"Multiple-Valued Galois Field S/D Trees for GFSOP Minimization and their Complexity"

Anas Al-Rabadi, and Marek Perkowski

Portland State University
ECE Department
1800 S.W. 6th Ave., Portland, Oregon 97201 U.S.A.
[alrabadi, mperkows@ee.pdx.edu]

MOTIVATION FOR THIS RESEARCH

- Binary ESOP (exclusive or sum of products) are used in minimization of data path and communication logic
- Exact solutions are found only for small functions
- Quality of minimization of large functions is still uncertain
- Several new techniques programmed in Exorcism-mv-4
- Inclusive Forms can theoretically find exact solution without complete exhaustive search
- IF-based methods can potentially lead to smaller circuits
- ESOP can be realized in reversible logic using Toffoli and Feynman gates.
- Toffoli and Feynman gates can be generalized to multiple-valued logic
- Binary IFs can be generalized to multiple-valued IFs.

Presentation Overview

- Known families of binary canonical forms
- Two new families of canonical forms were presented in "**A Family of Canonical AND/EXOR Forms That Includes Exact Minimum ESOPs" by Malgorzata Chrzanowska-Jeske, Alan Mishchenko, Marek Perkowski**
- One of them includes minimum ESOPs!
- The concept of S/D trees
- Inclusive forms for binary logic
- Inclusive Forms (IF) for Galois Logic
- Enumeration of IFs for arbitrary multiple-valued logic.
- Conclusions and future work

Known Families of Canonical Forms

- Positive Polarity Reed-Müller Form - FPRM
- Fixed Polarity Reed-Müller Forms - FPRM
- Generalized Reed-Müller Forms - GRM
- Kronecker Forms - KRO
- Pseudo-Kronecker Forms - PDSKRO
- Generalized Kronecker Forms - GK
- Pseudo-Generalized Kronecker Forms-PGK
- Free Pseudo-Generalized Kronecker Forms

Positive Polarity Reed-Müller Form

$f(x_1, x_2, \ldots, x_n) =$

$a_0 \oplus a_1x_1 \oplus a_2x_2 \oplus \ldots \oplus a_nx_n \oplus$

$a_{12}x_1x_2 \oplus a_{13}x_1x_3 \oplus \ldots \oplus a_{n-1,n}x_{n-1}x_n \oplus$

$\ldots \oplus$

$a_{12\ldots n}x_1x_2\ldots x_n$

There is only 1 PPRM

$F = 1 \oplus x_1 \oplus x_2 \oplus x_1x_2$

Fixed Polarity Reed-Müller Form

$$f(x_1, x_2, \ldots, x_n) =$$
$$a_0 \oplus a_1 \tilde{x}_1 \oplus a_2 \tilde{x}_2 \oplus \ldots \oplus a_n \tilde{x}_n \oplus$$
$$a_{12} \tilde{x}_1 \tilde{x}_2 \oplus a_{13} \tilde{x}_1 \tilde{x}_3 \oplus \ldots \oplus a_{n-1,n} \tilde{x}_{n-1} \tilde{x}_n \oplus$$
$$\ldots \oplus$$
$$a_{12\ldots n} \tilde{x}_1 \tilde{x}_2 \ldots \tilde{x}_n$$

There are 2^n FPRMs

Example: $F = 1 \oplus x_1 \oplus \bar{x}_2 \oplus x_1 \bar{x}_2$

Generalized Reed-Muller Forms

$$f(x_1, x_2, \ldots, x_n) =$$
$$a_0 \oplus a_1 \tilde{x}_1 \oplus a_2 \tilde{x}_2 \oplus \ldots \oplus a_n \tilde{x}_n \oplus$$
$$a_{12} \tilde{x}_1 \tilde{x}_2 \oplus a_{13} \tilde{x}_1 \tilde{x}_3 \oplus \ldots \oplus a_{n-1,n} \tilde{x}_{n-1} \tilde{x}_n \oplus$$
$$\ldots \oplus$$
$$a_{12\ldots n} \tilde{x}_1 \tilde{x}_2 \ldots \tilde{x}_n$$

There are $2^{n 2^{n-1}}$ GRMs

Example: $F = 1 \oplus x_1 \oplus \bar{x}_2 \oplus \bar{x}_1 \bar{x}_2$

Cofactors

Negative cofactor

$f_0(x_2, \ldots, x_n) = f(0, x_2, \ldots, x_n)$

Positive cofactor

$f_1(x_2, \ldots, x_n) = f(1, x_2, \ldots, x_n)$

Sum of cofactors

$f_2(x_2, \ldots, x_n) = f(0, x_2, \ldots, x_n) \oplus f(1, x_2, \ldots, x_n)$

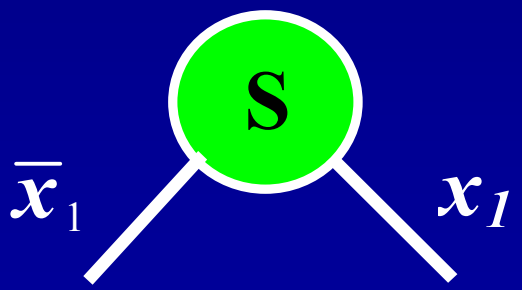
Fundamental Expansions - 1,2

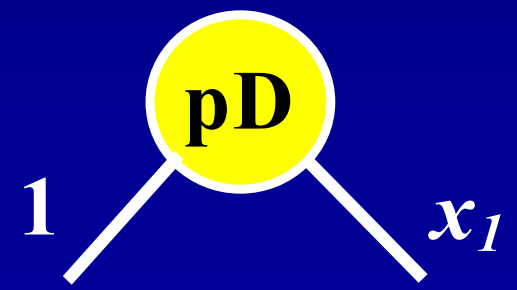
- Shannon Expansion - S

$$f(x_1, x_2, \ldots, x_n) = \bar{x}_1 \, f_0(x_2, \ldots, x_n) \oplus x_1 \, f_1(x_2, \ldots, x_n)$$

- Positive Davio Expansion - pD

$$f(x_1, x_2, \ldots, x_n) = 1 \cdot f_0(x_2, \ldots, x_n) \oplus x_1 \, f_2(x_2, \ldots, x_n)$$





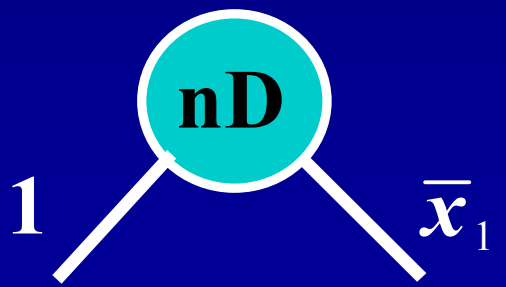
Fundamental Expansions - 3,4

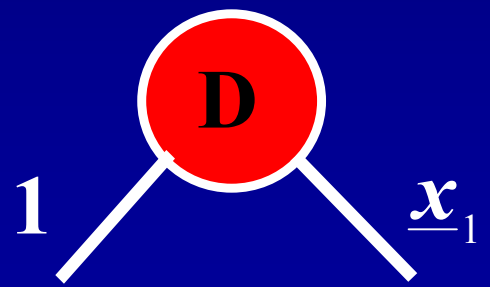
- Negative Davio Expansion - nD

$$f(x_1, x_2, \ldots, x_n) = 1 \cdot f_1(x_2, \ldots, \overline{x}_n) \oplus f_2(x_2, \ldots, x_n)$$

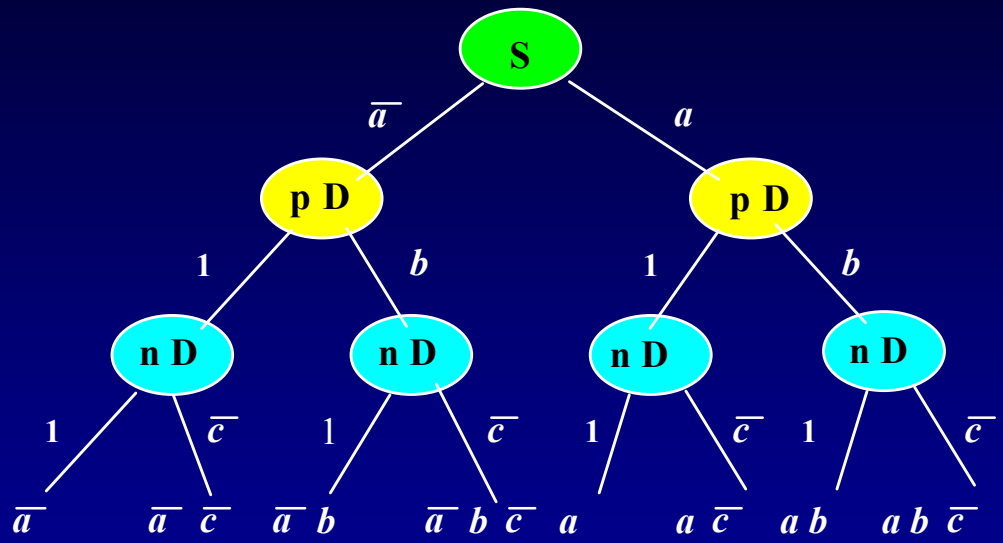
- Generalized Davio Expansion - D

$$f(x_1, x_2, \ldots, x_n) = 1 \cdot f_1(x_2, \ldots, \underline{x}_n) \oplus f_2(x_2, \ldots, x_n)$$

