Sets
Intersection, Union, Null Set
A ⊕ B (symmetric difference) = A ∪ B − A ∩ B
Addition principal - |A ∪ B ∪ C| = |A| + |B| + |C| − |A ∩ B| − |B ∩ C| − |A ∩ C| + |A ∩ B ∩ C|
Ā = U − A (complement of A)
Subsets
Sets containing other sets
Set builder notation
Cardinality

Functions and integers
Everywhere defined – everything in A is used and each element in A goes to only 1 element in B
Onto – every element in B can be gotten to with the function ie. Range(f) = B
1-to-1 – each element in B can be gotten to by at most one element in A
Invertible – 1-to-1 and onto
A function is invertible if its inverse \((f^{-1})\) is also a function
\(g ∘ f = (g ∘ f)(a) = g(f(a))\)
\(f ∘ g = (f ∘ g)(a) = f(g(a))\)
Floor – round down
Ceiling – round up
LCM = \(2^{\max(a,b)} \cdot 3^{\max(a,b)} \cdot 5^{\max(a,b)} \cdot \ldots \cdot \{\text{all primes}\}^{\max(a,b)}\)
GCD = \(2^{\min(a,b)} \cdot 3^{\min(a,b)} \cdot 5^{\min(a,b)} \cdot \ldots \cdot \{\text{all primes}\}^{\min(a,b)}\)
Euclid's algorithm – \(d = sa + tb\)
Base conversion – alternate using / and %

Sequences
Sequence – order matters
Finite – countable, has specific start and end points
Infinite – has no end point. The book calls it countable, but how do you count infinity?
Countable – can be arranged in a list, has a start
Uncountable – anything not countable, an example is all real numbers between 0 and 1
Recursive – element depends on previous values, may be infinite, but has a specific starting point
Explicit – element depends only upon itself, has a specific starting point
String – sequence of letters, set corresponding to a sequence
Regular Expressions – defining a set of strings

Counting and Probability
Multiplication principle
Permutation – order matters \(P(n,r) = n!/(n-r)!\)
Combination – order doesn't matter \(C(n,r) = n!/(r!(n-r)!))\)
Permutation with repeats – \(P(n,r) = n^r\)
Combination with repeats – \(C(n,r) = \) rewrite this one as a regular combination using \(C(n+r-1, r)\)
Event (E) – the desired outcome or combination of outcomes
Sample space (S) – all possible outcomes
Probability – \(P = |E|/|S|\)
Pigeonhole principle
Matrices – Boolean and Regular
Add, $\land, \lor$ - only exact same sizes
Multiply – $M \times N \times J \times K$ is possible only if $N=J$, result is a $M \times K$ matrix
Transpose – flip around the diagonal (first row becomes first column, etc.). Is symmetric if $A = A^T$
Identity matrix – binary matrix where the diagonal is all 1’s, all other values are 0, is always square
Inverse – only computable for a $2 \times 2$ matrix (bigger can be done, but not in this class)

Propositions and logical operations
Truth tables for logical operators
Statement – true or false declaration (not opinion, question, command, changing value, etc.)

Graphs
Matrix in-degree = number of arrows into node, number of 1's in the column
out-degree = number of arrows out of node, number of 1's in the row
Paths
Cycle – begin and end at the same vertex
Connectivity relation showing all paths of all lengths
$R^n$ path of length $n$

Relations
reflexive – $R$ is reflexive if $aRa$ for all $a$ in $A$
irreflexive – $R$ is irreflexive if $aRa$ for all $a$ in $A$
symmetric – $aRb$ and $bRa$
asymmetric – $aRb$ and $bRa$ (a$\neq$b or both 0, diagonal is 0)
antisymmetric – if $aRb$ and $bRa$ then $a=b$, else $aRb$ and $bRa$, or both 0
transitive – if $aRb$ and $bRc$ then $aRc$

Digraph representations of relations
Matrix representations of relations

Graphs
reflexive – all nodes need a cycle of length 1
irreflexive – no node can have a cycle of length 1
symmetric – all edges go both ways, cycles of length 1 are ok
asymmetric – no cycle of length 1, all edges are single path
antisymmetric – all edges between vertices are single path, cycle of length 1 is ok
transitive – if there is a path of length 2 from $a$ to $c$, passing through $b$, then there must also be a path of length 1 between $a$ and $c$. If no path of length 2 exists, it is still transitive.

Growth of Function
Big-Theta and Big-Oh Notation
**Trees**
root – first or top vertex in the tree, has a height of 0
leaf – bottom vertex, has 0 children
n-tree – all vertices have at most n children
complete – all vertices except leaves have the same number of children
balanced – height of all leaves differ by at most 1
sub-tree – any vertex of a tree may be partitioned off (with all children etc) to become a new tree

**Grammar**
\[ G=(V, S, v_0, \rightarrow) \]
machine
\( G \) – grammar
\( V \) – everything, similar to the universe
\( S \) – set of terminal symbols
\( N \) – set of non-terminal symbols
\( \rightarrow \) - the production

**Machine**
\[ M=(S, I, F) \]
\( M = (S, I, F) \) machine or (\( S, I, F, s_0, T \)) Moore
\( S \) – state set
\( I \) – input set
\( F \) – state transition function
\( T \) – terminal state set
\( s_0 \) – starting state