## Assignment-2

1)a) If $G = (\{S, A\}, \{a, b\}, P, S)$ where P consists of $\{S \to aAS/aSS/\varepsilon, A \to SbA/ba\}$? Show that the grammar G is ambiguous for the string aabaa.

Ans→ $S \to aAS/aSS/\varepsilon$

$A \to SbA/ba$

Here, $G = (V, \Sigma, R, S)$

$V = \{S, A\}$, $\Sigma = \{a, b\}$, S → starting variable

R → P (the given rules / productions)

Using Left most derivation →

i) $S \Rightarrow aSS$
$\to aaAS$
$\to aabaS$
$\to aababaS$
$\to aababaa\varepsilon$
$\to aabaa$

ii) $S \Rightarrow aSS$
$\to a\varepsilon$
$\to aS$
$\Rightarrow aaAS$
$\to aabaS$
$\to aababaS$
$\to aababaa\varepsilon$
$\to aabaa$

∵ A Grammar is said to be ambiguous if it has more than one leftmost or right most derivation.

∴ G is ambiguous.

b) Let the context-free grammar G be given by the productions $\{S \to AB, A \to a, B \to b, B \to C, E \to c\}$? Find the context-free grammar G1 such that every variable in G1 derives some terminal string.

Ans→ Given, $S \to AB$
$A \to a$
$B \to b$
$B \to C$
$E \to c$

For the context free grammar G1, we need to remove the unit rules / productions as well the null productions.

Name: ——————————

Regd. Number: ——————————

S → AB
A → a
B → b
E → c

c) Construct a grammar in Chomsky normal form (CNF) from the equivalent CFG, G with production rules S → aAbB, A → aAla, B → bBlb.

Ans → Hence, there is no start symbol given in the right side, we do not need to add one extra start state.

S → aAbB, A → aAla, B → bBlb

Now removing the null productions.

As there are no null l∈ productions, we do not need to remove anything in this step.

Now, removing unit productions that are also not available.

Now → S → aAbB
A → aAla
B → bBlb

adding new productions.

S → ~~X~~A ~~Y~~B
A → XA ~~aa~~ | X
B → YB ~~bb~~ | Y
X → a
Y → b

again adding new productions

S → AB
A → XA | X
B → YB | Y
X → a
Y → b

(Q2) a) Explain Chomsky Hierarchy & show that the relationship between the different types of grammars.

Ans → The Chomsky Hierarchy is a classification of formal grammars that describe the syntax of languages. The Chomsky Hierarchy levels are explained further →

Name: ——————————— Regd. Number: ———————————

Type 0: Unrestricted Grammars → They have no restrictions on production rules. They are recognized by turing machines.

Form of rules → $\alpha - \beta$, where $\alpha$ & $\beta$ are the strings of terminals & non-terminals and $\alpha$ must contain at least one non-terminal.

Type 1: Context Sensitive - Grammars (CSG) → These are the grammer where the length of the string on the left-hand side is less than or equal to the length of the string on the right hand side. ($|\alpha| \leq |\beta|$). They are recognised by Linear Bounded Automation (LBA)

Form of rules → $\alpha A \gamma \rightarrow \alpha \beta \gamma$, where A → Non-terminal

$\beta, \alpha, \gamma$ → strings

Type 2: Context - Free Grammar (CFG) → These are the grammar where every rule has a single non-terminal on the left-hand side. They are recognised by the Pushdown Automata (PDA).

Form of Rules → A → B where A → non-terminal
B → string

Type 3: Regular Grammars → These are the grammers where rules generate strings by concatenating terminals & at most a single non-terminal. They are recognised by the Finite Automata.

Form of Rules → A → aB or A → a where
A, B → non-terminal
a → string

The relationship among them are :-

Regular Grammar ⊆ Context Free Grammar

Then, Context Free Grammar ⊆ Context ~~Free~~ Sensitive Grammar

finally, Context Sensitive Grammar ⊆ Unrestricted Grammar.

Name: ——————————— Regd. Number: ———————————

b) Using pumping lemma prove that the language $L = \{a^n | n \text{ is a prime}\}$ is not a Context Free Language.