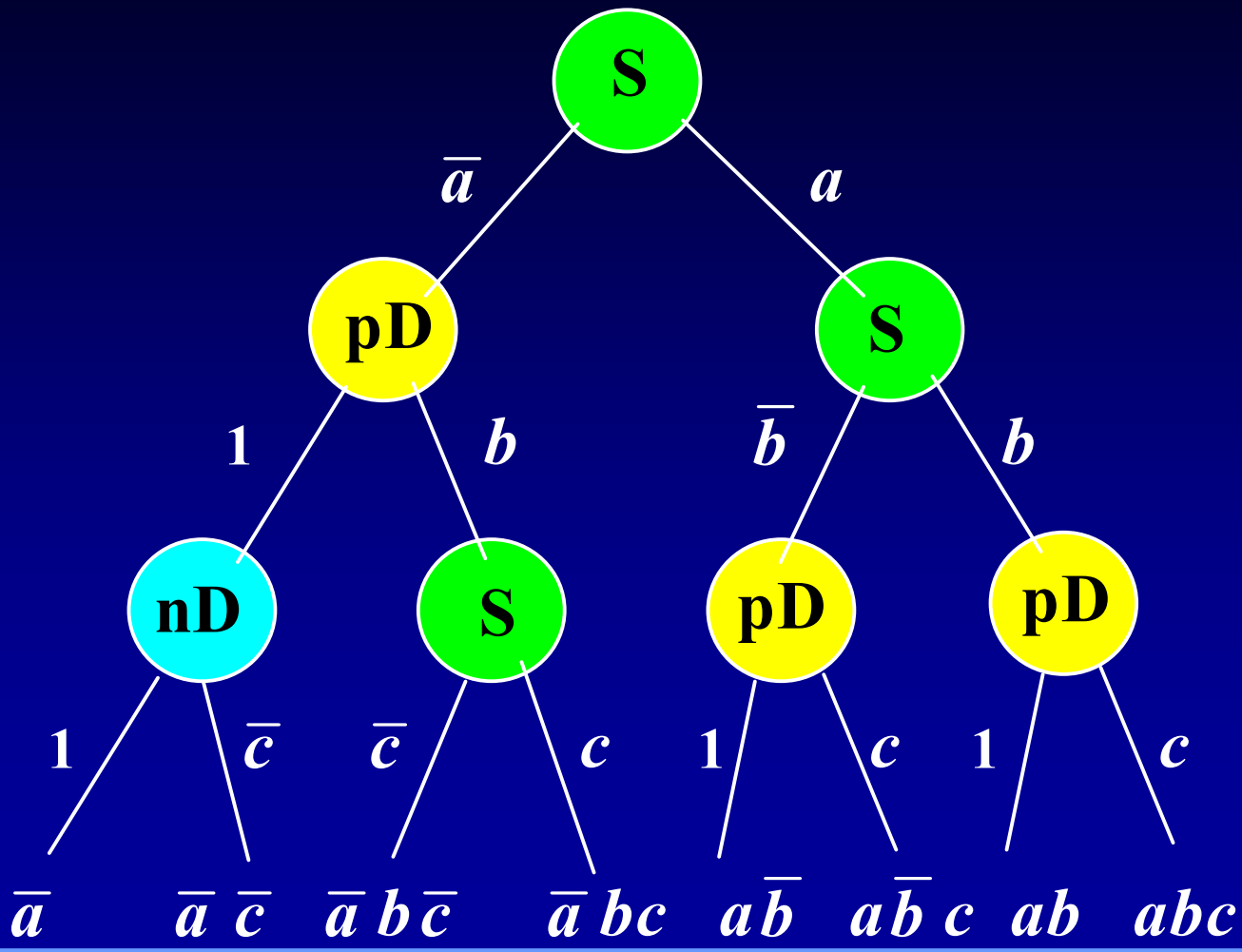




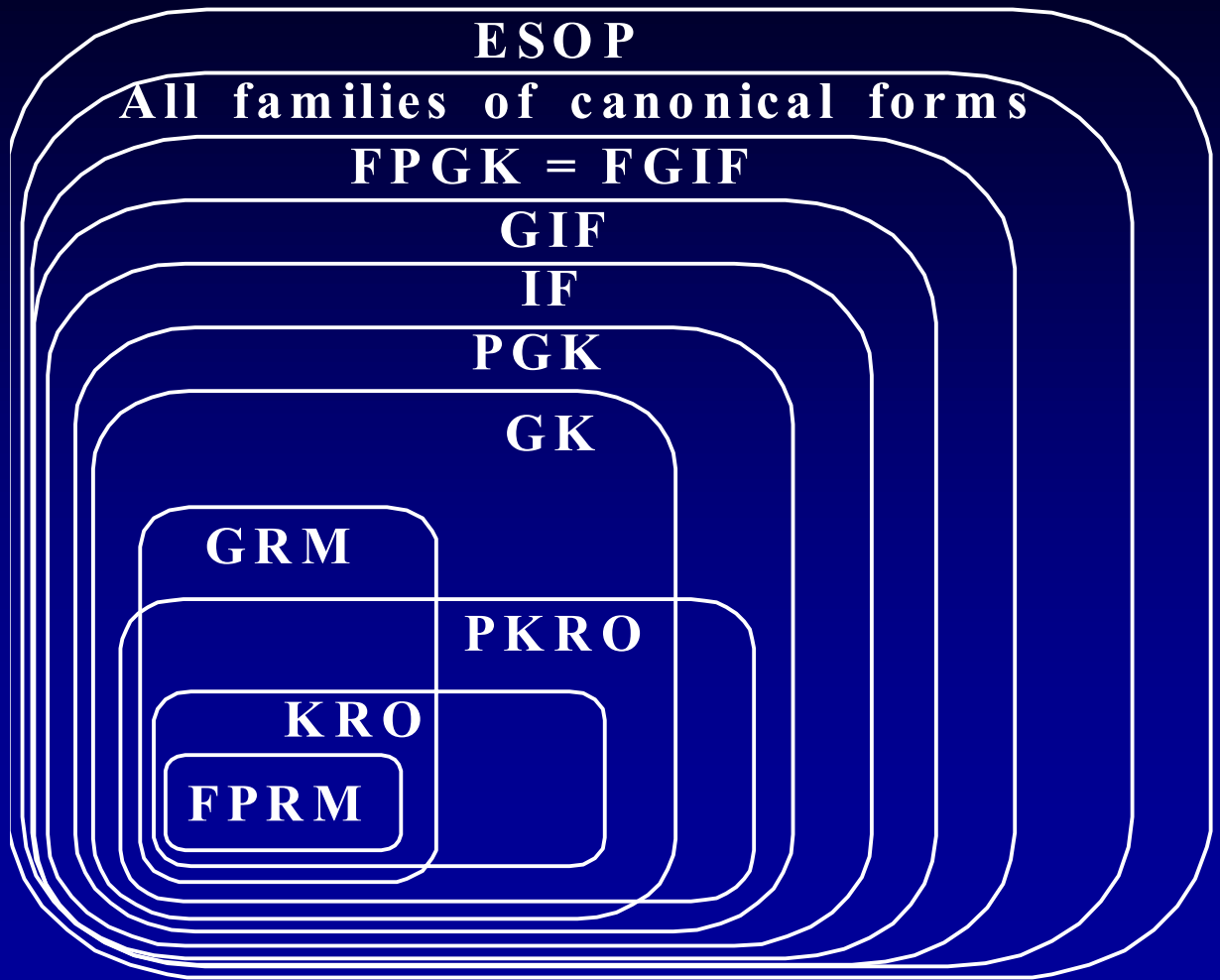
Example of a KRO Form



Example of a Pseudo-KRO Form



Hierarchy of Families

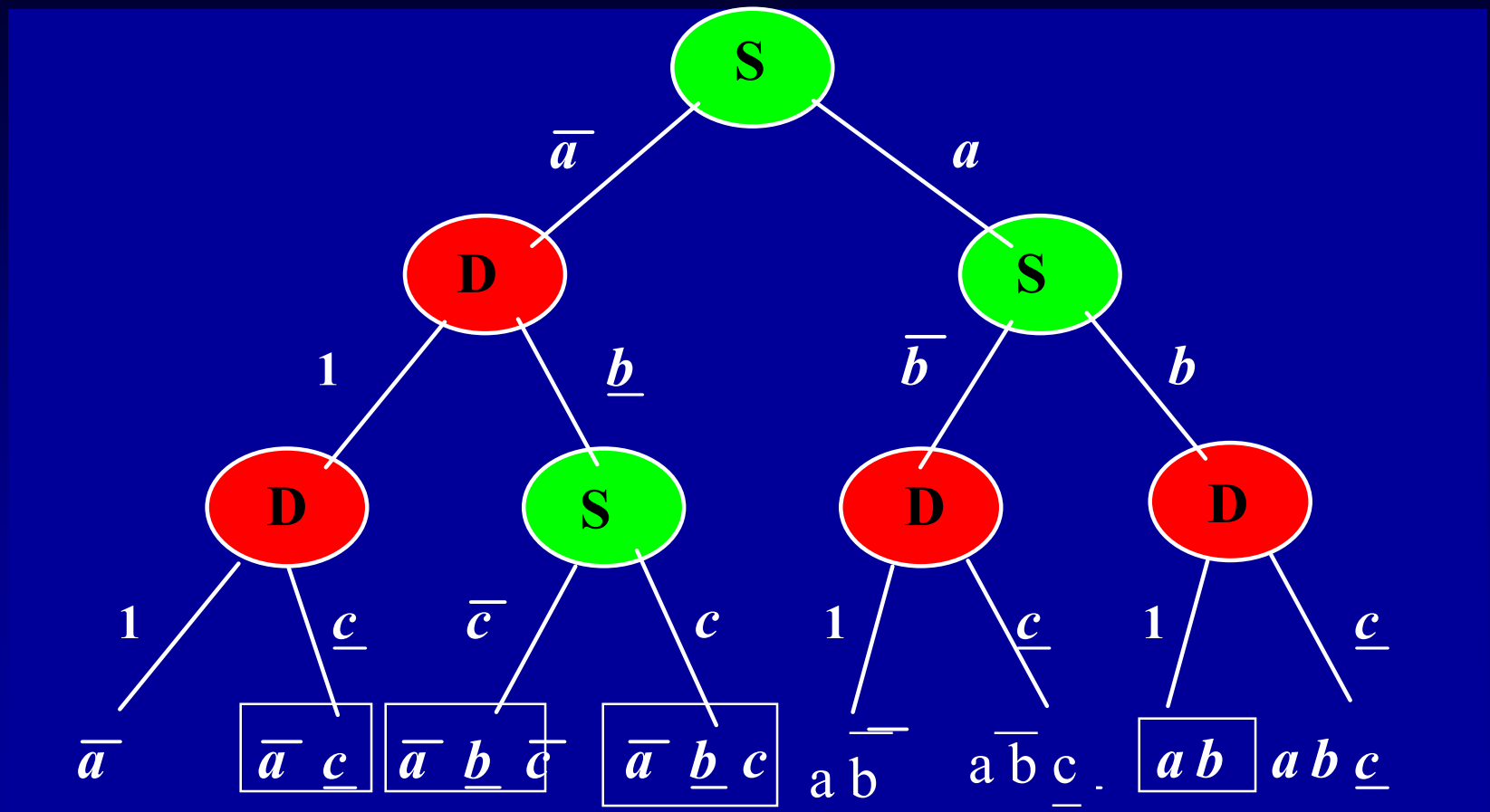


S/D Trees

Definition. An S/D tree is created by:

- selecting one variable order
- building the binary tree for this order
- choosing an arbitrary assignment of Shannon (S) and Generalized Davio (D) expansions for the nodes

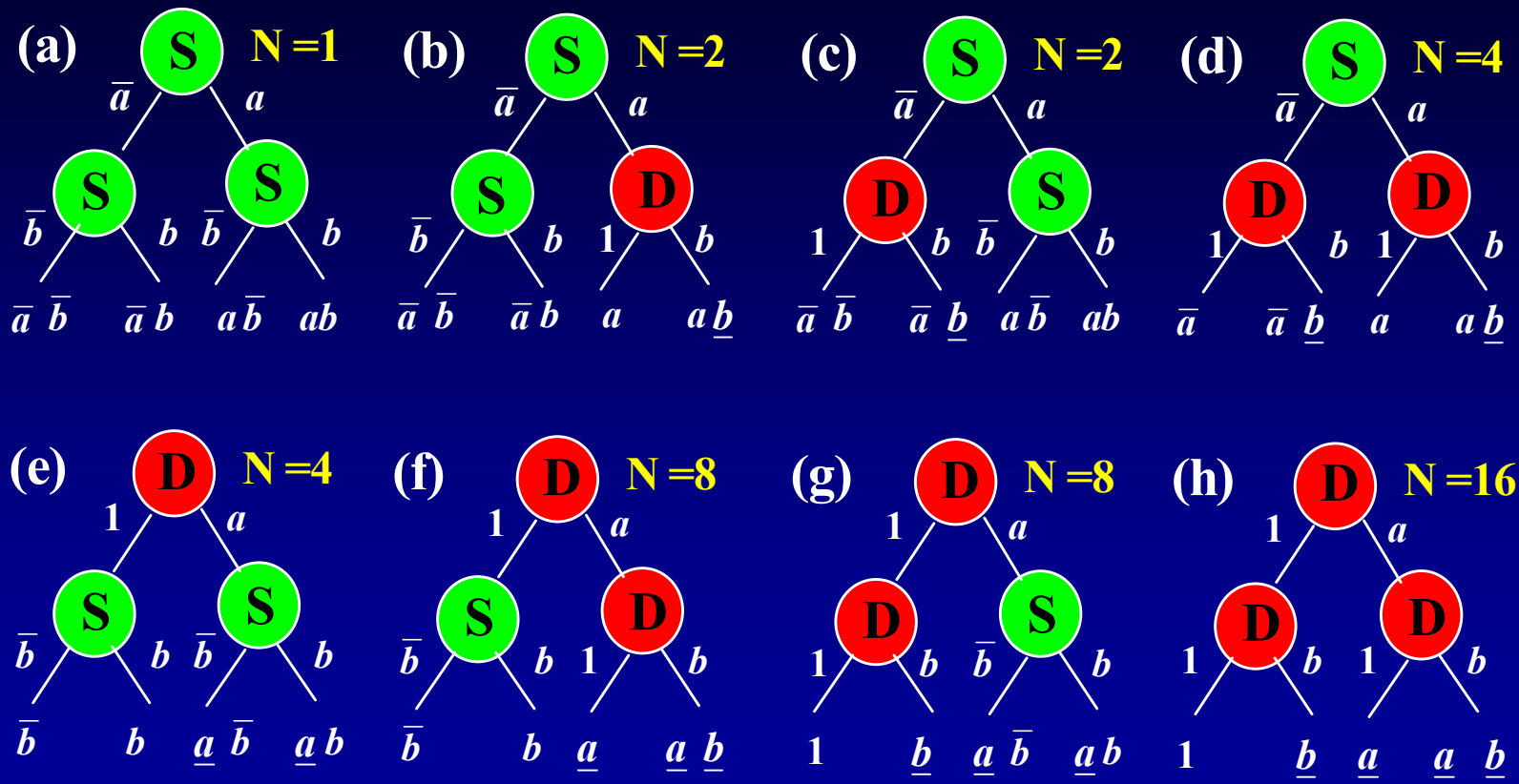
An S/D Tree for Order {a,b,c}



Generation of Inclusive Forms for a Given Variable Order

- Generate all S/D trees for the given variable order
- For each S/D tree, generate a set of forms created by replacing all generalized literals by literals in arbitrary polarities
