Discrete Mathematics & Probability Theory

Welcome to CS 70!

Discrete Mathematics & Probability Theory

Welcome to CS 70!

CS 70 is a math course.

Discrete Mathematics & Probability Theory

Welcome to CS 70!

CS 70 is a math course.

- Learn key ideas in math: proofs, induction/recursion, ...

Discrete Mathematics & Probability Theory

Welcome to CS 70!

CS 70 is a math course.

- Learn key ideas in math: proofs, induction/recursion, ...
- Practice “mathematical thinking”.

Discrete Mathematics & Probability Theory

Welcome to CS 70!

CS 70 is a math course.

- Learn key ideas in math: proofs, induction/recursion, ...
- Practice "mathematical thinking".

But CS 70 is also a EECS course.

Discrete Mathematics & Probability Theory

Welcome to CS 70!

CS 70 is a math course.

- Learn key ideas in math: proofs, induction/recursion, ...
- Practice “mathematical thinking”.

But CS 70 is also a EECS course.

- Graphs, modular arithmetic, probability, etc. all find immense applications in both EE and CS.

Discrete Mathematics & Probability Theory

Welcome to CS 70!

CS 70 is a math course.

- Learn key ideas in math: proofs, induction/recursion, ...
- Practice "mathematical thinking".

But CS 70 is also a EECS course.

- Graphs, modular arithmetic, probability, etc. all find immense applications in both EE and CS.
- Schedule jobs, encrypt communication, design reliable systems, learn from data (machine learning/AI), ...

Discrete Mathematics & Probability Theory

Welcome to CS 70!

CS 70 is a math course.

- Learn key ideas in math: proofs, induction/recursion, ...
- Practice "mathematical thinking".

But CS 70 is also a EECS course.

- Graphs, modular arithmetic, probability, etc. all find immense applications in both EE and CS.
- Schedule jobs, encrypt communication, design reliable systems, learn from data (machine learning/AI), ...

Mathematical rigor teaches you how to *think clearly*—don't be discouraged.

Sinho Chewi

- Just finished a B.S. in Engineering Mathematics & Statistics[1].

[1] Wow that's a mouthful—but it's one major.

Sinho Chewi

- Just finished a B.S. in Engineering Mathematics & Statistics[1].
- Upcoming fall: PhD in Applied Mathematics at MIT.

[1] Wow that's a mouthful—but it's one major.

Sinho Chewi

- Just finished a B.S. in Engineering Mathematics & Statistics[1].
- Upcoming fall: PhD in Applied Mathematics at MIT.
- I will cover discrete mathematics (first three weeks) and the last week of instruction.

[1] Wow that's a mouthful—but it's one major.

Sinho Chewi

- Just finished a B.S. in Engineering Mathematics & Statistics[1].
- Upcoming fall: PhD in Applied Mathematics at MIT.
- I will cover discrete mathematics (first three weeks) and the last week of instruction.
- I took CS 70 in Fall 2015.

[1] Wow that's a mouthful—but it's one major.

Sinho Chewi

- Just finished a B.S. in Engineering Mathematics & Statistics[1].
- Upcoming fall: PhD in Applied Mathematics at MIT.
- I will cover discrete mathematics (first three weeks) and the last week of instruction.
- I took CS 70 in Fall 2015.
- Teaching: TA for CS 70 (5 times) and EECS 126 (3 times).

[1] Wow that's a mouthful—but it's one major.

Sinho Chewi

- Just finished a B.S. in Engineering Mathematics & Statistics[1].
- Upcoming fall: PhD in Applied Mathematics at MIT.
- I will cover discrete mathematics (first three weeks) and the last week of instruction.
- I took CS 70 in Fall 2015.
- Teaching: TA for CS 70 (5 times) and EECS 126 (3 times).
- Email me anytime (or come to my OH).

[1] Wow that's a mouthful—but it's one major.

Vrettos Moulos

- He is a PhD student at UC Berkeley (out of town this week).

Vrettos Moulos

- He is a PhD student at UC Berkeley (out of town this week).
- His lectures (beginning after Midterm 1) will mostly cover probability.

Vrettos Moulos

- He is a PhD student at UC Berkeley (out of town this week).
- His lectures (beginning after Midterm 1) will mostly cover probability.
- Teaching: Instructor for CS 70, Summer 2017.

Vrettos Moulos

- He is a PhD student at UC Berkeley (out of town this week).
- His lectures (beginning after Midterm 1) will mostly cover probability.
- Teaching: Instructor for CS 70, Summer 2017.
- Meet him in his OH.

Introducing the TAs

They will teach you more than I ever will!

See course website for the full TA list with contact info.

Logistical Announcements

Website: `http://www.eecs70.org/`

Logistical Announcements

Website: `http://www.eecs70.org/`

Check Piazza for announcements and other communications.

Logistical Announcements

Website: `http://www.eecs70.org/`

Check Piazza for announcements and other communications.

Reminders:

- Homework 0 is due Wednesday, 6/20, 10 PM.

Logistical Announcements

Website: `http://www.eecs70.org/`

Check Piazza for announcements and other communications.

Reminders:

- Homework 0 is due Wednesday, 6/20, 10 PM.
- Homework 1 released today, due Friday, 6/22, 10 PM.

Logistical Announcements

Website: `http://www.eecs70.org/`

Check Piazza for announcements and other communications.

Reminders:

- Homework 0 is due Wednesday, 6/20, 10 PM.
- Homework 1 released today, due Friday, 6/22, 10 PM.
- Discussions start **today**.

Logistical Announcements

Website: `http://www.eecs70.org/`

Check Piazza for announcements and other communications.

Reminders:

- Homework 0 is due Wednesday, 6/20, 10 PM.
- Homework 1 released today, due Friday, 6/22, 10 PM.
- Discussions start **today**.

No more "grading options"—everyone does homeworks, two midterms, and final.

Logic Puzzles

True or False? Harry and Ron cannot both date Hermione. Harry will either date Ginny or Hermione. Ron will date Hermione. Therefore, Harry will date Ginny.

Logic Puzzles

True or False? Harry and Ron cannot both date Hermione. Harry will either date Ginny or Hermione. Ron will date Hermione. Therefore, Harry will date Ginny.

True or False? Either Iron Man or Captain America is being honest. Either Iron Man or Thor is being dishonest. Therefore, either Captain America is being honest or Thor is being dishonest.

Logic Puzzles

True or False? Harry and Ron cannot both date Hermione. Harry will either date Ginny or Hermione. Ron will date Hermione. Therefore, Harry will date Ginny.

True or False? Either Iron Man or Captain America is being honest. Either Iron Man or Thor is being dishonest. Therefore, either Captain America is being honest or Thor is being dishonest.

Can you program a computer figure these out?

Logic Puzzles

True or False? Harry and Ron cannot both date Hermione. Harry will either date Ginny or Hermione. Ron will date Hermione. Therefore, Harry will date Ginny.

True or False? Either Iron Man or Captain America is being honest. Either Iron Man or Thor is being dishonest. Therefore, either Captain America is being honest or Thor is being dishonest.

