DECOMPOSITION OF RELATIONS:

A NEW APPROACH TO CONSTRUCTIVE INDUCTION IN MACHINE LEARNING AND DATA MINING - AN OVERVIEW

Marek Perkowski

Portland State University

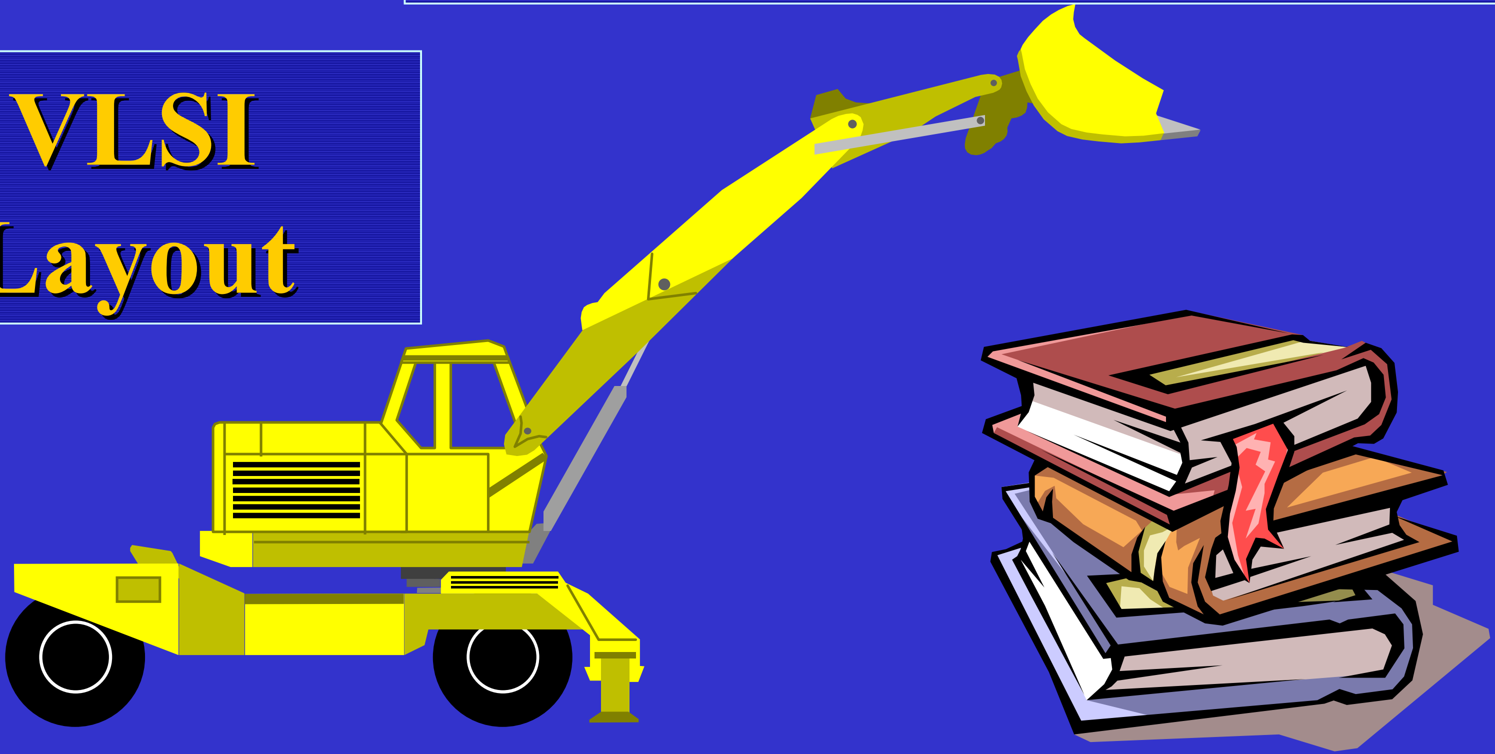
Data Mining Application for Epidemiologists

Control of a robot

FPGA

Machine Learning from Medical databases

VLSI Layout



- This is a review paper that presents work done at Portland State University and associated groups in years 1989 - 2001 in the area of functional decomposition of multi-valued functions and relations, as well as some applications of these methods.

Group Members

Current Students:

Tu Dinh

Michael Levy

Faculty

Marek Perkowski

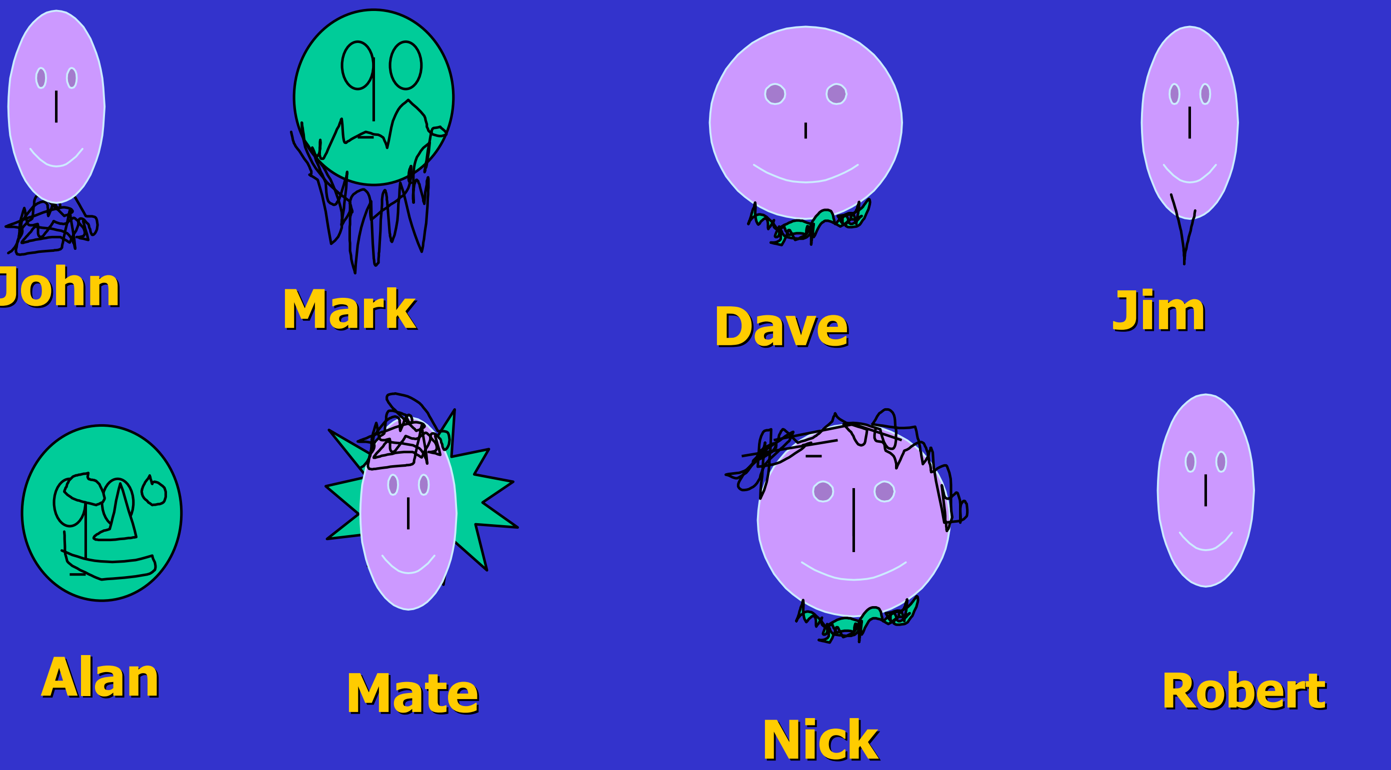
Alan Mishchenko

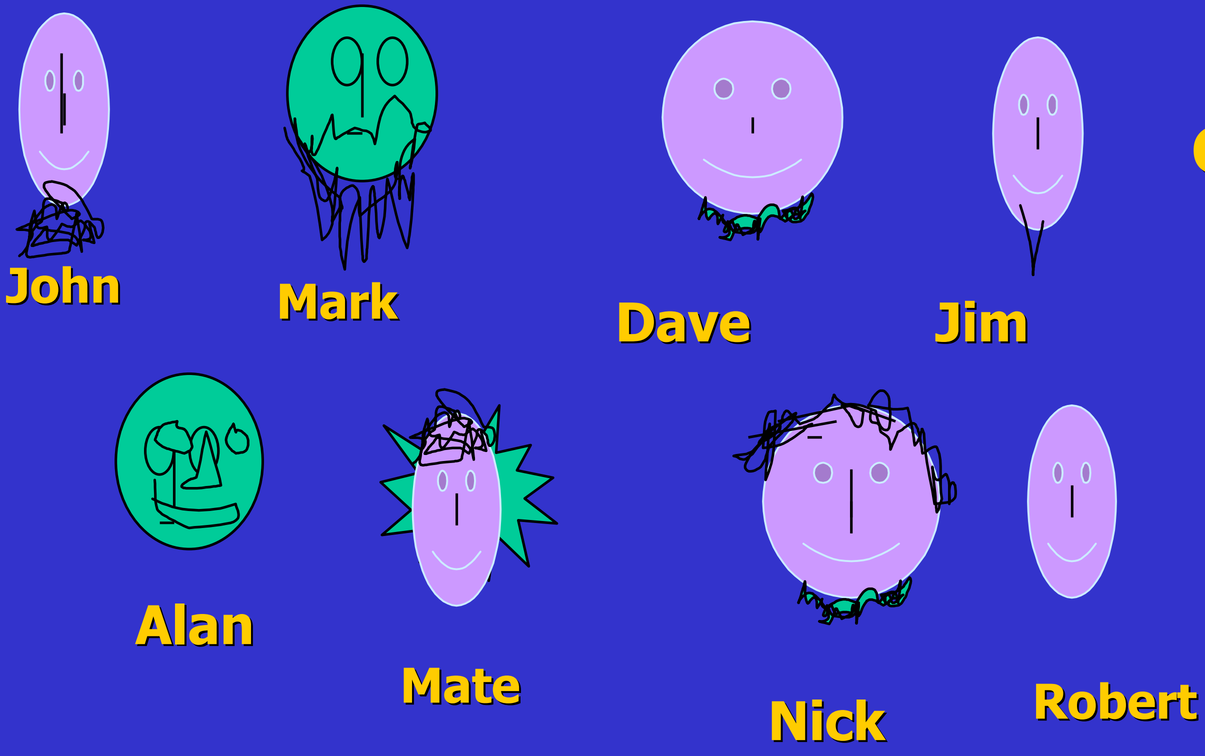
Researchers:

Stanislaw Grygiel, Ph.D., **Intel**
Craig Files, Ph.D., **AbTech.**
Paul Burkey, **Intel**
Rahul Malvi, **Synopsys**
Michael Burns, **Vlsi logic**,
Timothy Brandis, **OrCAD**

Essence of logic synthesis approach to learning

Example of Logical Synthesis





Good guys

Bad guys

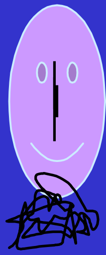
A - size of hair

B - size of nose

C - size of beard

D - color of eyes

Good guys

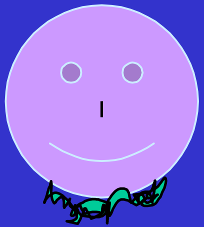


John

A′ BCD

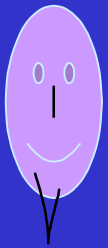
Mark

A′ BCD′



Dave

A′ B′CD



Jim

A′ B′CD

CD AB	00	01	11	10
00	-	-	1	-
01	–	-	1	1
11	-	–	-	-
10	-	-	-	-

A - size of hair

B - size of nose

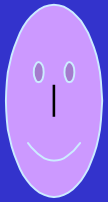
C - size of beard

D - color of eyes

Alan

Mate

Nick



Robert

Bad guys

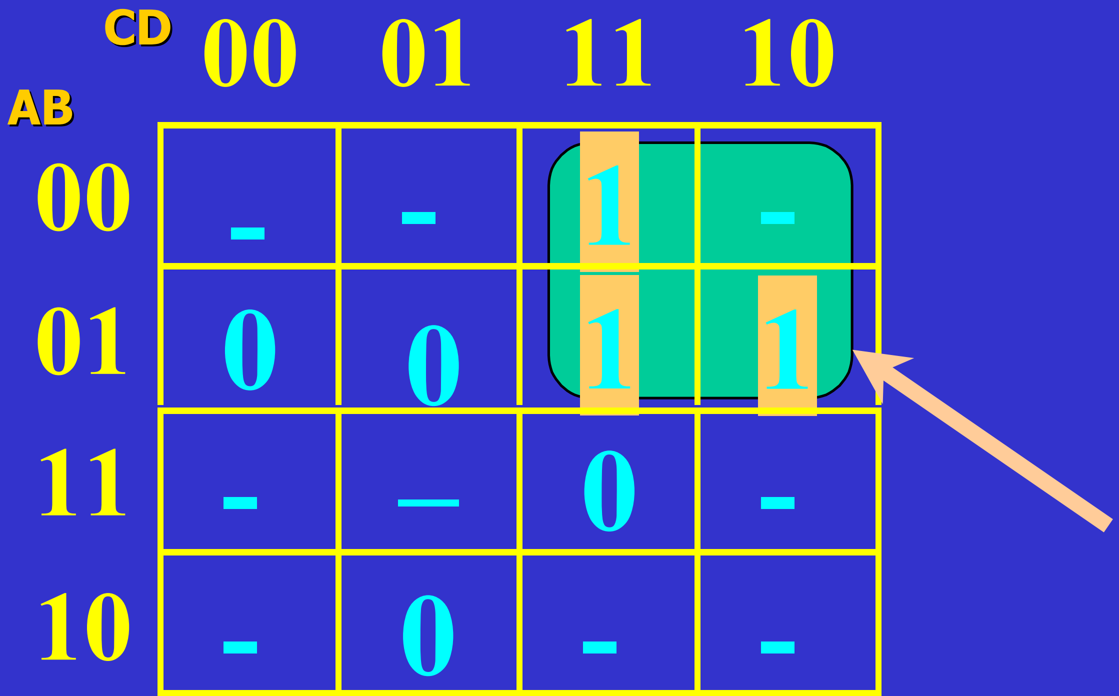
A′ BC′D′ AB′C′D ABCD A′ B′C′D

A - size of hair

B - size of nose

C - size of beard

D - color of eyes



CD AB	00	01	11	10
00	-	-	1	-
01	0	0	1	1
11	-	—	0	-
10	-	0	-	-

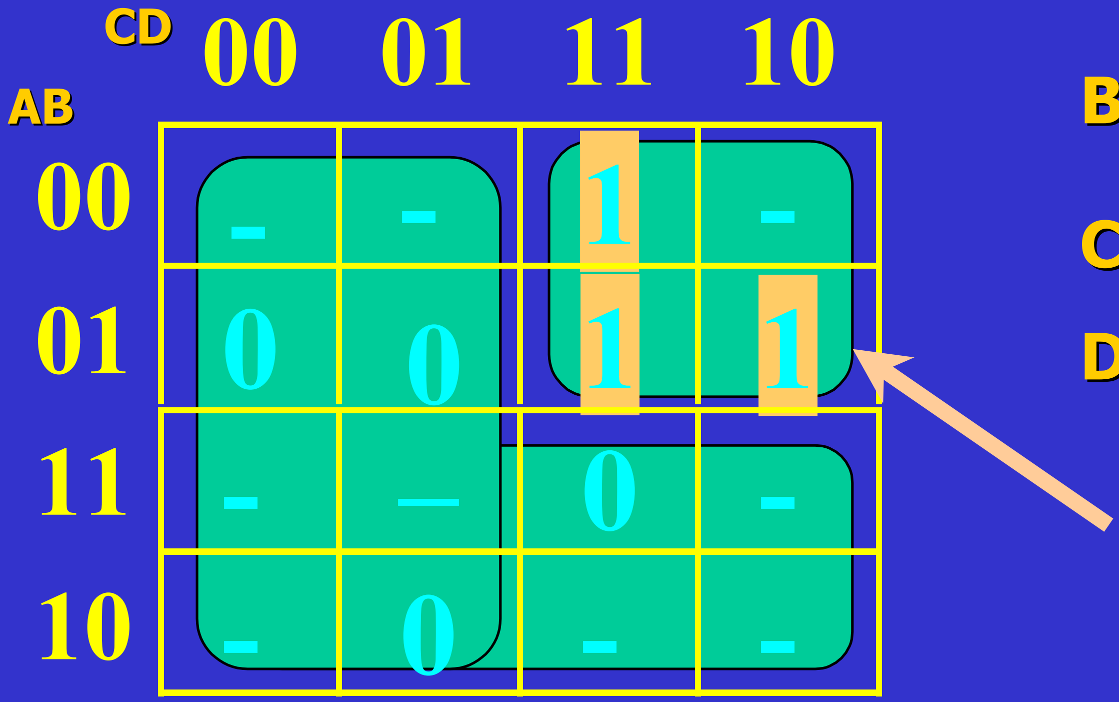
A′C

Generalization 1:

Bald guys with beards are good

Generalization 2:

All other guys are no good



AB \ CD	00	01	11	10
00	-	-	1	-
01	0	0	1	1
11	-	—	0	-
10	-	0	-	-

A′C

A - size of hair

B - size of nose

C - size of beard

D - color of eyes

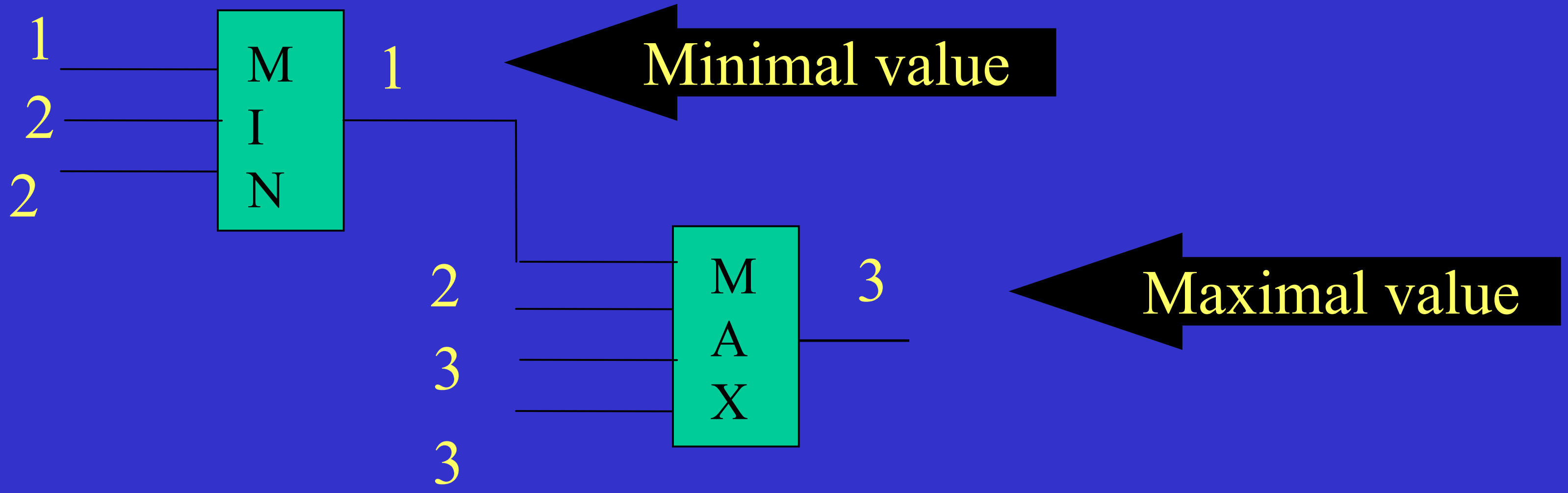
Short Introduction: multiple-valued logic

Signals can have values from some set, for instance {0,1,2}, or {0,1,2,3}

{0,1} - binary logic (a special case)

{0,1,2} - ***a ternary logic***

{0,1,2,3} ***- a quaternary logic,*** **etc**



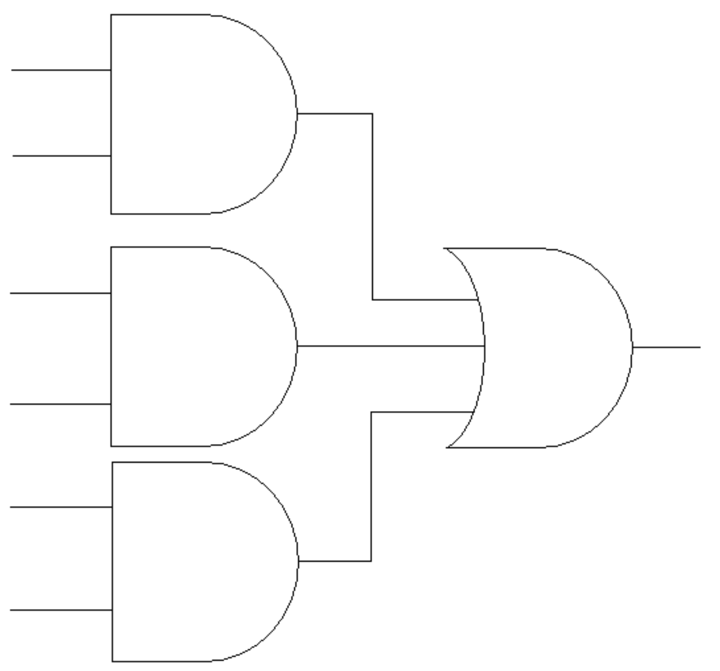
Types of Logical Synthesis

- Sum of Products
- Decision Diagrams
- Functional Decomposition

The method we are using

Sum of Products

AND gates, followed by an OR gate that produces the output. (Also, use Inverters as needed.)



Decision Diagrams

A Decision diagram breaks down a Karnaugh map into set of decision trees.

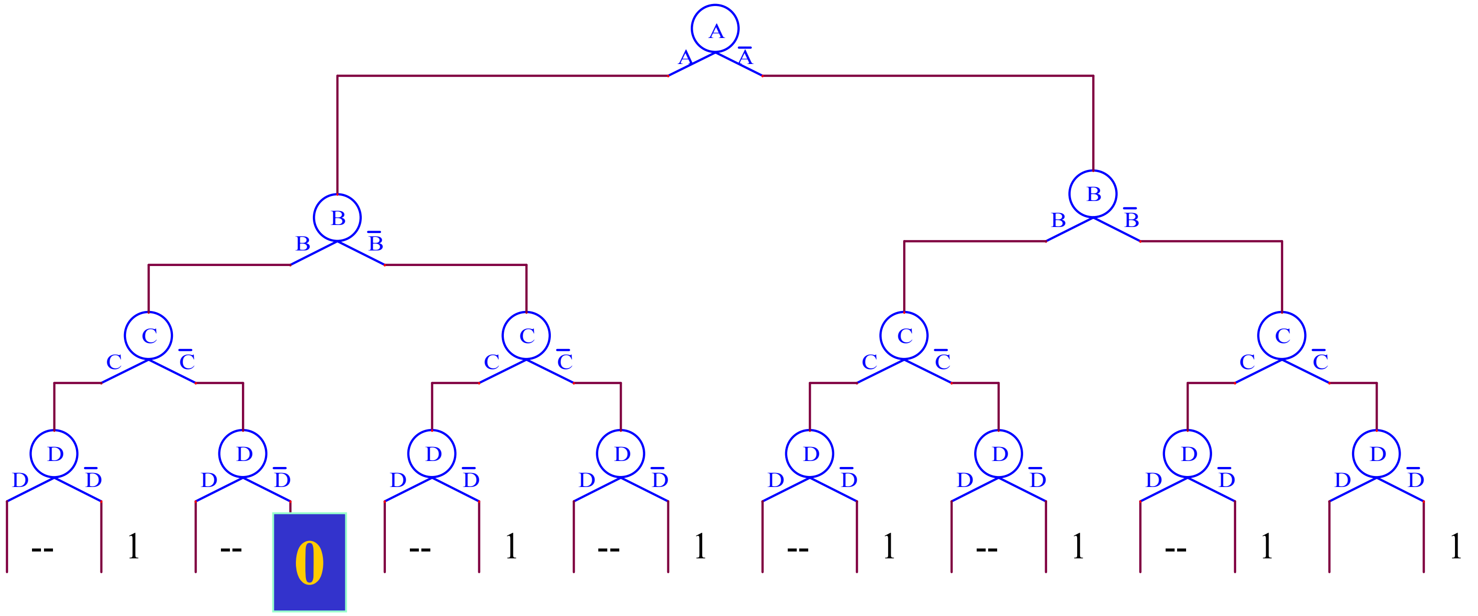
A decision diagram ends when all of branches have a yes, no, or do not care solution.