- Take the union of all these sets

Inclusive Forms for Two Variables



$$N_{IF} = (1 + 2 + 2 + 4) + (4 + 8 + 8 + 16) = 45$$

Properties of Inclusive Forms

- Inclusive Forms are canonical
- For a given order of n variables, there are

$$\prod_{k=0}^{n-1}\left(1+2^{2^{n-k-1}}\right)^{2^k}$$

unique Inclusive Forms

The Number of IFs for Three Variables

For n = 3, there are $\begin{pmatrix} 3^n \\ 2^n \end{pmatrix} = \begin{pmatrix} 27 \\ 8 \end{pmatrix} =$

2,220,075 possible expansions.

Only 527,121 of these expansions are linearly independent, or canonical.

According to the Formula, there are

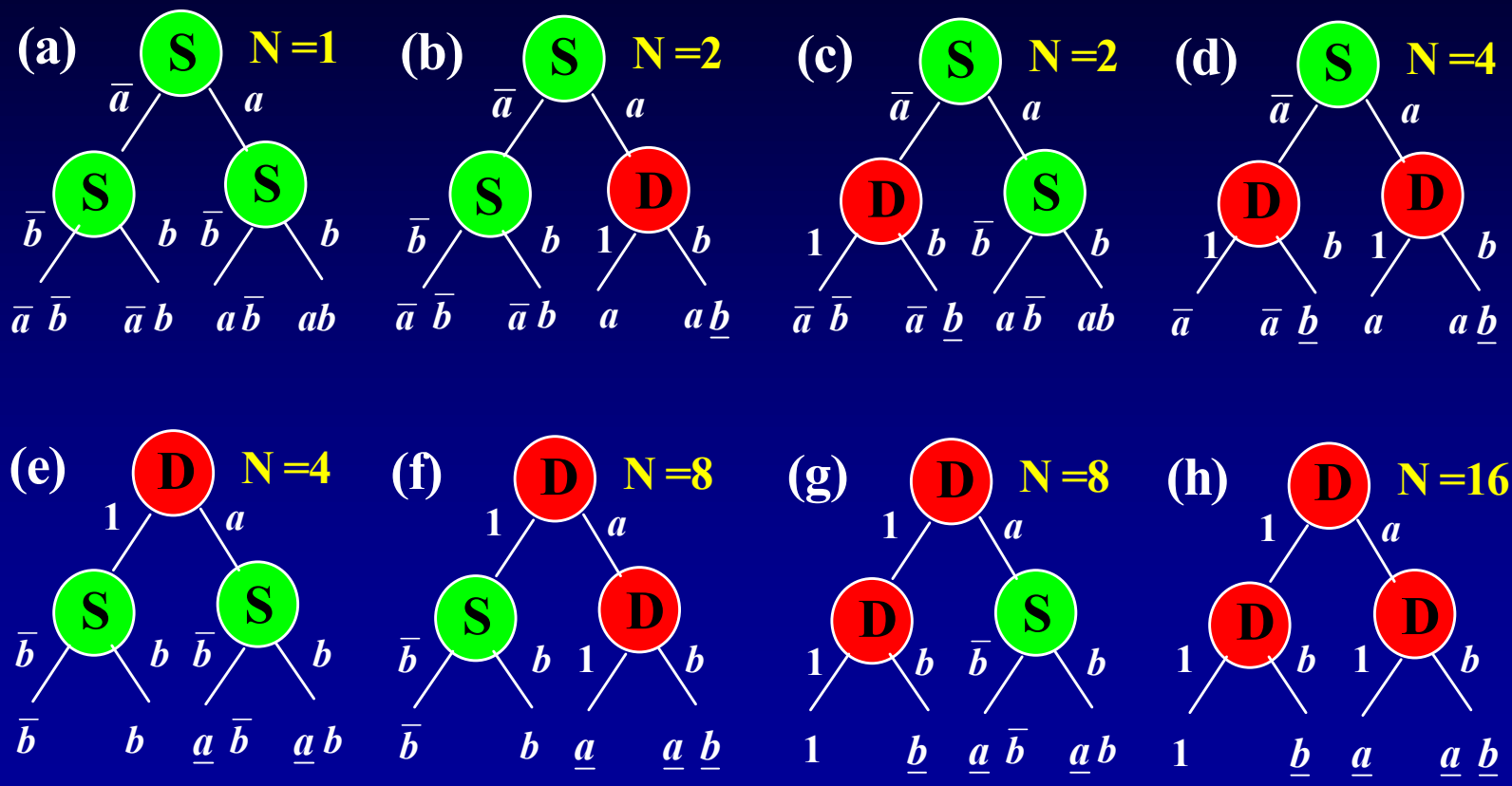
$N_{IF} = (1 + 16)^1(1 + 4)^2(1 + 2)^4 = 34{,}425$

