Information Theoretic Approach to Minimization of Logic Expressions, Trees, Decision Diagrams and Circuits

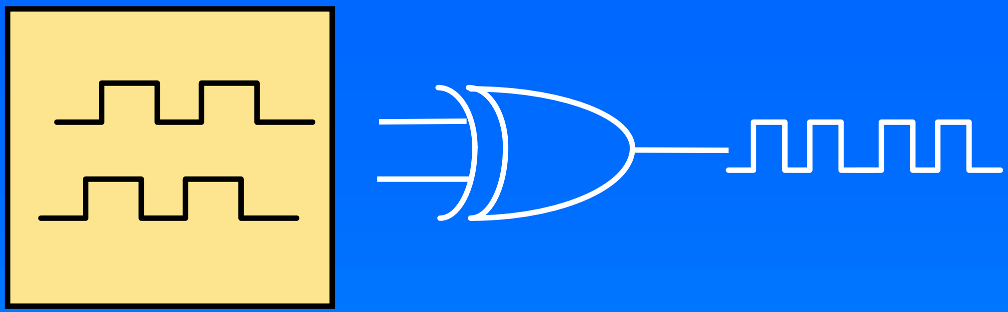
Outline

- Background
- Information Theoretic Model of Decision Trees (DTs) Design
- Minimization of Trees and Diagrams in Various Algebras
  - Arithmetic logic Expressions
  - Polynomial Expressions over GF(4)
- Experimental Study
- Summary and Future Work

Outline

- Information Theoretic Model of Free **Galois Decision Tree** Design
- Information Theoretic Model of Free **Word-Level Decision Tree** Design
- Galois-Sum of Galois-Multipliers **Circuit** Minimization
- **Arithmetical Expressions** Minimization Algorithm
- High generality of this type of methods

Shannon entropy



$p_{|f=1}$

$p_{|f=0}$

Entropy *H(f)* is a measure of switching activity

$$H(f) = p_{|f=0} \log_2 p_{|f=0} + p_{|f=1} \log_2 p_{|f=1}$$

Definition

- **Conditional entropy $H(f \mid x)$ is the information of event f under the assumption that a given event x had occurred**

- **Mutual information $I(f;x)$ is a measure of uncertainty removed by knowing x :**

$$I(f;x) = H(f) - H(f|x)$$

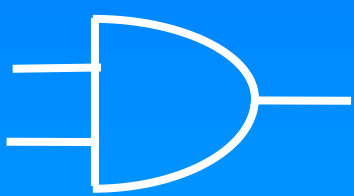
Shannon entropy

The information in an event *f* is a quantitative measure of the amount of uncertainty in this event

$$H(f) = - \sum_{i} p_{|f=i} \log_2 p_{|f=i}$$

Probability of 1 in f

Example.



x_1	x_2	f
0	0	0
0	1	0
1	0	0
1	1	1

$$H(f) = -(1/4)\log_2(1/4) - (3/4)\log_2(3/4) = 0.81\ bit$$

Probability of 0 in f

Definitions: Information theoretic measures

- Conditional entropy $H(f \mid x)$ is the information of event f under assumption that a given event x had occurred

$$H(f \mid x) = -\sum_i p_{|x=i} H(f|x=i)$$

Example.

x_1	x_2	f
0	0	0
0	1	0
1	0	0
1	1	1

$$H(f|x_1) = -(1/2)\cdot 0 - (1/2)\cdot 1 = 0.5 \text{ bit}$$

$$H(f_{|x1=0}) = -(2/2)\log_2(2/2) - (0/2)\log_2(0/2) = 0 \text{ bit}$$

$$H(f_{|x1=1}) = -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) = 1 \text{ bit}$$

Definitions: Information theoretic measures

- **Mutual information** $I(f;x)$ **is a measure of uncertainty removed by knowing** x **:**

$$I(f;x) = H(f) - H(f|x)$$

Example.

x_1	x_2	f
0	0	0
0	1	0
1	0	0
1	1	1

0.81 *bit*

0.5 *bit*

$$I(f;x_1) = H(f) - H(f|x_1) = 0.31\ bit$$

Mutual information

Entropy

Conditional Entropy

History

Not known to logic synthesis community but known to AI people

- **1938** - *Shannon expansion*
- **1948** - *Shannon entropy*
- **1980-** - *Shannon expansion + Shannon entropy for minimization of decision trees*
- *ID3*
- *C4.5 - Ross Quinlan*

Main results in application of information theory to logic functions minimization

- 1965 *ID3 algorithm* - a prototype
- 1990 A. Kabakciouglu et al.
  - »AND/OR decision trees design based on entropy measures
- 1993 A. Lloris et al.
  - »Minimization of multiple-valued logic functions using AND/OR trees
- 1998 D. Simovici, V.Shmerko et al.
  - »Estimation of entropy measures on Decision Trees

Application of Information Theory to Logic Design

- Logic function decomposition:
  - L. Jozwiak (Netherlands)
- Testing of digital circuits :
  - V. Agrawal, P. Varshney (USA)
- Estimation of power dissipation:
  - M. Pedram (USA)
- Logic functions minimization:

Example of ID3 algorithm

Attribute			Class
Furry?	*Age?*	*Size?*	*(Decision)*
Yes	Old	Large	Lion
No	Young	Large	Not Lion
Yes	Young	Medium	Lion
Yes	Old	Small	Not Lion
Yes	Young	Small	Not Lion
Yes	Young	Large	Lion
No	Young	Small	Not Lion
No	Old	Large	Not Lion
0.607	0.955	0.5	

