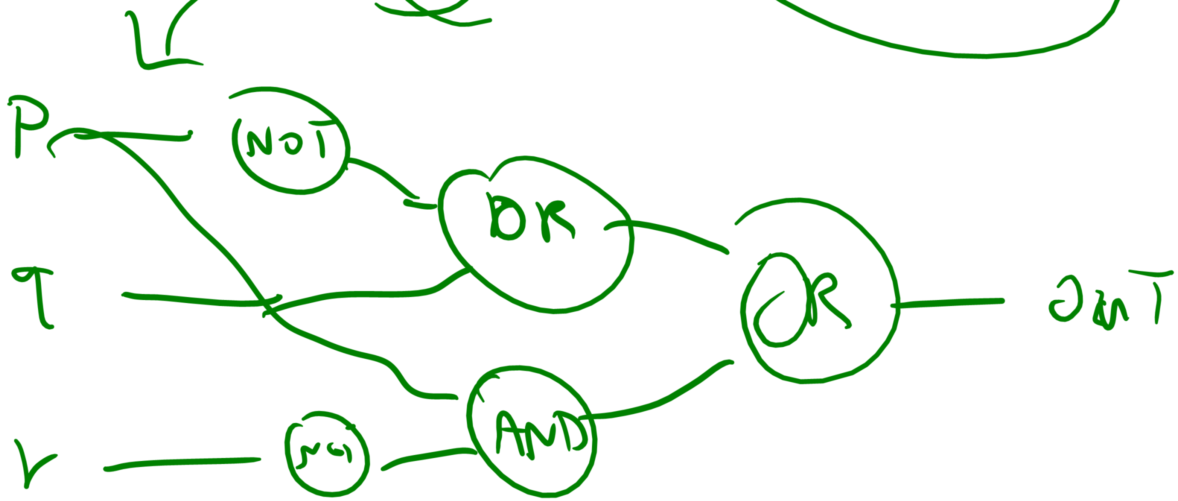
CSE 311

Foundations of Computing I

Pre-Lecture Problem

Do it! Do it now! What are you waiting for? ☺

Construct $(\neg p \vee q) \vee (\neg r \wedge p)$ as a circuit!

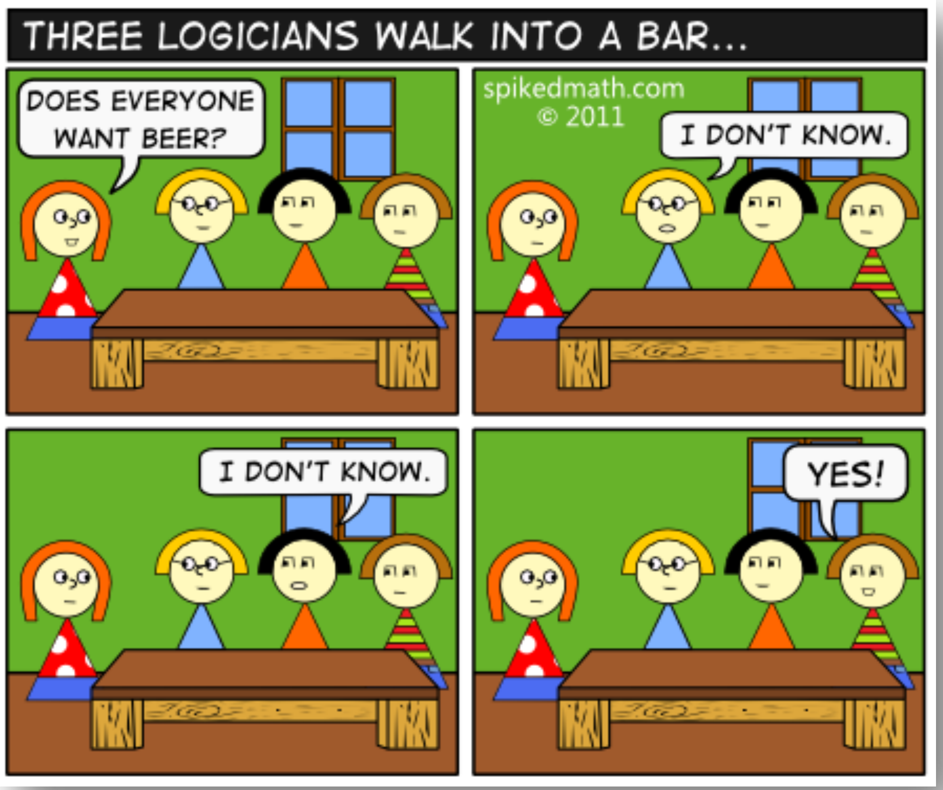


Administrivia

- Extra Credit: Included post-grades-calculation
- Tokens: Redos for **WRITTEN** questions
- Lying TAs

