Discrete Mathematics and Theory (2120)

Charlie Meyer

MoWeFri 1:00 - 1:50

MCS 3.4.2 and $\forall x$ 14-15 and equivalences and example proof and proof techniques and direct proof

MCS 3.4.2 - Proving Equivalences

A proposition with n variables has 2^n lines, so adding more and more propositions can become more and more tedious. An alternative approach is to use algebra to prove equivalence. To do this, try to remove all things except *not*, *and*, and *or*. Trick such as $A \to B$ can be turned into not(A) or (B). Below is a list of equivalence axioms.

- A and B $\leftrightarrow$ B and A (commutativity of *and*)
- (A and B) and C $\leftrightarrow$ A and (B and C) (associativity of *and*)
- $\top$ and A $\leftrightarrow$ A (identity for *and*)
- $\bot$ and A $\leftrightarrow \bot$ (zero for *and*)
- A and A $\leftrightarrow$ A (impotence for *and*)
- A and $\bar{A} \leftrightarrow \bot$ (contradiction for *and*)
- not($\bar{A}$) $\leftrightarrow$ A (double negation)
- A or $\bar{A} \leftrightarrow \top$
- not(A and B) $\leftrightarrow \bar{A}$ or $\bar{B}$ (DeMorgan for *and*)
- not(A or B) $\leftrightarrow \bar{A}$ and $\bar{B}$ (DeMorgan for *or*)

Example

Here's an example of converting any formula into a normal form. Take, for example, not((A and B) or (A and C)).

1. Start by applying DeMorgan's Law:
    1. not(A and B) and not(A and C)
2. Now Apply DeMorgan's Law again:
    1. ($\bar{A}$ or $\bar{B}$) and ($\bar{A}$ or $\bar{C}$)
3. Now apply the distributivity of *and* over or by distributing ($\bar{A}$ or $\bar{B}$) to get:
    1. (($\bar{A}$ and $\bar{A}$) or ($\bar{B}$ and $\bar{A}$)) or (($\bar{A}$ and $\bar{C}$) or ($\bar{B}$ and $\bar{C}$)).
4. Now we can use communativity and associativity to drop parenthesiss around things being *or*'ed:
    1. $\bar{A}$ or ($\bar{B}$ and $\bar{A}$) or ($\bar{A}$ and $\bar{C}$) or ($\bar{B}$ and $\bar{C}$).
5. With some extra magic, we now get:
    1. (A and $\bar{B}$ and $\bar{C}$) or ($\bar{A}$ and $\bar{B}$ and $\bar{C}$) or ($\bar{A}$ and B and C) or ($\bar{A}$ and B and $\bar{C}$) or ($\bar{A}$ and $\bar{B}$ and C) or ($\bar{A}$ and $\bar{B}$ and $\bar{C}$) or

$\forall X$ 14-15

Chapter 14: The Very Idea of Natural Deduction

The aim of a *natural deduction system* is to show that particular arguments are valid in a way that allows us to understand the reasoning that the arguments might involve. Unlike truth tables, we manipulate sentences in accordance with rules that we have set down as good rules.

One of the most reasonable reasons why one would use natural deduction is because evaluating truth tables with over 1024 lines (2^10) is unreasonable.

Chapter 15: Basic Rules for TFL

We will develop a *natural deduction* system. For each connective, there will be *introduction* rules that allow us to prove a sentence that has connective as the main logical operator, and *elimination* rules, that allow us to prove something given a sentence that has connective as the main logical operator.

15.1: The idea of a formal proof A *formal proof* is a sequence of sentences. The last line of a formal proof is the conclusion. As an illustration, consider $\neg(A \vee B) \therefore \neg A \wedge \neg B$.

We will start by writing the premise 1. $\neg(A \vee B)$ with a line underneath it. Everything written above the line is an *assumption*, and everything written below the line will either be something that follows form the assumption or a new assumption. Since we hope to conclude that $\neg A \wedge \neg B$, we will conclude our proof with n. $\neg A \wedge \neg B$.

As another illustration, suppose we wanted to consider $A \vee B, \neg(A \wedge C), \neg(B \wedge D) \therefore \neg C \vee D$. The argument has three premises:

1. $A \vee B$
2. $\neg(A \wedge C)$
3. $\neg(B \wedge \neg D)$

   n. $\neg C \vee D$

15.2: Conclusion

If we want to show that Ludwig is both reactionary and libertarian, we can use natural deduction to adopt the following:

- R: Ludwig is reactionary
- L: Ludwig is libertarian

If we were somewhere on lines 8 and 15 for R and L, respectively, then we could write the following proof as follows:

- 8 | R
- 15| L
- $R \wedge L \qquad \wedge I 8, 15$

Furthermore, if you had already proved that Ludwig was both libertarian and reactionary, then you could use *elimination* rules to prove conclude that ludwig is reactionary:

- x | $L \wedge R$
- $R \qquad \wedge Ex$

15.3: Conditional Consider the argument "If Jane is smart then she is fast. Jane is smart $\therefore$ Jane is fast.

Notice how this follows a straightforward conditional elimination rule ($\rightarrow E$):

- m | $A \rightarrow B$
- n | A
- $B \qquad \rightarrow Em, n$

This rule is also sometimes called *modus ponens*. THis is an elimination rule, because it allows us to obtain a sentence that may not contain '$\rightarrow$' having started with a sentence that does contain '$\rightarrow$.'

Now, let's consider the instance where we state: "Ludwig is reactionary. Therefore if Ludwig is libertarian, then Ludwig si both reactionary *and* libertarian."

1	R	
2	$\quad L$	
3	$\quad R \wedge L$	$\wedge$I 1, 2
4	$L \rightarrow (R \wedge L)$	$\rightarrow$I 2–3

Notice how an additional assumption ('Ludwig is libertarian') is necessary for the sake of the argument. To indicate that we're no longer dealing with just our original assumption 'R.' We are now in a position to use $\wedge I$. Now we have shown that, on the additional assumption 'L' that we can obtain '$R \wedge L$' or that we can conclude '$L \rightarrow (R \wedge L)$.' The idea of the rule $\rightarrow I$ is invoked when making additional assumptions.

However, one must be diligent when using these, as we must close the subproof to signify when we have returned to the main proof. Thus, we stipulate that "to cite individual lines when applying a rule, those lines must (1) come before the application of the rule, but (2) not occur within a closed subproof."

