Jay Patel
HW8
CS 331
Professor Oliver

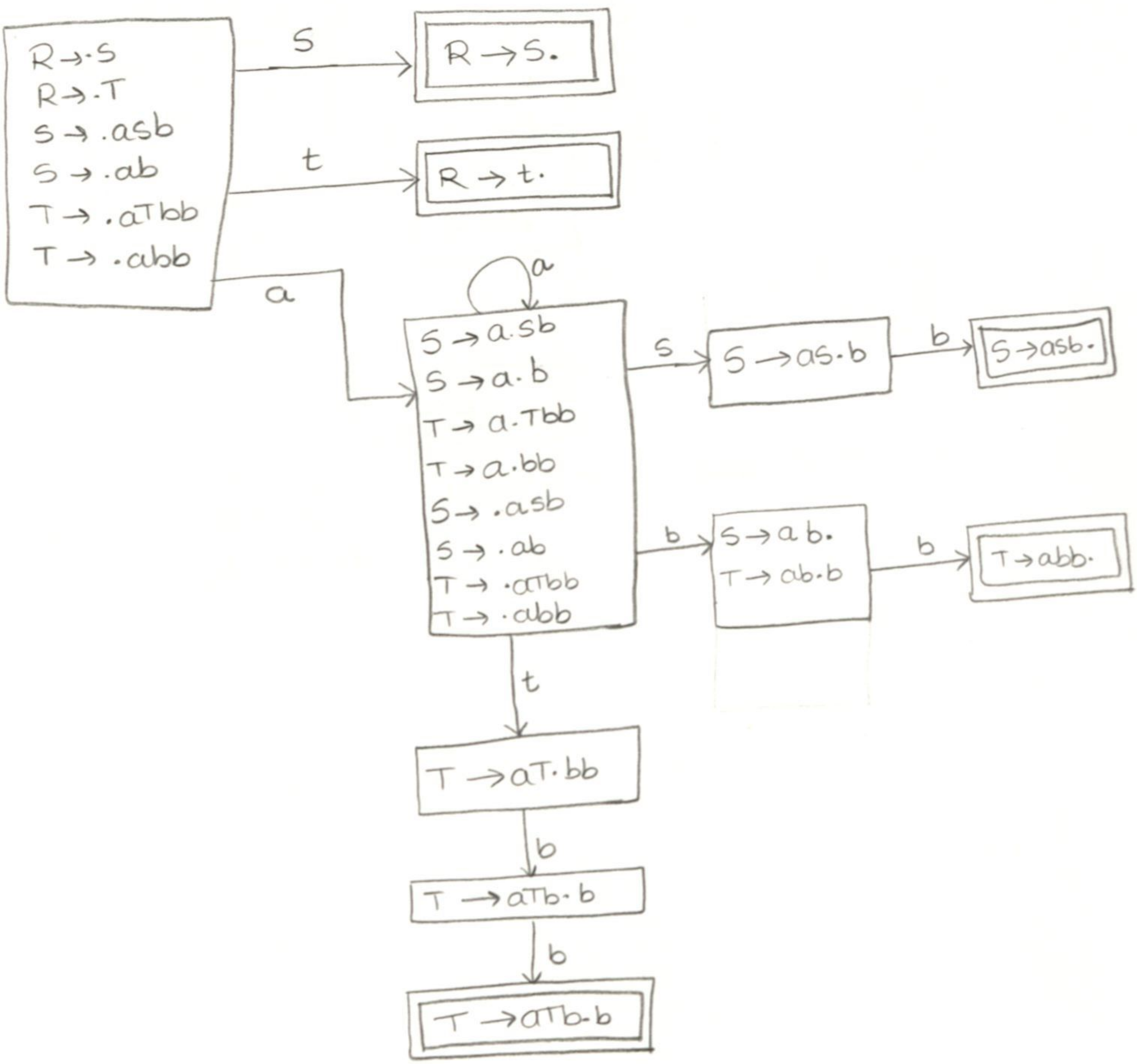
Q1.

