Logic

What is Logic?

Definition: *Philosophical Logic*

The study of thought and reasoning, including arguments and proof techniques.

This is the classical notion of logic

Definition: *Mathematical Logic*

The use of formal languages and grammars to represent syntax and semantics of computation

Propositional Logic

Propositional Logic is part of Mathematical Logic. Versions include:

What we use in this course

- *First Order Logic* (FOL a.ka. *First Order Predicate Calculus (FOPC)) includes simple term variables and quantifications*
- *Second Order Logic* allows its variable to represent more complex structures (in particular, predicates)
- *Modal Logic* adds support for modalities; that is concepts such as possibility and necessity.

Why Are We Studying Logic?

Logic has many applications to computer science:

- Logic is the foundation of computer operation
- Logical conditions are common in programs:
  - Selection: $\text{if (score} \ <= \ \text{max)} \ \{\ldots\}$
  - Iteration: $\text{while (i<limit \&\& list[i]!= stopValue)} \ldots$
- Structures in computing have properties that need to be proven
  - Examples: Trees, Graphs, Recursive Algorithms
- Whole programs can be proven correct
- Computational Linguistics must represent and reason about human language, and language represents thought (and thus logic)

Propositions

Definition: *Proposition*

A declarative sentence that is either true (**T**) or false (**F**), but not both.

Definition: *Atomic (simple) proposition*

A proposition that cannot be expressed in terms of simpler propositions (it includes no logical operators)

Propositions - Examples

- Propositions:
  - 1 + 1 = 2 **(True)**
  - 2 + 2 = 5 **(False)**
  - Tucson summers never get above 100 degrees **(False)**

Truth Value of Proposition

- Not Propositions:
  - $x * y < z$ **(depends on x, y, z)**
  - What time is it? **(Question)**
  - This sentence is false. **(Paradox)**

Propositional Variables

- To reduce writing required, we label propositions with lower-case letters, called *propositional variables*.
- Examples:
  - h: "Helena is the capital of Montana"
  - For brevity, we use either:
    1. A meaningful letter (h)
    2. p, q, r, s,… (these are the standard letters used)

Compound Propositions

Definition: *Compound Proposition*

A proposition formed by combining propositions using logical operators

What do we use to combine them?

Logical Operators

- *Connective Logical Operators* - Operators used to combine 2 or more propositions

  1. Conjunctions
  2. Disjunctions

- Logical Operators on a single proposition:

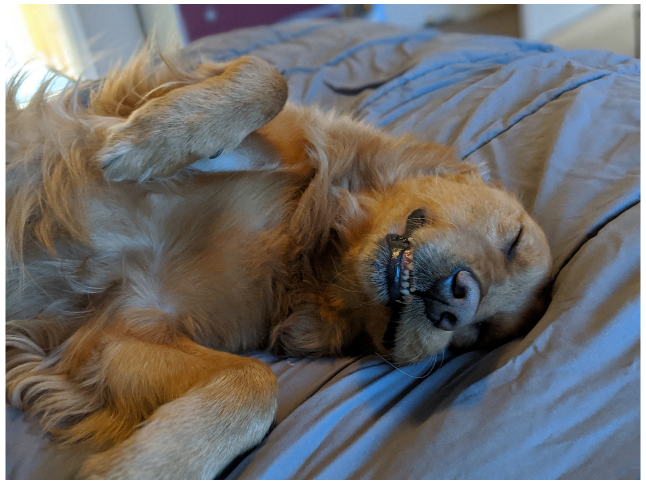
  3. Negations

Conjunctions

Definition: *Conjunction*

A conjunction of p and q is the proposition "p and q"

- Conjunctions are denoted by $p \wedge q$
- They are only true when both p and q are true
- Examples:
  - p: Rodger is a dog
  - q: Rodger likes to bark at cats.
  - $p \wedge q$: Rodger is a dog and likes to bark at cats.



Disjunctions

Definition: *Disjunction*

A disjunction of p and q is the proposition "p or q"

- Disjunctions are denoted by $p \vee q$
- Example:
  - p: Harry will destroy the Horcruxes.
  - q: Harry will find the Deathly Hallows.
  - $p \vee q$: Harry will destroy the Horcruxes or find the Deathly Hallows.

