VANDERBILT UNIVERSITY School of Engineering

Discrete Structures

CS 2212

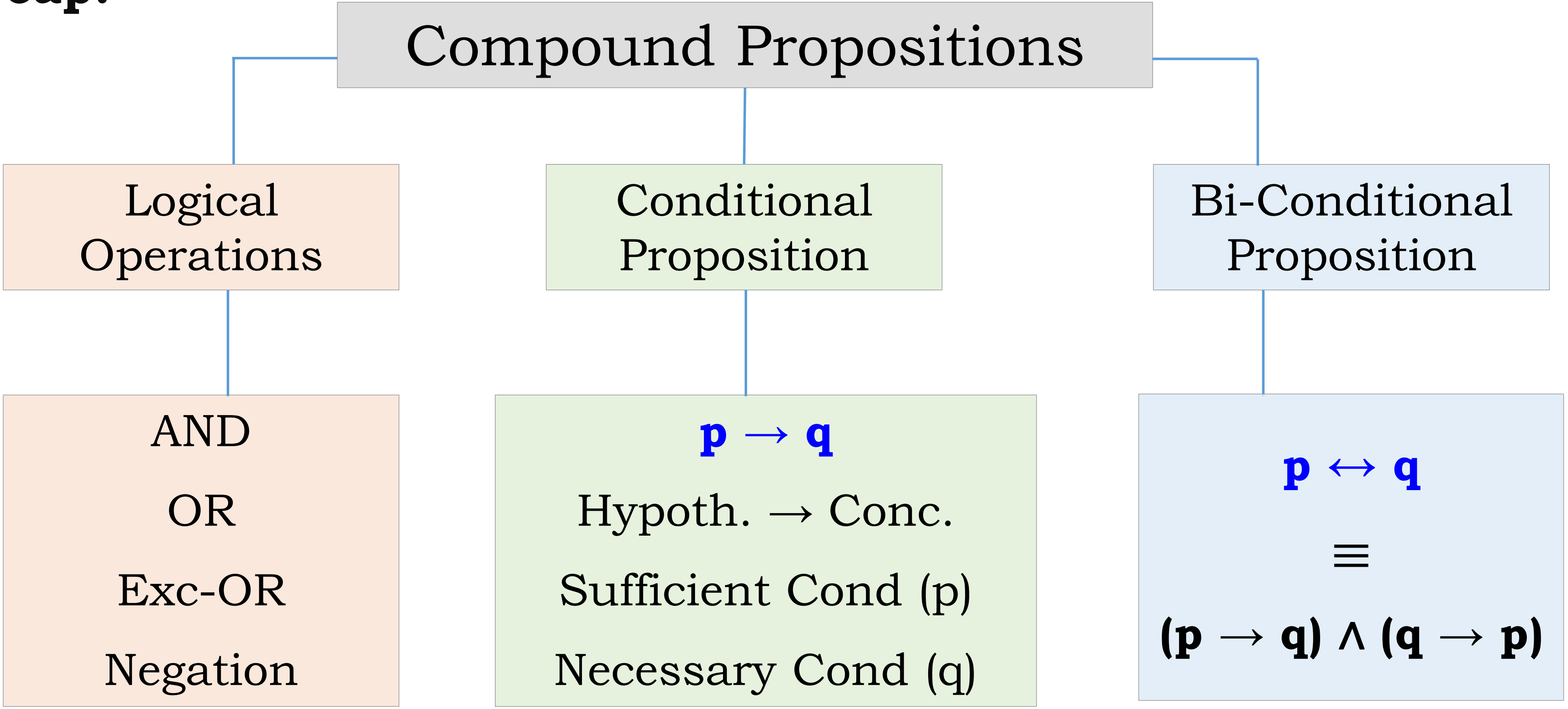
(Fall 2020)

3 – Logic

Reminder and Recap …

Reminder: **ZyBook Assig. 1A** and **1B** due **Sep. 06** (11:59 PM)

Recap:



Compound Propositions

Logical Operations

AND

OR

Exc-OR

Negation

Conditional Proposition

$\mathbf{p \rightarrow q}$

Hypoth. → Conc.

Sufficient Cond (p)

Necessary Cond (q)

Bi-Conditional Proposition

$\mathbf{p \leftrightarrow q}$

$\equiv$

$\mathbf{(p \rightarrow q) \land (q \rightarrow p)}$

Recap from last time

Conditional:	$\mathbf{p \to q}$
Converse:	$q \to p$
Contrapositive:	$\neg q \to \neg p$
Inverse:	$\neg p \to \neg q$

Tautology

Example: $\mathbf{p \lor \neg p}$

Contradiction

Example: $\mathbf{p \land \neg p}$

We also discussed when two statements are **equivalent**?

(If they always have same truth values)

Laws of Propositional Logic

- So far, we have used **truth tables** to show equivalence between statements.
- Not the easiest way if we have more complex propositions.
- Can there be another way?

In a complex statement, **substitute parts with equivalent statements** until we get the desired statement

Laws of Propositional Logic

It only makes sense to prove these simple but useful equivalences once, and then **re-use** them whenever they appear in complicated statements.

Lets see some useful equivalences that are often termed as laws.

Logical Equivalence

Consider the following statement with AND operation.

$$\underbrace{\text{Rachel has a cell phone}}_{p}\ \ \underset{\wedge}{\textbf{and}}\ \ \underbrace{\text{she has a laptop.}}_{q}$$

What is the negation of the following statement?

$$\neg\,(p \wedge q)\ ?$$

Logical Equivalence

Rachel has a cell phone **and** she has a laptop.
p $\land$ **q**

- For this statement to be true, ***both*** p and q must be **true**.
- Thus, if ***any*** of the p, q is **false**, the statement becomes false.
- So, the negation of the above statement is,

Rachel does not have a cell phone **or** she does not have a laptop.