Entropy

Furry - *Yes:* 3 Lions, 2 Non-Lion

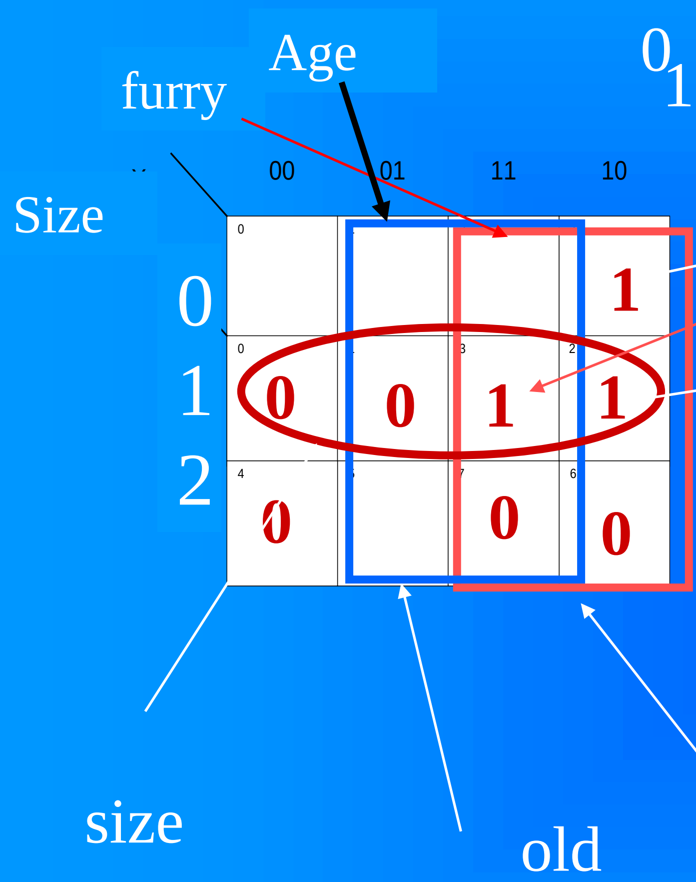
No: 0, 3 Non-Lions

$H = (5/8)\ H_{furry} + (3/8)\ H_{notfurry} = 0.607$

5 yeses

$H_{furry} = log(5/8)$

3 nos

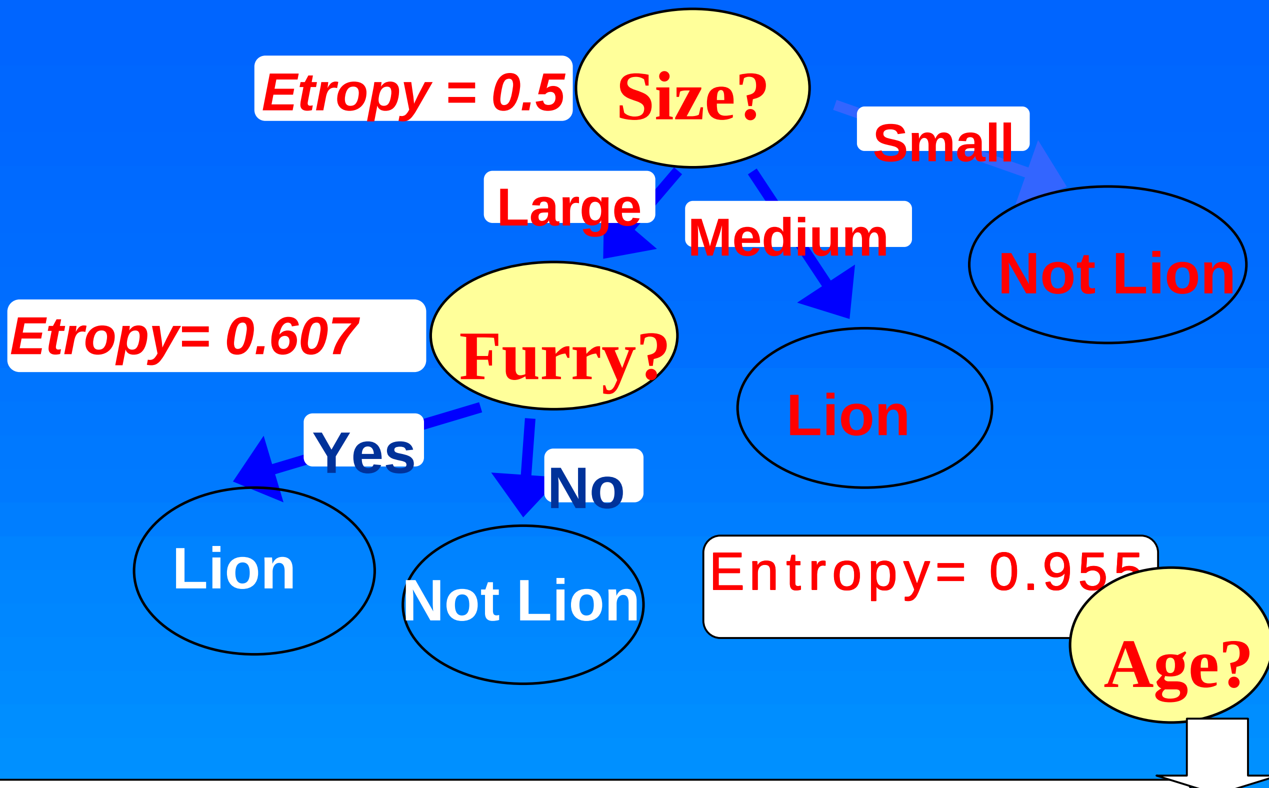


Attribute			Class
Furry?	*Age?*	*Size?*	*(Decision)*
1 **Yes**	1 **Old**	1 **Large**	**Lion**
0 **No**	0 **Young**	**Large**	**Not Lion**
1 **Yes**	0 **Young**	0 **Medium**	**Lion**
Yes	**Old**	**Small**	**Not Lion**
Yes	**Young**	2 **Small**	**Not Lion**
Yes	**Young**	**Large**	**Lion**
No	**Young**	**Small**	**Not Lion**
No	**Old**	**Large**	**Not Lion**
0.607	**0.955**	**0.5**	**Entropy**

Furry **-** ***Yes:*** **3 Lions, 2 Non-Lion**
No: **0, 3 Non-Lions**

$$H = (5/8)\, H_{furry} + (3/8)\, H_{notfurry} = 0.607$$

Optimal decision tree



Decision: ***Age***

Where did the idea come from?

- **ID3 algorithm have been used for long time in machine-learning systems for trees**
- ***The principal paradigm:*** **learning classification rules**
- **The rules are formed from a set of training examples**
- **The idea can be used not only to trees**

Summary

Idea

Consider the truth table of a logic function as a special case of the decision table with variables replacing the tests in the decision table

Arithmetic Spectrum

Use arithmetic operations to make logic decisions

- **Artificial Intelligence**
- **Testing of digital circuits :**
- **Estimation of power dissipation:**
- **Logic functions minimization for new technologies (quantum - Victor Varshavsky)**

Arithmetic Spectrum

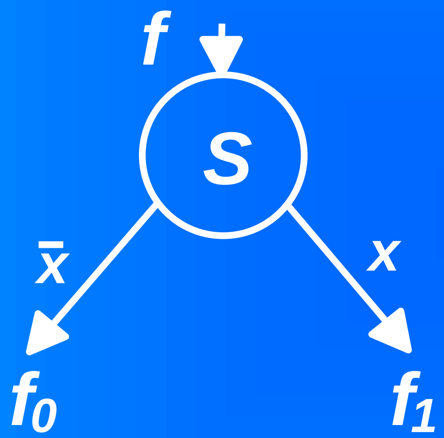
Use arithmetic operations to make logic decisions

- A or B becomes A + B - AB in arithmetics
- A exor B becomes A + B -2AB in arithmetics
- A and B becomes A * B in arithmetics
- not (A) becomes (1 - A) in arithmetics

Recall

Shannon expansion

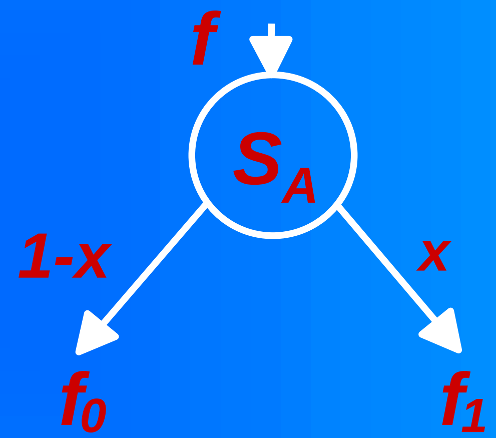
Bit-Level



$$f = \bar{x}\, f_{|x=0} \vee x\, f_{|x=1}$$

$$f = \bar{x}\, f_{|x=0} \oplus x\, f_{|x=1}$$

Word-Level

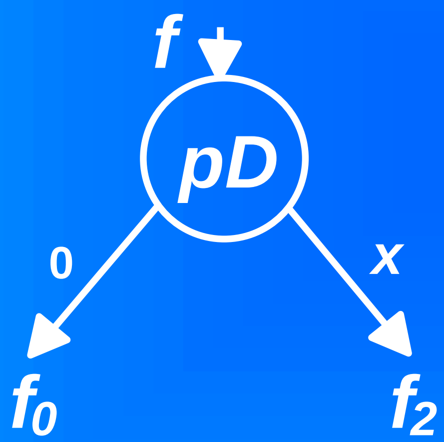


$$f = (1-x)\, f_{|x=0} + x\, f_{|x=1}$$

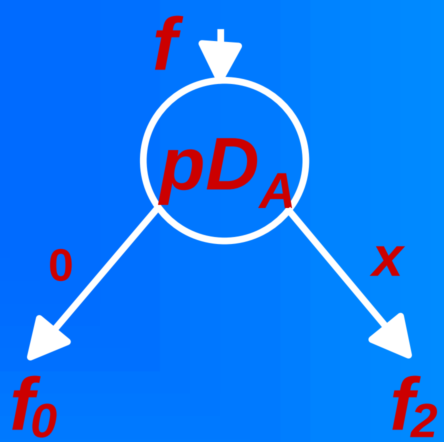
Recall

Positive Davio expansion

Bit-Level



Word-Level



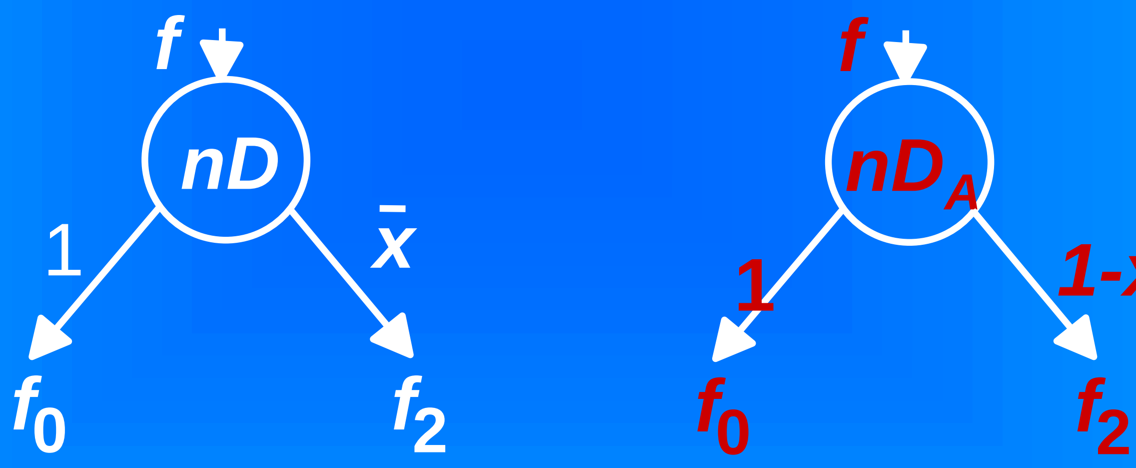
$$f = f_{|x=0} \oplus x(f_{|x=0} \oplus f_{|x=1})$$

$$f = f_{|x=0} + x(f_{|x=1} - f_{|x=0})$$

Recall

Negative Davio expansion

Bit-Level | **Word-Level**



$f = f_{|x=1} \oplus \bar{x}\,(f_{|x=0} \oplus f_{|x=1})$

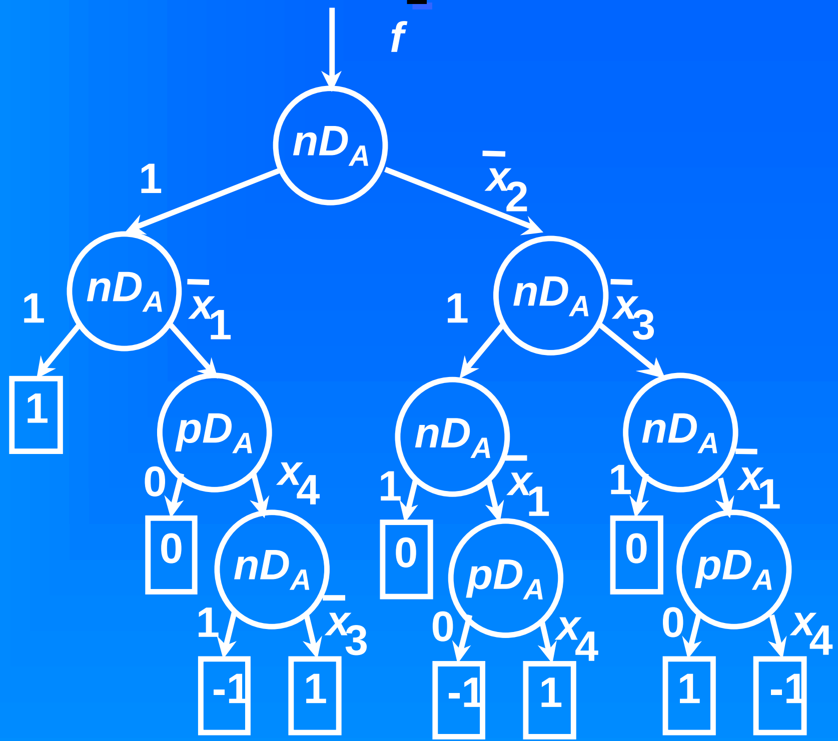
$f = f_{|x=1} + (1-x)(f_{|x=0} - f_{|x=1})$

