Part 3

More details on our approach

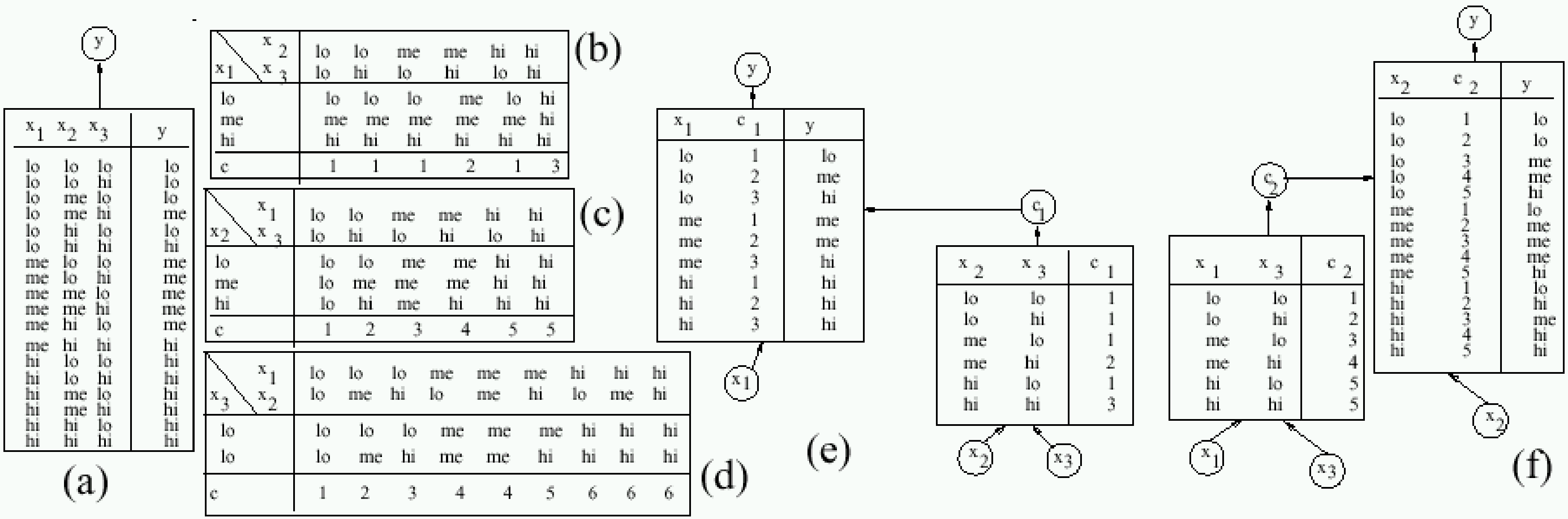
Generalization of the Ashenhurst-Curtis decomposition model

An example decision table $y = F(x_1, x_2, x_3)$

x_1	x_2	x_3	y
lo	lo	lo	lo
lo	lo	hi	lo
lo	me	lo	lo
lo	me	hi	me
lo	hi	lo	lo
lo	hi	hi	hi
lo	hi	hi	hi
me	lo	lo	me
me	lo	hi	me
me	me	lo	me
me	me	hi	me
me	hi	lo	me
me	hi	hi	hi
hi	lo	lo	hi
hi	lo	hi	hi
hi	me	lo	hi
hi	me	hi	hi
hi	hi	lo	hi
hi	hi	hi	hi

This kind of tables known from Rough Sets, Decision Trees, etc Data Mining

Two one-step decompositions



(a)

x_1	x_2	x_3	y
lo	lo	lo	lo
lo	lo	hi	lo
lo	me	lo	lo
lo	me	hi	me
lo	hi	lo	lo
lo	hi	hi	hi
me	lo	lo	me
me	lo	hi	me
me	me	lo	me
me	me	hi	me
me	hi	lo	me
me	hi	hi	hi
hi	lo	lo	hi
hi	lo	hi	hi
hi	me	lo	hi
hi	me	hi	hi
hi	hi	lo	hi
hi	hi	hi	hi

(b)

x_1 \ x_2 / x_3	lo / lo	lo / hi	me / lo	me / hi	hi / lo	hi / hi
lo	lo	lo	lo	me	lo	hi
me	me	me	me	me	me	hi
hi	hi	hi	hi	hi	hi	hi
c	1	1	1	2	1	3

(c)

x_2 \ x_1 / x_3	lo / lo	lo / hi	me / lo	me / hi	hi / lo	hi / hi
lo	lo	lo	me	me	hi	hi
me	lo	me	me	me	hi	hi
hi	lo	hi	me	hi	hi	hi
c	1	2	3	4	5	5

(d)

x_3 \ x_1 / x_2	lo / lo	lo / me	lo / hi	me / lo	me / me	me / hi	hi / lo	hi / me	hi / hi
lo	lo	lo	lo	me	me	me	hi	hi	hi
lo	lo	me	hi	me	me	hi	hi	hi	hi
c	1	2	3	4	4	5	6	6	6

(e)

x_1	c_1	y
lo	1	lo
lo	2	me
lo	3	hi
me	1	me
me	2	me
me	3	hi
hi	1	hi
hi	2	hi
hi	3	hi

x_2	x_3	c_1
lo	lo	1
lo	hi	1
me	lo	1
me	hi	2
hi	lo	1
hi	hi	3

(f)

x_1	x_3	c_2
lo	lo	1
lo	hi	2
me	lo	3
me	hi	4
hi	lo	5
hi	hi	5

x_2	c_2	y
lo	1	lo
lo	2	lo
lo	3	me
lo	4	me
lo	5	hi
me	1	lo
me	2	me
me	3	me
me	4	me
me	5	hi
hi	1	lo
hi	2	hi
hi	3	me
hi	4	hi
hi	5	hi

Decomposition is hierarchical

At every step many decompositions exist

A Standard Map of function ‘z’

Bound Set

Free Set

a b \ c	0	1	2
0 0	-	-	-
0 1	-	-	-
0 2	1	0 , 1	-
1 0	-	-	2
1 1	-	1	2
1 2	-	1	-
2 0	-	-	-
2 1	-	-	0
2 2	-	2 , 3	-

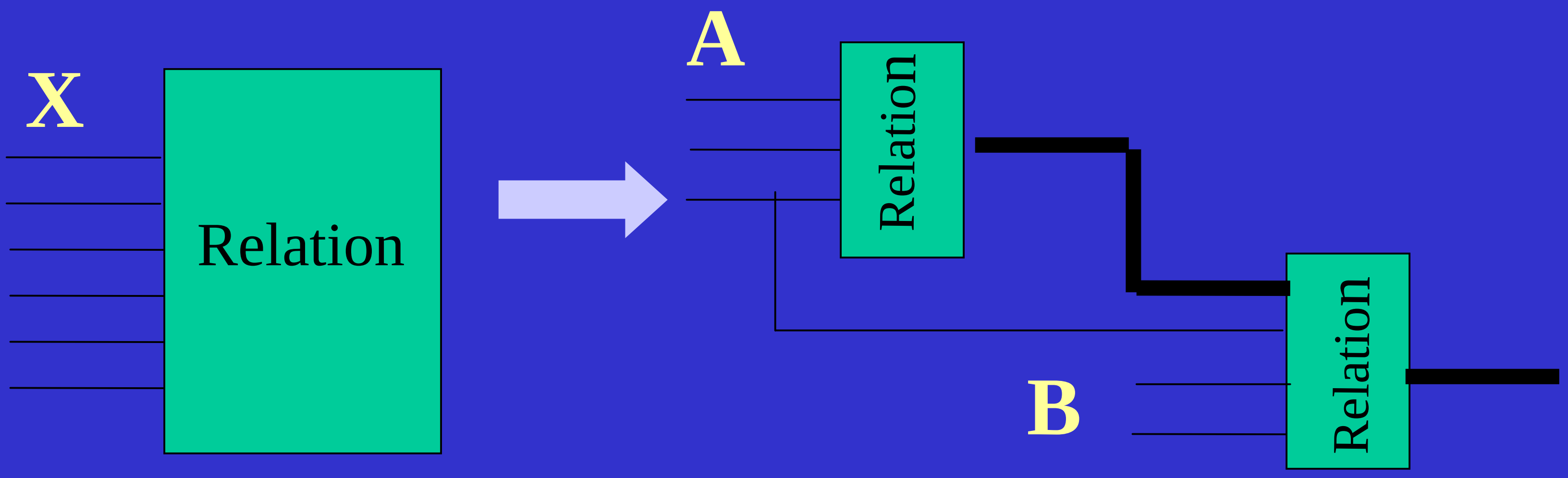
z

Columns 0 and 1
and
columns 0 and 2
are compatible

column
compatibility = 2

Decomposition of Multi-Valued Relations

F(X) = H(G(B), A), X = A $\cup$ B



if A $\cap$ B = $\emptyset$, it is ***disjoint decomposition***

if A $\cap$ B $\neq$ $\emptyset$, it is ***non-disjoint decomposition***

Forming a CCG from a K-Map

Bound Set

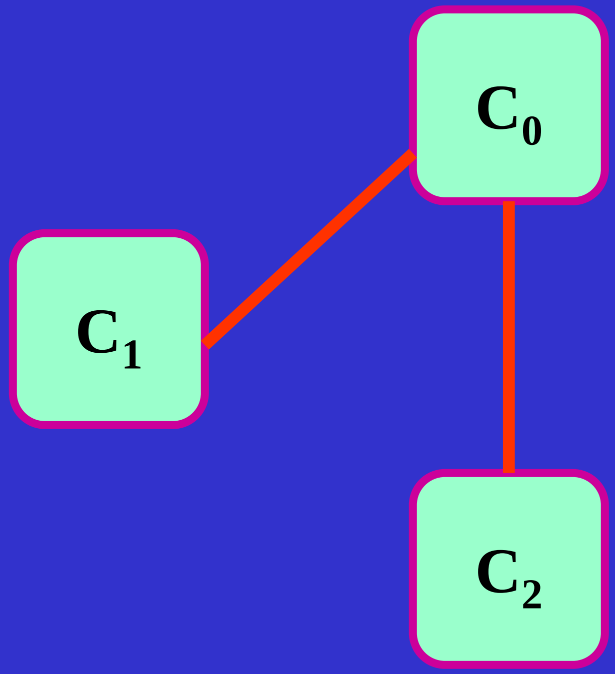
Free Set

a b \ c	0	1	2
0 0	-	-	-
0 1	-	-	-
0 2	1	0 , 1	-
1 0	-	-	2
1 1	-	1	2
1 2	-	1	-
2 0	-	-	-
2 1	-	-	0
2 2	-	2 , 3	-

z

Columns 0 and 1 and columns 0 and 2 are compatible

column compatibility index = 2



Column Compatibility Graph

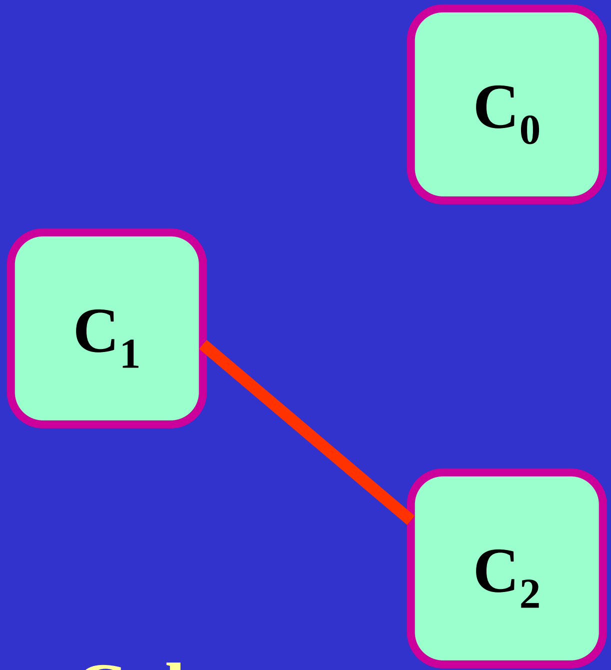
Forming a CIG from a K-Map

a b \ c	0	1	2
0 0	-	-	-
0 1	-	-	-
0 2	1	0 , 1	-
1 0	-	-	2
1 1	-	1	2
1 2	-	1	-
2 0	-	-	-
2 1	-	-	0
2 2	-	2 , 3	-

