SYMMETRY IN BOOLEAN FUNCTIONS
WITH EXAMPLES FOR TWO AND THREE VARIABLES

William Bricken
March 1997
Simplified, with figures added April 2002

Boolean function space is a mathematical abstraction of all possible combinational circuitry. Circuits have numerous forms that are functionally invariant. Functions themselves can be abstracted into groups with specific symmetries (this is the intent of group theory). The space of 2^2^n Boolean functions of n variables has symmetries that can reduce the number of functional entities needed to fully describe this space. In an experiment, I used algorithms to generate all possible Boolean functions (n < 5), then to reduce this space by identifying symmetries. The goal is to provide abstract manipulation and optimization tools for logic synthesis. FPGA architectures often incorporate four-input look-up tables (LUTs), so that the study of Boolean functions with (n < 5) is directly relevant to FPGA optimization.

The set of n-variable Boolean functions can be generated by Boolean cubes, by the Reed-Muller expansion (aka Binary Decision Diagrams), or by structural approaches. All of these approaches yield highly redundant forms. Given the goal of minimizing the form of each function, the challenge is to find a representation which both covers and abstracts.

The available symmetries for 2 and 3 variable Boolean functions are crisply summarizing by a variety of abstract Boolean *spatial* forms. The approach that yields the clearest insights is to use physical cubes in three-space to represent 3-variable functions. The possible combinations of zero to eight cubes in a 2x2x2 matrix yield the non-reduced function space. Whenever two or more cubes touched faces, a reduction was available. The morphic algebraic approach does not provide as clear an indication, since the algebraic forms have many varieties for each function. These varieties are suppressed in the cube physical model through the availability of (trivial) spatial rotation.

Here is a Table of the relations between the number of inputs, functions, and spatially-abstracted functional forms:

Inputs/variables	*Logical Functions*	*Spatial Forms*
0	2	1
1	4	2
2	16	4
3	256	14
4	65536	222
5	4294967296	616126
6	2^64	?

It is not unusual for circuits to have over 200 inputs.

There are 14 unique 3-variable spatial form abstractions which summarize 256 different three-variable functions (which in turn can be represented by an infinite variety of non-reduced structures). These can be classified by algebraic symmetries, by structural symmetries in 3-space, and by types of touching between the cubes in the form.

Technical note: The number of spatial forms of n variables can be determined using the Polya-Burnside group theoretic enumeration technique for counting the orbits of a set under the action of symmetry groups. Spatial forms are the set of unique block configurations. Group theoretic forms are the non-equivalent functions under axis-name permutation and complementation of variables and spaces (i.e. inputs and outputs). As best as I can determine, these techniques do not provide information about face touchings, and thus about potential minimization of algebraic forms.

Attachment I: Two variable Boolean functions
The availability and types of functional symmetry in an algebraic format.

Attachment II: Two-variable Boolean hypercube
The availability and types of functional symmetry in a geometric format.

Attachment III: Three variable Boolean functions
Functional symmetries in an algebraic format.

Attachment IV: Three variable Boolean cubes
A spatial interpretation and simplification of three variable functions.

Attachment V: Counts of various types of forms

Attachment VI: A listing of he three variable forms.

Attachment I: The two variable Boolean functions

The 16 two-variable Boolean functions expressed in parens notation:

NIL	(NIL)
x	(x)
y	(y)
x y	(x y)
x=y	(x=y)
x (y)	(x (y))
y (x)	(y (x))
(x)(y)	((x)(y))

Using symmetries, these 16 functional varieties reduce to four abstract forms.

Inverse function symmetry: negation of the entire function. Function negation removes half of the entities, under the rule: Given F, add (F). This generates either of the single columns directly above.

Inverse variable symmetry: negation of each variable. The notation ":x" indicates the literal x, either x or (x). Note that although this abstraction also reduces the space by half, it interacts with the previous symmetry of negating the entire function, since the variable negations are included in the function negations.

```
 NIL
 :x
:x=:y
 :y
:x :y      represents four forms:    x  y      (x) y
                                     x (y)     (x)(y)
```

N-variable symmetry: abstraction of variable names. A indicates either x or y. In the presence of A, B indicates the other variable.