Can you program a computer figure these out? Programming requires a *language*.

Logic Puzzles

True or False? Harry and Ron cannot both date Hermione. Harry will either date Ginny or Hermione. Ron will date Hermione. Therefore, Harry will date Ginny.

True or False? Either Iron Man or Captain America is being honest. Either Iron Man or Thor is being dishonest. Therefore, either Captain America is being honest or Thor is being dishonest.

Can you program a computer figure these out? Programming requires a *language*.

Today we develop a formal language: **propositional logic**.

Logic: The Foundations of Mathematics

Mathematics has **axioms**: statements whose truth is asserted and not proven.

[2] This is called the *Axiom of Extensionality*.

[3] This is called *commutativity of addition*.

Logic: The Foundations of Mathematics

Mathematics has **axioms**: statements whose truth is asserted and not proven.

- Example: Axioms of set theory. If two sets contain exactly the same elements, they are the same set. [2]

---

[2] This is called the *Axiom of Extensionality*.

[3] This is called *commutativity of addition*.

Logic: The Foundations of Mathematics

Mathematics has **axioms**: statements whose truth is asserted and not proven.

- Example: Axioms of set theory. If two sets contain exactly the same elements, they are the same set. [2]
- Example: Axioms of real numbers. If $x, y \in \mathbb{R}$, then $x + y = y + x$. [3]

[2] This is called the *Axiom of Extensionality*.

[3] This is called *commutativity of addition*.

Logic: The Foundations of Mathematics

Mathematics has **axioms**: statements whose truth is asserted and not proven.

- Example: Axioms of set theory. If two sets contain exactly the same elements, they are the same set. [2]
- Example: Axioms of real numbers. If $x, y \in \mathbb{R}$, then $x + y = y + x$. [3]

A **proposition** is a sentence with an *unambiguous* truth value: either T (True) or F (False).

[2] This is called the *Axiom of Extensionality*.

[3] This is called *commutativity of addition*.

Logic: The Foundations of Mathematics

Mathematics has **axioms**: statements whose truth is asserted and not proven.

- Example: Axioms of set theory. If two sets contain exactly the same elements, they are the same set. [2]
- Example: Axioms of real numbers. If $x, y \in \mathbb{R}$, then $x + y = y + x$. [3]

A **proposition** is a sentence with an *unambiguous* truth value: either T (True) or F (False).

- Example: $2 + 2 = 4$.

[2] This is called the *Axiom of Extensionality*.

[3] This is called *commutativity of addition*.

Logic: The Foundations of Mathematics

Mathematics has **axioms**: statements whose truth is asserted and not proven.

- Example: Axioms of set theory. If two sets contain exactly the same elements, they are the same set. [2]
- Example: Axioms of real numbers. If $x, y \in \mathbb{R}$, then $x + y = y + x$. [3]

A **proposition** is a sentence with an *unambiguous* truth value: either T (True) or F (False).

- Example: $2 + 2 = 4$. True.

---

[2] This is called the *Axiom of Extensionality*.

[3] This is called *commutativity of addition*.

Logic: The Foundations of Mathematics

Mathematics has **axioms**: statements whose truth is asserted and not proven.

- Example: Axioms of set theory. If two sets contain exactly the same elements, they are the same set. [2]
- Example: Axioms of real numbers. If $x, y \in \mathbb{R}$, then $x + y = y + x$. [3]

A **proposition** is a sentence with an *unambiguous* truth value: either T (True) or F (False).

- Example: $2 + 2 = 4$. True.
- Example: Every integer is even.

---

[2] This is called the *Axiom of Extensionality*.

[3] This is called *commutativity of addition*.

Logic: The Foundations of Mathematics

Mathematics has **axioms**: statements whose truth is asserted and not proven.

- Example: Axioms of set theory. If two sets contain exactly the same elements, they are the same set. [2]
- Example: Axioms of real numbers. If $x, y \in \mathbb{R}$, then $x + y = y + x$. [3]

A **proposition** is a sentence with an *unambiguous* truth value: either T (True) or F (False).

- Example: $2 + 2 = 4$. True.
- Example: Every integer is even. False.

---

[2] This is called the *Axiom of Extensionality*.

[3] This is called *commutativity of addition*.

Logic: The Foundations of Mathematics

Mathematics has **axioms**: statements whose truth is asserted and not proven.

- Example: Axioms of set theory. If two sets contain exactly the same elements, they are the same set. [2]
- Example: Axioms of real numbers. If $x, y \in \mathbb{R}$, then $x + y = y + x$. [3]

A **proposition** is a sentence with an *unambiguous* truth value: either T (True) or F (False).

- Example: $2 + 2 = 4$. True.
- Example: Every integer is even. False.

Mathematics: Deduce the truth of propositions from the axioms.

---

[2] This is called the *Axiom of Extensionality*.

[3] This is called *commutativity of addition*.

Logic: The Foundations of Mathematics

Mathematics has **axioms**: statements whose truth is asserted and not proven.

- Example: Axioms of set theory. If two sets contain exactly the same elements, they are the same set. [2]
- Example: Axioms of real numbers. If $x, y \in \mathbb{R}$, then $x + y = y + x$. [3]

A **proposition** is a sentence with an *unambiguous* truth value: either T (True) or F (False).

- Example: $2 + 2 = 4$. True.
- Example: Every integer is even. False.

Mathematics: Deduce the truth of propositions from the axioms.

What does “deduce” mean?

---

[2] This is called the *Axiom of Extensionality*.

[3] This is called *commutativity of addition*.

Combining Propositions

We build new propositions from old propositions via **logical symbols**: (,), $\neg$, $\wedge$, $\vee$.

[4] The symbol resembles an "A", for "AND".

Combining Propositions

We build new propositions from old propositions via **logical symbols**: (,), $\neg$, $\wedge$, $\vee$.

Given propositions P and Q, ...

- **Negation** ($\neg$): $\neg P$ has the opposite truth value of P.

[4] The symbol resembles an "A", for "AND".

Combining Propositions

We build new propositions from old propositions via **logical symbols**: (,), $\neg$, $\wedge$, $\vee$.

Given propositions P and Q, ...

- **Negation** ($\neg$): $\neg P$ has the opposite truth value of P.
- **Conjunction** ($\wedge$, "AND")[4]: $P \wedge Q$ is True only if P and Q are *both* true.

[4] The symbol resembles an "A", for "AND".

Combining Propositions

We build new propositions from old propositions via **logical symbols**: (,), $\neg$, $\wedge$, $\vee$.

Given propositions P and Q, ...

- **Negation** ($\neg$): $\neg P$ has the opposite truth value of P.
- **Conjunction** ($\wedge$, "AND")[4]: $P \wedge Q$ is True only if P and Q are *both* true.
- **Disjunction** ($\vee$, "OR"): $P \vee Q$ is True if either P is True or Q is True *or both are true*.

---

[4] The symbol resembles an "A", for "AND".

