Mathematical Logic

Part One

An Important Question

How do we formalize the logic we've been using in our proofs?

Where We're Going

- **Propositional Logic** (Today)
  - Basic logical connectives.
  - Truth tables.
  - Logical equivalences.
- **First-Order Logic** (Today/Friday)
  - Reasoning about properties of multiple objects.

Propositional Logic

A **proposition** is a statement that is, by itself, either true or false.

Some Sample Propositions

- Puppies are cuter than kittens.
- Kittens are cuter than puppies.
- Usain Bolt can outrun everyone in this room.
- CS103 is useful for cocktail parties.
- This is the last entry on this list.

More Propositions

- I came in like a wrecking ball.
- I am a champion.
- You're going to hear me roar.
- We all just entertainers.

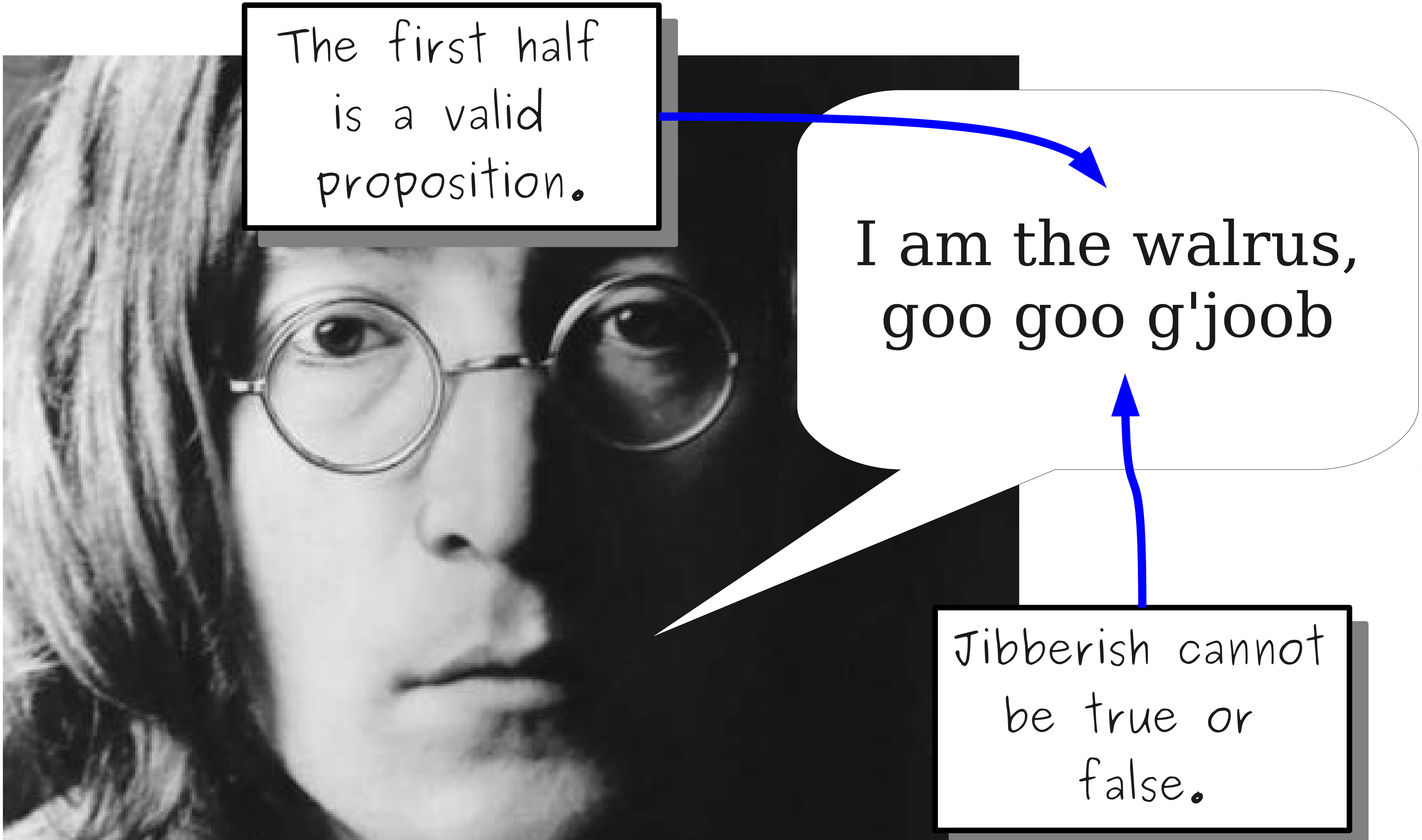
Things That Aren't Propositions

Commands cannot be true or false.

