Lecture 2
(Pre-Requisite)

We started with Mr. Mofajjel asking a question:

- How did the self-loop “R” come?
- Sir:

  When we are at state “0”, we can add 5 HKD or 10 HKD or we can press “R” (Rotate knob).

  If we add 5 HKD or 10 HKD we go to state “5” or “10” respectively.

  If we’re at state “0” and we rotate, there is no change of state.

Programming assignments are optional. Therefore 100 + PA Bonus

Sets

Set is an unordered collection of elements based on some criteria (criteria can be “No Criteria”).

An “Ordered Set” can exist but there may be a different name for that.

Things in a set are called Elements.

A set can be written like this {7, 14, 21} -----(“First 3 multiples of 7”)

* ”is an element of $\in$ ” and “is not an element of $\notin$ ” notation shown*

* ”subset $\subset$” and “Proper Subset $\subseteq$” notation shown*

If A = {1,2,3,4,5} B = {2,4} C= {2,4}

B $\subset$ A : All elements of B are also part of A. Not all elements of A are in B. |B| < |A|.

C $\subseteq$ B : C and B are identical. C $\equiv$ B.

MULTISET : {7} and {7, 7} are different sets. (We won’t be using this much)

Infinite Set : Set of Natural Numbers, Integers etc. A = {1, 2, 3, ... }

FINITE SET : Countable number of elements. B = {1, 2, 3, ... , n} (e.g “till 100 or 1 mill” We need to somehow close this)

Countable/Uncountable infinity discussed

EMPTY SET, ϕ = {} $|\phi| = 0$

UNION :

A = {5, 7, 3} B = {x, T}

To combine these, we may make $C \ = \ A \ \cup B$ where C is a set of alpha-numeric characters. So, C = {5, 7, 3, x, T}

A = {1, 2, 3, 4, 5} B = {2, 4}

$$C = A \cup B = \{1, 2, 3, 4, 5\}$$

We don’t repeat duplicate values

C = $\{c\_i \mid c\_i \in A \, or \, c\_i \in B\}$

INTERSECTION :

Only the common elements are counted.

$$A \cap B = \phi$$

C = {1, 2, 3, 4, 5} D = {2, 4}

E = $C \cap D = \{2,\ 4\} = \{e\_i \mid e\_i \in A \, and \, e\_i \in B\}$

DIFFERENCE :

If A = {5, 7, 3} B = {x, T}

A – B = {5, 7, 3} = *Present in a but not in B*

If C = {1, 2, 3, 4, 5} D = {2, 4}

E = C – D = {1, 3, 5} = $\{e\_i \mid e\_i \in A \, and \, e\_i \notin B\ \}$

COMPLEMENT :

If $U = \{1, 2, 3, 4, 5\}$ and A = {1,2}

A' = {3, 4, 5} = $\{x | x \in \mathbf{U} \ and \ x \notin A\}$

VENN DIAGRAMS FOR EACH OPERATION SHOWN

SEQUENCES AND TUPLES

Looks similar to Sets, but the lists are <u>ordered</u>. Therefore (2, 1) ≠ (1, 2).

Notation : (7, 14, 21) -----First Brackets Used

Coordinates in a graph (x, y, z) are tuples. (3-tuples to be exact)

Tuples are finite. Sequences can be finite/infinite.

Strings are Sequences. "Definition" ≠ "Difenition".

POWER SET :

Set of all subsets of a set.

If A = {1, 2, 3},

$P_{\mathbf{A}}$ = {ϕ, {1}, {2}, {3}, {1,2}, {1, 3}, {2, 3}, {1, 2, 3}}

CROSS PRODUCT OR CARTESIAN PRODUCT :

It is a set of sequences.

If A = (1, 2) B = (x, y) C = (+, -)

A × B = {(1, x), (1, y), (2, x), (2, y)}

A × B × C = {(1, x, +), (1, x, -), (1, y, +), (1, y, -), (2, x, +), (2, x, -), (2, y, +), (2, y, -)}

COMMUTIVITY: $A \cup B \ = \ B \cup A \ and \ A \cap B \ = \ B \cap A.$

ASSOCIATIVITY: $(A \cup B) \cup C = A \cup (B \cup C)$ and $(A \cap B) \cap C = A \cap (B \cap C)$.