Combining Propositions

We build new propositions from old propositions via **logical symbols**: (,), $\neg$, $\wedge$, $\vee$.

Given propositions P and Q, ...

- **Negation** ($\neg$): $\neg P$ has the opposite truth value of P.
- **Conjunction** ($\wedge$, "AND")[4]: $P \wedge Q$ is True only if P and Q are *both* true.
- **Disjunction** ($\vee$, "OR"): $P \vee Q$ is True if either P is True or Q is True *or both are true*. The "OR" is sometimes called an "*inclusive* OR".

[4] The symbol resembles an "A", for "AND".

Combining Propositions

We build new propositions from old propositions via **logical symbols**: (,), $\neg$, $\wedge$, $\vee$.

Given propositions P and Q, ...

- **Negation** ($\neg$): $\neg P$ has the opposite truth value of P.
- **Conjunction** ($\wedge$, "AND")[4]: $P \wedge Q$ is True only if P and Q are *both* true.
- **Disjunction** ($\vee$, "OR"): $P \vee Q$ is True if either P is True or Q is True *or both are true*. The "OR" is sometimes called an "*inclusive* OR".

Example: (CS 70 is fun) $\wedge$ (CS 70 is interesting).

[4] The symbol resembles an "A", for "AND".

Logical Operators Are Functions

We can think of $\neg$, $\wedge$, and $\vee$ as *Boolean functions*[5].

[5] A **Boolean function** is a function whose inputs and outputs are **Boolean** values, i.e., True or False.

Logical Operators Are Functions

We can think of $\neg$, $\wedge$, and $\vee$ as *Boolean functions*[5].

- $\neg$ is a **unary** (one argument) function, written $\neg : \{F, T\} \to \{F, T\}$.

[5] A **Boolean function** is a function whose inputs and outputs are **Boolean** values, i.e., True or False.

Logical Operators Are Functions

We can think of $\neg$, $\wedge$, and $\vee$ as *Boolean functions*[5].

- $\neg$ is a **unary** (one argument) function, written $\neg : \{F, T\} \to \{F, T\}$.
- $\wedge$ and $\vee$ are **binary** (two argument) functions, written $\wedge : \{F, T\} \times \{F, T\} \to \{F, T\}$.

[5] A **Boolean function** is a function whose inputs and outputs are **Boolean** values, i.e., True or False.

Logical Operators Are Functions

We can think of $\neg$, $\wedge$, and $\vee$ as *Boolean functions*[5].

- $\neg$ is a **unary** (one argument) function, written $\neg : \{F, T\} \to \{F, T\}$.
- $\wedge$ and $\vee$ are **binary** (two argument) functions, written $\wedge : \{F, T\} \times \{F, T\} \to \{F, T\}$.

We can specify functions by *specifying their outputs for each possible input*.

[5] A **Boolean function** is a function whose inputs and outputs are **Boolean** values, i.e., True or False.

Logical Operators Are Functions

We can think of $\neg$, $\wedge$, and $\vee$ as *Boolean functions*[5].

- $\neg$ is a **unary** (one argument) function, written $\neg : \{F, T\} \to \{F, T\}$.
- $\wedge$ and $\vee$ are **binary** (two argument) functions, written $\wedge : \{F, T\} \times \{F, T\} \to \{F, T\}$.

We can specify functions by *specifying their outputs for each possible input*.

P	$\neg P$
T	F
F	T

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

P	Q	$P \vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

[5] A **Boolean function** is a function whose inputs and outputs are **Boolean** values, i.e., True or False.

Logical Operators Are Functions

We can think of $\neg$, $\wedge$, and $\vee$ as *Boolean functions*[5].

- $\neg$ is a **unary** (one argument) function, written $\neg : \{F, T\} \to \{F, T\}$.
- $\wedge$ and $\vee$ are **binary** (two argument) functions, written $\wedge : \{F, T\} \times \{F, T\} \to \{F, T\}$.

We can specify functions by *specifying their outputs for each possible input*.

P	$\neg P$
T	F
F	T

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

P	Q	$P \vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

These are called **truth tables**.

[5] A **Boolean function** is a function whose inputs and outputs are **Boolean** values, i.e., True or False.

Propositional Equivalence

Suppose P and Q are propositions and we form sentences:
$P \wedge Q$, $P \vee Q$, $\neg P \wedge \neg Q$, ...

When is it true that two sentences *always* have the same truth value, *regardless* of the truth values of P and Q?

Propositional Equivalence

Suppose P and Q are propositions and we form sentences:
$P \wedge Q$, $P \vee Q$, $\neg P \wedge \neg Q$, $\ldots$

When is it true that two sentences *always* have the same truth value, *regardless* of the truth values of P and Q?

Consider $\neg(P \wedge Q)$ and $\neg P \vee \neg Q$.

Propositional Equivalence

Suppose P and Q are propositions and we form sentences:
$P \wedge Q$, $P \vee Q$, $\neg P \wedge \neg Q$, $\ldots$

When is it true that two sentences *always* have the same truth value, *regardless* of the truth values of P and Q?

Consider $\neg(P \wedge Q)$ and $\neg P \vee \neg Q$. Write out the truth tables.

Propositional Equivalence

Suppose P and Q are propositions and we form sentences:
$P \wedge Q$, $P \vee Q$, $\neg P \wedge \neg Q$, ...

When is it true that two sentences *always* have the same truth value, *regardless* of the truth values of P and Q?

Consider $\neg(P \wedge Q)$ and $\neg P \vee \neg Q$. Write out the truth tables.

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

P	Q	$\neg P$	$\neg Q$	$\neg P \vee \neg Q$
T	T	F	F	F
T	F	F	T	T
F	T	T	F	T
F	F	T	T	T

Propositional Equivalence

Suppose P and Q are propositions and we form sentences:
$P \wedge Q$, $P \vee Q$, $\neg P \wedge \neg Q$, $\ldots$

When is it true that two sentences *always* have the same truth value, *regardless* of the truth values of P and Q?

Consider $\neg(P \wedge Q)$ and $\neg P \vee \neg Q$. Write out the truth tables.

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

P	Q	$\neg P$	$\neg Q$	$\neg P \vee \neg Q$
T	T	F	F	F
T	F	F	T	T
F	T	T	F	T
F	F	T	T	T

The final columns match.

Propositional Equivalence

Suppose P and Q are propositions and we form sentences:
$P \wedge Q$, $P \vee Q$, $\neg P \wedge \neg Q$, $\ldots$

When is it true that two sentences *always* have the same truth value, *regardless* of the truth values of P and Q?

Consider $\neg(P \wedge Q)$ and $\neg P \vee \neg Q$. Write out the truth tables.

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

P	Q	$\neg P$	$\neg Q$	$\neg P \vee \neg Q$
T	T	F	F	F
T	F	F	T	T
F	T	T	F	T
F	F	T	T	T

The final columns match. This is called **propositional equivalence**, denoted $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$.

Useful Propositional Equivalences

Prove these for yourself.

Distributive laws:

- $P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$
- $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$

Useful Propositional Equivalences

Prove these for yourself.

