An Earley-style Recognition Algorithm for MCFGs

Daniel M. Albro

December 5, 2000

1 Introduction

The multiple context-free grammar (MCFG) formalism (Seki *et al.* 1991), an extension of the standard context-free grammar (CFG) formalism, is one of a class of grammars known as "mildly context sensitive." These grammars allow description of the type of phenomena typically analyzed as involving "movement." MCFGs have generative power strictly between tree adjoining grammars (TAGs) or head grammars (HGs) and context sensitive grammars (CSGs), and are equivalent in generative power to linear context-free rewriting systems (LCFRSs) (Seki *et al.* 1991). They are also equivalent to Stabler's (1997) formalization of Chomsky's (1995) minimalist grammars (MGs), and others .

The Earley algorithm (Earley 1970) is an efficient recognition algorithm for CFGs. Presented here is an extension of that algorithm for MCFGs, with a proof of correctness and complexity analysis.

2 Definitions for CFGs

A CFG is a 4-tuple $\langle N, \Sigma, P, S \rangle$, where N is a finite set of *nonterminal symbols*; Σ is a finite set of *terminal symbols* (words); P is a set of *production rules* $\langle A \in N, \alpha \rangle$, in which α is a sequence of terminals or nonterminals; and $S \in N$ is a *start symbol*. We will represent a production $\langle A, \alpha \rangle$ using the following notation: $A \to \alpha$. For example, a simple subset of English might be represented by the following grammar:

$$\left[\begin{array}{l} N = \{\mathrm{S}, \mathrm{NP}, \mathrm{N}, \mathrm{VP}, \mathrm{V}, \mathrm{IV}\}, \\ \Sigma = \{\mathrm{John}, \mathrm{Mary}, \mathrm{loves}, \mathrm{sits}\}, \\ P = \left\{\begin{array}{ll} \mathrm{S} \to \mathrm{NP\,VP}, & \mathrm{NP} \to \mathrm{N}, \\ \mathrm{N} \to \mathrm{John}, & \mathrm{N} \to \mathrm{Mary}, \\ \mathrm{VP} \to \mathrm{V\,NP}, & \mathrm{VP} \to \mathrm{IV}, \\ \mathrm{V} \to \mathrm{loves}, & \mathrm{IV} \to \mathrm{sits} \end{array}\right\}, \\ S = \mathrm{S} \end{array}\right] \tag{1}$$

This grammar generates the strings "John loves Mary," "Mary loves John," "John loves John," "Mary loves Mary," "John sits," "Mary sits," and no others.

2.1 Variable Conventions for CFGs

We will use the variables $A, B, C \in N$ to denote categories, $a, b, c \in \Sigma$ to denote terminals, $\alpha, \beta, \gamma, \delta, \varphi, \vartheta \in V^*$ (where $V = N \cup \Sigma$) to denote sequences of terminals or nonterminals, $v, w \in V$ to denote single terminals or nonterminals (left unspecified as to which), and $\theta \in \Sigma^*$ to denote a sequence of terminal symbols, *i.e.* a string.

2.2 Definitions for CFGs

We define a *rewrites* relation $\Rightarrow$ with respect to a grammar $G = \langle N, \Sigma, P, S \rangle$ as follows:

$$\alpha \Rightarrow \beta \text{ iff } \exists \alpha_1, \alpha_2 \text{ s.t. } \alpha = \alpha_1 A \alpha_2 \wedge \beta = \alpha_1 \gamma \alpha_2 \wedge A \rightarrow \gamma \in P$$

We say that α *derives* β in G if $\langle \alpha, \beta \rangle$ is in the reflexive, transitive closure $\Rightarrow_G^*$ of $\Rightarrow_G$. We can further note the number of rewrite steps involved in a derivation as follows:

$$\alpha \Rightarrow^n \beta \text{ iff } \alpha \Rightarrow \alpha_1 \Rightarrow \ldots \Rightarrow \alpha_{n-1} \Rightarrow \beta$$

If the productions of P have been indexed by a bijection $I : (N \times V^*) \rightarrow \mathbb{N}$, then we can indicate the first production used in a derivation as follows: $\alpha \Rightarrow_r \beta$. This indicates that the production $A \rightarrow \beta$ referred to in the definition of $\Rightarrow$ has index r (that is, $I(A \rightarrow \beta) = r$). The notation $\alpha \Rightarrow_r^* \beta$ is shorthand for $\alpha \Rightarrow_r \alpha' \Rightarrow^* \beta$.

A series of derivation steps defines a *derivation tree*, which consists of two binary relations: dominance (D) and precedence ($\prec$). The relations D and $\prec$ of a derivation tree are defined based on derivation steps $\alpha \Rightarrow \beta$ via the following rules:

$$A D v \text{ iff } \exists \alpha_1, \alpha_2 (\alpha \Rightarrow \beta \text{ via } A \rightarrow \alpha_1 v \alpha_2)$$
$$v \prec w \text{ iff } \exists \alpha_1, \alpha_2 (\alpha = \alpha_1 v w \alpha_2 \vee \beta = \alpha_1 v w \alpha_2)$$

Note that we will sometimes have cause to refer to the transitive closure $\prec^+$ of the precedence relation.

Following our example grammar in (1), and numbering the rules from left to right and top to bottom, we can derive the sentence "John loves Mary" as follows[1]:

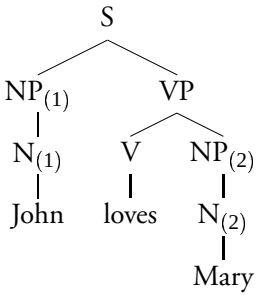
$$\begin{aligned} \text{S} \quad &\Rightarrow_1 \quad \text{NP}_{(1)}\,\text{VP} \Rightarrow_2 \text{N}_{(1)}\,\text{VP} \Rightarrow_3 \text{John VP} \Rightarrow_5 \text{John V NP}_{(2)} \Rightarrow_7 \text{John loves NP}_{(2)} \\ &\Rightarrow_2 \quad \text{John loves N}_{(2)} \Rightarrow_4 \text{John loves Mary} \end{aligned}$$

From this derivation we get the following relations:

$\text{NP}_{(1)}$	$\prec$	VP	S	D	$\text{NP}_{(1)}$
$\text{N}_{(1)}$	$\prec$	VP	S	D	VP
John	$\prec$	VP	$\text{NP}_{(1)}$	D	$\text{N}_{(1)}$
John	$\prec$	V	$\text{N}_{(1)}$	D	John
V	$\prec$	$\text{NP}_{(2)}$	VP	D	V
John	$\prec$	loves	VP	D	$\text{NP}_{(2)}$
loves	$\prec$	$\text{NP}_{(2)}$	V	D	loves
loves	$\prec$	$\text{N}_{(2)}$	$\text{NP}_{(2)}$	D	$\text{N}_{(2)}$
loves	$\prec$	Mary	$\text{N}_{(2)}$	D	Mary

[1] We are using parenthesized subscripts on nonterminals to distinguish between distinct nodes with identical labels.

The graph induced by these relations can be represented by the following tree:



The *yield* $Y(A)$ of a category A consists of all strings derivable from A. In mathematical terms, $Y(A) = \{\theta \in \Sigma^* | A \Rightarrow^* \theta\}$.

3 Definitions for MCFG

3.1 Components of an MCFG

An MCFG is much like a CFG, except that the yield of a category need not be a simple set of strings, but may instead be a set of tuples of strings. Just as for CFGs, an MCFG is a tuple $G = \langle N, \Sigma, P, S \rangle$, but the definitions of each of these components differs somewhat from the CFG definition[2]:

1. N is a finite set of nonterminals, as for CFGs. For each $A \in N$, we define a positive integer $\dim(A)$, called the *dimension* of A. If $\dim(A) = 1$, A is termed a *simple nonterminal*, otherwise it is a *complex nonterminal*. To distinguish the complex and simple nonterminals, we introduce two subsets of N: (1) $N_s = \{A \in N | \dim(A) = 1\}$ is the set of simple nonterminals, and (2) $N_c = N - N_s$ is the set of complex nonterminals. The yield $Y(A)$ of a category A is a tuple of arity $\dim(A)$, that is, a subset of $(\Sigma^*)^{\dim(A)}$. A may be written as $\langle A^{[1]}, \ldots, A^{[\dim(A)]} \rangle$, in which $A^{[1]}, \ldots, A^{[\dim(A)]}$ are the *component symbols* of A. In order to refer to the set of all component symbols for an MCFG, we define the set $N_{CMP} = \{A^{[i]} | A \in N, 1 \leq i \leq \dim(A)\}$. Taking the grammar as a whole, we may define its dimension as the maximum dimension of any of its component nonterminals: $\dim(G) = \max\{\dim(A) | A \in N\}$.

2. Σ is a finite set of terminals, as for CFGs.

