Formalizing Mirror Theory*

Gregory M. Kobele

1 Introduction

The primary empirical motivation for mirror theory is the linguistic phenomenon known as the mirror principle, which is summarized in (Baker, 1988):

> "...the characteristic morphology of GF [grammatical function - e.g. subject, object, etc (GMK)] changing processes always appears on the main verb in an order that exactly represents the order that those GF changing processes seem to have applied in." ((Baker, 1988), p. 363)

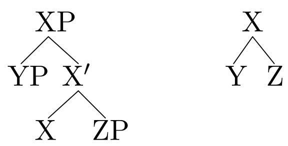
In his account of the mirror principle, Baker (1988) relies crucially on head (X^0) movement, to which he (compellingly) argues that the GF changing processes can be reduced. This move, along with the head movement constraint (Travis, 1984), allow Baker to account for the mirror principle. Head movement however, seems to have very different properties from phrasal (XP) movement. For example, head movement, but not phrasal movement, is a one-time deal: each head can move at most once (as excorporation is disallowed). Moreover, the domain of locality of heads seems much stricter than that of phrases (a head can move only to the next higher head position) (see (Brody, 1997) esp. chap. 3 for more and detailed examples of the differences between head and phrasal movement).

Mirror theory accounts both for the mirror principle, and for the difference between phrasal and head movement by its requirement that the

*This paper has appeared in **Grammars 5(3), 2002** in a slightly modified form. Citations should, when necessary, be made to the official version of this paper. Comments are welcome!

syntactic head-complement relation mirror certain morphological relations (such as suffix order). This requirement constrains the types of syntactic structures that can express a given phrase; the morphological constituency of the phrase determines part of the syntactic constituency, thereby ruling out other, weakly equivalent, alternatives.

A less fundamental, but superficially very noticeable feature of this theory is the elimination of phrasal projection. Thus the X-bar structure on the left becomes the mirror theoretic structure on the right:



(Brody calls this systematic collapse of X, X′ and XP nodes 'telescope'.) Every node may now have phonetic content, and children are identified as specifiers or complements depending on their direction of branching; left-daughters are specifiers and right-daughters are complements (previously, specifiers were children of XP, and complements were children of X′). Furthermore, the complement relation is a "word-forming" relation, where according to the "mirroring" relation, the phonetic content of each head is a suffix on the phonetic content of its complement.

This paper can be seen as making two contributions. The first, linguistic, one, is that it provides a formalization of Brody's mirror theory, and answers questions about the applicability of that theory to natural language such as whether mirror theory is strong enough to circumscribe the class of natural languages, and whether it overgenerates excessively. The second, formal, contribution, is as a case study in adding a certain kind of feature movement to the framework of minimalist grammars (Stabler, 1997; Stabler and Keenan, 2000).

2 Mathematical Preliminaries

Let S be a set, and n a natural number. Then S^n denotes the set of sequences of length n over S. $\langle S^n \rangle^m$ is the set of sequences of length m of sequences of length n over S. Define $S^? = S^0 \cup S^1$, $S^* = \bigcup_{n=0}^{\infty} S^n$, and $S^+ = S^* - \{\epsilon\}$, where ϵ is the sequence of length 0. Given sets of sequences S and T, $ST = \{st | s \in S \wedge t \in T\}$.

Fix a domain, D. A relation $R \subseteq D \times D$ over D is reflexive iff xRx for all $x \in D$. It is symmetric iff $R = R^{-1}$, where $xR^{-1}y$ iff yRx. R is antisymmetric iff $R \cap R^{-1} \subseteq \Delta$, where Δ denotes the diagonal relation over D ($x\Delta y$ iff $x = y$), and R is asymmetric iff $R \cap R^{-1} = \emptyset$. Given $R_1 \subseteq A \times B$, $R_2 \subseteq B \times C$, $R_2 \circ R_1 \subseteq A \times C$ denotes their composition, and $x(R_2 \circ R_1)y$ iff $\exists z\ xR_1z$ and zR_2y. Writing R^0 for Δ, $R^{n+1} = R^n \circ R$. $R^+ = \bigcup_{n=1}^{\infty} R^n$ is the transitive closure of R, and $R^* = R^0 \cup R^+$ is R's reflexive transitive closure. R is transitive when $R^+ \subseteq R$, and connected when every two distinct elements in D are related by R ($\forall x, y \in D\ x \neq y \rightarrow xRy \vee yRx$). A partial order is a binary relation that is transitive and asymmetric (in which case it is strict), or transitive, reflexive and antisymmetric (in which case it is weak). A total order is a connected partial order.

3 Mirror Theoretic Grammars

In this section I present two equivalent derivational descriptions of Mirror Theory. [1] A tree-based formalism is introduced first, as it bears obvious similarities to the existing expositions of Mirror Theory, and is thus hoped to be more readily accessible. The (formally simpler) chain-based formalism adopted in the remainder of this paper is then motivated and explained.

[1]Another motivation for mirror theory is the elimination of redundancies in the architecture of current popular generative syntactic theories. Brody (2000) (and elsewhere) argues that representational approaches to syntactic representations are to be preferred on these grounds. I view this debate as independant of the grammatical architecture of mirror theory, and study this formalism from a derivational perspective in order to facilitate comparison with other (and derivational) formalisms on the market. My reasons for this include the belief that Brody does not want to capture significantly different relations or structures than those described derivationally (or, perhaps more correctly, than those described derivationally and 'weakly' representationally with additional stipulations to ensure that 'bleeding' does not occur(see (Brody, 2000))), but instead is making a stand as to how the relevant grammatical knowledge of these structures is represented in the mind:

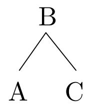
> "But it seems mistaken to conclude from the assumption that, say, move captures the properties of chain, that both chain and move are part of the grammar." (Brody, 2000)

This purely expository choice should not be taken as a stand on the issue of representations vs. derivations.

3.1 Trees

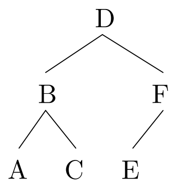
Trees in mirror theory are ordered by the standard dominance relation ('$\lhd$'), as well as by a strict total ordering called (left corner) precedence ('$\prec\!\prec$'). A child is a left child just in case it precedes its parent, and a right child otherwise. $\prec\!\prec$ and $\lhd$ are related by the requirement that all the decendants of a left (right) child precede (follow) that child's parent. If a node x dominates a node z, and if x precedes (follows) z, then all right (left) children of x dominate z. This forces our trees to be at most binary branching. Note that if a node A dominates only a single node B, B is regarded as a specifier of A or a complement of A depending on whether $B \prec\!\prec A$ or $A \prec\!\prec B$.

Example 1



Here A $\prec\!\prec$ B $\prec\!\prec$ C.

Example 2



Here, each node stands in the left corner precedence relation with every other node it is alphebetically less than (the nodes are labeled according to their recognition order by a left corner parse, with 'A' being recognized before 'B' etc. This is also the order in which the nodes are traversed in an inorder tree traversal. (See (Gerdemann, 1994))).

Definition 1 A **Mirror Theoretic Tree**(MTT) is a tuple $\tau = \langle N_\tau, S_\tau, \lhd_\tau, \prec\!\prec_\tau \rangle$[2] (omitting subscripts when τ is understood) such that

[2] There is an isomorphism between MTTs as defined above and subsets $T \subseteq \{0,1\}^*$ together with a distinguished subset $S \subseteq T$ such that $\forall u, v \; uv \in T \rightarrow u \in T$. This isomorphism is defined so that $hx \lhd^* hy$ iff $y = xz$, for some z. $hx \prec\!\prec hy$ iff $x = u0v$ and $y = u1w$, $x = y0v$, or $y = x1w$. $x \in S \leftrightarrow h(x) \in S$. See (Kobele et al., 2002) for a presentation in these terms.

1. N is a finite and non-empty set (of 'nodes'),
2. $S \subseteq N$,
3. $\lhd \subseteq N \times N$, s.t. $\lhd^+$ is a strict partial order,
4. $\prec\!\!\prec \subseteq N \times N$ is a strict total order,
5. the relations $\prec\!\!\prec$ and $\lhd$ satisfy the following axioms over N:

   Ax1 $\exists x \forall y \; x \lhd^* y$

   Ax2 $\forall xyz \; x \lhd^* z \wedge y \lhd^* z \rightarrow x \lhd^* y \vee y \lhd^* x$

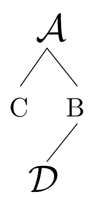
   Ax3 $\forall xy \; x \lhd y \wedge y \prec\!\!\prec x \rightarrow (\forall z \; y \lhd^* z \leftrightarrow (x \lhd^* z \wedge z \prec\!\!\prec x))$

   Ax4 $\forall xy \; x \lhd y \wedge x \prec\!\!\prec y \rightarrow (\forall z \; y \lhd^* z \leftrightarrow (x \lhd^* z \wedge x \prec\!\!\prec z))$

The set of nodes (N) has a unique least item with respect to dominance ($\lhd^*$) (**Ax1**), which is denoted $root_\tau$, and there is a unique domination path from the root to each node (**Ax2**, 'linear branching'). Left corner precedence is related to dominance in the manner outlined above; a parent's left (right) child only dominates nodes that precede (follow) its parent (left to right direction of the biconditional in **Ax3,4**), and if a node x dominates and precedes (follows) a node z, then all of x's right (left) children dominate z as well (right to left direction of the biconditional in **Ax3,4**).

For the sake of perspicuity, MTTs will be often depicted graphically rather than as tuples. The nodes in S (the strong nodes) will be drawn in cursive script (see fig. 1).

Figure 1:



The tree in figure 1 is the graphical depiction of $\tau = \langle \{A, B, C, D\}, \{A, D\}, \lhd_\tau, \prec\!\!\prec_\tau \rangle$, where $\lhd_\tau = \{\langle A, B\rangle, \langle A, C\rangle, \langle B, D\rangle\}$, and $C \prec\!\!\prec_\tau A \prec\!\!\prec_\tau D \prec\!\!\prec_\tau B$

Definition 2 Given τ, let $SPEC_\tau = \{\langle x, y\rangle | x \prec\!\!\prec y \wedge y \lhd x\}$ and $COMP_\tau = \{\langle x, y\rangle | y \prec\!\!\prec x \wedge y \lhd x\}$.

$aSPECb$ ($aCOMPb$) is read "a is a specifier (complement) of b."

Proposition 1 guarantees that a node has at most one specifier, and at most one complement, and thus that a MTT is at most binary branching. This, along with **Ax2**, ensures that $SPEC$ and $COMP$ are functional in both arguments, i.e. $SPEC$, $SPEC^{-1}$, $COMP$, and $COMP^{-1}$ are all (partial) functions over N.

Proposition 1
For any x, $|\{y|ySPECx\}| \leq 1$, and $|\{y|yCOMPx\}| \leq 1$.

