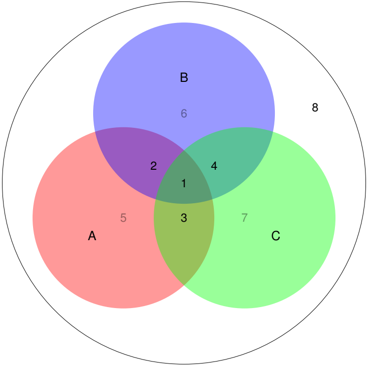
Contents

Section outline

Sum of products from sets

Regions

1. $A \cap B \cap C$
2. $A \cap B \cap \overline{C}$
3. $A \cap \overline{B} \cap C$
4. $\overline{A} \cap B \cap C$
5. $A \cap \overline{B} \cap \overline{C}$
6. $\overline{A} \cap B \cap \overline{C}$
7. $\overline{A} \cap \overline{B} \cap C$
8. $\overline{A} \cap \overline{B} \cap \overline{C}$

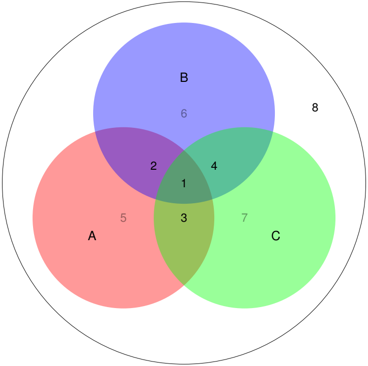


Selections

Sum of products from sets

Regions

1. $A \cap B \cap C$
2. $A \cap B \cap \overline{C}$
3. $A \cap \overline{B} \cap C$
4. $\overline{A} \cap B \cap C$
5. $A \cap \overline{B} \cap \overline{C}$
6. $\overline{A} \cap B \cap \overline{C}$
7. $\overline{A} \cap \overline{B} \cap C$
8. $\overline{A} \cap \overline{B} \cap \overline{C}$



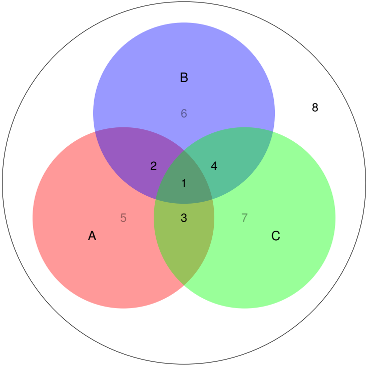
Selections

1∧2 $A \cap B$

Sum of products from sets

Regions

1. $A \cap B \cap C$
2. $A \cap B \cap \overline{C}$
3. $A \cap \overline{B} \cap C$
4. $\overline{A} \cap B \cap C$
5. $A \cap \overline{B} \cap \overline{C}$
6. $\overline{A} \cap B \cap \overline{C}$
7. $\overline{A} \cap \overline{B} \cap C$
8. $\overline{A} \cap \overline{B} \cap \overline{C}$



Selections

1∧2 $A \cap B$

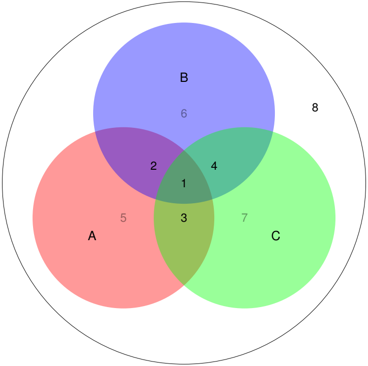
$(A \cap B \cap C) \cup$
$(A \cap B \cap \overline{C})$
$abc + ab\overline{c} = ab$

1∧2∧3∧5 A

Sum of products from sets

Regions

1. $A \cap B \cap C$
2. $A \cap B \cap \overline{C}$
3. $A \cap \overline{B} \cap C$
4. $\overline{A} \cap B \cap C$
5. $A \cap \overline{B} \cap \overline{C}$
6. $\overline{A} \cap B \cap \overline{C}$
7. $\overline{A} \cap \overline{B} \cap C$
8. $\overline{A} \cap \overline{B} \cap \overline{C}$



Selections

1∧2 $A \cap B$

$(A \cap B \cap C) \cup$
$(A \cap B \cap \overline{C})$
$abc + ab\overline{c} = ab$

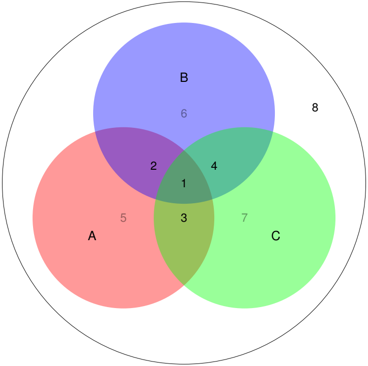
1∧2∧3∧5 A

$(A \cap B \cap C) \cup$
$(A \cap B \cap \overline{C}) \cup$
$(A \cap \overline{B} \cap C) \cup$
$(A \cap \overline{B} \cap \overline{C})$
$abc + ab\overline{c} + a\overline{b}c +$
$a\overline{b}\overline{c}$

Sum of products from sets

Regions

1. $A \cap B \cap C$
2. $A \cap B \cap \overline{C}$
3. $A \cap \overline{B} \cap C$
4. $\overline{A} \cap B \cap C$
5. $A \cap \overline{B} \cap \overline{C}$
6. $\overline{A} \cap B \cap \overline{C}$
7. $\overline{A} \cap \overline{B} \cap C$
8. $\overline{A} \cap \overline{B} \cap \overline{C}$



