CSE 311

Foundations of Computing I

Fall 2014

Administrivia

Course Web: http://www.cs.washington.edu/311

Office Hours: 10 hours available; check web

Homework #1: Posted. Turn in (stapled) at the **start** of class on Wednesday (Oct 1)

Extra Credit: Not required to get a 4.0.
Counts separately.
In total, may raise grade by ~0.1

Last Time: Logical Connectives

p	$\neg p$
T	F
F	T

NOT

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

AND

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

OR

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

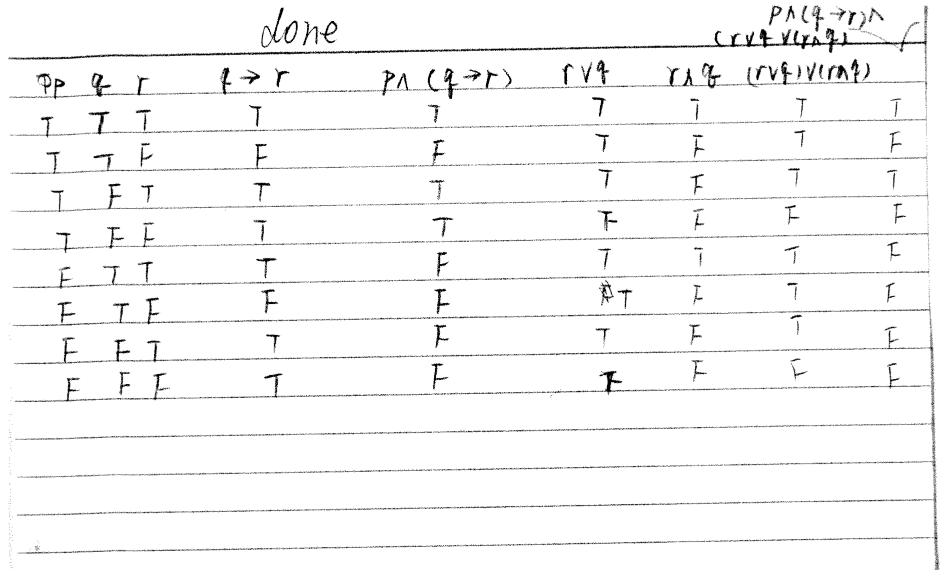
XOR

Index Card Questions!

Well formed? hard to grasp.

Does truth value have to be a single value or can it alternate?

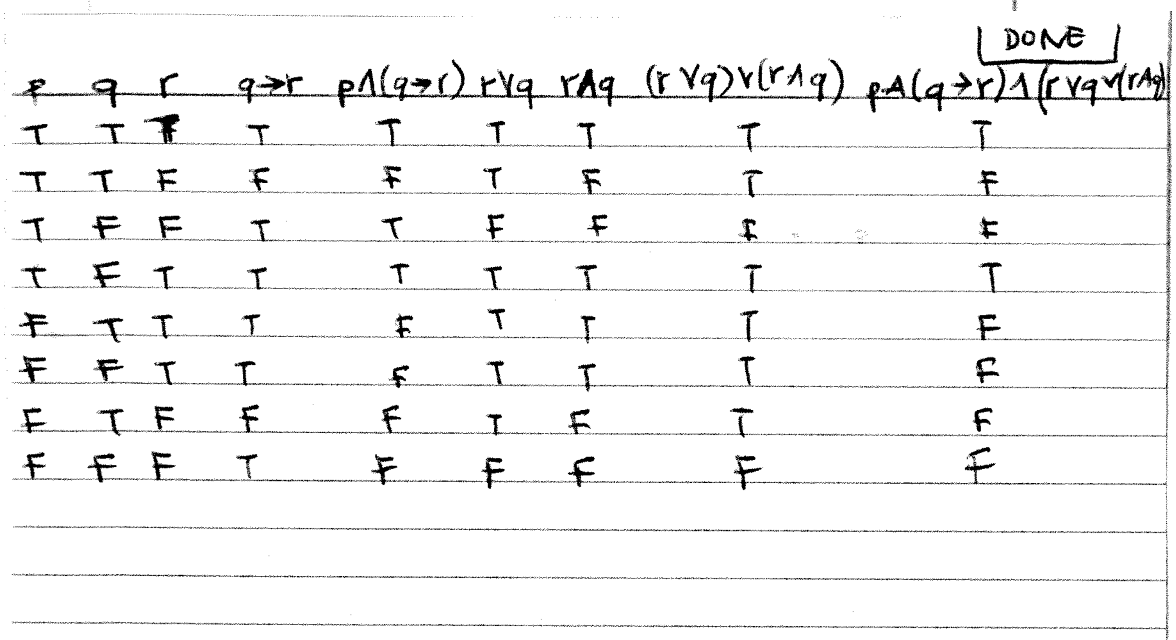
Index Card "Gotchas"



