Multiplicative Complexity of Boolean Functions

Meltem Sönmez Turan

National Institute of Standards and Technology, Gaithersburg, MD

WPI ECE Online Graduate Seminar Lecture
March 17 2021

In this presentation, ...

- Overview - Computer Security Division of NIST
- Circuit Complexity Problem
- Multiplicative Complexity
- Three results
  - Multiplicative Complexity of Boolean functions with $n \leq 6$
  - Boolean functions with Multiplicative Complexity 1,2,3 and 4
  - Multiplicative Complexity of Symmetric Boolean Functions
- Research Directions

National Institute of Standards and Technology

- Non-regulatory federal agency within U.S. Department of Commerce.
- Founded in 1901, known as the National Bureau of Standards (NBS) prior to 1988.
- Headquarters in Gaithersburg, Maryland, and laboratories in Boulder, Colorado.
- Employs around 6,000 employees and associates.

Computer Security Division (CSD) conducts research, development and outreach necessary to provide standards and guidelines, mechanisms, tools, metrics and practices to protect nation's information and information systems.

CSD Publications

- Federal Information Processing Standards (FIPS): Specify approved crypto standards.
- NIST Special Publications (SPs): Guidelines, technical specifications, recommendations and reference materials, including multiple sub-series.
- NIST Internal or Interagency Reports (NISTIR): Reports of research findings, including background information for FIPS and SPs.
- NIST Information Technology Laboratory (ITL) Bulletins: Monthly overviews of NIST's security and privacy publications, programs and projects.

Standard Development Process

- International "competitions": Engage community through an open competition (e.g., AES, SHA-3, PQC, Lightweight Crypto).
- Adoption of existing standards: Collaboration with accredited standards organizations (e.g., RSA, HMAC).
- Open call for proposals: Ongoing open invitation (e.g., modes of operations).
- Development of new algorithms: if no suitable standard exists (e.g., DRBGs).

Selected ongoing projects

- Lightweight Cryptography Project aims to provide symmetric key cryptography solutions specific to constrained devices.
- Post Quantum Cryptography Project aims to standardize one (or more) quantum-resistant public-key cryptographic algorithms.
- Threshold Cryptography Project studies security of the implementations, in *multi-party* and *single-device* settings.
- Crypto Publication Review Project aims to review quality of the standards and guidelines every five years, or more frequently if a need arises.
- Circuit Complexity Project aims to develop new techniques for constructing better circuits for use by academia and industry.

Boolean Function Complexity

Problem: Given a basis of Boolean gates, construct a circuit that computes a function that is optimal w.r.t. some criteria, such as

- Size complexity: The number of gates in the circuit.
- Depth complexity: The length of the longest path from an input gate to the output gate.

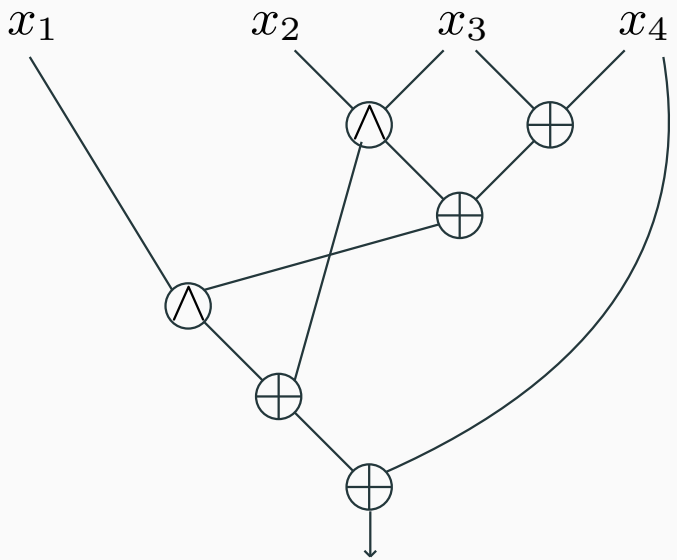
Target metric depends on the application.

- Circuits with small number of gates use less energy and occupy smaller area, and are desired for lightweight cryptography applications running on constrained devices.
- Circuits with small number of AND gates are desired for secure multi-party computation, zero-knowledge proofs and side channel protection.
- Circuits with small AND-depth are desired for homomorphic encryption schemes.

