1. Formal grammar

Def a formal way of description language, there are generative and analytical types.

Terminal symbol an object presented in words of language. Has semantic value for language user.

Non-Terminal symbol an meta-object of language(formula, command, …). Has no concrete meaning.

2. Generative grammar

Grammar describes how to form strings from alphabet of a formal language that are valid accordingly to language syntax. Does not describe the meaning of the strings - only their form.

alphabet a set of atomic symbols, which build words.

Grammar consists of:

1. Terminal alphabet $\sum$
2. Non-Terminal alphabet N
3. Inference rules P : left $\rightarrow$ right

left: nonempty sequence of terminal and non-terminals, has at least 1 non-terminal.

right: sequence of terminal and non-terminals

4. Define first non-terminal

2.1. Chomsky hierarchy

Def hierarchy in which lower is more strict, but more suitable for parsing.

0. Not limited
1. Context-sensitive
2. Context-free. (only non-terminals at left P)
3. Regular(right has one non-terminal and maybe terminals)

3. Chomsky Normal Form

For all P must: $A \rightarrow BC$ $A \rightarrow \alpha$

Must exist rule if language has epsilon: $S \rightarrow \varepsilon$

Where: $A, B, C \in \text{NonTerm}$, $\alpha \in \text{Term}$, S is start symbol, ε is empty string

4. Bottom-up parsing

5. offtop Dynamic programming

6. CYK

Def parsing algorithm for context-free grammars. It employs bottom-up parsing and dynamic programming.

Operates on Chomsky Normal Form. Noted because of good worst-case performance.

https://www.borealisai.com/research-blogs/tutorial-15-parsing-i-context-free-grammars-and-cyk-algorithm/