# Multiplicative complexity (MC)

**$c_\wedge(f)$: MC of a function $f$**

- min # nonlinear gates needed to implement $f$ by a Boolean circuit
- equivalently*: min # `AND` ($\wedge$) gates over the basis $(\wedge, \oplus, 1)$

* (since any fanin-2 nonlinear gate can be replaced by one `AND` gate and $\oplus$'s and 1's)

# Multiplicative complexity (MC)

## $c_\wedge(f)$: MC of a function $f$

- min # nonlinear gates needed to implement $f$ by a Boolean circuit
- equivalently*: min # AND ($\wedge$) gates over the basis $(\wedge, \oplus, 1)$

* (since any fanin-2 nonlinear gate can be replaced by one AND gate and $\oplus$'s and 1's)

## Why useful to find circuits with minimal MC?

- Shorter secure multi-party computation and zero-knowledge proofs:
  - non-linear gates are expensive; linear gates are "for free"
- Resistance to side-channel attacks:
  - threshold protection of leakage from non-linear gates has high cost

# Multiplicative complexity (MC)

### $c_\wedge(f)$: MC of a function $f$

- min # nonlinear gates needed to implement $f$ by a Boolean circuit
- equivalently*: min # AND ($\wedge$) gates over the basis $(\wedge, \oplus, 1)$

* (since any fanin-2 nonlinear gate can be replaced by one AND gate and $\oplus$'s and 1's)

**Why useful to find circuits with minimal MC?**

- Shorter secure multi-party computation and zero-knowledge proofs:
  - non-linear gates are expensive; linear gates are "for free"
- Resistance to side-channel attacks:
  - threshold protection of leakage from non-linear gates has high cost

Notes:

- Finding the MC of a Boolean function is hard
- Almost all $f \in \mathcal{B}_n$ have MC $\geq 2^{n/2} - n - 1$; all $\leq 3 \cdot 2^{(n-1)/2} - \mathcal{O}_n$

# Symmetric Boolean functions

$\mathcal{S}_n$: set of ($2^{n+1}$) symmetric functions with $n$ input bits

- Output invariant when swapping any pair of input variables.
- Output depends only on the Hamming weight (HW) of the input.

# Symmetric Boolean functions

$\mathcal{S}_n$: set of ($2^{n+1}$) symmetric functions with $n$ input bits

- Output invariant when swapping any pair of input variables.
- Output depends only on the Hamming weight (HW) of the input.

Examples of classes of symmetric $n$-bit functions:

- Elementary symmetric ($\Sigma_k^n$): sum of all monomials of degree $k$ (Note: Any $f \in \mathcal{S}_n$ is a linear sum of $\Sigma_i^n$'s)
- Counting ($E_k^n$): 1 if and only if $HW(x) = k$
- Threshold ($T_k^n$): 1 if and only if $HW(x) \geq k$

# Symmetric Boolean functions

$\mathcal{S}_n$: set of ($2^{n+1}$) symmetric functions with $n$ input bits

- Output invariant when swapping any pair of input variables.
- Output depends only on the Hamming weight (HW) of the input.

Examples of classes of symmetric $n$-bit functions:

- Elementary symmetric ($\Sigma_k^n$): sum of all monomials of degree $k$ (Note: Any $f \in \mathcal{S}_n$ is a linear sum of $\Sigma_i^n$'s)
- Counting ($E_k^n$): 1 if and only if $HW(x) = k$
- Threshold ($T_k^n$): 1 if and only if $HW(x) \geq k$

Example function:

$Maj_3$ — majority bit out of three (outputs 1 iff at least two 1s in input):
$T_2^3 = (x_1 \wedge x_2) \oplus (x_1 \wedge x_3) \oplus (x_2 \wedge x_3)$

# Symmetric Boolean functions

$\mathcal{S}_n$: set of $(2^{n+1})$ symmetric functions with $n$ input bits

- Output invariant when swapping any pair of input variables.
- Output depends only on the Hamming weight (HW) of the input.

Examples of classes of symmetric $n$-bit functions:

- Elementary symmetric ($\Sigma_k^n$): sum of all monomials of degree $k$ (Note: Any $f \in \mathcal{S}_n$ is a linear sum of $\Sigma_i^n$'s)
- Counting ($E_k^n$): 1 if and only if $HW(x) = k$
- Threshold ($T_k^n$): 1 if and only if $HW(x) \geq k$

Example function:

$Maj_3$ — majority bit out of three (outputs 1 iff at least two 1s in input):
$T_2^3 = (x_1 \wedge x_2) \oplus (x_1 \wedge x_3) \oplus (x_2 \wedge x_3) = ((x_1 \oplus x_2) \wedge (x_1 \oplus x_3)) \oplus x_1$

# MC of symmetric functions

**Why care about the MC of functions in $\mathcal{S}_n$?**

- **Building blocks for other functions**
  Improvements for $\mathcal{S}_n$ may carry to non-symmetric functions.

# MC of symmetric functions

**Why care about the MC of functions in $\mathcal{S}_n$?**

- **Building blocks for other functions**
  Improvements for $\mathcal{S}_n$ may carry to non-symmetric functions.

  E.g.: sum of two $n$-bit integers, via $n$ applications of $Maj_3$.
  Three-to-one `AND` gate reduction leads to 2/3 communic. reduction in crypto protocols (e.g., ZK proof of bit-commitments of an integer sum).

# MC of symmetric functions

**Why care about the MC of functions in $\mathcal{S}_n$?**

- **Building blocks for other functions**
  Improvements for $\mathcal{S}_n$ may carry to non-symmetric functions.

  E.g.: sum of two $n$-bit integers, via $n$ applications of $Maj_3$.
  Three-to-one `AND` gate reduction leads to 2/3 communic. reduction in crypto protocols (e.g., ZK proof of bit-commitments of an integer sum).

- **Easier start-point for certain MC analyses?**

# MC of symmetric functions

**Why care about the MC of functions in $\mathcal{S}_n$?**

- **Building blocks for other functions**
  Improvements for $\mathcal{S}_n$ may carry to non-symmetric functions.