IFs for each order of variables.

Generation of Generalized Inclusive Forms

- Generate sets of Inclusive Forms for all variable orders
- Take the union of all these sets

Generalized Inclusive Forms for Two Variables

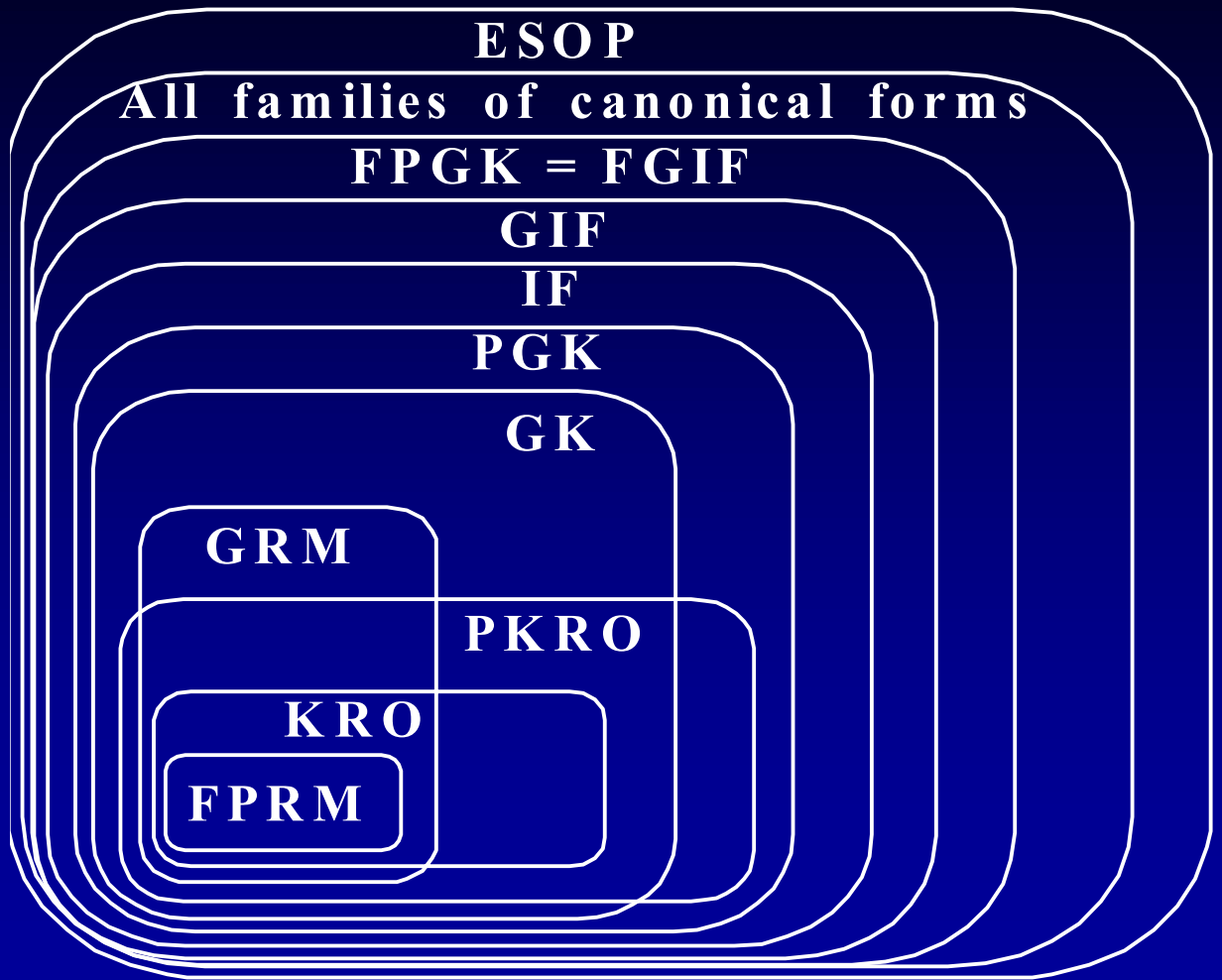


$$N_{GIFs} = 2 * 45 - (1 + 4 + 4 + 16) = 65$$

Properties of Free Generalized Inclusive Forms

- Free Generalized Inclusive Forms .
- Free Generalized Inclusive Forms are canonical
- The Theorem: The set of all FGIFs includes all minimum ESOPs for an arbitrary Boolean function

Family Relations



The Number of Canonical Forms, IFs, and GIFs over the Number of Variables

vars	# all	# can	# if	#gif	all/can	all/if	can/if	all/gif	can/gif
1	3*	3	3	3	1	1	1	1	1
2	126*	81	45	65	1.56	2.80	1.80	1.94	1.25
3	2,220,075*	527,121	34,425	109,361	4.21	64.5	15.3	20.3	4.8
4	100,000,000	1,037,459	175	1583	96.4	$5.7*10^5$	$5.9*10^3$	$6.3*10^4$	$6.6*10^2$
5	100,000,000	108,044	0	1	925	$>1.0*10^8$	$>1.0*10^5$	$>1.0*10^8$	$>1.0*10^5$

Canonical Forms for EXOR and Galois Logic

- Minimization of expressions for hardware realization.

- In binary, minimal ESOP has been found using binary S/D trees.

- Galois Fields are important algebraic structures.
  - GF addition and GF multiplication possess Latin Square property:
    - in any row and column the elements are all different, and the elements have a different order in each row and column.

- Due to this property, GF found applications in many areas, like testing of digital circuits.

Latin Square Property can be seen for instance in the following addition and multiplication tables over GF(4):

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

*	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

Notation;

$x' = x+1$

$x'' = x+2$

Counterparts of polarities

- **Example 1**. Let $f(x_1,x_2) = x_1''x_2 + x_2'''x_1$
- The quaternary truth vector of the function f is: $\mathbf{F} = [0,3,1,2,2,1,3,0,3,0,2,1,1,2,0,3]^T$
- we obtain the following quaternary Shannon expansion over GF(4) of the function f:

$$f = 2 \cdot {}^0x_1\,{}^1x_2 + 3\cdot {}^0x_1\,{}^2x_2 + {}^0x_1\,{}^3x_2 + 3\cdot {}^1x_1\,{}^0x_2 + \cdot {}^1x_1\,{}^1x_2 + 2\cdot {}^1x_1\,{}^3x_2 + {}^2x_1\,{}^0x_2 + 3\cdot {}^2x_1\,{}^1x_2 + 2\cdot {}^2x_1\,{}^2x_2 + 2\cdot {}^3x_1\,{}^0x_2 + {}^3x_1\,{}^2x_2 + 3\cdot {}^3x_1\,{}^3x_2.$$

- Using the axioms of GF(4), it can be derived that the 1-Reduced Post literals defined in equation (5) are related to the shifts of variables over GF(4) in terms of powers as follows:
- $^{0}x = x^3 + 1$ *(7)*
- $^{0}x = x' + (x')^2 + (x')^3$ *(8)*
- $^{0}x = 3(x'') + 2(x'')^2 + (x'')^3$ *(9)*
- $^{0}x = 2(x''') + 3(x''')^2 + (x''')^3$ *(10)*

- $^{1}x = x + (x)^2 + (x)^3$ *(11)*
- $^{1}x = (x')^3 + 1$ *(12)*
- $^{1}x = 2(x'') + 3(x'')^2 + (x'')^3$ *(13)*
- $^{1}x = 3(x''') + 2(x''')^2 + (x''')^3$ *(14)*

- $^{2}x = 3(x) + 2(x)^2 + (x)^3$ *(15)*
- $^{2}x = 2(x') + 3(x')^2 + (x')^3$ *(16)*
- $^{2}x = (x'')^3 + 1$ *(17)*
- $^{2}x = x''' + (x''')^2 + (x''')^3$ *(18)*

- $^{3}x = 2(x) + 3(x)^2 + (x)^3$ *(19)*
- $^{3}x = 3(x') + 2(x')^2 + (x')^3$ *(20)*
- $^{3}x = x'' + (x'')^2 + (x'')^3$ *(21)*
- $^{3}x = (x''')^3 + 1$ *(22)*

- Where: ^{0}x, ^{1}x, ^{2}x, ^{3}x are the: zeroth, first, second, and third polarities of the 1-Reduced Post literal, respectively. Also, x, x', x", x''' are the zeroth, first, second, and third shifts (inversions) of the variable x respectively, and variable x can take any value of the set {0, 1, 2, 3}.