1	A	
2	$\quad B$	
3	$\quad\quad C$	
4	$\quad\quad A \wedge B$	$\wedge$I 1, 2
5	$\quad C \rightarrow (A \wedge B)$	$\rightarrow$I 3–4
6	$B \rightarrow (C \rightarrow (A \wedge B))$	$\rightarrow$I 2–5

Biconditional In order to prove $W \leftrightarrow X$ we need to prove X on the assumption W and W on the assumption X. The biconditional rule $\leftrightarrow I$ therefore requires two subproofs. See below.

i	$\quad \mathscr{A}$	
j	$\quad \mathscr{B}$	
k	$\quad \mathscr{B}$	
l	$\quad \mathscr{A}$	
	$\mathscr{A} \leftrightarrow \mathscr{B}$	$\leftrightarrow$I i–j, k–l

Disjunction Suppose that Ludwig is reactionary. Then Ludwig is either reactionary or libertarian. TO say that Ludwig is either reactionary or libertarian is to say something weaker than to say that Ludwig is reactionary.

m	$\mathscr{A}$	
	$\mathscr{A} \vee \mathscr{B}$	$\vee$I m

and

m	$\mathscr{A}$	
	$\mathscr{B} \vee \mathscr{A}$	$\vee$I m

Contradiction and Negation The rule for introducing negation is that we can use it whenever we explicitly contradict ourselves, i.e. whenever we find both a sentence and its negation appearing in our proof:

There are more rules that I couldn't be bothered to include, so read them yourself. Considering what is taught in class almost never overlaps with $\forall X$, don't waste your time unless it's legitimately covered in class.

Logic Rules (Equivalences)

Two expressions are equivalent if they have the same truth value.

Simplifications

They are equivlaences that also work backwards. Here are "the big 5:"

long	simplified	name of rule
$\neg\neg P$	P	double negation
$\neg\top$	$\bot$	definition of $\bot$
$P \wedge \bot$	$\bot$	simplification
$P \wedge \top$	P	simplification
$P \vee \bot$	P	simplification
$P \vee \top$	$\top$	simplification

Here's another important table (remember how to read it!)

operands	$\rightarrow$	$\leftrightarrow$	$\oplus$	$\wedge$	$\vee$
P op P	$\top$	$\top$	$\bot$	P	P
P op $\neg P$	$\neg P$	$\bot$	$\top$	$\bot$	$\top$
$\neg P$ op P	$\neg P$	$\bot$	$\top$	$\bot$	$\top$
$\top$ op P	P	P	$\neg P$	P	$\top$
P op $\top$	P	P	$\neg P$	P	$\top$
$\bot$ op P	$\top$	$\neg P$	P	$\bot$	P
P op $\bot$	$\neg P$	$\neg P$	P	$\bot$	P

Associative and Commutative Properties A binary operators is commutative if its operands can be swapped without changing the meaning of the operation. A binary operator is associative if the pair of them can be re-parenthesized without changing the meaning of their joint operation.

Operator	Associativity	Commutativity
$\neg$	not a binary operator	not a binary operator
$\wedge$	$(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R)$	$P \wedge Q \equiv Q \wedge P$
$\vee$	$(P \vee Q) \vee R \equiv P \vee (Q \vee R)$	$P \vee Q \equiv Q \vee P$
$\oplus$	$(P \oplus Q) \oplus R \equiv P \oplus (Q \oplus R)$	$P \oplus Q \equiv Q \oplus P$
$\rightarrow$	not associative	not commutative
$\leftrightarrow$	$(P \leftrightarrow Q) \leftrightarrow R \equiv P \leftrightarrow (Q \leftrightarrow R)$	$P \leftrightarrow Q$

Note that mixing operators doesn't always hold. Also, it is common to write several operators in a row with/without parenthesis, such as $P \vee Q \vee R \vee S$ instead of $P \vee (Q \vee (R \vee S))$ Parenthesis can be changed with associativity or because they are redundant.

Other equivalences

Form 1	Form 2	Name of Rule
$A \to B$	$\neg A \lor B$	definition of implication
$A \land (B \lor C)$	$(A \land B) \lor (A \land C)$	Distributive Law
$A \lor (B \land C)$	$(A \lor B) \land (A \lor C)$	Distributive Law
$(A \land B) \lor C$	$(A \lor C) \land (B \lor C)$	Distributive Law
$(A \lor B) \land C$	$(A \land C) \lor (B \land C)$	Distributive Law
$\neg(A \land B)$	$(\neg A \lor \neg B)$	DeMorgan's
$\neg(A \lor B)$	$(\neg A \land \neg B)$	DeMorgan's
$(A \leftrightarrow B)$	$A \to B \land B \to A$	definition of implication
$(A \oplus B)$	$(A \lor B) \land \neg(A \land B)$	definition of exclusive or
$A \oplus B$	$\neg(A \leftrightarrow B)$	
$A \leftrightarrow B$	$\neg(A \oplus b)$	xnor
$P \to (A \lor Q)$	$(P \land \neg A) \to Q$	

Entails

Logical Entailment

Given	Entails	Name
$\bot$	x	
	$\top$	
	$A \lor \neg A$	excluded middle
$A \land B$	A	
A and B	$A \land B$	
A	$A \lor B$	
$A \lor B$ and $\neg B$	A	disjunctive syllogism
$A \to B$ and $B \to C$	$A \to C$	hypothetical syllogism; transitivity of implication
$A \to B$ and A	B	modus ponens
$A \to B$ and $\neg B$	$\neg A$	modus tolens
$A \leftrightarrow B$	$A \to B$	
$A \to C, B \to C$, and $A \lor B$	C	
$A \to B, C \to D$, and $A \lor C$	$B \lor D$	
$A \to B$	$A \to (A \land B)$	
$\neg(A \land B), A$	$\neg B$	

Assume-and-Prove entailment A proof assumes that A and derives B entails that $A \to B$. This is commonly used in the inductive step of a proof by induction:

- $A \vdash B$
- $\therefore A \to B$

A proof that assumes A and derives $\bot$ entails that $\neg A$. This is called "proof by contradiction" or an "indirect proof."

