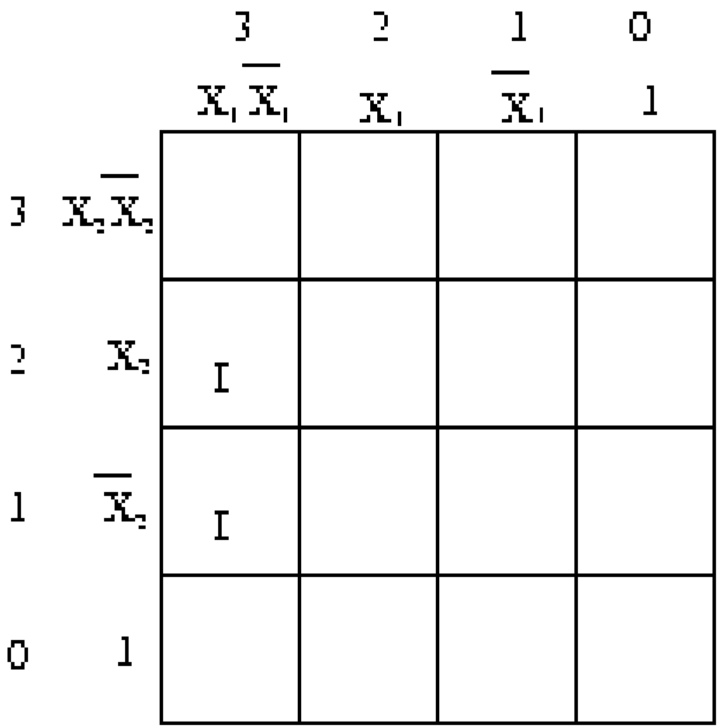
APPROACHES TO FUZZY LOGIC MINIMIZATION

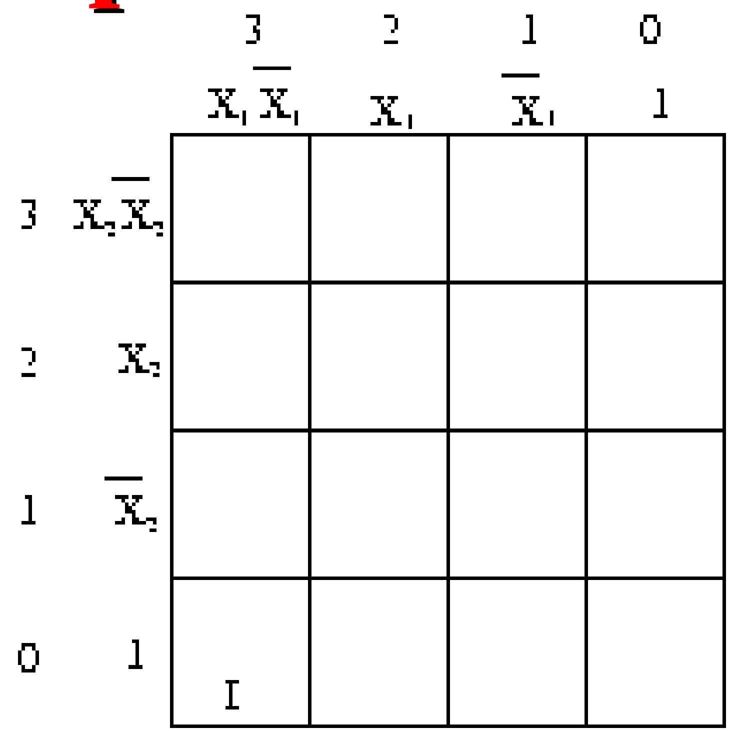
- ➜ Graphical Representations
- Kandel's and Francioni's Approach
- Fuzzy to Multiple-valued Function Conversion Approach
- Fuzzy Logic Decision Diagrams Approach
- Fuzzy Logic Multiplexer

Graphical Representations

- Fuzzy Maps
- Lattice of two variables
- The Subsumption rule
- Form to Decompose a Fuzzy Functions

Fuzzy Maps





(a) $f(x_1, x_2) = x1\, x'_1\, x_2 + x_1\, x'_1\, x'_2$

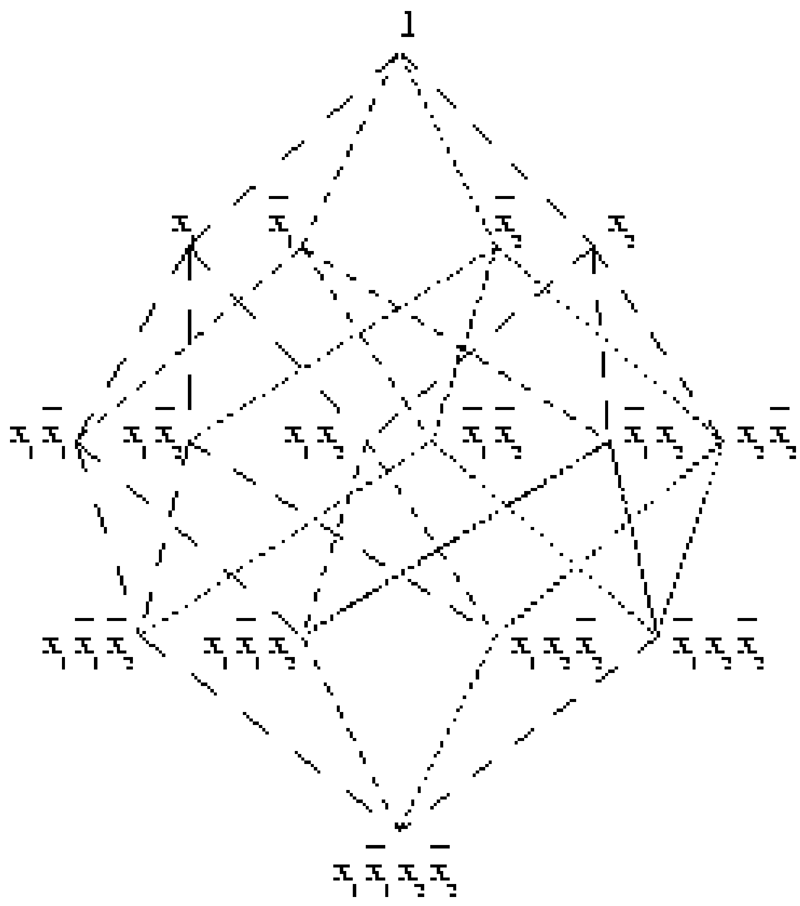
(b) $f(x_1, x_2) = x_1\, x'_1$

Fuzzy map may be regarded as an extension of the Veitch diagram, which forms also the basis for the Karnauph map.

The functions show in (a) and (b) are equivalent to $f(x_1, x_2) = x'_1\, x_2 + x_1\, x'_1\, x'_2 = x_1\, x'_1$

Lattice of Two Variables

- Shows the relationship of all the possible terms.
- Shows which two terms can be reduced to a single term.



(a)

	$x_1\bar{x}_1$	x_1	$\bar{x}_1$	1
$x_1\bar{x}_1$	5	4	4	3
x_1	4	3	3	2
$\bar{x}_1$	4	3	3	2
1	3	2	2	1

(b)

