1/19/23

The Towers of Hanoi

- Move tower from peg A to C
- One disk can move at a time
- Never place a larger disk atop a smaller one

Questions:

- Is there always a solution?
- How many moves are needed for a tower of n disks?
- Is there a concise program for this problem?
- How do I prove that the program correctly solves the program?

- **Graph theory:** 4 colors in US Map where states that touch each other must be different colors
- **Number theory:** securing communications channels
  - Account for eavesdropping
  - Ensuring that a message was sent from the right party (spoofing / phishing)
  - Man-in-the-middle attacks - interfering with messages sent / interactions

What is a "proof"?

- Proof means that something has been examined or evaluated
- Way of establishing a fact

What is truth and how is it established?

- ONLY two values when evaluating statements - True or False
- **Proposition:** declarative statement that is either TRue or False
  - EX: 4 + 3 = 7, 1 + 1 = 3, "I would like an A in CS135.", "Humans are mortal."
  - NOT EX: "Give me an A!"
    - There is no True or False answer to this statement - not declaring anything, more of a command
  - NOT EX: X + 1 = 2
    - You do not know what X is
      - If you stated that X is a number, then it could be a proposition
  - NOT EX: "This proposition is true.", "This proposition is false."
    - Has to be both false and true - if it's true then it's false, if it's false then it's true

- Self-referencing statement

Proposition	Not Proposition
4 + 3 = 7	"Give me an A!"
1 + 1 = 3	X + 1 = 2
1 + 1 = 3	"This proposition is true." (self-referencing statement)
"I would like an A in CS135."	"This proposition is false." (self-referencing statement)
"Humans are mortal."	

1/24/23

- Propositions must be either True or False
- We are not interested in what the statements mean themselves, but the form of the arguments
- Compound propositions: negation, conjunction, disjunction, implication
- Truth tables: tautologies, contradictions, logical equivalence
- If G is a proposition, then the negation of G ($\neg G$) is a proposition
  - EX: G = "I am grumpy.", $\neg G$ = "It is not the case that I am grumpy." / "I am not grumpy."
- Double negation - two negatives make a positive
- Conjunctions use $\wedge$ → only True if both sides of the condition are True
- Disjunctions use $\vee$ → True if either side of the $\vee$ is True, True if both sides are True
  - Exclusive OR is only or the other
  - Inclusive OR can have both be True and still have a True output
- Implications: G implies H is if G then H ($G \Rightarrow H$)
  - Always True until H is False at the same time that G is True
    - AKA antecedent is True but consequence is False
      - As soon as the antecedent is False, the implication is True
- Tautology = proposition that is always True

    - EX: True, “It is raining or it is not raining.”, 2 = 1+1
- Logical contradiction: proposition that is always False
    - Ex: False, 1 > 1, $Q \land \neg Q$
- Logically Equivalent: two prepositions have the same truth tables ($P \equiv Q$)
    - EX: $\neg(p \land q) \equiv (\neg p \lor \neg q)$
- Number of columns in truth table = 2^(number of variables)
- How to prove that two propositions are not equivalent?
    - Come up with counterargument
    - See if everything matches in their truth tables - the part(s) that do not match are counterexamples to the question of whether the propositions are equivalent
- Law of conditional identity: $(p \Rightarrow q) \equiv (\neg p \lor q)$

Idempotent laws:	$p \lor p \equiv p$	$p \land p \equiv p$
Associative laws:	$(p \lor q) \lor r \equiv p \lor (q \lor r)$	$(p \land q) \land r \equiv p \land (q \land r)$
Commutative laws:	$p \lor q \equiv q \lor p$	$p \land q \equiv q \land p$
Distributive laws:	$p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$	$p \land (q \lor r) \equiv (p \land q) \lor (p \land r)$
Identity laws:	$p \lor F \equiv p$	$p \land T \equiv p$
Domination laws:	$p \land F \equiv F$	$p \lor T \equiv T$
Double negation law:	$\neg\neg p \equiv p$	
Complement laws:	$p \land \neg p \equiv F$ $\neg T \equiv F$	$p \lor \neg p \equiv T$ $\neg F \equiv T$
De Morgan's laws:	$\neg(p \lor q) \equiv \neg p \land \neg q$	$\neg(p \land q) \equiv \neg p \lor \neg q$
Absorption laws:	$p \lor (p \land q) \equiv p$	$p \land (p \lor q) \equiv p$
Conditional identities:	$p \rightarrow q \equiv \neg p \lor q$	$p \leftrightarrow q \equiv (p \rightarrow q) \land (q \rightarrow p)$

