

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 $(A \leftrightarrow B) \equiv T$ have the same meaning.
tautology

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.

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

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

p	q	$p \wedge q$	$q \lor p$	$(p \land q) \to (q \lor p)$
F	F	F	F	Т
F	Т	F	Т	Т
Т	F	F	Т	Т
Т	Т	Т	Т	Т

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

> 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 $p \rightarrow q \equiv \neg p \lor q$ Contrapositive $p \rightarrow q \equiv \neg q \rightarrow \neg p$ Biconditional $p \leftrightarrow q \equiv (p \rightarrow q) \land (q \rightarrow p)$ 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

 $\begin{array}{l} p \wedge p \equiv p \\ p \lor p \equiv p \end{array}$

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

 $\begin{array}{l} p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r) \\ p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r) \end{array}$

Absorption

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

Negation

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

 $(p \land q) \rightarrow (q \lor p) \equiv \neg (p \land q) \lor (q \lor p)$ Law of implication \equiv \equiv

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 $p \rightarrow q \equiv \neg p \lor q$ Contrapositive $p \rightarrow q \equiv \neg q \rightarrow \neg p$ Biconditional $p \leftrightarrow q \equiv (p \rightarrow q) \land (q \rightarrow p)$ 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

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

$$(p \land q) \rightarrow (q \lor p) \equiv \neg (p \land q) \lor (q \lor p)$$
Law of implication
$$\equiv (\neg p \lor \neg q) \lor (q \lor p)$$
DeMorgan
$$\equiv$$
$$\equiv$$
$$\equiv$$
$$\equiv$$
$$\equiv$$
$$\equiv$$
$$\equiv$$
$$\equiv$$
$$\equiv$$

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 $p \rightarrow q \equiv \neg p \lor q$ Contrapositive $p \rightarrow q \equiv \neg q \rightarrow \neg p$ Biconditional $p \leftrightarrow q \equiv (p \rightarrow q) \land (q \rightarrow p)$ Double negation

 $p \equiv \neg \neg p$

dentity			
$p \land$	Т	≡	р

Domination

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

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

Idempotence

 $\begin{array}{l} p \wedge p \equiv p \\ p \lor p \equiv p \end{array}$

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

 $\begin{array}{l} p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r) \\ p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r) \end{array}$

Absorption

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

Negation

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

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 $p \rightarrow q \equiv \neg p \lor q$ Contrapositive $p \rightarrow q \equiv \neg q \rightarrow \neg p$ Biconditional $p \leftrightarrow q \equiv (p \rightarrow q) \land (q \rightarrow p)$ Double negation

 $p \equiv \neg \neg p$

Identity			
$p \wedge$	Т	\equiv	р

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

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

 $\begin{array}{l} p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r) \\ p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r) \end{array}$

Absorption

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

Negation

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

$$(p \land q) \rightarrow (q \lor p) \equiv \neg (p \land q) \lor (q \lor p)$$
Law of implication
$$\equiv (\neg p \lor \neg q) \lor (q \lor p)$$
DeMorgan
$$\equiv \neg p \lor (\neg q \lor (q \lor p))$$
Associativity
$$\equiv \neg p \lor ((\neg q \lor q) \lor p)$$
Associativity
$$\equiv$$
$$\equiv$$
$$\equiv$$
$$\equiv$$
$$\equiv$$
$$\equiv$$

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 $p \rightarrow q \equiv \neg p \lor q$ Contrapositive **Biconditional**

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

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

Double negation

 $p \equiv \neg \neg p$

dentity			
$p \land$	Т	≡	р

Domination

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

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

Idempotence

 $p \wedge p \equiv p$ $p \lor p \equiv p$

Commutativity

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

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

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

$$(p \land q) \rightarrow (q \lor p) \equiv \neg (p \land q) \lor (q \lor p)$$
Law of implication
$$\equiv (\neg p \lor \neg q) \lor (q \lor p)$$
DeMorgan
$$\equiv \neg p \lor (\neg q \lor (q \lor p))$$
Associativity
$$\equiv \neg p \lor ((\neg q \lor q) \lor p)$$
Associativity
$$\equiv \neg p \lor (p \lor (\neg q \lor q))$$
Commutativity
$$\equiv$$