Proof: I show $|\{y|ySPECx\}| \leq 1$. The proof that $|\{y|yCOMPx\}| \leq 1$ is similar. Let $a, b, c \in N$ such that $aSPECc$ and $bSPECc$. I show $a = b$:

1. $a \prec\!\!\prec c \wedge c \lhd a$ (by definition 2)
2. $b \prec\!\!\prec c \wedge c \lhd b$ (by definition 2)
3. $\forall z\ a \lhd^* z \leftrightarrow (c \lhd^* z \wedge z \prec\!\!\prec c)$ (from 1 by **Ax3**)
4. $c \lhd^* b \wedge b \prec\!\!\prec c$ (from 2)
5. $a \lhd^* b$ (from 3 and 4)
6. $\forall z\ b \lhd^* z \leftrightarrow (c \lhd^* z \wedge z \prec\!\!\prec c)$ (from 2 by **Ax3**)
7. $c \lhd^* a \wedge a \prec\!\!\prec c$ (from 1)
8. $b \lhd^* a$ (from 6 and 7)
9. $a = b$ (from 5 and 8 by the antisymmetry of $\lhd^*$)

$\square$

Given a set L of labels, an *expression* is defined to be a MTT together with a function $\mu : N \rightarrow L$, assigning to each $n \in N$ an element of L. Exp is the set of expressions.

Definition 3 (**Tree-based Mirror Theoretic Grammar**) Following (Stabler, 1997), a Tree-based Mirror Theoretic Grammar(MTG) $G = \langle V, Cat, Lex, \mathcal{F}\rangle$ is a bare grammar[3], where

1. V is a non-empty set (the pronounced elements)
2. Cat is the disjoint union of the following sets:
   - $base$, a non-empty finite set.
   - $cselect = \{\texttt{=}b | b \in base\}$
   - $sselect = \{b\texttt{=}\ | b \in base\}$
   - $licensees = \{\texttt{-}b | b \in base\}$
   - $licensors = \{\texttt{+}b | b \in base\}$

   A label is an element of V^*Cat^*.
3. Lex is a finite set of expressions $\langle \tau, \mu\rangle$, such that
   (a) $|N_\tau| = 1$, and
   (b) $\mu : N_\tau \to V^*\ cselect^?\ (sselect \cup licensors)^?\ base\ licensees^*$
4. $\mathcal{F} = \{merge, move\}$, where
   - $merge : Exp \times Exp \to Exp$ is the union of the following two functions:
     For $\alpha, \beta \in V^*$, $b \in base$, $\delta, \gamma \in Cat^*$,

     smerge
     $(E, E') \in dom(smerge)$ just in case
     (a) The specifier position of the root of τ, the tree component of E, is vacant. That is, $\forall x \in N_\tau\ x = root_\tau \vee root_\tau \prec\!\!\prec x$.

[3]A bare grammar in the sense of (Keenan and Stabler, forthcoming) is a pair $G = \langle Lex, \mathcal{F}\rangle$ where Lex is a set of generators (called lexical items) and $\mathcal{F}$ is a set of functions. The language generated by a bare grammar ($L(G)$)is the closure of the lexical items under $\mathcal{F}$. Alternatively, $\langle L(G), \mathcal{F}\rangle$ is a universal algebra generated by Lex. Sometimes elements of the language (called expressions) have some internal structure. To draw attention to this, sometimes the bare grammar $\langle Lex, \mathcal{F}\rangle$ is rewritten as $\langle X_1, \ldots, X_n, Lex, \mathcal{F}\rangle$, where the X_i's somehow capture the internal structure of the expressions.

(b) The label of the root of τ ($\mu(root_\tau)$) begins with an s-selection feature, and the label of the root of τ' (the tree component of E') begins with a base feature of the same type. That is, $\mu(root_\tau) = \alpha\ \mathsf{b{=}}\ \delta$, and $\mu'(root_{\tau'}) = \beta\ b\ \gamma$.

Asume that N_τ and $N_{\tau'}$ are disjoint. $smerge(E, E') = E''$, where $E'' = \langle \tau'', \mu'' \rangle$. τ'' is the tree formed by placing τ' in the specifier position of the root of τ. Formally,

$N_{\tau''} = N_\tau \cup N_{\tau'}$,

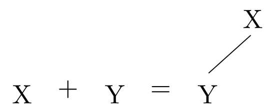
$S_{\tau''} = S_\tau \cup S_{\tau'}$,

$\lhd_{\tau''} = \lhd_\tau \cup \lhd_{\tau'} \cup \{\langle root_\tau, root_{\tau'} \rangle\}$,

$\prec\!\!\prec_{\tau''} = \prec\!\!\prec_\tau \cup \prec\!\!\prec_{\tau'} \cup (N_{\tau'} \times N_\tau)$, and

$\mu''(root_\tau) = \alpha\ \delta$, $\mu''(root_{\tau'}) = \beta\ \gamma$, and for $n \in N_\tau$, $\mu''(n) = \mu(n)$, for $n \in N_{\tau'}$, $\mu''(n) = \mu'(n)$.

smerge behaves as schematized below on the tree components of its arguments:



cmerge

$(E, E') \in dom(cmerge)$ just in case

(a) The complement position of the root of τ, the tree component of E, is vacant. That is, $\forall x \in N_\tau\ x = root_\tau \vee x \prec\!\!\prec root_\tau$.

(b) The label of the root of τ ($\mu(root_\tau)$) begins with a c-selection feature, and the label of the root of τ' (the tree component of E') begins with a base feature of the same type. That is, $\mu(root_\tau) = \alpha\ \mathsf{{=}b}\ \delta$, and $\mu'(root_{\tau'}) = \beta\ b\ \gamma$.

Assume that N_τ and $N_{\tau'}$ are disjoint. $cmerge(E, E') = E''$. τ'' is the tree formed by placing τ' in the complement position of the root of τ. Formally,

$N_{\tau''} = N_\tau \cup N_{\tau'}$,

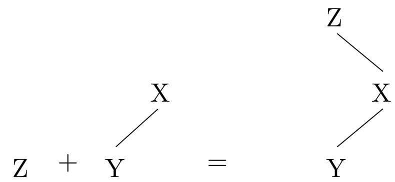
$S_{\tau''} = S_\tau \cup S_{\tau'}$,

$\lhd_{\tau''} = \lhd_\tau \cup \lhd_{\tau'} \cup \{\langle root_\tau, root_{\tau'} \rangle\}$,

$\prec\!\!\prec_{\tau''} = \prec\!\!\prec_\tau \cup \prec\!\!\prec_{\tau'} \cup (N_\tau \times N_{\tau'})$, and

$\mu''(root_\tau) = \alpha\ \delta\ \gamma$, $\mu''(root_{\tau'}) = \beta$, and for $n \in N_\tau$, $\mu''(n) = \mu(n)$, for $n \in N_{\tau'}$, $\mu''(n) = \mu'(n)$.

The behaviour of *cmerge* on the tree component of its arguments is illustrated below:



- *move* : *Exp* $\rightarrow$ *Exp* is, for $\alpha, \beta, \eta \in V^*$, $b \in base$, and $\delta, \gamma, \sigma \in Cat^*$, the following function:

move

$(E) \in dom(move)$ just in case

(a) The specifier position of the root of τ, the tree component of E, is not occupied.

(b) The label of the root of τ ($\mu(root_\tau)$) begins with a licensor feature, and there is exactly one proper subtree τ' of τ such that

- the root of τ' is the specifier of some node in τ, and
- the label of the root of τ' begins with a licensee feature of the same type.

That is, $\mu(root_\tau) = \alpha\ \mathtt{+}b\ \delta$, $\mu(root_{\tau'}) = \beta\ \mathtt{-}b\ \gamma$, and $root_{\tau'}$ is the unique $n \in \{w | \exists v\ wSPEC_\tau v\}$ such that $\mu(n) = \eta\ \mathtt{-}b\ \sigma$.

$move(E) = E''$. τ'' is the tree formed by moving τ' from its original position in τ to the specifier position of the root of τ. Remember that $N_{\tau'} \subset N_\tau$. Let v be the parent of $root_{\tau'}$ ($v = SPEC^{-1}(root_{\tau'})$). Formally,

$N_{\tau''} = N_\tau$,

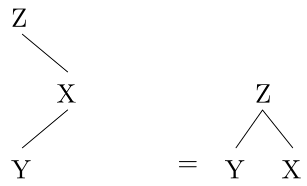
$S_{\tau''} = S_\tau$,

$\lhd_{\tau''} = (\lhd_\tau - \{\langle v, root_{\tau'}\rangle\}) \cup \{\langle root_\tau, root_{\tau'}\rangle\}$,

$\prec\!\!\prec_{\tau''} = (\prec\!\!\prec_\tau - ((N_\tau - N_{\tau'}) \times N_{\tau'})) \cup (N_{\tau'} \times (N_\tau - N_{\tau'}))$, and

$\mu''(root_\tau) = \alpha\ \delta$, $\mu''(root_{\tau'}) = \beta\ \gamma$, and $\mu''(x) = \mu(x)$ for all other $x \in N_\tau$.

move acts on the tree component of its argument by selecting a specifier-rooted subtree and moving it to the specifier of the root of the tree:



Definition 4 Let $G = \langle V, Cat, Lex, \mathcal{F} \rangle$ be a bare grammar. Then the language generated by G, $L(G)$, is the closure of Lex under $\mathcal{F}$

$$L(G) = \bigcap \{Y | Lex \subseteq Y \wedge Y \text{ is closed under } \mathcal{F}\}$$

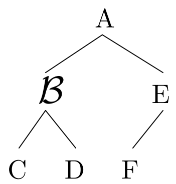
Definition 5 Let $E = \langle \tau, \mu \rangle$ be in $L(G)$. For any $c \in base$, E is a complete expression of category c if and only if, for all $n \in N_\tau$, there is some $\beta \in V^*$ such that

$$\mu(n) = \begin{cases} \beta\ c & \text{if } n = root_\tau \\ \beta & \text{otherwise} \end{cases}$$

Elements of S_τ are the *strong nodes*, and are used to determine the placement at spellout of what Brody calls morphological words (MWs).

Definition 6 For $x \in N$, let $[x]$ denote the closure of $\{x\}$ under $COMP$ and $COMP^{-1}$. $MW([x])$, the morphological word through $[x]$, is the sequence $\langle y_1, \ldots, y_{|[x]|} \rangle$, where each y_i is an element of $[x]$ and for $1 \leq j < |[x]|$, $y_{j+1} \lessdot\!\!\lessdot y_j$.

Figure 2:



Here $[A] = [E] = \{A, E\}$, $[B] = [D] = \{B, D\}$, $[C] = \{C\}$, and $[F] = \{F\}$.

$$\begin{array}{lcl} MW([A]) & = & \langle E, A \rangle \\ MW([B]) = MW([D]) & = & \langle D, B \rangle \\ MW([C]) & = & \langle C \rangle \\ MW([F]) & = & \langle F \rangle \end{array}$$

Note that $[\cdot]$ induces an equivalence relation $\approx$ on N_τ ($x \approx y$ iff $x \in [y]$).

Define $pr(\cdot)$ to be a predicate over N_τ such that $pr(\cdot)$ is true of a node x just in case either x is the least strong node in $[x]$ with respect to domination (nearest the root), or there are no strong nodes in $[x]$, and x is the greatest node in $[x]$ with respect to domination (farthest from the root).

Definition 7 Given a set R, let $max(R)/min(R)$ denote the greatest/least element of R with respect to $\lhd^*$.

$$pr(x) = \begin{cases} 1 & \text{if either } x \text{ is } min(S \cap [x]) \text{ and } S \cap [x] \neq \emptyset, \\ & \text{or } S \cap [x] \text{ is empty and } x \text{ is } max([x]). \\ 0 & \text{otherwise} \end{cases}$$

In figure 2, $pr(E) = pr(B) = pr(C) = pr(F) = 1$.