New

Types of Decision Trees

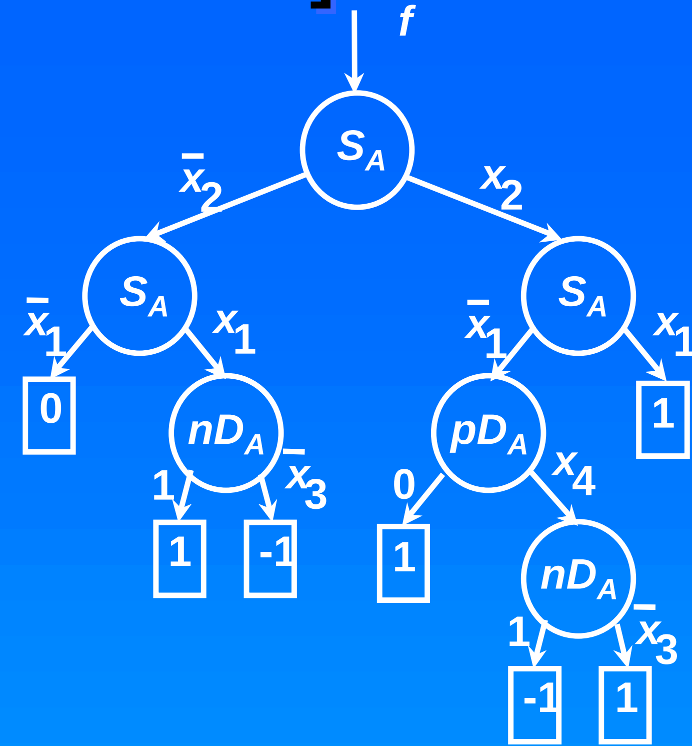
Example: switching function

$[0000\ 1110\ 0011\ 1111]^T$



pseudo Binary Moment Tree (BMT):

$\{pD_A, nD_A\}$



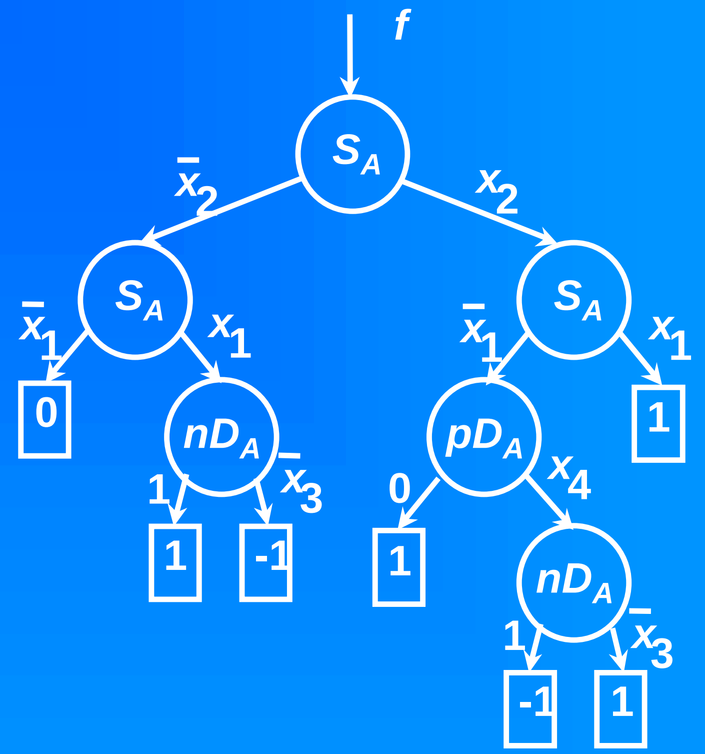
Free pseudo Kronecker Binary Moment Tree (KBMT):

$\{S_A, pD_A, nD_A\}$

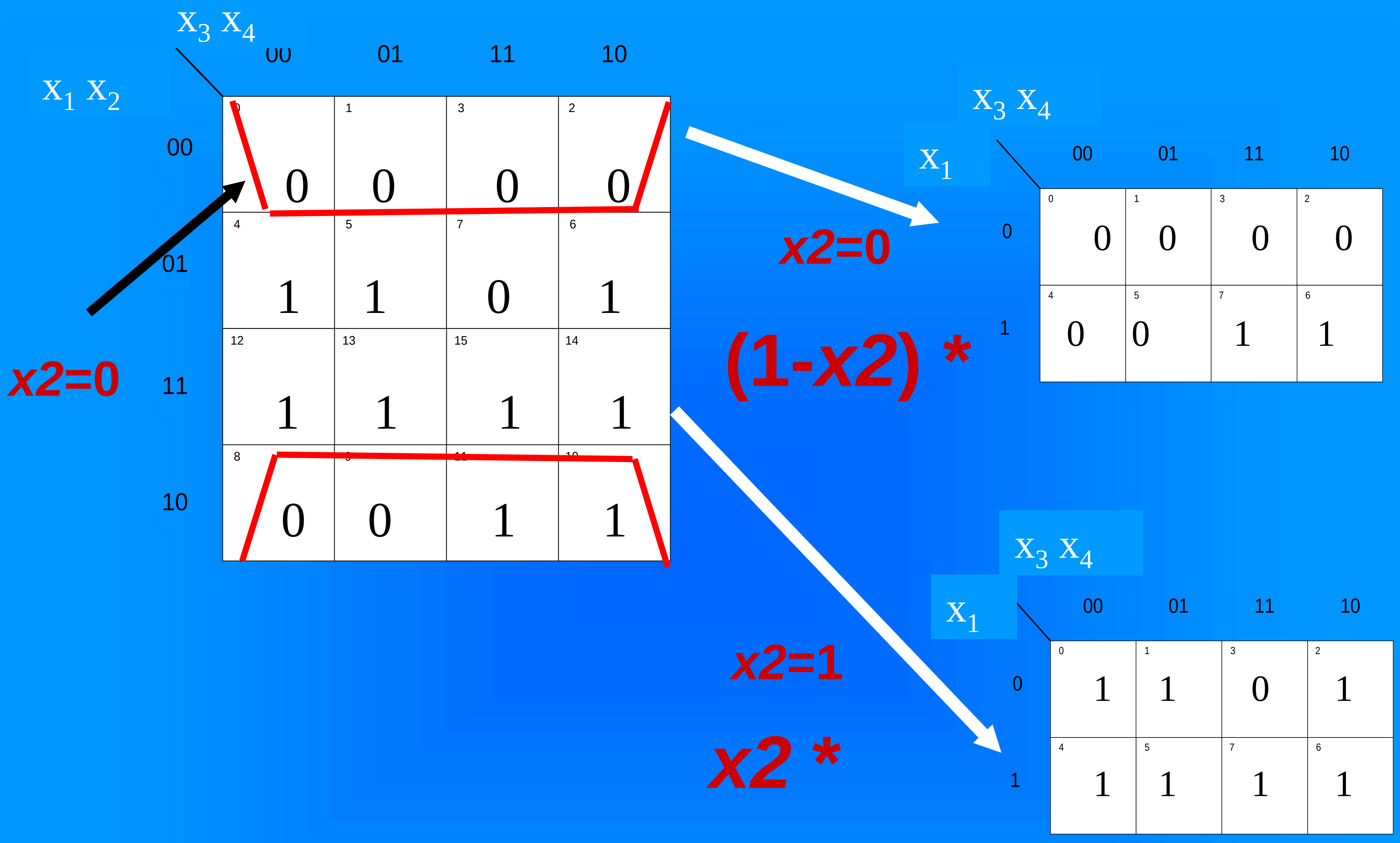
$[0000\ 1110\ 0011\ 1111]^T$

Free pseudo Kronecker Binary Moment Tree (KBMT): $\{S_A, pD_A, nD_A\}$

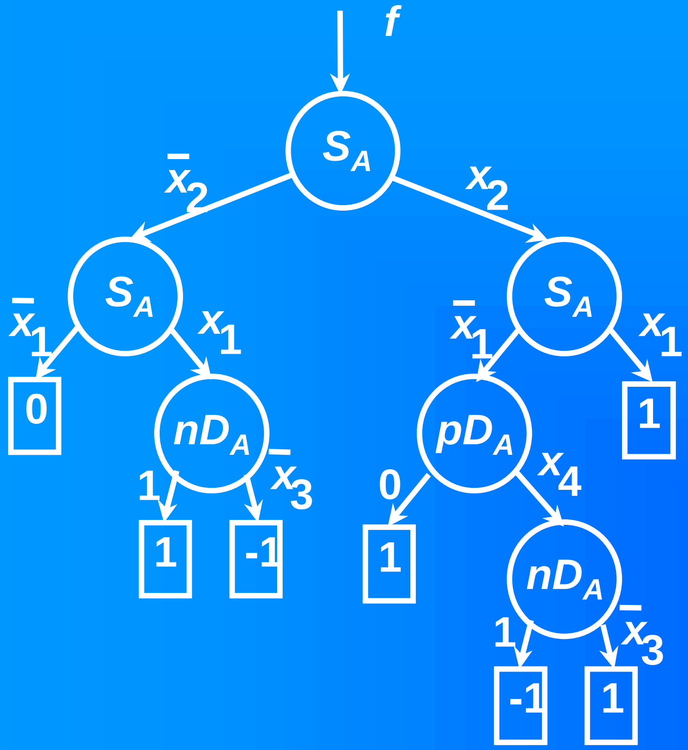
$x_1 x_2$ \ $x_3 x_4$	00	01	11	10
00	(0) 0	(1) 0	(3) 0	(2) 0
01	(4) 1	(5) 1	(7) 0	(6) 1
11	(12) 1	(13) 1	(15) 1	(14) 1
10	(8) 0	(9) 0	(11) 1	(10) 1