 $\equiv T$

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

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

Biconditional

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

Double negation

 $p \equiv \neg \neg p$

Identity						
$p \wedge T$	≡	р				
$p \lor F$	\equiv	р				

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

 $\begin{array}{l} p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r) \\ p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r) \end{array}$

Absorption

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

Negation

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

$$\begin{array}{ll} (p \land q) \rightarrow (q \lor p) \equiv \neg (p \land q) \lor (q \lor p) & \text{Law of implication} \\ \equiv (\neg p \lor \neg q) \lor (q \lor p) & \text{DeMorgan} \\ \equiv \neg p \lor (\neg q \lor (q \lor p)) & \text{Associativity} \\ \equiv \neg p \lor ((\neg q \lor q) \lor p) & \text{Associativity} \\ \equiv \neg p \lor (p \lor (\neg q \lor q)) & \text{Commutativity} \\ \equiv (\neg p \lor p) \lor (\neg q \lor q) & \text{Associativity} \\ \equiv \\ \end{array}$$

 $\equiv T$

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

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

Biconditional

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

Double negation

 $p \equiv \neg \neg p$

Identity $p \land T \equiv p$ $p \lor F \equiv p$

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

 $\begin{array}{l} p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r) \\ p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r) \end{array}$

Absorption

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

Negation

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

$$(p \land q) \rightarrow (q \lor p) \equiv \neg (p \land q) \lor (q \lor p) \qquad \text{Law of implication} \\ \equiv (\neg p \lor \neg q) \lor (q \lor p) \qquad \text{DeMorgan} \\ \equiv \neg p \lor (\neg q \lor (q \lor p)) \qquad \text{Associativity} \\ \equiv \neg p \lor ((\neg q \lor q) \lor p) \qquad \text{Associativity} \\ \equiv \neg p \lor (p \lor (\neg q \lor q)) \qquad \text{Commutativity} \\ \equiv (\neg p \lor p) \lor (\neg q \lor q) \qquad \text{Associativity} \\ \equiv (p \lor \neg p) \lor (q \lor \neg q) \qquad \text{Commutativity (twice)} \\ \equiv \\ = \\ = \\ \end{array}$$

 $\equiv T$

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** $p \rightarrow q \equiv \neg p \lor q$

Contrapositive

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

Biconditional

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

Double negation

 $p \equiv \neg \neg p$

Identity							
р	\wedge	Т	≡	р			
р	V	F	\equiv	p			

Domination

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

Idempotence

 $\begin{array}{l} p \land p \equiv p \\ p \lor p \equiv p \end{array}$

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

 $\begin{array}{l} p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r) \\ p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r) \end{array}$

Absorption

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

Negation

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

$$(p \land q) \rightarrow (q \lor p) \equiv \neg (p \land q) \lor (q \lor p)$$
Law of imp

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

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

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

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

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

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

$$\equiv T \lor T$$
Negation (

Law of implication DeMorgan Associativity Associativity Commutativity Associativity Commutativity (twice) Negation (twice)

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

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

Double negation

 $p \equiv \neg \neg p$

Identity $p \land T \equiv p$ $p \lor F \equiv p$

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

 $\begin{array}{l} p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r) \\ p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r) \end{array}$

Absorption

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

Negation

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

$$(p \land q) \rightarrow (q \lor p) \equiv \neg (p \land q) \lor (q \lor p)$$
Law of

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

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

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

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

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

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

$$\equiv T \lor T$$
Negati

$$\equiv T$$
Idempo

Law of implication DeMorgan Associativity Associativity Commutativity Associativity Commutativity (twice) Negation (twice) Idempotence

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

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

Double negation

 $p \equiv \neg \neg p$

Identity $p \land T \equiv p$ $p \lor F \equiv p$

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

 $\begin{array}{l} p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r) \\ p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r) \end{array}$

Absorption

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

Negation

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 {+, ·},
- a unary operation { ' }.

These correspond to the truth values $\{F, T\}$, and the logical connectives \lor, \land, \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 {+, ·},
- a unary operation { ' }.