  E.g.: sum of two $n$-bit integers, via $n$ applications of $Maj_3$.
  Three-to-one `AND` gate reduction leads to 2/3 communic. reduction in crypto protocols (e.g., ZK proof of bit-commitments of an integer sum).

- **Easier start-point for certain MC analyses?**
  $\mathcal{S}_n$ has $2^{n+1}$ functions; $\mathcal{B}_n$ has $2^{2^n}$ functions.
  Compared with $\mathcal{B}_n$, can we more easily characterize MC for $\mathcal{S}_n$?

# Summary of new results in this presentation

# Summary of new results in this presentation

- Devise "twin" technique to analyze MC of symmetric functions

# Summary of new results in this presentation

- Devise "twin" technique to analyze MC of symmetric functions
- Answer two open questions: $c_\wedge(\Sigma_4^8) = 6$; $c_\wedge(E_4^8) = 6$

# Summary of new results in this presentation

- Devise "twin" technique to analyze MC of symmetric functions
- Answer two open questions: $c_\wedge(\Sigma_4^8) = 6$; $c_\wedge(E_4^8) = 6$
- Characterize MC of functions in $\mathcal{S}_n$, for up to $n = 10$ variables:
  $n \in \{7, 8, 9, 10\} \wedge f \in \mathcal{B}_n \Rightarrow c_\wedge(f) \leq n - 1$

# Outline

# Affine equivalence

**Affine equivalence class.** $f$ and $g$ (from $\mathcal{B}_n$) are affine equivalent ($f \sim g$) if

$$f(x) = g(Ax + a) + b \cdot x + c, \text{ where:}$$

- $A$ is a non-singular $n \times n$ matrix over $\mathbb{F}_2$;
- $x$, $a$ are $n$-length column vectors over $\mathbb{F}_2$;
- $b$ is a $n$-length row vector over $\mathbb{F}_2$.

# Affine equivalence

**Affine equivalence class.** $f$ and $g$ (from $\mathcal{B}_n$) are affine equivalent ($f \sim g$) if

$$f(x) = g(Ax + a) + b \cdot x + c, \text{ where:}$$

- $A$ is a non-singular $n \times n$ matrix over $\mathbb{F}_2$;
- $x$, $a$ are $n$-length column vectors over $\mathbb{F}_2$;
- $b$ is a $n$-length row vector over $\mathbb{F}_2$.

**MC of equivalence class.** Multiplicative complexity is invariant under affine transformations: $f \sim g \Rightarrow c_\wedge(f) = c_\wedge(g)$

# Affine equivalence

**Affine equivalence class.** $f$ and $g$ (from $\mathcal{B}_n$) are affine equivalent ($f \sim g$) if

$$f(x) = g(Ax + a) + b \cdot x + c, \text{ where:}$$

- $A$ is a non-singular $n \times n$ matrix over $\mathbb{F}_2$;
- $x$, $a$ are $n$-length column vectors over $\mathbb{F}_2$;
- $b$ is a $n$-length row vector over $\mathbb{F}_2$.

**MC of equivalence class.** Multiplicative complexity is invariant under affine transformations: $f \sim g \Rightarrow c_\wedge(f) = c_\wedge(g)$

| $n \backslash k$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | Total |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | – | – | – | – | – | – | 1 |
| 2 | 1 | 1 | – | – | – | – | – | 2 |
| 3 | 1 | 1 | 1 | – | – | – | – | 3 |
| 4 | 1 | 1 | 3 | 3 | – | – | – | 8 |
| 5 | 1 | 1 | 3 | 17 | 26 | – | – | 48 |
| 6 | 1 | 1 | 3 | 24 | 914 | 148,483 | 931 [ÇTP18] | 150,357 [Mai91] |

Table 1: number of classes per $n$ (#vars) and $k$ (MC)

# Max MC of Boolean Functions with $n \leq 6$

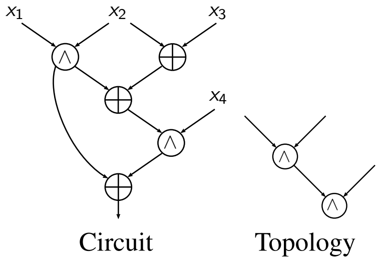
- $f \in \mathcal{B}_4$ (8 classes) $\Rightarrow c_\wedge(f) \leq 3$ [TP15]
- $f \in \mathcal{B}_5$ (48 classes) $\Rightarrow c_\wedge(f) \leq 4$ [TP15]
- $f \in \mathcal{B}_6$ (150,357 classes) $\rightarrow c_\wedge(f) \leq 6$ [ÇTP18]

# Max MC of Boolean Functions with $n \leq 6$

- $f \in \mathcal{B}_4$ (8 classes) $\Rightarrow c_\wedge(f) \leq 3$ [TP15]
- $f \in \mathcal{B}_5$ (48 classes) $\Rightarrow c_\wedge(f) \leq 4$ [TP15]
- $f \in \mathcal{B}_6$ (150,357 classes) $\rightarrow c_\wedge(f) \leq 6$ [ÇTP18]

## (Circuit) Topologies [CCFS15]

E.g.: $f = x_1x_2x_3 + x_1x_2 + x_1x_4 + x_2x_3 + x_4$

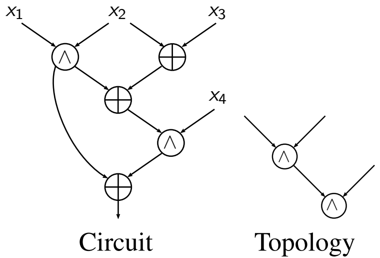


# Max MC of Boolean Functions with $n \leq 6$

- $f \in \mathcal{B}_4$ (8 classes) $\Rightarrow c_\wedge(f) \leq 3$ [TP15]
- $f \in \mathcal{B}_5$ (48 classes) $\Rightarrow c_\wedge(f) \leq 4$ [TP15]
- $f \in \mathcal{B}_6$ (150,357 classes) $\rightarrow c_\wedge(f) \leq 6$ [ÇTP18]

## (Circuit) Topologies [CCFS15]

E.g.: $f = x_1x_2x_3 + x_1x_2 + x_1x_4 + x_2x_3 + x_4$




## Method [ÇTP18]

- Iterate over all topologies with 1, 2, 3, … AND gates

| # AND gates | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| # topologies | 1 | 2 | 8 | 84 | 3,170 | 475,248 |

- For each topology, mark the classes generated by circuits.
- Max MC for $n = 6$ is found when all classes are marked.