$$f = (1-x2)\, f_{|x2=0} + x2\, f_{|x2=1}$$



$$f = (1-x2)\, f_{|x2=0} + x2\, f_{|x2=1}$$



x_1 \ $x_3 x_4$	00	01	11	10
0	0	0	0	0
1	0	0	1	1

$$f = (1-x1)\, f_{|x1=0} + x1\, f_{|x1=1}$$

x1=0	0	0	0	0
x1=1	0	0	1	1

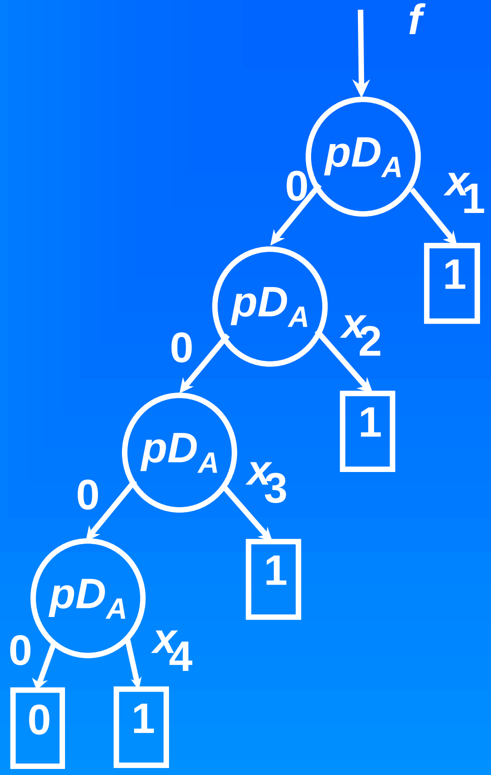
$$f = f_{|x3=1} + (1-x3)(f_{|x3=0} - f_{|x3=1})$$

x3=1

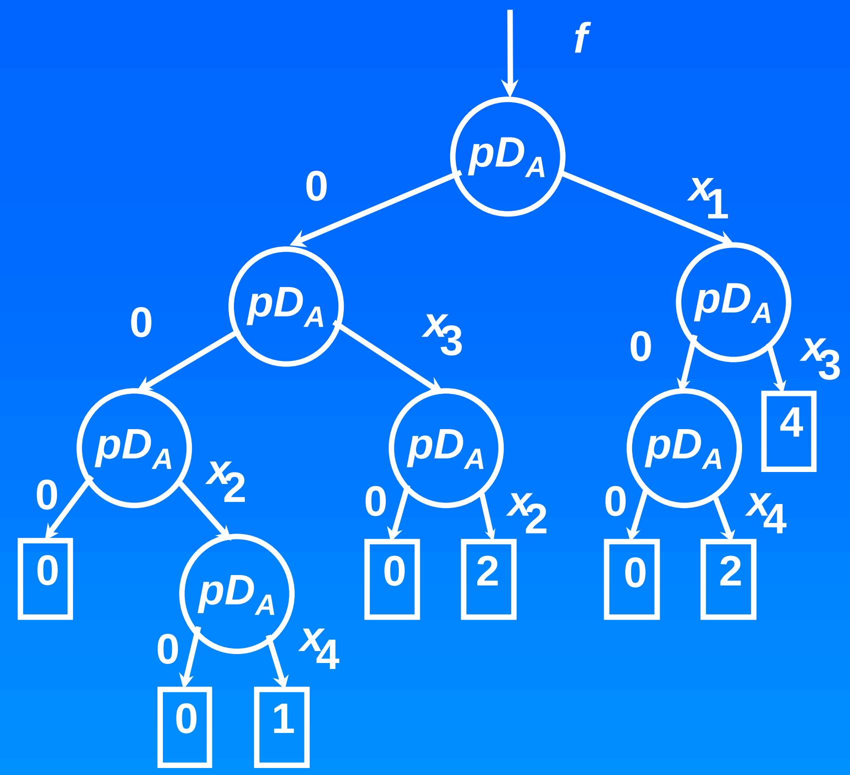
$f_{|x3=0} - f_{|x3=1}$	0	0	1	1

Example

Word-Level Decision Trees for arithmetic functions



2-bit half-adder



2-bit multiplier

Problems of Free Word-Level Decision Tree Design

- **Variable ordering**
- **Selection of decomposition**

Benefit in Minimization

Information theoretical measures

$x_1 x_2$ \ $x_3 x_4$	00	01	11	10
00	(0) 0	(1) 0	(3) 0	(2) 0
01	(4) 1	(5) 1	(7) 0	(6) 1
11	(12) 1	(13) 1	(15) 1	(14) 1
10	(8) 0	(9) 0	(11) 1	(10) 1

For a given switching function $[0000\ 1110\ 0011\ 1111]^T$

- ***Entropy***

$$H(f) = -(7/16)\log_2(7/16) - (9/16)\log_2(9/16) = 0.99 \textit{ bit}$$

- ***Conditional Entropy***

$$H(f|x_1) = -(5/16)\log_2(5/8) - (3/16)\log_2(3/8) - (2/16)\log_2(2/8) - (6/16)\log_2(6/8) = 0.88 \textit{ bit}$$

- ***Mutual Information***

$$I(f;x_1) = 0.99 - 0.88 = 0.11 \textit{ bit}$$

Example

$x_1 x_2$ \ $x_3 x_4$	00	01	11	10
00	(0) 0	(1) 0	(3) 0	(2) 0
01	(4) 1	(5) 1	(7) 0	(6) 1
11	(12) 1	(13) 1	(15) 1	(14) 1
10	(8) 0	(9) 0	(11) 1	(10) 1

9 ones

7 zeros

Entropy is large when there is as many zeros as ones

Entropy does not take into account where are they located

Entropy

$H(f) = -(7/16)\,log_2(7/16) - (9/16)\,log_2(9/16) = 0.99$ *bit*

- ***Conditional Entropy***

$H(f\,|x_1) = -(5/16)\,log_2(5/8) - (3/16)\,log_2(3/8)$
$- (2/16)\,log_2(2/8) - (6/16)\,log_2(6/8) = 0.88$ *bit*

- ***Mutual Information***

$I(f;x_1) = 0.99 - 0.88 = 0.11$ *bit*

Entropy is measure of function complexity

Now the same idea will be applied to Galois Logic

Shannon and Davio expansions in GF(4)

Shannon entropy

Information theoretic criterion in minimization of polynomial expressions in GF(4)

New Idea

Linearly Independent Expansions in any Logic

Shannon entropy

Information theoretic criterion in minimization of trees, lattices and flattened forms in this logic

Idea

Merge two concepts

Shannon decomposition

\+

Shannon entropy

Information theoretic approach to logic functions minimization

New Information Model

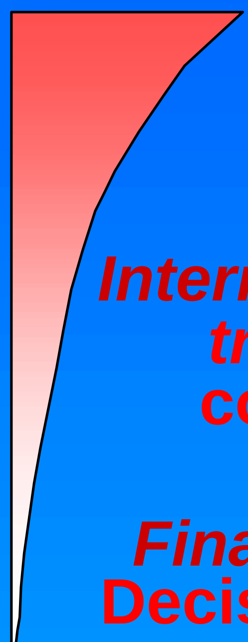
Entropy is reduced:

Information is increased:

H(*f*)

H(*f*|*tree*)

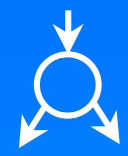
H(*f*|*Tree*)=0



***Initial state*:**
No decision *tree* for a given function

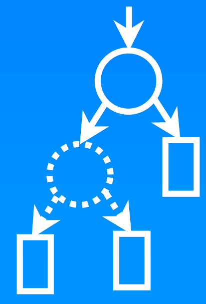
...

***Intermediate state*:**
***tree* is part-constructed**



...