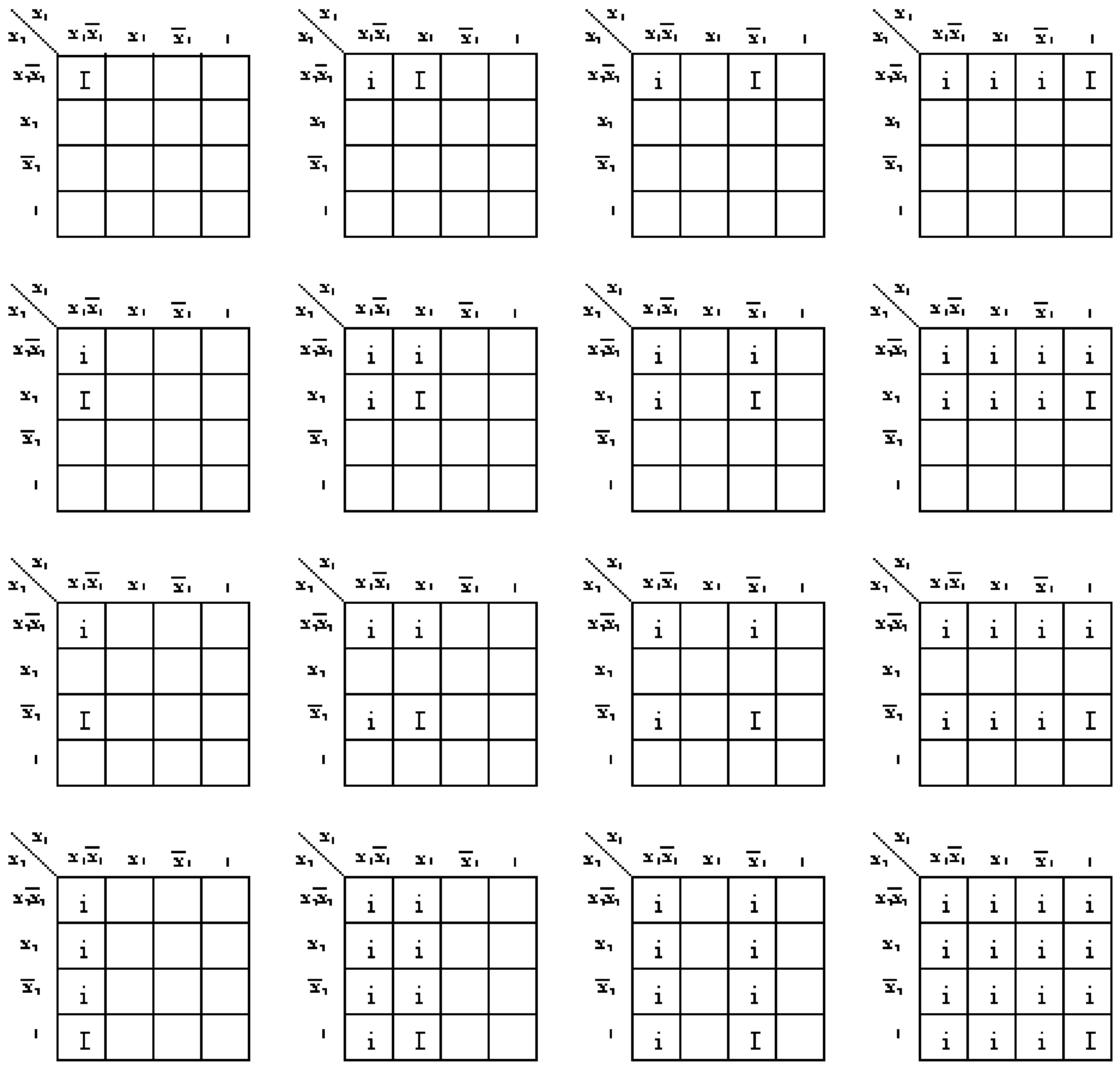
The Subsumption Rule

Used to reduce a fuzzy logic function.

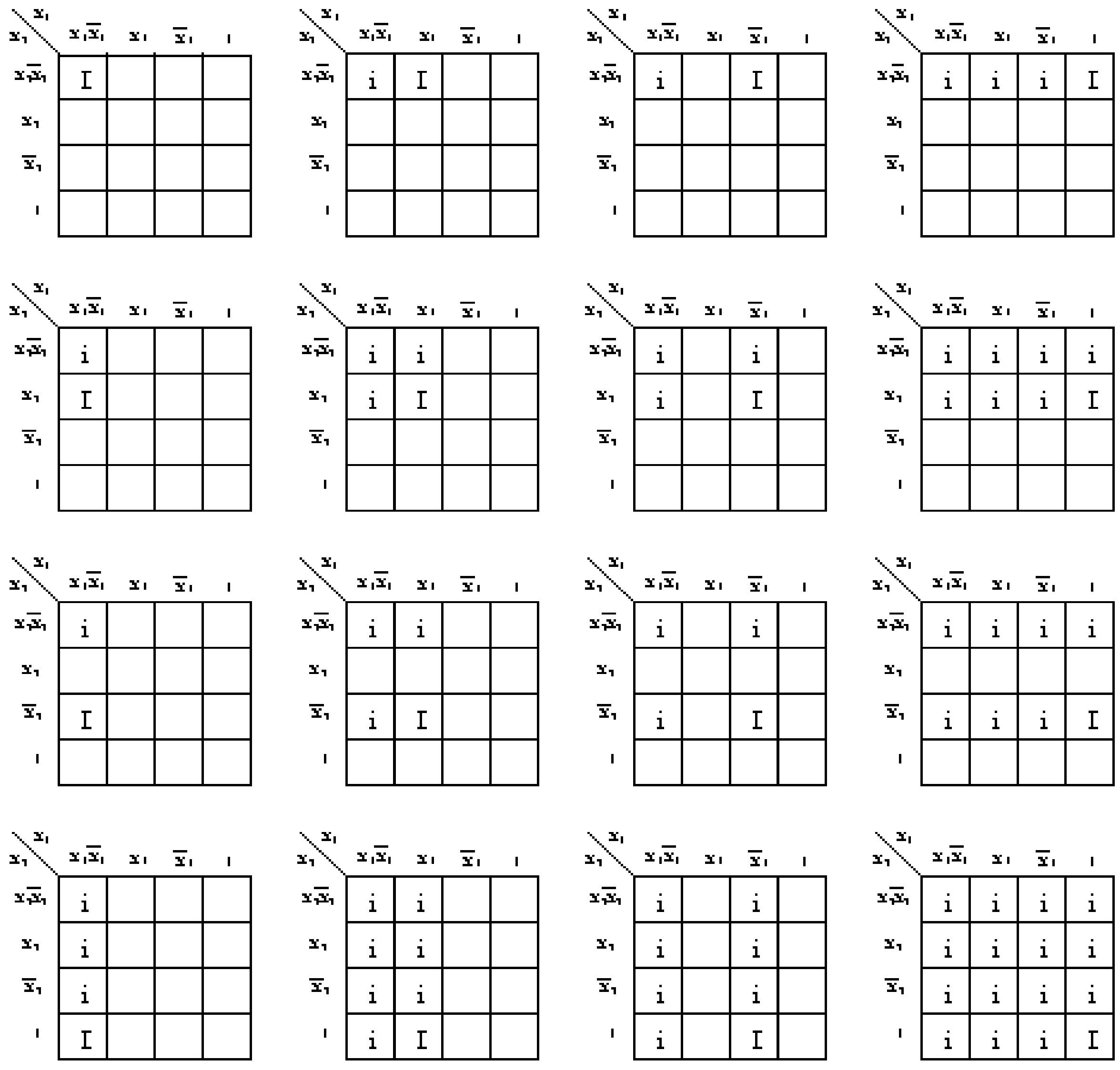
The subsumption rule is:

$\alpha\, x_i\, x'_I\, \beta + \alpha'\, x_i\, x'_I\, \beta = x_i\, x'_I\, \beta$

Operations on two variable map are shown with I subsuming i.



$$\alpha\, x_i\, x'_I\, \beta + \alpha'\, x_i\, x'_I\, \beta = x_i\, x'_I\, \beta$$



