Chapter 1: Foundations: Logic and Proofs

Foundations of Logic (§1.1-1.3)

Mathematical Logic is a tool for working with complicated *compound* statements. It includes:

- A language for expressing them.
- A concise notation for writing them.
- A methodology for objectively reasoning about their truth or falsity.
- It is the foundation for expressing formal proofs in all branches of mathematics.

Foundations of Logic: Overview

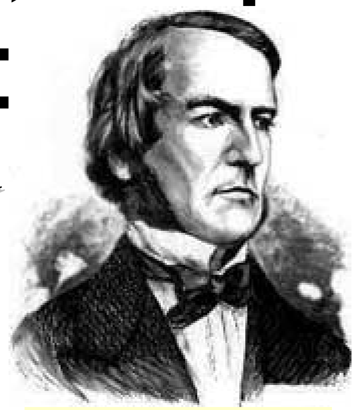
- Propositional logic (§1.1-1.2):
  - Basic definitions. (§1.1)
  - Equivalence rules & derivations. (§1.2)
- Predicate logic (§1.3-1.4)
  - Predicates.
  - Quantified predicate expressions.
  - Equivalences & derivations.

Propositional Logic (§1.1)

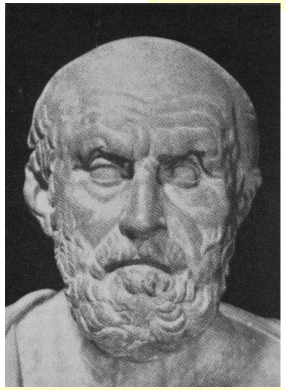
Propositional Logic is the logic of compound statements built from simpler statements using so-called *Boolean connectives*.

Some applications in computer science:

- Design of digital electronic circuits.
- Expressing conditions in programs.
- Queries to databases & search engines.



George Boole (1815-1864)



Chrysippus of Soli (ca. 281 B.C. – 205 B.C.)

Definition of a *Proposition*

A ***proposition*** (*p*, *q*, *r*, …) is simply a *statement* (*i.e.*, a declarative sentence) *with a definite meaning*, having a *truth value* that's either *true* (T) or *false* (F) (**never** both, neither, or somewhere in between).

(However, you might not *know* the actual truth value, and it might be situation-dependent.)

[Later we will study *probability theory,* in which we assign *degrees of certainty* to propositions. But for now: think True/False only!]

Examples of Propositions

- "It is raining." (In a given situation.)
- "Beijing is the capital of China." • "1 + 2 = 3"

But, the following are **NOT** propositions:

- "Who's there?" (interrogative, question)
- "La la la la la." (meaningless interjection)
- "Just do it!" (imperative, command)
- "Yeah, I sorta dunno, whatever..." (vague)
- "1 + 2" (expression with a non-true/false value)

Operators / Connectives

An *operator* or *connective* combines one or more *operand* expressions into a larger expression. (*E.g.*, "+" in numeric exprs.)

Unary operators take 1 operand (*e.g.*, −3); *Binary* operators take 2 operands (*eg* 3 × 4).

Propositional or *Boolean* operators operate on propositions or truth values instead of on numbers.

Some Popular Boolean Operators

Formal Name	**Nickname**	**Arity**	**Symbol**
Negation operator	NOT	Unary	$\neg$
Conjunction operator	AND	Binary	$\wedge$
Disjunction operator	OR	Binary	$\vee$
Exclusive-OR operator	XOR	Binary	$\oplus$
Implication operator	IMPLIES	Binary	$\rightarrow$
Biconditional operator	IFF	Binary	$\leftrightarrow$

The Negation Operator

The unary *negation operator* "¬" (*NOT*) transforms a prop. into its logical *negation*.

E.g. If p = "I have brown hair."

then $\neg p$ = "I do **not** have brown hair."

Truth table for NOT:

p	$\neg p$
T	
F	

Operand column — Result column

T :≡ True; F :≡ False

":≡" means "is defined as"

The Conjunction Operator

The binary *conjunction operator* "∧" (*AND*) combines two propositions to form their logical *conjunction*.

∧ND

E.g. If p="I will have salad for lunch." and q="I will have steak for dinner.", then $p \wedge q$="I will have salad for lunch **and** I will have steak for dinner."

Remember: "∧" points up like an "A", and it means "∧ND"

Conjunction Truth Table

- Note that a conjunction $p_1 \wedge p_2 \wedge \ldots \wedge p_n$ of n propositions will have 2^n rows in its truth table.

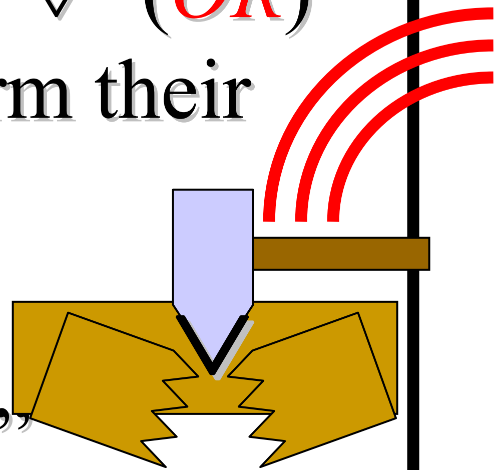
Operand columns

p	q	$p \wedge q$
F	F	
F	T	
T	F	
T	T	

- Also: ¬ and ∧ operations together are sufficient to express *any* Boolean truth table!

The Disjunction Operator

- The binary *disjunction operator* "∨" (*OR*) combines two propositions to form their logical *disjunction*.
- *p*="My car has a bad engine."
- *q*="My car has a bad carburetor."
- *p*∨*q*="Either my car has a bad engine, **or** my car has a bad carburetor."



After the downward-pointing "axe" of "∨" splits the wood, you can take 1 piece OR the other, or both.

Meaning is like "and/or" in English.

Disjunction Truth Table

- Note that $p \lor q$ means that p is true, or q is true, **or both** are true!
- So, this operation is also called *inclusive or,* because it **includes** the possibility that both p and q are true.
- "¬" and "∨" together are also universal.

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

Nested Propositional Expressions

- Use parentheses to *group sub-expressions*: "I just saw my old *f*riend, and either he's grown or I've *s*hrunk." = $f \land (g \lor s)$
  - $(f \land g) \lor s$ would mean something different
  - $f \land g \lor s$ would be ambiguous
- By convention, "¬" takes *precedence* over both "∧" and "∨".
  - $\neg s \land f$ means $(\neg s) \land f$, **not** $\neg (s \land f)$

A Simple Exercise

Let p="It rained last night",
q="The sprinklers came on last night,"
r="The lawn was wet this morning."

Translate each of the following into English:

$\neg p$ = "It didn't rain last night."

$r \wedge \neg p$ = "The lawn was wet this morning, and it didn't rain last night."

$\neg r \vee p \vee q$ = "Either the lawn wasn't wet this morning, or it rained last night, or the sprinklers came on last night."

The *Exclusive Or* Operator

The binary *exclusive-or operator* "⊕" (*XOR*) combines two propositions to form their logical "exclusive or" (exjunction?).

p = "I will earn an A in this course,"

q = "I will drop this course,"

$p \oplus q$ = "I will either earn an A for this course, or I will drop it (**but not both**!)"

Exclusive-Or Truth Table

- Note that $p \oplus q$ means that p is true, or q is true, but **not both**!
- This operation is called *exclusive or*, because it **excludes** the possibility that both p and q are true.
- "¬" and "⊕" together are **not** universal.

p	q	$p \oplus q$
F	F	
F	T	
T	F	
T	T	

Natural Language is Ambiguous

Note that English "or" can be ambiguous regarding the "both" case!