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

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

Closure $a + b \in B$ $a \cdot b \in B$	Distributivity $a + (b \cdot c) = (a + b) \cdot (a + c)$ $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$	Null $a + 1 = 1$ $a \cdot 0 = 0$
Commutativity a + b = b + a $a \cdot b = b \cdot a$	Identity a + 0 = a $a \cdot 1 = a$	Idempotency a + a = a $a \cdot a = a$
Associativity a + (b + c) = (a + b) + c $a \cdot (b \cdot c) = (a \cdot b) \cdot c$	Complementarity a + a' = 1 $a \cdot a' = 0$	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 ...

```
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 ...

```
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.



	Day	$d_2 d_1 d_0$	L	c_1c_0
<pre>switch (day) { case 0: // SUNDAY</pre>	SUN	000	0	
<pre>case 1: // MONDAY return lecture ? 3 : 1:</pre>	SUN	000	1	
case 2: // TUESDAY	MON	001	0	
return lecture ? 2 : 1;	MON	001	1	
<pre>case 4: // THURSDAY return lecture ? 1 : 1;</pre>	TUE	010	0	
case 5: // FRIDAY	TUE	010	1	
default: // SATURDAY etc.	WED	011	0	
<pre>return lecture ? 0 : 0; }</pre>	WED	011	1	
	THU	100	0	
day lecture	THU	100	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	

	Day	$d_2 d_1 d_0$	L	C_1C_0
<pre>switch (day) { case 0: // SUNDAY</pre>	SUN	000	0	01
<pre>case 1: // MONDAY return lecture ? 3 : 1;</pre>	SUN	000	1	11
case 2: // TUESDAY	MON	001	0	
return lecture ? 2 : 1;	MON	001	1	
<pre>case 4: // THURSDAY return lecture ? 1 : 1;</pre>	TUE	010	0	
case 5: // FRIDAY	TUE	010	1	
default: // SATURDAY etc.	WED	011	0	
<pre>return lecture ? 0 : 0; }</pre>	WED	011	1	
	THU	100	0	
day lecture	THU	100	1	



ION	001	1	
TUE	010	0	
TUE	010	1	
NED	011	0	
NED	011	1	
ΓHU	100	0	
ΓHU	100	1	
FRI	101	0	
FRI	101	1	
SAT	110	0	
SAT	110	1	
-	111	0	
-	111	1	

	Day	$d_2 d_1 d_0$	L	C_1C_0
case 0: // SUNDAY	SUN	000	0	01
<pre>case 1: // MONDAY return lecture ? 3 : 1:</pre>	SUN	000	1	11
case 2: // TUESDAY	MON	001	0	01
<pre>case 3: // WEDNESDAY return lecture ? 2 : 1;</pre>	MON	001	1	11
<pre>case 4: // THURSDAY return lecture ? 1 : 1;</pre>	TUE	010	0	
case 5: // FRIDAY	TUE	010	1	
default: // SATURDAY etc.	WED	011	0	
<pre>return lecture ? 0 : 0; }</pre>	WED	011	1	
	THU	100	0	
dav lecture	тин	100	1	



0011	000	-	<u> </u>
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	

	Day	$d_2 d_1 d_0$	Ì
<pre>switch (day) { case 0: // SUNDAY</pre>	SUN	000	(
<pre>case 1: // MONDAY return lecture ? 3 : 1:</pre>	SUN	000	
case 2: // TUESDAY	MON	001	(
return lecture ? 2 : 1;	MON	001	
<pre>case 4: // THURSDAY return lecture ? 1 : 1;</pre>	TUE	010	(
<pre>case 5: // FRIDAY return lecture ? 1 : 0:</pre>	TUE	010	
default: // SATURDAY etc.	WED	011	(
<pre>return lecture ? 0 : 0; }</pre>	WED	011	
	ТЦП	100	



Day	$d_2 d_1 d_0$	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	
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	

<pre>switch (day)</pre>	{				
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	et	C	•	
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	

<pre>switch (day)</pre>	{			
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	et	C	•
return	lecture ?	0	:	0;
3				



Day	$d_2 d_1 d_0$	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	

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



Day	$d_2 d_1 d_0$	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	00
FRI	101	1	01
SAT	110	0	
SAT	110	1	
-	111	0	
-	111	1	

switch (day)	{		
case 0: // .	SUNDAY		
case 1: // 1	MONDAY		
return	lecture ? 3	:	1;
case 2: //	TUESDAY		
case 3: //	WEDNESDAY		
return	lecture ? 2	:	1;
case 4: //	<i>THURSDAY</i>		
return	lecture ? 1	:	1;
case 5: // 1	FRIDAY		
return	lecture ? 1	:	0;
default: //	SATURDAY e	tc	•
return	lecture ? O	:	0;
3			



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	

<pre>switch (day)</pre>	{			
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	et	C.	•
return	lecture ?	0	:	0;
}				



Day	$d_2 d_1 d_0$	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	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 $c_1 = 1$.