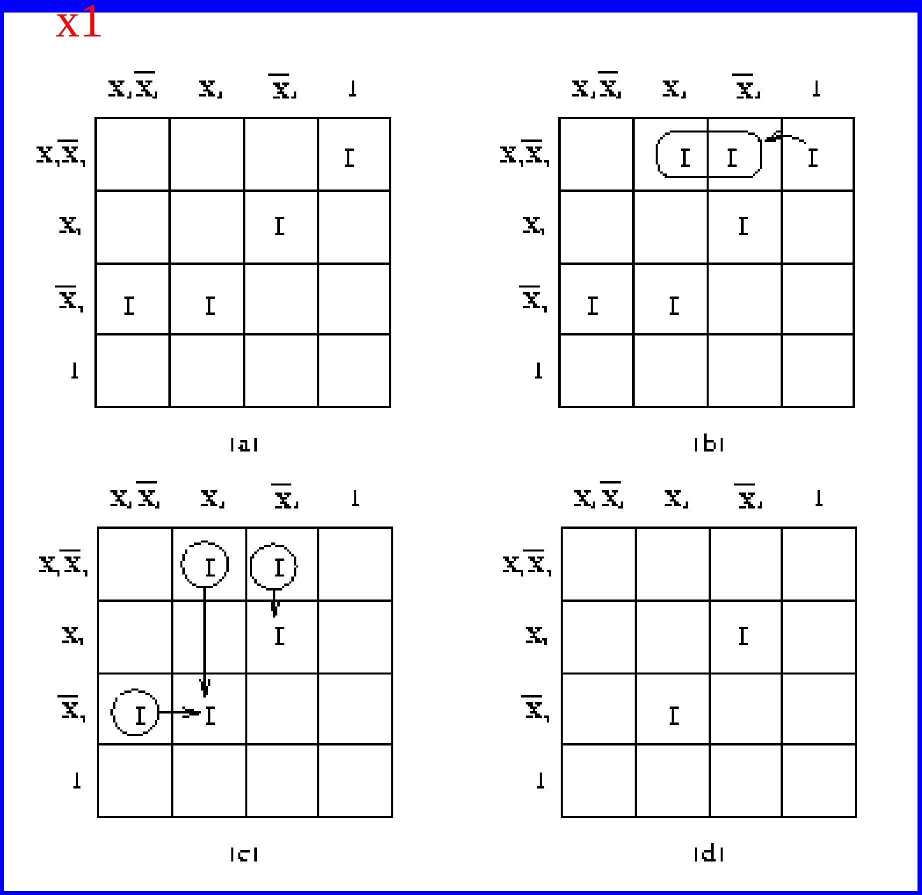
The Subsumption Rule

Form Needed to Decompose Fuzzy Functions

Form requirements:

- Sum-of-products
- Canonical

Figures show the function $x_2\, x'_2 + x'_1\, x_2 + x_1\, x'_2 + x_1\, x'_1\, x'_2$ before using the subsuming rules in (a) and after in (d) $x'_1\, x_2 + x_1\, x'_2$.



$\alpha\, x_i\, x'_I\, \beta + \alpha'\, x_i\, x'_I\, \beta = x_i\, x'_I\, \beta$

$$x_2\, x'_2 + x'_1\, x_2 + x_1\, x'_2 + x_1\, x'_1\, x'_2$$

$$= x_2\, x'_2 + x'_1\, x_2 + x_1\, x'_2\, (1 + x'_1)$$

$$= x_2\, x'_2 + x'_1\, x_2 + x_1\, x'_2$$

$$= x'_1\, x_2 + x_1\, x'_2\,.$$

APPROACHES TO FUZZY LOGIC DECOMPOSITION

- Graphical Representations

➔ Kandel's and Francioni's Approach

- Fuzzy to Multiple-valued Function Conversion Approach
- Fuzzy Logic Decision Diagrams Approach
- Fuzzy Logic Multiplexer

Kandel's and Francioni's Approach

- Decomposition Implicant Pattern (DIP)
- Variable Matching DIP’s Table
- S-Maps
- Example using Kandel and Francioni approach Example
  - Second Decomposition in Example
  - Fuzzy Logic Circuits from Example

Decomposition Implicant Pattern (DIP)

x_i \ x_t	3	2	1	0
3				
2		a		
1				$\bar{a}$
0			$\bar{a}$	g

x_i \ x_t	3	2	1	0
3				
2				$\bar{b}$
1			b	
0		$\bar{b}$		g

x_i \ x_t	3	2	1	0
3				
2			c	
1				$\bar{c}$
0		$\bar{c}$		g

x_i \ x_t	3	2	1	0
3				
2				$\bar{d}$
1		d		
0			$\bar{d}$	g

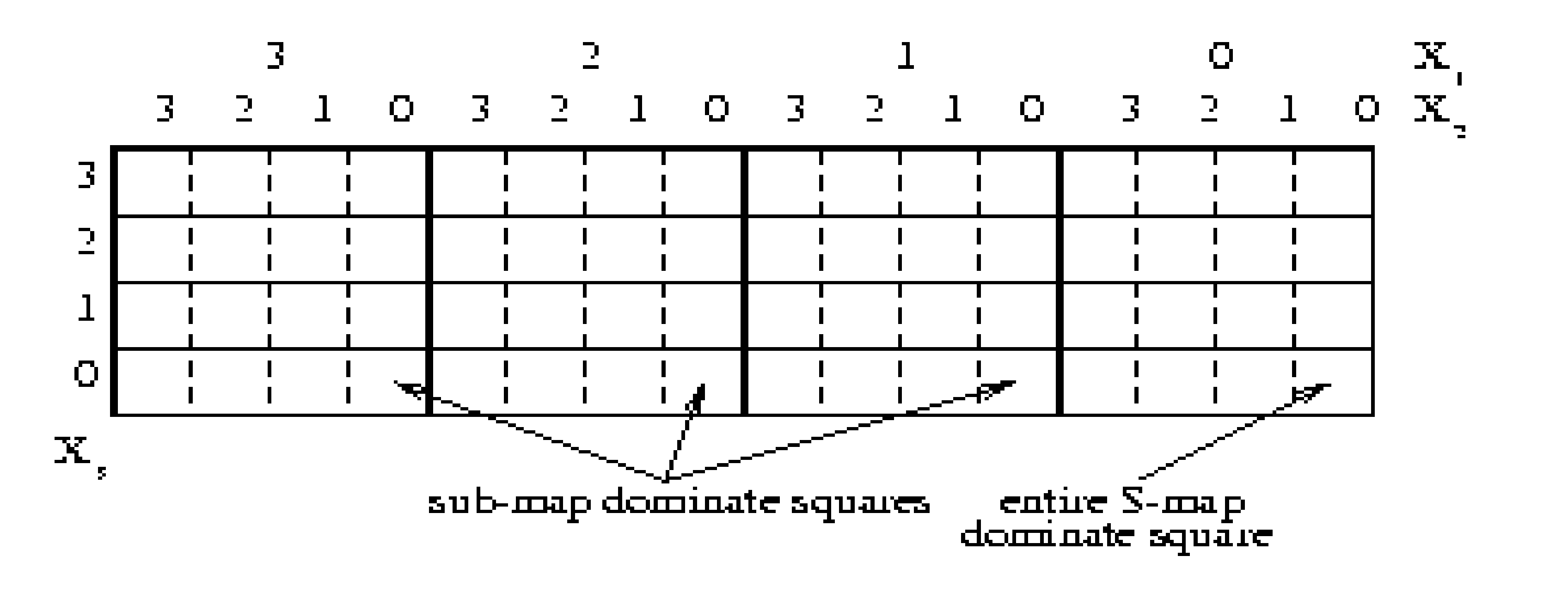
x_i \ x_t	3	2	1	0
3				
2		$\bar{e}$	e	
1		e	$\bar{e}$	
0				g

Variable Matching DIPs Table

Tabular form of Decomposition Implicant Pattern (DIP) used in Kandel's and Francioni's approach

$h(X)$	$h'(X)$	DIP
x_i	x'_i	-
$x_i x_j$	$x'_i + x'_j$	1
$x'_i x'_j$	$x_i + x_j$	2
$x'_i x_j$	$x_i + x'_j$	3
$x_i x'_j$	$x'_i + x_j$	4
$x_i x_j + x'_i x'_j$	$x_i x'_j + x'_i x_j$	5

S-Maps



- Arrange two-variable fuzzy maps for n variables.
- This method is just done by iteration to form an n variable S-map.
- This shows X_1 is made up of repeated X_2 and X_3 two variable maps.

