Assignment-3

~~Q1~~

Ans 1 a) $\Sigma = \{a, b\}$

$V = \{S, A\}$

Start symbol: S
Productions (P):

$S \rightarrow aaS \mid aaA$
$A \rightarrow bA \mid \varepsilon$

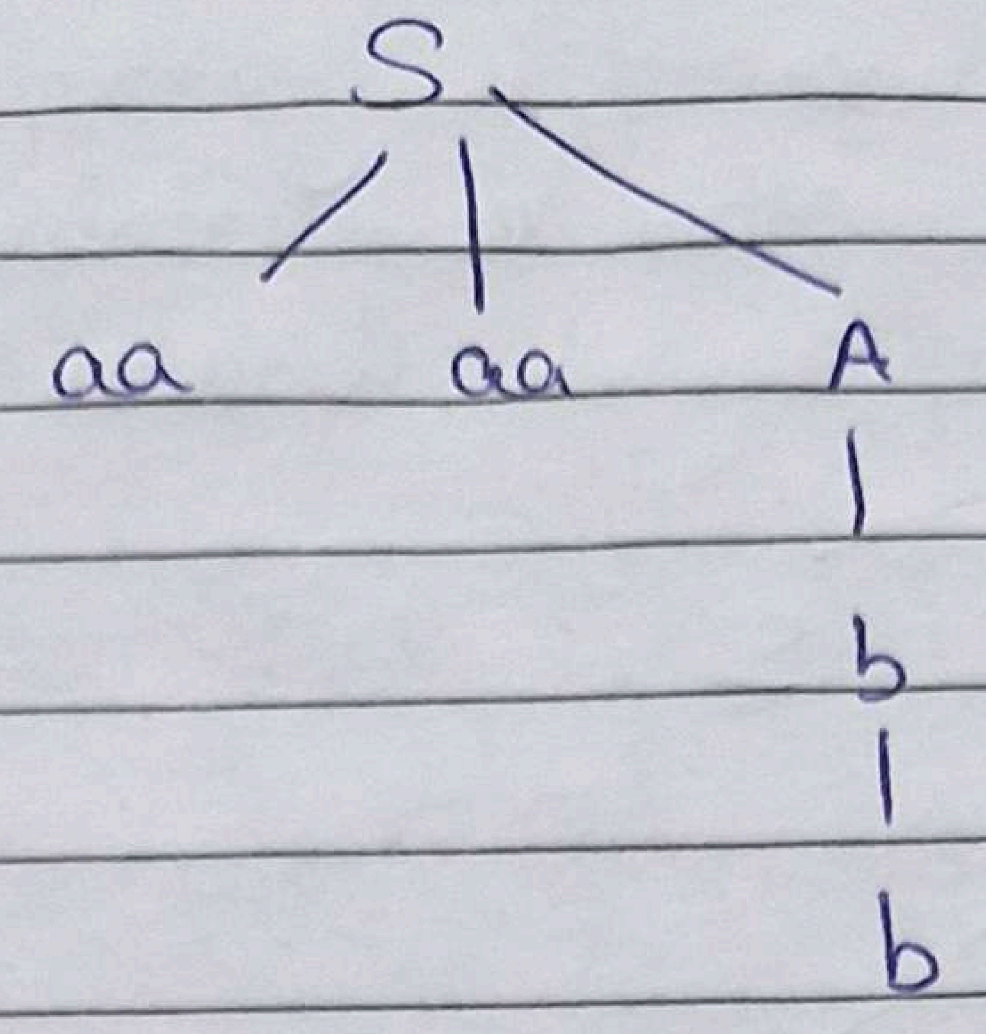
S generates an even number of a's (at least 2) and A generates any number of b's.

(b) String: aaaabb
Leftmost ~~ass~~ Derivation.

$S \rightarrow aaS$

$\rightarrow aa\ aaA$
$\rightarrow aa\ aabA$
$\rightarrow aa\ aab\ bA$
$\rightarrow aa\ aa\ bb$

(c)



Leaves : a a a a bb → "aaaabb"

(d) Regular expression for L:

$$(aa)^{+} b^{*}$$

Since this RE exists, L is a Regular language and every regular language is also context free.