-
    - Do not have to memorize names of these laws, but practice and be comfortable with them

“If I attend lectures then I will do well in CS135”
“If I do my homeworks ten I will do well in CS135”

“If I attend lectures and do my homeworks then I will do well in CS135” → is this equivalent to both above?? NO

1/26/23

- To solve for whether two propositions are equivalent, you can

    - Construct truth tables
    - Use laws to find equivalence
- The Contrapositive
    - $P \Rightarrow Q$
    - $\equiv \neg P \lor Q$ (conditional identity)
    - $\equiv Q \lor \neg P$ (commutative)
    - $\equiv \neg\neg Q \lor \neg P$ (double negation)
    - $\equiv \neg Q \Rightarrow \neg P$ (conditional)
- Logical reasoning
    - $(H1 \land H2) \Rightarrow C$
        - Must be a tautology to conclude that a argument is valid
    - An argument is valid if and only if its implication is a tautology
- Tree Method (finding counterexamples)
    - Write down each premise
    - Negate the conclusion
    - Look for a counter-example
        - If all leaves are checked off: <u>no counterexamples exist</u>
        - Else, if some leaf if not blocked off but all compound propositions are checked off: <u>counterexamples exist</u>
        - If there is an unchecked compound proposition, expand it <u>at every leaf below</u> and check off the proposition
        - Close off every lead whose path to the top contains a contradiction

<u>1/31/23</u>

- Tree method notes
    - You continue expanding at every root that is open one by one
    - If any of the roots are open when the problem is done, counterexamples exist
        - Goal is to make roots get canceled
        - Counter-example is a traceback up the root - look at parts that made it cancel and combine
    - Make sure to check your answers afterwards
    - You do not need to use all the hypotheses to conclude that an argument is valid

- If an argument is valid, it does not mean that the conclusion is correct!!
- Scheme notes
  - Identifiers are like variables
    - They can have a variety of variables
  - Identifiers = names assigned to quantities
    - They have value - they must be assigned values when printed etc. or there will be an error
  - Expressions
    - Atoms: constant (number, boolean value), variable
    - Lists: made up of Atoms and other lists
      - Atoms and Lists can be nested inside Lists as much as you want
  - Evaluating expressions
    - Every expression has a value
      - The value of #t is #t etc.
      - The value of any variable x is whatever it was defined as
  - Once you assign a value to an expression, you can't change it
  - First element of list/expression should be a defined function
    - EX: (+1 2) gives 3
    - EX: (*2 4) is 8
    - EX: (+ (* 2 5) (*3 4)) = (+ 10 12) = 22
  - Functions
    - (define (f x y) (+ (* x x) (* y y)))
  - quote suppresses evaluation of its argument
  - Functions
    - car returns the first element of a list
    - cdr returns the elements of a list excluding the first
    - cons = construct, adds element to start of list
  - Lists
    - A list is stored in memory as a sequence of <u>cons</u>tructor boxes
    - (1 2 3) = (cons 1 (cons 2 (cons 3 '())))
    - car = address part of register
    - cdr = decrement part of the register