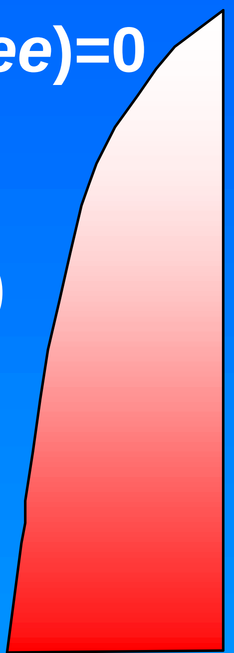
***Final state*:**
Decision *tree* represents the given function



I(*f*;*Tree*)=0

I(*f*;*tree*)

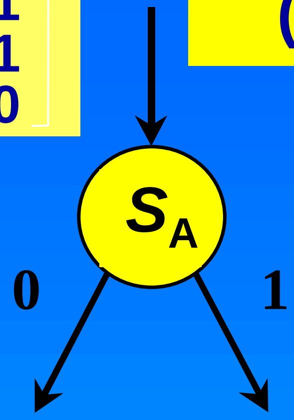
I(*f*;*Tree*)=*H*(*f*)



IDEA: Shannon Entropy + Decision Tree

x_1	x_2	f
0	0	0
0	1	1
1	0	1
1	1	0

$$H(f) = -(2/4)\log_2(2/4) - (2/4)\log_2(2/4) = 1\ bit$$



x_1	x_2	$f_{\mid x1=0}$
0	0	0
0	1	1

x_1	x_2	$f_{\mid x1=1}$
1	0	1
1	1	0

$$H(f_{\mid x1=0}) = -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) = 1\ bit$$

$$H(f_{\mid x1=1}) = -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) = 1\ bit$$

New

Information theoretic measures for arithmetic

- ***Arithmetic Shannon***

$$H^{S_A}(f\,|x) = p_{|x=0}\cdot H(f_{|x=0}) + p_{|x=1}\cdot H(f_{|x=1})$$

- ***Arithmetic positive Davio***

$$H^{pD_A}(f\,|x) = p_{|x=0}\cdot H(f_{|x=0}) + p_{|x=1}\cdot H(f_{|x=1} - f_{|x=0})$$

- ***Arithmetic negative Davio***

$$H^{nD_A}(f\,|x) = p_{|x=1}\cdot H(f_{|x=1}) + p_{|x=0}\cdot H(f_{|x=0} - f_{|x=1})$$

Information theoretic criterion for Decision Tree design

- Each pair (x,ω) brings the portion of information

$$I(f\,;x) = H(f\,) - H^{\omega}(f\,|x)$$

- The criterion to choose the variable x **and the decomposition type** ω

$$H^{\omega}(f\,|x)=min(H^{\omega_j}(f\,|x_i)|\ pair\ (x_i,\omega_j))$$

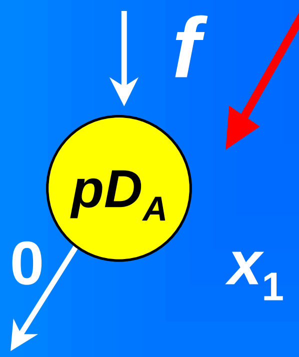
Algorithm to minimize arithmetic expressions *INFO-A*

- **Evaluate information measures: $H^{\omega}(f\,|x_i)$ for each variable**

- **Pair (x, ω) that corresponds to $min(H^{\omega}(f\,|x))$ is assigned to the current node**

Example

How does the algorithm work?



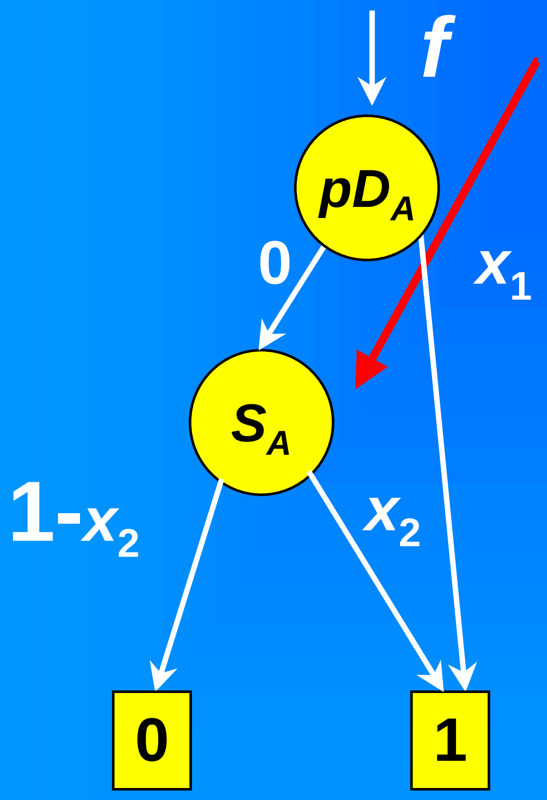
1. Evaluate Information measures for 1-bit half-adder:

$H^{S_A}(f\,|x_1)$, $H^{S_A}(f\,|x_2)$,
$H^{pD_A}(f\,|x_1)$, $H^{pD_A}(f\,|x_2)$,
$H^{nD_A}(f\,|x_1)$, $H^{nD_A}(f\,|x_2)$

2. Pair (x_1, pD_A) that corresponds to $min(H^{pD_A}(f\,|x_1))=0.5$ bit is assigned to the current node

Example

How does the algorithm work?



$f = x_1 + x_2$

1. Evaluate Information measures:

$H^{S_A}(f \mid x_2)$,

$H^{pD_A}(f \mid x_2)$,

$H^{nD_A}(f \mid x_2)$

2. Pair (x_2, S_A) that corresponds to $\min(H^{S_A}(f \mid x_2))=0$ is assigned to the current node

Main idea

Decision Table

Conversion with optimization of:

- variables ordering,
- decomposition type

Decision Tree

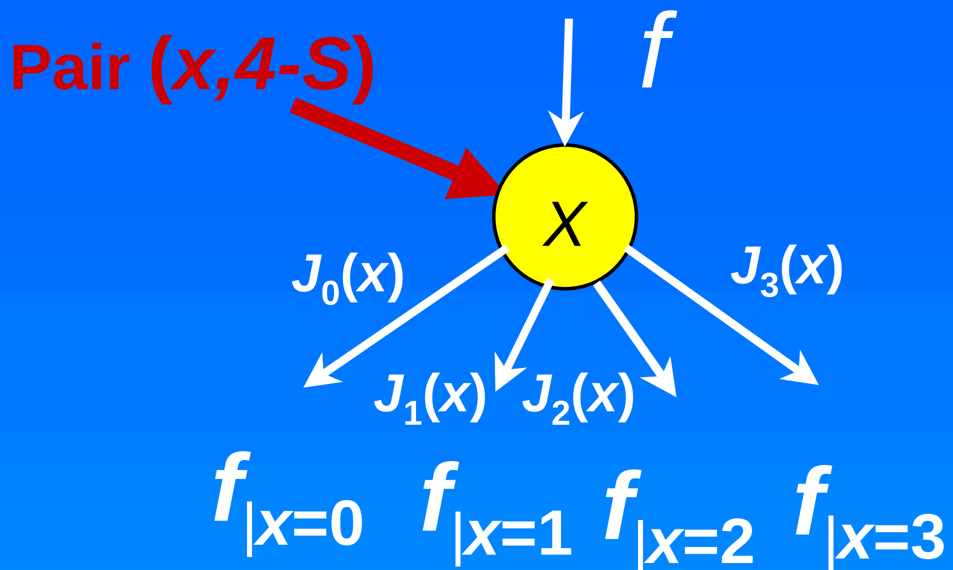
New information criterion

Decision Trees and expansion types

Multi-terminal GF(4)	*4-S*
Pseudo Reed-Muller GF(4)	*4-pD, 1-4-nD, 2-4-nD, 3-4-nD*
Pseudo Kronecker GF(4)	*4-S, 4-pD, 1-4-nD, 2-4-nD, 3-4-nD*

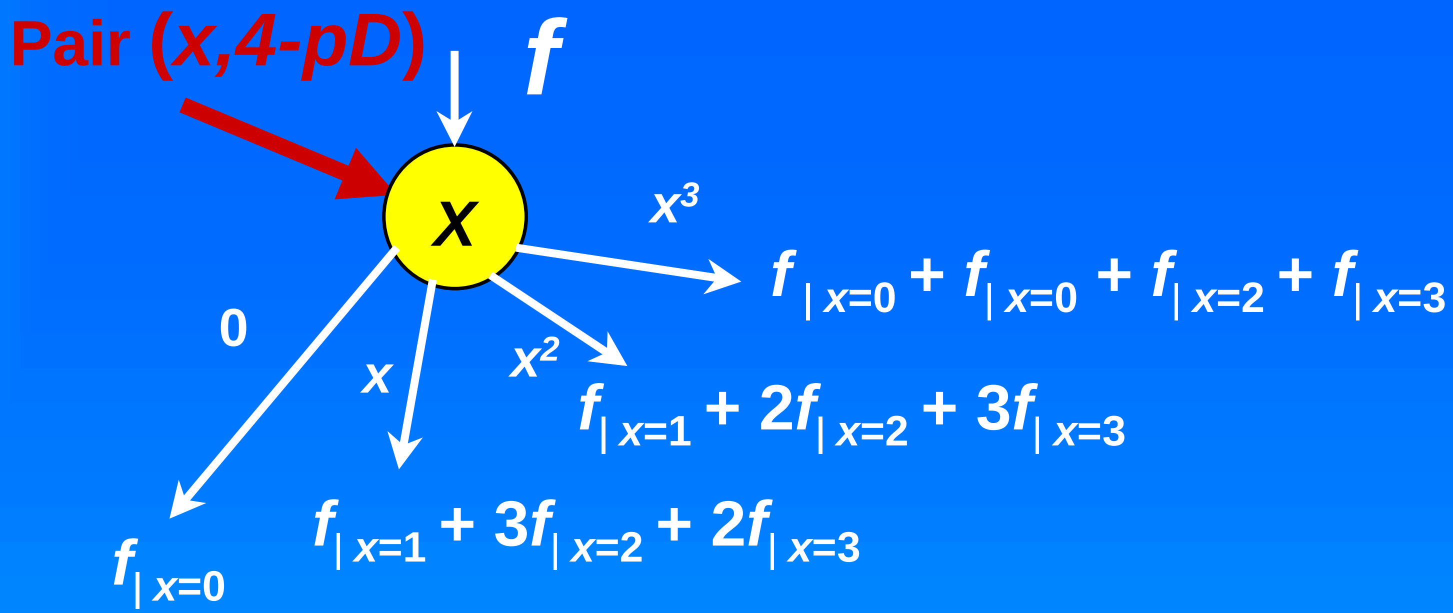
We always use FREE Decision Trees

Analogue of Shannon decomposition in GF(4)



