CSE 311 Lecture 04: Boolean Algebra

Emina Torlak and Sami Davies

Topics

Equivalence and proofs

A brief review of Lecture 03.

Boolean algebra

A notation for combinational circuits.

Simplification and proofs

Optimizing circuits and proving theorems.

Equivalence and proofs

A brief review of Lecture 03.

Equivalence via truth tables and proofs

$A \equiv B$ is an **assertion** that two propositions A and B have the same truth values in all possible cases.

$A \equiv B$ and $\underbrace{(A \leftrightarrow B) \equiv \mathsf{T}}_{\text{tautology}}$ have the same meaning.

Checking equivalence has many real-world applications.

Verification, optimization, and more!

There are two ways to check equivalence of propositional formulas.

Brute-force: compare their truth tables.

Proof-based: apply equivalences to transform one into the other.

Example: show tautology with a truth table and proof

$(p \land q) \to (q \lor p)$

A **truth table** for $(p \land q) \to (q \lor p) \equiv \mathsf{T}$.

p	q	$p \land q$	$q \lor p$	$(p \land q) \to (q \lor p)$
F	F	F	F	T
F	T	F	T	T
T	F	F	T	T
T	T	T	T	T

Example: show tautology with a truth table and proof

$$(p \wedge q) \rightarrow (q \vee p)$$

$$
\begin{aligned}
(p \wedge q) \rightarrow (q \vee p) &\equiv \\
&\equiv \\
&\equiv \\
&\equiv \\
&\equiv \\
&\equiv \\
&\equiv \\
&\equiv \\
&\equiv \mathsf{T}
\end{aligned}
$$

DeMorgan's laws

$\neg(p \wedge q) \equiv \neg p \vee \neg q$
$\neg(p \vee q) \equiv \neg p \wedge \neg q$

Law of implication

$p \rightarrow q \equiv \neg p \vee q$

Contrapositive

$p \rightarrow q \equiv \neg q \rightarrow \neg p$

Biconditional

$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$

Double negation

$p \equiv \neg\neg p$

Identity

$p \wedge \mathsf{T} \equiv p$
$p \vee \mathsf{F} \equiv p$

Domination

$p \wedge \mathsf{F} \equiv \mathsf{F}$
$p \vee \mathsf{T} \equiv \mathsf{T}$

Idempotence

$p \wedge p \equiv p$
$p \vee p \equiv p$

Commutativity

$p \wedge q \equiv q \wedge p$
$p \vee q \equiv q \vee p$

Associativity

$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
$(p \vee q) \vee r \equiv p \vee (q \vee r)$

Distributivity

$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$

Absorption

$p \wedge (p \vee q) \equiv p$
$p \vee (p \wedge q) \equiv p$

Negation

$p \wedge \neg p \equiv \mathsf{F}$
$p \vee \neg p \equiv \mathsf{T}$

Example: show tautology with a truth table and proof

$$(p \wedge q) \rightarrow (q \vee p)$$

$$\begin{aligned}(p \wedge q) \rightarrow (q \vee p) &\equiv \neg(p \wedge q) \vee (q \vee p) && \text{Law of implication}\\ &\equiv \\ &\equiv \\ &\equiv \\ &\equiv \\ &\equiv \\ &\equiv \\ &\equiv \\ &\equiv \mathsf{T}\end{aligned}$$

DeMorgan's laws

$\neg(p \wedge q) \equiv \neg p \vee \neg q$

$\neg(p \vee q) \equiv \neg p \wedge \neg q$

Law of implication

$p \rightarrow q \equiv \neg p \vee q$

Contrapositive

$p \rightarrow q \equiv \neg q \rightarrow \neg p$

Biconditional

$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$

Double negation

$p \equiv \neg\neg p$

Identity

$p \wedge \mathsf{T} \equiv p$

$p \vee \mathsf{F} \equiv p$

Domination

$p \wedge \mathsf{F} \equiv \mathsf{F}$

$p \vee \mathsf{T} \equiv \mathsf{T}$

Idempotence

$p \wedge p \equiv p$

$p \vee p \equiv p$

Commutativity

$p \wedge q \equiv q \wedge p$

$p \vee q \equiv q \vee p$

Associativity

$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$

$(p \vee q) \vee r \equiv p \vee (q \vee r)$

Distributivity

$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$

$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$

Absorption

$p \wedge (p \vee q) \equiv p$

$p \vee (p \wedge q) \equiv p$

Negation

$p \wedge \neg p \equiv \mathsf{F}$

$p \vee \neg p \equiv \mathsf{T}$

Example: show tautology with a truth table and proof

$$(p \wedge q) \rightarrow (q \vee p)$$

$$\begin{aligned}
(p \wedge q) \rightarrow (q \vee p) &\equiv \neg(p \wedge q) \vee (q \vee p) && \textbf{Law of implication} \\
&\equiv (\neg p \vee \neg q) \vee (q \vee p) && \textbf{DeMorgan} \\
&\equiv \\
&\equiv \\
&\equiv \\
&\equiv \\
&\equiv \\
&\equiv \\
&\equiv \mathsf{T}
\end{aligned}$$

DeMorgan's laws

$\neg(p \wedge q) \equiv \neg p \vee \neg q$
$\neg(p \vee q) \equiv \neg p \wedge \neg q$

Law of implication

$p \rightarrow q \equiv \neg p \vee q$

Contrapositive

$p \rightarrow q \equiv \neg q \rightarrow \neg p$

Biconditional

$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$

Double negation

$p \equiv \neg\neg p$

Identity

$p \wedge \mathsf{T} \equiv p$
$p \vee \mathsf{F} \equiv p$

Domination

$p \wedge \mathsf{F} \equiv \mathsf{F}$
$p \vee \mathsf{T} \equiv \mathsf{T}$

Idempotence

$p \wedge p \equiv p$
$p \vee p \equiv p$

Commutativity

$p \wedge q \equiv q \wedge p$
$p \vee q \equiv q \vee p$

Associativity

$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
$(p \vee q) \vee r \equiv p \vee (q \vee r)$

Distributivity

$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$

Absorption

$p \wedge (p \vee q) \equiv p$
$p \vee (p \wedge q) \equiv p$

Negation

$p \wedge \neg p \equiv \mathsf{F}$
$p \vee \neg p \equiv \mathsf{T}$

Example: show tautology with a truth table and proof

$$(p \wedge q) \rightarrow (q \vee p)$$

$$\begin{aligned}
(p \wedge q) \rightarrow (q \vee p) &\equiv \neg(p \wedge q) \vee (q \vee p) && \text{Law of implication} \\
&\equiv (\neg p \vee \neg q) \vee (q \vee p) && \text{DeMorgan} \\
&\equiv \neg p \vee (\neg q \vee (q \vee p)) && \text{Associativity} \\
&\equiv \\
&\equiv \\
&\equiv \\
&\equiv \\
&\equiv \\
&\equiv \mathsf{T}
\end{aligned}$$

DeMorgan's laws

$\neg(p \wedge q) \equiv \neg p \vee \neg q$

$\neg(p \vee q) \equiv \neg p \wedge \neg q$

Law of implication

$p \rightarrow q \equiv \neg p \vee q$

Contrapositive

$p \rightarrow q \equiv \neg q \rightarrow \neg p$

Biconditional

$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$

Double negation

$p \equiv \neg\neg p$

Identity

$p \wedge \mathsf{T} \equiv p$

$p \vee \mathsf{F} \equiv p$

Domination

$p \wedge \mathsf{F} \equiv \mathsf{F}$

$p \vee \mathsf{T} \equiv \mathsf{T}$

Idempotence

$p \wedge p \equiv p$

$p \vee p \equiv p$

Commutativity

$p \wedge q \equiv q \wedge p$

$p \vee q \equiv q \vee p$

Associativity

$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$

$(p \vee q) \vee r \equiv p \vee (q \vee r)$

Distributivity

$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$

$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$

Absorption

$p \wedge (p \vee q) \equiv p$

$p \vee (p \wedge q) \equiv p$

Negation

$p \wedge \neg p \equiv \mathsf{F}$

$p \vee \neg p \equiv \mathsf{T}$

Example: show tautology with a truth table and proof

$$(p \land q) \to (q \lor p)$$

$$\begin{aligned}
(p \land q) \to (q \lor p) &\equiv \neg(p \land q) \lor (q \lor p) && \text{Law of implication} \\
&\equiv (\neg p \lor \neg q) \lor (q \lor p) && \text{DeMorgan} \\
&\equiv \neg p \lor (\neg q \lor (q \lor p)) && \text{Associativity} \\
&\equiv \neg p \lor ((\neg q \lor q) \lor p) && \text{Associativity} \\
&\equiv \\
&\equiv \\
&\equiv \\
&\equiv \\
&\equiv \mathsf{T}
\end{aligned}$$