Ao 2

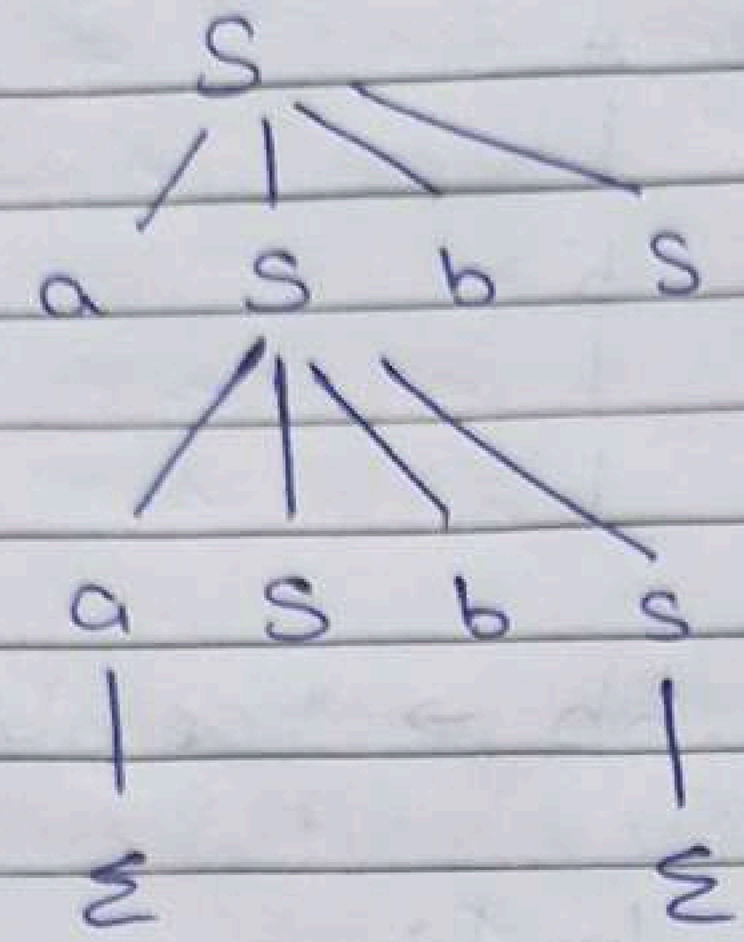
(a) Derive aabb

$S \rightarrow aSbS$

$\rightarrow a(aSbS)bS$

$\rightarrow aa\epsilon b\epsilon b\epsilon = aabb$

Parse tree



leaves (left-to-right): aa bb → aabb

(b) A grammar is ambiguous if there exists at least one string in the language that has two distinct parse trees (equivalently two different leftmost or rightmost derivations).

Claim: The grammar $S \to aSbS \mid \varepsilon$ is unambiguous

Proof: Any nonempty string generated by this grammar must begin with a (because the only production introducing terminals start with a) and therefore must be produced by one application of $S \to aS_1bS_2$.

- In ~~any~~ any derivation of a non empty string N, the first a in N must match some b later in N. let that matching b be at position

k. The grammar forces that matching b to be the b produced in the same production aS_1bS_2 that produced the first a.

- a (the first terminal)
- substring generated by S_1 (between this a and its matching b)
- b (the matching b)
- substring generated by S_2 (the remainder)

- The position k (first a's matching b) is uniquely determined by the usual balance argument (count a minus b scanning from left) – it is the smallest index where balance returns to the level before the first a. Hence the split into parts for S_1 and S_2 is unique.

- By induction on string length, the derivations (and therefore parse tree) of each of the two substrings are unique.

- Therefore the whole parse tree is unique

So no string has two different parse trees is unambiguous.