done

p∧(q→r)∧ (r∨q)∨(r∧q)

p	q	r	q→r	p∧(q→r)	r∨q	r∧q	(r∨q)∨(r∧q)	
T	T	T	T	T	T	T	T	T
T	T	F	F	F	T	F	T	F
T	F	T	T	T	T	F	T	T
T	F	F	T	T	F	F	F	F
F	T	T	T	F	T	T	T	F
F	T	F	F	F	T	F	T	F
F	F	T	T	F	T	F	T	F
F	F	F	T	F	F	F	F	F



DONE

p	q	r	q→r	p∧(q→r)	r∨q	r∧q	(r∨q)∨(r∧q)	p∧(q→r)∧((r∨q)∨(r∧q))
T	T	T	T	T	T	T	T	T
T	T	F	F	F	T	F	T	F
T	F	F	T	T	F	F	F	F
T	F	T	T	T	T	T	T	T
F	T	T	T	F	T	T	T	F
F	F	T	T	F	T	T	T	F
F	T	F	F	F	T	F	T	F
F	F	F	T	F	F	F	F	F

Index Card "Gotchas"

p	q	r	$q \to r$	$p \wedge (q \to r)$	$r \vee q$	$r \wedge q$	$(r \vee q) \vee (r \wedge q)$	$p \wedge (q \to r) \wedge ((r \vee q) \vee (r \wedge q))$
0	0	0	1	0	0	0	0	0
0	0	1	1	0	1	0	1	0
0	1	0	0	0	1	0	1	0
0	1	1	1	0	1	1	1	0
1	0	0	1	1	0	0	0	0
1	0	1	1	1	1	0	1	1
1	1	0	0	0	1	0	1	0
1	1	1	1	1	1	1	1	1

Last Time: $p \rightarrow q$

- "If p, then q" is a **promise**:
  - Whenever p is true, then q is true
  - Ask "has the promise been broken"

p	q	$p \rightarrow q$
F	F	T
F	T	T
T	**F**	**F**
T	T	T

If it's raining, then I have my umbrella

Suppose it's not raining…

First Question: It's not raining, and I don't bring my umbrella. Have I broken the promise?

Second Question: It's not raining, and I bring my umbrella. Have I broken the promise?

In both cases, the pre-requisite to my promise isn't met. So, I haven't in either case. In fact, the only case in which I've lied to you is when it's raining, but I don't have my umbrella.

Last Time: Related Implications

- **Implication:** $p \rightarrow q$
- **Converse:** $q \rightarrow p$
- **Contrapositive:** $\neg q \rightarrow \neg p$
- **Inverse:** $\neg p \rightarrow \neg q$

How do these relate to each other?

Example:

p: x is divisible by 2

q: x is divisible by 4

$p \rightarrow q$	False
$q \rightarrow p$	True
$\neg q \rightarrow \neg p$	False
$\neg p \rightarrow \neg q$	True