Under what circumstances is $p \vee q$ true?

Disjunctions - Inclusive

- Proposition: Harry will destroy the Horcruxes (p) or find the Deathly Hallows (q)
- $p \vee q$ is true if:
  1. p is true (Harry destroys the Horcruxes)
  2. q is true (Harry finds the deathly hallows)
  3. Both p and q are true, (Harry destroys the Horcruxes and finds the Deathly Hallows)
- Since the third option is acceptable, the disjunction is *inclusive*. By default, $p \vee q$ denotes *inclusive disjunctions*.

Disjunctions - Exclusive

- Consider the proposition: Harry will destroy the Horcruxes (p) or Voldemort will be immortal (q)
- $p \vee q$ is true if:
  1. p is true (Harry destroys the Horcruxes)
  2. q is true (Voldemort is immortal)
- But it's not true if:
  - Both p and q are true, (Harry destroys the Horcruxes but Voldemort is still immortal.)
- Since the third option is not acceptable, the disjunction is *exclusive.* This is denoted by $p \oplus q$ (or XOR).

Disjunction Examples

- The computer was a MacBook or a Thinkpad
- For dessert, you can have cake or ice cream
- His birthday is in June or July.

Negation

Definition: *Negation*

The negation of proposition p, is the statement “it is not the case that p.”

- Negations are denoted by $\neg p$ (also denoted $\overline{p}$)
- Example:
  - p : I love computers
  - $\neg p$: It is not the case that I love computers
    - I do not love computers
    - I hate computers

Negations - Examples

- p: Eleanor took a nap
  - $\neg p$:Eleanor did not take a nap
  - Eleanor skipped her nap
- $\neg p$: they will lose the game
  - p : they will win the game

Truth Tables

- *Truth tables* show us all possible truth values of a given proposition
- Structure of truth table for $\neg(p \wedge q)$:

Proposition Labels → p, q

Sequence of propositions (building to the proposition of interest) → $(p \wedge q)$, $\neg(p \wedge q)$

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

All possible combinations of logical values → columns p, q

Evaluations → columns $(p \wedge q)$, $\neg(p \wedge q)$

Truth Tables

- *Truth tables* of $\wedge$,$\vee$, $\oplus$, and $\neg$.

NOT ($\neg$)

p	$\neg p$
T	F
F	T

AND ($\bigwedge$)

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

OR ($\bigvee$)

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

XOR ($\bigoplus$)

p	q	$(p \oplus q)$
T	T	F
T	F	T
F	T	T
F	F	F

Truth Tables

- Example: $\neg(q \wedge p) \vee \neg s$

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

Precedence of Logical Operators

- Rosen suggests the precedence order:

Precedence	Operator
Highest	$\neg$
	$\wedge$
$\Updownarrow$	$\vee$
	$\rightarrow$
Lowest	$\leftrightarrow$

(we'll cover these soon)

- There is disagreement among mathematicians
- Always use parentheses to avoid confusion

Operator Associativity

- Given $\neg\neg p$, we evaluate it right to left, $\neg(\neg p)$
  - Negation is right associative
- Given $p \wedge q \wedge r$, we evaluate it left to right $(p \wedge q) \wedge r$
  - This holds for $\vee$ and $\oplus$
  - Conjunctions and both disjunctions are left associative

Equivalence of Propositions

Definition: *Logically Equivalent*

> Two propositions p and q are logically equivalent if they have the same truth values in all possible inputs

- Equivalences are denoted by $p \equiv q$
- Example: is $p \equiv (p \wedge q) \vee p$?

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

Equivalence of Propositions

- Example: Distributive Law - $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$

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

Converting Natural Language to Propositions

- Is *The sky is cloudy* a proposition?
  - Yes, it is an atomic proposition
- Is the following sentence a proposition?
  - *Either Walter deposits his mortgage payment or else he will lose his house and move in with Donna.*
  - Yes!
  - It is a compound proposition built of 3 atomic propositions

Converting Natural Language to Propositions

- Step 1: Identify the atomic (simple) propositions

Either Walter deposits his mortgage payment or else he will lose his house and move in with Donna.