- $A \vdash \bot$
- $\therefore \neg A$

A proof $x \in S \vdash P(x)$ entails $\forall_x \in S.P(x)$. This is called "universal induction."

- $x \in S \vdash P(x)$
- $\therefore \forall_x \in S.P(x)$

If P(x) and x is some specific member of S, that entails $\exists_x \in S.P(x)$. This is called "existential instantiation."

- $x \in S$
- P(x)
- $\therefore \exists_x \in S.P(x)$

Set Entailment

Given	Entails
P(x) and $x \in S$	$\exists_x \in S.P(x)$
$\forall_x \in S.P(x)$ and $T \subseteq S$	$\forall_x \in T.P(x)$
$\exists_x \in S.P(x)$ and $T \supseteq S$	$\exists_x \in T.P(x)$

Given	Entails
$\forall_x \in S.P(x)$ and $S \neq \emptyset$	$\exists_x \in S.P(x)$
$\lvert S\rvert \neq \lvert T\rvert$	$S \neq T$
$\lvert S\rvert < \lvert T\rvert$	$S \not\supseteq T$
$\exists_x \in S.P(x)$	$S \neq \emptyset$

Qualified Entailments

Given	Entails	Names
$\exists_x \in S.P(x)$	P(s), for any $s \in S$ we care to pick	universal instantiation
$\exists_x \in S.P(x)$	$s \in S \wedge P(s)$ where s is an otherwise-undefined new variable	existential instantiation
$s \in S \vdash P(s)$	$\forall_x \in S.P(x)$	universal generalization
$P(s) \wedge s \in S$	$\exists_x \in S.P(x)$	existential generalization

Mathematical Identities

The following are all true for all real numbers where both sides of the equal sign are defined:

- $\log_a(a^x) = x$
- $a^{\log_a(x)} = x$
- $\log_a(xy) = \log_a(x) + \log_a(y)$
- $\log_a(\frac{x}{y}) = \log_a(x) - \log_a(y)$
- $\log_a(x^y) = y \log_a(x)$
- $\log_a(x) = \frac{\log_b(x)}{\log_b(a)}$
- $\log_{a^b}(x) = b^{-1} \log_a(x)$

Also the following are true:

- $(a \in \mathbb{Z}) \wedge (a > 1) \vDash (a$ has at least two factors$)$
- $(a \in \mathbb{Z}) \wedge (a > 1) \wedge (a$ has exactly two factors$) \equiv (a$ is prime$)$
- Each integer greater than 1 has exactly one prime factorization

Example Proof: DeMorgan's Law

Theorem 1: For any expressions P and Q, the expression $\neg(P \wedge Q)$ is equivalent to the expression $(\neg P) \vee (\neg Q)$

Option 1: Brute-Force a Truth Table

P	Q	$\neg$	$(P \wedge Q)$	$(\neg P)$	$\vee$	$(\neg Q)$
false	false	**true**	false	true	**true**	true
false	true	**true**	false	true	**true**	false
true	false	**true**	false	false	**true**	true
true	true	**false**	true	false	**false**	false

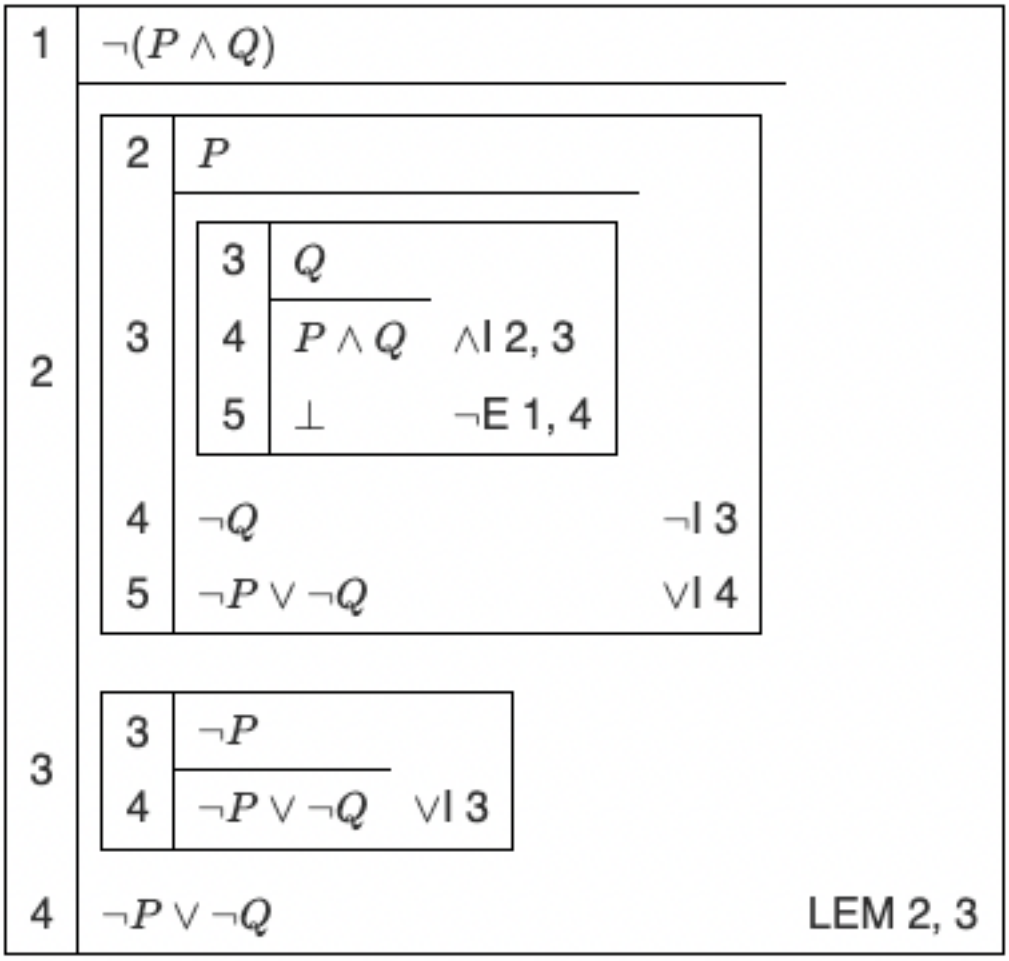
Option 2: Proof by Cases: prose

Proof: Let **N** represent the expression $\neg(P \wedge Q)$ and **O** represent the expression $(\neg P) \vee (\neg Q)$. The proof is by case analysis. There are two cases, either P is true or it is false.