$\neg$ **p** $\lor$ $\neg$ **q**

Logical Equivalence

De Morgan's Laws:

not (p and q) is equivalent to **(not p) or (not q)**

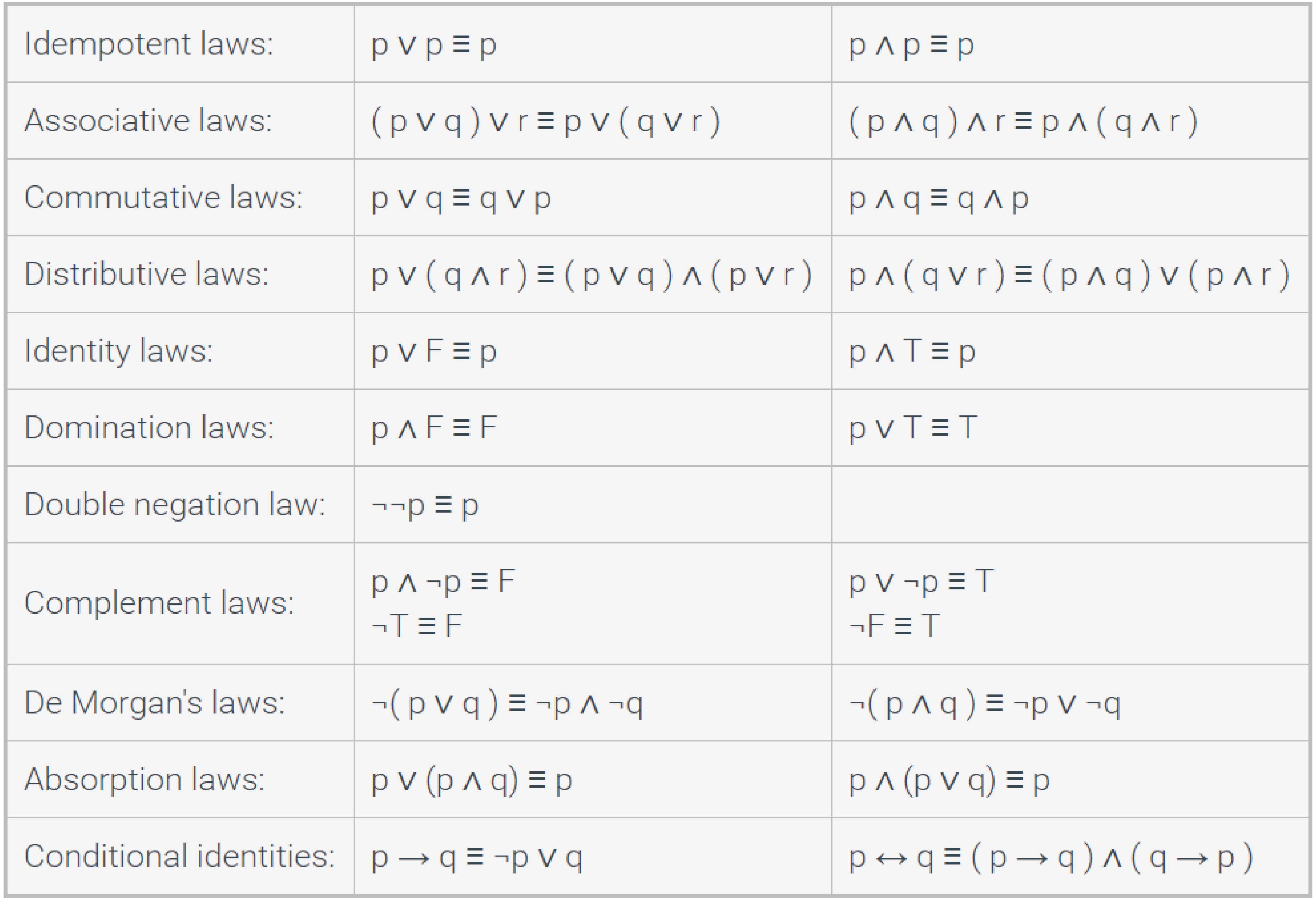
$$\neg (p \wedge q) \equiv \neg p \vee \neg q$$

not (p or q) is equivalent to **(not p) and (not q)**

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

(verify using truth tables)

Laws of Proportional Logic



Idempotent laws:	$p \vee p \equiv p$	$p \wedge p \equiv p$
Associative laws:	$(p \vee q) \vee r \equiv p \vee (q \vee r)$	$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
Commutative laws:	$p \vee q \equiv q \vee p$	$p \wedge q \equiv q \wedge p$
Distributive laws:	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
Identity laws:	$p \vee F \equiv p$	$p \wedge T \equiv p$
Domination laws:	$p \wedge F \equiv F$	$p \vee T \equiv T$
Double negation law:	$\neg\neg p \equiv p$	
Complement laws:	$p \wedge \neg p \equiv F$ $\neg T \equiv F$	$p \vee \neg p \equiv T$ $\neg F \equiv T$
De Morgan's laws:	$\neg(p \vee q) \equiv \neg p \wedge \neg q$	$\neg(p \wedge q) \equiv \neg p \vee \neg q$
Absorption laws:	$p \vee (p \wedge q) \equiv p$	$p \wedge (p \vee q) \equiv p$
Conditional identities:	$p \rightarrow q \equiv \neg p \vee q$	$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$

Equivalence through Proportional Logic Laws

Regarding all these equivalence rules:

- They will be handed out for exams.
- You don't need to memorize them.