"Pat is a singer or Pat is a writer." - $\vee$

"Pat is a man or Pat is a woman." - $\oplus$

p	q	p "or" q
F	F	
F	T	
T	F	
T	T	

Need context to disambiguate the meaning!

For this class, assume "or" means inclusive.

The *Implication* Operator

antecedent consequent

The *implication* $p \rightarrow q$ states that p implies q.

I.e., If p is true, then q is true; but if p is not true, then q could be either true or false.

E.g., let p = "You study hard."
q = "You will get a good grade."

$p \rightarrow q$ = "If you study hard, then you will get a good grade." (else, it could go either way)

Implication Truth Table

- $p \rightarrow q$ is **false** only when p is true but q is **not** true.
- $p \rightarrow q$ does **not** say that p causes q!
- $p \rightarrow q$ does **not** require that p or q **are ever true**!
- *E.g.* "(1=0) → pigs can fly" is TRUE!

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

Examples of Implications

- "If this lecture ends, then the sun will rise tomorrow." *True* or *False*?
- "If Tuesday is a day of the week, then I am a penguin." *True* or *False*?
- "If 1+1=6, then Bush is president." *True* or *False*?
- "If the moon is made of green cheese, then I am richer than Bill Gates." *True* or *False*?

Why does this seem wrong?

- Consider a sentence like,
  - "If I wear a red shirt tomorrow, then the U.S. will attack Iraq the same day."
- In logic, we consider the sentence **True** so long as either I don't wear a red shirt, or the US attacks.
- But in normal English conversation, if I were to make this claim, you would think I was lying.
  - Why this discrepancy between logic & language?

Resolving the Discrepancy

- In English, a sentence "if *p* then *q*" usually really *implicitly* means something like,
  - "In all possible situations, if *p* then *q*."
    - That is, "For *p* to be true and *q* false is *impossible*."
    - Or, "I *guarantee* that no matter what, if *p*, then *q*."
- This can be expressed in *predicate logic* as:
  - "For all situations *s*, if *p* is true in situation *s*, then *q* is also true in situation *s*"
  - Formally, we could write: $\forall s, P(s) \rightarrow Q(s)$
- This sentence is logically ***False*** in our example, because for me to wear a red shirt and the U.S. *not* to attack Iraq is a *possible* (even if not actual) situation.
  - Natural language and logic then agree with each other.

English Phrases Meaning $p \rightarrow q$

- "*p* implies *q*"
- "if *p*, then *q*"
- "if *p*, *q*"
- "when *p*, *q*"
- "whenever *p*, *q*"
- "*p* only if *q*" "
- *p* is sufficient for *q*"
- "*q* if *p*"

- "*q* when *p*"
- "*q* whenever *p*"
- "*q* is necessary for *p*"
- "*q* follows from *p*"
- "*q* is implied by *p*"

We will see some equivalent logic expressions later.

Converse, Inverse, Contrapositive

Some terminology, for an implication $p \rightarrow q$:

- Its *converse* is:
- Its *inverse* is:
- Its *contrapositive*:
- One of these three has the *same meaning* (same truth table) as $p \rightarrow q$. Can you figure out which?

Contrapositive

How do we know for sure?

Proving the equivalence of $p \rightarrow q$ and its contrapositive using truth tables:

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

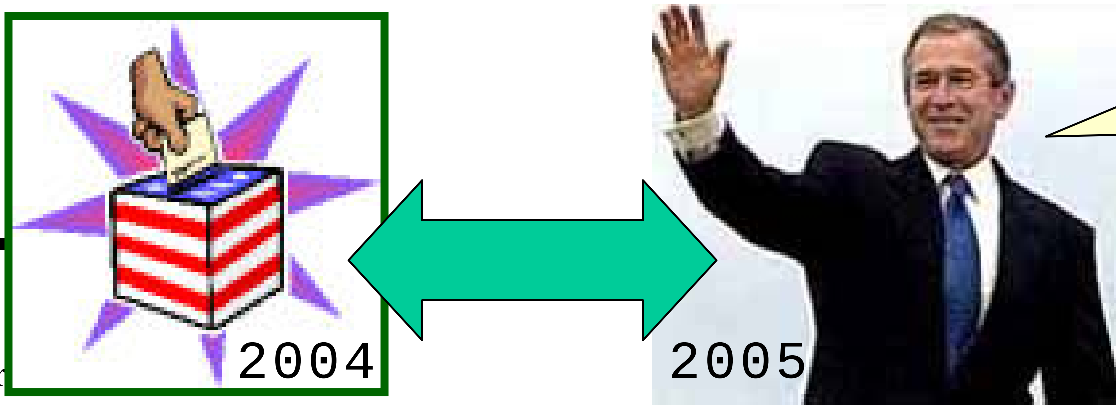
The *biconditional* operator

The *biconditional* $p \leftrightarrow q$ states that p is true *if and only if (IFF)* q is true.

p = “Bush wins the 2004 election.”

q = “Bush will be president for all of 2005.”

$p \leftrightarrow q$ = “If, and only if, Bush wins the 2004 election, Bush will be president for all of 2005.”



Biconditional Truth Table

- $p \leftrightarrow q$ means that p and q have the **same** truth value.
- Note this truth table is the exact **opposite** of $\oplus$'s!
  - $p \leftrightarrow q$ means $\neg(p \oplus q)$
- $p \leftrightarrow q$ does **not** imply p and q are true, or cause each other.

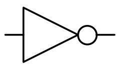
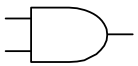
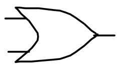
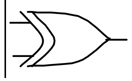
p	q	$p \leftrightarrow q$
F	F	
F	T	
T	F	
T	T	

Boolean Operations Summary

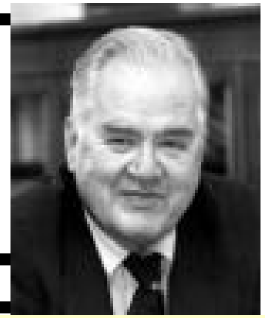
- We have seen 1 unary operator (out of the 4 possible) and 5 binary operators (out of the 16 possible). Their truth tables are below.

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \rightarrow q$	$p \leftrightarrow q$
F	F	T	F	F	F	T	T
F	T	T	F	T	T	T	F
T	F	F	F	T	T	F	F
T	T	F	T	T	F	T	T

Some Alternative Notations

Name:	not	and	or	xor	implies	iff
Propositional logic:	$\neg$	$\wedge$	$\vee$	$\oplus$	$\rightarrow$	$\leftrightarrow$
Boolean algebra:	$\bar{p}$	pq	$+$	$\oplus$		
C/C++/Java (wordwise):	!	&&	||	!=		==
C/C++/Java (bitwise):	~	&	|	^		
Logic gates:						

Bits and Bit Operations



John Tukey (1915-2000)

- A *bit* is a binary (base 2) digit: 0 or 1.
- Bits may be used to represent truth values.
- By convention:
  0 represents "false"; 1 represents "true".
- *Boolean algebra* is like ordinary algebra except that variables stand for bits, + means "or", and multiplication means "and".
  - See chapter 10 for more details.

Bit Strings

- A *Bit string* of *length n* is an ordered series or sequence of $n \geq 0$ bits.
  - More on sequences in §2.4.
- By convention, bit strings are written left to right: *e.g.* the first bit of "1001101010" is 1.
- When a bit string represents a base-2 number, by convention the first bit is the *most significant* bit. *Ex.* $1101_2 = 8+4+1=13$.