# Some prior results on the MC of symmetric functions

- Functions in $\mathcal{B}_n$ have circuits with $\leq n + 3\sqrt{n}$ `AND` gates [BPP00]
- The MC of an $n$-bit *nonlinear* symmetric function is at least $\lfloor \frac{n}{2} \rfloor$ [BP08]
- The MC of $\Sigma_2^n$ is $\lfloor \frac{n}{2} \rfloor$; the MC of $\Sigma_3^n$ is $\lceil \frac{n}{2} \rceil$, ... [BP08]

Table A.1 from [BP08]:
MC complexity of the elementary symm $\Sigma_i^n$

| $n \backslash i$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 3 | 1 | 2 | – | – | – | – | – |
| 4 | 2 | 2 | 3 | – | – | – | – |
| 5 | 2 | 3 | 3 | 4 | – | – | – |
| 6 | 3 | 3 | 4 | 4 | 5 | – | – |
| 7 | 3 | 4 | 4 | 5 | 5 | 6 | – |
| 8 | 4 | 4 | **5–6** | 5 | 6 | 6 | 7 |

Table A.3 from [BP08]:
MC complexity of the counting function $E_i^n$

| $n \backslash i$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 3 | 2 | 2 | 2 | 2 | – | – | – | – | – |
| 4 | 3 | 2 | 2 | 2 | 3 | – | – | – | – |
| 5 | 4 | 4 | 3 | 3 | 4 | 4 | – | – | – |
| 6 | 5 | 4 | 3 | 3 | 5 | 4 | 5 | – | – |
| 7 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | – |
| 8 | 7 | 6 | 6 | 6 | **6–7** | 6 | 6 | 6 | 7 |

# Some prior results on the MC of symmetric functions

- Functions in $\mathcal{B}_n$ have circuits with $\leq n + 3\sqrt{n}$ AND gates [BPP00]
- The MC of an $n$-bit *nonlinear* symmetric function is at least $\lfloor \frac{n}{2} \rfloor$ [BP08]
- The MC of $\Sigma_2^n$ is $\lfloor \frac{n}{2} \rfloor$; the MC of $\Sigma_3^n$ is $\lceil \frac{n}{2} \rceil$, ... [BP08]

Table A.1 from [BP08]:
MC complexity of the elementary symm $\Sigma_i^n$

| $n \backslash i$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 3 | 1 | 2 | – | – | – | – | – |
| 4 | 2 | 2 | 3 | – | – | – | – |
| 5 | 2 | 3 | 3 | 4 | – | – | – |
| 6 | 3 | 3 | 4 | 4 | 5 | – | – |
| 7 | 3 | 4 | 4 | 5 | 5 | 6 | – |
| 8 | 4 | 4 | **5–6** | 5 | 6 | 6 | 7 |

Table A.3 from [BP08]:
MC complexity of the counting function $E_i^n$

| $n \backslash i$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 3 | 2 | 2 | 2 | 2 | – | – | – | – | – |
| 4 | 3 | 2 | 2 | 2 | 3 | – | – | – | – |
| 5 | 4 | 4 | 3 | 3 | 4 | 4 | – | – | – |
| 6 | 5 | 4 | 3 | 3 | 5 | 4 | 5 | – | – |
| 7 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | – |
| 8 | 7 | 6 | 6 | 6 | **6–7** | 6 | 6 | 6 | 7 |

**Two concrete open questions:**

1. What is the MC of $\Sigma_4^8$? (Is it 5 or 6?)
2. What is the MC of $E_4^8$? (Is it 6 or 7?)

# Outline

# Boolean Functions with Twin Variables

(Towards facilitating the analysis of symmetric Boolean functions)

Definition (twin variables): Let $f(x) = \boldsymbol{x_i x_j} g(x) + h(x)$, where $g$ and $h$ do not depend on $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$. Then, $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are called **twins** in $f$.

$\mathcal{T}_n$: set of functions in $\mathcal{B}_n$ and with twins.

# Boolean Functions with Twin Variables

(Towards facilitating the analysis of symmetric Boolean functions)

Definition (twin variables): Let $f(x) = \boldsymbol{x_i x_j} g(x) + h(x)$, where $g$ and $h$ do not depend on $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$. Then, $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are called **twins** in $f$.

$\mathcal{T}_n$: set of functions in $\mathcal{B}_n$ and with twins.

Example: $f(\boldsymbol{x_1}, x_2, x_3, \boldsymbol{x_4}) = \boldsymbol{x_1 x_4}(1 + x_2 + x_2 x_3) + x_3$

# Boolean Functions with Twin Variables

(Towards facilitating the analysis of symmetric Boolean functions)

Definition (twin variables): Let $f(x) = \boldsymbol{x_i x_j} g(x) + h(x)$, where $g$ and $h$ do not depend on $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$. Then, $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are called **twins** in $f$.

$\mathcal{T}_n$: set of functions in $\mathcal{B}_n$ and with twins.

Example: $f(\boldsymbol{x_1}, x_2, x_3, \boldsymbol{x_4}) = \boldsymbol{x_1 x_4}(1 + x_2 + x_2 x_3) + x_3$

---

What can we do with this?

# Boolean Functions with Twin Variables

(Towards facilitating the analysis of symmetric Boolean functions)

Definition (twin variables): Let $f(x) = x_i x_j g(x) + h(x)$, where $g$ and $h$ do not depend on $x_i$ and $x_j$. Then, $x_i$ and $x_j$ are called **twins** in $f$.

$\mathcal{T}_n$: set of functions in $\mathcal{B}_n$ and with twins.

Example: $f(x_1, x_2, x_3, x_4) = x_1 x_4 (1 + x_2 + x_2 x_3) + x_3$

---

What can we do with this?