The given grammar itself is already non-ambiguous. You can present an equivalent (and maybe clearer) unambiguous grammar

$S \to TS \mid \epsilon$
$T \to aSb$

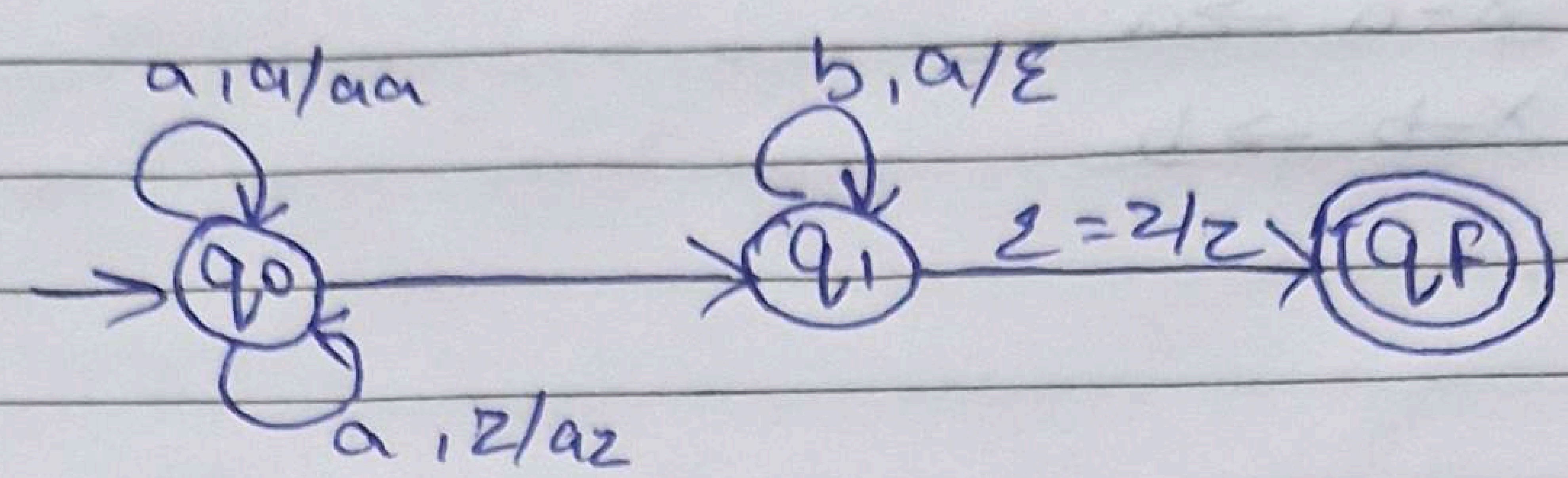
T generates one matched pair a.....b where the .... is a balanced substring (generated by S) and then $S \to TS$ allows concatenation of such matched blocks. This grammar is the same as the original $S \to aSbS \mid \epsilon$ but written to emphasize the unique decomposition $S = (aSb)S$ - hence unambiguous.

Q3 let $M = (Q, \Sigma, T, \delta, q_0, z_0, F)$ where:

- $Q = \{q_0, q_1, q_f\}$
- $\Sigma = \{a, b\}$
- $T = \{z_0, X\}$ - z_0 is initial stack symbol
  X is marker for each a
- q_0 is start state.
- z_0 is initial stack symbol.
- $F = \{q_f\}$ (accept by final state)

(b) • $\delta(q0, a, z0) = \{(q0, xz0)\}$
• $\delta(q0, \epsilon, z0) = \{(qf, z0)\}$
• $\delta(q0, a, x) = \{(q0, xx)\}$
• $\delta(q1, \varepsilon, z0) = \{(qf, z0)\}$
• $\delta(q0, b, x) = \{(q1, \varepsilon)\}$

d)



Ans 4

Step 1 - find nullable non-terminals:

A is nullable (since $A \to \varepsilon$)
B is nullable (since $B \to \varepsilon$). S is not nullable

Step 2 - add productions obtained by omitting nullable occurences:

from $S \to aA$: because A nullable, add $S \to a$
from $S \to bB$: because B nullable, add $S \to b$
from $A \to aA$: because A nullable, add $A \to a$
from $B \to bB$: because B nullable, add $B \to b$.

