WEEK 1 – PROPOSITIONAL LOGIC

PROPOSITIONS

- **proposition:** statement that is either true or false. True = 1 False = 0 (boolean values).
- **propositional variable:** *proposition* represented by a small caps letter symbol. Usually start with p, q, r, s. *= atom*
- **mathematical generality:** the substition of a long *proposition* with a *propositional variable* being equivalent.
- **argument:** conjuction of premises implying that when these are true, so must be the conclusion.
- **conjuction:** AND, both statements must be true for the conjuction to be true, else the conjuction is false
- **premise:** set of assumed propositions accepted as true for the sake of argument.
- **argument conclusion:** *deduced* proposition from logic inference. If the conjuction of all premises, implies the conclusion, and such gives a tautology, the conclusion is logically valid.

LOGICAL OPERATORS

- **logical operator**
  = logical connective: Like the arithmetic operators +,*,% etc. logical operators are the symbols placed generally between two *propositional variables* although they can take multiple paramaters into account. If so these terciary, quartenary and so forth operators are most likely not standard and their meaning (aka description of the output) is unkwonw. The hidden meaning of these unkwonw +2 operand operators can be found by observing the outputs of its truth table and translate it to DNF.
- **operand:** the propositional variable taken as input for the logical operator.
- **DNF/sum of products:** Disjunctive Normal Form. Further explained in page 3.
- **truth table:** table that shows the output for each of the of propositional variables values and a operator
- **unary operator:** takes one variable, the most common is "no change" it's represented by "nothing" in front of the variable. The second most common is **negation** (¬), which *toggles* (inverts) the current value of the propositional variable. The remaining unary operators are one that would make all outputs 1 and other one all outputs 0 regardless of the input.
- **binary operator:** returns one output from two operands. Most common are defined in the table below.

Truth table

Latex:		\lnot	\lor	\oplus	\land	\to	\leftrightarrow
p	q	$\neg p$	$p \vee q$	$p \oplus q$	$p \wedge q$	$p \to q \ \equiv \neg p \vee q$	$p \leftrightarrow q \ \equiv$ $(p \to q) \wedge (q \to p)$
0	0	1	0	0	0	1	1
0	1	1	1	1	0	1	0
1	0	0	1	1	0	0	0
1	1	0	1	0	1	1	1
English		Not p **negation** **logical complement**	p or q or both English 'or' is ambigous, as it can mean both or and xor	**exclusive or** **either** p or q **but not both**	and	p **implies** q p **is sufficient for** q q **is necessary for** p if p then q **conditional operator**	q **if and only if** p q **iff** p if p then q **and conversely** **biconditional**

* this english equals only applies to the operator output. p and q are different variables and can have a different state from each at the same time (i.e. 01 or 10) so you can't apply mathematical generality with these 2 variables.

- **ternary operator:** CSE1300 doesn't have any standard, so better to convert to DNF and use boolean logic, or a karnaugh map to generate a simplified compound proposition that generates the same output.

operator question: What's the maximum number of operators you can create for 1, 2 and 3 variables? **formula**: $2^{(2^n)}$.

Take n = 1, such as “p”; p has two possible states 0 and 1 (rows in a truth table), which is equivalent to $2^n = 2^1 = 2$ (p,q,r would have 2^3 rows). Each of these possible states of p can be arbitrarily modified by the operator and take two new forms. The maximum number of new states combinations is 2^{states}, therefore $2^2 = 4$.

FUNCTIONALLY COMPLETE, UNIVERSAL, EXPRESSIVELY ADEQUATE SET OF OPERATORS

- **functionally complete operators:** set of operators that can express all possible situations, such as ¬, ∧ *and* ∨

functionally complete question: What’s the minimum number of operators needed for a functionally complete set?

Answer: 1. The binary connectives **NAND** ↑ and **NOR** ↓ are functionally complete. This is due to the fact that when applied to the same *atom* it provides its inverse (not), it also has similar OR/AND properties. Thus, 2 operators in 1.

PRECEDENCE RULES

- **compound proposition:** proposition variable made from sub proposition(s) represented with a capital letter.
- **precedence rules:** (, ¬, ∧, ∨, ⊕, →, ↔, …*if equal precende precedence goes from left to right*.
- **associative operator:** the order of the of the operands doesn’t chage the output. $p \wedge q \wedge r \equiv r \wedge q \wedge p$ some of them are: ∧, ∨, ↔, ≡ (equivalence operator yields either a tautology or a contradiction)
- **main connective:** is the operator that es evaluated last, according to the precedence rules and parentheses.

IMPLICATIONS IN ENGLISH

Take ($p \rightarrow q$ *as a reference point*)

- **implication**: p implies q **= conditional**
- **hypotheses**: **p** in p implies q **= antecedent**
- **conclusion**: **q** in p implies q **= consequent**
- **sufficient**: p is sufficient for q in $p \rightarrow q$
- **necessary**: q is necessary for p in $p \rightarrow q$
- **converse:** $q \rightarrow p$ (*flip* p → q)
- **inverse:** $\neg \boldsymbol{p} \rightarrow \neg \boldsymbol{q}$ (*negate atoms in* p → q)
- **contrapositive:** $\neg \boldsymbol{q} \rightarrow \neg \boldsymbol{p}$ ***= combination of inverse and converse***
- **biconditional*:*** $p \leftrightarrow q$

LOGICAL EQUIVALENCE

- **situation:** each possible combination of values of the propositional variables that a **truth table** contains (row)
- **logically equivalent** (≡): propositions are logically equivalent if they have the same values in every situation. Which can be confirmed if $P \leftrightarrow Q$ is a *tautology.* The symbol ≡ has essentially the same meaning as =.

CLASSIFYING PROPOSITIONS

- **tautology (T):** (compound) proposition that is true for every situation in its truth table.
- **contradiction (F):** (compound) propoistion that is false for every situation in its truth table.
- **contingency:** (compound) proposition where at least one situation is false and at least one situation is true.

BOOLEAN ALGEBRA

- **George Boole:** accounted for introducing Boolean algebra in 1854.

• Double negation	$\neg(\neg p) \equiv p$	
• Excluded middle	$p \vee \neg p \equiv \mathbf{T}$	(variable + complement = 1)
• Contradiction	$p \wedge \neg p \equiv \mathbf{F}$	
• Identity laws	$\mathbf{T} \wedge p \equiv p$	$\mathbf{F} \vee p \equiv p$
• Impotent laws	$p \wedge p \equiv \mathbf{p}$	$p \vee p \equiv \mathbf{p}$
• Cummulative law	$p \wedge q \equiv q \wedge p$	$p \vee q \equiv q \vee p$
• Associative law	$p \wedge q \wedge r \equiv r \wedge q \wedge p$	$p \vee q \vee r \equiv r \vee q \vee p$
• Distribute law	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
• Demorgan’s law	$\neg(p \wedge q) \equiv \neg p \vee \neg q$	$\neg(p \vee q) \equiv \neg p \wedge \neg q$

- **Laws of Boolean algebra**

Trick: use CSE1400 Computer Organisation technique of replacing T for 1, F for 0, ∨ for $+$, ∧ for * and ¬ for $-1*$ (or just minus) and then all laws, except impotent laws, are equal to highschool math. For the impotent law just assume that any number larger than 1 becomes immediately 1 again. Thus $p^2 = p^1 = p \text{ and } 2p = p$.

For the second distributive law open the brackets and simplify using the first distributive law and the T identity law:

$$(p+q)(q+r) = p^2 + pr + pq + qr = p(\text{"}1\text{"} + r + q) + qr = p + qr$$

But it is easier to remember I want Pasta or (Quesadillas and Rice). I'd be equally happy if I get (Pasta or Quesadillas) and (Pasta or Rice)

- **duality:** from a tautology that uses only the operators ∧, ∨, and ¬, another tautology can be obtained by interchanging ∧ with ∨ and T with F.

SUBSTITUTION LAWS

- **statement of the same form:** proposition that can be obtained by substituting **all** instances of the same variable with a different variable. *Substitute p for Q. (p for q means we have a redundant atom).*
- **1st Substition Law:** quoted when applying logical equivalence between a compound proposition and an atom. $Q = p$
- **2nd Substition Law:** quoted when applying logical equivalence between 2 compound propositions. $Q = P$
- **chain of equivalences:** resulting conclusion that two proposotions are equivalent by finding a chain of equivalences using the substition laws.
- **simplification**: the use of chain of equivalences to provide a logically equivalent proposition that contains less logical connectives and less atoms.
- **tri-state boolean**: Model that includes a third state representing "unknown" or "not proven" which lead to **fuzzy logic**. It follows non-stardad logics because it allows for a middle ground.

LOGIC CIRCUITS*

* Next to a title means not CSE1300 exam material.

- **logic gates**: electronic components (often transistors) that compute the values of simple propositions. They are equivalent to logic connectives. Most logic gates are made out of **NAND** and **NOR** gates.
- **logic circuit:** combination of logic gates, equivalent to a compound proposition.

DISJUNCTIVE NORMAL FORM (DNF)

Any compound proposition (or logic circuit) of any arbitrary size has a logically equivalent simplified form.

- **DNF/*sum of products*:**. Is a compound proposition made out of a "*disjunction of conjuctions*" of "*simple terms*", and neither the terms nor the atoms in the terms are duplicated (no redundancy).
  In other words a sum (or) of products (and) where each of the products in the conjuction (and) represent the state of the variables ($1 = p, 0 = \neg p$) in situations that have an output of true/1.
- **simple term**: refers to atoms and their complements (the products).
- **conjuction of simple terms**: a product of simple terms.

The DNF or sum of products does not necessarily need to be in its most simplified way (minimal). Boolean logic or *Karnaugh Maps* can be used to obtain the minimal form. The products of the DNF must be atoms, not compound propositions. DNF shall not rely on parentheses nor on any other operator besides ¬, ∨, ∧.

p	**q**	**output**	
0	0	0	
0	1	**1**	$\boldsymbol{\neg p \wedge q}$
1	0	**1**	$\boldsymbol{p \wedge \neg q}$
1	1	0	

DNF: $\boldsymbol{\neg p \wedge q \vee p \wedge \neg q} = (-\mathbf{p})\mathbf{q} + \mathbf{p}(-\mathbf{q})$

Thanks to the Substition Laws, it is sufficient to just consider the DNF proposition of a theorem when proving it.

