CSE2353 - BOOLEAN ALGEBRA OUTLINE

• What is a Boolean Algebra?

• Relationship to Sets and Propositions

• Boolean Functions

• Minterms/Maxterms

• Boolean Algebra in Computer Science
**INTRODUCTION**

\[(\{0,1\},+,\ast,\neg,0,1)\]

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[(P(\{x\}),\cup,\cap,\neg,\emptyset,\{x\})\]

<table>
<thead>
<tr>
<th></th>
<th>\emptyset</th>
<th>{x}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\cup</td>
<td>\emptyset</td>
<td>{x}</td>
</tr>
<tr>
<td>{x}</td>
<td>{x}</td>
<td>{x}</td>
</tr>
<tr>
<td>\emptyset</td>
<td>\emptyset</td>
<td>\emptyset</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>\emptyset</th>
<th>{x}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\cap</td>
<td>\emptyset</td>
<td>{x}</td>
</tr>
<tr>
<td>{x}</td>
<td>\emptyset</td>
<td>\emptyset</td>
</tr>
<tr>
<td>\emptyset</td>
<td>{x}</td>
<td></td>
</tr>
</tbody>
</table>

\[(\{f,t\},\lor,\land,\neg,f,t)\]

<table>
<thead>
<tr>
<th></th>
<th>f</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>\lor</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>t</td>
<td>t</td>
<td>t</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>f</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>\land</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>t</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>f</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>\neg</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>t</td>
<td></td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td></td>
</tr>
</tbody>
</table>
WHAT IS A BOOLEAN ALGEBRA?

- $(B, +, *, \neg, 0, 1)$
- Binary Operator $+$ (Sum)
- Binary Operator $*$ (Product)
- Unary Operator $\neg$ (Complement)
- $0$ (Identity): $\forall b \in B, b + 0 = 0 + b = b$
- $1$ (Identity): $\forall b \in B, b * 1 = 1 * b = b$
- $+$ and $*$ are Associative
- $+$ and $*$ are Commutative
- $+$ Distributes over $*$
- $*$ Distributes over $+$
- $\forall b \in B, b + \overline{b} = 1$
- $\forall b \in B, b * \overline{b} = 0$
BOOLEAN ALGEBRA PROPERTIES

- Dual - Change Binary operators and Identities

- The dual of a Boolean algebra theorem is a Boolean algebra theorem

- The identity elements are unique

- The complement of an element is unique

- \( \forall b \in B, b + b = b \) and \( b \ast b = b \)

- \( \forall b \in B, 1 + b = 1 \) and \( 0 \ast b = 0 \)

- Absorption: \( \forall b_1, b_2 \in B, b_1 + (b_1 \ast b_2) = b_1 \)
  and \( b_1 \ast (b_1 + b_2) = b_1 \)

- Involution: \( \forall b \in B, \overline{\overline{b}} = b \)

- DeMorgan’s Laws: \( \forall b_1, b_2 \in B, (\overline{b_1 + b_2}) = \overline{b_1} \ast \overline{b_2} \)
  and \( (\overline{b_1 \ast b_2}) = \overline{b_1} + \overline{b_2} \)
BOOLEAN EXPRESSIONS

- Given \((B, +, \cdot, \neg, 0, 1)\), a Boolean Variable is a variable over the set \(B\)

- A Literal is a Boolean variable, \(x\), or its complement \(\bar{x}\)

- Boolean Expressions
  - Identity Elements \(0, 1\)
  - Boolean Variables \(x_1, x_2, \ldots, x_n\)
  - \((X + Y), (X \cdot Y), \overline{X}\) where \(X\) and \(Y\) are Boolean Expressions

- Two Boolean expressions are equivalent (equal) if one can be obtained from the other by a finite sequence of applications of the Boolean algebra axioms
BOOLEAN FUNCTIONS

• A Boolean Function $f: B^n \rightarrow B$ such that $f(x_1, x_2, \ldots, x_n)$ is a Boolean expression

• Examples: $f_1(x_1, x_2) = x_1 + x_2; f_2(x_1, x_2, x_3) = \overline{x_1} \cdot (x_2 + \overline{x_3})$

• Notation: Use $x_1 x_2$ to mean $x_1 \cdot x_2$

• Precedence Order: Complement, Product, Sum, Left to Right
BOOLEAN FUNCTION ON ONE VARIABLE

• $f(x) = 1$, $g(x) = x$

• Examine $f(x) + f(x)$, $f(x) + g(x)$, $f(x)g(x)$, $g(x) + g(x)$, $g(x)g(x)$, $(f(x))$

• Any function $f(x)$ can be written as $f(x) = f(0)x + f(1)x$.