z

Columns 1 and 2 are incompatible

chromatic number = 2

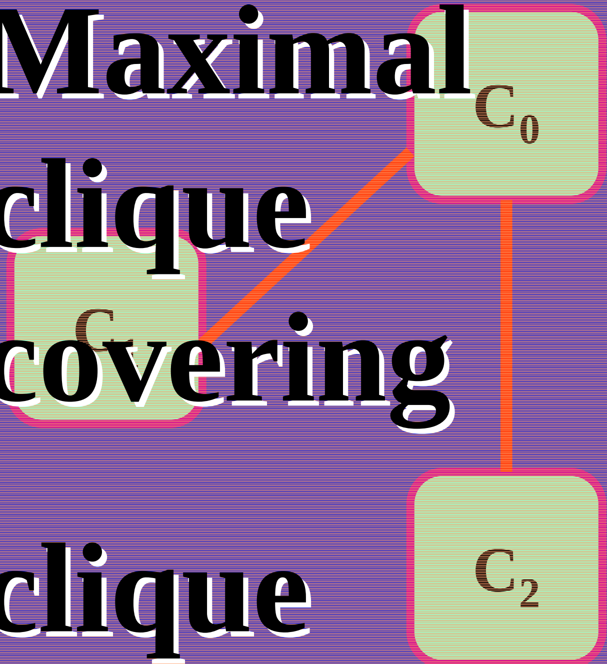


Column Incompatibility Graph

CCG and CIG are complementary

Maximal clique covering

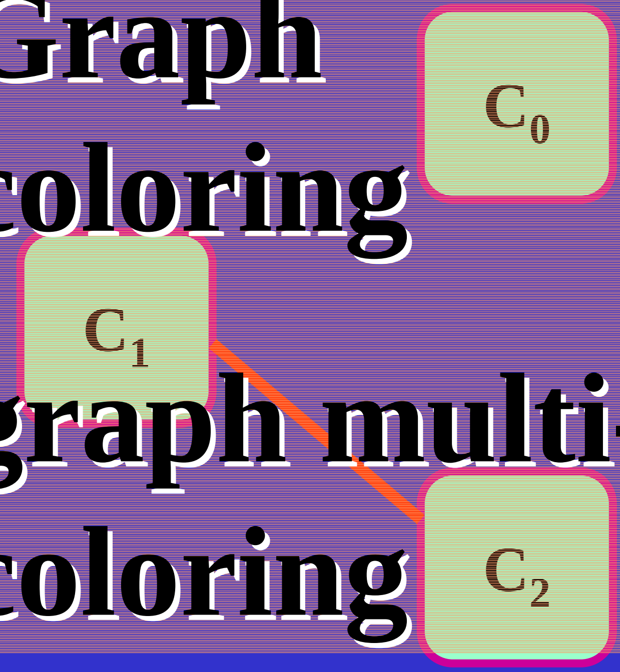
clique partitioning



Column Compatibility Graph

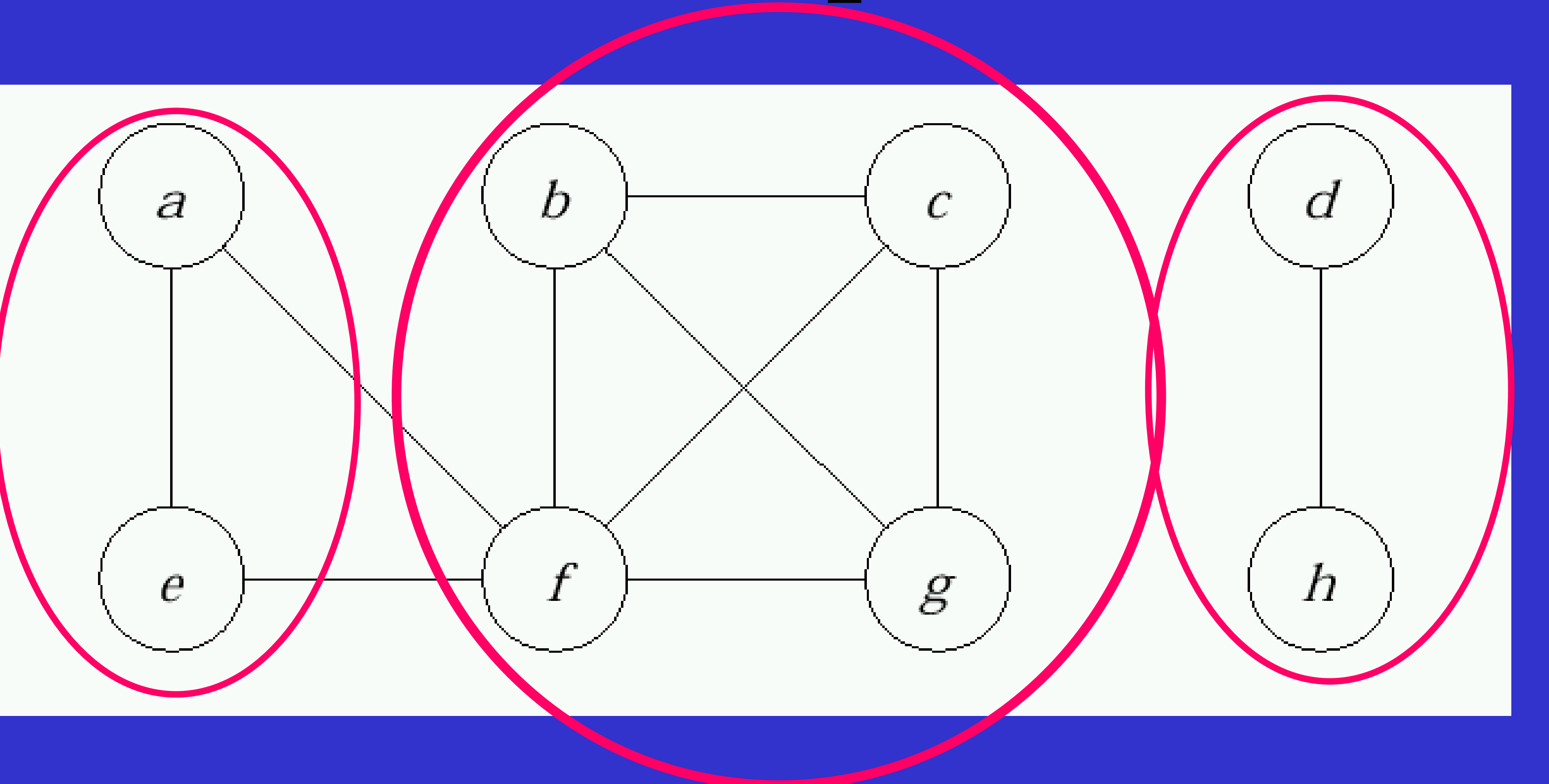
Graph coloring

graph multi-coloring

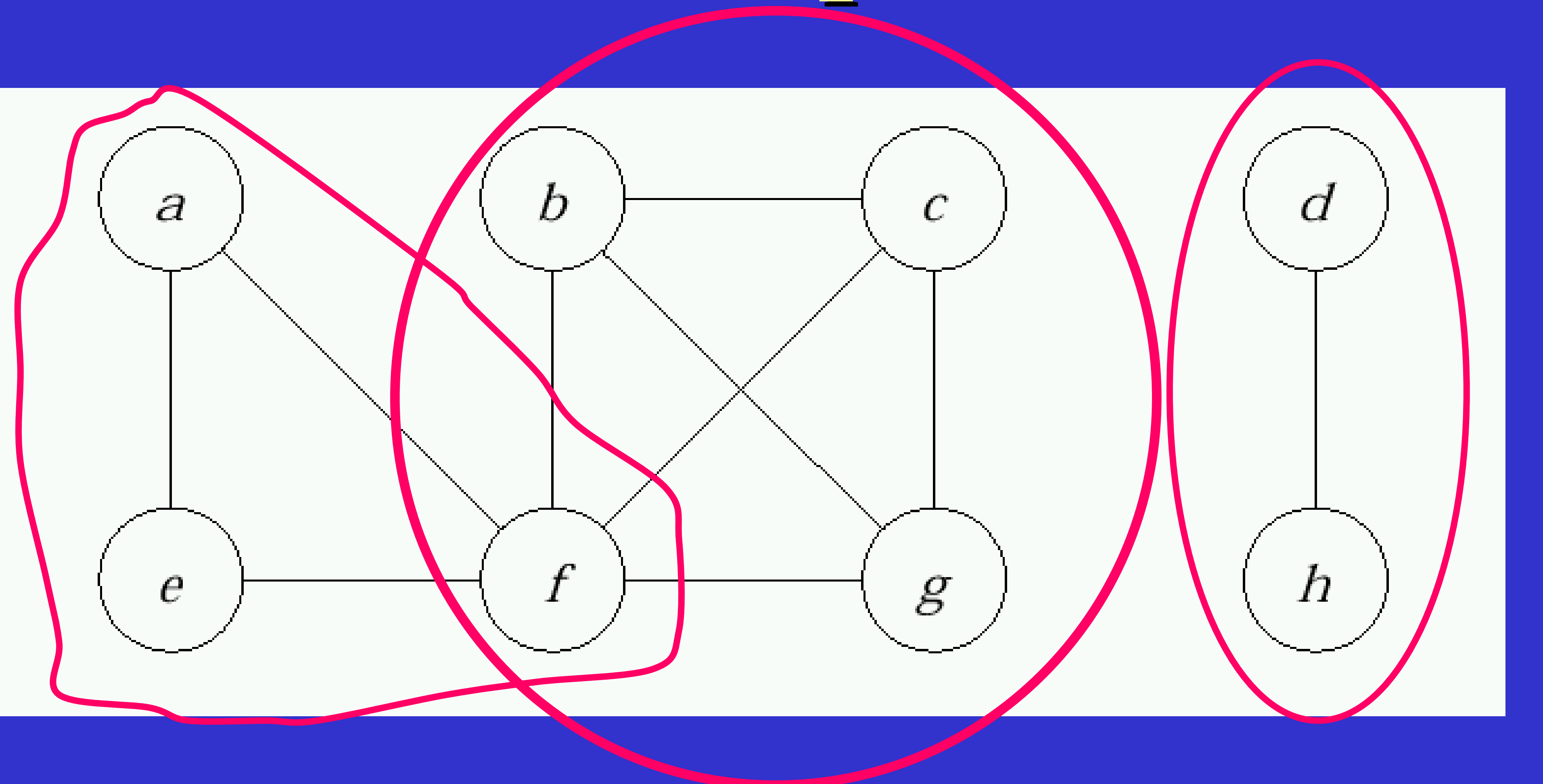


Column Incompatibility Graph

clique partitioning example.



Maximal clique covering example.



Map of relation G

\ c

0	1	2
0 , 1	0	1

G

From CIG

\ c

0	1	2
0	0	1

G

After induction

g = a high pass filter whose acceptance threshold begins at

$c > 1$

Cost Function

Decomposed Function Cardinality **is the total cost of all blocks.**

Cost is defined for a single block in terms of the block's n inputs and m outputs