Converting Natural Language to Propositions

- Step 2: Assign easy to remember statement labels

d

Either <u>Walter deposits his mortgage payment</u> or else <u>he will lose his house</u> and <u>move in with Donna</u>.

k m

Converting Natural Language to Propositions

- Step 3: Identify the logical operators

Exclusive Disjunction $\oplus$

d

Either Walter deposits his mortgage payment or else he will lose his house and move in with Donna.

k m

Negation $\neg$

Conjunction $\wedge$

Converting Natural Language to Propositions

- Step 4: Construct the matching logical expression

Exclusive Disjunction $\oplus$

d

Either Walter deposits his mortgage payment or else he will lose his house and move in with Donna.

k m

Negation $\neg$

Conjunction $\wedge$

$$d \oplus (\neg k \wedge m)$$

Converting Natural Language to Propositions

- Why do we need to do this?
  - Expressing Program Conditions

    (x!=6) or (y==‘Y’) and flag

  - Natural Language Understanding

    “Route me to campus with a stop for gas."

  - Proof Setup

    Converting conjectures to logic:

    “The sum of the squares of two odd integers is never a perfect square”

Three Categories of Propositions

Definition: *Tautology*

A proposition that is always **true**, no matter the truth values of proposition variables

Definition: *Contradiction*

A proposition that is always **false**, no matter the truth values of proposition variables

Definition: *Contingency*

A proposition that is neither a *tautology* or *contradiction*

Three Categories of Propositions

- Examples:

Tautology

p	$\neg p$	$(p \vee \neg p)$
T	F	T
F	T	T

Contradiction

p	$\neg p$	$(p \wedge \neg p)$
T	F	F
F	T	F

Contingency

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

Aside: Logical Bit Operations

- Bit operations correspond to logical connectives

<table>
<tr><th>Logical Operator</th><th>Bit Operator</th><th>Name</th><th>Example</th></tr>
<tr><td>¬</td><td>~</td><td>Complement (not)</td><td>~1100 = 0011</td></tr>
<tr><td>∧</td><td>&</td><td>AND</td><td>1100
& 1011
1000</td></tr>
<tr><td>∨</td><td>|</td><td>OR</td><td>1100
| 1011
1111</td></tr>
<tr><td>⊕</td><td>^</td><td>XOR</td><td>1100
^ 1011
0111</td></tr>
</table>

Aside: Logical Bit Operations

- Default Linux File Permissions

```
$ ls -l
-rw-rw-r-- 1 liw liw 836 Nov 10 16:33 drafts/liw-permissions.mdwn
```

[unmask]		000 011 111
[complement of unmask]	&	111 100 000
[default permissions]		110 110 110
[the file's permissions]		110 100 000
		rw- r-- ---

Conditional Propositions

Definition: *Conditional Proposition*

A conditional proposition is one that can be expressed as "if p then q", denoted $p \to q$, where p and q are propositions.

- Example:
  - If the doorbell rings, then my dog will bark.

Conditional Propositions

- In "if p then q", p and q are known by various names:

p		q
(1) Antecedent	—	consequent
(2) Hypothesis	—	conclusion
(3) Sufficient	—	necessary

- Common forms of "if p then q":

  - ▷ if p, then q
  - ▷ if p,q
  - ▷ p implies q
  - ▷ p only if q
  - ▷ p is sufficient for q
  - ▷ a necessary condition for p is q
  - ▷ q unless $\neg p$
  - ▷ q if p
  - ▷ q when p
  - ▷ q whenever p
  - ▷ q follows from p
  - ▷ q is necessary for p
  - ▷ a sufficient condition for q is p
  - ▷ q provided that p

Conditional Propositions

- Example: Rewrite the proposition in the given from:
  - If the bike has 2 wheels you can ride it.

    You can ride the bike if it has 2 wheels

  - The engine will start only if the tank has fuel.

    If the engine will start then the tank has fuel

Truth of Conditional Propositions

- When are conditionals ‘true’?