Replace $x_1 x_n$ by $y_1$ and let $f'(y_1, x_2, ..., x_{n-1}) = f(x_1, x_2, ..., x_n)$.

# Boolean Functions with Twin Variables

(Towards facilitating the analysis of symmetric Boolean functions)

Definition (twin variables): Let $f(x) = \boldsymbol{x_i x_j} g(x) + h(x)$, where $g$ and $h$ do not depend on $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$. Then, $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are called **twins** in $f$.

$\mathcal{T}_n$: set of functions in $\mathcal{B}_n$ and with twins.

Example: $f(\boldsymbol{x_1}, x_2, x_3, \boldsymbol{x_4}) = \boldsymbol{x_1 x_4}(1 + x_2 + x_2 x_3) + x_3$

---

What can we do with this?

Replace $\boldsymbol{x_1 x_n}$ by $y_1$ and let $f'(y_1, x_2, ..., x_{n-1}) = f(\boldsymbol{x_1}, x_2, ..., \boldsymbol{x_n})$.

Fact: $c_\wedge(\boldsymbol{f}) \leq 1 + c_\wedge(f')$.

# Boolean Functions with Twin Variables

(Towards facilitating the analysis of symmetric Boolean functions)

Definition (twin variables): Let $f(x) = \boldsymbol{x_i x_j} g(x) + h(x)$, where $g$ and $h$ do not depend on $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$. Then, $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are called **twins** in $f$.

$\mathcal{T}_n$: set of functions in $\mathcal{B}_n$ and with twins.

Example: $f(\boldsymbol{x_1}, x_2, x_3, \boldsymbol{x_4}) = \boldsymbol{x_1 x_4}(1 + x_2 + x_2 x_3) + x_3$

---

What can we do with this?

Replace $\boldsymbol{x_1 x_n}$ by $y_1$ and let $f'(y_1, x_2, \dots, x_{n-1}) = f(\boldsymbol{x_1}, x_2, \dots, \boldsymbol{x_n})$.

Fact: $c_\wedge(\boldsymbol{f}) \leq 1 + c_\wedge(f')$. Twin Conjecture: $c_\wedge(\boldsymbol{f}) = 1 + c_\wedge(f')$

# Boolean Functions with Twin Variables

(Towards facilitating the analysis of symmetric Boolean functions)

Definition (twin variables): Let $f(x) = \boldsymbol{x_i x_j} g(x) + h(x)$, where $g$ and $h$ do not depend on $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$. Then, $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are called **twins** in $f$.

$\mathcal{T}_n$: set of functions in $\mathcal{B}_n$ and with twins.

Example: $f(\boldsymbol{x_1}, x_2, x_3, \boldsymbol{x_4}) = \boldsymbol{x_1 x_4}(1 + x_2 + x_2 x_3) + x_3$

---

What can we do with this?

Replace $\boldsymbol{x_1 x_n}$ by $y_1$ and let $f'(y_1, x_2, ..., x_{n-1}) = f(\boldsymbol{x_1}, x_2, ..., \boldsymbol{x_n})$.

Fact: $c_\wedge(\boldsymbol{f}) \leq 1 + c_\wedge(f')$. Twin Conjecture: $c_\wedge(\boldsymbol{f}) = 1 + c_\wedge(f')$

Result: Analyzing $c_\wedge(\boldsymbol{f} \in \mathcal{T}_n)$ is reduced to analyzing $c_\wedge(f' \in \mathcal{B}_{n-1})$

# Boolean Functions with Twin Variables

(Towards facilitating the analysis of symmetric Boolean functions)

Definition (twin variables): Let $f(x) = \boldsymbol{x_i x_j} g(x) + h(x)$, where $g$ and $h$ do not depend on $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$. Then, $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are called **twins** in $f$.

$\mathcal{T}_n$: set of functions in $\mathcal{B}_n$ and with twins.

Example: $f(\boldsymbol{x_1}, x_2, x_3, \boldsymbol{x_4}) = \boldsymbol{x_1 x_4}(1 + x_2 + x_2x_3) + x_3$

---

What can we do with this?

Replace $\boldsymbol{x_1 x_n}$ by $y_1$ and let $f'(y_1, x_2, ..., x_{n-1}) = f(\boldsymbol{x_1}, x_2, ..., \boldsymbol{x_n})$.

Fact: $c_\wedge(\boldsymbol{f}) \leq 1 + c_\wedge(f')$. Twin Conjecture: $c_\wedge(\boldsymbol{f}) = 1 + c_\wedge(f')$

Result: Analyzing $c_\wedge(\boldsymbol{f} \in \mathcal{T}_n)$ is reduced to analyzing $c_\wedge(f' \in \mathcal{B}_{n-1})$

But what about symmetric functions ($\mathcal{S}_n$)? (next slide)

# Symmetric Functions and Twin Variables

Theorem: Any symmetric Boolean function ($f \in \mathcal{S}_n$) is affine equivalent to a Boolean function ($f' \in \mathcal{T}_n$) with twins.

# Symmetric Functions and Twin Variables

Theorem: Any symmetric Boolean function ($f \in \mathcal{S}_n$) is affine equivalent to a Boolean function ($f' \in \mathcal{T}_n$) with twins.

Example with elementary symmetric function:

- $f = \Sigma_2^3 = x_1x_2 \oplus x_1x_3 \oplus x_2x_3 = (x_1 \oplus x_3)(x_2 \oplus x_3) \oplus x_3$
- Var transform ($\boldsymbol{\tau}$): $x_1 \to A + C$; $x_2 \to A + B$; $x_3 \to A + B + C + 1$
- Result: $\Sigma_2^3 = (B \oplus 1)(C \oplus 1) \oplus A \oplus B \oplus C \oplus 1 = A \oplus \boldsymbol{BC}$

# Symmetric Functions and Twin Variables

Theorem: Any symmetric Boolean function ($f \in \mathcal{S}_n$) is affine equivalent to a Boolean function ($f' \in \mathcal{T}_n$) with twins.