Boolean Circuits

A *Boolean circuit* with n inputs and m outputs is a directed acyclic graph (DAG), where

- the inputs and the gates are *nodes*,
- the edges correspond to the Boolean-valued *wires*,
- *fanin*/*fanout* of a node is the number of wires going in/out the node,
- the nodes with fanin zero are called the *input nodes* and are labeled with $\{x_1, \ldots, x_n\}$,
- the nodes with fanout zero is the *output node*.



Boolean Gates and Functional Completeness

x	0	0	1	1
y	0	1	0	1
f	*	*	*	*
0	0	0	0	0
x AND y	0	0	0	1
x AND $\overline{y}$	0	0	1	0
x	0	0	1	1
$\overline{x}$ AND y	0	1	0	0
y	0	1	0	1
x XOR y	0	1	1	0
x OR y	0	1	1	1
x NOR y	1	0	0	0
NOT (x OR y)	1	0	0	1
$\overline{y}$	1	0	1	0
x OR $\overline{y}$	1	0	1	1
$\overline{x}$	1	1	0	0
$\overline{x}$ OR y	1	1	0	1
x NAND y	1	1	1	0
1	1	1	1	1

A functionally complete set of Boolean operators is one which can be used to express all possible truth tables by combining members of the set into a Boolean expression.

- {NAND}, {NOR}, {NOT,AND,XOR}, {NOT,AND,OR} and {AND,NOT} are complete.
- {AND,OR} is not complete,
  - not possible to express NOT using ANDs and ORs.

Boolean circuit from the truth table

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Every n-variable Boolean function can be expressed by a Boolean expression of size $O(n2^n)$,

- using Disjunctive Normal Form, or
- using Conjunctive Normal Form.

Heuristic approaches using graphical techniques (e.g. Karnaugh map, Quince-McCluskey algorithm) or SAT solvers etc.

Boolean circuit from the truth table

Disjunctive Normal Form: sum of products (minterms)

x	y	z	f	DNF
0	0	0	0	0
0	0	1	1	$\overline{\texttt{x}}$ AND $\overline{\texttt{y}}$ AND z
0	1	0	1	$\overline{\texttt{x}}$ AND y AND $\overline{\texttt{z}}$
0	1	1	1	$\overline{\texttt{x}}$ AND y AND z
1	0	0	1	x AND $\overline{\texttt{y}}$ AND $\overline{\texttt{z}}$
1	0	1	1	$\overline{\texttt{x}}$ AND y AND $\overline{\texttt{z}}$
1	1	0	0	0
1	1	1	0	0

$f =$ ($\overline{\texttt{x}}$ AND $\overline{\texttt{y}}$ AND z) OR ($\overline{\texttt{x}}$ AND y AND $\overline{\texttt{z}}$) OR ($\overline{\texttt{x}}$ AND y AND z) OR (x AND $\overline{\texttt{y}}$ AND $\overline{\texttt{z}}$) OR ($\overline{\texttt{x}}$ AND y AND $\overline{\texttt{z}}$)

Boolean circuit from the truth table

Conjunctive Normal Form: product of sums (maxterms)

x	y	z	f	
0	0	0	0	(x OR y OR z)
0	0	1	1	
0	1	0	1	
0	1	1	1	
1	0	0	1	
1	0	1	1	
1	1	0	0	($\overline{\text{x}}$ OR $\overline{\text{y}}$ OR z)
1	1	1	0	($\overline{\text{x}}$ OR $\overline{\text{y}}$ OR $\overline{\text{z}}$)

f = (x OR y OR z) AND ($\overline{\text{x}}$ OR $\overline{\text{y}}$ OR z) AND ($\overline{\text{x}}$ OR $\overline{\text{y}}$ OR $\overline{\text{z}}$)

Straight Line Program (SLP)

```
begin CIRCUIT MAJ3
# Description: The majority of x1,x2,x3
Inputs: x1:x3;
Outputs: y1;
GateSyntax: GateName Output Inputs
begin SLP
  XOR t1 x1 x2;
  XOR t2 x1 x3;
  AND t3 t1 t2;
  XOR y1 t3 x1
end SLP end CIRCUIT
```

Relevant Functions

Any function used in cryptographic designs:

- Polynomial multiplication of degree n over GF(2)
- n-bit Boolean functions or classes of Boolean functions (e.g., cubic, symmetric)
- Vectorial Boolean functions (e.g., AES S-box, MDS)