Things That Aren't Propositions

Questions cannot be true or false.

Things That Aren't Propositions



Propositional Logic

- **Propositional logic** is a mathematical system for reasoning about propositions and how they relate to one another.
- Every statement in propositional logic consists of **propositional variables** combined via **logical connectives**.
  - Each variable represents some proposition, such as “You liked it” or “You should have put a ring on it.”
  - Connectives encode how propositions are related, such as “If you liked it, then you should have put a ring on it.”

Propositional Variables

- Each proposition will be represented by a **propositional variable**.
- Propositional variables are usually represented as lower-case letters, such as *p*, *q*, *r*, *s*, etc.
- Each variable can take one one of two values: true or false.

Logical Connectives

- **Logical NOT: $\neg p$**
  - Read "**not** p"
  - $\neg p$ is true if and only if p is false.
  - Also called **logical negation**.
- **Logical AND: $p \land q$**
  - Read "p **and** q."
  - $p \land q$ is true if both p and q are true.
  - Also called **logical conjunction**.
- **Logical OR: $p \lor q$**
  - Read "p **or** q."
  - $p \lor q$ is true if at least one of p or q are true (inclusive OR)
  - Also called **logical disjunction**.

Truth Tables

p	q	$p \land q$
F	F	F
F	T	F
T	F	F
T	T	T

Truth Tables

p	q	$p \lor q$
F	F	F
F	T	T
T	F	T
T	T	T

This "or" is an inclusive or.

Truth Tables

p	$\neg p$
F	T
T	F

Truth Table for Implication

p	q	$p \to q$
F	F	T
F	T	T
T	F	
T	T	

In both of these cases, p is false, so the statement "if p, then q" is vacuously true.

Truth Table for Implication

p	q	$p \to q$
F	F	T
F	T	T
T	F	F
T	T	

$p \to q$ should mean when p is true, q is true as well. But here p is true and q is false!

Truth Table for Implication

p	q	$p \to q$
F	F	T
F	T	T
T	F	F
T	T	T

$p \to q$ means that if we ever find that p is true, we'll find that q is true as well.

Truth Table for Implication

p	q	$p \to q$
F	F	T
F	T	T
T	F	F
T	T	T

The <u>only way</u> for $p \to q$ to be false is for p to be true and q to be false.

The Biconditional

- The **biconditional** connective $p \leftrightarrow q$ is read "p if and only if q."
- Intuitively, either both p and q are true, or neither of them are.

p	q	$p \leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

One interpretation of $\leftrightarrow$ is to think of it as equality: the two propositions must have equal truth values.

True and False

- There are two more “connectives” to speak of: true and false.
  - The symbol $\top$ is a value that is always true.
  - The symbol $\bot$ is value that is always false.
- These are often called connectives, though they don't connect anything.
  - (Or rather, they connect zero things.)

Operator Precedence

- How do we parse this statement?

$$\neg x \to y \lor z \to x \lor y \land z$$

- Operator precedence for propositional logic:

$$\neg$$
$$\land$$
$$\lor$$
$$\to$$
$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Operator Precedence

- How do we parse this statement?

$$(\neg x) \rightarrow ((y \lor z) \rightarrow (x \lor (y \land z)))$$

- Operator precedence for propositional logic:

$$\neg$$
$$\land$$
$$\lor$$
$$\rightarrow$$
$$\leftrightarrow$$

- All operators are right-associative.
- We can use parentheses to disambiguate.