The x in $[x]$ such that $pr(x) = 1$, is the position of the MW through $[x]$ at spellout (i.e. where the MW is *pr*onounced). For $x, y \in [x]$, x is left of y at spellout just in case x occurs earlier in the MW through $[x]$ than does y. And for $[x] \neq [y]$, all members of $[x]$ are left of all members of $[y]$ at spellout just in case the position of the MW through $[x]$ at spellout left corner precedes the position of the MW through $[y]$.

Definition 8 x is left of y at spellout, $x \ll y$, iff

1. $[x] = [y] \wedge y \lessdot x$, or

2. $[x] \neq [y] \wedge z \lessdot w$, for $z \in [x], w \in [y]$ such that $pr(z) = pr(w) = 1$

In figure 2, $C \ll D \ll B \ll F \ll E \ll A$.

Proposition 2
$\forall x, y, z \; x \ll y \wedge y \ll z \rightarrow x \ll z$

Proof: Let $x \ll y \wedge y \ll z$.

CASE 1: $[x] = [y] = [z]$
Then by condition 1 in definition 8, $y \lessdot x$ and $z \lessdot y$. Since $\lessdot$ is transitive, $z \lessdot x$, and thus $x \ll z$.

CASE 2: The sets$[x]$, $[y]$, and $[z]$ are pairwise distinct.
Let $t \in [x], u \in [y], v \in [z]$ be such that $pr(t) = pr(u) = pr(v) = 1$. By condition 2 in definition 8, $t \lessdot u$ and $u \lessdot v$. Again, the transitivity of $\lessdot$ gives us that $t \lessdot v$, whence $x \ll z$.

CASE 3: Exactly two of $[x]$, $[y]$, and $[z]$ are identical.

Let $t \in [x], u \in [y], v \in [z]$ be such that $pr(t) = pr(u) = pr(v) = 1$. t cannot equal v to the exclusion of u, as that would entail that $t \prec\!\!\prec u$ (by $x \ll y$), and that $u \prec\!\!\prec t$ (by $y \ll z$). Assume $t = u$. Then $t \prec\!\!\prec v$ (as $y \ll z$), and thus $x \ll z$. Now assume $u = v$. Then $t \prec\!\!\prec v$ (as $x \ll y$), and thus $x \ll z$.

□

Proposition 3
$\forall x, y\ x \neq y \rightarrow x \ll y \vee y \ll x$

Proof: Let $x, y \in N_\tau$ be arbitrary such that $x \neq y$. There are two cases.

CASE 1: $[x] = [y]$

Since $\prec\!\!\prec$ is total, $x \prec\!\!\prec y$ or $y \prec\!\!\prec x$. Thus $y \ll x$ or $x \ll y$ by condition 1 of definition 8.

CASE 2: $[x] \neq [y]$

Let $t \in [x], u \in [y]$ be such that $pr(t) = pr(u) = 1$. Since $\prec\!\!\prec$ is total, $t \prec\!\!\prec u$ or $u \prec\!\!\prec t$. So by condition 2 of definition 8, $x \ll y$ or $y \ll x$.

□

Propositions 2 and 3, together with the asymmetry of the $\prec\!\!\prec$ relation, give that $\ll$ is a total ordering on N_τ.

Definition 9 For an expression $E = \langle \tau, \mu \rangle$, the yield of E is the concatenation of the phonetic features of the sequence $\langle y_1, \ldots, y_{|N_\tau|} \rangle$, where for $1 \leq i, j \leq |N_\tau|$, $y_i \in N_\tau$, and for $i < j$, $y_i \ll y_j$.

Definition 10 For $c \in base$, the string language of category c, $L_c(G)$, is

$$\{yield(E) | E \in L(G) \wedge E \textit{ is a complete expression of category } c\}$$

Given a MTG G, we are normally interested in the string language of G at a particular category (the start category). In this paper, unless mentioned otherwise, the start category of our grammars will be c.

3.2 An Example Grammar

To illustrate the workings of the formalism given above, I give below a naive grammar for a fragment of English.
The lexicon consists of the following five expressions (in graphical notation for convenience):

$E_1 = \langle \tau_1, \mu_1 \rangle, \quad \tau_1 = \text{N}, \quad \mu_1 : \text{N} \mapsto$ who n -wh

$E_2 = \langle \tau_2, \mu_2 \rangle, \quad \tau_2 = \mathcal{D}, \quad \mu_2 : \text{D} \mapsto \epsilon$ =n d -k

$E_3 = \langle \tau_3, \mu_3 \rangle, \quad \tau_3 = \text{V}, \quad \mu_3 : \text{V} \mapsto$ dream d= v

$E_4 = \langle \tau_4, \mu_4 \rangle, \quad \tau_4 = \mathcal{I}, \quad \mu_4 : \text{I} \mapsto$ -s =v +k i

$E_5 = \langle \tau_5, \mu_5 \rangle, \quad \tau_5 = \text{C}, \quad \mu_5 : \text{C} \mapsto \epsilon$ =i +wh c

This grammar generates only the sentence "who dreams", the derivation of which is as follows:

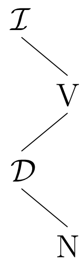
$cmerge(E_2, E_1) = E_6,$

where $\tau_6 = \begin{array}{l} \mathcal{D} \\ \quad \diagdown \\ \qquad \text{N} \end{array}$ and $\mu_6 : \begin{array}{l} \text{D} \mapsto \epsilon \text{ d -k -wh} \\ \text{N} \mapsto \text{who} \end{array}$

$smerge(E_3, E_6) = E_7,$

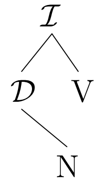
where $\tau_7 = \begin{array}{l} \qquad \text{V} \\ \quad \diagup \\ \mathcal{D} \\ \quad \diagdown \\ \qquad \text{N} \end{array}$ and $\mu_7 : \begin{array}{l} \text{V} \mapsto \text{dream v} \\ \text{D} \mapsto \epsilon \text{ -k -wh} \\ \text{N} \mapsto \text{who} \end{array}$

$cmerge(E_4, E_7) = E_8,$

where $\tau_8 =$  and μ_8 :

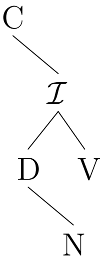
I $\mapsto$ -s +k i
V $\mapsto$ dream
D $\mapsto$ ϵ -k -wh
N $\mapsto$ who

$move(E_8) = E_9$,

where $\tau_9 =$  and μ_9 :

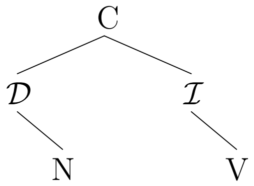
I $\mapsto$ -s i
V $\mapsto$ dream
D $\mapsto$ ϵ -wh
N $\mapsto$ who

$cmerge(E_5, E_9) = E_{10}$,

where $\tau_{10} =$  and μ_{10} :

C $\mapsto$ ϵ +wh c
I $\mapsto$ -s
V $\mapsto$ dream
D $\mapsto$ ϵ -wh
N $\mapsto$ who

$move(E_{10}) = E_{11}$,

where τ_{11} :  and μ_{11} :

C $\mapsto$ ϵ c
I $\mapsto$ -s
V $\mapsto$ dream
D $\mapsto$ ϵ
N $\mapsto$ who

In τ_{11}, [D] = [N] = $\{$D, N$\}$, and [C] = [I] = [V] = $\{$C, I, V$\}$. pr(I) = pr(D) = 1, $MW([D]) = \langle \mathrm{N}, \mathrm{D} \rangle$, $MW([I]) = \langle \mathrm{V}, \mathrm{I}, \mathrm{C} \rangle$. Since $D \prec\!\!\prec I$, $N \ll D \ll V \ll I \ll C$, and thus $yield(E_{12}) = who \cdot \epsilon \cdot dream \cdot -s \cdot \epsilon$.

3.3 Chains

The tree representation of MTGs is more expressive than we need it to be from our derivational perspective. The idea behind the more deductive "chain-based" approach of (Stabler and Keenan, 2000) is to collapse distinctions made in trees between heads or phrases that have exhausted their feature strings and will thus remain forevermore adjacent to one another. In our grammars, we need to keep track of specifier positions relative to the position of MWs. We do this by splitting the phonetic yield of an expression into three parts: exhausted specifiers above the current position of the MW, the MW itself, and exhausted specifiers below the current position of the MW. Consider the following examples:

Example 3

- $E_0 = \langle \tau_0, \mu_0 \rangle$, $\tau_0 = \mathrm{A}$, $\mu_0 : \mathrm{A} \mapsto \mathrm{a\ B{=}\ A}$
- $E_1 = \langle \tau_1, \mu_1 \rangle$, $\tau_1 = \mathrm{B}$, $\mu_1 : \mathrm{B} \mapsto \mathrm{b\ B}$

Then smerge(E_0,E_1) = E_2,

where $\tau_2 = \begin{array}{cc} & \mathrm{A} \\ / & \\ \mathrm{B} & \end{array}$ and $\mu_2 : \begin{array}{l} \mathrm{A} \mapsto \mathrm{a\ A} \\ \mathrm{B} \mapsto \mathrm{b} \end{array}$

Here $MW([root_{\tau_2}]) = \langle \mathrm{A} \rangle$, and $pr(\mathrm{A}) = 1$. B is a specifier which has no movement features, and B $-\!\!\!\!\prec\!\!\!\prec$ A. Thus we represent the phonetic yield of E_2 as the triple (b, a, ϵ). Note that B $\ll$ A, and $yield(E_2) = b \cdot a$.

Example 4

- $E_3 = \langle \tau_3, \mu_3 \rangle$, $\tau_3 = \mathcal{C}$, $\mu_3 : \mathrm{C} \mapsto \mathrm{c\ {=}A\ C\ \text{-}C}$

Then cmerge(E_3, E_2) = E_4,

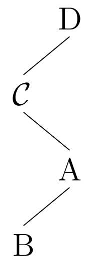
where $\tau_4 = \begin{array}{ccc} \mathcal{C} & & \\ & \backslash & \\ & & \mathrm{A} \\ & / & \\ \mathrm{B} & & \end{array}$ and $\mu_4 : \begin{array}{l} \mathrm{A} \mapsto \mathrm{a} \\ \mathrm{B} \mapsto \mathrm{b} \\ \mathrm{C} \mapsto \mathrm{c\ C\ \text{-}C} \end{array}$

Now $MW([root_{\tau_4}]) = \langle \mathrm{A}, \mathrm{C} \rangle$, and $pr(\mathrm{C}) = 1$. B is as before, but now C $-\!\!\!\!\prec\!\!\!\prec$ B. Thus we represent the phonetic yield of E_4 as the triple (ϵ, ac, b). Note again that A $\ll$ C $\ll$ B, and $yield(E_4) = a \cdot c \cdot b$.

Example 5

- $E_5 = \langle \tau_5, \mu_5 \rangle$, $\tau_5 =$ D, $\mu_5 :$ D $\mapsto$ d C= D

Then smerge$(E_5, E_4) = E_6$,

where $\tau_6 =$  and $\mu_6 :$ $\begin{array}{l} \text{A} \mapsto \text{a} \\ \text{B} \mapsto \text{b} \\ \text{C} \mapsto \text{c -C} \\ \text{D} \mapsto \text{d D} \end{array}$

Here $MW([root_{\tau_6}]) = \langle \text{D} \rangle$, and $pr(D) = 1$. Now, even though C $\prec\!\prec$ D, C has not exhausted all of its movement features, and must move to a higher position in order for E_6 to be part of a successful derivation. Thus we represent the phonetic yield of E_6 as the triple (ϵ, d, ϵ), and keep track of the yield of the specifier of D in a separate one-tuple, (acb).

