Goals

- Translate between simple natural language statements and propositional logic.
- Given a propositional logic statement, apply an equivalent rule to create an equivalent statement.
- Explore alternate forms of propositional logic statements by application of equivalence rules.
- Evaluate propositional logic as a “model of computation” for combinational circuits, including at least one explicit shortfall (e.g., referencing gate delays, fan-out, transistor count, wire length, instabilities, shared sub-circuits, etc.).
Conditional

<table>
<thead>
<tr>
<th>if $p$ then $q$</th>
<th>$p$ only if $q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$ implies $q$</td>
<td>$q$ if $p$</td>
</tr>
</tbody>
</table>

If you show up for work Monday morning, then you will get the job.

- $p$ is called the hypothesis (or antecedent) and $q$ is called the conclusion (or consequent).
- The sentence says nothing about what will happen if the condition is not met.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$p \rightarrow q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
$p \rightarrow q$

- Equivalence: $\sim p \lor q$
- Negation: $p \land \sim q$
- Contrapositive: $\sim q \rightarrow \sim p$
- Converse: $q \rightarrow p$
- Inverse: $\sim p \rightarrow \sim q$

Order of Operations for Logical Operators:

- Negation
- Disjunction and Conjunction
- Conditional and Biconditional
It is true if both p and q have the same truth values and is false if p and q have opposite truth values.

- p is a sufficient condition for q means if p then q.
- p is a necessary condition for q means if q then p.
- p is a necessary and sufficient condition for q means p if, and only if, q.

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p ↔ q</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
Very well done overall!!! \o/

Question 3: Which of the following has the same meaning as $p \rightarrow \sim q$?

A  Anytime that $p$ is true, $q$ must be false.
B  $p$ can not be true unless $q$ is false.
C  $q$ can not be false unless $p$ is true.
D  if $p$ is true then $q$ is false.
E  $q$ and $p$ can never have the same truth value (both true or both false).
p = Metallica is playing in Vancouver
q = I’m going to a Metallica’s concert

Let me being happy means that the statement is True.

**Conditional:** \( p \rightarrow q 

If Metallica plays in Vancouver and I miss, I will be sad. If Metallica doesn’t come to Vancouver, then maybe I’ll travel to see them or maybe I’ll be home, but I am not sad because I am not missing their concert in Vancouver.
p = Metallica is playing in Vancouver
q = I’m going to a Metallica’s concert

Let me being happy means that the statement is True.

Inverse: \( \sim p \rightarrow \sim q \)

If Metallica doesn’t play in Vancouver, I’m not going to see them. I will be sad if I have to travel to see them and I don’t like them so much to be sad if they come here and I miss it.
Examples

\[ p = \text{Metallica is playing in Vancouver} \]
\[ q = \text{I’m going to a Metallica’s concert} \]

Let me being happy means that the statement is True.

Converse: \( q \rightarrow p \)

If I’m going to a Metallica’s concert, then they are playing in Vancouver. But maybe they are playing here and I’m not there. I’m not such a fan, so I’m not traveling and I would not be sad if I missed it.
Examples

\[ p = \text{Metallica is playing in Vancouver} \]
\[ q = \text{I’m going to a Metallica’s concert} \]

Let me being happy means that the statement is True.

**Contrapositive:** \(~q \rightarrow \sim p\)

If I’m not going to a Metallica’s concert, then they are not playing in Vancouver, because if they were, I’d be crying about missing it.
Examples

\[ p = \text{Metallica is playing in Vancouver} \]
\[ q = \text{I’m going to a Metallica’s concert} \]

Let me being happy means that the statement is True.

**Biconditional:**  \( p \iff q \)

If Metallica comes to Vancouver, I’m at their show and I am only at their show here. Traveling to see them makes me as sad as loosing their concert here.
Logic vs Everyday English

Be careful!

The meaning of \textit{if} \textit{p} then \textit{q} in propositional language is not quite the same as in normal language.