Counting in Binary

- Did you know that you can count to 1,023 just using two hands?
  - How? Count in binary!
    - Each finger (up/down) represents 1 bit.
- To increment: Flip the rightmost (low-order) bit.
  - If it changes 1→0, then also flip the next bit to the left,
    - If that bit changes 1→0, then flip the next one, *etc.*
- 0000000000, 0000000001, 0000000010, …
  …, 1111111101, 1111111110, 1111111111

Bitwise Operations

- Boolean operations can be extended to operate on bit strings as well as single bits.
- E.g.:
  01 1011 0110
  11 0001 1101

  Bit-wise OR
  Bit-wise AND
  Bit-wise XOR

End of §1.1

You have learned about:

- Propositions: What they are.
- Propositional logic operators'
  - Symbolic notations.
  - English equivalents.
  - Logical meaning.
  - Truth tables.
- Atomic vs. compound propositions.
- Alternative notations.
- Bits and bit-strings.
- Next section: §1.2
  - Propositional equivalences.
  - How to prove them.

Propositional Equivalence (§1.2)

Two *syntactically* (*i.e.*, textually) different compound propositions may be the *semantically* identical (*i.e.*, have the same meaning). We call them *equivalent*. Learn:

- Various *equivalence rules* or *laws*.
- How to *prove* equivalences using *symbolic derivations*.

Tautologies and Contradictions

A ***tautology*** is a compound proposition that is **true** *no matter what* the truth values of its atomic propositions are!

Ex. $p \lor \neg p$ [What is its truth table?]

A ***contradiction*** is a compound proposition that is **false** no matter what! *Ex.* $p \land \neg p$ [Truth table?]

Other compound props. are ***contingencies***.

Logical Equivalence

- Compound proposition p is *logically equivalent* to compound proposition q, written **$p \Leftrightarrow q$**, **IFF** the compound proposition **$p \leftrightarrow q$ is a tautology**.
- Compound propositions p and q are logically equivalent to each other **IFF *p* and *q* contain the same truth values** as each other in all rows of their truth tables.

Proving Equivalence via Truth Tables

Ex. Prove that $p \vee q \Leftrightarrow \neg(\neg p \wedge \neg q)$.

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

Equivalence Laws

- These are similar to the arithmetic identities you may have learned in algebra, but for propositional equivalences instead.
- They provide a pattern or template that can be used to match all or part of a much more complicated proposition and to find an equivalence for it.

Equivalence Laws - Examples

- *Identity*: $p \wedge \mathbf{T} \Leftrightarrow$ $p \vee \mathbf{F} \Leftrightarrow$
- *Domination*: $p \vee \mathbf{T} \Leftrightarrow$ $p \wedge \mathbf{F} \Leftrightarrow$
- *Idempotent*: $p \vee p \Leftrightarrow$ $p \wedge p \Leftrightarrow$
- *Double negation*: $\neg\neg p \Leftrightarrow$
- *Commutative*: $p \vee q \Leftrightarrow q \vee p$ $p \wedge q \Leftrightarrow q \wedge p$
- *Associative*: $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$
  $(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$

More Equivalence Laws

- *Distributive*: $p\lor(q\land r) \Leftrightarrow$
  $p\land(q\lor r) \Leftrightarrow$
- *De Morgan's*:
  $\neg(p\land q) \Leftrightarrow$
  $\neg(p\lor q) \Leftrightarrow$
- *Trivial tautology/contradiction*:
  $p \lor \neg p \Leftrightarrow$ $p \land \neg p \Leftrightarrow$

Augustus De Morgan (1806-1871)

Defining Operators via Equivalences

Using equivalences, we can *define* operators in terms of other operators.

- Exclusive or: $p \oplus q \Leftrightarrow (p \vee q) \wedge \neg(p \wedge q)$
  $p \oplus q \Leftrightarrow (p \wedge \neg q) \vee (q \wedge \neg p)$
- Implies: $p \rightarrow q \Leftrightarrow$
- Biconditional: $p \leftrightarrow q \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$
  $p \leftrightarrow q \Leftrightarrow$

An Example Problem

- Check using a symbolic derivation whether $(p \wedge \neg q) \rightarrow (p \oplus r) \Leftrightarrow \neg p \vee q \vee \neg r$.

$(p \wedge \neg q) \rightarrow (p \oplus r) \Leftrightarrow$

[Expand definition of $\rightarrow$]

[Defn. of $\oplus$] $\Leftrightarrow \neg(p \wedge \neg q) \vee ((p \vee r) \wedge \neg(p \wedge r))$

[DeMorgan's Law]

$\Leftrightarrow \qquad \vee ((p \vee r) \wedge \neg(p \wedge r))$

$\Leftrightarrow$ [associative law] *cont.*

Example Continued...

$(\neg \mathbf{p} \vee \mathbf{q}) \vee ((p \vee r) \wedge \neg(p \wedge r)) \Leftrightarrow$ [$\vee$ commutes]

$\Leftrightarrow \quad \vee ((p \vee r) \wedge \neg(p \wedge r))$ [$\vee$ associative]

$\Leftrightarrow \mathbf{q} \vee (\underline{\neg \mathbf{p} \vee} ((p \vee r) \wedge \neg(p \wedge r)))$ [distrib. $\vee$ over $\wedge$]

$\Leftrightarrow q \vee ((\underline{\neg \mathbf{p} \vee} (\mathbf{p} \vee \mathbf{r})) \wedge (\underline{\neg \mathbf{p} \vee} \neg(\mathbf{p} \wedge \mathbf{r})))$

[assoc.] $\Leftrightarrow q \vee ((\quad) \wedge (\quad))$

[trivail taut.] $\Leftrightarrow q \vee ((\quad) \wedge (\neg p \vee \neg(p \wedge r)))$

[domination] $\Leftrightarrow q \vee (\quad \wedge (\neg p \vee \neg(p \wedge r)))$

[identity] $\Leftrightarrow q \vee (\neg \mathbf{p} \vee \neg(\mathbf{p} \wedge \mathbf{r})) \Leftrightarrow$ *cont.*

End of Long Example

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

[DeMorgan's] $\Leftrightarrow q \vee (\neg p \vee (\qquad))$

[Assoc.] $\Leftrightarrow q \vee ((\neg p \vee \neg p) \vee \neg r)$

[Idempotent] $\Leftrightarrow q \vee (\qquad \vee \neg r)$

[Assoc.] $\Leftrightarrow (q \vee \neg p) \vee \neg r$

[Commut.] $\Leftrightarrow \neg p \vee q \vee \neg r$

Q.E.D. (quod erat demonstrandum)

(Which was to be shown.)

Review: Propositional Logic (§1.1-1.2)

- Atomic propositions: p, q, r, …
- Boolean operators: $\neg \wedge \vee \oplus \rightarrow \leftrightarrow$
- Compound propositions: $s :\equiv (p \wedge \neg q) \vee r$
- Equivalences: $p \wedge \neg q \Leftrightarrow \neg(p \rightarrow q)$
- Proving equivalences using:
  - Truth tables.
  - Symbolic derivations. $p \Leftrightarrow q \Leftrightarrow r$ …

Predicate Logic (§1.3)

- ***Predicate logic*** is an extension of propositional logic that permits concisely reasoning about whole *classes* of entities.
- Propositional logic (recall) treats simple *propositions* (sentences) as atomic entities.
- In contrast, *predicate* logic distinguishes the ***subject*** of a sentence from its ***predicate***.
  - Remember these English grammar terms?

Applications of Predicate Logic

It is *the* formal notation for writing perfectly clear, concise, and unambiguous mathematical *definitions*, *axioms*, and *theorems* (more on these in chapter 3) for *any* branch of mathematics.

Predicate logic with function symbols, the "=" operator, and a few proof-building rules is sufficient for defining *any* conceivable mathematical system, and for proving anything that can be proved within that system!