An **initial chain** is an element of $V^* \times V^* \times V^* \{::, :::\} Cat^*$, and represents the exhausted specifiers and MW as described above, along with the features of the root. A **non-initial chain** is an element of $V^*\ Cat^*$, and represents those specifiers which have not exhausted their features, and thus whose relative position in the expression is not yet determined. The expression representing E_6 in example 5 is composed of the initial chain (ϵ, d, ϵ) ::: D (the three colons indicate that D is a strong node), and the non-initial chain (acb) -C. A **chain** is either an initial chain or a non-initial chain, and an **expression** is an element of $E = (\textit{initial chain})(\textit{non-initial chain})^*$.

Definition 11 (**Chain-based Mirror Theoretic Grammar**) A chain-based mirror theoretic grammar is a tuple $G = \langle V, Cat, Lex, \mathcal{F} \rangle$, such that

1. V is a non-empty set (the pronounced elements)

2. Cat is the disjoint union of the following sets:

    - $base$, a non-empty finite set.
    - $cselector = \{\texttt{=}b | b \in base\}$
    - $sselector = \{b\texttt{=}\ | b \in base\}$

- $licensee = \{\text{-}b | b \in base\}$
- $licensor = \{\text{+}b | b \in base\}$
- $str = \{::, :::\}$

3. The **lexicon**

   $$Lex \subset \{\epsilon\} \times V^* \times \{\epsilon\}\ str\ cselector^?(sselector \cup licensor)^? base\ licensee^*$$

   is a finite set (of initial chains), where each has a strength feature $\cdot \in \{:::, ::\}$ that distinguishes **strong** initial chains from **weak** initial chains, respectively.[4] [5]

4. The generating functions $\mathcal{F} = \{merge, move\}$ are partial functions from tuples of expressions to expressions. It will be convenient to define these functions in a deductive format, with the arguments as premises and the values as the conclusion.

   - $merge : (E \times E) \to E$ is the union of the following 4 functions, for $s, t \in V^*$, for $\cdot, \cdot\cdot \in \{::, :::\}$, for $f \in base$, $\gamma, \nu \in Cat^*$, $\delta \in Cat^+$, and for non-initial chains $\alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l$ $(0 \leq k, l)$

$$\frac{(s,t,u) \cdot f\text{=}\, \gamma, \alpha_1, \ldots, \alpha_k \qquad (v,w,x) \cdot\cdot f, \iota_1, \ldots, \iota_l}{(vwxs,t,u) \cdot \gamma, \alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l}\text{s-merge1}$$

---

[4]Abels (2001) suggests that facts about English verb raising and negation might be elegantly accounted for if strength can be left unspecified on lexical items. He proposes that a strengthless lexical item inherit the strength feature of 'the next affix down'. His proposal could be implemented in the present system by adding a third strength feature, ':', and extending the *merge* function with the following two *cmerge* variants:

$$\frac{(s,t,u) : \text{=}f\gamma, \alpha_1, \ldots, \alpha_k \qquad (v,w,x) :: f\nu, \iota_1, \ldots, \iota_l}{(suv,wt,x) :: \gamma\nu, \alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l}\text{c-merge-unspecified1}$$

$$\frac{(s,t,u) : \text{=}f\gamma, \alpha_1, \ldots, \alpha_k \qquad (v,w,x) ::: f\nu, \iota_1, \ldots, \iota_l}{(s,wt,uvx) ::: \gamma\nu, \alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l}\text{c-merge-unspecified2}$$

This extension would not affect the language theoretic results obtained in the next section.

[5]Brody (p.c.) suggests that morphological words be spellt out uniformly in the highest node in the MW, regardless of strength. This proposal is easily implemented in this formalization by requiring that all lexical items be strong. Restricting the lexica of MTGs in this manner does not affect the results of the next section.

$$\frac{(s,t,u) \cdot f\texttt{=}\ \gamma, \alpha_1, \ldots, \alpha_k \qquad (v,w,x) \cdot\!\cdot f\delta, \iota_1, \ldots, \iota_l}{(s,t,u) \cdot \gamma, \alpha_1, \ldots, \alpha_k, (vwx)\delta, \iota_1, \ldots, \iota_l} \text{s-merge2}$$

$$\frac{(s,t,u) :: \texttt{=}f\gamma, \alpha_1, \ldots, \alpha_k \qquad (v,w,x) \cdot f\nu, \iota_1, \ldots, \iota_l}{(suv, wt, x) :: \gamma\nu, \alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l} \text{c-merge1}$$

$$\frac{(s,t,u) ::: \texttt{=}f\gamma, \alpha_1, \ldots, \alpha_k \qquad (v,w,x) \cdot f\nu, \iota_1, \ldots, \iota_l}{(s, wt, uvx) ::: \gamma\nu, \alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l} \text{c-merge2}$$

Note that since the domains of these four functions are disjoint, their union is a function.

- *move* : $E \to E$ is the union of the following 2 functions, for $s, t \in V^*$, $f \in base$, $\gamma \in Cat^*$, $\delta \in Cat^+$, and for non-initial chains $\alpha_1, \ldots, \alpha_k$ $(0 \leq k)$ satisfying the following condition: (SMC) none of $\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k$ has $\texttt{-}f$ as its first feature.

$$\frac{(s,t,u) \cdot \texttt{+}f\gamma, \alpha_1, \ldots, \alpha_{i-1}, (v)\texttt{-}f, \alpha_{i+1}, \ldots, \alpha_k}{(vs,t,u) \cdot \gamma, \alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k} \text{move1}$$

$$\frac{(s,t,u) \cdot \texttt{+}f\gamma, \alpha_1, \ldots, \alpha_{i-1}, (v)\texttt{-}f\delta, \alpha_{i+1}, \ldots, \alpha_k}{(s,t,u) \cdot \gamma, \alpha_1, \ldots, \alpha_{i-1}, (v)\delta, \alpha_{i+1}, \ldots, \alpha_k} \text{move2}$$

Notice that the domains of move1 and move2 are disjoint, so their union is a function. The (SMC) restriction on the domain of *move* is a simple version of the "shortest move condition" (Chomsky, 1995).

Definition 12 An expression *exp* is a complete expression of a category $c \in base$ $(exp \in L_G(c))$ iff $exp = (\sigma_1, \sigma_2, \sigma_3) \cdot c$, for $\sigma_1, \sigma_2, \sigma_3 \in V^*$, and $\cdot \in \{::, :::\}$. The yield of a complete expression $(\sigma_1, \sigma_2, \sigma_3) \cdot c$ is $\sigma_1\sigma_2\sigma_3$.

Definition 13 Given a MTG $G = \langle V, Cat, Lex, \mathcal{F} \rangle$, for $c \in base$ the strings of category c are $L_c(G) = \{yield(exp) | exp \in L_G(c)\}$.

Definition 14 The mirror theoretic languages are those in

$$MTL = \{L | \exists G \in MTG \text{ such that } L \text{ is } L_c(G) \text{ for some } c \in base_G\}$$

3.4 An Example Grammar Revisited

The grammar from section 3.2 is here translated into our chain based formalism, and the sentence "who dreams" is derived again. The reader is advised to compare each step in the following with the corresponding step in the previous example, partial depictions of which are included here for convenience. In the interests of brevity, I leave out the labeling function μ, instead writing the phonetic features of a node adjacent to it in the tree (thereby obtaining a representation virtually identical to the one familiar from Brody's work).

The lexicon consists of the following five expressions:

$E_1 =$ $(\epsilon,\text{who},\epsilon)$ :: n -wh

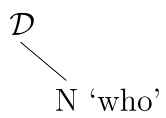
$E_2 =$ $(\epsilon,\epsilon,\epsilon)$:::=n d -k

$E_3 =$ $(\epsilon,\text{dream},\epsilon)$ :: d= v

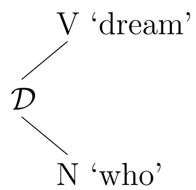
$E_4 =$ $(\epsilon,\text{-s},\epsilon)$:::=v +k i

$E_5 =$ $(\epsilon,\epsilon,\epsilon)$ :: =i +wh c

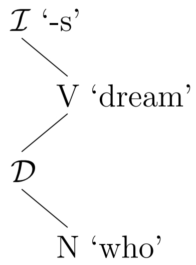
c-merge2$(E_2, E_1) = E_6 =$ $(\epsilon,\text{who},\epsilon)$ :::d -k -wh



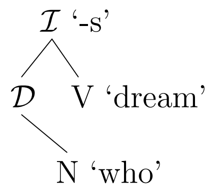
s-merge2$(E_3, E_6) = E_7 =$ $(\epsilon,\text{dream},\epsilon)$ :: v, (who) -k -wh



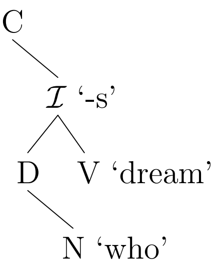
c-merge2(E_4, E_7) $= E_8 =$ (ϵ,dream -s,ϵ):::+k i, (who) -k -wh



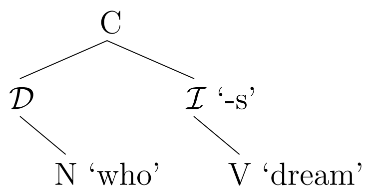
move2(E_8) $= E_9 =$ (ϵ,dream -s,ϵ) ::: i, (who) -wh



c-merge1(E_5, E_9) $= E_{10} =$ (ϵ,dream -s,ϵ) :: +wh c, (who) -wh



move1(E_{10}) $= E_{11} =$ (who,dream -s,ϵ) :: c



4 Formal Properties of Mirror Theoretic Grammars

The expressive power of certain classes of MTGs is considered in this section. First the unrestricted formalism given in §3 is shown to be outside of the class of grammars thought appropriate for linguistic investigations. Then a lexical restriction called the non-recursiveness condition (NRC) is motivated in section 4.2. Sections 4.4 and 4.5 establish that MTGs meeting this condition are weakly equivalent to minimalist grammars (MGs) (Stabler, 1997; Stabler and Keenan, 2000), and multiple context-free languages (MCFLs) (Seki et al., 1991) respectively. Next, a subclass of MTGs that doesn't use movement features is introduced, and is shown to generate non-context-free languages.

4.1 Mirror Theory and Mild Context Sensitivity

The (language theoretic) complexity of natural languages is often assumed to be somewhere in between context-sensitive and context-free. Joshi (1985) gives an intuitive characterization (in the form of a list of conditions) of a family of languages which depart from the context-free class 'just enough' to adequately describe the structure of natural language sentences. Of particular interest here is the condition of 'constant growth', as mirror theoretic grammars, as presented in this paper, are able to define languages which are not of constant growth. This formalism thus does not explain the apparent constant growth property of natural language (though see (Michaelis and Kracht, 1997) for arguments that natural language is not actually of constant growth). Below is a grammar for the strictly context sensitive language $\{b^{2^n} | n \in \mathbf{N}\}$.

