

Lecture #16, Figure

①

Example #1

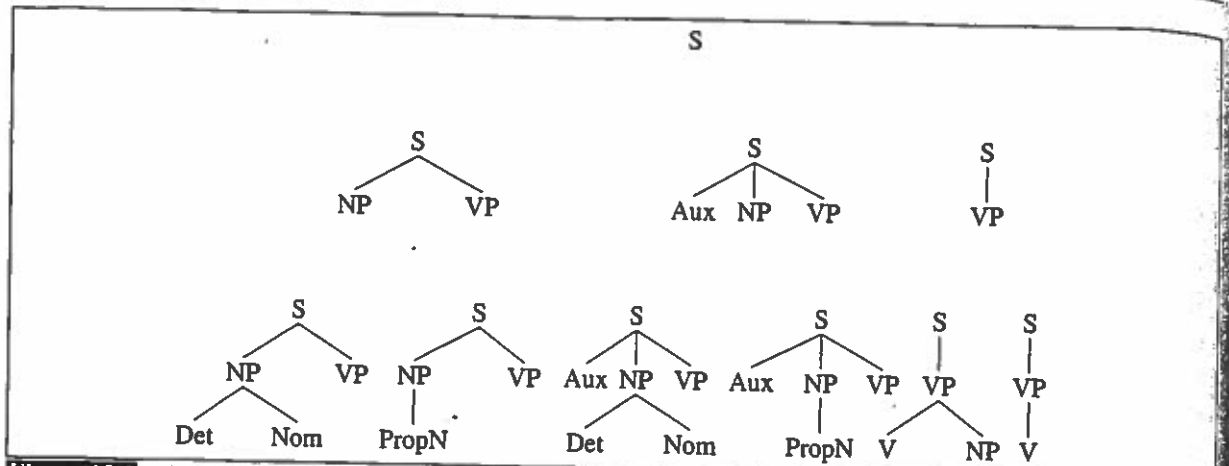


Figure 13.3 An expanding top-down search space. We create each ply by taking each tree from the previous ply, replacing the leftmost non-terminal with each of its possible expansions and collecting each of these trees into a new ply.