- Case 1: P is true:
  - **N** is $\neg(\top \wedge Q)$; using the identity of *and*, this can be re-written as $\neg Q$. **O** is $(\neg\top) \vee (\neg Q)$, which is equivalent to $\bot \vee (\neg Q)$; using the identity of **or**, this can be re-written as $\neg Q$. Because **N** and **O** are equivalent to the same thing, they are equivalent to each other. Thus the theorem holds in this case.
- Case 2: P is false:
  - **N** is $\neg(\bot \wedge Q)$; using the zero of *and* this can be re-written as $\neg\bot$ which is $\top$. **O** is $(\neg\bot) \vee (\neg Q)$ which is equivalent to $\top \vee (\neg Q)$; using the zero of *or*, this can be re-written as $\top$. Thus, because **N** and **O** are equivalent to the same thing, they are equivalent to each other. Thus, the theorem holds in this case.

Because the theorem holds in all cases, it is true. ■.

Option 3: TFL



TFL Proofs by cases helps demonstrate two important principles:

1. The goal of proof by cases is to add extra information in each case, visible here as the extra given in each sub-proof
2. The cases need to be exhaustive, so that the *or* of all of them is equivalent to $\top$. In TFL, this is ensured by adding specific proof rules (such as LEM) that only apply when this is the case.

Proof Techniques

Apply Equivalence Rules

In a small-step proof, write an equivalent expression and cite the rule used to reach it.

Example — The following uses a note per line to show how it is equivalent to the preceding line

1	$A \vee (B \vee C)$	
2	$(A \vee B) \vee C$	Associative property of $\vee$
3	$(B \vee A) \vee C$	Commutative property of $\vee$
4	$B \vee (A \vee C)$	Associative property of $\vee$
5	$(\neg(\neg B)) \vee (A \vee C)$	Double negation
6	$(\neg B) \rightarrow (A \vee C)$	Disjunction to implication

You can also use prose instead:

> *Example* — This is the same example as the previous one, but written in prose style instead.
>
> $A \vee (B \vee C)$ can be re-written as $(\neg\neg B) \vee (A \vee C)$, which is equivalent to $(\neg B) \to A \vee C$ by the equivalence of implication and disjuction.

Also apply rearranging - utilizing the associative, communities, and distributive properties of operators. Use simplifying - removing double negation and the ones and zeroes effects of tautologies and contradictions.

Apply Entailment

Because $A \equiv B$ implies $A \vDash B$, you can use equivalence rules in a proof that applies entailment. There are many more entailments (called "proof rules") than equivalence rules, so using them can make a proof construction much easier. A proof using "proof rules" is called a *direct proof*.

(Guide to a) Direct Proof

Propositions and *operators*: MCS 1.1 and $\forall x$ 4.3 and MCS3-3.2.0 and $\forall x$ 5 and about $\to$ and operators

MCS 1.1

Propositions

A *proposition* is a statement that is either true or false. In the examples below, the first is true and the second is false.

- **Proposition 1.1.1** $2 + 3 = 5$
- **Proposition 1.1.2** $1 + 1 = 3$

Being true or false excludes statements that are not binary answers, and also excludes questions whose truth varies on circumstance. Unfortunately, not all propositions are easy:

- **Proposition 1.1.3**: For every nonnegative integer, n, the value of $n^2 + n + 41$ is prime.

To check this, we can let $p(n) ::= n^2 + n + 41$ (::= means "equal by definition). However, we see that p(40) $= 40^2 + 40 + 41 = 41 * 41$ which is not prime. But, the point of this was to show that in general, you can't check a claim about an infinite set by checking a finite set of its elements, no matter how large the finite set. And, **Proposition 1.1.3** could also be rewritten as: $\forall_n \in \mathbb{N}$. p(n) is prime.

Some other cool theorems:

- Fermat's Last Theorem: _There are no positive integers x, y, and z such that $x^n + y^n = z^n$ for some integer $n > 2$. This was eventually proved incorrect.
- Goldbach's Conjecture: *Every even integer greater than 2 is the sum of two primes.* To this day, no one nows whether it's true or false.

$\forall_x$ 4.3 - Atomic Sentences

Entirely unhelpful?

MCS 3 - 3.2.0

While there is a ton of ambiguity in the english language in common speech, in mathematics it is necessary to formulate speech precisely, as we can't make an exact argument if we're not sure what the statements we're even arguing mean!

Propositions from Propositions

Propositional variables are used in place of specific propositions and can only take on the values of $\top$ or $\bot$.

3.1.1 - Not, and, and or If P is a proposition, then so is not(P).

P	not(P)
$\top$	$\bot$
$\bot$	$\top$

The truth table for the proposition "P and Q" can be seen below:

P	Q	P and Q
T	T	T
T	F	F
F	T	F
F	F	F

The truth table for the proposition "P or Q" can be seen below:

P	Q	P or Q
T	T	T
T	F	T
F	T	T
F	F	F

If you want to exclude the possibility of having both, then use the "xor," exclusive-or, opeation:

P	Q	P xor Q
T	T	F
T	F	T
F	T	T
F	F	F

3.1.2 - Implies The combinding operation with the least intuitive technical meaning is "implies." This can also be understood as "An implication is true when the if-part is false or the then-part is true." Here is the truth table:

P	Q	P implies Q
T	T	T (tt)
T	F	F (tf)
F	T	T (ft)
F	F	F (ff)

The idea of "implies" can allow us to see whether the proposition evaluates to true or not. For example, "If Goldbach's Conjecture is true, then $x^2 \geq 0$ for every real number x." Now, this sentence is a proposition in the form "P implies Q." The hypothesis, P, is "Goldbach's Conjecture is true" and the conclusion, Q, is "$x^2 \geq 0$ for every real number x." Since the conclusion is definitely true, we're either on (tt) or (ft) of the truth table - either way, the proposition as a whole is *true*!

Here's an example of a false implication: "If the moon shines white, then the moon is made of white cheddar." Yes, the moon shines white. But no, the moon is not made of white cheddar cheese. So we're on line (tf) of the truth table, and thus the proposition is *false*.