WEEK 2 – PREDICATE LOGIC/PREDICATE CALCULUS

PREDICATES

Disclaimer: not every predicate has to correspond to an English sentence.

- **Charles Sanders Peirce** (1839–1914): Father of predicate logic and logic circuits early thinker.
- **predicates:** the elements of predicate logic that are applied to an object (a subject in grammar). Itself is an incomplete propoosition P(x) = x is red. **You can complete a predicate by inputting an entity P(a)** = a is red. **A completed predicate is a proposition.**
- **applying P to a**: P(a), "applying" is exclusively reserved for using the predicate with one entity at a time.
- **object:** the subject in a predicate logic statement to which the predicate is applied. Such as 'a'.
  = entity = subject
- **domain of discourse for the predicate**: the domain of object inputs accepted as a paramater (subject) to which the predicate can be applied. Such as "humans" in the statement T(x) = x pays taxes. Only humans pay them.
- **one-place predicate:** predicate with only 1 place holder value (object).
- **two-place predicate:** predicate that takes two paramaters (object/subject/entity). L(x,y) = x loves y.
  Each place holder variable ('slot') can have its own domain of discourse.

QUANTIFIERS

Predicates can only be ***applied*** to entities. **Quantifiers** can **be applied to predicates**, to turn them into **propositions**.

- **quantifiers:** These specify the extent to which an incomplete predicate can be applied to the whole domain. *All/no, some/not all* "domain" is "red". If the predicate is D(x) = is a doctor. And the domain is "humans". A quantifier that would make the statement true ***in our current specific universe*** would be "some". Making the incomplete statement look in "logic English" like: "some human *is a* doctor"*.*

*Predicate logic is bound to a universe. Sometimes it's the real world, sometimes it's made up (such as a **Tarksi world**). Sometimes you can build a made up world to prove that the a predicate statement is a contradiction.*

Quantifiers:

Latex:	\forall	\exists	\lnot \forall	\lnot \exists
Symbol	∀	∃	¬∀	¬∃
English	for all, all **universal quantifier**	there exists, at list one **existential quantifier**	not all	there doesn't exist there is no, no

$\exists P(x)$ = there exists an x in the domain of discourse for P for which P(x) is true = at least one x is P(x).

- **open statement:** incomplete predicate that contains one or more unfilled place holder values (entity variables), which becomes a proposition when these are substituied for an entity. Incomplete predicates can also become a proposition when a quantifier is applied to them.
- **free variables:** the placeholders/variables unfilled by an entity in an open statement.
- **bound variable:** variable to which the quantifier is applied to, i.e. 'for all $\boldsymbol{x}$' = $\forall x$ and is not "free" anymore.

Quantifiers are not associative: The **order** of the quantifier **does change** the output.

$$\exists x\left(\forall \mathrm{y}\big(L(x,y)\big)\right) \not\equiv \forall \mathrm{y}\left(\exists \mathrm{x}\big(L(x,y)\big)\right)$$

Trick: *(this is a trick relying on English Language heuristics, not a proof) if* L(x,y) = x loves y

The arrows below do not mean the implication arrow, just the direction of the verb (direct to indirect object)

$\forall \mathrm{y}\big(L(x,y)\big)$ • **All y's:** $(\mathrm{x} \xrightarrow[loves]{} \mathbf{y}) = \mathrm{x} \xrightarrow[loves]{} \forall y$ = x loves all y's $\exists x\left(\forall \mathrm{y}\big(L(x,y)\big)\right)$ • **At least an x:** $(\mathbf{x} \xrightarrow[loves]{} \forall \mathrm{y})$ = At least an x loves all y's = At least someone loves everyone At least **one guy loves everyone**	$\exists \mathrm{x}\big(L(x,y)\big)$ • **At least an x:** $[(\mathbf{x} \xrightarrow[loves]{} \mathrm{y}) = (\mathrm{y} \xleftarrow[loves]{} \mathbf{x})] = \mathrm{y} \xleftarrow[loves]{} \exists \boldsymbol{x}$ **=** y <u>is loved</u> by **at least an x** (flipped verb) $\forall \mathrm{y}\left(\exists \mathrm{x}\big(L(x,y)\big)\right)$ • **All y's:** $(\mathbf{y} \xleftarrow[loves]{} \exists \boldsymbol{x})$ = All y's is loved by at least an x = Everyone is loved by at least someone **Everybody is loved**, but **not necessarily by the same guy**

1. You organize the verb (flip if necessary) to **keep the inner quantified (bonded) variable to the right**.
   a. Iff swap is made, then goes from present to participle and viceversa.
2. **Replace the variable with the bonded version** (apply quantity to the variable).
3. Copy the updated predicate and apply the 2nd step to the remaining free variable.
4. Replace “at least an x”/”at least an y” for “someone” and “all y’s”/”all x’s” for everyone.
5. **Take the Natural English result as the literal result**.

$$Someone \rightarrow everyone \;\neq Everyone \leftarrow someone(\boldsymbol{s})$$

   a. $Someone \rightarrow everyone$: $Someone\ is\ the\ direct\ object, everyone\ is\ the\ indirect\ object$
      i. Someone loves everyone: A single instance applies its action to all instances **(Unique to All)**
      ii. At least a single person loves all the persons (including himself).
   b. $Everyone \leftarrow someone(\boldsymbol{s})$: $someone(\boldsymbol{s})\ \boldsymbol{are}\ the\ direct\ \boldsymbol{objects}, everyone\ is\ the\ indirect\ object$
      i. Everyone is loved by at least someone. **(All to 1/Different instances)**
      ii. Those someone(s) can be different persons.
6. **Fill in quantifiers in the order in which they appear**.

Quantifier question: how many different propositions can be obtained from L(x, y) by applying quantifiers? Answer: **six distinct meanings,** among them:

$\forall x$	$\forall y$	outcome	flipped (-1 = redundant)
0	0	$\exists x\ \exists y$	$\exists y\ \exists x$ *-1*
0	1	$\exists x\ \forall y$	$\forall y\ \exists x$
1	0	$\forall x\ \exists y$	$\exists y\ \forall x$
1	1	$\forall x\ \forall y$	$\forall y\ \forall x$ *-1*

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

- **Predicate DeMorgan’s** (1,2) **and Associative** (3,4) **Laws**

- Attempt with L(x,y,z) 1 + 6*4 + 1 = 26

$\forall x$	$\forall y$	$\forall z$	outcome + *guessed extra flips*
0	0	0	$\exists x\ \exists y\ \exists z + 0$
0	0	1	$\exists x\ \exists y\ \forall z\ +$ $\exists x\ \forall z\ \exists y\ +$ $\exists y\ \forall z\ \exists x\ +$ $\forall z\ \exists x\ \exists y$
0	1	0	4
0	1	1	4
1	0	0	4
1	0	1	4
1	1	0	4
1	1	1	1

- **English to predicate conversions** (in CSE1300 often you have to guess the domain, it’s not always explicit):

Adjectives	
SINGLE ENTITY **Red rose** (it is a red and it is rose) **Mortal human** **Long black train**	Domain x = things $Red(x) \wedge Rose(x)$ $Human(x) \wedge Mortal(x)$ $Long(x) \wedge Black(x) \wedge Train(x)$
Copulative verbs	
SINGLE ENTITY **The sky is blue** (x = the sky) **The rose is red** (x = the rose)	Domain x = things $Blue(x)$ $Red(x)$
QUANTIFIED ALL **Everyone is mortal** (1 simple version) **All humans are mortal** (2 advanced version)	Domain x = 1 Humans/ 2 Animals $\forall x Mortal(x)$ $\forall x(Human(x) \rightarrow Mortal(x))$ *Living things can be mortal without being human*
QUANTIFIED EXISTS **At least one swan is black** = There exists a black swan **There is a human that is mortal** = There is a mortal human	Domain x = things $\exists x(Swan(x) \wedge Black(x))$ $\exists x(Human(x) \wedge Mortal(x))$

Non-Copulative verbs	
SINGLE ENTITY **x writes** **x reads and writes**	Domain x = persons $Writes(x)$ $Reads(x) \wedge Writes(x)$
MULTIPLE ENTITIES **x loves y**	Domain x = persons $Loves(x, y)$ *The predicate “formula” is given in present form. With the first paramater being the direct object and the second parameter the indirect object.*
QUANTIFIED ALL (ONE OBJECT) **All mammals sleep** (all animals which are mamals sleep) (for all animals, if it’s a mammal, it sleeps)	Domain x = animals $\forall x\big(Mammal(x) \rightarrow Sleep(x)\big)$ *Same as a copulative verb*
QUANTIFIED EXISTS (ONE OBJECT) **At least one parrot talks**	Domain x = birds $\exists x\ (Parrot(x) \wedge Talk(x))$ *Same as a copulative verb*
QUANTIFIED (MULTIPLE OBJECTS ALL/EXISTS)	Domain x = persons/things
Someone loves y	$\exists x(L(x, y))$
x loves everyone	$\forall y\big(L(x, y)\big)$
Someone loves everyone (not equal to statement below)	$\exists x\left(\forall y\big(L(x, y)\big)\right)$
Everyone is loved by someone (not necessarily the same guy)	$\forall y\left(\exists x\big(L(x, y)\big)\right)$
Jack owns y (y is owned by jack)	$O(jack, x)$
“Owned by jack” computer (it is owned by jack and a comp)	$O(jack, x) \wedge C(x)$
Jack owns a computer (there exists a thing that is (owned by jack) and is a computer)	$\exists x\big(O(jack, x) \wedge C(x)\big)$
Everything jack owns is a computer (For all things, if jack owns it, it is a computer)	$\forall x\big(O(jack, x) \rightarrow C(x)\big)$
If jack owns a computer then he is happy	$\left(\exists x\big(O(jack, x) \wedge C(x)\big)\right) \rightarrow H(jack)$
Everyone who owns a computer is hapy	$\forall x\left(\exists y\big(O(x, y) \wedge C(y)\big) \rightarrow Happy(x)\right)$
Everyone owns a computer	$\forall x\left(\exists y\big(O(x, y) \wedge C(y)\big)\right)$
A single computer is owned by everyone	$\exists y\left(\forall x\big(O(x, y) \wedge C(y)\big)\right)$
Quantifier comparision with stereotype: **Blondes are Stupid.** Domain x = girls	
1. All (blondes are stupid) = All girls that are blonde are stupid = For all girls: If blonde, then stupid	$\forall x\big(Blonde(x) \rightarrow Stupid(x)\big)$
2. Some blondes are stupid = There **exist a (stupid blonde)** = There exists at least a girl that: is blonde and is stupid	$\exists x\big(Blonde(x) \wedge Stupid(x)\big)$
3. Not (all (blondes are stupid)) = Negation of (1. All blondes are stupid) = There exists at least a girl that is: blonde and not stupid	$\neg\left(\forall x\big(Blonde(x) \rightarrow Stupid(x)\big)\right)$ $\equiv \neg\left(\forall x\big(\neg\, Blonde(x)\ \vee\ Stupid(x)\big)\right)$ $\equiv \exists x\big(\ Blonde(x)\ \wedge\ \neg\, Stupid(x)\big)$
4. No blondes are stupid = There does**n’t (exist a (stupid blonde))** = Negation of (2. Some blondes are stupid) = For all girls: if blonde, then not stupid = **(All) Blondes are not supid**	$\neg\left(\exists x\big(Blonde(x) \wedge Stupid(x)\big)\right)$ $\equiv \forall x\big(\neg Blonde(x) \vee \neg Stupid(x)\big)$ $\equiv \forall x\big(Blonde(x) \rightarrow \neg Stupid(x)\big)$