Recap So Far

- A **propositional variable** is a variable that is either true or false.
- The **logical connectives** are
  - Negation: $\neg p$
  - Conjunction: $p \land q$
  - Disjunction: $p \lor q$
  - Implication: $p \to q$
  - Biconditional: $p \leftrightarrow q$
  - True: $\top$
  - False: $\bot$

Translating into Propositional Logic

Some Sample Propositions

a: There is a velociraptor outside my apartment.

b: Velociraptors can open windows.

c: I am in my apartment right now.

d: My apartment has windows.

e: I am going to be eaten by a velociraptor

"I won't be eaten by a velociraptor if there isn't a velociraptor outside my apartment."

$$\neg a \to \neg e$$

“p if q”

translates to

$$q \rightarrow p$$

It does *not* translate to

$$p \rightarrow q$$

Some Sample Propositions

a: There is a velociraptor outside my apartment.

b: Velociraptors can open windows.

c: I am in my apartment right now.

d: My apartment has windows.

e: I am going to be eaten by a velociraptor

"If there is a velociraptor outside my apartment, but velociraptors can't open windows, I am not going to be eaten by a velociraptor."

$$a \wedge \neg b \rightarrow \neg e$$

“p, but q”

translates to

$$p \wedge q$$

The Takeaway Point

- When translating into or out of propositional logic, be very careful not to get tripped up by nuances of the English language.
  - In fact, this is one of the reasons we have a symbolic notation in the first place!
- Many prepositional phrases lead to counterintuitive translations; make sure to double-check yourself!

More Elaborate Truth Tables

This gives the final truth value for the expression.

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

Logical Equivalence

Negations

- $p \land q$ is false if and only if $\neg(p \land q)$ is true.
- Intuitively, this is only possible if either p is false or q is false (or both!)
- In propositional logic, we can write this as $\neg p \lor \neg q$.
- How would we prove that $\neg(p \land q)$ and $\neg p \lor \neg q$ are equivalent?
- **Idea**: Build truth tables for both expressions and confirm that they always agree.

Negating AND

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

p	q	$\neg p$	$\lor$	$\neg q$
F	F	T	T	T
F	T	T	T	F
T	F	F	T	T
T	T	F	F	F

These two statements are always the same!

Logical Equivalence

- If two propositional logic statements φ and ψ always have the same truth values as one another, they are called **logically equivalent**.
- We denote this by **φ ≡ ψ**.
- ≡ is not a connective. It is a statement used to describe propositional formulas.
  - **φ ↔ ψ** is a propositional statement that can take on different truth values based on how φ and ψ evaluate. Think of it as a function of φ and ψ.
  - **φ ≡ ψ** is an assertion that the formulas always take on the same values. It is either true or it isn't.

De Morgan's Laws

- Using truth tables, we concluded that

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

- We can also use truth tables to show that

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

- These two equivalences are called **De Morgan's Laws**.

Another Important Equivalence

- When is $p \to q$ false?
- **Answer**: p must be true and q must be false.
- In propositional logic:

$$\boldsymbol{p \land \neg q}$$

- Is the following true?

$$\boldsymbol{\neg(p \to q) \equiv p \land \neg q}$$

Negating Implications

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

p	q	p	$\land$	$\neg q$
F	F	F	F	T
F	T	F	F	F
T	F	T	T	T
T	T	T	F	F

$$\neg(p \to q) \equiv p \land \neg q$$

An Important Observation

- We have just proven that

$$\neg(p \to q) \equiv p \land \neg q$$

- If we negate both sides, we get that

$$p \to q \equiv \neg(p \land \neg q)$$

- By De Morgan's laws:

$$p \to q \equiv \neg(p \land \neg q)$$

$$p \to q \equiv \neg p \lor \neg\neg q$$

$$p \to q \equiv \neg p \lor q$$

- Thus $\boldsymbol{p \to q \equiv \neg p \lor q}$

If p is false, the whole thing is true and we gain no information. If p is true, then q has to be true for the whole expression to be true.

Why This Matters

- Understanding these equivalences helps justify how proofs work and what to prove.
- Unsure what to prove? Try translating it into logic first and see what happens.

Announcements!

Problem Set Three Checkpoint

- Problem Set Three checkpoints graded and solutions are released.
- ***Please review the feedback and solution set***. Parts (ii) and (iv) are trickier than they might seem.
- On-time Problem Set Two's should be graded and returned by tomorrow at noon in the homework return bin.
  - Please keep everything sorted!
  - Please don't leave papers sitting out!

A Note on Induction

- In an inductive proof, $P(n)$ must be a statement that is either true or false for a particular choice of n.
- Examples:
  - $P(n)$ = "$a_n = 2^n$."
  - $P(n)$ = "any tournament with n players has a winner."
- Non-examples:
  - $P(n)$ = "a game of Nim with n stones in each pile"
  - $P(n)$ = "for any $n \in \mathbb{N}$, $a_n = 2^n$."