False Hypotheses An illustrative example of false hypotheses is a system specification which consisted of a series of a dozen rules:

- if C_i: the system sensors are in condition i, then A_i: the system takes action i. Or, more concisely, C_i implies A_i for $1 \leq i \leq 12$. Then the fact that the system obeys the specification would be expressed by saying that the :
- [C_1 implies A_1] and [C_2 implies A_2] and ... and [C_{12} implies A_{12}] (3.1)

Suppose conditions C_2 and C_5 are true, and the system indeed takes on the specified actions A_2 and A_5. This means that in this case the system is behaving according to the specification, and accordingly we want the formula 3.1 to come out to *true*. Now the implications C_2 implies A_2 and C_5 implies A_5 are both true because their hypotheses and their conclusions are true. But in order for 3.1 to be true, we need all other implications with the false hypotheses C_i for $i \neq 2, 5$ to be true. This is exactly what the rule for implications with false hypotheses accomplishes.

If and Only If Mathematics joins propositions in one additional way: The proposition "P if and only if Q" asserts that P and Q have the same truth value - either both are true or both are false.

P	Q	P iff Q
T	T	T
F	T	F

P	Q	P iff Q
T	F	T
F	F	T

for example, the following *iff* statement is true for every real number x: $x^2 - 4 \geq 0 \iff |x| \geq 2$.

- Note that: "iff" is $\iff$.
- For some values of x, *both* inequalities are true. For other values of x, *neither* inequality is true. In every case, however, the IFF proposition as a whole is true.

3.2 - Propositional Logic in Computer Programs

Consider the code snippet: `if (x > 0) || (x <= 0 && y > 100)`. Let A be the proposition taht `x>0` and let B be the proposition that `y > 100`. Then we can rewrite the condition as: "A or (not(A) and B).

3.2.1 - Truth table calculation Let's evaluate the truth table of the expression "A or (not(A) and B)":

A	B	A or (not(A) and B)	A or B
T	T	T	T
T	F	T	T
F	T	T	T
F	F	F	F

Note that somehow, the outputs of A or B are the same as "A or (not(A) and B). Expressions whose truth tables always match are called *equivalent*. So, we can simplify the code without changing its behavior to `if (x > 0 || y > 100)`.

Notation

English	Symbolic Notation
Not(P)	$\neg P$ (alternatively, $\bar{P}$)
P and Q	$P \wedge Q$
P or Q	$P \vee Q$
P implies Q	$P \to Q$
if P then Q	$P \to Q$
P iff Q	$P \leftrightarrow Q$
P xor Q	$P \oplus Q$

For example, the notation "If P and not(Q), then R" would be written: $(P \wedge \bar{Q}) \to R$.

$\forall_X$ **5 - Connections**

symbol	what it is called	rough meaning
$\neg$	negation	"it is not the case that"
$\wedge$	conjunction	"both... and..."
$\vee$	disjunction	"either... or..."
$\to$	conditional	"if... then..."
$\leftrightarrow$	biconditional	"... if and only if ..."

5.1 - Negation Consider the statement B: "Mary is in Barcelona." Using $\neg$, we can symbolize the sentence "It is not the case that Mary is in Barcelona" using $\neg B$.

You can also express things like "The wrench is not irreplaceable" by saying $\neg\neg B$. Be careful of tricks like "Jane is unhappy." If we were to say $\neg B$, then we would be saying "It is not the case that Jane is happy," however the phrase "Jane is unhappy" could mean she could be in a state of neither happiness nor unhappiness.

5.2 - Conjunction Consider:

1. Adam is athletic
2. Barbra is athletic
3. Adam is athletic, and Barbara is also athletic.

The third expression can be written as a conjunction of the two - 2. $\wedge$ 3..

You can then use the previous statements and use parenthesis to express expressions like "it's not the cae that you will get both soup and salad" like $\neg(S_1 \wedge S_2)$ where S_1 means "you will get soup" and S_2 means "you will get salad."

5.3 - Disjunction Consider:

1. Either Fatima will play videogames, or she will watch movies.
2. Either Fatima or Omar will play videogames.
    - F: Fatima will play videogames
    - O: Omar will play videogames
    - M: Fatima will watch movies

To express these expressions, we need to express sentence 1. using $F \vee M$, called *disjunction*. We can express sentence 2 as $F \vee O$.

5.4 - Conditional Consider these sentences:

1. If Jean is in paris, then Jean is in France
2. Jean is in France only if Jean is in Paris.
    - P: Jean is in Paris
    - F: Jean is in France

We can symbolize sentence 1. as $P \to F$. This connective is called "the conditional." Here, P is the *antecedent* and F is the *consequent*. We can symbolize sentence one as $F \to P$.

5.5 - Biconditional Consider:

1. Josie is a dog only if she is a mammal
2. Josie is a dog if she is a mammal
3. Josie is a dog if and only if she is a mammal
    - J: Josie is a dog
    - M: Josie is a mammal

Sentence 1 can be symbolized as $J \to M$. Sentence 2 can be symbolized as $M \to J$. Sentence 3 says something stronger than either 1 or 2 - it can be paraphrased as "Josie is a dog if josie is a mammal, and josie is a dog only if josie is a mammal. This can be expressed by $J \leftrightarrow M$. This is called the"biconditional," or "iff" - if and only if. A sentence can be symbolized as $A \leftrightarrow B$ if it can be paraphrased in english as 'A iff B,' as 'A if and only if B.'

5.6 - Unless We can use all the connectives to express many kinds of sentences with complex structures. For example, the sentence "Unless you wear a jacket, you will catch a cold" where J="You will wear a jacket" and D="You will catch a cold" can be expressed as $\neg J \to D$.

About $\to$: "If" and Logic

If

If it rains, I'll get wet. Logical part is $R \to W$.

Raining?	Wet?	Consistent with the phrase?
No	no	yes
No	yes	yes
Yes	no	no
Yes	yes	yes

Beyond Logic - there's a connotation which propositional logic cannot encode that rain starts shortly before wetness and there's a causal connotation that propositional logic cannot encode that rain creates wetness.

