GRAMMARS

A grammar is defined (formally) as a 4-tuple \( G = (N, \Sigma, P, S) \) where

- \( N \) is the set of nonterminal symbols.
- \( \Sigma \) is the set of terminal symbols.
- \( P \) is the set of production rules, each of the form \( \alpha \to \beta \).
- \( S \) is the start symbol.

The Chomsky hierarchy defines four distinct types of grammars based on restrictions to the types of production rules allowed in the grammar. These types of grammars are:

1. **Regular grammars.** Productions must be of the form \( A \to xB \), where \( A \) and \( B \) are nonterminal symbols and \( x \) is a string of terminal symbols. For example, the grammar below defining identifiers in a programming language is regular.

   \[
   G = \{ \{ I, X \}, \{ a, b, c, \ldots, z, 0, \ldots, 9 \}, P, I \} \\
   I \to aX | bX | cX | \ldots | zX | a | b | \ldots | z \\
   X \to aX | bX | cX | \ldots | zX | 0X | 1X | \ldots | 9X | a | b | \ldots | z | 0 | 1 | \ldots | 9
   \]

   A sample derivation is

   \( I \Rightarrow cX \Rightarrow csX \Rightarrow cs4X \Rightarrow cs40X \Rightarrow cs405 \)

2. **Context-free grammars.** Productions must be of the form \( A \to \alpha \), where \( A \) is a nonterminal symbol and \( \alpha \) is a string of terminals and nonterminals in any order. (This is essentially the same as a BNF grammar.) The grammar below is an example of a context-free grammar.

   \[
   G = \{ \{ E, T, F \}, \{ a, +, *, (, ) \}, P, E \} \\
   E \to E + T | T \\
   T \to T * F | F \\
   F \to (E) | a
   \]

3. **Context-sensitive grammars.** Productions must be of the form \( \alpha \to \beta \), where \( \alpha \) and \( \beta \) are strings on terminals and nonterminals with the restriction that the number of symbols in \( \alpha \) must not exceed the number of symbols in \( \beta \). An example of a context-sensitive grammar is:

   \[
   G = \{ \{ S, A, B, C \}, \{ a, b, c \}, P, S \} \\
   S \to aSBC | abC \\
   bB \to bb \\
   CB \to BC \\
   bC \to bc \\
   cC \to cc
   \]

   This grammar generates strings of the form \( a^n b^n c^n \) where \( n > 0 \). For example,

   \( S \Rightarrow aSBC \Rightarrow aabC \Rightarrow aabBC \Rightarrow aabbCC \Rightarrow aabbcC \Rightarrow aabbcC \Rightarrow aabbcc \)

4. **Unrestricted grammars.** Production rules are of the form \( \alpha \to \beta \) with no restrictions. For example,

   \[
   G = \{ \{ S, A, B, C, D \}, \{ a, b \}, P, S \} \\
   S \to CD \\
   C \to aCA \\
   bB \to bB \\
   AD \to aD \\
   BD \to bD \\
   \]

   This grammar generates symmetrical strings containing \( a \)'s and \( b \)'s (e.g. aabaab, baabaa, abbbabbb, etc.). For example,

   \( S \Rightarrow CD \Rightarrow aCAD \Rightarrow abCBAD \Rightarrow abBAD \Rightarrow abBaD \Rightarrow abaBD \Rightarrow ababD \Rightarrow abab \)