Example using Kandel and Francioni approach

$f = x'y'zz' + xz + w'x'zz' + wyz$

From DIP 1 implies:

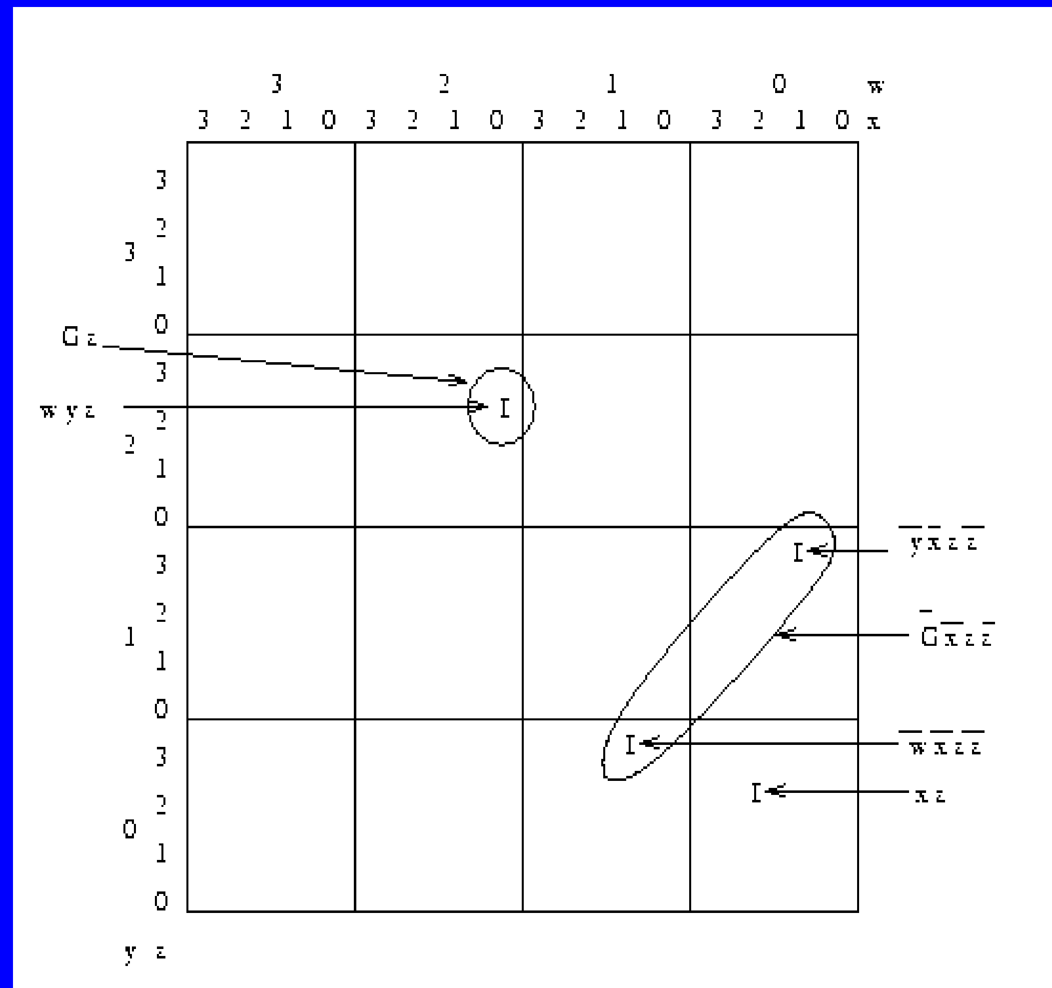
$g(w,y) = wy$, $G'(w,y) = w' + y'$

$f = (wy)z + (w' + y')x'zz' + xz$

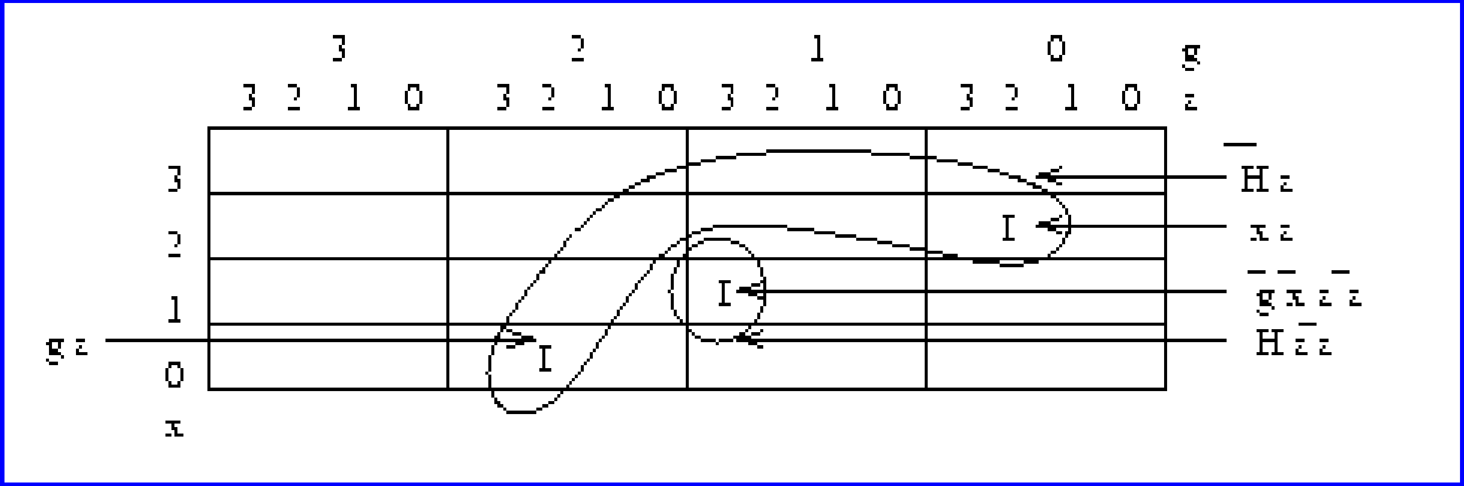
By substituting: $G(w,y) = G(Y)$ and $G'(w,y) = G'(Y)$

$f = F[(G(w,y), x,z)]$

$\quad = G(Y)z + G'(Y)x'zz' + xz$



Second Decomposition in Example



Let g = G(w, y) and g' = G'(w,y) Then F(g,x,z) =gz + g'x'zz' + xz

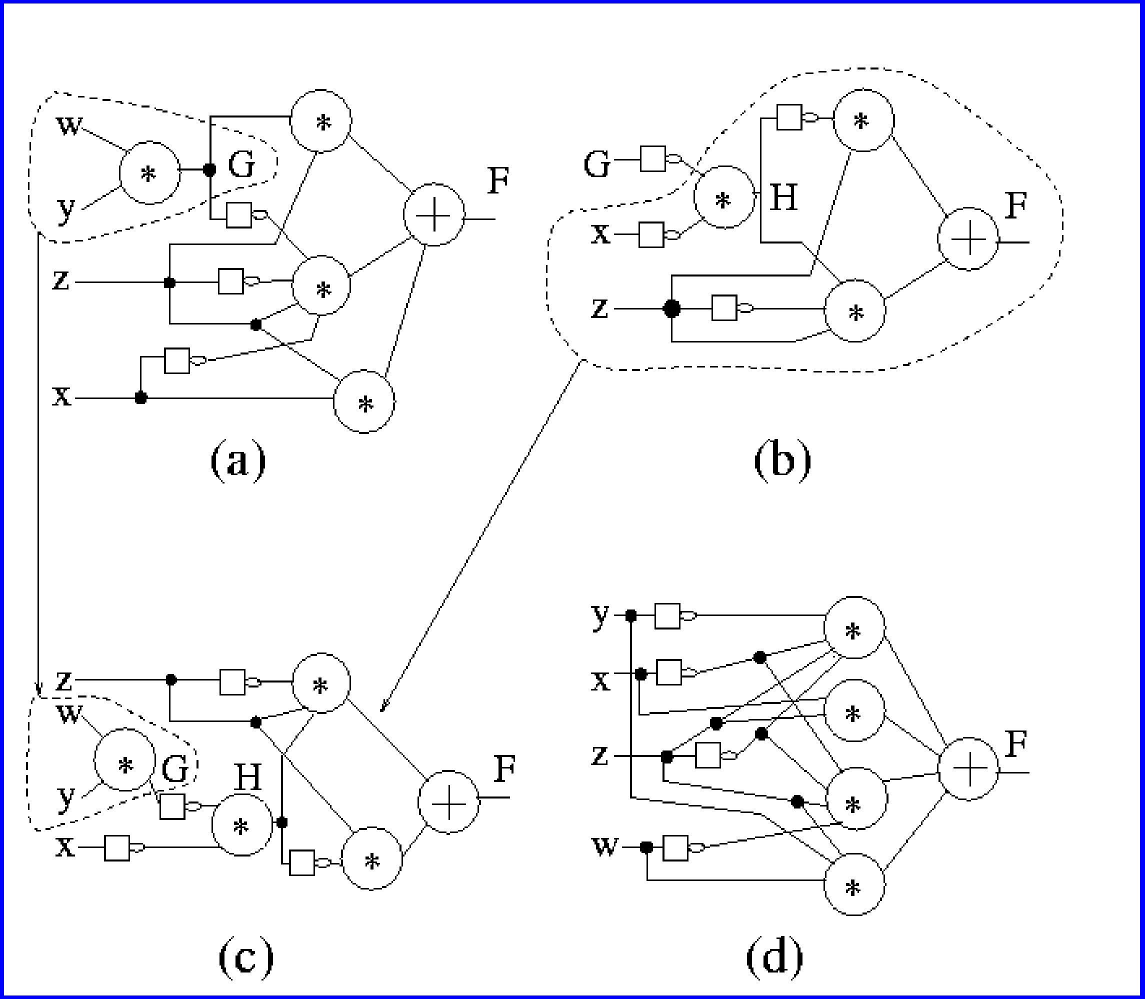
From DIP 2 implies: H(g,x) = g'x', H'(g,x) = g + x
g = G(w,y) = wy, g' = G(w,y) = w' + y'