3. P is a finite set of production rules. A rule r has the form $A \rightarrow \langle \alpha_1, \ldots, \alpha_{\dim(A)} \rangle$, where $A \in N$, $\gamma_1, \ldots, \gamma_{\dim(A)} \in (N_{CMP} \cup \Sigma)^*$. Each rule r satisfies the following condition:

   > **Weak Right-linearity:** For each complex nonterminal B and each component symbol $B^{[i]}$ of B $(1 \leq i \leq \dim(B))$, $B^{[i]}$ appears in the right hand side (*rhs*) of r at most once.

   A is called the nonterminal in the left hand side (*lhs*) of r. $B \in N$ is called a nonterminal in the rhs if some $B^{[i]}$ appears in the rhs. We call γ_k the k-th component of the rhs. For each production r we

[2] This formulation of an MCFG is based on that of Nakanishi *et al.* (2000).

define a constant c_r equal to the number of nonterminals in the rhs of r. Similarly we define c'_r as the number of complex nonterminals in the rhs. For a grammar as a whole we say $c = \max\{c_r | r \in P\}$ and $c' = \max\{c'_r | r \in P\}$.

4. $S \in N$ is the start symbol, with $\dim(S) = 1$.

In the context of an MCFG the set V refers to $N_{CMP} \cup \Sigma$. As before, it is the set of permissible symbols, that is the vocabulary for the right hand sides of production rules.

3.2 Derivations in an MCFG

For a context free grammar the rewrite relation $\Rightarrow$ is defined on sequences. For a multiple context free grammar we define the rewrite relation on *linked sequences*. A linked sequence is a sequence in which some elements are linked to other elements. In such a sequence, a given element may be a member of no more than one *linkage group*, where a linkage group is a set of linked elements. The linkage group of which the element is a member is indicated as a superscript integer on the element. If no linkage group is indicated for an element, that element is linked to no other elements. For example, in the sequence $\alpha_1 A^{[i],1} \alpha_2 A^{[j],2} \alpha_3 A^{[k],1} \alpha_4$, $A^{[i]}$ is linked to $A^{[k]}$, but neither of these is linked to $A^{[j]}$. In the case of MCFGs a linkage group may only contain distinct components of a single nonterminal symbol. For example, a sequence might link together elements $A^{[1]}$, $A^{[7]}$, and $A^{[3]}$, but it could not link together two of the same component ($A^{[3]}$ and $A^{[3]}$, for example) or components of different nonterminals, as in $A^{[1]}$, $B^{[2]}$.

For MCFGs, then, the variables $\alpha, \beta, \gamma, \delta, \varphi, \vartheta$ refer to linked sequences from the set $(N_{CMP} \cup N_{CMP} \times \mathbb{N} \cup \Sigma)^*$.

We define the rewrite relation for MCFGs as follows. For linked sequences α and β, $\alpha \Rightarrow \beta$ iff:

1. $\alpha = \alpha_1 A^{[q_1],i} \alpha_2 \dots \alpha_{2m-1} A^{[q_m],i} \alpha_{2m}$ for some $Q = \{q_1, \dots, q_m\}$ such that $\forall q_i, 1 \leq q_i \leq \dim(A)$. In this expression no α_j may contain a nonterminal $A^{[q],i}$. In other words, the linked sequence α contains a set $\{A^{[q_1]}, \dots, A^{[q_m]}\}$ of linked nonterminals.

2. There exists a production rule $A \rightarrow \langle \gamma_1, \dots, \gamma_{\dim(A)} \rangle \in P$ with index r. Note that in all right hand sides of MCFG productions all components of any given complex category are considered to be linked, with a different linkage group for each. For example, $\mathrm{A} \rightarrow \langle \mathrm{B}^{[1]}\mathrm{C}^{[1]}, \mathrm{dB}^{[1]}\mathrm{C}^{[2]}\mathrm{D}^{[2]}, \mathrm{D}^{[1]} \rangle$, where categories C and D are complex, could be written $\mathrm{A} \rightarrow \langle \mathrm{B}^{[1]}, \mathrm{C}^{[1],1}, \mathrm{dB}^{[1]}\mathrm{C}^{[2],1}\mathrm{D}^{[2],2}, \mathrm{D}^{[1],2} \rangle$, or even more explicitly as $\mathrm{A} \rightarrow \langle \mathrm{B}^{[1],1}, \mathrm{C}^{[1],2}, \mathrm{dB}^{[1],3}\mathrm{C}^{[2],2}\mathrm{D}^{[2],4}, D^{[1],4} \rangle$.

3. $\beta = \alpha_1 \gamma'_{q_1} \alpha_2 \dots \alpha_{2m-1} \gamma'_{q_m} \alpha_{2m}$, where $\gamma_i = v_1 \dots v_n$ and $\gamma'_i = h(v_1) \dots h(v_n)$. The function h is defined as follows:

$$\begin{aligned} h(A^{[j]}) &= A^{[j],k} \text{ if } A \in N_c \\ h(A^{[j]}) &= A^{[j]} \text{ if } A \in N_s \\ h(a) &= a \text{ if } a \in \Sigma \end{aligned}$$

where the k values are chosen such that there are no linkage groups k mentioned in $\alpha_1 \dots \alpha_{2m}$ and all components of the same complex category in r are assigned the same linkage group k.

For example, given the following grammar

$$\left[\begin{array}{lcl} N & = & \{\mathrm{S},\mathrm{A},\mathrm{B},\mathrm{N},\mathrm{V},\mathrm{C}\} \\ \Sigma & = & \{\mathrm{John},\mathrm{Jim},\mathrm{George},\mathrm{Sue},\mathrm{what},\mathrm{eats},\mathrm{drinks}\} \\ P & = & \left\{\begin{array}{lcl} \mathrm{S} & \rightarrow & \mathrm{C}^{[1]}\,\mathrm{A}^{[1]}\,\mathrm{B}^{[1]}\,\mathrm{A}^{[2]} \\ \mathrm{C} & \rightarrow & \mathrm{N}^{[1]}\,\mathrm{N}^{[2]} \\ \mathrm{B} & \rightarrow & \mathrm{what} \\ \mathrm{A} & \rightarrow & (\mathrm{N}^{[1]}\,\mathrm{V}^{[1]},\mathrm{N}^{[2]}\,\mathrm{V}^{[2]}) \\ \mathrm{N} & \rightarrow & (\mathrm{John},\mathrm{Jim}) \\ \mathrm{N} & \rightarrow & (\mathrm{George},\mathrm{Sue}) \\ \mathrm{V} & \rightarrow & (\mathrm{eats},\mathrm{eats}) \\ \mathrm{V} & \rightarrow & (\mathrm{drinks},\mathrm{drinks}) \end{array}\right\} \\ S & = & \mathrm{S} \end{array}\right] \tag{2}$$

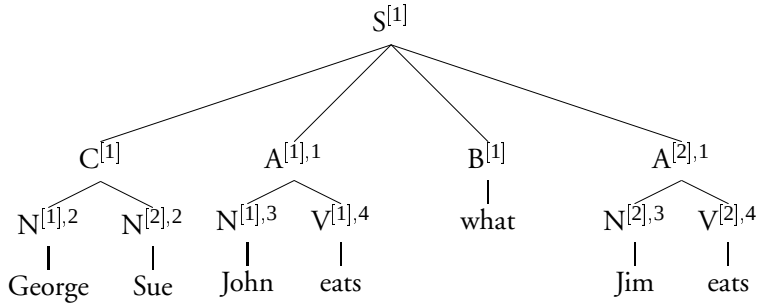
we have the following derivation of "George,Sue–John eats what Jim eats":

$$\begin{array}{rl} \mathrm{S}^{[1]} & \Rightarrow_1 \ \mathrm{C}^{[1]}\,\mathrm{A}^{[1],1}\,\mathrm{B}^{[1]}\,\mathrm{A}^{[2],1} \\ & \Rightarrow_2 \ \mathrm{N}^{[1],2}\,\mathrm{N}^{[2],2}\,\mathrm{A}^{[1],1}\,\mathrm{B}^{[1]}\,\mathrm{A}^{[2],1} \\ & \Rightarrow_4 \ \mathrm{N}^{[1],2}\,\mathrm{N}^{[2],2}\,\mathrm{N}^{[1],3}\,\mathrm{V}^{[1],4}\,\mathrm{B}^{[1]}\,\mathrm{N}^{[2],3}\,\mathrm{V}^{[2],4} \\ & \Rightarrow_6 \ \mathrm{George}\,\mathrm{Sue}\,\mathrm{N}^{[1],3}\,\mathrm{V}^{[1],4}\,\mathrm{B}^{[1]}\,\mathrm{N}^{[2],3}\,\mathrm{V}^{[2],4} \\ & \Rightarrow_5 \ \mathrm{George}\,\mathrm{Sue}\,\mathrm{John}\,\mathrm{V}^{[1],4}\,\mathrm{B}^{[1]}\,\mathrm{Jim}\,\mathrm{V}^{[2],4} \\ & \Rightarrow_7 \ \mathrm{George}\,\mathrm{Sue}\,\mathrm{John}\,\mathrm{eats}\,\mathrm{B}^{[1]}\,\mathrm{Jim}\,\mathrm{eats} \\ & \Rightarrow_3 \ \mathrm{George}\,\mathrm{Sue}\,\mathrm{John}\,\mathrm{eats}\,\mathrm{what}\,\mathrm{Jim}\,\mathrm{eats} \end{array}$$

We can define a derivation tree from a series of rewrite relations $\alpha \Rightarrow \beta$ in the same way as for CFGs. A derivation tree is a pair of relations $\langle \prec, D \rangle$, defined in the much the same way as before:

$$\begin{array}{r} AD\nu \text{ iff } \exists \alpha_1,\alpha_2,\gamma_1,\gamma_2(\alpha_1 A^{[q]}\alpha_2 \Rightarrow \beta \text{ via } A \rightarrow \langle \ldots,\gamma_q = \gamma_1\nu\gamma_2,\ldots\rangle) \\ \nu \prec w \text{ iff } \exists \alpha_1,\alpha_2(\alpha = \alpha_1\nu w\alpha_2 \vee \beta = \alpha_1\nu w\alpha_2) \end{array}$$

The derivation tree for the example derivation above thus looks like:



where linkage group superscripts have been included for illustration only.