This diagram can become quite complex if the data is spread out as in the following example.

Example Karnaugh Map

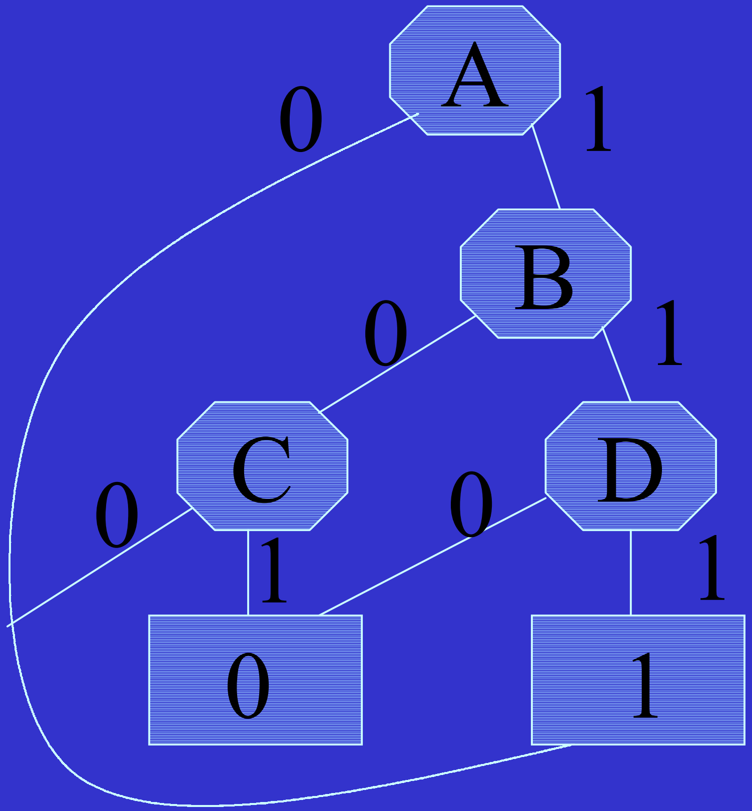
AB\CD	00	01	10	11
00	1	-	1	-
01	-	1	-	1
10	1	-	1	-
11	0	1	-	1

Decision Tree for Example Karnaugh Map



BDD Representation of function

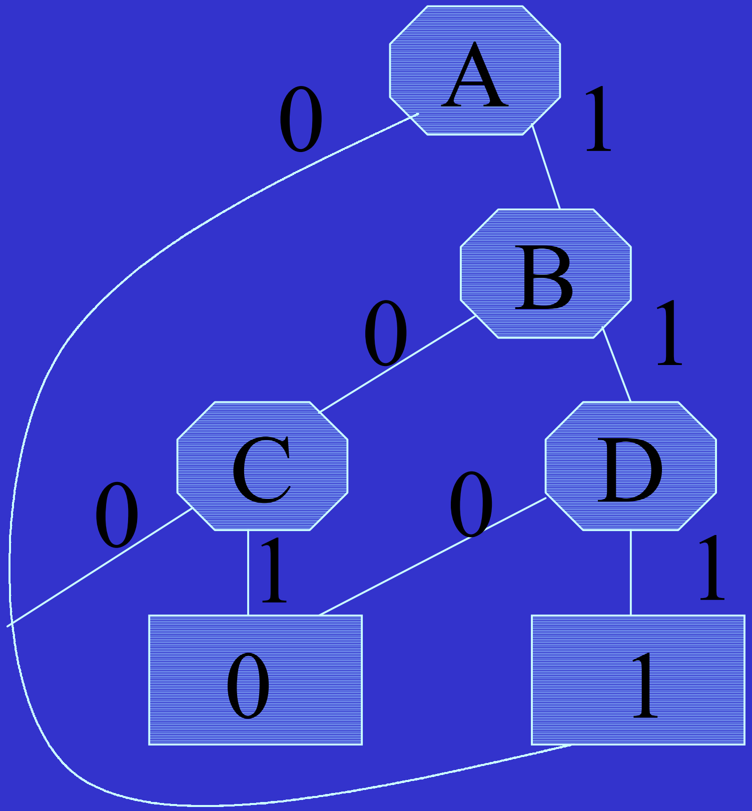
CD AB	00	01	11	10
00	-	1	1	-
01	1	-	1	1
11	0	1	-	0
10	-	1	0	-



Incompletely specified function

BDD Representation of function

AB \ CD	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	0	1	1	0
10	1	1	0	0

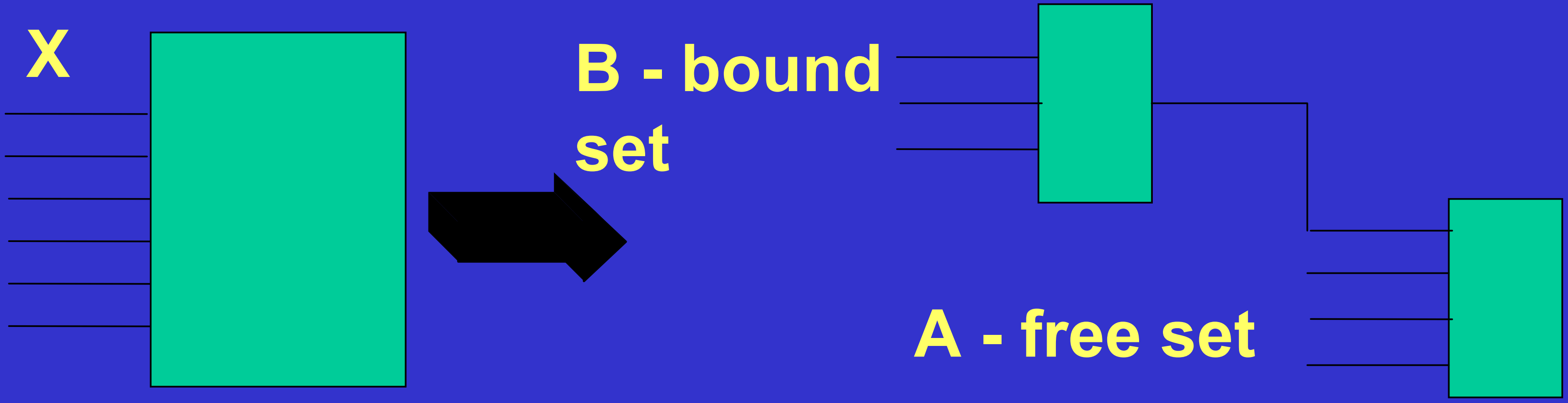


Completely specified function

Functional Decomposition

Evaluates the data function and attempts to decompose into simpler functions.

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



X

B - bound set

A - free set

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

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

Pros and cons

In generating the final combinational network, BDD decomposition, based on multiplexers, and SOP decomposition, trade flexibility in circuit topology for time efficiency

Generalized functional decomposition sacrifices speed for a higher likelihood of minimizing the complexity of the final network

Overview of data mining

What is Data Mining?

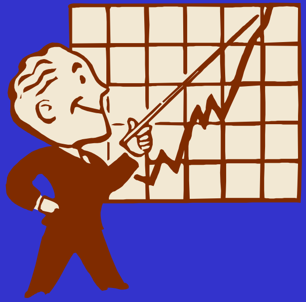
Databases with millions of records and thousands of fields are now common in business, medicine, engineering, and the sciences.

To extract useful information from such data sets is an important practical problem.

Data Mining is the study of methods to find useful information from the database and use data to make predictions about the people or events the data was developed from.

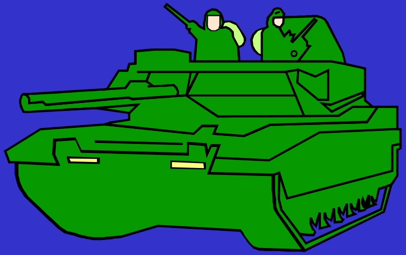
Some Examples of Data Mining

1) Stock Market Predictions



2) Large companies tracking sales

3) Military and intelligence applications



Data Mining in Epidemiology

Epidemiologists track the spread of infectious disease and try to determines the diseases original source

Often times Epidemiologist only have an initial suspicions about what is causing an illness. They interview people to find out what those people that got sick have in common.

Currently they have to sort through this data by hand to try and determine the initial source of the disease.

A data mining application would speed up this process and allow them to quickly track the source of an infectious diseases