DeMorgan's laws

$\neg(p \land q) \equiv \neg p \lor \neg q$
$\neg(p \lor q) \equiv \neg p \land \neg q$

Law of implication

$p \to q \equiv \neg p \lor q$

Contrapositive

$p \to q \equiv \neg q \to \neg p$

Biconditional

$p \leftrightarrow q \equiv (p \to q) \land (q \to p)$

Double negation

$p \equiv \neg\neg p$

Identity

$p \land \mathsf{T} \equiv p$
$p \lor \mathsf{F} \equiv p$

Domination

$p \land \mathsf{F} \equiv \mathsf{F}$
$p \lor \mathsf{T} \equiv \mathsf{T}$

Idempotence

$p \land p \equiv p$
$p \lor p \equiv p$

Commutativity

$p \land q \equiv q \land p$
$p \lor q \equiv q \lor p$

Associativity

$(p \land q) \land r \equiv p \land (q \land r)$
$(p \lor q) \lor r \equiv p \lor (q \lor r)$

Distributivity

$p \land (q \lor r) \equiv (p \land q) \lor (p \land r)$
$p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$

Absorption

$p \land (p \lor q) \equiv p$
$p \lor (p \land q) \equiv p$

Negation

$p \land \neg p \equiv \mathsf{F}$
$p \lor \neg p \equiv \mathsf{T}$

Example: show tautology with a truth table and proof

$$(p \land q) \to (q \lor p)$$

$$\begin{aligned}
(p \land q) \to (q \lor p) &\equiv \neg(p \land q) \lor (q \lor p) && \text{Law of implication} \\
&\equiv (\neg p \lor \neg q) \lor (q \lor p) && \text{DeMorgan} \\
&\equiv \neg p \lor (\neg q \lor (q \lor p)) && \text{Associativity} \\
&\equiv \neg p \lor ((\neg q \lor q) \lor p) && \text{Associativity} \\
&\equiv \neg p \lor (p \lor (\neg q \lor q)) && \text{Commutativity} \\
&\equiv \\
&\equiv \\
&\equiv \\
&\equiv \mathsf{T}
\end{aligned}$$

DeMorgan's laws

$\neg(p \land q) \equiv \neg p \lor \neg q$

$\neg(p \lor q) \equiv \neg p \land \neg q$

Law of implication

$p \to q \equiv \neg p \lor q$

Contrapositive

$p \to q \equiv \neg q \to \neg p$

Biconditional

$p \leftrightarrow q \equiv (p \to q) \land (q \to p)$

Double negation

$p \equiv \neg\neg p$

Identity

$p \land \mathsf{T} \equiv p$

$p \lor \mathsf{F} \equiv p$

Domination

$p \land \mathsf{F} \equiv \mathsf{F}$

$p \lor \mathsf{T} \equiv \mathsf{T}$

Idempotence

$p \land p \equiv p$

$p \lor p \equiv p$

Commutativity

$p \land q \equiv q \land p$

$p \lor q \equiv q \lor p$

Associativity

$(p \land q) \land r \equiv p \land (q \land r)$

$(p \lor q) \lor r \equiv p \lor (q \lor r)$

Distributivity

$p \land (q \lor r) \equiv (p \land q) \lor (p \land r)$

$p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$

Absorption

$p \land (p \lor q) \equiv p$

$p \lor (p \land q) \equiv p$

Negation

$p \land \neg p \equiv \mathsf{F}$

$p \lor \neg p \equiv \mathsf{T}$

Example: show tautology with a truth table and proof

$$(p \land q) \to (q \lor p)$$

$$\begin{aligned}
(p \land q) \to (q \lor p) &\equiv \neg(p \land q) \lor (q \lor p) && \text{Law of implication} \\
&\equiv (\neg p \lor \neg q) \lor (q \lor p) && \text{DeMorgan} \\
&\equiv \neg p \lor (\neg q \lor (q \lor p)) && \text{Associativity} \\
&\equiv \neg p \lor ((\neg q \lor q) \lor p) && \text{Associativity} \\
&\equiv \neg p \lor (p \lor (\neg q \lor q)) && \text{Commutativity} \\
&\equiv (\neg p \lor p) \lor (\neg q \lor q) && \text{Associativity} \\
&\equiv \\
&\equiv \\
&\equiv \mathsf{T}
\end{aligned}$$

DeMorgan's laws

$\neg(p \land q) \equiv \neg p \lor \neg q$

$\neg(p \lor q) \equiv \neg p \land \neg q$

Law of implication

$p \to q \equiv \neg p \lor q$

Contrapositive

$p \to q \equiv \neg q \to \neg p$

Biconditional

$p \leftrightarrow q \equiv (p \to q) \land (q \to p)$

Double negation

$p \equiv \neg\neg p$

Identity

$p \land \mathsf{T} \equiv p$

$p \lor \mathsf{F} \equiv p$

Domination

$p \land \mathsf{F} \equiv \mathsf{F}$

$p \lor \mathsf{T} \equiv \mathsf{T}$

Idempotence

$p \land p \equiv p$

$p \lor p \equiv p$

Commutativity

$p \land q \equiv q \land p$

$p \lor q \equiv q \lor p$

Associativity

$(p \land q) \land r \equiv p \land (q \land r)$

$(p \lor q) \lor r \equiv p \lor (q \lor r)$

Distributivity

$p \land (q \lor r) \equiv (p \land q) \lor (p \land r)$

$p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$

Absorption

$p \land (p \lor q) \equiv p$

$p \lor (p \land q) \equiv p$

Negation

$p \land \neg p \equiv \mathsf{F}$

$p \lor \neg p \equiv \mathsf{T}$

Example: show tautology with a truth table and proof

$$(p \wedge q) \rightarrow (q \vee p)$$

$$\begin{aligned}
(p \wedge q) \rightarrow (q \vee p) &\equiv \neg(p \wedge q) \vee (q \vee p) && \text{Law of implication} \\
&\equiv (\neg p \vee \neg q) \vee (q \vee p) && \text{DeMorgan} \\
&\equiv \neg p \vee (\neg q \vee (q \vee p)) && \text{Associativity} \\
&\equiv \neg p \vee ((\neg q \vee q) \vee p) && \text{Associativity} \\
&\equiv \neg p \vee (p \vee (\neg q \vee q)) && \text{Commutativity} \\
&\equiv (\neg p \vee p) \vee (\neg q \vee q) && \text{Associativity} \\
&\equiv (p \vee \neg p) \vee (q \vee \neg q) && \text{Commutativity (twice)} \\
&\equiv \\
&\equiv \mathsf{T}
\end{aligned}$$

DeMorgan's laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$
$\neg(p \vee q) \equiv \neg p \wedge \neg q$

Law of implication
$p \rightarrow q \equiv \neg p \vee q$

Contrapositive
$p \rightarrow q \equiv \neg q \rightarrow \neg p$

Biconditional
$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$

Double negation
$p \equiv \neg\neg p$

Identity
$p \wedge \mathsf{T} \equiv p$
$p \vee \mathsf{F} \equiv p$

Domination
$p \wedge \mathsf{F} \equiv \mathsf{F}$
$p \vee \mathsf{T} \equiv \mathsf{T}$

Idempotence
$p \wedge p \equiv p$
$p \vee p \equiv p$

Commutativity
$p \wedge q \equiv q \wedge p$
$p \vee q \equiv q \vee p$

Associativity
$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
$(p \vee q) \vee r \equiv p \vee (q \vee r)$

Distributivity
$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$

Absorption
$p \wedge (p \vee q) \equiv p$
$p \vee (p \wedge q) \equiv p$

Negation
$p \wedge \neg p \equiv \mathsf{F}$
$p \vee \neg p \equiv \mathsf{T}$

Example: show tautology with a truth table and proof

$(p \land q) \to (q \lor p)$

$$
\begin{aligned}
(p \land q) \to (q \lor p) &\equiv \neg(p \land q) \lor (q \lor p) && \text{Law of implication}\\
&\equiv (\neg p \lor \neg q) \lor (q \lor p) && \text{DeMorgan}\\
&\equiv \neg p \lor (\neg q \lor (q \lor p)) && \text{Associativity}\\
&\equiv \neg p \lor ((\neg q \lor q) \lor p) && \text{Associativity}\\
&\equiv \neg p \lor (p \lor (\neg q \lor q)) && \text{Commutativity}\\
&\equiv (\neg p \lor p) \lor (\neg q \lor q) && \text{Associativity}\\
&\equiv (p \lor \neg p) \lor (q \lor \neg q) && \text{Commutativity (twice)}\\
&\equiv \mathsf{T} \lor \mathsf{T} && \text{Negation (twice)}\\
&\equiv \mathsf{T}
\end{aligned}
$$

