Consider an ordered sequence of tokens, $\tau = (\tau_1, \tau_2, ..., \tau_n)$. A parse of this sequence provides, under one interpretation, a nested partitioning of the tokens, e.g. either

$$\{\ \{\texttt{The cat}\}\ \{\texttt{sat}\ \{\texttt{on}\ \{\texttt{the mat}\}\ \}\ \}\ \},$$

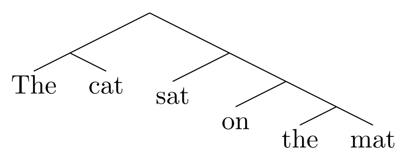
Figure 1: Representing the parse as a bracketing of the tokens.

or

`The cat sat on the mat` .

Figure 2: Representing the parse as a collection of token sub-sequences.

Under another interpretation, the parse forms a tree of nodes, e.g.



.

Figure 3: Representing the parse as a tree of nodes.

Thus, a parse tree can be characterised by a set of nodes and a set of relations or rules linking these nodes. The parse nodes are divided into leaf nodes and derived nodes. A *leaf* node represents all of the known information about a token, including the position of that token in the token sequence. A *derived* node represents a combination of a contiguous sub-sequence of leaf and/or derived nodes.

In order to define node contiguity, first observe from Figure 3 that each node ν spans some contiguous sub-sequence of tokens, e.g. $(\tau_i, \tau_{i+1}, ..., \tau_{i+k})$. Let *beta*($\nu$) be the index of the first token in the sub-sequence, e.g. $\beta(\nu) = i$, and let $\varepsilon(\nu)$ be the index of the last token, e.g. $\varepsilon(\nu) = i + k$. Then the *span* of node ν is defined as the set of token indices

$$\sigma(\nu) = \{j \in \mathcal{I}_n \mid \beta(\nu) \leq j \leq \varepsilon(\nu)\},$$

where $\mathcal{I}_n = \{1, 2, ..., n\}$. Hence, for two arbitrary nodes ν_1 and ν_2 to be said to be *non-overlapping*, their spans must be mutually exclusive, i.e. obey $\sigma(\nu_1) \bigcap \sigma(\nu_2) = \{\}$. Furthermore, for ν_1 and ν_2 to be *adjacent*, their spans must further obey either $\beta(\nu_1) = \varepsilon(\nu_2) + 1$ or $\beta(\nu_2) = \varepsilon(\nu_1) + 1$. Finally, an arbitrary sequence $(\nu_1, \nu_2, ..., \nu_m)$ of nodes is *contiguous* if and only if

$$\beta(\nu_{i+1}) = \varepsilon(\nu_i) + 1,\ \forall i \in \mathcal{I}_{m-1}\,.$$

Continuing now from previous remarks, a parse tree $\mathcal{P}$ may be thought of as a set of node combination rules of the form

$$\nu_1\ \nu_2\ ...\ \nu_m \xrightarrow{\rho} \nu\ .$$

In order to be a valid rule, the sequence $\pi(\rho) = (\nu_1, \nu_2, ..., \nu_m)$ of so-called *predecessor* nodes must be contiguous, such that the resulting derived node $\delta(\rho) = \nu$ has span

$$\sigma(\nu) = \bigcup_{i=1}^{m} \sigma(\nu_i)\ ,$$

where $\beta(\nu) = \beta(\nu_1)$ and $\varepsilon(\nu) = \varepsilon(\nu_m)$. In effect, the predecessor nodes partition the tokens spanned by ν into mutually exclusive sub-sequences.

Taken together, the set $\mathcal{P}$ of rules forms a parse of the whole token sequence, i.e.

$$\exists \rho \in \mathcal{P},\ \sigma(\delta(\rho)) = \mathcal{I}_n\ .$$

Furthermore, the rules must partition the token sequence into a tree of nested sub-sequences, i.e.

$$\forall \rho_1 \in \mathcal{P},\ \forall \rho_2 \in \mathcal{P} \backslash \{\rho_1\},\ \sigma(\delta(\rho_1)) \bigcap \sigma(\delta\rho_2)) \neq \{\} \Rightarrow \sigma(\delta(\rho_1)) \subset \sigma(\delta(\rho_2)) \text{ or } \sigma(\delta(\rho_2)) \subset \sigma(\delta(\rho_1))\ .$$