Types of Data Mining

Data Mining applications use, among others, three methods to process data

1) Neural Nets

2) Statistical Analysis

3) Logical Synthesis

The method we are using

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	-

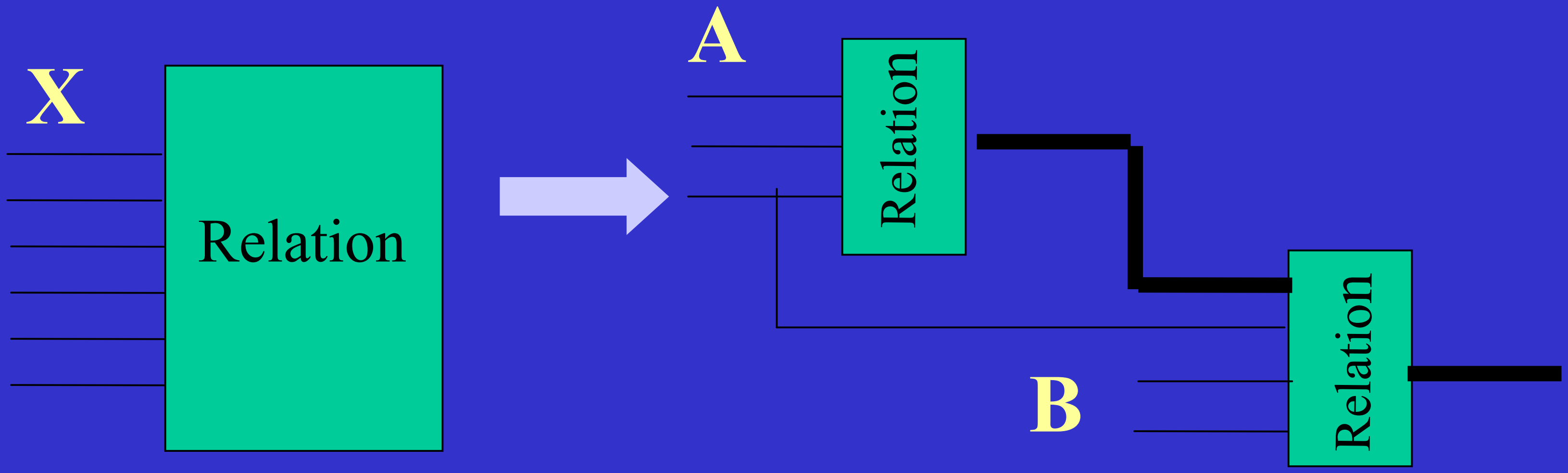
z

Columns 0 and 1
and
columns 0 and 2
are compatible

column
compatibility = 2

Decomposition of Multi-Valued Relations

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



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

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

Forming a CCG from a K-Map

Bound Set

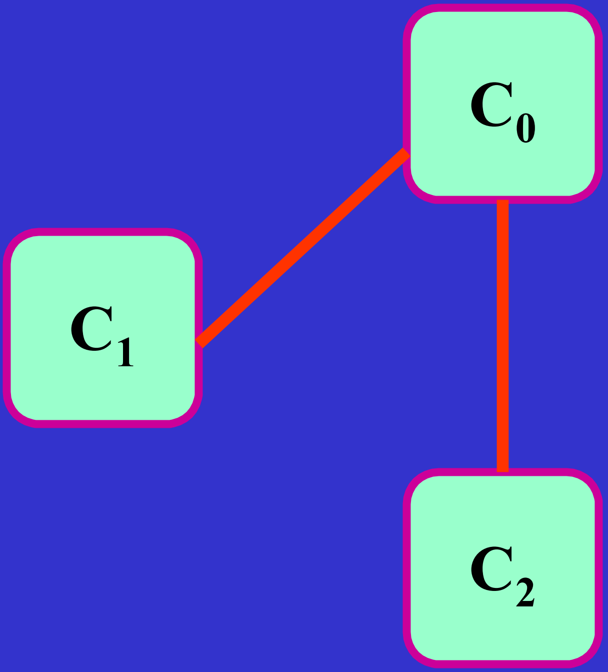
Free Set

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

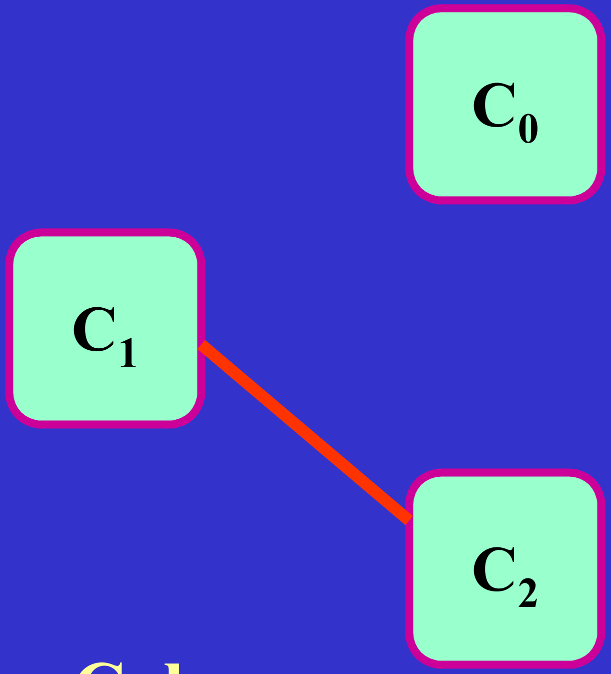
Forming a CIG from a K-Map

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

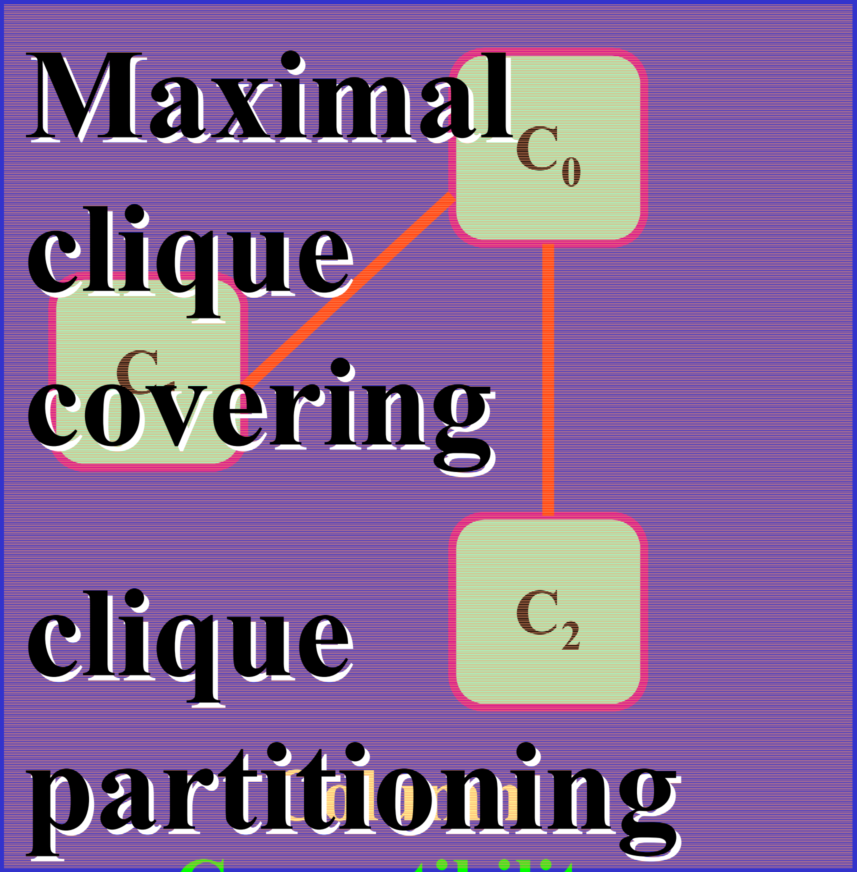
Columns 1 and 2 are incompatible

chromatic number = 2



Column Incompatibility Graph

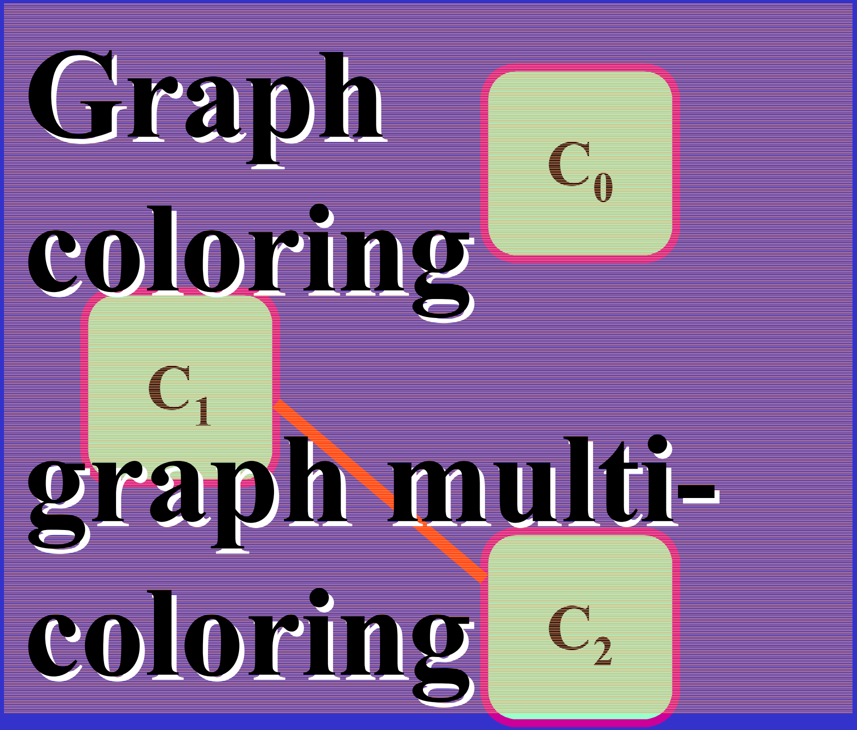
CCG and CIG are complementary



Maximal clique covering

clique partitioning

Column
Compatibility
Graph

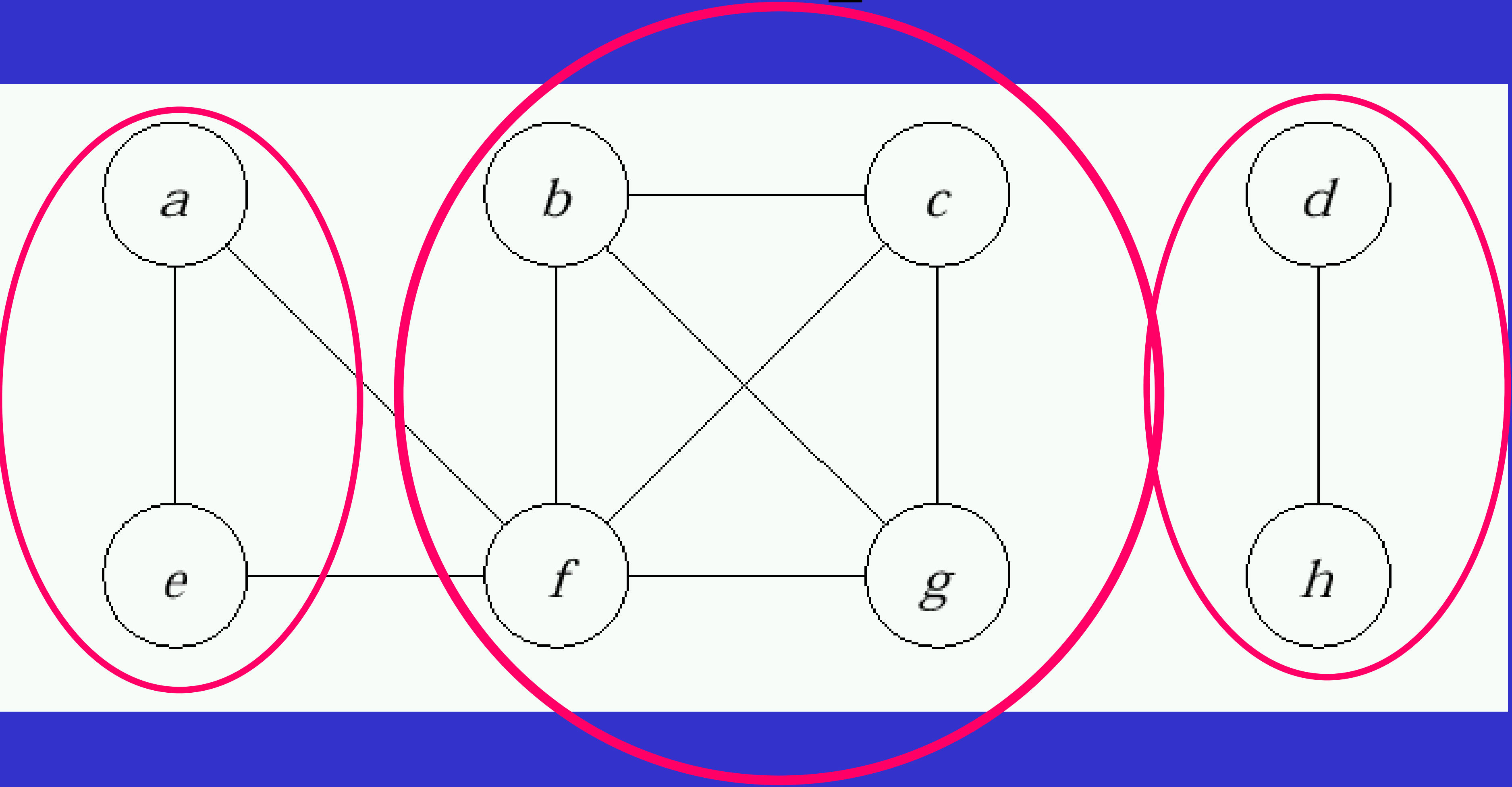


Graph coloring

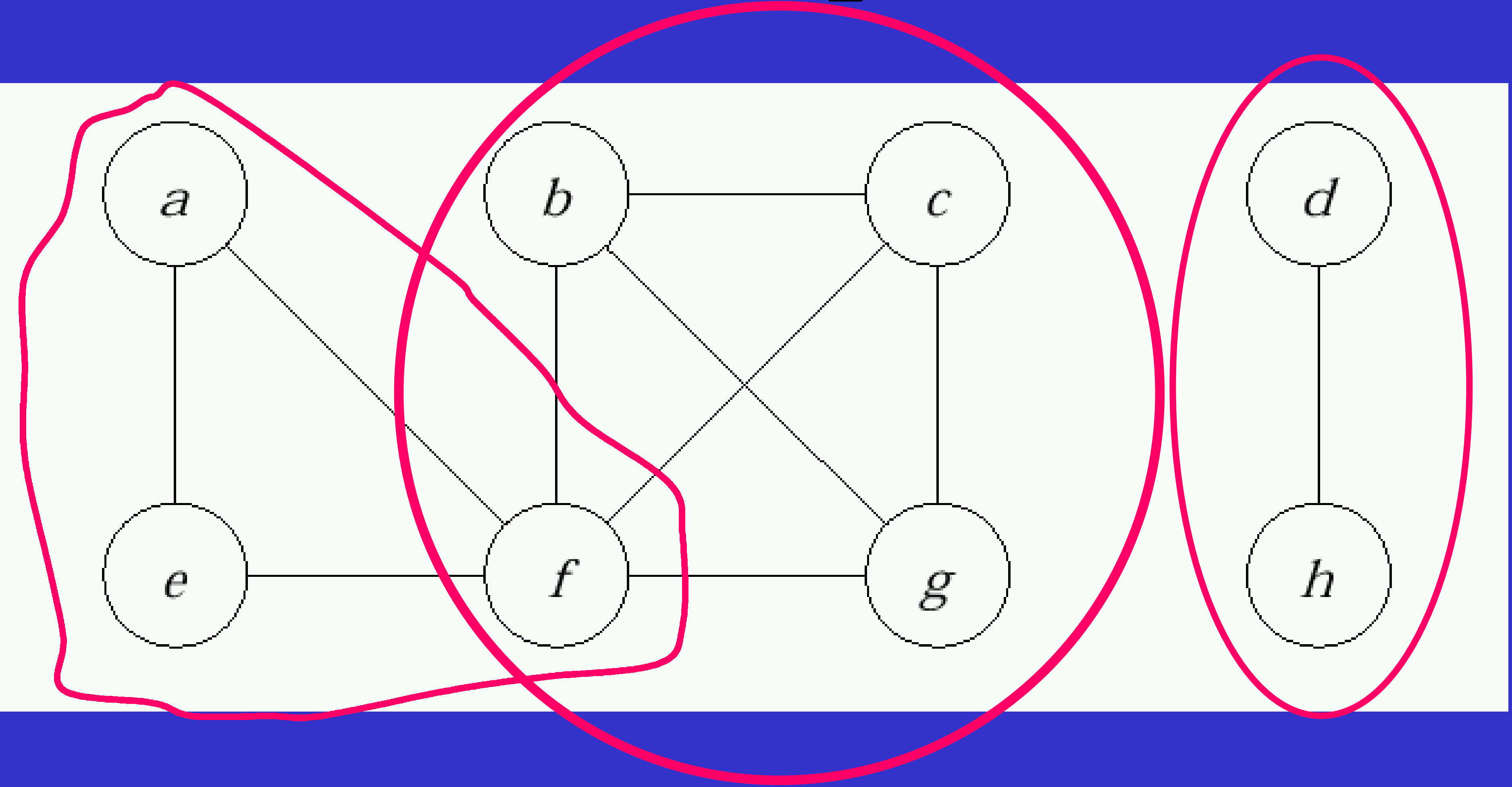
graph multi-coloring

Column
Incompatibility
Graph

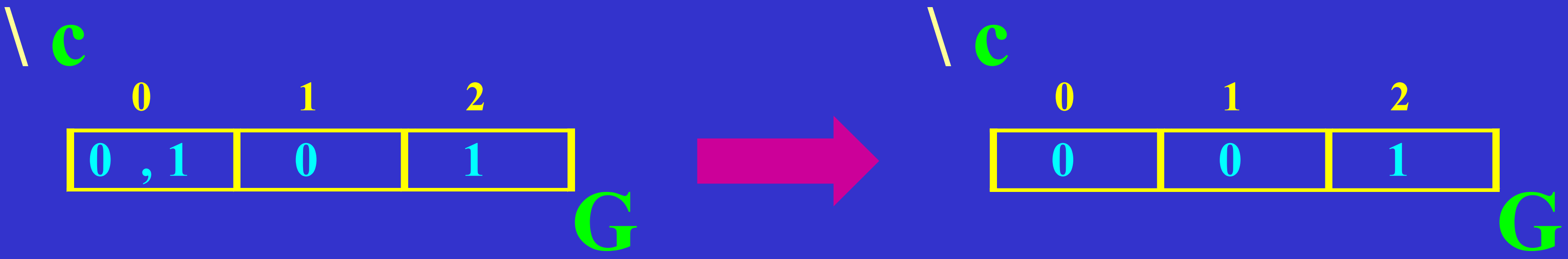
clique partitioning example.



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

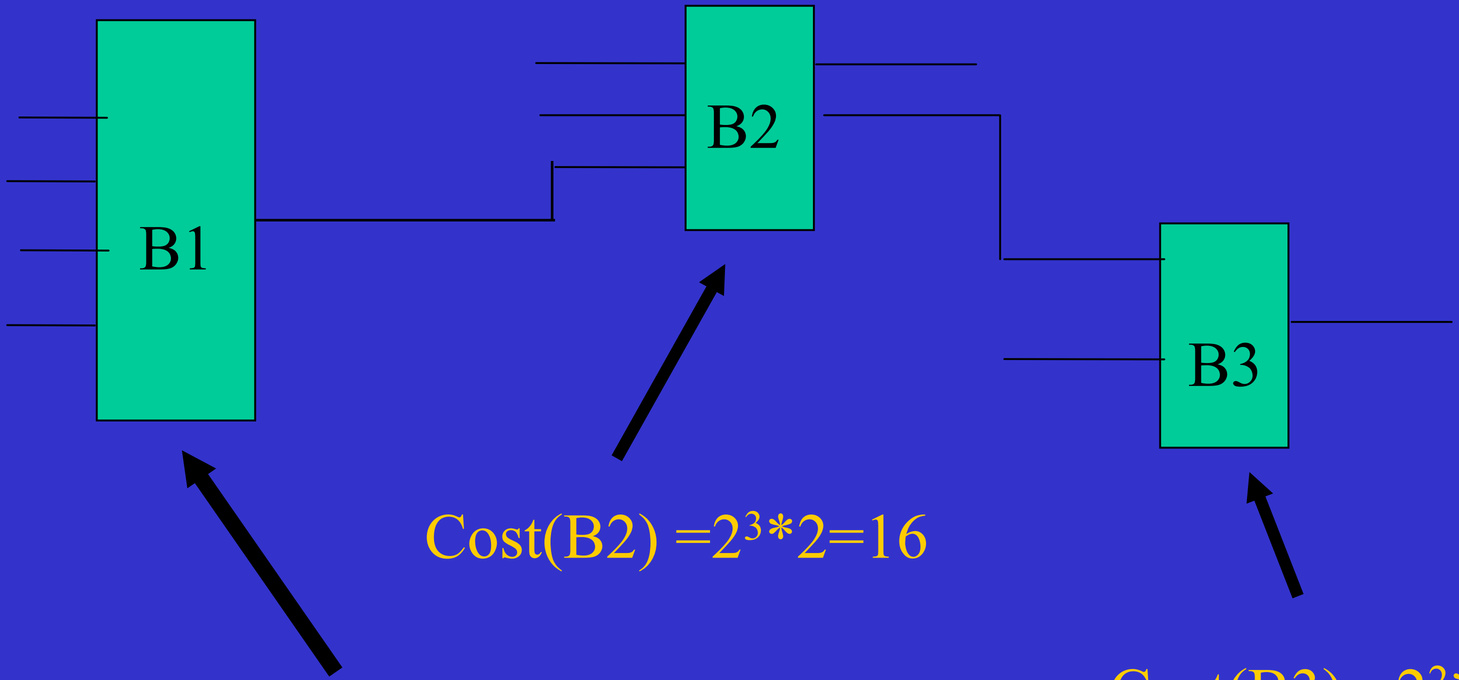
$$\text{Cost} := m * 2^n$$

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

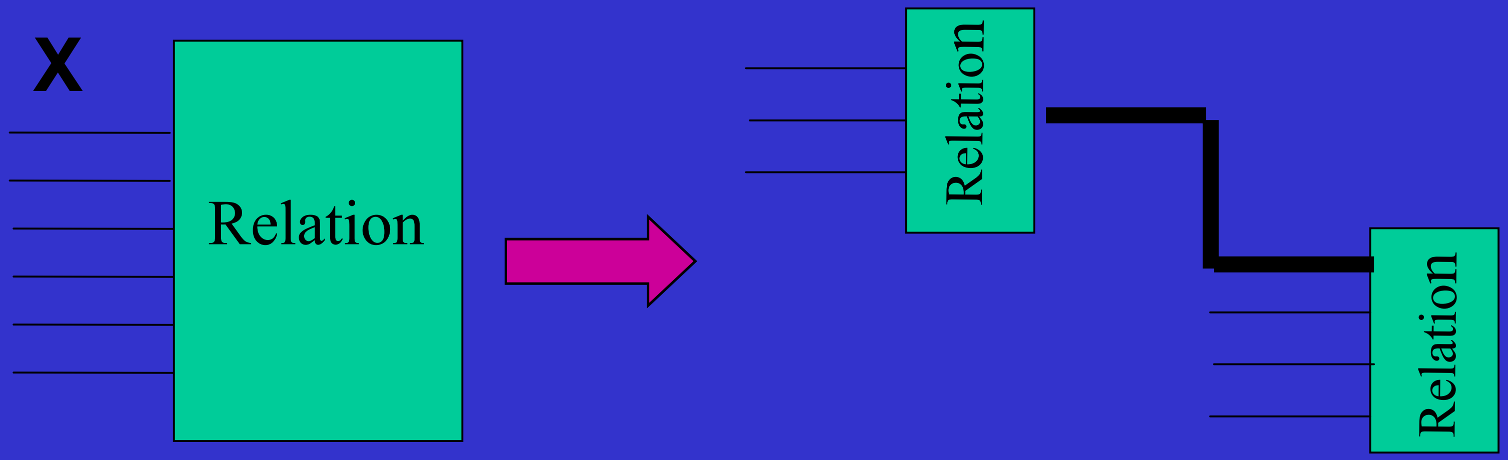
New Complexity Measures

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

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



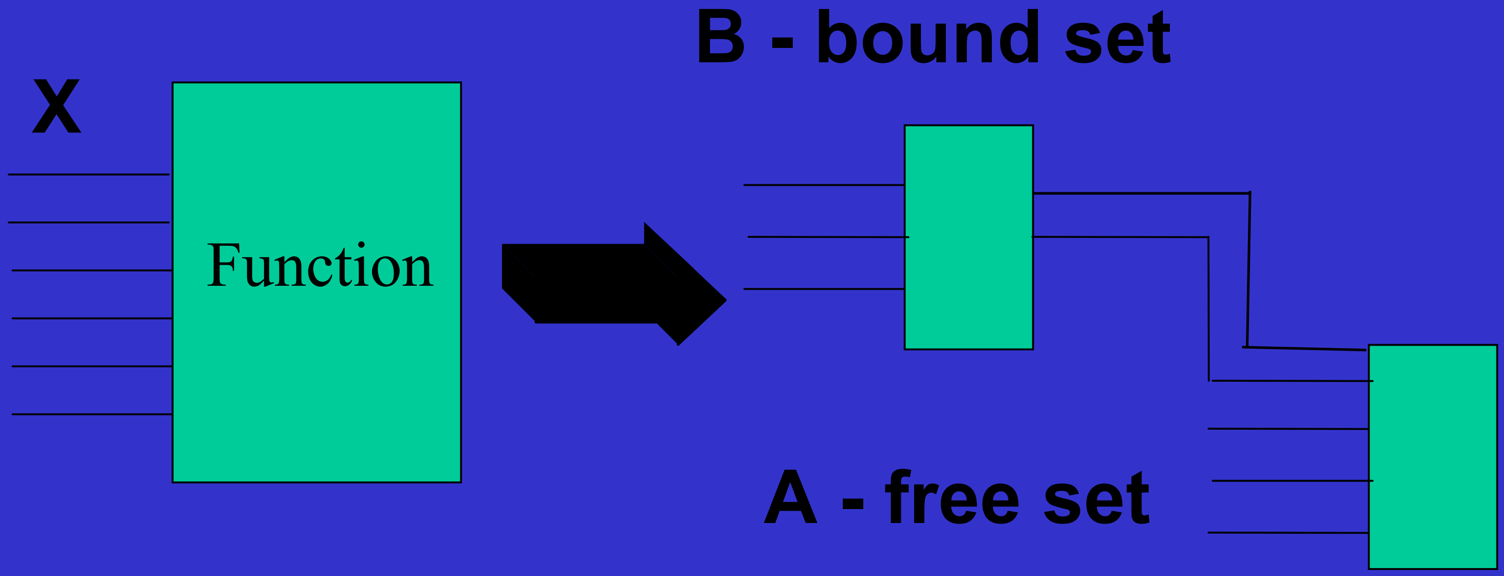
$RC_{before} = (3*3*3)*(\log_2 4) = 54$

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

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

Two-Level Curtis Decomposition

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



B - bound set

A - free set

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

if $A \cap B \neq \varnothing$, 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

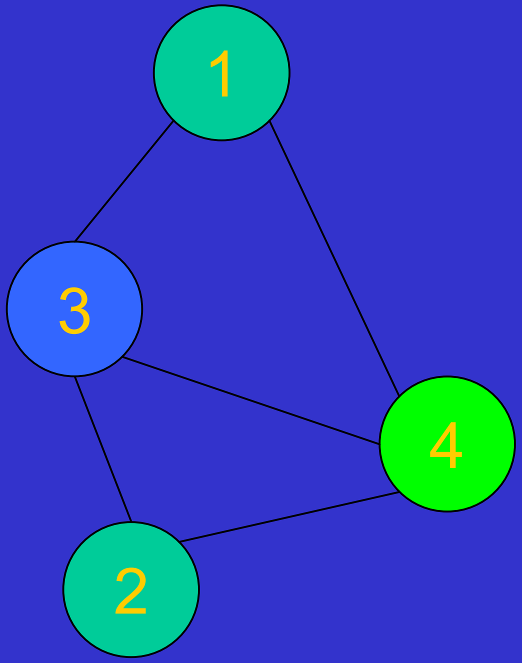
Column Multiplicity

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

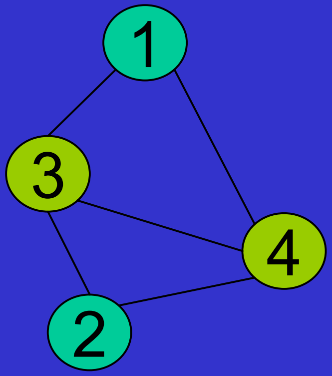


Column Multiplicity-other example

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

X=G(C,D)

X=C in this case

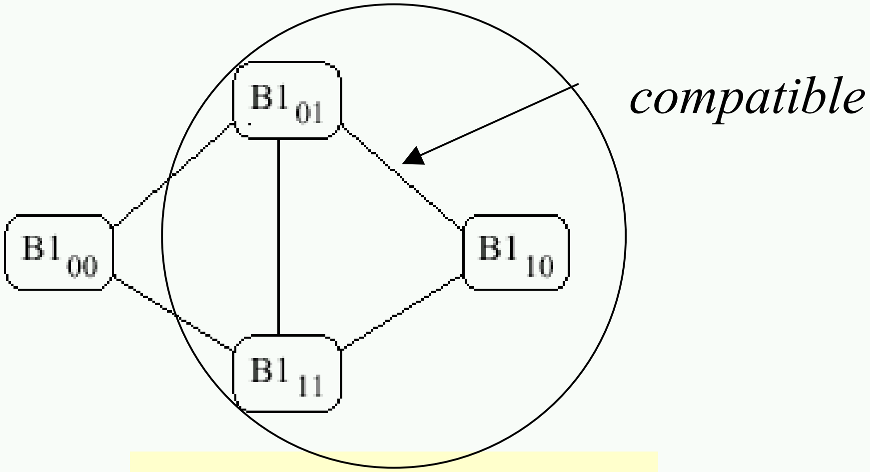
But how to calculate function H?

Decomposition of multiple-valued relation

	$B1_{00}$	$B1_{01}$	$B1_{11}$	$B1_{10}$
ab \ cd	00	01	11	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

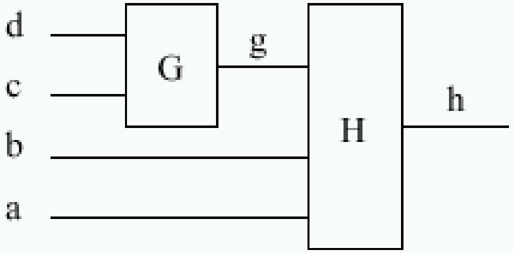
c \ d	0	1
0	0	0,1
1	1	0,1

Kmap of block G

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

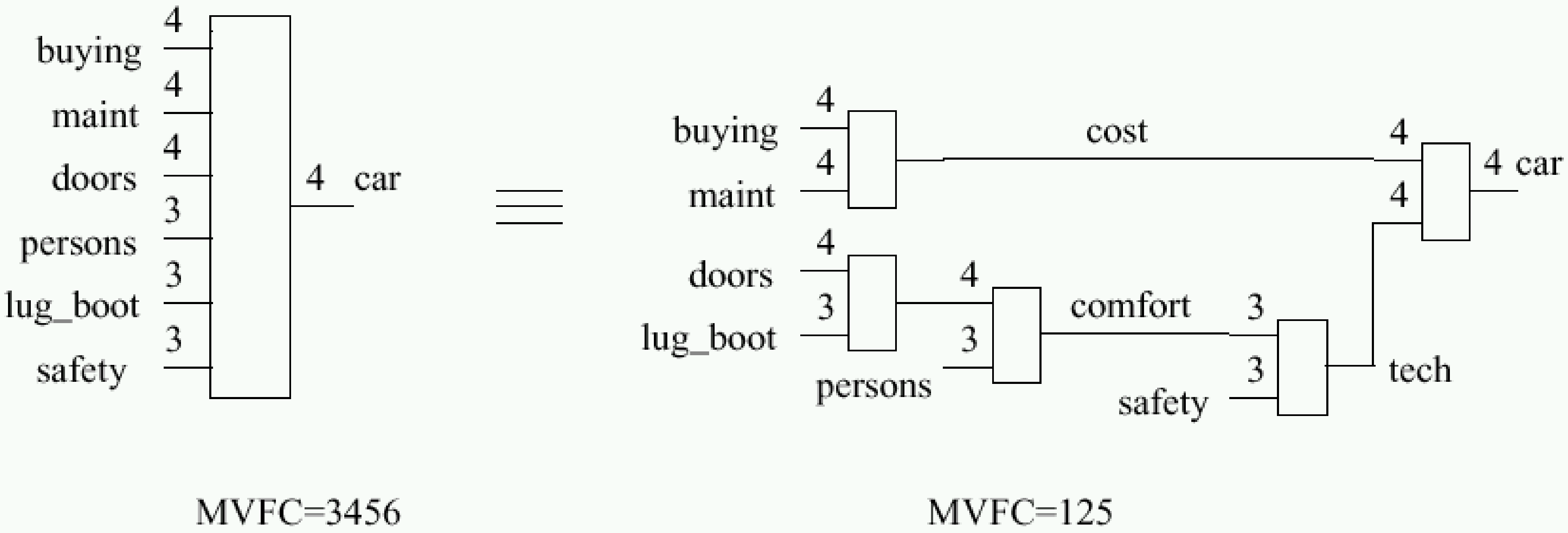
h

Kmap of block H



One level of decomposition

Discovering new concepts



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

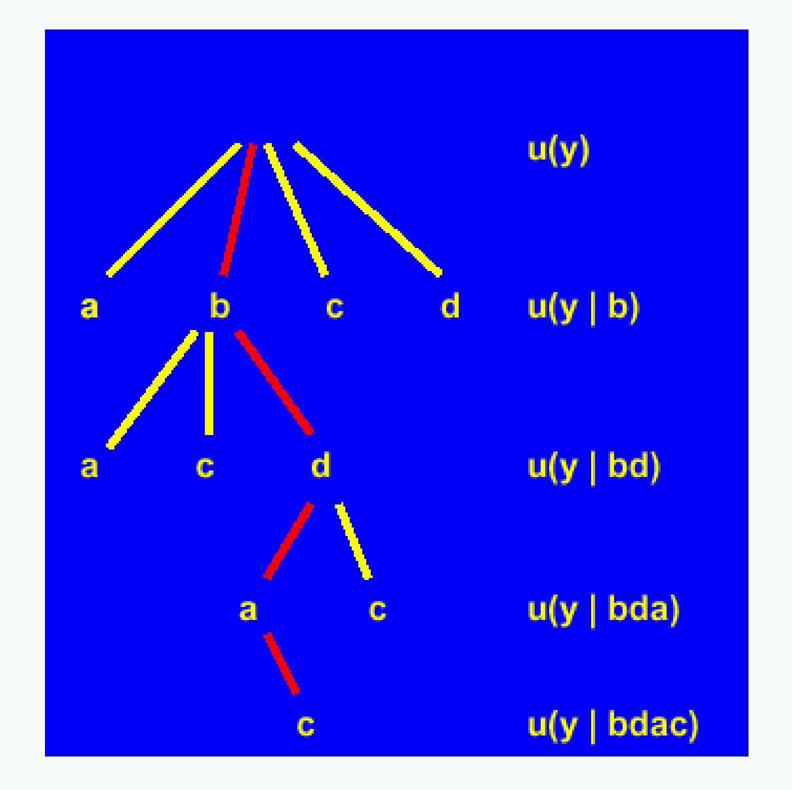
Variable ordering

- **Uncertainty (Shannon):**

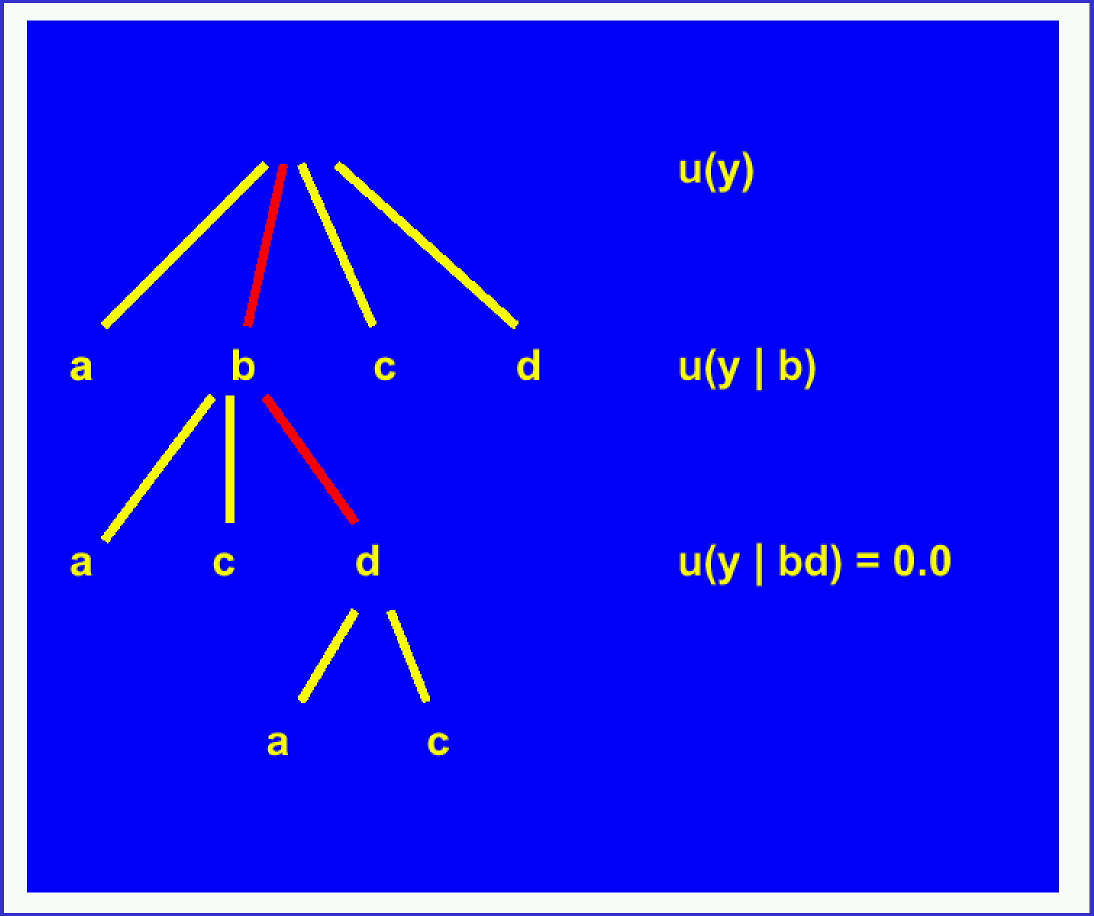
$$u(a) = -\sum_i p(a = a_i) \log_2 p(a = a_i)$$