The extension of binary Shannon and Davio expansions to higher radix logics.

Let us define the 1-Reduced Post literal as follows:

$^{i}x = 1$ *iff* $x = i$ *else* $^{i}x = 0$

Utilizing such literal, the following are Shannon and Davio expansions For 4-valued GF logic, respectively:

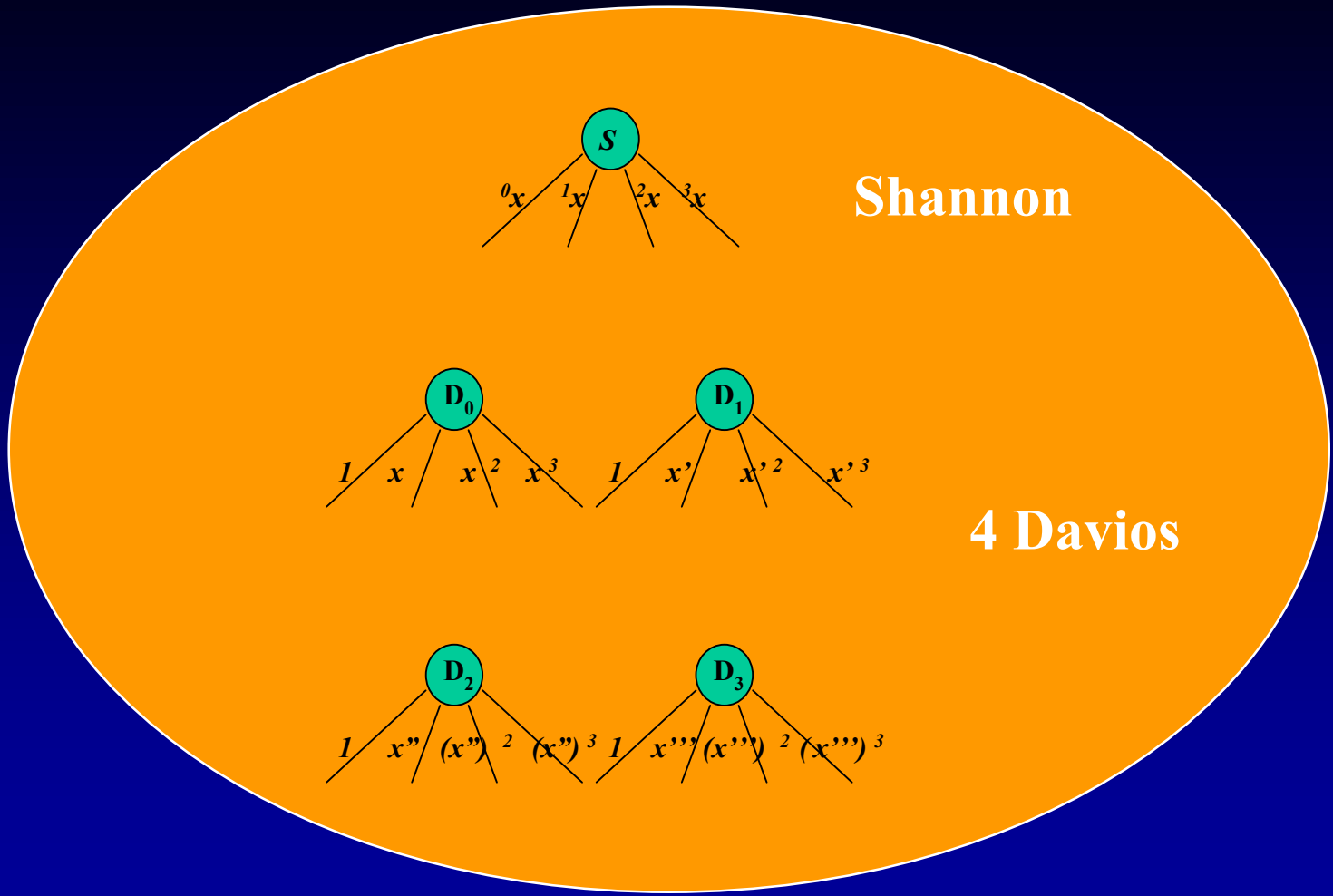
$$f = {}^{0}x\, f_0 + {}^{1}x\, f_1 + {}^{2}x\, f_2 + {}^{3}x\, f_3$$

Shannon

$$f = 1 \cdot f_0 + x\,(f_1+3f_2+2f_3) + (x)^2(f_1+2f_2+3f_3)+(x)^3(f_0+f_1+f_2+f_3)$$
$$f = 1 \cdot f_1 + (x')\,(f_0+2f_2+3f_3) + (x')^2(f_0+3f_2+2f_3) + (x')^3(f_0+f_1+f_2+f_3)$$
$$f = 1 \cdot f_2 + (x'')(3f_0+2f_1+f_3) + (x'')^2(2f_0+3f_1+f_3) + (x'')^3(f_0+f_1+f_2+f_3)$$
$$f = 1 \cdot f_3 + (x''')(f_2+3f_1+2f_0) + (x''')^2(f_2+2f_1+3f_0) + (x''')^3(f_0+f_1+f_2+f_3)$$

Davio

The corresponding 4-valued Shannon and Davio DTs for single variable are as follows, respectively:

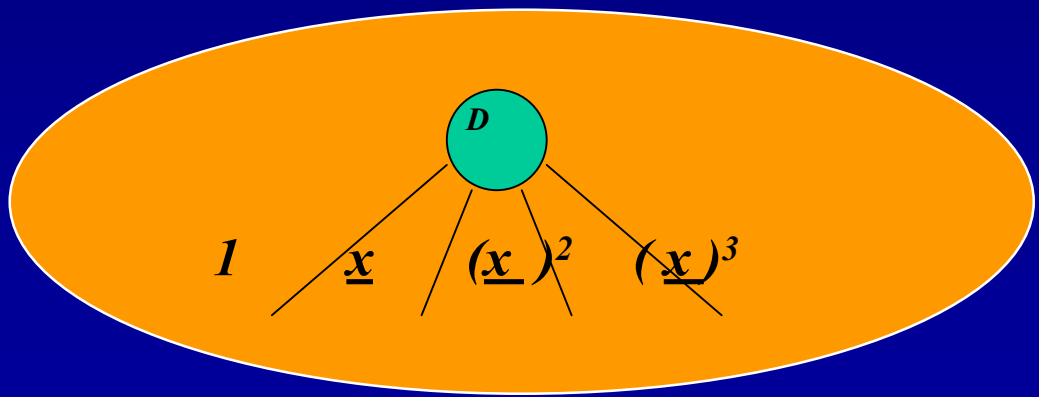


Generalized-Davio nodes

- S/D trees have proven to be useful in minimizing 2-valued expressions.

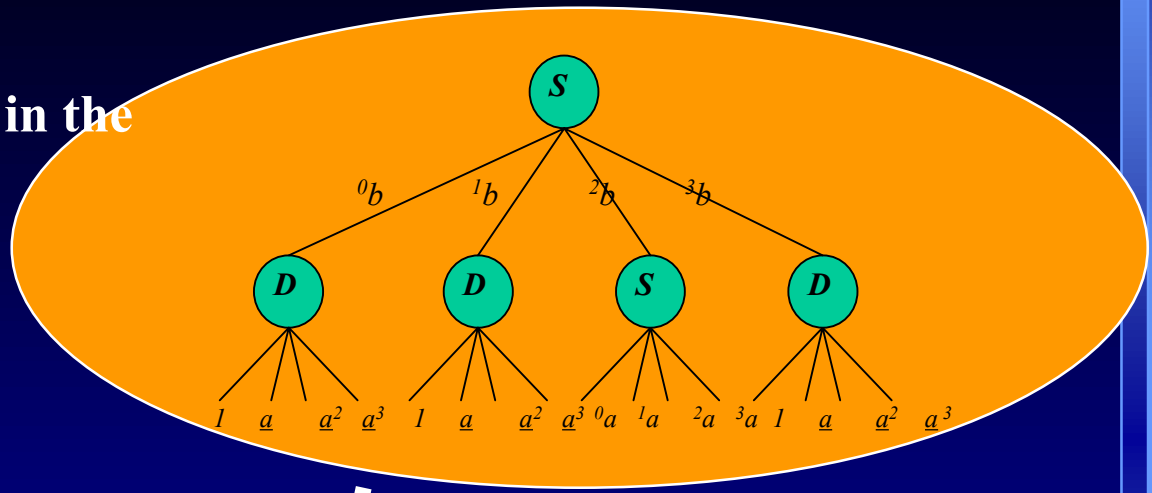
  - the creation of the generalized-Davio nodes that generate all possible Davio expansions.