Other Applications

Kurt Gödel
1906-1978

- Predicate logic is the foundation of the field of *mathematical logic*, which culminated in *Gödel's incompleteness theorem*, which revealed the ultimate limits of mathematical thought:
  - Given any finitely describable, consistent proof procedure, there will still be *some* true statements that can *never be proven* by that procedure.
- *I.e.*, we can't discover *all* mathematical truths, unless we sometimes resort to making *guesses*.

Practical Applications

- Basis for clearly expressed formal specifications for any complex system.
- Basis for *automatic theorem provers* and many other Artificial Intelligence systems.
- Supported by some of the more sophisticated *database query engines* and *container class libraries* (these are types of programming tools).

Subjects and Predicates

- In the sentence "The dog is sleeping":
  - The phrase "**the dog**" denotes the ***subject*** - the *object* or *entity* that the sentence is about.
  - The phrase "**is sleeping**" denotes the ***predicate***- a property that is true **of** the subject.
- In predicate logic, a ***predicate*** is modeled as a *function* $P(\cdot)$ from objects to propositions.
  - $P(x)$ = "x is sleeping" (where x is any object).

More About Predicates

- Convention: Lowercase variables ***x*, *y*, *z*... denote objects/entities**; uppercase variables ***P*, *Q*, *R*… denote propositional functions (predicates)**.
- Keep in mind that the *result of applying* a predicate P to an object x is the *proposition* $P(x)$. But the predicate P **itself** (*e.g.* P=“is sleeping”) is **not** a proposition (not a complete sentence).
  - *E.g.* if $P(x)$ = “x is a prime number”, $P(3)$ is the *proposition* “3 is a prime number.”

Propositional Functions

- Predicate logic *generalizes* the grammatical notion of a predicate to also include propositional functions of **any** number of arguments, each of which may take **any** grammatical role that a noun can take.
  - *E.g.* let $P(x,y,z)$ = "x gave y the grade z", then if x="Mike", y="Mary", z="A", then $P(x,y,z)$ = "Mike gave Mary the grade A."

Universes of Discourse (U.D.s)

- The power of distinguishing objects from predicates is that it lets you state things about *many* objects at once.
- E.g., let $P(x)$="$x+1>x$". We can then say, "For *any* number x, $P(x)$ is true" instead of $(\mathbf{0}+1>\mathbf{0}) \wedge (\mathbf{1}+1>\mathbf{1}) \wedge (\mathbf{2}+1>\mathbf{2}) \wedge \ldots$
- The collection of values that a variable x can take is called ***x's universe of discourse***.

Quantifier Expressions

- ***Quantifiers*** provide a notation that allows us to *quantify* (count) *how many* objects in the univ. of disc. satisfy a given predicate.
- "$\forall$" is the FOR$\forall$LL or *universal* quantifier. $\forall x\ P(x)$ means *for all* x in the u.d., P holds.
- "$\exists$" is the $\exists$XISTS or *existential* quantifier. $\exists x\ P(x)$ means there *exists* an x in the u.d. (that is, 1 or more) such that $P(x)$ is true.

The Universal Quantifier ∀

- Example:
  Let the u.d. of *x* be parking spaces at UF.
  Let *P*(*x*) be the *predicate* "*x* is full."
  Then the *universal quantification of P*(*x*), ∀*x P*(*x*), is the *proposition*:
  - "All parking spaces at UF are full."
  - *i.e.*, "Every parking space at UF is full."
  - *i.e.*, "For each parking space at UF, that space is full."

The Existential Quantifier ∃

- Example:
  Let the u.d. of x be parking spaces at UF.
  Let $P(x)$ be the *predicate* "x is full."
  Then the *existential quantification of* $P(x)$, $\exists x\, P(x)$, is the *proposition*:
  - "Some parking space at UF is full."
  - "There is a parking space at UF that is full."
  - "At least one parking space at UF is full."

Free and Bound Variables

- An expression like $P(x)$ is said to have a ***free variable x*** (meaning, x is undefined).
- A quantifier (either ∀ or ∃) *operates* on an expression having one or more free variables, and *binds* one or more of those variables, to produce an expression having one or more ***bound variables***.

Example of Binding

- $P(x,y)$ has 2 free variables, x and y.
- $\forall x\, P(x,y)$ has 1 free variable, and one bound variable. [Which is which?]
- "$P(x)$, where $x=3$" is another way to bind x.
- An expression with zero free variables is a bona-fide (actual) proposition.
- An expression with one or more free variables is still only a predicate: $\forall x\, P(x,y)$

Nesting of Quantifiers

Example: Let the u.d. of x & y be people.

Let $L(x,y)$=“x likes y” (a predicate w. 2 f.v.’s)

Then $\exists y\, L(x,y)$ = “There is someone whom x likes.” (A predicate w. 1 free variable, x)

Then $\forall x\, (\exists y\, L(x,y))$ =
“Everyone has someone whom they like.”
(A Proposition with ___ free variables.)

Review: Predicate Logic (§1.3)

- Objects x, y, z, …
- Predicates P, Q, R, … are functions mapping objects x to propositions $P(x)$.
- Multi-argument predicates $P(x, y)$.
- Quantifiers: $[\forall x\ P(x)] :\equiv$ "For all x's, $P(x)$."
  $[\exists x\ P(x)] :\equiv$ "There is an x such that $P(x)$."
- Universes of discourse, bound & free vars.

Quantifier Exercise

If $R(x,y)$="x relies upon y," express the following in unambiguous English:

$\forall x(\exists y\ R(x,y))$= Everyone has *someone* to rely on.

$\exists y(\forall x\ R(x,y))$= There's a poor overburdened soul whom *everyone* relies upon (including himself)!

$\exists x(\forall y\ R(x,y))$= There's some needy person who relies upon *everybody* (including himself).

$\forall y(\exists x\ R(x,y))$= Everyone has *someone* who relies upon them.

$\forall x(\forall y\ R(x,y))$= *Everyone* relies upon *everybody*, (including themselves)!

Natural language is ambiguous!

- "Everybody likes somebody."
  - For everybody, there is somebody they like,
    - $\forall x\, \exists y\, Likes(x,y)$ [Probably more likely.]
  - or, there is somebody (a popular person) whom everyone likes?
    - $\exists y\, \forall x\, Likes(x,y)$
- "Somebody likes everybody."
  - Same problem: Depends on context, emphasis.

Game Theoretic Semantics

- Thinking in terms of a competitive game can help you tell whether a proposition with nested quantifiers is true.
- The game has two players, both with the same knowledge:
  - Verifier: Wants to demonstrate that the proposition is true.
  - Falsifier: Wants to demonstrate that the proposition is false.
- The Rules of the Game "Verify or Falsify":
  - Read the quantifiers from left to right, picking values of variables.
  - When you see "∀", the falsifier gets to select the value.
  - When you see "∃", the verifier gets to select the value.
- If the verifier can always win, then the proposition is true.
- If the falsifier can always win, then it is false.

Let's Play, "Verify or Falsify!"

Let $B(x,y) :\equiv$ "x's birthday is followed within 7 days by y's birthday."

Suppose I claim that among you:

$\underline{\forall x}\ \exists y\ B(x,y)$

Your turn, as falsifier:
You pick any x → *(so-and-so)*

$\underline{\exists y}\ B(\text{so-and-so},y)$

My turn, as verifier:
I pick any y → *(such-and-such)*

$B(\text{so-and-so},\text{such-and-such})$

- Let's play it in class.
- Who wins this game?
- What if I switched the quantifiers, and I claimed that

$$\exists y\ \forall x\ B(x,y)?$$

Who wins in that case?