- In logic, a hypothesis (p) and conclusion (q) are not required to have related subject matters.
- In informal language, simple conditionals are often used to mean biconditionals.

\textit{If you rob a bank, then you will go to jail!}

- Assuming the statement is truth, we need to evaluate:
  - The truth value of \textit{p} (whether or not I lied)
  - The truth value of \textit{q} (whether or not you will go to jail)
If you rob a bank, will you go to jail?

a. Yes
b. No
c. Maybe
If you go to jail, have you robbed a bank?

a. Yes
b. No
c. Maybe
Logical Equivalence

Two propositions are logically equivalent if they have the same Truth Table.

Showing equivalence:

1. Draw both truth tables and check if they are equal.
2. Use rules to change one proposition into the other:
   - We state the theorem we want to prove.
   - We indicate the beginning of the proof by \textbf{Proof:}
   - We start with one side and work towards the other, one step at a time, justifying each step.
   - We indicate the end of the proof by \textbf{QED} (\textit{Quod erat demonstrandum}) or ■.

Tip: Usually we will simplify the more complicated proposition, instead of trying to complicate the simpler one.
<table>
<thead>
<tr>
<th>Logical Equivalence Proofs: Laws</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Identity Laws</strong></td>
</tr>
<tr>
<td>( p \land T \equiv p )</td>
</tr>
<tr>
<td>( p \lor F \equiv p )</td>
</tr>
<tr>
<td><strong>Universal Bounds Laws</strong></td>
</tr>
<tr>
<td>( p \land F \equiv F )</td>
</tr>
<tr>
<td>( p \lor T \equiv T )</td>
</tr>
<tr>
<td><strong>Idempotent Laws</strong></td>
</tr>
<tr>
<td>( p \land p \equiv p )</td>
</tr>
<tr>
<td>( p \lor p \equiv p )</td>
</tr>
</tbody>
</table>
## Logical Equivalence Proofs: Laws

### Commutative Laws

<table>
<thead>
<tr>
<th>Formula</th>
<th>Equivalent Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p \land q$</td>
<td>$q \land p$</td>
</tr>
<tr>
<td>$p \lor q$</td>
<td>$q \lor p$</td>
</tr>
</tbody>
</table>

### Associative Laws

<table>
<thead>
<tr>
<th>Formula</th>
<th>Equivalent Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p \land (q \land r)$</td>
<td>$(p \land q) \land r$</td>
</tr>
<tr>
<td>$p \lor (q \lor r)$</td>
<td>$(p \lor q) \lor r$</td>
</tr>
</tbody>
</table>

### Distributive Laws

<table>
<thead>
<tr>
<th>Formula</th>
<th>Equivalent Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p \lor (q \land r)$</td>
<td>$(p \lor q) \land (p \lor r)$</td>
</tr>
<tr>
<td>$p \land (q \lor r)$</td>
<td>$(p \land q) \lor (p \land r)$</td>
</tr>
</tbody>
</table>
### Absorption Laws

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

### Negation Laws

- $p \land \sim p \equiv F$
- $p \lor \sim p \equiv T$

### Double Negation Law

- $\sim \sim p \equiv p$
### Logical Equivalence Proofs: Laws

#### DeMorgan’s Laws

\[
\neg(p \land q) \equiv (\neg p) \lor (\neg q) \\
\neg(p \lor q) \equiv (\neg p) \land (\neg q)
\]

#### Definition of XOR

\[
p \oplus q \equiv (p \lor q) \land \neg (p \land q)
\]

#### Definition of IF

\[
p \rightarrow q \equiv \neg p \lor q
\]