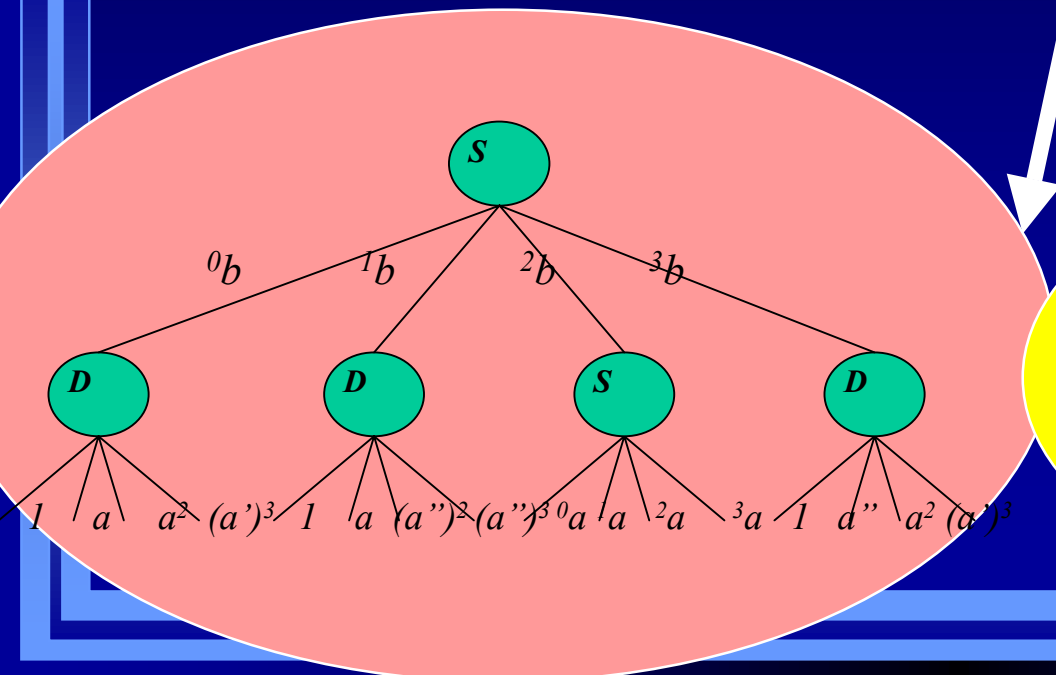
- This is generalized for 4-valued logic as follows:

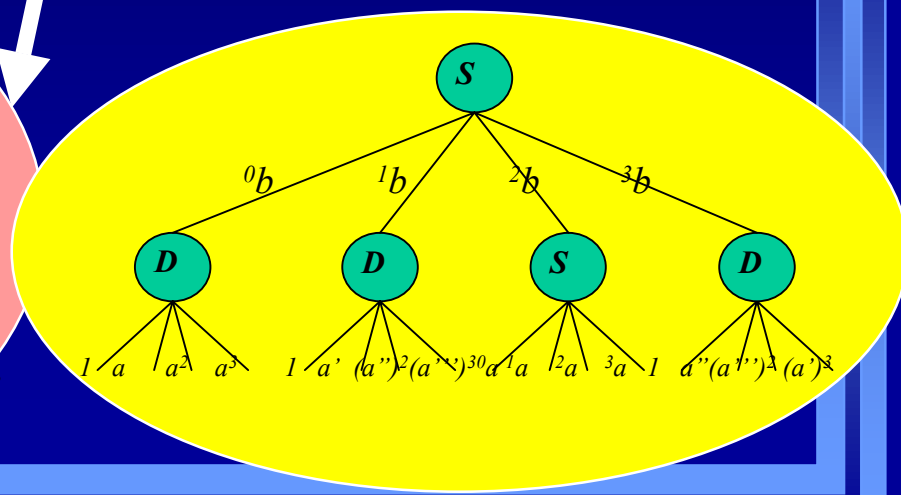


Utilizing such Generalized-Davio nodes in DTs, one obtains many possible corresponding quaternary DTs from the corresponding quaternary S/D tree.

This can be illustrated as in the following example:







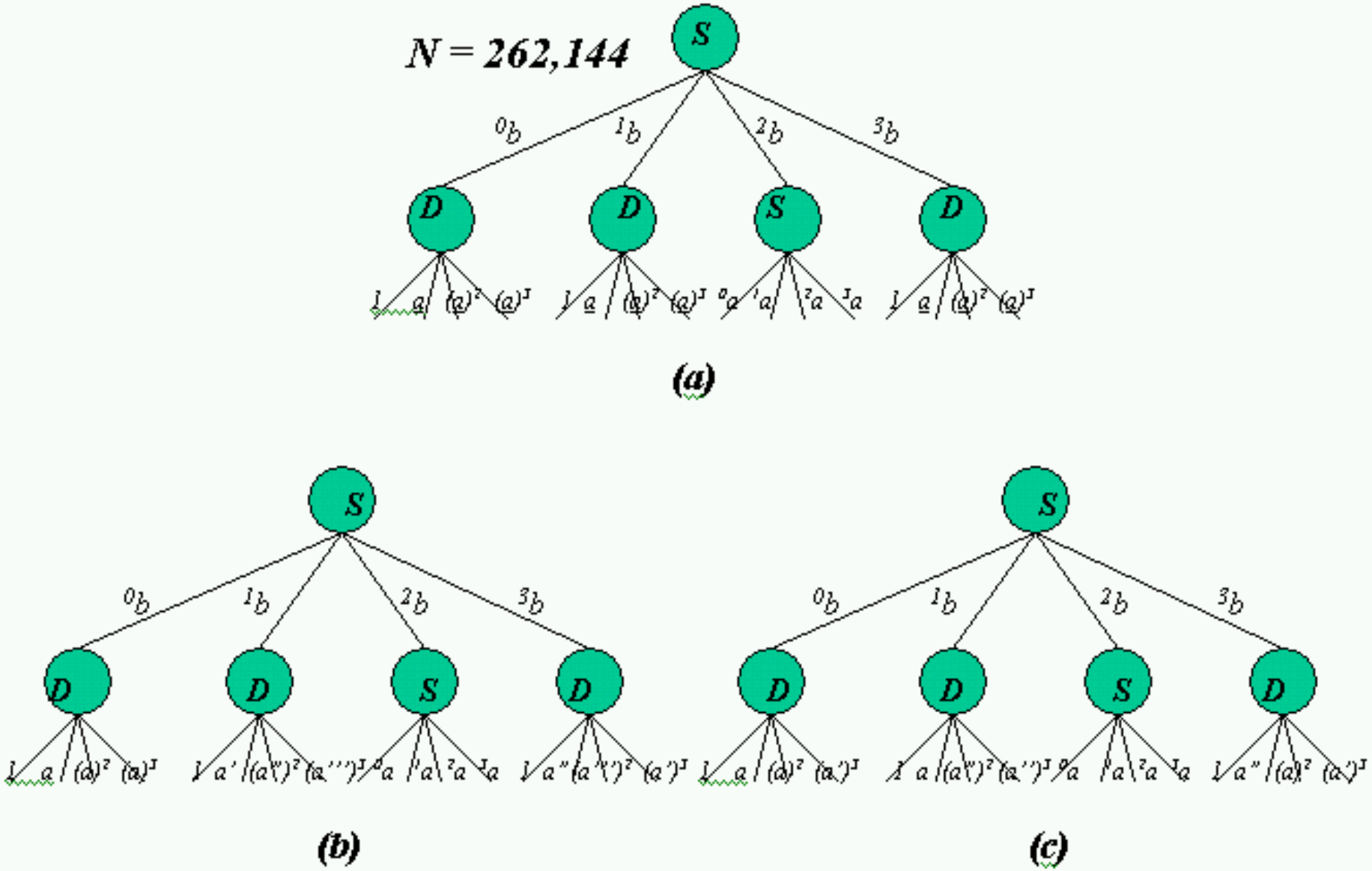


Figure 4. A quaternary S/D tree for two variables of order (b, a) with two Shannon nodes and three Generalized Davio nodes (a), and some of the quaternary trees that it generates (b and c).

Counting of the forms

•The number of possible **Inclusive Forms** (**IFs**) that can be generated from such quaternary S/D trees is very big.

• Counting such numbers is important in searching for minimum GFSOP forms, as the count provides an upper bound for the exhaustive search for minimal forms in the S/D space.

• The following formula provides the count for such quaternary inclusive forms:

$$\sum_{k1=0}^{(4)^{N-1}} \sum_{k2=0}^{(4)^{N-2}} \cdots \sum_{kN=0}^{(4)^{0}} \{ [\frac{4^{(N-1)}!}{(4^{(N-1)} - k_1)! k_1!} \cdots \frac{4^{0}!}{(4^{0} - k_N)! k_N!}] [(4^{3.(4)^{0}})^{k_1} (4^{3.(4)^{1}})^{k_2} \cdots (4^{3.(4)^{(N-1)}})^{k_N}] \}$$

where **N** is the **number of variables.**

• **For instance,** for two variables we obtain approximately **2.99483809211 * 10^{14}** quaternary IFs.

