HEURISTIC PARSING AND SEARCH SPACE PRUNING

José António Mendes †
Pedro Rangel Henriques
INESC-Braga, Universidade do Minho
Largo do Paço, 4719 Braga - Portugal

ABSTRACT

When parsing a Language the search space is traversed in a pre-defined way – top-down or bottom-up, depth-first or breath-first, typically. However, these might not be the ideal strategies: there are languages for which that process is very expensive or even, some valid sentences are not parsed straightly. Our aim is to describe an alternative method where the traversal may be heuristically defined during the process of parsing, depending only on the linguistic phenomena being analyzed. Such parser is data-driven by the string, and heuristically-driven by the annotated grammar rules. This makes it possible to mix top-down and bottom-up and also, to alternate parsing to the left or to the right of the symbol which causes the rule to be triggered. A bidirectional chart parsing was implemented in Prolog to express this idea. The results then obtained gave us some rules of thumb to get the best annotation of a grammar rule set and showed us the gain in time analysis by the spectacular pruning (65% in average for any phenomenon) of the search space.

1. INTRODUCTION

For a given grammar and for a given string, parsing is the process which verifies if the string belongs to the Language, which is generated by the grammar rule set and, if so, it assigns a structure to the string (accordingly to the hypotheses in the grammar). Parsing is, therefore, a search process to match an instance with a set of structural patterns.

The grammar rule set defines the search space. The search strategy defines the way that space is traversed. The search space path is defined by the rules which are verified until the structure of the string is found.

The depth-first strategy creates backtracking problems. By going back and attempt trough another path, it causes a duplication of effort since it tries to prove multiple instances of the same evidence. Another problem is that it may cause an infinite cycle when, depending on the variant of the method used, either left or right recursive rules are used.

In the breath-first strategy there is a duplication of efforts since multiple instances of alternative solutions, which belong to different paths, are found. This implies high cost of memory space since all alternative paths waiting to be analyzed are kept.

† Prof. Assistant in Esc. Superior de Tecnologia de Tomar doing a Ph.D. at the University of Minho
The top-down strategy is predictive and so it can parse rules which derive in the empty string. However it may formulate all hypotheses and traverse the whole search space, even when there is no need to try some hypotheses.

Bottom-up as a data-driven strategy constrains the search space, but it cannot straightly parse the null string since there is no lexical evidence\(^2\). Besides, the rules are always triggered by the first category on the right hand side (RHS) of the rule and, nothing tells us that such category is the best one to trigger the rule in any circumstances.

The method we will use to parse, in the context of ambiguous sentences, is Chart-Parsing, [Win83a] and [Tho85a]. By using charts the evidence are corroborated only once and then kept [Men90a]. So, the evidence may be shared by all hypotheses which need to be corroborated. Therefore, there is no duplication of efforts, there is a reduction of memory space and, very important, backtracking is avoided. The problems of top-down and bottom-up strategies are solved.

We will analyze a strategy based on the idea (coming from [SD87a]) of mixing standard search strategies, combining them not in a pre defined way but in a way that takes into account the string (phenomenon) being analyzed. This idea implies the concept of annotated context-free grammar, which means to annotate a rule by saying if it is to be triggered top-down or bottom-up (in the latter case, also saying which is the best evidence in the RHS to do that triggering), [Men91a].

If the rules are to be triggered by the best evidence (not necessarily the left-most category of the RHS) then the chart parser needs to be modified also to the left of the category that makes the triggering [DE86a], in order to corroborate hypotheses against such constituents\(^3\). This gives rise to the so called bidirectional chart parser [SD87a].

In the next section we present a brief review of search and strategies. In section 3 the concept of annotation of a grammar rule set with heuristics is discussed and some criteria to get the best annotation is presented [MD88a]. Then, in section 4 we talk about chart parsing discussing the modifications to get a bidirectional chart parser. The gain in efficiency due to an heuristic mixed strategy during parsing is analyzed in section 5\(^4\).

2. SEARCH AND STRATEGIES

Parsing the string *I saw a girl with a telescope* will produce a phrase structure tree (PST) as in Fig.1; another PST, in Fig.2, can be derived which show us ambiguity, i.e., there are two

\(^2\) As the left-to-right bottom-up parser is the most used, from now on we will refer to standard bottom-up, or just bottom-up, assuming that it relies on the first symbol on the right hand side of the rule. Problems here presented as inherent to bottom-up strategy can be solved using well known techniques but, one would get a deterministic parser which is not our goal.

\(^3\) Some substructure which belongs to the string.

\(^4\) The examples presented follow the grammar defined on page 13