Boolean Algebra

Juri Kolčák

Friday 31st October, 2025

Boolean Algebra

Boolean algebra consists of the value set $\mathbb{B} = \{0, 1\}$ and three operators, $\wedge$, $\vee$, $\neg$.

OPERATORS:

$$a \wedge b = \begin{cases} 1 & \text{if } a = b = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$a \vee b = \begin{cases} 0 & \text{if } a = b = 0 \\ 1 & \text{otherwise} \end{cases}$$

$$\neg a = \begin{cases} 0 & \text{if } a = 1 \\ 1 & \text{if } a = 0 \end{cases}$$

Boolean Algebra

Boolean algebra consists of the value set $\mathbb{B} = \{0, 1\}$ and three operators, $\wedge$, $\vee$, $\neg$.

Operators:

$$a \wedge b = \begin{cases} 1 & \text{if } a = b = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$a \vee b = \begin{cases} 0 & \text{if } a = b = 0 \\ 1 & \text{otherwise} \end{cases}$$

$$\neg a = \begin{cases} 0 & \text{if } a = 1 \\ 1 & \text{if } a = 0 \end{cases}$$

Truth tables:

a	b	$a \wedge b$	$a \vee b$
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	1

a	$\neg a$
0	1
1	0

Boolean Functions

Boolean functions on $n \in \mathbb{N}$ variables $v_1, \ldots, v_n$, are defined recursively as follows:

$$\varphi ::= c \in \mathbb{B} \mid v \in \{v_1, \ldots, v_n\} \mid \neg(\varphi) \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2)$$

Boolean Functions

Boolean functions on $n \in \mathbb{N}$ variables $v_1, \ldots, v_n$, are defined recursively as follows:

$$\varphi ::= c \in \mathbb{B} \mid v \in \{v_1, \ldots, v_n\} \mid \neg(\varphi) \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2)$$

FUNCTIONS ON 1 VARIABLE:

$f_{neg}(a) = \neg(a) = \neg a$
$f_{id}(a) = a$
$f_{const}(a) = 0$

Boolean Functions

Boolean functions on $n \in \mathbb{N}$ variables $v_1, \ldots, v_n$, are defined recursively as follows:

$$\varphi ::= c \in \mathbb{B} \mid v \in \{v_1, \ldots, v_n\} \mid \neg(\varphi) \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2)$$

Functions on 1 variable:

$f_{neg}(a) = \neg(a) = \neg a$
$f_{id}(a) = a$
$f_{const}(a) = 0$

Functions on 2 variables:

$f_{and}(a, b) = a \wedge b$
$f_{or}(a, b) = a \vee b$
$f_{imp}(a, b) = \neg a \vee b = a \Rightarrow b$
$f_{equiv}(a, b) = (a \wedge b) \vee (\neg a \wedge \neg b) = a \Leftrightarrow b$
$f_{xor}(a, b) = (a \wedge \neg b) \vee (\neg a \wedge b) = a \oplus b$

Boolean Algebra Laws

Precedence:

$\neg a \wedge b \stackrel{\Delta}{=} \neg(a) \wedge b$

$\neg a \vee b \stackrel{\Delta}{=} \neg(a) \vee b$

Associativity:

$a \wedge (b \wedge c) = (a \wedge b) \wedge c$

$a \vee (b \vee c) = (a \vee b) \vee c$

Commutativity:

$a \wedge b = b \wedge a$

$a \vee b = b \vee a$

Distributivity:

$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$

$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$

Idempotence:

$a \wedge a = a$

$a \vee a = a$

Absorption:

$a \wedge (a \vee b) = a$

$a \vee (a \wedge b) = a$

Identities:

$a \wedge 1 = a$

$a \vee 0 = a$

Annihilators:

$a \wedge 0 = 0$

$a \vee 1 = 1$

Complementation:

$a \wedge \neg a = 0$

$a \vee \neg a = 1$

Double negation:

$\neg(\neg a) = a$

De Morgan's laws:

$\neg(a \wedge b) = \neg a \vee \neg b$

$\neg(a \vee b) = \neg a \wedge \neg b$

Example: XOR is the complement of equivalence

$$
\begin{aligned}
\neg(a \Leftrightarrow b) = &\qquad \text{/primary operators}\\
\neg((a \wedge b) \vee (\neg a \wedge \neg b)) = &\qquad \text{/De Morgan}\\
\neg(a \wedge b) \wedge \neg(\neg a \wedge \neg b) = &\qquad \text{/De Morgan x2}\\
(\neg a \vee \neg b) \wedge (\neg(\neg a) \vee \neg(\neg b)) = &\qquad \text{/Double negation x2}\\
(\neg a \vee \neg b) \wedge (a \vee b) = &\qquad \text{/Distributivity}\\
((\neg a \vee \neg b) \wedge a) \vee ((\neg a \vee \neg b) \wedge b) = &\qquad \text{/Commutativity x2}\\
(a \wedge (\neg a \vee \neg b)) \vee (b \wedge (\neg a \vee \neg b)) = &\qquad \text{/Distributivity} \times 2\\
(a \wedge \neg a) \vee (a \wedge \neg b) \vee (b \wedge \neg a) \vee (b \wedge \neg b) = &\qquad \text{/Complementation x2}\\
0 \vee (a \wedge \neg b) \vee (b \wedge \neg a) \vee 0 = &\qquad \text{/Commutativity}\\
(a \wedge \neg b) \vee (b \wedge \neg a) \vee 0 \vee 0 = &\qquad \text{/Identity x2}\\
(a \wedge \neg b) \vee (b \wedge \neg a) = &\qquad \text{/Commutativity}\\
(a \wedge \neg b) \vee (\neg a \wedge b) &
\end{aligned}
$$

Normal Forms

CONJUNCTIVE NORMAL FORM (CNS) – Conjunction of disjunctive clauses.

$$
\begin{aligned}
C &::= (D) \mid (D) \wedge C \\
D &::= L \mid L \vee D \\
L &::= v \mid \neg v & \qquad v \in \{v_1, \ldots, v_n\}
\end{aligned}
$$

DISJUNCTIVE NORMAL FORM (DNS) – Disjunction of conjunctive clauses.

$$
\begin{aligned}
D &::= (C) \mid (C) \vee D \\
C &::= L \mid L \wedge C \\
L &::= v \mid \neg v & \qquad v \in \{v_1, \ldots, v_n\}
\end{aligned}
$$

Constructing CNF – Semantic Approach

For each line in the truth table which evaluates to 0, add a disjunctive clause containing negations of the line to the CNF.