Still More Conventions

- Sometimes the universe of discourse is restricted within the quantification, *e.g.*,
  - $\forall x>0\ P(x)$ is shorthand for "For all x that are greater than zero, $P(x)$."
    $=\forall x\ ($ $)$
  - $\exists x>0\ P(x)$ is shorthand for "There is an x greater than zero such that $P(x)$."
    $=\exists x\ ($ $)$

More to Know About Binding

- $\forall x\, \exists x\, P(x)$ - x is not a free variable in $\exists x\, P(x)$, therefore the $\forall x$ binding <u>isn't used</u>.
- $(\forall x\, P(x)) \wedge Q(x)$ - The variable x is outside of the *scope* of the $\forall x$ quantifier, and is therefore free. Not a proposition!
- $(\forall x\, P(x)) \wedge (\exists x\, Q(x))$ – This is legal, because there are 2 <u>different</u> x's!

Quantifier Equivalence Laws

- Definitions of quantifiers: If u.d.=a,b,c,…
  $\forall x\ P(x) \Leftrightarrow P(\text{a}) \wedge P(\text{b}) \wedge P(\text{c}) \wedge \ldots$
  $\exists x\ P(x) \Leftrightarrow P(\text{a}) \vee P(\text{b}) \vee P(\text{c}) \vee \ldots$
- From those, we can prove the laws:
  $\forall x\ P(x) \Leftrightarrow$
  $\exists x\ P(x) \Leftrightarrow$
- Which *propositional* equivalence laws can be used to prove this?

More Equivalence Laws

- $\forall x\ \forall y\ P(x,y) \Leftrightarrow \forall y\ \forall x\ P(x,y)$
  $\exists x\ \exists y\ P(x,y) \Leftrightarrow \exists y\ \exists x\ P(x,y)$
- $\forall x\ (P(x) \wedge Q(x)) \Leftrightarrow (\forall x\ P(x)) \wedge (\forall x\ Q(x))$
  $\exists x\ (P(x) \vee Q(x)) \Leftrightarrow (\exists x\ P(x)) \vee (\exists x\ Q(x))$
- Exercise:
  See if you can prove these yourself.
  – What propositional equivalences did you use?

Review: Predicate Logic (§1.3)

- Objects x, y, z, …
- Predicates P, Q, R, … are functions mapping objects x to propositions $P(x)$.
- Multi-argument predicates $P(x, y)$.
- Quantifiers: $(\forall x\ P(x))$ =“For all x’s, $P(x)$.” $(\exists x\ P(x))$=“There is an x such that $P(x)$.”

More Notational Conventions

- Quantifiers bind as loosely as needed: parenthesize $\forall x(P(x) \wedge Q(x))$
- Consecutive quantifiers of the same type can be combined: $\forall x\ \forall y\ \forall z\ P(x,y,z) \Leftrightarrow \forall x,y,z\ P(x,y,z)$ or even $\forall xyz\ P(x,y,z)$
- All quantified expressions can be reduced to the canonical *alternating* form $\forall x_1 \exists x_2 \forall x_3 \exists x_4 \ldots P(x_1, x_2, x_3, x_4, \ldots)$

Defining New Quantifiers

As per their name, quantifiers can be used to express that a predicate is true of any given *quantity* (number) of objects.

Define $\exists! x\, P(x)$ to mean "$P(x)$ is true of *exactly one* x in the universe of discourse."

$\exists! x\, P(x) \Leftrightarrow \exists x\, \big(P(x) \wedge \neg\exists y\, (P(y) \wedge y \neq x)\big)$

"There is an x such that $P(x)$, where there is no y such that P(y) and y is other than x."

Some Number Theory Examples

- Let u.d. = the *natural numbers* 0, 1, 2, …
- "A number x is *even*, $E(x)$, if and only if it is equal to 2 times some other number."
  $\forall x\, (E(x) \leftrightarrow (\exists y\;\; x=2y))$
- "A number is *prime*, $P(x)$, iff it's greater than 1 and it isn't the product of two non-unity numbers."
  $\forall x\, (P(x) \leftrightarrow (x>1 \land \neg\exists yz\;\; x=yz \land y\neq 1 \land z\neq 1))$

Goldbach's Conjecture (unproven)

Using $E(x)$ and $P(x)$ from previous slide,

$\forall E(x>2): \exists P(p),P(q): p+q = x$

or, with more explicit notation:

$\forall x\ [x>2 \wedge E(x)] \rightarrow$

$\exists p\ \exists q\ P(p) \wedge P(q) \wedge p+q = x.$

"Every even number greater than 2 is the sum of two primes."

Calculus Example

- One way of precisely defining the calculus concept of a *limit*, using quantifiers:

$$\left(\lim_{x\to a} f(x) = L\right) \Leftrightarrow$$

$$\left(\begin{array}{l}\forall \varepsilon > 0 : \exists \delta > 0 : \forall x : \\ \left(|x-a| < \delta\right) \rightarrow \left(|f(x)-L| < \varepsilon\right)\end{array}\right)$$

Deduction Example

- Definitions:
  s :≡ Socrates (ancient Greek philosopher);
  $H(x)$:≡ "x is human";
  $M(x)$:≡ "x is mortal".
- Premises:
  $H(\text{s})$ *Socrates is human.*
  $\forall x\, H(x) \rightarrow M(x)$ *All humans are mortal.*

Deduction Example Continued

Some valid conclusions you can draw:

$H(s) \rightarrow M(s)$ **[Instantiate universal.]** *If Socrates is human then he is mortal.*

$\neg H(s) \vee M(s)$ *Socrates is inhuman or mortal.*

$H(s) \wedge (\neg H(s) \vee M(s))$

Socrates is human, and also either inhuman or mortal.

$(H(s) \wedge \neg H(s)) \vee (H(s) \wedge M(s))$ **[Apply distributive law.]**

$\mathbf{F} \vee (H(s) \wedge M(s))$ **[Trivial contradiction.]**

$H(s) \wedge M(s)$ **[Use identity law.]**

$M(s)$ *Socrates is mortal.*

Another Example

- Definitions: $H(x) :\equiv$ "x is human"; $M(x) :\equiv$ "x is mortal"; $G(x) :\equiv$ "x is a god"
- Premises:
  - $\forall x\, H(x) \rightarrow M(x)$ ("Humans are mortal") and
  - $\forall x\, G(x) \rightarrow \neg M(x)$ ("Gods are immortal").
- Show that $\neg\exists x\, (H(x) \wedge G(x))$ ("No human is a god.")

The Derivation

- $\forall x\ H(x) \rightarrow M(x)$ and $\forall x\ G(x) \rightarrow \neg M(x)$.
- $\forall x\ \neg M(x) \rightarrow$ **[Contrapositive.]**
- $\forall x\ [G(x) \rightarrow \neg M(x)] \wedge [\neg M(x) \rightarrow \neg H(x)]$
- $\forall x\ G(x) \rightarrow$ **[Transitivity of →.]**
- $\forall x$ **[Definition of →.]**
- $\forall x$ **[DeMorgan's law.]**
- $\neg \exists x\ G(x) \wedge H(x)$ **[An equivalence law.]**

End of §1.3-1.4, Predicate Logic

- From these sections you should have learned:
  - Predicate logic notation & conventions
  - Conversions: predicate logic ↔ clear English
  - Meaning of quantifiers, equivalences
  - Simple reasoning with quantifiers
- Upcoming topics:
  - Introduction to proof-writing.
  - Then: Set theory –
    - a language for talking about collections of objects.