So by DeMorgan's Law and distributive law we obtain:
H(w,y,z) = (wy)'x' = (w' + y')x' = w'x'+ y'x'
f(w,x,y,z) = H(w,y,x) = (g'x')zz' + (g + x)z

And therefore: F(w,x,y,z) = F[H(w,x,y),z] = H(w,y,x)zz' + H'(w,y,x)z

Fuzzy Logic Circuits from Example

- (a) First level of decomposition.
- (b) Second level of decomposition.
- (c) Result of first and second level of decomposition.
- (d) Original function.



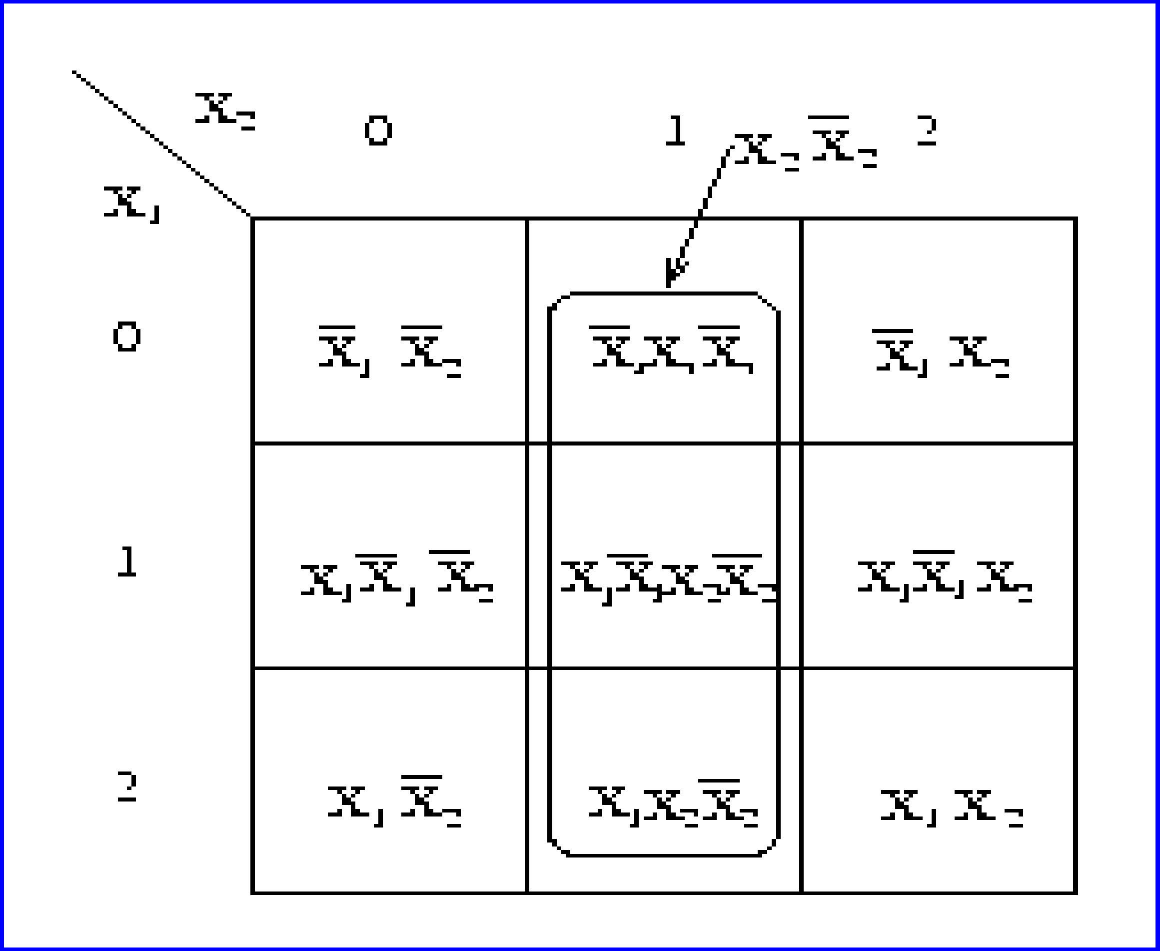
APPROACHES TO FUZZY LOGIC DECOMPOSITION

- Graphical Representations
- Kandel's and Francioni's Approach
- ➔ Fuzzy to Multiple-valued Function Conversion Approach
- Fuzzy Logic Decision Diagrams Approach
- Fuzzy Logic Multiplexer

Fuzzy to Multiple-valued Function Conversion Approach

- Fuzzy Function Ternary Map
- Fuzzy Function to Three-valued Function Conversion Example
  - The MAX operation form the result
  - The result from canonical form same as non-canonical form

Fuzzy Function Ternary Map



This shows the mapping between the fuzzy terms and terms in the ternary map.

Fuzzy Function to Three-valued Function **Conversion** Example

Conversion of the Fuzzy function terms:

$x_2 x'_2$

$x'_1 x_2$

$x_1 x'_2$

$x_1 x'_1 x_2$

In non-canonical form using the MIN operation as shown from

$f = x_2 x'_2 + x'_1 x_2 + x_1 x'_2 + x_1 x'_1 x'_2$

x_2

$x_2 \backslash x_1$	0	1	2
0	0	0	0
1	1	1	1
2	2	2	2

•

$\bar{x}_2$

$x_2 \backslash x_1$	0	1	2
0	2	2	2
1	1	1	1
2	0	0	0

=

$x_2 \bullet \bar{x}_2$

$x_2 \backslash x_1$	0	1	2
0	0	0	0
1	1	1	1
2	0	0	0

$\bar{x}_1$

$x_2 \backslash x_1$	0	1	2
0	2	1	0
1	2	1	0
2	2	1	0

•

x_2

$x_2 \backslash x_1$	0	1	2
0	0	0	0
1	1	1	1
2	2	2	2

=

$\bar{x}_1 \bullet x_2$

$x_2 \backslash x_1$	0	1	2
0	0	0	0
1	1	1	0
2	2	1	0

x_1

$x_2 \backslash x_1$	0	1	2
0	0	1	2
1	0	1	2
2	0	1	2

•

$\bar{x}_2$

$x_2 \backslash x_1$	0	1	2
0	2	2	2
1	1	1	1
2	0	0	0

=

$x_1 \bullet \bar{x}_2$

$x_2 \backslash x_1$	0	1	2
0	0	1	2
1	0	1	1
2	0	0	0

x_1

$x_2 \backslash x_1$	0	1	2
0	0	1	2
1	0	1	2
2	0	1	2

•

$\bar{x}_1$

$x_2 \backslash x_1$	0	1	2
0	2	1	0
1	2	1	0
2	2	1	0

•

$\bar{x}_2$

$x_2 \backslash x_1$	0	1	2
0	2	2	2
1	1	1	1
2	0	0	0

=

$x_1 \bullet \bar{x}_1 \bullet \bar{x}_2$

$x_2 \backslash x_1$	0	1	2
0	0	1	0
1	0	1	0
2	0	0	0

The MAX operation form the result

- Combining the three-valued term functions into a single three-valued function is performed using the MAX Operation