- $d_2d_1d_0 == 000$ && L==1
- $d_2d_1d_0 == 001$ && L==1
- $d_2d_1d_0 == 010$ && L==1
- $d_2d_1d_0 == 011$ && L==1

Day	$d_2 d_1 d_0$	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	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 $c_1 = 1$.

- $d_2d_1d_0 == 000$ && L==1
- $d_2d_1d_0 == 001$ && L==1
- $d_2d_1d_0 == 010$ && L==1
- $d_2d_1d_0 == 011$ && 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

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 $c_1 = 1$.

- $d_2d_1d_0 == 000$ && L==1
- $d_2d_1d_0 == 001$ && L==1
- $d_2d_1d_0 == 010$ && 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 .

•
$$d'_2 \cdot d'_1 \cdot d'_0 \cdot L$$

• $d'_2 \cdot d'_1 \cdot d_0 \cdot L$

- $d'_2 \cdot d_1 \cdot d'_0 \cdot L$
- $d'_2 \cdot d_1 \cdot d_0 \cdot L$

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 $c_1 = 1$.

- $d_2d_1d_0 == 000$ && L==1
- $d_2d_1d_0 == 001$ && L==1
- $d_2d_1d_0 == 010$ && L==1
- $d_2d_1d_0 == 011$ && 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 .

•
$$d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L$$

• $d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L$
• $d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L$
• $d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L$

 $c_{1} = d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L +$ $d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L +$ $d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L +$ $d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L +$ $d'_{2} \cdot d_{1} \cdot d_{0} \cdot L$

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

$$c_1 = d'_2 \cdot d'_1 \cdot d'_0 \cdot L + d'_2 \cdot d'_1 \cdot d_0 \cdot L + d'_2 \cdot d_1 \cdot d'_0 \cdot L + d'_2 \cdot d_1 \cdot d'_0 \cdot L + d'_2 \cdot d_1 \cdot d_0 \cdot L$$

Now we repeat this process to get c_0 .

 $c_0 =$

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

$$c_1 = d'_2 \cdot d'_1 \cdot d'_0 \cdot L + d'_2 \cdot d'_1 \cdot d_0 \cdot L + d'_2 \cdot d_1 \cdot d'_0 \cdot L + d'_2 \cdot d_1 \cdot d'_0 \cdot L + d'_2 \cdot d_1 \cdot d_0 \cdot L$$

$$c_0 = d'_2 \cdot d'_1 \cdot d'_0 \cdot L' + d'_2 \cdot d'_1 \cdot d'_0 \cdot L +$$

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

$$c_{1} = d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L + d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d_{1} \cdot d_{0} \cdot L + d'_{2} \cdot d_{1} \cdot d_{0} \cdot L$$

$$c_{0} = d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L +$$

Day	$d_2 d_1 d_0$	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	00
FRI	101	1	01
SAT	110	0	00
SAT	110	1	00
-	111	0	00
-	111	1	00

$$c_1 = d'_2 \cdot d'_1 \cdot d'_0 \cdot L + d'_2 \cdot d'_1 \cdot d_0 \cdot L + d'_2 \cdot d_1 \cdot d'_0 \cdot L + d'_2 \cdot d_1 \cdot d_0 \cdot L + d'_2 \cdot d_1 \cdot d_0 \cdot L$$

$$c_{0} = d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L' +$$

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

$$c_{1} = d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L + d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d_{1} \cdot d_{0} \cdot L + d'_{2} \cdot d_{1} \cdot d_{0} \cdot L$$

$$c_{0} = d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L + d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L' +$$