§1.5-1.7 : Basic Proof Methods

Nature & Importance of Proofs

- In mathematics, a ***proof*** is:
  - a ***correct*** (well-reasoned, logically valid) and ***complete*** (clear, detailed) argument that rigorously & undeniably establishes the truth of a mathematical statement.
- Why must the argument be correct & complete?
  - ***Correctness*** prevents us from fooling ourselves.
  - ***Completeness*** allows anyone to verify the result.
- In this course (& throughout mathematics), a very high standard for correctness and completeness of proofs is demanded!!

Overview of §1.5 -1.7

- Methods of mathematical argument (*i.e.*, proof methods) can be formalized in terms of ***rules of logical inference***.
- Mathematical *proofs* can themselves be represented formally as discrete structures.
- We will review both **correct** & **fallacious** inference rules, & several proof methods.

Applications of Proofs

- An exercise in clear communication of logical arguments in any area of study.
- The fundamental activity of mathematics is the discovery and elucidation, through proofs, of interesting new theorems.
- Theorem-proving has applications in program verification, computer security, automated reasoning systems, *etc.*
- Proving a theorem allows us to rely upon on its correctness even in the most critical scenarios.

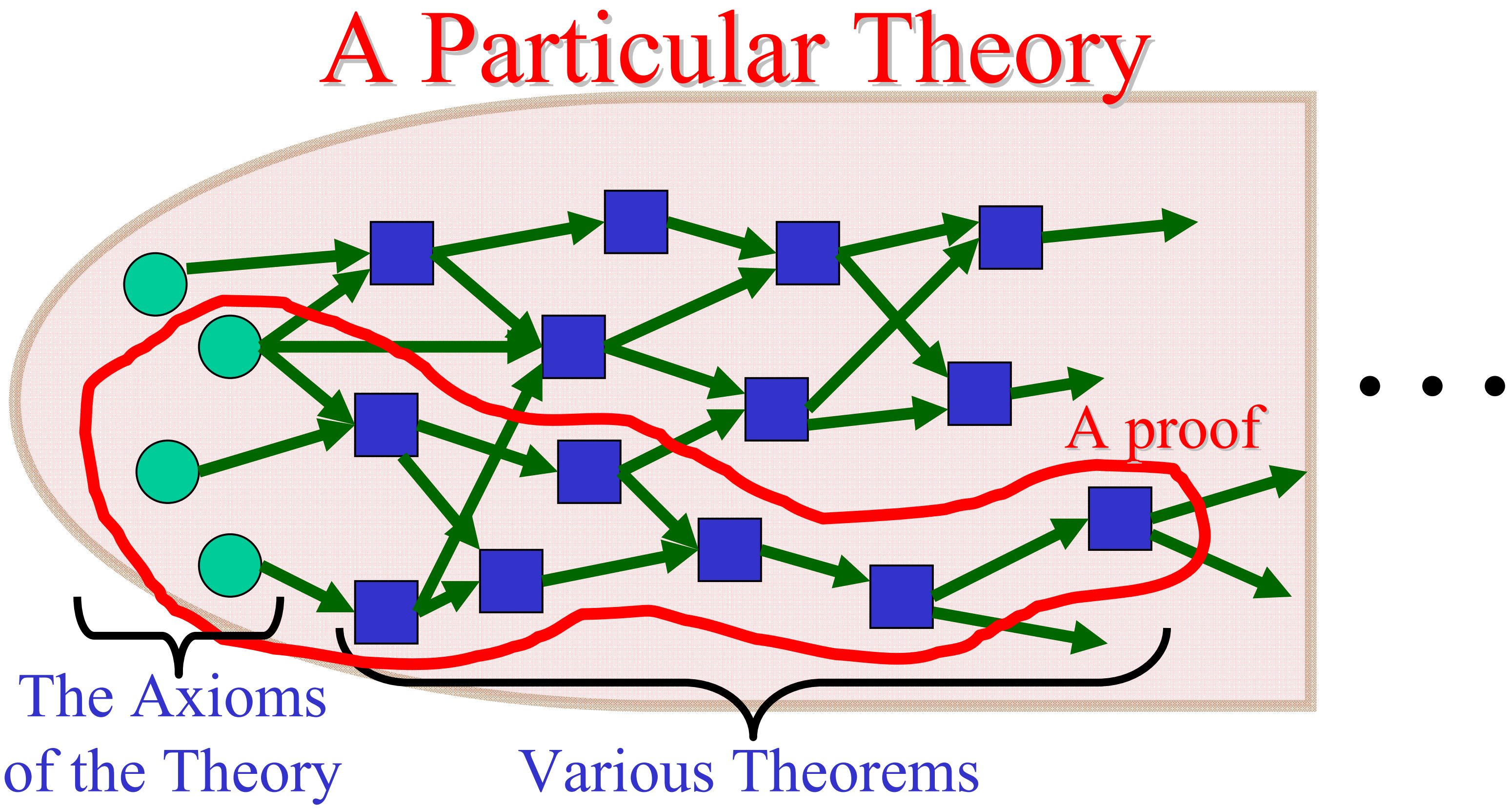
Proof Terminology

- ***Theorem***
  - A statement that has been proven to be true.
- ***Axioms**, **postulates, hypotheses, premises***
  - Assumptions (often unproven) defining the structures about which we are reasoning.
- ***Rules of inference***
  - Patterns of logically valid deductions from hypotheses to conclusions.

More Proof Terminology

- ***Lemma*** - A minor theorem used as a stepping-stone to proving a major theorem.
- ***Corollary*** - A minor theorem proved as an easy consequence of a major theorem.
- ***Conjecture*** - A statement whose truth value has not been proven. (A conjecture may be widely believed to be true, regardless.)
- ***Theory*** – The set of all theorems that can be proven from a given set of axioms.

Graphical Visualization



Inference Rules - General Form

- ***Inference Rule*** –
  - Pattern establishing that if we know that a set of *antecedent* statements of certain forms are all true, then a certain related *consequent* statement is true.
- *antecedent 1*
  antecedent 2 …
  $\therefore$ *consequent*

"$\therefore$" means "therefore"

Inference Rules & Implications

- Each logical inference rule corresponds to an implication that is a tautology.
- Inference rule

$$\begin{array}{l} \textit{antecedent 1} \\ \underline{\textit{antecedent 2} \;\ldots} \\ \therefore\ \textit{consequent} \end{array}$$

- Corresponding tautology:

$$((\textit{ante. 1}) \land (\textit{ante. 2}) \land \ldots) \rightarrow \textit{consequent}$$

Some Inference Rules

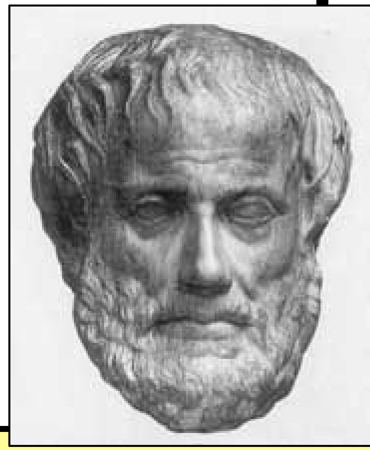
- $$\frac{p}{\therefore\ p \vee q}$$ Rule of Addition
- $$\frac{p \wedge q}{\therefore\ p}$$ Rule of Simplification
- $$\frac{\begin{matrix} p \\ q \end{matrix}}{\therefore\ p \wedge q}$$ Rule of Conjunction

Modus Ponens & Tollens

- p
  $\underline{p \rightarrow q}$
  $\therefore q$
  Rule of *modus ponens* (a.k.a. *law of detachment*)
  "the mode of affirming"
- $\neg q$
  $\underline{p \rightarrow q}$
  $\therefore \neg p$
  Rule of *modus tollens*
  "the mode of denying"

Syllogism Inference Rules

- $$\begin{array}{c} p \rightarrow q \\ q \rightarrow r \\ \hline \therefore p \rightarrow r \end{array}$$ Rule of hypothetical syllogism
- $$\begin{array}{c} p \vee q \\ \neg p \\ \hline \therefore q \end{array}$$ Rule of disjunctive syllogism



Aristotle
(ca. 384-322 B.C.)