  If the doorbell rings, then my dog will bark.

- The possibilities:

  1. Antecedent true, Consequent true; statement is: __**T**__
  2. Antecedent true, Consequent false; statement is: __**F**__
  3. Antecedent false, Consequent true; statement is: __**T**__
  4. Antecedent false, Consequent false; statement is: __**T**__

Truth of Conditional Propositions

- Example:

```
if (y < x){
  temp = x; x = y; y = temp;
}
```

Truth of Conditional Propositions

- Example:

```
if (y < x){
    temp = x; x = y; y = temp;
}
```

p q

$$p \rightarrow q$$

When **p** is **False**, **q** is irrelevant, yet the Java statement is still legal (or **True**)

Truth of Conditional Propositions

- Other Examples:
  - “If elected, I will lower taxes.”
  - “If it is below 90 this evening, I will go for a run”.
  - “If it rains today, I won’t water my plants.”
  - “If you push on the door, it will open”

Equivalences of OR, AND, Implication

- Truth tables for OR, AND, and Implication

All 3 operators have one value that differs!

OR ($\bigvee$)

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

AND ($\bigwedge$)

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

Implies ($\rightarrow$)

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

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

Can get proposition equivalent to implication from AND and OR

Inverse, Converse, & Contrapositive

Definition: *Inverse*

Given $p \rightarrow q$, the inverse is $\neg p \rightarrow \neg q$

Definition: *Converse*

Given $p \rightarrow q$, the converse is $q \rightarrow p$

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

Note: Inverse $\equiv$ Converse $\not\equiv$ Original

Inverse, Converse, & Contrapositive

Definition: *Contrapositive*

Given $p \rightarrow q$, the contrapositive is $\neg q \rightarrow \neg p$

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

Note: $p \rightarrow q \equiv \neg q \rightarrow \neg p$

Examples: English Translation

- Proposition: If you got an A on the final, you pass the class.
- Converse: If you pass the class, you got an A on the final.
- Inverse: If you did not get an A on the final, you do not pass the class.
- Contrapositive: If you do not pass the class, you did not get an A on the final.

English -> Logic

- Remember our steps for converting natural language to propositional logic:
  - Step 1: Identify the atomic (simple) propositions
  - Step 2: Assign easy to remember statement labels
  - Step 3: Identify the logical operators
  - Step 4: Construct the matching logical expression

English -> Logic

- Translate the proposition: *When she loses the poker tournament, she will keep her job and won't buy a round of drinks*

- Following our step defined earlier:

When she loses the poker tournament (p), she will keep her job (j) and won't buy a round of drinks (d)

p: she wins the poker tournament

j: she will keep her job

d: she will buy a round of drinks

English -> Logic

- Translate the proposition: *When she loses the poker tournament, she will keep her job and won't buy a round of drinks*

- Following our step defined earlier:

If $\quad \neg \quad p$

When she loses the poker tournament, she will keep her job and won't buy a round of drinks

$j \quad \wedge \quad \neg \quad d$

$$\neg p \rightarrow (j \wedge \neg d)$$

English -> Logic

- Translate the proposition: *If I don't take my dog for a walk or a run, then he won't be tired for bed.*

- Following our step defined earlier:

w

If I don't take my dog for a walk or a run, then he won't be tired for bed.

t

w : I take my dog for a walk

r : I take my dog for a run

t : he is tired for bed

English -> Logic

- Translate the proposition: *If I don't take my dog for a walk or a run, then he won't be tired for bed.*

- Following our step defined earlier:

If ¬ w $\oplus$ r

If I don't take my dog for a walk or a run, then he won't be tired for bed.

¬ t

English -> Logic

- Translate the proposition: *If I don't take my dog for a walk or a run, then he won't be tired for bed.*

- Following our step defined earlier:

If ¬ w ⊕ r

If I don't take my dog for a walk or a run, then he won't be tired for bed.

¬ t

Two possibilities: (1) $(\neg w \oplus \neg r) \rightarrow \neg t$ **(2)** $\neg(w \oplus r) \rightarrow \neg t$

Which is correct?