A CFG is nothing but is unambiguous. If every string has at most one valid derivation according to the CFG otherwise the grammar is ambiguous. If you have a CFG and can produce two different derivation trees for some string have an ambiguous grammar. A CFL is inherently ambiguous if and only if it is not the language of any unambiguous CFG. For **Ambiguous grammar**, they are defined as being able to have multiple parse trees, which means that it will have at least one string. In order to define ambiguity, it is basically a CFG for which there exists a string that can have more than one **left most** derivation whereas on the other hand unambiguous grammar is a CGG for every valid string has a unique left more derivation. Which will then result into some left most reduction for some grammar example;
S → T -|
T → T (T) | ∈
Which will then result it **diverging into two different rules**, which results in that the grammar has at least **one string** where there are more than one handles. As we know that **right most derivation is equivalence to the left most reductant**. Which will then result in grammar as not deterministic if the grammar is ambiguous since the string cannot have multiple derivations.

Jay Patel

Q2]
a

$R \to S \mid T$
$S \to aSb \mid ab$
$T \to aTbb \mid abb$



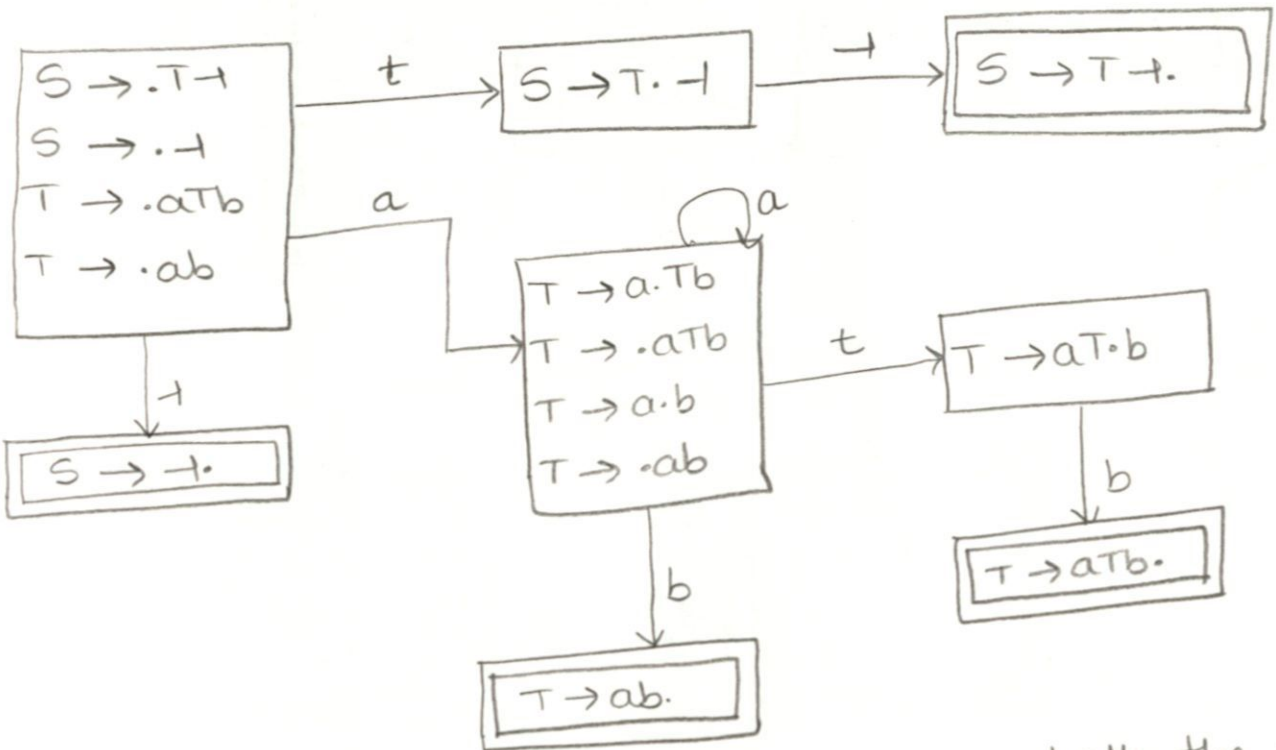
Non deterministic, because] fails the second rule of DK test also in other word "dot" before the terminal

Q2]

b]