Circuit	#Gates	# AND	# XOR	# XNOR	# NOT	Depth	AND depth
AES S-Box	113	32	77	4	0	27	6
AES S-Box^{-1}	121	34	83	4	0	21	4
AES-128(k,m)	28 600	6400	21 356	844	0	326	60
AES-128(0,m)	21 392	5120	14 652	1620	0	325	60
SHA-256(m)	115 882	22 385	89 248	3894	355	5403	1604
SHA-256(cv,m)	118 287	22 632	92 802	2840	13	5458	1607

k: AES key, cv : chaining value, m: message (128-bit for AES; 512-bit for SHA-256)

Multiplicative Complexity

Multiplicative Complexity (MC) of f, denoted $C_\wedge(f)$, is the minimum number of AND gates that is sufficient to evaluate f over the basis (AND, XOR, NOT).

- MC of an affine functions is zero.
- MC of a quadratic function is at most $\lfloor n/2 \rfloor$.
- Multiplicative complexity of a randomly selected n-bit Boolean function is at least $2^{n/2} - \mathcal{O}(n)$.
  - No specific n-variable function had been proven to have MC larger than n.
- Degree Bound: MC of a function with degree d is at least $d - 1$.
- The number of n-variable Boolean functions with MC k is at most $2^{k^2-k+2kn+n+1}$.

Why do we count the AND gates?

- **Lightweight Cryptography:** Efficient implementations needed for resource-constrained devices (e.g. RFID tags). The technique of minimizing the number of AND gates, and then optimizing the linear components leads to the implementations with low gate complexity.
- **Secure multi-party computation:** Reducing the number of AND gates improves the efficiency of secure multi-party protocols (e.g. conducting online auctions in a way that the winning bid can be determined without opening the losing bids).
- **Side channel attacks:** Minimizing the number of AND gates is necessary when implementing a masking scheme to prevent side-channel attacks.
- **Cryptanalysis of cryptographic primitives:** Primitives with low multiplicative complexity may be susceptible to algebraic cryptanalysis.

- Multiplicative Complexity of Boolean functions with $n \leq 6$
- Boolean functions with Multiplicative Complexity 1,2,3 and 4
- Multiplicative Complexity of Symmetric Boolean Functions

- Multiplicative Complexity of Boolean functions with $n \leq 6$
- Boolean functions with Multiplicative Complexity 1,2,3 and 4
- Multiplicative Complexity of Symmetric Boolean Functions

Main observation

Multiplicative Complexity is affine invariant.

- Boolean functions $f, g \in B_n$ are affine equivalent if there exists a transformation of the form $f(x) = g(Ax + a) + b \cdot x + c$, where A is a non-singular $n \times n$ matrix over $\mathbb{F}_2$;
- The set of affine equivalent functions constitute an equivalence class denoted by $[f]$, where f is an arbitrary function from the class.
- Affine equivalent Boolean functions have the same MC.

MC of 4- and 5-bit Boolean Functions [1]

Method

1. Find a simple representative from each equivalence class.
2. Find a circuit with small number of AND gates.
3. Check if it is optimal using the degree bound.

The multiplicative complexity is

- ≤ 3 for $f \in B_4$ (8 equivalence classes),
- ≤ 4 for $f \in B_5$ (48 equivalence classes).

Equivalence classes for $n = 4$

Class	Representative
1	x_1
2	x_1x_2
3	$x_1x_2 + x_3x_4$
4	$x_1x_2x_3$
5	$x_1x_2x_3 + x_1x_4$
6	$x_1x_2x_3x_4$
7	$x_1x_2x_3x_4 + x_1x_2$
8	$x_1x_2x_3x_4 + x_1x_2 + x_3x_4$

[1] M. Sönmez Turan, R. Peralta: The Multiplicative Complexity of Boolean Functions on Four and Five Variables. LightSec 2014: 21-33a

MC of 6-bit Boolean Functions [2]

The same approach does not work for $n = 6$, since

- The number of equivalence classes is 150 537, and
- Simple heuristics do not find optimal circuits, as representatives are more complex.
- For some classes, it is not possible to verify optimality using the degree bound.

Method

Exhaustively construct all Boolean circuits with 1,2, 3, ... AND gates, and mark the Boolean functions that can be generated by the circuits until all 6-bit Boolean functions are generated.