The lexicon consists of the following six expressions:

E_1. $(\epsilon, \epsilon, \epsilon) :: k\ \texttt{-}k$
E_2. $(\epsilon, \epsilon, \epsilon) :: k\texttt{=}\ \ g\ \texttt{-}k$
E_3. $(\epsilon, \epsilon, \epsilon) :: \texttt{=}k\ g\ \texttt{-}k$
E_4. $(\epsilon, \epsilon, \epsilon) :: \texttt{=}g\ \texttt{+}k\ k\ \texttt{-}k$
E_5. $(\epsilon, \epsilon, \epsilon) :: k\texttt{=}\ \ c$
E_6. $(\epsilon, \mathrm{b}, \epsilon) :: \texttt{=}c\ \texttt{+}k\ c$

$E_1 - E_4$ recursively build expressions with base category k such that each successive cycle doubles the number of movement features from the last cycle.

The expressions thus generated have no pronounced material, and function as 'counters' in the next step of the derivation. E_5 selects an expression of category k to get an expression of category c, and then E_6 recursively selects expressions of base category c, each time checking a licensee feature, until all licensee features are checked. An example derivation of *bbbb* follows. There are 2 'cycles' present: 1 – 3, and 4 – 9.

1.	s-merge2(E_2, E_1)	=	E_7	=	$(\epsilon, \epsilon, \epsilon) :: g\ \text{-}k, (\epsilon)\text{-}k$
2.	c-merge1(E_4, E_7)	=	E_8	=	$(\epsilon, \epsilon, \epsilon) :: \text{+}k\ k\ \text{-}k\ \text{-}k, (\epsilon)\text{-}k$
3.	move1(E_8)	=	E_9	=	$(\epsilon, \epsilon, \epsilon) :: k\ \text{-}k\ \text{-}k$
4.	s-merge2(E_2, E_9)	=	E_{10}	=	$(\epsilon, \epsilon, \epsilon) :: g\ \text{-}k, (\epsilon)\text{-}k\ \text{-}k$
5.	c-merge1(E_4, E_{10})	=	E_{11}	=	$(\epsilon, \epsilon, \epsilon) :: \text{+}k\ k\ \text{-}k\ \text{-}k, (\epsilon)\text{-}k\ \text{-}k$
6.	move2(E_{11})	=	E_{12}	=	$(\epsilon, \epsilon, \epsilon) :: k\ \text{-}k\ \text{-}k, (\epsilon)\text{-}k$
7.	c-merge1(E_3, E_{12})	=	E_{13}	=	$(\epsilon, \epsilon, \epsilon) :: g\ \text{-}k\ \text{-}k\ \text{-}k, (\epsilon)\text{-}k$
8.	c-merge1(E_4, E_{13})	=	E_{14}	=	$(\epsilon, \epsilon, \epsilon) :: \text{+}k\ k\ \text{-}k\ \text{-}k\ \text{-}k\ \text{-}k, (\epsilon)\text{-}k$
9.	move1(E_{14})	=	E_{15}	=	$(\epsilon, \epsilon, \epsilon) :: k\ \text{-}k\ \text{-}k\ \text{-}k\ \text{-}k$
10.	smerge2(E_5, E_{15})	=	E_{16}	=	$(\epsilon, \epsilon, \epsilon) :: c, (\epsilon)\text{-}k\ \text{-}k\ \text{-}k\ \text{-}k$
11.	cmerge1(E_6, E_{16})	=	E_{17}	=	$(\epsilon, b, \epsilon) :: \text{+}k\ c, (\epsilon)\text{-}k\ \text{-}k\ \text{-}k\ \text{-}k$
12.	move2(E_{17})	=	E_{18}	=	$(\epsilon, b, \epsilon) :: c, (\epsilon)\text{-}k\ \text{-}k\ \text{-}k$
13.	cmerge1(E_6, E_{18})	=	E_{19}	=	$(\epsilon, bb, \epsilon) :: \text{+}k\ c, (\epsilon)\text{-}k\ \text{-}k\ \text{-}k$
14.	move2(E_{19})	=	E_{20}	=	$(\epsilon, bb, \epsilon) :: c, (\epsilon)\text{-}k\ \text{-}k$
15.	cmerge1(E_6, E_{20})	=	E_{21}	=	$(\epsilon, bbb, \epsilon) :: \text{+}k\ c, (\epsilon)\text{-}k\ \text{-}k$
16.	move2(E_{21})	=	E_{22}	=	$(\epsilon, bbb, \epsilon) :: c, (\epsilon)\text{-}k$
17.	cmerge1(E_6, E_{22})	=	E_{23}	=	$(\epsilon, bbbb, \epsilon) :: \text{+}k\ c, (\epsilon)\text{-}k$
18.	move1(E_{23})	=	E_{24}	=	$(\epsilon, bbbb, \epsilon) :: c$

The SMC condition on the domain of the *move* operation ensures that all features in a cycle be checked in the next cycle (in the derivation above, line 6 could not be followed with line 7′. smerge2(E_2, E_{12})), which in turn guarantees that the number of movement features will double at the close of each cycle. Note that this grammar does not depend on the particular implementation of the feature percolation used here, other than the fact that the ratio of the total number of licensee features to those which are accessible in any given expression is unbounded (only the first syntactic feature in a feature string is accessible to any generating function). It is this, in fact, which provides the crucial tweak to derive languages of non-constant growth.

4.2 Placing a Bound on Lexical Recursion

As stated above, the fact that an unbounded number of features can be added to a MW without increasing the number of features visible to the move operation is at the heart of the non-constant growth property of the formalism. Linguistic evidence is presented here which, when analyzed in the framework of mirror theory, provides support for a restriction on the MTG formalism which will be seen in later sections to reenforce constant growth.

Stabler (1994) notices that there seems to be an upper bound of about three on the acceptability of iterated morphological causative markers in natural languages. This is in surprising contrast with the apparent naturalness of iteration of periphrastic causatives (such as 'make'). This difference seems to persist even when morphological processes in general are considered. To account for this peculiarity of natural language, only those grammars are considered which impose a finite bound on the depth of any complement sequence (i.e. to those which require that there is an $n \in \mathbf{N}$ which bounds the cardinality of all MWs).[6] This restriction is most naturally formalized as a restriction on lexicons.

First I make precise the notion of a 'sequence of complements'.

Definition 15 Given a MTG $G = \langle V, Cat, Lex, \mathcal{F}\rangle$,

$CS_1 = \{\langle \ell \rangle | \ell \in Lex\}$, and

$CS_{n+1} = \{s^\frown\langle \ell \rangle | s = \langle \ell_1, \ldots, \ell_n \rangle \in CS_n \wedge \langle \ell \rangle \in CS_1$ such that ℓ_n has a cselect feature =z, and ℓ has a base feature z$\}$

Intuitively each CS_n contains all n-ary sequences of complements. A naive formalization of the intuition above would be to require that $\bigcup_{n=0}^{\infty} CS_n$ be finite. This requirement however, is equivalent to the following:[7] [8]

[6] In Mirror Theory, a morphological causative marker would be in the same MW as the verb, whereas a periphrastic causative would not be.

[7] Assume that $cs \in CS_n$ such that $cs_i = cs_j$, for $i < j$. Then the subsequence $cs_{i+1,j} \in CS_{j-i+1}$. However, $cs_{i+1,j} \frown cs_{i+1,j}$ is also a complement sequence, as $cs_i (= cs_j)$ has a cselect feature that matches with the base feature of cs_{i+1}. In fact, for any m, $cs_{i+1,j}^m \in \bigcup_{n=0}^{\infty} CS_n$, whence $\bigcup_{n=0}^{\infty} CS_n$ is infinite.

[8] This restriction in effect nullifies the formalism's feature-percolation ability. A similar formalism without feature percolation is given in (Kobele et al., 2002), which can be proven to be equivalent to this restricted version.

Definition 16 Non-recursiveness Condition (NRC)
A Grammar $G = \langle V, Cat, Lex, \mathcal{F}\rangle$ meets the non-recursiveness condition iff for all n, and for all $s \in CS_n$, $i \neq j$ implies $\ell_i \neq \ell_j$ for $1 \leq i, j \leq n$.

The expressive power of this restricted class of MTGs is considered in the next few subsections.

4.3 Preliminaries

The mirror theoretic languages with grammars meeting the NRC (NRC-MTGs) are shown to contain the languages generated by minimalist grammars (MGs) (Stabler, 1997; Stabler and Keenan, 2000) in §4.4, and to be contained in the multiple context-free languages (MCFLs) (Seki et al., 1991) in §4.5. I use MGs without head movement or affix hopping, as they are the simplest formally.[9] MGs were shown to be equivalent to MCFGs in (Harkema, 2001; Michaelis, 1998; Michaelis, 2001), and thus, by the results of §4.4 and §4.5, $MG \equiv NRC\text{-}MTG \equiv MCFG$. In §4.4 and §4.5 I will sometimes refer to mirror theoretic grammars meeting the non-recursiveness condition simply as MTGs. All references to MTGs or MTLs in these sections should be understood to be shorthand for NRC-MTGs or MTLs with grammars meeting the NRC respectively, unless explicitly stated otherwise.

Both containment proofs map derivations in one grammar to appropriately similar derivations in another. In the remainder of this section I introduce concepts on which the proofs below rely.

The derivation-tree language for a grammar is a subset of the term algebra with elements of the lexicon as constants, and with function symbols those in $\mathcal{F}$ ($\Gamma(G) \subseteq TermAlg(Lex, \mathcal{F})$). I help myself to a surjective partial map from elements of this term algebra to expressions in the language, an 'interpretation' function. If $t \in TermAlg(Lex, \mathcal{F})$, $ev(t)$ is the expression generated by the derivation described by t, if one exists.

Definition 17 The derivation-tree language of a bare grammar $G = \langle Lex, \mathcal{F}\rangle$ is the set $\Gamma(G) = \bigcup_{n=0}^{\infty} \Gamma_n(G)$, where

1. $\Gamma_0(G) = Lex$

[9]As shown by (Harkema, 2001; Michaelis, 1998; Michaelis, 2001; Stabler, 2001), adding head movement and/or affix hopping doesn't change the weak expressive power of the MG formalism.

2. $\Gamma_n(G) = \Gamma_{n-1}(G) \cup \{\sigma(t_1,\ldots,t_i) | t_1,\ldots,t_i \in \Gamma_{n-1}(G) \wedge \sigma \in \mathcal{F} \wedge \langle ev(t_1),\ldots,ev(t_i)\rangle \in dom(\sigma)\}$

Definition 18 Given $d \in \Gamma(G)$, $Subderiv(d)$, the set of subderivations of d, is:

for $d \in Lex$, $Subderiv(d) = \{d\}$

for $d = \sigma(t_1,\ldots,t_i)$, $Subderiv(d) = \{d\} \cup \bigcup_{n=1}^{i} Subderiv(t_n)$

Definition 19 The root of a derivation tree d is defined to be

d, if $d \in \Gamma_0(G)$

σ, if $d = \sigma(t_1,\ldots,t_n)$

In the grammars we are interested in (i.e. MTGs), the features of the first (linear) argument to the generating functions determine the further behaviour of the resultant expression. We are often interested in the particular lexical item that contributed these features. Given a derivation d of an expression, we can determine what this lexical item was as follows.

Definition 20 $head : \Gamma(G) \rightarrow Lex$ is:

$head(\ell) = \ell$, $\ell \in Lex$

$head(\sigma(t_1,\ldots,t_n)) = head(t_1)$, for $\sigma(t_1,\ldots,t_n) \in \Gamma(G)$

Where $b = head(d)$, for some $d \in \Gamma(G)$, b is called the **head** of d.

4.4 A Lower Bound on Weak Generative Capacity

In what follows, a proof will be given showing the inclusion of the minimalist languages (Stabler, 1997; Stabler and Keenan, 2000) in the mirror theoretic ones. First, minimalist grammars without head movement or affix hopping are introduced, as presented in (Stabler and Keenan, 2000). Then an example MG is given, along with the corresponding MTG by the translation presented next. Finally, the inclusion proof is presented.