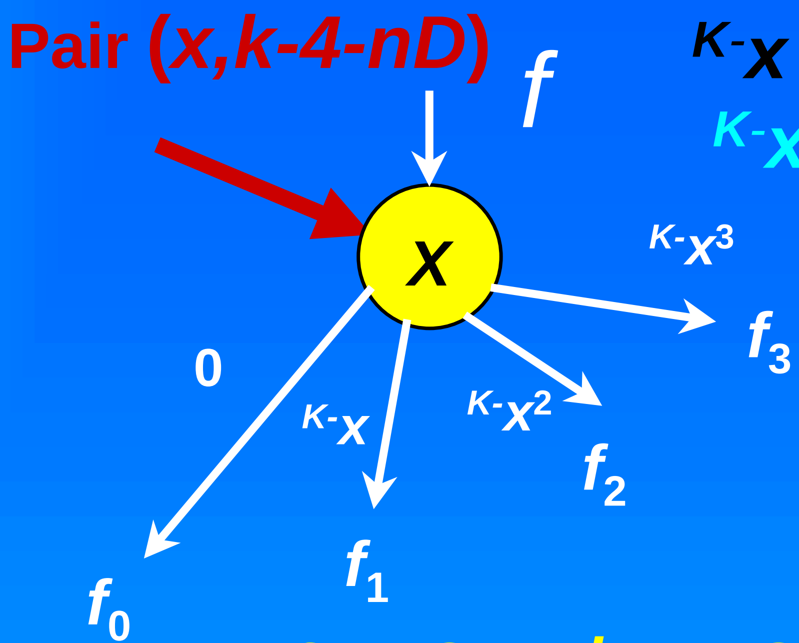
$$f = J_0(x)\cdot f_{|x=0} + J_1(x)\cdot f_{|x=1} + J_2(x)\cdot f_{|x=2} + J_3(x)\cdot f_{|x=3}$$

Analogue of positive Davio decomposition in GF(4)



$$f = f_{|x=0} + x\cdot(f_{|x=1} + 3f_{|x=2} + 2f_{|x=3}) \\ + x^2\cdot(f_{|x=1} + 2f_{|x=2} + 3f_{|x=3}) \\ + x^3\cdot(f_{|x=0} + f_{|x=0} + f_{|x=2} + f_{|x=3})$$

Analogue of negative Davio decomposition in GF(4)



${}^{K-}x$ is a complement of *x:*

${}^{K-}x = x + k$, $k = 1,\ldots,4$

$$f = f_0 + {}^{k-}x \cdot f_1 + {}^{k-}x^2 \cdot f_2 + {}^{k-}x^3 \cdot f_3$$

How to minimize polynomial expressions via Decision Tree

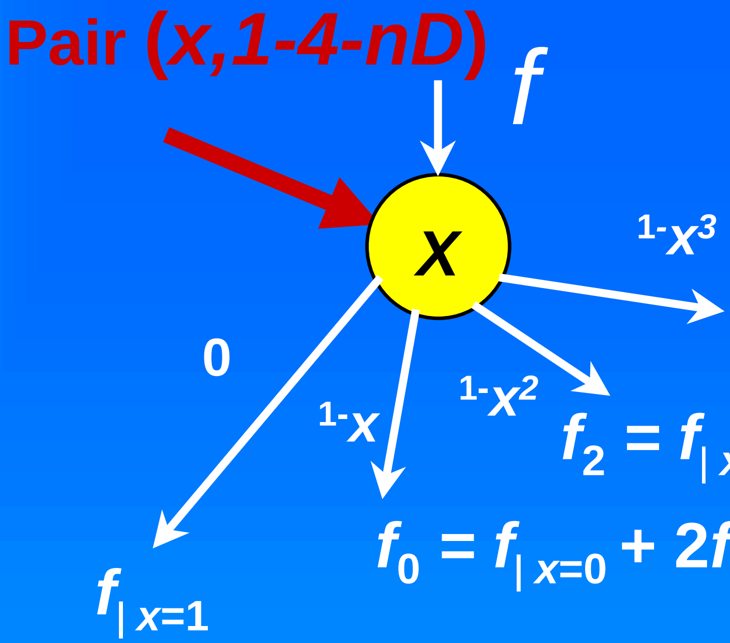
- A path in the Decision tree corresponds to a product term
- The best product terms (with minimal number of literals) to appear in the quasi-minimal form can be searched via Decision Tree design
- The order of assigning variables and decomposition types to nodes needs a criterion

Summary of GF(4) logic

Minimization of polynomial expressions in GF(4) means the design of Decision Trees with variables ordered by using some criterion

This is true for any type of logic.

Shannon entropy + decomposition in GF(4)



$$f_3 = f_{|\,x=0} + f_{|\,x=0} + f_{|\,x=2} + f_{|\,x=3}$$

$$f_2 = f_{|\,x=0} + 3f_{|\,x=2} + 2f_{|\,x=3}$$

$$f_0 = f_{|\,x=0} + 2f_{|\,x=2} + 3f_{|\,x=3}$$

$$H(f \mid x) = p_{|x=0} \cdot H(f_0) + p_{|x=2} \cdot H(f_2) + p_{|x=3} \cdot H(f_3) + p_{|x=1} \cdot H(f_{|x=1})$$

Information theoretic criterion for Decision Tree design

- Each pair (x,ω) carries a portion of information

$$I(f\,;x) = H(f\,) - H^{\omega}(f\,|x)$$

- The criterion to choose the variable x and the decomposition type ω

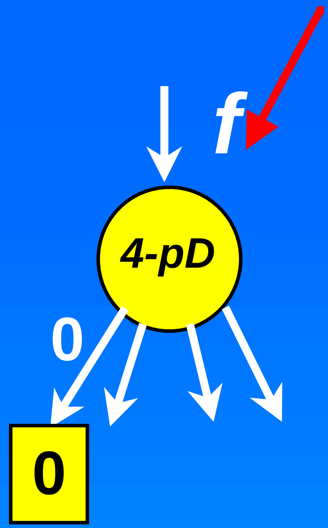
$$H^{\omega}(f\,|x)=min(H^{\omega_j}(f\,|x_i)|\;pair\;(x_i,\omega_j)\;)$$

INFO-MV Algorithm

- Evaluate information measures: $H^{\omega}(f\,|x_i)$ for each variable
- Pair (x,ω) that corresponds to $min(H^{\omega}(f\,|x))$ is assigned to the current node

Example:

How does the algorithm work?



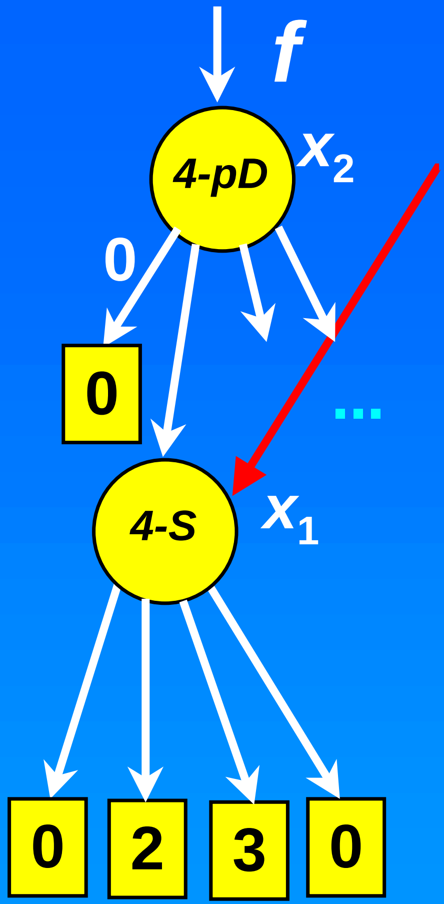
f=[000 0231 0213 0321]

1. Evaluate Information measures:

$H^{4\text{-}S}(f\,|x_1)$, $H^{4\text{-}S}(f\,|x_2)$,
$H^{4\text{-}pD}(f\,|x_1)$, $H^{4\text{-}pD}(f\,|x_2)$,
$H^{1\text{-}4\text{-}nD}(f\,|x_1)$, $H^{1\text{-}4\text{-}nD}(f\,|x_2)$
$H^{2\text{-}4\text{-}nD}(f\,|x_1)$, $H^{2\text{-}4\text{-}nD}(f\,|x_2)$
$H^{3\text{-}4\text{-}nD}(f\,|x_1)$, $H^{3\text{-}4\text{-}nD}(f\,|x_2)$

2. Pair $(x_2, 4\text{-}pD)$ that corresponds to $min(H^{4\text{-}pD}(f\,|x_2))=0.75$ *bit* is assigned to the current node

How does the algorithm work?