Distributive laws:

- $P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$
- $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$

Double negation:

- $\neg\neg P \equiv P$

Useful Propositional Equivalences

Prove these for yourself.

Distributive laws:

- $P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$
- $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$

Double negation:

- $\neg\neg P \equiv P$

De Morgan's Laws: (distribute and flip)

- $\neg(A \wedge B) \equiv \neg A \vee \neg B$
- $\neg(A \vee B) \equiv \neg A \wedge \neg B$

Useful Propositional Equivalences

Prove these for yourself.

Distributive laws:

- $P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$
- $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$

Double negation:

- $\neg\neg P \equiv P$

De Morgan's Laws: (distribute and flip)

- $\neg(A \wedge B) \equiv \neg A \vee \neg B$
- $\neg(A \vee B) \equiv \neg A \wedge \neg B$

And more...

Recap: Building a Language

What have we done so far?

Recap: Building a Language

What have we done so far?

- Introduced a new language!

Recap: Building a Language

What have we done so far?

- Introduced a new language!

  That is, a set of *symbols* with rules for combining them.

Recap: Building a Language

What have we done so far?

- Introduced a new language!
  That is, a set of *symbols* with rules for combining them.
- Given the language a meaning!

Recap: Building a Language

What have we done so far?

- Introduced a new language!
  That is, a set of *symbols* with rules for combining them.
- Given the language a meaning!
  The symbols $\neg$, $\wedge$, and $\vee$ are given logical *interpretations*.

Recap: Building a Language

What have we done so far?

- Introduced a new language!
  That is, a set of *symbols* with rules for combining them.
- Given the language a meaning!
  The symbols $\neg$, $\wedge$, and $\vee$ are given logical *interpretations*.
- Defined an *algorithm* for evaluating the truth of a sentence and determining if two propositions are equivalent!

Recap: Building a Language

What have we done so far?

- Introduced a new language!
  That is, a set of *symbols* with rules for combining them.
- Given the language a meaning!
  The symbols $\neg$, $\wedge$, and $\vee$ are given logical *interpretations*.
- Defined an *algorithm* for evaluating the truth of a sentence and determining if two propositions are equivalent!
  Write out a truth table.

Logic Puzzles (Revisited)

True or False? Harry and Ron cannot both date Hermione. Harry will either date Ginny or Hermione. Ron will date Hermione. Therefore, Harry will date Ginny.

Logic Puzzles (Revisited)

True or False? Harry and Ron cannot both date Hermione. Harry will either date Ginny or Hermione. Ron will date Hermione. Therefore, Harry will date Ginny.

Translate to propositional logic:

- *HG* means “Harry dates Ginny”;
- *HH* means “Harry dates Hermione”;
- *RH* means “Ron dates Hermione”.

Logic Puzzles (Revisited)

True or False? Harry and Ron cannot both date Hermione. Harry will either date Ginny or Hermione. Ron will date Hermione. Therefore, Harry will date Ginny.

Translate to propositional logic:

- HG means "Harry dates Ginny";
- HH means "Harry dates Hermione";
- RH means "Ron dates Hermione".

The statements are:

- $\neg(HH \wedge RH)$.

Logic Puzzles (Revisited)

True or False? Harry and Ron cannot both date Hermione. Harry will either date Ginny or Hermione. Ron will date Hermione. Therefore, Harry will date Ginny.

Translate to propositional logic:

- HG means "Harry dates Ginny";
- HH means "Harry dates Hermione";
- RH means "Ron dates Hermione".

The statements are:

- $\neg(HH \wedge RH)$.
- $HG \vee HH$.

Logic Puzzles (Revisited)

True or False? Harry and Ron cannot both date Hermione. Harry will either date Ginny or Hermione. Ron will date Hermione. Therefore, Harry will date Ginny.

Translate to propositional logic:

- HG means "Harry dates Ginny";
- HH means "Harry dates Hermione";
- RH means "Ron dates Hermione".

The statements are:

- $\neg(HH \wedge RH)$.
- $HG \vee HH$.
- RH.

Solution to the Logic Puzzle, I

We translated the logic puzzle to the three propositions $\neg(HH \wedge RH)$, $HG \vee HH$, and RH.

Solution to the Logic Puzzle, I

We translated the logic puzzle to the three propositions $\neg(HH \wedge RH)$, $HG \vee HH$, and RH.

Approach 1: Use a truth table!

Solution to the Logic Puzzle, I

We translated the logic puzzle to the three propositions $\neg(HH \wedge RH)$, $HG \vee HH$, and RH.

Approach 1: Use a truth table!

HG	HH	RH	$\neg(HH \wedge RH)$	$HG \vee HH$	RH
T	T	T	F	T	T
T	T	F	T	T	F
T	F	T	T	T	T
T	F	F	T	T	F
F	T	T	F	T	T
F	T	F	T	T	F
F	F	T	T	F	T
F	F	F	T	F	F

Solution to the Logic Puzzle, I

We translated the logic puzzle to the three propositions $\neg(HH \wedge RH)$, $HG \vee HH$, and RH.

Approach 1: Use a truth table!

HG	HH	RH	$\neg(HH \wedge RH)$	$HG \vee HH$	RH
T	T	T	F	T	T
T	T	F	T	T	F
T	F	T	T	T	T
T	F	F	T	T	F
F	T	T	F	T	T
F	T	F	T	T	F
F	F	T	T	F	T
F	F	F	T	F	F

There is only one satisfying assignment, and in this assignment, $HG = T$.

Solution to the Logic Puzzle, I

We translated the logic puzzle to the three propositions $\neg(HH \wedge RH)$, $HG \vee HH$, and RH.

Approach 1: Use a truth table!

HG	HH	RH	$\neg(HH \wedge RH)$	$HG \vee HH$	RH
T	T	T	F	T	T
T	T	F	T	T	F
T	F	T	T	T	T
T	F	F	T	T	F
F	T	T	F	T	T
F	T	F	T	T	F
F	F	T	T	F	T
F	F	F	T	F	F

There is only one satisfying assignment, and in this assignment, $HG = T$. Harry must date Ginny.

Solution to the Logic Puzzle, II

Propositions: $\neg(HH \wedge RH)$, $HG \vee HH$, RH.

[6] This is a major unsolved question in computer science, called the $P = NP$ question. If you solve it, you win $1000000. Take CS 170 to learn more.

Solution to the Logic Puzzle, II

Propositions: $\neg(HH \wedge RH)$, $HG \vee HH$, RH.

Notes on Approach 1.

The truth table is *large*: for n variables, there are 2^n entries in the truth table.

[6] This is a major unsolved question in computer science, called the P $=$ NP question. If you solve it, you win \$1000000. Take CS 170 to learn more.

Solution to the Logic Puzzle, II

Propositions: $\neg(HH \wedge RH)$, $HG \vee HH$, RH.

Notes on Approach 1.