Everyone, At least, At most, Exactly (Simple domain x = humans)	
Everyone is happy	$\forall x\big(Happy(x)\big)$
At least one person is happy	$\exists x\big(Happy(x)\big)$
At least two people are happy	$\exists x\exists y\big(Happy(x) \wedge Happy(y) \wedge (x \neq y)\big)$
At least three people are happy	$\exists x\exists y\exists z(Happy(x) \wedge Happy(y) \wedge \mathrm{Happy(z)} \wedge (x \neq y) \wedge (x \neq z) \wedge (y \neq z))$ *There exists a set with at least 3 objects that have the properties of being happy and these objects are not the same entity.*
At most 1 person is happy = opposite of at least 2 happy	$\neg\left(\exists x\exists y\big(Happy(x) \wedge Happy(y) \wedge (x \neq y)\big)\right)$ $\equiv \neg\left(\exists x\exists y\left(\big(Happy(x) \wedge Happy(y)\big) \wedge (x \neq y)\right)\right)$ $\equiv \forall \mathrm{x}\forall \mathrm{y}\left(\neg\big(Happy(x) \wedge Happy(y)\big) \vee \neg(x \neq y)\right)$ $\equiv \forall \mathrm{x}\forall \mathrm{y}\left(\neg\big(Happy(x) \wedge Happy(y)\big) \vee (x = y)\right)$ $\equiv \forall \mathrm{x}\forall \mathrm{y}\left(\big(Happy(x) \wedge Happy(y)\big) \rightarrow (x = y)\right)$
At most 2 peoeple are happy = opposite of at least 3 happy	$\neg(\exists x\exists y\exists z(Happy(x) \wedge Happy(y) \wedge \mathrm{Happy(z)} \wedge (x \neq y) \wedge (x \neq z) \wedge (y \neq z)))$ $\neg(\exists x\exists y\exists z((Happy(x) \wedge Happy(y) \wedge \mathrm{Happy(z)}) \wedge ((x \neq y) \wedge (x \neq z) \wedge (y \neq z))))$ $\forall \mathrm{x}\forall \mathrm{y}\forall \mathrm{z}(\neg(Happy(x) \wedge Happy(y) \wedge \mathrm{Happy(z)}) \vee \neg((x \neq y) \wedge (x \neq z) \wedge (y \neq z)))$ $\forall \mathrm{x}\forall \mathrm{y}\forall \mathrm{z}(\neg(Happy(x) \wedge Happy(y) \wedge \mathrm{Happy(z)}) \vee ((x = y) \vee (x = z) \vee (y = z)))$ $\forall \mathrm{x}\forall \mathrm{y}\forall \mathrm{z}((Happy(x) \wedge Happy(y) \wedge \mathrm{Happy(z)}) \rightarrow ((x = y) \vee (x = z) \vee (y = z)))$
At most n people are happy	*Opposite of at least n+1 happy*
There is exactly 1 happy person = at least 1 happy and at most 1 = at least 1 happy object (x), that also has the quality that no other y is happy and not him (x).	Conjuction: $\exists x\big(Happy(x)\big) \wedge \forall \mathrm{x}\forall \mathrm{y}\left(\big(Happy(x) \wedge Happy(y)\big) \rightarrow (x = y)\right)$ Short version: $\exists x\left(Happy(x) \wedge \neg\exists y\big(Happy(y) \wedge (x \neq y)\big)\right)$ $\equiv \exists x\left(Happy(x) \wedge \forall \mathrm{y}\big(\neg Happy(y) \vee (x = y)\big)\right)$ $\equiv \exists x\left(Happy(x) \wedge \forall y\big(Happy(y) \rightarrow (x = y)\big)\right)$ *There exists an object (x) that has the propety of being happy and the property that no other objects (y) are happy and different from it (y not equal to x)*
There is exactly 2 happy persons = at least 2 happy and at most 2 = at least 2 happy objects (x, y), that also have the quality that no other z is happy and not x nor y.	Conjuction (short steps, long formula): $\exists x\exists y\big(Happy(x) \wedge Happy(y) \wedge (x \neq y)\big) \wedge$ $\forall \mathrm{x}\forall \mathrm{y}\forall \mathrm{z}((Happy(x) \wedge Happy(y) \wedge Happy(z)) \rightarrow ((x = y) \vee (x = z) \vee (y = z)))$ Short formula, more steps: $\exists x\exists y\left(Happy(x) \wedge Happy(y) \wedge (x \neq y) \wedge \neg\exists z\big(Happy(z) \wedge (z \neq x) \wedge (z \neq y)\big)\right)$ $\exists x\exists y\left(Happy(x) \wedge Happy(y) \wedge (x \neq y)\ \neg\exists z\left(Happy(z) \wedge \big((z \neq x) \wedge (z \neq y)\big)\right)\right)$ $\exists x\exists y\left(Happy(x) \wedge Happy(y) \wedge (x \neq y)\ \forall \mathrm{z}\left(\neg Happy(z) \vee \big((z = x) \vee (z = y)\big)\right)\right)$ $\exists x\exists y\left(Happy(x) \wedge Happy(y) \wedge (x \neq y)\ \forall \mathrm{z}\left(Happy(z) \rightarrow \big((z = x) \vee (z = y)\big)\right)\right)$ *There exists an object (x) and an other object (y) that have the propeties of being happy and the property that no other objects (z) are happy and different from them (z not equal to x, or z not equal to y).* Can also be read as: There exists at least an object x and an object y that each have the property of being happy and that for all other objects z, if these are happy, then they must refer to x or y
There is exactly 3 happy persons	$\exists x\exists y\exists z\left(H(x) \wedge H(y) \wedge H(z) \wedge (x \neq y) \wedge (x \neq z) \wedge (y \neq z) \wedge \forall w\left(H(w) \rightarrow \big((w = x) \vee (w = y) \vee (w = z)\big)\right)\right)$
There is exactly 4 [equals = n] [not equals = $(n^2 - n)/2$]	Equals part: $(s = x) \vee (s = y) \vee (s = z) \vee (s = w)$ Not equals part: $(x \neq y) \wedge (x \neq z) \wedge (\mathrm{x} \neq \mathrm{w}) \wedge (y \neq z) \wedge (y \neq w) \wedge (z \neq w)$

LOGICAL EQUIVALENCE

In predicate logic, two formulas are logically equivalent if they have the same truth value for all possible predicates. It is generally considered **simpler to have the negation operator applied to basic propositions** such as R(y), rather than to quantified expressions such as ∀y(R(y) ∨ Q(y)). Replacing placeholders with predicates keeps it equivalent.

DEDUCTION

- **conclusion:** proposotion that is logically deduced from a set of **premises.**
- **premise:** proposotion accepted as true for the sake of argument
- **argument:** claim that a certain conclusion follows from a given set of premises. **Traditional format**:

$$\begin{array}{l} p \to \mathrm{q} \\ p \\ \hline \therefore q \end{array}$$

It can be proved with a truth table that it is always true: If $p \to q$ AND p is true then q is true $\equiv (p \to q \land p) \to q \equiv$ T

p	q	$p \to q$	$((p$	$\to$	$q \land p)$	$\to$	q)
0	0	1	0	1	0	1	0
0	1	1	0	1	0	1	1
1	0	0	1	0	0	1	0
1	1	1	1	1	1	1	1

- **if the premises are true, then the conclusion must be true:** That's all the truth table proves, but not that the conclusion is necesarily true (the premises might be wrong in real life). This just makes the argument.
  Let P represent the Disjunction of Premises and Q represent the conclusion. These statements are all equal:
- $\boldsymbol{P \to Q}$ **is a Tautology**
- $\boldsymbol{P \Rightarrow Q}$ which can also be read as premises P do lead to conclusion Q, and
- Q **follows logically from** P
- **in all cases where** P **is true,** Q **is also true**
- Q **can be logically deduced from** P
- P **logically implies Q**

- **valid argument:** Argument where $P \Rightarrow Q$ holds (the conclusion follows logically from the premises).
- **logical deduction:** the formulation of propositions that satisfy $P \Rightarrow Q$

RULES OF INFERENCE, INFERENCE RULES OR TRANSFORMATION RULES – PROPOSITIONAL LOGIC

- **modus ponens:** second premise confirms the left side of the implication, deducing the right side

$$\begin{array}{l} p \to \mathrm{q} \\ p \\ \hline \therefore q \end{array}$$

- **modus tollens:** 2nd premise contradicts the right side of the implication, deducing the left side is false

$$\begin{array}{l} p \to \mathrm{q} \\ \neg q \\ \hline \therefore \neg p \end{array}$$

- **Basic rules**:

$$\begin{array}{l} p \lor q \\ \neg p \\ \hline \therefore q \end{array} \qquad \begin{array}{l} p \\ q \\ \hline \therefore p \land q \end{array} \qquad \begin{array}{l} p \land q \\ \hline \therefore p \end{array} \qquad \begin{array}{l} p \\ \hline \therefore p \lor q \end{array}$$

- **law of syllogism**: (chain of implications allows you to "jump" to a next one(s))

$$\begin{array}{l} p \to \mathrm{q} \\ p \to \mathrm{r} \\ \hline \therefore p \to \mathrm{r} \end{array}$$

- **"Free"/"cheap" conclusion**: if $P \equiv Q$

$$\begin{array}{l} P \\ \hline \therefore Q \end{array}$$

With all these rules, instead of making a truth table of the Disjunction of Premises → Conclusion, we can demonstrate the validity of an argument by deducing the conclusion from the premises in a sequence of steps.

