

Categorial Grammar – Introduction

Phrase-structure grammars may be thought of as formalisations of the notion that sentences can be analysed in terms of hierarchies of progressively more inclusive phrases or syntactic categories. The analysis of a sentence is effected by the application of phrase-structure rules, often called rewriting rules. The analysis process begins with the application of rules such as the following:

- (1) (a) $A \rightarrow \text{nice}$
 (b) $N \rightarrow \text{dogs}$
 (c) $V \rightarrow \text{like}$

that serve to identify the parts of speech or lexical categories of individual words. Then other rules such as the following are applied:

- (2) (a) $NP \rightarrow A N$
 (b) $VP \rightarrow V NP$
 (c) $S \rightarrow NP VP$

Rules of this kind govern how words can be combined into phrases, and ultimately into a sentence, on the basis of the lexical categories of the words.

A categorial grammar, in contrast, does not include a separate collection of word-combining rules. Rather, the lexical categories of words such as verbs and adjectives constitute functions that determine how these words can combine with other categories. For example, a lexical item such as ‘nice,’ an adjective, is in a category corresponding to a function that maps from the domain of nouns into the range of noun phrases. The association of this item with a function is represented by writing

- (3) $\text{nice} := NP/N$

wherein ‘:=’ denotes the syntactic type association operator. The argument N of the function NP/N appears to the right of the forward slant or slash character, and the value or result NP is on the left. The NP/N category is referred to as a functional syntactic type; the N and NP represent instances of elementary or atomic syntactic types.

The fact that the slash in the functional NP/N type slants rightward indicates that the noun must be to the right of the adjective with which it is combined. For example, the word ‘dogs’ can be associated as follows with the atomic syntactic type N :

- (4) $\text{dogs} := N$

Application of the function NP/N associated with the word ‘nice’ results in the substring ‘nice dogs’ being analysed as having atomic syntactic type NP .

The forward function application operation may be represented in this example by writing

$$(5) \quad NP/N \ N \Rightarrow \ NP$$

Note that this does not constitute a rule of grammar in the sense of a phrase-structure rule: it simply describes application of the function NP/N to its argument N to derive the result NP .

A verb such as ‘fly’ can be associated as follows with the functional syntactic type $S \backslash NP$:

$$(6) \quad \text{fly} := S \backslash NP$$

The back or leftward slash in $S \backslash NP$ indicates that its argument, an NP , must be to the left. If the item ‘birds’ is classified as a collective noun associated with the atomic type NP as follows:

$$(7) \quad \text{birds} := NP$$

then the string ‘birds fly’ can be analysed as a sentence, with atomic syntactic type S . In this instance, the backward application operation can be represented by writing:

$$(8) \quad NP \ S \backslash NP \Rightarrow \ S$$

A verb such as ‘like’ is usually thought of as taking two arguments and can be associated as follows with an appropriate functional type:

$$(9) \quad \text{like} := (S \backslash NP) / NP$$

The function $(S \backslash NP) / NP$ maps from a domain of atomic NP types into a range comprised of functions with the form $S \backslash NP$. For example, if the second word, ‘cats,’ of the substring ‘like cats’ is associated the NP type as follows:

$$(10) \quad \text{cats} := NP$$

then application of the function $(S \backslash NP) / NP$ results in the two words being identified as having syntactic type $S \backslash NP$. This type is analogous to the verb phrase category constituting the predicate of a sentence.

If the substring ‘like cats’ is preceded by the substring ‘nice dogs’ previously analysed as being in the NP category, then application of the function $S \backslash NP$ yields the result category S . This result signals that the following string:

$$(11) \quad \text{nice dogs like cats}$$

can be analysed as a sentence.

The analysis process can be represented by the following derivation:

$$(12) \quad \frac{\frac{\text{nice} \quad \text{dogs}}{NP/N \quad N} \quad \frac{\text{like} \quad \text{cats}}{(S \backslash NP) / NP \quad NP}}{NP \quad S \backslash NP} \begin{matrix} > \\ < \end{matrix} \\ \frac{}{S}$$

wherein the horizontal lines beneath each functional type and its argument are indexed with the symbols ‘>’ or ‘<’ to mark whether the functional application operation that yields the type below the line is forward or backward, respectively.

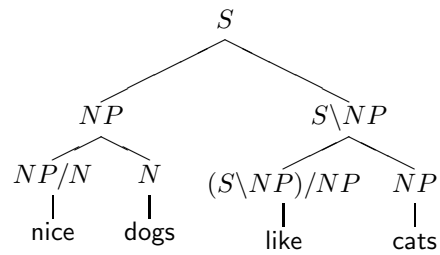


Figure 1: Tree representation of the derivation of the sentence ‘nice dogs like cats.’

The derivation of the sentence in (11) may also be represented with a tree structure as illustrated in Fig. 1; but note that a categorial grammar is normally not considered to yield syntactic trees such as those generated by context-free phrase-structure grammars.