English -> Logic

- Translate the proposition: *If I don't take my dog for a walk or a run, then he won't be tired for bed.*

- Following our step defined earlier:

If $\neg$ w $\oplus$ r

If I don't take my dog for a walk or a run, then he won't be tired for bed.

$\neg$ t

Two possibilities: (1) $(\neg w \oplus \neg r) \rightarrow \neg t$ (2) $\neg(w \oplus r) \rightarrow \neg t$

Consider English Contrapositive:
If my dog is tired for bed, I took him for a walk or a run.

This [$t \rightarrow (w \oplus r)$] is the contrapositive of (2) so (2) is correct.

English -> Logic

- Translate the proposition: *If I don't take my dog for a walk or a run, then he won't be tired for bed.*

- Following our step defined earlier:

If ¬ w $\oplus$ r

If I don't take my dog for a walk or a run, then he won't be tired for bed.

¬ t

Two possibilities: (1) $(\neg w \oplus \neg r) \rightarrow \neg t$ **(2)** $\neg(w \oplus r) \rightarrow \neg t$

This $[t \rightarrow (w \oplus r)]$ **is the contrapositive of (2) so (2) is correct.**

Note: $w \oplus r \equiv \neg w \oplus \neg r \not\equiv \neg(w \oplus r)$

Biconditional Propositions

- What is the meaning of:

A triangle is equilateral if and only if all three angles are equal

$\underbrace{\text{triangle is equilateral}}_{t}$ ... $\underbrace{\text{all three angles are equal}}_{a}$

IF	AND	ONLY IF
t **if** a		t **only if** a
if a**, then** t		**if** t**, then** a
$a \rightarrow t$	$\wedge$	$t \rightarrow a$

$$(a \rightarrow t) \wedge (t \rightarrow a)$$

Biconditional Propositions

Definition: *Biconditional Proposition*

A biconditional statement is the proposition "p if and only if q" (p iff q). It is denoted by the symbol $\leftrightarrow$ ($p \leftrightarrow q$).

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

p	q	$p \rightarrow q$	$q \rightarrow p$	$(p \rightarrow q) \land (q \rightarrow p)$	$p \leftrightarrow q$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	T	F	F	F
F	F	T	T	T	T

Biconditionals and Logical Equivalence

- Previously, we defined *Logically Equivalent* as

  Two propositions p and q are logically equivalent if they have the same truth values in all possible inputs

- We can introduce a second definition using Biconditionals
- Before we do that:
  - Remember: *Tautology*

    A proposition that is always **true**, no matter the truth values of proposition variables

Biconditionals and Logical Equivalence

Definition: *Logically Equivalent (2)*

Two propositions p and q are logically equivalent $(p \equiv q)$ if $p \leftrightarrow q$ is a *tautology*

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

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

De Morgan's Laws

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

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

- Show $\neg(p \wedge q) \equiv \neg p \vee \neg q$:

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

Example: Using De Morgan's

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

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

- Show $\neg(a \vee (b \vee c)) \equiv \neg a \wedge \neg b \wedge \neg c$.

$$\begin{aligned} \neg(a \vee (b \vee c)) &\equiv \neg a \wedge \neg(b \vee c) && \text{(De Morgan 2)} \\ &\equiv \neg a \wedge (\neg b \wedge \neg c) && \text{(De Morgan 2)} \\ &\equiv \neg a \wedge \neg b \wedge \neg c && \text{(Associativity of } \wedge \text{)} \end{aligned}$$

Example: De Morgan's Laws and Programming

- Checking to see if a score is ***not*** a 'B'

- Version 1: $\underbrace{(x < 80)}_{p} \,||\, \underbrace{(x >= 90)}_{q}$ $\quad p \vee q$

- Version 2: $!\,(\underbrace{x >= 80}_{\neg p} \;\&\&\; \underbrace{x < 90}_{\neg q})$ $\quad \neg(\neg p \wedge \neg q)$

$$p \vee q \equiv \neg\neg(p \vee q)$$ **Double negative**

$$\equiv \neg(\neg p \wedge \neg q)$$ **De Morgan's (2)**

Common Logical Equivalences

Table I: Some Equivalences using AND ($\wedge$) and OR ($\vee$):