Yield of simple categories can be defined in terms of the rewrite relation given above in the expected way: for $A \in N_s$, $Y(A) = \{\theta \in \Sigma^* | A \Rightarrow^* \theta\}$. However, category yield for MCFGs, as well as derivation trees, is usually defined in a bottom-up manner rather than top-down as we have done here. The standard formulation, from Seki *et al.* (1991) and Nakanishi *et al.* (2000), is as follows:

1. If a terminating rule $A \to \langle \theta_1, \ldots, \theta_{\dim(A)} \rangle$ is in P, then $\langle \theta_1, \ldots, \theta_{\dim(A)} \rangle$ is in $Y(A)$.
2. Let $r : A \to \langle \gamma_1, \ldots, \gamma_{\dim(A)} \rangle \in P$ be a nonterminating rule and let $B_1, \ldots, B_n$ be the nonterminals in the right hand side of r. Suppose $\langle \theta_{i,1}, \ldots, \theta_{i,\dim(B)} \rangle \in Y(B_i)$ for $1 \leq i < n$. Then $Y(A)$ contains the $\dim(A)$-tuple of strings obtained from $\langle \gamma_1, \ldots, \gamma_{\dim(A)} \rangle$ by replacing each component symbol $B_i^{[j]}$ with $\theta_{i,j}$ for $1 \leq i < n, 1 \leq j \leq \dim(B_i)$.
3. $Y(A)$ has no element other than those obtained by (1) or (2).

3.3 Connections between MCFGs and CFGs

For a given normal-form MCFG grammar $G = \langle N, \Sigma, P, S \rangle$ we can define a *superset CFG* $\mathrm{cfg}(G) = \langle N_{CMP}, \Sigma, P', S^{[1]} \rangle$ where $P' = \{A^{[q]} \to \gamma_q | A \to \langle \gamma_1, \ldots, \gamma_q, \ldots \gamma_{\dim(A)} \rangle \in P\}$. The language of the superset CFG of an MCFG is a superset of the language of the MCFG. Of value later will be a relation that can be defined between derivation trees in the two grammars. In particular, a derivation tree t' for $\mathrm{cfg}(G)$ is a *CFG subtree* of a derivation tree t for G iff the following holds of it:

$$\begin{array}{rcl} v \prec_{t'}^{+} w & \text{implies} & v \prec_{t}^{+} w \\ vD_{t'}w & \text{implies} & vD_t w \end{array}$$

That is, t' is a CFG subtree of t if and only if the relations by which t' is defined are a subset of the relations by which t is defined.

4 The Earley-style Recognition Algorithm

We present two Earley-style recognizers, one for CFGs and one for MCFGs. The formulation of the CFG recognizer is taken from Sikkel (1998). These are agenda-driven, chart-based recognizers, grounded in a logical perspective on parsing as presented in Shieber *et al.* (1995). The definition of the recognizer is in two parts: a deduction procedure and a grammatical deductive system. The deductive system uses formulae known as *items* which express claims about grammatical properties of strings. For a given grammar and input string the deductive system begins with a set of items, the axioms, which are given without proof and computes the closure of this set using some specified inference rules. The input string has been proven to be in the language of the grammar if a *goal item* is present in the closure. Such an item in fact represents the claim that the input string is grammatical.

4.1 Deduction Procedure

The deduction procedure used here is that of Shieber *et al.* (1995). It employs a *chart* and an *agenda*, both of which are data structures (for our purposes we can treat them as sets) which hold items. The chart holds unique

items in order to avoid applying any rule of inference more than once to the same set of items. The agenda holds items whose consequences under the inference rules have not yet been generated. The procedure is as follows:

1. Initialize the chart to the empty set of items and the agenda to the axioms of the deductive system.
2. Repeat until the agenda is empty:
   (a) Select an item from the agenda—the *trigger* item—and remove it.
   (b) Add the trigger item to the chart, if it isn't there already.
   (c) If the trigger item was added to the cart, generate all items that can be derived from the trigger and any items in the chart by one application of a rule of inference, and add these generated items to the agenda.
3. If the goal item is in the chart, then the string is recognized; otherwise, it is not.

Shieber *et al.* (1995) have proved that the deductive procedure is correct in the sense that it generates all and only the items that are derivable from the axioms via the inference rules. Because of this it is now necessary only to prove that the inference rules are correct and axioms are correct.

4.2 For CFGs

We will begin by presenting the deductive system for CFGs to make the terminology clear and to allow comparisons later.

4.2.1 Items

An item is a 3-tuple $\langle A \to \alpha \cdot \beta, i, j \rangle$, which can be interpreted as saying that category A can be derived from the start category S, that a production rule $A \to \alpha\beta$ exists in P, and that α derives the portion of the input between i and j, that is $\alpha \Rightarrow^* a_{i+1} \dots a_j$.

4.2.2 Deductive Steps

A particular input $\theta = a_1 \dots a_n$ is in the language of G (that is, in $L(G)$) if and only if the item $\langle S' \to S\cdot, 0, n \rangle$ is in the closure of the set $\{\langle S' \to \cdot S, 0, 0 \rangle\}$[3] (where $S' \notin N$) under the following deductive operations:

$\mathbf{D}_{\text{scan}}$: $\frac{\langle A \to \alpha \cdot a\beta, i, j \rangle}{\langle A \to \alpha a \cdot \beta, i, j+1 \rangle}$ if $a = a_{j+1}$

$\mathbf{D}_{\text{comp}}$: $\frac{\langle A \to \alpha \cdot B\beta, i, j \rangle, \langle B \to \gamma \cdot, j, k \rangle}{\langle A \to \alpha B \cdot \beta, i, k \rangle}$

$\mathbf{D}_{\text{pred}}$: $\frac{\langle A \to \alpha \cdot B\beta, i, j \rangle}{\langle B \to \cdot \gamma, j, j \rangle}$ if $B \to \gamma \in P$

[3] Note that this item makes the claim that the higher start symbol S' derives nothing before the start of the string, which is clearly correct.

4.2.3 Example

To check whether the sentence "John loves Mary" is in the grammar defined in example (1) we would start with a single item $\langle \mathrm{S}' \to \cdot \mathrm{S}, 0, 0\rangle$ in our closure. We could then apply operation D_{pred} to that item to deduce $\langle \mathrm{S} \to \cdot \mathrm{NP\,VP}, 0, 0\rangle$. From $\langle \mathrm{S} \to \cdot \mathrm{NP\,VP}, 0, 0\rangle$ we deduce $\langle \mathrm{NP} \to \cdot \mathrm{N}, 0, 0\rangle$, also via D_{pred}, and from there we can get $\langle \mathrm{N} \to \cdot \mathrm{John}, 0, 0\rangle$. At this point D_{scan} comes into play and we can deduce $\langle \mathrm{N} \to \mathrm{John} \cdot, 0, 1\rangle$. This last item is the first *completed item* in the closure, and it allows us to apply the D_{comp} deduction rule to get $\langle \mathrm{NP} \to \mathrm{N} \cdot, 0, 1\rangle$, which gets us $\langle \mathrm{S} \to \mathrm{NP} \cdot \mathrm{VP}, 0, 1\rangle$ in the same way. From here we revert to D_{pred} to get $\langle \mathrm{VP} \to \cdot \mathrm{V\,NP}, 1, 1\rangle$ (we also get $\langle \mathrm{VP} \to \cdot \mathrm{IV}, 1, 1\rangle$, but it won't turn out to be useful), and thence $\langle \mathrm{V} \to \cdot \mathrm{loves}, 1, 1\rangle$, which starts the scan-complete cycle over again, giving us $\langle \mathrm{V} \to \mathrm{loves} \cdot, 1, 2\rangle$ and then $\langle \mathrm{VP} \to \mathrm{V} \cdot \mathrm{NP}, 1, 2\rangle$. Moving on to further predictions, we get $\langle \mathrm{NP} \to \cdot \mathrm{N}, 2, 2\rangle$ and $\langle \mathrm{N} \to \cdot \mathrm{Mary}, 2, 2\rangle$, which leads us to a final round of scan-complete: $\langle \mathrm{N} \to \mathrm{Mary} \cdot, 2, 3\rangle$ leads to $\langle \mathrm{NP} \to \mathrm{N} \cdot, 2, 3\rangle$, from there to $\langle \mathrm{VP} \to \mathrm{V\,NP} \cdot, 1, 3\rangle$, then to $\langle \mathrm{S} \to \mathrm{NP\,VP} \cdot, 0, 3\rangle$, and finally to $\langle \mathrm{S}' \to \mathrm{S} \cdot, 0, 3\rangle$. We can think of this closure-building operation as a proof that the sentence is in the grammar, presenting it as follows:

$$
\begin{aligned}
\langle \mathrm{S}' \to \cdot S, 0, 0\rangle \quad &\vdash \quad \langle \mathrm{S} \to \cdot \mathrm{NP\,VP}, 0, 0\rangle \, [D_{pred}] && (3)\\
&\vdash \quad \langle \mathrm{NP} \to \cdot \mathrm{N}, 0, 0\rangle \, [D_{pred}(3)] && (4)\\
&\vdash \quad \langle \mathrm{N} \to \cdot \mathrm{John}, 0, 0\rangle \, [D_{pred}(4)] && (5)\\
&\vdash \quad \langle \mathrm{N} \to \mathrm{John} \cdot, 0, 1\rangle \, [D_{scan}(5)] && (6)\\
&\vdash \quad \langle \mathrm{NP} \to \mathrm{N} \cdot, 0, 1\rangle \, [D_{comp}(4, 6)] && (7)\\
&\vdash \quad \langle \mathrm{S} \to \mathrm{NP} \cdot \mathrm{VP}, 0, 1\rangle \, [D_{comp}(3, 7)] && (8)\\
&\vdash \quad \langle \mathrm{VP} \to \cdot \mathrm{V\,NP}, 1, 1\rangle \, [D_{pred}(8)] && (9)\\
&\vdash \quad \langle \mathrm{V} \to \cdot \mathrm{loves}, 1, 1\rangle \, [D_{pred}(9)] && (10)\\
&\vdash \quad \langle \mathrm{V} \to \mathrm{loves} \cdot, 1, 2\rangle \, [D_{scan}(10)] && (11)\\
&\vdash \quad \langle \mathrm{VP} \to \mathrm{V} \cdot \mathrm{NP}, 1, 2\rangle \, [D_{comp}(9, 11)] && (12)\\
&\vdash \quad \langle \mathrm{NP} \to \cdot \mathrm{N}, 2, 2\rangle \, [D_{pred}(12)] && (13)\\
&\vdash \quad \langle \mathrm{N} \to \cdot \mathrm{Mary}, 2, 2\rangle \, [D_{pred}(13)] && (14)\\
&\vdash \quad \langle \mathrm{N} \to \mathrm{Mary} \cdot, 2, 3\rangle \, [D_{scan}(14)] && (15)\\
&\vdash \quad \langle \mathrm{NP} \to \mathrm{N} \cdot, 2, 3\rangle \, [D_{comp}(13, 15)] && (16)\\
&\vdash \quad \langle \mathrm{VP} \to \mathrm{V\,NP} \cdot, 1, 3\rangle \, [D_{comp}(12, 16)] && (17)\\
&\vdash \quad \langle \mathrm{S} \to \mathrm{NP\,VP} \cdot, 0, 3\rangle \, [D_{comp}(8, 17)] && (18)\\
&\vdash \quad \langle \mathrm{S}' \to \mathrm{S} \cdot, 0, 3\rangle \, [D_{comp}(3', 18)], \; Q.E.D. && (19)
\end{aligned}
$$

4.3 For MCFGs

The components of the deductive system for MCFGs are essentially extensions of the CFG components. MCFG items contain all of the information that CFG items do, plus some extra information, and the derivation steps are essentially the same as for CFGs, but with some extra twists.

4.3.1 Items

An item is a 5-tuple $\langle A^{[q]} \to \alpha \cdot \beta, i, j, r, M \rangle$, which can be interpreted as saying that component q of category A can be derived from the start symbol S, that there is a production rule $A \to \langle \ldots, \gamma_q, \ldots \rangle$ (nule number r), where $\gamma_q = \alpha\beta$, and that $\alpha \Rightarrow^* a_{i+1} \ldots a_j$. Additionally, the M component gives some information about the derivational history of the item. This component is a function from a category B to the $\langle q, r, i, j \rangle$ values of the item $\langle B^{[q]} \to \gamma \cdot, i, j, r, M' \rangle$ that last completed an element of category B in the derivation level of α from S (that is, for some $Q = \{s_1, \ldots, s_p\}$, where $\forall s_i \in Q, 1 \leq s_i \leq \dim(A)$ and $q \in Q$, we have a derivation $S^{[1]} \Rightarrow^* \alpha_1 A^{[s_1]} \alpha_2 \ldots \alpha_m A^{[s]} \alpha_{m+1} \ldots \alpha_{2p-1} A^{[s_p]} \alpha_{2p} \Rightarrow \delta_1 \alpha \beta \delta_2$ and M records for each multiarity category B the values of $\langle q, r, i, j \rangle$ for the item that completed the final appearance of B in the sequence $\delta_1 \alpha$ in order to derive the item $\langle A^{[q]} \to \alpha \cdot \beta, i, j, r, M \rangle$.

For convenience, in the following sections we will make use of a function modification operator $\oplus$, defined as follows:

$$M \oplus \langle B, q, r, i, j \rangle = \begin{cases} (M - \{\langle B, q', r', i', j' \rangle\}) \cup \{\langle B, q, r, i, j \rangle\} & \text{if } \exists q', r', i', j' (\langle B, q', r', i', j' \rangle \in M) \\ M \cup \{\langle B, q, r, i, j \rangle\} & \text{otherwise} \end{cases}$$

We will also use the notation $M(B)\downarrow$ to indicate that the function M is undefined for the category B, or in other words, that $\nexists q, r, l, m (\langle B, q, r, l, m \rangle \in M)$.

4.3.2 Deduction Steps

A particular input $\theta = a_1 \ldots a_n$ is in the language defined by G (that is, the set $L(G)$ of strings in $Y(S)$ for grammar G) if and only if the item $\langle S'^{[1]} \to S^{[1]} \cdot, 0, n, 0, \emptyset \rangle$ is in the closure of the set $\{\langle S'^{[1]} \to \cdot S^{[1]}, 0, 0, 0, \emptyset \rangle\}$[4] (where $S' \notin N$) under the following deductive operations:

$\mathbf{D}_{\mathbf{scan}}$: $\frac{\langle A^{[q]} \to \alpha \cdot a\beta, i, j, r, M \rangle}{\langle A^{[q]} \to \alpha a \cdot \beta, i, j+1, r, M \rangle}$ if $a = a_{j+1}$

$\mathbf{D}_{\mathbf{comp_1}}$: $\frac{\langle A^{[q_1]} \to \alpha \cdot B^{[q_2]}\beta, i, j, r_1, M_1 \rangle, \langle B^{[q_2]} \to \gamma \cdot, j, k, r_2, M_2 \rangle}{\langle A^{[q_1]} \to \alpha B^{[q_2]} \cdot \beta, i, k, r_1, M_1 \rangle}$ if $\dim(B) = 1$

$\mathbf{D}_{\mathbf{comp_2}}$: $\frac{\langle A^{[q_1]} \to \alpha \cdot B^{[q_2]}\beta, i, j, r_1, M_1 \rangle, \langle B^{[q_2]} \to \gamma \cdot, j, k, r_2, M_2 \rangle}{\langle A^{[q_1]} \to \alpha B^{[q_2]} \cdot \beta, i, k, r_1, M_1 \oplus \langle B, q_2, r_2, j, k \rangle \rangle}$ if $\dim(B) > 1 \wedge (M_1(B)\downarrow \vee \exists q, l, m (M_1(B) = \langle q, r_2, l, m \rangle))$

$\mathbf{D}_{\mathbf{pred_1}}$: $\frac{\langle A^{[q_1]} \to \alpha \cdot B^{[q_2]}\beta, i, j, r_1, M_1 \rangle}{\langle B^{[q_2]} \to \cdot \gamma_{q_2}, j, j, r_2, \emptyset \rangle}$ if $r_2 : B \to \langle \ldots, \gamma_{q_2}, \ldots \rangle \in P$, and $\dim(B) = 1$ or $M_1(B)\downarrow$.