1. Evaluate information measures:
   $H^{4\text{-}S}(f\,|x_1)$, $H^{4\text{-}pD}(f\,|x_1)$, $H^{1\text{-}4\text{-}nD}(f\,|x_1)$, $H^{2\text{-}4\text{-}nD}(f\,|x_1)$, $H^{3\text{-}4\text{-}nD}(f\,|x_1)$
2. Pair $(x_1, 4\text{-}S)$ that corresponds to $\min(H^{4\text{-}S}(f\,|x_1))=0$ is assigned to the current node

Experiments

Plan of study

Comparison with arithmetic generalization of *Staircase* strategy

(Dueck et.al., Workshop on Boolean problems 1998)

Comparison with *INFO* algorithm (bit-level trees)

(Shmerko et.al., TELSIKS’1999)

INFO-A algorithm

Experiments

INFO-A against Staircase strategy

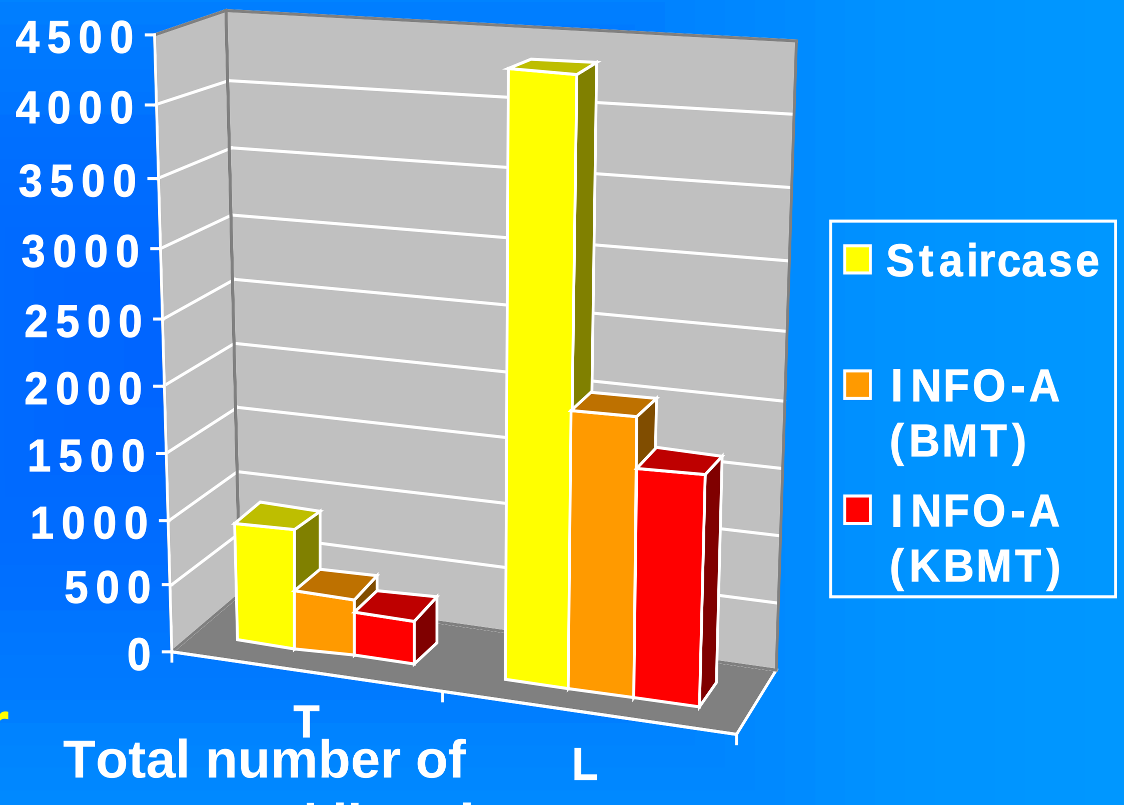
Test	*Staircase* (Dueck et.al. 98)	*INFO-A*
	L / t	*L / t*
xor5	80/0.66	80/0.00
squar5	56/0.06	24/0.00
rd73	448/0.80	333/0.01
newtpla2	1025/185.20	55/0.12
Total	1609/186.72	492/0.13

EFFECT 3.3 times

L / t - the number of literals / run time in seconds

INFO-A against table-based generalized *Staircase*

- Staircase strategy manipulates matrices
- *INFO-A* is faster and produces 70 % less literals
- BMT- free binary moment tree (word-level)
- KBMT - free Kronecker Binary Moment tree (word-level)



Total number of terms and literals, for 15 benchmarks

Experiments

INFO-A against bit-level algorithm *INFO*

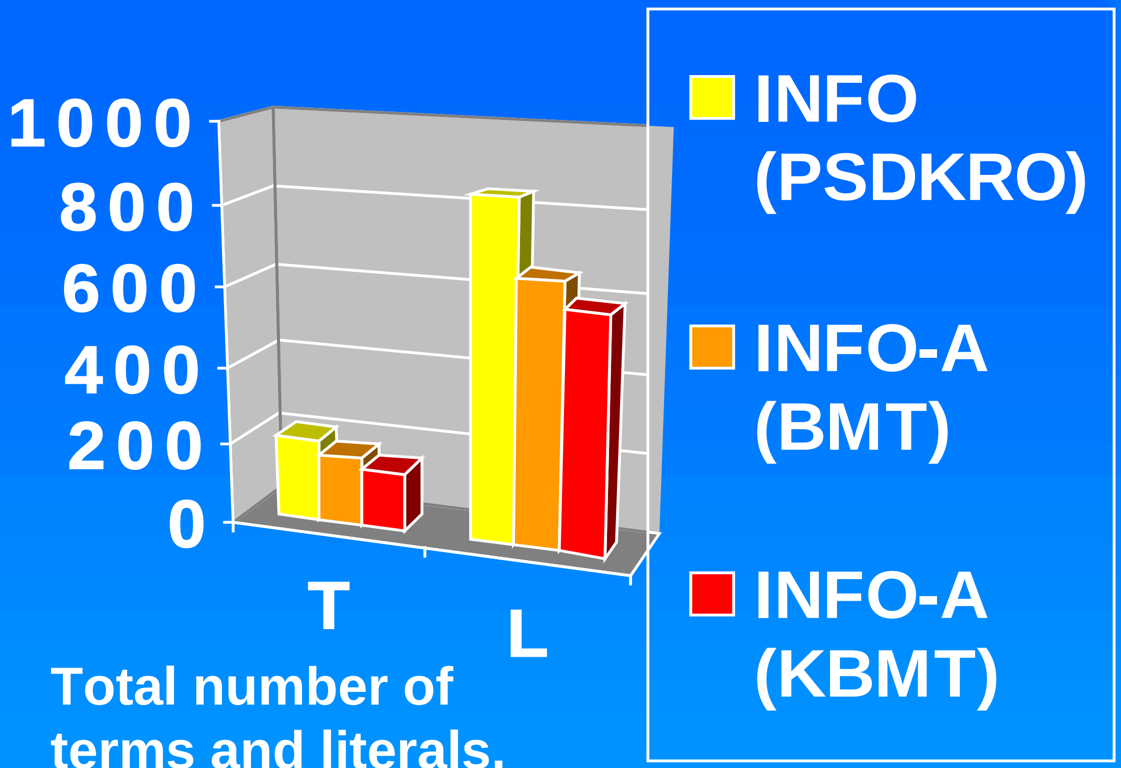
Test	*INFO* (Shmerko et.al. 99)	*INFO-A*
	T / t	*T / t*
xor5	5/0.00	31/0.00
z4	32/0.04	7/0.00
inc	32/0.20	41/0.45
log8mod	39/1.77	37/0.03
Total	109/2.01	116/0.48

EFFECT 4 times

T / t - the number of products / run time in seconds

Advantages of using Word-Level Decision Trees to minimize arithmetic functions (squar, adder, root, log)

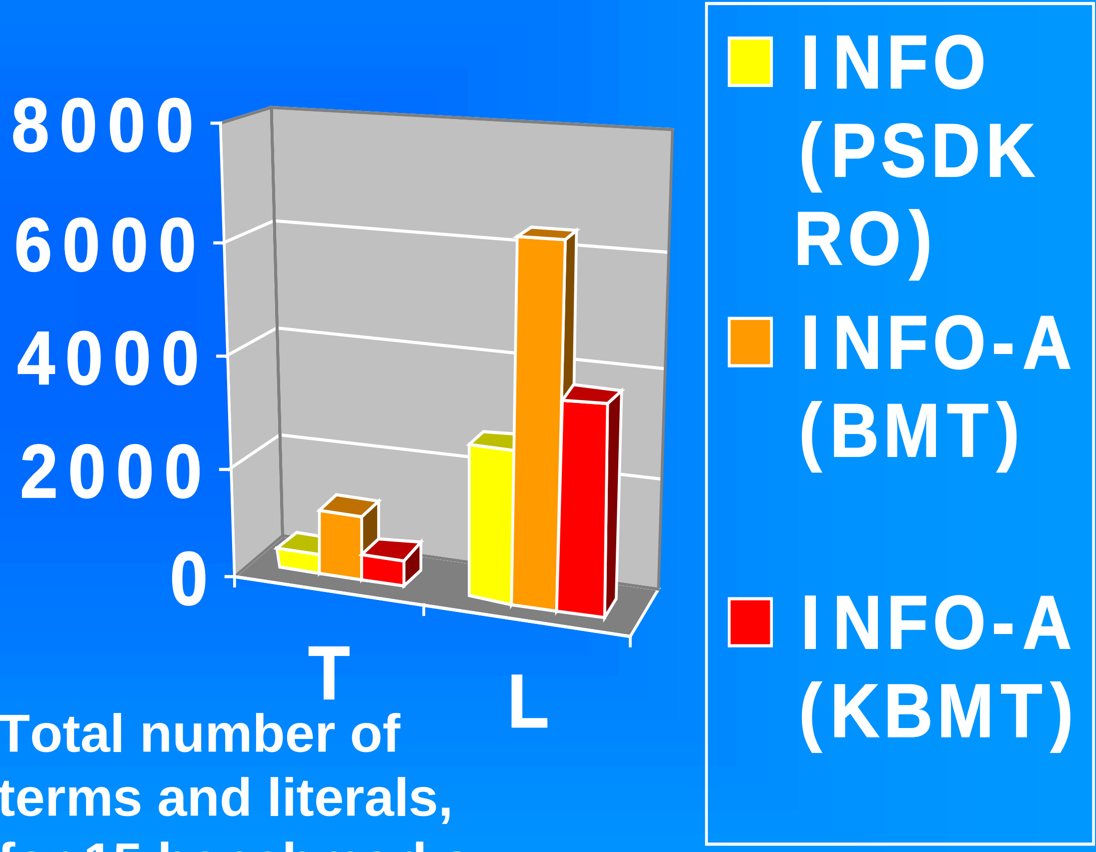
- PSDKRO - free pseudo Kronecker tree (bit-level)
- BMT - free binary moment tree (word-level)
- KBMT- free Kronecker Binary Moment tree (word-level)