CSE 311: Foundations of Computing

Lecture 3: More Equivalence & Boolean Algebra



Some Familiar Properties of Arithmetic

What are there logical versions of these rules?

- $x + y = y + x$ $\quad x \cdot y = y \cdot x$ **(Commutativity)**

$x \wedge y \equiv y \wedge x$ ✓

$x \vee y \equiv y \vee x$

- $x \cdot (y + z) = x \cdot y + x \cdot z$ **(Distributivity)**

$x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$

- $(x + y) + z = x + (y + z)$ **(Associativity)**

Some Familiar Properties of Arithmetic

What are there logical versions of these rules?

- $x + y = y + x$ **(Commutativity)**
  - $p \vee q \equiv q \vee p$
  - $p \wedge q \equiv q \wedge p$

$x \to (y \wedge z) \stackrel{?}{\equiv} (x \to y) \wedge (x \to z)$

↳ proof by TT (yay.)

- $x \cdot (y + z) = x \cdot y + x \cdot z$ **(Distributivity)**
  - $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
  - $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
- $(x + y) + z = x + (y + z)$ **(Associativity)**
  - $(p \vee q) \vee r \equiv p \vee (q \vee r)$
  - $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$

Properties of Logical Connectives

We will always give you this list!

DO NOT mem.! please !!

- **Identity**
  - $p \wedge \mathrm{T} \equiv p$
  - $p \vee \mathrm{F} \equiv p$
- **Domination**
  - $p \vee \mathrm{T} \equiv \mathrm{T}$
  - $p \wedge \mathrm{F} \equiv \mathrm{F}$
- **Idempotent**
  - $p \vee p \equiv p$
  - $p \wedge p \equiv p$
- **Commutative**
  - $p \vee q \equiv q \vee p$
  - $p \wedge q \equiv q \wedge p$
- **Associative**
  - $(p \vee q) \vee r \equiv p \vee (q \vee r)$
  - $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
- **Distributive**
  - $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
  - $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
- **Absorption**
  - $p \vee (p \wedge q) \equiv p$
  - $p \wedge (p \vee q) \equiv p$
- **Negation**
  - $p \vee \neg p \equiv \mathrm{T}$
  - $p \wedge \neg p \equiv \mathrm{F}$

Understanding Connectives

- Reflect basic rules of reasoning and logic
- Allow manipulation of logical formulas
  - Simplification
  - Testing for equivalence
- Applications
  - Query optimization
  - Search optimization and caching
  - Artificial Intelligence
  - Program verification

Computing Equivalence

Given two propositions, can we write an algorithm to determine if they are equivalent?

What is the runtime of our algorithm?

n vars?

$\hookrightarrow O(2^n)$

Computing Equivalence

Given two propositions, can we write an algorithm to determine if they are equivalent?

Yes! Generate the truth tables for both propositions and check if they are the same for every entry.

What is the runtime of our algorithm?

Every atomic proposition has two possibilities (T, F). If there are n atomic propositions, there are 2^n rows in the truth table.

Logical Proofs

To show A is equivalent to B:

Apply a series of logical equivalences to sub-expressions to convert A to B

Example:

Let A be "$p \lor (p \lor p)$", and B be "p".

Our general proof looks like:

$$p \lor \overset{p}{(p \cancel{\lor} p)} \equiv (\; p \lor p \;) \qquad \text{Idempot.}$$

$$\equiv p \qquad \text{Idempot.}$$

Logical Proofs

To show A is equivalent to B:

Apply a series of logical equivalences to sub-expressions to convert A to B

Example:

Let A be "$p \vee (p \vee p)$", and B be "p".