$X_2 \bullet \overline{X}_2$

$X_2 \backslash X_1$	0	1	2
0	0	0	0
1	1	1	1
2	0	0	0

+

$\overline{X}_1 \bullet X_2$

$X_2 \backslash X_1$	0	1	2
0	0	0	0
1	1	1	0
2	2	1	0

+

$X_1 \bullet \overline{X}_2$

$X_2 \backslash X_1$	0	1	2
0	0	1	2
1	0	1	1
2	0	0	0

+

$X_1 \bullet \overline{X}_1 \bullet \overline{X}_2$

$X_2 \backslash X_1$	0	1	2
0	0	1	0
1	0	1	0
2	0	0	0

=

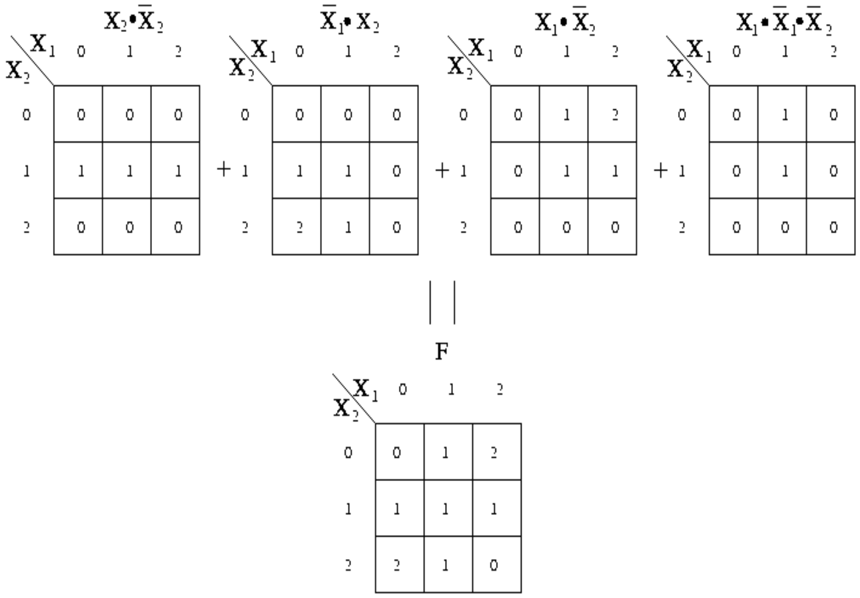
F

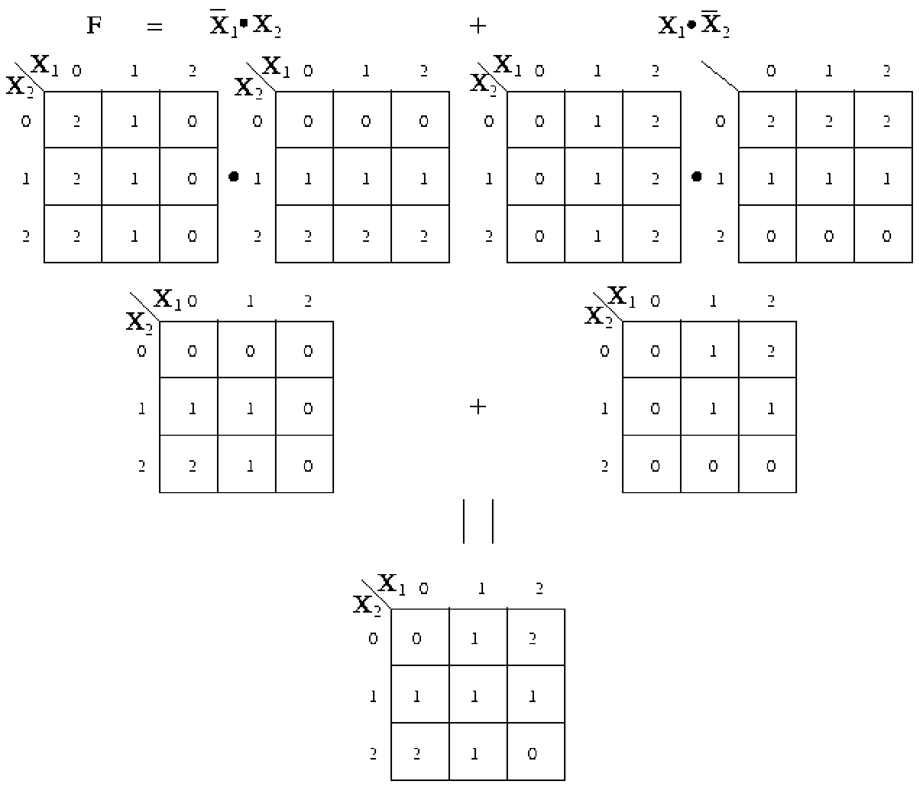
$X_2 \backslash X_1$	0	1	2
0	0	1	2
1	1	1	1
2	2	1	0

The result form canonical form same as non-canonical form

- $F = x_2x'_2 + x'_1x_2 + x_1x'_2 + x_1x'_1x'_2$
  conversion is equal to
  $F(x_1x_2) = x'_1x_2 + x_1x'_2$

  (See slide 29)





Fuzzy to Multiple-valued Function Conversion Approach Example

- Fuzzy to Multiple-valued Function Conversion Example
  - Fuzzy function to Multiple-valued function
  - Input and results of decomposition
  - Multiple-valued function to fuzzy function with circuit
- Method of Doing More Examples
  - Using Mathcad to do the MIN, MAX Operations
- Fuzzy Function Decomposition Results

Fuzzy Function to Multiple-valued Function

XZ

Y	0			1			2		
X \ Z	0	1	2	0	1	2	0	1	2
0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	1	1	0	1	1
2	0	1	2	0	1	2	0	1	2

+

$\overline{X}\,\overline{Y}\,Z\,\overline{Z}$

Y	0			1			2		
X \ Z	0	1	2	0	1	2	0	1	2
0	0	1	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0

+

YZ

Y	0			1			2		
X \ Z	0	1	2	0	1	2	0	1	2
0	0	0	0	0	1	1	0	1	2
1	0	0	0	0	1	1	0	1	2
2	0	0	0	0	1	1	0	1	2

=

Y	0			1			2		
X \ Z	0	1	2	0	1	2	0	1	2
0	0	1	0	0	1	1	0	1	2
1	0	1	1	0	1	1	0	1	2
2	0	1	2	0	1	2	0	1	2

$F(x,y,z) = xz + x'y'zz' + yz$

Input Function			
X	Y	Z	F
0	0	0	0
0	0	1	1
0	0	2	0
1	0	0	0
1	0	1	1
1	0	2	1
2	0	0	0
2	0	1	1
2	0	2	2
0	1	0	0
0	1	1	1
0	1	2	1
1	1	0	0
1	1	1	1
1	1	2	1
2	1	0	0
2	1	1	1
2	1	2	2
0	2	0	0
0	2	1	1
0	2	2	2
1	2	0	0
1	2	1	1
1	2	2	2
2	2	0	0
2	2	1	1
2	2	2	2

(a)

G Function		
X	Y	G
0	1	1
1	0	1
2	0	2
0	0	0
1	2	2
2	2	2
0	2	2
1	1	1
2	1	2

(b)

H Function		
Z	G	F
2	2	2
1	2	1
0	2	0
2	1	1
1	1	1
0	1	0
2	0	0
1	0	1
0	0	0

(c)

F(x,y,z) = xz + x'y'zz' + yz

X Z

Y	0			1			2		
X \ Z	0	1	2	0	1	2	0	1	2
0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	1	1	0	1	1
2	0	1	2	0	1	2	0	1	2

+

$\bar{X}\ \bar{Y}\ Z\ \bar{Z}$

Y	0			1			2		
X \ Z	0	1	2	0	1	2	0	1	2
0	0	1	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0