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

$$c_1 = d'_2 \cdot d'_1 \cdot d'_0 \cdot L + d'_2 \cdot d'_1 \cdot d_0 \cdot L + d'_2 \cdot d_1 \cdot d'_0 \cdot L + d'_2 \cdot d_1 \cdot d_0 \cdot L + d'_2 \cdot d_1 \cdot d_0 \cdot L$$

$$c_{0} = d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L + d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L +$$

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

$$c_1 = d'_2 \cdot d'_1 \cdot d'_0 \cdot L + d'_2 \cdot d'_1 \cdot d_0 \cdot L + d'_2 \cdot d_1 \cdot d'_0 \cdot L + d'_2 \cdot d_1 \cdot d_0 \cdot L + d'_2 \cdot d_1 \cdot d_0 \cdot L$$

$$c_{0} = d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L' + d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L +$$

From combinational logic to circuits

$$c_{0} = d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L' + d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L +$$

 $c_1 = d'_2 \cdot d'_1 \cdot d'_0 \cdot L +$

 $d_2^{\tilde{\prime}} \cdot d_1^{\tilde{\prime}} \cdot d_0 \cdot L +$

 $d_2^{\overline{\prime}} \cdot d_1 \cdot d_0^{\prime} \cdot L +$

 $d_2^{\overline{\prime}} \cdot d_1 \cdot d_0^{\widetilde{}} \cdot L$

Here is
$$c_1$$
 as a circuit \ldots



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.

$$c_{0} = d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L' + c_{1} = d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L + d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d'$$

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 $p \land q \rightarrow r \equiv T$

$$c_{0} = d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L' + c_{1} = d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d_{0} \cdot L + d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d_{1} \cdot d'_{0} \cdot L' + d'_{2} \cdot d'_{1} \cdot d'_{0} \cdot L + d'_{2} \cdot d'$$

Day	$d_2 d_1 d_0$	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	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 \lor, \land, \neg .

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

Closure Distributivity Null $a + (b \cdot c) = (a + b) \cdot (a + c)$ $a + b \in B$ a + 1 = 1 $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ $a \cdot b \in B$ $a \cdot 0 = 0$ Commutativity Identity Idempotency a + 0 = aa + b = b + aa + a = a $a \cdot b = b \cdot a$ $a \cdot 1 = a$ $a \cdot a = a$ Associativity Complementarity Involution a + a' = 1a + (b + c) = (a + b) + c(a')' = a $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ $a \cdot a' = 0$

We can use the basics to prove some useful theorems

Uniting

$$\begin{aligned} a \cdot b + a \cdot b' &= a \\ (a + b) \cdot (a + b') &= a \end{aligned}$$

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

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

Uniting	$X \cdot Y + X \cdot Y' =$
8	=
	= X
Absorption	$X + X \cdot Y =$
	=
	=
	=
	= X

Closure	Distributivity	Null
$a+b\in B$	$a + (b \cdot c) = (a + b) \cdot (a + c)$	a + 1 = 1
$a \cdot b \in B$	$a \cdot (b+c) = (a \cdot b) + (a \cdot c)$	$a \cdot 0 = 0$
Commutativity	Identity	Idempotency
a + b = b + a	a + 0 = a	
$a \cdot b = b \cdot a$	$a \cdot 1 = a$	a + a - a
Associativity	Complementarity	$a \cdot a = a$
a + (b + c) = (a + b) + c	a + a' = 1	Involution
$a \cdot (b \cdot c) = (a \cdot b) \cdot c$	$a \cdot a' = 0$	$(a^{\prime})^{\prime} = a$

Uniting	$\begin{array}{l} X \cdot Y + X \cdot Y' \ = \ X \cdot (Y + Y') \\ = \end{array}$	Distributivity
	= X	
Absorption	$X + X \cdot Y =$	
	=	
	=	
	= X	
Closure $a + b \in B$ $a \cdot b \in B$	Distributivity $a + (b \cdot c) = (a + b) \cdot (a + c)$ $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$	Null $a+1=1$

Commutativity

Associativity

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

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

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

a + (b - a) - (a + b) - (a + a)	NULL
$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \perp 1 - 1$
$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$	u + 1 = 1
	$a \cdot 0 = 0$