[2] Ç. Çalık, M. Sönmez Turan, R. Peralta: The multiplicative complexity of 6-variable Boolean functions. Cryptogr. Commun. 11(1): 93-107 (2019)

MC of 6-bit Boolean Functions [2]

The same approach does not work for $n = 6$, since

- The number of equivalence classes is 150 537, and
- Simple heuristics do not find optimal circuits, as representatives are more complex.
- For some classes, it is not possible to verify optimality using the degree bound.

Method

Exhaustively construct all Boolean circuits with 1,2, 3, ... AND gates, and mark the Boolean functions that can be generated by the circuits until ~~all 6-bit Boolean functions are generated~~ a function from each equivalence class is generated.

[2] Ç. Çalık, M. Sönmez Turan, R. Peralta: The multiplicative complexity of 6-variable Boolean functions. Cryptogr. Commun. 11(1): 93-107 (2019)

MC of 6-bit Boolean Functions [2]

The same approach does not work for $n = 6$, since

- The number of equivalence classes is 150 537, and
- Simple heuristics do not find optimal circuits, as representatives are more complex.
- For some classes, it is not possible to verify optimality using the degree bound.

Method

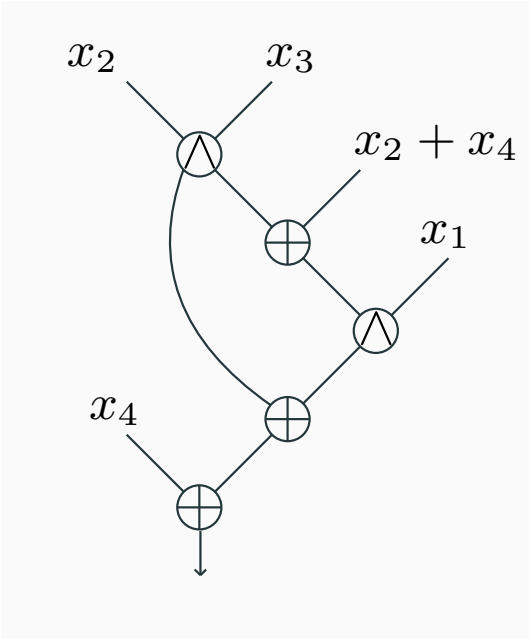
Exhaustively construct all Boolean ~~circuits~~ topologies with 1,2, 3, ... AND gates, and mark the Boolean functions that can be generated by the circuits until a function from each equivalence class is generated.

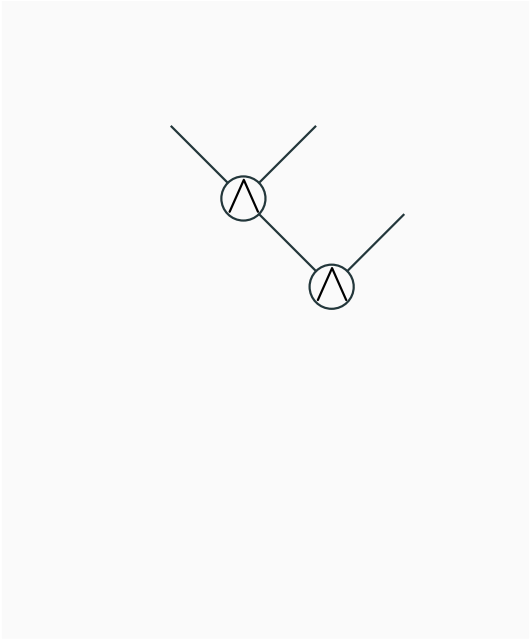
[2] Ç. Çalık, M. Sönmez Turan, R. Peralta: The multiplicative complexity of 6-variable Boolean functions. Cryptogr. Commun. 11(1): 93-107 (2019)

Topology of a circuit

Topology is an abstraction of a Boolean circuit that shows the relations between AND gates

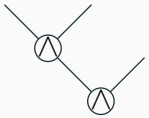
Example: Let $f = x_1x_2x_3 + x_1x_2 + x_1x_4 + x_2x_3 + x_4$.