Some Important Points:

- Know the Objective
- Be mindful of the order of terms
- Justify each step
- There could be many ways

So, what’s the trick here?

- Practice, and
- Some more practice
- Keep in mind the objective.

Equivalence through Proportional Logic Laws

Show that $\neg(p \lor (\neg p \land q))$ and $\neg p \land \neg q$ are logically equivalent

LHS:	$\neg(p \lor (\neg p \land q))$	
	$\equiv \neg p \land \neg(\neg p \land q)$	by the second De Morgan law
	$\equiv \neg p \land [\neg(\neg p) \lor \neg q]$	by the first De Morgan law
	$\equiv \neg p \land (p \lor \neg q)$	by the double negation law
	$\equiv (\neg p \land p) \lor (\neg p \land \neg q)$	by the second distributive law
	$\equiv \mathbf{F} \lor (\neg p \land \neg q)$	because $\neg p \land p \equiv \mathbf{F}$
	$\equiv (\neg p \land \neg q) \lor \mathbf{F}$	by the commutative law for disjunction
	$\equiv \neg p \land \neg q$	by the identity law for $\mathbf{F}$

So far, we have been dealing with propositions, which have *fixed* truth values.

What if our statements have *variable* truth values?

For this, we need:

Predicates

Predicate Logic

Often times mathematical statements involve **variables**, and truth value of the statement depends on **particular values** of variables.

Example: **x is an odd integer**

Is this a proposition?

Example: **if** (x > 10), **then** print “Hello”

Is our code going to print 10?

There is **no definite** true or false here.

Thus, propositional logic cannot deal with such situations.

We need to **“generalize”** the notion of propositions.

Predicate Logic

Predicate: It is a logical statement that contains **variables**, and the truth value of the statement depends on the particular values of variables.

P(x): ***x* is an odd integer** **(Predicate)**

P(5): **5** is an odd integer (True)

P(4): **4** is an odd integer (False)

These statements have definite truth values, so they are **propositions**

A predicate with its variables instantiated is a **proposition**.

P(1), P(2), P(3), …

Each of the above has a true or false value and is a proposition

Predicate Logic

P(x): x is an odd integer **(Predicate)**

- ***x*** is our **variable** here. So can x have ***any*** value ???
- How about x = banana?

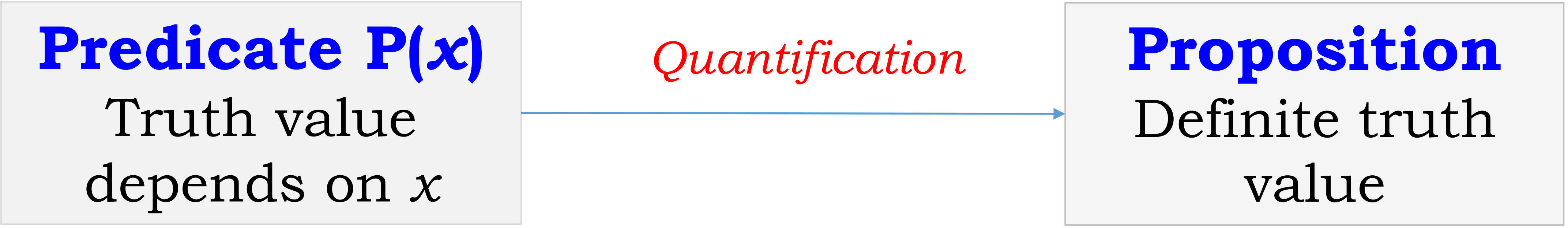
P(banana): banana is an odd integer

- We must define a **set** or **Domain** which contains all possible values of variables. In other words, our variables can have values only from that domain or set.
- Such a domain is typically called **Domain of Discourse**

P(x): x is an odd integer

Domain: Set of integers ($\mathbb{Z}$)

Quantifying Predicates



Predicate P(x)

Truth value depends on x

Quantification

Proposition

Definite truth value

There are different ways of doing it.

Here, we see three different ways.

Quantifying Predicates

1. Select a specific value of x from the domain.

   Proposition: **P(x) for that specific value of x**

Example:

Predicate:	**Person is older than 30 years.**
Variable (x):	Person
Domain:	Persons in this class

Quantifier:	**Specific value of person x = Waseem**
Proposition:	**Waseem is older than 30 years.**
Truth Value:	True

Quantifying Predicates (Summarizing)

2. For all values of *x* in the domain

Proposition: **For all *x*, P(*x*)**

True: if P(*x*) is *true* for *every value* of *x*.

False: if there is *at least one value* of *x* for which P(*x*) is *false*.

Example:

Predicate:	**Person is older than 30 years.**
Variable (*x*):	Person
Domain:	Persons in this class

Quantifier:	**For all values of *x***
Proposition:	**All persons in the class are older than 30 years.**
Truth Value:	False

Quantifying Predicates

3. There exists some *x* in the domain

Proposition: **For some *x*, P(*x*)**

True: if there is *at least one value* of *x* for which P(*x*) is *true*.

False: if P(*x*) is *false* for all values of *x*.