Identity

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

Complementarity

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

Idempotency

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

Involution

(a')' = a

Uniting	$X \cdot Y + X \cdot Y' = X \cdot (Y + Y')$ $= X \cdot 1$ $= X$	Distributivity Complementarity
Absorption	$\begin{array}{rcl} X + X \cdot Y &= \\ &= \\ &= \\ &= \\ &= \\ &= \\ &= X \end{array}$	
Closure $a + b \in B$ $a \cdot b \in B$ Commutativity a + b = b + a $a \cdot b = b \cdot a$	Distributivity $a + (b \cdot c) = (a + b) \cdot (a + c)$ $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ Identity a + 0 = a $a \cdot 1 = a$	Null a + 1 = 1 $a \cdot 0 = 0$ Idempotency a + a = a

$$a \cdot a = a$$

Associativity

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

Complementarity

 $\begin{aligned} a + a' &= 1\\ a \cdot a' &= 0 \end{aligned}$

Involution

(a')' = a

Uniting	$X \cdot Y + X \cdot Y' = X \cdot (Y + Y')$ $= X \cdot 1$ $= X$	Distributivity Complementarity Identity
Absorption	$\begin{array}{rcl} X + X \cdot Y &= \\ &= \\ &= \\ &= \\ &= \\ &= \\ &= \\ &=$	
Closure $a + b \in B$ $a \cdot b \in B$	Distributivity $a + (b \cdot c) = (a + b) \cdot (a + c)$ $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$	Null $a + 1 = 1$

Commutativity

Associativity

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

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

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

	Null
$a + (b \cdot c) = (a + b) \cdot (a + c)$	a + 1 - 1
$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$	a + 1 - 1
a (b + c) = (a - b) + (a - c)	$a \cdot 0 = 0$

Identity

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

Complementarity

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

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

Involution (a')' = a

Uniting	$X \cdot Y + X \cdot Y'$	$= X \cdot (Y + Y')$ $= X \cdot 1$ $= X$	Distributivity Complementarity Identity
Absorption	$X + X \cdot Y$	$= X \cdot 1 + X \cdot Y$ $=$	Identity
		=	
		=	
		= X	

Closure	Distributivity	NUI
$a + b \in B$	$a + (b \cdot c) = (a + b) \cdot (a + c)$	a + 1 = 1
$a \cdot b \in B$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$	$a + 1 = 1$ $a \cdot 0 = 0$
Commutativity	Identity	Idomnotonov
a + b = b + a	a + 0 = a	Idempotency
$a \cdot b = b \cdot a$	$a \cdot 1 = a$	a + a = a
Associativity	Complementarity	$a \cdot a = a$
		Involution
a + (b + c) = (a + b) + c	a + a' = 1	(a')' - a
$a \cdot (b \cdot c) = (a \cdot b) \cdot c$	$a \cdot a' = 0$	(u) - u

$X \cdot Y + X \cdot Y'$	$= X \cdot (Y + Y')$ = X \cdot 1 = X	Distributivity Complementarity Identity
$X + X \cdot Y$	$= X \cdot 1 + X \cdot Y$ $= X \cdot (1 + Y)$ $=$	ldentity Distributivity
	= = X	
	$X \cdot Y + X \cdot Y'$ $X + X \cdot Y$	$X \cdot Y + X \cdot Y' = X \cdot (Y + Y')$ = X \cdot 1 = X $X + X \cdot Y = X \cdot 1 + X \cdot Y$ = X \cdot (1 + Y) = = X

Closure	Distributivity	NIII
$a + b \in B$	$a + (b \cdot c) = (a + b) \cdot (a + c)$	a + 1 = 1
$a \cdot b \in B$	$a \cdot (b+c) = (a \cdot b) + (a \cdot c)$	$a \cdot 0 = 0$
Commutativity	Identity	Idempotency
a + b = b + a	a + 0 = a	idempotency
$a \cdot b = b \cdot a$	$a \cdot 1 = a$	a + a = a
Accoriativity	Complementarity	$a \cdot a = a$
ASSOCIATIVITY	Complementanty	Involution
a + (b + c) = (a + b) + c	a + a' = 1	(a')' - a
$a \cdot (b \cdot c) = (a \cdot b) \cdot c$	$a \cdot a' = 0$	(u) = u