Example with elementary symmetric function:

- $f = \Sigma_2^3 = x_1x_2 \oplus x_1x_3 \oplus x_2x_3 = (x_1 \oplus x_3)(x_2 \oplus x_3) \oplus x_3$
- Var transform ($\boldsymbol{\tau}$): $x_1 \to A + C$; $x_2 \to A + B$; $x_3 \to A + B + C + 1$
- Result: $\Sigma_2^3 = (B \oplus 1)(C \oplus 1) \oplus A \oplus B \oplus C \oplus 1 = A \oplus \boldsymbol{BC}$

Intuition:

- For any $n$ and $k$, $\boldsymbol{\tau}$ applied to $\Sigma_k^n$ combines $\boldsymbol{B}$ and $\boldsymbol{C}$ as **twins**

# Symmetric Functions and Twin Variables

Theorem: Any symmetric Boolean function ($f \in \mathcal{S}_n$) is affine equivalent to a Boolean function ($f' \in \mathcal{T}_n$) with twins.

Example with elementary symmetric function:

- $f = \Sigma_2^3 = x_1x_2 \oplus x_1x_3 \oplus x_2x_3 = (x_1 \oplus x_3)(x_2 \oplus x_3) \oplus x_3$
- Var transform ($\boldsymbol{\tau}$): $x_1 \to A + C$; $x_2 \to A + B$; $x_3 \to A + B + C + 1$
- Result: $\Sigma_2^3 = (B \oplus 1)(C \oplus 1) \oplus A \oplus B \oplus C \oplus 1 = A \oplus \boldsymbol{BC}$

Intuition:

- For any $n$ and $k$, $\boldsymbol{\tau}$ applied to $\Sigma_k^n$ combines $\boldsymbol{B}$ and $\boldsymbol{C}$ as **twins**
- Any $f \in \mathcal{S}_n$ is a sum of elementary symmetric functions ($\Sigma_i^n$)

# Symmetric Functions and Twin Variables

Theorem: Any symmetric Boolean function ($f \in \mathcal{S}_n$) is affine equivalent to a Boolean function ($f' \in \mathcal{T}_n$) with twins.

Example with elementary symmetric function:

- $f = \Sigma_2^3 = x_1x_2 \oplus x_1x_3 \oplus x_2x_3 = (x_1 \oplus x_3)(x_2 \oplus x_3) \oplus x_3$
- Var transform ($\boldsymbol{\tau}$): $x_1 \to A + C$; $x_2 \to A + B$; $x_3 \to A + B + C + 1$
- Result: $\Sigma_2^3 = (B \oplus 1)(C \oplus 1) \oplus A \oplus B \oplus C \oplus 1 = A \oplus \boldsymbol{BC}$

Intuition:

- For any $n$ and $k$, $\boldsymbol{\tau}$ applied to $\Sigma_k^n$ combines $\boldsymbol{B}$ and $\boldsymbol{C}$ as **twins**
- Any $f \in \mathcal{S}_n$ is a sum of elementary symmetric functions ($\Sigma_i^n$)
- Each disjoint var triplet becomes one **twin pair** and another variable

# Symmetric Functions and Twin Variables

Theorem: Any symmetric Boolean function ($f \in \mathcal{S}_n$) is affine equivalent to a Boolean function ($f' \in \mathcal{T}_n$) with twins.

Example with elementary symmetric function:

- $f = \Sigma_2^3 = x_1x_2 \oplus x_1x_3 \oplus x_2x_3 = (x_1 \oplus x_3)(x_2 \oplus x_3) \oplus x_3$
- Var transform ($\boldsymbol{\tau}$): $x_1 \to A + C$; $x_2 \to A + B$; $x_3 \to A + B + C + 1$
- Result: $\Sigma_2^3 = (B \oplus 1)(C \oplus 1) \oplus A \oplus B \oplus C \oplus 1 = A \oplus \boldsymbol{BC}$

Intuition:

- For any $n$ and $k$, $\boldsymbol{\tau}$ applied to $\Sigma_k^n$ combines $\boldsymbol{B}$ and $\boldsymbol{C}$ as **twins**
- Any $f \in \mathcal{S}_n$ is a sum of elementary symmetric functions ($\Sigma_i^n$)
- Each disjoint var triplet becomes one **twin pair** and another variable
- For $c_\wedge(\cdot)$ analysis, each **twin pair** is replaced by a new variable

# Symmetric Functions and Twin Variables

Theorem: Any symmetric Boolean function ($f \in \mathcal{S}_n$) is affine equivalent to a Boolean function ($f' \in \mathcal{T}_n$) with twins.

Example with elementary symmetric function:

- $f = \Sigma_2^3 = x_1x_2 \oplus x_1x_3 \oplus x_2x_3 = (x_1 \oplus x_3)(x_2 \oplus x_3) \oplus x_3$
- Var transform ($\boldsymbol{\tau}$): $x_1 \to A + C$; $x_2 \to A + B$; $x_3 \to A + B + C + 1$
- Result: $\Sigma_2^3 = (B \oplus 1)(C \oplus 1) \oplus A \oplus B \oplus C \oplus 1 = A \oplus \boldsymbol{BC}$

Intuition:

- For any $n$ and $k$, $\boldsymbol{\tau}$ applied to $\Sigma_k^n$ combines $\boldsymbol{B}$ and $\boldsymbol{C}$ as **twins**
- Any $f \in \mathcal{S}_n$ is a sum of elementary symmetric functions ($\Sigma_i^n$)
- Each disjoint var triplet becomes one **twin pair** and another variable
- For $c_\wedge(\cdot)$ analysis, each **twin pair** is replaced by a new variable

Result: $f \in \mathcal{S}_n$ is mapped to $f' \in \mathcal{B}_{n-\lfloor n/3 \rfloor}$

# Symmetric Functions and Twin Variables

Theorem: Any symmetric Boolean function ($f \in \mathcal{S}_n$) is affine equivalent to a Boolean function ($f' \in \mathcal{T}_n$) with twins.