The truth table is *large*: for n variables, there are 2^n entries in the truth table. Can we find satisfying assignments *faster*? [6]

[6] This is a major unsolved question in computer science, called the $\mathrm{P} = \mathrm{NP}$ question. If you solve it, you win $1000000. Take CS 170 to learn more.

Solution to the Logic Puzzle, II

Propositions: $\neg(HH \wedge RH)$, $HG \vee HH$, RH.

Notes on Approach 1.

The truth table is *large*: for n variables, there are 2^n entries in the truth table. Can we find satisfying assignments *faster*? [6]

Approach 2: Use inference rules.

[6] This is a major unsolved question in computer science, called the $\mathrm{P} = \mathrm{NP}$ question. If you solve it, you win \$1000000. Take CS 170 to learn more.

Solution to the Logic Puzzle, II

Propositions: $\neg(HH \wedge RH)$, $HG \vee HH$, RH.

Notes on Approach 1.

The truth table is *large*: for n variables, there are 2^n entries in the truth table. Can we find satisfying assignments *faster*? [6]

Approach 2: Use inference rules.

- Set RH to be True.

[6] This is a major unsolved question in computer science, called the P $=$ NP question. If you solve it, you win $1000000. Take CS 170 to learn more.

Solution to the Logic Puzzle, II

Propositions: $\neg(HH \wedge RH)$, $HG \vee HH$, RH.

Notes on Approach 1.

The truth table is *large*: for n variables, there are 2^n entries in the truth table. Can we find satisfying assignments *faster*? [6]

Approach 2: Use inference rules.

- Set RH to be True.
- Then, $\neg(HH \wedge RH)$ becomes $\neg(HH \wedge T) \equiv \neg HH$.

[6] This is a major unsolved question in computer science, called the P $=$ NP question. If you solve it, you win $1000000. Take CS 170 to learn more.

Solution to the Logic Puzzle, II

Propositions: $\neg(HH \wedge RH)$, $HG \vee HH$, RH.

Notes on Approach 1.

The truth table is *large*: for n variables, there are 2^n entries in the truth table. Can we find satisfying assignments *faster*? [6]

Approach 2: Use inference rules.

- Set RH to be True.
- Then, $\neg(HH \wedge RH)$ becomes $\neg(HH \wedge T) \equiv \neg HH$.
- Set HH to be False.

---

[6] This is a major unsolved question in computer science, called the $\mathrm{P} = \mathrm{NP}$ question. If you solve it, you win $1000000. Take CS 170 to learn more.

Solution to the Logic Puzzle, II

Propositions: $\neg(HH \wedge RH)$, $HG \vee HH$, RH.

Notes on Approach 1.

The truth table is *large*: for n variables, there are 2^n entries in the truth table. Can we find satisfying assignments *faster*? [6]

Approach 2: Use inference rules.

- Set RH to be True.
- Then, $\neg(HH \wedge RH)$ becomes $\neg(HH \wedge T) \equiv \neg HH$.
- Set HH to be False.
- Then, $HG \vee HH$ becomes $HG \vee F \equiv HG$.

[6] This is a major unsolved question in computer science, called the $P = NP$ question. If you solve it, you win \$1000000. Take CS 170 to learn more.

Solution to the Logic Puzzle, II

Propositions: $\neg(HH \wedge RH)$, $HG \vee HH$, RH.

Notes on Approach 1.

The truth table is *large*: for n variables, there are 2^n entries in the truth table. Can we find satisfying assignments *faster*? [6]

Approach 2: Use inference rules.

- Set RH to be True.
- Then, $\neg(HH \wedge RH)$ becomes $\neg(HH \wedge T) \equiv \neg HH$.
- Set HH to be False.
- Then, $HG \vee HH$ becomes $HG \vee F \equiv HG$.
- Set HG to be True.

[6] This is a major unsolved question in computer science, called the $\mathrm{P} = \mathrm{NP}$ question. If you solve it, you win \$1000000. Take CS 170 to learn more.

Solution to the Logic Puzzle, II

Propositions: $\neg(HH \wedge RH)$, $HG \vee HH$, RH.

Notes on Approach 1.

The truth table is *large*: for n variables, there are 2^n entries in the truth table. Can we find satisfying assignments *faster*? [6]

Approach 2: Use inference rules.

- Set RH to be True.
- Then, $\neg(HH \wedge RH)$ becomes $\neg(HH \wedge T) \equiv \neg HH$.
- Set HH to be False.
- Then, $HG \vee HH$ becomes $HG \vee F \equiv HG$.
- Set HG to be True. Harry must date Ginny.

[6] This is a major unsolved question in computer science, called the $P = NP$ question. If you solve it, you win $1000000. Take CS 170 to learn more.

Solution to the Logic Puzzle, II

Propositions: $\neg(HH \wedge RH)$, $HG \vee HH$, RH.

Notes on Approach 1.
The truth table is *large*: for n variables, there are 2^n entries in the truth table. Can we find satisfying assignments *faster*? [6]

Approach 2: Use inference rules.

- Set RH to be True.
- Then, $\neg(HH \wedge RH)$ becomes $\neg(HH \wedge T) \equiv \neg HH$.
- Set HH to be False.
- Then, $HG \vee HH$ becomes $HG \vee F \equiv HG$.
- Set HG to be True. Harry must date Ginny.

English: Ron dates Hermione, so Harry does not.

[6] This is a major unsolved question in computer science, called the $P = NP$ question. If you solve it, you win $1000000. Take CS 170 to learn more.

Solution to the Logic Puzzle, II

Propositions: $\neg(HH \wedge RH)$, $HG \vee HH$, RH.

Notes on Approach 1.

The truth table is *large*: for n variables, there are 2^n entries in the truth table. Can we find satisfying assignments *faster*? [6]

Approach 2: Use inference rules.

- Set RH to be True.
- Then, $\neg(HH \wedge RH)$ becomes $\neg(HH \wedge T) \equiv \neg HH$.
- Set HH to be False.
- Then, $HG \vee HH$ becomes $HG \vee F \equiv HG$.
- Set HG to be True. Harry must date Ginny.

English: Ron dates Hermione, so Harry does not. Since Harry dates either Ginny or Hermione, then Harry must date Ginny.

---

[6] This is a major unsolved question in computer science, called the $P = NP$ question. If you solve it, you win $1000000. Take CS 170 to learn more.

Replicating Truth Tables

Our language is already fully expressive: given any truth table, we can write an equivalent sentence using only $\neg$, $\wedge$, and $\vee$.

Replicating Truth Tables

Our language is already fully expressive: given any truth table, we can write an equivalent sentence using only $\neg$, $\wedge$, and $\vee$.

Example: XOR (“exclusive OR”).

P	Q	$P \oplus Q$
T	T	F
T	F	T
F	T	T
F	F	F

Replicating Truth Tables

Our language is already fully expressive: given any truth table, we can write an equivalent sentence using only $\neg$, $\wedge$, and $\vee$.

Example: XOR ("exclusive OR").

P	Q	$P \oplus Q$
T	T	F
T	F	T
F	T	T
F	F	F

- Represent the T rows using "AND".

