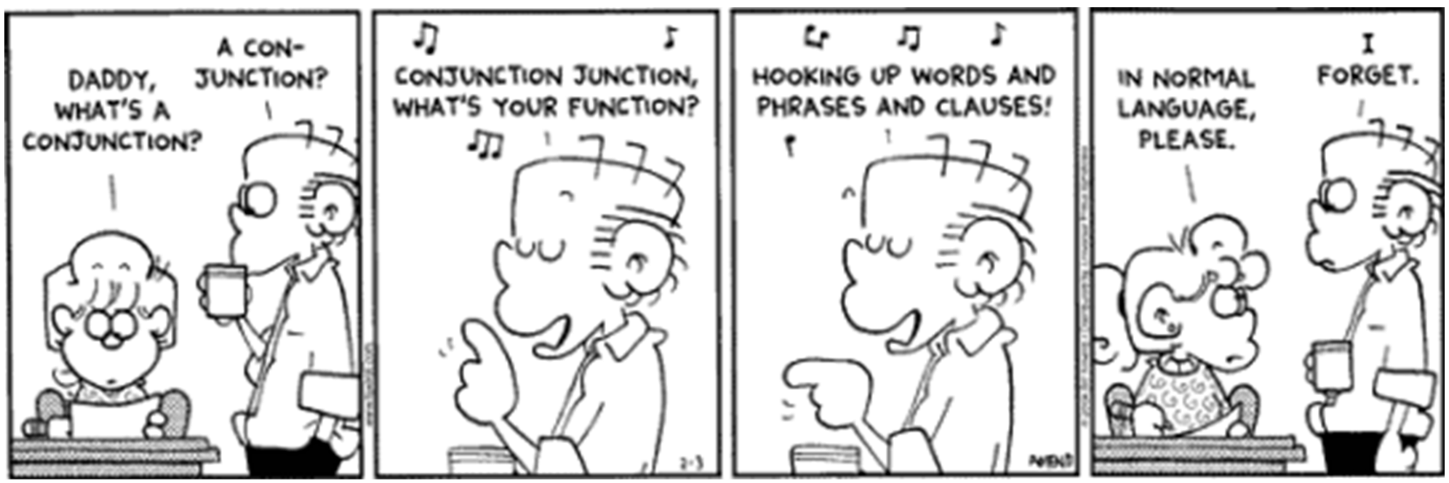
# CSE 311: Foundations of Computing

## Lecture 2: More Logic, Equivalence & Digital Circuits



# Last class: Some Connectives & Truth Tables

## Negation (not)

| $p$ | $\neg p$ |
|---|---|
| T | F |
| F | T |

## Conjunction (and)

| $p$ | $q$ | $p \wedge q$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

## Disjunction (or)

| $p$ | $q$ | $p \vee q$ |
|---|---|---|
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

## Exclusive Or

| $p$ | $q$ | $p \oplus q$ |
|---|---|---|
| T | T | F |
| T | F | T |
| F | T | T |
| F | F | F |

# Last class: Implication

*"If it's raining, then I have my umbrella"*

| $p$ | $q$ | $p \rightarrow q$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | T |
| F | F | T |

## $p \rightarrow q$

*(1) "I have collected all 151 Pokémon if I am a Pokémon master"*

*(2) "I have collected all 151 Pokémon only if I am a Pokémon master"*

These sentences are implications in opposite directions:

## $p \rightarrow q$

*(1) “I have collected all 151 Pokémon if I am a Pokémon master”*

*(2) “I have collected all 151 Pokémon only if I am a Pokémon master”*

These sentences are implications in opposite directions:

(1) “Pokémon masters have all 151 Pokémon”

(2) “People who have 151 Pokémon are Pokémon masters”

So, the implications are:

*(1) If* I am a Pokémon master, *then* I have collected all 151 Pokémon.

*(2) If* I have collected all 151 Pokémon, *then* I am a Pokémon master.