#### Contrapositive Law

\[
p \rightarrow q \equiv (\neg q) \rightarrow (\neg p)
\]
<table>
<thead>
<tr>
<th>Identity Laws</th>
<th>( p \land T \equiv p )</th>
<th>Universal Bounds Laws</th>
<th>( p \land F \equiv F )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p \lor F \equiv p )</td>
<td></td>
<td>( p \lor T \equiv T )</td>
<td></td>
</tr>
<tr>
<td>Idempotent Laws</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( p \land p \equiv p )</td>
<td></td>
<td>( p \lor p \equiv p )</td>
</tr>
<tr>
<td>Commutative Laws</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( p \land q \equiv q \land p )</td>
<td></td>
<td>( p \lor q \equiv q \lor p )</td>
</tr>
<tr>
<td>Associative Laws</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>((p \land q) \land r \equiv (p \land q) \land r )</td>
<td></td>
<td>((p \lor q) \lor r \equiv (p \lor q) \lor r )</td>
</tr>
<tr>
<td>Distributive Laws</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>((p \land q) \lor r \equiv (p \lor q) \lor (p \lor r) )</td>
<td></td>
<td>((p \lor q) \land r \equiv (p \lor q) \lor (p \lor r) )</td>
</tr>
<tr>
<td>Absorption Laws</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p \lor (p \land q) \equiv p )</td>
<td></td>
<td>( p \land (p \lor q) \equiv p )</td>
<td></td>
</tr>
<tr>
<td>Negation Laws</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p \land \sim p \equiv F )</td>
<td></td>
<td>( p \lor \sim p \equiv T )</td>
<td></td>
</tr>
<tr>
<td>Double Negation Law</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \sim \sim p \equiv p )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DeMorgan’s Laws</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \sim (p \land q) \equiv (\sim p) \lor (\sim q) )</td>
<td></td>
<td>( \sim (p \lor q) \equiv (\sim p) \land (\sim q) )</td>
<td></td>
</tr>
<tr>
<td>Definition of XOR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p \oplus q \equiv (p \lor q) \land \sim (p \land q) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Definition of IF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p \rightarrow q \equiv \sim p \lor q )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contrapositive Law</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p \rightarrow q \equiv (\sim q) \rightarrow (\sim p) )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Prove that \((\sim a \land b) \lor a \equiv a \lor b\).

Proof: \((\sim a \land b) \lor a \equiv a \lor (\sim a \land b)\)  
\[\equiv (a \lor \sim a) \land (a \lor b)\]  
\[\equiv ????\]  
\[\equiv a \lor b\]  
Commutative Law  
Distributive Law  
Identity Law

What is missing?

a. \((a \lor b)\)

b. \(F \land (a \lor b)\)

c. \(a \land (a \lor b)\)

d. Something else

e. Not enough info to tell
It has 3 inputs $a$, $b$, select and just one output $m$. It outputs $a$ if select is False, and $b$ if select is True.
Let’s implement a circuit based on this Truth Table.

\[ m = (\neg sel \land a) \lor (sel \land b) \]
Let’s implement a circuit based on this Truth Table.

\[ m = (\sim sel \land a) \lor (sel \land b) \]

But there is a problem with this circuit...
Gates are not instantaneous!!!

- If we change the input of a gate at time $t = 0$.
- The output of the gate will only reflect the change some time later.
- This time gap is called the gate delay.
Assume the gate delay is 10 ns and $a$, $b$, $sel$ are initially True.

How long will it take before output correctly reflects any changes in $a$, $b$, $sel$?

- a. 10 ns
- b. 20 ns
- c. 30 ns
- d. 40 ns
- e. It may never happen.
t = 0 ns: we are going to change sel from True to False and track what happens.
**t = 5 ns:** since the gate delay is 10 ns, no reflection is seen on the circuit.
Multiplexers

\[ t = 10 \text{ ns}: \text{ all gates can reflect the changes seen so far, but not those made now.} \]
Multiplexers

\[ t = 20 \text{ ns}: \text{ all gates can reflect the changes seen so far and this time it changes the output.} \]
\[ t = 30 \text{ ns} : \] finally the output gate changes again to reflect the correct answer.
So the output changed from T (old output) to F and then to T (new output).