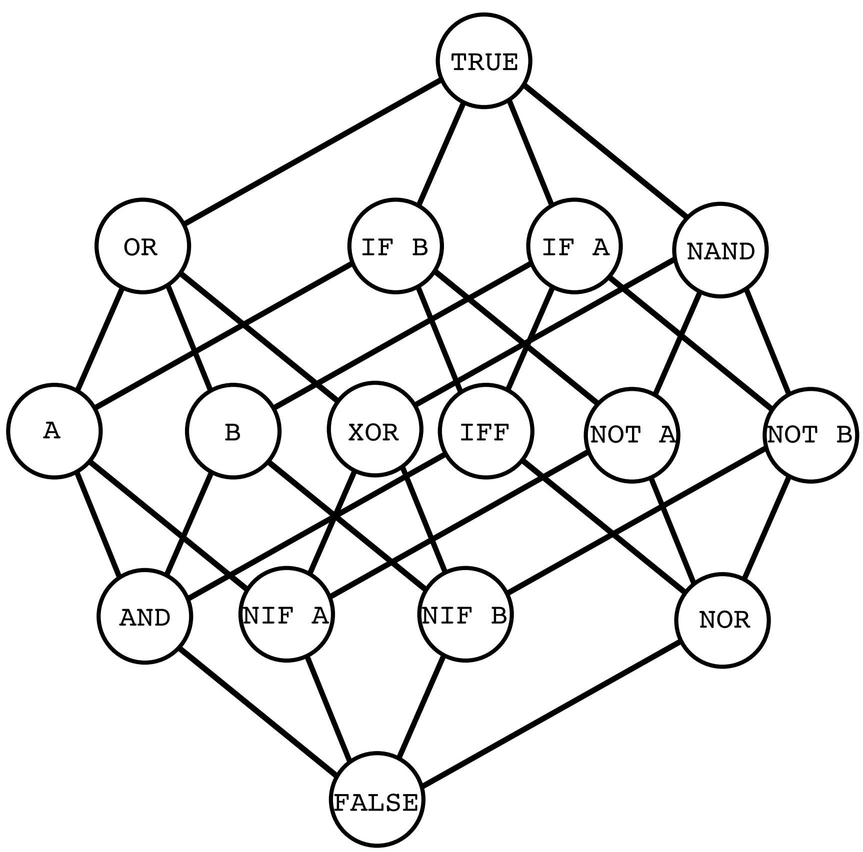
```
 NIL
 :A
:A=:B
:A :B
```

Dual function symmetry: DeMorgan transformation of the space. This exchanges NIL for (NIL) while at the same time exchanging space for bounding. In logical terms, DeMorgan exchanges TRUE for FALSE, and AND for OR. The DeMorgan transform is: bound every variable in a space, and bound the entire space; then reduce through Involution ((A)) = A. DeMorgan transformation turns the Boolean space upside-down (rotates it by 180 degrees), but does not provide an abstraction of forms.

Attachment II: Two variable functions as a 4D Boolean hypercube

A Boolean hypercube is a distributed, complemented lattice. Two-variable functions form a 4D hypercube with 4 different projections into 3D. The projections can be visualized by forming 3D cubes using triples of connections to the TRUE node. Numbering these connections 1 through 4 from left to right yields four triples. Each connection in a triple forms a different axis of its 3D projection cube. The unincorporated nodes form the complementary 3D projection cube, which has a symmetric connection triple with the FALSE node.

Connections	*Projection Space*	*Symmetry*
1-2-3 (TRUE OR IFB IFA A B IFF AND)	unbounded function	inverse function
1-2-4 (TRUE OR IFB NAND A XOR NOTA NIFA)	variable A	N-variable
1-3-4 (TRUE OR IFA NAND B XOR NOTB NIFB)	variable B	N-variable
2-3-4 (TRUE IFB IFA NAND IFF NOTA NOTB NOR)	bounded function	inverse variable



Attachment III: Algebraic abstraction of three-variable functions

The topology of each distinct abstract Boolean function of three variables can be characterized by boundary logic algebraic structures and by Boolean cube formations in three-space (See Attachment IV).