2/2/23

- Propositions only deal with nouns
- Predicates: the part of a sentence or clause containing a verb and stating something about the subject
    - Have arguments and variables (x,y) of the predicate
    - Are used in identifying patterns
    - Value is True / False
    - EX: $T(x,y) \wedge T(y,z) \rightarrow T(x,z)$
        - Generalizing names to variables to work in all situations
- Quantifiers
    - $\forall x$, $\forall y$, $\forall z$
        - "For all x", "for all y", "for all z"
    - Binds each variable to the domain of discourse
        - Giving information about the value of that variable
    - Every variable in an expression MUST be quantified
    - "Free variables" are variables that are not quantified
- Predicates over numbers
    - EX: $1+... +n = n(n+1) / 2$
- Predicates and quantifiers
    - Possible values are combined with $\vee$

2/7/23

- Quantified propositions declare a domain
- Universal instantiation
    - If you have a universally quantified proposition, then you can plug in any value for x and it will be True
- Existential instantiation
    - a rule of inference which says that, given a formula of the form , one may infer for a new constant symbol c
- Universal generalization
- Existential generalization
    - Generalization statements make conclusions about P(x)'s domain

2/9/23

- Scheme

  - Don’t use cons on an atom or you will get a dotted expression as a result
  - Conditionals
    - (eq? x y) - tells you whether x and y are equal or not equal
    - (if cond expr1 expr2)
    - Conditional expression has a bunch of conditions are arguments
  - Recursive functions
    - Define the value on a bigger argument by defining the value on a smaller argument and doing minimal work
    - (cdr L) - if L is empty, cdr is undefined
    - You can have a list of conditions for recursive functions
- Sets
  - Φ - means that the set is empty
  - {{Φ}} has one member
  - A set cannot contain itself
    - If it does contain itself, it fits into neither the group of sets that contain themselves nor the group of sets that don’t contain themselves
    - If A contains B then B cannot contain A.

★ Did recursion exercises in Lecture 7

2/14/23

- A ⊆ B means that every member of A is also a member of B
- A ⊂ B means that every member of A is a member of B, but B has elements that are not members of A
- Set notation consists of symbols used to represent real numbers, natural numbers, etc.
- Union = ∪ (similar to or)
- Intersection = ∩ (similar to and)
- Complement EX: $\bar{A}$ (similar to negation of A)
- Cartesian Product: combination of all elements in two sets
- (1, a) ≠ (a, 1) → not commutative
- Power set = the set of all subsets of S
- Set identities and laws of propositional logic are very similar
- $A - B \equiv (A \cap \overline{B})$
- A relation R with domain and range B is a subset of A x B

- A relation R over set A is a subset of A x A
- Relations properties
  - Reflexive - every element points back to itself
  - Anti-reflexive - no element points back to itself
  - Symmetric - (x, y) ←> (y, x)
  - Anti-symmetric - x and y are not interchangeable
  - Transitive - (x, y) → (y, z) → (x, z)
- Relations can be symmetric but not reflective and vice-versa
  - Symmetric and reflexive are diff properties
- Equivalence relations: reflexive, symmetric, and transitive

2/16/23

- Equivalence relations: reflexive, symmetric, and transit
- Reflexive relations need every element to point to itself
  - Reflexive closure: smallest reflexive relation
- Symmetric closures are the smallest symmetric relation
  - Needs (a, b) and (b, a)
- Transitive: if you have two arrows, they should be connected in some way (like a triangle)
- Composing relations
  - R = direct flights, R^2 = one-stop flights, R^3 = two-stop flights, etc.
    - R^k gives k - 1 stops
- Functions
  - A relation where every element in the domain has exactly one arrow coming out of it
    - No restriction on where the arrow leads
  - One-to-one: every domain element is mapped to a unique element in the target
    - Every element of domain is matched
  - Surjective: every element in the target is the target of at least one domain element
    - Every element of target is matched
  - Bijective or one-to-one correspondence: every domain element is matched with exactly one element in the target and vice versa
    - Domain size = range size
    - Every element is matched up
  - EX:
    - f(x) = x^2 : one-to-one but not onto

