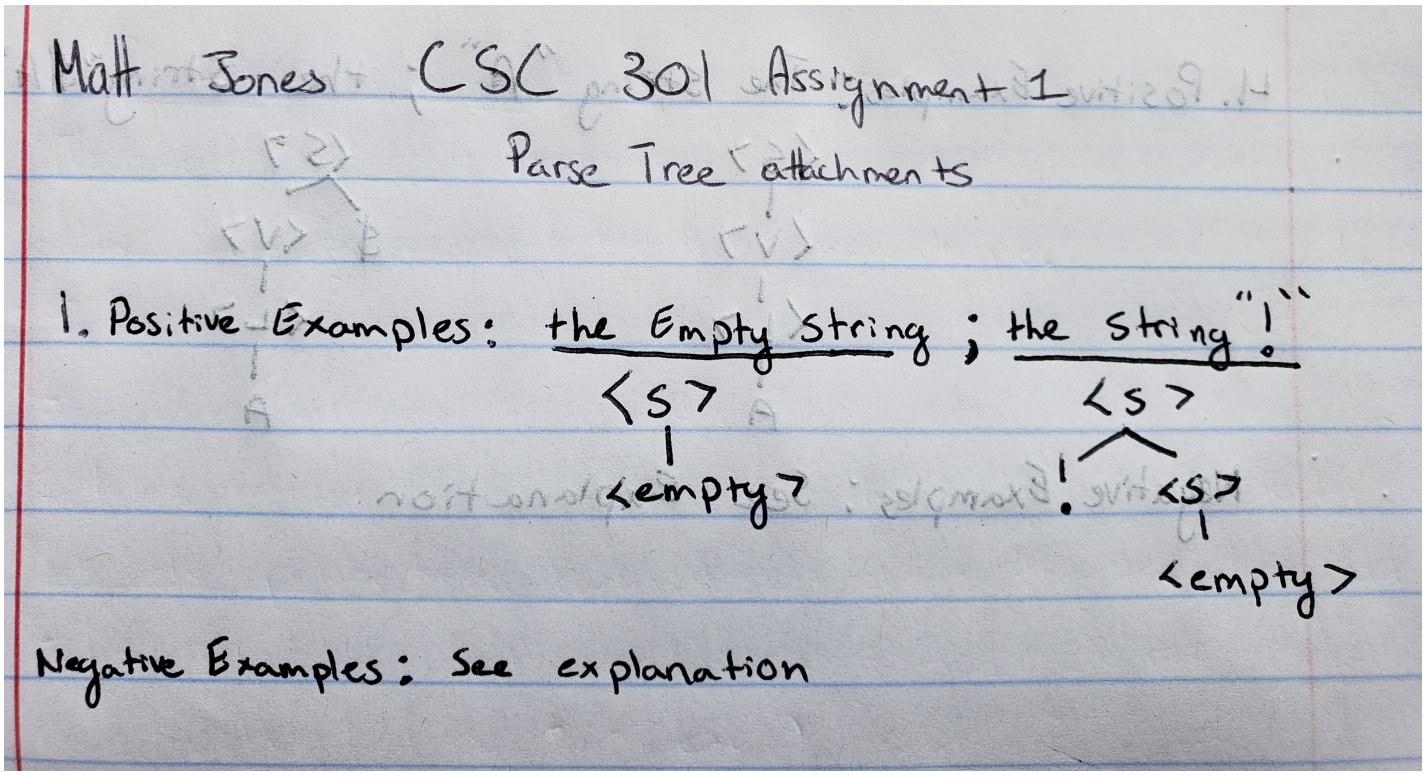