Selections

1∧2 $A \cap B$

$(A \cap B \cap C) \cup$
$(A \cap B \cap \overline{C})$
$abc + ab\overline{c} = ab$

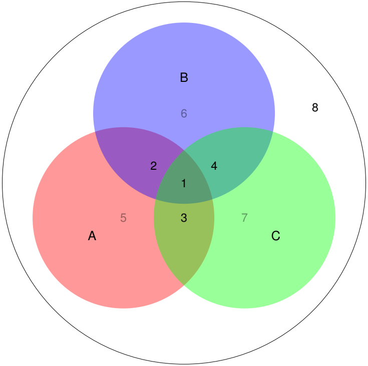
1∧2∧3∧5 A

$(A \cap B \cap C) \cup$
$(A \cap B \cap \overline{C}) \cup$
$(A \cap \overline{B} \cap C) \cup$
$(A \cap \overline{B} \cap \overline{C})$
$abc + ab\overline{c} + a\overline{b}c +$
$a\overline{b}\overline{c} = ab + a\overline{b} = a$

Sum of products from sets

Regions

1. $A \cap B \cap C$
2. $A \cap B \cap \overline{C}$
3. $A \cap \overline{B} \cap C$
4. $\overline{A} \cap B \cap C$
5. $A \cap \overline{B} \cap \overline{C}$
6. $\overline{A} \cap B \cap \overline{C}$
7. $\overline{A} \cap \overline{B} \cap C$
8. $\overline{A} \cap \overline{B} \cap \overline{C}$



Selections

1∧2 $A \cap B$

$(A \cap B \cap C) \cup$
$(A \cap B \cap \overline{C})$
$abc + ab\overline{c} = ab$

1∧2∧3∧5 A

$(A \cap B \cap C) \cup$
$(A \cap B \cap \overline{C}) \cup$
$(A \cap \overline{B} \cap C) \cup$
$(A \cap \overline{B} \cap \overline{C})$
$abc + ab\overline{c} + a\overline{b}c +$
$a\overline{b}\overline{c} = ab + a\overline{b} = a$

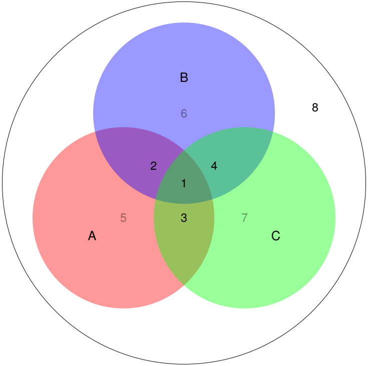
a I have an item from A

$\overline{a}$ I don't have an item from A

Sum of products from sets

Regions

1. $A \cap B \cap C$
2. $A \cap B \cap \overline{C}$
3. $A \cap \overline{B} \cap C$
4. $\overline{A} \cap B \cap C$
5. $A \cap \overline{B} \cap \overline{C}$
6. $\overline{A} \cap B \cap \overline{C}$
7. $\overline{A} \cap \overline{B} \cap C$
8. $\overline{A} \cap \overline{B} \cap \overline{C}$



Selections

1∧2 $A \cap B$

$(A \cap B \cap C) \cup$
$(A \cap B \cap \overline{C})$
$abc + ab\bar{c} = ab$

1∧2∧3∧5 A

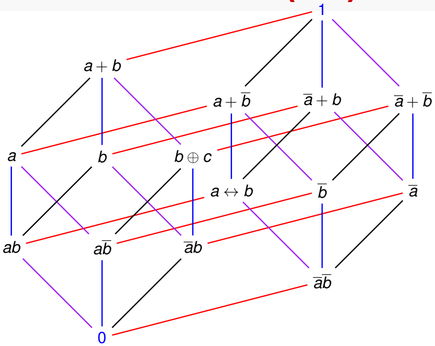
$(A \cap B \cap C) \cup$
$(A \cap B \cap \overline{C}) \cup$
$(A \cap \overline{B} \cap C) \cup$
$(A \cap \overline{B} \cap \overline{C})$
$abc + ab\bar{c} + a\bar{b}c +$
$a\bar{b}\bar{c} = ab + a\bar{b} = a$

a I have an item from A

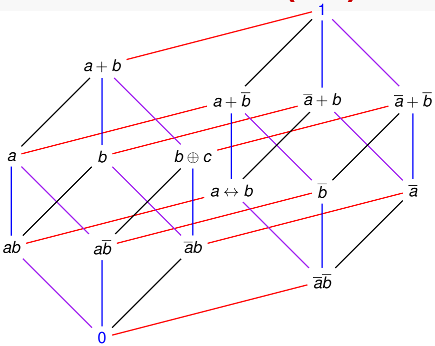
$\bar{a}$ I don't have an item from A

$a\bar{b} + c$ I have an item from A but not from B or an item from C

Boolean lattice (BL) for 2 variables

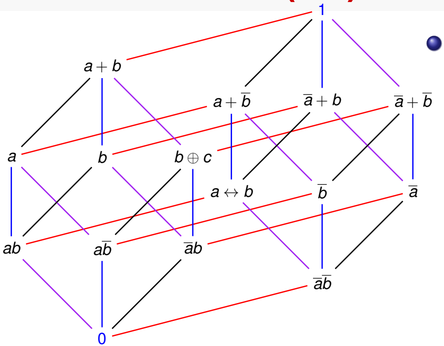


Boolean lattice (BL) for 2 variables



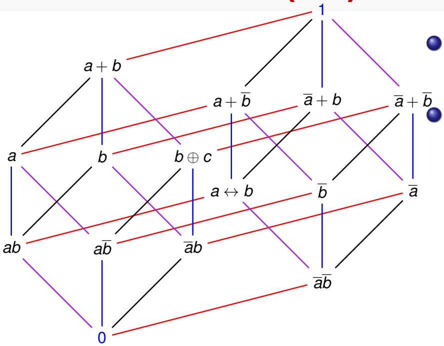
- A *literal* is a variable (a) or its complement ($\overline{a}$)
- A Boolean expression is a string built from literals and the Boolean operators without violating their arity
- Grouping with parentheses is permitted