- **Conditional Uncertainty (Shannon):**

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

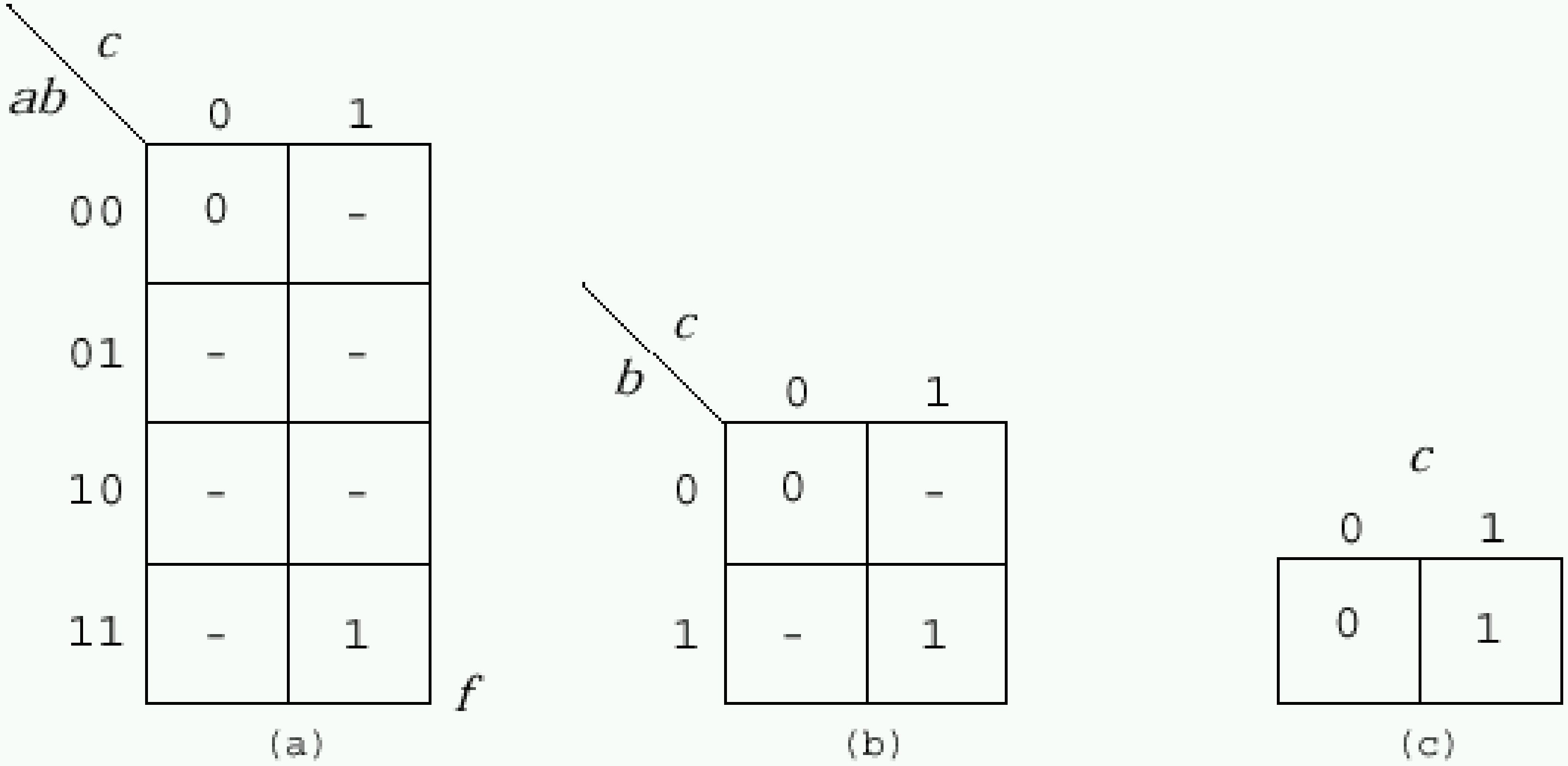


Vacuous variables removing



- 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, variable c is no longer inessential.



ab \ c	0	1
00	0	-
01	-	-
10	-	-
11	-	1

(a)

b \ c	0	1
0	0	-
1	-	1

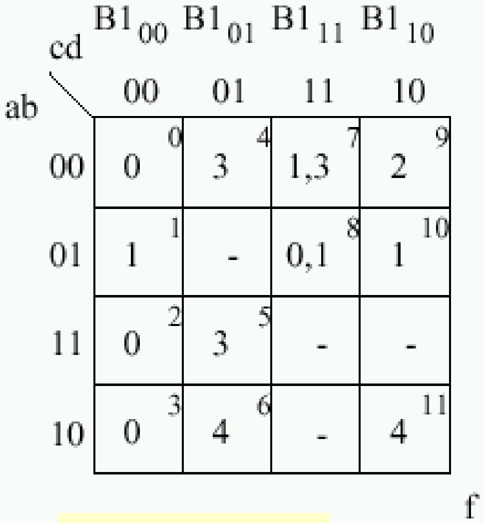
(b)

c	0	1
	0	1

(c)

Generalization of the Ashenhurst-Curtis decomposition model

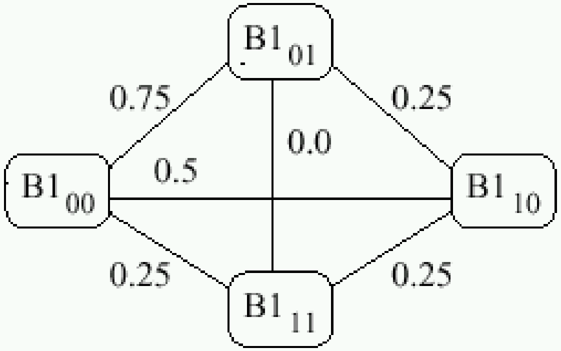
Compatibility graph construction for data with noise



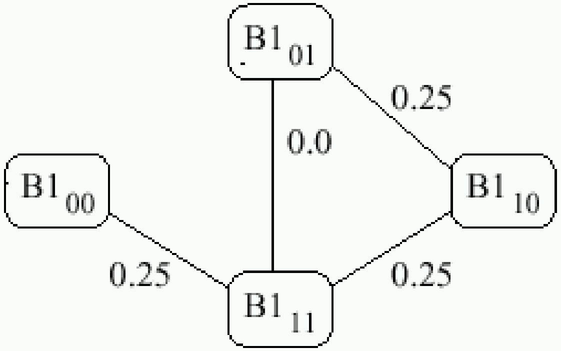
	$B1_{00}$	$B1_{01}$	$B1_{11}$	$B1_{10}$
ab \ cd	00	01	11	10
00	0 ^{0}	3 ^{4}	1,3 ^{7}	2 ^{9}
01	1 ^{1}	-	0,1 ^{8}	1 ^{10}
11	0 ^{2}	3 ^{5}	-	-
10	0 ^{3}	4 ^{6}	-	4 ^{11}

f

Kmap

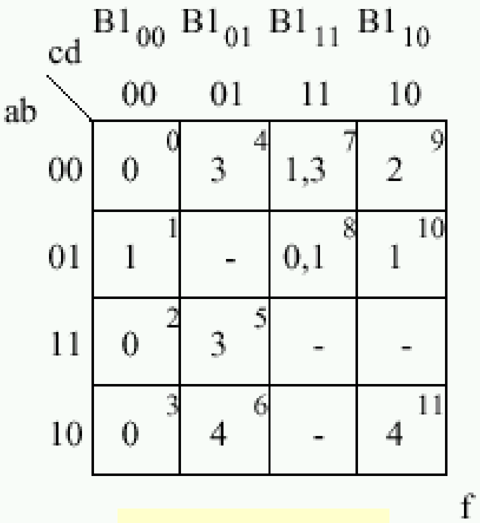


Compatibility Graph for Threshold 0.75



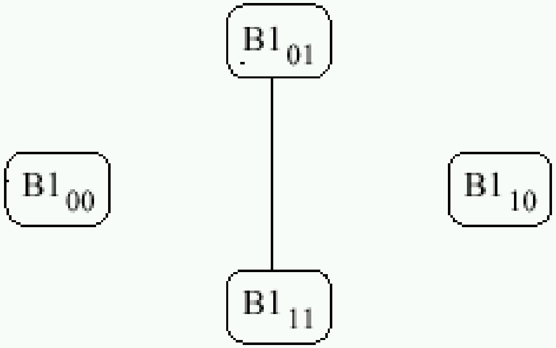
Compatibility Graph for Threshold 0.25

Compatibility graph for metric data

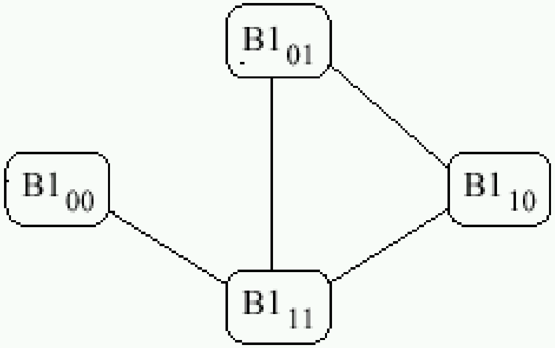


	B1₀₀	B1₀₁	B1₁₁	B1₁₀
ab \ cd	00	01	11	10
00	0 0	4 3	7 1,3	9 2
01	1 1	-	8 0,1	10 1
11	2 0	5 3	-	-
10	3 0	6 4	-	11 4

Kmap



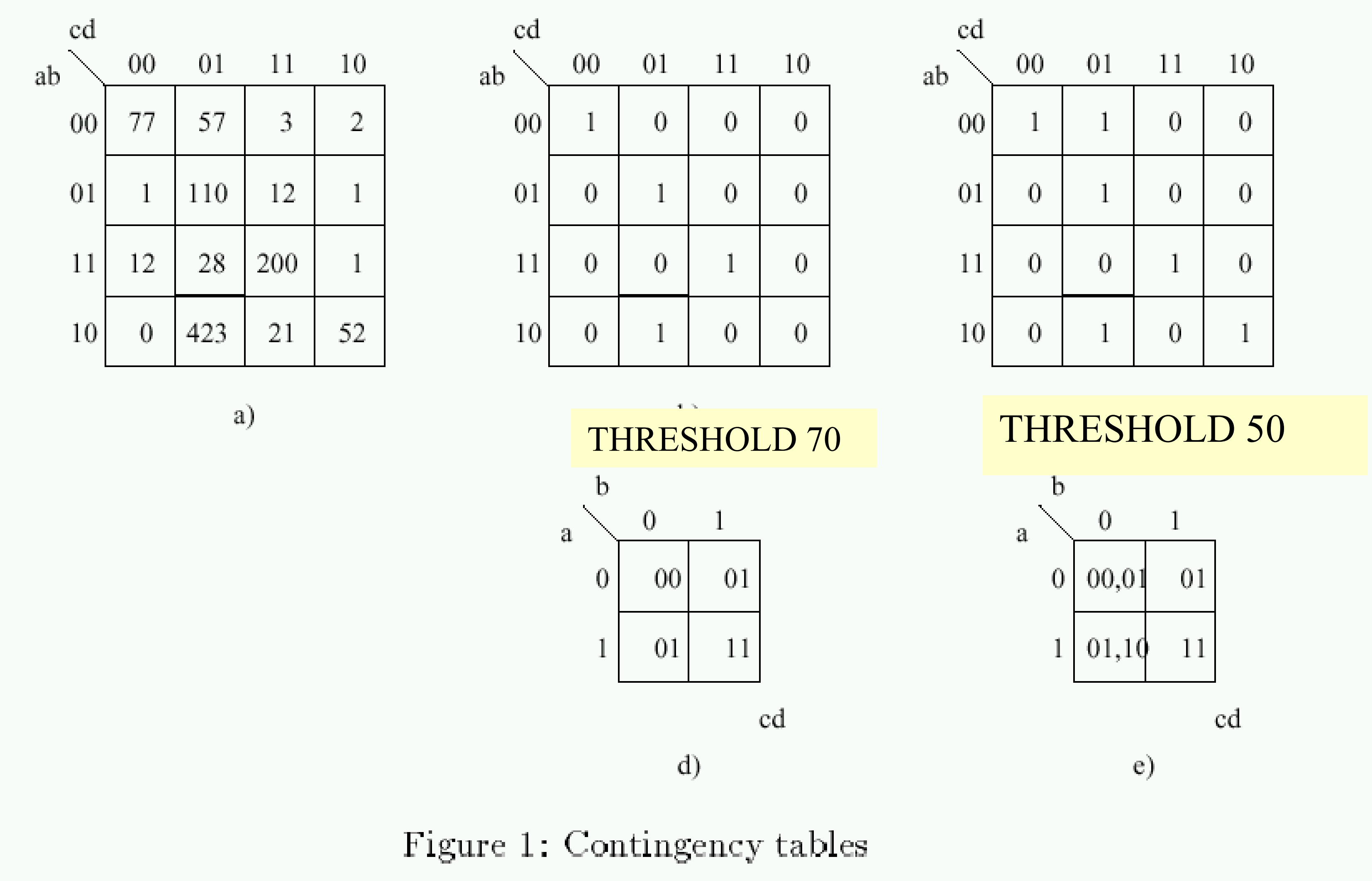
Compatibility Graph for nominal data



Compatibility Graph for metric data

Difference of 1

MV relations can be created from contingency tables



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

a)

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 70

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

THRESHOLD 50

a \ b	0	1
0	00	01
1	01	11

cd

d)

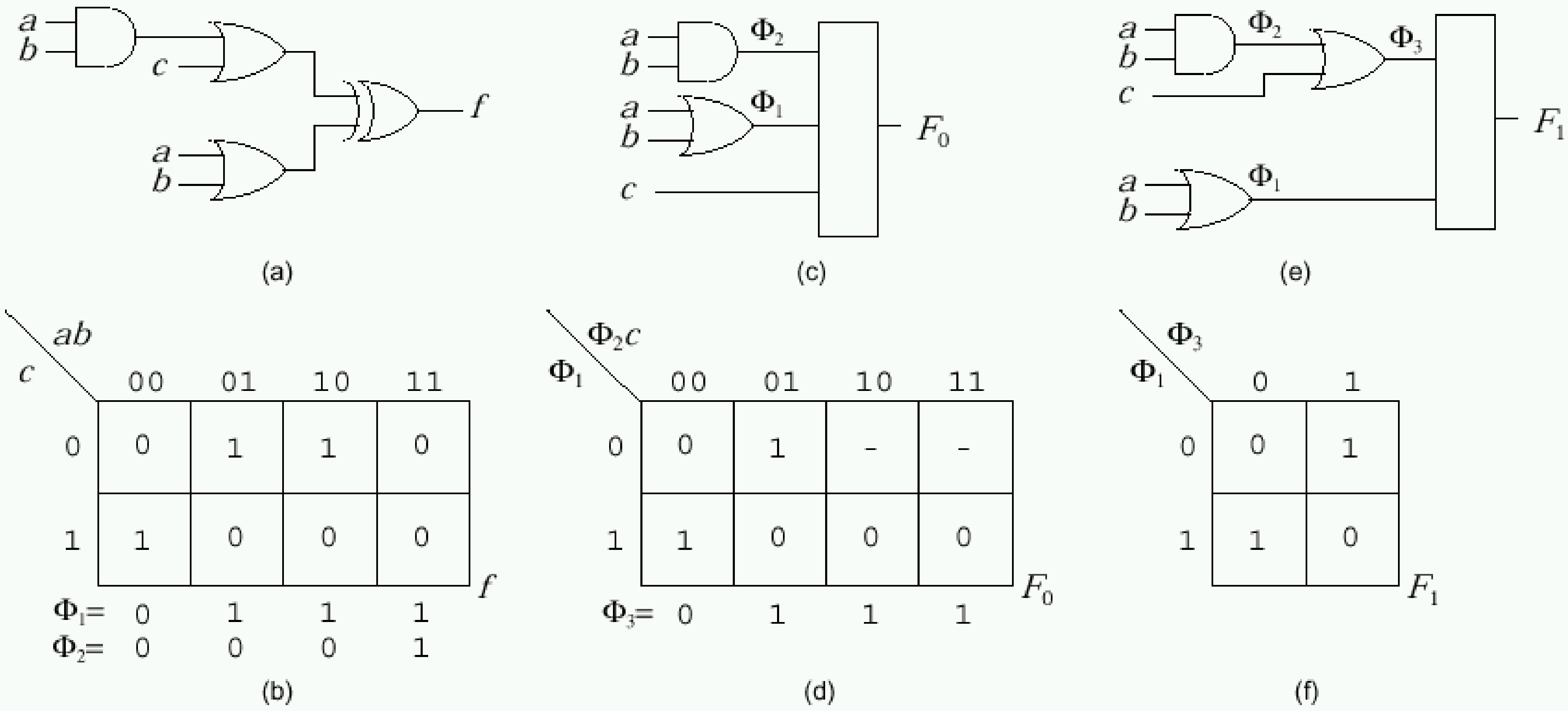
a \ b	0	1
0	00,01	01
1	01,10	11

cd

e)

Figure 1: Contingency tables

Example of decomposing a Curtis non-decomposable function.



ab c	00	01	10	11
0	0	1	1	0
1	1	0	0	0
$\Phi_1=$	0	1	1	1
$\Phi_2=$	0	0	0	1

f

(b)

$\Phi_2 c$ Φ_1	00	01	10	11
0	0	1	-	-
1	1	0	0	0
$\Phi_3=$	0	1	1	1

F_0

(d)

Φ_3 Φ_1	0	1
0	0	1
1	1	0

F_1

(f)

Evaluation of numerical results

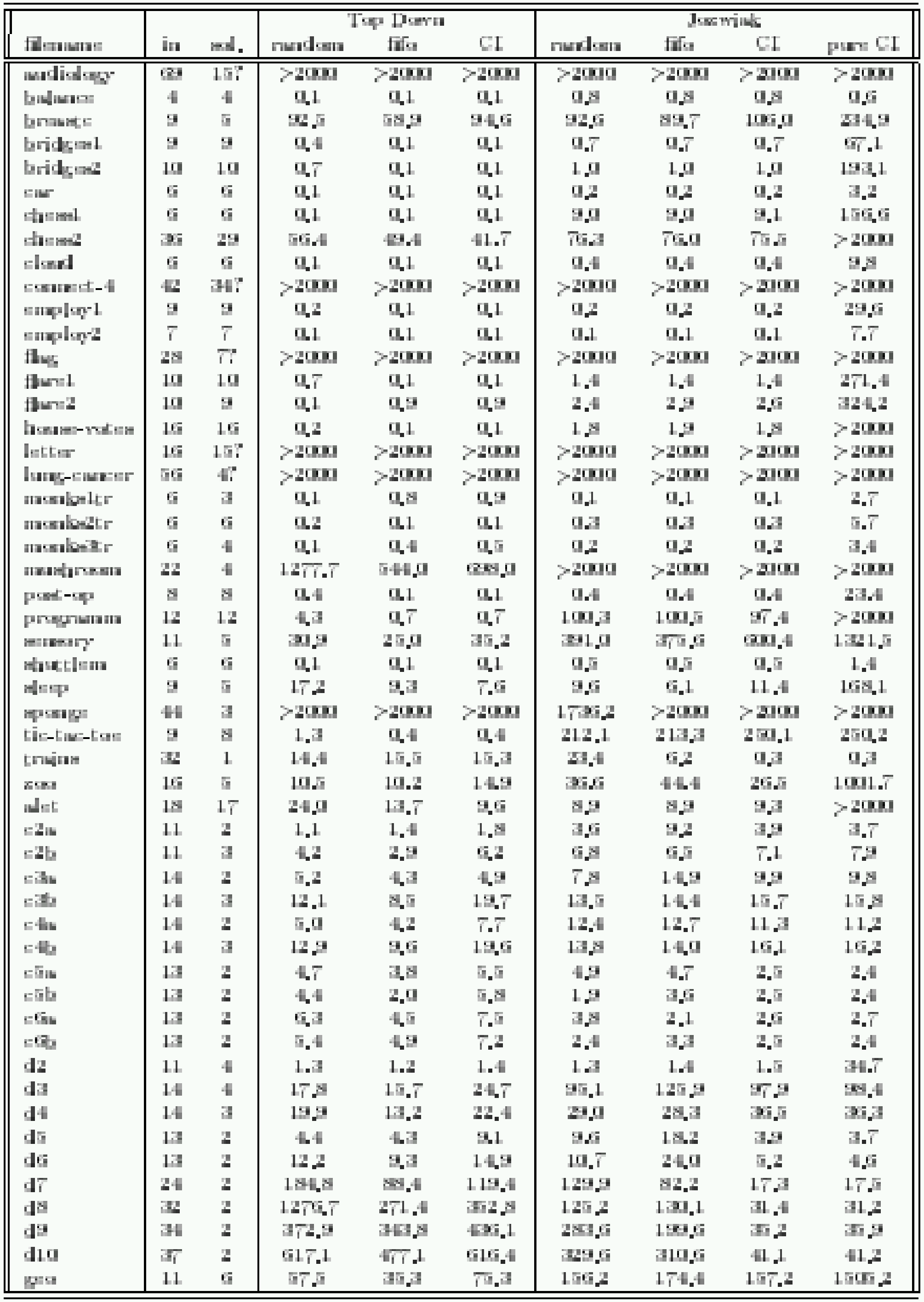
Decomposition of binary (MCNC) benchmarks

		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]

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)	236	11.0
9sym	9/1	640	984	400	224 (7)	160 (5)			104	26.4
con1	7/2	80	68	60					70	2.3
duke2	22/29	6516	2428	2200	3456 (108)				2896	11289.0
ex5p	8/63		3720	1560					2104	208.0
f5lm	8/8	372	392	240	256 (8)				177	10.1
misex1	8/7	472	208	224	256 (8)	354 (11)	304 (19)	288 (18)	229	8.6
misex2	25/18	548	464	436	768 (24)				392	1086.0
misex3	14/14	9816	4204	3028					1744	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%

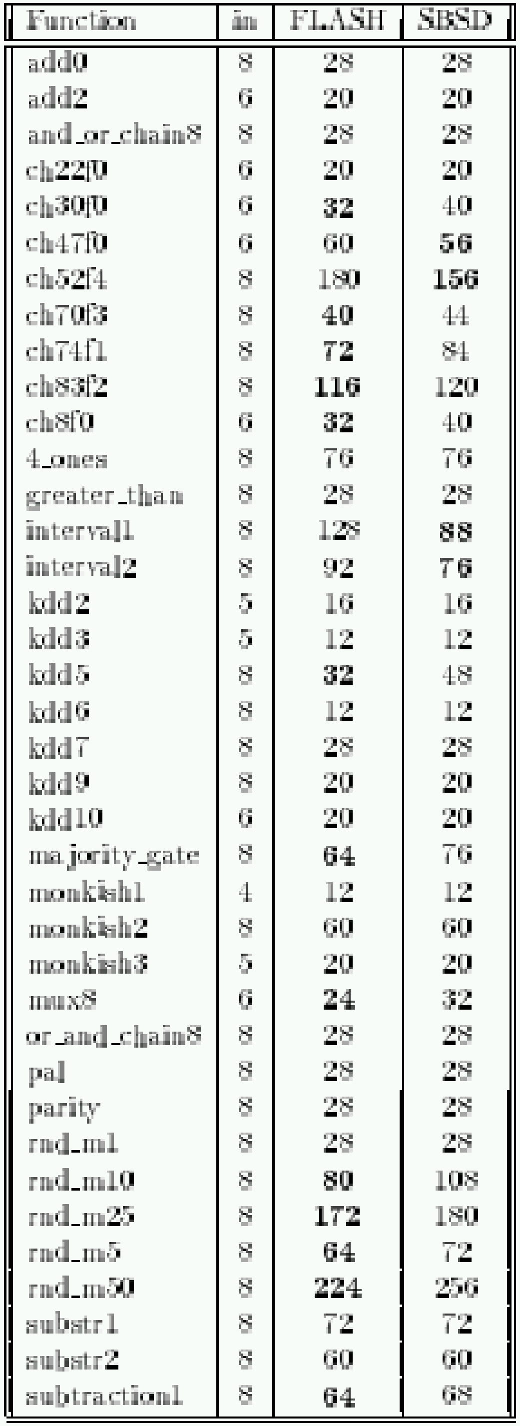
Table 3.2: MVDD and MBDD size comparisons.



filename	in	sol.	Top Down random	Top Down fifo	Top Down CI	Jozwiak random	Jozwiak fifo	Jozwiak CI	Jozwiak 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
breastc	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	688.0	>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	13.7	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	13.6	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	349.8	496.1	289.6	139.6	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.

