SAPIENZA
Università di Roma
Dipartimento di Informatica

Minimizing Boolean Expressions

Prof. Daniele Gorla

Minimizing BEs

Minimizing expressions arises from **reducing the number of logic gates** needed to realize a circuit.

This fact has consequences on:

COST: With the technology of medium-scale, high-scale and very high-scale integrated circuits (MSI, LSI and VLSI) this aspect is less crucial. However, the size of the circuit still matters (mostly with the development of nano technologies).

CROSSING TIME: The time to produce an output depends on the number of traversed gates: reducing this number can affect performances.

Minimization

The idea is to repeatedly use the following simplification:

$$axa' + a\overline{x}a' = a(x+\overline{x})a' = aa'$$

where a and a' are any sequence of literals.

Probl.: By only having the truth table, this simplification is not always apparent!

x_3	x_2	x_1	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$\overline{x}_3 x_2 \overline{x}_1 + \overline{x}_3 x_2 x_1 = \overline{x}_3 x_2 (\overline{x}_1 + x_1) = \overline{x}_3 x_2$$

Adjacent 1s correspond to simplifiable minterms

$$\overline{x}_3 x_2 x_1 + x_3 x_2 x_1 = (\overline{x}_3 + x_3) x_2 x_1 = x_2 x_1$$

Simplifiable minterms are NON adjacent 1s

Adjacent 1s correspond to NON simplifiable minterms

Karnaugh Maps

They are a different way of writing truth tables.

These maps sort the elements of $\{0,1\}^n$ in a way that elements at unitary distance (i.e., seuqences that differ just for one bit) are adjacent on the map.

They work well (in a pencil-and-paper fashion) only up-to $n = 4$; for more variables, we need more sophisticated methods (not considered in this course)

To our purposes, BF with more than 4 variables would lead to truth tables that are too big for being developed by hand

KMs for 2 variables functions

SAPIENZA Università di Roma Dipartimento di Informatica

x	y	f
0	0	f_{00}
0	1	f_{01}
1	0	f_{10}
1	1	f_{11}

→

y \ x	0	1
0	f_{00}	f_{10}
1	f_{01}	f_{11}

m_0 must be close to m_1 and m_2

m_1 must be close to m_0 and m_3

m_2 must be close to m_0 and m_3

m_3 must be close to m_1 and m_2

m_0	m_2
m_1	m_3

KMs for 3 variables functions

SAPIENZA Università di Roma Dipartimento di Informatica

x	y	z	f
0	0	0	f_{000}
0	0	1	f_{001}
0	1	0	f_{010}
0	1	1	f_{011}
1	0	0	f_{100}
1	0	1	f_{101}
1	1	0	f_{110}
1	1	1	f_{111}

→

z \ xy	00	01	11	10
0	f_{000}	f_{010}	f_{110}	f_{100}
1	f_{001}	f_{011}	f_{111}	f_{101}

!!!

m_0 must be close to m_1, m_2 and m_4
m_1 must be close to m_0, m_3 and m_5
m_2 must be close to m_0, m_3 and m_6
m_3 must be close to m_1, m_2 and m_7
m_6 must be close to m_2, m_7 and m_4
m_7 must be close to m_3, m_6 and m_5
m_4 must be close to m_0, m_5 and m_6
m_5 must be close to m_1, m_4 and m_7

m_4	m_0	m_2	m_6	m_4
m_5	m_1	m_3	m_7	m_5

↓

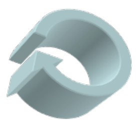
Tridimensional (circular) structure

KMs for 4 variables functions

SAPIENZA Università di Roma Dipartimento di Informatica

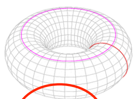
m_0	m_4	m_{12}	m_8
m_1	m_5	m_{13}	m_9
m_3	m_7	m_{15}	m_{11}
m_2	m_6	m_{14}	m_{10}

Longitudinally curved



Traversally curved

It is a *thorus*



x	y	w	z	f
0	0	0	0	f_{0000}
0	0	0	1	f_{0001}
	…			…
1	1	1	1	f_{1111}

→

wz \ xy	00	01	11	10
00	f_{0000}	f_{0100}	f_{1100}	f_{1000}
01	f_{0001}	f_{0101}	f_{1101}	f_{1001}
11	f_{0011}	f_{0111}	f_{1111}	f_{1011}
10	f_{0010}	f_{0110}	f_{1110}	f_{1010}

!!! !!!