Ans → For L to be a CFL, it must be divided into 5 parts satisfying the following conditions.

i) for each $i \geq 0$, $uv^ixy^iz \in A$

ii) $|vy| > 0$

iii) $|vxy| \leq p$, thus $v = a^m$ & $y = a^n$, where $m+n > 0$.

Let L be a CFL.

Let $w = a^q$, where q is a prime number & $q \geq p$

For any $i \geq 0$, the pumped string → $w' = uv^ixy^iz = a^{q+(i-1)(m+n)}$

For $i = 2$, $|w'| = uv^2xy^2z = a^{q+(i-1)(m+n)}$

$q+(m+n)$

if $q = 7$ (prime number) & $m+n = 2$

then $|w'| = 7+2 = 9$ (which is not a prime no.)

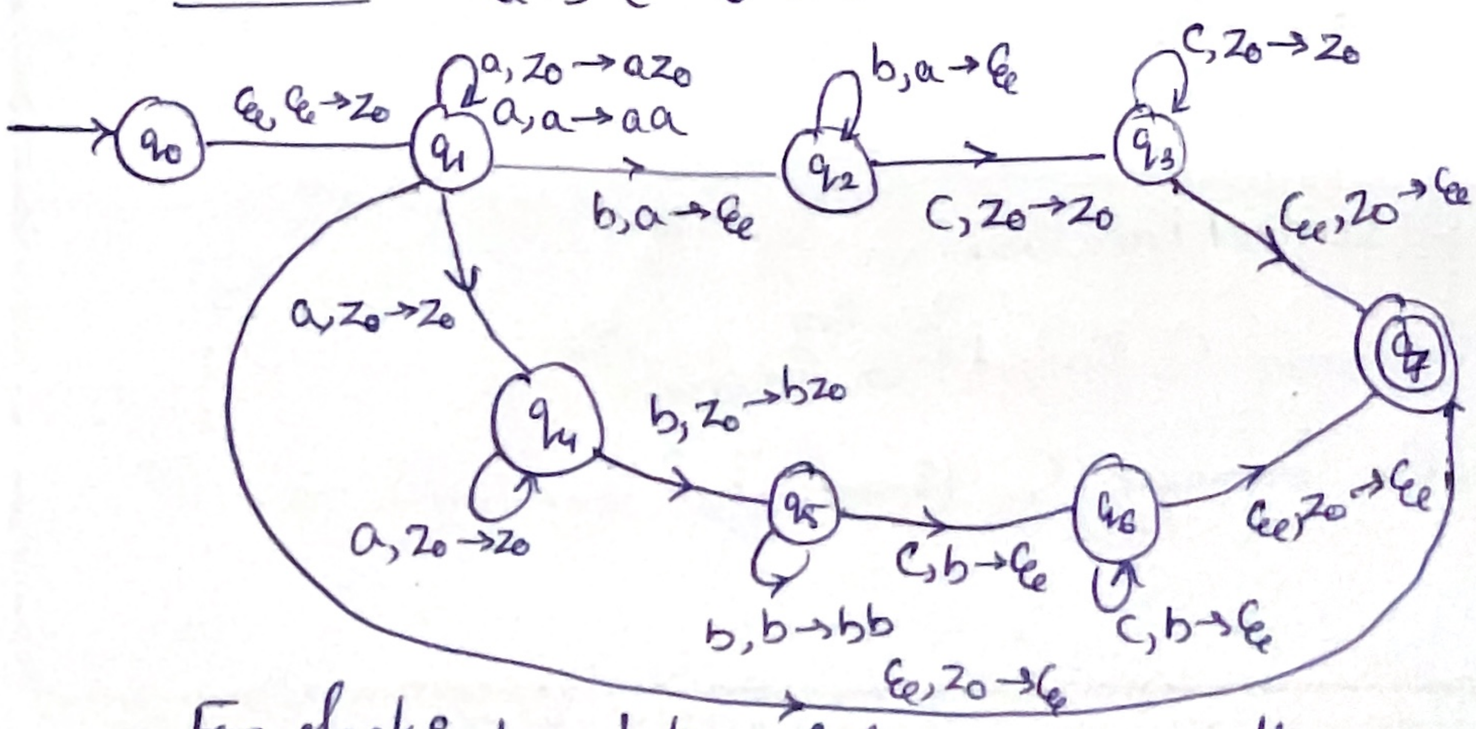
Thus, $w' \in L$ contradicting the pumping lemma.