DeMorgan's laws

$\neg(p \land q) \equiv \neg p \lor \neg q$

$\neg(p \lor q) \equiv \neg p \land \neg q$

Law of implication

$p \to q \equiv \neg p \lor q$

Contrapositive

$p \to q \equiv \neg q \to \neg p$

Biconditional

$p \leftrightarrow q \equiv (p \to q) \land (q \to p)$

Double negation

$p \equiv \neg\neg p$

Identity

$p \land \mathsf{T} \equiv p$

$p \lor \mathsf{F} \equiv p$

Domination

$p \land \mathsf{F} \equiv \mathsf{F}$

$p \lor \mathsf{T} \equiv \mathsf{T}$

Idempotence

$p \land p \equiv p$

$p \lor p \equiv p$

Commutativity

$p \land q \equiv q \land p$

$p \lor q \equiv q \lor p$

Associativity

$(p \land q) \land r \equiv p \land (q \land r)$

$(p \lor q) \lor r \equiv p \lor (q \lor r)$

Distributivity

$p \land (q \lor r) \equiv (p \land q) \lor (p \land r)$

$p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$

Absorption

$p \land (p \lor q) \equiv p$

$p \lor (p \land q) \equiv p$

Negation

$p \land \neg p \equiv \mathsf{F}$

$p \lor \neg p \equiv \mathsf{T}$

Example: show tautology with a truth table and proof

$$(p \wedge q) \rightarrow (q \vee p)$$

$$\begin{aligned}
(p \wedge q) \rightarrow (q \vee p) &\equiv \neg(p \wedge q) \vee (q \vee p) && \text{Law of implication}\\
&\equiv (\neg p \vee \neg q) \vee (q \vee p) && \text{DeMorgan}\\
&\equiv \neg p \vee (\neg q \vee (q \vee p)) && \text{Associativity}\\
&\equiv \neg p \vee ((\neg q \vee q) \vee p) && \text{Associativity}\\
&\equiv \neg p \vee (p \vee (\neg q \vee q)) && \text{Commutativity}\\
&\equiv (\neg p \vee p) \vee (\neg q \vee q) && \text{Associativity}\\
&\equiv (p \vee \neg p) \vee (q \vee \neg q) && \text{Commutativity (twice)}\\
&\equiv \mathsf{T} \vee \mathsf{T} && \text{Negation (twice)}\\
&\equiv \mathsf{T} && \text{Idempotence}
\end{aligned}$$

DeMorgan's laws

$\neg(p \wedge q) \equiv \neg p \vee \neg q$

$\neg(p \vee q) \equiv \neg p \wedge \neg q$

Law of implication

$p \rightarrow q \equiv \neg p \vee q$

Contrapositive

$p \rightarrow q \equiv \neg q \rightarrow \neg p$

Biconditional

$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$

Double negation

$p \equiv \neg\neg p$

Identity

$p \wedge \mathsf{T} \equiv p$

$p \vee \mathsf{F} \equiv p$

Domination

$p \wedge \mathsf{F} \equiv \mathsf{F}$

$p \vee \mathsf{T} \equiv \mathsf{T}$

Idempotence

$p \wedge p \equiv p$

$p \vee p \equiv p$

Commutativity

$p \wedge q \equiv q \wedge p$

$p \vee q \equiv q \vee p$

Associativity

$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$

$(p \vee q) \vee r \equiv p \vee (q \vee r)$

Distributivity

$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$

$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$

Absorption

$p \wedge (p \vee q) \equiv p$

$p \vee (p \wedge q) \equiv p$

Negation

$p \wedge \neg p \equiv \mathsf{F}$

$p \vee \neg p \equiv \mathsf{T}$

Truth tables versus proofs

Proofs are not smaller than truth tables where there are a few propositional variables.

But proofs are usually much smaller when there are many variables.

We can extend the proof method to reason about richer logics for which truth tables don't apply.

Theorem provers use a combination of search (truth tables) and deduction (proofs) to automate equivalence checking.

Boolean algebra

A notation for combinational circuits.

Recall the relationship between logic and circuits …

Digital circuits implement propositional logic:

- T corresponds to 1 or high voltage.
- F corresponds to 0 or low voltage.

Digital circuits are functions that

- take values 0/1 as inputs and produce 0/1 as output;
- $out = F(input)$, where F is built out of wires and gates; and
- every bit of output is computed from some bits of the input.

Recall the relationship between logic and circuits …

Digital circuits implement propositional logic:

- T corresponds to 1 or high voltage.
- F corresponds to 0 or low voltage.

Digital circuits are functions that

- take values 0/1 as inputs and produce 0/1 as output;
- $out = F(input)$, where F is built out of wires and gates; and
- every bit of output is computed from some bits of the input.

We call these types of digital circuits *combinational logic circuits*. There are other kinds of digital circuits (called *sequential circuits*) but we'll focus on combinational circuits in this course.

Boolean algebra is a notation for combinational logic

Think of it as a notation for propositional logic used in circuit design.

Boolean algebra consists of the following elements and operations:

- a set of elements $B = \{0, 1\}$,
- binary operations $\{+, \cdot\}$,
- a unary operation $\{\,'\,\}$.

These correspond to the truth values $\{\mathsf{F}, \mathsf{T}\}$, and the logical connectives $\vee, \wedge, \neg$.

Boolean algebra is a notation for combinational logic

Think of it as a notation for propositional logic used in circuit design.

Boolean algebra consists of the following elements and operations:

- a set of elements $B = \{0, 1\}$,
- binary operations $\{+, \cdot\}$,
- a unary operation $\{\,'\,\}$.

These correspond to the truth values {F, T}, and the logical connectives $\vee$, $\wedge$, $\neg$.

Boolean operations satisfy the following axioms for any $a, b, c \in B$:

Closure

$$a + b \in B$$
$$a \cdot b \in B$$

Commutativity

$$a + b = b + a$$
$$a \cdot b = b \cdot a$$

Associativity

$$a + (b + c) = (a + b) + c$$
$$a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

Distributivity

$$a + (b \cdot c) = (a + b) \cdot (a + c)$$
$$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

Identity

$$a + 0 = a$$
$$a \cdot 1 = a$$

Complementarity

$$a + a' = 1$$
$$a \cdot a' = 0$$

Null

$$a + 1 = 1$$
$$a \cdot 0 = 0$$

Idempotency

$$a + a = a$$
$$a \cdot a = a$$

Involution

$$(a')' = a$$

Example: from spec to code to logic to circuits

Suppose that we want to compute the number of lectures or sections remaining *at the start* of a given day of the week.

Example: from spec to code to logic to circuits

Suppose that we want to compute the number of lectures or sections remaining *at the start* of a given day of the week.

The function for this computation has the following signature:

- **Inputs**: day of the week (integers from 0 to 6), lecture flag (boolean).
- **Output**: number of sessions left (integer from 0 to 3).

Example: from spec to code to logic to circuits

Suppose that we want to compute the number of lectures or sections remaining *at the start* of a given day of the week.

The function for this computation has the following signature:

- **Inputs**: day of the week (integers from 0 to 6), lecture flag (boolean).
- **Output**: number of sessions left (integer from 0 to 3).

Here are some examples of the function's input/output behavior:

- Input: (`Wednesday`, `Lecture`), Output: 2
- Input: (`Monday`, `Section`), Output: 1

How would you implement this function in Java?

From spec to code …

```java
public class Sessions {

    public static int classesLeft(int day, boolean lecture) {
        switch (day) {
            case 0: // SUNDAY
            case 1: // MONDAY
                return lecture ? 3 : 1;
            case 2: // TUESDAY
            case 3: // WEDNESDAY
                return lecture ? 2 : 1;
            case 4: // THURSDAY
                return lecture ? 1 : 1;
            case 5: // FRIDAY
                return lecture ? 1 : 0;
            default: //case 6: // SATURDAY
                return lecture ? 0 : 0;
        }
    }

     public static void main(String []args){
        System.out.println("(W, L) -> " + classesLeft(3,true));
        System.out.println("(M, S) -> " + classesLeft(1,false));
     }
}
```

From spec to code …

```
public class Sessions {

    public static int classesLeft(int day, boolean lecture) {
        switch (day) {
            case 0: // SUNDAY
            case 1: // MONDAY
                return lecture ? 3 : 1;
            case 2: // TUESDAY
            case 3: // WEDNESDAY
                return lecture ? 2 : 1;
            case 4: // THURSDAY
                return lecture ? 1 : 1;
            case 5: // FRIDAY
                return lecture ? 1 : 0;
            default: //case 6: // SATURDAY
                return lecture ? 0 : 0;
        }
    }

     public static void main(String []args){
        System.out.println("(W, L) -> " + classesLeft(3,true));
        System.out.println("(M, S) -> " + classesLeft(1,false));
     }
}
```

Suppose that we need this function to run *really* fast …

From spec to code …

```java
public class Sessions {

    public static int classesLeft(int day, boolean lecture) {
        switch (day) {
            case 0: // SUNDAY
            case 1: // MONDAY
                return lecture ? 3 : 1;
            case 2: // TUESDAY
            case 3: // WEDNESDAY
                return lecture ? 2 : 1;
            case 4: // THURSDAY
                return lecture ? 1 : 1;
            case 5: // FRIDAY
                return lecture ? 1 : 0;
            default: //case 6: // SATURDAY
                return lecture ? 0 : 0;
        }
    }

     public static void main(String []args){
        System.out.println("(W, L) -> " + classesLeft(3,true));
        System.out.println("(M, S) -> " + classesLeft(1,false));
     }
}
```

Suppose that we need this function to run *really* fast … To do that, we'll implement a custom circuit (hardware accelerator!).

