Appendix 2. Parsing

This section is limited to a detailed examination of Chart Parsing only. Parsers to process DCGs effectively behave in a similar way to a Prolog interpreter [Pereira & Warren]. Parsers which process ATNs are top-down "tree-divers" which use ATN sub-networks to generate their trees.

A2.1 Chart Parsing

Chart Parsing [Gazdar & Mellish] is based around creating and extending instances of *edge*, the parser's main data object. Each edge (also called a chart) spans one or more words of a sentence and is of a specified syntactic category or phrase type. The parser creates an edge from a grammar rule when there is evidence to suggest that a phrase, of the type the rule defines, exists in the sentence. This edge is placed over the sentence starting from the first word that contributes to the phrase. The parser extends the edge as it finds more words that belong to the phrase it represents. The process is seeded by an initial set of edges taken directly from the lexicon definitions of the words that make up the sentence.

The parser maintains lists of active edges and complete edges. Complete edges relate to complete phrases: edges associated with grammar rules whose components have all been found. Active edges are those describing incomplete phrases: edges waiting to be extended as more information about their phrases is discovered by the parser.

For example, given a sentence containing the noun phrase "the black cat" and the grammar rule:

Rule1: NounPhrase → Determiner Adjective Noun

a chart parser (using obvious lexical definitions) creates an active edge (Edge#1 below) on encountering the determiner of the phrase. This active edge requires an adjective followed by a noun to complete.

Edge#1

Created from	Rule1
Status	Active
Category	NounPhrase
Words Used	"the"
Targets	Adjective Noun

At some time later when the adjective "black" is encountered the parser creates a second active edge which encompasses the first but only requires a noun to complete.

Edge#2

Created from	Rule1
Status	Active
Category	NounPhrase
Words Used	"the black"
Targets	Noun

A third edge is created when the noun is found. This one is complete conforming to the description of its syntactic category as described by its parent rule.

Edge#3

Created from	Rule1
Status	Complete
Category	NounPhrase
Words Used	"the black cat"
Targets	nil

New edges are only triggered by other complete edges. Edge#1 above is active so would not trigger edge creation from a rule like:[Rule2: Sentence → NounPhrase VerbPhrase] but Edge#3 (a complete edge) could trigger such a rule.

The chart parser proceeds by grouping edges together to form new edges, Edge#2 above is grouped with a completed Noun edge (formed from the word "cat") to form Edge#3 which is an extended version of Edge#2. An obvious and fundamental constraint on this grouping is that two edges can only be grouped if their relative positions in a sentence make grouping sensible, ie: one follows on from the other.

The parser does not ever modify or destroy edges once they are created so there is never any need to backtrack in order to undo poor grouping decisions. This produces a significant efficiency gain over many other types of parser which do need to backtrack and rebuild phrase structures.

The following example shows the operation of the parser for a complete sentence. The sentence "the old man the boats" exhibits sub-sentence ambiguity because the lexicon describes old as both an adjective and a noun (a collective noun meaning a group of elderly people) and describes man as both a noun and a verb (to control/take charge of).

The diagram shows the production of edges as it would occur in a chart parser implementing a breadth first exhaustive search. Active edges are shown as arrows, completed edges as bars. The grammar is shown below along with the abbreviations used for syntactic categories. The order in which edges are formed is in general indicated by their position in the diagram - those nearer the top of the diagram are formed before those lower down (though some compromise has been made for the sake of readability). The number of the rule which originally creates each edge is shown on the diagram, edges which are created by lexical entries (those corresponding to single words) are labelled *Lex*.

Grammar

R1: S → NP VP
R2: NP → Det Adj Noun
R3: NP → Det Noun
R4: VP → Verb NP

Abbreviations	
S	Sentence
NP	NounPhrase
VP	VerbPhrase
Det	Determiner
Adj	Adjective