Example:

Predicate:	**Person is older than 30 years.**
Variable:	Person
Domain:	Persons in this class

Quantifier:	**There exists some *x***
Proposition:	**There is a person in the class are older than 30 years.**
Truth Value:	True

Quantifying Predicates

Predicate P(*x*)
Truth value depends on *x*

Quantification →

Proposition
Definite truth value

Quantification:

1. Select a specific value of *x* from domain.
   Proposition: **P(*x*) for that specific value of *x***

2. For all values of x in domain
   Proposition: **For all *x*, P(*x*)**
   True: if P(*x*) is *true* for *every value* of *x*.
   False: if there is *at least one value* of *x* for which P(*x*) is *false*.

3. There exists an x in domain
   Proposition: **For some *x*, P(*x*)**
   True: if there is *at least one value* of *x* for which P(*x*) is *true*.
   False: if P(*x*) is *false* for *all values* of *x*.

Quantifying Predicates

Example:

Predicate: **Person is older than 30 years.**
Variable: Person
Domain: Persons in this class

Quantifier	Proposition	Truth Value
Specific variable (Person = Waseem)	Waseem is older than 30 years	**True**
Specific variable (Person = *xyz*)	*xyz* is older than 30 years	**False**
For all	All persons in this class are older than 30 years	**False**
There exists	There is a person in the class older than 30 years	**True**

Quantifiers

Quantifier: For all

Name: **Universal** quantifier

Symbol: $\forall$

Proposition $\forall x \; P(x)$

Quantifier: There exists

Name: **Existential** quantifier

Symbol: $\exists$

Proposition $\exists x \; P(x)$

Quantifying Predicates

Example:

Predicate P(x): Person is older than 10 years.

Variable (x): Person

Domain: Students in this class

Quantifier	Proposition	Truth Value
For all ∀	$\forall x\ P(x)$	True
There exists ∃	$\exists x\ P(x)$	True

Quantifying Predicates

To **prove** universally quantified statements, we need to show that the statement is true for *all possible values* in the domain

Quantifying Predicates

To **disprove** a universally quantified statement, all we need to show is that there *exists at least one value* (case) for which the statement is not true. (also referred to as the **counterexample**).

Quantifying Predicates

Always be mindful of the **domain**

$\forall x\,(x^2 > x)$ (domain: x is a positive integer)

This statement is false.

Counterexample: $x = 1$

$\forall x\,(x^2 > x)$

(domain: x is a positive integer greater than 1)

This statement is true.

A question:

If a *universally* quantified statement is *false*, does that mean the *existentially* quantified statement is *always true*?

Quantified Statements

Think of $\forall x \, P(x)$ as a Conjunction

Domain = $\{ x_1, x_2, x_3, \ldots, x_n \}$

$\forall x \, P(x)$ is true means

($P(x_1)$ is true) **and** ($P(x_2)$ is true) **and** … ($P(x_n)$ is true)

$$\forall x \, P(x) = P(x_1) \land P(x_2) \land P(x_3) \land \ldots \land P(x_n)$$

Quantified Statements

Think of $\exists x\ P(x)$ as a Disjunction

Domain = $\{ x_1, x_2, x_3, \ldots, x_n \}$

$\exists x\ P(x)$ is true means

($P(x_1)$ is true) **or** ($P(x_2)$ is true) **or** … ($P(x_n)$ is true)

$$\forall x\ P(x) = P(x_1) \lor P(x_2) \lor P(x_3) \lor \ldots \lor P(x_n)$$

Quantified Statements

Example:

Let P(x) means x is prime and

let O(x) means x is odd.

Given the proposition, $\forall x\ (P(x) \rightarrow O(x))$, x is a positive integer

- what does it mean?
- Is it *true*?

Solution:

$\forall x\ (P(x) \rightarrow O(x))$ says that for every positive integer x, if x is prime then x is odd (we can word this in many ways).

This proposition is ***false*****.** We just need one counter example.

Letting x =2, which is prime and even.

Quantified Statements

Example:

Let $P(x)$ means x is prime and

let $O(x)$ means x is odd.

Given the proposition, $\exists x\ (P(x) \land \neg O(x))$,

- what does it mean?
- Is it *true*?

Solution:

$\exists x\ (P(x) \land \neg O(x))$, says that there is a positive integer x which is prime and even.

This proposition is ***true***.

Letting $x = 2$, which is prime and even.

Quantifiers and Precedence of Operations

The quantifiers $\forall$ and $\exists$ are applied **before** the logical operations ($\land$, $\lor$, $\rightarrow$, and $\leftrightarrow$) used for propositions.

Example:

What does $\forall x\, P(x) \land Q(x)$ mean?

$(\forall x\, P(x)) \land Q(x)$ or $\forall x\, (P(x) \land Q(x))$

✓ (Correct)

✗ (Wrong)

Logic and Predicates: Free and Bound Variables

- A variable x in the predicate $P(x)$ is called a **free variable** because the variable is free to take on **any** value in the domain.

- The variable x in the statement $\forall x\, P(x)$ or $\exists\, x\, P(x)$ is a **bound variable** because the variable is bound to the quantifier within the domain.

- A statement with **no free variables is a proposition** because the statement's truth value can be determined.