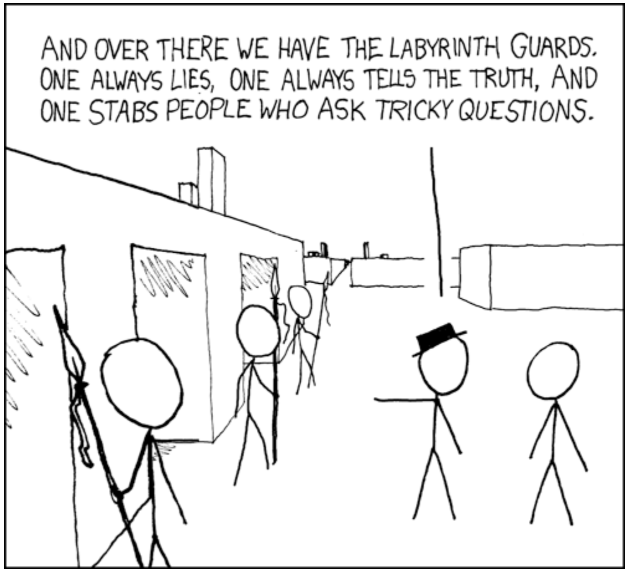
Last Time: $p \leftrightarrow q$

- *p* iff *q*
- *p* is equivalent to *q*
- *p* implies *q* and *q* implies *p*

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

CSE 311: Foundations of Computing

Fall 2014
Lecture 2: Digital Circuits & More Logic



Digital Circuits

Computing With Logic

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

Gates

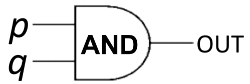
- Take inputs and produce outputs (functions)
- Several kinds of gates
- Correspond to propositional connectives (most of them)

And Gate

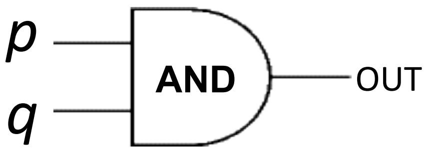
AND Connective vs. AND Gate

$p \wedge q$

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F



p	q	OUT
1	1	1
1	0	0
0	1	0
0	0	0



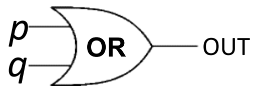
"block looks like D of AND"

Or Gate

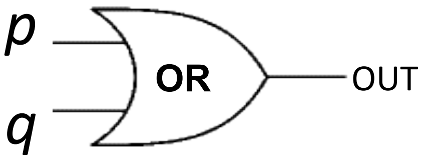
OR Connective vs. OR Gate

p* ∨ *q

p	*q*	*p* ∨ *q*
T	T	T
T	F	T
F	T	T
F	F	F



p	*q*	OUT
1	1	1
1	0	1
0	1	1
0	0	0

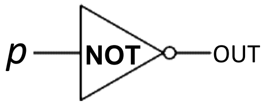


"arrowhead block looks like V"

Not Gates

NOT Connective vs. **NOT Gate**

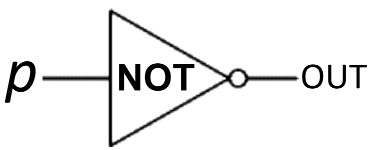
$\neg p$



Also called *inverter*

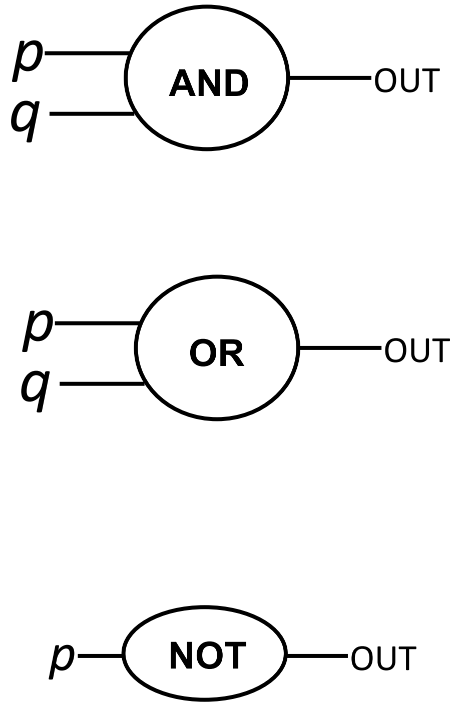
p	$\neg p$
T	F
F	T

p	OUT
1	0
0	1



Blobs are Okay!