$\mathbf{D}_{\mathbf{pred_2}}$: $\frac{\langle A^{[q_1]} \to \alpha \cdot B^{[q_2]}\beta, i, j, r_1, M_1 \rangle, \langle B^{[q]} \to \gamma_{q'}, l, m, r_2, M_2 \rangle}{\langle B^{[q_2]} \to \cdot \gamma_{q_2}, j, j, r_2, M_2 \rangle}$ if $r_2 : B \to \langle \ldots, \gamma_q, \ldots, \gamma_{q_2}, \ldots \rangle \in P$, $\dim(B) > 1$, and $M_1(B) = \langle q, r_2, l, m \rangle$[5].

[4] This set of axioms claims that nothing is derivable from S' before the start of the input string using the fake rule $0 : S' \to S^{[1]}$ and with no category history. This claim is obviously true.

[5] No order is implied between γ_q and γ_{q_2} within production rule r_2.

Discussing each of these deductive steps in turn, we see that D_{scan} for MCFGs is essentially identical to D_{scan} for CFGs; the only difference is that a few extra variables are carried along.

D_{comp_1} is the complete rule for simple categories, and is essentially identical to D_{comp} for CFGs. When completing over complex categories, however, we make use of a new rule D_{comp_2} which does two new things. First, it requires that either (1) category B is present neither in α nor in any of the sequences dominated by components of A linked to $A^{[q_1]}$ and found earlier in the derivation or (2) function M_1 records that the last instance of category B in those places was completed using the same production rule r_2 by which $B^{[q_2]}$ is being completed. Second, it notes in the M function for the resulting item that the last instance of category B was completed by an item that derives $a_{j+1} \ldots a_k$ from component $B^{[q_2]}$ via production rule r_2. In this way we enforce the requirement that derivations for MCFGs replace all linked components of a category using the same rule.

D_{pred_1} is the prediction rule employed either for simple categories or for instances where the category B has not yet been encountered in sequences dominated by categories linked to the parent category A, in other words, in situations where no information is present in the chart about the restrictions on children of components linked to $B^{[q_2]}$. This rule is essentially the same as D_{pred} for CFGs, except that we set the function M for the new item to be undefined for all categories. In situations where information *is* present in the chart about the restrictions on children of components linked to $B^{[q_2]}$ we will be able to find out what item(s) represent the last instance(s) of a component of B linked to $B^{[q_2]}$ dominated by a category linked to $A^{[q_1]}$ by looking in M_1. Matching chart items contain the relevant restriction information, which D_{pred_2} passes on to the new item.

For more discussion of these deductive steps, refer to the proof of correctness below.

5 Proof of Correctness

The following subsets of the set I of possible items for the MCFG Earley Recognition algorithm will be useful in proving the algorithm correct:

- $\mathcal{F}$ is the set of final items (that is, the set of items representing a sucessful recognition).
- $\mathcal{C}$ is the set of correct items (that is, the set of items that represent strings in the language).
- $\mathcal{V}$ is the set of valid items, of which $\langle S'^{[1]} \rightarrow \cdot S^{[1]}, 0, n, 0, \emptyset \rangle$ must be a member, by hypothesis. This set must have the following properties in order to ensure soundness and completeness of the parsing system:
  - Soundness: any items derived from valid items (or hypotheses) must also be valid (soundness).
  - Completeness: any valid item must be able to be derived from other valid items, and those items must not have been derived from the item under consideration, *i.e.,* derivation must not be necessarily circular.

The recognition algorithm is correct iff $\mathcal{C} = \mathcal{V} \cap \mathcal{F}$.

For this parsing system, we define the subsets $\mathcal{F}_{\text{mcfg}}$ and $\mathcal{C}_{\text{mcfg}}$, for an input of length n as follows:

- $\mathcal{F}_{\text{mcfg}} = \{\langle S^{[1]} \rightarrow \alpha\cdot, 0, n, r, M \rangle\}$

- $\mathcal{C}_{\text{mcfg}} = \{\langle S^{[1]} \to \alpha\cdot, 0, n, r, M\rangle | S^{[1]} \Rightarrow_r^* a_1 \ldots a_n\}$

For the purpose of our proof we will define the set $\mathcal{W}$, which is a guess at the definition for $\mathcal{V}_{\text{mcfg}}$:

$$\mathcal{W} = \{\langle A^{[q]} \to \alpha\cdot\beta, i, j, r, M\rangle | \text{Conditions A, B, C, and D hold}\}$$

where the conditions are defined as follows.

Condition A:

$$\exists\gamma,\delta(S^{[1]} \Rightarrow^* \gamma A^{[q]}\delta)$$

In other words, $A^{[q]}$ can be derived from the start symbol in zero or more steps. Variables γ and δ will be used in the other conditions as well.

Condition B:

$$\gamma \Rightarrow^* a_1 \ldots a_i$$

That is, when $A^{[q]}$ is derived from the start symbol, the string ultimately derived from it is situated immediately to the right of $a_1 \ldots a_i$.

Condition C:

$$\alpha \Rightarrow^* a_{i+1} \ldots a_j$$

In other words, the stuff covered so far under $A^{[q]}$ derives the part of the input between a_{i+1} and a_j, inclusive.

To set up for Condition D we will need the following variable definitions:

$$\begin{aligned}
&\text{Let } Q = \{s_1, \ldots s_p\} \text{ for some } p, \text{ where } \forall s_i \in Q, 1 \leq s_i \leq \dim(A), s_i \neq q\\
&\gamma \text{ can be rewritten as } \alpha_1 A^{[s_1]} \alpha_2 \ldots \alpha_{2p-1} A^{[s_p]} \alpha_{2p}\\
&\text{Rule } r = A \to \langle \gamma_1, \ldots, \gamma_{\dim(A)}\rangle\\
&\text{Let } L = \gamma_{s_1} \ldots \gamma_{s_p} \alpha
\end{aligned}$$

In other words, γ contains zero or more components of A linked to $A^{[q]}$, and we refer to all of the stuff dominated by them so far in the derivation tree as the concatenated sequence L.

Condition D:

$$\forall B \in N_c[(M(B)\downarrow \wedge B \notin L) \vee (\exists q_1, r_1, l, m \in \mathbb{N}(\text{Conditions D.1 and D.2 hold}))]$$

D.1: $\langle B, q_1, r_1, l, m\rangle \in M$. That is, M records the fact that $B^{[q_1]}$ is the last component of B in sequence L (to be precise, $\exists\varphi,\vartheta \in V^*(L = \varphi B^{[q_1]}\vartheta \wedge B \notin \vartheta)$), which represents the totality of the sequences covered so far under components linked to $A^{[q]}$. M also records the rule number (r_1) by which $B^{[q_1]}$ was completed and the range of the input covered by $B^{[q_1]}$ (that is, we are assured that $B^{[q_1]} \Rightarrow_{r_1}^* a_{l+1} \ldots a_m$).

D.2: $\alpha_1\gamma_{s_1}\alpha_2\ldots\alpha_{2p-1}\gamma_{s_p}\alpha_{2p}\alpha \Rightarrow^*_{r_1} a_1\ldots a_j$. This says that the first step in the derivation from the string that will remain after applying rule r to $\gamma A^{[q]}$ to the subsequence of the input covered by $\gamma A^{[q]}$ can be r_1.

If we can prove that $\mathcal{W} = \mathcal{V}_{\text{mcfg}}$ (that is, if $\mathcal{W}$ is internally sound and complete), then Condition C is enough to show that $\mathcal{C}_{\text{mcfg}} = \mathcal{V}_{\text{mcfg}} \cap \mathcal{F}_{\text{mcfg}}$ and thus the recognition algorithm is sound and complete.

5.1 Soundness

To prove soundness of the parsing system, we need first to show for any deductive step $\eta_1,\ldots,\eta_n \vdash \xi$, where $\eta_1,\ldots,\eta_n \in \mathcal{W}$, that $\xi \in \mathcal{W}$ as well. For this we will proceed through each type of deduction step.

5.1.1 Scan

For D_{scan}, $\eta_1 = \langle A^{[q]} \rightarrow \alpha\cdot a\beta, i, j, r, M\rangle$ and $\xi = \langle A^{[q]} \rightarrow \alpha a\cdot\beta, i, j+1, M\rangle$. We are given that $\eta_1 \in \mathcal{W}$. ξ differs from η_1 in terms of the definition of variables necessary to declare them members of $\mathcal{W}$ only in that $\alpha_\xi = \alpha_{\eta_1} a$ and $j_\xi = j_{\eta_1} + 1$. This means that Conditions A and B from the definition of $\mathcal{W}$ are unchanged and thus still valid for ξ. The fact that α_ξ is modified from α_{η_1} only by the addition of a terminal element signifies that Condition D must still hold, since it concerns only complex nonterminals. Finally, it can be seen from the definition of $\Rightarrow^*$ that Condition C holds for ξ if it holds for η_1, therefore $\xi \in \mathcal{W}$.