Boolean lattice (BL) for 2 variables



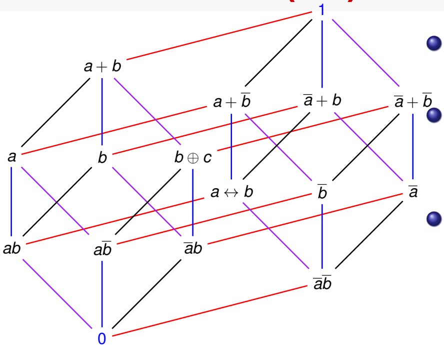
- A *literal* is a variable (a) or its complement ($\overline{a}$)
- A Boolean expression is a string built from literals and the Boolean operators without violating their arity
- Grouping with parentheses is permitted
- Such an expression is *well formed* or syntactically correct

Boolean lattice (BL) for 2 variables



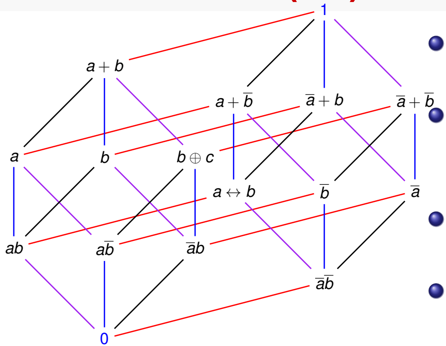
- A *literal* is a variable (a) or its complement ($\overline{a}$)
- A Boolean expression is a string built from literals and the Boolean operators without violating their arity
- Grouping with parentheses is permitted
- Such an expression is *well formed* or syntactically correct
- A fundamental product (FP) is a literal or a product of two or more literals arising from distinct variables

Boolean lattice (BL) for 2 variables



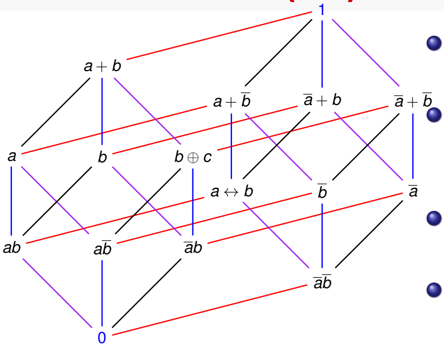
- A *literal* is a variable (a) or its complement ($\overline{a}$)
- A Boolean expression is a string built from literals and the Boolean operators without violating their arity
- Grouping with parentheses is permitted
- Such an expression is *well formed* or syntactically correct
- A fundamental product (FP) is a literal or a product of two or more literals arising from distinct variables
- A FP involving all the variables is a *minterm* – atoms in the BL

Boolean lattice (BL) for 2 variables



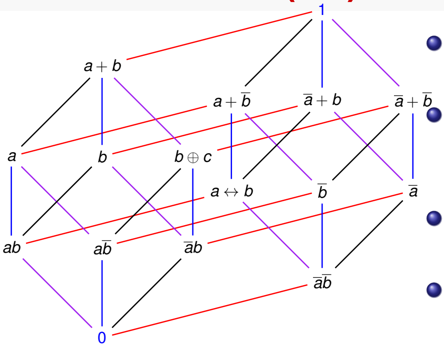
- A *literal* is a variable (a) or its complement ($\overline{a}$)
- A Boolean expression is a string built from literals and the Boolean operators without violating their arity
- Grouping with parentheses is permitted
- Such an expression is *well formed* or syntactically correct
- A fundamental product (FP) is a literal or a product of two or more literals arising from distinct variables
- A FP involving all the variables is a *minterm* – atoms in the BL
- A FP P_1 is contained or included in P_2 if P_2 has all the literals of P_1; then $P_2 \Rightarrow P_1$ (P_2 implies P_1)

Boolean lattice (BL) for 2 variables



- A *literal* is a variable (a) or its complement ($\overline{a}$)
- A Boolean expression is a string built from literals and the Boolean operators without violating their arity
- Grouping with parentheses is permitted
- Such an expression is *well formed* or syntactically correct
- A fundamental product (FP) is a literal or a product of two or more literals arising from distinct variables
- A FP involving all the variables is a *minterm* – atoms in the BL
- A FP P_1 is contained or included in P_2 if P_2 has all the literals of P_1; then $P_2 \Rightarrow P_1$ (P_2 implies P_1)
- A *sum of products* (SOP) expression is FP or a sum of two or more FPs $P_1, \ldots, P_n$ and $\forall i, j, P_i \not\Rightarrow P_j$

Boolean lattice (BL) for 2 variables



- A *literal* is a variable (a) or its complement ($\overline{a}$)
- A Boolean expression is a string built from literals and the Boolean operators without violating their arity
- Grouping with parentheses is permitted
- Such an expression is *well formed* or syntactically correct
- A fundamental product (FP) is a literal or a product of two or more literals arising from distinct variables
- A FP involving all the variables is a *minterm* – atoms in the BL
- A FP P_1 is contained or included in P_2 if P_2 has all the literals of P_1; then $P_2 \Rightarrow P_1$ (P_2 implies P_1)
- A *sum of products* (SOP) expression is FP or a sum of two or more FPs $P_1, \ldots, P_n$ and $\forall i, j, P_i \not\Rightarrow P_j$
- DeMorgan's laws, distributivity, commutativity, idempotence, involution may be used to transform a Boolean expression to SOP