4.4.1 Minimalist Grammars

Definition 21 A Minimalist Grammar (Stabler and Keenan, 2000) is a bare grammar $MG = \langle V, Cat, Lex, \mathcal{F}\rangle$, where

1. V is a finite set, the *pronounced elements*,

2. Cat is the disjoint union of the following sets:

   - *base*, a non-empty finite set.
   - *selector* $= \{=b | \mathrm{b} \in base\}$
   - *licensee* $= \{\text{-b} | \mathrm{b} \in base\}$
   - *licensor* $= \{+\mathrm{b} | \mathrm{b} \in base\}$

3. Lex, a finite set of lexical items, is a subset of

$$V^*\{::\}(selector \cup (selector \cup licensor)^*)^? base\ licensee^*$$

4. $\mathcal{F} = \{merge, move\}$ is a set of partial functions from tuples of expressions to expressions. Expressions $E \doteq V^*\{:, ::\}Cat^*$. Each expression has a type $\cdot \in \{:, ::\}$ that indicates whether it is *simple* (::) or *complex* (:).

   - *merge* is the union of the following three functions: for $\sigma, \tau \in V^*$, $\gamma \in Cat^*$, $\delta \in Cat^+$, $\cdot \in \{:, ::\}$,

$$\frac{\sigma :: =x\gamma, \phi_1, \ldots, \phi_n \quad \tau \cdot x, \psi_1, \ldots, \psi_m}{\sigma\tau : \gamma, \phi_1, \ldots, \phi_n, \psi_1, \ldots, \psi_m}\ mrg1$$

$$\frac{\sigma : =x\gamma, \phi_1, \ldots, \phi_n \quad \tau \cdot x, \psi_1, \ldots, \psi_m}{\tau\sigma : \gamma, \phi_1, \ldots, \phi_n, \psi_1, \ldots, \psi_m}\ mrg2$$

$$\frac{\sigma \cdot =x\gamma, \phi_1, \ldots, \phi_n \quad \tau \cdot x\delta, \psi_1, \ldots, \psi_m}{\sigma : \gamma, \phi_1, \ldots, \phi_n, \tau : \delta, \psi_1, \ldots, \psi_m}\ mrg3$$

   - *move* is the union of the following two functions: for $\sigma, \tau \in V^*$, $\gamma \in Cat^*$,$\delta \in Cat^+$, and every ϕ_i has a different first feature (the SMC from §1),

$$\frac{\sigma : +\mathrm{x}\gamma, \phi_1, \ldots, \phi_{i-1}, \tau : \text{-x}, \phi_{i+1}, \ldots, \phi_m}{\tau\sigma : \gamma, \phi_1, \ldots, \phi_{i-1}, \phi_{i+1}, \ldots, \phi_m}\ mv1$$

$$\frac{\sigma : +\mathrm{x}\gamma, \phi_1, \ldots, \phi_{i-1}, \tau : \text{-x}\delta, \phi_{i+1}, \ldots, \phi_m}{\sigma : \gamma, \phi_1, \ldots, \phi_{i-1}, \tau : \delta, \phi_{i+1}, \ldots, \phi_m}\ mv2$$

The language generated by a Minimalist Grammar G, $L(G)$, is the closure of the lexicon under the functions *merge* and *move*. For any $b \in base$, the string language of G at b is $\{\sigma | \sigma \cdot b \in L(G)\}$, and is denoted $L_b(G)$. When b is understood, I sometimes drop the subscript and write simply $L(G)$. If the first argument to *merge* is a simple expression (one of type ::), the second argument to *merge* is a complement. Otherwise, the second argument is a specifier. Note that though an expression can have at most one complement, there is no restriction on the number of specifiers it may have (other than finitude). However, as shown in (Kobele and Kandybowicz, 2001; Michaelis, 2001), minimalist grammars in which no expression has more than one specifier yield the same set of languages.[10] G is in K normal form (KNF) just in case no expression in $L(G)$ has more than one specifier per head (i.e. $Lex \subseteq selector^{?}(selector \cup licensor)^{?}\ base\ licensee^{*}$).

4.4.2 A Minimalist Example

The MG $G = \langle V_G, Cat_G, Lex_G, \mathcal{F}_G \rangle$ below is in K normal form, and generates the sentence 'who scream -s' of type c.
$Lex_G =$

$E_0 =$ who :: d -k -wh

$E_1 =$ scream :: =d v -v

$E_2 =$ -s :: =v +v t

$E_3 =$ ϵ :: =t +k agrs

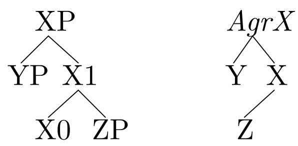
$E_4 =$ ϵ :: =agrs +wh c

1.	mrg3(E_1, E_0)	=	E_5	=	scream : v -v, who : -k -wh
2.	mrg3(E_2, E_5)	=	E_6	=	-s : +v t, scream : -v, who : -k -wh
3.	mv1(E_6)	=	E_7	=	scream -s : t, who : -k -wh
4.	mrg1(E_3, E_7)	=	E_8	=	scream -s : +k agrs, who : -k -wh
5.	mv2(E_8)	=	E_9	=	scream -s : agrs, who : -wh
6.	mrg1(E_4, E_9)	=	E_{10}	=	scream -s : +wh c, who : -wh
7.	mv1(E_{10})	=	E_{11}	=	who scream -s : c

[10] The proof idea is simple: every lexical item is mapped to a sequence of lexical items in an equivalent grammar - for every specifier a lexical item in G has past the first (this can be read off from the shape of the lexical item), an additional item is needed in the sequence in G' (to host the specifier position).

4.4.3 Transforming MGs into MTGs

With each MG G in K normal form, one can associate with it a MTG $T(G)$ that will be proven later to be weakly equivalent to it. The idea behind the transformation is found in (Brody, 1997). Brody notes that the standard X-bar tree below must be represented in his system as the neighboring mirror theoretic tree:



Intuitively, the idea behind the proof of equivalence is simply that the two trees above are functionally equivalent in that YP/Y and ZP/Z are the only 'movable' items (ignoring head movement). Thus operations performed by one can be simulated by the other.

In mirror theory, complements are different from specifiers not only with respect to precedence relative to a head, but also in that complements form a tighter unit with the selecting head than do specifiers; complements cannot extract - specifiers can. Thus, we represent both minimalist specifiers and complements by mirror theoretic specifiers, as per the picture above. Proposition 1 requires us to factor each minimalist lexical item that selects (by merge or move) a complement and a specifier into two mirror theoretic lexical items that form a morphological word, and each of which select a specifier. To ensure the minimalist specifier-head-complement pronunciation order, the least lexical item in the morphological word with respect to domination (the one which c-selects the other) must be strong. In the above example, $MW([X])$ is pronounced in AgrX; following Y, but preceding Z at spellout ($Y \ll X \ll AgrX \ll Z$). The minimalist lexical items are uniformly mirrored by two mirror theoretic lexical items in the definitions to follow, even though not strictly necessary in the case where the minimalist item has no selection features (i.e. neither YP nor ZP is present).

Definition 22 Let G be a minimalist grammar in KNF, and let $enum(\cdot)$ be an enumeration of Lex_G. $T(G) = \langle V, Cat_{T(G)}, Lex_{T(G)}, \mathcal{F}_{T(G)} \rangle$, where

1. $base_{T(G)} = base_G \cup \{x_i | enum(i) \in Lex_G\}$, $x_i \notin Cat_G$ for $1 \leq i \leq |Lex_G|$

2. $Lex_{T(G)} = \{\ell | \exists i\ \ell \in f(enum(i))\}$, where $f(enum(i)) =$

$\langle(\epsilon,\sigma,\epsilon)$ ::: =x_i d -w$_1$ ... -w$_n$, $(\epsilon,\epsilon,\epsilon)$ ::: x$_i\rangle$,
if $enum(i) = \sigma$ :: d -w$_1$ ... -w$_n$,

$\langle(\epsilon,\sigma,\epsilon)$ ::: =x_i d -w$_1$ ... -w$_n$, $(\epsilon,\epsilon,\epsilon)$ ::: y= x$_i\rangle$,
if $enum(i) = \sigma$ :: =y d-w$_1$ ... -w$_n$

$\langle(\epsilon,\sigma,\epsilon)$ ::: =x_i z= d -w$_1$ ... -w$_n$, $(\epsilon,\epsilon,\epsilon)$ ::: y= x$_i\rangle$,
if $enum(i) = \sigma$ :: =y z= d -w$_1$ ... -w$_n$

$\langle(\epsilon,\sigma,\epsilon)$ ::: =x_i +z d -w$_1$ ... -w$_n$, $(\epsilon,\epsilon,\epsilon)$ ::: y= x$_i\rangle$,
if $enum(i) = \sigma$ :: =y +z d -w$_1$ ... -w$_n$

A lexical item ℓ in the minimalist grammar is associated with a set S of lexical items in the mirror theoretic grammar just in case f maps ℓ to a sequence containing each member of S. S is the **association class** of ℓ, $f(\ell)_1$ is the first element in the sequence f maps ℓ to, and $f(\ell)_2$ is the second.

One sees immediately that if c-merge is defined on $\langle ev(d), ev(d')\rangle$, for $d, d' \in \Gamma(T(G))$, then, as $head(d)$ must contain =x_i, and $head(d')$ x_i, and since only members of the same association class share features =x_j and x_j, $\langle head(d), head(d')\rangle$ must be the image of some $\ell \in Lex_G$ under f.

Moreover, if one member of an association class occurs in a derivation d such that $ev(d)$ begins with some $b \in base_G$, then both members of the association class must occur in d. Since only the first member of each $f(\ell)$ has a category symbol $b \in base_G$, it must occur in d, and because the first member of an $f(\ell)$ begins with a complement selection feature =x_i, the second member of $f(\ell)$ must occur as well, by the previous observation.

4.4.4 An Example Transformation

Given the 'who scream -s' grammar in §4.4.2, a MTG $T(G)$ is constructed by the method described above. In §4.4.6 I introduce the notion of a mapping between minimalist and mirror theoretic derivations, and then compare the derivations of 'who scream -s' in both grammars.
Take $enum : \{0,1,2,3,4\} \to Lex_G$ to map numbers to lexical items of G in the order given in §4.4.2.