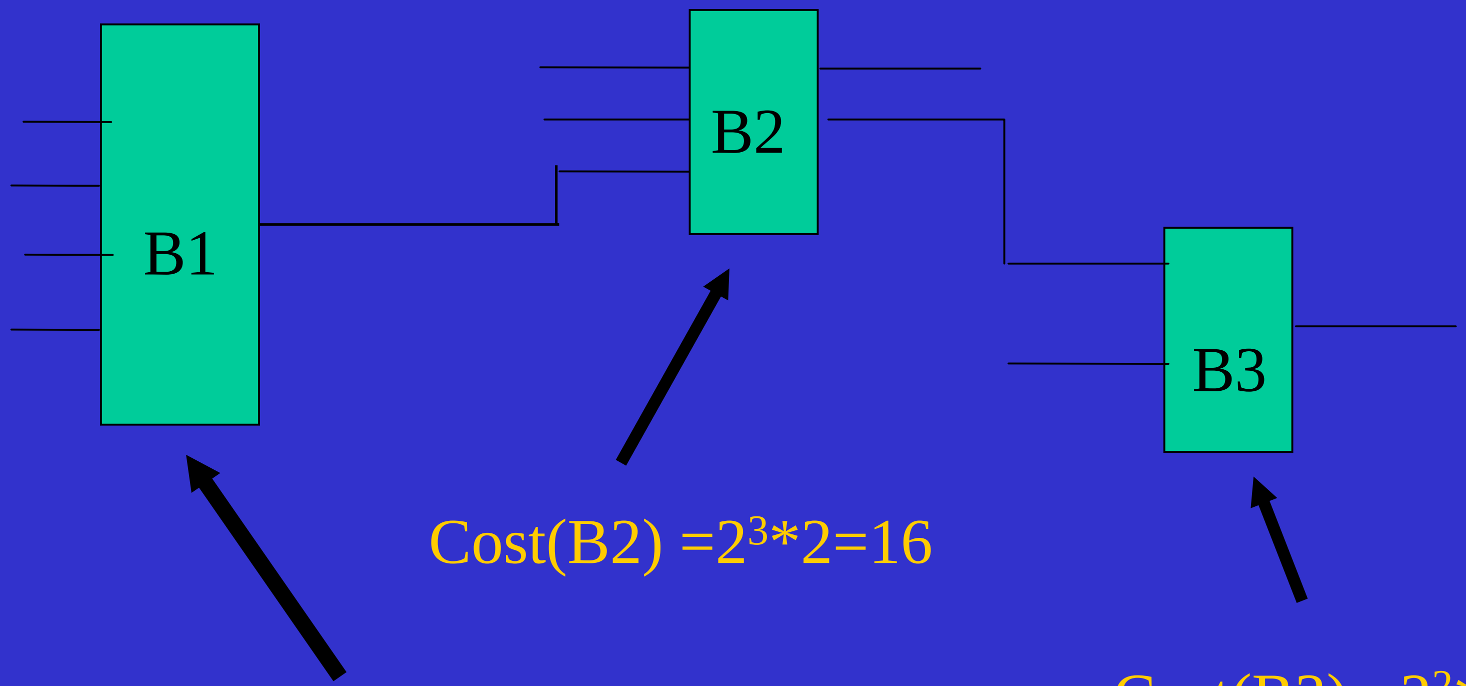
$$\textbf{Cost} := m * 2^n$$

DFC = Decomposed Function Cardinality

$$C_x(f) = \log_2 \min\{cost\ of\ \Gamma : \Gamma\ simulates\ f\}$$

$$cost(f) = 2^{|X|}|Y|$$

Example of DFC calculation



Cost(B2) $=2^3*2=16$

Cost(B1) $=2^4*1=16$

Cost(B3) $=2^2*1=4$

Total DFC = 16 + 16 + 4 = 36

Other cost functions

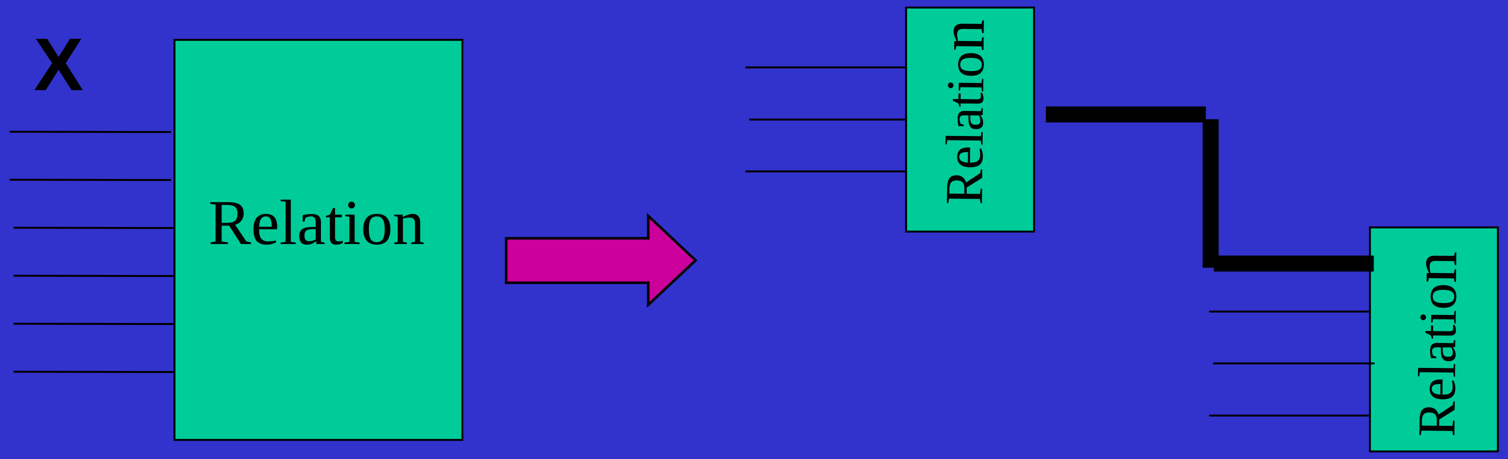
New Complexity Measures

$$C_x = \log_2 \left(\prod_{x_i \in X} |x_i| \, \log_2 \prod_{y_j \in Y} |y_j| \right)$$

where: $|x_i|$ is cardinality of variable $x_i \in X$,
$|y_j|$ is cardinality of variable $y_j \in Y$.

$$C_x = \log_2 \left(\prod_{y_j \in Y} |y_j| \right)^{\prod_{x_i \in X} |x_i|} = \prod_{x_i \in X} |x_i| \, \log_2 \prod_{y_j \in Y} |y_j|$$

Comparison of RC before and after decomposition



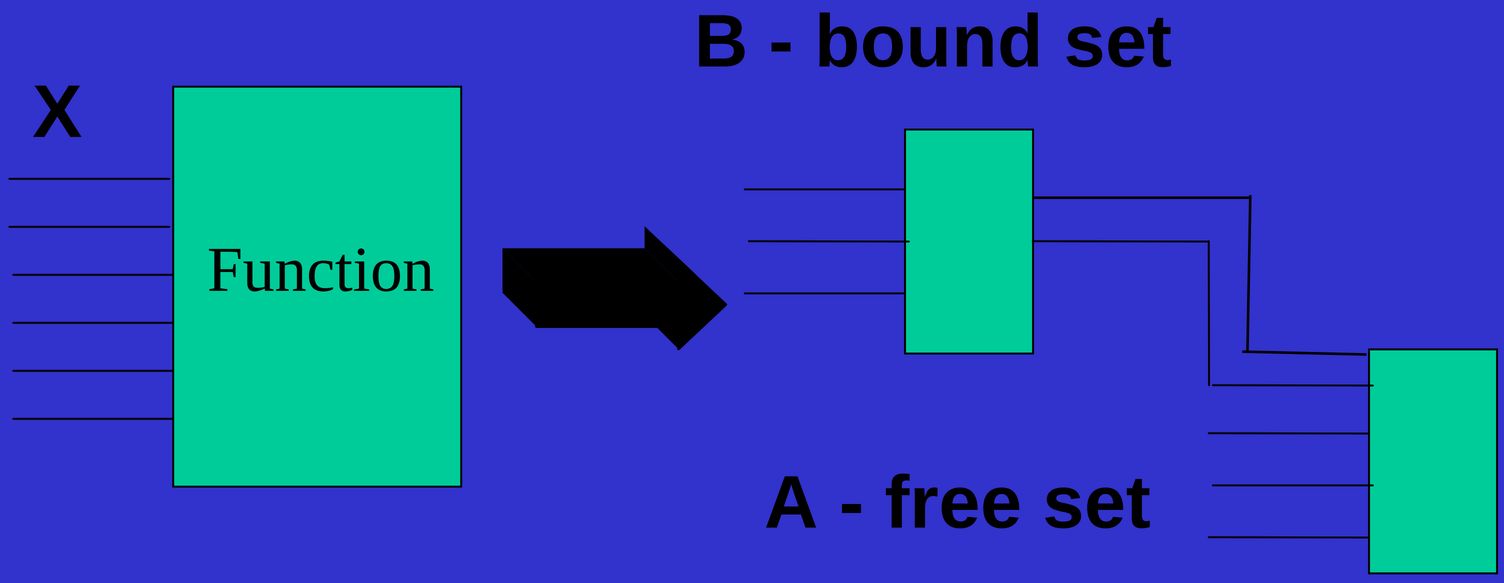
$RC_{before} = (3*3*3)*(\log_2 4) = 54$

$RC_{after} = [(3)*(\log_2 2)] +$

$[(2*3*3)*(\log_2 4)] = 3 + 36 = 39$

Two-Level Curtis Decomposition

F(X) = H(G(B), A), X = A ∪ B



if A ∩ B = ∅, it is *disjoint decomposition*

if A ∩ B ≠ ∅, it is *non-disjoint decomposition*

Decomposition Algorithm

- Find a set of partitions (A_i, B_i) of input variables (X) into free variables (A) and bound variables (B)
- For each partitioning, find decomposition $F(X) = H_i(G_i(B_i), A_i)$ such that column multiplicity is minimal, and calculate DFC
- Repeat the process for all partitioning until the decomposition with minimum DFC is found.

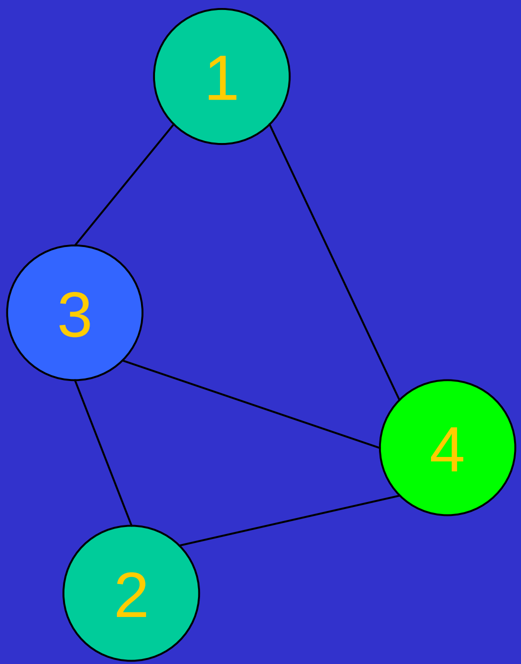
Algorithm Requirements

- Since the process is iterative, it is of high importance that minimization of the column multiplicity index is done as fast as possible.
- At the same time, for a given partitioning, it is important that the value of the column multiplicity is as close to the **absolute minimum** value

Column Multiplicity

Bound Set

Free Set	00	01	11	10
00	0	0	–	1
01	–	1	0	0
11	1	–	1	0
10	1	1	0	0



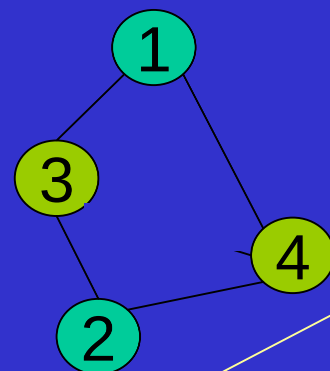
1 2 3 4

Column Multiplicity-other example

Bound Set

Free Set

AB \ CD	00	01	11	10
00	0	0	–	1
01	–	1	0	0
11	1	–	1	-
10	1	1	0	0
	1	2	3	4



C \ D	0	1
0	0	0
1	1	1

X

Color green

X=G(C,D)

X=C in this case

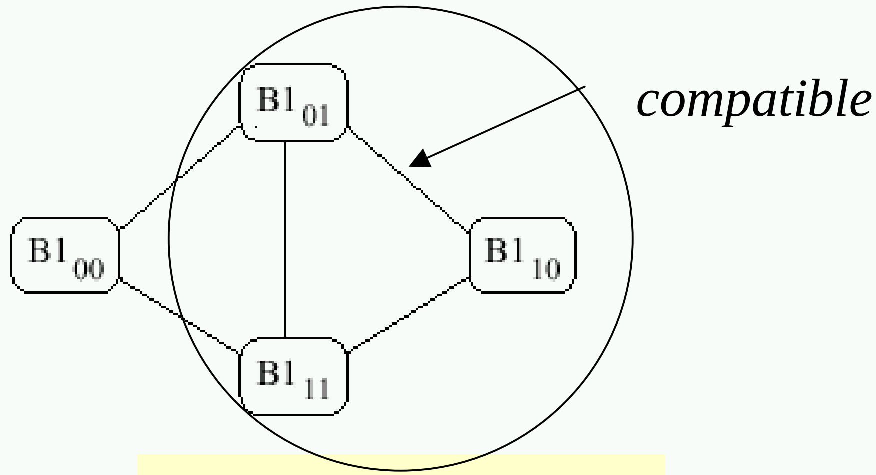
But how to calculate function H?

Decomposition of multiple-valued relation

ab \ cd	$B1_{00}$ 00	$B1_{01}$ 01	$B1_{11}$ 11	$B1_{10}$ 10
00	0,3 [0]	0,3 [4]	1,3 [7]	2,3 [9]
01	1,2 [1]	-	0,1 [8]	1,3 [10]
11	0 [2]	0,3 [5]	-	-
10	0,3 [3]	0,4 [6]	-	1,4 [11]

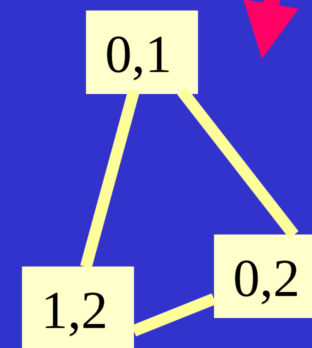
f

Karnaugh Map



Compatibility Graph for columns

But in general compatibility is not transitive and the whole group must be checked



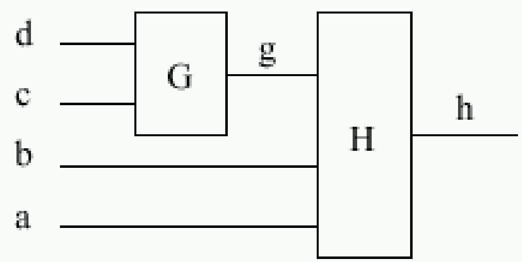
c \ d	0	1
0	0	0,1
1	1	0,1

Kmap of block G

ab \ g	0	1
00	3	3
01	1	1
11	0	0,3
10	0	4

h

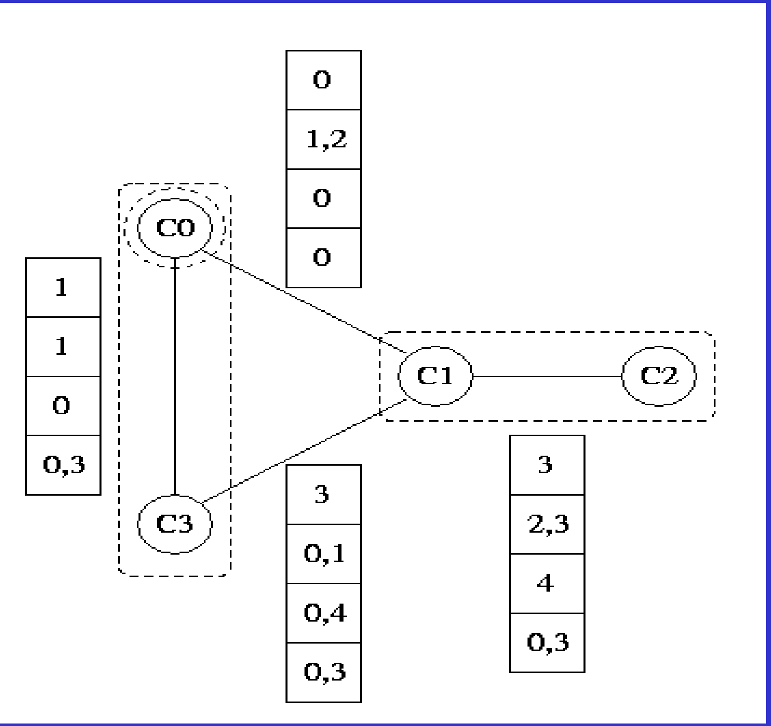
Kmap of block H



One level of decomposition

Compatibility of columns for Relations is not transitive !