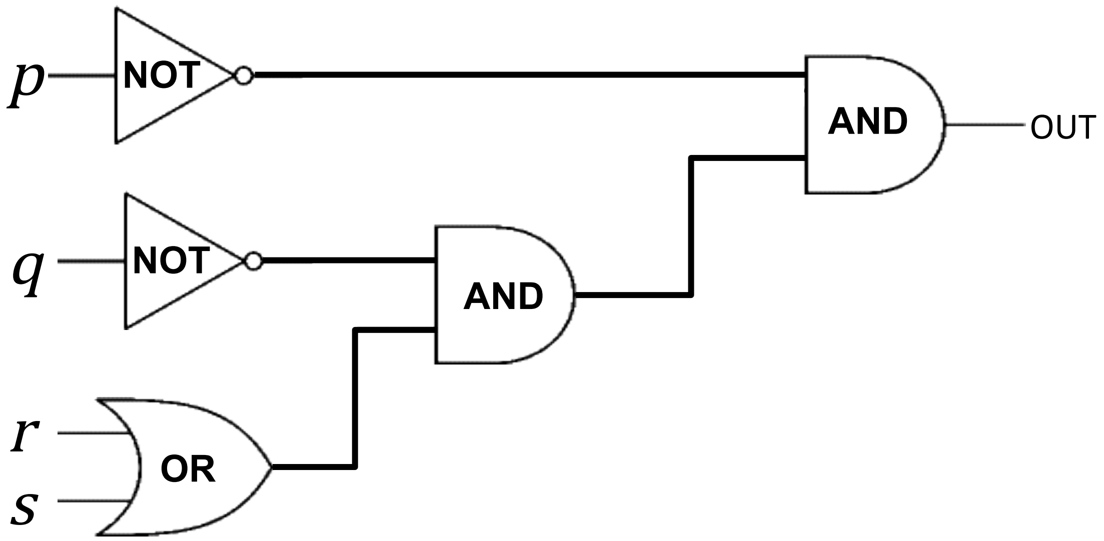
You may write gates using blobs instead of shapes!



Let's Try Something New…

GOTO: http://tinyurl.com/ynlecture

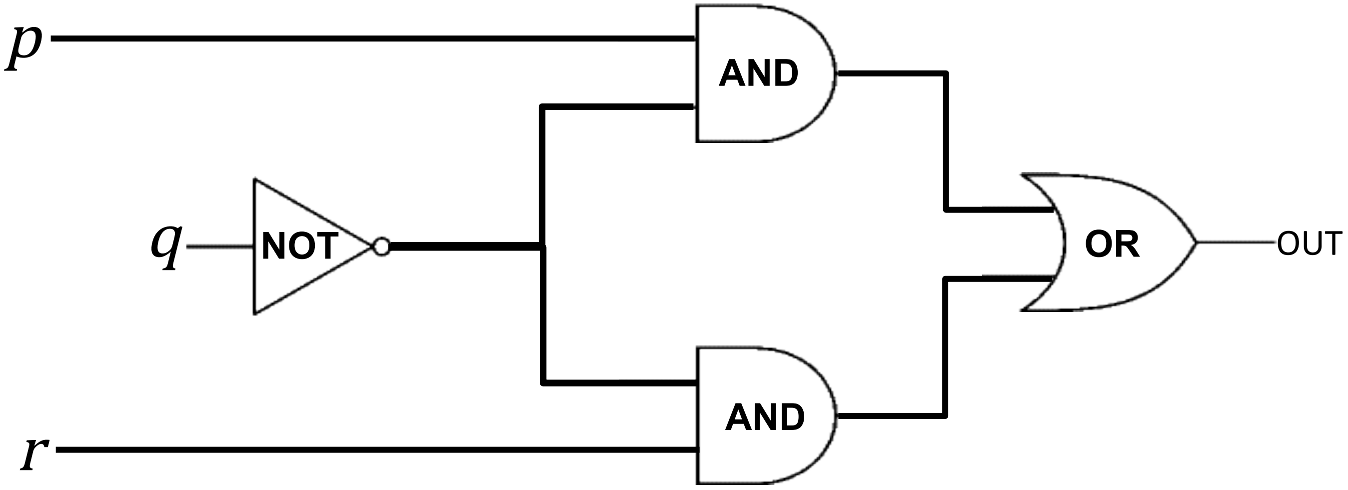
Combinational Logic Circuits



Values get sent along wires connecting gates

$$\neg p \land (\neg q \land (r \lor s))$$

Combinational Logic Circuits



Wires can send one value to multiple gates!

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

Logical Equivalence

Terminology: A compound proposition is a…

- *Tautology* if it is always true
- *Contradiction* if it is always false
- *Contingency* if it can be either true or false

$p \vee \neg p$ Tautology!

$p \oplus p$ Contradiction!

$(p \rightarrow q) \wedge p$ Contingency (note: in lecture the and was an or)!

$(p \wedge q) \vee (p \wedge \neg q) \vee (\neg p \wedge q) \vee (\neg p \wedge \neg q)$ Tautology!