Functional completeness

- May be derived from the Boolean lattice

Functional completeness

- May be derived from the Boolean lattice
- OR is required to compute the joins on the elements

Functional completeness

- May be derived from the Boolean lattice
- OR is required to compute the joins on the elements
- NOT and AND are required to compute the atoms from the proposition variables

x	y	$\overline{x}$	$x \cdot y$	$x + y$
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

Functional completeness

- May be derived from the Boolean lattice
- OR is required to compute the joins on the elements
- NOT and AND are required to compute the atoms from the proposition variables

x	y	$\overline{x}$	$x \cdot y$	$x + y$
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

Functional completeness

- May be derived from the Boolean lattice
- OR is required to compute the joins on the elements
- NOT and AND are required to compute the atoms from the proposition variables

x	y	$\overline{x}$	$x \cdot y$	$x + y$
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

NAND $\overline{x \cdot y}$

Functional completeness

- May be derived from the Boolean lattice
- OR is required to compute the joins on the elements
- NOT and AND are required to compute the atoms from the proposition variables

x	y	$\overline{x}$	$x \cdot y$	$x + y$
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

NAND $\overline{x \cdot y}$
NOR $\overline{x + y}$

Functional completeness

- May be derived from the Boolean lattice
- OR is required to compute the joins on the elements
- NOT and AND are required to compute the atoms from the proposition variables

x	y	$\overline{x}$	$x \cdot y$	$x + y$
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

NAND $\overline{x \cdot y}$

NOR $\overline{x + y}$

XOR,AND $x \oplus y, x \cdot y$

Functional completeness

- May be derived from the Boolean lattice
- OR is required to compute the joins on the elements
- NOT and AND are required to compute the atoms from the proposition variables

x	y	$\overline{x}$	$x \cdot y$	$x + y$
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

NAND $\overline{x \cdot y}$
NOR $\overline{x + y}$
XOR,AND $x \oplus y, x \cdot y$
MUX $s \cdot x + \overline{s} \cdot y$

Functional completeness

- May be derived from the Boolean lattice
- OR is required to compute the joins on the elements
- NOT and AND are required to compute the atoms from the proposition variables

x	y	$\overline{x}$	$x \cdot y$	$x + y$
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

NAND $\overline{x \cdot y}$
NOR $\overline{x + y}$
XOR,AND $x \oplus y, x \cdot y$
MUX $s \cdot x + \overline{s} \cdot y$
RAM Random access memory

Functional completeness

- May be derived from the Boolean lattice
- OR is required to compute the joins on the elements
- NOT and AND are required to compute the atoms from the proposition variables

x	y	$\overline{x}$	$x \cdot y$	$x + y$
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

NAND $\overline{x \cdot y}$

NOR $\overline{x + y}$

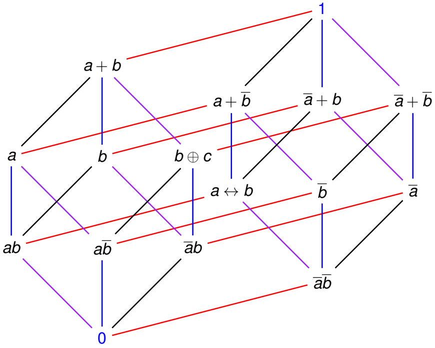
XOR,AND $x \oplus y, x \cdot y$

MUX $s \cdot x + \overline{s} \cdot y$

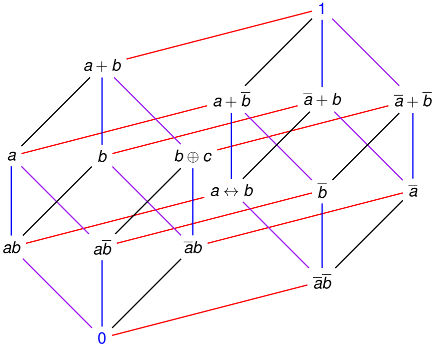
RAM Random access memory

Minority Minority value among given inputs

Boolean expressions

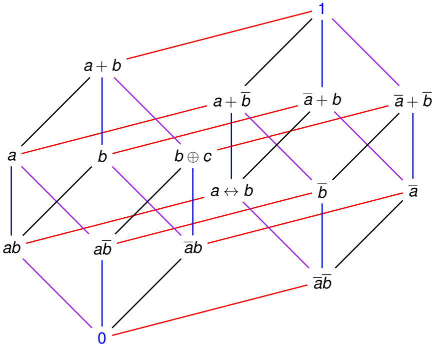


Boolean expressions



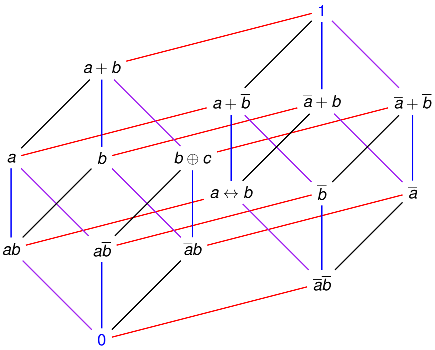
- $E = x\overline{z} + \overline{y}z + xy\overline{z}$

Boolean expressions



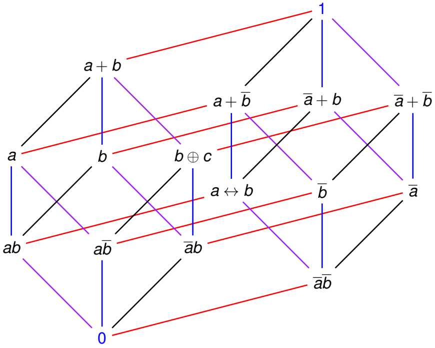
- $E = x\overline{z} + \overline{y}z + xy\overline{z}$
- $E = \overline{(\overline{(xy)}z)}\,\overline{((\overline{x}+z)(\overline{y}+\overline{z}))}$