- **formal proof (of an argument): sequence of proposotions** such that the **last proposition** in the sentence is the **conclusion** of the argument and every proposition in the sequence is either a premise of the argument or follows by logical deduction from propositions that precede in the list. The existence of such a proof shows that the conclusion follows logically from the premises, and therefore that the argument is valid

Example Argument:

$$\begin{array}{l} (p \land \mathrm{r}) \to \mathrm{s} \\ q \to p \\ t \to r \\ q \\ t \\ \hline \therefore s \end{array}$$

Proof.

1. $q \to p$	premise
2. q	premise
3. p	from 1 and 2 (modus ponens)
4. $t \to r$	premise
5. t	premise
6. r	from 4 and 5 (mods ponens)
7. $p \land r$	from 3 and 5
8. $(q \land r) \to s$	premise
9. s	from 7 and 8 (mods ponens)

Q.E.D

Remember! no proof is complete if it does not start with *Proof.* and ends with Q.E.D. or □

The argument is valid if in all cases where all the premises are true, the conclusion is also true. The argument is invalid if there is at least one case where all the premises are true and the conclusion is false.

- **counterexample**: Situation where the premises are true and the conclusion is false, needed always to be provided to disprove the validity of an argument. Example:

  $$\begin{array}{l} p \to \mathrm{q} \\ q \\ \hline \therefore p \end{array}$$

  False, Proof. Counter example: $\neg p \ \land q$

  □

INFERENCE RULES - PEDICATE LOGIC

- **basic predicate rule:** $\boldsymbol{(\forall x P(x)) \Rightarrow P(a)}$**.** If a predicate is true for all entities, then it is true for a specific entity.
- **predicate modus ponens:**

  $$\begin{array}{l} \forall x\big(P(x) \to Q(x)\big) \\ P(a) \\ \hline \therefore Q(a) \end{array}$$

- **predicate modus tollens:**

  $$\begin{array}{l} \forall x\big(P(x) \to Q(x)\big) \\ \neg Q(a) \\ \hline \therefore \neg P(a) \end{array}$$

To disprove validity of arguments in predicate logic, you again need to provide a **counterexample**. You can literally make up any formal structure as counterexample to disprove it. Consider the following argument:

$$\begin{array}{l} \exists x P(x) \\ \forall x\big(P(x) \to Q(x)\big) \\ \hline \therefore \forall x Q(x) \end{array}$$

Not valid. *Proof.*

Counter example given in the following structure A: D = {a, b}; P^A = {a}; Q^A = {a}

It says two things, the 2nd premise is that all objects of domain x that have property P, also have property Q. And the 1st premise is that there exists at least an object x with property P. The conclusion says that all object x have property Q, however it is enteriely possible to have a situation (structure) where there is an additional object that does not have property P and therefore not having Q wouldnt violate the second premise, and make the conclusion false. Therefore Q(x) does not hold for all.

WEEK 3+ PROOFS, SETS, RELATIONS, FUNCTIONS (AND TARKSI WORLD FROM PREVIOUS WEEK)

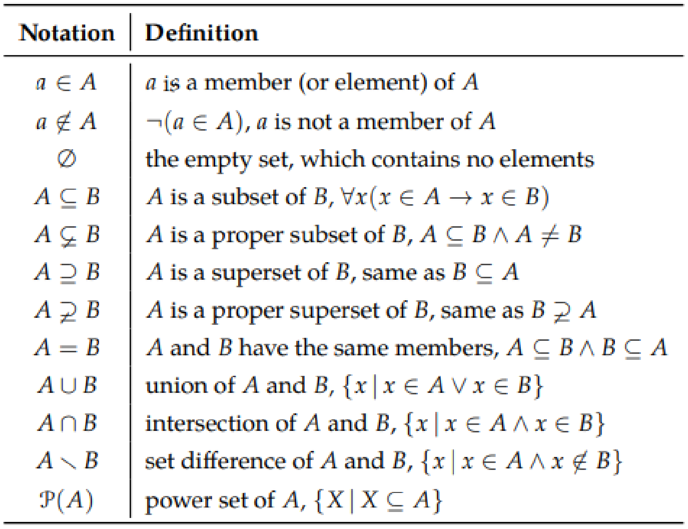
This is something I was not able to summarize well. The best way to learn this is by doing a lot of exercises!

I recommend Book of Proof By Richard Hammack, Chapter: 1, 4-12, 14. (for both, explanations and exercises)

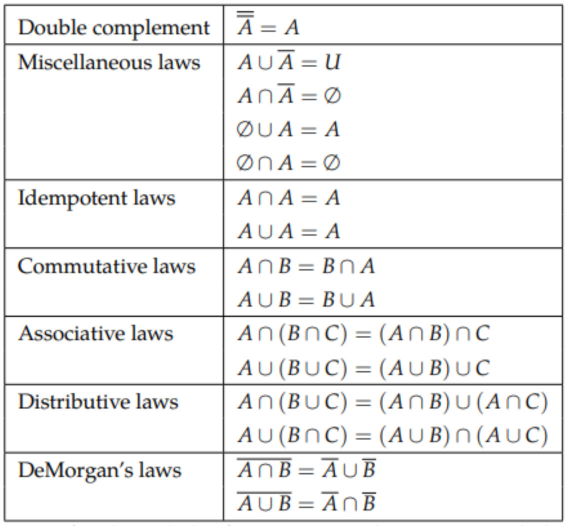
It has a free pdf version on the original page, a quick google search should display within the first results. Chapter 2 is the logic (I covered it already here and you'll have to use CSE1300 logic style (i.e. small caps for atoms, big caps for composed propositions, 1 and 0s (instead of T and F) and 0 is an element of N) 13 is calculus proofs.

This Summary does not cover the Tarksi World. I recommend to check the assigned book and lectures. It regards a very specific "construction" of sets. I recommend to get familiar with Set notation before trying to learn Tarksi World.

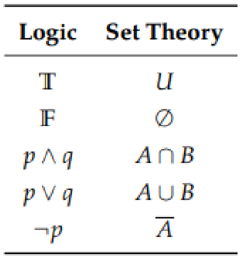
BOOLEAN ALGEBRA OF SETS



Notation	Definition
$a \in A$	a is a member (or element) of A
$a \notin A$	$\neg(a \in A)$, a is not a member of A
$\emptyset$	the empty set, which contains no elements
$A \subseteq B$	A is a subset of B, $\forall x(x \in A \rightarrow x \in B)$
$A \subsetneq B$	A is a proper subset of B, $A \subseteq B \wedge A \neq B$
$A \supseteq B$	A is a superset of B, same as $B \subseteq A$
$A \supsetneq B$	A is a proper superset of B, same as $B \supsetneq A$
$A = B$	A and B have the same members, $A \subseteq B \wedge B \subseteq A$
$A \cup B$	union of A and B, $\{x \mid x \in A \vee x \in B\}$
$A \cap B$	intersection of A and B, $\{x \mid x \in A \wedge x \in B\}$
$A \setminus B$	set difference of A and B, $\{x \mid x \in A \wedge x \notin B\}$
$\mathcal{P}(A)$	power set of A, $\{X \mid X \subseteq A\}$



Double complement	$\overline{\overline{A}} = A$
Miscellaneous laws	$A \cup \overline{A} = U$ $A \cap \overline{A} = \emptyset$ $\emptyset \cup A = A$ $\emptyset \cap A = \emptyset$
Idempotent laws	$A \cap A = A$ $A \cup A = A$
Commutative laws	$A \cap B = B \cap A$ $A \cup B = B \cup A$
Associative laws	$A \cap (B \cap C) = (A \cap B) \cap C$ $A \cup (B \cup C) = (A \cup B) \cup C$
Distributive laws	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
DeMorgan's laws	$\overline{A \cap B} = \overline{A} \cup \overline{B}$ $\overline{A \cup B} = \overline{A} \cap \overline{B}$

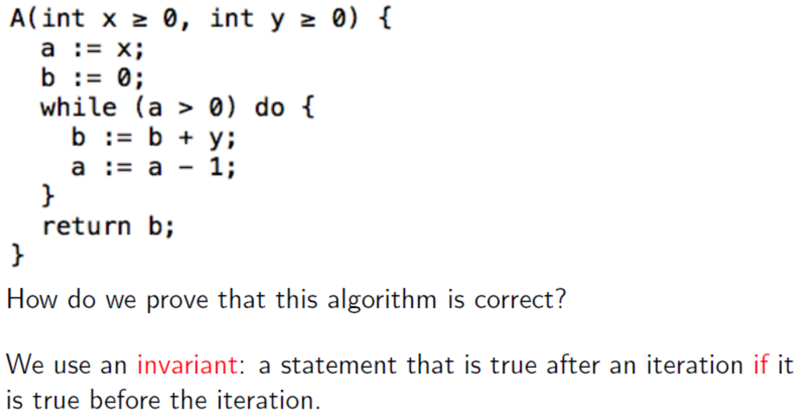


Logic	Set Theory
$\mathbb{T}$	U
$\mathbb{F}$	$\emptyset$
$p \wedge q$	$A \cap B$
$p \vee q$	$A \cup B$
$\neg p$	$\overline{A}$

APPENDIX – SOME PROOFS (MIGHT NOT BE PERFECT), MISCELANIOUS EXERCISES AND NOTES

Regarding the proofs I did: Focus mostly on how I structure my proofs grammatically rather than on the actual algebraic development within the body of the proof.

LOOP INVARIANT



```
A(int x ≥ 0, int y ≥ 0) {
  a := x;
  b := 0;
  while (a > 0) do {
    b := b + y;
    a := a - 1;
  }
  return b;
}
```

How do we prove that this algorithm is correct?

