AI-12

## Implementation Aspects of Syntactic Analysis

There are a number of algorithms researchers have developed for syntactic analysis, but we consider only the following simple methods −

* Context-Free Grammar
* Top-Down Parser

Let us see them in detail −

### Context-Free Grammar

It is the grammar that consists rules with a single symbol on the left-hand side of the rewrite rules. Let us create grammar to parse a sentence −

“The bird pecks the grains”

Articles (DET) − a | an | the

Nouns − bird | birds | grain | grains

Noun Phrase (NP) − Article + Noun | Article + Adjective + Noun

= DET N | DET ADJ N

Verbs − pecks | pecking | pecked

Verb Phrase (VP) − NP V | V NP

Adjectives (ADJ) − beautiful | small | chirping

The parse tree breaks down the sentence into structured parts so that the computer can easily understand and process it. In order for the parsing algorithm to construct this parse tree, a set of rewrite rules, which describe what tree structures are legal, need to be constructed.

These rules say that a certain symbol may be expanded in the tree by a sequence of other symbols. According to first order logic rule, if there are two strings Noun Phrase (NP) and Verb Phrase (VP), then the string combined by NP followed by VP is a sentence. The rewrite rules for the sentence are as follows −

S → NP VP

NP → DET N | DET ADJ N

VP → V NP

Lexocon −

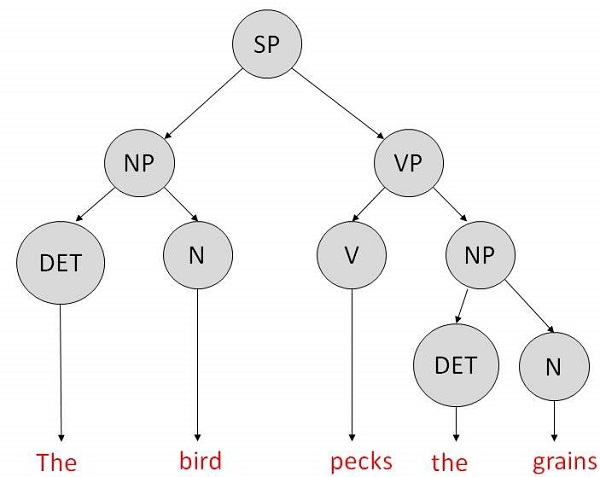
DET → a | the

ADJ → beautiful | perching

N → bird | birds | grain | grains

V → peck | pecks | pecking

The parse tree can be created as shown −



Now consider the above rewrite rules. Since V can be replaced by both, "peck" or "pecks", sentences such as "The bird peck the grains" can be wrongly permitted. i. e. the subject-verb agreement error is approved as correct.

Merit − The simplest style of grammar, therefore widely used one.

Demerits −

* They are not highly precise. For example, “The grains peck the bird”, is a syntactically correct according to parser, but even if it makes no sense, parser takes it as a correct sentence.
* To bring out high precision, multiple sets of grammar need to be prepared. It may require a completely different sets of rules for parsing singular and plural variations, passive sentences, etc., which can lead to creation of huge set of rules that are unmanageable.

### Top-Down Parser

Here, the parser starts with the S symbol and attempts to rewrite it into a sequence of *terminal symbols* that matches the classes of the words in the input sentence until it consists entirely of terminal symbols.

These are then checked with the input sentence to see if it matched. If not, the process is started over again with a different set of rules. This is repeated until a specific rule is found which describes the structure of the sentence.

Merit − It is simple to implement.

Demerits −

* It is inefficient, as the search process has to be repeated if an error occurs.
* Slow speed of working.