Your Questions

What are some practical applications of cardinality? Why is it useful?

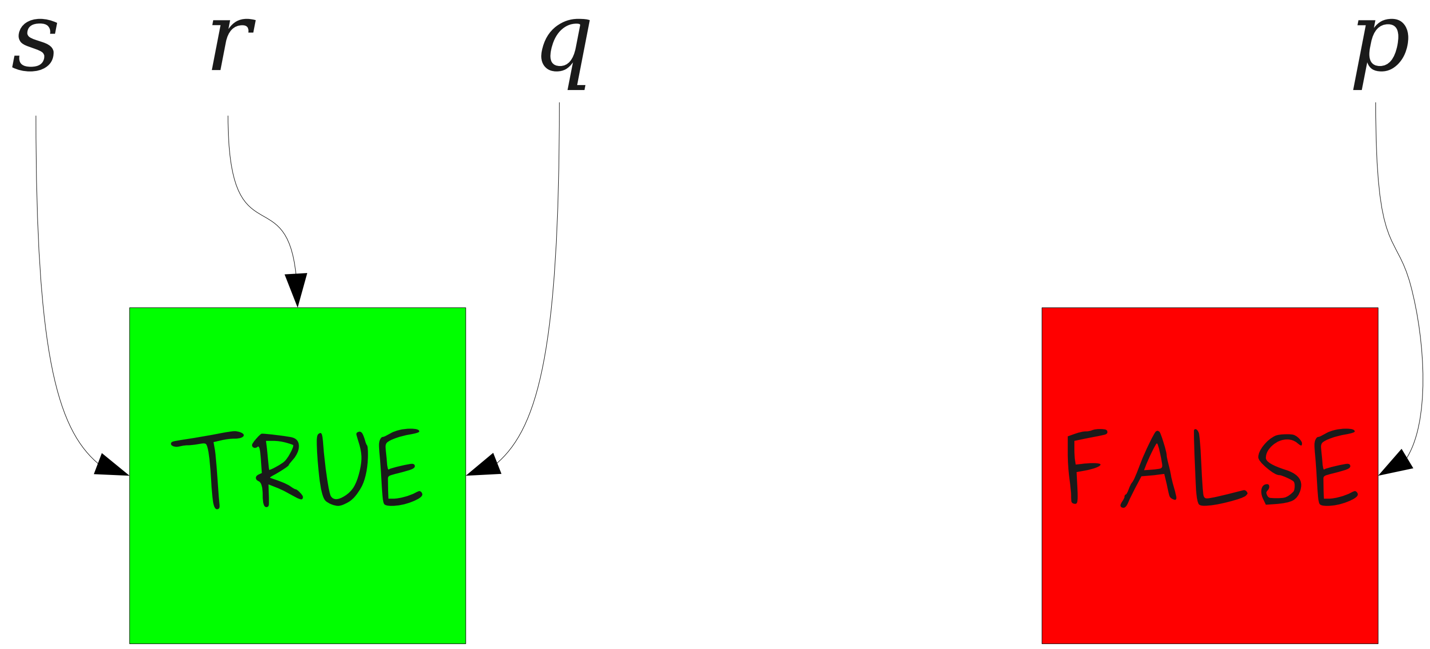
First-Order Logic

What is First-Order Logic?

- **First-order logic** is a logical system for reasoning about properties of objects.
- Augments the logical connectives from propositional logic with
  - **predicates** that describe properties of objects, and
  - **functions** that map objects to one another,
  - **quantifiers** that allow us to reason about multiple objects simultaneously.