Boolean expressions



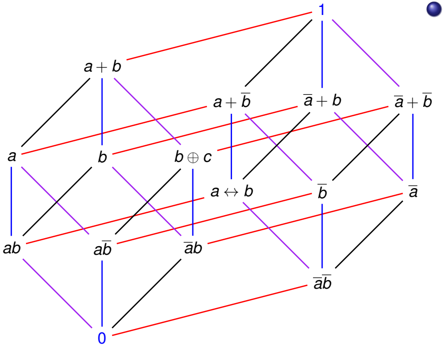
- $E = x\overline{z} + \overline{y}z + xy\overline{z}$
- $E = \overline{(\overline{(xy)}z)}\,\overline{((\overline{x}+z)(\overline{y}+\overline{z}))}$
- $E = x\overline{(\overline{y}z)}$

Boolean expressions



- $E = x\overline{z} + \overline{y}z + xy\overline{z}$
- $E = \overline{(\overline{(xy)}z)}\,\overline{((\overline{x}+z)(\overline{y}+\overline{z}))}$
- $E = x\overline{(\overline{y}z)}$

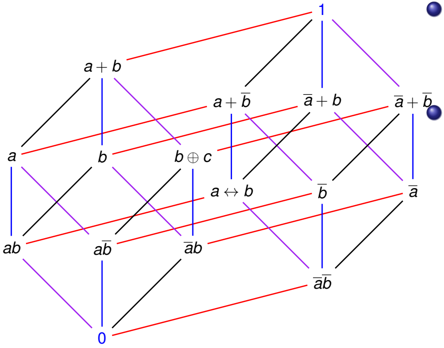
Boolean expressions



- $E = x\bar{z} + \bar{y}z + xy\bar{z}$
- $E = \overline{(\overline{(xy)}z)\overline{((\bar{x} + z)(\bar{y} + \bar{z}))}}$
- $E = x(\overline{\bar{y}z})$

- A SOP expression where each FP is a minterm is said to be in *disjunctive normal form* (DNF)

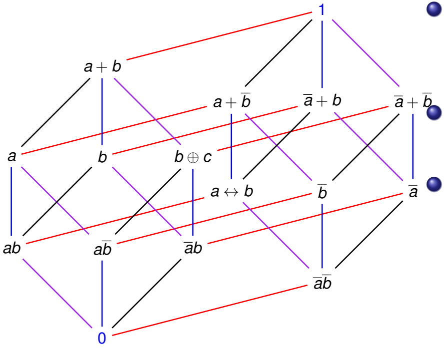
Boolean expressions



- A SOP expression where each FP is a minterm is said to be in *disjunctive normal form* (DNF)
- The DNF of any SOP is unique (why?) – cannonical SOP

- $E = x\overline{z} + \overline{y}z + xy\overline{z}$
- $E = \overline{(\overline{(xy)}z)}\,\overline{((\overline{x}+z)(\overline{y}+\overline{z}))}$
- $E = x\overline{(\overline{y}z)}$

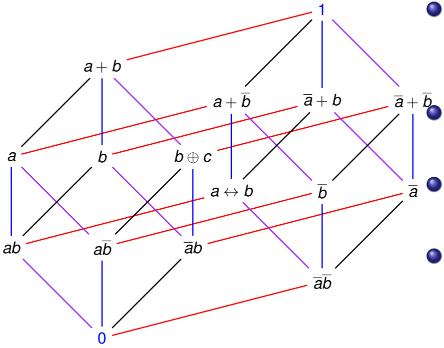
Boolean expressions



- A SOP expression where each FP is a minterm is said to be in *disjunctive normal form* (DNF)
- The DNF of any SOP is unique (why?) – cannonical SOP
- An element x in a BL is *maxterm* if it has 1 as its only successor

- $E = x\overline{z} + \overline{y}z + xy\overline{z}$
- $E = \overline{(\overline{(xy)}z)\overline{((\overline{x}+z)(\overline{y}+\overline{z}))}}$
- $E = x\overline{(\overline{y}z)}$

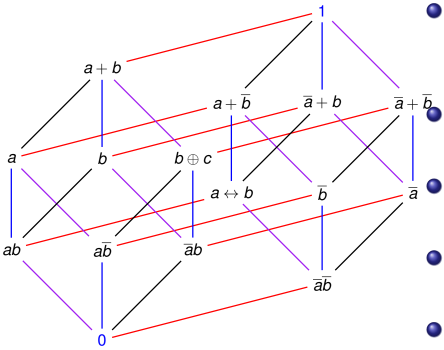
Boolean expressions



- $E = x\overline{z} + \overline{y}z + xy\overline{z}$
- $E = \overline{((\overline{xy})z)}\,\overline{((\overline{x}+z)(\overline{y}+\overline{z}))}$
- $E = x\overline{(\overline{y}z)}$

- A SOP expression where each FP is a minterm is said to be in *disjunctive normal form* (DNF)
- The DNF of any SOP is unique (why?) – cannonical SOP
- An element x in a BL is *maxterm* if it has 1 as its only successor
- A maxterm is a sum of literals involving all the variables