If you earn a 93% or more, you get an A Logical part: $E \leftrightarrow A$

Earned 93?	Get A?	Consistent with phrase?
No	no	yes
no	yes	no
yes	no	no
yes	yes	yes

Beyond logic - there's a temporal condition - you get the 93% before you get the A - which cannot be encoded in propositional logic. Further, the $\leftarrow$ part is "fuzzy." A 92.8% might get you an A, but an 81% won't. The $\rightarrow$ part is implicitly "or more": an A+ would not be a problem.

This program crashes if you type Ctrl+Q Logical part: $C \leftarrow Q$

it crashes?	Ctrl+Q?	Consistent with phrase?
No	no	yes
no	yes	no
yes	no	yes
yes	yes	yes

Beyond logic - there's a temporal connection that you press Ctrl+Q before it crashes and theres a causal connotation that typing Ctrl+Q triggers the crash that logic cannot encode

When

I love it when it rains Logical part: $L \leftarrow R$

I love it?	It rains?	Consistent with the phrase?
no	no	yes
no	yes	no
yes	no	yes
yes	yes	yes

Beyond logic - "when" connotes that the truth of R comes and goes. It also connotes that rain *will* happen eventually; "if" in this phrase would have some logical meaning, but it would imply doubt that rain would ever occur.

I sing when in the shower Logical part - $\top$

I sing?	Showering?	consistent with phrase?
no	no	yes
no	yes	yes
yes	no	yes
yes	yes	yes

Beyond logic - This means that there exist some times when i am both in the shower and singing, but does not rule out either singing outside the shower nor that I might have some in-shower time when I'm not singing. We can encode that "there exists some time" idea with first order logic - $\exists t.S(t) \wedge H(t)$ but not with propositional logic.

Only

It only rains at night Logical part - $R \rightarrow N$

it rains?	It's night?	consistent with phrase?
no	no	yes
no	yes	yes
yes	no	no
yes	yes	yes

I only love my spouse Logical part - $L \leftrightarrow S$

I love it?	It's my spouse?	Consistent with phrase?
No	no	yes
no	yes	no
yes	no	no
yes	yes	yes

Beyond logic - the use of "only" implies "love" means something more specific than it would like a phrase "I love cake"and possibly something even more specific than "I love my spouse."

Operators

Propositions

Concept	Java/C	Python	This class	Bitwise	Name	other
true	`true`	`True`	$\top$ or 1	-1	tautology	T
false	`false`	`false`	$\bot$ or 0	0	contradiction	F
not P	`!p`	`not p`	$\neg P$ or $\bar{P}$	~p	negation	
P and Q	`p && q`	`p and q`	$P \wedge Q$	p & q	conjunction	PQ, $P \cdot Q$
P or Q	p || q	p or q	$P \vee Q$	P | Q	disjunction	P + Q
P xor Q	`p!=q`	`p!=q`	$P \oplus Q$	p ^ q	parity	$P \veebar Q$
P implies Q			$P \rightarrow Q$		implication	$P \supset Q, P \implies Q$
P iff Q	`p==q`	`p==q`	$P \leftrightarrow Q$		bi-implication	$P \iff Q$, P xnor Q

Proofs

Concept	Symbol	Meaning
equivalent	$\equiv$	"$A \equiv B$" means "$A \leftrightarrow B$ is a tautology"
entails	$\models$	"$A \models B$" means "$A \rightarrow B$ is a tautology."
provable	$\vdash$	"$A \vdash B$" means "A *proves* B"; it means both "$A \vdash B$ and"I know B is true because A is true. "$\vdash B$" means "I know B is true."
therefore	$\therefore$	"$\therefore A$" means "the lines above this $\vdash A$" and also connotes "A is the thing we wanted to show."
proof done	$\blacksquare_{q.e.d}$	marks the end of a written (prose) proof.
hypothesis		something we expect is true
theorem		something we've proven is true
corollary		small theorem that builds off of the main theorem
lemma		small theorem that helps set up the proof of the main theorem

Arithmetic

Concept	Symbol	Meaning
floor	$\lfloor x \rfloor$	the largest integer not larger than x; x rounded down to an integer
ceiling	$\lceil x \rceil$	the smallest integer not smaller than x; x rounded up to an integer
exponent	x^y	x multiplied by itself y times
sum	$\sum_{x \in \mathbb{S}} f(x)$	the sum of all members of $\{f(x) \vert x \in \mathbb{S}\}$. By definition, 0 if S = {}.
Product	$\prod_{x \in \mathbb{S}} f(x)$	The product of all members of $\{f(x) \vert x \in \mathbb{S}\}$. By definition, 1 if S = {}

Concept	Symbol	Meaning
product	$\prod_{x=a}^{b} f(x)$	$\prod_{x\in\mathbb{S}} f(x)$ where $\mathbb{S} = \{x \mid (x \in \mathbb{Z}) \wedge (a \leq x \leq b)\}$. The product of f(x) applied to integers between a and b inclusive
factorial	x!	$\prod_{i=1}^{x} i$. The product of all positive integers less than or equal to x. The number of permutations of a length-x sequence with distinct members.
choose	$\binom{n}{k}$	$\frac{n!}{(n-k)!k!}$. The number of k-member subsets of an n-element set.

Sets: Sets are Values, Sets vs. Sequences, Sets Writeup

Sets Writeup

Defining Sets

1. A set contains numbers. **set** is thus a term referring to a collection.
   - Member is the word for something inside a set.
2. We write a set with curly braces and commas.
   - {1, 3} is a set. so is {dog, cat, mouse}. Arguably {x?y:z, satisfaction, 2102, :dragon:} is a set but sets whose members do not have some single unifying defined type are so uncommon that many say they're not sets.
3. A member of a set has no other set-related properties besides membership in the set: not order or position in the set, not number of copies in the set.
   - Thus, {1, 2} and {2, 1} are two ways of writing the same set and {1, 2, 2} is nonsensical.
4. The only property of a set is its members
5. A set can be empty
6. A set can be infinite. So is the set of positive integers. . . .
7. A set can have sets as members.
   - {1} is a set, so is {{1}, {}, {84, 5}}, so is the set of all two-element sets of integers.
   - Having sets within sets does not change any other rules - i.e. the set {{1, 2}, {2,1}} is nonsensical.

Set operations and notation

"m is a member of S" is written as $m \in S$.

- Note that $m \in S$ is a predicate: $1 \in \{1, 2, 3\}$ is $\top$ and $1 \in \{2, 4, 6\}$ is $\bot$.