The Universe of Propositional Logic

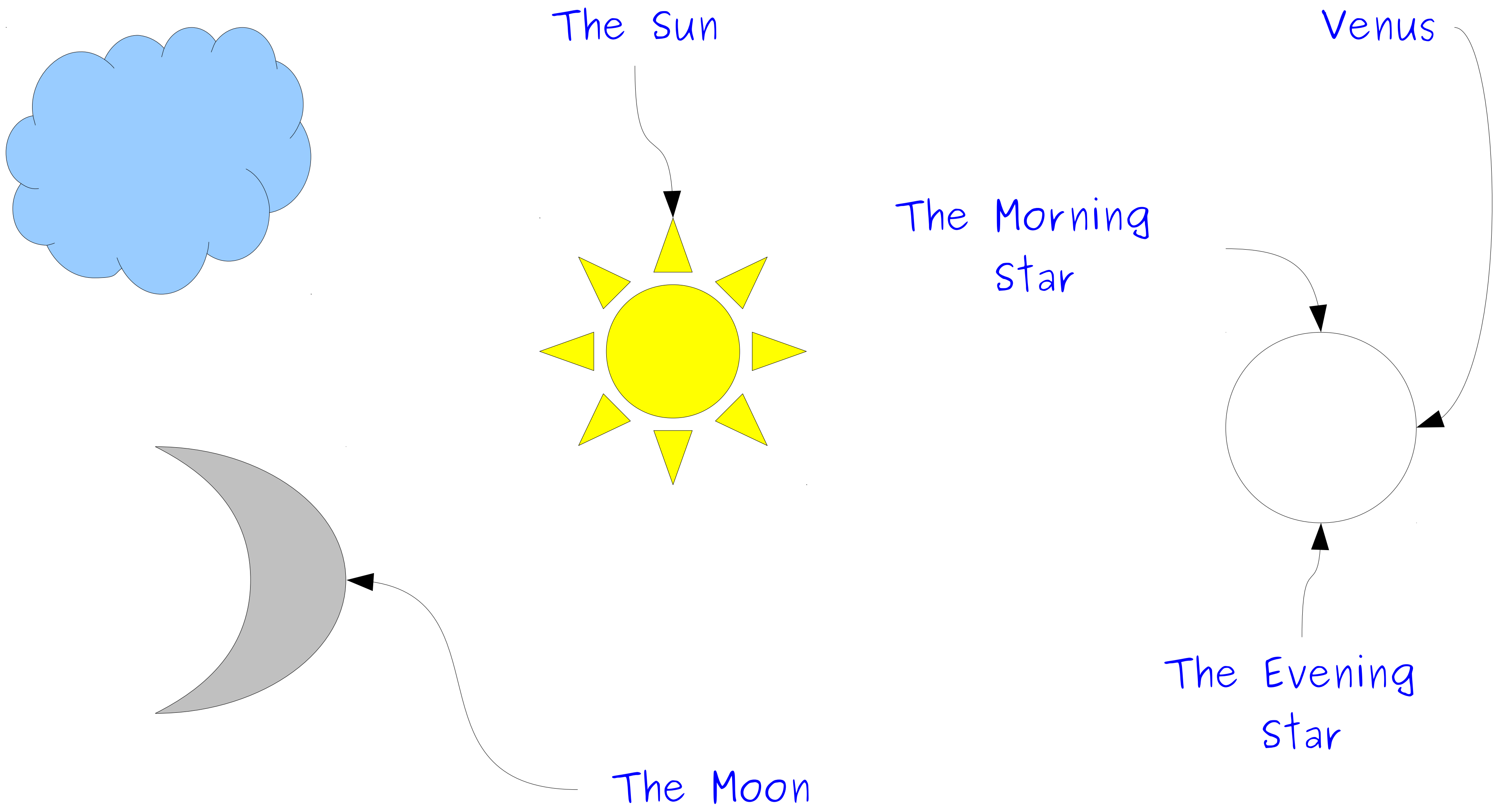
$$p \land q \to \neg r \lor \neg s$$



Propositional Logic

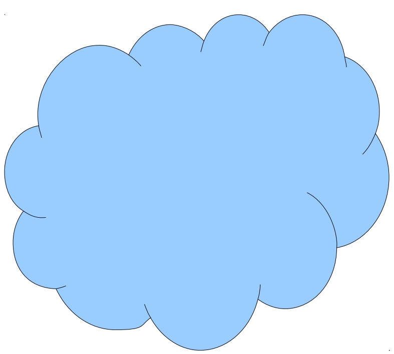
- In propositional logic, each variable represents a **proposition**, which is either true or false.
- We can directly apply connectives to propositions:
  - $p \to q$
  - $\neg p \land q$
- The truth of a statement can be determined by plugging in the truth values for the input propositions and computing the result.
- We can see all possible truth values for a statement by checking all possible truth assignments to its variables.