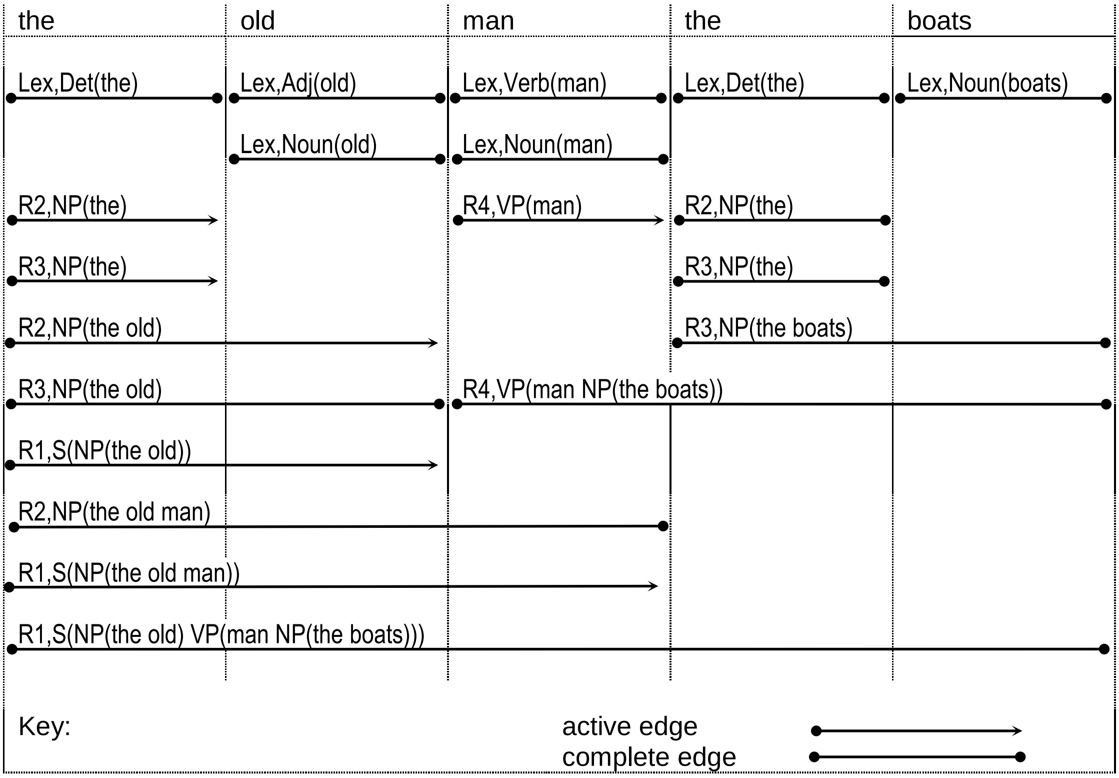


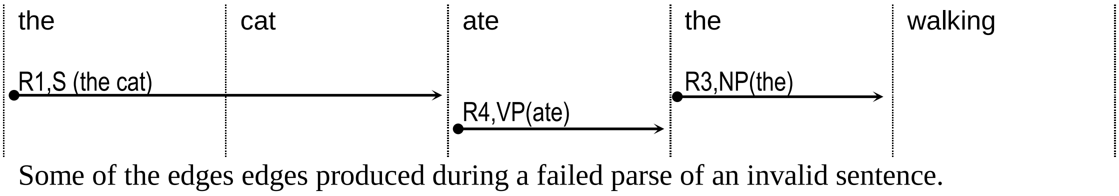
Diagram shows the rule number, syntactic category and word organisation for each edge.

The central control of the parser and its grouping processes is implemented as a search process. As with other search processes minor modifications can be made to effect different types of search. The order in which edges are explored defines the type of search that occurs. If edges are queued for exploration a breadth first search occurs, if they are stacked a depth first search occurs. Other heuristic ordering can effect a variety of searches including best first. Heuristic pruning can be used to delay or prevent explosive state-space growth. However, as in other searching, heuristic pruning may result in a search that is not admissible (one that is no longer guaranteed to find a solution eventhough one exists in state-space). A further complication is that a parser which stops when it finds the first syntactically correct parse structure may not have found the most desirable parse. An ambiguous sentence may have multiple parses, all valid, whose relative merits can only be assessed by further analysis.

Chart parsers can often produce useful information in the event of failed parses. This information comes from examining partial edge data to identify unrecognised or incomplete phrase structures. Consider the syntactically invalid sentence:

*the cat ate the walking

A failed parse would generate an active sentence edge requiring a verb phrase, an active verb phrase requiring a noun phrase and an active noun phrase all requiring a noun to be found where the word *walking* occurs.



Some of the edges edges produced during a failed parse of an invalid sentence.