SBSD comparison to FLASH on Wright Lab benchmark functions.



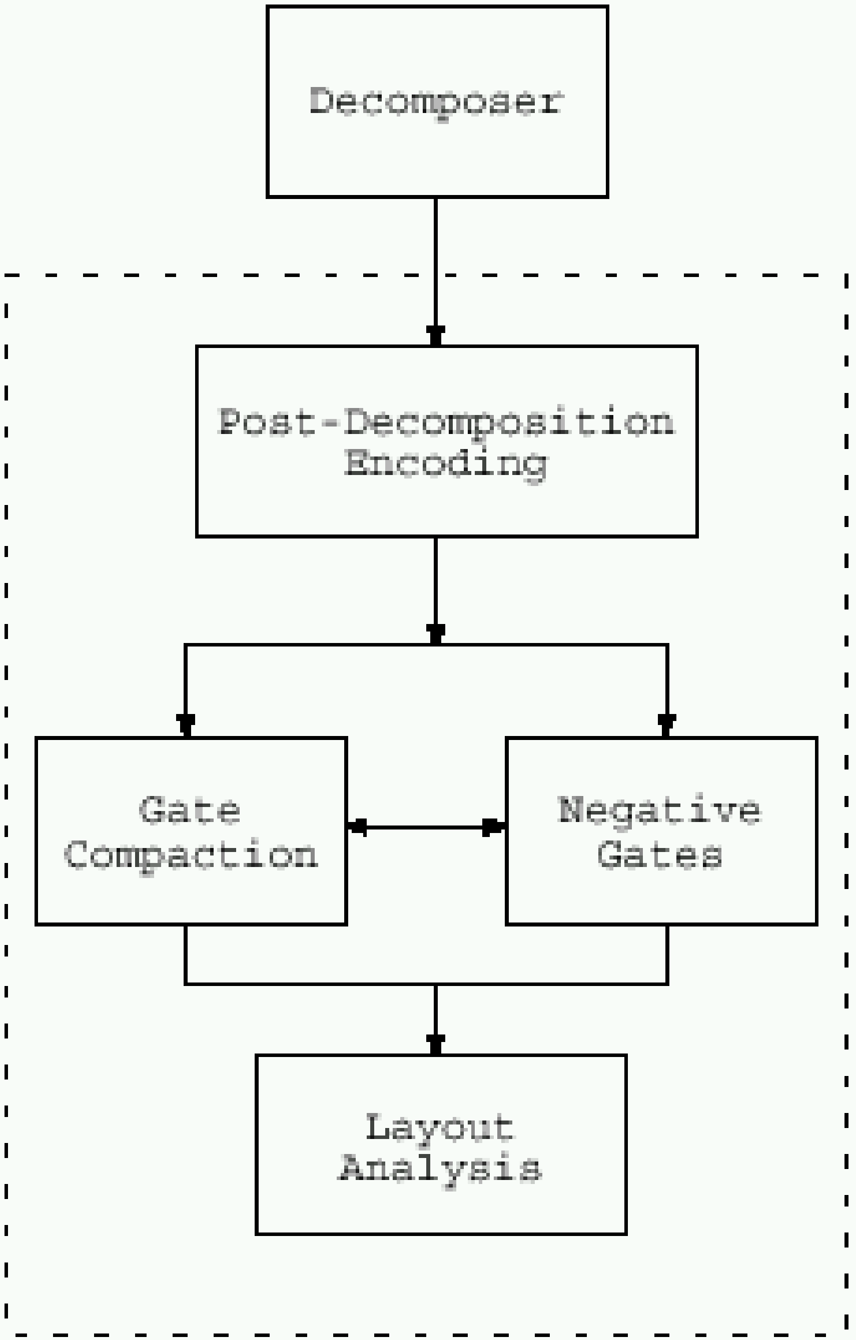
Function	in	FLASH	SBSD
add0	8	28	28
add2	6	20	20
and_or_chain8	8	28	28
ch22f0	6	20	20
ch30f0	6	**32**	40
ch47f0	6	60	**56**
ch52f4	8	180	**156**
ch70f3	8	**40**	44
ch74f1	8	**72**	84
ch83f2	8	**116**	120
ch8f0	6	**32**	40
4_ones	8	76	76
greater_than	8	28	28
interval1	8	128	**88**
interval2	8	92	**76**
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

APPLICATIONS

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

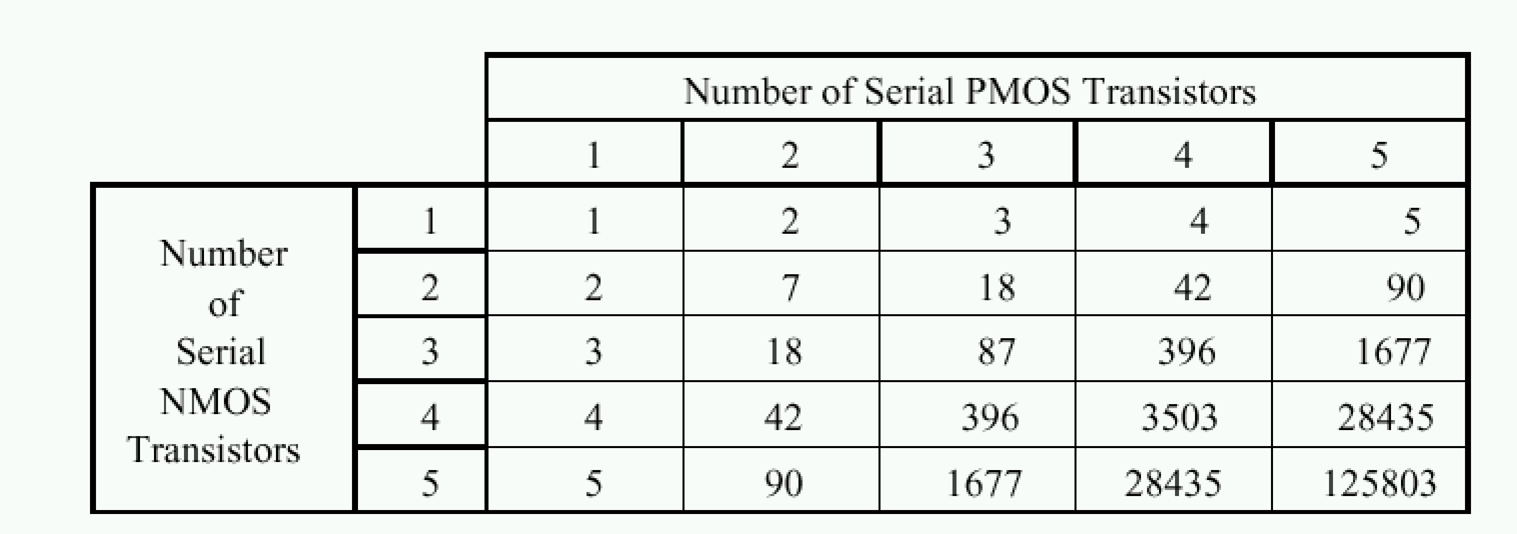
Example of a application

VLSI Layout



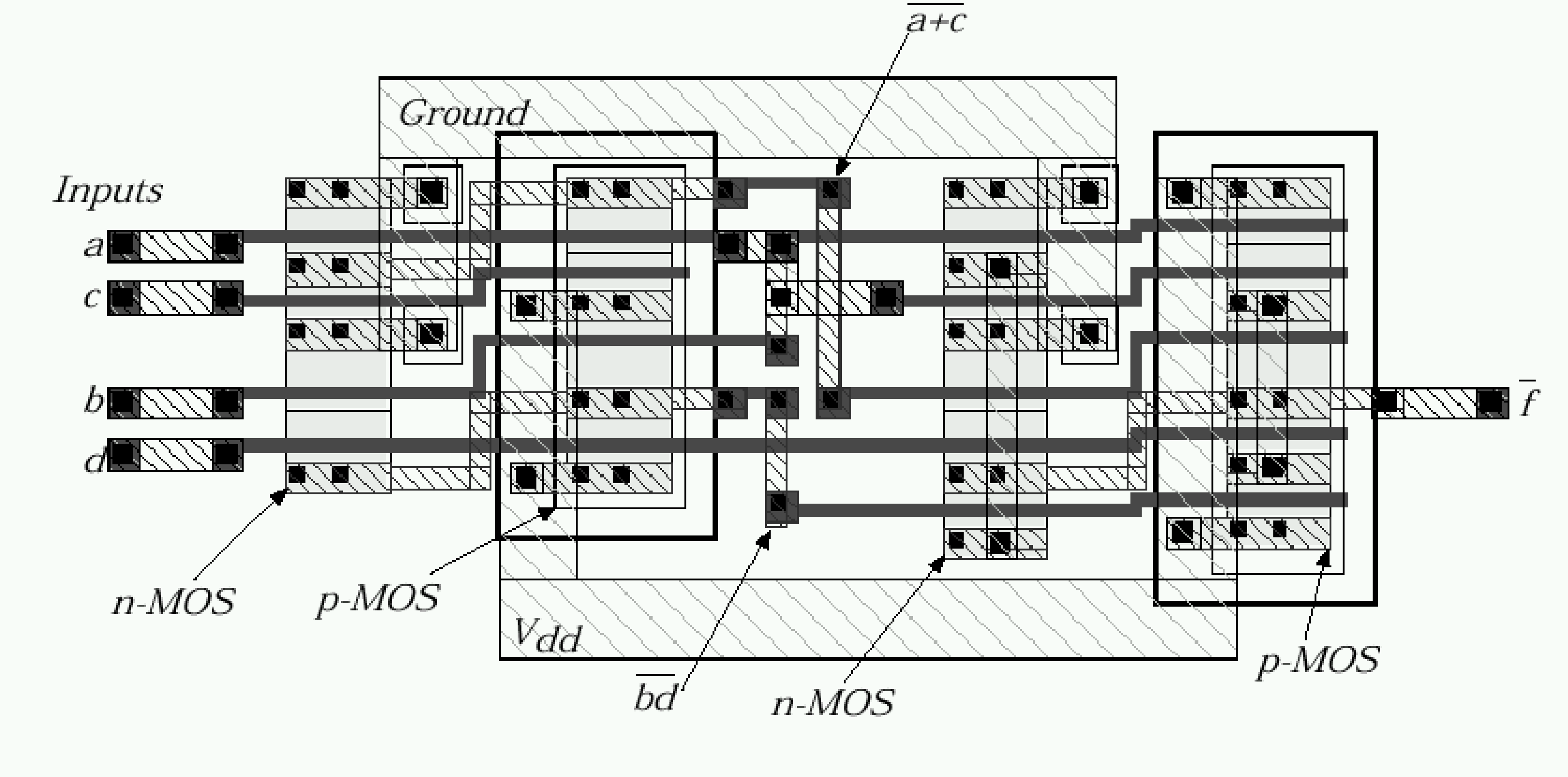
Layout decomposition block diagram.

Number of complex gates with limited serial transistors

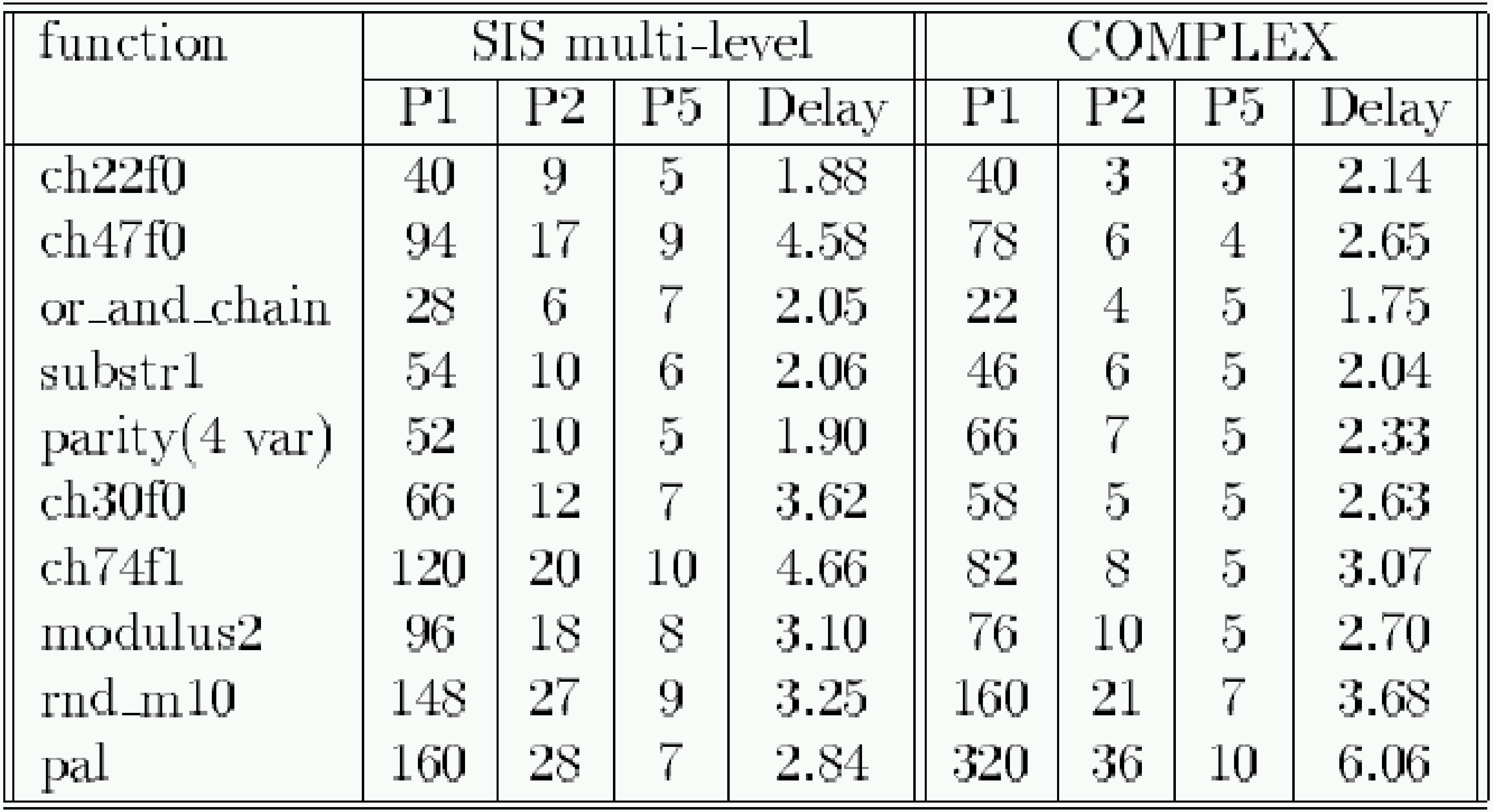


		Number of Serial PMOS Transistors				
		1	2	3	4	5
Number of Serial NMOS Transistors	1	1	2	3	4	5
	2	2	7	18	42	90
	3	3	18	87	396	1677
	4	4	42	396	3503	28435
	5	5	90	1677	28435	125803

VLSI layout of $\overline{f} = d\overline{(a+c)} + (b+c)\overline{(bd)}$.

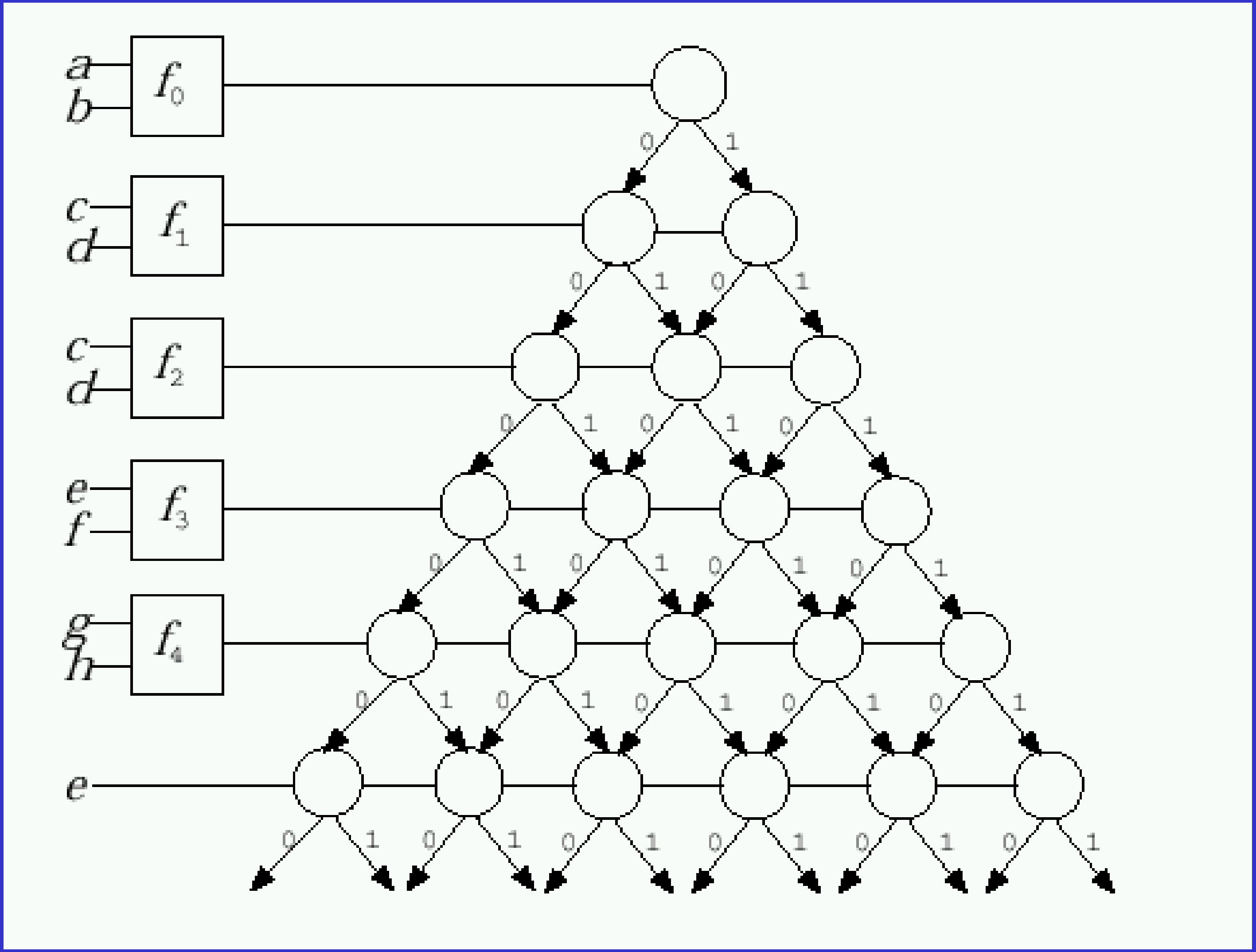


Comparison of SIS and COMPLEX



function	SIS multi-level				COMPLEX			
	P1	P2	P5	Delay	P1	P2	P5	Delay
ch22f0	40	9	5	1.88	40	3	3	2.14
ch47f0	94	17	9	4.58	78	6	4	2.65
or_and_chain	28	6	7	2.05	22	4	5	1.75
substr1	54	10	6	2.06	46	6	5	2.04
parity(4 var)	52	10	5	1.90	66	7	5	2.33
ch30f0	66	12	7	3.62	58	5	5	2.63
ch74f1	120	20	10	4.66	82	8	5	3.07
modulus2	96	18	8	3.10	76	10	5	2.70
rnd_m10	148	27	9	3.25	160	21	7	3.68
pal	160	28	7	2.84	320	36	10	6.06

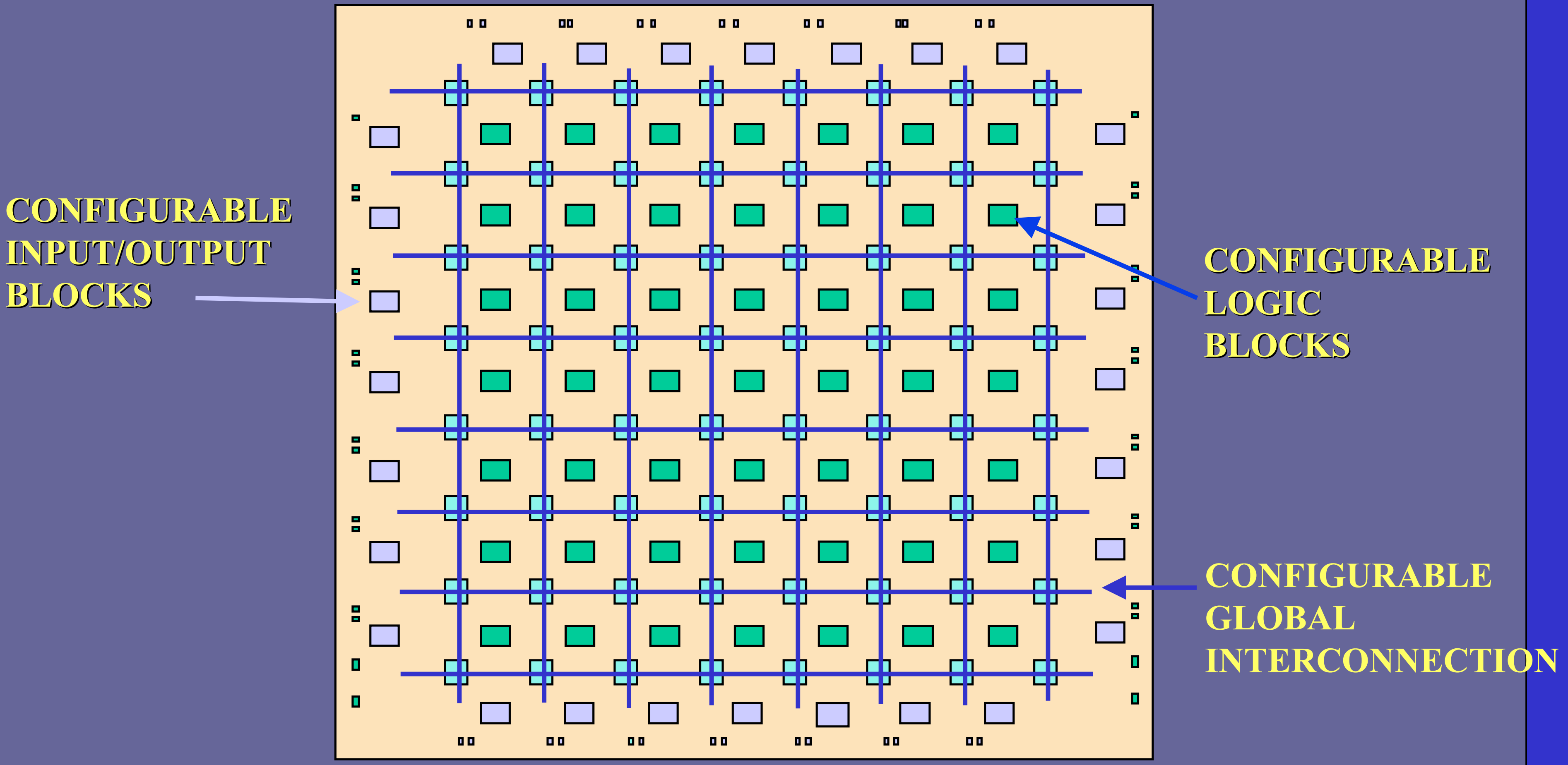
Example of decomposition based synthesis for lattice diagrams.