(Q3) a) Design a PDA that recognizes the language $L = \{a^ib^jc^k | i,j,k \geq 0 \text{ and } i=j \text{ or } i=k\}$. Show the stack implementation for the string aaabbccc.

Ans →

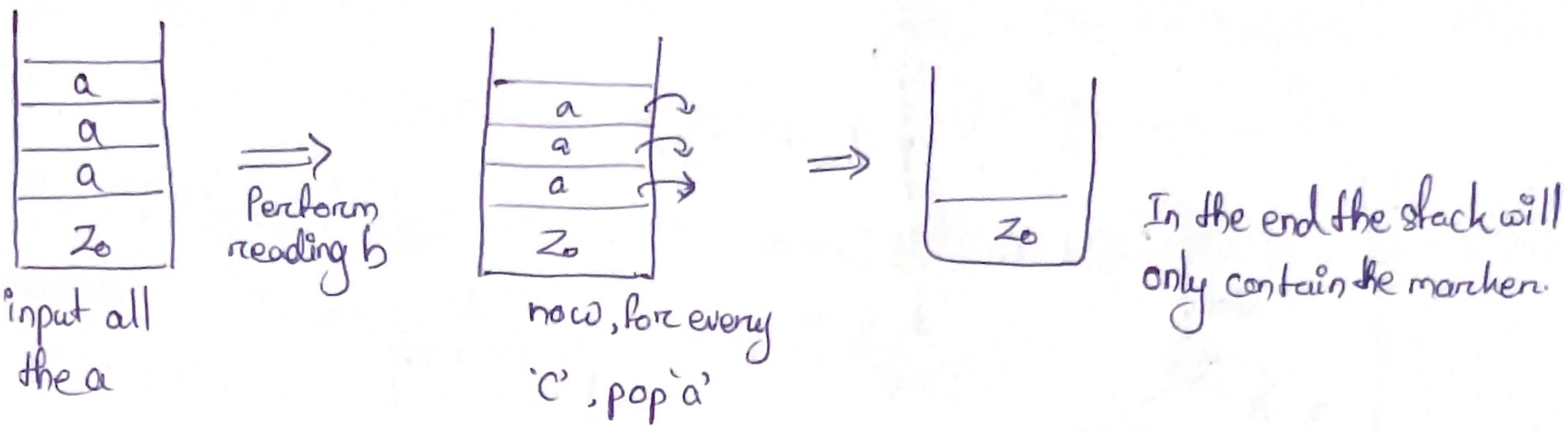
| $a^ib^ic^k$ | $a^ib^ic^k$ |
|---|---|
| here $i=j$ | here $i=k$ |
| let $i=j=m$ | let $i=k=n$ |
| $\Rightarrow a^mb^mc^k, m,k \geq 0$ | $\Rightarrow a^nb^ic^n, n,j \geq 0$ |