Example #2

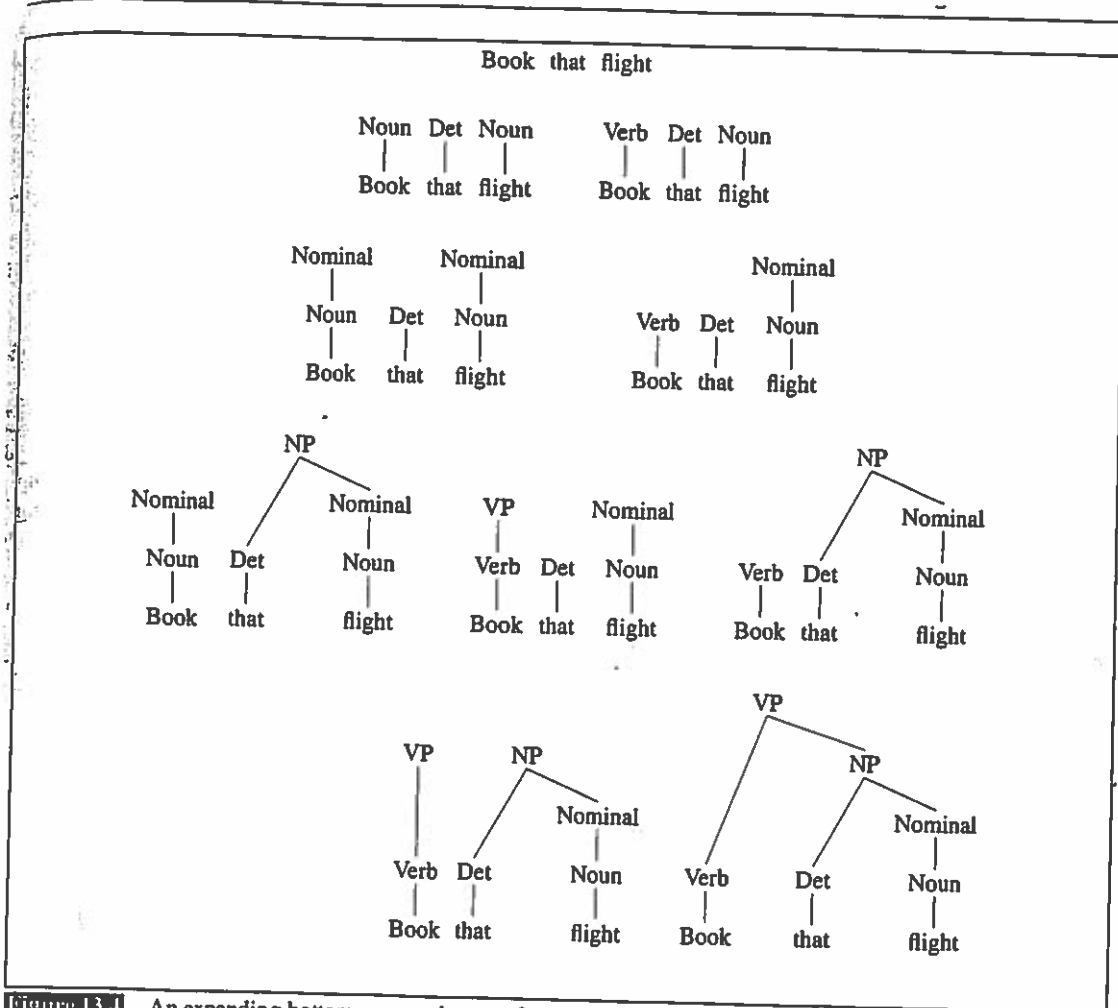


Figure 13.4 An expanding bottom-up search space for the sentence *Book that flight*. This figure does not show the final phase of the search with the correct parse tree (see Fig. 13.2). Make sure you understand how that final parse tree follows from the search space in this figure.

Lecture # 16, Figure

2

Example # 3

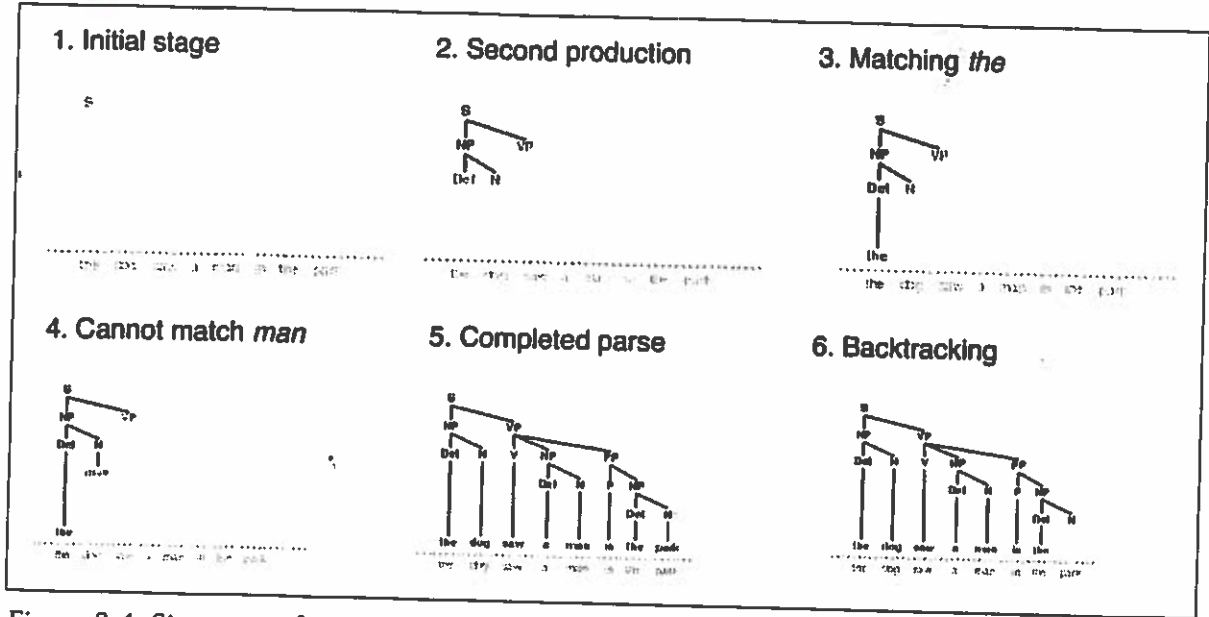


Figure 8-4. Six stages of a recursive descent parser: The parser begins with a tree consisting of the node S; at each stage it consults the grammar to find a production that can be used to enlarge the tree; when a lexical production is encountered, its word is compared against the input; after a complete parse has been found, the parser backtracks to look for more parses.