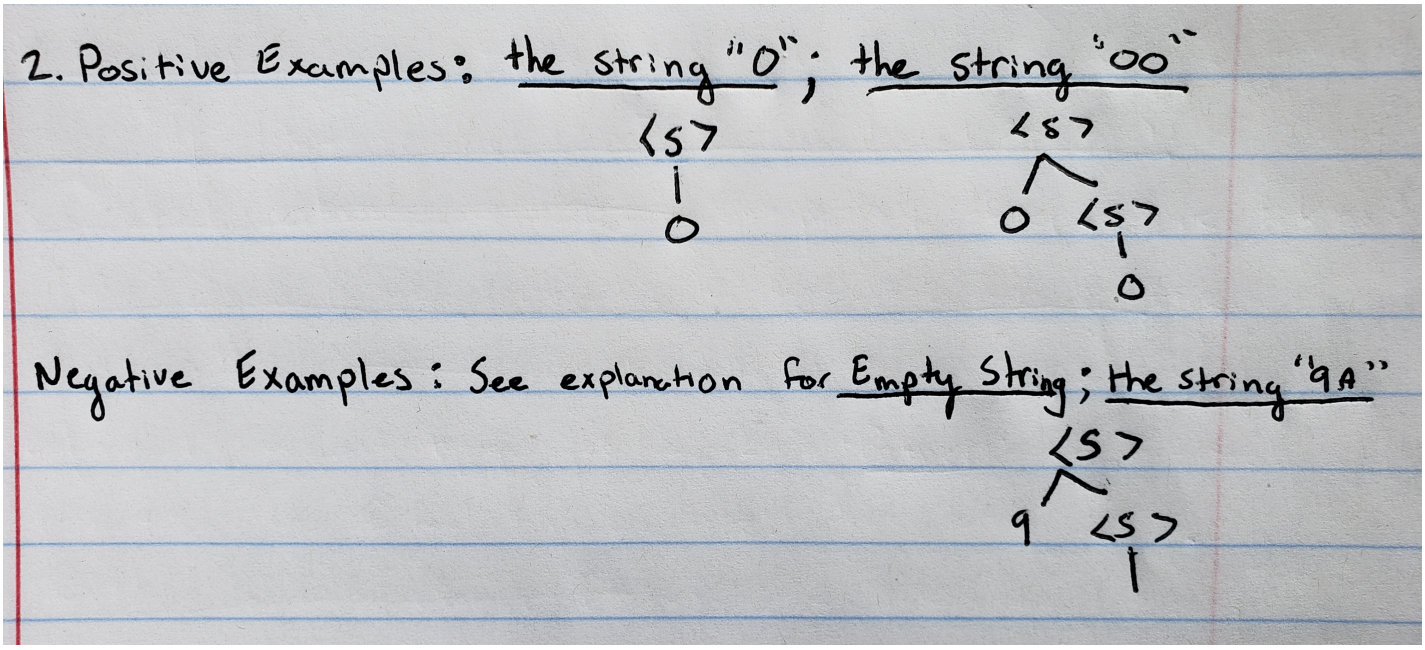
Matt Jones