+

Y Z

Y	0			1			2		
X \ Z	0	1	2	0	1	2	0	1	2
0	0	0	0	0	1	1	0	1	2
1	0	0	0	0	1	1	0	1	2
2	0	0	0	0	1	1	0	1	2

=

Y	0			1			2		
X \ Z	0	1	2	0	1	2	0	1	2
0	0	1	0	0	1	1	0	1	2
1	0	1	1	0	1	1	0	1	2
2	0	1	2	0	1	2	0	1	2

Input and results of decomposition

- The input function table in generated from the result of the Figure 3.29.
- The input table is shown in Figure 3.30a, along with the resulting multiple-valued decomposition functions G and H in Figure 3.30b and Figure 3.30c, respectively.
- Input function data (b) output G function, and © output H function

Input Function

X	Y	Z	F
0	0	0	0
0	0	1	1
0	0	2	0
1	0	0	0
1	0	1	1
1	0	2	1
2	0	0	0
2	0	1	1
2	0	2	2
0	1	0	0
0	1	1	1
0	1	2	1
1	1	0	0
1	1	1	1
1	1	2	1
2	1	0	0
2	1	1	1
2	1	2	2
0	2	0	0
0	2	1	1
0	2	2	2
1	2	0	0
1	2	1	1
1	2	2	2
2	2	0	0
2	2	1	1
2	2	2	2

(a)

G Function

X	Y	G
0	1	1
1	0	1
2	0	2
0	0	0
1	2	2
2	2	2
0	2	2
1	1	1
2	1	2

(b)

H Function

Z	G	F
2	2	2
1	2	1
0	2	0
2	1	1
1	1	1
0	1	0
2	0	0
1	0	1
0	0	0

(c)

Multiple-valued function to fuzzy function with circuit

The results of the decomposition process, functions G and H, are shown in Figure 3.31, a,b, respectively, as multiple-valued maps.
Fuzzy terms Gz, G'zz' and zz' of H are shown. Two solutions are obtained, G(x,y) = x+y, H(x,y) = Gz+zz' (Fig. 3.31c)
and
G(x,y) = x+y, H(x,y) = Gz+G'zz'(Fig.3.3ld)

Results of decomposition
(a) resulting multiple-valued map of functions G,
(b) resulting map
of function H from Example 8,
(c) first decomposition
H (x,y) = Gz + G'zz',
(d) Second decomposition H(x,y) = Gz + zz'.

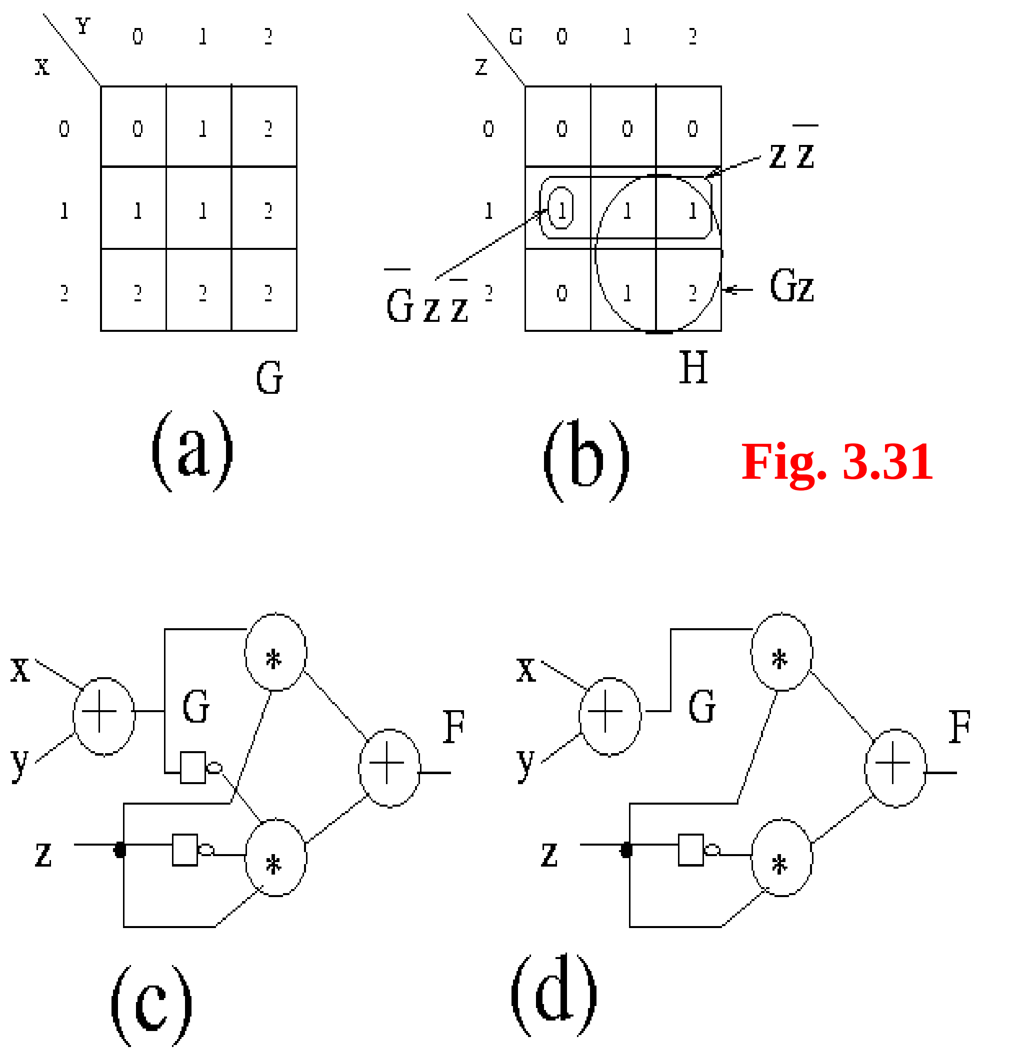


Fig. 3.31

Method for Larger Examples

W

	w	0			1			2		
y	z \ x	0	1	2	0	1	2	0	1	2
0	0	0	0	0	1	1	1	2	2	2
	1	0	0	0	1	1	1	2	2	2
	2	0	0	0	1	1	1	2	2	2
1	0	0	0	0	1	1	1	2	2	2
	1	0	0	0	1	1	1	2	2	2
	2	0	0	0	1	1	1	2	2	2
2	0	0	0	0	1	1	1	2	2	2
	1	0	0	0	1	1	1	2	2	2
	2	0	0	0	1	1	1	2	2	2

$\overline{W}$

	w	0			1			2		
y	z \ x	0	1	2	0	1	2	0	1	2
0	0	2	2	2	1	1	1	0	0	0
	1	2	2	2	1	1	1	0	0	0
	2	2	2	2	1	1	1	0	0	0
1	0	2	2	2	1	1	1	0	0	0
	1	2	2	2	1	1	1	0	0	0
	2	2	2	2	1	1	1	0	0	0
2	0	2	2	2	1	1	1	0	0	0
	1	2	2	2	1	1	1	0	0	0
	2	2	2	2	1	1	1	0	0	0

X

	w	0			1			2		
y	z \ x	0	1	2	0	1	2	0	1	2
0	0	0	1	2	0	1	2	0	1	2
	1	0	1	2	0	1	2	0	1	2
	2	0	1	2	0	1	2	0	1	2
1	0	0	1	2	0	1	2	0	1	2
	1	0	1	2	0	1	2	0	1	2
	2	0	1	2	0	1	2	0	1	2
2	0	0	1	2	0	1	2	0	1	2
	1	0	1	2	0	1	2	0	1	2
	2	0	1	2	0	1	2	0	1	2

$\overline{X}$

	w	0			1			2		
y	z \ x	0	1	2	0	1	2	0	1	2
0	0	2	1	0	2	1	0	2	1	0
	1	2	1	0	2	1	0	2	1	0
	2	2	1	0	2	1	0	2	1	0
1	0	2	1	0	2	1	0	2	1	0
	1	2	1	0	2	1	0	2	1	0
	2	2	1	0	2	1	0	2	1	0
2	0	2	1	0	2	1	0	2	1	0
	1	2	1	0	2	1	0	2	1	0
	2	2	1	0	2	1	0	2	1	0

Y

	w	0			1			2		
y	z \ x	0	1	2	0	1	2	0	1	2
0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1
2	0	2	2	2	2	2	2	2	2	2
	1	2	2	2	2	2	2	2	2	2
	2	2	2	2	2	2	2	2	2	2