1. $base_{T(G)} = base_G \cup \{x_0, x_1, x_2, x_3, x_4\}$

2. $Lex_{T(G)} = \{\ell | \ell \in f(enum(i))$ for some $0 \le i \le 4\}$

   $D_0 = f(enum(0))_1 = (\epsilon, \text{who}, \epsilon)$ ::: =x_0 d -k -wh

$D_1 = f(enum(0))_2 = (\epsilon, \epsilon, \epsilon) ::: x_0$
$D_2 = f(enum(1))_1 = (\epsilon, \text{scream}, \epsilon) ::: \texttt{=}x_1 \text{ v } \texttt{-v}$
$D_3 = f(enum(1))_2 = (\epsilon, \epsilon, \epsilon) ::: d\texttt{=}\ x_1$
$D_4 = f(enum(2))_1 = (\epsilon, \text{-s}, \epsilon) ::: \texttt{=}x_2 \ \texttt{+v t}$
$D_5 = f(enum(2))_2 = (\epsilon, \epsilon, \epsilon) ::: \text{v}\texttt{=}\ x_2$
$D_6 = f(enum(3))_1 = (\epsilon, \epsilon, \epsilon) ::: \texttt{=}x_3 \ \texttt{+k} \text{ agrs}$
$D_7 = f(enum(3))_2 = (\epsilon, \epsilon, \epsilon) ::: \text{t}\texttt{=}\ x_3$
$D_8 = f(enum(4))_1 = (\epsilon, \epsilon, \epsilon) ::: \texttt{=}x_4 \ \texttt{+wh} \text{ c}$
$D_9 = f(enum(4))_2 = (\epsilon, \epsilon, \epsilon) ::: \text{agrs}\texttt{=}\ x_4$

What follows is a derivation of 'who scream -s' in $T(G)$:

1.	c-merge2(D_0, D_1)	$=$	D_{10}
		$=$	$(\epsilon, \text{who}, \epsilon)$ ::: d -k -wh
2.	s-merge2(D_3, D_{10})	$=$	D_{11}
		$=$	$(\epsilon, \epsilon, \epsilon) ::: x_1$, (who) -k -wh
3.	c-merge2(D_2, D_{11})	$=$	D_{12}
		$=$	$(\epsilon, \text{scream}, \epsilon)$ ::: v -v, (who) -k -wh
4.	s-merge2(D_5, D_{12})	$=$	D_{13}
		$=$	$(\epsilon, \epsilon, \epsilon) ::: x_2$, (scream) -v, (who) -k -wh
5.	c-merge2(D_4, D_{13})	$=$	D_{14}
		$=$	$(\epsilon, \text{-s}, \epsilon)$ ::: +v t, (scream) -v, (who) -k -wh
6.	move1(D_{14})	$=$	D_{15}
		$=$	$(\text{scream}, \text{-s}, \epsilon)$ ::: t, (who)-k -wh
7.	s-merge1(D_7, D_{15})	$=$	D_{16}
		$=$	$(\text{scream -s}, \epsilon, \epsilon) ::: x_3$, (who) -k -wh
8.	c-merge2(D_6, D_{16})	$=$	D_{17}
		$=$	$(\epsilon, \epsilon, \text{scream -s})$ ::: +k agrs, (who) -k -wh
9.	move2(D_{17})	$=$	D_{18}
		$=$	$(\epsilon, \epsilon, \text{scream -s})$ ::: agrs, (who) -wh
10.	s-merge1(D_9, D_{18})	$=$	D_{19}
		$=$	$(\text{scream -s}, \epsilon, \epsilon) ::: x_4$, (who) -wh
11.	c-merge2(D_8, D_{19})	$=$	D_{20}
		$=$	$(\epsilon, \epsilon, \text{scream -s})$ ::: +wh c, (who) -wh
12.	move1(D_{20})	$=$	D_{21}
		$=$	$(\text{who}, \epsilon, \text{scream -s})$ ::: c

4.4.5 Transforming Derivations

To prove the equivalence of $G \in MG$ and $T(G) \in MTG$, we define a string preserving bijection from (a subset of) the derivations of G to (a subset of) the derivations of $T(G)$. We are interested in the derivations whose heads have satisfied their selection (x= , =x) and licensor (+x) features, and thus can themselves be selected for. Intuitively (in GB/Minimalist terms), we are interested just in the derivations of (completed) maximal projections (XPs).

Definition 23

- For $G = \langle V, Cat, Lex, \mathcal{F} \rangle$ a MG, and for any $c \in Cat$ and $\cdot \in \{:, ::\}$,

$$derive_c(G) = \{d \in \Gamma(G) | ev(d) = \sigma \cdot c\delta, \phi_1, \ldots, \phi_n\}$$

- For $G' = \langle V', Cat', Lex', \mathcal{F}' \rangle$ a MTG, and for any $c \in Cat'$ and $\cdot \in str$,

$$derive_c(G') = \{d \in \Gamma(G') | ev(d) = (\sigma_1, \sigma_2, \sigma_3) \cdot c\delta, \phi_1, \ldots, \phi_n\}$$

I define a function relating the structure building minimalist functions with the mirror theoretic ones. As mentioned above, the intuition is that minimalist complements (base generated by merge1 or merge3) and specifiers (base generated by merge2 or merge3) are treated uniformly as mirror theoretic specifiers (base generated by smerge1 or smerge2).

Definition 24

$g(\text{mrg1}) = g(\text{mrg2}) = \text{s-merge1}$,

$g(\text{mrg3}) = \text{s-merge2}$,

$g(\text{mv1}) = \text{move1}$, and

$g(\text{mv2}) = \text{move2}$.

Definition 25
Let $h : \bigcup_{b \in base_G} derive_b(G) \rightarrow \bigcup_{b \in base_G} derive_b(T(G))$ be defined by term induction as follows:
for a a lexical item, $\Phi, \Psi \in \bigcup_{b \in base_G} derive_b(G)$, and $\sigma, \tau \in \mathcal{F}_G$,

1. a is α :: b -w$_1$ ... -w$_i$, and
   $h(a) = \text{c-merge2}(f(a)_1, f(a)_2)$

2. a is α :: =y b -w$_1$ ... -w$_i$, and
   $h(\sigma(a, \Phi)) = \text{c-merge2}(f(a)_1, g(\sigma)(f(a)_2, h(\Phi)))$

3. a is α :: =x =y b -w$_1$ ... -w$_i$, and
   $h(\sigma(\tau(a, \Phi), \Psi)) = g(\sigma)(\text{c-merge2}(f(a)_1, g(\tau)(f(a)_2, h(\Phi))), h(\Psi))$

4. a is α :: =x +y b -w$_1$ ... -w$_i$, and
   $h(\sigma(\tau(a, \Phi))) = g(\sigma)(\text{c-merge2}(f(a)_1, g(\tau)(f(a)_2, h(\Phi))))$

The domain of h is the set of maximally projected derivations in G, the range is the set of maximally projected derivations in $T(G)$ *of just the categories in G*.

4.4.6 A Derivation Transformed

The minimalist derivation of 'who scream -s' in the grammar G in §4.4.2 is here reproduced as an element in $\Gamma(G)$. The morphism h defined above is applied to it, yielding the derivation tree in $\Gamma(T(G))$ corresponding to the mirror theoretic derivation of 'who scream -s' in §4.4.4.

$$\begin{aligned}
&h(\text{mv1}(\text{mrg1}(E_4, \text{mv2}(\text{mrg1}(E_3, \text{mv1}(\text{mrg3}(E_2, \text{mrg3}(E_1, E_0)))))))) \\
&= \; h(\text{mv1}(\text{mrg1}(E_4, \Phi))) \\
&= \; g(\text{mv1})(\text{c-merge2}(f(E_4)_1, g(\text{mrg1})(f(E_4)_2, h(\Phi)))) \\
&= \; \text{move1}(\text{c-merge2}(D_8, \text{s-merge1}(D_9, h(\Phi))))
\end{aligned}$$

$$\begin{aligned}
&h(\Phi) \\
&= \; h(\text{mv2}(\text{mrg1}(E_3, \text{mv1}(\text{mrg3}(E_2, \text{mrg3}(E_1, E_0)))))) \\
&= \; h(\text{mv2}(\text{mrg1}(E_3, \Psi)))))) \\
&= \; g(\text{mv2})(\text{c-merge2}(f(E_3)_1, g(\text{mrg1})(f(E_3)_2, h(\Psi)))) \\
&= \; \text{move2}(\text{c-merge2}(D_6, \text{s-merge1}(D_7, h(\Psi))))
\end{aligned}$$

$$\begin{aligned}
&h(\Psi) \\
&= \; h(\text{mv1}(\text{mrg3}(E_2, \text{mrg3}(E_1, E_0)))) \\
&= \; h(\text{mv1}(\text{mrg3}(E_2, \Upsilon))) \\
&= \; g(\text{mv1})(\text{c-merge2}(f(E_2)_1, g(\text{mrg3})(f(E_2)_2, h(\Upsilon)))) \\
&= \; \text{move1}(\text{c-merge2}(D_4, \text{s-merge2}(D_5, h(\Upsilon))))
\end{aligned}$$

$h(\Upsilon)$

$$
\begin{aligned}
&= h(\text{mrg3}(E_1, E_0)) \\
&= \text{c-merge2}(f(E_1)_1, g(\text{mrg3})(f(E_1)_2, h(E_0))) \\
&= \text{c-merge2}(D_2, \text{s-merge2}(D_3, h(E_0)))
\end{aligned}
$$

$h(E_0)$

$$
\begin{aligned}
&= \text{c-merge2}(f(E_0)_1, f(E_0)_2) \\
&= \text{c-merge2}(D_0, D_1)
\end{aligned}
$$

4.4.7 $ML \subseteq MTL$

The proof strategy is as follows. First, I show that the derivation tree d' in $\Gamma(T(G))$ gotten from d in $\Gamma(G)$ by h in the above manner has the same category and licensee features as d, and also the 'same' string component. Next I show that every d' in $\Gamma(T(G))$ such that $d' \in derive_c(T(G))$ for $c \in base_G$ is the image of h under some d in $\Gamma(G)$. Finally, using a result of (Hale and Stabler, 2001), I conclude that two distinct trees in $\Gamma(G)$ are mapped by h to distinct trees in $\Gamma(T(G))$. This shows the injectivity of h. The inclusion of the minimalist languages in the mirror theoretic ones is then a special case of proposition 4, namely, one in which only the complete expressions of a particular category are relevant.

Proposition 4

1. h is a bijection, and
2. $ev(d) = \alpha \cdot \gamma\delta, \beta_1 \cdot \chi_1, \ldots, \beta_k \cdot \chi_k$ if and only if $ev(h(d)) = (\alpha_1, \alpha_2, \alpha_3) \cdot \gamma\delta, (\beta_1)\chi_1, \ldots, (\beta_k)\chi_k$, for $\alpha = \alpha_1\alpha_2\alpha_3$.

Proof: The 'only if' direction of part 2 of proposition 4 is shown first. The 'if' direction will then follow from the bijectivity of h. The proof is by an induction on the height of the term d:

base d is a lexical item, and falls under condition 1 in the definition of h. Since for some $b \in base_G$, $d \in Subtree_b(G)$, $d = \alpha :: \text{b -w}_1 \ldots \text{-w}_n = enum(i)$, for some i. By the definition of f, $f(d) = \langle(\epsilon, \alpha, \epsilon) ::: \text{=x}_i \text{ b -w}_1 \ldots \text{-w}_n, (\epsilon, \epsilon, \epsilon) ::: \text{x}_i\rangle$. Then $h(d) = \text{c-merge2}(f(d)_1, f(d)_2)$, and $ev(\text{c-merge2}(f(d)_1, f(d)_2))$ is
$(\epsilon, \alpha, \epsilon) ::: \text{b -w}_1 \ldots \text{-w}_n$.

induction There are ten cases to consider here, corresponding to the possible instantiations of the function variables σ and τ above. Two are worked out here; the remainder are left to the interested reader. The first case is an instance of condition 2 in the definition of h, and the eighth is one of condition 4.