**CSC 301 Assignment 1**

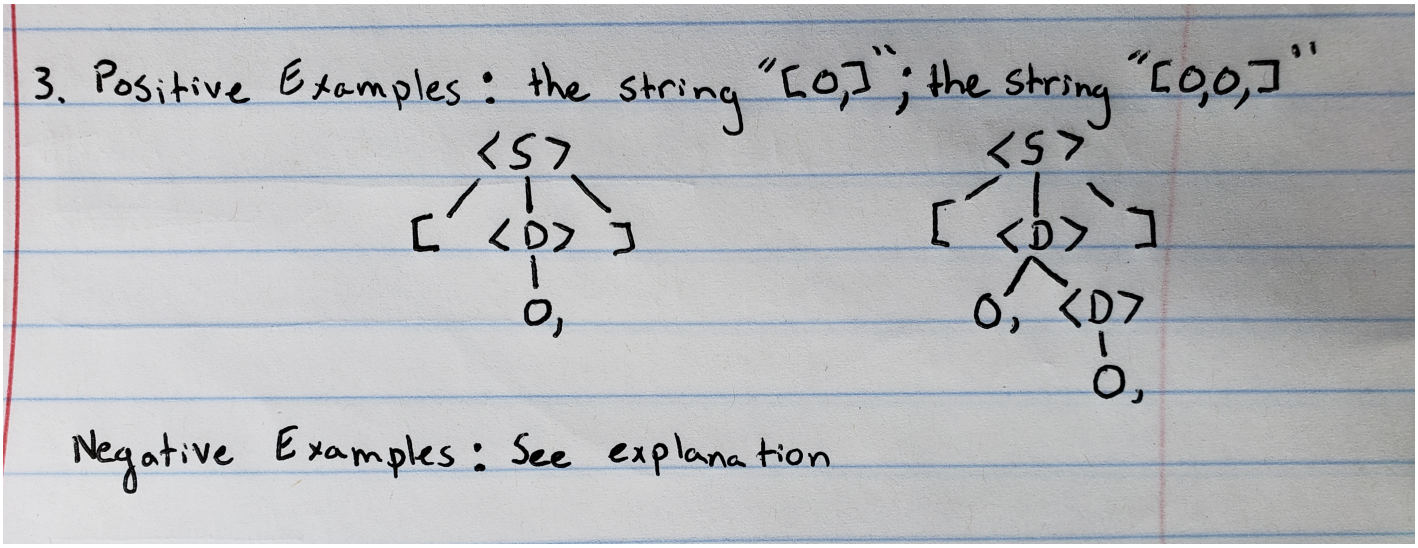
1. The set of all strings containing zero or more exclamation points.
    - <S>* :: = ! <S> | <empty>
    - Positive Examples: The Language can generate the Empty String, and the string “!”.
    - Negative Examples: The language can’t generate the string “A”, or the string “A!”.
      Explanation: The language can only generate “!” and the Empty String because those are the only terminal nodes defined in <S>*. A is not a terminal node, so any string containing “A” can’t be generated.




2. The set of all strings containing one or more decimal digits.
    - <S>* :: =0| 0 <S> |1|1 <S> | ... |9 |9 <S>
    - Positive Examples: The Language can generate the string “0”, and the string “00”.
    - Negative Examples: The Language can’t generate the Empty String, or the string “9A”.
      Explanation: The language can’t generate the Empty String or “9A”, because both the Empty String and “A” aren’t defined as a terminal nodes. Every terminal node is defined as having at least one decimal number, or more through recursion (ie. 0 <S>).




3. The set of all strings consisting of an open bracket followed by a comma separated list of decimal integers, followed by a closing bracket.
   - <S>* :: = [ <D> ]
     < D > :: = 0,| 0 <D>, |1,| 1 <D>, | ... |9,|9 <D>,
   - Positive Examples: The Language can generate the string “[0,]’, and the string “[0,0,]”.
   - Negative Examples: The Language can’t generate the Empty String, or string “0”. Explanation: See above explanation for the Empty String. Additionally, the language can’t generate the Empty String or “0” because the start symbol is defined as having two terminals “[“ & “]” and the non-terminal <D>, which represents decimal integers followed by a comma. Neither the Empty String or “0” have those two mandatory terminals “[“ &”]” or the comma.



4. The set of all legal variable names in Commodore Basic. Variable names can just be one or two characters. The first character must be an uppercase letter. This language only has string and float variables. The “$” in front of character(s) distinguishes these types of variables.
    - <S>* :: = $<V> | <V>
      <V> :: = <L>|<L> <L>|<L> <D>
      <L> :: = A | B| … | Z|
      < D > :: = 0 | 1| ... |9
    - Positive Examples: The language can generate the string “A”, and the string “$A”.
    - Negative Examples:The language can’t generate the string “A$”, or the String “1A”.
      Explanation: The start symbol is strictly defined as $<V> or <V>, so the string “A$” can never be generated. Further <V> is defined as <L>, <L> <L>, or <L> <D> (Where <L> represents all uppercase letters, and <D> represents numbers 0-9). So an uppercase letter must come first, therefore the string “1A” can’t be generated.