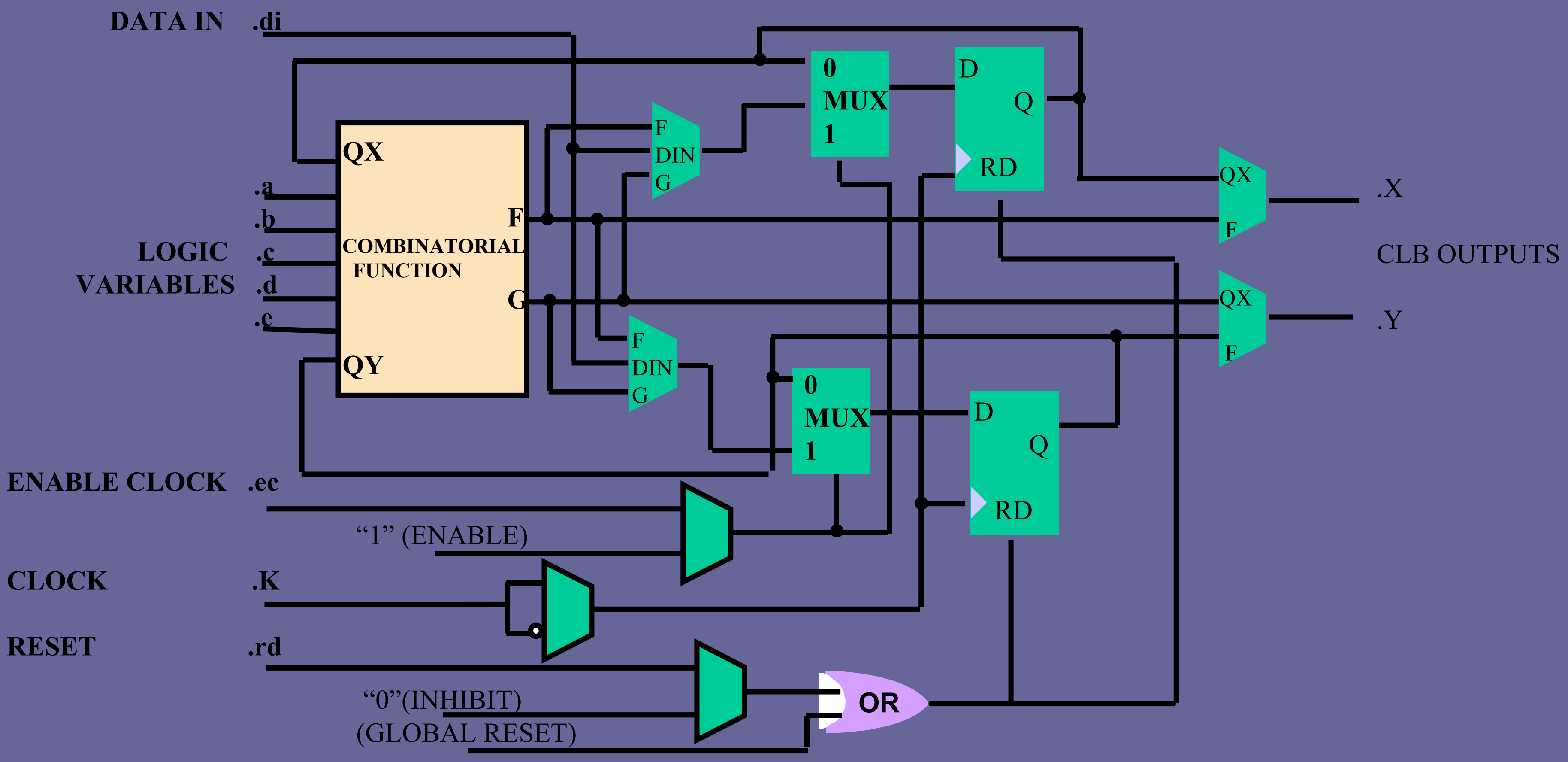
Example of a application

Synthesis for FPGAs

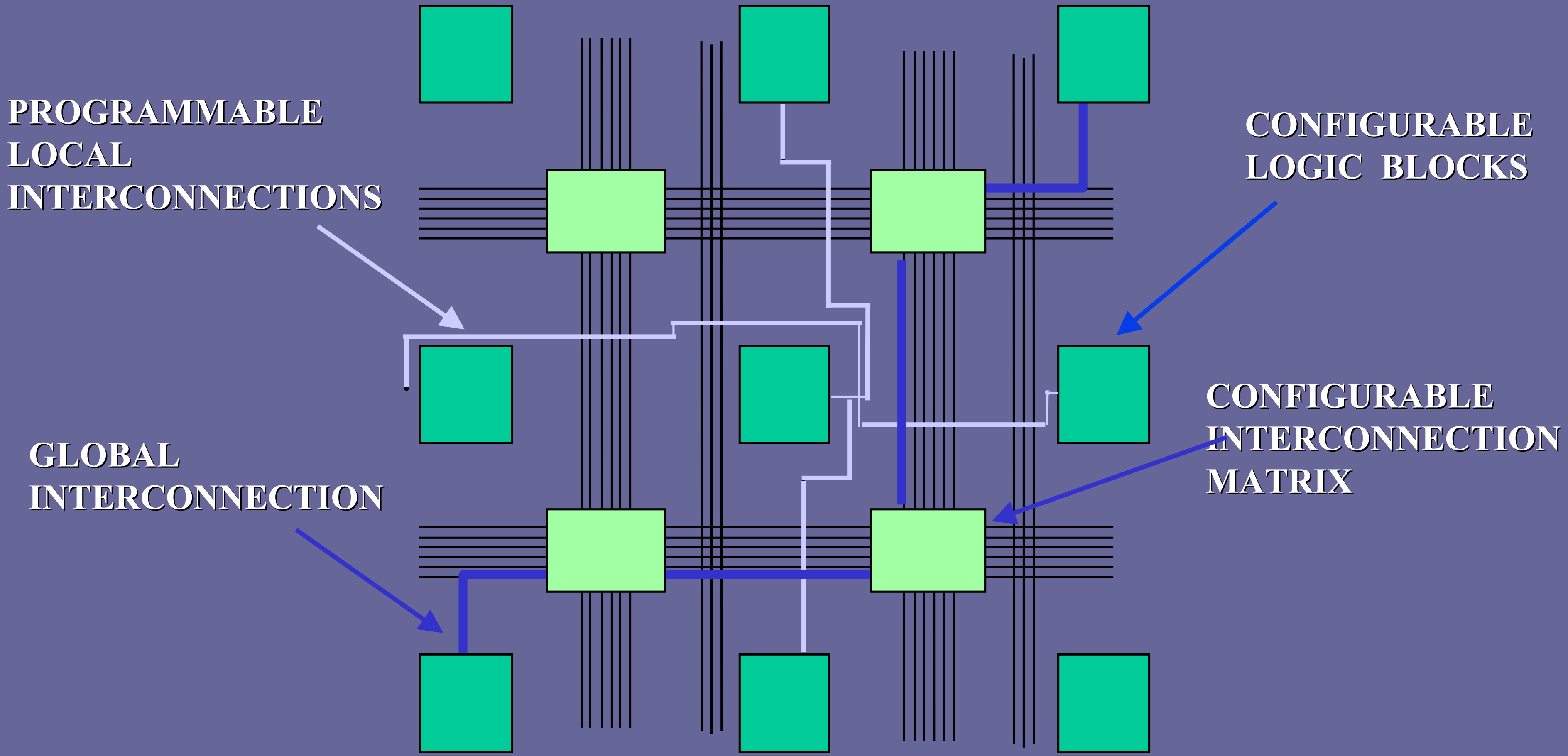
XILINX Field Programmable Gate Array



Configurable Logic Block



Interconnections

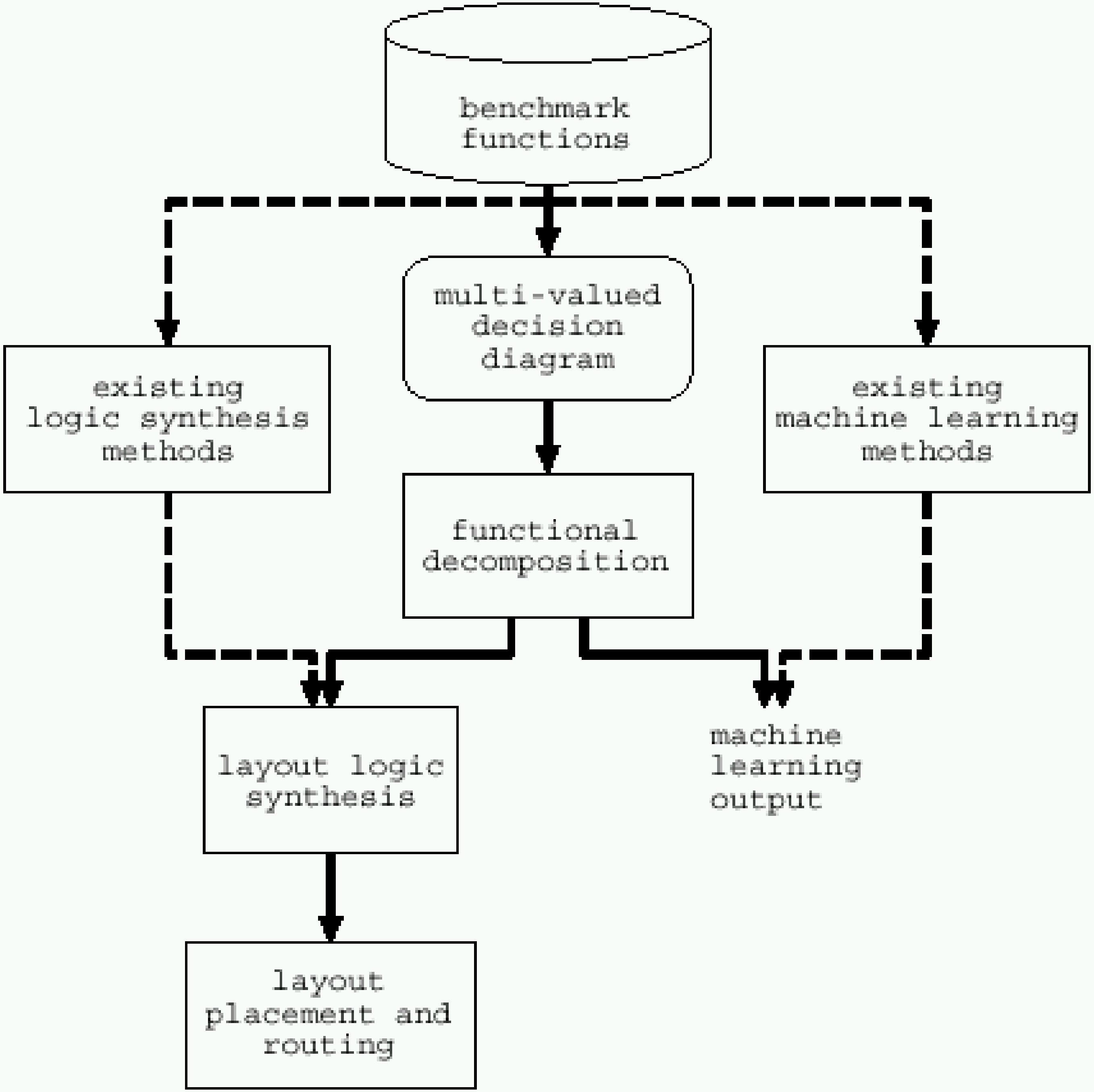


PROGRAMMABLE
LOCAL
INTERCONNECTIONS

CONFIGURABLE
LOGIC BLOCKS

CONFIGURABLE
INTERCONNECTION
MATRIX

GLOBAL
INTERCONNECTION

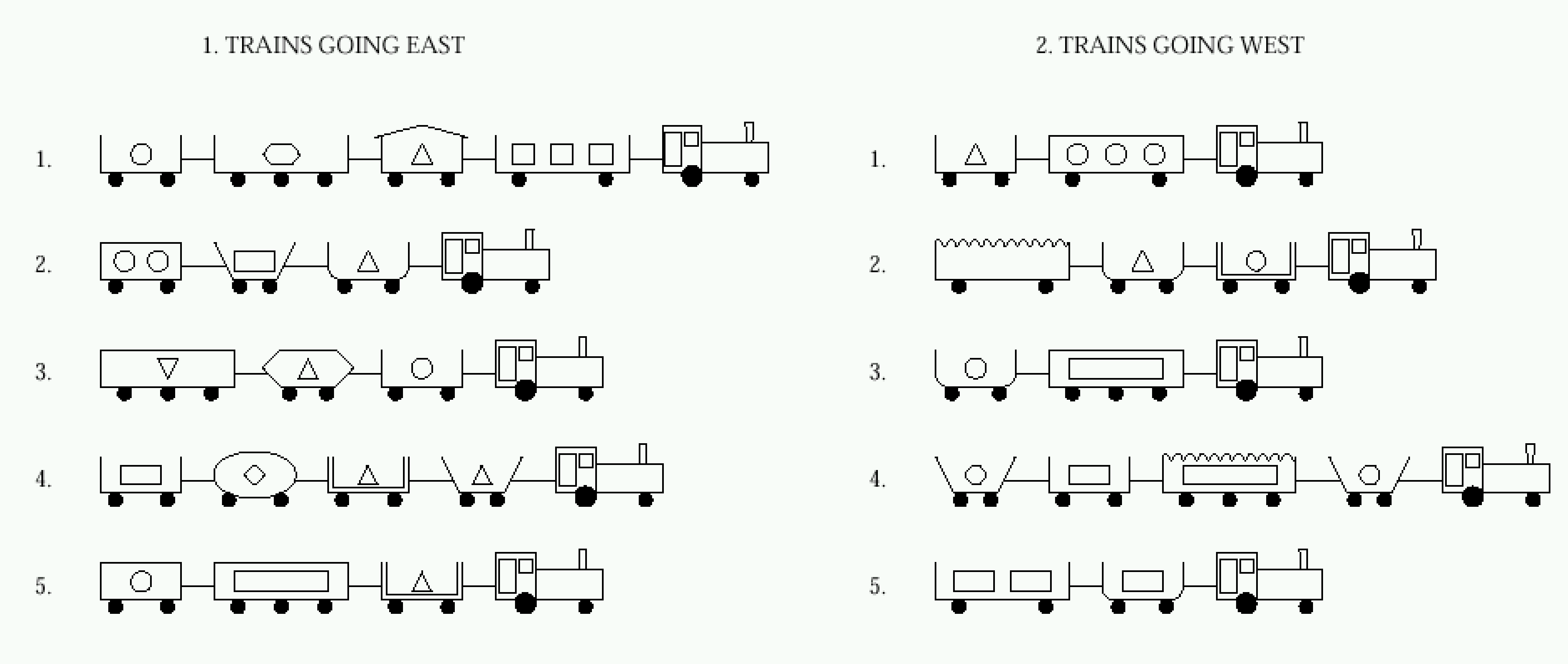


complete decomposition system.

Example of a application

Knowledge discovery in data with no error

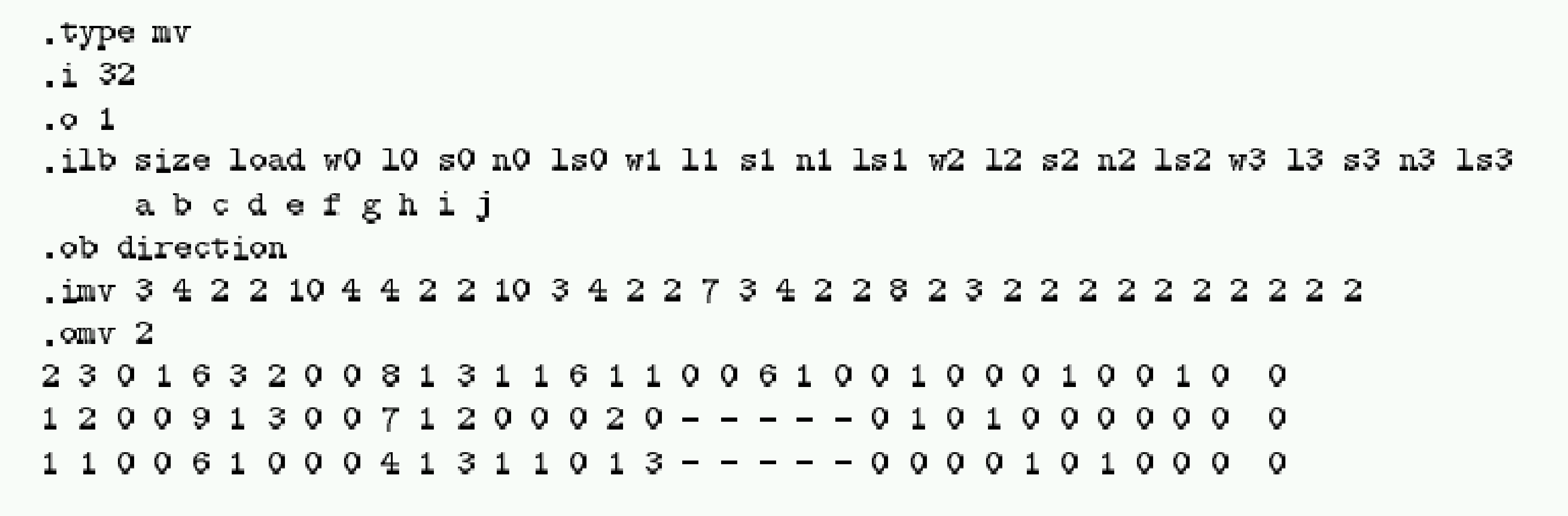
Michalski's Trains



Michalski's Trains

- Multiple-valued functions.
- There are 10 trains, five going East, five going West, and the problem is to find 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)

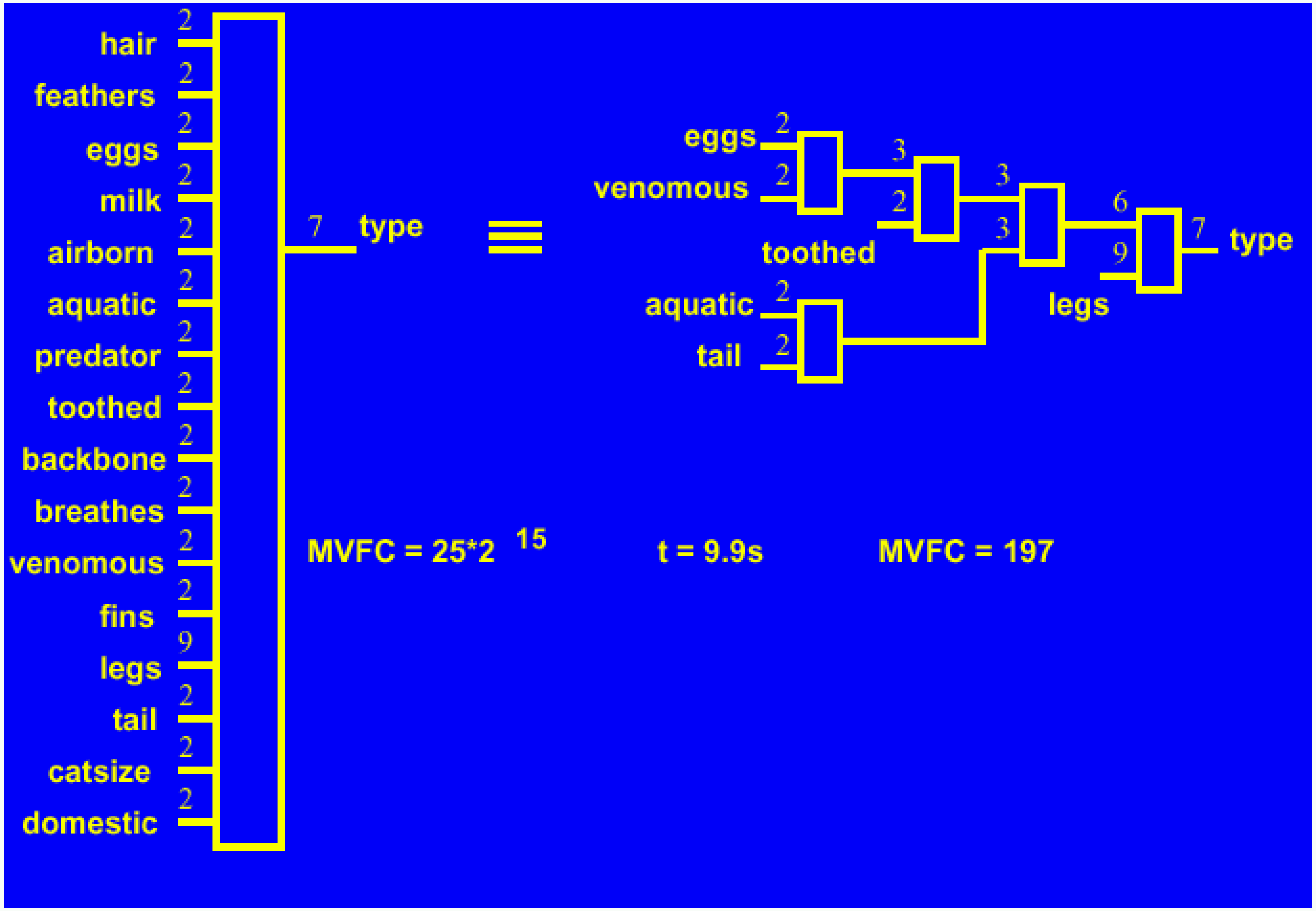
Variables 23-32: 10 Boolean attributes describing whether 2 types of loads are on adjacent cars of the train