Formal Proofs

- A formal proof of a conclusion *C*, given premises $p_1, p_2, \ldots, p_n$ consists of a sequence of *steps*, each of which applies some inference rule to premises or to previously-proven statements (as antecedents) to yield a new true statement (the consequent).
- A proof demonstrates that *if* the premises are true, *then* the conclusion is true.

Formal Proof Example

- Suppose we have the following premises:
  "It is not sunny and it is cold."
  "We will swim(*p*) only if it is sunny(*q*)."(*p* -->*q*)
  "If we do not swim, then we will canoe."
  "If we canoe, then we will be home early."
- Given these premises, prove the theorem **"We will be home early"** using inference rules.

Proof Example *cont.*

- Let us adopt the following abbreviations:
  - *sunny* = **"It is sunny"**; *cold* = **"It is cold"**; *swim* = **"We will swim"**; *canoe* = **"We will canoe"**; *early* = **"We will be home early"**.
- Then, the premises can be written as:
  (1) $\neg sunny \wedge cold$ (2) $swim \rightarrow sunny$
  (3) $\neg swim \rightarrow canoe$ (4) $canoe \rightarrow early$

Proof Example *cont.*

Step	Proved by
1. $\neg sunny \wedge cold$	Premise #1.
2. $\neg sunny$	Simplification of 1.
3. $swim \rightarrow sunny$	Premise #2.
4.	Modus tollens on 2,3.
5. $\neg swim \rightarrow canoe$	Premise #3.
6.	Modus ponens on 4,5.
7. $canoe \rightarrow early$	Premise #4.
8.	Modus ponens on 6,7.

Inference Rules for Quantifiers

- $\dfrac{\forall x\, P(x)}{\therefore P(o)}$ **Universal instantiation** (substitute *any* object *o*)
- $\dfrac{P(g)}{\therefore \forall x\, P(x)}$ (for *g* a *general* element of u.d.) **Universal generalization**
- $\dfrac{\exists x\, P(x)}{\therefore P(c)}$ **Existential instantiation** (substitute a *new constant c*)
- $\dfrac{P(o)}{\therefore \exists x\, P(x)}$ (substitute any extant object *o*) **Existential generalization**

Common Fallacies

- A *fallacy* is an inference rule or other proof method that is not logically valid.
  - May yield a false conclusion!
- Fallacy of *affirming the conclusion*:
  - "$p \rightarrow q$ is true, and q is true, so p must be true." (No, because **F**$\rightarrow$**T** is true.)
- Fallacy of *denying the hypothesis*:
  - "$p \rightarrow q$ is true, and p is false, so q must be false." (No, again because **F**$\rightarrow$**T** is true.)

Circular Reasoning

- The fallacy of (explicitly or implicitly) assuming the very statement you are trying to prove in the course of its proof. Example:
- Prove that an integer n is even, if n^2 is even.
- Attempted proof: "Assume n^2 is even. Then $n^2=2k$ for some integer k. Dividing both sides by n gives $n = (2k)/n = 2(k/n)$. So there is an integer j (namely k/n) such that $n=2j$. Therefore n is even."

Begs the question: How do you show that $j=k/n=n/2$ is an integer, without first assuming n is even?

Removing the Circularity

Suppose n^2 is even $\therefore 2|n^2 \therefore n^2 \bmod 2 = 0$. Of course $n \bmod 2$ is either 0 or 1. If it's 1, then $n \equiv 1 \pmod 2$, so $n^2 \equiv 1 \pmod 2$, using the theorem that if $a \equiv b \pmod m$ and $c \equiv d \pmod m$ then $ac \equiv bd$ (mod m), with $a=c=n$ and $b=d=1$. Now $n^2 \equiv 1 \pmod 2$ implies that $n^2 \bmod 2 = 1$. So by the hypothetical syllogism rule, $(n \bmod 2 = 1)$ implies $(n^2 \bmod 2 = 1)$. Since we know $n^2 \bmod 2 = 0 \neq 1$, by *modus tollens* we know that $n \bmod 2 \neq 1$. So by disjunctive syllogism we have that $n \bmod 2 = 0 \therefore 2|n \therefore n$ is even.

Proof Methods for Implications

For proving implications $p \rightarrow q$, we have:

- ***Direct*** proof: Assume p is true, and prove q.
- ***Indirect*** proof: Assume $\neg q$, and prove $\neg p$.
- ***Vacuous*** proof: Prove $\neg p$ by itself.
- ***Trivial*** proof: Prove q by itself.
- **Proof by cases**:
  Show $p \rightarrow (a \vee b)$, and $(a \rightarrow q)$ and $(b \rightarrow q)$.

Direct Proof Example

- **Definition:** An integer n is called *odd* iff $n=2k+1$ for some integer k; n is *even* iff $n=2k$ for some k.
- **Axiom:** Every integer is either odd or even.
- **Theorem:** (For all numbers n) If n is an odd integer, then n^2 is an odd integer.
- **Proof:**

Indirect Proof Example

- **Theorem:** (For all integers n)
  If $3n+2$ is odd, then n is odd.
- **Proof:**

Vacuous Proof Example

- **Theorem:** (For all n) If n is both odd and even, then $n^2 = n + n$.
- **Proof:** The statement "n is both odd and even" is necessarily false, since no number can be both odd and even. So, the theorem is vacuously true. □

Trivial Proof Example

- **Theorem:** (For integers *n*) If *n* is the sum of two prime numbers, then either *n* is odd or *n* is even.
- **Proof:** Any integer *n* is either odd or even. So the conclusion of the implication is true regardless of the truth of the antecedent. Thus the implication is true trivially. □

Proof by Contradiction

- A method for proving p.
- Assume $\neg p$, and prove both q and $\neg q$ for some proposition q.
- Thus $\neg p \rightarrow (q \wedge \neg q)$
- $(q \wedge \neg q)$ is a trivial contradition, equal to **F**
- Thus $\neg p \rightarrow \mathbf{F}$, which is only true if $\neg p = \mathbf{F}$
- Thus p is true.

Review: Proof Methods So Far

- ***Direct, indirect, vacuous***, and ***trivial*** proofs of statements of the form $p \rightarrow q$.
- ***Proof by contradiction*** of any statements.
- Next: ***Constructive*** and ***nonconstructive existence proofs***.

Proving Existentials

- A proof of a statement of the form $\exists x\, P(x)$ is called an ***existence proof***.
- If the proof demonstrates how to actually find or construct a specific element a such that $P(a)$ is true, then it is a ***constructive*** proof.
- Otherwise, it is *nonconstructive*.

Constructive Existence Proof

- **Theorem:** There exists a positive integer n that is the sum of two perfect cubes in two different ways:
  - equal to $j^3 + k^3$ and $l^3 + m^3$ where j, k, l, m are positive integers, and $\{j,k\} \neq \{l,m\}$
- **Proof:**

Another Constructive Existence Proof

- **Theorem:** For any integer $n>0$, there exists a sequence of n consecutive composite integers.
- Same statement in predicate logic: $\forall n>0\ \exists x\ \forall i\ (1\leq i\leq n)\rightarrow(x+i \text{ is composite})$
- Proof follows on next slide…

The proof...