Example with elementary symmetric function:

- $f = \Sigma_2^3 = x_1x_2 \oplus x_1x_3 \oplus x_2x_3 = (x_1 \oplus x_3)(x_2 \oplus x_3) \oplus x_3$
- Var transform ($\tau$): $x_1 \to A + C$; $x_2 \to A + B$; $x_3 \to A + B + C + 1$
- Result: $\Sigma_2^3 = (B \oplus 1)(C \oplus 1) \oplus A \oplus B \oplus C \oplus 1 = A \oplus \boldsymbol{BC}$

Intuition:

- For any $n$ and $k$, $\tau$ applied to $\Sigma_k^n$ combines $\boldsymbol{B}$ and $\boldsymbol{C}$ as **twins**
- Any $f \in \mathcal{S}_n$ is a sum of elementary symmetric functions ($\Sigma_i^n$)
- Each disjoint var triplet becomes one **twin pair** and another variable
- For $c_\wedge(\cdot)$ analysis, each **twin pair** is replaced by a new variable

Result: $f \in \mathcal{S}_n$ is mapped to $f' \in \mathcal{B}_{n-\lfloor n/3 \rfloor}$; and $c_\wedge(f) \leq \lfloor n/3 \rfloor + c_\wedge(f')$

# Symmetric Functions and Twin Variables

Theorem: Any symmetric Boolean function ($f \in \mathcal{S}_n$) is affine equivalent to a Boolean function ($f' \in \mathcal{T}_n$) with twins.

Example with elementary symmetric function:

- $f = \Sigma_2^3 = x_1x_2 \oplus x_1x_3 \oplus x_2x_3 = (x_1 \oplus x_3)(x_2 \oplus x_3) \oplus x_3$
- Var transform ($\tau$): $x_1 \to A + C$; $x_2 \to A + B$; $x_3 \to A + B + C + 1$
- Result: $\Sigma_2^3 = (B \oplus 1)(C \oplus 1) \oplus A \oplus B \oplus C \oplus 1 = A \oplus \boldsymbol{BC}$

Intuition:

- For any $n$ and $k$, $\tau$ applied to $\Sigma_k^n$ combines $\boldsymbol{B}$ and $\boldsymbol{C}$ as **twins**
- Any $f \in \mathcal{S}_n$ is a sum of elementary symmetric functions ($\Sigma_i^n$)
- Each disjoint var triplet becomes one **twin pair** and another variable
- For $c_\wedge(\cdot)$ analysis, each **twin pair** is replaced by a new variable

Result: $f \in \mathcal{S}_n$ is mapped to $f' \in \mathcal{B}_{n-\lfloor n/3 \rfloor}$; and $c_\wedge(f) \le \lfloor n/3 \rfloor + c_\wedge(f')$

Example: analysis of $f \in \mathcal{S}_8$ becomes analysis of $f' \in \mathcal{B}_{\boldsymbol{6}}$

# Multiplicative Complexity of $E_4^8$ and $\Sigma_4^8$

Using the Twin technique:

- Reduce # variables (from 8 to 6): $f \in \left\{E_4^8, \Sigma_4^8\right\} \to f' \in \mathcal{B}_6$
- Find MC-optimal circuit for $f' \in \mathcal{B}_6$
- $c_\wedge(f) \leq c_\wedge(f') + 2$

# Multiplicative Complexity of $E_4^8$ and $\Sigma_4^8$

Using the Twin technique:

- Reduce # variables (from 8 to 6): $f \in \{E_4^8, \Sigma_4^8\} \to f' \in \mathcal{B}_6$
- Find MC-optimal circuit for $f' \in \mathcal{B}_6$
- $c_\wedge(f) \leq c_\wedge(f') + 2$

Case $f = E_4^8$ (counting function):

- It was known that $c_\wedge(f) \in \{6, 7\}$
- We find that $c_\wedge$(f')=4
- If follows that $c_\wedge(f) = 4 + 2 = 6$

# Multiplicative Complexity of $E_4^8$ and $\Sigma_4^8$

Using the Twin technique:

- Reduce # variables (from 8 to 6): $f \in \left\{E_4^8, \Sigma_4^8\right\} \to f' \in \mathcal{B}_6$
- Find MC-optimal circuit for $f' \in \mathcal{B}_6$
- $c_\wedge(f) \leq c_\wedge(f') + 2$

Case $f = E_4^8$ (counting function):

- It was known that $c_\wedge(f) \in \{6, 7\}$
- We find that $c_\wedge$(f')=4
- If follows that $c_\wedge(f) = 4 + 2 = 6$

Case $f = \Sigma_4^8$ (elementary symmetric function):

- It was known that $c_\wedge(f) \in \{5, 6\}$
- (Cheap) If twin-conj true: $c_\wedge(f') = 4$ directly implies $c_\wedge(f) = 4+2 = 6$
- (Expensive) No 5-AND topology can generate $f$, hence $c_\wedge(f) = 6$

# Transformation and SLPs (just for a glimpse)

Affine transformation from $f \in \mathcal{S}_8$ to $f' \in \mathcal{T}_6$:

- $(x_1, x_2, x_8) \to (x_1 \oplus x_2 \oplus x_8 \oplus 1,\ \ x_2 \oplus x_8 \oplus 1,\ \ x_1 \oplus x_2 \oplus 1)$
- $(x_3, x_4, x_7) \to (x_3 \oplus x_4 \oplus x_7 \oplus 1,\ \ x_4 \oplus x_7 \oplus 1,\ \ x_3 \oplus x_4 \oplus 1)$
- $(x_5, x_6) \to (x_5, x_6)$