Final PDA → $a^mb^mc^k \cup a^nb^ic^n$

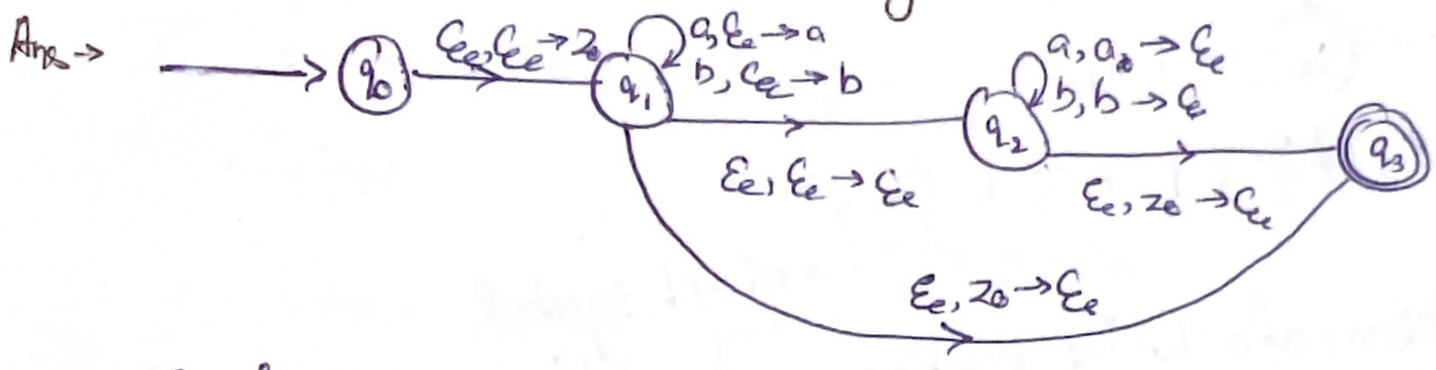



For stack implementation of string aaabbccc. Here no. of a = no. of c, hence push & pop has to be done on a & c.

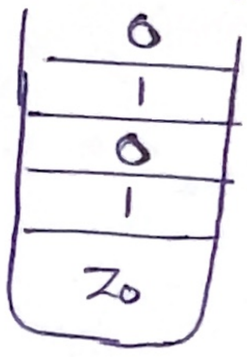
Name: ——————————— Regd. Number: ———————————



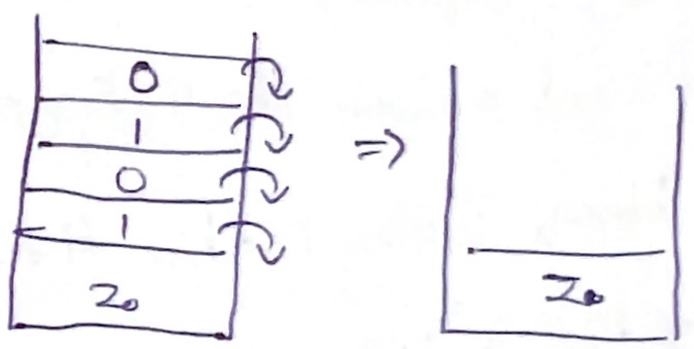

b) Construct the PDA that recognizes the language $L = \{wcw^R \mid w \in \{1,0\}^*\}$. Show the stack implementation for the string 10100101.



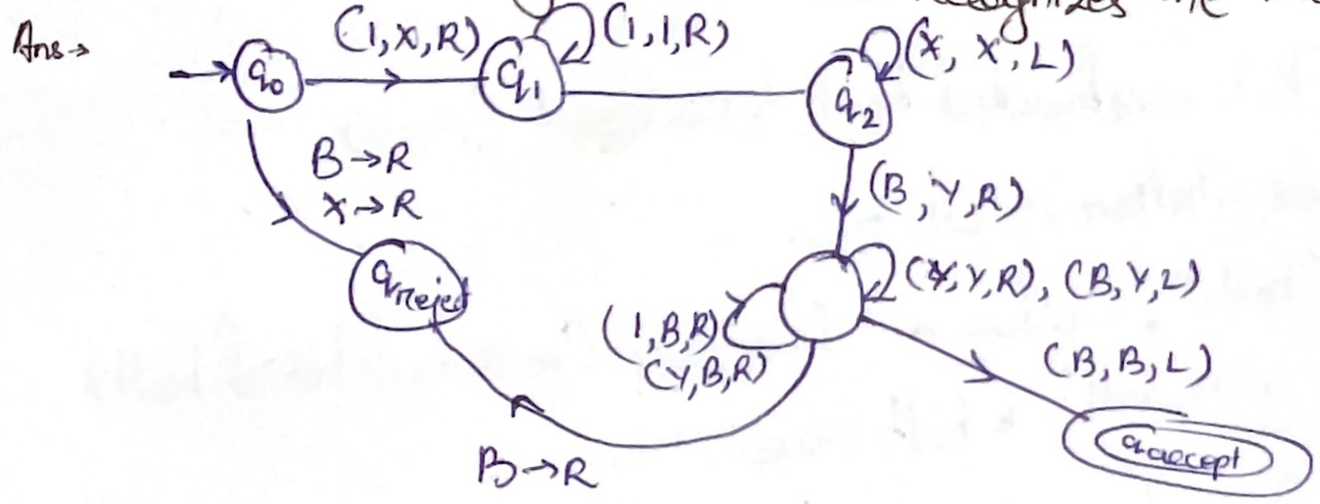

For stack implementation for the string 10100101




for 1→0→1→0 push the symbols ⟹ then pop, the same symbols




(Q4) a) Construct a Turing Machine that recognizes the Language $L = \{0^n 2 0^n \mid n >= 0\}$?



