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:

- *First Order Logic* (FOL a.ka. *First Order Predicate Calculus (FOPC)) includes simple term variables and quantifications*

What we use in this course

- *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: *Truth Value* of Proposition
  - 1 + 1 = 2 **(True)**
  - 2 + 2 = 5 **(False)**
  - Tucson summers never get above 100 degrees **(False)**
- Not Propositions:
  - $x * y < z$ **(depends on x, y, z)**
  - What time is it? **(Question)**
  - This sentence is false. **(Paradox)**

Playposit Question

- Which of the following are propositions?
  - Red Rising is a great book.
  - $x^2 > 15$
  - I want a cat.
  - $3^2 > 15$
  - How hot is it outside?

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)

Examples

- Using the propositions from the last playposit question:
  - r: Red Rising is a great book.
  - c: I want a cat.
  - q: $3^2 > 15$

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:

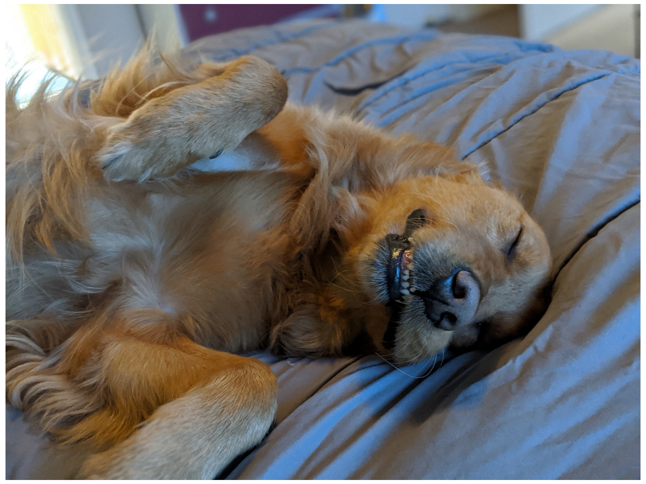
  3. Negations

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.



In Programming…

- Let's say we are creating a program for finding a car
- People often want to cars with good gas mileage (>30mpg) and low cost (<$12,000), so we write a function that checks this

```
def check_car(Car c):
    return (c.gas_mileage > 30) && (c.price < 12000);
```

g**:** the car (c) has gas mileage is greater than 30mpg

p: the car (c) costs less than $12,000

We can rewrite the return statement as :

return $g \wedge p$

Disjunctions

Definition: *Disjunction*

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

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

Under what circumstances is $p \lor q$ 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.

Playposit Question

- Which of the following are exclusive disjunctions?
  - My favorite show is either The Office or Brooklyn 99
  - You can major in CS or Math
  - For Saturday, I will bake rye cookies or apple cake.
  - At 7pm, we will walk Rodger or go to Andrew's house
  - Tomorrow, the high will be above 90 or below 80

In Programming...

- Let’s say we are creating a program for finding a restaurant

- You want to eat burgers or Mediterranean food.

```
def check_restaurant(Restaurant r):
    return (r.type==“Mediterranean”)||
           (r.has_burgers==true)
```

m: the restaurant (r) serves Mediterranean food

b: the restaurant (r) serves burgers

We can rewrite the return statement as :

return $m \vee b$

In Programming...

- If we wanted a Restaurant that either serves steak or vegan food

```
def check_restaurant(Restaurant r):
        return (r.has_steak==true)^(r.is_vegan==true)
```

s: the restaurant (r) serves steak

v: the restaurant (r) is vegan

We can rewrite the return statement as :

return $s \vee v$

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

In Programming…

- We are writing a program to decide what board game to play.

- We don't want to play a card game

```
def check_game(Game g):
    return !(g.card == true)
```

c: the game (g) is a card game

We can rewrite the return statement as :

return $\neg c$

Playposit Question

- Which of the following are negations of “Olives are delicious”?
  - Olives are gross.
  - Pickles are delicious.
  - Olives taste bad.
  - Olives are not delicious.
  - Olives are too squishy.

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

Sequence of propositions (building to the proposition of interest)

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

Evaluations

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

- Break down the compound proposition so that we have 1 column for each nested proposition:

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

There are 7 nested propositions :

The three, labeled, atomic propositions: p, q, r

Four compound proposition to build to the final one:

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

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

Playposit Question

How may columns will the proposition $\neg q \vee (p \wedge q)$ have?

A. 3

B. 4

C. 5

D. 6

E. 7

Playposit Question

- Which of the blanks in the truth table below (labeled (1), (2),(3) and (4)) evaluate to TRUE?

p	q	$\neg q$	$(p \wedge q)$	$\neg q \vee (p \wedge q)$
T	T	F	T	(1)
T	F	T	F	(2)
F	T	F	F	(3)
F	F	T	F	(4)

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

Precedence of Logical Operators

Example: $\neg p \wedge r \vee \neg q \vee s$

This will be evaluated as: $((((\neg p) \wedge r) \vee (\neg q)) \vee s)$

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

To ensure that we read it correctly, it's better to write your compound propositions like this!