"m is not a member of S" is written as $m \notin S$, that is, $\neg(m \in S)$

- You may also see $\ni$ and $\notin$ which means the same thing as $\in$ and $\notin$ but with the set on the left and the member on the right instead of the other way around.

Sub- and super-sets

$A \subseteq B$ is defined to mean "every member of A is also a member of B," that is, $\forall x.((x \in A) \rightarrow (x \in B))$

- The $\subseteq$ symbol is pronounced "is a subset of." It intentionally looks somewhat like the $\leq$ symbol because if $A \subseteq B$ then A is "less than or equal to B in the sense that everything A has, B has too, and B may have more.

$\top$: True	$\bot$: False
$\{\} \subseteq \{1, 2\}$	$\{1, 2\} \subseteq \{2, 3\}$
$\{1\} \subseteq \{1, 2\}$	$\{1, 2\} \subseteq \{1\}$

$\{1, 2\} \subseteq \{1, 2\}$ $\{1, 2\} \subseteq$ the integers | $\{1, 2.3\} \subseteq$ the integers

$A \subset B$ is defined to mean both "$A \subseteq B$" and "$A \neq B$"; that is, $(A \subseteq B) \wedge (A \neq B)$. The symbol $\subseteq$ is pronounced "is a proper subset of."

$\top$: True	$\bot$: False
$\{\} \subset \{1, 2\}$	$\{1\} \subset \{2, 3\}$
$\{1\} \subset \{1, 2\}$	$\{1, 2\} \subset \{1\}$
	$\{1, 2\} \subseteq \{1, 2\}$

$\top$: True	$\bot$: False
$\{1, 2\} \subset$ the integers	$\{1, 2.3\} \subset$ the integers

$A \supseteq B$ means $B \subseteq A$; $A \supset B$ means $B \subset A$

- the symbol "$\supseteq$" is pronounced "is a superset of". The "$\supset$" synbol is pronounced "is a proper superset of."

Sets from other sets

$A \cup B$ is defined to mean "A set containing every member of A every member of B and no other members. That is,"a set S such that $\forall x.(x \in S \leftrightarrow ((x \in A) \vee (x \in B)))$"

- The $\cup$ symbol is pronounced "union" and $A \cup B$ is called the union of A and B." Note that $(A \subseteq) \leftrightarrow ((A \cup B) = B)$

A	B	$A \cup B$
{1, 2}	{1, 2}	{1, 2}
{1, 2}	{0}	{1, 2}
{1, 2}	{5, 6}	{1, 2, 5, 6}
{1, 2, 4, 8}	{2, 4, 6, 8}	{1, 2, 4, 6, 8}
the positive numbers	{0}	the non-negative numbers
the even numbers	the odd numbers	the integers

$A \cap B$ is defined to mean "a set containing every member shared both by A and B, and no other members"; that is; "a set S such that $\forall x.(x \in S \leftrightarrow ((x \in A) \wedge (x \in B)))$

- the $\cap$ symbol is pronounced "intersection" and $A \cap B$ is called the "intersection of A and B". It intentionally looks similar to $\wedge$ to suggest a similar value to the member of intersection of A and B if it is a member of A and a member of B.
- Note that $(A \subseteq B) \leftrightarrow ((A \cap B) = A)$

A	B	$A \cap B$
{1, 2}	{1, 2}	{1, 2}
{1, 2}	{}	{}
{1, 2}	{5, 6}	{}
{1, 2, 4, 8}	{2, 4, 6, 8}	{2, 4, 8}
the positive numbers	the integers	the positive integers

If we say that $A \cap B = \{\}$, then we say that A and B are disjoint. Less commonly, **overlap** is used to mean "are not disjoint."

A B is defined to mean "a set containing every member of A that is not a member of B, and no other members"; that is, "a set S such that $\forall x.(x \in S \leftrightarrow ((x \in A) \vee (x \notin B)))$"

- the symbol is pronounced either "Minus" or "set minus." A B has everything A has provided B does not have it. It is rarely used in computing. Note that $(A \subseteq B) \leftrightarrow ((A \backslash B)) = \emptyset$

A	B	$A \backslash B$
{1, 2}	{1, 2}	{}
{1, 2}	{}	{1, 2}
{1, 2}	{5, 6}	{1, 2}
{1, 2, 4, 8}	{2, 4, 6, 8}	{1}
the integers	the positive numbers	the negative integers

pow(A) or P(A) is defined to mean "the set of all subsets of A"; that is, "a set S such that $\forall x.(x \in S \leftrightarrow (x \subseteq A))$". pow(A) or P(A) is called the power set of A and is the set of all subsets of A. Note that every set, even {}, has one subset: {}; thus, a power set is never empty, $\emptyset \in P(A)$ is a tautology, and $\emptyset = P(A)$ is a contradiction regardless of what set A is.

A	P(A)
{}	{{}}
{1}	{{}, {1}}
{1, 2}	{{}, {1}, {2}, {1, 2}}

A	P(A)
{{1, 2}, {3}}	{{}, {{1, 2}, {3}, {{1, 2}, 3}}
{{}}	{{}, {{}}}

Concept	Set Notation	Actuality
"add" x to S	$S \cup \{x\}$	a set like S except it also has x in it
"remove" x to S	$S \backslash \{x\}$	a set like S except it does not have x in it.
A "xor" B	$(A \cup B) \backslash (A \cap B)$	a set with elements in A or B but not both

Counting Members

$|A|$ means "the number of distinct values that are members of A." We call this notion the **cardinality** and read $|A|$ as "the cardinality of A."

A	$\|A\|$
{}	0
{1, 2}	2
{{1, 2}}	2
{1, {2, 3}}	2
P({1, 3, 5, 7})	16
P(A)	$2^{\|A\|}$

Set-builder notation

Set-builder notation $\{x \in \mathbb{Z} | x^3 - 30x + 1 > 0\}$ would be written: "x for x in Z if $x^3 - 30x + 1 > 0$", although you'd have to define a sensible, finite Z range.