- Given n>0, let $x = (n + 1)! + 1$.
- Let $i \geq 1$ and $i \leq n$, and consider $x+i$.
- Note $x+i =$
- Note , since $2 \leq i+1 \leq n+1$.
- Also $(i+1)|(i+1)$. So,
- $\therefore$ $x+i$ is composite.
- $\therefore$ $\forall n \, \exists x \, \forall 1 \leq i \leq n : x+i$ is composite. Q.E.D.

Nonconstructive Existence Proof

- **Theorem:**
  "There are infinitely many prime numbers."
- Any finite set of numbers must contain a maximal element, so we can prove the theorem if we can just show that there is ***no*** largest prime number.
- *I.e.*, show that for any prime number, there is a larger number that is *also* prime.
- More generally: For *any* number, ∃ a larger prime.
- Formally: Show $\forall n\ \exists p>n : p$ is prime.

The proof, using *proof by cases*...

- Given $n>0$, prove there is a prime $p>n$.
- Consider $x = n!+1$. Since $x>1$, we know (x is prime)$\vee$(x is composite).
- **Case 1:** x is prime.
- **Case 2:** x has a prime factor p.

Limits on Proofs

- Some very simple statements of number theory haven't been proved or disproved!
  - *E.g. Goldbach's conjecture*: Every integer $n \geq 2$ is exactly the average of some two primes.
  - $\forall n \geq 2\ \exists$ primes p,q: $n=(p+q)/2$.
- There are true statements of number theory (or any sufficiently powerful system) that can *never* be proved (or disproved) (Gödel).

More Proof Examples

- Quiz question 1a: Is this argument correct or incorrect?
  - "All TAs compose easy quizzes. Ramesh is a TA. Therefore, Ramesh composes easy quizzes."
- First, separate the premises from conclusions:
  - Premise #1: All TAs compose easy quizzes.
  - Premise #2: Ramesh is a TA.
  - Conclusion: Ramesh composes easy quizzes.

Answer

Next, re-render the example in logic notation.

- Premise #1: All TAs compose easy quizzes.
  - Let U.D. = all people
  - Let $T(x) :\equiv$ "x is a TA"
  - Let $E(x) :\equiv$ "x composes easy quizzes"
  - Then Premise #1 says: $\forall x, T(x) \rightarrow E(x)$

Answer cont…

- Premise #2: Ramesh is a TA.
  - Let R :≡ Ramesh
  - Then Premise #2 says: T(R)
  - And the Conclusion says: E(R)
- The argument is correct, because it can be reduced to a sequence of applications of valid inference rules, as follows:

The Proof in Gory Detail

- <u>Statement</u> <u>How obtained</u>

1. $\forall x, T(x) \rightarrow E(x)$ **(Premise #1)**
2. $T(\text{Ramesh}) \rightarrow E(\text{Ramesh})$ **(Universal instantiation)**
3. $T(\text{Ramesh})$ **(Premise #2)**
4. $E(\text{Ramesh})$ **(*Modus Ponens* from statements #2 and #3)**

Another example

- Quiz question 2b: Correct or incorrect: At least one of the 280 students in the class is intelligent. *Y* is a student of this class. Therefore, *Y* is intelligent.
- First: Separate premises/conclusion, & translate to logic:
  - Premises: (1) $\exists x\ \mathrm{InClass}(x) \wedge \mathrm{Intelligent}(x)$
    (2) $\mathrm{InClass}(Y)$
  - Conclusion: $\mathrm{Intelligent}(Y)$

Answer

- No, the argument is invalid; we can disprove it with a counter-example, as follows:
- Consider a case where there is only one intelligent student X in the class, and $X \neq Y$.
  - Then the premise $\exists x$ **InClass(x)** $\wedge$ **Intelligent(x)** is true, by existential generalization of **InClass(X)** $\wedge$ **Intelligent(X)**
  - But the conclusion **Intelligent(Y)** is false, since X is the only intelligent student in the class, and $Y \neq X$.
- Therefore, the premises *do not* imply the conclusion.

Another Example

- Quiz question #2: Prove that the sum of a rational number and an irrational number is always irrational.
- First, you have to understand exactly what the question is asking you to prove:
  - "For all real numbers x,y, if x is rational and y is irrational, then $x+y$ is irrational."
  - $\forall x,y$: Rational(x) $\wedge$ Irrational(y) $\rightarrow$ Irrational($x+y$)

Answer

- Next, think back to the definitions of the terms used in the statement of the theorem:
  - $\forall$ reals r: Rational(r) $\leftrightarrow$ $\exists$ Integer(i) $\wedge$ Integer(j): $r = i / j$.
  - $\forall$ reals r: Irrational(r) $\leftrightarrow$ $\neg$Rational(r)
- You almost always need the definitions of the terms in order to prove the theorem!
- Next, let's go through one valid proof:

What you might write

- **Theorem:**
  $\forall x, y$: Rational(x) $\wedge$ Irrational(y) $\rightarrow$ Irrational($x + y$)
- **Proof:** Let x, y be any rational and irrational numbers, respectively. … (universal generalization)
- Now, just from this, what do we know about x and y? You should think back to the definition of rational:
- … Since x is rational, we know (from the very definition of rational) that there must be some integers i and j such that $x = i / j$. So, let i_x, j_x be such integers ...
- We give them unique names so we can refer to them later.

What next?

- What do we know about y? Only that y is irrational: $\neg\exists$ integers $i, j : y = i / j$.
- But, it's difficult to see how to use a direct proof in this case. We could try indirect proof also, but in this case, it is a little simpler to just use proof by contradiction (very similar to indirect).
- So, what are we trying to show? Just that $x+y$ is irrational. That is, $\neg\exists i, j : (x + y) = i / j$.
- What happens if we hypothesize the negation of this statement?

More writing…

- Suppose that $x+y$ were not irrational. Then $x+y$ would be rational, so $\exists$ integers $i,j: x+y=i/j$. So, let i_s and j_s be any such integers where $x+y=i_s/\ j_s$.
- Now, with all these things named, we can start seeing what happens when we put them together.
- So, we have that $(i_x/j_x)+y=(\ i_s/j_s)$.
- Observe! We have enough information now that we can conclude something useful about y, by solving this equation for it.

Finishing the proof.

- Solving that equation for y, we have:

$$y =$$

$$=$$

Now, since the numerator and denominator of this expression are both integers, y is (by definition) rational. This contradicts the assumption that y was irrational. Therefore, our hypothesis that $x + y$ is rational must be false, and so the theorem is proved.

Example wrong answer

- 1 is rational. $\sqrt{2}$ is irrational. $1+\sqrt{2}$ is irrational. Therefore, the sum of a rational number and an irrational number is irrational. (Direct proof.)
- Why does this answer merit no credit?
  - The student attempted to use an example to prove a universal statement. **This is always wrong!**
  - Even as an example, it's incomplete, because the student never even proved that $1+\sqrt{2}$ is irrational!

Proofs of Equivalence

- How to prove "$p \leftrightarrow q$", i.e., " p if and only if q"?
  - You must prove "$p \rightarrow q$" and "$q \rightarrow p$"
- How to prove that $p_1, p_2, p_3, \ldots, p_n$ are equivalent, i.e., $p_1 \leftrightarrow p_2 \leftrightarrow p_3 \leftrightarrow \ldots \leftrightarrow p_n$?
  - You only need to prove "$p_1 \rightarrow p_2$" $\wedge$ "$p_2 \rightarrow p_3$" $\wedge$ "$p_3 \rightarrow p_4$" $\wedge \ldots \wedge$ "$p_{n-1} \rightarrow p_n$" $\wedge$ "$p_n \rightarrow p_1$"!

Uniqueness Proofs

- Existence: show that an element x with the desired property exists.
- Uniqueness: show that if $y \neq x$, then y does not have the desired property, or if x, y both have the desired property, then $y = x$.