$\overline{Y}$

	w	0			1			2		
y	z \ x	0	1	2	0	1	2	0	1	2
0	0	2	2	2	2	2	2	2	2	2
	1	2	2	2	2	2	2	2	2	2
	2	2	2	2	2	2	2	2	2	2
1	0	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0

Z

	w	0			1			2		
y	z \ x	0	1	2	0	1	2	0	1	2
0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1
	2	2	2	2	2	2	2	2	2	2
1	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1
	2	2	2	2	2	2	2	2	2	2
2	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1
	2	2	2	2	2	2	2	2	2	2

$\overline{Z}$

	w	0			1			2		
y	z \ x	0	1	2	0	1	2	0	1	2
0	0	2	2	2	2	2	2	2	2	2
	1	1	1	1	1	1	1	1	1	1
	2	0	0	0	0	0	0	0	0	0
1	0	2	2	2	2	2	2	2	2	2
	1	1	1	1	1	1	1	1	1	1
	2	0	0	0	0	0	0	0	0	0
2	0	2	2	2	2	2	2	2	2	2
	1	1	1	1	1	1	1	1	1	1
	2	0	0	0	0	0	0	0	0	0

Method of Doing More Examples

Four variable Multiple-valued maps used to convert four valued fuzzy functions using Mathcad.

W

y	z \ x	w = 0: 0	1	2	w = 1: 0	1	2	w = 2: 0	1	2
0	0	0	0	0	1	1	1	2	2	2
	1	0	0	0	1	1	1	2	2	2
	2	0	0	0	1	1	1	2	2	2
1	0	0	0	0	1	1	1	2	2	2
	1	0	0	0	1	1	1	2	2	2
	2	0	0	0	1	1	1	2	2	2
2	0	0	0	0	1	1	1	2	2	2
	1	0	0	0	1	1	1	2	2	2
	2	0	0	0	1	1	1	2	2	2

$\overline{W}$

y	z \ x	w = 0: 0	1	2	w = 1: 0	1	2	w = 2: 0	1	2
0	0	2	2	2	1	1	1	0	0	0
	1	2	2	2	1	1	1	0	0	0
	2	2	2	2	1	1	1	0	0	0
1	0	2	2	2	1	1	1	0	0	0
	1	2	2	2	1	1	1	0	0	0
	2	2	2	2	1	1	1	0	0	0
2	0	2	2	2	1	1	1	0	0	0
	1	2	2	2	1	1	1	0	0	0
	2	2	2	2	1	1	1	0	0	0

X

y	z \ x	w = 0: 0	1	2	w = 1: 0	1	2	w = 2: 0	1	2
0	0	0	1	2	0	1	2	0	1	2
	1	0	1	2	0	1	2	0	1	2
	2	0	1	2	0	1	2	0	1	2
1	0	0	1	2	0	1	2	0	1	2
	1	0	1	2	0	1	2	0	1	2
	2	0	1	2	0	1	2	0	1	2
2	0	0	1	2	0	1	2	0	1	2
	1	0	1	2	0	1	2	0	1	2
	2	0	1	2	0	1	2	0	1	2

$\overline{X}$

y	z \ x	w = 0: 0	1	2	w = 1: 0	1	2	w = 2: 0	1	2
0	0	2	1	0	2	1	0	2	1	0
	1	2	1	0	2	1	0	2	1	0
	2	2	1	0	2	1	0	2	1	0
1	0	2	1	0	2	1	0	2	1	0
	1	2	1	0	2	1	0	2	1	0
	2	2	1	0	2	1	0	2	1	0
2	0	2	1	0	2	1	0	2	1	0
	1	2	1	0	2	1	0	2	1	0
	2	2	1	0	2	1	0	2	1	0

Y

y	z \ x	w = 0: 0	1	2	w = 1: 0	1	2	w = 2: 0	1	2
0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1
2	0	2	2	2	2	2	2	2	2	2
	1	2	2	2	2	2	2	2	2	2
	2	2	2	2	2	2	2	2	2	2

$\overline{Y}$

y	z \ x	w = 0: 0	1	2	w = 1: 0	1	2	w = 2: 0	1	2
0	0	2	2	2	2	2	2	2	2	2
	1	2	2	2	2	2	2	2	2	2
	2	2	2	2	2	2	2	2	2	2
1	0	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0

Z

y	z \ x	w = 0: 0	1	2	w = 1: 0	1	2	w = 2: 0	1	2
0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1
	2	2	2	2	2	2	2	2	2	2
1	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1
	2	2	2	2	2	2	2	2	2	2
2	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1
	2	2	2	2	2	2	2	2	2	2

$\overline{Z}$

y	z \ x	w = 0: 0	1	2	w = 1: 0	1	2	w = 2: 0	1	2
0	0	2	2	2	2	2	2	2	2	2
	1	1	1	1	1	1	1	1	1	1
	2	0	0	0	0	0	0	0	0	0
1	0	2	2	2	2	2	2	2	2	2
	1	1	1	1	1	1	1	1	1	1
	2	0	0	0	0	0	0	0	0	0
2	0	2	2	2	2	2	2	2	2	2
	1	1	1	1	1	1	1	1	1	1
	2	0	0	0	0	0	0	0	0	0

Using Mathcad to perform the MIN, MAX Operations

MIN

	w	0			1			2		
y	z \ x	0	1	2	0	1	2	0	1	2
	0	0	0	0	1	1	1	0	0	0
0	1	0	0	0	1	1	1	0	0	0
	2	0	0	0	1	1	1	0	0	0
	0	0	0	0	1	1	1	0	0	0
1	1	0	0	0	1	1	1	0	0	0
	2	0	0	0	1	1	1	0	0	0
	0	0	0	0	1	1	1	0	0	0
2	1	0	0	0	1	1	1	0	0	0
	2	0	0	0	1	1	1	0	0	0

(a)

MAX

	w	0			1			2		
y	z \ x	0	1	2	0	1	2	0	1	2
	0	2	2	2	1	1	1	2	2	2
0	1	2	2	2	1	1	1	2	2	2
	2	2	2	2	1	1	1	2	2	2
	0	2	2	2	1	1	1	2	2	2
1	1	2	2	2	1	1	1	2	2	2
	2	2	2	2	1	1	1	2	2	2
	0	2	2	2	1	1	1	2	2	2
2	1	2	2	2	1	1	1	2	2	2
	2	2	2	2	1	1	1	2	2	2

(b)

Application of Fuzzy Logic

- Fuzzy Logic history
- Fuzzy applications
- The Fuzzy World
- Contradiction & Uncertainty
- Vagueness & Doubt
- Multivalued logic & Fuzzy logic
- Membership Degree of belonging
- Membership Degree of membership
- Fuzzy inference
- Fuzzy set theory
- Membership function
- Many membership functions for one variable
- Logic operations (pseudo Boolean algebra)
- Operation examples
- Fuzzification (evaluating the degree of membership)
- Other operations

Application of Fuzzy Logic

- Fuzzy inference rules
- Sum of Products
- Multiple rules
- Defuzzication/inference methods
- Defuzzification action value
- Defuzzification average weighted action value
- Defuzzification TVFI
- Mamdani (COG) defuzzication Cut
- Mamdani (COG) defuzzication aggregate
- Mamdani (COG) defuzzication COG
- Fuzzy calculus & function approximation
- Membership function as interpolation functions (single input variable)
- Membership function as interpolation functions (two input variables)

Application of Fuzzy Logic

- Fuzzy microcontroller: a hardwire approach NeuraLogix NLX230
- Command menu for Neuralogix software
- Truck Parking Exercise
- Fuzzy microcontroller: a hardwire approach VLSI 86C500
- Fuzzy microcontroller: a software approach
- Neuron
- Artificial neuron
- Neural Netowork and Fuzzy Logic
- project using Fide

Application of Fuzzy Logic

- project: temperature & discomfortness control (interpretation of the curves by thermo8)
- Use of the 6805 assembler
- procedure used for the heater box project
- Use of thermo8
- Suggested I/O variables
- I/O variables
- I/O interface
- Lab : Fuzzy logic for Fuzzy problem

Sources

- **Paul Burkey**
- **Weilin Pan**
- **Xuekun Kou**