Logical Equivalence

A and B are *logically equivalent* if and only if $A \leftrightarrow B$ is a tautology

i.e. A and B have the same truth table

The notation $A \equiv B$ denotes A and B are logically equivalent

Example: $p \equiv \neg\neg p$

p	$\neg p$	$\neg\neg p$	$p \leftrightarrow \neg\neg p$

Logical Equivalence

A and *B* are *logically equivalent* if and only if $A \leftrightarrow B$ is a tautology

i.e. A and *B* have the same truth table

The notation $A \equiv B$ denotes *A* and *B* are logically equivalent

Example: $p \equiv \neg\neg p$

p	$\neg p$	$\neg\neg p$	$p \leftrightarrow \neg\neg p$
T	F	T	T
F	T	F	T

$A \leftrightarrow B$ vs. $A \equiv B$

$A \equiv B$ says that ***two*** propositions A and B *always* ***mean*** *the same thing*

$A \leftrightarrow B$ is a ***single*** proposition that may be true or false depending on the truth values of the variables in A and B

- but $A \equiv B$ and $(A \leftrightarrow B) \equiv \mathbf{T}$ have the same meaning

Note: Why write $A \equiv B$ and not $A{=}B$?

We use $A{=}B$ to say that A and B are precisely the same proposition (same sequence of symbols)

De Morgan's Laws

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

My code compiles or there is a bug.

The negation of this statement is:

It's not the case that my code compiles or there is a bug

My code doesn't compile and there isn't a bug

De Morgan's Laws

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

p	q	$\neg p$	$\neg q$	$\neg p \lor \neg q$	$p \land q$	$\neg (p \land q)$	$\neg (p \land q) \leftrightarrow (\neg p \lor \neg q)$
T	T						
T	F						
F	T						
F	F						

De Morgan's Laws

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

p	q	$\neg p$	$\neg q$	$\neg p \vee \neg q$	$p \wedge q$	$\neg (p \wedge q)$	$\neg (p \wedge q) \leftrightarrow (\neg p \vee \neg q)$
T	T	F	F	F	T	F	T
T	F	F	T	T	F	T	T
F	T	T	F	T	F	T	T
F	F	T	T	T	F	T	T

De Morgan's laws

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

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

```
if (!(front != null && value > front.data))
    front = new ListNode(value, front);
else {
    ListNode current = front;
    while (current.next != null && current.next.data < value))
        current = current.next;
    current.next = new ListNode(value, current.next);
}
```

This code inserts *value* into a sorted linked list.

The first if runs when…front is null or value is smaller than the first item.

The while loop stops when…we've reached the end of the list or the next value is bigger.

Law of Implication

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

p	q	$p \rightarrow q$	$\neg\ p$	$\neg\ p \vee q$	$(p \rightarrow q) \leftrightarrow (\neg\ p \vee q)$
T	T				
T	F				
F	T				
F	F				

Law of Implication

$$(p \rightarrow q) \equiv (\neg\, p \vee q)$$

p	q	$p \rightarrow q$	$\neg\, p$	$\neg\, p \vee q$	$(p \rightarrow q) \leftrightarrow (\neg\, p \vee q)$
T	T	T	F	T	T
T	F	F	F	F	T
F	T	T	T	T	T
F	F	T	T	T	T

Computing Equivalence

Describe an algorithm for computing if two logical expressions/circuits are equivalent.

What is the run time of the algorithm?

Compute the entire truth table for both of them!

There are 2^n entries in the column for *n* variables.

Some Familiar Properties of Arithmetic

- $x + y = y + x$ **(Commutativity)**
  - $p \lor q \equiv q \lor p$
  - $p \land q \equiv q \land p$
- $x \cdot (y + z) = x \cdot y + x \cdot z$ **(Distributivity)**
  - $p \land (q \lor r) \equiv (p \land q) \lor (p \land r)$
  - $p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$
- $(x + y) + z = x + (y + z)$ **(Associativity)**
  - $(p \lor q) \lor r \equiv p \lor (q \lor r)$
  - $(p \land q) \land r \equiv p \land (q \land r)$

Properties of Logical Connectives

We will always give you this list!

- **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**
  - $p \vee \neg p \equiv \mathrm{T}$
  - $p \wedge \neg p \equiv \mathrm{F}$

Some Equivalences Related to Implication

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

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

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

$$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$$

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

Next Time...

x is 0 $\vee$ 1

$\vee$

Difficult-ware