• Due to such huge numbers, one expects to find minimal GFSOP forms in such huge space of total forms.

The following formula provides the counting of IFs for

any radix,

arbitrary number of variables,

and arbitrary algebraic structures,

as follows:

$$\sum_{k_1=0}^{(n)^{N-1}} \sum_{k_2=0}^{(n)^{N-2}} \cdots \sum_{k_N=0}^{(n)^{0}} \{ [\frac{n^{(N-1)}!}{(n^{(N-1)} - k_1)! k_1!} \cdots \frac{n^{0}!}{(n^{0} - k_N)! k_N!}] [(n^{(n-1).(n)^{0}})^{k_1} (n^{(n-1).(n)^{1}})^{k_2} \cdots (n^{(n-1).(n)^{(N-1)}})^{k_N}] \}$$

where N is number of variables, and n is the radix.

- Utilizing MVL map representation, we can easily prove that there are 4^{16} = 4,294,967,296 quaternary functions of two variables, and $2.99483809211*10^{14}$ quaternary Inclusive Forms generated by the S/D trees.
- Thus, on the average every function of two variables can be realized in approximately 69,729 ways.
- This high number of realizations means that most functions of two variables are realized with less than five expansions, and all functions with at most five expansions.

From the previous general formula, it can be immediately observed that the formula is complicated enough

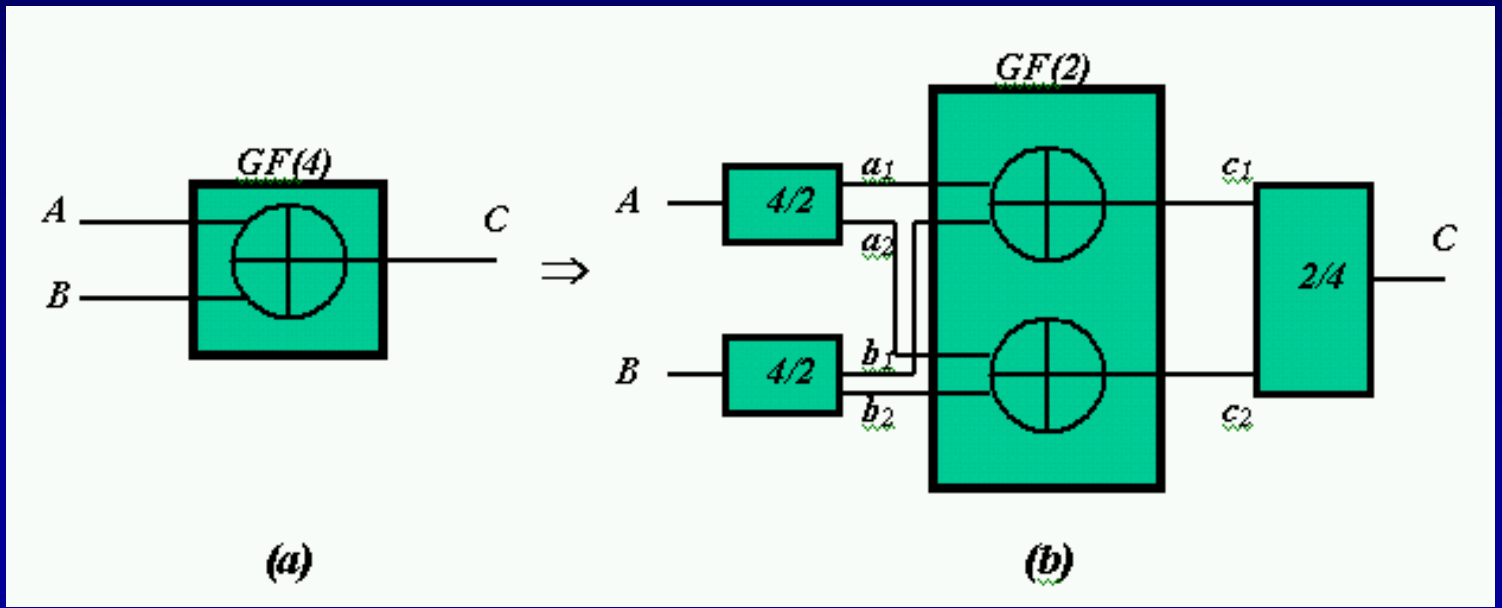
such that for high number of variables and high radix it will be virtually impossible to be executed using an ordinary PC due to the extensive amount of time that will be needed.

One solution is to use some patterns for counting ➔ The IF triangles.

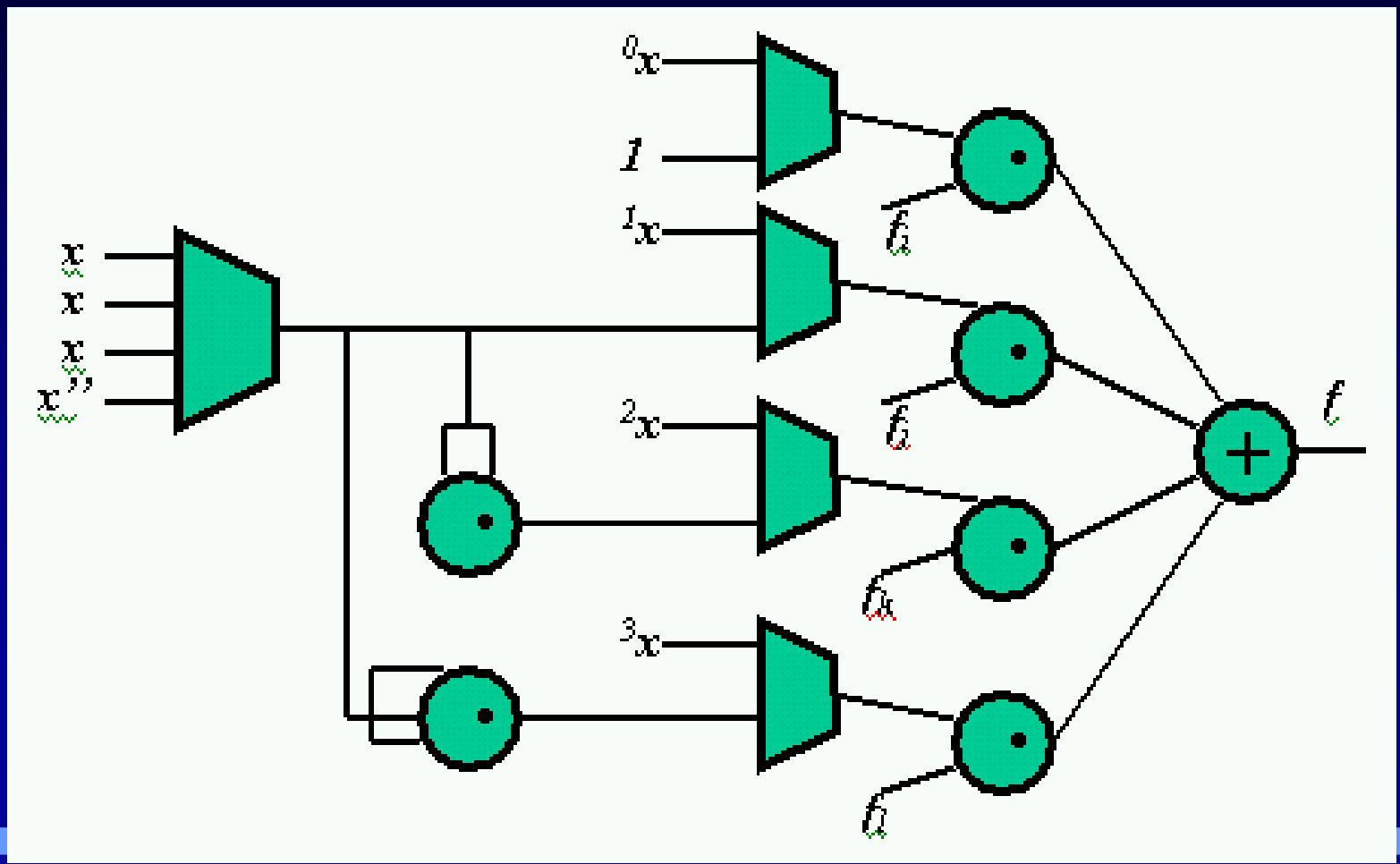
Functions with two variables are attractive in logic synthesis since many functional decomposition methods exist that produce two control inputs for primitive cells in a library of standard cells (such as a multiplexer with two address lines).

Universal Logic Modules (ULMs) can be produced for pairs of control variables that generalize Shannon and Davio expansion modules.

Realization of GF(4) addition (a) as GF(2) addition (b) (i.e. *vector of EXORs*).



Quaternary ULM that produces quaternary Shannon expansions, and all quaternary Davio expansions



Therefore, a fast method to calculate the number of Inclusive Forms for functions with two input variables over an arbitrary radix of Galois field is very useful.

The following pattern is therefore very useful for counting IFs for pairs of variables.