Our general proof looks like:

$$
\begin{aligned}
p \vee (p \vee p) &\equiv (\quad p \vee p \quad) && \text{By Idempotency} \\
&\equiv p && \text{By Idempotency}
\end{aligned}
$$

Logical Proofs

To show A is a Tautology:

Apply a series of logical equivalences to sub-expressions to convert P to T.

Example:

Let A be "$\neg p \lor (p \lor p)$".

Our general proof looks like:

Comm.

Idem $\quad p \lor \neg p$

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

$$\equiv \mathbf{T}$$

negation

Logical Proofs

To show A is a Tautology:

Apply a series of logical equivalences to sub-expressions to convert P to T.

Example:

Let A be "$\neg p \vee (p \vee p)$".

Our general proof looks like:

$$
\begin{aligned}
\neg p \vee (p \vee p) &\equiv (\quad \neg p \vee p \quad) && \text{By Idempotency} \\
&\equiv \mathbf{T} && \text{By Negation}
\end{aligned}
$$

Prove this is a Tautology: Option 1

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

Make a Truth Table and show:

$$(p \wedge q) \rightarrow (p \vee q) \equiv \mathbf{T}$$

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

Prove this is a Tautology: Option 2

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

Use a series of equivalences like so:

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

Prove this is a Tautology: Option 2

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

Use a series of equivalences like so:

$(p \wedge q) \rightarrow (p \vee q)$	$\equiv \neg(p \wedge q) \vee (p \vee q)$	By Law of Implication
	$\equiv (\neg p \vee \neg q) \vee (p \vee q)$	By DeMorgan's Laws
	$\equiv \neg p \vee (\neg q \vee (p \vee q))$	By Associativity
	$\equiv \neg p \vee (\neg q \vee (q \vee p))$	By Commutativity
	$\equiv \neg p \vee ((\neg q \vee q) \vee p)$	By Associativity
	$\equiv \neg p \vee ((q \vee \neg q) \vee p)$	By Commutativity
	$\equiv \neg p \vee (\mathbf{T} \vee p)$	By Negation
	$\equiv \neg p \vee (p \vee \mathbf{T})$	By Commutativity
	$\equiv \neg p \vee \mathbf{T}$	By Domination
	$\equiv \mathbf{T}$	By Domination

Prove these propositions are equivalent

Prove: $p \wedge (p \rightarrow q) \equiv p \wedge q$

$$p \wedge (\overset{p}{p} \rightarrow \overset{q}{q}) \equiv \overbrace{p}^{p} \wedge (\overbrace{\neg p}^{q} \vee \overbrace{q}^{r}) \qquad \text{imp.}$$

$$\equiv (p \wedge \neg p) \vee (p \wedge q) \qquad \text{dist.}$$

$$\equiv F \vee (p \wedge q) \qquad \text{neg.}$$

$$\equiv (p \wedge q) \vee F \qquad \text{comm}$$

$$\equiv p \wedge q \qquad \text{id.}$$

Prove these propositions are equivalent

Prove: $p \wedge (p \rightarrow q) \equiv p \wedge q$

$p \wedge (p \rightarrow q) \equiv p \wedge (\neg p \vee q)$	By Law of Implication
$\equiv (p \wedge \neg p) \vee (p \wedge q)$	By Distributivity
$\equiv \mathbf{F} \vee (p \wedge q)$	By Negation
$\equiv (p \wedge q) \vee \mathbf{F}$	By Commutativity
$\equiv p \wedge q$	By Identity

Prove these are not equivalent

$(p \rightarrow q) \rightarrow r$ $\qquad\qquad$ $p \rightarrow (q \rightarrow r)$

Consider: p is F, q is F, and r is F…

$(F \rightarrow F) \rightarrow F \equiv T \rightarrow F \equiv F$

$F \rightarrow (F \rightarrow F) \equiv F \rightarrow T \equiv T$

Not equal.