From code to combinational logic …

Recall the signature of our function:

- **Inputs**: day of the week (integers from 0 to 6), lecture flag (boolean).
- **Output**: number of sessions left (integer from 0 to 3).

How many bits for each input/output?

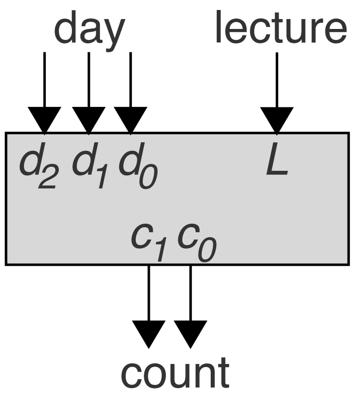
From code to combinational logic …

Recall the signature of our function:

- **Inputs**: day of the week (integers from 0 to 6), lecture flag (boolean).
- **Output**: number of sessions left (integer from 0 to 3).

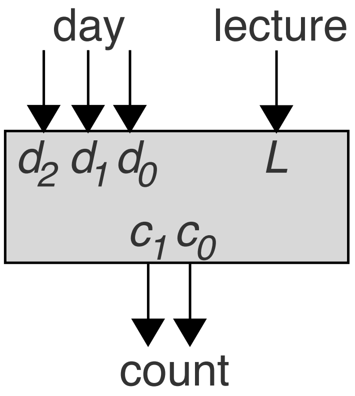
How many bits for each input/output?

- **Inputs**: 3 bits for day of the week, 1 bit for the lecture flag.
- **Output**: 2 bits for the number of sessions left.



From code to combinational logic via a truth table

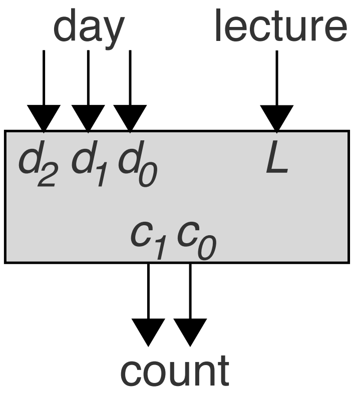
```
switch (day) {
  case 0: // SUNDAY
  case 1: // MONDAY
      return lecture ? 3 : 1;
  case 2: // TUESDAY
  case 3: // WEDNESDAY
      return lecture ? 2 : 1;
  case 4: // THURSDAY
      return lecture ? 1 : 1;
  case 5: // FRIDAY
      return lecture ? 1 : 0;
  default: // SATURDAY etc.
      return lecture ? 0 : 0;
}
```



Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	
SUN	000	1	
MON	001	0	
MON	001	1	
TUE	010	0	
TUE	010	1	
WED	011	0	
WED	011	1	
THU	100	0	
THU	100	1	
FRI	101	0	
FRI	101	1	
SAT	110	0	
SAT	110	1	
-	111	0	
-	111	1	

From code to combinational logic via a truth table

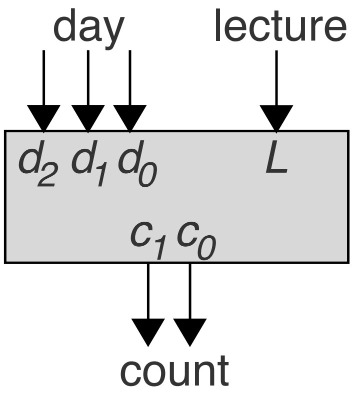
```
switch (day) {
  case 0: // SUNDAY
  case 1: // MONDAY
      return lecture ? 3 : 1;
  case 2: // TUESDAY
  case 3: // WEDNESDAY
      return lecture ? 2 : 1;
  case 4: // THURSDAY
      return lecture ? 1 : 1;
  case 5: // FRIDAY
      return lecture ? 1 : 0;
  default: // SATURDAY etc.
      return lecture ? 0 : 0;
}
```



Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	
MON	001	1	
TUE	010	0	
TUE	010	1	
WED	011	0	
WED	011	1	
THU	100	0	
THU	100	1	
FRI	101	0	
FRI	101	1	
SAT	110	0	
SAT	110	1	
-	111	0	
-	111	1	

From code to combinational logic via a truth table

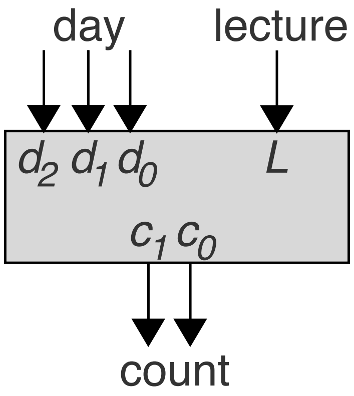
```
switch (day) {
  case 0: // SUNDAY
  case 1: // MONDAY
      return lecture ? 3 : 1;
  case 2: // TUESDAY
  case 3: // WEDNESDAY
      return lecture ? 2 : 1;
  case 4: // THURSDAY
      return lecture ? 1 : 1;
  case 5: // FRIDAY
      return lecture ? 1 : 0;
  default: // SATURDAY etc.
      return lecture ? 0 : 0;
}
```



Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	
TUE	010	1	
WED	011	0	
WED	011	1	
THU	100	0	
THU	100	1	
FRI	101	0	
FRI	101	1	
SAT	110	0	
SAT	110	1	
-	111	0	
-	111	1	

From code to combinational logic via a truth table

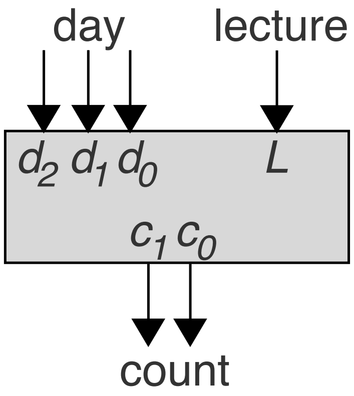
```
switch (day) {
  case 0: // SUNDAY
  case 1: // MONDAY
      return lecture ? 3 : 1;
  case 2: // TUESDAY
  case 3: // WEDNESDAY
      return lecture ? 2 : 1;
  case 4: // THURSDAY
      return lecture ? 1 : 1;
  case 5: // FRIDAY
      return lecture ? 1 : 0;
  default: // SATURDAY etc.
      return lecture ? 0 : 0;
}
```



Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	
WED	011	1	
THU	100	0	
THU	100	1	
FRI	101	0	
FRI	101	1	
SAT	110	0	
SAT	110	1	
-	111	0	
-	111	1	

From code to combinational logic via a truth table

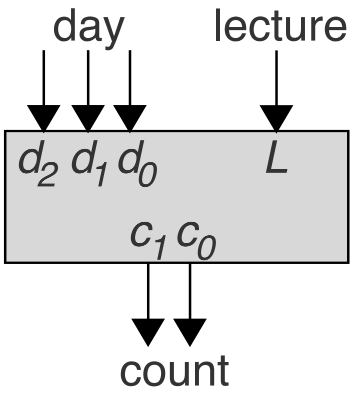
```
switch (day) {
  case 0: // SUNDAY
  case 1: // MONDAY
      return lecture ? 3 : 1;
  case 2: // TUESDAY
  case 3: // WEDNESDAY
      return lecture ? 2 : 1;
  case 4: // THURSDAY
      return lecture ? 1 : 1;
  case 5: // FRIDAY
      return lecture ? 1 : 0;
  default: // SATURDAY etc.
      return lecture ? 0 : 0;
}
```



Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	
THU	100	1	
FRI	101	0	
FRI	101	1	
SAT	110	0	
SAT	110	1	
-	111	0	
-	111	1	