`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

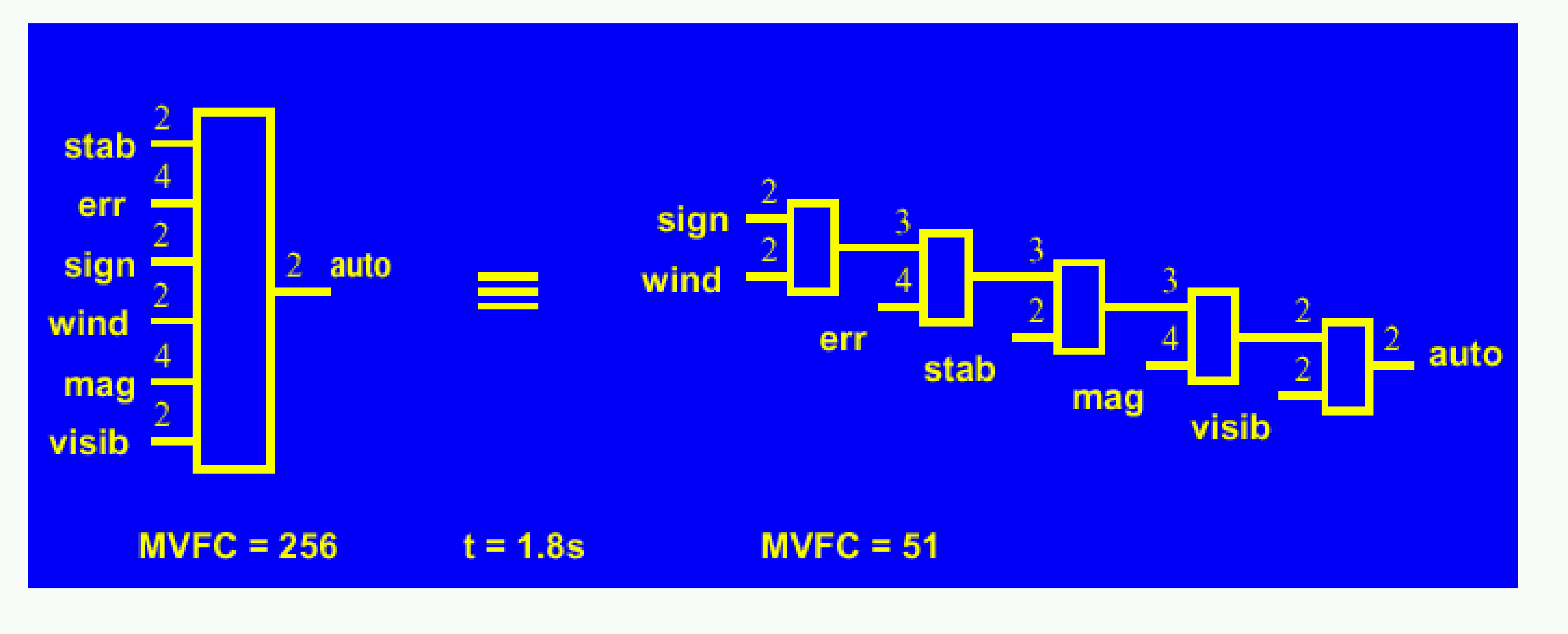


MVFC = 25*2 ^{15}

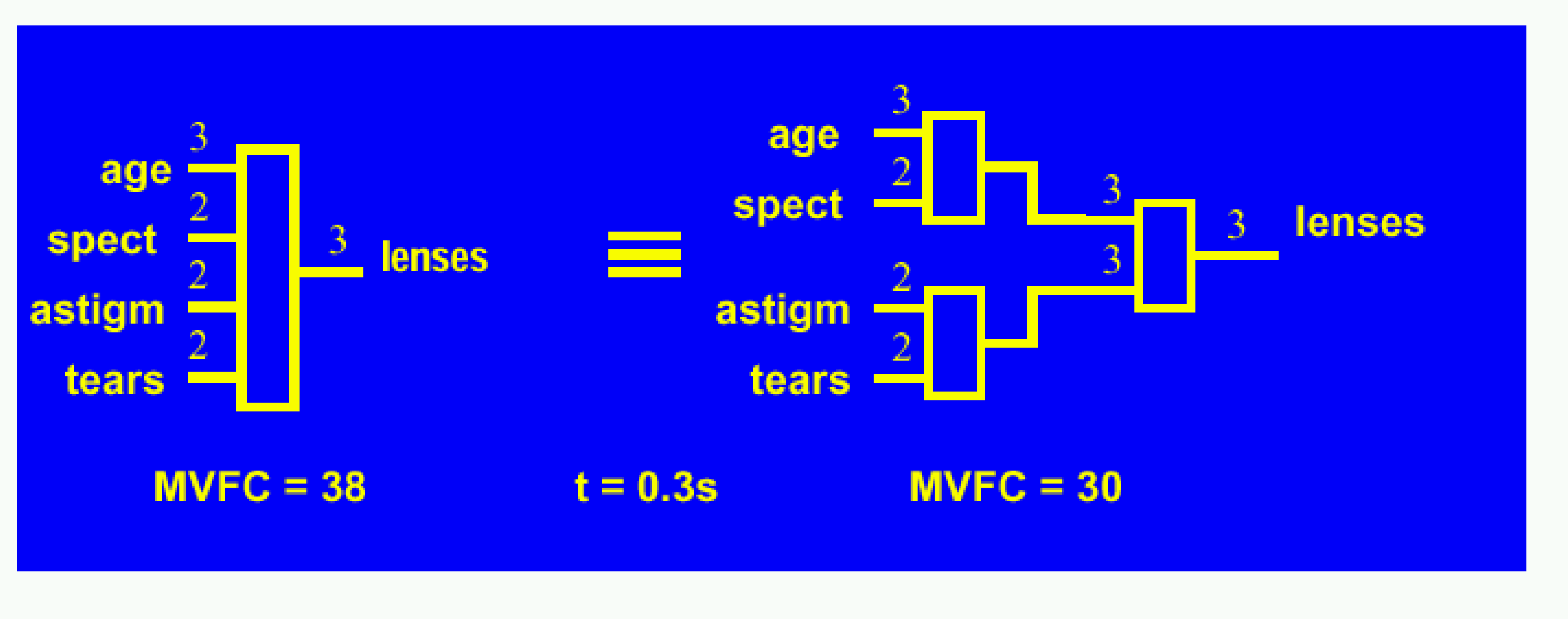
t = 9.9s

MVFC = 197

MV benchmarks: shuttle



MV benchmarks: lenses



Example of a application

Medical data bases with error

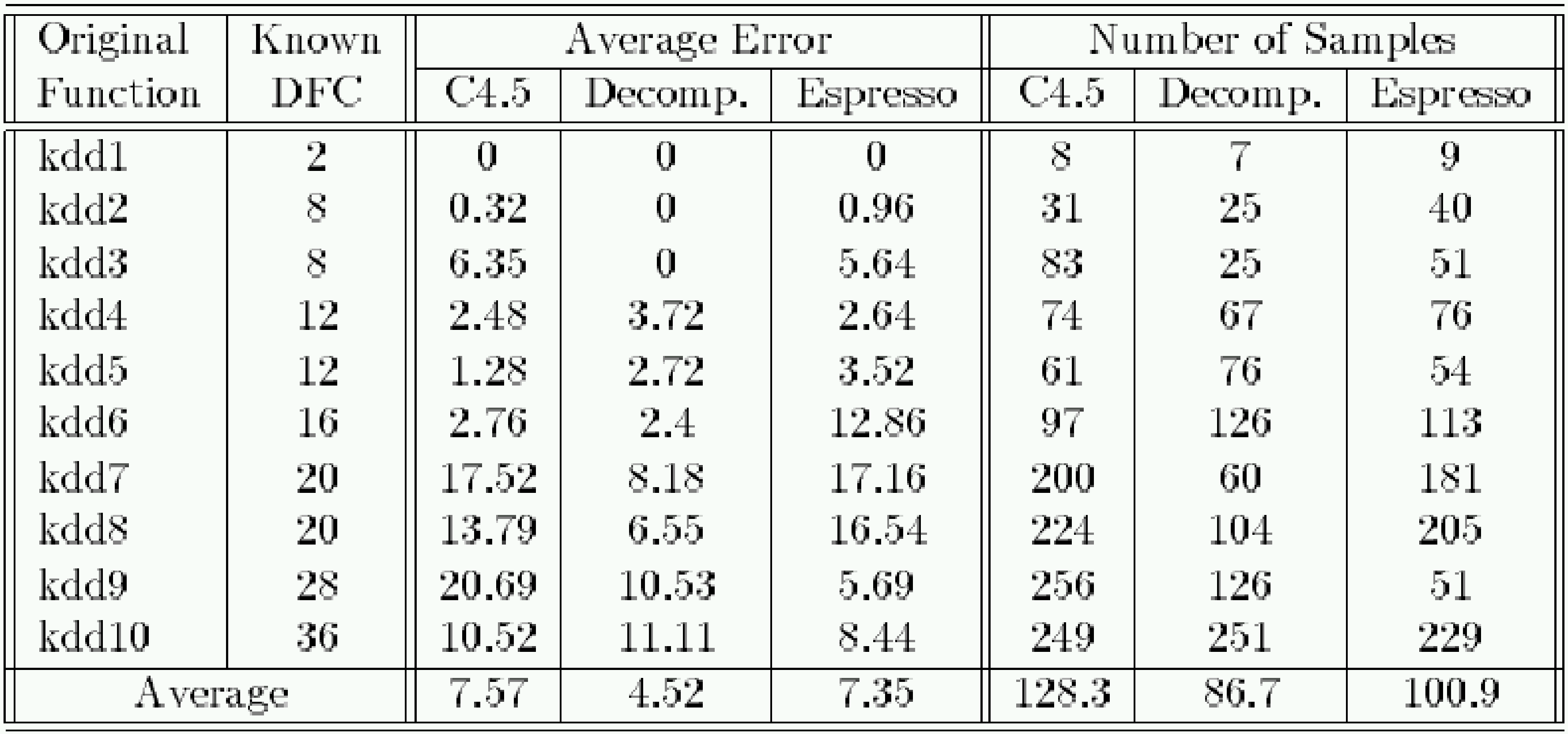
Evaluation of results for learning

- 1. Learning Error

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

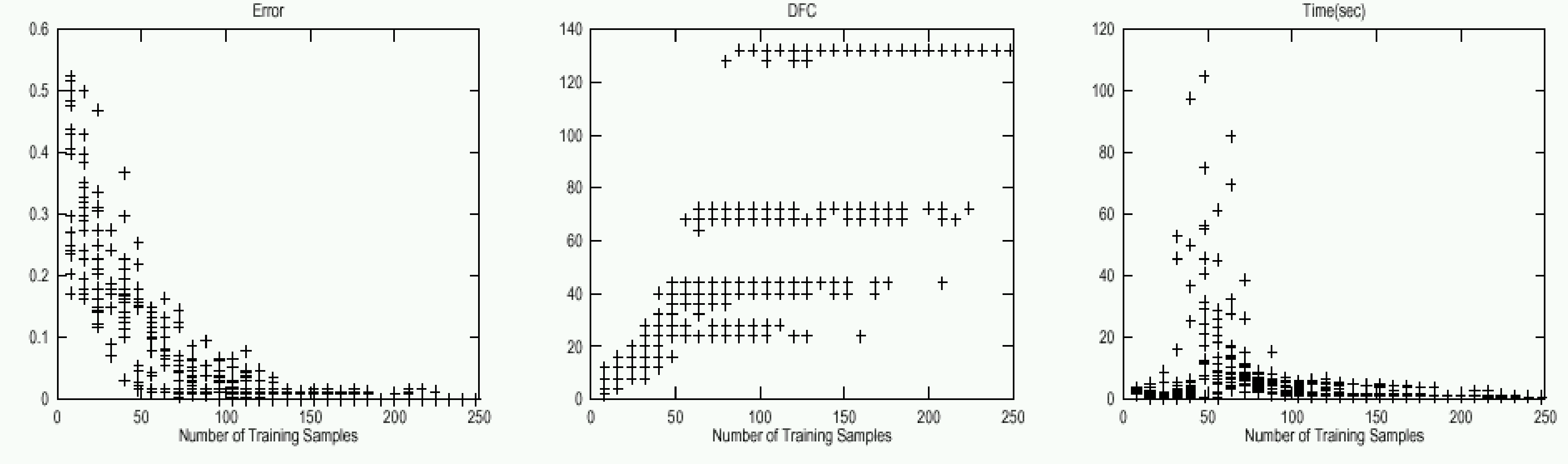
- 2. Occam Razor , complexity

A machine learning approach versus several logic synthesis approaches

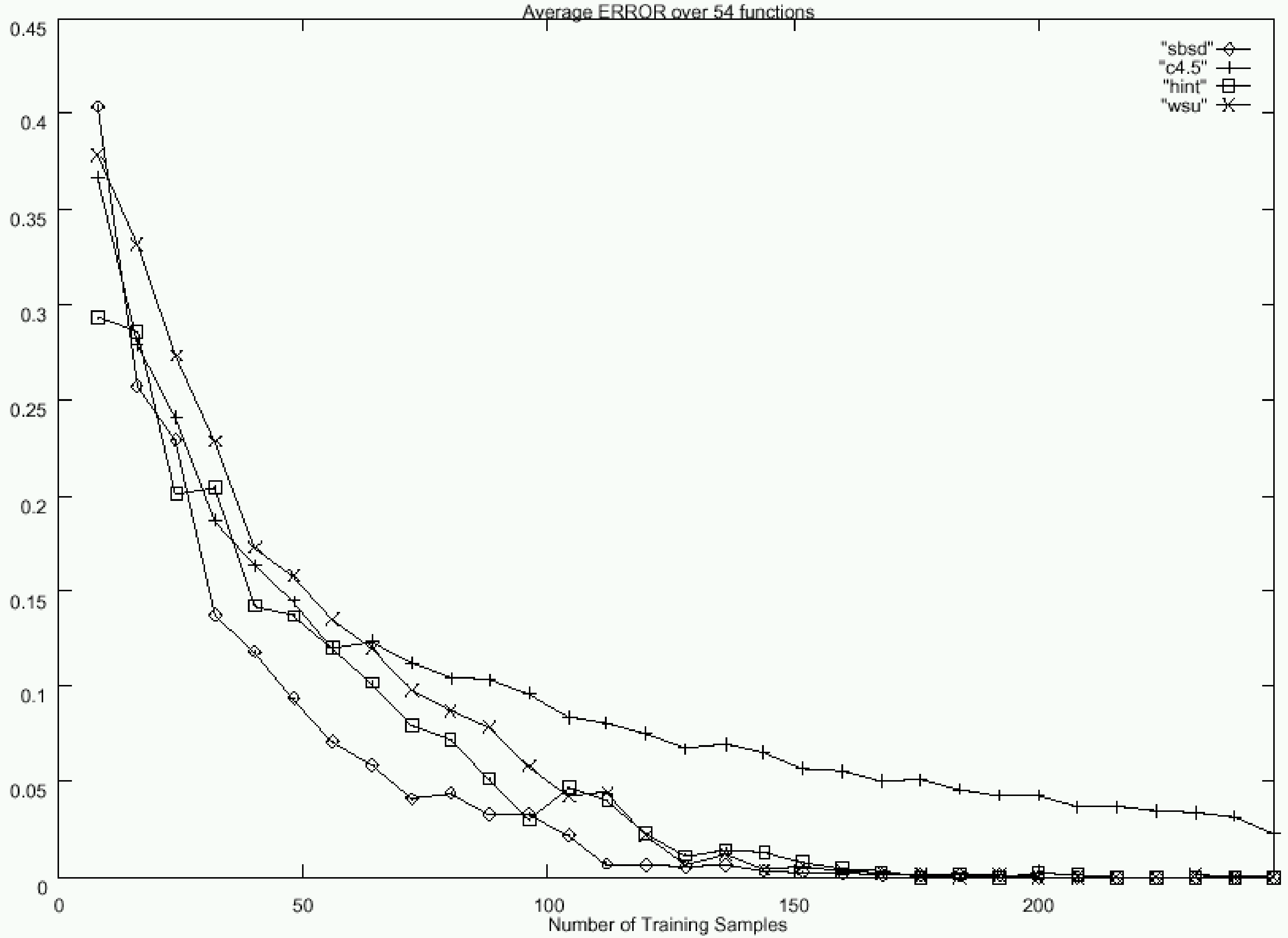


Original Function	Known DFC	Average Error			Number of Samples		
		C4.5	Decomp.	Espresso	C4.5	Decomp.	Espresso
kdd1	2	0	0	0	8	7	9
kdd2	8	0.32	0	0.96	31	25	40
kdd3	8	6.35	0	5.64	83	25	51
kdd4	12	2.48	3.72	2.64	74	67	76
kdd5	12	1.28	2.72	3.52	61	76	54
kdd6	16	2.76	2.4	12.86	97	126	113
kdd7	20	17.52	8.18	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

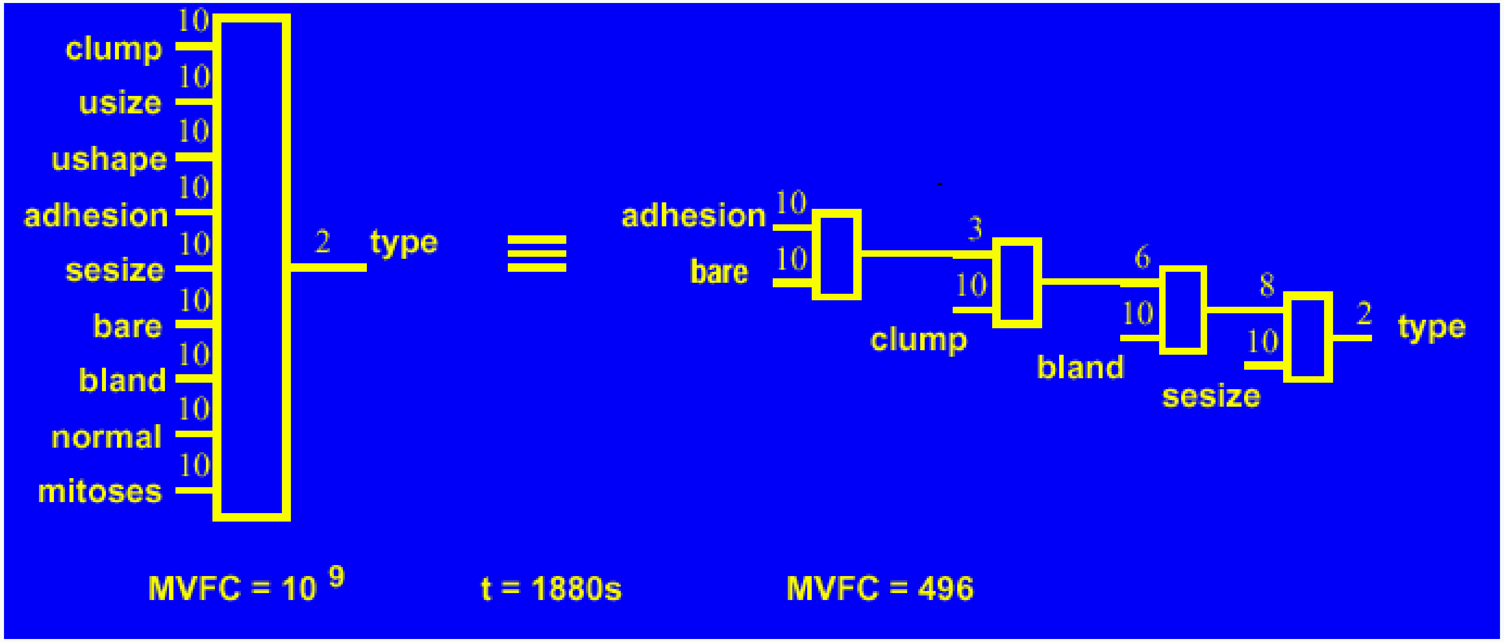
Finding the error, DFC, and time of the decomposer on the benchmark **kdd5**.



The average error over 54 benchmark functions.



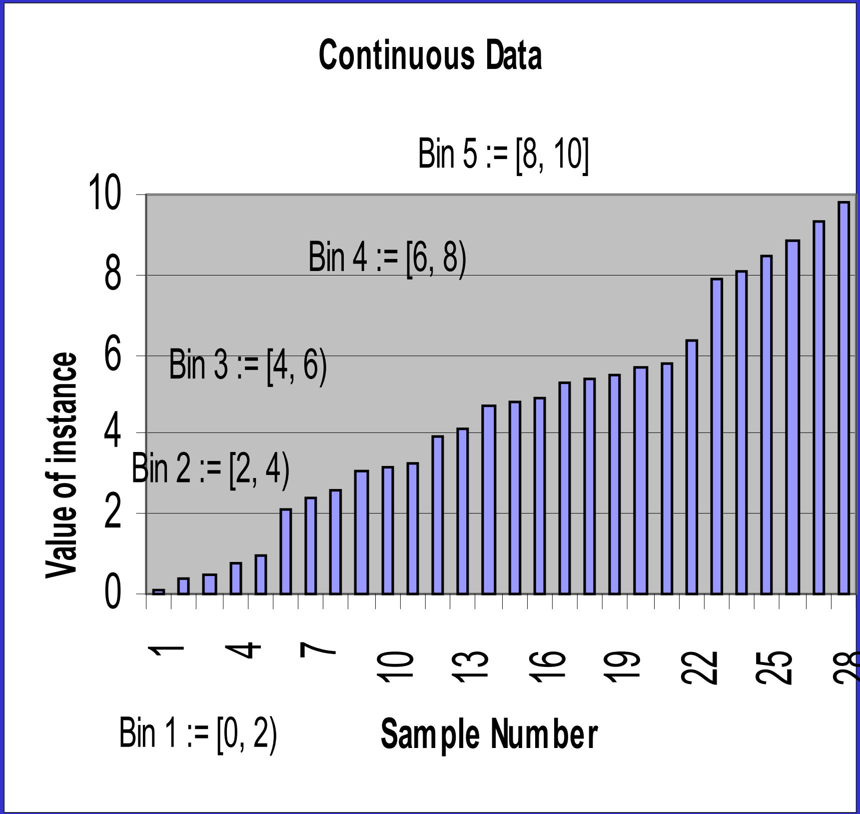
MV benchmarks: breastc

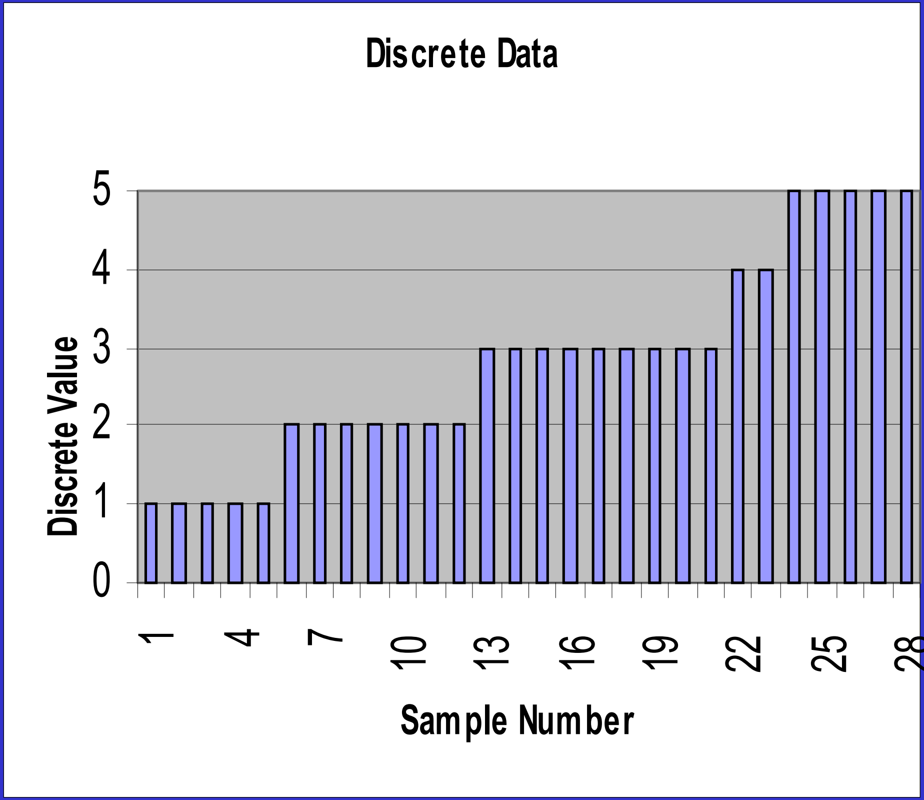


Example of a application

Data mining system for epidemiologists

Binning Strategy #1: Linear Mapping





7/22/01

Epidemiological Survey

Race:

_____ (W) White

_____ (B) Black

_____ (O) Other

Did you [Name of child] have contact with or change any diapers while at Battleground State Park?