Replicating Truth Tables

Our language is already fully expressive: given any truth table, we can write an equivalent sentence using only $\neg$, $\wedge$, and $\vee$.

Example: XOR ("exclusive OR").

P	Q	$P \oplus Q$
T	T	F
T	F	T
F	T	T
F	F	F

- Represent the T rows using "AND".
  - $P = T$ and $Q = F$ becomes $P \wedge \neg Q$.

Replicating Truth Tables

Our language is already fully expressive: given any truth table, we can write an equivalent sentence using only $\neg$, $\wedge$, and $\vee$.

Example: XOR ("exclusive OR").

P	Q	$P \oplus Q$
T	T	F
T	F	T
F	T	T
F	F	F

- Represent the T rows using "AND".
  - $P = T$ and $Q = F$ becomes $P \wedge \neg Q$.
  - $P = F$ and $Q = T$ becomes $\neg P \wedge Q$.

Replicating Truth Tables

Our language is already fully expressive: given any truth table, we can write an equivalent sentence using only $\neg$, $\wedge$, and $\vee$.

Example: XOR (“exclusive OR”).

P	Q	$P \oplus Q$
T	T	F
T	F	T
F	T	T
F	F	F

- Represent the T rows using “AND”.
  - $P = T$ and $Q = F$ becomes $P \wedge \neg Q$.
  - $P = F$ and $Q = T$ becomes $\neg P \wedge Q$.
- Put the rows together with “OR”.

Replicating Truth Tables

Our language is already fully expressive: given any truth table, we can write an equivalent sentence using only $\neg$, $\wedge$, and $\vee$.

Example: XOR ("exclusive OR").

P	Q	$P \oplus Q$
T	T	F
T	F	T
F	T	T
F	F	F

- Represent the T rows using "AND".
  - $P = T$ and $Q = F$ becomes $P \wedge \neg Q$.
  - $P = F$ and $Q = T$ becomes $\neg P \wedge Q$.
- Put the rows together with "OR".
  - $(P \wedge \neg Q) \vee (\neg P \wedge Q)$

Replicating Truth Tables

Our language is already fully expressive: given any truth table, we can write an equivalent sentence using only $\neg$, $\wedge$, and $\vee$.

Example: XOR ("exclusive OR").

P	Q	$P \oplus Q$
T	T	F
T	F	T
F	T	T
F	F	F

- Represent the T rows using "AND".
  - $P = T$ and $Q = F$ becomes $P \wedge \neg Q$.
  - $P = F$ and $Q = T$ becomes $\neg P \wedge Q$.
- Put the rows together with "OR".
  - $(P \wedge \neg Q) \vee (\neg P \wedge Q)$
- Can you see why this works?

Implications

Can we use even fewer operators?

Implications

Can we use even fewer operators? *Yes*, for a fully expressive language, *one* operator suffices: "NAND" (Note 1, Exercise).

Implications

Can we use even fewer operators? *Yes*, for a fully expressive language, *one* operator suffices: "NAND" (Note 1, Exercise).

However, more symbols are convenient!

Implications

Can we use even fewer operators? *Yes*, for a fully expressive language, *one* operator suffices: "NAND" (Note 1, Exercise).

However, more symbols are convenient!

- **Implication** ($\implies$): $P \implies Q$ is True *unless* P is True and Q is False.

P	Q	$P \implies Q$
T	T	T
T	F	F
F	T	T
F	F	T

Implications

Can we use even fewer operators? *Yes*, for a fully expressive language, *one* operator suffices: "NAND" (Note 1, Exercise).

However, more symbols are convenient!

- **Implication** ($\Longrightarrow$): $P \Longrightarrow Q$ is True *unless* P is True and Q is False.

P	Q	$P \Longrightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

- **Biconditional** ($\Longleftrightarrow$): $P \Longleftrightarrow Q$ means $(P \Longrightarrow Q) \wedge (Q \Longrightarrow P)$.

Interpretation of Implications

In $P \implies Q$, the **hypothesis** is P and the **conclusion** is Q.

Interpretation of Implications

In $P \implies Q$, the **hypothesis** is P and the **conclusion** is Q.

If $P \implies Q$ is True, then this is a *promise*: if P is True, then Q must be True; if P is False, all bets are off.

Interpretation of Implications

In $P \implies Q$, the **hypothesis** is P and the **conclusion** is Q.

If $P \implies Q$ is True, then this is a *promise*: if P is True, then Q must be True; if P is False, all bets are off.

English translations:

- if P then Q

Interpretation of Implications

In $P \implies Q$, the **hypothesis** is P and the **conclusion** is Q.

If $P \implies Q$ is True, then this is a *promise*: if P is True, then Q must be True; if P is False, all bets are off.

English translations:

- if P then Q
- P implies Q

Interpretation of Implications

In $P \implies Q$, the **hypothesis** is P and the **conclusion** is Q.

If $P \implies Q$ is True, then this is a *promise*: if P is True, then Q must be True; if P is False, all bets are off.

English translations:

- if P then Q
- P implies Q
- Q is implied by P

Interpretation of Implications

In $P \implies Q$, the **hypothesis** is P and the **conclusion** is Q.

If $P \implies Q$ is True, then this is a *promise*: if P is True, then Q must be True; if P is False, all bets are off.

English translations:

- if P then Q
- P implies Q
- Q is implied by P
- P is a sufficient condition for Q

Interpretation of Implications

In $P \implies Q$, the **hypothesis** is P and the **conclusion** is Q.

If $P \implies Q$ is True, then this is a *promise*: if P is True, then Q must be True; if P is False, all bets are off.

English translations:

- if P then Q
- P implies Q
- Q is implied by P
- P is a sufficient condition for Q
- Q is a necessary condition for P

Contrapositive

For an implication $P \implies Q$,

Contrapositive

For an implication $P \implies Q$,

- $Q \implies P$ is the **converse** implication.

Contrapositive

For an implication $P \implies Q$,

- $Q \implies P$ is the **converse** implication.
- $\neg Q \implies \neg P$ is the **contrapositive** implication.

Contrapositive

For an implication $P \implies Q$,

- $Q \implies P$ is the **converse** implication.
- $\neg Q \implies \neg P$ is the **contrapositive** implication.

Key idea: If an implication is True, then the contrapositive is True; the converse need **NOT** be true.

Contrapositive

For an implication $P \implies Q$,

- $Q \implies P$ is the **converse** implication.
- $\neg Q \implies \neg P$ is the **contrapositive** implication.

Key idea: If an implication is True, then the contrapositive is True; the converse need **NOT** be true.

- Example: If I am an apple, I am a fruit. (True)

Contrapositive

For an implication $P \implies Q$,

- $Q \implies P$ is the **converse** implication.
- $\neg Q \implies \neg P$ is the **contrapositive** implication.

Key idea: If an implication is True, then the contrapositive is True; the converse need **NOT** be true.

- Example: If I am an apple, I am a fruit. (True)
- Converse: If I am a fruit, I am an apple. (**False**—I could be an orange.)