KM for the ternary OR

SAPIENZA Università di Roma Dipartimento di Informatica

$OR(a,b,c) = 0$ if and only if $a = b = c = 0$

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

KMs covering and minimal DNFs

An *implicant* corresponds to a rectangle made up by 2^k 1s, i.e. 2^k minterms at unitary distance

An implicant is said *prime* if no other bigger implicant fully contains it

An implicant is said *essential* if it contains at least one 1 that is covered by any other implicant

A *minimal covering* of a KM is a minimal set of prime and essential implicants

A minimal DNF for the BF associated to the KM is built by summing the products associated to a minimal covering

Implicants for the KM of the ternary OR

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

This rectangle is an implicant that represents the minterm $\bar{a}bc$

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

This rectangle is an implicant that represents the sum of minterms $\bar{a}bc + \bar{a}b\bar{c} = \bar{a}b$

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

This rectangle is a prime implicant that represents the sum of minterms

$$\bar{a}bc + \bar{a}b\bar{c} + ab\bar{c} + abc = \bar{a}b + ab = b$$

Identifying the product associated to an implicant

The product is formed by the variables that assume the same value on all the 1s of the rectangle

Every variable is affirmed, if its value is 1 on all the rectangle, negated otherwise

Ex.:

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

1×1 Rectangle, where a=0 & b=c=1 hence, the associated product is m_3

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

2×1 Rectangle, where a=0 & b=1 hence, the associated product is $\bar{a}b$

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

2×2 Rectangle, where b=1 hence, the associated product is b

Prime Implicants for the KM of the ternary OR

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

b

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

c

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

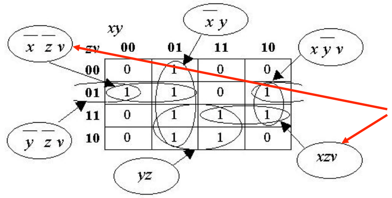
a

Hence, a (actually, the) minimal DNF for this BF is $a+b+c$

Another example

x	y	z	v	o		x	y	z	v	o
0	0	0	0	0		1	0	0	0	0
0	0	0	1	1		1	0	0	1	1
0	0	1	0	0		1	0	1	0	0
0	0	1	1	0		1	0	1	1	1
0	1	0	0	1		1	1	0	0	0
0	1	0	1	1		1	1	0	1	0
0	1	1	0	1		1	1	1	0	1
0	1	1	1	1		1	1	1	1	1

REMARK: all these implicants are prime, but they are NOT all essential



Not essential, by having the other ones

Minimal expressions by using other gates

KMs give minimal BEs *in DNF*

We can obtain BEs with less operators by working on the minimal DNFs,
By using other operators (NOR, NAND, XOR, XNOR)

Ex.:

zt \ xy	00	01	11	10
0 0	1	0	1	1
0 1	0	0	1	1
1 1	1	1	0	0
1 0	1	1	0	1

9 logic gates

$$\bar{x}z + x\bar{z} + \bar{y}\bar{t}$$

$$= (x \oplus z) + \overline{(y+t)}$$

$$= XOR(x,z) + NOR(y,t)$$

3 logic gates

Minimal CNFs

zt \ xy	00	01	11	10
0 0	0	0	0	0
0 1	0	1	1	1
1 1	0	1	1	1
1 0	0	1	1	1

minimal DNF: $yt + xt + yz + xz$

Simplifiable with boolean algebra:

$$yt + xt + yz + xz = (y + x)t + (y + x)z$$
$$= (y + x)(t + z)$$

This is a CNF!!

How can we obtain a minimal CNF from the KM?

Procedure for the minimal POS

1. Cover the 0s with rectangles of size 1, 2, 4, 8 or 16
2. For every rectangle, build the associated sum of literals
   (by duality, we take the negated literal if the variable holds 1, affirmed otherwise)

Ex.:

zt \ xy	00	01	11	10
0 0	0	0	0	0
0 1	0	1	1	1
1 1	0	1	1	1
1 0	0	1	1	1

minimal CNF: $(x + y)(z + t)$

Partial BFs

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

It is possible that the BF is not defined on all $\{0,1\}^n$ but only on some n-tuples

In this case, the n-tuples where it is NOT defined can be associated to any boolean value, so that the DNF (or CNF) becomes as small ss possible

Ex.:

x y z	f
0 0 0	0
0 0 1	0
0 1 0	1
0 1 1	0
1 0 0	1
1 0 1	0
1 1 0	-
1 1 1	-

z \ $x\,y$	00	01	11	10
0	0	1	-	1
1	0	0	-	0

We should consider as 1 and as 0

We know that x and y cannot be simultaneously at 1