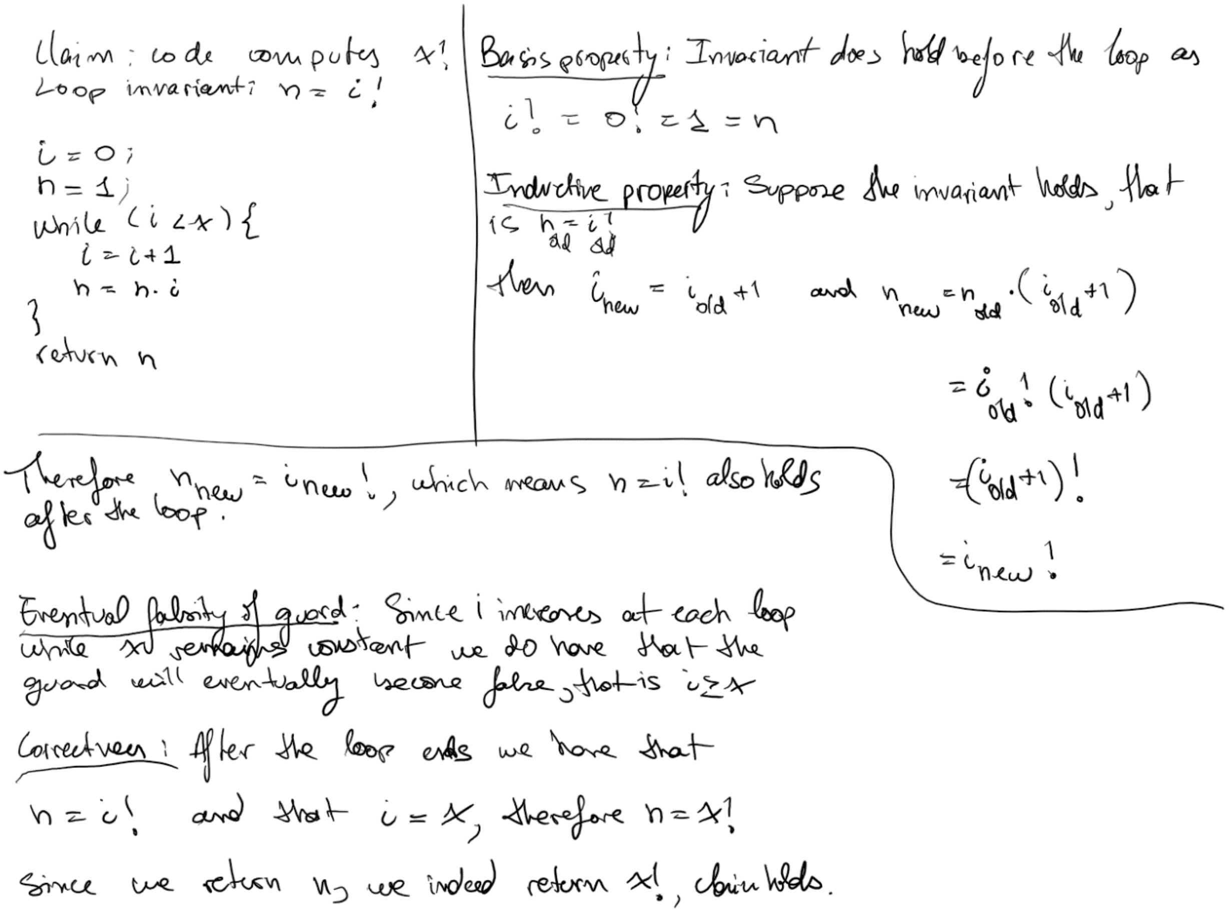
We use an invariant: a statement that is true after an iteration if it is true before the iteration.

An **invariant** is a **property P** of a loop such that P is **true before and after the loop is executed**, but not necessarily during the execution of the loop.

The **post-condition** of an algorithm i.e. **property Q** is **proved by finding an invariant P such that P → Q** and showing that the **condition of the loop** ("guard") has become **false**.

A PROOF OF CORRECTNESS USING INDUCTION

1. **Initialisation** or precondition or basis property: the invariant is true before the loop starts.
2. **Inductive case** or maintenance: the invariant is true after an iteration of the loop.
3. **Termination** or eventual falsity of the guard: the loop stops at some point.
4. **Postcondition**: the invariant shows that after the loop is done the algorithm has produced the output we want.



Claim: code computes $x!$
Loop invariant: $n = i!$

```
i = 0;
n = 1;
while (i < x) {
    i = i+1
    n = n · i
}
return n
```

Basis property: Invariant does hold before the loop as
$i! = 0! = 1 = n$

Inductive property: Suppose the invariant holds, that is $n_{old} = i_{old}!$
then $i_{new} = i_{old} + 1$ and $n_{new} = n_{old} \cdot (i_{old}+1)$

$= i_{old}! \, (i_{old}+1)$

$= (i_{old}+1)!$

$= i_{new}!$

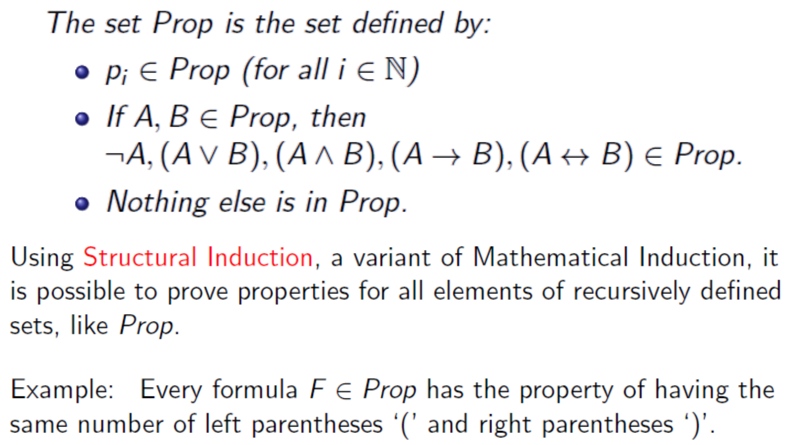
Therefore $n_{new} = i_{new}!$, which means $n = i!$ also holds after the loop.

Eventual falsity of guard: Since i increases at each loop while x remains constant we do have that the guard will eventually become false, that is $i \geq x$

Correctness: After the loop ends we have that
$n = i!$ and that $i = x$, therefore $n = x!$
Since we return n, we indeed return $x!$, claim holds.

SET OF ALL PROPOSITIONS

The set PROP represents all valid formula in propositional logic, therefore it is defined by:



The set Prop is the set defined by:

- $p_i \in Prop$ (for all $i \in \mathbb{N}$)
- If $A, B \in Prop$, then $\neg A, (A \lor B), (A \land B), (A \to B), (A \leftrightarrow B) \in Prop$.
- Nothing else is in Prop.

Using Structural Induction, a variant of Mathematical Induction, it is possible to prove properties for all elements of recursively defined sets, like *Prop*.

Example: Every formula $F \in Prop$ has the property of having the same number of left parentheses '(' and right parentheses ')'.

A partition of a set A is a set of subsets of A

WEAK INDUCTION

Proof

$$P(n): \quad \sum_{i=1}^{n} \frac{1}{i(i+1)} = \frac{n}{n+1}$$

Base case:

$$P(1): \quad \sum_{1}^{1} \frac{1}{1\cdot(1+1)} = \frac{1}{2} = \frac{1}{1+1} = \frac{n}{n+1}$$, P(1) holds

Inductive case:

We will show that

Given any integer $k \geq 1$ $P(k) \to P(k+1)$

Suppose P(k) holds (the inductive hypothesis IH), so that

$\sum_{i=1}^{k} \frac{1}{i(i+1)} = \frac{k}{k+1}$ holds then P(k+1) should be true, that is

$\sum_{i=1}^{k+1} \frac{1}{i(i+1)} = \frac{k+1}{k+1+1}$ should hold

Then:

$$\sum_{i=1}^{k+1} \frac{1}{i(i+1)} = \frac{1}{(k+1)(k+2)} + \sum_{i=1}^{k} \frac{1}{i(i+1)}$$

We apply the IH

$$\sum_{i=1}^{k+1} \frac{1}{i(i+1)} = \frac{1}{(k+1)(k+2)} + \frac{k}{k+1}$$

$$= \frac{1 + k(k+2)}{(k+1)(k+2)}$$

$$= \frac{k^2+2k+1}{(k+1)(k+2)}$$

$$= \frac{(k+1)^2}{(k+1)(k+2)}$$

$$= \frac{k+1}{k+2} = \frac{n}{n+1}$$, P(k+1) holds

Since k is an arbitrary integer ≥ 1, the implication holds for all $n \geq 1$. It follows by mathematical induction that P(n) is true for all $n \geq 1$

Q.E.D

Proof - $P(n)$: $3 \mid F_{4n}$, F = fibonacci sequence indexed at 1

Base Case:

$P(1)$: $3 \mid F_4 = 3 \mid 3$, holds

Inductive step:

We will show that given any integer $k \geq 1$ $P(k) \to P(k+1)$ holds

Suppose $P(k)$ holds (Inductive Hypothesis) so that $3 \mid F_{4k} \Rightarrow$

$F_{4k} = 3 \cdot m \in m\mathbb{Z}$

then;

$$3 \mid F_{4(k+1)} = 3 \mid F_{4k+4}$$

$$= 3 \mid F_{4k+3} + F_{4k+2}$$

$$= 3 \mid F_{4k+2} + F_{4k+1} + F_{4k+1} + F_{4k}$$

$$= 3 \mid F_{4k+1} + F_{4k} + F_{4k+1} + F_{4k+1} + F_{4k}$$

$$= 3 \mid 2 \cdot F_{4k} + 3 \cdot F_{4k+1}$$

We apply the IH:

$$= 3 \mid 2 \cdot 3m + 3 \cdot F_{4k+1}$$

$$= 3 \mid 3(2m + F_{4k+1})$$

$$= 3 \mid 3 \cdot l \text{ for } (l \in \mathbb{Z} \wedge l = 2m + F_{4k+1})$$

which holds

Since k was an arbitrary integer ≥ 1, the implication holds for all $n \geq 1$. It follows by the principle of mathematical induction that $P(n)$ is true for all $n \geq 1$.

Q.E.D

STRONG INDUCTION

definition (true)

(8 points) Consider the sequence: $a_1 = 1$, $a_2 = 8$, $\underline{a_n = a_{n-1} + 2a_{n-2}}$ for all $n > 2$. Prove that $\underline{a_n = 3 \cdot 2^{n-1} + 2 \cdot (-1)^n}$ for all $n \geq 1$. *Hint:* use strong induction for your proof.

claims to prove

Proof. Let P(n) denote $a_n = 3 \cdot 2^{n-1} + 2 \cdot (-1)^n$

Base cases:

P(1): $a_1 = 3 \cdot 2^{1-1} + 2 \cdot (-1)^1 = 3 - 2 = 1$, holds

P(2): $a_2 = 3 \cdot 2^{2-1} + 2 \cdot (-1)^2 = 6 + 2 = 8$, holds

Inductive step:

We will show that given any integer $k \geq 1$

$(P(1) \wedge P(2) \wedge P(3) \wedge \ldots P(k)) \rightarrow P(k+1)$ holds.