Contrapositive

For an implication $P \implies Q$,

- $Q \implies P$ is the **converse** implication.
- $\neg Q \implies \neg P$ is the **contrapositive** implication.

Key idea: If an implication is True, then the contrapositive is True; the converse need **NOT** be true.

- Example: If I am an apple, I am a fruit. (True)
- Converse: If I am a fruit, I am an apple. (**False**—I could be an orange.)
- Contrapositive: If I am not a fruit, I am not an apple. (True)

Contrapositive

For an implication $P \implies Q$,

- $Q \implies P$ is the **converse** implication.
- $\neg Q \implies \neg P$ is the **contrapositive** implication.

Key idea: If an implication is True, then the contrapositive is True; the converse need **NOT** be true.

- Example: If I am an apple, I am a fruit. (True)
- Converse: If I am a fruit, I am an apple. (**False**—I could be an orange.)
- Contrapositive: If I am not a fruit, I am not an apple. (True)

P	Q	$\neg P$	$\neg Q$	$P \implies Q$	$\neg Q \implies \neg P$
T	T	F	F	T	T
T	F	F	T	F	F
F	T	T	F	T	T
F	F	T	T	T	T

Quantifiers

Consider the statement "$x > 0$". **True or False?**

[7] Read this as "there exists an x such that $P(x)$ holds".
[8] Read this as "for all x, $P(x)$ holds".

Quantifiers

Consider the statement "$x > 0$". **True or False?**

Depends on what x is.

[7] Read this as "there exists an x such that $P(x)$ holds".
[8] Read this as "for all x, $P(x)$ holds".

Quantifiers

Consider the statement "$x > 0$". **True or False?**

Depends on what x is. Say $P(x)$ is the statement "$x > 0$". Here, x is called a **free variable**.

---

[7] Read this as "there exists an x such that $P(x)$ holds".
[8] Read this as "for all x, $P(x)$ holds".

Quantifiers

Consider the statement "$x > 0$". **True or False?**

Depends on what x is. Say $P(x)$ is the statement "$x > 0$". Here, x is called a **free variable**.

Introducing **quantifiers**.

[7] Read this as "there exists an x such that $P(x)$ holds".
[8] Read this as "for all x, $P(x)$ holds".

Quantifiers

Consider the statement "$x > 0$". **True or False?**

Depends on what x is. Say $P(x)$ is the statement "$x > 0$". Here, x is called a **free variable**.

Introducing **quantifiers.**

- **Existential Quantifier** ($\exists$): $\exists x\ P(x)$[7] is a *proposition* whose truth value is True if $P(x)$ is True for *some* x.

[7] Read this as "there exists an x such that $P(x)$ holds".

[8] Read this as "for all x, $P(x)$ holds".

Quantifiers

Consider the statement "$x > 0$". **True or False?**

Depends on what x is. Say $P(x)$ is the statement "$x > 0$". Here, x is called a **free variable**.

Introducing **quantifiers**.

- **Existential Quantifier** ($\exists$): $\exists x\ P(x)$[7] is a *proposition* whose truth value is True if $P(x)$ is True for *some* x.
- **Universal Quantifier** ($\forall$): Similarly, $\forall x\ P(x)$[8] is True if $P(x)$ is True for *all* x.

[7] Read this as "there exists an x such that $P(x)$ holds".
[8] Read this as "for all x, $P(x)$ holds".

Quantifiers

Consider the statement "$x > 0$". **True or False?**

Depends on what x is. Say $P(x)$ is the statement "$x > 0$". Here, x is called a **free variable**.

Introducing **quantifiers**.

- **Existential Quantifier** ($\exists$): $\exists x\ P(x)$[7] is a *proposition* whose truth value is True if $P(x)$ is True for *some* x.
- **Universal Quantifier** ($\forall$): Similarly, $\forall x\ P(x)$[8] is True if $P(x)$ is True for *all* x.

Formally, this is an *extension* of propositional logic, called **first-order logic**.

[7] Read this as "there exists an x such that $P(x)$ holds".
[8] Read this as "for all x, $P(x)$ holds".

Quantifiers

Consider the statement "$x > 0$". **True or False?**

Depends on what x is. Say $P(x)$ is the statement "$x > 0$". Here, x is called a **free variable**.

Introducing **quantifiers.**

- **Existential Quantifier** ($\exists$): $\exists x\ P(x)$[7] is a *proposition* whose truth value is True if $P(x)$ is True for *some* x.
- **Universal Quantifier** ($\forall$): Similarly, $\forall x\ P(x)$[8] is True if $P(x)$ is True for *all* x.

Formally, this is an *extension* of propositional logic, called **first-order logic.**

Example: $\forall x\ (x > 0 \implies x^2 > 0)$

[7] Read this as "there exists an x such that $P(x)$ holds".
[8] Read this as "for all x, $P(x)$ holds".

Quantifier Universes

Sometimes, we are only interested in members of a set S.

Quantifier Universes

Sometimes, we are only interested in members of a set S.

To restrict our discussion to elements of S, we can write

$$\exists x\, (x \in S \wedge \cdots)$$

or

$$\forall x\, (x \in S \implies \cdots).$$

Quantifier Universes

Sometimes, we are only interested in members of a set S.

To restrict our discussion to elements of S, we can write

$$\exists x\,(x \in S \land \cdots)$$

or

$$\forall x\,(x \in S \implies \cdots).$$

Equivalently, we can write

$$\exists x \in S\,(\cdots),$$
$$\forall x \in S\,(\cdots).$$

Quantifier Universes

Sometimes, we are only interested in members of a set S.

To restrict our discussion to elements of S, we can write

$$\exists x\,(x \in S \wedge \cdots)$$

or

$$\forall x\,(x \in S \implies \cdots).$$

Equivalently, we can write

$$\exists x \in S\,(\cdots),$$
$$\forall x \in S\,(\cdots).$$

Example: $\forall x \in \mathbb{N}\,(x \text{ is even} \vee x \text{ is odd})$

De Morgan’s Laws for Quantifiers

“Unicorns do not exist.”

De Morgan's Laws for Quantifiers

"Unicorns do not exist."

"For all x, x is not a unicorn."

De Morgan’s Laws for Quantifiers

“Unicorns do not exist.”

“For all x, x is not a unicorn.”

The two statements above are equivalent.

De Morgan's Laws for Quantifiers

"Unicorns do not exist."

"For all x, x is not a unicorn."

The two statements above are equivalent. If $U(x)$ is the statement that "x is a unicorn", then the two statements are:

De Morgan's Laws for Quantifiers

"Unicorns do not exist."

"For all x, x is not a unicorn."

The two statements above are equivalent. If $U(x)$ is the statement that "x is a unicorn", then the two statements are:

- $\neg\exists x\, U(x)$,

De Morgan's Laws for Quantifiers

"Unicorns do not exist."

"For all x, x is not a unicorn."

The two statements above are equivalent. If $U(x)$ is the statement that "x is a unicorn", then the two statements are:

- $\neg\exists x\, U(x)$,
- $\forall x\, \neg U(x)$.