This is called an **instability**.

The cause of the problem:

- The information from select travels on two different paths to the output
- These paths contain different numbers of gates
- The shorter path may affect the output until the information on the longer path catches up
Which one(s) of the following operation may cause an instability?

a. Changing a or b only
b. Changing select, when at exactly one of a, b is F
c. Changing select, when both a, b are F
d. Both (a) and (b)
e. None of (a), (b) or (c).
When \( a \) and \( b \) are False, the output of both AND is False no matter the value of sel so changing sel does not affect on the output and there is no instability.
When $a$ is False and $b$ is True, the AND with $a$ as input is always False, so the output will come only from the AND with $b$ as input. Since there is only one path for $sel$, then there is no instability.
When $a$ is True and $b$ is False, the AND with $b$ as input is always False, so the output will come only from the AND with $a$ as input. Since there is only one path for $sel$, then there is no instability.
When \( a \) and \( b \) are True, the output should be True, we can enforce that by adding an AND gate in the middle. For all other cases it will be False and maintain the original output.
Module 02. Conditionals and Logical Equivalences

Questions?

Ask CPSC 121

http://www.cs.ubc.ca/~mochetti/askCPSC121.html
Exercise

Prove:

\[(a \land \sim b) \lor (\sim a \land b) \equiv (a \lor b) \land \sim (a \land b)\]
Exercise: Solution

Prove:

\[(a \land \sim b) \lor (\sim a \land b) \equiv (a \lor b) \land \sim(a \land b)\]

\[(a \lor b) \land \sim(a \land b) \equiv (a \lor b) \land (\sim a \land \sim b) \quad \text{(De Morgan’s Law)}
\equiv ((a \lor b) \land \sim a) \lor ((a \lor b) \land \sim b) \quad \text{(Distributive Law)}
\equiv (\sim a \land (a \lor b)) \lor (\sim b \land (a \lor b)) \quad \text{(Commutative Law)}
\equiv ((\sim a \land a) \lor (\sim a \land b)) \lor ((\sim b \land a) \lor (\sim b \land b)) \quad \text{(Dist. Law)}
\equiv (F \lor (\sim a \land b)) \lor ((\sim b \land a) \lor F) \quad \text{(Negation Law)}
\equiv (\sim a \land b) \lor (\sim b \land a) \quad \text{(Identity Law)}
\equiv (\sim a \land b) \lor (a \land \sim b) \quad \text{(Commutative Law)}
\equiv (a \land \sim b) \lor (\sim a \land b) \quad \text{(Commutative Law)}\]
Consider the code:

```python
if target = value then
    if lean-left-mode = true then
        call the go-left() routine
    else
        call the go-right() routine
else if target < value then
    call the go-left() routine
else
    call the go-right routine
```

Let \( g_l \) mean “the go-left() routine is called”. Complete the following:

\[ g_l \leftrightarrow \]
The Java [String] equals() method returns true if and only if the argument is not null and is a String object that represents the same sequence of characters as this object.

Let
- \( n_1 \): the string is null
- \( n_2 \): the argument is null
- \( n_t \): the method returns true
- \( s \): the two objects are strings that represent the same sequence of characters.

Is the sentence logically equivalent to \( n_t \leftrightarrow (n_1 \land n_2) \lor s \)? Why or why not?
STATEMENT: IF YOU'RE NOT PART OF THE SOLUTION, YOU'RE PART OF THE PROBLEM.

IN SYMBOLIC LOGIC: $\neg S \rightarrow P$

(1) $\neg S \rightarrow P$ (given)
(2) $\neg P \rightarrow S$ (law of contraposition)

NEW STATEMENT: IF YOU'RE NOT PART OF THE PROBLEM, YOU'RE PART OF THE SOLUTION.

YOU DID NOTHING ALL DAY!

I'M HELPING SOLVE THE RACE PROBLEM!