From code to combinational logic via a truth table

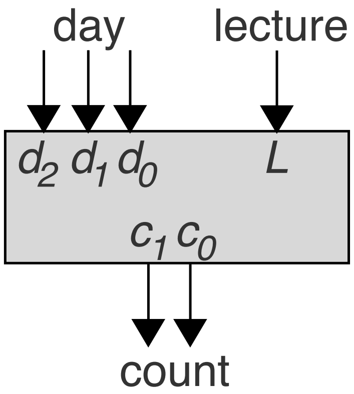
```
switch (day) {
  case 0: // SUNDAY
  case 1: // MONDAY
      return lecture ? 3 : 1;
  case 2: // TUESDAY
  case 3: // WEDNESDAY
      return lecture ? 2 : 1;
  case 4: // THURSDAY
      return lecture ? 1 : 1;
  case 5: // FRIDAY
      return lecture ? 1 : 0;
  default: // SATURDAY etc.
      return lecture ? 0 : 0;
}
```



Day	$d_2d_1d_0$	L	c_1c_0
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	01
THU	100	1	01
FRI	101	0	
FRI	101	1	
SAT	110	0	
SAT	110	1	
-	111	0	
-	111	1	

From code to combinational logic via a truth table

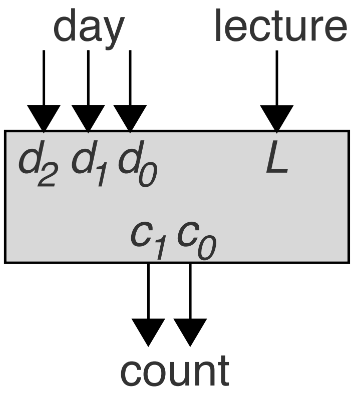
```
switch (day) {
  case 0: // SUNDAY
  case 1: // MONDAY
      return lecture ? 3 : 1;
  case 2: // TUESDAY
  case 3: // WEDNESDAY
      return lecture ? 2 : 1;
  case 4: // THURSDAY
      return lecture ? 1 : 1;
  case 5: // FRIDAY
      return lecture ? 1 : 0;
  default: // SATURDAY etc.
      return lecture ? 0 : 0;
}
```



Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	01
THU	100	1	01
FRI	101	0	00
FRI	101	1	01
SAT	110	0	
SAT	110	1	
-	111	0	
-	111	1	

From code to combinational logic via a truth table

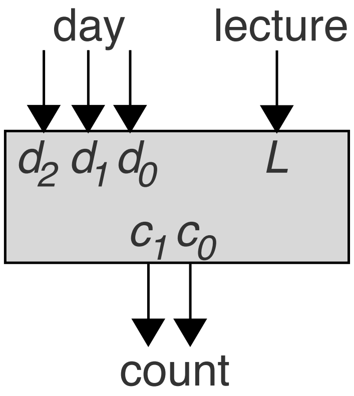
```
switch (day) {
  case 0: // SUNDAY
  case 1: // MONDAY
      return lecture ? 3 : 1;
  case 2: // TUESDAY
  case 3: // WEDNESDAY
      return lecture ? 2 : 1;
  case 4: // THURSDAY
      return lecture ? 1 : 1;
  case 5: // FRIDAY
      return lecture ? 1 : 0;
  default: // SATURDAY etc.
      return lecture ? 0 : 0;
}
```



Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	01
THU	100	1	01
FRI	101	0	00
FRI	101	1	01
SAT	110	0	00
SAT	110	1	00
-	111	0	
-	111	1	

From code to combinational logic via a truth table

```
switch (day) {
  case 0: // SUNDAY
  case 1: // MONDAY
      return lecture ? 3 : 1;
  case 2: // TUESDAY
  case 3: // WEDNESDAY
      return lecture ? 2 : 1;
  case 4: // THURSDAY
      return lecture ? 1 : 1;
  case 5: // FRIDAY
      return lecture ? 1 : 0;
  default: // SATURDAY etc.
      return lecture ? 0 : 0;
}
```



Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	01
THU	100	1	01
FRI	101	0	00
FRI	101	1	01
SAT	110	0	00
SAT	110	1	00
-	111	0	00
-	111	1	00

From code to combinational logic via a truth table: c_1

Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	01
THU	100	1	01
FRI	101	0	00
FRI	101	1	01
SAT	110	0	00
SAT	110	1	00
-	111	0	00
-	111	1	00

To find an expression for c_1, look at the rows where $c_1 = 1$.

- $d_2 d_1 d_0$`==000 && L==1`
- $d_2 d_1 d_0$`==001 && L==1`
- $d_2 d_1 d_0$`==010 && L==1`
- $d_2 d_1 d_0$`==011 && L==1`

From code to combinational logic via a truth table: c_1

Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	01
THU	100	1	01
FRI	101	0	00
FRI	101	1	01
SAT	110	0	00
SAT	110	1	00
-	111	0	00
-	111	1	00

To find an expression for c_1, look at the rows where $c_1 = 1$.

- $d_2 d_1 d_0$`==000 && L==1`
- $d_2 d_1 d_0$`==001 && L==1`
- $d_2 d_1 d_0$`==010 && L==1`
- $d_2 d_1 d_0$`==011 && L==1`

Split up the bits of the day to get a formula for each row.

- d_2`==0 &&` d_1`==0 &&` d_0`==0 && L==1`
- d_2`==0 &&` d_1`==0 &&` d_0`==1 && L==1`
- d_2`==0 &&` d_1`==1 &&` d_0`==0 && L==1`
- d_2`==0 &&` d_1`==1 &&` d_0`==1 && L==1`

From code to combinational logic via a truth table: c_1

Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	01
THU	100	1	01
FRI	101	0	00
FRI	101	1	01
SAT	110	0	00
SAT	110	1	00
-	111	0	00
-	111	1	00

To find an expression for c_1, look at the rows where $c_1 = 1$.

- $d_2 d_1 d_0$`==000 && L==1`
- $d_2 d_1 d_0$`==001 && L==1`
- $d_2 d_1 d_0$`==010 && L==1`
- $d_2 d_1 d_0$`==011 && L==1`

Split up the bits of the day to get a formula for each row.

- d_2`==0 &&` d_1`==0 &&` d_0`==0 && L==1`
- d_2`==0 &&` d_1`==0 &&` d_0`==1 && L==1`
- d_2`==0 &&` d_1`==1 &&` d_0`==0 && L==1`
- d_2`==0 &&` d_1`==1 &&` d_0`==1 && L==1`

Translate to Boolean algebra to get an expression for c_1.

- $d_2' \cdot d_1' \cdot d_0' \cdot L$
- $d_2' \cdot d_1' \cdot d_0 \cdot L$
- $d_2' \cdot d_1 \cdot d_0' \cdot L$
- $d_2' \cdot d_1 \cdot d_0 \cdot L$

From code to combinational logic via a truth table: c_1

Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	01
THU	100	1	01
FRI	101	0	00
FRI	101	1	01
SAT	110	0	00
SAT	110	1	00
-	111	0	00
-	111	1	00

To find an expression for c_1, look at the rows where $c_1 = 1$.

- $d_2 d_1 d_0$`==000 && L==1`
- $d_2 d_1 d_0$`==001 && L==1`
- $d_2 d_1 d_0$`==010 && L==1`
- $d_2 d_1 d_0$`==011 && L==1`

Split up the bits of the day to get a formula for each row.

- d_2`==0 &&` d_1`==0 &&` d_0`==0 && L==1`
- d_2`==0 &&` d_1`==0 &&` d_0`==1 && L==1`
- d_2`==0 &&` d_1`==1 &&` d_0`==0 && L==1`
- d_2`==0 &&` d_1`==1 &&` d_0`==1 && L==1`

Translate to Boolean algebra to get an expression for c_1.

- $d_2' \cdot d_1' \cdot d_0' \cdot L$
- $d_2' \cdot d_1' \cdot d_0 \cdot L$
- $d_2' \cdot d_1 \cdot d_0' \cdot L$
- $d_2' \cdot d_1 \cdot d_0 \cdot L$

$$\begin{aligned} c_1 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L + \\ & d_2' \cdot d_1' \cdot d_0 \cdot L + \\ & d_2' \cdot d_1 \cdot d_0' \cdot L + \\ & d_2' \cdot d_1 \cdot d_0 \cdot L \end{aligned}$$

From code to combinational logic via a truth table: c_0

Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	01
THU	100	1	01
FRI	101	0	00
FRI	101	1	01
SAT	110	0	00
SAT	110	1	00
-	111	0	00
-	111	1	00

$$\begin{aligned} c_1 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L + \\ & d_2' \cdot d_1' \cdot d_0 \cdot L + \\ & d_2' \cdot d_1 \cdot d_0' \cdot L + \\ & d_2' \cdot d_1 \cdot d_0 \cdot L \end{aligned}$$

Now we repeat this process to get c_0.