Playposit Question

Based on the precedence table and operator associativity rules in the previous 2 slides, which of the following correctly adds parentheses to $q \vee \neg p \wedge \neg s \vee q$?

A. $q \vee \neg(p \wedge \neg(s \vee q))$

B. $(q \vee \neg p) \wedge (\neg s \vee q)$

C. $q \vee (\neg p \wedge \neg(s \vee q))$

D. $(q \vee (\neg p \wedge \neg s)) \vee q$

In Programming...

- Programs evaluate logic in a similar way.
- Consider the Java code:

```java
int x = 10;
int y = 8;
System.out.println((y==8)||(y==7)&&!(x==10));
```

- Java follows our precedence table and the proposition as follows:

(y==8)	(y==7)	(x==10)	!(x==10)	(y==7)&&!(x==10)	(y==8)||(y==7)&&!(x==10)
T	**F**	**T**	**F**	**F**	**T**

In Programming...

- Programs evaluate logic in a similar way.
- Consider the Java code:

Use Parentheses!

```
int x = 10;
int y = 8;
System.out.println((y==8)||(y==7)&&!(x==10));
```

- If Java treated && and || equally and just evaluated left to right:

`(y==8)`	`(y==7)`	`(x==10)`	`(y==8)||(y==7)`	`!(x==10)`	`(y==8)||(y==7)&&!(x==10)`
T	**F**	**T**	**T**	**F**	**F**

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

Playposit Question

Given the truth table below, which of the following is **not** an equivalence:

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

A. $\neg(\neg p \wedge \neg q) \equiv p \wedge q$

C. $p \equiv \neg(\neg p)$

B. $\neg q \equiv \neg p$

D. $\neg(p \wedge q) \equiv \neg p \wedge \neg q$

Converting Natural Language to Propositions

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

Either $\underbrace{\text{Walter deposits his mortgage payment}}^{d}$ or else $\underbrace{\text{he will lose his house}}_{k}$ and $\underbrace{\text{move in with Donna}}_{m}$.

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 $\wedge$

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

Playposit Question

Which of propositional logic statement correctly represents the following English statement:

"You drive to campus and either pay for parking or get a parking ticket"

Let d : you drive to campus,

p : you pay for parking,

t : you get a parking ticket

A. $d \wedge (p \oplus t)$

B. $(d \wedge p) \oplus t$

C. $d \wedge (p \vee t)$

D. $(d \wedge p) \vee t$

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

Playposit Question

Which of the following is **true**?

A. $\neg p$ is a contingency, $p \vee \neg p$ is a contradiction, and $p \wedge \neg p$ is a tautology?

B. $p \wedge \neg p$ is a contingency, $\neg p$ is a contradiction, and $p \vee \neg p$ is a tautology?

C. $p \wedge \neg p$ is a contingency, $p \vee \neg p$ is a contradiction, and $\neg p$ is a tautology?

D. $\neg p$ is a contingency, $p \wedge \neg p$ is a contradiction, and $p \vee \neg p$ is a tautology?

Aside: Logical Bit Operations

- Bit operations correspond to logical connectives

Logical Operator	Bit Operator	Name	Example
$\neg$	~	Complement (not)	~1100 = 0011
$\wedge$	&	AND	1100 & 1011 1000
$\vee$	|	OR	1100 | 1011 1111
$\oplus$	^	XOR	1100 ^ 1011 0111

Aside: Logical Bit Operations

- Default Linux File Permissions
  - Defined for 3 user types: owner, group members, and everyone else.
  - 3 permission types: read (r), write (w), and execute (x)

```
$ ls -l
-rw-rw-rw- 1 rjf users 836 Nov 10 16:33 filename.mdwn
```

  - Default file creation permission: rw- rw- rw-
  - Can use linux umask utility to change file permissions

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-- ---

Playposit Question

- What is the result of the following bit operation (**<u>include</u>** any leading zeros, if necessary)

```
  101011
& 101100
________
```

Conditional Propositions

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

Playposit Question

Write the following conditional proposition in the specified form:

“When I am bored, I watch The Office”