$S \rightarrow T \dashv | \dashv$

$T \rightarrow aTb | ab$

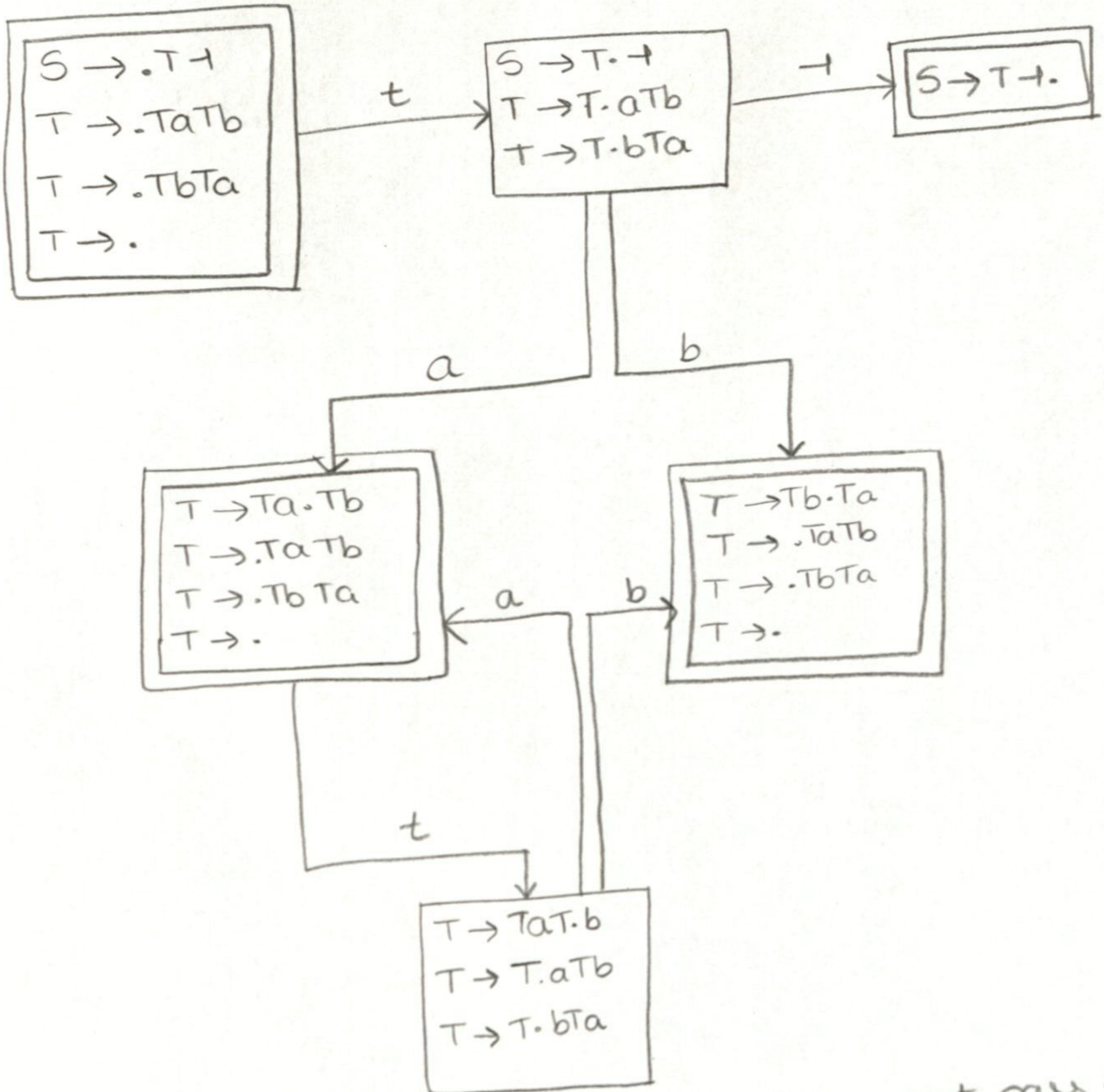


It is deterministic since it passes both the rules of DK-test

Q2]

c] $S \rightarrow T \dashv$

$T \rightarrow TaTb \mid TbTa \mid \varepsilon$



Non deterministic because it does not pass the DK test because there's a dot before the terminal