Example for: $(a \vee b) \Rightarrow (b \wedge \neg c)$

Constructing CNF – Semantic Approach

For each line in the truth table which evaluates to 0, add a disjunctive clause containing negations of the line to the CNF.

Example for: $(a \vee b) \Rightarrow (b \wedge \neg c)$

a	b	c	$\neg c$	$a \vee b$	$b \wedge \neg c$	$(a \vee b) \Rightarrow (b \wedge \neg c)$
0	0	0	1	0	0	1
1	0	0	1	1	0	0
0	1	0	1	1	1	1
1	1	0	1	1	1	1
0	0	1	0	0	0	1
1	0	1	0	1	0	0
0	1	1	0	1	0	0
1	1	1	0	1	0	0

Constructing CNF – Semantic Approach

For each line in the truth table which evaluates to 0, add a disjunctive clause containing negations of the line to the CNF.

Example for: $(a \vee b) \Rightarrow (b \wedge \neg c)$

a	b	c	$\neg c$	$a \vee b$	$b \wedge \neg c$	$(a \vee b) \Rightarrow (b \wedge \neg c)$
0	0	0	1	0	0	1
1	0	0	1	1	0	0
0	1	0	1	1	1	1
1	1	0	1	1	1	1
0	0	1	0	0	0	1
1	0	1	0	1	0	0
0	1	1	0	1	0	0
1	1	1	0	1	0	0

$$(\neg a \vee b \vee c) \wedge (\neg a \vee b \vee \neg c) \wedge (a \vee \neg b \vee \neg c) \wedge (\neg a \vee \neg b \neg c) =$$

$$= (\neg a \vee b) \wedge (\neg b \vee \neg c)$$

Constructing DNF – Semantic Approach

For each line in the truth table which evaluates to 1, add a conjunctive clause containing the line to the DNF.

Example for: $(a \vee b) \Rightarrow (b \wedge \neg c)$

a	b	c	$\neg c$	$a \vee b$	$b \wedge \neg c$	$(a \vee b) \Rightarrow (b \wedge \neg c)$
0	0	0	1	0	0	1
1	0	0	1	1	0	0
0	1	0	1	1	1	1
1	1	0	1	1	1	1
0	0	1	0	0	0	1
1	0	1	0	1	0	0
0	1	1	0	1	0	0
1	1	1	0	1	0	0

Constructing DNF – Semantic Approach

For each line in the truth table which evaluates to 1, add a conjunctive clause containing the line to the DNF.

Example for: $(a \vee b) \Rightarrow (b \wedge \neg c)$

a	b	c	$\neg c$	$a \vee b$	$b \wedge \neg c$	$(a \vee b) \Rightarrow (b \wedge \neg c)$
0	0	0	1	0	0	1
1	0	0	1	1	0	0
0	1	0	1	1	1	1
1	1	0	1	1	1	1
0	0	1	0	0	0	1
1	0	1	0	1	0	0
0	1	1	0	1	0	0
1	1	1	0	1	0	0

$$(\neg a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge b \wedge \neg c) \vee (a \wedge b \wedge \neg c) \vee (\neg a \wedge \neg b \wedge c) =$$

$$= (\neg a \wedge \neg b) \vee (b \wedge \neg c)$$

Constructing DNF – Syntactic Approach

Formula using only primary operators can be converted to a DNF using the following term rewriting system:

$$
\begin{aligned}
\neg(\neg a) &\rightsquigarrow a && \text{(Double negation)}\\
\neg(a \wedge b) &\rightsquigarrow \neg a \vee \neg b && \text{(De Morgan)}\\
\neg(a \vee b) &\rightsquigarrow \neg a \wedge \neg b && \text{(De Morgan)}\\
a \wedge (b \vee c) &\rightsquigarrow (a \wedge b) \vee (a \wedge c) && \text{(Distributivity)}\\
a \vee (b \wedge c) &\rightsquigarrow (a \vee b) \wedge (a \vee c) && \text{(Distributivity)}
\end{aligned}
$$

Constructing DNF – Syntactic Approach

Formula using only primary operators can be converted to a DNF using the following term rewriting system:

$$\begin{aligned}
\neg(\neg a) &\rightsquigarrow a && \text{(Double negation)}\\
\neg(a \wedge b) &\rightsquigarrow \neg a \vee \neg b && \text{(De Morgan)}\\
\neg(a \vee b) &\rightsquigarrow \neg a \wedge \neg b && \text{(De Morgan)}\\
a \wedge (b \vee c) &\rightsquigarrow (a \wedge b) \vee (a \wedge c) && \text{(Distributivity)}\\
a \vee (b \wedge c) &\rightsquigarrow (a \vee b) \wedge (a \vee c) && \text{(Distributivity)}
\end{aligned}$$

CNF of a formula φ can be constructed syntactically by first constructing a DNF of $\neg\varphi$ and subsequently applying De Morgan laws to the negation of this DNF.