NOTATION

( )	parens are logic/connectives/boundaries
x	names a variable in the positive state, x=1
(x)	names a variable in the negative state, x=0
:x	identifies a literal, either positive or negative
:x=y	refers to bounding the entire equality
(x y)	concatenation in bounded space is OR
%x	identifies a second occurrence of a variable, for which it must take on the opposite value of its paired literal
+x	identifies a second occurrence of a variable; when the first occurrence is in one specific form, +x is bound to other form

Abstract form	*Physical form*	*Touching form*
()	solid	meaningless, tautology
(x y z)	cube	0 origin
(:x :y)	face-pair	face
(:x :y=z)	edge-pair	edge
((x :y :z)((x) %y %z))	point-pair	point
(:x (:y :z))	face-face-pairs	face-face-edge
((:x :y) (%x %y :z))	face-edge-pairs	face-edge-point
((:x y=z) (%x +:y +:z))	edge-edge-pairs	edge-edge-edge
(x)	four-plane	4face-2edge
((:y :z) (:x (%y %z)))	tetrahedron	3face-3edge
((:x :y) (%x :z))	three-way-L	3face-2edge-point
((x :(:y :z)) (%x %(:y :z)))	L+not-fill	2face-3edge-point
(x=y)	equal-pairs	2face-2edge-2point
(x==y=z)	four-corners	6edge

This description differentiates

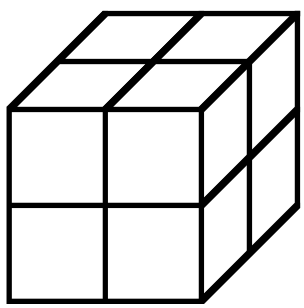
1) the count of types of block touching (faces, edges, points)
2) the 14 core 3-variable functions.
3) the available types of simplification through symmetries:
   faces eliminate a variable reference
   edges introduce a partial equals, nor/and
   points hold up structure, no reduction

Attachment IV: Fourteen 3-variable forms as freely rotated cubes

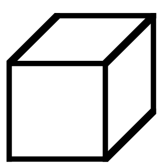
Each single cube in a 2x2x2 space represents a unique combination of different polarities of three different variables. Free rotation of the Boolean 3space cube removes the polarity distinctions of each variable, abstracting the function space by a factor of eight. Cubes that touch faces identify reductions in the algebraic form of the cubes. Touching edges identify varieties of the 2 variable XOR function embedded in 3 variable cube space.

Below, boundary logic and physical descriptions are followed by a 3-space visualization with free rotation. The boundary logic forms are reduced from the Boolean cube forms, to indicate the effect of face touching.

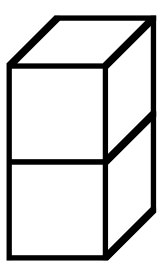
Boundary Logic reduced form	*Physical description* *Logical example*
()	solid, ground TRUE



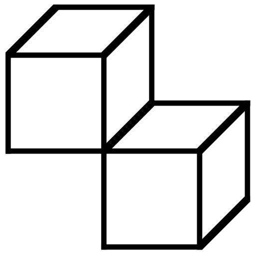
(:x :y :z)	small cube (AND x y z)



(:x :y)	face-pair (AND x y)

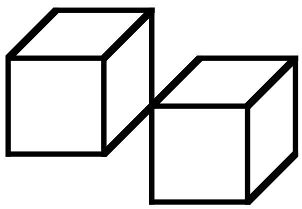


(:x :y=z)



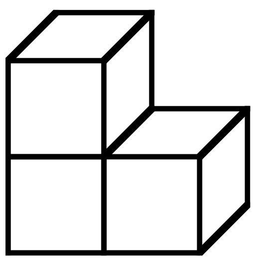
edge-pair
(AND x y=z)

((x :y :z)((x) %y %z))



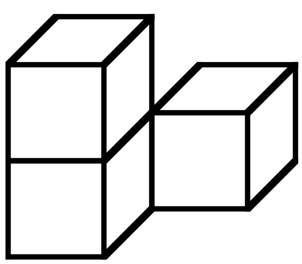
point-pair
(AND (OR x y z)
(OR (NOT x)(NOT y)(NOT z)))

(:x (:y :z))



face-face-pairs
(AND x (OR y z))