Suppose P(m) holds for all values smaller or equal to k.

That is: $a_m = 3 \cdot 2^{m-1} + 2 \cdot (-1)^m$ holds for all $m \leq k$ (Inductive Hypothesis)

Then we have to prove that P(k+1) should also hold,

that is: $a_{k+1} = 3 \cdot 2^k + 2(-1)^{k+1}$

So, by definition we have:

$$a_{k+1} = a_k + 2a_{k-1}$$

Applying the Inductive Hypothesis yields:

$$a_{k+1} = 3 \cdot 2^{k-1} + 2(-1)^k + 2(3 \cdot 2^{k-2} + 2(-1)^{k-1})$$

$$= 3 \cdot 2^{k-1} + 2 \cdot (-1)^k + 3 \cdot 2^{k-1} + 4 \cdot (-1)^{k-1}$$
$$= 2(3 \cdot 2^{k-1}) + 2 \cdot (-1)^k - 4 \cdot (-1)^k$$
$$= 3 \cdot 2^k + 2 \cdot -1 \cdot (-1)^k$$
$$= 3 \cdot 2^k + 2 \cdot (-1)^{k+1}$$

, P(k+1) holds

Since k is an arbitrary integer ≥ 1, the implication holds for all $n \geq 1$

It follows by mathematical induction that P(n) is true for all $n \geq 1$.

Q.E.D.

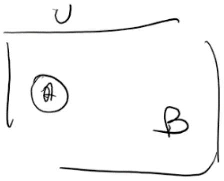
SETS PROOFS

Proving $A \subseteq B$

$A^C = \overline{A}$

$\overline{A} \subseteq B \to \overline{B} \subseteq A$

Proof involving sets:



Proof. Suppose $\overline{A} \subseteq B$, To prove $\overline{B} \subseteq A$

take an arbitrary object $e \in \overline{B}$,

Since $\overline{A} \subseteq B$, "an object that is not in A is in B". The contrapositive is:

"an object that is not in B is in A"

$e \in \overline{B} = e \notin B = e \in A$

Since e was an arbitrary object, the claim holds for all objects.

Q.E.D

Proving $A = B$

Claim: $\overline{A \cap B} = \overline{A} \cup \overline{B}$

Proof. Suppose $\overline{A \cap B} \subseteq \overline{A} \cup \overline{B} \land \overline{A} \cup \overline{B} \subseteq \overline{A \cap B}$

take an arbitrary object x

$$
\begin{aligned}
x \in \overline{(A \cap B)} &\leftrightarrow x \notin (A \cap B)\\
&\leftrightarrow x \notin A \cup x \notin B\\
&\leftrightarrow x \in \overline{A} \cup x \in \overline{B}\\
&\leftrightarrow x \in (\overline{A} \cup \overline{B})
\end{aligned}
$$

Read from bottom to top for the second subset relation

Since x was arbitrarily chosen both subset relations hold and thus the sets are equal

Q.E.D.

Claim: $P(A) \cup P(B) \subseteq P(A \cup B)$

Proof. Contrapositive

Suppose $x \notin P(A \cup B)$

Then $x \notin \{\emptyset, A, B, A \cup B\}$

$x \notin A \wedge x \notin B$

$x \notin P(A) \wedge x \notin P(B)$

$x \notin (P(A) \cup P(B))$

So it follows that $P(A) \cup P(B) \subseteq P(A \cup B)$

Since x was arbitrarily...

Q.E.D

$A = \{x \in \mathbb{Z} : \exists y \in \mathbb{Z}(\frac{1}{2}x = y)\}$ $B = \{x \in \mathbb{Z} \mid \exists y \in \mathbb{Z}(x = 2y + 1)\}$

multiples of 2 $\{0, 2, 4, 6, 8 \ldots\}$ odd numbers $\{1, 3, 5, \ldots\}$

$f : A \to B$, with $f(x) = x+1$

Cardinality of sets proof

Proof.

Consider the function $f(x) = x+1$. This function is a bijection thus for every $y \in B$ there is one element $x \in A$ such that $f(x) = y$ (Surjective), thus $|A| \geq |B|$. Also, for every $x \in A$ there is one element $y \in B$ such that $f(x) = y$ (injective), thus $|B| \geq |A|$. Thus $|A| = |B|$, which means A and B have the same cardinality.

Q.E.D

Structural Induction:

Recursively defined set C

Base: $3 \in C$

Recursion: If $c \in C$, then $3c \in C \wedge -3c \in C$

Restriction: Nothing is in C except what can be derived using the above rules

Claim: $\forall n : n \in C \rightarrow n$ is odd

Proof. We use property P(n) to denote that an n generated using the definition above is odd.

Basis: $3 = 2k+1$ where k is an integer, 1, therefore 3 is odd.

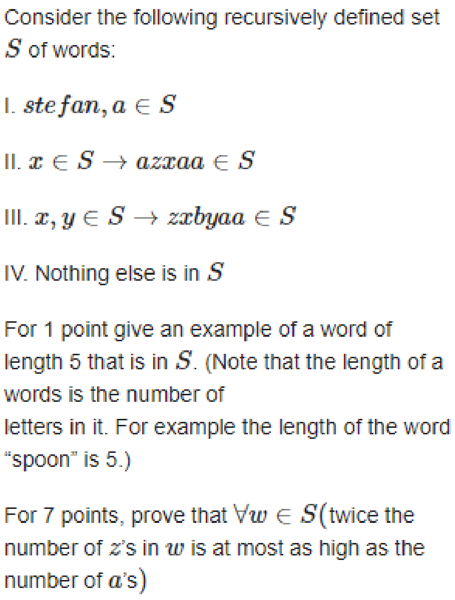
Inductive step: Suppose that we have an arbitrary element $p \in C$ that is odd. Then $p = 2q+1$ for some integer q

$$\begin{aligned} 3p &= 3(2q+1) \\ &= 6q+3 \\ &= 6q+2+1 \\ &= 2(3q+1)+1 \\ &= 2l+1 \text{ for some integer } q \end{aligned}$$

Therefore 3p is odd. Without loss of generality, $-3p$ is also odd

Since p and q are arbitrary integers, P(n) holds for any n constructed using the recursion rule of the recursive definition above

According to the principle of induction, P(n) holds for all $n \in C$



Consider the following recursively defined set S of words:

I. $stefan, a \in S$

II. $x \in S \to azxaa \in S$

III. $x, y \in S \to zxbyaa \in S$

IV. Nothing else is in S

For 1 point give an example of a word of length 5 that is in S. (Note that the length of a words is the number of letters in it. For example the length of the word "spoon" is 5.)

For 7 points, prove that $\forall w \in S$(twice the number of z's in w is at most as high as the number of a's)

Claim: $\forall w\,(2 \cdot f_z(w) \leq f_a(w))$

with $f_z(w) =$ number of z's in w and same for f_a

Proof.

Base: $2 \cdot f_z(w) \leq f_a(w)$

Case 1: $2 \cdot f_z(stefan) \leq f_a\, stefan$

$2 \cdot 0 \leq 1$, holds

Case 2: $2 f_z(a) \leq f_a(a)$

$0 \leq 1$, holds

Inductive step: Suppose that we have an arbitrary word $k, l \in S$ such that the claim holds.

That is $2 f_z(k) \leq f_a(k)$ (IH) and the same for l without loss of generality.

Then it's triggered new recursively added words will be:

Case 1: $azkaa$

Case 2: $azlaa$

Case 3: $zkblaa$

Case 1: $2 \cdot f_z(azkaa) \leq f_a(azkaa)$

$$2\,(3 f_z(a) + f_z(z) + f_z(k)) \leq 3 f_a(a) + f_a(z) + f_a(k)$$

$$2\,(0 + 1 + f_z(k)) \leq 3 + 0 + f_a(k)$$

$$2 + 2 f_z(k) \leq 3 + f_a(k)$$

$$2 f_z(k) \leq f_a(k) + 1$$

Apply IH:

$$\frac{-\;2 f_z(k) \leq f_a(k) + 1}{0 \leq 1}$$, which holds

Without loss of generality, the same applies to $azlaa$

Case 3:

$$2 \cdot f_z(zkblaa) \leq f_a(zkblaa)$$

$$2 \cdot (f_z(z) + f_z(k) + f_z(b) + f_z(l) + 2 f_z(a) \leq f_a(z) + f_a(k) + f_a(b) + f_a(l) + 2 f_a(a)$$

$$2\,(1 + 2 f_z(k)) \leq 2 + 2 f_a(k)$$

$$4 f_z(k) + \cancel{2} \leq 2 f_a(k) + \cancel{2}$$

$$2 f_z(k) \leq f_a(k)$$

Apply IH:

$$-\;2 f_z(k) \leq f_a(k)$$

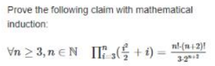
$$0 \leq 0$$

which holds.

Since k and l were arbitrarily chosen, according to mathematical induction, it holds to all defined elements.

Q.E.D.

WEAK INDUCTION



Prove the following claim with mathematical induction:

$\forall n \geq 3, n \in \mathbb{N} \quad \prod_{i=3}^{n}(\frac{i^2}{2}+i) = \frac{n!\cdot(n+2)!}{3\cdot 2^{n+2}}$

Proof.

Let P(n) denote $\prod_{i=3}^{n}\left(\frac{i^2}{2}+i\right) = \frac{n!(n+2)!}{3\cdot 2^{n+2}}$

Base case: P(3) gives

$$\prod_{i=3}^{3}\left(\frac{i^2}{2}+i\right) = \frac{3!(3+2)!}{3\cdot 2^{3+2}}$$

$$\frac{3^2}{2}+3 = \frac{6\cdot 5\cdot 4\cdot 3\cdot 2\cdot 1}{3\cdot 2^5}$$

$$\frac{9}{2}+\frac{6}{2} = \frac{6\cdot 5\cdot 4\cdot 2}{2^5}$$

$$\frac{15}{2} = \frac{240}{32} = \frac{15}{2}$$

which holds.