- The part of logical expression to which a quantifier is applied is called the **scope** of the quantifier, $\exists\, x\, \boxed{R(x)}$

scope

Logic and Predicates: Free and Bound Variables

Example:

$$(\forall x\, P(\underset{\substack{\uparrow\\ \text{Bound variable}}}{x})) \land Q(\underset{\substack{\uparrow\\ \text{Free variable}}}{x})$$

Not a proposition

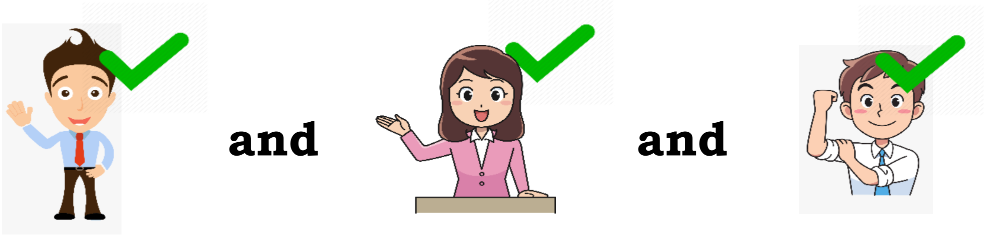
$$\forall x\, (P(x) \land Q(x))$$

x is a bound variable

Proposition

DeMorgans Law with Quantifiers

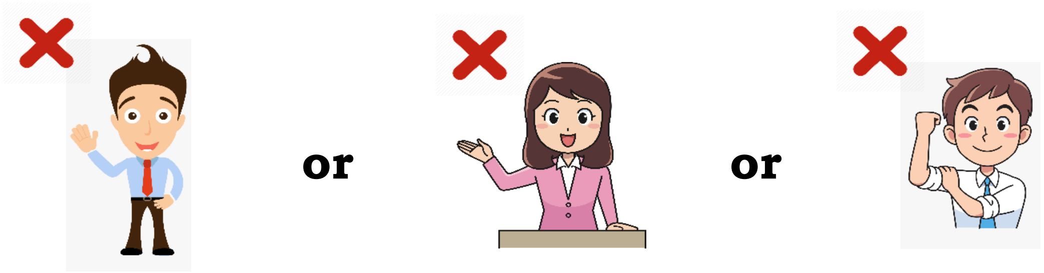
$\forall x\ P(x)$: Every student in this class has taken a calculus course.



What is the negation of the above statement, that is, $\neg\,(\forall x\ P(x))$?

It is not true that every student in this class has taken calculus.

There is at least one student that has not taken calculus.



$\exists x\ \neg P(x)$

$$\boxed{\begin{array}{c}\neg\,(\forall x\ P(x)) \\ \equiv \\ \exists x\ \neg P(x)\end{array}}$$

DeMorgans Law with Quantifiers

Domain of discourse = $\{x_1 \;, x_2 \,, \ldots \,, x_n\}$

$$\begin{array}{ccc}
\neg\forall x\, P(x) & \equiv & \neg(P(x_1) \wedge P(x_2) \wedge \ldots \wedge P(x_n)) \\
||| & & ||| \\
\exists x\, \neg P(x) & \equiv & \neg P(x_1) \vee \neg P(x_2) \vee \ldots \vee \neg P(x_n)
\end{array}$$

DeMorgans Law with Quantifiers

Similarly, we have

$$\begin{array}{ccc} \neg \exists x \, P(x) & \equiv & \neg(P(x_1) \vee P(x_2) \vee \ldots \vee P(x_n)) \\ ||| & & ||| \\ \forall x \, \neg P(x) & \equiv & \neg P(x_1) \wedge \neg P(x_2) \wedge \ldots \wedge \neg P(x_n)) \end{array}$$

Nested Quantifiers

- If a predicate has more than one variable, each variable **must be bound** by a separate quantifier.
- A logical expression with more than one quantifier that bind different variables in the same predicate is said to have **nested quantifiers.**
- **Example:**

$$\forall \mathrm{x}\ \exists \mathrm{y}\ \mathrm{P}(x, y)$$

Nested Quantifiers

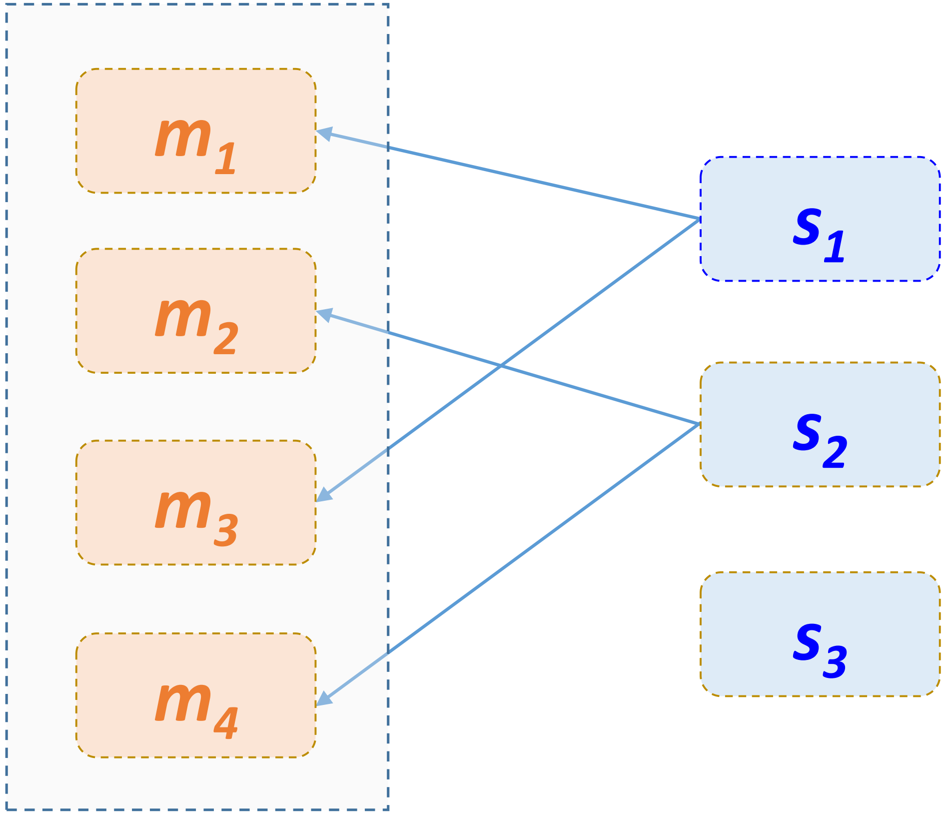
Lets see which are bound and free variables here.