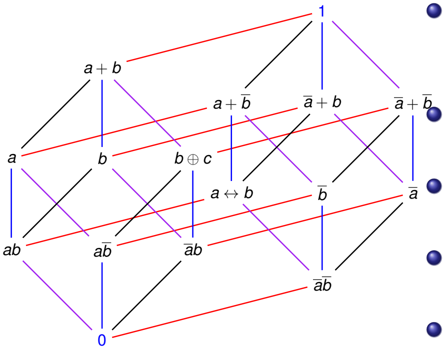
Boolean expressions



- $E = x\overline{z} + \overline{y}z + xy\overline{z}$
- $E = \overline{((\overline{xy})z)}\,\overline{((\overline{x}+z)(\overline{y}+\overline{z}))}$
- $E = x(\overline{\overline{y}z})$

- A SOP expression where each FP is a minterm is said to be in *disjunctive normal form* (DNF)
- The DNF of any SOP is unique (why?) – cannonical SOP
- An element x in a BL is *maxterm* if it has 1 as its only successor
- A maxterm is a sum of literals involving all the variables
- Similar to SOP, *product of sums* (POS) may be defined

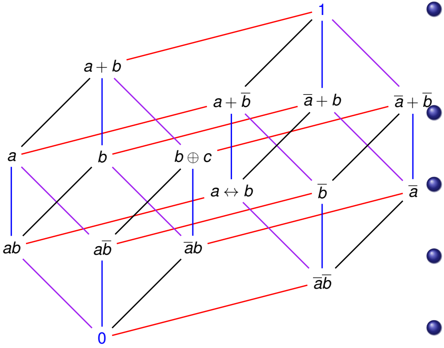
Boolean expressions



- $E = x\overline{z} + \overline{y}z + xy\overline{z}$
- $E = \overline{((\overline{xy})z)}\,\overline{((\overline{x}+z)(\overline{y}+\overline{z}))}$
- $E = x(\overline{\overline{y}z})$

- A SOP expression where each FP is a minterm is said to be in *disjunctive normal form* (DNF)
- The DNF of any SOP is unique (why?) – cannonical SOP
- An element x in a BL is *maxterm* if it has 1 as its only successor
- A maxterm is a sum of literals involving all the variables
- Similar to SOP, *product of sums* (POS) may be defined
- A Boolean expression which is a product of maxterms is said to be in *conjunctive normal form* (CNF)

Boolean expressions



- $E = x\overline{z} + \overline{y}z + xy\overline{z}$
- $E = \overline{((\overline{xy})z)}\,\overline{((\overline{x}+z)(\overline{y}+\overline{z}))}$
- $E = x(\overline{\overline{y}z})$

- A SOP expression where each FP is a minterm is said to be in *disjunctive normal form* (DNF)
- The DNF of any SOP is unique (why?) – cannonical SOP
- An element x in a BL is *maxterm* if it has 1 as its only successor
- A maxterm is a sum of literals involving all the variables
- Similar to SOP, *product of sums* (POS) may be defined
- A Boolean expression which is a product of maxterms is said to be in *conjunctive normal form* (CNF)
- The CNF of any POS is unique (why?) – cannonical POS

Alternate argument for minterm expansion

Acceptance for complements: $\overline{x} = 1$ iff $x = 0$

Alternate argument for minterm expansion

Acceptance for complements: $\overline{x} = 1$ iff $x = 0$

Acceptance for products: $xy = 1$ iff $x = 1$ and $y = 1$

Alternate argument for minterm expansion

Acceptance for complements: $\overline{x} = 1$ iff $x = 0$

Acceptance for products: $xy = 1$ iff $x = 1$ and $y = 1$

Acceptance for sums: $u + v = 1$ iff $u = 1$ or $v = 1$

Alternate argument for minterm expansion

Acceptance for complements: $\overline{x} = 1$ iff $x = 0$

Acceptance for products: $xy = 1$ iff $x = 1$ and $y = 1$

Acceptance for sums: $u + v = 1$ iff $u = 1$ or $v = 1$

Minterm expansion: sum of distinct minterms

Alternate argument for minterm expansion

Acceptance for complements: $\overline{x} = 1$ iff $x = 0$

Acceptance for products: $xy = 1$ iff $x = 1$ and $y = 1$

Acceptance for sums: $u + v = 1$ iff $u = 1$ or $v = 1$

Minterm expansion: sum of distinct minterms

Alternate argument for minterm expansion

Acceptance for complements: $\overline{x} = 1$ iff $x = 0$

Acceptance for products: $xy = 1$ iff $x = 1$ and $y = 1$

Acceptance for sums: $u + v = 1$ iff $u = 1$ or $v = 1$

Minterm expansion: sum of distinct minterms

Acceptance for minterm expansion:
- An acceptance for minterm expansion on truth assignment of variables happens due to acceptance of exactly one minterm

Alternate argument for minterm expansion

Acceptance for complements: $\overline{x} = 1$ iff $x = 0$

Acceptance for products: $xy = 1$ iff $x = 1$ and $y = 1$

Acceptance for sums: $u + v = 1$ iff $u = 1$ or $v = 1$

Minterm expansion: sum of distinct minterms

Acceptance for minterm expansion:

- An acceptance for minterm expansion on truth assignment of variables happens due to acceptance of exactly one minterm
- If m_i and m_j are two distinct minterms on variables $x_1, \ldots, x_k$

Alternate argument for minterm expansion

Acceptance for complements: $\overline{x} = 1$ iff $x = 0$

Acceptance for products: $xy = 1$ iff $x = 1$ and $y = 1$

Acceptance for sums: $u + v = 1$ iff $u = 1$ or $v = 1$

Minterm expansion: sum of distinct minterms

Acceptance for minterm expansion:

- An acceptance for minterm expansion on truth assignment of variables happens due to acceptance of exactly one minterm
- If m_i and m_j are two distinct minterms on variables $x_1, \ldots, x_k$
- Let m_i and m_j differ on x_p