5.1.2 Complete-1

For D_{comp_1}, $\eta_1 = \langle A^{[q_1]} \rightarrow \alpha\cdot B^{[q_2]}\beta, i, j, r_1, M_1\rangle$, $\eta_2 = \langle B^{[q_2]} \rightarrow \gamma\cdot, j, k, r_2, M_2\rangle$ and $\xi = \langle A^{[q_1]} \rightarrow \alpha B^{[q_2]}\cdot \beta, i, k, r_1, M_1\rangle$. We are given that $\eta_1, \eta_2 \in \mathcal{W}$. ξ differs from η_1 in terms of the definitions of variables necessary to declare them members of $\mathcal{W}$ only in that $\alpha_\xi = \alpha_{\eta_1} B^{[q_2]}$ (where B is a simple category) and $j_\xi = k$, whereas $j_{\eta_1} = j$ ($j \leq k$). By the same argument as for D_{scan}, Conditions A and B of the definition of $\mathcal{W}$ are unchanged and thus hold for ξ, and since we haven't added a complex nonterminal to α_ξ, Condition D must still hold as well. At this point it remains only to show that $\alpha B^{[q_2]} \Rightarrow^* a_{i+1}\ldots a_k$ (Condition C). By the fact that $\eta_1 \in \mathcal{W}$ we know that $\alpha \Rightarrow^* a_{i+1}\ldots a_j$, and by $\eta_2 \in \mathcal{W}$ we know that $\gamma \Rightarrow^* a_{j+1}\ldots a_k$ and thus that $B^{[q_2]} \Rightarrow^* a_{j+1}\ldots a_k$. Since B is not a complex category, we can thus conclude from the definition of $\Rightarrow^*$ that $\alpha B^{[q_2]} \Rightarrow^* a_{i+1}\ldots a_k$. All of the conditions of $\mathcal{W}$ are satisfied for ξ, so $\xi \in \mathcal{W}$.

5.1.3 Complete-2

For D_{comp_2}, $\eta_1 = \langle A^{[q_1]} \rightarrow \alpha\cdot B^{[q_2]}\beta, i, j, r_1, M_1\rangle$, $\eta_2 = \langle B^{[q_2]} \rightarrow \gamma\cdot, j, k, r_2, M_2\rangle$ and $\xi = \langle A^{[q_1]} \rightarrow \alpha B^{[q_2]}\cdot \beta, i, k, r_1, M_1 \oplus \langle B^{[q_2]}, r_2, j, k\rangle\rangle$. We are given that $\eta_1, \eta_2 \in \mathcal{W}$. The variables for ξ differ from those of η_1 only in that $\alpha_\xi = \alpha_{\eta_1} B^{[q_2]}$ (where B is a complex category) and $j_\xi = k$, whereas $j_{\eta_1} = j$ ($j \leq k$). Since the variables in Conditions A and B have the same denotation in ξ and η_1, these conditions hold for ξ.

Condition D For discussion of Condition D, it should be noted that $L_\xi = L_{\eta_1} B^{[q_2]}$. Since L_ξ differs from L_{η_1} only by the addition of a member of category B, Condition D must hold with respect to ξ for all other categories, and to prove that Condition D is valid for ξ it remains only to show that it holds for category B. Here there are two cases:

1. If M_{η_1} is not defined for category B (that is, if there is no $\langle B,\cdot,\cdot,\cdot,\cdot\rangle$ element in M_{η_1}), then we are assured by the definition $M_\xi = M_1 \oplus \langle B, q_2, r_2, j, k\rangle$ that the first disjunct of Condition D will be false, and therefore we must prove that for category B conditions D.1 and D.2 hold. Setting $\vartheta_\xi = \varepsilon$($\varepsilon$ being the empty sequence), $l_\xi = j$, $m_\xi = k$, $q_{1\xi} = q_2$, $r_{1\xi} = r_2$, $\varphi_\xi = \gamma_{s_1\xi}\ldots\gamma_{s_p\xi}\alpha$, it should be clear that $\langle B, q_{1\xi}, r_{1\xi}, l_\xi, m_\xi\rangle = \langle B, q_2, r_2, j, k\rangle \in M_\xi$, $L = \varphi_\xi B^{[q_{1\xi}]}\vartheta_\xi$, and $B \notin \vartheta_\xi = \varepsilon$. Since $\eta_2 \in \mathcal{W}$, we know that $\alpha_{\eta_2} \Rightarrow^* a_{i_{\eta_2}+1}\ldots a_{j_{\eta_2}}$, which is equivalent to $\gamma \Rightarrow^* a_{j+1}\ldots a_k$. From this and the definition of η_2 it can be seen that $B^{[q_2]} \Rightarrow^*_{r_2} a_{j+1}\ldots a_k$ and thus all three parts of condition D.1 hold for M_ξ. At this point we need only to prove that $\alpha_1\gamma_{s_1}\alpha_2\ldots\alpha_{2p-1}\gamma_{s_p}\alpha_{2p}\alpha B^{[q_2]} \Rightarrow^*_{r_2} a_1\ldots a_k$. Since M_{η_1} is undefined for B there must be no component symbols of B in $\gamma_{s_1}\ldots\gamma_{s_p}\alpha$[6], that $\alpha_1\gamma_{s_1}\alpha_2\ldots\alpha_{2p-1}\gamma_{s_p}\alpha_{2p}\alpha \Rightarrow^* a_1\ldots a_j$, and that therefore $\alpha_1\gamma_{s_1}\alpha_2\ldots\alpha_{2p-1}\gamma_{s_p}\alpha_{2p}\alpha B^{[q_2]} \Rightarrow^*_{r_2} a_1\ldots a_k$, since we already know that $B^{[q_2]} \Rightarrow^*_{r_2} a_{j+1}\ldots a_k$ and that starting with production r_2 does not affect $\alpha_1\gamma_{s_1}\alpha_2\ldots\alpha_{2p-1}\gamma_{s_p}\alpha_{2p}\alpha$.

2. If M_{η_1} is defined for category B, we are still assured by the definition $M_\xi = M_{\eta_1} \oplus \langle B^{[q_2]}, r_2, j, k\rangle$ that the first disjunct of Condition D will be false. Further, the same arguments as for case (1) apply with respect to condition D.1, so only conditon D.2 remains to be proved. Here we are given by the fact that $\eta_1 \in \mathcal{W}$ that $\alpha_1\gamma_{s_1}\alpha_2\ldots\alpha_{2p-1}\gamma_{s_p}\alpha_{2p}\alpha \Rightarrow^*_{r_2} a_1\ldots a_j$ (from condition D.2, since B is defined in M_{η_1} and must have used r_2 given the definition of D_{comp_2}), and this, combined with the fact that $B^{[q_2]} \Rightarrow^*_{r_2} a_{j+1}\ldots a_k$ (from $\eta_2 \in \mathcal{W}$), using the definition of $\Rightarrow^*$, leads to the inevitable conclusion that D.2 must hold, that is $\alpha_1\gamma_{s_1}\alpha_2\ldots\alpha_{2p-1}\gamma_{s_p}\alpha_{2p}\alpha B^{[q_2]} \Rightarrow^*_{r_2} a_1\ldots a_k$.

Condition C All that remains now is to show that $\alpha B^{[q_2]} \Rightarrow^* a_{i+1}\ldots a_k$. We know from the fact that $\eta_1 \in \mathcal{W}$ that $\alpha \Rightarrow^*_{r_2} a_{i+1}\ldots a_j$ (see also proof of Condition D) and from the fact that $\eta_2 \in \mathcal{W}$ that $B^{[q_2]} \Rightarrow^*_{r2} a_{j+1}\ldots a_k$. These facts combine to show that $\alpha B^{[q_2]} \Rightarrow^*_{r2} a_{i+1}\ldots a_k$, which is a stronger conclusion.

5.1.4 Predict-1

For D_{pred_1}, $\eta_1 = \langle A^{[q_1]} \to \alpha\cdot B^{[q_2]}\beta, i, j, r_1, M_1\rangle$ and $\xi = \langle B^{[q_2]} \to \cdot\gamma, j, j, r_2, \emptyset\rangle$. Here the difference in $\mathcal{W}$ variables between η_1 and ξ is pretty large. From η_1 we can say $S^{[1]} \Rightarrow^* \gamma_{\eta_1}A^{[q_1]}\delta_{\eta_1}$ and then that $\gamma_{\eta_1}A^{[q_1]}\delta_{\eta_1} \Rightarrow_{r_1} \gamma'_{\eta_1}\alpha B^{[q_2]}\beta\delta'_{\eta_1}$, so $\gamma_\xi = \gamma'_{\eta_1}\alpha$ and $\delta_\xi = \beta\delta'_{\eta_1}$, and we have proved Condition A for ξ. Moving on to Condition B, we can see that from conditions B,C and D of η_1 that $\gamma'_{\eta_1}\alpha \Rightarrow^* a_1\ldots a_j$, which is equivalent to $\gamma_\xi \Rightarrow^* a_1\ldots a_{i_\xi}$. For Condition C we can see that $\alpha_\xi = \varepsilon$, so the condition applies vacuously. Finally, since $M_\xi = \emptyset$ and $Q_\xi = \varepsilon$ (thus $L_\xi = \varepsilon$ as well), the first disjunct applies in Condition D for all complex categories, so therefore Condition D is true of ξ as well.