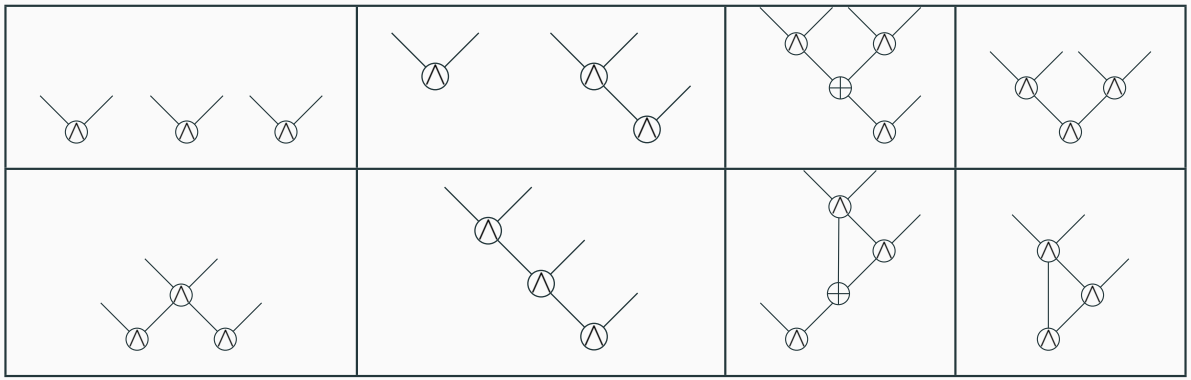


Constructing Circuit Topologies

Topologies with 1 AND gate: 

Topologies with 2 AND gates:   and 

Topologies with 3 AND gates

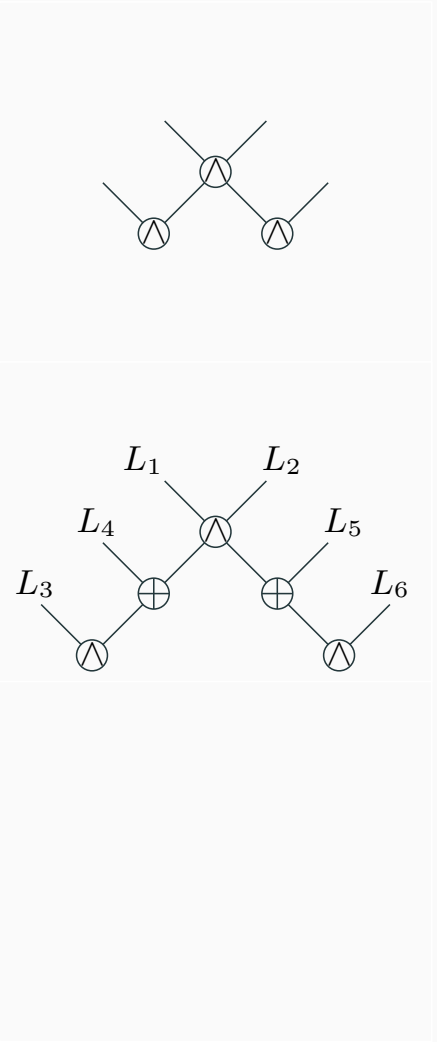


Number of topologies

# ANDs	1	2	3	4	5	6
# Topologies	1	2	8	84	3 170	475 248

Evaluating Topologies to Generate Boolean Functions

- A topology with k AND gates can be supplied $2k$ linear function inputs $X = (L_1, \ldots, L_{2k})$. Trying all inputs becomes quickly infeasible since there are 2^{2kn} choices (2^{60} inputs for $n = 6,\ k = 5$).
- Any affine transformation of the inputs $A(X) = (A(L_1), \ldots, A(L_{2k}))$ will produce a function from the same equivalence class. Hence, the inputs that are affine transformations of each other need not be considered.
- The number of inputs corresponds to the Gaussian binomial coefficient $\binom{2k}{n}_2$ ($\approx 2^{26}$ inputs for $n = 6,\ k = 5$).



Method

- Construct topologies having $k = 1, 2, 3, 4, 5$ AND gates.
- For each topology, find the equivalence classes it can produce.
  - 149 426 equivalence classes out of 150 357 generated with at most 5 AND gates.
  - Remaining 931 equivalence classes were generated from a selection of 6 AND gate topologies.
- Computations were done on a cluster (Intel Xeon E5-2630 processor, 64GB RAM) and took 38 422 core hours.

Multiplicative Complexity Distribution for $n = 6$

The multiplicative complexity is ≤ 6 for $f \in B_6$.