Alternate argument for minterm expansion

Acceptance for complements: $\overline{x} = 1$ iff $x = 0$

Acceptance for products: $xy = 1$ iff $x = 1$ and $y = 1$

Acceptance for sums: $u + v = 1$ iff $u = 1$ or $v = 1$

Minterm expansion: sum of distinct minterms

Acceptance for minterm expansion:

- An acceptance for minterm expansion on truth assignment of variables happens due to acceptance of exactly one minterm
- If m_i and m_j are two distinct minterms on variables $x_1, \ldots, x_k$
- Let m_i and m_j differ on x_p
- Let x_p occur as literal x_{pi} in m_i and x_{pj} in m_j

Alternate argument for minterm expansion

Acceptance for complements: $\overline{x} = 1$ iff $x = 0$

Acceptance for products: $xy = 1$ iff $x = 1$ and $y = 1$

Acceptance for sums: $u + v = 1$ iff $u = 1$ or $v = 1$

Minterm expansion: sum of distinct minterms

Acceptance for minterm expansion:

- An acceptance for minterm expansion on truth assignment of variables happens due to acceptance of exactly one minterm
- If m_i and m_j are two distinct minterms on variables $x_1, \ldots, x_k$
- Let m_i and m_j differ on x_p
- Let x_p occur as literal x_{pi} in m_i and x_{pj} in m_j
- Then $x_{pi} = \overline{x_{pj}}$, so if m_i accepts then m_j doesn't accept and vice versa

Alternate argument for minterm expansion

Acceptance for complements: $\overline{x} = 1$ iff $x = 0$

Acceptance for products: $xy = 1$ iff $x = 1$ and $y = 1$

Acceptance for sums: $u + v = 1$ iff $u = 1$ or $v = 1$

Minterm expansion: sum of distinct minterms

Acceptance for minterm expansion:

- An acceptance for minterm expansion on truth assignment of variables happens due to acceptance of exactly one minterm
- If m_i and m_j are two distinct minterms on variables $x_1, \ldots, x_k$
- Let m_i and m_j differ on x_p
- Let x_p occur as literal x_{pi} in m_i and x_{pj} in m_j
- Then $x_{pi} = \overline{x_{pj}}$, so if m_i accepts then m_j doesn't accept and vice versa
- This ensures that the minterm expansion is unique

Number of Boolean functions

By lattice:

Number of Boolean functions

By lattice:

- A Boolean lattice for a Boolean function of k variables has $n = 2^k$ atoms as minterms

Number of Boolean functions

By lattice:

- A Boolean lattice for a Boolean function of k variables has $n = 2^k$ atoms as minterms
- A Boolean lattice with n atoms has 2^n elements by the Stone representation theorem

Number of Boolean functions

By lattice:

- A Boolean lattice for a Boolean function of k variables has $n = 2^k$ atoms as minterms
- A Boolean lattice with n atoms has 2^n elements by the Stone representation theorem
- Each non-zero element has a unique representation in terms of the atoms (minterms)

Number of Boolean functions

By lattice:

- A Boolean lattice for a Boolean function of k variables has $n = 2^k$ atoms as minterms
- A Boolean lattice with n atoms has 2^n elements by the Stone representation theorem
- Each non-zero element has a unique representation in terms of the atoms (minterms)
- Thus there are $2^n = 2^{2^k}$ distinct Boolean functions

Number of Boolean functions

By lattice:

- A Boolean lattice for a Boolean function of k variables has $n = 2^k$ atoms as minterms
- A Boolean lattice with n atoms has 2^n elements by the Stone representation theorem
- Each non-zero element has a unique representation in terms of the atoms (minterms)
- Thus there are $2^n = 2^{2^k}$ distinct Boolean functions

By minterm expansion:

Number of Boolean functions

By lattice:

- A Boolean lattice for a Boolean function of k variables has $n = 2^k$ atoms as minterms
- A Boolean lattice with n atoms has 2^n elements by the Stone representation theorem
- Each non-zero element has a unique representation in terms of the atoms (minterms)
- Thus there are $2^n = 2^{2^k}$ distinct Boolean functions

By minterm expansion:

- A Boolean function on k variables has $n = 2^k$ possible minterms

Number of Boolean functions

By lattice:

- A Boolean lattice for a Boolean function of k variables has $n = 2^k$ atoms as minterms
- A Boolean lattice with n atoms has 2^n elements by the Stone representation theorem
- Each non-zero element has a unique representation in terms of the atoms (minterms)
- Thus there are $2^n = 2^{2^k}$ distinct Boolean functions

By minterm expansion:

- A Boolean function on k variables has $n = 2^k$ possible minterms
- A minterm expansion results in a unique acceptance

Number of Boolean functions

By lattice:

- A Boolean lattice for a Boolean function of k variables has $n = 2^k$ atoms as minterms
- A Boolean lattice with n atoms has 2^n elements by the Stone representation theorem
- Each non-zero element has a unique representation in terms of the atoms (minterms)
- Thus there are $2^n = 2^{2^k}$ distinct Boolean functions

By minterm expansion:

- A Boolean function on k variables has $n = 2^k$ possible minterms
- A minterm expansion results in a unique acceptance
- The minterms may be chosen in $\sum_{k=0}^{k=n} \binom{n}{k} = 2^n = 2^{2^k}$ ways

Number of Boolean functions

By lattice:

- A Boolean lattice for a Boolean function of k variables has $n = 2^k$ atoms as minterms
- A Boolean lattice with n atoms has 2^n elements by the Stone representation theorem
- Each non-zero element has a unique representation in terms of the atoms (minterms)
- Thus there are $2^n = 2^{2^k}$ distinct Boolean functions