$c_0 =$

From code to combinational logic via a truth table: c_0

Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	01
THU	100	1	01
FRI	101	0	00
FRI	101	1	01
SAT	110	0	00
SAT	110	1	00
-	111	0	00
-	111	1	00

$$\begin{aligned} c_1 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L + \\ & d_2' \cdot d_1' \cdot d_0 \cdot L + \\ & d_2' \cdot d_1 \cdot d_0' \cdot L + \\ & d_2' \cdot d_1 \cdot d_0 \cdot L \end{aligned}$$

Now we repeat this process to get c_0.

$$\begin{aligned} c_0 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L' + \\ & d_2' \cdot d_1' \cdot d_0' \cdot L + \end{aligned}$$

From code to combinational logic via a truth table: c_0

Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	01
THU	100	1	01
FRI	101	0	00
FRI	101	1	01
SAT	110	0	00
SAT	110	1	00
-	111	0	00
-	111	1	00

$$
\begin{aligned}
c_1 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L + \\
& d_2' \cdot d_1' \cdot d_0 \cdot L + \\
& d_2' \cdot d_1 \cdot d_0' \cdot L + \\
& d_2' \cdot d_1 \cdot d_0 \cdot L
\end{aligned}
$$

Now we repeat this process to get c_0.

$$
\begin{aligned}
c_0 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L' + \\
& d_2' \cdot d_1' \cdot d_0' \cdot L + \\
& d_2' \cdot d_1' \cdot d_0 \cdot L' + \\
& d_2' \cdot d_1' \cdot d_0 \cdot L +
\end{aligned}
$$

From code to combinational logic via a truth table: c_0

Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	01
THU	100	1	01
FRI	101	0	00
FRI	101	1	01
SAT	110	0	00
SAT	110	1	00
-	111	0	00
-	111	1	00

$$\begin{aligned} c_1 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L + \\ & d_2' \cdot d_1' \cdot d_0 \cdot L + \\ & d_2' \cdot d_1 \cdot d_0' \cdot L + \\ & d_2' \cdot d_1 \cdot d_0 \cdot L \end{aligned}$$

Now we repeat this process to get c_0.

$$\begin{aligned} c_0 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L' + \\ & d_2' \cdot d_1' \cdot d_0' \cdot L + \\ & d_2' \cdot d_1' \cdot d_0 \cdot L' + \\ & d_2' \cdot d_1' \cdot d_0 \cdot L + \\ & d_2' \cdot d_1 \cdot d_0' \cdot L' + \end{aligned}$$

From code to combinational logic via a truth table: c_0

Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	01
THU	100	1	01
FRI	101	0	00
FRI	101	1	01
SAT	110	0	00
SAT	110	1	00
-	111	0	00
-	111	1	00

$$
\begin{aligned}
c_1 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L + \\
& d_2' \cdot d_1' \cdot d_0 \cdot L + \\
& d_2' \cdot d_1 \cdot d_0' \cdot L + \\
& d_2' \cdot d_1 \cdot d_0 \cdot L
\end{aligned}
$$

Now we repeat this process to get c_0.

$$
\begin{aligned}
c_0 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L' + \\
& d_2' \cdot d_1' \cdot d_0' \cdot L + \\
& d_2' \cdot d_1' \cdot d_0 \cdot L' + \\
& d_2' \cdot d_1' \cdot d_0 \cdot L + \\
& d_2' \cdot d_1 \cdot d_0' \cdot L' + \\
& d_2' \cdot d_1 \cdot d_0 \cdot L' +
\end{aligned}
$$

From code to combinational logic via a truth table: c_0

Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	01
THU	100	1	01
FRI	101	0	00
FRI	101	1	01
SAT	110	0	00
SAT	110	1	00
-	111	0	00
-	111	1	00

$$
\begin{aligned}
c_1 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L + \\
& d_2' \cdot d_1' \cdot d_0 \cdot L + \\
& d_2' \cdot d_1 \cdot d_0' \cdot L + \\
& d_2' \cdot d_1 \cdot d_0 \cdot L
\end{aligned}
$$

Now we repeat this process to get c_0.

$$
\begin{aligned}
c_0 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L' + \\
& d_2' \cdot d_1' \cdot d_0' \cdot L + \\
& d_2' \cdot d_1' \cdot d_0 \cdot L' + \\
& d_2' \cdot d_1' \cdot d_0 \cdot L + \\
& d_2' \cdot d_1 \cdot d_0' \cdot L' + \\
& d_2' \cdot d_1 \cdot d_0 \cdot L' + \\
& d_2 \cdot d_1' \cdot d_0' \cdot L' + \\
& d_2 \cdot d_1' \cdot d_0' \cdot L +
\end{aligned}
$$

From code to combinational logic via a truth table: c_0

Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	01
THU	100	1	01
FRI	101	0	00
FRI	101	1	01
SAT	110	0	00
SAT	110	1	00
-	111	0	00
-	111	1	00

$$\begin{aligned} c_1 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L + \\ & d_2' \cdot d_1' \cdot d_0 \cdot L + \\ & d_2' \cdot d_1 \cdot d_0' \cdot L + \\ & d_2' \cdot d_1 \cdot d_0 \cdot L \end{aligned}$$

Now we repeat this process to get c_0.

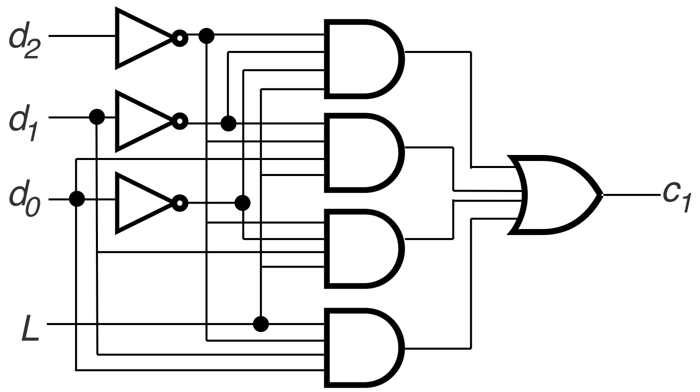
$$\begin{aligned} c_0 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L' + \\ & d_2' \cdot d_1' \cdot d_0' \cdot L + \\ & d_2' \cdot d_1' \cdot d_0 \cdot L' + \\ & d_2' \cdot d_1' \cdot d_0 \cdot L + \\ & d_2' \cdot d_1 \cdot d_0' \cdot L' + \\ & d_2' \cdot d_1 \cdot d_0 \cdot L' + \\ & d_2 \cdot d_1' \cdot d_0' \cdot L' + \\ & d_2 \cdot d_1' \cdot d_0' \cdot L + \\ & d_2 \cdot d_1' \cdot d_0 \cdot L \end{aligned}$$

From combinational logic to circuits

$$\begin{aligned} c_0 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L' + \\ & d_2' \cdot d_1' \cdot d_0' \cdot L + \\ & d_2' \cdot d_1' \cdot d_0 \cdot L' + \\ & d_2' \cdot d_1' \cdot d_0 \cdot L + \\ & d_2' \cdot d_1 \cdot d_0' \cdot L' + \\ & d_2' \cdot d_1 \cdot d_0 \cdot L' + \\ & d_2 \cdot d_1' \cdot d_0' \cdot L' + \\ & d_2 \cdot d_1' \cdot d_0' \cdot L + \\ & d_2 \cdot d_1' \cdot d_0 \cdot L \end{aligned}$$

$$\begin{aligned} c_1 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L + \\ & d_2' \cdot d_1' \cdot d_0 \cdot L + \\ & d_2' \cdot d_1 \cdot d_0' \cdot L + \\ & d_2' \cdot d_1 \cdot d_0 \cdot L \end{aligned}$$

Here is c_1 as a circuit …



What can we do with the logic encoding?

Create hardware implementations!

And perform program verification …

Example: verify that `classesLeft` returns 3 only if `lecture` is true.

$$c_0 = d_2' \cdot d_1' \cdot d_0' \cdot L' + d_2' \cdot d_1' \cdot d_0' \cdot L + d_2' \cdot d_1' \cdot d_0 \cdot L' + d_2' \cdot d_1' \cdot d_0 \cdot L + d_2' \cdot d_1 \cdot d_0' \cdot L' + d_2' \cdot d_1 \cdot d_0 \cdot L' + d_2 \cdot d_1' \cdot d_0' \cdot L' + d_2 \cdot d_1' \cdot d_0' \cdot L + d_2 \cdot d_1' \cdot d_0 \cdot L$$

$$c_1 = d_2' \cdot d_1' \cdot d_0' \cdot L + d_2' \cdot d_1' \cdot d_0 \cdot L + d_2' \cdot d_1 \cdot d_0' \cdot L + d_2' \cdot d_1 \cdot d_0 \cdot L$$

Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	01
THU	100	1	01
FRI	101	0	00
FRI	101	1	01
SAT	110	0	00
SAT	110	1	00
-	111	0	00
-	111	1	00

What can we do with the logic encoding?