Prove these are not equivalent

$(p \rightarrow q) \rightarrow r$

$p \rightarrow (q \rightarrow r)$

Consider: p is F, q is F, and r is F…

$$(\mathbf{F} \rightarrow \mathbf{F}) \rightarrow \mathbf{F} \equiv \mathbf{T} \rightarrow \mathbf{F}$$
$$\equiv \mathbf{F}$$

$$\mathbf{F} \rightarrow (\mathbf{F} \rightarrow \mathbf{F}) \equiv \mathbf{F} \rightarrow \mathbf{F}$$
$$\equiv \mathbf{T}$$

Boolean Logic

$(y+x)'$

y'

Combinational Logic

– output = F(input)

Sequential Logic

– $output_t = F(output_{t-1}, input_t)$

- output dependent on history
- concept of a time step (clock, t)

Boolean Algebra consists of…

"$x + y'$"

– a set of elements B = {0, 1}

– binary operations { + , • } (OR, AND) $x \vee \neg y$

– and a unary operation { ’ } (NOT)

A Combinational Logic Example

Sessions of Class:

We would like to compute the number of lectures or quiz sections remaining *at the start* of a given day of the week.

- **Inputs:** Day of the Week, Lecture/Section flag
- **Output:** Number of sessions left

Examples: Input: (Wednesday, Lecture) Output: 2

Input: (Monday, Section) Output: 1

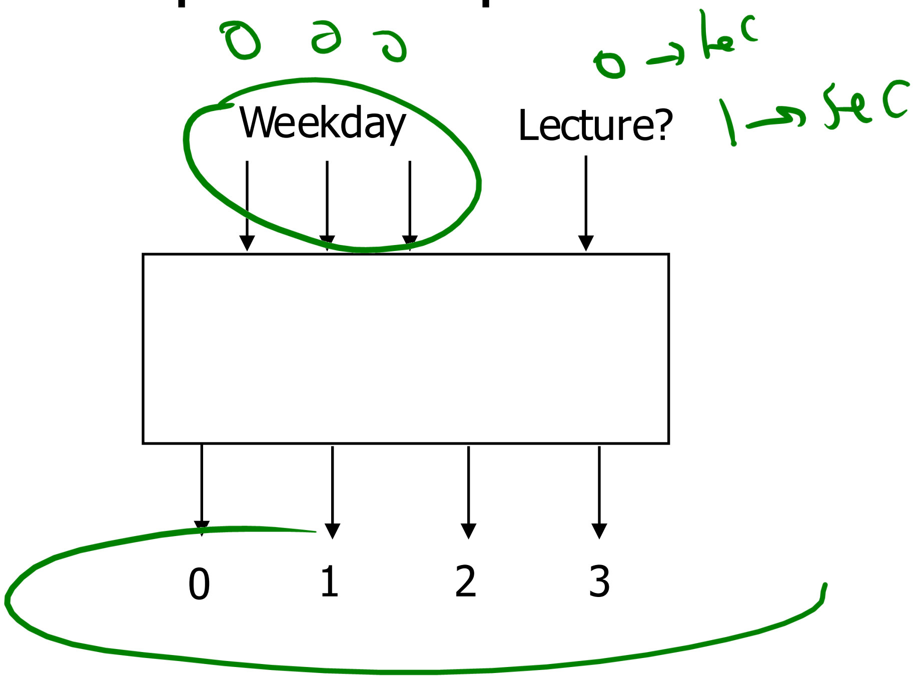
Implementation in Software

```
public int classesLeftInMorning(weekday, lecture_flag) {
    switch (weekday) {
        case SUNDAY:
        case MONDAY:
            return lecture_flag ? 3 : 1;
        case TUESDAY:
        case WEDNESDAY:
            return lecture_flag ? 2 : 1;
        case THURSDAY:
            return lecture_flag ? 1 : 1;
        case FRIDAY:
            return lecture_flag ? 1 : 0;
        case SATURDAY:
            return lecture_flag ? 0 : 0;
    }
}
```

Implementation with Combinational Logic

Encoding:

- How many bits for each input/output?
- Binary number for weekday
- One bit for each possible output



Defining Our Inputs!

Weekday Input:

- Binary number for weekday
- Sunday = 0, Monday = 1, …
- We care about these in binary:

Weekday	Number	Binary
Sunday	0	$(000)_2$
Monday	1	$(001)_2$
Tuesday	2	$(010)_2$
Wednesday	3	$(011)_2$
Thursday	4	$(100)_2$
Friday	5	$(101)_2$
Saturday	6	$(110)_2$

111

Combinational Logic

- Switches
- Basic logic and truth tables
- Logic functions
- Boolean algebra
- Proofs by re-writing and by truth table