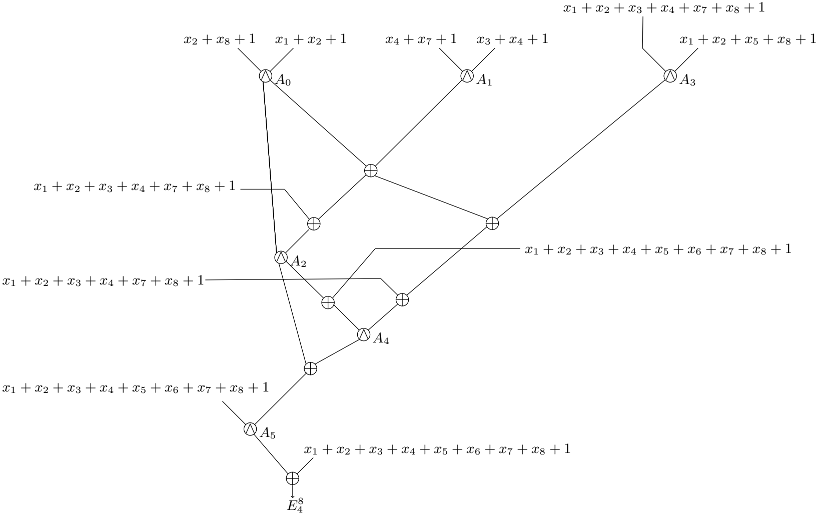
SLP for $f = E_4^8$ (counting function):

$a_0 = (1 \oplus x_2 \oplus x_8) \wedge (1 \oplus x_1 \oplus x_2)$
$a_1 = (1 \oplus x_4 \oplus x_7) \wedge (1 \oplus x_3 \oplus x_4)$
$a_2 = (a_0 \oplus a_1 \oplus 1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_7 \oplus x_8) \wedge (a_0)$
$a_3 = (1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_7 \oplus x_8) \wedge (1 \oplus x_1 \oplus x_2 \oplus x_5 \oplus x_8)$
$a_4 = (a_2 \oplus 1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_8) \wedge (a_0 \oplus a_1 \oplus a_3 \oplus 1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_7 \oplus x_8)$
$a_5 = (1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_8) \wedge (a_2 \oplus a_4)$
$f = a_5 \oplus 1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_8$

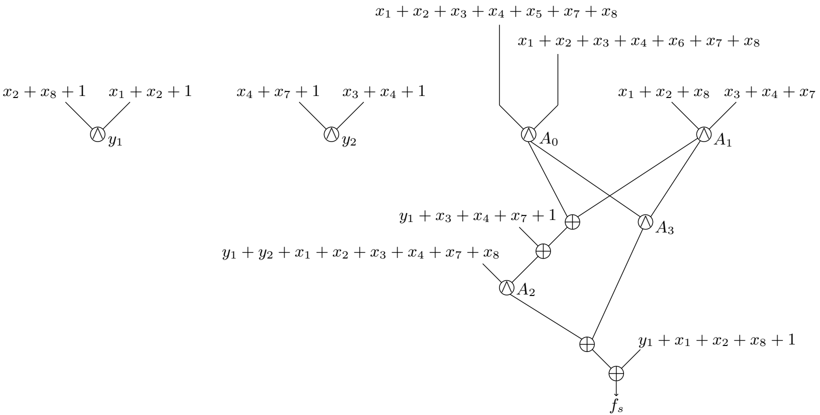
SLP for $f = \Sigma_4^8$ (elementary symmetric function):

$a_0 = (1 \oplus x_2 \oplus x_8) \wedge (1 \oplus x_1 \oplus x_2)$
$a_1 = (1 \oplus x_4 \oplus x_7) \wedge (1 \oplus x_3 \oplus x_4)$
$a_2 = (x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_7 \oplus x_8) \wedge (x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_7 \oplus x_8)$
$a_3 = (x_1 \oplus x_2 \oplus x_8) \wedge (x_3 \oplus x_4 \oplus x_7)$
$a_4 = (a_0 \oplus a_1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_7 \oplus x_8) \wedge (a_0 \oplus a_2 \oplus a_3 \oplus 1 \oplus x_3 \oplus x_4 \oplus x_7)$
$a_5 = (a_2) \wedge (a_3)$
$f = a_0 \oplus a_4 \oplus a_5 \oplus 1 \oplus x_1 \oplus x_2 \oplus x_8$

# MC-optimal circuit for $E_4^8$ (just for a glimpse)



# MC-optimal circuit for $\Sigma_4^8$ (just for a glimpse)



# MC of Symmetric Functions with $n \leq 10$

Prior lemma ([BP08]): $c_{\wedge}(f \in \mathcal{S}_7) \leq 8$

Using the *twin* technique and the ability to find MC for $f \in \mathcal{B}_{n \leq 6}$, we get:

$$n \in \{7, 8, 9, 10\} \Rightarrow c_{\wedge}(f \in \mathcal{S}_n) \leq n - 1$$

# MC of Symmetric Functions with $n \leq 10$

Prior lemma ([BP08]): $c_\wedge(f \in \mathcal{S}_7) \leq 8$

Using the *twin* technique and the ability to find MC for $f \in \mathcal{B}_{n \leq 6}$, we get:

$$n \in \{7, 8, 9, 10\} \Rightarrow c_\wedge(f \in \mathcal{S}_n) \leq n - 1$$

| | # Symmetric Boolean Functions | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $n\backslash k$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Total | $*$ |
| 1 | 4 | | | | | | | | | | 4 | |
| 2 | 4 | 4 | | | | | | | | | 8 | |
| 3 | 4 | 4 | 8 | | | | | | | | 16 | |
| 4 | 4 | | 12 | 16 | | | | | | | 32 | ✓ |
| 5 | 4 | | 4 | 24 | 32 | | | | | | 64 | |
| 6 | 4 | | | 12 | 48 | 64 | | | | | 128 | |
| 7 | 4 | | | 4 | 16 | 104 | 128 | | | | 256 | Twin |
| 8 | 4 | | | | 12 | 16 | 224 | 256 | | | 512 | conj. |
| 9 | 4 | | | | 4 | 8 | 48 | 448 | 512 | | 1024 | (TC) |
| 10 | 4 | | | | | 12 | 0 | 96 | 712 | 1224 | 2048 | |

Legend: $n$ (# input vars); $k$ (# `AND` gates); TC (twin conjecture)