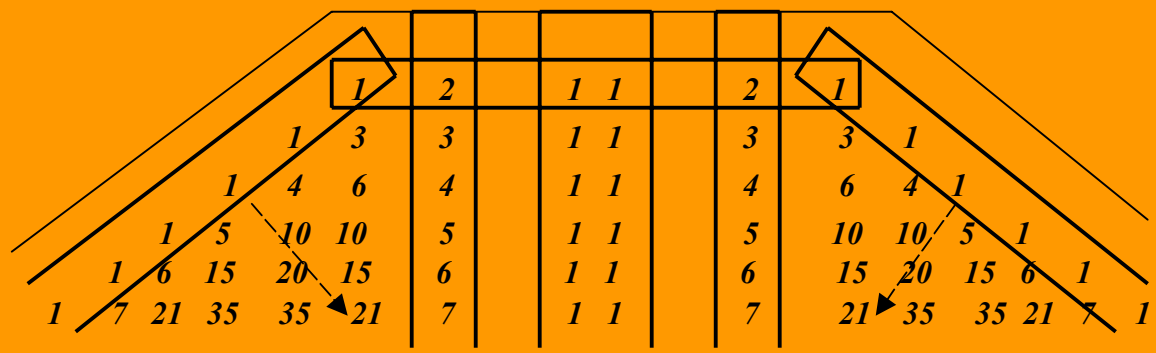
Such pattern makes the process of counting forms for two variables much easier, and much less time consuming.

$IF_{n,2}$ Triangles

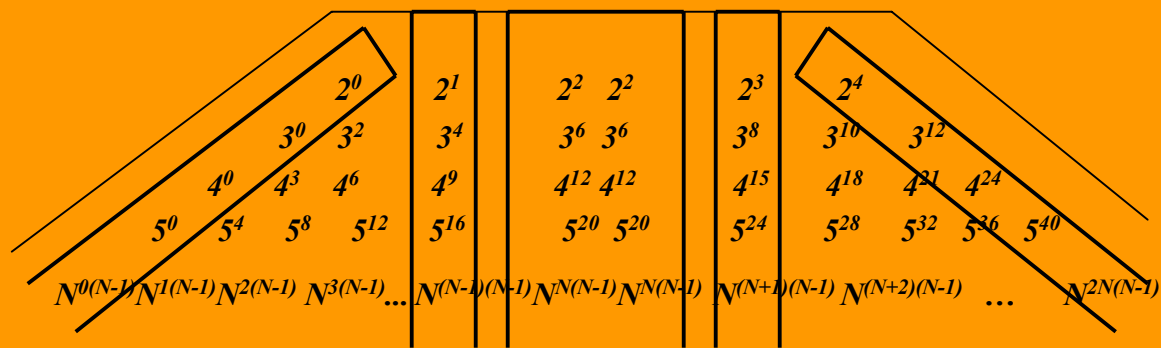
the Triangle of Coefficients

the Triangle of Values

for a fast calculation of the number of Inclusive Forms for an arbitrary radix Galois field and functions of two input variables (N=2).



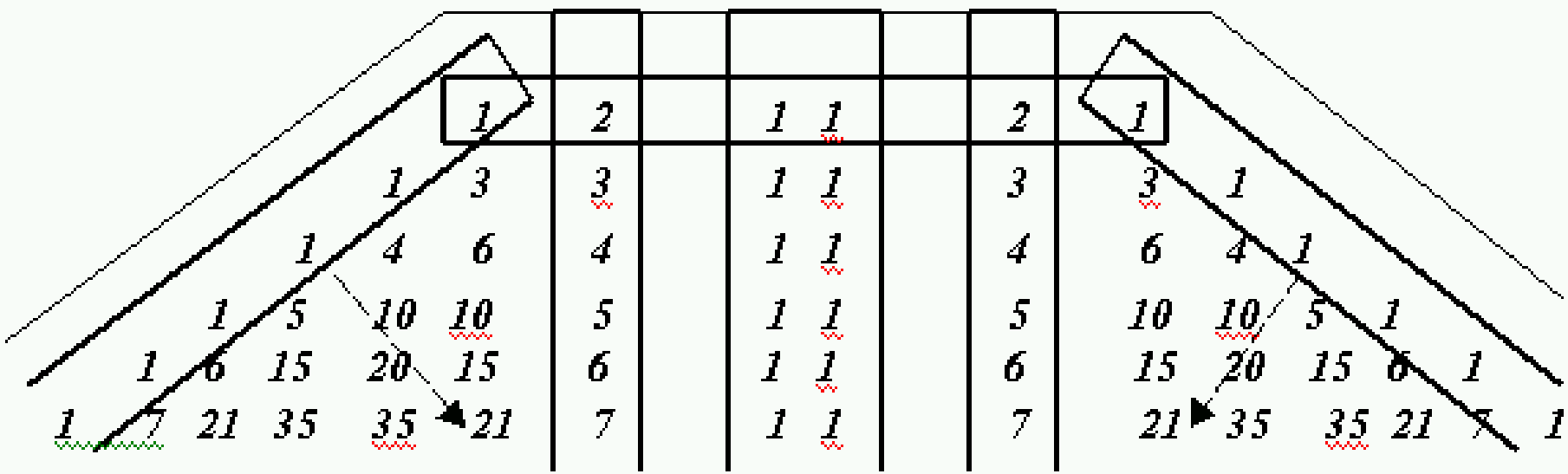
(a)



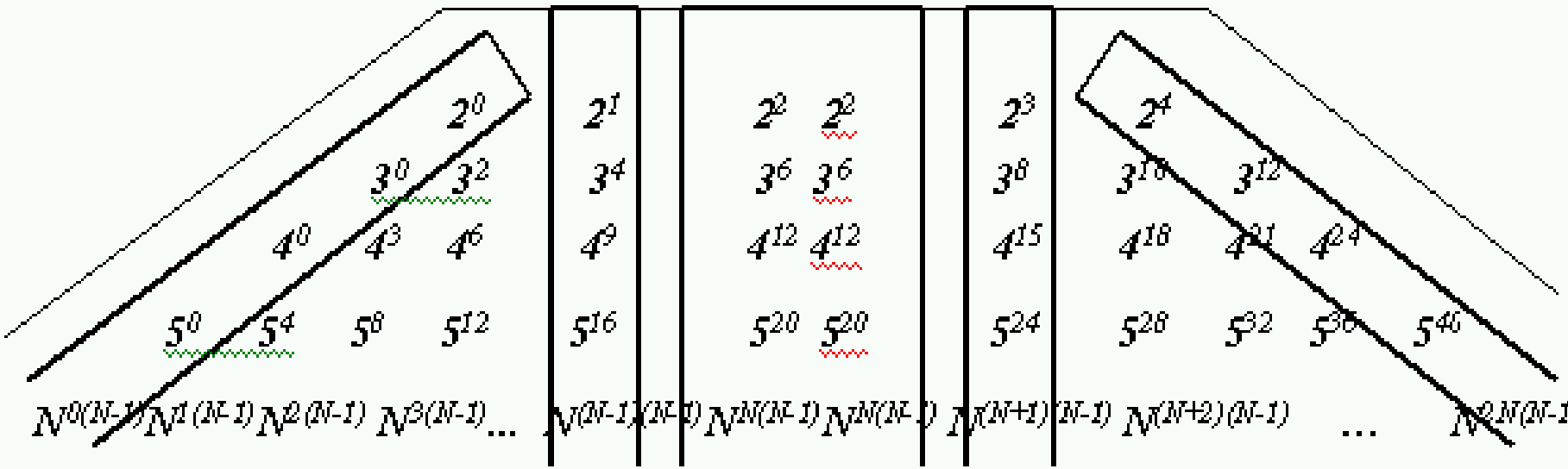
(b)

$IF_{n,2}$ Triangles; the Triangle of Coefficients (a) and the Triangle of Values (b) for a fast calculation of the number of Inclusive Forms for an arbitrary radix Galois field and functions of two input variables (N=2).

the Triangle of Coefficients



the Triangle of Values



- **Example.** Utilizing the *$IF_{n,2}$ Triangles* we can calculate the number of Inclusive Forms for GF(2), GF(3), and GF(4) for two variables, respectively, as follows:

- $\Phi_{2,2} = 1 \cdot 2^0 + 2 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^2 + 2 \cdot 2^3 + 1 \cdot 2^4 = 1 + 4 + 4 + 4 + 16 + 16 = 45$
- $\Phi_{3,2} = 1 \cdot 3^0 + 3 \cdot 3^2 + 3 \cdot 3^4 + 1 \cdot 3^6 + 1 \cdot 3^6 + 3 \cdot 3^8 + 3 \cdot 3^{10} + 1 \cdot 3^{12} = 730{,}000$
- $\Phi_{4,2} = 1 \cdot 4^0 + 4 \cdot 4^3 + 6 \cdot 4^6 + 4 \cdot 4^9 + 1 \cdot 4^{12} + 1 \cdot 4^{12} + 4 \cdot 4^{15} + 6 \cdot 4^{18} + 4 \cdot 4^{21} + 1 \cdot 4^{24} = 2.99483809211 * 10^{14}$

Conclusions

- We introduced a new family of canonical forms: **Quaternary Galois Generalized Inclusive Forms**
- We proved earlier that binary GIFs contain all exact minimum ESOPs
- We showed that binary GIFs lead to a significant reduction of the search space for exact minimum ESOP.

Further Research

Further research should concentrate on:

- investigating the properties of binary and multiple-valued GIFs,
- creating efficient algorithms for exact and heuristic **GFSOP minimization** based on them
- investigating applications to reversible logic.

•S/D DTs create the challenge of inventing new algorithms for the search in such big S/D spaces for minimal forms.

• Such difficulties already exist in the 2-valued case.

• We expect that creation of such algorithms for multi-valued S/D DTs will be very challenging.

• The hope is in the implicit methods.