5.1.5 Predict-2

For D_{pred_2}, $\eta_1 = \langle A^{[q_1]} \to \alpha\cdot B^{[q_2]}\beta, i, j, r_1, M_1\rangle$, $\eta_2 = \langle B^{[q]} \to \delta\cdot, l, m, r_2, M_2\rangle$, and $\xi = \langle B^{[q_2]} \to \cdot\gamma, j, j, r_2, M_2\rangle$. For Condition A the argument is the same as for D_{pred_1}: from η_1 we can say $S^{[1]} \Rightarrow^* \gamma_{\eta_1}A^{[q_1]}\delta_{\eta_1}$ and then that $\gamma_{\eta_1}A^{[q_1]}\delta_{\eta_1} \Rightarrow_{r_1} \gamma'_{\eta_1}\alpha B^{[q_2]}\beta\delta'_{\eta_1}$, so $\gamma_\xi = \gamma'_{\eta_1}\alpha$ and $\delta_\xi = \beta\delta'_{\eta_1}$. For Condition B the argument is the same as well: we can see that from conditions B, C, and D of η_1 that $\gamma'_{\eta_1}\alpha \Rightarrow^* a_1\ldots a_j$, which is equivalent to $\gamma_\xi \Rightarrow^* a_1\ldots a_{i_\xi}$.

[6] There can be B components in $\alpha_1\ldots\alpha_{2p}$ because these are not children of the components linked to $A^{[q_1]}$ and therefore there is no same-rule linking requirement imposed by the definition of rewrite for MCFGs.

Condition C applies vacuously here as well. Finally we need to prove Condition D. Here we look at M_2. From the definition of D_{pred_2} and from the fact that $\eta_2 \in \mathcal{W}$, we can see that $L_{\eta_2} = L_\xi$, so the fact that Condition D applies for η_2 implies that it applies for ξ as well, since the first disjunct and D.1 will be unchanged for both of them, and D.2 differs only in the makeup of the term $\alpha_{2m}\alpha$, which by definition doesn't mention category B, therefore that condition applies in both cases.

5.1.6 Conclusion

We have shown that all of the deductive steps are sound, so $\mathcal{W}$ is a sound subset of items. We now need to show that it is complete as well.

5.2 Completeness

For a completeness proof, we need to show that any valid item was derived from valid items and/or hypotheses. This proof has two parts, conceptually: we need to show that for each derivation step $\eta_1, \ldots, \eta_n \vdash \xi$, if ξ is valid ($\xi \in \mathcal{W}$), then $\eta_1, \ldots, \eta_n$ are valid as well ($\eta_1, \ldots, \eta_n \in \mathcal{W}$); we also need to show that $\eta_1, \ldots, \eta_n$ can be derived before ξ. For this second part we need to define a *derivation length function* $d : I \to \mathbb{N}$ that mirrors the length of derivations. Upon examination of the derivation steps proposed here, it becomes clear that these steps mirror the derivation steps for the Earley algorithm for CFGs, and thus follow a CFG derivation, not necessarily an MCFG derivation. Therefore we will have to have some definitions to bridge the gap:

> The *CFG-equivalent rewrite relation for MCFGs* $\mapsto$ is defined as follows: Given an MCFG G and its corresponding superset CFG $\text{cfg}(G)$, $\alpha \mapsto_G \beta$ iff there exists some derivation $S^{[1]} \Rightarrow_G^* \theta$ where θ is a string of terminal elements and the derivation tree defined by $\alpha \Rightarrow_{\text{cfg}(G)}^* \beta$ is a *CFG subset tree* of the derivation tree defined by $S^{[1]} \Rightarrow_G^* \theta$. We write $\alpha \mapsto^n \beta$ if $\alpha \Rightarrow_{\text{cfg}(G)}^n \beta$.

We then define the derivation length function d as follows:

$$d(\langle A^{[q]} \to \alpha \cdot \beta, i, j, r, M\rangle) = min\{\pi + 2\lambda + 2\mu + j \mid \begin{array}{l} S^{[1]} \mapsto^\pi \gamma A^{[q]} \delta \\ \gamma \mapsto^\lambda a_1 \ldots a_i \\ \alpha \mapsto^\mu a_{i+1} \ldots a_j \end{array}\}$$

Now we just have to show for each deductive step $\eta_1, \ldots, \eta_m \vdash \xi$ that $\xi \in \mathcal{W}$ implies that $\eta_1, \ldots, \eta_m \in \mathcal{W}$ and that $\forall i, 1 \leq i \leq m, d(\eta_i) < d(\xi)$.

5.2.1 Scan

For D_{scan}, $\eta_1 = \langle A^{[q]} \to \alpha \cdot a\beta, i, j, r, M\rangle$, and $\xi = \langle A^{[q]} \to \alpha a \cdot \beta, i, j+1, r, M\rangle$. Since the only difference between η_1 and ξ is the position of a before or after the dot, the definition for membership in $\mathcal{W}$ must hold for η_1 if it does for ξ, and it should be clear that $\pi_{\eta_1} = \pi_\xi$, $\lambda_{\eta_1} = \lambda_\xi$, and $\mu_{\eta_1} = \mu_\xi$, whereas $j_{\eta_1} = j_\xi - 1$, so therefore $d(\eta_1) = d(\xi) - 1$, and thus $d(\eta_1) < d(\xi)$.

5.2.2 Complete-1

For D_{comp_1}, $\eta_1 = \langle A^{[q_1]} \to \alpha \cdot B^{[q_2]}\beta, i, j, r_1, M_1\rangle$, $\eta_2 = \langle B^{[q_2]} \to \gamma\cdot, j, k, r_2, M_2\rangle$, and $\xi = \langle A^{[q_1]} \to \alpha B^{[q_2]} \cdot \beta, i, k, r_1, M_1\rangle$. Here we can see that η_1 differs from ξ only by the position of a simple nonterminal before or after the dot, so the argument of section 5.1.2 that $\eta_1 \in \mathcal{W}$ implies $\xi \in \mathcal{W}$ applies in reverse. The item η_2 is a bit more complicated because we have to specify the contents of M_2 and choose r_2. Since $\xi \in \mathcal{W}$, there must be a derivation from $\alpha B^{[q_2]}$ to $a_{i+1}\dots a_k$, which implies that there must be some derivation from $B^{[q_2]}$ to $a_{j+1}\dots a_k$ via some rule r_2. Looking at this derivation we can form M_2 as follows: for all complex categories C mentioned in γ there must be an component number $q \in \mathbb{N}$ such that $\gamma = \gamma' C^{[q]} \delta$ where category C is not mentioned in δ, and further there must be a tuple $\langle C^{[q]}, r, l, m\rangle$ such that $C^{[q]} \Rightarrow_r^* a_{l+1}\dots a_m$; M_2 is the smallest set that contains all such tuples. If r_2 and M_2 are chosen in this way, it should be easy to see that $\eta_2 \in \mathcal{W}$. Finally, we can see that $\pi_\xi = \pi_{\eta_1} = \pi_{\eta_2} - 1$, $\lambda_\xi = \lambda_{\eta_1}$, $\lambda_{\eta_2} = \lambda_{\eta_1} + \mu_{\eta_1}$, and $\mu_\xi = \mu_{\eta_1} + \mu_{\eta_2} + 1$. In terms of η_1, $d(\xi) = \pi_{\eta_1} + 2\lambda_{\eta_1} + 2(\mu_{\eta_1} + \mu_{\eta_2} + 1) + k$, which is clearly greater than $d(\eta_1) = \pi_{\eta_1} + 2\lambda_{\eta_1} + 2\mu_{\eta_1} + j$. In terms of η_2, $d(\xi) = \pi_{\eta_2} - 1 + 2(\lambda_{\eta_2} - \mu_{\eta_1}) + 2(\mu_{\eta_1} + \mu_{\eta_2} + 1) + k = \pi_{\eta_2} + 2\lambda_{\eta_2} + 2\mu_{\eta_2} + k + 1 = d(\eta_2) + 1$, so $d(\xi) > d(\eta_2)$.