ab \ cd	C0 00	C1 01	C2 10	C3 11
00	0,1	0,3	2,3	1,3
01	1,2	—	2,3	0,1
10	0,3	0,4	1,4	—
11	0	0,3	—	—

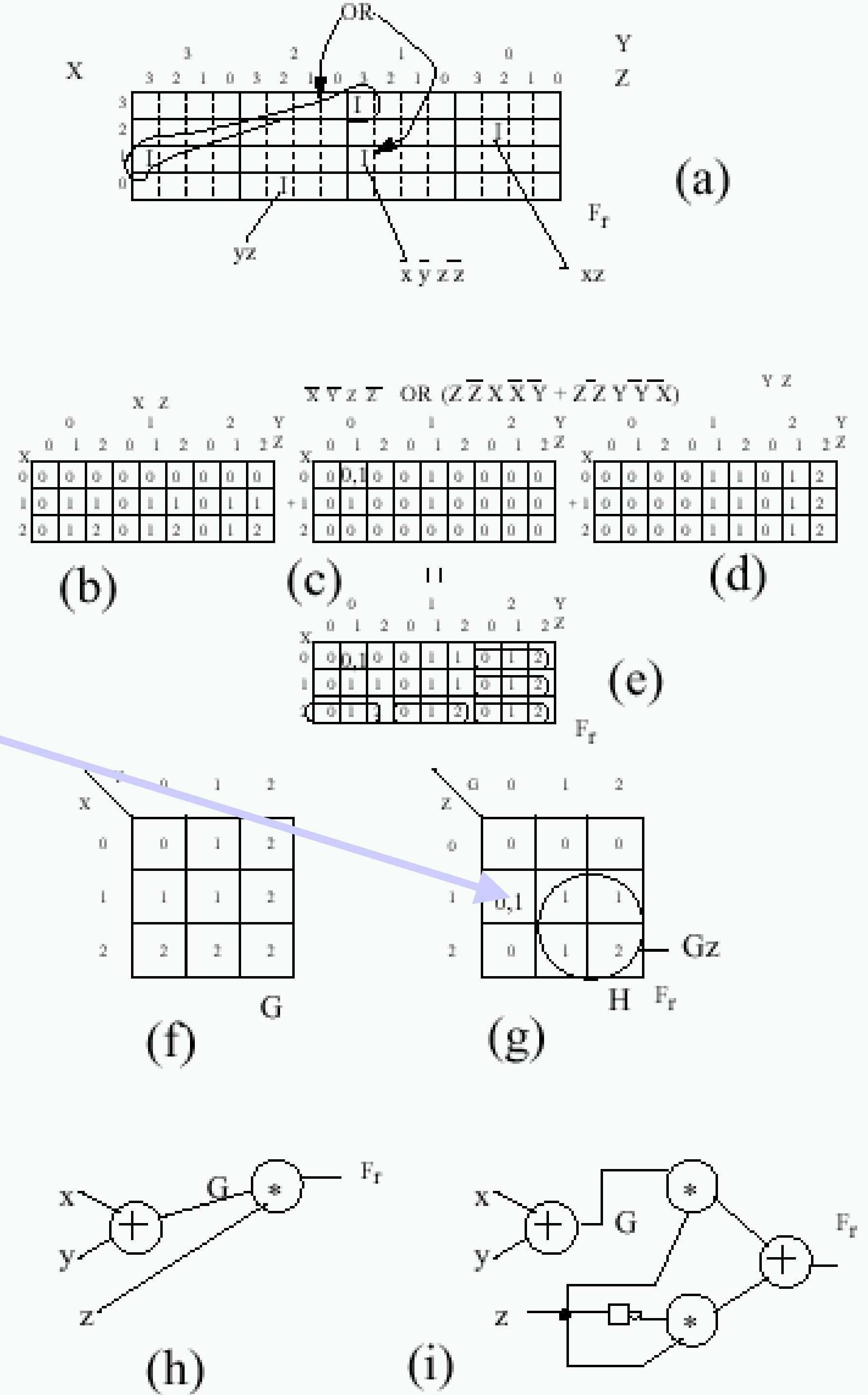


This is an important difference between decomposing functions and relations

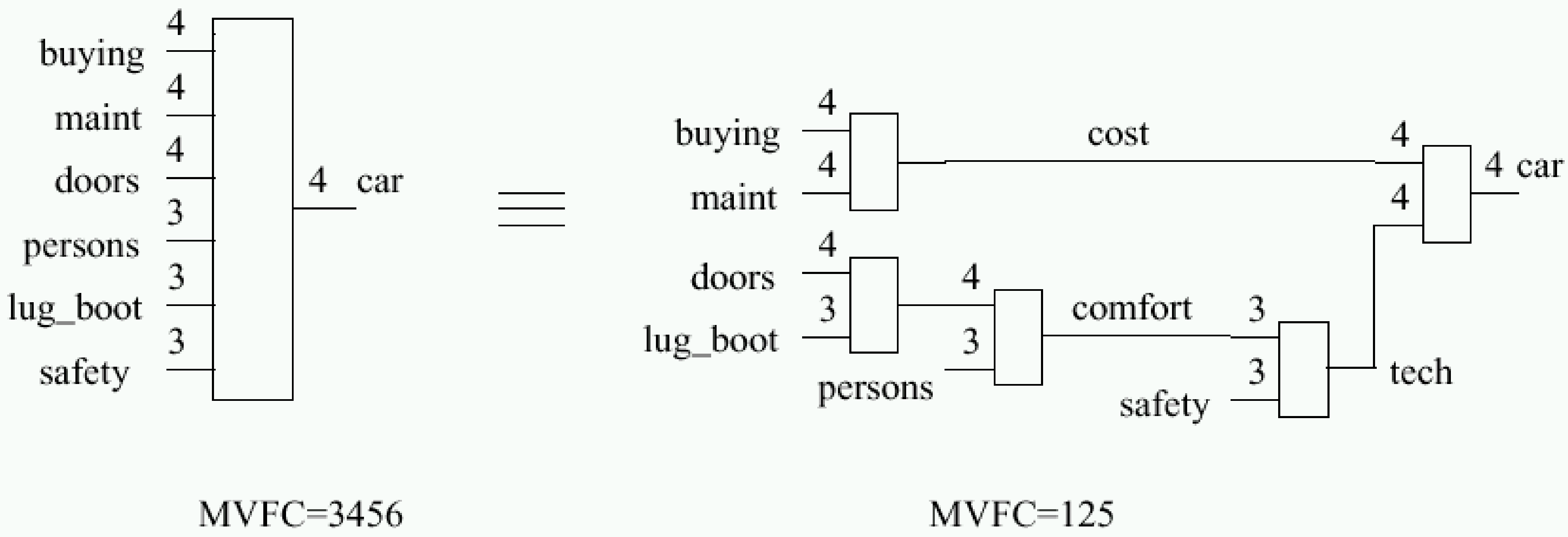
Decomposition of Relations

Now H is a relation

which can be either decomposed or minimized directly in a sum-of-products fashion



Discovering new concepts



- Discovering concepts useful for **purchasing a car**

Variable ordering

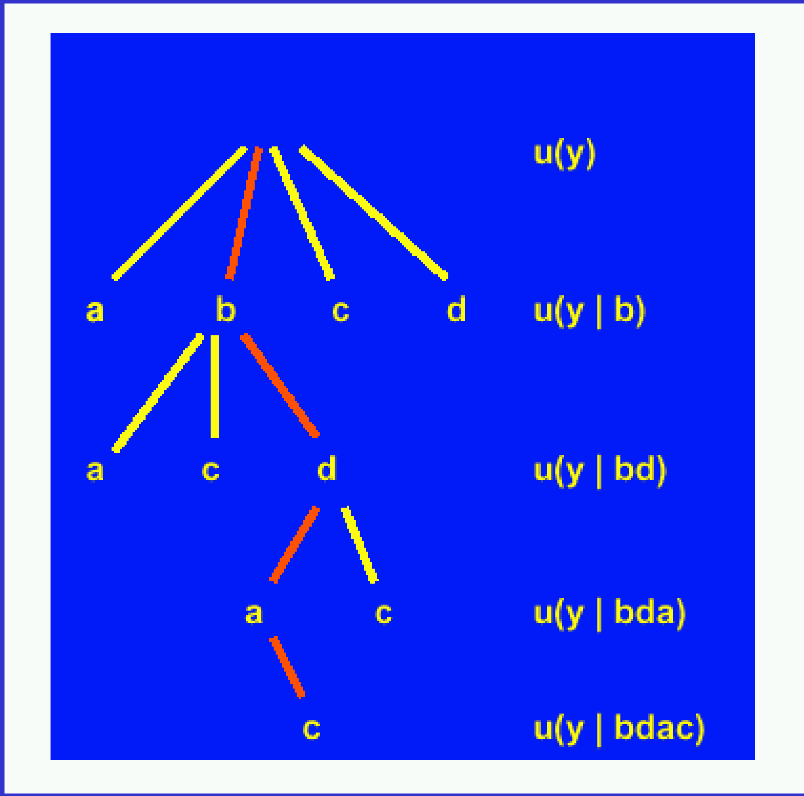
- **Uncertainty (Shannon):**

$$u(a) = -\sum_i p(a = a_i) \log_2 p(a = a_i)$$

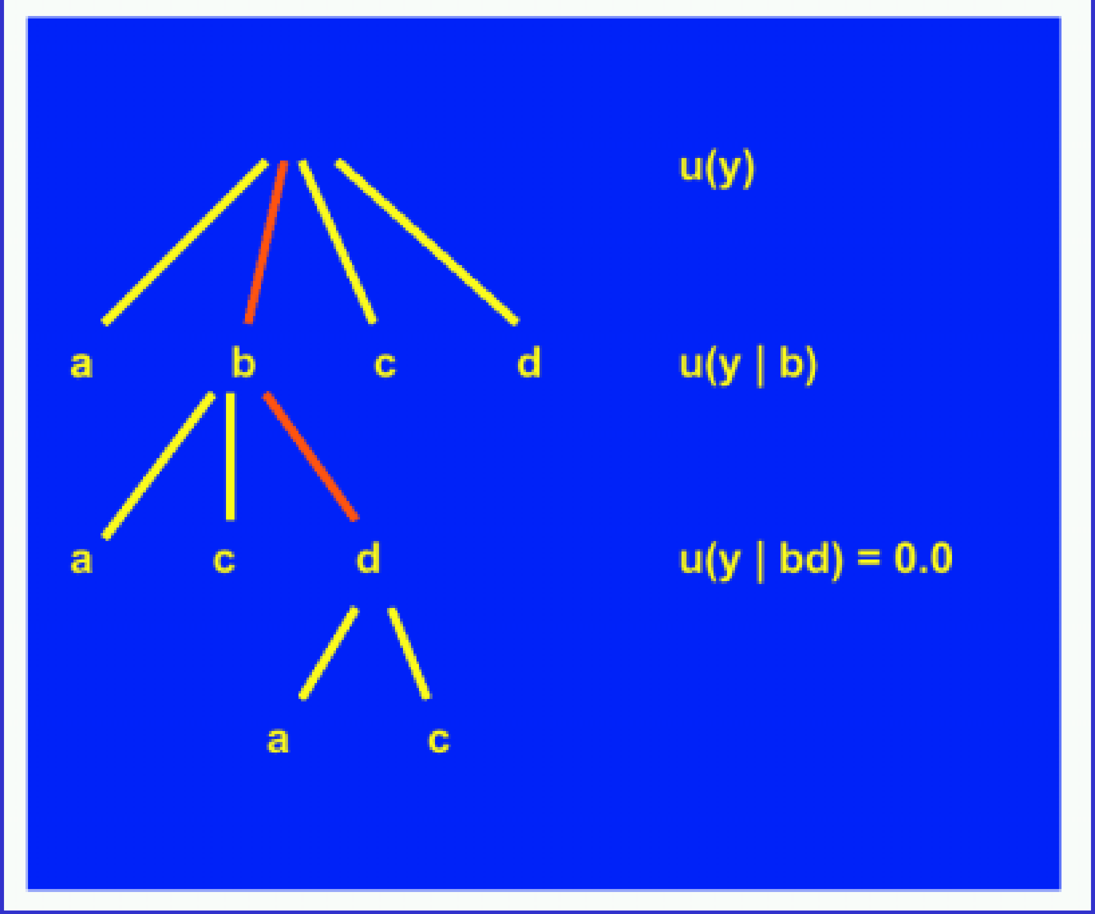
- **Conditional Uncertainty (Shannon):**

$$u(a|b) = u(ab) - u(b)$$

Select variables that reduce the uncertainty the most - the best way to separate zeros from ones