# MC of Symmetric Functions with $n \leq 10$

Prior lemma ([BP08]): $c_\wedge(f \in \mathcal{S}_7) \leq 8$

Using the *twin* technique and the ability to find MC for $f \in \mathcal{B}_{n \leq 6}$, we get:

$$n \in \{7, 8, 9, 10\} \Rightarrow c_\wedge(f \in \mathcal{S}_n) \leq n - 1$$

| | # Symmetric Boolean Functions | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $n \backslash k$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Total | $*$ |
| 1 | 4 | | | | | | | | | | 4 | ✓ |
| 2 | 4 | 4 | | | | | | | | | 8 | |
| 3 | 4 | 4 | 8 | | | | | | | | 16 | |
| 4 | 4 | | 12 | 16 | | | | | | | 32 | |
| 5 | 4 | | 4 | 24 | 32 | | | | | | 64 | |
| 6 | 4 | | | 12 | 48 | 64 | | | | | 128 | |
| 7 | 4 | | | 4 | 16 | 104 | 128 | | | | 256 | Twin conj. (TC) |
| 8 | 4 | | | | 12 | 16 | 224 | 256 | | | 512 | |
| 9 | 4 | | | | 4 | 8 | 48 | 448 | 512 | | 1024 | |
| 10 | 4 | | | | | 12 | 0 | 96 | 712 | 1224 | 2048 | |

Legend: $n$ (# input vars); $k$ (# AND gates); TC (twin conjecture)

$*$: if TC holds, all results are exact; otherwise some MCs might be smaller by 1.

# MC of Symmetric Functions with $n \leq 10$

Prior lemma ([BP08]): $c_\wedge(f \in \mathcal{S}_7) \leq 8$

Using the *twin* technique and the ability to find MC for $f \in \mathcal{B}_{n \leq 6}$, we get:

$$n \in \{7, 8, 9, 10\} \Rightarrow c_\wedge(f \in \mathcal{S}_n) \leq n - 1$$

| | # Symmetric Boolean Functions | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $n \backslash k$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Total | $*$ |
| 1 | 4 | | | | | | | | | | 4 | ✓ |
| 2 | 4 | 4 | | | | | | | | | 8 | |
| 3 | 4 | 4 | 8 | | | | | | | | 16 | |
| 4 | 4 | | 12 | 16 | | | | | | | 32 | |
| 5 | 4 | | 4 | 24 | 32 | | | | | | 64 | |
| 6 | 4 | | | 12 | 48 | 64 | | | | | 128 | |
| 7 | 4 | | | 4 | 16 | 104 | 128 | | | | 256 | Twin conj. (TC) |
| 8 | 4 | | | | 12 | 16 | 224 | 256 | | | 512 | |
| 9 | 4 | | | | 4 | 8 | 48 | 448 | 512 | | 1024 | |
| 10 | 4 | | | | | 12 | 0 | 96 | 712 | 1224 | 2048 | |

Legend: $n$ (# input vars); $k$ (# `AND` gates); TC (twin conjecture)

$*$: if TC holds, all results are exact; otherwise some MCs might be smaller by 1.

Note: all cells are multiple of 4, since MC is independent of sum by $\Sigma_0^n$ and $\Sigma_1^n$

# Outline

# Summary and further research

**Summary**

- Studied the MC of symmetric functions
- Devised the twin method for reducing # variables
- Answered two open questions: $c_\wedge(\Sigma^8_4) = 6$; $c_\wedge(E^8_4) = 6$
- Gave upper bounds (conjectured tight) for up to $n = 10$ variables
- (Not shown here) new non-tight upper-bounds for higher $n$

# Summary and further research

## Summary

- Studied the MC of symmetric functions
- Devised the twin method for reducing # variables
- Answered two open questions: $c_\wedge(\Sigma^8_4) = 6$; $c_\wedge(E^8_4) = 6$
- Gave upper bounds (conjectured tight) for up to $n = 10$ variables
- (Not shown here) new non-tight upper-bounds for higher $n$

## Further research

- Prove (or disprove?) the *twin* conjecture
- How to enable tight characterizations for higher $n$?

# Summary and further research

**Summary**

- Studied the MC of symmetric functions
- Devised the twin method for reducing # variables
- Answered two open questions: $c_\wedge(\Sigma^8_4) = 6$; $c_\wedge(E^8_4) = 6$
- Gave upper bounds (conjectured tight) for up to $n = 10$ variables
- (Not shown here) new non-tight upper-bounds for higher $n$

**Further research**

- Prove (or disprove?) the *twin* conjecture
- How to enable tight characterizations for higher $n$?

Thank you for your attention!

# References


[Sch89] C.P. Schnorr, *The multiplicative complexity of Boolean functions*, in: Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 6th International Conference, LNCS, vol. 357, pp. 4558. Springer, 1989. doi:10.1007/3-540-51083-4_47

[Mai91] J.A. Maiorana, *A Classification of the Cosets of the Reed-Muller Code* $\mathcal{R}(1, 6)$, in: Mathematics of Computation, vol. 57, no. 195, pp. 403–414. July 1991. doi:10.2307/2938682

[BPP00] J. Boyar, R. Peralta, D. Pochuev, *On the multiplicative complexity of Boolean functions over the basis* $(\wedge, \oplus, 1)$, Theoretical Computer Science, 2000 - Elsevier doi:10.1016/S0304-3975(99)00182-6

[Fin04] M.G. Find,*On the Complexity of Computing Two Nonlinearity Measures*, Computer Science — Theory and Applications, pp. 167–175, 2014, Springer. doi:10.1007/978-3-319-06686-8_13

[BP08] J. Boyar, R. Peralta, *Tight bounds for the multiplicative complexity of symmetric functions*, Theoretical Computer Science 396, (2008), pp. 223-246. doi:10.1016/j.tcs.2008.01.030

[TP15] M.S. Turan, R. Peralta, *The Multiplicative Complexity of Boolean Functions on four and five variables*, International Workshop on Lightweight Cryptography for Security and Privacy, 2014. doi:10.1007/978-3-319-16363-5_2

[CCFS15] Michael Codisha, Luís Cruz-Filipe, Michael Franka, Peter Schneider-Kamp *When Six Gates are Not Enough*, Jr. CoRR, 2015. arXiv:1508.05737

[ÇTP18] Ç. Çalık, M. S. Turan, R. Peralta, *The multiplicative complexity of 6-variable Boolean functions*, R. Cryptogr. Commun. (2018). doi:10.1007/s12095-018-0297-2