Create hardware implementations!

And perform program verification …

Example: verify that `classesLeft` returns 3 only if `lecture` is true.

$p, q,$ and r represent propositions

$c_1 = 1, c_0 = 1,$ and $L = 1.$

Check that $p \wedge q \rightarrow r \equiv T$

$$\begin{aligned} c_0 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L' + \\ & d_2' \cdot d_1' \cdot d_0' \cdot L + \\ & d_2' \cdot d_1' \cdot d_0 \cdot L' + \\ & d_2' \cdot d_1' \cdot d_0 \cdot L + \\ & d_2' \cdot d_1 \cdot d_0' \cdot L' + \\ & d_2' \cdot d_1 \cdot d_0 \cdot L' + \\ & d_2 \cdot d_1' \cdot d_0' \cdot L' + \\ & d_2 \cdot d_1' \cdot d_0' \cdot L + \\ & d_2 \cdot d_1' \cdot d_0 \cdot L \end{aligned}$$

$$\begin{aligned} c_1 = {} & d_2' \cdot d_1' \cdot d_0' \cdot L + \\ & d_2' \cdot d_1' \cdot d_0 \cdot L + \\ & d_2' \cdot d_1 \cdot d_0' \cdot L + \\ & d_2' \cdot d_1 \cdot d_0 \cdot L \end{aligned}$$

Day	$d_2 d_1 d_0$	L	$c_1 c_0$
SUN	000	0	01
SUN	000	1	11
MON	001	0	01
MON	001	1	11
TUE	010	0	01
TUE	010	1	10
WED	011	0	01
WED	011	1	10
THU	100	0	01
THU	100	1	01
FRI	101	0	00
FRI	101	1	01
SAT	110	0	00
SAT	110	1	00
-	111	0	00
-	111	1	00

Simplification and proofs

Optimizing circuits and proving theorems.

So far, we've used the basics of Boolean algebra ...

Boolean algebra consists of the following elements and operations:

- a set of elements $B = \{0, 1\}$,
- binary operations $\{+, \cdot\}$,
- a unary operation $\{\,'\,\}$.

These correspond to the truth values {F, T}, and the logical connectives $\vee$, $\wedge$, $\neg$.

Boolean operations satisfy the following axioms for any $a, b, c \in B$:

Closure

$$a + b \in B$$
$$a \cdot b \in B$$

Commutativity

$$a + b = b + a$$
$$a \cdot b = b \cdot a$$

Associativity

$$a + (b + c) = (a + b) + c$$
$$a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

Distributivity

$$a + (b \cdot c) = (a + b) \cdot (a + c)$$
$$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

Identity

$$a + 0 = a$$
$$a \cdot 1 = a$$

Complementarity

$$a + a' = 1$$
$$a \cdot a' = 0$$

Null

$$a + 1 = 1$$
$$a \cdot 0 = 0$$

Idempotency

$$a + a = a$$
$$a \cdot a = a$$

Involution

$$(a')' = a$$

We can use the basics to prove some useful theorems

Uniting

$$a \cdot b + a \cdot b' = a$$
$$(a + b) \cdot (a + b') = a$$

Absorption

$$a + a \cdot b = a$$
$$a \cdot (a + b) = a$$
$$(a + b') \cdot b = a \cdot b$$
$$(a \cdot b') + b = a + b$$

Factoring

$$(a + b) \cdot (a' + c) = a \cdot c + a' \cdot b$$
$$a \cdot b + a' \cdot c = (a + c) \cdot (a' + b)$$

Consensus

$$(a \cdot b) + (b \cdot c) + (a' \cdot c) = a \cdot b + a' \cdot c$$
$$(a + b) \cdot (b + c) \cdot (a' + c) = (a + b) \cdot (a' + c)$$

DeMorgan's

$$(a + b + \ldots)' = a' \cdot b' \cdot \ldots$$
$$(a \cdot b \cdot \ldots)' = a' + b' + \ldots$$

Example: proving theorems using the axioms

Uniting

$$\begin{aligned} X \cdot Y + X \cdot Y' &= \\ &= \\ &= X \end{aligned}$$

Absorption

$$\begin{aligned} X + X \cdot Y &= \\ &= \\ &= \\ &= \\ &= X \end{aligned}$$

Closure

$a + b \in B$

$a \cdot b \in B$

Commutativity

$a + b = b + a$

$a \cdot b = b \cdot a$

Associativity

$a + (b + c) = (a + b) + c$

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

Distributivity

$a + (b \cdot c) = (a + b) \cdot (a + c)$

$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

Identity

$a + 0 = a$

$a \cdot 1 = a$

Complementarity

$a + a' = 1$

$a \cdot a' = 0$

Null

$a + 1 = 1$

$a \cdot 0 = 0$

Idempotency

$a + a = a$

$a \cdot a = a$

Involution

$(a')' = a$

Example: proving theorems using the axioms

Uniting

$$\begin{aligned} X \cdot Y + X \cdot Y' &= X \cdot (Y + Y') \quad \textbf{Distributivity} \\ &= \\ &= X \end{aligned}$$

Absorption

$$\begin{aligned} X + X \cdot Y &= \\ &= \\ &= \\ &= \\ &= X \end{aligned}$$

Closure

$a + b \in B$

$a \cdot b \in B$

Commutativity

$a + b = b + a$

$a \cdot b = b \cdot a$

Associativity

$a + (b + c) = (a + b) + c$

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

Distributivity

$a + (b \cdot c) = (a + b) \cdot (a + c)$

$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

Identity

$a + 0 = a$

$a \cdot 1 = a$

Complementarity

$a + a' = 1$

$a \cdot a' = 0$

Null

$a + 1 = 1$

$a \cdot 0 = 0$

Idempotency

$a + a = a$

$a \cdot a = a$

Involution

$(a')' = a$

Example: proving theorems using the axioms

Uniting

$$
\begin{aligned}
X \cdot Y + X \cdot Y' &= X \cdot (Y + Y') && \text{Distributivity} \\
&= X \cdot 1 && \text{Complementarity} \\
&= X
\end{aligned}
$$

Absorption

$$
\begin{aligned}
X + X \cdot Y &= \\
&= \\
&= \\
&= \\
&= X
\end{aligned}
$$

Closure

$a + b \in B$

$a \cdot b \in B$

Commutativity

$a + b = b + a$

$a \cdot b = b \cdot a$

Associativity

$a + (b + c) = (a + b) + c$

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

Distributivity

$a + (b \cdot c) = (a + b) \cdot (a + c)$

$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

Identity

$a + 0 = a$

$a \cdot 1 = a$

Complementarity

$a + a' = 1$

$a \cdot a' = 0$

Null

$a + 1 = 1$

$a \cdot 0 = 0$

Idempotency

$a + a = a$

$a \cdot a = a$

Involution

$(a')' = a$

Example: proving theorems using the axioms

Uniting

$$
\begin{aligned}
X \cdot Y + X \cdot Y' &= X \cdot (Y + Y') && \text{Distributivity} \\
&= X \cdot 1 && \text{Complementarity} \\
&= X && \text{Identity}
\end{aligned}
$$

Absorption

$$
\begin{aligned}
X + X \cdot Y &= \\
&= \\
&= \\
&= \\
&= X
\end{aligned}
$$

Closure

$a + b \in B$

$a \cdot b \in B$

Commutativity

$a + b = b + a$

$a \cdot b = b \cdot a$

Associativity

$a + (b + c) = (a + b) + c$

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

Distributivity

$a + (b \cdot c) = (a + b) \cdot (a + c)$

$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

Identity

$a + 0 = a$

$a \cdot 1 = a$

Complementarity

$a + a' = 1$

$a \cdot a' = 0$

Null

$a + 1 = 1$

$a \cdot 0 = 0$

Idempotency

$a + a = a$

$a \cdot a = a$

Involution

$(a')' = a$

Example: proving theorems using the axioms

Uniting

$$
\begin{aligned}
X \cdot Y + X \cdot Y' &= X \cdot (Y + Y') && \text{Distributivity} \\
&= X \cdot 1 && \text{Complementarity} \\
&= X && \text{Identity}
\end{aligned}
$$

Absorption

$$
\begin{aligned}
X + X \cdot Y &= X \cdot 1 + X \cdot Y && \text{Identity} \\
&= \\
&= \\
&= \\
&= X
\end{aligned}
$$

