1. Given the grammar

\[ G : S \to SBA | A \\
    A \to aA | \lambda \\
    B \to Bba | \lambda, \]

convert to an equivalent non-contracting grammar with no recursive start.

*We compute Null*(\(G\)) = \{A, B, S\}.

*So the conversion is:*

\[ G_L : S \to SBA | SB | SA | BA | S | B | A \\
    A \to aA | a \\
    B \to Bba | ba, \]

*convert to an equivalent non-contracting grammar with no recursive start. in which the rule S \(\to\) S can be removed.*

*Removing the non-recursive start we define a new start symbol S':*

\[ G_L : S' \to S \\
    S \to SBA | SB | SA | BA | S | B | A \\
    A \to aA | a \\
    B \to Bba | ba, \]

*convert to an equivalent non-contracting grammar with no recursive start.*

2. Given the grammar

\[ G : S \to AS | A \\
    A \to a^2A | bB | C \\
    B \to b^2B | b \\
    C \to c^2C | B \]

Compute \(G_c\), the equivalent grammar with no chain rules.

*Chain*(\(S\)) = \{S, A, B, C\}

*Chain*(\(A\)) = \{A, B, C\}

*Chain*(\(B\)) = \{B\}

*Chain*(\(C\)) = \{B, C\}

\[ G : S \to AS | a^2A | bB | b^2B | b | c^2C \\
    A \to a^2A | bB | b^2B | b | c^2C \\
    B \to b^2B | b \\
    C \to c^2C | b^2B | b \]