_____ (1) YES _____ (2) NO _____ (9) DK

Estimate the amount of time you [Name of child] spent in the water (total time):

> 2 hours	____ (3)
15 minutes – 2 hours	____ (2)
< 15 minutes	____ (1)

How serious was your child's illness?

____ (1) No illness ____ (2) diarrhea but no fever ____ (3) diarrhea and fever ____ (9) DK

Survey Encoding

Input Variable ‘a’

White encodes to ‘0’
Black encodes to ‘1’
Other encodes to ‘2’

Input Variable ‘b’

DK encodes to ‘2’
NO encodes to ‘1’
YES encodes to ‘0’

Input Variable ‘c’

2 hr < encodes to ‘2’
[.25, 2) hr encodes to ‘1‘
< .25 hr encodes to ‘0’

Output Variable ‘z’

Don’t Know encodes to ‘3’
Diarrhea and fever encodes to ‘2’
Diarrhea but no fever encodes to ‘1’
No illness encodes to ‘0’

Survey Data: Sample 0

Race:

_____ (W) White
__X__ (B) Black
_____ (O) Other

Did you [Name of child] have contact with or change any diapers while at Battleground State Park?

__X__ (1) YES _____ (2) NO _____ (9) DK

Estimate the amount of time you [Name of child] spent in the water (total time):

> 2 hours __X__ (3)
15 minutes – 2 hours ____ (2)
< 15 minutes ____ (1)

How serious was your child’s illness?

____ (1) No illness ____ (2) diarrhea but no fever __X__ (3) diarrhea and fever ____ (9) DK

Encoded Survey Data: Sample 0

Sample #	a	b	c	f
0	1	0	2	2

Ten Encoded Surveys

Sample #	a	b	c	z
0	1	0	2	2
1	2	1	2	0
2	2	2	1	3
3	0	2	1	1
4	2	1	2	0
5	2	2	1	2
6	0	2	1	0
7	0	2	0	1
8	1	1	2	2
9	1	1	1	0

Multi-valued Relation Represented Tabular Form

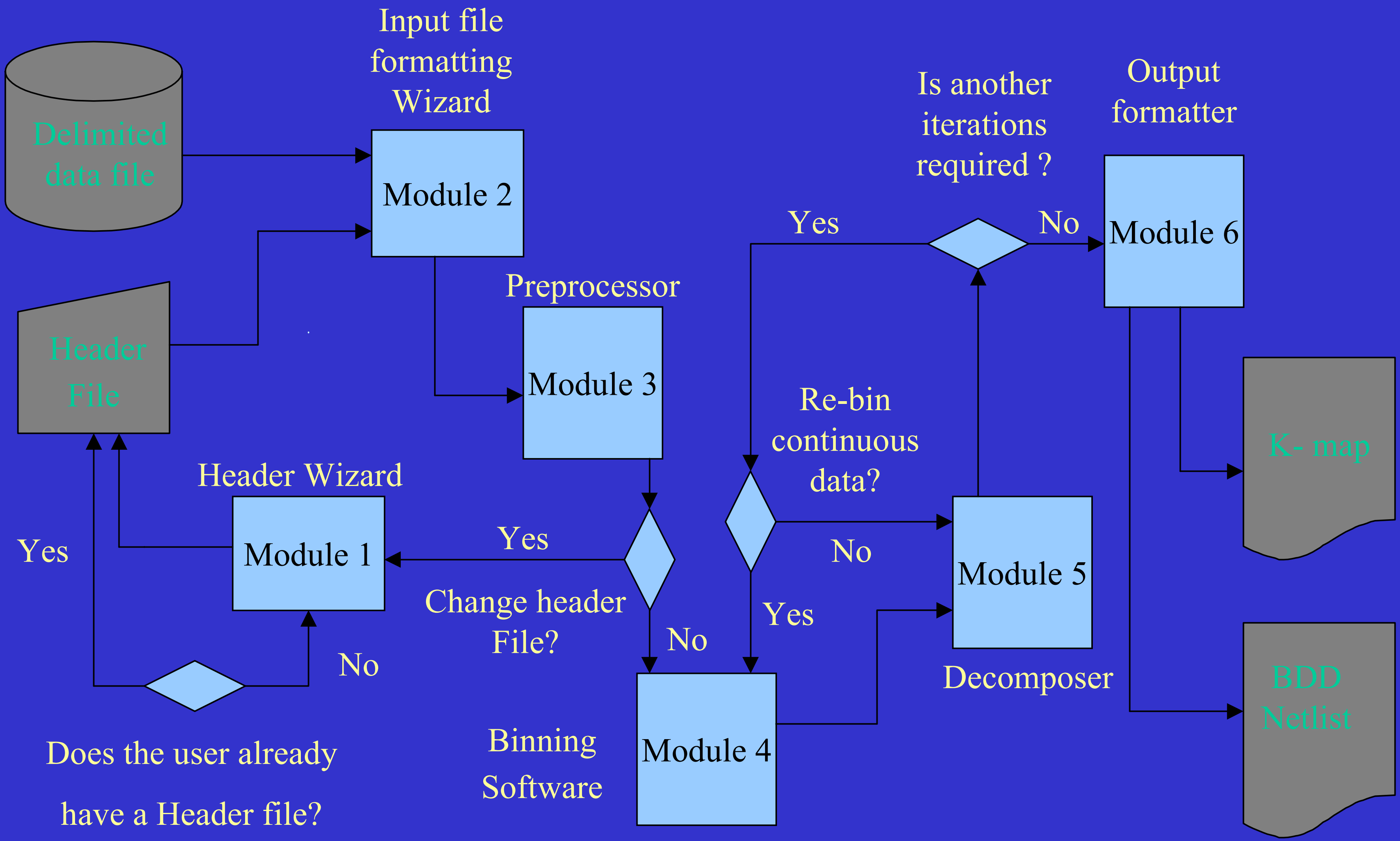
Market

- **Current intended market**
- State and federal epidemiologists working within the United States of America.
- **Anticipated market demand**
- There are approximately 1000 epidemiologists in the United States.
- **Predicable future markets**
- Any application where there is a data set with many unknown values and a user that wishes to generate hypothesis from the data.

Competition

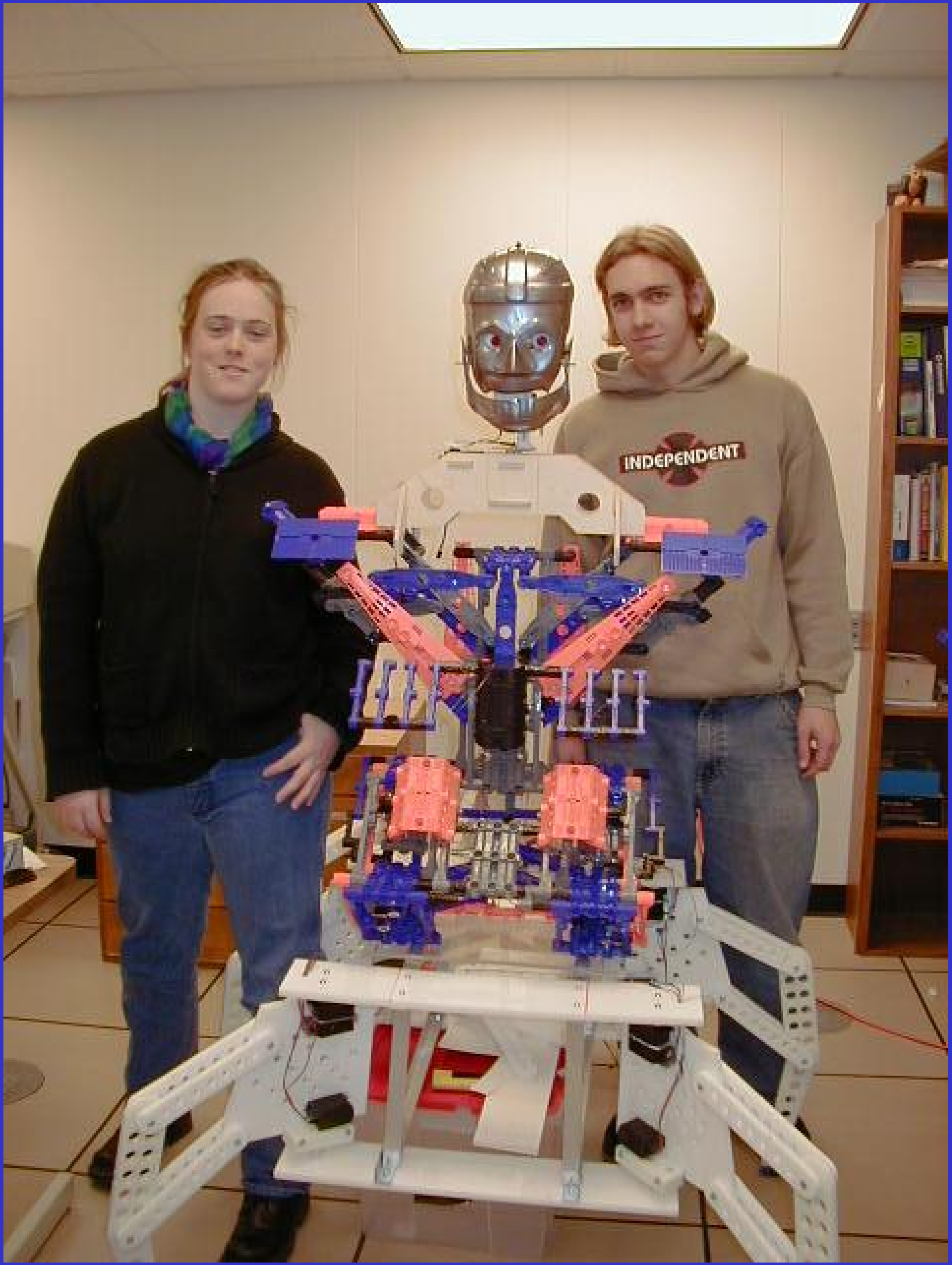
- **Oracle's Darwin®**
  - Darwin's one-click data import wizards accept data in all popular formats, including ODBC, ASCII, and SAS
  - Array of techniques increases modeling accuracy. These techniques include regression trees, neural networks, *k*-nearest neighbors, regression, and clustering algorithms
- **Wizsoft'sWizRule**
  - Reports the rules, and the cases deviating from the norm
  - Sorts the deviated cases by their level of unlikelihood
- **Information Discovery's Data Mining Suite**
  - Uses relational and multi-dimensional data
  - Results are delivered to the user in plain English, accompanied by tables and graph that highlight the key patterns
- **Center for Disease Control's Epi Info**
  - Tailored for Epidemiologist
  - DOS based suite of Application

Flow of the Program



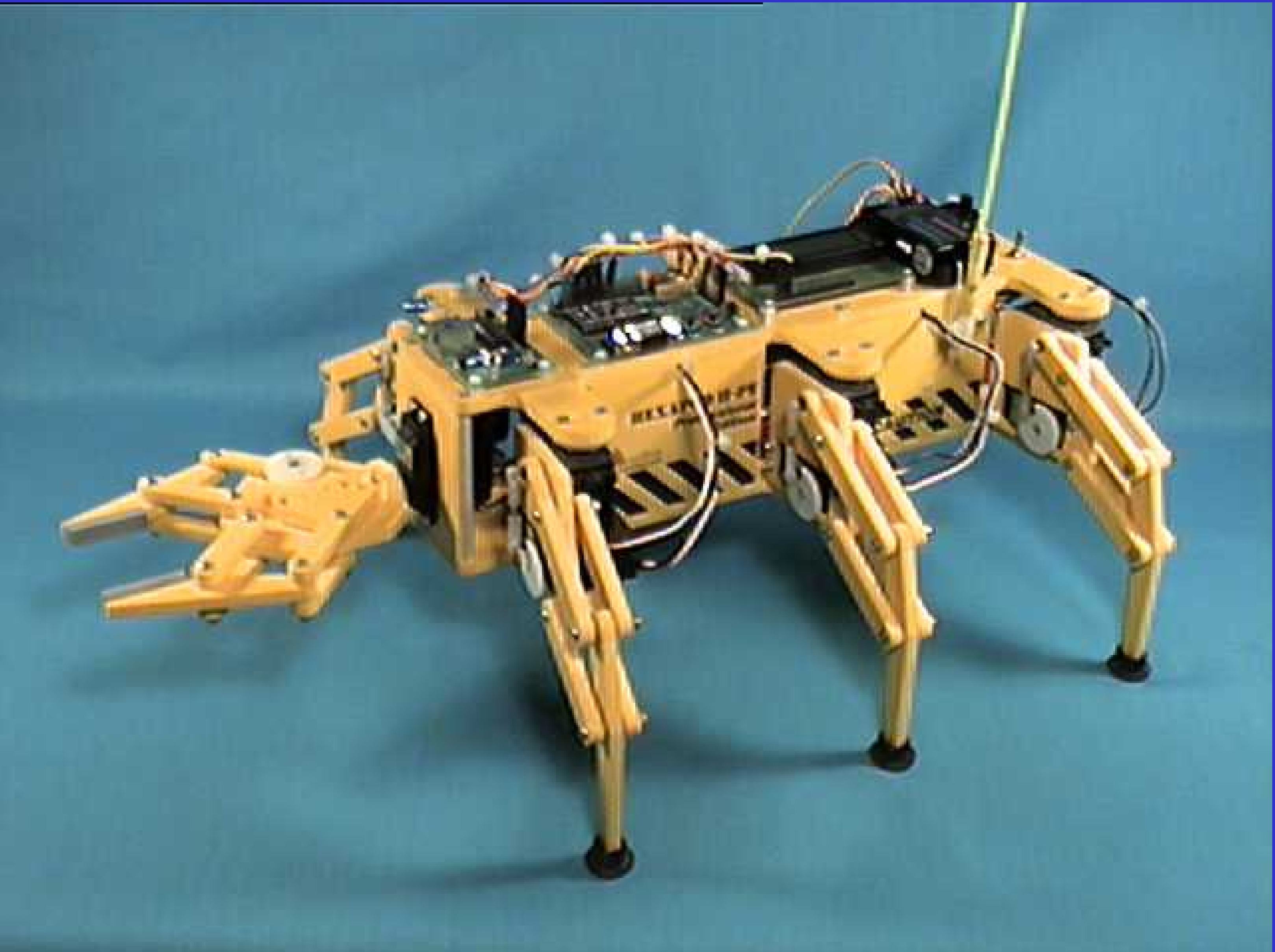
Example of a application

Gait control of a robot puppet for Oregon Cyber Theatre

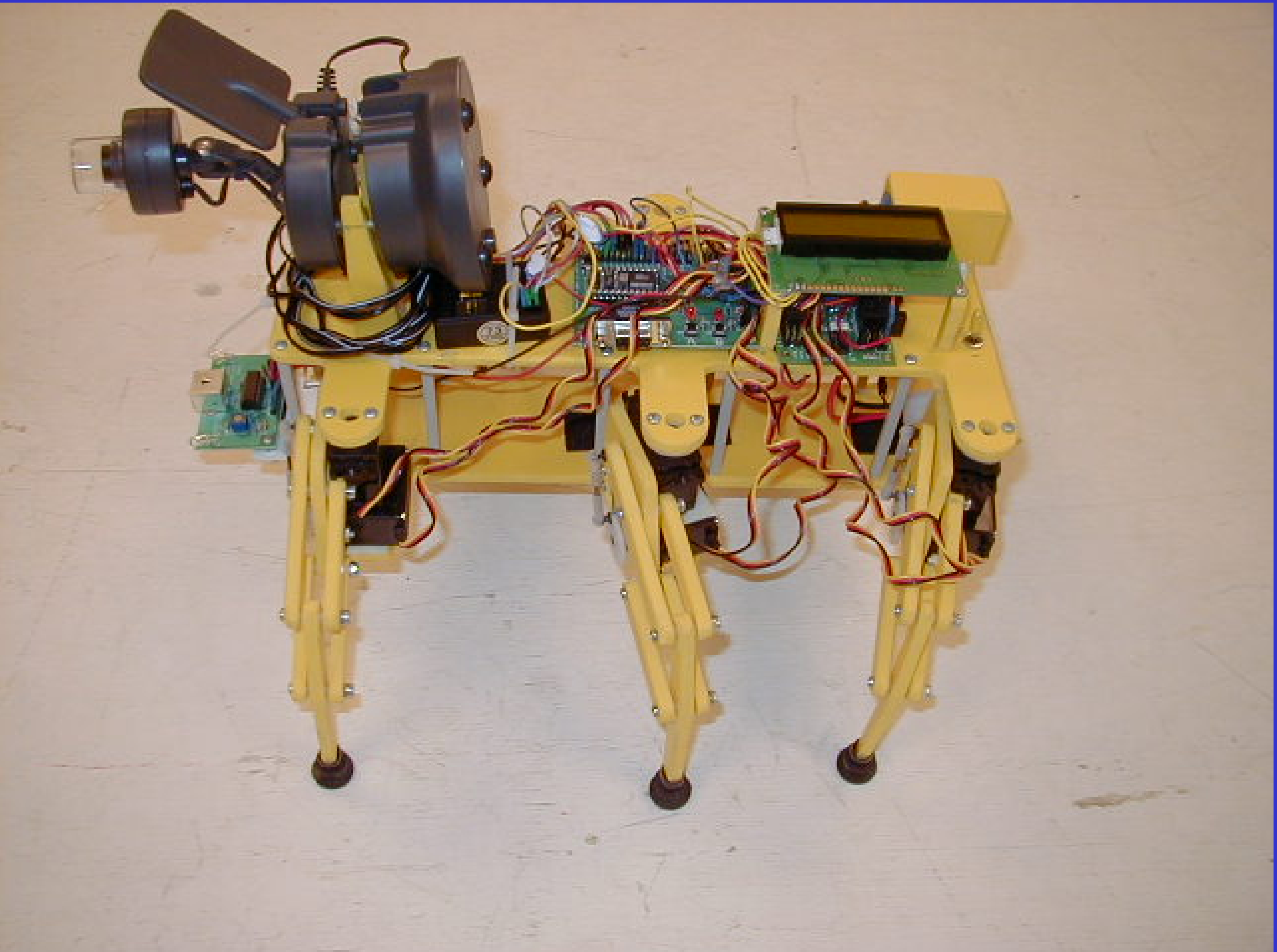


INDEPENDENT

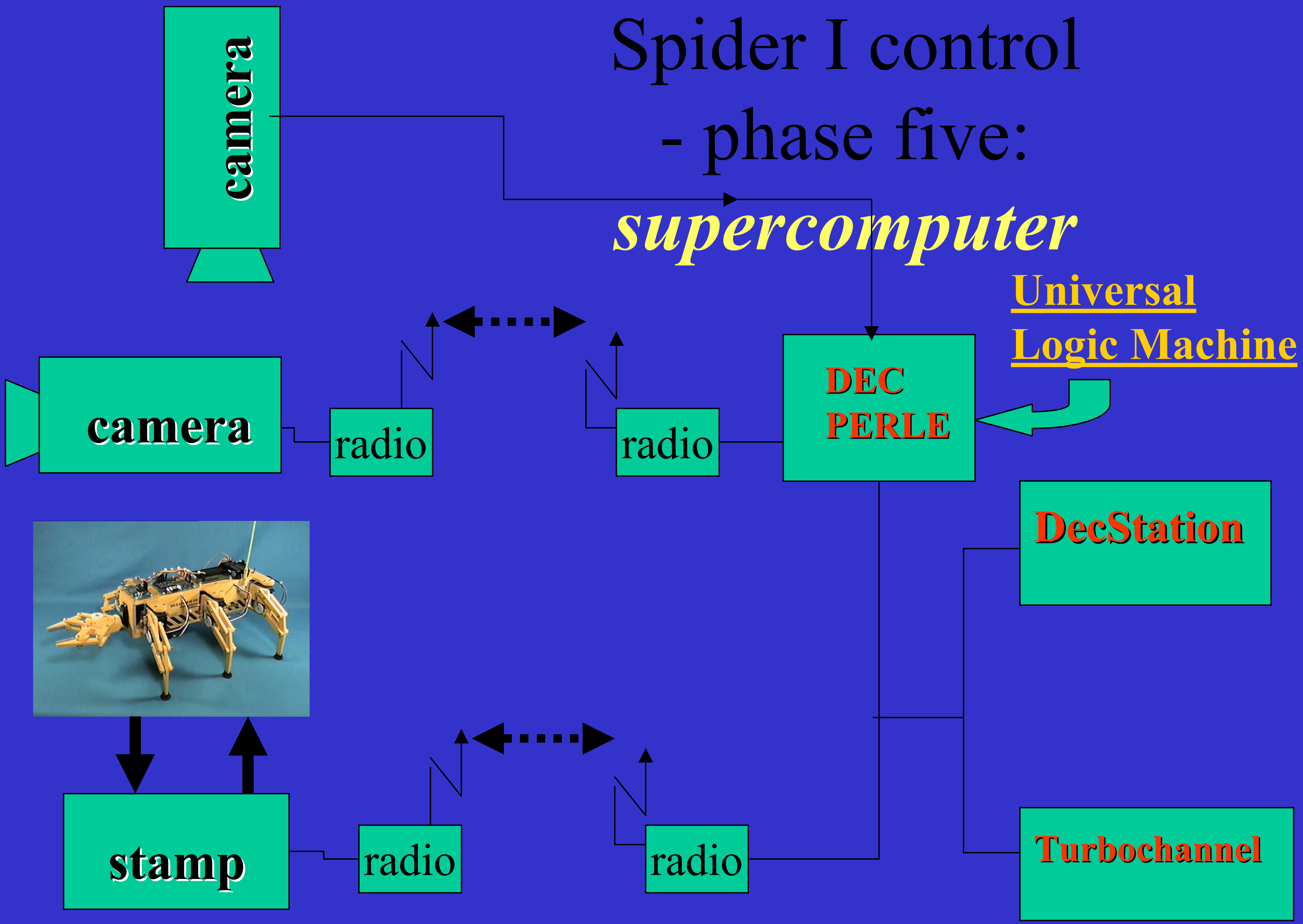
Model with a gripper



Model with an internet camera



Spider I control - phase five: *supercomputer*


camera

camera

radio

radio

DEC PERLE

Universal Logic Machine

DecStation

stamp

radio

radio

Turbochannel

teaching a hexapod to walk

- The following formula describes the exact motion of the shaft of every servo.

$$\theta_i(t) = \theta_o + A_i \sin\left(\omega_i * t + \phi_i\right)$$

- Theta, the angle of the servo’s shaft, is a function of time.
- Theta naught is a base value corresponding to the servo’s middle position. Theta naught will be the same for all the servos.
- ‘A’ is called the amplitude of the oscillation. It relates to how many degrees the shaft is able to rotate through.
- Omega relates to how fast the servo’s shaft rotates back and forth. Currently, for all servos, there are only four possible value that omega may take
- Phi is the relative phase angle.

And a familiar table again

Trial	Inputs							Outputs		
	Servo 1			…	Servo 12			x	y	z
	Amp	Freq	Phase	…	Amp	Freq	Phase			
1	0	1	4	…	1	1	2	-1	1	0
…	…	…	…	…	…	…	…	…	…	…
n	1	1	5	…	1	0	0	-1	-1	1

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.
- A set of tools 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.

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.