Closure

$a + b \in B$

$a \cdot b \in B$

Commutativity

$a + b = b + a$

$a \cdot b = b \cdot a$

Associativity

$a + (b + c) = (a + b) + c$

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

Distributivity

$a + (b \cdot c) = (a + b) \cdot (a + c)$

$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

Identity

$a + 0 = a$

$a \cdot 1 = a$

Complementarity

$a + a' = 1$

$a \cdot a' = 0$

Null

$a + 1 = 1$

$a \cdot 0 = 0$

Idempotency

$a + a = a$

$a \cdot a = a$

Involution

$(a')' = a$

Example: proving theorems using the axioms

Uniting

$$\begin{aligned} X \cdot Y + X \cdot Y' &= X \cdot (Y + Y') && \text{Distributivity} \\ &= X \cdot 1 && \text{Complementarity} \\ &= X && \text{Identity} \end{aligned}$$

Absorption

$$\begin{aligned} X + X \cdot Y &= X \cdot 1 + X \cdot Y && \text{Identity} \\ &= X \cdot (1 + Y) && \text{Distributivity} \\ &= \\ &= \\ &= X \end{aligned}$$

Closure

$a + b \in B$

$a \cdot b \in B$

Commutativity

$a + b = b + a$

$a \cdot b = b \cdot a$

Associativity

$a + (b + c) = (a + b) + c$

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

Distributivity

$a + (b \cdot c) = (a + b) \cdot (a + c)$

$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

Identity

$a + 0 = a$

$a \cdot 1 = a$

Complementarity

$a + a' = 1$

$a \cdot a' = 0$

Null

$a + 1 = 1$

$a \cdot 0 = 0$

Idempotency

$a + a = a$

$a \cdot a = a$

Involution

$(a')' = a$

Example: proving theorems using the axioms

Uniting

$$\begin{aligned} X \cdot Y + X \cdot Y' &= X \cdot (Y + Y') && \text{Distributivity} \\ &= X \cdot 1 && \text{Complementarity} \\ &= X && \text{Identity} \end{aligned}$$

Absorption

$$\begin{aligned} X + X \cdot Y &= X \cdot 1 + X \cdot Y && \text{Identity} \\ &= X \cdot (1 + Y) && \text{Distributivity} \\ &= X \cdot (Y + 1) && \text{Commutativity} \\ &= \\ &= X \end{aligned}$$

Closure

$a + b \in B$

$a \cdot b \in B$

Commutativity

$a + b = b + a$

$a \cdot b = b \cdot a$

Associativity

$a + (b + c) = (a + b) + c$

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

Distributivity

$a + (b \cdot c) = (a + b) \cdot (a + c)$

$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

Identity

$a + 0 = a$

$a \cdot 1 = a$

Complementarity

$a + a' = 1$

$a \cdot a' = 0$

Null

$a + 1 = 1$

$a \cdot 0 = 0$

Idempotency

$a + a = a$

$a \cdot a = a$

Involution

$(a')' = a$

Example: proving theorems using the axioms

Uniting

$$\begin{aligned} X \cdot Y + X \cdot Y' &= X \cdot (Y + Y') && \text{Distributivity} \\ &= X \cdot 1 && \text{Complementarity} \\ &= X && \text{Identity} \end{aligned}$$

Absorption

$$\begin{aligned} X + X \cdot Y &= X \cdot 1 + X \cdot Y && \text{Identity} \\ &= X \cdot (1 + Y) && \text{Distributivity} \\ &= X \cdot (Y + 1) && \text{Commutativity} \\ &= X \cdot 1 && \text{Null} \\ &= X \end{aligned}$$

Closure

$a + b \in B$

$a \cdot b \in B$

Commutativity

$a + b = b + a$

$a \cdot b = b \cdot a$

Associativity

$a + (b + c) = (a + b) + c$

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

Distributivity

$a + (b \cdot c) = (a + b) \cdot (a + c)$

$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

Identity

$a + 0 = a$

$a \cdot 1 = a$

Complementarity

$a + a' = 1$

$a \cdot a' = 0$

Null

$a + 1 = 1$

$a \cdot 0 = 0$

Idempotency

$a + a = a$

$a \cdot a = a$

Involution

$(a')' = a$

Example: proving theorems using the axioms

Uniting

$$\begin{aligned} X \cdot Y + X \cdot Y' &= X \cdot (Y + Y') && \text{Distributivity} \\ &= X \cdot 1 && \text{Complementarity} \\ &= X && \text{Identity} \end{aligned}$$

Absorption

$$\begin{aligned} X + X \cdot Y &= X \cdot 1 + X \cdot Y && \text{Identity} \\ &= X \cdot (1 + Y) && \text{Distributivity} \\ &= X \cdot (Y + 1) && \text{Commutativity} \\ &= X \cdot 1 && \text{Null} \\ &= X && \text{Identity} \end{aligned}$$

Closure

$a + b \in B$

$a \cdot b \in B$

Commutativity

$a + b = b + a$

$a \cdot b = b \cdot a$

Associativity

$a + (b + c) = (a + b) + c$

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

Distributivity

$a + (b \cdot c) = (a + b) \cdot (a + c)$

$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

Identity

$a + 0 = a$

$a \cdot 1 = a$

Complementarity

$a + a' = 1$

$a \cdot a' = 0$

Null

$a + 1 = 1$

$a \cdot 0 = 0$

Idempotency

$a + a = a$

$a \cdot a = a$

Involution

$(a')' = a$

Example: proving theorems using truth tables

DeMorgan's law

$(X + Y)' = X' \cdot Y'$

NOR is equivalent to AND with inputs complemented

X	Y	X'	Y'	$(X+Y)'$	$X' \cdot Y'$
0	0	1	1	1	1
0	1	1	0	0	0
1	0	0	1	0	0
1	1	0	0	0	0

DeMorgan's law

$(X \cdot Y)' = X' + Y'$

NAND is equivalent to OR with inputs complemented

X	Y	X'	Y'	$(X \cdot Y)'$	$X' + Y'$
0	0	1	1	1	1
0	1	1	0	1	1
1	0	0	1	1	1
1	1	0	0	0	0

Example: simplifying (circuits) using Boolean algebra

$$
\begin{aligned}
c_1 &= d_2' \cdot d_1' \cdot d_0' \cdot L + d_2' \cdot d_1' \cdot d_0 \cdot L + d_2' \cdot d_1 \cdot d_0' \cdot L + d_2' \cdot d_1 \cdot d_0 \cdot L && \text{(from classesLeft)} \\
&= \ldots && \text{(HW2)} \\
&= d_2' \cdot L && \text{(HW2)}
\end{aligned}
$$

Closure

$a + b \in B$

$a \cdot b \in B$

Commutativity

$a + b = b + a$

$a \cdot b = b \cdot a$

Associativity

$a + (b + c) = (a + b) + c$

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

Distributivity

$a + (b \cdot c) = (a + b) \cdot (a + c)$

$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

Identity

$a + 0 = a$

$a \cdot 1 = a$

Complementarity

$a + a' = 1$

$a \cdot a' = 0$

Null

$a + 1 = 1$

$a \cdot 0 = 0$

Idempotency

$a + a = a$

$a \cdot a = a$

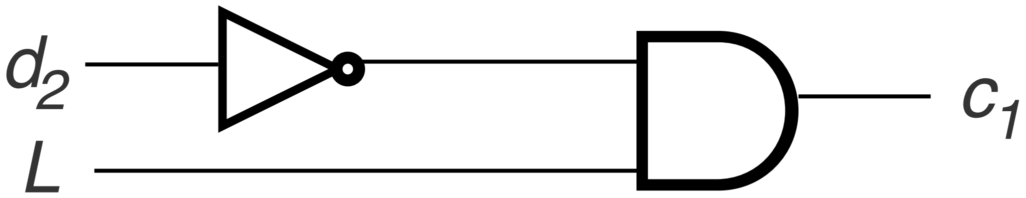
Involution

$(a')' = a$

Example: simplifying (circuits) using Boolean algebra

$$
\begin{aligned}
c_1 &= d_2' \cdot d_1' \cdot d_0' \cdot L + d_2' \cdot d_1' \cdot d_0 \cdot L + d_2' \cdot d_1 \cdot d_0' \cdot L + d_2' \cdot d_1 \cdot d_0 \cdot L && \text{(from classesLeft)} \\
&= \ldots && \text{(HW2)} \\
&= d_2' \cdot L && \text{(HW2)}
\end{aligned}
$$

Here is the simplified c_1 circuit …



Summary

Boolean algebra is a notation for combinational circuits.

It consists of elements $\{0, 1\}$ and operations $\{+, \cdot, '\}$.

The operations satisfy the axioms of Boolean algebra.

We can translate specs to code to logic and to circuits for

faster implementation in hardware, and

program verification.

We can use axioms of Boolean algebra and truth tables to

prove useful theorems, and

simplify and optimize combinational circuits.