## $p \rightarrow q$

| $p$ | $q$ | $p \rightarrow q$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | T |
| F | F | T |

**Implication:**

- *p* implies *q*
- whenever *p* is true *q* must be true
- if *p* then *q*
- *q* if *p*
- *p* is sufficient for *q*
- *p* only if *q*
- *q is necessary for p*

# Biconditional: $p \leftrightarrow q$

- *p* iff *q*
- *p* is equivalent to *q*
- *p* implies *q* and *q* implies *p*
- *p* is necessary and sufficient for *q*

| $p$ | $q$ | $p \leftrightarrow q$ |
|---|---|---|
| | | |
| | | |
| | | |
| | | |

# Biconditional: $p \leftrightarrow q$

- *p* iff *q*
- *p* is equivalent to *q*
- *p* implies *q* and *q* implies *p*
- *p* is necessary and sufficient for *q*

| $p$ | $q$ | $p \leftrightarrow q$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | T |

# Last class: Using Logical Connectives

Measles:
**"You can get measles"**
Mumps:
**"You can get mumps"**
MMR:
**"You had the MMR vaccine"**

**"You can get measles and mumps if you didn't have the MMR vaccine, but if you had the MMR vaccine then you can't get either."**

↓

((Measles **and** Mumps) **if not** MMR) **and** (**if** MMR **then not** (Measles **or** Mumps))

↓

((Measles ∧ Mumps) **if** ¬MMR) ∧ (**if** MMR **then** ¬(Measles ∨ Mumps))

# Understanding the Vaccine Sentence

**"You can get measles and mumps if you didn't have the MMR vaccine, but if you had the MMR vaccine you can't get either."**

((Measles ∧ Mumps) **if** ¬MMR) ∧ (**if** MMR **then** ¬(Measles ∨ Mumps))

(¬MMR → (Measles ∧ Mumps)) ∧ (MMR → ¬(Measles ∨ Mumps))

# Understanding the Vaccine Sentence

**"You can get measles and mumps if you didn't have the MMR vaccine, but if you had the MMR vaccine you can't get either."**

((Measles ∧ Mumps) **if** ¬MMR) ∧ (**if** MMR **then** ¬(Measles ∨ Mumps))

(¬MMR → (Measles ∧ Mumps)) ∧ (MMR → ¬(Measles ∨ Mumps))

**Define shorthand …**

$p$ : MMR
$q$ : Measles
$r$ : Mumps

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

# Analyzing the Vaccine Sentence with a Truth Table

| $p$ | $q$ | $r$ | $\neg p$ | $q \wedge r$ | $\neg p \longrightarrow (q \wedge r)$ | $q \vee r$ | $\neg(q \vee r)$ | $p \rightarrow \neg(q \vee r)$ | $\big(\neg p \longrightarrow (q \wedge r)\big) \wedge (p \rightarrow \neg(q \vee r))$ |
|---|---|---|---|---|---|---|---|---|---|
| T | T | T | | | | | | | |
| T | T | F | | | | | | | |
| T | F | T | | | | | | | |
| T | F | F | | | | | | | |
| F | T | T | | | | | | | |
| F | T | F | | | | | | | |
| F | F | T | | | | | | | |
| F | F | F | | | | | | | |

# Analyzing the Vaccine Sentence with a Truth Table

| $p$ | $q$ | $r$ | $\neg p$ | $q \wedge r$ | $\neg p \longrightarrow (q \wedge r)$ | $q \vee r$ | $\neg(q \vee r)$ | $p \rightarrow \neg(q \vee r)$ | $\big(\neg p \longrightarrow (q \wedge r)\big) \wedge (p \rightarrow \neg(q \vee r))$ |
|---|---|---|---|---|---|---|---|---|---|
| T | T | T | F | T | T | T | F | F | F |
| T | T | F | F | F | T | T | F | F | F |
| T | F | T | F | F | T | T | F | F | F |
| T | F | F | F | F | T | F | T | T | T |
| F | T | T | T | T | T | T | F | T | T |
| F | T | F | T | F | F | T | F | T | F |
| F | F | T | T | F | F | T | F | T | F |
| F | F | F | T | F | F | F | T | T | F |

# Converse, Contrapositive

**Implication:**

$$p \rightarrow q$$

**Converse:**

$$q \rightarrow p$$

**Contrapositive:**

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

**Inverse:**

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

**Consider**

*p*: *x* is divisible by 2

*q*: *x* is divisible by 4

| | |
|---|---|
| $p \rightarrow q$ | |
| $q \rightarrow p$ | |
| $\neg q \rightarrow \neg p$ | |
| $\neg p \rightarrow \neg q$ | |