Example # 4

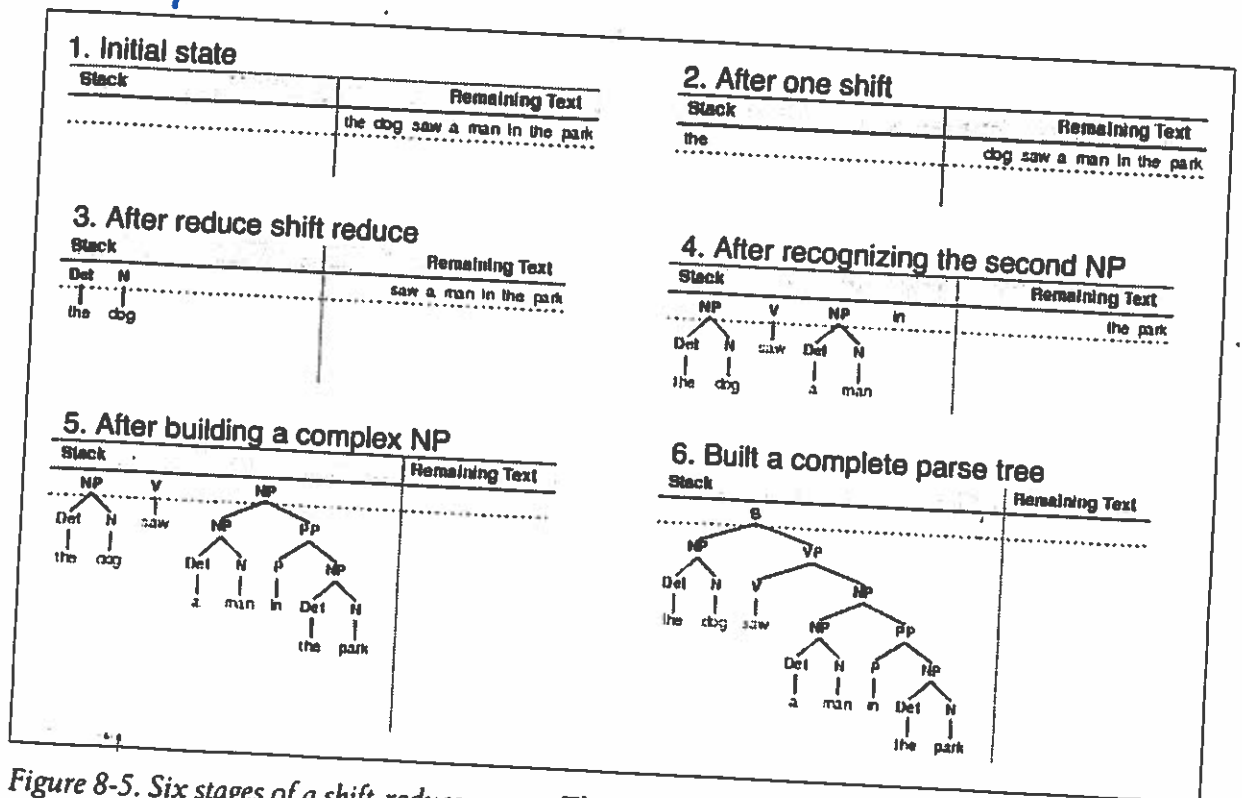


Figure 8-5. Six stages of a shift-reduce parser: The parser begins by shifting the first input word onto its stack; once the top items on the stack match the righthand side of a grammar production, they can be replaced with the lefthand side of that production; the parser succeeds once all input is consumed and one S item remains on the stack.