Common sets

There are some common sets that have their own symbols. Many of them are listed in MCS 4.1.1

Symbol	Set	Elements
	the empty set	none
$\mathbb{N}$	nonnegative integers	{0, 1, 2, 3, . . . }
$\mathbb{Z}$	integers	{. . . , -3, 2, -1, 0, 1, 2, 3, . . . }
$\mathbb{Q}$	rational numbers	$\{\frac{1}{2}, -\frac{5}{3}$, 16, etc.$\}$
$\mathbb{R}$	real numbers	$\{\pi, e, -9, \sqrt{2}, etc\}$
$\mathbb{C}$	complex numbers	$\{i, \frac{19}{2}, \sqrt{2} - 2i, etc\}$

Quantifiers and Sets

Quantifiers are often used with sets. Set-notation quantifiers and domain-bound quantifiers can be each defined in terms of the other.

Core Notation

The notation $\forall x \in S.P(x)$ means "the predicate P(x) is true for every x in the set S." It does not say anything about the true or falsehood of P(x) for x not in S, nor does it assert that there are any members of S.

The notation "$\exists x \in S.P(x)$" means "there is at least one element of S, and at least one element of S makes the predicate P(x) true. It does not say anything about the truth or falsehood of P(x) for x not in S, nor if there are more (or even at all) members of S that also make P(x) true.

- "$\forall x, y \in S$" is shorthand for "$\forall x \in S . \forall y \in S$".
- "$\exists x, y \in S$" is shorthand for "$\exists x, y \in S . \exists y \in S$".

Converting "$\forall x \in S ...$" to "$\forall x ...$"

If a domain is not specifed and all quantifiers are given with sets, the implicit domain is union of all such sets or any superset containing that union.

Converting "$\forall x$...." to "$\forall x \in S$..."

Define a set U representing the entire domain. The symbol U is not required, but is often used with the intent that it suggest the "universal set" or the "universe of discourse."

Sets vs Sequences

Sets ands sequences are "composite" values because they are made up of other values - true, false, and numbers. Both sets and sequences:

- May contain any other value, including other sets and sequences
- Are values, not entities
- are used to define other structures of interest in discrete math
- are commonly written with their contained values separated by commas

Set:

- Written with curly braces, like {1, 2}
- Cannot contain the same value more than once
- Members have no order, i.e. {2, 3} = {3, 2}
- The empty set (the only set with cardinality 0) is written {} or $\emptyset$
- Has many operators and special notations like $\{1, 2\} \cup \{x^2 | x \in \mathbb{N}^+\}$
- A singleton set is always distinct from its member; $\{2\} \neq 2$
- always called a "set"
- contained values are called "members"; "element" is also sometimes used

Sequence

- Written with parenthesis, like (1, 2)
- Can contain the same value any number of times; (1, 1) is a sequence and is distinct from (1) and (1, 1, 1)
- Items have an order, i.e. (2, 3) $\neq$ (3, 2)
- the empty sequence (the only sequence with length 0) is written () or ϵ
- has no operators that are commonly used in computing
- A singleton sequence is often considered equal to its item: i.e. (2) = 2
- Called a "sequence" or a "tuple", with special words for some lengths (i.e. "pair" or "triple") and some element types (i.e. "string")
- contained values are called "items"; "elements" is also sometimes used

Sets are Values

By example 7 is a value. Charlie is an entity. If we let x and y both refer to 7 and make x bigger, x is no longer 7. This gets to the idea that 7 is a value: an unchanging Platonic ideal. It is impossible, by definition, to make 7 itself bigger. Similarly, if Charlie eats a ton and grows, he's still Charlie.

Because x and y both refer to a value, changing the value of x has no impact on y whereas "Charles" and "Charlie" both refer to an entity, so making "Charlie" richer means "Charles" is richer.

Entities and Values

A value can be written in multiple ways "3+4" and "7" represent the same *integer* but "3+4" and "7" represent different *arithmetic expressions.*

Variables name just one value within a given context Within a single problem, if one assigns *x=2*, then x=2 and this cannot change.

Discrete math values have related programming entitites Sets are values in discrete mathematics, just like numbers are: unchangeable Platonic ideals. Everything in discrete mathematics is a mathematical construct and thus a value, not an entity.

Good Proofs in Practice

The purpose of a proof is to establish the truth of an assertion with absolute certainty. To be understanding and helpful, a proof should not only worry about correctness, but also be clear. here are some helpful tactics to achieve this:

- State your game plan: a proof begins by explaining the general line of reasoning, for example "we use case analysis" or "we argue by contradiction."
- Keep a linear flow: proofs are written like mathematical mosaics.. the steps of an argument should follow one another in an intelligible order.

- A proof is an essay: a proof looks like an essay with some equations thrown in.
- Avoid symbolism: use words when you can!
- Introduce notation thoughtfully: An argument can be easily simplified with a variable or defining a new term, but do this sparingly becuase the reader then has to remember all of that!
- Structure long proofs: Use preliminary lemmas to cite repeatedly.
- Be wary of the obvious: what is obvious to one reader might not be to another - you don't have to prove every single claim you say, but don't use phrases like "clearly" or "obviously" to bully your reader into thinking they're dumb or accept it as true just because

Skills from Mathematics

1. Discussing definitions
    1. Mathematicians begin with thinking hard about definitions early in their undergraduate career and fluency in discussing definitions is something that can benefit everyone.
2. Coming up with counterexamples
    1. When coming up with a new definition, one has as et of examples and counterexamples that one wants the definition to adhere to. So examples and counterexamples help build good definitions.
    2. When encountering an existing definition, the first thing a mathematician does is write down examples/counterexamples.
3. Being wrong and often admitting it
    1. Fostering doubt, being wrong, admitting it, and starting over distinguishes mathematical discourse even from much praised scientific discourse. There's no fame or money behind it, just the personal insight for truth.
    2. The mathematical habit is putting your personal pride or embarrassment aside for the sake of insight.
4. Evaluating many possible consequences of a claim
    1. The limits of an argument result in an even better and more elegant theorem that includes the original claim. More often, you realize you were wrong. So this habit is a less formal variation of being wrong often and coming up with counterexamples.
5. Teasing apart the assumptions underlying an argument
    1. Treating claims mathematically rather than through emotion allows us to tease apart claims without bias.
6. Scaling the ladder of abstraction
    1. Mathematicians "scale the ladder" of abstraction - i.e. they start at the lowest rung where they understand some examples of a paper, jump to the main theorem of the paper, and then synthesize that information to the real world.