• $0x + 0x = 0$

• $0x + 1x = x$

• $1x + 0x = x$

• $1x + 1x = x + x = 1$
MINTERMS & MAXTERMS

- One variable: \( x, \overline{x} \)

- Two variables: \( x_1 x_2, x_1 \overline{x}_2, \overline{x_1} x_2, \overline{x_1} \overline{x}_2 \)

- Minterm on \( n \) variables \( x_1, x_2, \ldots, x_n \) is a Boolean expression which has the form of the product of each Boolean variable or its complement.

- Two minterms on one variable, four on two, eight on 3, and \( 2^n \) on \( n \).

- Notation: \( x^0 = \overline{x} \) and \( x^1 = x \).

- \( m_{e_1 e_2 \ldots e_n} = x_1^{e_1} x_2^{e_2} \ldots x_n^{e_n} \)

- Examples: \( m_{11} = x_1 x_2, m_{10} = x_1 \overline{x}_2, m_{01} = \overline{x_1} x_2, m_{00} = \overline{x_1} \overline{x}_2 \)

- Maxterm on \( n \) variables \( x_1, x_2, \ldots, x_n \) is a Boolean expression which has the form of the sum of each Boolean variable or its complement.

- Examples: \( M_{11} = x_1 + x_2, M_{10} = x_1 + \overline{x}_2, m_{01} = \overline{x_1} + x_2, m_{00} = \overline{x_1} + \overline{x}_2 \)
CANONICAL FORMS OF BOOLEAN EXPRESSIONS

• Any Boolean expression can be written as the sum of minterms (or product of maxterms)
• Disjunctive Normal Form - Boolean expression written as sum of minterms
• Conjuctive Normal Form - Boolean expression written as product of maxterms
• We will normally use DNF
• To convert a Boolean expression, \( f(x_1, x_2, ..., x_n) \) into DNF, we need to determine the values for the constant prefixes (That is the \( f(e_1, e_2, ..., e_n) \)).
• Example: Convert \( x_1 + x_2 \) into DNF
CONVERTING TO CANONICAL FORM

- Algorithm:

  1. Create table showing values of $f(e_1, ..., e_n)$

  2. Rewrite $f$ using the output values shown as the prefixes for the corresponding terms

- Example:

<table>
<thead>
<tr>
<th>$e_1$</th>
<th>$e_2$</th>
<th>$f(e_1, e_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
GENERAL BOOLEAN ALGEBRAS

• The Power Set for any finite set can be used to define a Boolean Algebra

• Example: Look at $A = \{x,y\}$

• The cardinality of a finite Boolean algebra is a power of 2

• In Computer Science we are primarily interested in the Boolean algebra of cardinality 2
BOOLEAN ALGEBRA IN COMPUTER SCIENCE

- Switching Circuits
- Logic Networks
- Karnaugh Maps
SWITCHING CIRCUITS

• Two-state Device: On or Off (1 or 0; T or F)

• Switch - Open (No current) or Closed (Current)

• Switches Can be combined:
  – Parallel - Boolean algebra +
  – Series - Boolean algebra *
  – If $S$ denotes a switch then $\overline{S}$ denotes a switch which always has the opposite state

• Switching functions can be created to represent any Boolean function

• Switches can be mechanical or electronic
LOGIC NETWORKS

- Switching circuits either conduct (switch closed) or do not conduct (switch open) electricity. For example voltage-operated circuits could define 1 as a signal with $\geq 3$ volts and 0 with 0 volts and an acceptable tolerance.

- Gate - hardware that implements logic operations of AND, OR, NOT (inverter).

- Linking gates together creates a logic network.

- Adders

\[
\begin{align*}
\text{AND} & : x \text{ AND } y = xy \\
\text{XOR} & : x \text{ XOR } y = x \oplus y \\
\text{NOT} & : \overline{x} = \bar{x}
\end{align*}
\]