Inductive step:

We will show that, given any integer k ≥ 3 P(k)→P(k+1)

Suppose P(k) holds (the inductive hypothesis) so that

$\prod_{i=3}^{k}\left(\frac{i^2}{2}+i\right) = \frac{k!(k+2)!}{3\cdot 2^{k+2}}$ holds.

Then

$$\prod_{i=3}^{k+1}\left(\frac{i^2}{2}+i\right) = \prod_{i=3}^{k}\left(\frac{i^2}{2}+1\right)\cdot\left(\frac{(k+1)^2}{2}+k+1\right)$$

Apply IH:

$$= \frac{k!(k+2)!}{3\cdot 2^{k+2}}\cdot\frac{(k+1)^2}{2}+k+1$$

$$= \frac{k!(k+2)!}{3\cdot 2^{k+2}}\cdot\frac{k^2+4k+3}{2}$$

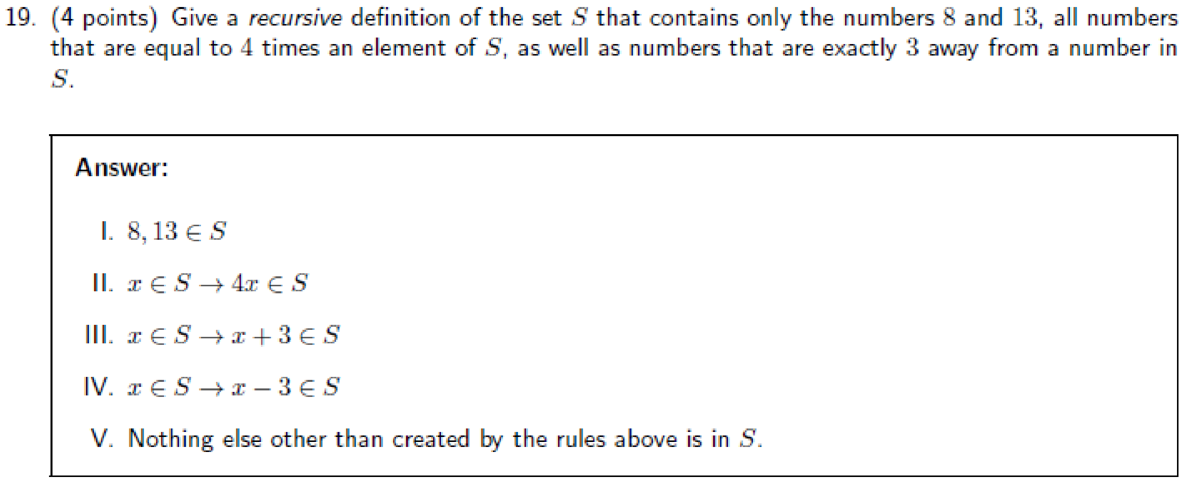
$$= \frac{k!(k+2)!}{3\cdot 2^{k+2}}\cdot\frac{(k+3)(k+1)}{2}$$

$$= \frac{(k+1)!(k+3)!}{3\cdot 2^{k+3}}$$

Since k was arbitrarily chosen it holds for all integers ≥3

By the principle of induction the claim holds.

Q.E.D



19. (4 points) Give a *recursive* definition of the set S that contains only the numbers 8 and 13, all numbers that are equal to 4 times an element of S, as well as numbers that are exactly 3 away from a number in S.

Answer:

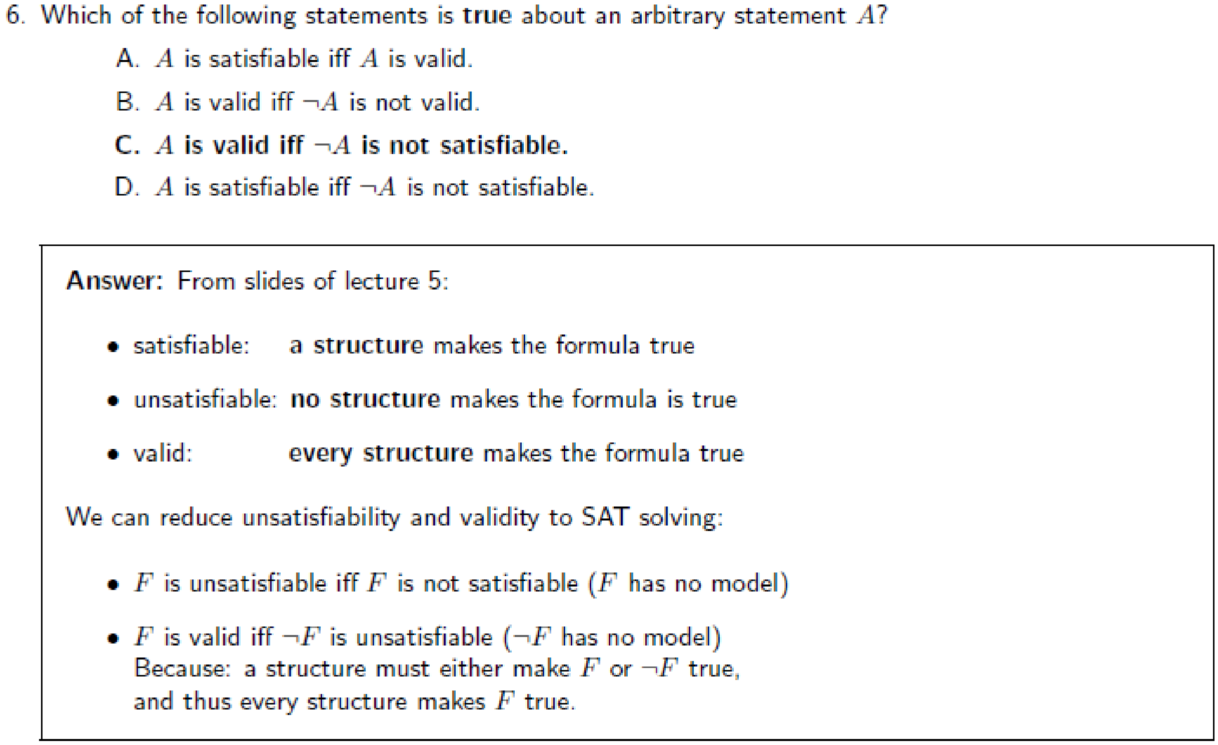
I. $8, 13 \in S$

II. $x \in S \to 4x \in S$

III. $x \in S \to x + 3 \in S$

IV. $x \in S \to x - 3 \in S$

V. Nothing else other than created by the rules above is in S.



6. Which of the following statements is **true** about an arbitrary statement A?

A. A is satisfiable iff A is valid.

B. A is valid iff $\neg A$ is not valid.

C. A is valid iff $\neg A$ is not satisfiable.

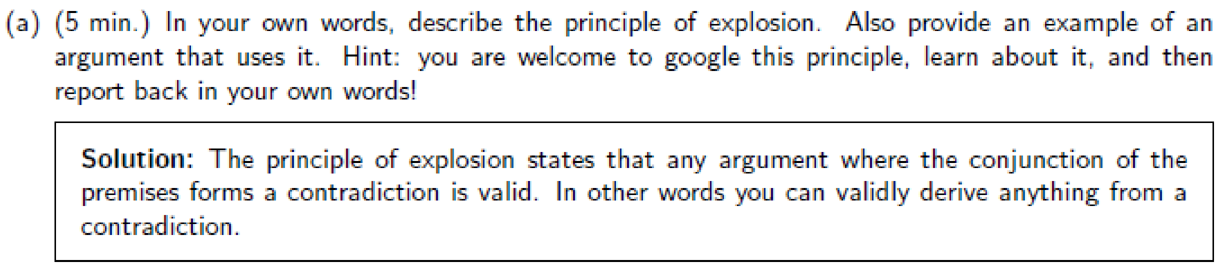
D. A is satisfiable iff $\neg A$ is not satisfiable.

Answer: From slides of lecture 5:

- satisfiable: a **structure** makes the formula true
- unsatisfiable: **no structure** makes the formula is true
- valid: **every structure** makes the formula true

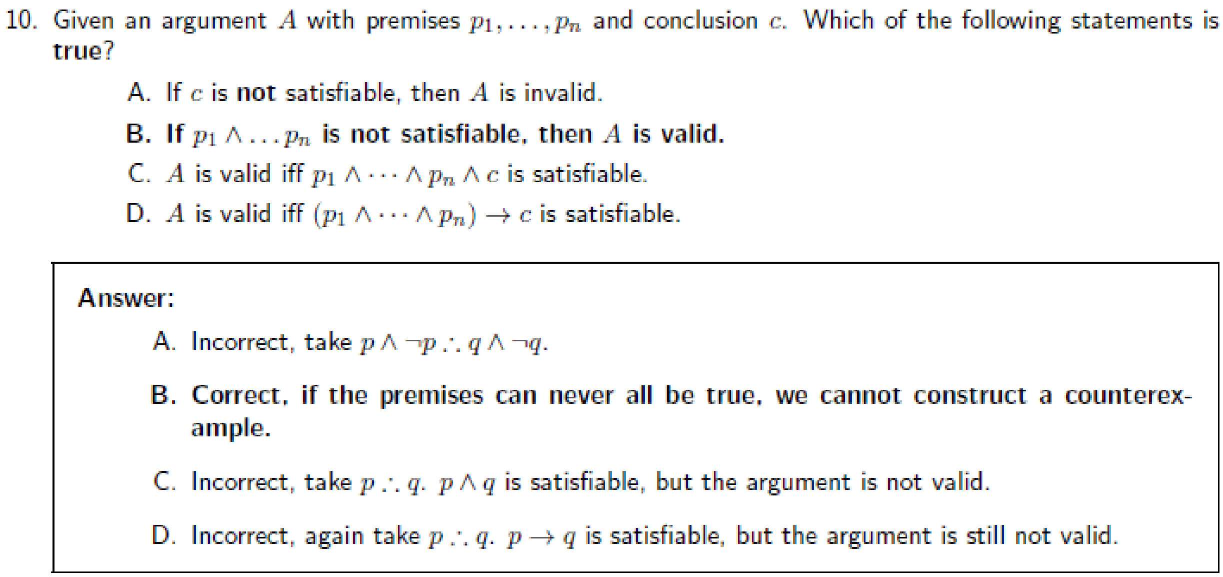
We can reduce unsatisfiability and validity to SAT solving:

- F is unsatisfiable iff F is not satisfiable (F has no model)
- F is valid iff $\neg F$ is unsatisfiable ($\neg F$ has no model)
  Because: a structure must either make F or $\neg F$ true,
  and thus every structure makes F true.