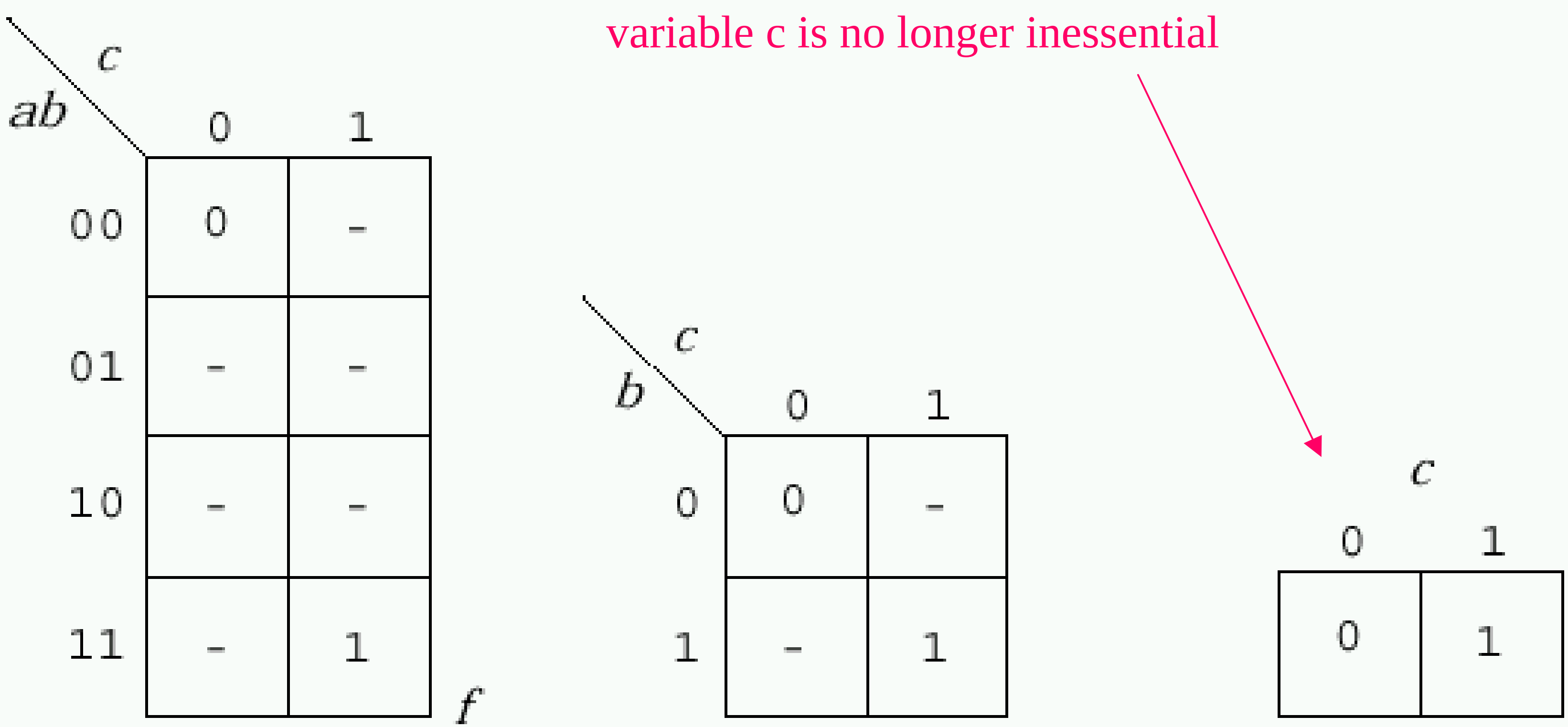


Vacuous variables removing



- Variables b and d reduce uncertainty of y to 0 which means they provide all the information necessary for determination of the output **y**
- Variables a and c are vacuous

Example of removing inessential variables



(a) original function

(b) variable a removed

(c) variable b removed,

Part 4

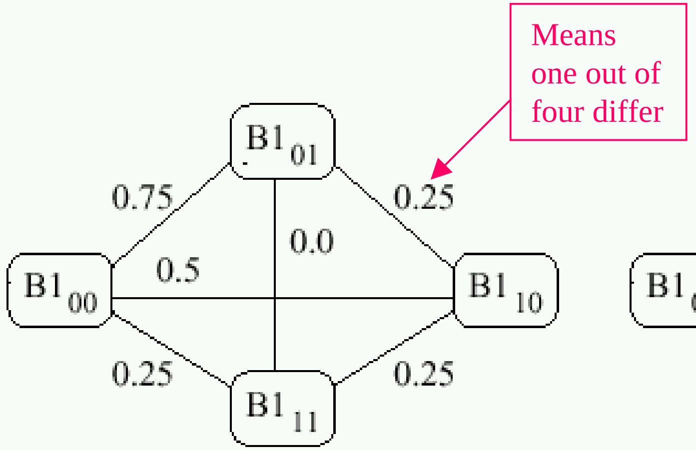
Generalization of the Ashenhurst-Curtis decomposition model

Compatibility graph construction for data with noise

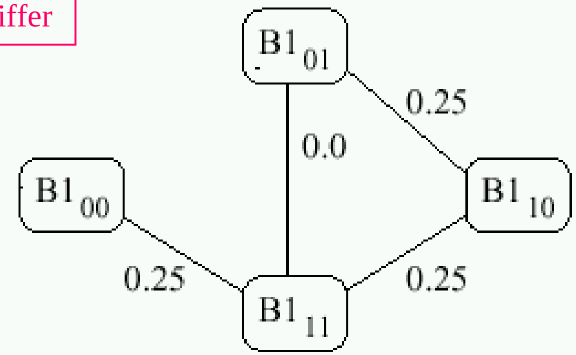
ab \ cd	00 ($B1_{00}$)	01 ($B1_{01}$)	11 ($B1_{11}$)	10 ($B1_{10}$)
00	0 ^{0}	3 ^{4}	1,3 ^{7}	2 ^{9}
01	1 ^{1}	-	0,1 ^{8}	1 ^{10}
11	0 ^{2}	3 ^{5}	-	-
10	0 ^{3}	4 ^{6}	-	4 ^{11}

f

Kmap



Compatibility Graph for Threshold 0.75



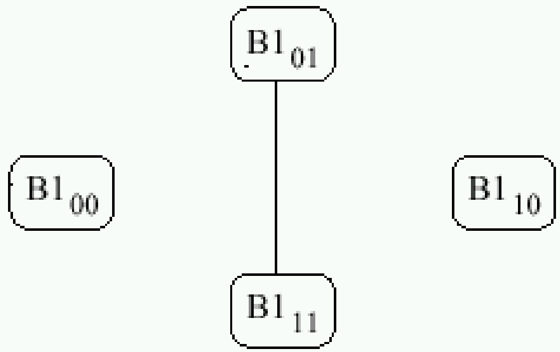
Compatibility Graph for Threshold 0.25

Compatibility graph for metric data

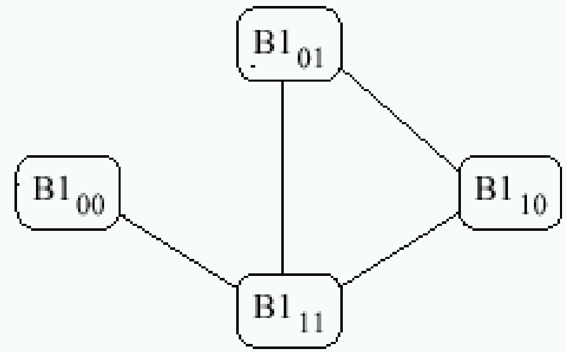
ab \ cd	00 ($B1_{00}$)	01 ($B1_{01}$)	11 ($B1_{11}$)	10 ($B1_{10}$)
00	0^{0}	3^{4}	$1,3^{7}$	2^{9}
01	1^{1}	-	$0,1^{8}$	1^{10}
11	0^{2}	3^{5}	-	-
10	0^{3}	4^{6}	-	4^{11}

f

Kmap



Compatibility Graph for nominal data



Compatibility Graph for metric data

Compatible when Difference of 1 or less

MV relations can be created from contingency tables

a)

ab \ cd	00	01	11	10
00	77	57	3	2
01	1	110	12	1
11	12	28	200	1
10	0	423	21	52

THRESHOLD 70

ab \ cd	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	1	0	0

THRESHOLD 50

ab \ cd	00	01	11	10
00	1	1	0	0
01	0	1	0	0
11	0	0	1	0
10	0	1	0	1

d) (cells: cd)

a \ b	0	1
0	00	01
1	01	11

e) (cells: cd)

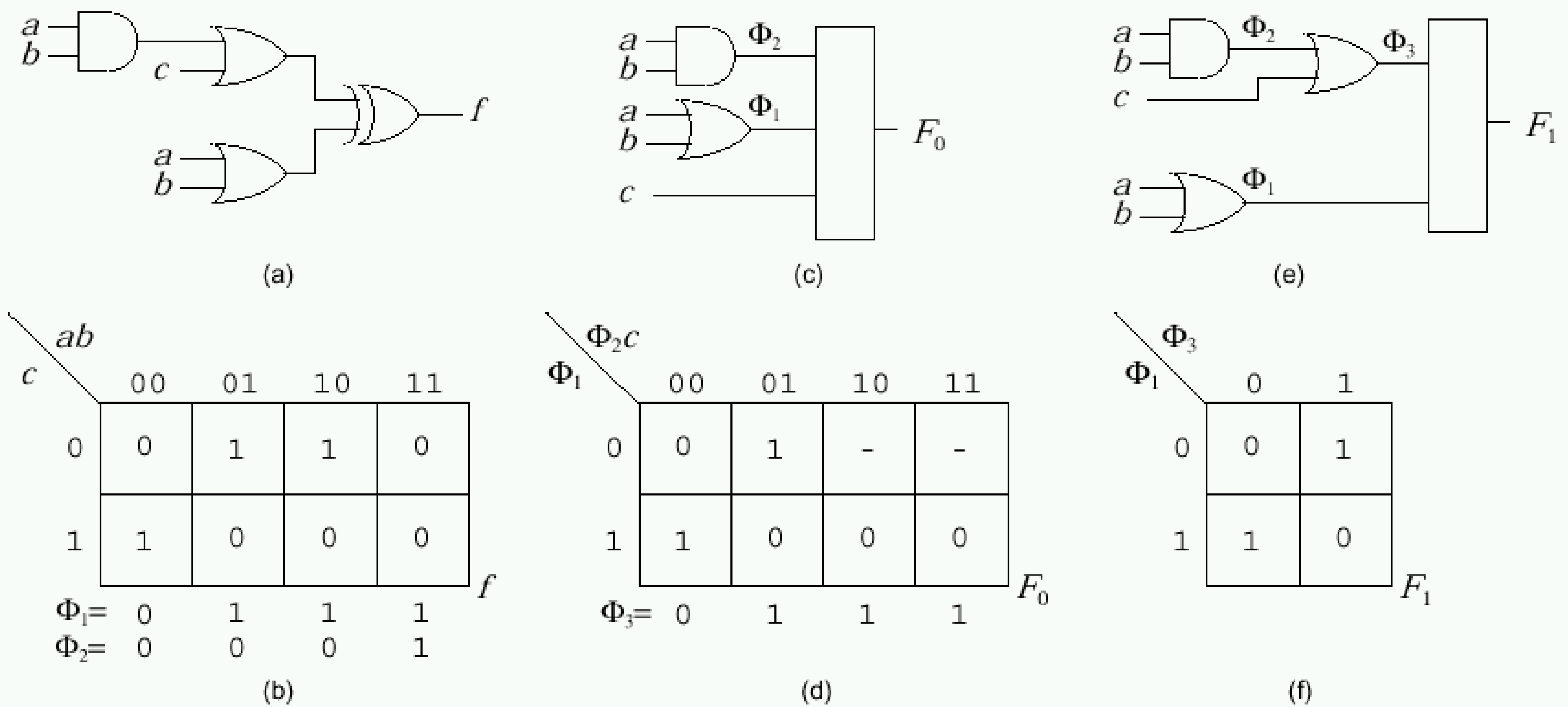
a \ b	0	1
0	00,01	01
1	01,10	11

Thus our method can handle also probabilistic and continuous data

Figure 1: Contingency tables

Our method generalizes the concept of decomposition even in standard binary case

Example of decomposing a Curtis non-decomposable function.



Concluding on decomposition principles

- The most general decomposition ever.
- Binary, multi-valued, fuzzy, continuous, probabilistic, nominal, metric, reversible, quantum…..
- Synthesis can be exact or for noisy data.
- Many applications: Field Programmable Gate Arrays (Xilinx), VLSI design (Intel), robot control, epidemiology, layout……..