Total number of terms and literals, for 15 benchmarks

Advantages of using bit-level DT to minimize symmetric functions

- PSDKRO - free pseudo Kronecker tree (bit-level)
- BMT - free binary moment tree (word-level)
- KBMT - free Kronecker Binary Moment tree (word-level)



Total number of terms and literals, for 15 benchmarks

Concluding remarks for arithmetic

What new results have been obtained?

- New information theoretic interpretation of arithmetic Shannon and Davio decomposition

- New technique to minimize arithmetic expressions via new types of word-level Decision Trees

What improvements did it provide?

70% products and 60% literals less against known Word-level Trees, for arithmetic functions

Now do the same for Galois Logic

Organization of Experiments

Symbolic Manipulations approach - *EXORCISM* (Song et.al.,1997)

Staircase strategy on Machine Learning benchmarks (Shmerko et.al., 1997)

INFO-MV algorithm

Experiments:

INFO against Symbolic Manipulation

Test	*EXORCISM* (Song and Perkowski 97)	*INFO*
bw	319 / 1.1	65 / 0.00
rd53	57 / 0.4	45 / 0.00
adr4	144 / 1.7	106 / 0.00
misex1	82 / 0.2	57 / 0.50
Total	602 / 3.4	273 / 0.50

EFFECT 2 times

L / *t* - the number of literals / run time in seconds

Experiments:
INFO-MV against Staircase strategy

Test	*Staircase* (Shmerko et.al., 97)	*INFO-MV*
monks1te	13 / 0.61	7 / 0.04
monks1tr	7 / 0.06	7 / 0.27
monks2te	13 / 0.58	7 / 0.04
monks2tr	68 / 1.27	21 / 1.29
Total	101 / 2.52	42 / 1.64

EFFECT 2.5 times

T / t - the number of terms / run time in seconds

Experiments:
4-valued benchmarks (*INFO-MV*)

Test	*Type of DT in GF(4)* *Multi-Terminal*	*Pseudo Reed-Muller*	*Pseudo Kronecker*
5xp1	256/ 1024	165/ 521	142/ 448
clip	938/ 4672	825/ 3435	664/ 2935
inc	115/ 432	146/ 493	65/ 216
misex1	29/ 98	48/ 108	15/ 38
sao2	511/ 2555	252/ 1133	96/ 437
Total	1849/ 8781	1436/ 5690	982/ 4074

***T / L* - the number of terms / literals**

Extension of the Approach

Minimization on **Word-level** Trees	**70%** products and **60%** literals less against known Word-level Trees
Minimization on **Ternary** Decision Trees	**15%** reduction of the number of products against ROETTD
Minimization of **incompletely** specified functions	**30%** improvement in the number of products against EXORCISM

Summary

Contributions of this approach

- New information theoretic interpretation of arithmetic Shannon and Davio decomposition
- New information model for different types of decision trees to represent AND/EXOR expressions in GF(4)
- New technique to minimize 4-valued AND/EXOR expressions in GF(4) via FREE Decision Tree design
- Very general approach to any kind of decision diagrams, trees, expressions, forms, circuits, etc
- Not much published - opportunity for our class and M.S or PH.D. thesis

Future work

Calculation of information measures on Decision Diagrams (no truth table is needed) → Dramatical extension of size of the problem

Extension toward other types of Decision Trees and Diagrams → Enlarging the area of application

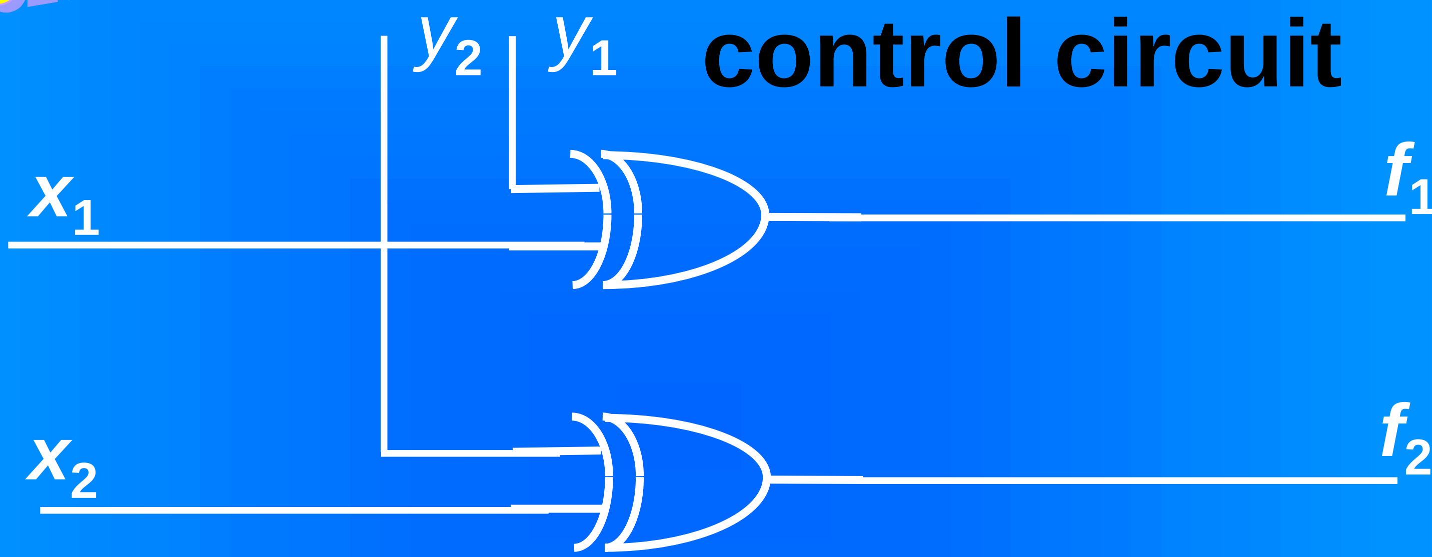
Future work (cont)

Focus of our todays research is the linear arithmetic representation of circuits:

- linear word-level DTs
- linear arithmetic expressions

Example

Linear arithmetic expression of parity control circuit



$$\left.\begin{array}{l} f_1 = x_1 + y_1 \\ f_2 = x_2 + y_2 \end{array}\right\} \quad f = 2f_2 + f_1 = 2x_2 + 2y_2 + x_1 + y_1$$

We use masking operator Ξ to extract the necessary bits from integer value of the function

Other Future Problems and Ideas

- Decision Trees are the most popular method in industrial learning system
- Robust and easy to program.
- Nice user interfaces with graphical trees and mouse manipulation.
- Limited type of rules and expressions
- AB @ CD is easy, tree would be complicated.
- Trees should be combined with functional decomposition - this is our research
- *A Problem for ambitious* - how to do this combination??
- More tests on real-life robotics data, not only medical databases

Questions and Problems

- 1. Write a Lisp program to create decision diagrams based on entropy principles
- 2. Modify this program using Davio Expansions rather than Shannon Expansions
- 3. Modify this program by using Galois Field Davio expansions for radix of Galois Field specified by the user.
- 4. Explain on example of a function how to create pseudo Binary Moment Tree (BMT), and write program for it.
- 5. As you remember the Free pseudo Kronecker Binary Moment Tree (KBMT) uses the following expansions $\{S_A, pD_A, nD_A\}$:
    - 1) Write Lisp program for creating such tree
    - 2) How you can generalize the concept of such tree?

Questions and Problems (cont)

- 6. Use the concepts of arithmetic diagrams for analog circuits and for multi-output digital circuits. Illustrate with circuits build from such diagrams.
- 7. How to modify the method shown to the GF(3) logic?
- 8. Decomposition:
    - A) Create a function of 3 ternary variables, describe it by a Karnaugh-like map.
    - B) Using Ashenhurst/Curtis decomposition, decompose this function to blocks
    - C) Realize each of these blocks using the method based on decision diagrams.

Partially based on slides from

Information Theoretic Approach to Minimization of Arithmetic Expressions

D. Popel, S. Yanushkevich
M. Perkowski*, P. Dziurzanski,
V. Shmerko

Technical University of Szczecin, Poland

* *Portland State University*

Information Theoretic Approach to Minimization of Polynomial Expressions over GF(4)

D. Popel, S. Yanushkevich
P. Dziurzanski,
V. Shmerko