# Converse, Contrapositive

**Implication:**

$p \rightarrow q$

**Converse:**

$q \rightarrow p$

**Contrapositive:**

$\neg q \rightarrow \neg p$

**Inverse:**

$\neg p \rightarrow \neg q$

**Consider**

*p*: *x* is divisible by 2

*q*: *x* is divisible by 4

| | |
|---|---|
| $p \rightarrow q$ | |
| $q \rightarrow p$ | |
| $\neg q \rightarrow \neg p$ | |
| $\neg p \rightarrow \neg q$ | |

| | Divisible By 2 | Not Divisible By 2 |
|---|---|---|
| Divisible By 4 | | |
| Not Divisible By 4 | | |

# Converse, Contrapositive

**Implication:**

$$p \rightarrow q$$

**Converse:**

$$q \rightarrow p$$

**Contrapositive:**

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

**Inverse:**

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

**Consider**

***p*: *x* is divisible by 2**

***q*: *x* is divisible by 4**

| | |
|---|---|
| $p \rightarrow q$ | |
| $q \rightarrow p$ | |
| $\neg q \rightarrow \neg p$ | |
| $\neg p \rightarrow \neg q$ | |

| | Divisible By 2 | Not Divisible By 2 |
|---|---|---|
| Divisible By 4 | 4,8,12,... | Impossible |
| Not Divisible By 4 | 2,6,10,... | 1,3,5,... |

# Converse, Contrapositive

**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?**

| $p$ | $q$ | $p \rightarrow q$ | $q \rightarrow p$ | $\neg p$ | $\neg q$ | $\neg p \rightarrow \neg q$ | $\neg q \rightarrow \neg p$ |
|---|---|---|---|---|---|---|---|
| T | T | | | | | | |
| T | F | | | | | | |
| F | T | | | | | | |
| F | F | | | | | | |

# Converse, Contrapositive

**Implication:**

$p \rightarrow q$

**Contrapositive:**

$\neg q \rightarrow \neg p$

**Converse:**

$q \rightarrow p$

**Inverse:**

$\neg p \rightarrow \neg q$

**An implication and it's contrapositive have the same truth value!**

| $p$ | $q$ | $p \rightarrow q$ | $q \rightarrow p$ | $\neg p$ | $\neg q$ | $\neg p \rightarrow \neg q$ | $\neg q \rightarrow \neg p$ |
|---|---|---|---|---|---|---|---|
| T | T | T | T | F | F | T | T |
| T | F | F | T | F | T | T | F |
| F | T | T | F | T | F | F | T |
| F | F | T | T | T | T | T | T |

# Tautologies!

**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

$p \lor \neg p$

$p \oplus p$

$(p \to q) \land p$

# Tautologies!

**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

$p \lor \neg p$

This is a tautology. It's called the "law of the excluded middle.
If p is true, then $p \lor \neg p$ is true. If p is false, then $p \lor \neg p$ is true.

$p \oplus p$

This is a contradiction. It's always false no matter what truth value p takes on.

$(p \to q) \land p$

This is a contingency. When p=T, q=T, $(T \to T)\land T$ is true.
When p=T, q=F, $(T \to F)\land T$ is false.

# Logical Equivalence

A = B means A and B are identical "strings":

- $p \wedge q = p \wedge q$
- $p \wedge q \neq q \wedge p$

# Logical Equivalence

A = B means A and B are identical "strings":

- $p \wedge q = p \wedge q$

  These are equal, because they are character-for-character identical.

- $p \wedge q \neq q \wedge p$

  These are NOT equal, because they are different sequences of characters. They "mean" the same thing though.

A ≡ B means A and B have identical truth values:

- $p \wedge q \equiv p \wedge q$
- $p \wedge q \equiv q \wedge p$
- $p \wedge q \not\equiv q \vee p$

# Logical Equivalence

A = B means A and B are identical "strings":

– $p \wedge q = p \wedge q$

These are equal, because they are character-for-character identical.

– $p \wedge q \neq q \wedge p$

These are NOT equal, because they are different sequences of characters. They "mean" the same thing though.

A ≡ B means A and B have identical truth values:

– $p \wedge q \equiv p \wedge q$

Two formulas that are equal also are equivalent.

– $p \wedge q \equiv q \wedge p$

These two formulas have the same truth table!

– $p \wedge q \not\equiv q \vee p$

When $p$=T and $q$=F, $p \wedge q$ is false, but $p \vee q$ is true!

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

$A \equiv B$ is an assertion over all possible truth values that A and B always have the same truth values.

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

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

# De Morgan's Laws

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

Negate the statement:

"My code compiles or there is a bug."

To negate the statement, ask "when is the original statement false".