By minterm expansion:

- A Boolean function on k variables has $n = 2^k$ possible minterms
- A minterm expansion results in a unique acceptance
- The minterms may be chosen in $\sum_{k=0}^{k=n} \binom{n}{k} = 2^n = 2^{2^k}$ ways
- Each choice denotes a distinct Boolean function

Boolean expression manipulation

- $xy + \overline{x}z + yz = xy + \overline{x}z$
- $(x + y)(\overline{x} + z)(y + z) = (x + y)(\overline{x} + z)$
- $T = (x + y)\overline{[\overline{x}(\overline{y} + \overline{z})]} + \overline{x}\,\overline{y} + \overline{x}\,\overline{z}$
- $xy + \overline{x}\,\overline{y} + yz = xy + \overline{x}\,\overline{y} + \overline{x}z$

Exclusive OR

- $a \oplus b = b \oplus a$

Exclusive OR

- $a \oplus b = b \oplus a$
- $(a \oplus b) \oplus c = a \oplus (b \oplus c) = a \oplus b \oplus c$

Exclusive OR

- $a \oplus b = b \oplus a$
- $(a \oplus b) \oplus c = a \oplus (b \oplus c) = a \oplus b \oplus c$
- $a(b \oplus c) = (ab) \oplus (ac)$

Exclusive OR

- $a \oplus b = b \oplus a$
- $(a \oplus b) \oplus c = a \oplus (b \oplus c) = a \oplus b \oplus c$
- $a(b \oplus c) = (ab) \oplus (ac)$
- if $a \oplus b = c$ then $\begin{cases} a \oplus c = b \\ b \oplus c = a \\ a \oplus b \oplus c = 0 \end{cases}$

Series-parallel switching circuits

- A transmission device may be treated as a gate (pass or block)

Series-parallel switching circuits

- A transmission device may be treated as a gate (pass or block)
- MOS transistor, relay, pneumatic valve

Series-parallel switching circuits

- A transmission device may be treated as a gate (pass or block)
- MOS transistor, relay, pneumatic valve
- Normally closed (primed: $\overline{x}$) or normally open (unprimed: x)

Series-parallel switching circuits

- A transmission device may be treated as a gate (pass or block)
- MOS transistor, relay, pneumatic valve
- Normally closed (primed: $\overline{x}$) or normally open (unprimed: x)
- Series connection denoted by AND

Series-parallel switching circuits

- A transmission device may be treated as a gate (pass or block)
- MOS transistor, relay, pneumatic valve
- Normally closed (primed: $\overline{x}$) or normally open (unprimed: x)
- Series connection denoted by AND
- Parallel connection denoted by OR

Series-parallel switching circuits

- A transmission device may be treated as a gate (pass or block)
- MOS transistor, relay, pneumatic valve
- Normally closed (primed: $\overline{x}$) or normally open (unprimed: x)
- Series connection denoted by AND
- Parallel connection denoted by OR
- $T = x\overline{y} + (\overline{x} + y)z$

Series-parallel switching circuits

- A transmission device may be treated as a gate (pass or block)
- MOS transistor, relay, pneumatic valve
- Normally closed (primed: $\overline{x}$) or normally open (unprimed: x)
- Series connection denoted by AND
- Parallel connection denoted by OR
- $T = x\overline{y} + (\overline{x} + y)z$
- $T = x\overline{y} + \overline{x}z + \overline{y}z + yz = x\overline{y} + \overline{x}z + z = x\overline{y} + z$

Series-parallel switching circuits

- A transmission device may be treated as a gate (pass or block)
- MOS transistor, relay, pneumatic valve
- Normally closed (primed: $\overline{x}$) or normally open (unprimed: x)
- Series connection denoted by AND
- Parallel connection denoted by OR
- $T = x\overline{y} + (\overline{x} + y)z$
- $T = x\overline{y} + \overline{x}z + \overline{y}z + yz = x\overline{y} + \overline{x}z + z = x\overline{y} + z$
- CMOS NAND, NOR

Shannon decomposition

- $f(x_1, x_2, \ldots, x_n) = x_1 \cdot f(1, x_2, \ldots, x_n) + \overline{x_1} \cdot f(0, x_2, \ldots, x_n)$

Shannon decomposition

- $f(x_1, x_2, \ldots, x_n) = x_1 \cdot f(1, x_2, \ldots, x_n) + \overline{x_1} \cdot f(0, x_2, \ldots, x_n)$
- $f(x_1, x_2, \ldots, x_n) = (\overline{x_1} + f(1, x_2, \ldots, x_n)) \cdot (x_1 + f(0, x_2, \ldots, x_n))$

Shannon decomposition

- $f(x_1, x_2, \ldots, x_n) = x_1 \cdot f(1, x_2, \ldots, x_n) + \overline{x_1} \cdot f(0, x_2, \ldots, x_n)$
- $f(x_1, x_2, \ldots, x_n) = (\overline{x_1} + f(1, x_2, \ldots, x_n)) \cdot (x_1 + f(0, x_2, \ldots, x_n))$
- Multiplexer realisation by Shannon decomposition or Shannon expansion

Shannon decomposition

- $f(x_1, x_2, \ldots, x_n) = x_1 \cdot f(1, x_2, \ldots, x_n) + \overline{x_1} \cdot f(0, x_2, \ldots, x_n)$
- $f(x_1, x_2, \ldots, x_n) = (\overline{x_1} + f(1, x_2, \ldots, x_n)) \cdot (x_1 + f(0, x_2, \ldots, x_n))$
- Multiplexer realisation by Shannon decomposition or Shannon expansion
- Repeated application to obtain CNF or DNF of a given Boolean function