PRINT NAME: _____

$\mathcal{SIGN}$: _____

1. (5 **pts**) Give a regular grammar for the language of all strings in $\{a, b\}^*$ such that aba is not a substring.

Of course, your grammar should generate every string in this language, and no strings not in the language.

♣ The request is for a regular grammar, so I will plan the design according to the prefix of the sentential form.

S - the start symbol.

A - the prefix is in the language and ends in a.

B - the prefix is in the language and ends in ab.

C - the prefix is in the language and ends in b, but not ab.

No other variables should be needed, and all that remains is to write rules expressing how the prefix expands:

$$
\begin{aligned}
G : S &\rightarrow aA \mid bB \mid a \mid b \mid \lambda \\
A &\rightarrow aA \mid bB \mid a \mid b \\
B &\rightarrow bC \mid b \\
C &\rightarrow aA \mid bC \mid a \mid b
\end{aligned}
$$

Incidently, the grammar is unambiguous and has no useless symbols.   ♣
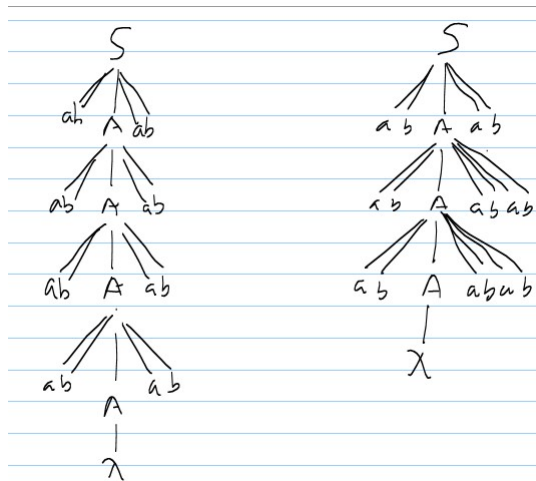
2. (5 **pts**) a) Show that grammar below is ambiguous by giving a word in the language with two different derivation trees.

$$G : S \;\rightarrow\; abAab \mid \lambda$$
$$A \;\rightarrow\; abAab \mid abAabab \mid \lambda$$

b) Give an unambiguous grammar which yields the same language.

♣ It is best to first decide about the language, $L(G) = \{\lambda, (ab)^2\} \cup \{(ab)^k \mid k \geq 4\}$, so all powers of ab except the first and third power.

The smallest word with an ambiguous derivation is $(ab)^8$:

For b), as we discussed in class, there is no easy algorithm for detecting or removing ambiguity, so it is often a matter of just writing a new grammar from scratch, like this one:

$$G : S \;\rightarrow\; \lambda \mid abab \mid abababab \mid ababababB$$
$$B \;\rightarrow\; ab \mid abB \quad ♣$$