(a)	$p \wedge p \equiv p, \quad p \vee p \equiv p$	Idempotent Laws
(b)	$p \vee \mathbf{T} \equiv \mathbf{T}, \quad p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination Laws
(c)	$p \wedge \mathbf{T} \equiv p, \quad p \vee \mathbf{F} \equiv p$	Identity Laws
(d)	$p \wedge q \equiv q \wedge p$ $p \vee q \equiv q \vee p$	Commutative Laws
(e)	$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$ $(p \vee q) \vee r \equiv p \vee (q \vee r)$	Associative Laws
(f)	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$ $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	Distributive Laws
(g)	$p \wedge (p \vee q) \equiv p$ $p \vee (p \wedge q) \equiv p$	Absorption Laws

Table II: Some More Equivalences (adding $\neg$):

(a)	$\neg(\neg p) \equiv p$	Double Negation
(b)	$p \vee \neg p \equiv \mathbf{T}, \quad p \wedge \neg p \equiv \mathbf{F}$	Negation Laws
(c)	$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's Laws

Common Logical Equivalences

Table III: Still More Equivalences (adding $\rightarrow$):

(a)	$p \rightarrow q \equiv \neg p \vee q$	Law of Implication
(b)	$p \rightarrow q \equiv \neg q \rightarrow \neg p$	Law of the Contrapositive
(c)	$\mathbf{T} \rightarrow p \equiv p$	"Law of the True Antecedent"
(d)	$p \rightarrow \mathbf{F} \equiv \neg p$	"Law of the False Consequent"
(e)	$p \rightarrow p \equiv \mathbf{T}$	Self-implication (a.k.a. Reflexivity)
(f)	$p \rightarrow q \equiv (p \wedge \neg q) \rightarrow \mathbf{F}$	Reductio Ad Absurdum
(g)	$\neg p \rightarrow q \equiv p \vee q$	
(h)	$\neg(p \rightarrow q) \equiv p \wedge \neg q$	
(i)	$\neg(p \rightarrow \neg q) \equiv p \wedge q$	
(j)	$(p \rightarrow q) \vee (q \rightarrow p) \equiv \mathbf{T}$	Totality
(k)	$(p \wedge q) \rightarrow r \equiv p \rightarrow (q \rightarrow r)$	Exportation Law (a.k.a. Currying)
(l)	$(p \wedge q) \rightarrow r \equiv (p \rightarrow r) \vee (q \rightarrow r)$	
(m)	$(p \vee q) \rightarrow r \equiv (p \rightarrow r) \wedge (q \rightarrow r)$	
(n)	$p \rightarrow (q \wedge r) \equiv (p \rightarrow q) \wedge (p \rightarrow r)$	
(o)	$p \rightarrow (q \vee r) \equiv (p \rightarrow q) \vee (p \rightarrow r)$	
(p)	$p \rightarrow (q \rightarrow r) \equiv q \rightarrow (p \rightarrow r)$	Commutativity of Antecedents

Common Logical Equivalences

Table IV: Yet More Equivalences (adding $\oplus$ and $\leftrightarrow$):

(a)	$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$	Definition of Biimplication
(b)	$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$	
(c)	$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$	
(d)	$p \oplus q \equiv (p \wedge \neg q) \vee (\neg p \wedge q)$	Definition of Exclusive Or
(e)	$p \oplus q \equiv \neg(p \leftrightarrow q)$	
(f)	$p \oplus q \equiv p \leftrightarrow \neg q \equiv \neg p \leftrightarrow q$	

You **do not** need to memorize these tables…

…but you **do** need to know how to use them!

Applications of Logical Equivalences

- Question: Is $(p \land q) \to p$ is a *tautology*? (1)
  - Using Truth tables, we see:

p	q	$(p \land q)$	$(p \land q) \to p$
T	T	T	**T**
T	F	F	**T**
F	T	F	**T**
F	F	F	**T**

- Because the expression evaluates to **True** for all possible truth values, the expression is a *tautology*.