____________________ if ____________________

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){
    int temp = x;
    x = y;
    y = temp;
}
```

Truth of Conditional Propositions

- Example:

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

$$p \rightarrow q$$

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”

Playposit Question

Select all propositions that cause the following conditional proposition to be **true**:

If I get a cat, then my dog will chase it.

- I don’t get a cat
- I get a cat and my dog does not chase it.
- I get a cat and my dog chases it

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 $p \to q$, the contrapositive is $\neg q \to \neg p$

p	q	$p \to q$	$\neg p$	$\neg q$	$\neg q \to \neg p$	$\neg p \to \neg q$	$q \to p$
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: $p \to q \equiv \neg q \to \neg p$

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.

Playposit Question

Give the inverse, converse and contrapositive of the following proposition:

"If I play squash, then I will eat Time Market pizza."

Inverse: If ______________________ then _______________

Converse: If ___________________ then _______________

Contrapositive: If _________________ then _______________

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 $\overbrace{\text{she loses the poker tournament}}^{p}$, $\underbrace{\text{she will keep her job}}_{j}$ and $\underbrace{\text{won't buy a round of drinks}}_{d}$

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 $\qquad \neg \qquad p$

$\boxed{\text{When}}$ $\underline{\text{she } \boxed{\text{loses}} \text{ the poker tournament}}$, $\underline{\text{she will keep}}$
$\underline{\text{her job}}$ $\boxed{\text{and}}$ $\underline{\boxed{\text{won't}} \text{ buy a round of drinks}}$

$j \qquad \wedge \qquad \neg \qquad d$

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

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 $\neg$ 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

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)** $\neg(w \oplus r) \rightarrow \neg t$

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

$\boxed{\text{If}}$ I $\boxed{\text{don't}}$ take my dog for a walk $\boxed{\text{or}}$ a run, then he $\boxed{\text{won't}}$ 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 $\neg$ 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) $\neg(w \oplus r) \rightarrow \neg t$

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

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

Playposit Question

What is the propositional representation for the following statement:

"I will go skiing only if there is snow and I don't have work"

Where s : I go skiing, n : there is snow, w : I have to work

A. $s \rightarrow (n \wedge \neg w)$

B. $(n \wedge \neg w) \rightarrow s$

C. $s \rightarrow (n \wedge w)$

D. $(n \wedge w) \rightarrow s$

Biconditional Propositions

Biconditional Propositions

- What is the meaning of:

A $\underbrace{\text{triangle is equilateral}}_{t}$ if and only if $\underbrace{\text{all three angles are equal}}_{a}$

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 $\leftrightarrow$ ($p \leftrightarrow q$).

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

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

Playposit question

Using the below truth table, is $(\neg p \vee q) \equiv p \rightarrow q$?

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

A. Yes, they are equivalent.

B. No, they are not equivalent.

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

$$
\begin{aligned}
p \vee q &\equiv \neg\neg(p \vee q) && \textbf{Double negative} \\
&\equiv \neg(\neg p \wedge \neg q) && \textbf{De Morgan's (2)}
\end{aligned}
$$

Playposit Question

Which of the following is the complete simplification of $\neg(\neg p \wedge (p \vee q))$?

A. $\neg p \vee \neg(p \vee q)$

B. $p \vee \neg(p \vee q)$

C. $p \vee (\neg p \wedge \neg q)$

D. $p \vee (p \wedge q)$

Common Logical Equivalences

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

(a)	$p \wedge p \equiv p$, $p \vee p \equiv p$	Idempotent Laws
(b)	$p \vee \mathbf{T} \equiv \mathbf{T}$, $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination Laws
(c)	$p \wedge \mathbf{T} \equiv p$, $p \vee \mathbf{F} \equiv p$	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)	$p \vee \neg p \equiv \mathbf{T}$, $p \wedge \neg p \equiv \mathbf{F}$	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 $(p \wedge q) \rightarrow p$ is a *tautology*? (1)
  - Using Truth tables, we see:

p	q	$(p \wedge q)$	$(p \wedge q) \rightarrow p$
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 $(p \wedge q) \rightarrow p$ is a *tautology*? (2)
  - By application of logical equivalences

$(p \wedge q) \rightarrow p \equiv p \rightarrow (q \rightarrow p)$	**Table 3 (k)**
$\equiv q \rightarrow (p \rightarrow p)$	**Table 3 (p)**
$\equiv q \rightarrow \mathrm{T}$	**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 games is an integer

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

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

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

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**

Playposit Question

Which equivalences from table 3 are used in the following:

$$
\begin{aligned}
& \neg(p \to q) \to \neg q \\
\equiv \quad & (p \wedge \neg q) \to \neg q \\
\equiv \quad & (p \to \neg q) \vee (\neg q \to \neg q) \\
\equiv \quad & (p \to \neg q) \vee T \\
\equiv \quad & T
\end{aligned}
$$

Table III: Still More Equivalences (adding -

(a)	$p \to q \equiv \neg p \vee q$
(b)	$p \to q \equiv \neg q \to \neg p$
(c)	$\mathbf{T} \to p \equiv p$
(d)	$p \to \mathbf{F} \equiv \neg p$
(e)	$p \to p \equiv \mathbf{T}$
(f)	$p \to q \equiv (p \wedge \neg q) \to \mathbf{F}$
(g)	$\neg p \to q \equiv p \vee q$
(h)	$\neg(p \to q) \equiv p \wedge \neg q$
(i)	$\neg(p \to \neg q) \equiv p \wedge q$
(j)	$(p \to q) \vee (q \to p) \equiv \mathbf{T}$
(k)	$(p \wedge q) \to r \equiv p \to (q \to r)$
(l)	$(p \wedge q) \to r \equiv (p \to r) \vee (q \to r)$

- (a)
- (c)
- (e)
- (h)
- (k)
- (l)