        - f(x) = x^2 -1 : NOT A FUNCTION
        - f(x) = (x-1)^2 : not one-to-ne, not onto
        - f(x) = x : one-to-one and onto
- Pigeonhole Principle
    - If k+1 pigeon occupy pigeonholes, then at least 2 pigeons share a pigeonhole
    - No function from a domain size k+1 to a target of size k is injective
- Well-ordering principle
    - Every non-empty subset of N has at least one element (least element)
- Pigeonhole principle = well-ordering principle
- Proof techniques
    - Direct method
    - Proof by contradiction

2/21/23

- What does it mean for 2 sets to be the same size?
    - Bijective = injective and subjective
    - Associate every member of A with a unique member of B
    - |A| = |B| if and only if there is a bijection from A to B
- A set S is countable if there is an injective function f : S → N
    - Every finite set is countable
    - Every subset of N is countable
    - If we can list all the elements of an infinite set without repetition, then the set is countably infinite
- A set is countably infinite if there is a bijective function f : N → S
- Countability
    - N x N is countable
    - The set of Q of Rationals is countable
- The power set of the set of natural numbers (N) has a cardinality that is a larger infinity than the cardinality of N
- Mapping subsets to binary strings
    - {0,2) = 10100000000
    - {0,2,4,5} = 1010110000
    - Odd = 010101010101010
    - Even = 101010101010101
        - Every subset has a unique string

- Cardinality of power sets → infinite hierarchy of infinite sets
- Cardinality of sets of infinities (power sets of power sets) → cantor's continuum hypothesis

2/23/23

- Propositions have quantifiers in front of them
- To prove that P(x) is true for all natural numbers:
  - Step 1: basis, show that P(0) is True
  - Step 2: inductive hypothesis, state P(k), k is an arbitrary number
  - Step 3: inductive step, show that P(k) → P(k+1) is true for all k
- Induction: assuming at P(k) is true, we can conclude P(k+1) is true etc.

2/28/23

- Lem's proof
  - The hypothesis should all be true and the conclusion should be true as a result
- Induction:
  - Base case
  - Inductive hypothesis
  - Inductive step
  - Conclusions
- Hole agriculture question
  - You need to place a hole in each of the 4 courtyards to do induction
  - Use the 3 pieces to touch each quadrant
- Chocolate question
  - Must expand because one cut may not be the end result
    - 80 pieces, one cut does not break all the pieces individually
- Strong induction: hypothesis has more info + assume more;

Now that we know how standard induction works, it's time to look at a variant of it, strong induction. In many ways, strong induction is similar to normal induction. There is, however, a difference in the inductive hypothesis. Normally, when using induction, we assume that $P(k)$ is true to prove $P(k+1)$. In strong induction, we assume that all of $P(1), P(2), \ldots, P(k)$ are true to prove $P(k+1)$.

- https://brilliant.org/wiki/strong-induction/

3/2/23 (zoom) + 3/7/23

- Proofs by strong induction
- Steps for induction
    - Basis
    - Inductive hypothesis
    - Inductive step
- Normal fibonacci function
    - Makes the same calculations over and over again - very slow!
        - Solution: pass completed calculations as arguments of the function to keep it iterative but more practical
            - Tail recursion

3/21/23 + 3/23/23

- Greatest common divisor (gcd) - greatest common factor between two numbers
- Linear combination EX: $\{12s + 30t : s, t \in z\}$
    - Find the smallest positive linear combination using the equation made!
- GCD Theorem: The smallest positive linear combination m of two integers a, b (at least one of which is non-zero) equals g = gcd(a, b)
    - EX: The smallest positive linear combination of 12, 30 is gcd(12, 30) = 6

  Proof of GCD included in slides
- Euclid's algorithm for GCD - repeated use (recursive) of GCD theorem formula until one of the values used equals 0

3/28/23

- Perfect numbers
- Prime numbers: not divisible by any number smaller than it that is greater than 1
- How many numbers are prime?
    - Theorem: infinitely many primes
    - P = 1 + p1 + p2 + p… etc.