Showed that there exists $f \in B_6$ with multiplicative complexity 6, e.g.,

- A function with 6 monomials:

  $x_1x_5 + x_3x_6 + x_3x_4x_5 + x_2x_4 + x_1x_2x_6 + x_1x_2x_3x_4x_5x_6$
- A function with algebraic degree 4:

  $x_4x_5 + x_3x_4x_5 + x_2x_5 + x_2x_4 + x_2x_4x_6 + x_1x_5x_6 + x_1x_4 + x_1x_3 + x_1x_2x_4x_5 + x_1x_2x_3x_6$

MC	#classes	#functions	$\log_2(\#functions)$
0	1	128	7.00
1	1	83 328	16.34
2	3	73 757 184	26.13
3	24	281 721 079 808	38.03
4	914	7 944 756 861 878 272	52.81
5	148 483	18 344 082 080 963 133 440	63.99
6	931	94 716 954 089 619 456	56.39

The method is infeasible for $n \geq 7$, due to the large number of affine equivalence classes and topologies.

- Multiplicative Complexity of Boolean functions with $n \leq 6$
- Boolean functions with Multiplicative Complexity 1,2,3 and 4
- Multiplicative Complexity of Symmetric Boolean Functions

Dimension of a Boolean function

The following functions are all affine equivalent and have MC=1:

$$x_1x_2$$

$$x_1 + x_2x_3$$

$$(x_1 + x_2)(x_3 + x_4) = x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4$$

It is easier to work on smaller number of variables.

Definition. Let L_f be the number of input variables that appear in the *algebraic normal form* (ANF) of a Boolean function f. The dimension of f is the smallest number of variables that appear in the ANF among the functions that are affine equivalent to f:

$$dim(f) = \min_{g \in [f]} L_g.$$

Example. $dim(x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4) = dim(x_1x_2) = 2$

Linear Structures and Dimension of a Boolean Function

$\alpha \in \mathbb{F}_2^n$ is a linear structure of f if $f(x) + f(x + \alpha)$ is constant [Nyb92, Lai94].

The set of linear structures of a Boolean function form a vector space, whose dimension $d_l(f)$ is called the linearity dimension of f, where

$$d_l(f) = \log_2 \#\{f(x) + f(x + \alpha), \alpha \in \mathbb{F}_2^n\}.$$

The dimension of an n-variable Boolean function is:

$$dim(f) = n - d_l(f).$$

Main Observation

The MC of f is at least $\lceil \dim(f)/2 \rceil$.

Sketch of the proof.

1. Let $C_\wedge(f) = k$, consider a circuit implementing f with k AND gates.
2. The topology with k AND gates has $2k$ linear function inputs.
3. The rank of $2k$ linear functions can be at most $2k$.
4. Any set of $2k$ linear functions on $n > 2k$ variables can be affine transformed to functions having at most $2k$ variables.
5. Therefore, $dim(f) \leq 2k$, which implies $C_\wedge(f) \geq \lceil dim(f)/2 \rceil$.

Boolean functions with MC 1 and 2

Boolean functions with MC 1 [FP02]

- Functions with MC 1 are affine equivalent to x_1x_2.
- The number of n-variable Boolean functions with MC 1 is $2\binom{2^n}{3}$.

Boolean functions with MC 2 [FTT17]

- Functions with MC 2 are affine equivalent to one of the functions from the set $\{x_1x_2x_3,\ x_1x_2x_3 + x_1x_4,\ x_1x_2 + x_3x_4\}$.
- The number of n-variable Boolean functions with MC 2 is

$$2^n(2^n-1)(2^n-2)(2^n-4)\left(\frac{2}{21} + \frac{2^n-8}{12} + \frac{2^n-8}{360}\right).$$

Boolean functions with MC 3 and 4

Find exhaustive list of equivalence classes with MC 3 and 4.

Approach

Step 1. Construct Boolean circuits (topologies) with 3 and 4 AND gates.

Step 2. Evaluate the circuits to generate Boolean functions.

Step 3. Identify distinct affine equivalence classes with MC 3 and 4.