Statement	Bound variable	Free variable
$\forall x\, \exists \mathrm{y}\, L(x, y)$	Both x and y	
$\forall x\, L(x, y)$	x	y
$\forall x\, \exists y\, L(x, y, z)$	Both x and y	z

Example - Nested Quantifiers

There are 4 machines in a plant and three supervisors.

Plant works if every machine is operated by at least 1 supervisor.

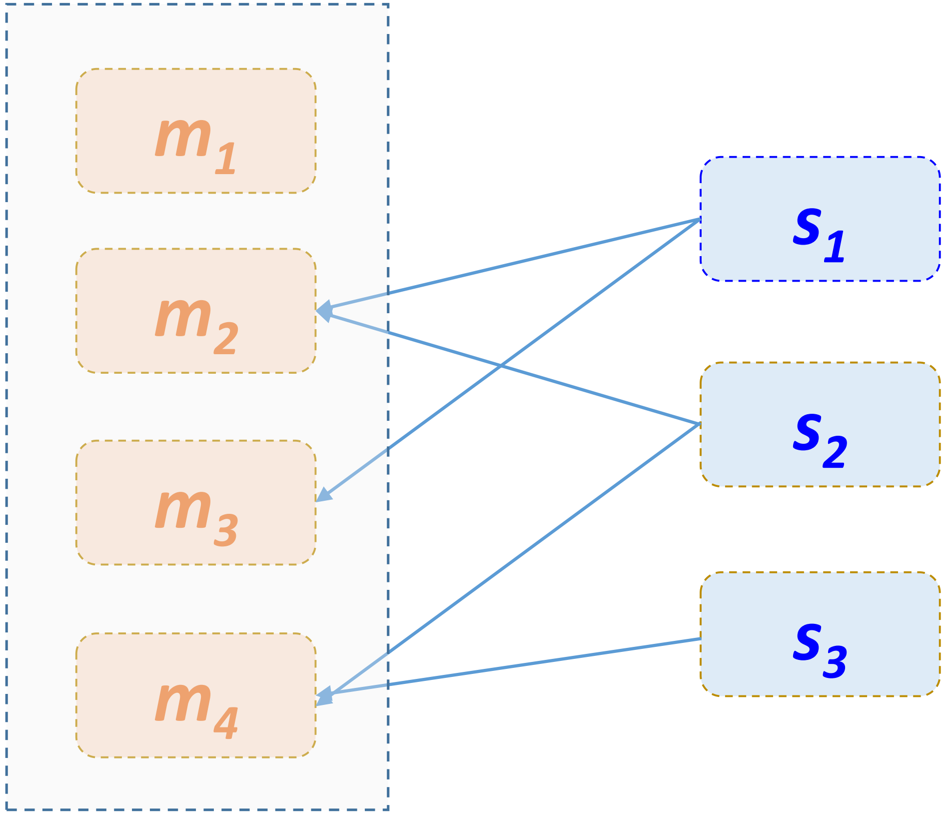


Does the plant work?

Example - Nested Quantifiers

There are 4 machines in a plant and three supervisors.

Plant works if every machine is operated by at least 1 supervisor.



Does the plant work now?

Example - Nested Quantifiers

Machine: m

Supervisor: s

$P(m,s)$: m is operated by s.

Plant works if every machine is operated by at least one supervisor.

For every machine, there is a supervisor that operates it.

$$\forall m\, \exists s\, P(m, s)$$

Example - Nested Quantifiers

Domain of m: $\{m_1, m_2, m_3, m_4\}$

Domain of s: $\{s_1, s_2, s_3\}$

$$\forall m \, \exists s \, P(m, s)$$

$$\equiv$$

$$(\exists s \, P(m_1,s)) \land (\exists s \, P(m_2,s)) \land (\exists s \, P(m_3, s)) \land (\exists s \, P(m_4,s))$$

Example - Nested Quantifiers

There are 4 students and 3 courses in a semester.

A new whiteboard is installed in a classroom if there is a student with A’s in all courses.

	Course 1	**Course 2**	**Course 3**
Student 1	A	A	C
Student 2	A	B	B
Student 3	A	B	B
Student 4	A	B	B

Do we have a new whiteboard?

Example - Nested Quantifiers

There are 4 students and 3 courses in a semester.

A new whiteboard is installed in a classroom if there is a student with A’s in all courses.

	Course 1	**Course 2**	**Course 3**
Student 1	A	A	C
Student 2	A	B	B
Student 3	A	A	A
Student 4	A	B	B

Do we have a new whiteboard now?

Example - Nested Quantifiers

Student: s

Course: c

$G(s,c)$: s scored A in c

There is a student with A's in all courses..