De Morgan's Laws for Quantifiers

"Unicorns do not exist."

"For all x, x is not a unicorn."

The two statements above are equivalent. If $U(x)$ is the statement that "x is a unicorn", then the two statements are:

- $\neg \exists x\, U(x)$,
- $\forall x\, \neg U(x)$.

These are called **De Morgan's Laws for Quantifiers**:

$$\neg \exists x\, P(x) \equiv \forall x\, \neg P(x)$$
$$\neg \forall x\, P(x) \equiv \exists x\, \neg P(x)$$

De Morgan's Laws for Quantifiers

"Unicorns do not exist."

"For all x, x is not a unicorn."

The two statements above are equivalent. If $U(x)$ is the statement that "x is a unicorn", then the two statements are:

- $\neg \exists x\, U(x)$,
- $\forall x\, \neg U(x)$.

These are called **De Morgan's Laws for Quantifiers**:

$$\neg \exists x\, P(x) \equiv \forall x\, \neg P(x)$$
$$\neg \forall x\, P(x) \equiv \exists x\, \neg P(x)$$

"If you move the negation through a quantifier, flip the quantifier."

Do Quantifiers Commute?

Is $\forall x\, \exists y\, P(x, y)$ the same as $\exists y\, \forall x\, P(x, y)$?

Do Quantifiers Commute?

Is $\forall x \, \exists y \; P(x, y)$ the same as $\exists y \, \forall x \; P(x, y)$?

NO.

Do Quantifiers Commute?

Is $\forall x\, \exists y\, P(x, y)$ the same as $\exists y\, \forall x\, P(x, y)$?

NO. Let $P(x, y)$ be the statement “x loves y”.

Do Quantifiers Commute?

Is $\forall x\, \exists y\, P(x, y)$ the same as $\exists y\, \forall x\, P(x, y)$?

NO. Let $P(x, y)$ be the statement "x loves y".

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

Do Quantifiers Commute?

Is $\forall x\, \exists y\, P(x, y)$ the same as $\exists y\, \forall x\, P(x, y)$?

NO. Let $P(x, y)$ be the statement "x loves y".

- $\forall x\, \exists y\, P(x, y)$. Everyone has someone that he/she loves.

Do Quantifiers Commute?

Is $\forall x \, \exists y \, P(x, y)$ the same as $\exists y \, \forall x \, P(x, y)$?

NO. Let $P(x, y)$ be the statement "x loves y".

- $\forall x \, \exists y \, P(x, y)$. Everyone has someone that he/she loves.
- $\exists y \, \forall x \, P(x, y)$.

Do Quantifiers Commute?

Is $\forall x\, \exists y\, P(x, y)$ the same as $\exists y\, \forall x\, P(x, y)$?

NO. Let $P(x, y)$ be the statement "x loves y".

- $\forall x\, \exists y\, P(x, y)$. Everyone has someone that he/she loves.
- $\exists y\, \forall x\, P(x, y)$. There is a person who everyone loves.

Do Quantifiers Commute?

Is $\forall x\, \exists y\, P(x, y)$ the same as $\exists y\, \forall x\, P(x, y)$?

NO. Let $P(x, y)$ be the statement "x loves y".

- $\forall x\, \exists y\, P(x, y)$. Everyone has someone that he/she loves.
- $\exists y\, \forall x\, P(x, y)$. There is a person who everyone loves.

Not the same!

Do Quantifiers Commute?

Is $\forall x \, \exists y \; P(x, y)$ the same as $\exists y \, \forall x \; P(x, y)$?

NO. Let $P(x, y)$ be the statement "x loves y".

- $\forall x \, \exists y \; P(x, y)$. Everyone has someone that he/she loves.
- $\exists y \, \forall x \; P(x, y)$. There is a person who everyone loves.

Not the same!

However, $\forall x \, \forall y \; P(x, y) \equiv \forall y \, \forall x \; P(x, y)$ and
$\exists x \, \exists y \; P(x, y) \equiv \exists y \, \exists x \; P(x, y)$.

Intuition for Quantifiers

Suppose our universe consists of three elements: $\{x_1, x_2, x_3\}$.

Intuition for Quantifiers

Suppose our universe consists of three elements: $\{x_1, x_2, x_3\}$.

$\forall x\ P(x)$ is like "AND": $P(x_1) \wedge P(x_2) \wedge P(x_3)$.

Intuition for Quantifiers

Suppose our universe consists of three elements: $\{x_1, x_2, x_3\}$.

$\forall x\ P(x)$ is like "AND": $P(x_1) \wedge P(x_2) \wedge P(x_3)$.

$\exists x\ P(x)$ is like "OR": $P(x_1) \vee P(x_2) \vee P(x_3)$.

Intuition for Quantifiers

Suppose our universe consists of three elements: $\{x_1, x_2, x_3\}$.

$\forall x\ P(x)$ is like "AND": $P(x_1) \wedge P(x_2) \wedge P(x_3)$.

$\exists x\ P(x)$ is like "OR": $P(x_1) \vee P(x_2) \vee P(x_3)$.

Intuition: $\forall$ and $\exists$ let us express *infinite* strings of "AND" and "OR" statements.

Intuition for Quantifiers

Suppose our universe consists of three elements: $\{x_1, x_2, x_3\}$.

$\forall x\ P(x)$ is like "AND": $P(x_1) \wedge P(x_2) \wedge P(x_3)$.

$\exists x\ P(x)$ is like "OR": $P(x_1) \vee P(x_2) \vee P(x_3)$.

Intuition: $\forall$ and $\exists$ let us express *infinite* strings of "AND" and "OR" statements.

- Moreover, $\forall x\ (P(x) \wedge Q(x)) \equiv (\forall x\ P(x)) \wedge (\forall x\ Q(x))$.

Intuition for Quantifiers

Suppose our universe consists of three elements: $\{x_1, x_2, x_3\}$.

$\forall x\ P(x)$ is like "AND": $P(x_1) \wedge P(x_2) \wedge P(x_3)$.

$\exists x\ P(x)$ is like "OR": $P(x_1) \vee P(x_2) \vee P(x_3)$.

Intuition: $\forall$ and $\exists$ let us express *infinite* strings of "AND" and "OR" statements.

- Moreover, $\forall x\ (P(x) \wedge Q(x)) \equiv (\forall x\ P(x)) \wedge (\forall x\ Q(x))$.
- Similarly, $\exists x\ (P(x) \vee Q(x)) \equiv (\exists x\ P(x)) \vee (\exists x\ Q(x))$.

Summary

- The language of propositional logic: (,), $\neg$, $\wedge$, $\vee$, $\implies$, $\iff$.
- Two sentences are equivalent if they have the same truth values regardless of the truth values of their component propositions.
- Truth tables help us evaluate logical statements and prove propositional equivalences.
- Useful propositional equivalences: distributivity, double negatives, De Morgan's Laws.
- Implications are equivalent to their contrapositive implications.
- The language of first-order logic: $\exists$, $\forall$.