5.2.3 Complete-2

For D_{comp_2}, $\eta_1 = \langle A^{[q_1]} \to \alpha \cdot B^{[q_2]}\beta, i, j, r_1, M_1\rangle$, $\eta_2 = \langle B^{[q_2]} \to \gamma\cdot, j, k, r_2, M_2\rangle$, and $\xi = \langle A^{[q_1]} \to \alpha B^{[q_2]} \cdot \beta, i, k, r_1, M_1 \oplus \langle B^{[q_2]}, r_2, j, k\rangle\rangle$. Since $\xi \in \mathcal{W}$, there must be a derivation $\alpha_1\gamma_{s_1}\alpha_2\dots\alpha_{2p-1}\gamma_{s_p}\alpha_{2p}\alpha B^{[q_2]} \Rightarrow_{r_2}^* a_1\dots a_k$ and a derivation $\alpha_1\gamma_{s_1}\alpha_2\dots\alpha_{2p-1}\gamma_{s_p}\alpha_{2p}\alpha B^{[q_2]} \Rightarrow_r^* a_1\dots a_k$ for every $\langle C^{[q]}, r, l, m\rangle$ in M_1. These two facts are sufficient to guarantee that there are also derivations $\alpha_1\gamma_{s_1}\alpha_2\dots\alpha_{2p-1}\gamma_{s_p}\alpha_{2p}\alpha \Rightarrow_{r_2}^* a_1\dots a_j$ if $\exists q, l, m(\langle B^{[q]}, r_2, l, m\rangle \in M_1)$ and certainly $\alpha_1\gamma_{s_1}\alpha_2\dots\alpha_{2p-1}\gamma_{s_p}\alpha_{2p}\alpha \Rightarrow_r^* a_1\dots a_j$ derivations for every $\langle C^{[q]}, r, l, m\rangle$ in M_1. These fact, plus the fact that $\gamma_\xi = \gamma_{\eta_1}$ and $\delta_\xi = \delta_{\eta_1}$ are enough to show that necessarily $\eta_1 \in \mathcal{W}$. Furthermore, the existence of these derivations is sufficient to reconstruct M_2 such that $\eta_2 \in \mathcal{W}$. As for the derivation length functions for D_{comp_2}, these are identical to the functions for D_{comp_1}, so it is clear that $d(\eta_1) < d(\xi)$ and $d(\eta_2) < d(\xi)$.

5.2.4 Predict-1

For D_{pred_1}, $\eta_1 = \langle A^{[q_1]} \to \alpha \cdot B^{[q_2]}\beta, i, j, r_1, M_1\rangle$ and $\xi = \langle B^{[q_2]} \to \cdot\gamma, j, j, r_2, \emptyset\rangle$. Here the arguments are much the same as for all the above. The fact that $\xi \in \mathcal{W}$ implies the existence of derivations upon which the components of η_1 can be built to ensure that $\eta_1 \in \mathcal{W}$. As for the derivation length functions, we can define the constants for ξ in terms of η_1 as follows: $\pi_\xi = \pi_{\eta_1} + 1$, $\lambda_\xi = \lambda_{\eta_1} + \mu_{\eta_1}$, $\mu_\xi = 0$, and $j_\xi = j_{\eta_1} = j$, so $d(\xi) = d(\eta_1) + 1$, thus $d(\eta_1) < d(\xi)$.

5.2.5 Predict-2

For D_{pred_2}, $\eta_1 = \langle A^{[q_1]} \to \alpha \cdot B^{[q_2]}\beta, i, j, r_1, M_1\rangle$, $\eta_2 = \langle B^{[q]} \to \delta\cdot, l, m, r_2, M_2\rangle$, and $\xi = \langle B^{[q_2]} \to \cdot\gamma, j, j, r_2, M_2\rangle$. Here the arguments are much the same as for all the above. The fact that $\xi \in \mathcal{W}$ implies the existence of derivations upon which the components of η_1 and η_2 can be built to ensure that $\eta_1, \eta_2 \in \mathcal{W}$. The derivation length argument for η_1 is the same as before. For η_2 we can say that $\pi_{\eta_2} \leq \pi_\xi$ and that $\lambda_{\eta_2} + \mu_{\eta_2} < \lambda_\xi + \mu_\xi$, since $\mu_\xi = 0$ and the steps for λ_ξ must necessarily include all of the steps counted in λ_{η_2} and μ_{η_2} plus the step from $B^{[q]}$ to δ. Since $j_{\eta_2} \leq j_\xi$, this means $d(\eta_2) < d(\xi)$.

6 Complexity

For a given MCFG $\langle N, \Sigma, P, S \rangle$ the number of items is polynomially bounded by the length n of the input string. All items are of the form $\langle A^{[q]} \rightarrow \alpha \cdot \beta, i, j, r, M \rangle$. The number of possible $A^{[q]} \rightarrow \alpha \cdot \beta, r$ combinations is bounded by the grammar itself at $c \cdot r \cdot d \cdot l$ where c is the number of categories in the grammar, r is the maximum number of rules headed by a category, d is the maximum dimension of the grammar, and l is the maximum length of a component of the right hand side of a rule. Leaving aside M, then, the number of possible $\langle A^{[q]} \rightarrow \alpha \cdot \beta, i, j, r \rangle$ combinations is $crdln^2$, which is $O(n^2)$. Looking at M, we can see that it has at most c' tuples $\langle B, q, r, l, m \rangle$ where $c\prime$ is the maximum number of complex categories mentioned in the right hand side of a rule. In the worst case the number of possibilities for each slot in M is bounded by $r\prime \cdot d \cdot n^2$, where r' is the maximum number of rules headed by a complex category. Therefore, we can bound the number of items at $crdln^2 r'^{c'} d^{c'} n^{2c'}$, which is $O(n^{2c'+2})$. The situation is actually somewhat better than this, however, as there are limitations on the possible l, m values within M. With no loss of generality we can restrict ourselves to grammars in which no category found on the right hand side of a rule rewrites to ε. In such grammars the following restrictions exist on the l, m pairs:

1. For all such pairs $l < m$.
2. For all such pairs $l, m \leq j$.
3. For no two pairs l_1, m_1 and l_2, m_2 may the intervals (l_1, m_1), (l_2, m_2) overlap.

Given these restrictions the choice of m values reverts to an ordered selection without repetition of c' values from a set of n (actually j) possible values, and the choice of l values is the same. From combinatorics we can therefore see that the overall number of possible items is thus $O(n^2 \left(\frac{n!}{(n-c')!}\right)^2)$, which is a significant improvement.

As for the time complexity of the recognizer, step 2.b of the deduction procedure specified in section 4.1 requires every item on the agenda to be compared with items in the chart. Since the number of items in the chart is $O(n^{2c'+2})$ (ignoring the somewhat messier improved result), we can employ a binary search to complete this step in $O(\log_2 n^{2c'+2})$ operations per item on the agenda. For any item on the agenda but not already in the chart, step 2.c of the deduction procedure checks whether any rule of inference will apply. Looking at the inference rules it can be seen that a limited binary search can be employed to complete this step in $O(n^{2c'+1} \log_2 n)$ checks per item on the agenda. Given an item on the agenda and an item in the chart, actually verifying whether an inference rule applies takes constant time. Since step 2.a takes constant time and the steps 2.a–2.c are taken in sequence, the overall time complexity of step 2 is $O(n^{2c'+1} \log_2 n)$ per item on the agenda. Steps 1 and 3 do not exceed this bound. The number of items that will be put on the agenda while recognizing a string is $O(n^{2c'+2})$. This is the upper bound on the number of possible items. There will be duplicates in the agenda, but their number is finite and does not depend on n, essentially because the number of axioms and the number of inference rules is finite and all items in the chart are unique. Thus, the overall time complexity of the recognizer is $O(n^{4c'+3} \log_2 n)$ or, with the refinement from the previous paragraph, $O(n^3 \left(\frac{n!}{(n-c')!}\right)^4 \log_2 n)$, which is somewhat better.

An interesting fact about this algorithm is that, in contrast with many recognizers proposed for MCFGs (Seki *et al.* 1991; Nakanishi *et al.* 2000), the dimension of the grammar is not present in the exponent of the polynomial describing its complexity.

References

CHOMSKY, NOAM. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.

EARLEY, JAY. 1970. An efficient context-free parsing algorithm. *Comm. of the ACM* 6.451–455.

NAKANISHI, R., K. TAKADA, and H. SEKI. 2000. An efficient recognition algorithm for multiple context-free languages. In *Proceedings of the Fifth Meeting on the Mathematics of Language*, 119–123, Saarbrücken, Germany.

SEKI, HIROYUKI, TAKASHI MASUMURA, MAMORU FUJII, and TADAO KASAMI. 1991. On multiple context-free grammars. *Theoretical Computer Science* 88.

SHIEBER, STUART M., YVES SHABES, and FERNANDO C. N. PEREIRA. 1995. Principles and implementation of deductive parsing. *Journal of Logic Programming* 24.

SIKKEL, KLAAS. 1998. Parsing schemata and correctness of parsing algorithms. *Theoretical Computer Science* 199.87–103.

STABLER, EDWARD P. 1997. Derivational minimalism. In *Logical Aspects of Computational Linguistics*, ed. by Christian Retoré, number 1328 in Lecture Notes in Computer Science, 68–95. NY: Springer-Verlag.