Step 3 → remove original ε-productions: remove $A \to \varepsilon$ and $B \to \varepsilon$

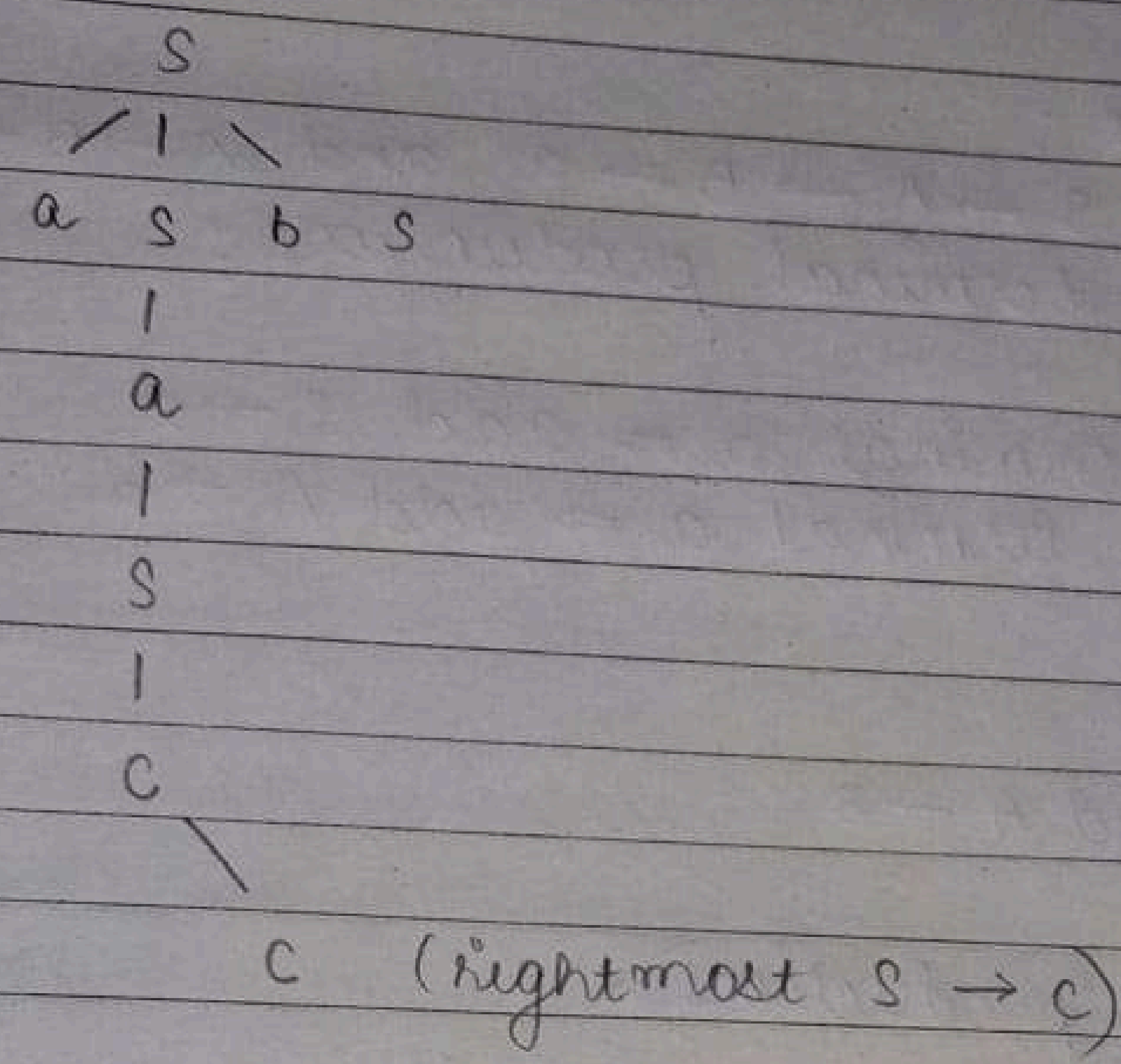
No unit productions (of the form $X \to Y$) are present now.

b) CNF require productions of the form $X \to YZ$ (two - nonterminals) or $X \to a$ (single terminal).