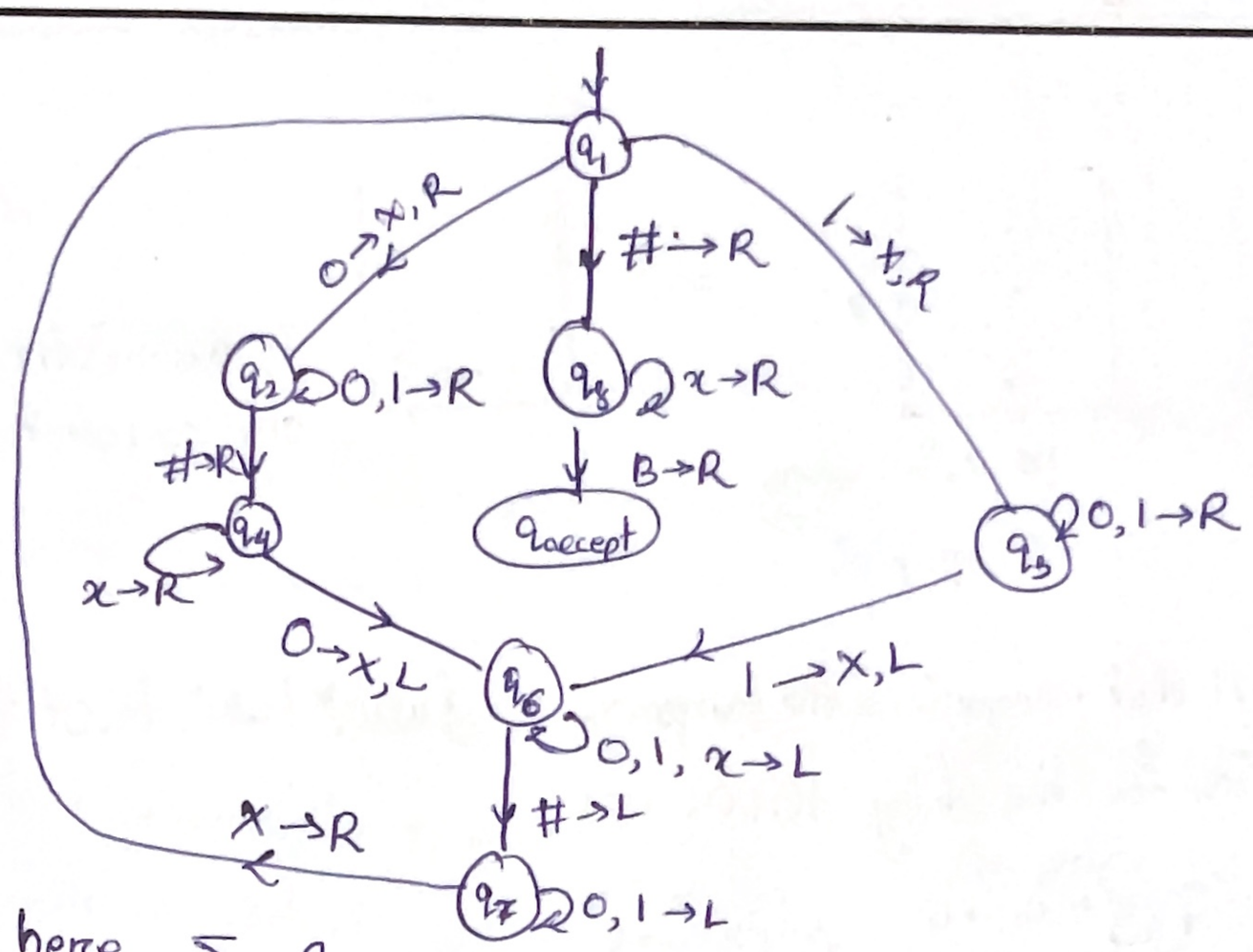

Here, $TM = (Q, \Sigma, \delta, \Gamma, q_0, B, F)$

$Q = \{q_0, q_1, q_2, q_3, q_{accept}\}$

b) Construct a Turing Machine that decides the language $L = \{w\#w \mid w \in \{0,1\}^*\}$.