# De Morgan's Laws

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

**Negate the statement:**

**"My code compiles or there is a bug."**

**To negate the statement, ask "when is the original statement false".**

It's false when not(my code compiles) AND not(there is a bug).

Translating back into English, we get:

My code doesn't compile and there is not a bug.

# 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 | | | | | | |
| 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);
}
```

# De Morgan's Laws

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

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

```
!(front != null && value > front.data)
```

$$\equiv$$

```
front == null || value <= front.data
```

You've been using these for a while!

# Law of Implication

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

| $p$ | $q$ | $p \to q$ | $\neg p$ | $\neg p \vee q$ | $p \to q \leftrightarrow \neg p \vee q$ |
|---|---|---|---|---|---|
| T | T | | | | |
| T | F | | | | |
| F | T | | | | |
| F | F | | | | |

# Law of Implication

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

| $p$ | $q$ | $p \rightarrow q$ | $\neg p$ | $\neg p \lor q$ | $p \rightarrow q \leftrightarrow \neg p \lor q$ |
|---|---|---|---|---|---|
| T | T | T | F | T | T |
| T | F | F | F | F | T |
| F | T | T | T | T | T |
| F | F | T | T | T | T |

# Some Equivalences Related to Implication

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

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

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

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

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

# Digital Circuits

## Computing With Logic

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

## Gates

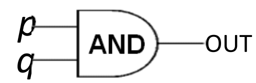
- Take inputs and produce outputs (functions)
- Several kinds of gates
- Correspond to propositional connectives (most of them)

# And Gate

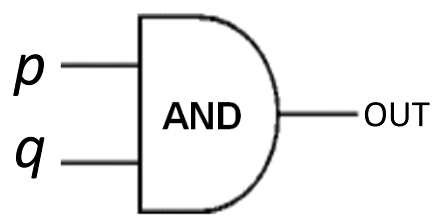
## AND Connective vs. AND Gate

**$p \wedge q$**

| $p$ | $q$ | $p \wedge q$ |
|---|---|---|
| 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 |



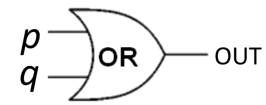

"block looks like D of AND"

# Or Gate

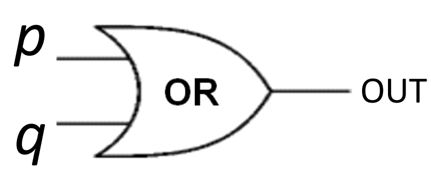
## OR Connective vs. OR Gate

$p \vee q$

| $p$ | $q$ | $p \vee q$ |
|---|---|---|
| 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 |