Evaluation of numerical results

Decomposition of binary (MCNC) benchmarks

In underlined cases in this column we are the best

misII wins only once

		cost				
File	i/o	TRADE	MISII	DSGN174	mvgud	[time]
5xp1	7/10	496	384	292	**236**	[11.0]
9sym	9/1	640	984	400	**104**	[26.4]
con1	7/2	80	68	**60**	70	[2.3]
duke2	22/29	6516	2428	**2200**	2896	[11289.0]
ex5p	8/63	-	3720	**1560**	2104	[208.0]
f51m	8/8	372	392	240	**177**	[10.1]
misex1	8/7	472	**208**	224	229	[8.6]
misex2	25/18	548	464	436	**392**	[1086.0]
misex3	14/14	9816	4204	3028	**1744**	[1316.0]
rd53	5/3	120	96	84	**60**	[1.8]
rd73	7/3	320	352	256	**113**	[13.1]
rd84	8/4	508	672	320	**171**	[32.6]
sao2	10/4	1848	516	468	**441**	[47.2]

Our program

Benchmark		Cost for Various Decomposers *								
Name	i(o)	TR	MI	St	SC	LU	Js	Jh	MV	Time, s
5xpl	7/10	496	384	292	288 (9)	288 (9)	320 (20)	336 (21)	<u>236</u>	11.0
9sym	9/1	640	984	400	224 (7)	160 (5)			<u>104</u>	26.4
con1	7/2	80	68	60					<u>70</u>	2.3
duke2	22/29	6516	2428	<u>2200</u>	3456 (108)				2896	11289.0
ex5p	8/63		3720	<u>1560</u>					2104	208.0
f5lm	8/8	372	392	240	256 (8)				<u>177</u>	10.1
misex1	8/7	472	208	224	256 (8)	354 (11)	304 (19)	288 (18)	<u>229</u>	8.6
misex2	25/18	548	464	436	768 (24)				<u>392</u>	1086.0
misex3	14/14	9816	4204	3028					<u>1744</u>	1316.0

Function	in	MBDD in	MBDD nodes	MVDD nodes	MBDD size	MVDD size	Size %
audiology	69	80	7039	6668	28156	34021	82%
breastc	9	36	4093	1119	16372	14547	112%
bridges1	9	16	359	195	1140	1137	100%
bridges2	10	18	503	262	1576	1537	102%
chess1	6	16	7820	3091	31280	33981	92%
chess2	36	37	8802	8446	34900	42538	82%
connect-4	42	84	82639	40724	273252	244344	111%
flag	28	57	6651	3557	26284	25854	101%
house-votes	16	16	407	407	1628	2035	80%
letter	16	64	318883	77004	1275532	1463076	87%
lung-cancer	56	112	2953	1472	11812	10304	114%
programm	12	24	33317	16419	115496	104737	110%
sensory	11	19	1853	1074	6992	6541	106%
sleep	9	31	933	238	3328	3143	105%
sponge	44	86	3472	1745	13888	11987	115%
tic-tac-toe	9	18	779	338	2400	2028	118%
trains	32	51	314	193	1256	1247	100%
alet	18	72	21967	5316	79500	69108	115%
d4	14	29	486	219	1872	1543	121%
d7	24	61	1123	416	4284	3647	117%
d8	32	80	1527	588	5800	4869	119%
d9	34	84	1616	629	6156	5162	119%
d10	37	89	1720	688	6572	5554	118%
geo	11	32	3163	831	11556	8879	130%
let	18	72	21910	5304	79296	68952	115%
u1	60	153	22552	9839	90208	73631	122%
u1_4	60	91	329	237	1316	1319	99%
u1_5	60	98	437	295	1748	1701	102%
u1_10	60	129	1106	571	4424	3773	117%
u2	60	144	21344	10085	85376	71369	119%
u3	60	151	22363	9831	89452	71898	124%
u4	60	144	21492	9989	85968	70693	121%
u5	60	143	21779	10064	87116	71157	122%
total			645,731	227,854	2,485,936	2,536,120	98%
w/o letter			326,848	150,850	1,210,404	1,073,044	113%

Table 3.2: MVDD and MBDD size comparisons.

Here we compare various functional representations for data

			Top Down			Jozwiak			
filename	in	sol.	random	fifo	CI	random	fifo	CI	pure CI
audiology	69	157	>2000	>2000	>2000	>2000	>2000	>2000	>2000
balance	4	4	0.1	0.1	0.1	0.8	0.8	0.8	0.6
breast	9	5	92.5	58.9	94.6	92.6	89.7	106.0	234.9
bridges1	9	9	0.4	0.1	0.1	0.7	0.7	0.7	67.1
bridges2	10	10	0.7	0.1	0.1	1.0	1.0	1.0	193.1
car	6	6	0.1	0.1	0.1	0.2	0.2	0.2	3.2
chess1	6	6	0.1	0.1	0.1	9.0	9.0	9.1	156.6
chess2	36	29	56.4	49.4	41.7	76.3	76.0	75.5	>2000
cloud	6	6	0.1	0.1	0.1	0.4	0.4	0.4	9.8
connect-4	42	347	>2000	>2000	>2000	>2000	>2000	>2000	>2000
employ1	9	9	0.2	0.1	0.1	0.2	0.2	0.2	29.6
employ2	7	7	0.1	0.1	0.1	0.1	0.1	0.1	7.7
flag	28	77	>2000	>2000	>2000	>2000	>2000	>2000	>2000
flare1	10	10	0.7	0.1	0.1	1.4	1.4	1.4	271.4
flare2	10	9	0.1	0.9	0.9	2.4	2.9	2.6	324.2
house-votes	16	16	0.2	0.1	0.1	1.8	1.9	1.8	>2000
letter	16	157	>2000	>2000	>2000	>2000	>2000	>2000	>2000
lung-cancer	56	47	>2000	>2000	>2000	>2000	>2000	>2000	>2000
monks1tr	6	3	0.1	0.8	0.9	0.1	0.1	0.1	2.7
monks2tr	6	6	0.2	0.1	0.1	0.3	0.3	0.3	5.7
monks3tr	6	4	0.1	0.4	0.5	0.2	0.2	0.2	3.4
mushroom	22	4	1277.7	544.0	698.0	>2000	>2000	>2000	>2000
post-op	8	8	0.4	0.1	0.1	0.4	0.4	0.4	23.4
programm	12	12	4.3	0.7	0.7	100.3	100.5	97.4	>2000
sensory	11	5	30.9	25.0	35.2	391.0	375.6	600.4	1321.5
shuttlem	6	6	0.1	0.1	0.1	0.5	0.5	0.5	1.4
sleep	9	5	17.2	9.3	7.6	9.6	6.1	11.4	168.1
sponge	44	3	>2000	>2000	>2000	1736.2	>2000	>2000	>2000
tic-tac-toe	9	8	1.3	0.4	0.4	212.1	213.3	250.1	250.2
trains	32	1	14.4	15.5	15.3	23.4	6.2	0.3	0.3
zoo	16	5	10.5	10.2	14.9	36.6	44.4	26.5	1001.7
alut	18	17	24.0	13.7	9.6	8.9	8.9	9.3	>2000
c2a	11	2	1.1	1.4	1.8	3.6	9.2	3.9	3.7
c2b	11	3	4.2	2.9	6.2	6.8	6.5	7.1	7.9
c3a	14	2	5.2	4.3	4.9	7.8	14.9	9.9	9.8
c3b	14	3	12.1	8.5	19.7	13.5	14.4	15.7	15.8
c4a	14	2	5.0	4.2	7.7	12.4	12.7	11.3	11.2
c4b	14	3	12.9	9.6	19.6	13.8	14.0	16.1	16.2
c5a	13	2	4.7	3.8	5.5	4.9	4.7	2.5	2.4
c5b	13	2	4.4	2.0	5.8	1.9	3.6	2.5	2.4
c6a	13	2	6.3	4.5	7.5	3.8	2.1	2.6	2.7
c6b	13	2	5.4	4.9	7.2	2.4	3.3	2.5	2.4
d2	11	4	1.3	1.2	1.4	1.3	1.4	1.5	34.7
d3	14	4	17.8	15.7	24.7	95.1	125.9	97.9	98.4
d4	14	3	19.9	13.2	22.4	29.0	28.3	36.5	36.3
d5	13	2	4.4	4.3	9.1	9.6	18.2	3.9	3.7
d6	13	2	12.2	9.3	14.9	10.7	24.0	5.2	4.6
d7	24	2	184.8	89.4	119.4	129.9	82.2	17.3	17.5
d8	32	2	1276.7	271.4	352.8	125.2	130.1	31.4	31.2
d9	34	2	372.9	343.8	436.1	283.6	199.6	35.2	35.9
d10	37	2	617.1	477.1	616.4	329.6	310.6	41.1	41.2
gcu	11	6	57.5	35.3	75.3	156.2	174.4	157.2	1505.2

Top Down algorithm comparison with Jozwiak's algorithm.

In all these cases Jozwiak cannot complete

Function	in	FLASH	SBSD
add0	8	28	28
add2	6	20	20
and_or_chain8	8	28	28
ch22f0	6	20	20
ch30f0	6	**32**	40
ch47f0	6	60	**56**
ch52f4	8	180	**156**
ch70f3	8	**40**	44
ch74f1	8	**72**	84
ch83f2	8	**116**	120
ch8f0	6	**32**	40
4_ones	8	76	76
greater_than	8	28	28
interval1	8	128	**88**
interval2	8	92	**76**
kdd2	5	16	16
kdd3	5	12	12
kdd5	8	**32**	48
kdd6	8	12	12
kdd7	8	28	28
kdd9	8	20	20
kdd10	6	20	20
majority_gate	8	**64**	76
monkish1	4	12	12
monkish2	8	60	60
monkish3	5	20	20
mux8	6	**24**	32
or_and_chain8	8	28	28
pal	8	28	28
parity	8	28	28
rnd_m1	8	28	28
rnd_m10	8	**80**	108
rnd_m25	8	**172**	180
rnd_m5	8	**64**	72
rnd_m50	8	**224**	256
substr1	8	72	72
substr2	8	60	60
subtraction1	8	**64**	68

SBSD comparison to FLASH on Wright Lab benchmark functions.

Here we show that we are comparable to program from Wright Labs, which is however much slower since it uses exhaustive search while we use heuristics for variable partitioning.

Recent Publications

- **Stanislaw Grygiel and Marek Perkowski,** ``Labeled Rough Partitions - A New General Purpose Representation for Multiple-Valued Functions and Relations," *Journal of Systems Architecture*, Vol. 47, Issue 1, January 2001, pp. 29-59.

- **Craig Files and Marek Perkowski,** ``New Multivalued Functional Decomposition Algorithms Based on MDDs," ***IEEE Transactions on CAD***, Vol. 19, September 2000, pp. 1081-1086.

Top journal and conference in the field

- **Alan Mishchenko, Bernd Steinbach, and Marek Perkowski,** ``An Algorithm for Bi-Decomposition of Logic Functions," ***Proceedings of Design Automation Conference,*** *DAC 2001*, June 18-22, Las Vegas, pp. 103 - 108.

- **Alan Mishchenko, Bernd Steinbach, and Marek Perkowski,** ``Bi-Decomposition of Multi-Valued Relations," *Proc. 10-th International Workshop on Logic and Synthesis, IWLS'01* , pp. 35 - 40, Granlibakken, CA, June 12 - 15, 2001, IEEE Computer Society and ACM SIGDA.

Part 5

Some Applications

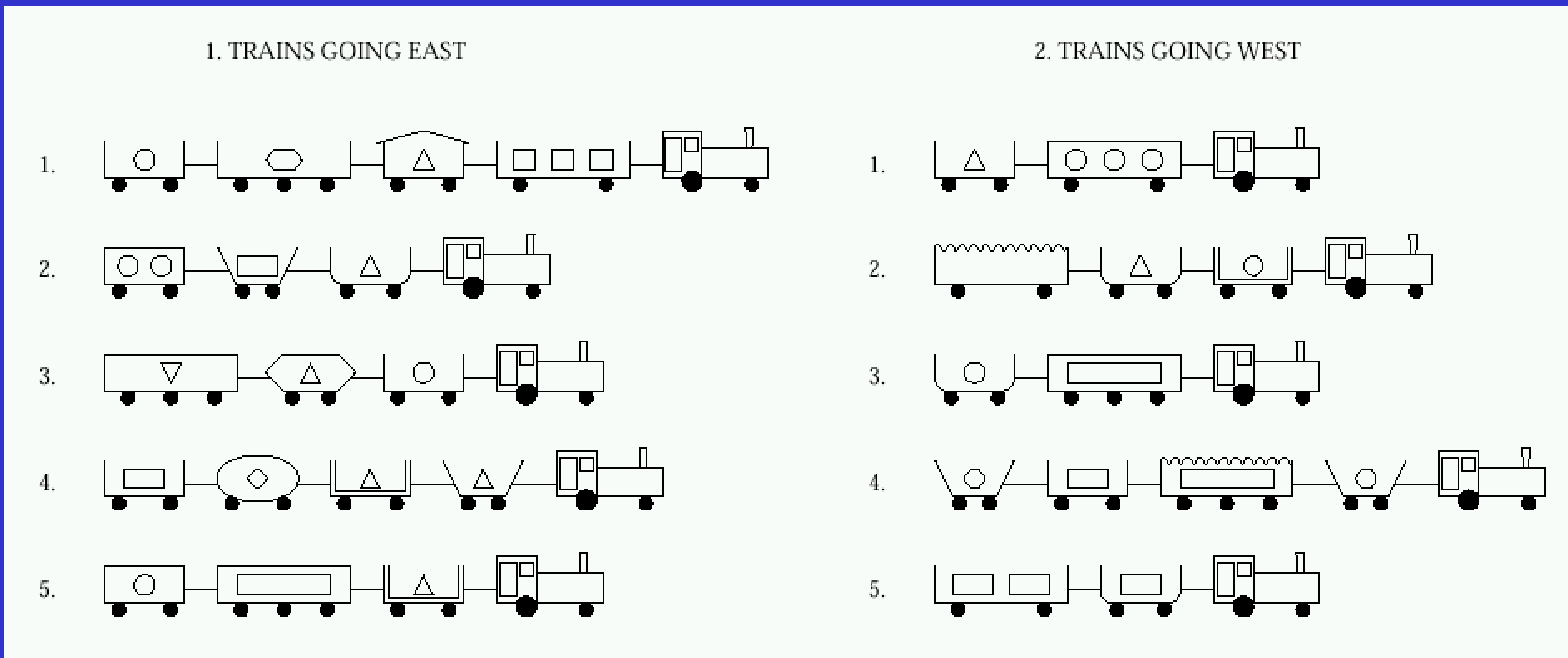
APPLICATIONS

- FPGA SYNTHESIS
- VLSI LAYOUT SYNTHESIS
- DATA MINING AND KNOWLEDGE DISCOVERY
- MEDICAL DATABASES
- EPIDEMIOLOGY
- ROBOTICS
- FUZZY LOGIC DECOMPOSITION
- CONTINUOUS FUNCTION DECOMPOSITION

Example of a application

Knowledge discovery in data with no error

Michalski's Trains



Michalski's Trains

- Multiple-valued functions.
- There are 10 trains, five going East, five going West, and the problem is to nd the simplest rule which, for a given train, would determine whether it is East or Westbound.
- The best rules discovered at that time were:
  - 1. If a train has a short closed car, then it Eastbound and otherwise Westbound.
  - 2. If a train has two cars, or has a car with a jagged roof then it is Westbound and otherwise Eastbound.
- Espresso format. MVGUD format.