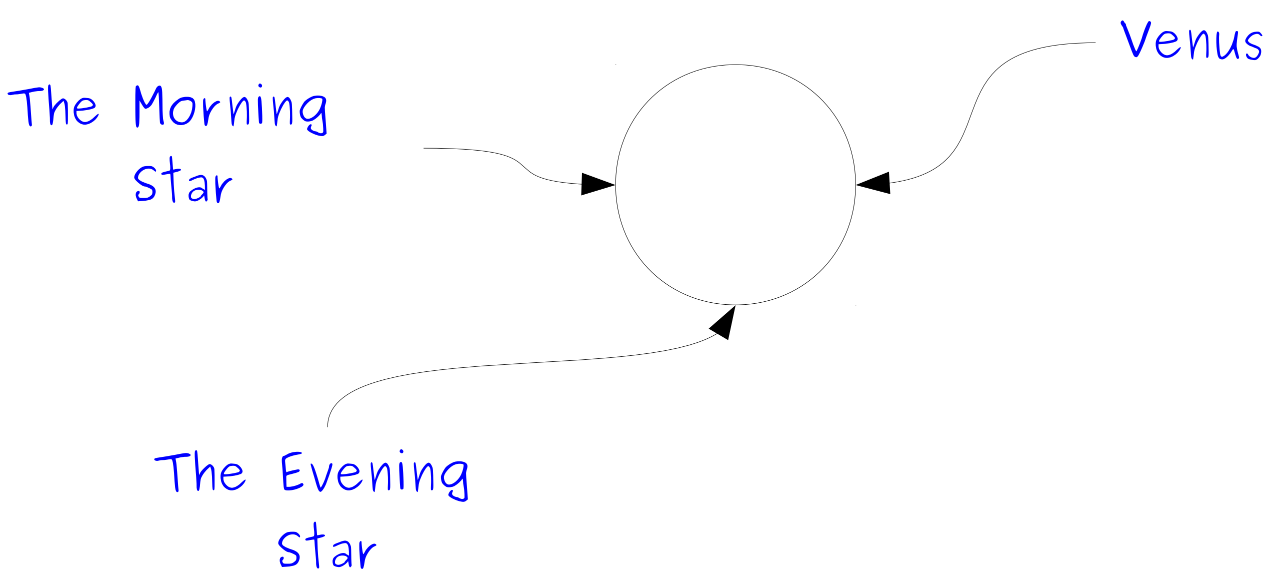
The Universe of First-Order Logic



First-Order Logic

- In first-order logic, each variable refers to some object in a set called the **domain of discourse**.
- Some objects may have multiple names.
- Some objects may have no name at all.





Propositional vs. First-Order Logic

- Because propositional variables are either true or false, we can directly apply connectives to them.

$$\boldsymbol{p \to q} \qquad\qquad \boldsymbol{\neg p \leftrightarrow q \land r}$$

- Because first-order variables refer to arbitrary objects, it does not make sense to apply connectives to them.

$$\boldsymbol{Venus \to Sun} \qquad\qquad \boldsymbol{137 \leftrightarrow \neg 42}$$

- ***This is not C!***

Reasoning about Objects

- To reason about objects, first-order logic uses **predicates**.
- Examples:
  - $ExtremelyCute(Quokka)$
  - $DeadlockEachOther(House, Senate)$
- Predicates can take any number of arguments, but each predicate has a fixed number of arguments (called its **arity**)
- Applying a predicate to arguments produces a proposition, which is either true or false.

First-Order Sentences

- Sentences in first-order logic can be constructed from predicates applied to objects:

$$LikesToEat(V, M) \land Near(V, M) \to WillEat(V, M)$$

$$Cute(t) \to Dikdik(t) \lor Kitty(t) \lor Puppy(t)$$

$$x < 8 \to x < 137$$

The notation $\mathbf{x < 8}$ is just a shorthand for something like **LessThan(x, 8)**. Binary predicates in math are often written like this, but symbols like **<** are not a part of first-order logic.

Equality

- First-order logic is equipped with a special predicate **=** that says whether two objects are equal to one another.
- Equality is a part of first-order logic, just as $\to$ and $\neg$ are.
- Examples:

$$MorningStar = EveningStar$$

$$Glinda = GoodWitchOfTheNorth$$

- Equality can only be applied to **objects**; to see if **propositions** are equal, use $\leftrightarrow$.

For notational simplicity, define $\neq$ as

$$x \neq y \quad \equiv \quad \neg(x = y)$$

Next Time

- **First-Order Logic II**
  - Functions and quantifiers.
  - How do we translate statements into first-order logic?
  - Why does any of this matter?