DISTRIBUTIVITY: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ and $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

GRAPHS :

A set of points (nodes/vertices), with some lines (edges) connecting them.

Nodes can have numbers, names or nothing. (e.g Facebook is a huge graph of people)

Edges can have weights (Weighted Graphs). We won't be too concerned with it, but FB is. (e.g how many interests do two people share or how many ways are two people related)

Machine Learning deals with Weighted Graphs.

DEGREE OF A NODE: How many edges are connected to a node.

PATH: Moving from one node to the other via edges (where the sequence is important)

NOTATION: G = (V, E), where G is a graph (a pair of vertices and edges), V is the set of vertices. E is the set of edges. Edges are tuples (Especially true for undirected graphs).

Therefore, a graph, G, can be = ({1, 2, 3, 4, 5}, {(1,2), (2,3), (3,4), (4,5), (5,1), (2,1), (3,2), (4,3), (5,4), (1,5)})

To avoid excess writing {(1,2), (2,1)}, we can state that we are dealing with an Undirected Graph and just write {(1,2)}.

SUBGRAPHS are a smaller part of a Graph.

TREES: A type of graph where each node has indegree of at most 1. The first node which has indegree 0 is called the root. The nodes with outdegree 0 are called leaves. Nodes that have both indegree and outdegree nodes are internal nodes. Trees only go down. Nodes that are in the same "level" are called siblings/cousins etc.

CYCLES: For directed graphs, when a path from a starting node leads back to the starting node. {(1,2), (2,1)} and {(1,2), (2,3), (3,4), (4,1)} are cycles.

CONNECTED GRAPHS: If two graphs have no nodes connected by an edge, they are not connected. (Strongly/weakly connected graphs not needed)

ALPHABETS AND STRINGS :

(Combination of sets and sequences)

ALPHABETS are a finite set of letters (more appropriately for this course: a collection of symbols). For computers, it's alphabets are 0 and 1.

STRINGS

They are a sequence of alphabets.

In this course Σ represents a finite set. And the set usually represents alphabets.

Σ_1 = {a, b, c, d, . . . , z}: the set of letters in English.

Σ_2 = {0, 1, . . . , 9}: the set of base-10 digits.

$\Sigma_3 = \{$a, b, c, d, $\ldots$, z, #$\} = \Sigma_1 \cup \{$#$\}$: the set of letters in English.

$\Sigma_4 = \Sigma_1 - \{$a, e, i, o , u$\}$: set of consonants.

More formally, STRINGS over alphabet Σ are a FINITE SEQUENCE of symbols in Σ.

abfbz is a string over $\Sigma_1 = \{$a,b,c,d,$\ldots$, z$\}$

ab#fbz is a string over $\Sigma_3 = \{$a,b,c,d,$\ldots$, z, #$\}$

abfbz $\neq$ afbbz.

EMPTY STRING is denoted, ε. (Empty Tuple is a 0-tuple. Notation: ϕ)

Σ^* is a set of all possible strings over Σ. Therefore $\Sigma_1^* = \{\varepsilon$, a, b, $\ldots$, aa, bb, $\ldots\}$ a Countably Finite set.

LANGUAGES :

A set of strings over the same alphabet.

L_1 = All strings that contain the SUBSTRING “to” over Σ_1 are “stop”, “to”, “toe”.

Since Strings are a sequence, all substrings of a string must have the same sequence.

Therefore $L_1 = \{x \in \Sigma_1^* : x$ contains the substring “to”$\}$

$L_2 = \{x \in \Sigma_2^* : x$ is divisible by 7$\}$, $\Sigma_2 = \{0, 1, \ldots, 9\}$

$= \{7, 14, 21, \ldots\}$

$L_3 = \{s\#s \in \Sigma_1^*\}$ ab#ab is in L_3, ab#ba is not in L_3

If A = abcde, B = fgh

LENGTH, |A| = 5.

SUBSTRING of B = f, g, h, fg, gh, fgh.

PREFIX: is a substring that must contain the first symbol.

SUFFIX: is a substring that must contain the last symbol.

CONCATENATION, A and B = abcdefgh.

REVERSE: abcd $\rightarrow$ dcba

EQUALITY: abcd = abcd, abcd $\neq$abdc.