There exists some s for which $G(s,c)$ is true for all c.

$$\exists s \, \forall c \, G(s,c)$$

Example - Nested Quantifiers

Domain of s: $\{s_1, s_2, s_3, s_4\}$

Domain of c: $\{c_1, c_2, c_3\}$

$$\exists s\ \forall c\ \mathrm{G}(s,c)$$

$$\equiv$$

$$(\forall c\ \mathrm{G}(s_1,\ c)) \lor (\forall c\ \mathrm{G}(s_1,\ c)) \lor (\forall \mathrm{c}\ \mathrm{G}(\mathrm{s}_1,\ \mathrm{c})))$$

Nested Quantifiers Precedence

Operator	**Precedence**
$\forall$, $\exists$	1
$\neg$	2
$\wedge$ $\vee$	3 4
$\rightarrow$ $\leftrightarrow$	5 6

The quantifiers $\forall$ and $\exists$ have *higher* precedence than all the logical operators.

Nested Quantifiers Precedence

Predicate precedence with no presence of parentheses:

1.	$\forall, \exists$
2.	$\neg$
3. 4.	$\wedge, \vee$
5. 6.	$\rightarrow, \leftrightarrow$

Example:

$$\forall x \neg \exists y\, p(x, y) \rightarrow \forall x\, q(x)$$

$$\equiv (\forall x \neg \exists y\, p(x, y)) \rightarrow (\forall x\, q(x))$$

$$\equiv (\forall x \neg (\exists y\, p(x, y))) \rightarrow (\forall x\, q(x))$$

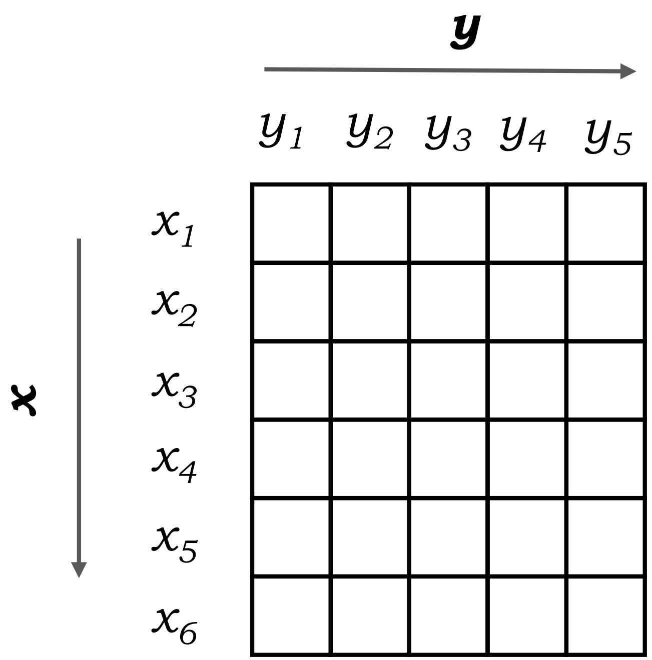
$$\equiv (\forall x\, (\neg (\exists y\, p(x, y)))) \rightarrow (\forall x\, q(x))$$

Nested Quantifiers

Two variable predicate: **P(x,y)**

Variable $\boldsymbol{x}$ = $\{x_1 , x_2 , \ldots , x_n\}$

Variable $\boldsymbol{y}$ = $\{y_1 , y_2 , \ldots , y_m\}$

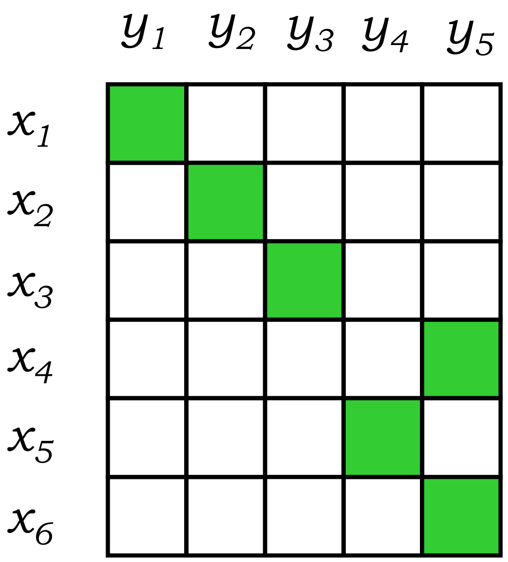


P(x,y)

$$\forall x\, \exists y\, \mathrm{P}(x, y)$$

It means that in **every row**, there should be at least one true value (green block). We don't care where this true value is in the row, but each row must contain one.

Example:



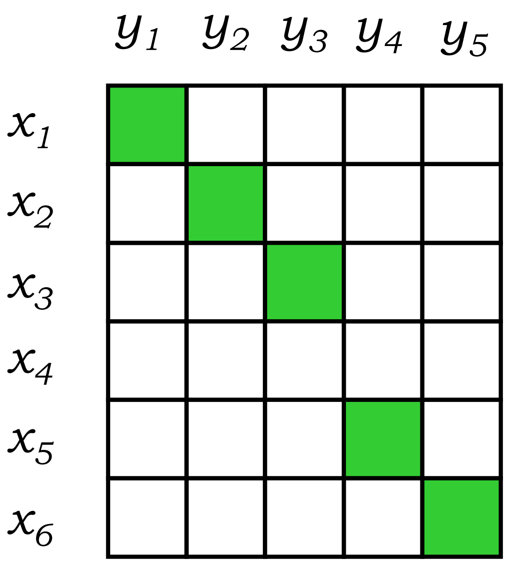
	y_1	y_2	y_3	y_4	y_5
x_1	🟩				
x_2		🟩			
x_3			🟩		
x_4					🟩
x_5				🟩	
x_6					🟩

$\forall x\, \exists y\, \mathrm{P}(x, y)$ is **true**

Nested Quantifiers

$\forall x \, \exists y \, P(x, y)$ is **false**

It means there is at least one **row** that **does not** have any true value.