Affine Equivalence Classes with MC 3

Dimension 4:

$x_1x_2x_3x_4$
$x_1x_2 + x_1x_2x_3x_4$
$x_2x_3 + x_1x_4 + x_1x_2x_3x_4$

Dimension 5:

$x_3x_4 + x_1x_5 + x_1x_2x_5 + x_1x_2x_3x_4$	$x_3x_4 + x_1x_3x_4 + x_1x_2x_5$
$x_2x_4 + x_1x_5 + x_1x_2x_3$	$x_4x_5 + x_1x_2x_3$
$x_1x_2x_5 + x_1x_2x_3x_4$	$x_1x_3x_4 + x_1x_2x_5$
$x_2x_3x_5 + x_1x_4x_5 + x_1x_2x_3x_4$	$x_3x_5 + x_1x_2x_5 + x_1x_2x_3x_4$
$x_1x_3 + x_1x_2x_5 + x_1x_2x_3x_4$	$x_3x_4 + x_1x_2x_5 + x_1x_2x_3x_4$
$x_1x_5 + x_1x_2x_3x_4$	$x_2x_3 + x_1x_5 + x_1x_2x_3x_4$
$x_2x_3 + x_2x_3x_5 + x_1x_4x_5 + x_1x_2x_3x_4$	$x_1x_5 + x_1x_2x_5 + x_1x_2x_3x_4$

Dimension 6:

$x_3x_4 + x_2x_5 + x_1x_6$	$x_1x_6 + x_1x_3x_4 + x_1x_2x_5$
$x_3x_4 + x_1x_6 + x_1x_3x_4 + x_1x_2x_5$	$x_4x_5 + x_1x_6 + x_1x_2x_3$
$x_1x_6 + x_1x_2x_5 + x_1x_2x_3x_4$	$x_5x_6 + x_3x_4x_5 + x_1x_2x_6 + x_1x_2x_3x_4$
$x_3x_4 + x_1x_6 + x_1x_2x_5 + x_1x_2x_3x_4$	

Affine Equivalence Classes with MC 4

After evaluating 84 topologies with 4 AND gates, we obtained

- 26 classes with dimension 5,
- 888 classes with dimension 6,
- 321 classes with dimension 7,
- 42 classes with dimension 8.

Complete list is available at:

`https://github.com/usnistgov/Circuits/tree/master/data/mc_dim`

How about Boolean functions with MC 5?

MC	dimension											
	2	3	4	5	6	7	8	9	10	11	12	Total
1	1											1
2		1	2									3
3			3	14	7							24
4				26	888	321	42					1277
5					148483	*	*	*	575			*
6					931	*	*	*	*	*	*	*

Table 1: The Distribution of Classes w.r.t MC and Dimension.

- Multiplicative Complexity of Boolean functions with $n \leq 6$
- Boolean functions with Multiplicative Complexity 1,2,3 and 4
- Multiplicative Complexity of Symmetric Boolean Functions

Symmetric Boolean Functions

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is said to be symmetric, if the output depends only on the Hamming weight of the input.

Representation: Symmetric functions can be represented using $(n+1)$-bit vector $v(f) = (v_0, \ldots, v_n)$ such that $f(x) = v_i$ if the weight of x is i.

- Elementary symmetric functions, Σ_k^n, composed of all degree k monomials.
- Counting function, E_k^n, outputs 1 if and only if $w_h(x) = k$.
- Threshold function, T_k^n, outputs 1 if and only if $w_h(x) \geq k$.

Every symmetric function can be written as a summation of elementary symmetric functions.

Can we use the inherent symmetries to find efficient implementations?

Example - Threshold function

$$T_3^5(x_1, x_2, x_3, x_4, x_5) = \begin{cases} 1, & \text{if majority of } \{x_1, x_2, x_3, x_4, x_5\} \text{ is } 1, \\ 0, & \text{otherwise.} \end{cases}$$

The algebraic normal form:

$$\begin{aligned} T_3^5 \quad = \quad & x_1x_2x_3 + x_1x_2x_4 + x_1x_2x_5 + x_1x_3x_4 + x_1x_3x_5 + x_1x_4x_5 + x_2x_3x_4 \\ & + x_2x_3x_5 + x_2x_4x_5 + x_3x_4x_5 + x_1x_2x_3x_4 + x_1x_2x_3x_5 + x_1x_2x_4x_5 \\ & + x_1x_3x_4x_5 + x_2x_3x_4x_5 \end{aligned}$$

The equation contains 35 multiplications, but **only 3 multiplications** are sufficient to compute it.