Name: ——————————— Regd. Number: ———————————

Soln →




here, $\Sigma = \{0, 1, \#\}$ & $\Gamma = \Sigma \cup \{x, B\}$

↳ Blank symbol

Q.5)a) Show that halting problem of a Turing Machine is undecidable.

Ans → We can show this by using proof by contradiction.

For that,

Proof → Let assume the Halting problem is decidable.

Here, Turing Machine H that decides the Halting problem. For any Turing machine M & input w ⇒ H(M, w) → "halt" if M halts on w & "nohalt" if M does no halt on w.

Now, new Turing machine K is constructed that takes input (M, w).

↳ Run H(M, w) to decide whether M halts on w.

↳ If H(M, w) outputs "halts", K enters on infinite loop (ie. does not ~~exit~~ halt)

↳ If H(M, w) outputs "no halt", K halts immediately.

Hence, if M halts on w, K does not halt or if M ~~not~~ does not halt on w, K halts.

The points of contradictions arises likes →

i) If K halts, then H(M, w) must have ~~been~~ output "no halt" for M. But if H(M, w) output "no halt", K would not halt, which again is contradiction.

ii) If K does not halt, then H(M, w) must have output "halt" for M. But if H(M, w) output "halt", K should halt, which again is contradiction.

Therefore, our assumption of halting problem is decidable is false.

∴ The halting problem for turing machine is undecidable.

Name: ———————— Regd. Number: ————————

b) What is P, NP, NP-complete & NP-Hard problem? Explain the relationship of these concepts with the help of a Venn-diagram & give ~~the~~ some examples for each problem.

Soln → P (Polynomial Time) → These problems can be solved in polynomial time by a DTM. If the problem is in P, we can find a solution efficiently.

e.g. → Sorting (Merge Sort, Quick Sort), etc.

NP (Non-Deterministic Polynomial Time) → These are problems where a solution, if given, can be verified in polynomial time by a deterministic Turing machine.

e.g. → Sudoku, Subset-Sum problem.

NP-Complete → These are the hardest problems in NP.

A problem is NP-complete if:

i) It is in NP

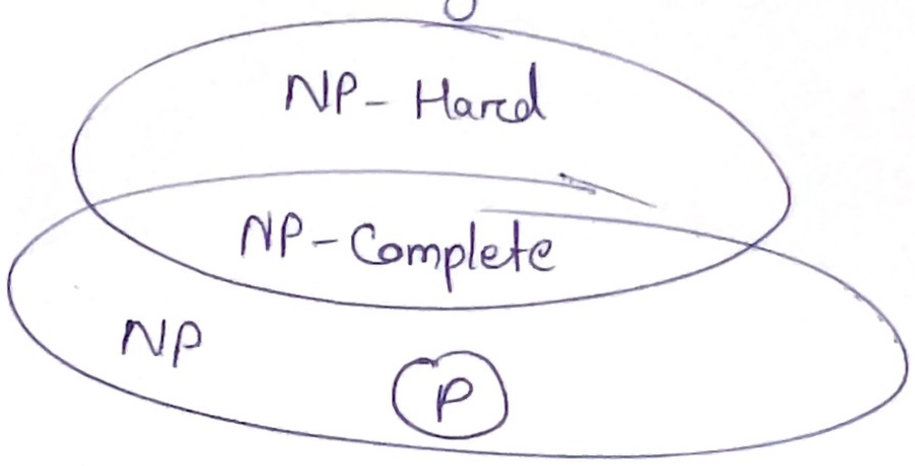
ii) Every problem in NP can be reduced to it in polynomial time.

e.g. → Travelling Salesman problem, 3-SAT problem.

NP-Hard → These are problems at least as hard as NP-complete but are not necessarily in NP.

e.g. → Halting problem, etc.

Relationship in Venn-Diagram.




Name: ——————————

Regd. Number: ——————————