(a) (5 min.) In your own words, describe the principle of explosion. Also provide an example of an argument that uses it. Hint: you are welcome to google this principle, learn about it, and then report back in your own words!

Solution: The principle of explosion states that any argument where the conjunction of the premises forms a contradiction is valid. In other words you can validly derive anything from a contradiction.



10. Given an argument A with premises $p_1, \ldots, p_n$ and conclusion c. Which of the following statements is true?

    A. If c is **not** satisfiable, then A is invalid.

    B. If $p_1 \wedge \ldots p_n$ is not satisfiable, then A is valid.

    C. A is valid iff $p_1 \wedge \cdots \wedge p_n \wedge c$ is satisfiable.

    D. A is valid iff $(p_1 \wedge \cdots \wedge p_n) \rightarrow c$ is satisfiable.

Answer:

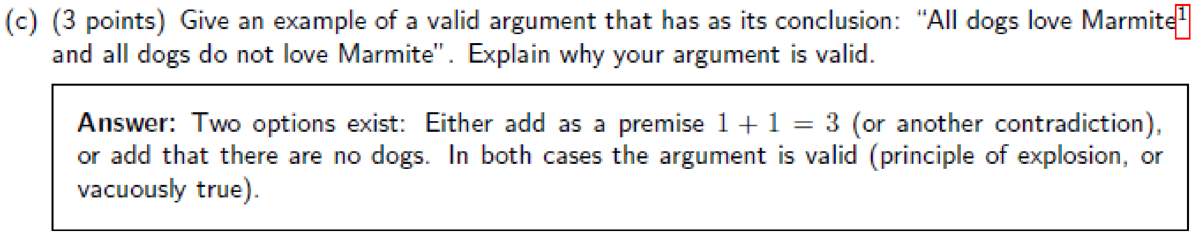
A. Incorrect, take $p \wedge \neg p \therefore q \wedge \neg q$.

B. Correct, if the premises can never all be true, we cannot construct a counterexample.

C. Incorrect, take $p \therefore q$. $p \wedge q$ is satisfiable, but the argument is not valid.

D. Incorrect, again take $p \therefore q$. $p \rightarrow q$ is satisfiable, but the argument is still not valid.

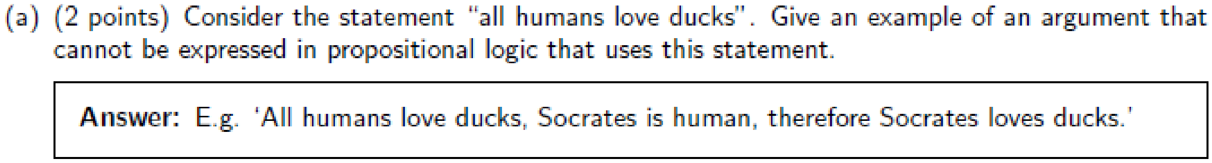
But if it has no premises at all, then the only way to be valid is if the conclusion is a tautology.



(c) (3 points) Give an example of a valid argument that has as its conclusion: "All dogs love Marmite[1] and all dogs do not love Marmite". Explain why your argument is valid.

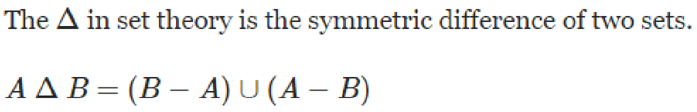
Answer: Two options exist: Either add as a premise $1 + 1 = 3$ (or another contradiction), or add that there are no dogs. In both cases the argument is valid (principle of explosion, or vacuously true).

“this argument is not sound as 2≠3, but because of this contradiction the argument is valid”



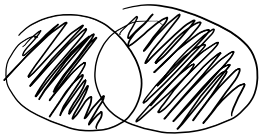
(a) (2 points) Consider the statement "all humans love ducks". Give an example of an argument that cannot be expressed in propositional logic that uses this statement.

Answer: E.g. 'All humans love ducks, Socrates is human, therefore Socrates loves ducks.'



The Δ in set theory is the symmetric difference of two sets.

$$A \,\Delta\, B = (B - A) \cup (A - B)$$



(a) (8 points) Consider the recursive definition of the set A:

I. $3 \in A$, $15 \in A$

II. $x \in A \rightarrow 8x + 24 \in A$

III. $x, y \in A \rightarrow 2x - 7y \in A$

IV. Nothing other than created by the rules above is in A.

Prove that every number in A is divisible by 3.

Answer:

Proof. Proof by structural induction:

- Base cases: $3 = 3 \cdot 1$ and $15 = 3 \cdot 5$, thus $3 \mid 3$ and $3 \mid 15$ both hold.
- Inductive step: Take arbitrary $k, m \in A$, such that $3 \mid k$ and $3 \mid m$ (IH). To prove: $3 \mid 8k + 24$, $3 \mid 2k - 7m$.

$$\begin{aligned} 8k + 24 &\overset{\text{IH}}{=} 8(3c) + 24 \\ &= 3(8c + 8) \\ &= 3d \\ 2k - 7m &\overset{\text{IH}}{=} 2(3c) - 7(3d) \\ &= 3(2c - 7d) \\ &= 3e \end{aligned}$$

So $3 \mid 8k + 24$ and $3 \mid 2k - 7m$.

Thus by the principle of induction it holds for all elements of A that they are divisible by 3. QED

Note: Many people lost one or two points for the IH here. You should include the following notions:

- k, m are arbitrary
- $k, m \in A$
- $3 \mid k$ and $3 \mid m$
- or alternatively $\exists c, d \in \mathbb{Z}$ such that $k = 3c$ and $m = 3d$

3. (1 point) Consider the argument: $A, B \therefore C$. Which of the following methods can we use to prove the validity of the argument?

A. Show that $B \wedge C$ is a contradiction.

B. Show that $\neg A \wedge C$ is a contradiction.

[✓ C.] Show that $B \wedge \neg C$ is a contradiction.

D. Show that $\neg A \wedge \neg B$ is a contradiction.

16. (1 point) Consider the following description of a function f. f takes a function g and an integer and returns a fraction. g is a function that takes a real number and an integer and returns a fraction and an integer. Which of the following describes the function f formally?

[✓ A.] $f : (\mathbb{Q} \times \mathbb{N})^{\mathbb{R} \times \mathbb{N}} \times \mathbb{N}$ -

B. $f : (\mathbb{R} \times \mathbb{N})^{\mathbb{Q} \times \mathbb{N}} \times \mathbb{N}$ -

C. $f : (\mathbb{Q})^{\mathbb{R} \times \mathbb{N}} \times \mathbb{N} \rightarrow \mathbb{Q}$

D. $f : (\mathbb{Q})^{\mathbb{Q} \times \mathbb{N}} \times \mathbb{N} \rightarrow \mathbb{R}$

$g : R \times N \rightarrow Q \times N$

$f : (Q \times N)^{R \times N} \times N \rightarrow Q$

(b) (4 points) The set T contains the fraction $\frac{1}{2}$. Furthermore every number t in the set can be written as: $t = \frac{2^k}{2^l}$ where $k, l \in \mathbb{N}$.

Answer:

I. $\frac{1}{2} \in T$

II. $\forall x : x \in T \rightarrow 2x \in T$

III. $\forall x : x \in T \rightarrow x/2 \in T$

IV. Nothing else is in T.

6. (1 point) Which of the following **cannot** be used to show that two propositions P and Q are equivalent?

A. Show that $(P \rightarrow Q) \wedge (Q \rightarrow P)$ is a tautology.

B. Show that $\neg Q \rightarrow \neg P$ and $\neg P$ are both contradictions.

C. Show that $(P \rightarrow R) \wedge (R \rightarrow Q) \wedge (Q \rightarrow P)$ is a tautology.

D. Show that P is a contradiction and that $\neg Q$ is a tautology.

Answer:

A. This shows that $P \leftrightarrow Q$ is a tautology, which is a correct method to show that $P \equiv Q$.

B. This is nonsense, it shows that P and $\neg Q \wedge P$ are tautologies. This in fact proves that Q is a contradiction and that P is a tautology, so they are not equivalent.

C. This proves that $P \leftrightarrow Q$ ($P \rightarrow Q$ due to transitivity, and $Q \rightarrow P$ is already listed) is a tautology, which is a correct method to show that $P \equiv Q$.

D. This implies that Q is also a contradiction and as all contradictions are equivalent, this shows that $P \equiv Q$.

(a) (5 points) Give a recursive definition of the set A that only contains the number 120, and for any numbers in the set, the set also contains:

- all of the factors/divisors of those numbers.
- all of the products of those numbers.

Answer:

I. $120 \in A$.

II. if $x \in A$, then $\forall y(y \mid x \to y \in A)$.

III. if $x, y \in A$, then $x \cdot y \in A$.

IV. Nothing else is in A other than the numbers constructed with the rules above.

(b) (8 points) Consider again the recursive definition from Question 13. Prove the following claim: Each word in S contains an odd number of a's.

Answer:

Proof. Define $f(x)$ to return the number of a's in x.
To prove: $\forall x \in S(2 \nmid f(x))$.

Base case ($x = a$): $f(a) = 1$ and $2 \nmid 1$.
Inductive step:
Take arbitrary $x, y \in S$ and assume $2 \nmid f(x) \wedge 2 \nmid f(y)$ (IH).
To prove: $2 \nmid f(xi)$, $2 \nmid f(axa)$, $2 \nmid f(ixiyixi)$.
$f(xi) = f(x) = 2m + 1$ by the IH.
$f(axa) = 2 + f(x) = 2 + 2m + 1 = 2(m + 1) + 1$ by the IH.
$f(ixiyixi) = 2f(x) + f(y) = 2(2m + 1) + 2n + 1 = 4m + 2 + 2n + 1 = 2(2m + n + 1) + 1$ by the IH.
So by the principle of induction, it holds that all words in S have an odd number of a's in it. QED

(a) (2 points) Describe in your own words the two differences between functions and relations. Answer in at most 5 lines.

Answer: In relations we can map a single object in the domain to multiple objects in the range. This is not allowed in a well-defined function.