- Fundamental theorem of arithmetic: every number greater than 1 is uniquely expressed as a product of primes
    - Lemma in slides
- Modular arithmetic
    - r = rem(a, b)
    - DOES NOW FOLLOW standard arithmetic properties (may be true but also may not be true! - okay in some examples, not in others)
        - Add common term to both sides
        - Multiply by common term on both sides
        - Add equal numbers
        - Multiply equal numbers
    - When can you use arithmetic properties??
        - If gcd(c, m) == 1 in a * c = b * c (mod m)
            - Proof included in slides
- Congruence modulo m → is a relation (specifically an equivalence relation!!)
    - Equivalence relation
        - reflective , commutative, transitive
        - Equivalence classes are created by the relation
    - Congruence by itself is not a relation
    - Only if (a - b) is divisible into m → (a - b) is congruence modulo m
    - EX: m = 7, a = 25, b = 4
- Basic theorem example
    - r1 and r2 are equal so r1 - r2 is 0

3/30/23 + 4/4/23

- Modulus arithmetic does not always follow standard arithmetic
    - Using the pulverizer can be useful
- Examples in notebook
- A series of calculating remainders and a list of pairwise relatively prime moduli can be used in combination with the chinese remainder theorem to do arithmetic with larger numbers
- Theorems
    - Fermat’s lit
    - Euler’s totient function
    - Euler’s generalization of fermat’s little theorem
- Cryptosystem: hiding messages from outside people

4/6/23 + 4/11/23

- Pairwise relatively prime: choosing any two values, the gcd will be 1
- Chinese remainder theorem: representing numbers repeatedly up until m
    - Least common multiple = m = all values multiplied together
    - Combine solutions to individual pieces
    - Requirement: moduli are pairwise relatively prime
    - After the theorem, you still need to prove that the solution is unique
- CRT Proof of Uniqueness
    - Base case of lemma: = 1

4/13/23

- Graph Theory Intro 🙂
- Terminology:
    - Vertex = node
    - Edge = set of two nodes
    - Degree of a vertex = # of edges indecent to it
    - Walk : sequence of vertices and edges
        - Closed walk: same start and end node
        - Trail: a walk in which no edge is repeated
            - Circuit: closed walk combined with no edge repeated (closed walk + trail regulations)
        - Cycle: circuit of length >= with the same first and last vertices and no repeated vertex
    - DEFINITELY WILL BE ON TRUE/FALSE ON FINAL
- Directed graphs terminology
    - Outdegree = number of outgoing edges
    - Indegree = number of incoming edges
    - Length of walk: # of edges in the walk
    - Path: a walk with no repeated nodes
    - Cycle: walk that begins and ends at same node with no repeated nodes
- Strongly connected graph: a directed graph where there is a directed path from every node to every other node
    - If you can connect all the codes in a cycle, a graph is strongly connected

- Directed acyclic graph (DAG): directed graph that does not have a direct cycle to it
- Proofs use recursion :)

4/18/23 + 4/20/23

- Undirected graph: finite set V of vertices and a set of edges E = {e: e = {a,b}, a, b ∈ V}
- Property of a eulerian trail:
  - In an eulerian trail at every vertex, other than possibly the start and end vertices, has even degree.
- Connected graph: graph with connection between any two given vertices
- Color theorems
  - Proving 4, 5 colors is possible
  - It is impossible to prove / tell if 3 colors can be true
- Planar Graph: can be drawn in the plane without any crossing edges
- Complete graph: k3,3 or K5
- Every non-planar graph has a combo of K3,3 or K5
- Euler's formula in graphs only applies to connected planar graphs

4/25/23

- For every connected planar graph with n vertices, m edges, and r regions: n - m + r = 2
  - The number of regions in every planar graph is invariant
- Theorems:
  - For every connected planar graph G with n >= 3 vertices: m <= 3n - 6
  - Five color theorem: every planar graph can be colored with 5 or fewer colors
    - Proof by induction included in slides
  - Hall's theorem: a biparte graph G(A,B,E) has a perfect matching if and only if |N(S)| >= |S| for every subset S ⊆ A
- K5 and K3,3 are not planar → a graph that is not planar may not be planar because it may have K5 or K3,3 incorporated within it
- Look on slides for more info