Uniting	$X \cdot Y + X \cdot Y'$	$= X \cdot (Y + Y')$ = X \cdot 1 = X	Distributivity Complementarity Identity
Absorption	$X + X \cdot Y$	$= X \cdot 1 + X \cdot Y$ = $X \cdot (1 + Y)$ = $X \cdot (Y + 1)$ = X	ldentity Distributivity Commutativity
Closure	Distributivity		

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

a

$a + (b \cdot c) = (a + b) \cdot (a + c)$	NULL
$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$	a + 1

Identity

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

Complementarity

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

= 1 $a \cdot 0 = 0$

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

Involution

(a')' = a

Uniting	$X \cdot Y + X \cdot Y' = X \cdot (Y + Y')$ $= X \cdot 1$ $= X$	Distributivity Complementarity Identity
Absorption	$X + X \cdot Y = X \cdot 1 + X \cdot Y$ $= X \cdot (1 + Y)$ $= X \cdot (Y + 1)$ $= X \cdot 1$ $= X$	Identity Distributivity Commutativity Null
Closure $a + b \in B$ $a \cdot b \in B$	Distributivity $a + (b \cdot c) = (a + b) \cdot (a + c)$ $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$	Null a + 1 = 1 a + 0 = 0
Commutativity a + b = b + a $a \cdot b = b \cdot a$	Identity a + 0 = a $a \cdot 1 = a$	Idempotency a + a = a $a \cdot a = a$
Associativity a + (b + c) = (a + b) + c $a \cdot (b \cdot c) = (a \cdot b) \cdot c$	Complementarity a + a' = 1 $a \cdot a' = 0$	Involution (a')' = a

Uniting	$X \cdot Y + X \cdot Y' = X \cdot (Y + X)$ $= X \cdot 1$ $= X$	Y') Distributivity Complementarity Identity
Absorption	$X + X \cdot Y = X \cdot 1 + X$ $= X \cdot (1 + Y)$ $= X \cdot (Y + 1)$ $= X \cdot 1$ $= X$	 Y Identity Distributivity Commutativity Null Identity
Closure $a + b \in B$ $a \cdot b \in B$ Commutativity a + b = b + a $a \cdot b = b \cdot a$	Distributivity $a + (b \cdot c) = (a + b) \cdot (a + c)$ $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ Identity a + 0 = a $a \cdot 1 = a$	Null a + 1 = 1 $a \cdot 0 = 0$ Idempotency a + a = a $a \cdot a = a$
Associativity a + (b + c) = (a + b) + c $a \cdot (b \cdot c) = (a \cdot b) \cdot c$	Complementarity a + a' = 1 $a \cdot a' = 0$	$a \cdot a = a$ 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

$$c_1 = d'_2 \cdot d'_1 \cdot d'_0 \cdot L + d'_2 \cdot d'_1 \cdot d_0 \cdot L + d'_2 \cdot d_1 \cdot d'_0 \cdot L + d'_2 \cdot d_1 \cdot d_0 \cdot L$$
(from classesLeft)
= ...
= $d'_2 \cdot L$ (HW2)
(HW2)

Closure $a + b \in B$ $a \cdot b \in B$	Distributivity $a + (b \cdot c) = (a + b) \cdot (a + c)$ $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$	Null $a + 1 = 1$ $a \cdot 0 = 0$
Commutativity a + b = b + a $a \cdot b = b \cdot a$	Identity a + 0 = a $a \cdot 1 = a$	Idempotency a + a = a $a \cdot a = a$
Associativity a + (b + c) = (a + b) + c $a \cdot (b \cdot c) = (a \cdot b) \cdot c$	Complementarity a + a' = 1 $a \cdot a' = 0$	Involution $(a')' = a$

Example: simplifying (circuits) using Boolean algebra

$$c_1 = d'_2 \cdot d'_1 \cdot d'_0 \cdot L + d'_2 \cdot d'_1 \cdot d_0 \cdot L + d'_2 \cdot d_1 \cdot d'_0 \cdot L + d'_2 \cdot d_1 \cdot d_0 \cdot L$$
(from classesLeft)
= ...
= $d'_2 \cdot L$ (HW2)
(HW2)

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.