Michalski's Trains

```
.type mv
.i 32
.o 1
.ilb size load w0 l0 s0 n0 ls0 w1 l1 s1 n1 ls1 w2 l2 s2 n2 ls2 w3 l3 s3 n3 ls3
     a b c d e f g h i j
.ob direction
.imv 3 4 2 2 10 4 4 2 2 10 3 4 2 2 7 3 4 2 2 8 2 3 2 2 2 2 2 2 2 2 2 2
.omv 2
2 3 0 1 6 3 2 0 0 8 1 3 1 1 6 1 1 0 0 6 1 0 0 1 0 0 0 1 0 0 1 0  0
1 2 0 0 9 1 3 0 0 7 1 2 0 0 0 2 0 - - - - - 0 1 0 1 0 0 0 0 0 0  0
1 1 0 0 6 1 0 0 0 4 1 3 1 1 0 1 3 - - - - - 0 0 0 0 1 0 1 0 0 0  0
```

Michalski's Trains

```
2 1 0 0 7 1 3 0 0 1 1 3 0 0 2 1 2 0 0 6 1 2 1 1 0 0 1 0 0 0 0 0  0
1 2 0 0 1 1 3 1 1 0 1 2 0 0 0 1 0 - - - - - 0 1 0 1 0 0 0 0 0 0  0
0 1 0 1 0 3 0 0 0 6 1 3 - - - - - - - - - - 0 0 0 0 0 0 1 0 0 0  1
1 1 0 0 1 1 0 0 0 9 1 3 0 1 5 0 - - - - - - 0 0 0 0 0 0 1 0 0 0  1
0 1 1 1 0 1 2 0 0 9 1 0 - - - - - - - - - - 0 0 0 1 0 0 0 0 0 0  1
2 1 0 0 7 1 0 0 1 5 1 2 0 0 6 1 2 0 0 7 1 0 1 0 0 1 0 0 0 0 0 0  1
0 0 0 0 9 1 2 0 1 6 2 2 - - - - - - - - - - 1 0 0 0 0 0 0 0 0 0  1
.end
```

where:

`.i`	number of input variables (attributes)
`.o`	number of output variables (attributes)
`.ilb`	input variable names
`.ob`	output variable names
`.imv`	cardinalities of input variables
`.omv`	cardinalities of output variables

Variables 1-2: general attributes

- `size` number of cars (integer in [3-5])
- `load` number of different loads (integer in [1-4])

Variables 3-22: 5 attributes for each of cars 2 through 5: (20 attributes total)

- `w` number of wheels (integer in [2-3])
- `l` length (short or long)
- `s` shape (closedrect,dblopnrect,ellipse,engine,hexagon, jaggedtop, openrect, opentrap, slopetop, ushaped)
- `n` number of loads (integer in [0-3])
- `ls` load shape (circlelod, hexagonlod, rectanglod, trianglod)

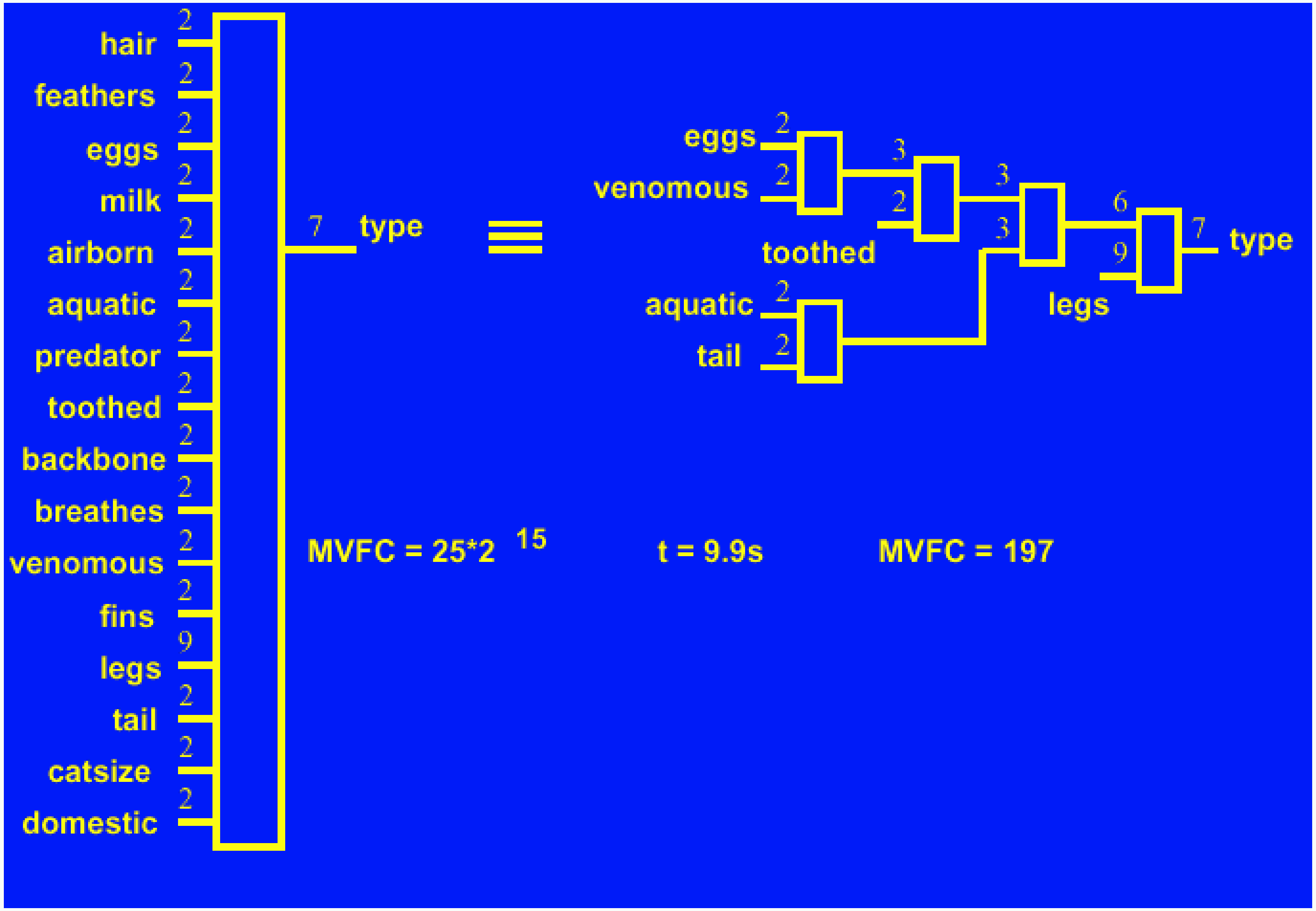
Variables 23-32: 10 Boolean attributes describing whether 2 types of loads are on adjacent cars of the train

- `a` rectangle next to rectangle (0 if false, 1 if true)
- `b` rectangle next to triangle (0 if false, 1 if true)
- `c` rectangle next to hexagon (0 if false, 1 if true)
- `d` rectangle next to circle (0 if false, 1 if true)
- `e` triangle next to triangle (0 if false, 1 if true)
- `f` triangle next to hexagon (0 if false, 1 if true)
- `g` triangle next to circle (0 if false, 1 if true)
- `h` hexagon next to hexagon (0 if false, 1 if true)
- `i` hexagon next to circle (0 if false, 1 if true)
- `j` circle next to circle (0 if false, 1 if true)

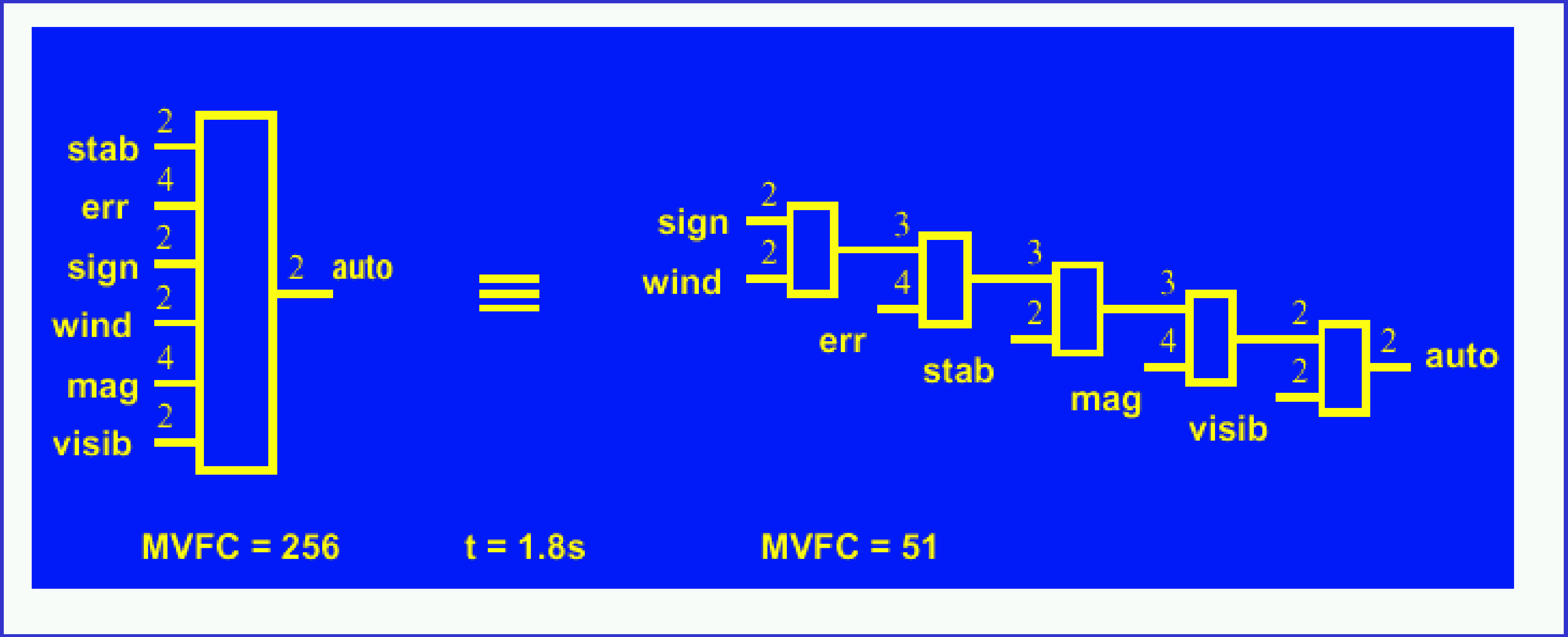
Michalski's Trains

- Attribute 33: Class attribute (east or west)
  - direction (east = 0, west = 1)
- The number of cars vary between 3 and 5. Therefore, attributes referring to properties of cars that do not exist (such as the 5 attributes for the "5th" car when the train has fewer than 5 cars) are assigned a value of "-".
- Applied to the trains problem our program discovered the following rules:
  - 1. If a train has triangle next to triangle or rectangle next to triangle on adjacent cars then it is Eastbound and otherwise Westbound.
  - 2. If the shape of car 1 (s1) is jagged top or open rectangle or u-shaped then it is Westbound and otherwise Eastbound.