Also introduce helper nonterminals for terminals when they appear in longer right - hand sides.

$X_a \to a$

$X_b \to b$



⇒ The two trees are structurally different (in A the left child of the root is 'a' then a subtree aSbS; in B the root expands to aSbS immediately). Therefore the grammar is ambiguous (exists a string with two distinct parse trees).

⇒ A grammar is ambiguous iff some string has two different parse trees. We exhibited two distinct parse trees for aacbc. Hence the grammar is ambiguous.

Ques8 Give the CFG $G = (\{S, A, B\}, \{a\}, \{S \to A, A \to B, B \to a\}, S)$. Remove unit productions and rewrite the grammar.

Given CFG:

$G = (\{S, A, B\}, \{a\}, P, S)$ with

$P: S \to A, \; A \to B, \; B \to a$

⇒ Remove unit productions :

We have unit chains $S \to A \to B \to a$ and $A \to B \to a$.

Replace them by direct terminal productions :

- From S follow chain to terminal a ⇒ add $S \to a$.
- From A follow chain to terminal a ⇒ add $A \to a$.
- Keep $B \to a$.

Remove unit rules $S \to A$ and $A \to B$

Resulting grammar (no unit productions) :

$S \to a$

$A \to a$

$B \to a$

Note : all non-terminals still generate a. The language is $\{a\}$.

Ques^a Give the CFG $G = (\{S, A, B\}, \{a, b, c\}, \{S \to A, A \to aB, B \to c\}, S)$. Remove useless productions and write the updated grammar.

Given :

$G = (\{S, A, B\}, \{a, b, c\}, P, S)$ with

$P : S \to A,\ A \to aB,\ B \to c$.

Step 1 — Find non terminals that generate terminals (useful for generating) :

Date.....................

- $B \to c \Rightarrow B$ generates.
- $A \to aB$ and B generates $\Rightarrow A$ generates.
- $S \to A$ and A generates $\Rightarrow S$ generates.

So all S, A, B are generating

Step 2 – Find reachable non terminals from start S:

- S is start (reachable).
- From $S \to A \Rightarrow A$ reachable
- From $A \to aB \Rightarrow B$ reachable

So, all S, A, B are reachable.

$\Rightarrow$ There are no useless productions (no unreachable or non-generating non-terminals).

* terminal grammar (after removing useless productions):

$S \to A$
$A \to aB$
$B \to c$

Language produced: strings of the form ac (specifically only "ac").

Ques10 Convert the given CFG to CNF. Consider the given grammar G1:
$S \to a \mid aA \mid B$

Date....................

$A \rightarrow aBB \mid \varepsilon$
$B \rightarrow Aa \mid b$

Step 1 – Remove ε- production:
$A \rightarrow \varepsilon$ is nullable ⇒ update others.

$S \rightarrow a \mid aA \mid B$
$A \rightarrow aBB$
$B \rightarrow Aa \mid a \mid b$

Step 2 – Remove unit productions:
$S \rightarrow B$ ⇒ replace with B's RHS.

$S \rightarrow a \mid b \mid aA \mid Aa$
$A \rightarrow aBB$
$B \rightarrow Aa \mid a \mid b$

Step 3 – Convert to CNF form:
Introduce $X\_a \rightarrow a$, $Y \rightarrow BB$.
Replace terminals in long RHS and make binary:

$S \rightarrow a \mid b \mid X\_a A \mid A X\_a$
$A \rightarrow X\_a Y$
$Y \rightarrow B B$
$B \rightarrow A X\_a \mid a \mid b$
$X\_a \rightarrow a$

Final CNF: All rules are either $A \rightarrow BC$ or $A \rightarrow a$.