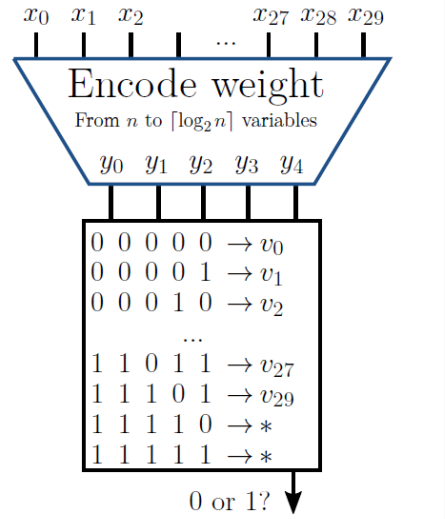
Multiplicative Complexity of the Symmetric Functions

- A construction of circuits for symmetric Boolean functions on n variables that requires $\leq n + 3\sqrt{n}$ AND gates. (Boyar et al., 2008)
- The MC of an n-bit nonlinear symmetric function is at least $\lfloor \frac{n}{2} \rfloor$.
- The MC of Σ_2^n is $\lfloor \frac{n}{2} \rfloor$.
- The MC of Σ_3^n is $\lceil \frac{n}{2} \rceil$.

Constructing Circuits for Symmetric Functions

Generic approach:

- Compute the binary representation of Hamming weight of $(x_1, \ldots, x_n)$
- Construct a second function based on the hamming weight with $\lceil log_n \rceil$ number of variables.



Composition of optimal sub-circuits do not necessarily result in an optimal circuit.

Approach: Use weight encodings that also considers the second part of the function.

- (Optimal) circuits for all symmetric functions with up to 25 variables.
- Upper bounds on the maximum MC of symmetric Boolean functions for $n \leq 132$.

Open Problems

- New heuristics for constructing circuits with small number of `AND` gates
- New lower and upper bounds on the MC of Boolean functions.

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$MC \leq$	1	2	3	4	6	13	26	41	57	88	120	183	247	374	502

- Prove that MC of a specific Boolean function is more than n.
- New circuit for AES s-box with less than 32 `AND` gates
- Results on special classes of functions (e.g., partially symmetric, or rotation symmetric functions)
- MC of vectorial Boolean functions on $n \geq 5$ bits.

Selected References

M. Sönmez Turan and R. Peralta. The Multiplicative Complexity of Boolean functions on Four and Five Variables. LightSec 2014, Turkey.

Ç. Çalık, M. Sönmez Turan, R. Peralta, The Multiplicative Complexity of 6-variable Boolean Functions, Cryptography and Communications 2018.

M. G. Find, D. Smith-Tone, M. Sönmez Turan, The Number of Boolean Functions with Multiplicative Complexity 2, International Journal of Information and Coding Theory, 2017.

Ç. Çalık, M. Sönmez Turan, R. Peralta, Boolean functions with multiplicative complexity 3 and 4. Cryptogr. Commun. 12, 935–946 (2020).

L. Brandao, Ç. Çalık, M. Sönmez Turan, R. Peralta: Upper bounds on the multiplicative complexity of symmetric Boolean functions. Cryptogr. Commun. 11(6): 1339-1362 (2019)

J. Boyar, R. Peralta, and D. Pochuev, "On the multiplicative complexity of Boolean functions over the basis $(\wedge, \oplus, 1)$, Theoretical Computer Science, vol. 235, no. 1, pp. 43 – 57, 2000.

J. Boyar, P. Matthews, and R. Peralta. Logic Minimization Techniques with Applications to Cryptology. Journal of Cryptology, vol. 26, pp. 280–312 (2013).

J. Boyar, and R. Peralta. Tight bounds for the multiplicative complexity of symmetric functions, Theoretical Computer Science, Volume 396, Issues 1–3, (2008)

M. Codish, L. Cruz-Filipe, M. Frank, P. Scheneider-Kamp, "When Six Gates are Not Enough", 2015

X. Lai, Additive and Linear Structures of Cryptographic Functions, FSE 1994, LNCS 1008, Springer-Verlag, pp. 75–85, 1994.

More Information

NIST Circuit Complexity Project Webpage:

`https://csrc.nist.gov/Projects/Circuit-Complexity`

GitHubLink:

`https://github.com/usnistgov/Circuits/`

Contact email:

`meltem.turan@nist.gov`

`circuit_complexity@nist.gov`