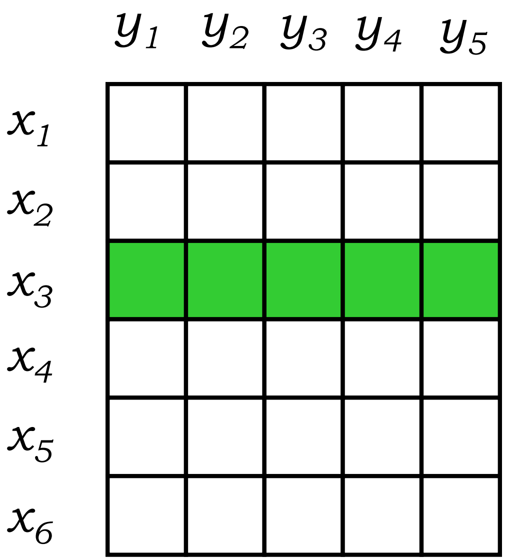


	y_1	y_2	y_3	y_4	y_5
x_1	🟩				
x_2		🟩			
x_3			🟩		
x_4					
x_5				🟩	
x_6					🟩

$\neg\forall x \, \exists y \, P(x, y)$ is **true**

$\exists x \, \forall y \, P(x, y)$

We are looking for a **row** with all true values



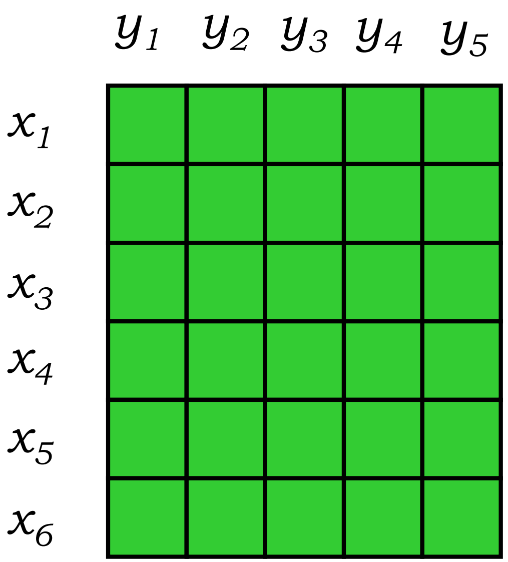
	y_1	y_2	y_3	y_4	y_5
x_1					
x_2					
x_3	🟩	🟩	🟩	🟩	🟩
x_4					
x_5					
x_6					

$\exists x \, \forall y \, P(x, y)$ is **true**

Nested Quantifiers

$\forall x \, \forall y \, P(x, y)$

All blocks should be true

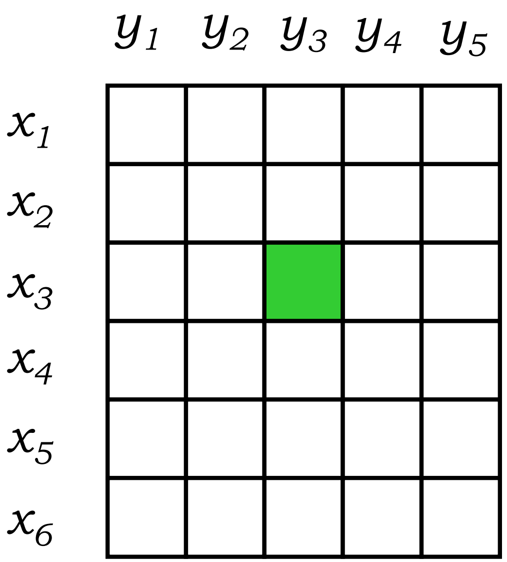


	y_1	y_2	y_3	y_4	y_5
x_1	🟩	🟩	🟩	🟩	🟩
x_2	🟩	🟩	🟩	🟩	🟩
x_3	🟩	🟩	🟩	🟩	🟩
x_4	🟩	🟩	🟩	🟩	🟩
x_5	🟩	🟩	🟩	🟩	🟩
x_6	🟩	🟩	🟩	🟩	🟩

$\forall x \, \forall y \, P(x, y)$ is **true**

$\exists x \, \exists y \, P(x, y)$

We are looking for a **at least one block** to be true.



	y_1	y_2	y_3	y_4	y_5
x_1					
x_2					
x_3			🟩		
x_4					
x_5					
x_6					

$\exists x \, \exists y \, P(x, y)$ is **true**

Nested Quantifiers

Statement	**When True?**	**When False?**
$\forall x \forall y P(x, y)$, $\forall y \forall x P(x, y)$	$P(x, y)$ is true for every pair x, y.	There is a pair x, y for which $P(x, y)$ is false.
$\forall x \exists y P(x, y)$	For every x there is a y for which $P(x, y)$ is true.	There is an x such that $P(x, y)$ is false for every y.
$\exists x \forall y P(x, y)$	There is an x for which $P(x, y)$ is true for every y.	For every x there is a y for which $P(x, y)$ is false.
$\exists x \exists y P(x, y)$, $\exists y \exists x P(x, y)$	There is a pair x, y for which $P(x, y)$ is true.	$P(x, y)$ is false for every pair x, y.

(Summary)