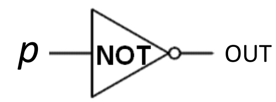



"arrowhead block looks like V"

# Not Gates

**NOT Connective** vs. **NOT Gate**

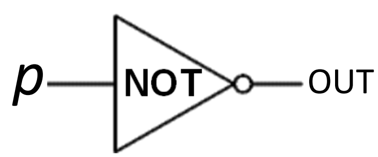
¬*p*




Also called *inverter*

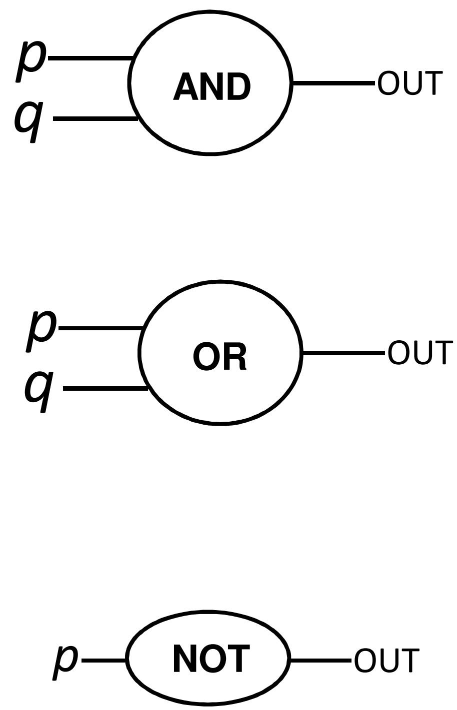
| *p* | ¬ *p* |
|---|---|
| T | F |
| F | T |

| *p* | OUT |
|---|---|
| 1 | 0 |
| 0 | 1 |

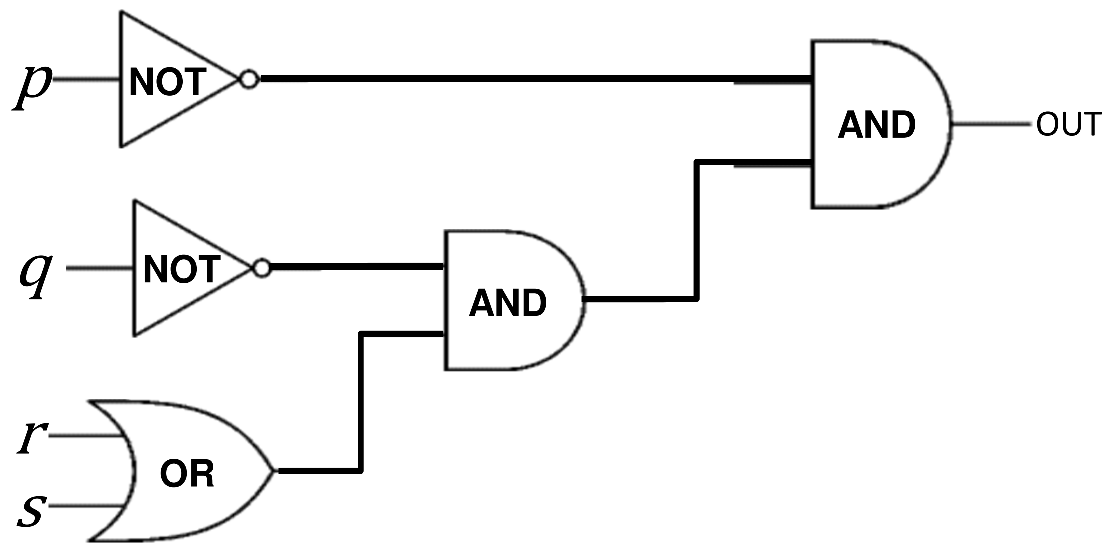


# Blobs are Okay!

You may write gates using blobs instead of shapes!

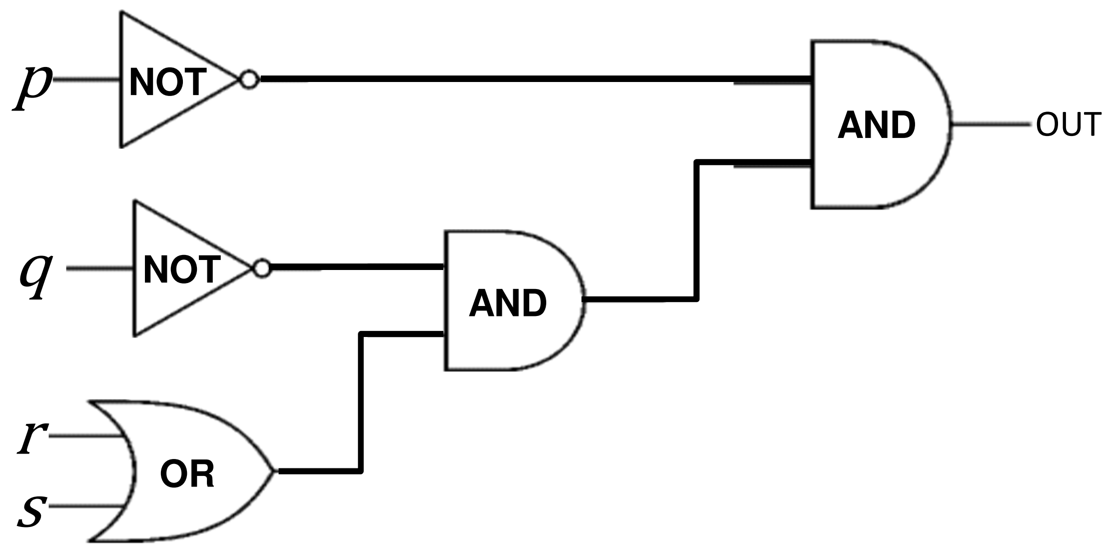


# Combinational Logic Circuits




Values get sent along wires connecting gates

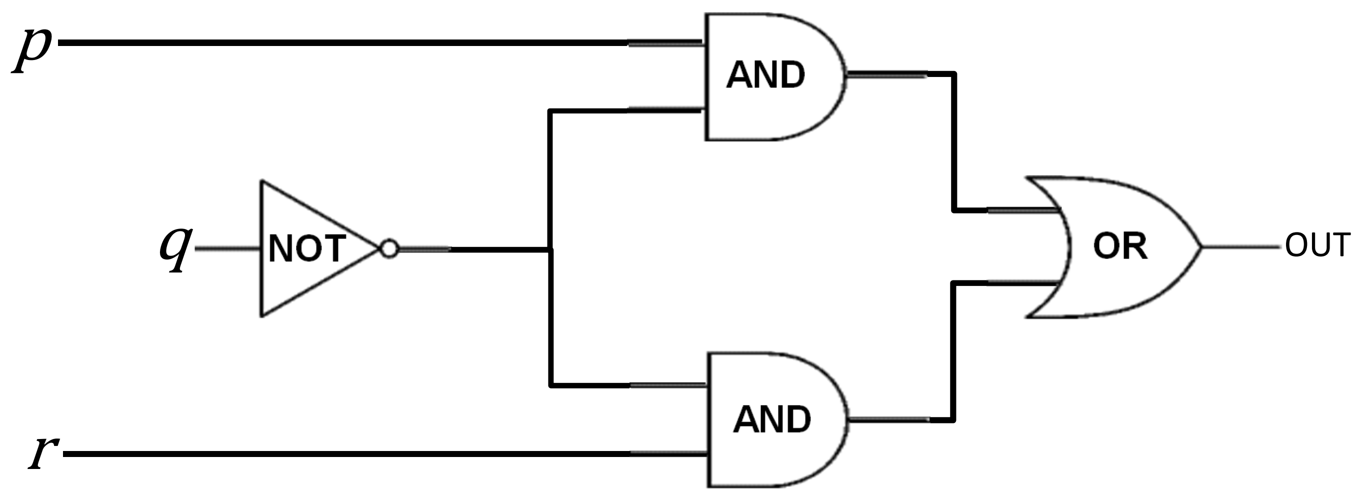
# Combinational Logic Circuits




Values get sent along wires connecting gates

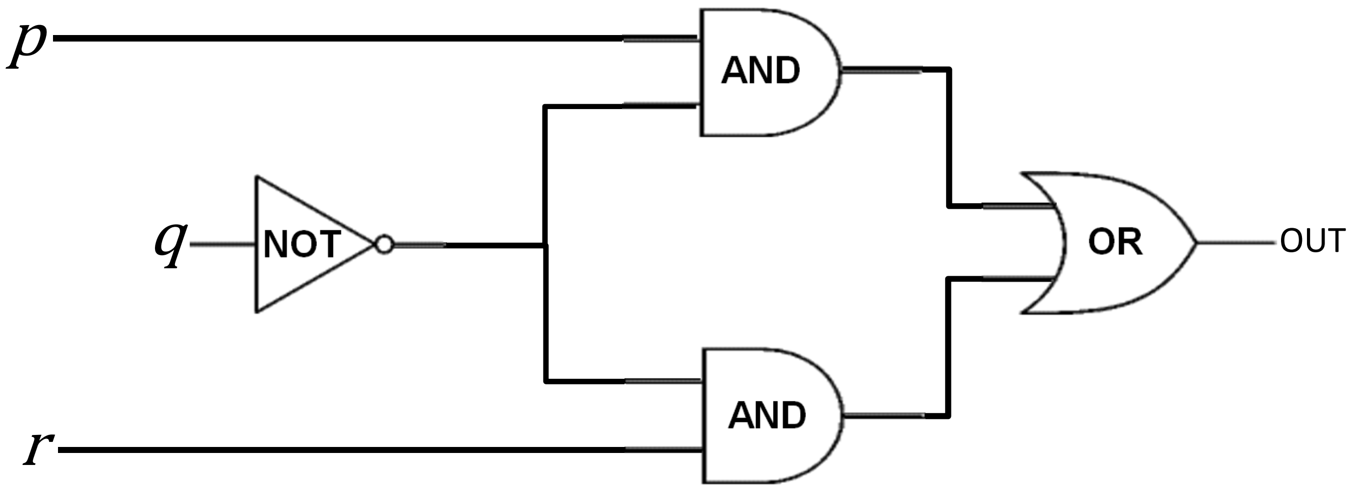
$$\neg p \wedge (\neg q \wedge (r \vee s))$$

# Combinational Logic Circuits




Wires can send one value to multiple gates!

# Combinational Logic Circuits




Wires can send one value to multiple gates!

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

# 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)**

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

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

# 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