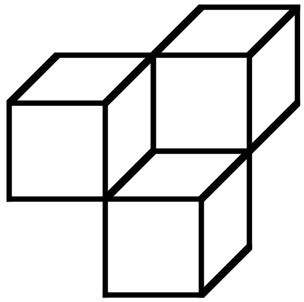
((:x :y) (%x %y :z))



face-edge-pairs, part-equal
(AND (OR x y)
(OR (NOT x)(NOT y) z))

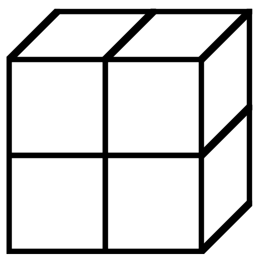
((:x y=z) (%x +:y +:z))

edge-edge-pairs



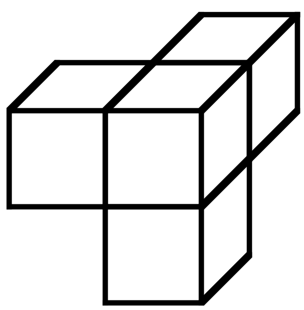
(x)

four-plane, literal
(NOT x)



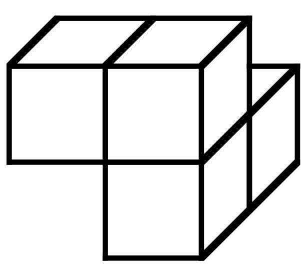
((:y :z) (:x (%y %z)))

tetrahedron, 2/3 majority
(AND (OR y z)
(OR x (AND y z)))



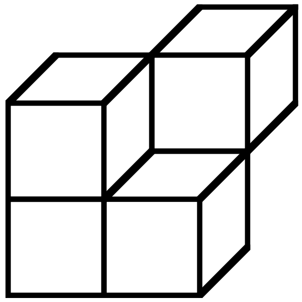
((:x :y) (%x :z))

three-way-L, if-then-else
(IF x THEN y ELSE z)



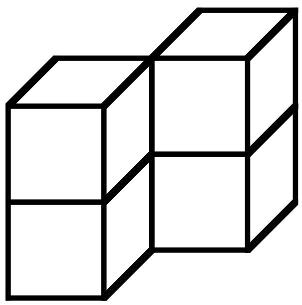
((x :(:y :z)) (%x %(:y :z)))

L+not-fill
(AND (OR x (AND y z))
(OR (NOT x) y z))



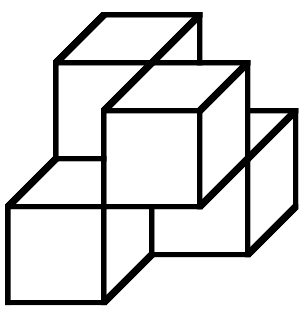
(x=y)

equal-pairs
(XOR x y)



(x==y=z)

four-corners
((XOR x y) XOR z)



Attachment V: Counts of various types of forms

n-variable Boolean functions:

2^2^n = sum(0 to i=2^n)[choose i 2^n]

n-variable spatial forms

N	BLOCKS		#functions	total	spatial types
0	0/1		2		
total				2	1
1	0/2		2		
	1/1		2		
total				4	2
2	0/4		2		
	1/3		8		
	2/2	face	4		
		point	2		
total				16	4
3	0/8		2		
	1/7		16		
	2/6	face	24		
		edge	24		
		point	8		
	3/5	2face	48		
		1face	48		
		0face	16		
	4/4	4face	6		
		3face-1	8		
		3face-2	24		
		2face-1	24		
		2face-2	6		
		0face	2		
total				256	14

note:

3face-n = n blocks participate in the shared faces

Attachment VI: The 256 three variable Boolean functions

Symmetry simplifications:

compliments omitted, equalities abstracted, boundary forms reduced

```
(NIL)       (A)         (B)         (C)

( A  B )    ( A  C )    ( B  C )
( A (B))    ( A (C))    ( B (C))    (B (A))     (C (A))     (C (B))
((A)(B))    ((A)(C))    ((B)(C))
( A=B  )    ( A=C  )    ( B=C  )

(A  B  C )  (A  B (C))  (A  C (B))  ( B  C (A))
(A (B)(C))  (B (A)(C))  (C (A)(B))  ((A)(B)(C))

( A ( B  C ))      ( B ( A  C ))      ( C ( A  B ))
( A ( B (C)))      ( B ( A (C)))      ( C ( A (B)))
( A ( C (B)))      ( B ( C (A)))      ( C ( B (A)))
( A ((B)(C)))      ( B ((A)(C)))      ( C ((A)(B)))
( A    B=C  )      ( B    A=C  )      ( C    A=B  )
( A   (B=C) )      ( B   (A=C) )      ( C   (A=B) )
((A)( B  C ))      ((B)( A  C ))      ((C)( A  B ))
((A)( B (C)))      ((B)( A (C)))      ((C)( A (B)))
((A)( C (B)))      ((B)( C (A)))      ((C)( B (A)))
((A)((B)(C)))      ((B)((A)(C)))      ((C)((A)(B)))
((A)   B=C  )      ((B)   A=C  )      ((C)   A=B  )
((A)  (B=C) )      ((B)  (A=C) )      ((C)  (A=B) )

((A (B)) ( C (A)))      ((B (A)) ( C (B)))      ((C (A)) ((B)(C)))
((A (B)) ((A)(C)))      ((B (A)) ((B)(C)))      ((C (B)) ((A)(C)))
((A (C)) ( B (A)))      ((B (C)) ( C (A)))
((A (C)) ( C (B)))      ((B (C)) ((A)(B)))
((A (C)) ((A)(B)))

(( A  B ) (C (A)(B)))   (( A  B ) ((A)(B)(C)))
(( A (B)) (B  C (A)))   (( A (B)) ( B (A)(C)))
(( B (A)) (A  C (B)))   (( B (A)) ( A (B)(C)))
(((A)(B)) (A  B  C ))   (((A)(B)) ( A  B (C)))
(( A  C ) (B (A)(C)))   (( A  C ) ((A)(B)(C)))
(( A (C)) (B  C (A)))   (( A (C)) ( C (A)(B)))
(( C (A)) (A  B (C)))   (( C (A)) ( A (B)(C)))
(((A)(C)) (A  B  C ))   (((A)(C)) ( A  C (B)))
(( B  C ) (A (B)(C)))   (( B  C ) ((A)(B)(C)))
(( B (C)) (A  C (B)))   (( B (C)) ( C (A)(B)))
(((B)(C)) (A  B  C ))   (((B)(C)) ( B  C (A)))
(( C (B)) (A  B (C)))   (( C (B)) ( B (A)(C)))
```

```
((A (B)) (A (C)) ( B (C)))    ((B (A)) (B (C)) ((A)(C)))
((A (B)) (A (C)) ((B)(C)))    ((C (A)) (C (B)) ((A)(B)))

((A B  C ) ((A)(B)(C)))       ((A  B (C)) (C (A)(B)))
((A C (B)) ( B (A)(C)))       ((A (B)(C)) (B  C (A)))

((A B (C)) ( C (A  B )))  ((A C (B)) ( B (A  C )))  ((B C (A)) ( A (B  C )))
((A B (C)) ((A)(B (C))))  ((A C (B)) ((A)(C (B))))  ((B C (A)) ((B)(C (A))))
((A B (C)) ((B)(A (C))))  ((A C (B)) ((C)(A (B))))  ((B C (A)) ((C)(B (A))))

((A (B)(C)) ((A)((B)(C))))
((B (A)(C)) ((B)((A)(C))))
((C (A)(B)) ((C)((A)(B))))

(((B)(C)) (B (A)(C)) (C (A)(B)))

((A B C) (A (B)(C)) (B (A)(C)))     ((A B (C)) (A C (B)) (B C (A)))
((A B C) (A (B)(C)) (C (A)(B)))     ((A B (C)) (A C (B)) ((A)(B)(C)))
((A B C) (B (A)(C)) (C (A)(B)))     ((A B (C)) (B C (A)) ((A)(B)(C)))

((A C (B)) (B C (A)) ((A)(B)(C)))   ((A (B)(C)) (B (A)(C)) (C (A)(B)))

((A B (C)) (A C (B)) (B C (A)) ((A)(B)(C)))
```

NOTE: The symmetries of these 128 forms have not been developed rigorously in the above display.