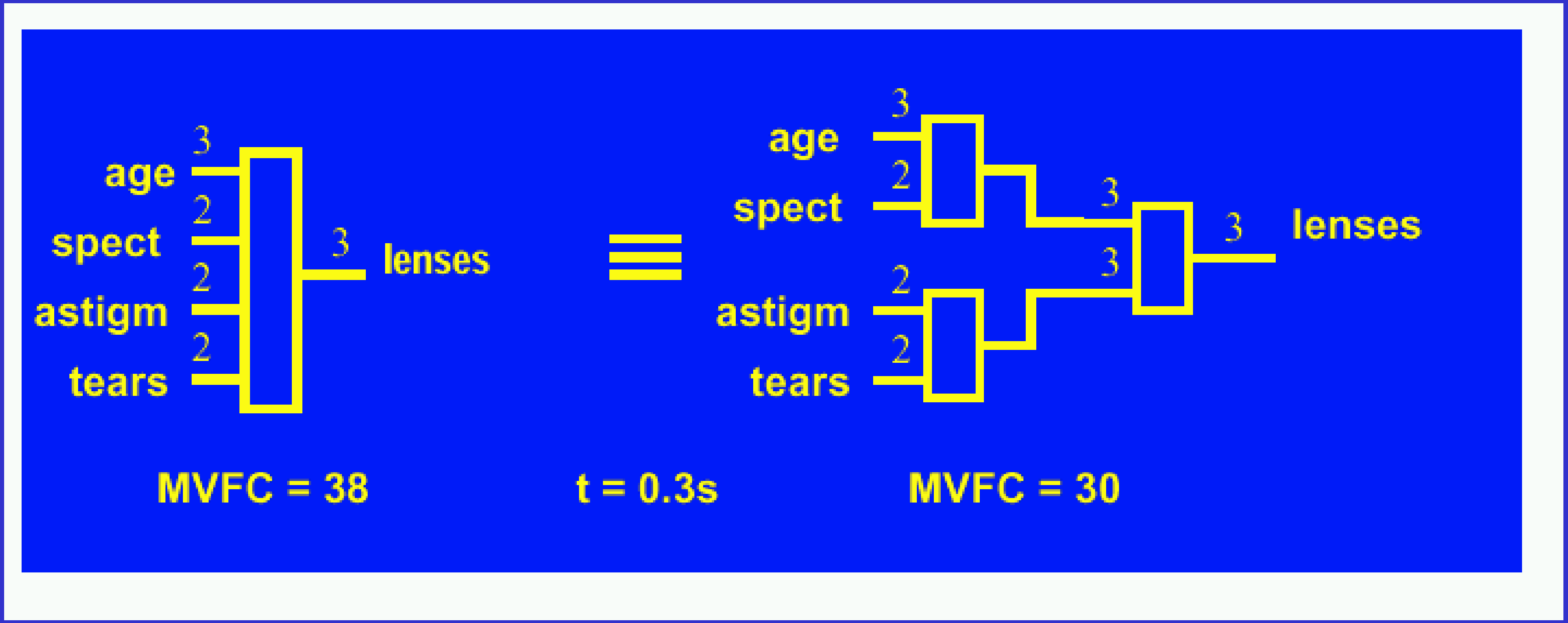
MV benchmarks: zoo



MV benchmarks: shuttle



MV benchmarks: lenses



Example of a application

Medical data bases with error

Evaluation of results for learning

- 1. Learning Error

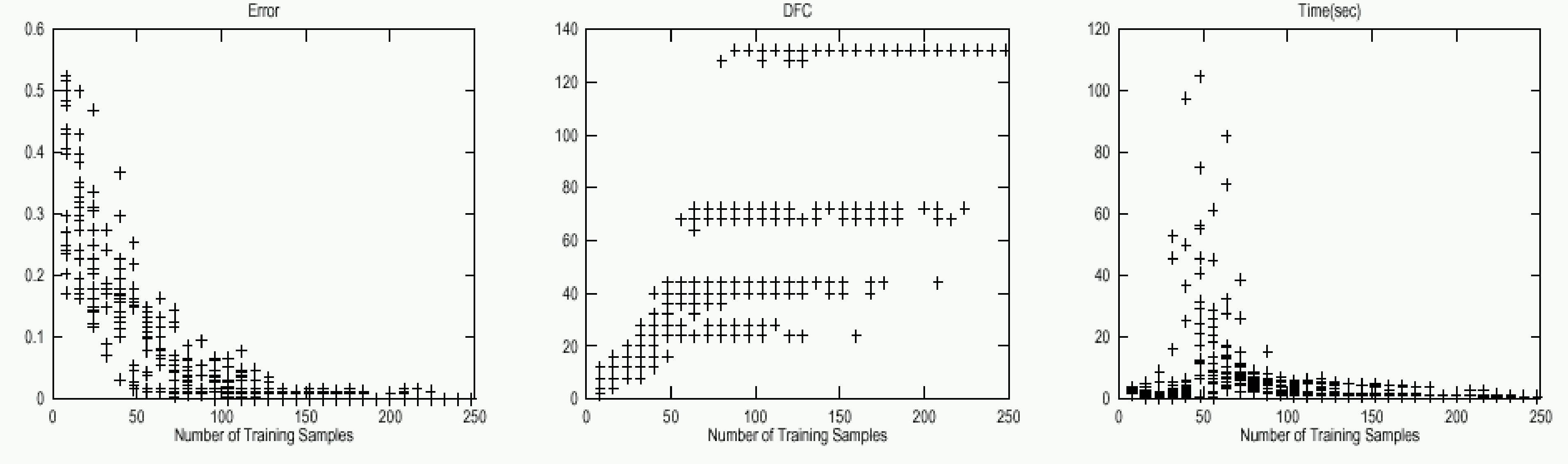
$$error = \frac{\#\ of\ incorrectly\ classified\ samples}{total\ \#\ of\ samples}$$

- 2. Occam Razor , complexity

A machine learning approach versus several logic synthesis approaches

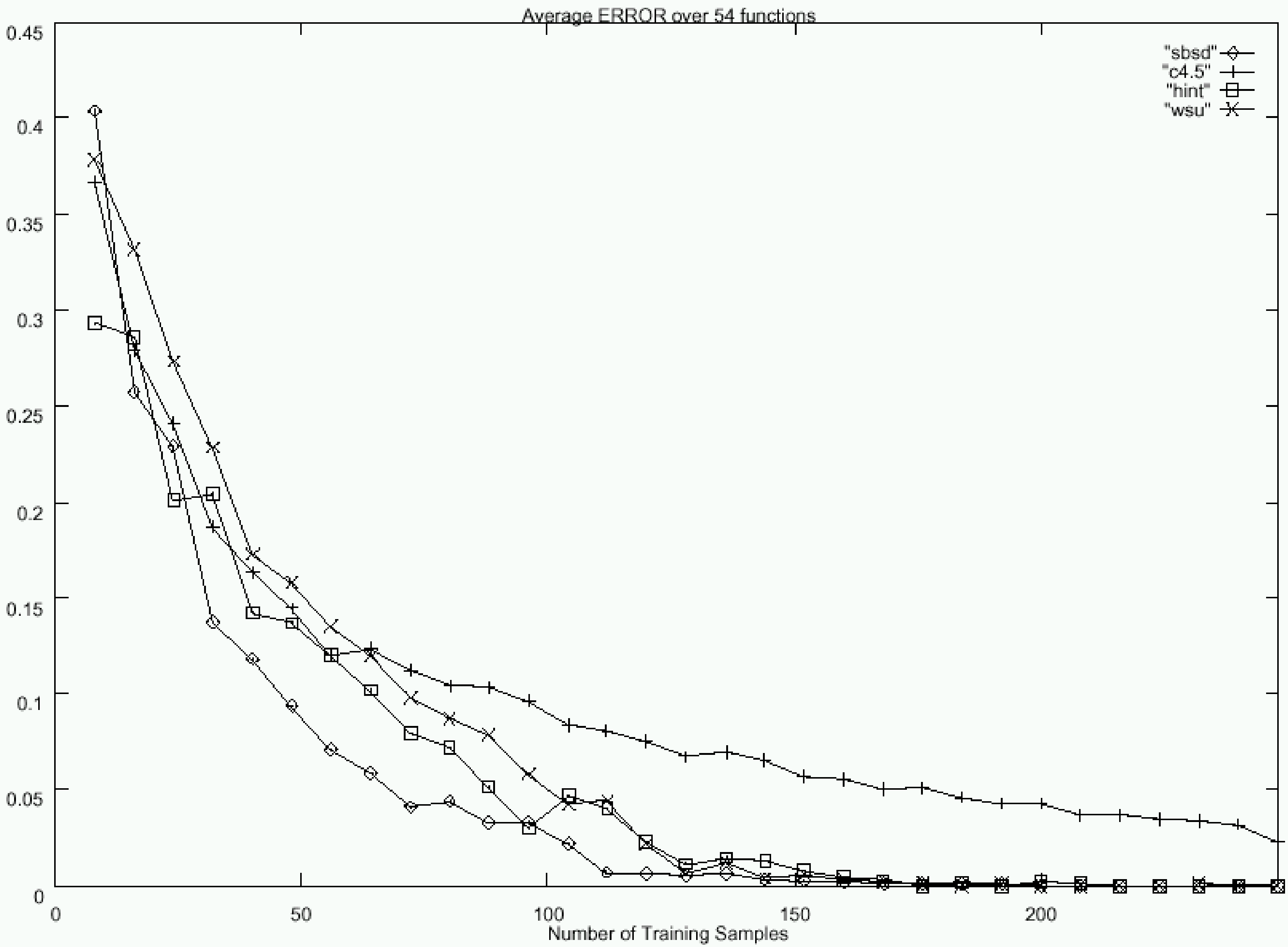
Original Function	Known DFC	Average Error			Number of Samples		
		C4.5	Decomp.	Espresso	C4.5	Decomp.	Espresso
kdd1	2	0	0	0	8	7	9
kdd2	8	0.32	0	0.96	31	25	40
kdd3	8	6.35	0	5.64	83	25	51
kdd4	12	2.48	3.72	2.64	74	67	76
kdd5	12	1.28	2.72	3.52	61	76	54
kdd6	16	2.76	2.4	12.86	97	126	113
kdd7	20	17.52	8.18	17.16	200	60	181
kdd8	20	13.79	6.55	16.54	224	104	205
kdd9	28	20.69	10.53	5.69	256	126	51
kdd10	36	10.52	11.11	8.44	249	251	229
Average		7.57	4.52	7.35	128.3	86.7	100.9

Finding the error, DFC, and time of the decomposer on the benchmark kdd5.

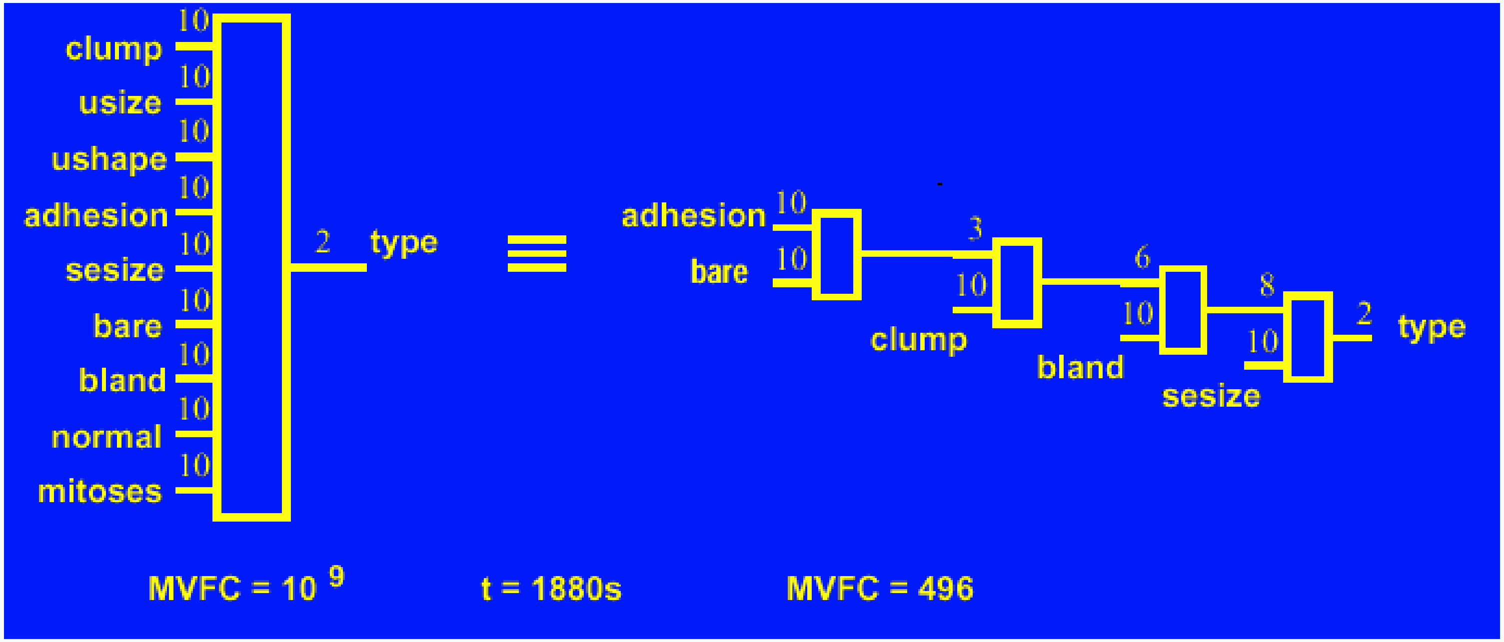


We use learning curves to evaluate quality of our software variants and compare our software to the competitors

The average error over 54 benchmark functions.



MV benchmarks: breastc



Conclusion

- Stimulated by practical hard problems:
  - Field Programmable Gate Arrays (FPGA),
  - Application Specific Integrated Circuits (ASIC)
  - high performance custom design (Intel)
  - Very Large Scale of Integration (VLSI) layout-driven synthesis for custom processors,
  - robotics (hexapod gaits, face recognition),
  - Machine Learning,
  - Data Mining.

Conclusion

- Developed 1989-present
- Intel, Washington County epidemiology office, Northwest Family Planning Services, Lattice Logic Corporation, Cypress Semiconductor, AbTech Corp., Air Force Office of Scientific Research, Wright Laboratories.
- <u>A set of tools</u> for decomposition of binary and multi-valued functions and relations.
- Extended to fuzzy logic, reconstructability analysis and real-valued functions.

Conclusion

- Our recent software allows also for bi-decomposition, removal of vacuous variables and other preprocessing/postprocessing operations.
- Variants of our software are used in several commercial companies.
- The applications of the method are unlimited and it can be used whenever decision trees or artificial neural nets are used now.
- The quality of learning was better than in the top decision tree creating program C4.5 and various neural nets.
- The only problem that remains is speed in some applications.
- Recent version included in MVSIS tools from U.C. Berkeley.
- This is still work in progress and you can contribute to new applications and software variants tuned to them.

Conclusion

- **On our WWW page,**

  http:// www.ee.pdx.edu/~cfiles/papers.html

  the reader can find many benchmarks from various disciplines that can be used for comparison of machine learning and logic synthesis programs.
- We plan to continue work on decomposition and its various practical applications such as epidemiology or robotics which generate large real-life benchmarks.
- We work on FPGA-based reconfigurable hardware accelerator for decomposition to be used on a mobile robot.
- We are interested in potential other applications for which large database exist and the is large benefit of practical application.