Applications of Logical Equivalences

- Question: Is $(p \wedge q) \to p$ is a *tautology*? (2)
  - By application of logical equivalences

$(p \wedge q) \to p \equiv p \to (q \to p)$	**Table 3 (k)**
$\equiv q \to (p \to p)$	**Table 3 (p)**
$\equiv q \to \mathrm{T}$	**Table 3 (e) (reflexivity)**
$\equiv \neg q \vee \mathrm{T}$	**Law of Implication**
$\equiv \mathrm{T}$	**Law of Domination**

Applications of Logical Equivalences

- Question: Is $(p \wedge q) \rightarrow p$ is a *tautology*? (3)
  - By reasoning:
    - When p is **True**: $(\mathrm{T} \wedge q) \rightarrow \mathrm{T} \equiv \mathrm{T}$
      - Anything $\rightarrow \mathrm{T}$ is T (by the definition of $\rightarrow$)
    - When p is **False**:

$$
\begin{aligned}
(\mathrm{F} \wedge q) \rightarrow \mathrm{F} &\equiv \mathrm{F} \rightarrow \mathrm{F} \\
&\equiv \mathrm{T}
\end{aligned}
$$

      - Thus, $(p \wedge q) \rightarrow p$ is a *tautology*?

What we just learned

- Three quick ways to prove that something is a *tautology*:

    1. **Truth Table:** Do all cases resolve to **TRUE**?

    2. **Logical Equivalences**: Can we convert the expression to **TRUE**?

    3. **Reasoning:** Any argument you make; our example did “proof by cases”.

Proving that something is a contradiction

- How to prove that something is a *contradiction*:

  1. **Truth Table:** Do all cases resolve to **FALSE**?
  2. **Logical Equivalences**: Can we convert the expression to **FALSE**?
  3. **Reasoning:** Any argument you make.
  4. **Bonus:** Negate the expression and prove that it is a *tautology*!

Proving that something is a contingency

- How to prove that something is a *contingency*:

  1. **Truth Table:** can we find one case that resolves to **TRUE** and another that resolves to **FALSE**?

  2. **Logical Equivalences**: Can we convert the expression to a simpler expression which is obviously a contingency?

  3. **Reasoning:** Any argument you make. Typically involves simplifying the expression (as with logical equivalences)

Applications of Logical Equivalences

- Programming Example: Assume $\texttt{games}$ is an integer

$$\texttt{if ((games <= 10 || ties > 2) \&\& } \underbrace{\texttt{games >= 11}}_{\neg g}\texttt{)}$$

  - Let g: $\texttt{games <= 10}$ and t: $\texttt{ties > 2}$

$$
\begin{aligned}
(g \lor t) \land \neg g &\equiv (g \land \neg g) \lor (t \land \neg g) && \textbf{Distribution} \\
&\equiv \mathrm{F} \lor (t \land \neg g) && \textbf{Negation} \\
&\equiv (t \land \neg g) && \textbf{Identity}
\end{aligned}
$$

Thus we can rewrite the statement more efficiently as:

$$\texttt{if (ties > 2 \&\& games >= 11) ...}$$

Applications of Logical Equivalences

- Question: Are $(p \wedge q) \vee (p \wedge r)$ and $p \wedge \neg(\neg q \wedge \neg r)$ logically equivalent?

$(p \wedge q) \vee (p \wedge r)$	$\equiv p \wedge (q \vee r)$	**Distributive Law**
	$\equiv p \wedge (\neg q \rightarrow r)$	**Table 3 (g)**
	$\equiv p \wedge \neg\neg(\neg q \rightarrow r)$	**Double Negation**
	$\equiv p \wedge \neg(\neg q \wedge \neg r)$	**Table 3 (h)**

$(p \wedge q) \vee (p \wedge r)$	$\equiv p \wedge (q \vee r)$	**Distributive Law**
	$\equiv p \wedge \neg\neg(q \vee r)$	**Double Negation**
	$\equiv p \wedge \neg(\neg q \wedge \neg r)$	**De Morgan's**