CASE 1 $d = \text{mrg1}(\ell, \Phi)$, where ℓ is $\alpha ::$ =y b -w$_1 \ldots$ -w$_n$, and $enum(\ell) = k$

$h(d)$ then, is c-merge2$(f(\ell)_1, g(mrg1)(f(\ell)_2, h(\Phi))) =$ c-merge2$((\epsilon, \alpha, \epsilon)$ ::: x$_k$ b -w$_1 \ldots$ -w$_n$, s-merge1$((\epsilon, \epsilon, \epsilon)$ :::y= x$_k$, $h(\Phi)))$. As Φ is the second argument to $merge1$, we know that $ev(\Phi)$ must be of the form: $\beta \cdot$ y, $\phi_1, \ldots, \phi_j$ and thus $ev(d) = \alpha\beta$: b -w$_1 \ldots$ -w$_n$, $\phi_1, \ldots, \phi_j$. By the inductive hypothesis, we have that $h(\Phi) = (\beta_1, \beta_2, \beta_3) \cdot$y, $\phi_1, \ldots, \phi_j$. Then ev(s-merge1$((\epsilon, \epsilon, \epsilon)$ ::: y= x$_k$, $h(\Phi))) = (\beta, \epsilon, \epsilon)$ ::: x$_k$, $\phi_1, \ldots,$ ϕ_j, and finally $ev(h(d)) = (\epsilon, \alpha, \beta)$ ::: b -w$_1 \ldots$ -w$_n$, $\phi_1, \ldots, \phi_j$

CASE 2 $d = \text{mrg3}(\ell, \Phi)$

CASE 3 $d = \text{mrg2}(\text{mrg1}(\ell, \Phi), \Psi)$

CASE 4 $d = \text{mrg2}(\text{mrg3}(\ell, \Phi), \Psi)$

CASE 5 $d = \text{mrg3}(\text{mrg1}(\ell, \Phi), \Psi)$

CASE 6 $d = \text{mrg3}(\text{mrg3}(\ell, \Phi), \Psi)$

CASE 7 $d = \text{mv1}(\text{mrg1}(\ell, \Phi))$

CASE 8 $d = \text{mv1}(\text{mrg3}(\ell, \Phi))$, where ℓ is $\alpha ::$ =z +y b-w$_1 \ldots$ -w$_n$.

$h(d) =$ move1(c-merge2$((\epsilon, \alpha, \epsilon)$ ::: =x$_k$+y b -w$_1 \ldots$ -w$_n$, s-merge2$((\epsilon, \epsilon, \epsilon)$ ::: z= x$_k$, $h(\Phi))))$. As in case 1, we know $ev(\Phi)$ to be $\beta\cdot$z -l$_1 \ldots$ -l$_i$, $\phi_1, \ldots, \phi_{j-1}, \gamma$-y, $\phi_{j+1}, \ldots, \phi_m$ and thus $ev(d) = \gamma\alpha$: b -w$_1 \ldots$ -w$_n$, β-l$_1 \ldots$ -l$_i$, $\phi_1, \ldots, \phi_m$.

By the inductive hypothesis, $h(\Phi) = (\beta_1, \beta_2, \beta_3) \cdot$ z -l$_1 \ldots$ -l$_i$, $\phi_1, \ldots, \phi_{j-1}, (\gamma)$-y, $\phi_{j+1}, \ldots, \phi_m$. Then ev(s-merge2$((\epsilon, \epsilon, \epsilon)$ ::: z= x$_k$, $h(\Phi))) =$ $(\epsilon, \epsilon, \epsilon)$ ::: x$_k$, (β)-l$_1 \ldots$ -l$_i$, $\phi_1, \ldots, \phi_{j-1}, (\gamma)$-y, $\phi_{j+1}, \ldots, \phi_m$, and thus $ev(h(d)) = (\gamma, \alpha, \epsilon)$ ::: b -w$_1 \ldots$ -w$_n$, (β)-l$_1 \ldots$ -l$_i$, $\phi_1, \ldots, \phi_{j-1}, \phi_{j+1}, \ldots, \phi_m$.

CASE 9 $d = \text{mv2}(\text{mrg1}(\ell, \Phi))$

CASE 10 $d = \text{mv2}(\text{mrg3}(\ell, \Phi))$

I next show that h is onto.

Let $d' \in Subtree_b(T(G))$, for some $b \in base_G$. We construct $d \in Subtree_b(G)$ such that $h(d) = d'$. The construction is via term induction on the height of d'.

base

Let $d' \in \Gamma_1(T(G)) \cap Subtree_b(T(G))$ (By construction, $\Gamma_0(T(G)) \cap Subtree_b(T(G))$ is empty)

Then $d' =$ c-merge2($(\epsilon, \alpha, \epsilon)$ ::: =x$_i$ b -w$_1 \ldots$ -w$_n$, $(\epsilon, \epsilon, \epsilon)$ ::: x$_i$), for some i. Then by definition 22.2, there is some $d \in Lex_G$ such that $f(d)_1 = (\epsilon, \alpha, \epsilon)$ ::: x$_i$ b -w$_1 \ldots$ -w$_n$ and $f(d)_2 = (\epsilon, \epsilon, \epsilon)$ ::: x$_i$. Then by definition 25.1 $h(d) =$ c-merge2($f(d)_1, f(d)_2$) $= d'$.

induction

Assume that for every $d'' \in \Gamma_n(T(G)) \cap Subtree_b(T(G))$ there is some $d \in Subtree_b(G)$ such that $h(d) = d''$. Let $d' \in \Gamma_{n+1}(T(G)) \cap Subtree_b(T(G))$. We construct $d \in Subtree_b(G)$ such that $h(d) = d'$. By definition 22, the head of d' can be one of

- $(\epsilon, \alpha, \epsilon)$ ::: =x$_k$ b-w$_1 \ldots$ -w$_n$,
- $(\epsilon, \alpha, \epsilon)$ ::: =x$_k$ y= b-w$_1 \ldots$ -w$_n$, or
- $(\epsilon, \alpha, \epsilon)$ ::: =x$_k$ +z b-w$_1 \ldots$ -w$_n$

We examine the third of these cases in detail.

CASE 3 $head(d') = (\epsilon, \alpha, \epsilon)$ ::: =x$_k$ +z b -w$_1 \ldots$ -w$_n$.

By definition 22, there is a unique $\ell \in Lex_G$ such that $f(\ell)_1 = head(d')$ and $f(\ell)_2 = (\epsilon, \epsilon, \epsilon)$ ::: y= x$_k$. Again by definition 22, $\ell = \alpha$ :: =y +z b-w$_1 \ldots$ -w$_n$. As both members of an association class must co-occur, d' must be σ(c-merge2($head(d')$, $\tau((\epsilon, \epsilon, \epsilon)$ ::: y= x$_k$, Φ))), for $\sigma \in \{$move1, move2$\}$, and $\tau \in \{$s-merge1, s-merge2$\}$. There are four cases, depending on the values of σ and τ. Let $\sigma =$ move2 and $\tau =$ s-merge1. Then $ev(\Phi) = (\beta_1, \beta_2, \beta_3)$ ::: y$, \phi_1, \ldots, \phi_{i-1}, (\gamma)$-l$_1 \ldots$-l$_j, \phi_{i+1}, \ldots,$
ϕ_m. By the inductive hypothesis, there is some $\Phi' \in \Gamma(G)$ such that $h(\Phi') = \Phi$. By the result of the immediately preceding proof, $ev(\Phi') = \beta$: y$, \phi_1, \ldots, \phi_{i-1}, \gamma$: -l$_1 \ldots$-l$_j, \phi_{i+1}, \ldots, \phi_m$. Thus

mv2(mrg1(ℓ, Φ')) is defined, and so by definition 25, h(mv2(mrg1(d, Φ)) = move2(c-merge2($f(d)_1$, s-merge1($f(d)_2, h(\Phi')$))) $= d'$

Note that if $d_1, \ldots, d_n$ is the sequence of lexical items of some $d \in \Gamma(G)$, then $f(d_1), \ldots, f(d_n)$ is the sequence of lexical items of $h(d) \in \Gamma(T(G))$. The previous proof demonstrated that every lexical sequence of a tree in $\Gamma(T(G))$ is the image of some lexical sequence of a tree in $\Gamma(G)$. To show injectivity, note that if $a, b \in \Gamma(G)$ are distinct, then so are the lexical sequences of both, $\langle a_1, \ldots, a_m \rangle$ and $\langle b_1, \ldots, b_m \rangle$ (Hale and Stabler, 2001), and thus so must be $\langle f(a_1), \ldots, f(a_m) \rangle$ and $\langle f(b_1), \ldots, f(b_m) \rangle$, as $\exists i\ a_i \neq b_i$, and for all $\ell, \ell' \in Lex_G$ with $\ell \neq \ell'$, $f(\ell) \neq f(\ell')$. □

Theorem 1
$ML \subseteq MTL$.

Proof: Let L be an arbitrary language in ML. There is a KNF grammar G such that $L_c(G) = L$, for some $c \in base_G$. Let $\ell \in L_c(G)$. By definition, $\ell \in \{\delta | \delta \cdot c \in Cl(Lex_G, \mathcal{F}_G)\}$. Thus there is some $d \in \Gamma(G)$ such that $ev(d) = \ell \cdot c$. By definition 25, $h(d) \in \Gamma(T(G))$ such that $ev(h(d)) = (\ell_1, \ell_2, \ell_3) ::: c$. Then $\ell \in \{\alpha\beta\gamma | (\alpha, \beta, \gamma) \cdot c \in Cl(Lex_{T(G)}, \mathcal{F}_{T(G)})\}$, and thus $\ell \in L_c(T(G))$. Now let $\ell \in L_c(T(G))$, for some MG G. By definition 13, $\ell = \sigma_1\sigma_2\sigma_3$ and for some $d \in \Gamma(T(G))$ and $\cdot \in \{::, :::\}$, $ev(d) = (\sigma_1, \sigma_2, \sigma_3) \cdot c$. By proposition 4.1, $\exists d' \in \Gamma(G)$ such that $h(d') = d$. By proposition 4.2, $ev(d') = \sigma_1\sigma_2\sigma_3 \cdot c$. Thus $\ell \in L_b(G)$.

□

4.5 An Upper Bound for MTGs

4.5.1 Multiple Context-Free Grammars

I preface the formal definition of MCFGs below with a more intuitive description. The reader familiar with MCFGs should skip to §4.5.2.

Take a CFG $G = \langle V, T, P, S \rangle$. Each production in P of the form $A \longrightarrow \sigma$, for $A \in V$ and $\sigma \in T^*$, associates with a non-terminal (in this case, 'A') a string ('σ'), or, equivalently, a function $f \in [1 \rightarrow T^*]$. Each production in P of the form $A \longrightarrow \sigma_1 A_1 \ldots \sigma_n A_n \sigma_{n+1}$ associates with a non-terminal ('A') a set of strings, which in the case above contains all strings of the form σ_1 concatenated with a string associated with $A_1, \ldots$, concatenated with σ_n, concatenated with a string associated with A_n, concatenated with σ_{n+1}.

Equivalently, A is associated with a function $g \in [T^* \times \ldots \times T^* \rightarrow T^*]$, such that g maps strings of type A_1 through A_n to the string described above. An MCFG is an extension of the CFG described above, in that the functions f, g are from tuples of tuples of strings to tuples of strings (i.e. $f \in \left[1 \rightarrow \langle T^* \rangle^k\right]$, and $g \in \left[\langle T^* \rangle^{k_1} \times \ldots \times \langle T^* \rangle^{k_n} \rightarrow \langle T^* \rangle^{k_{n+1}}\right]$). Moreover, g is allowed to be nearly *any* function with the appropriate domain and codomain (not just the concatenation function), as long as it does not 'copy' any string in its input. An m-MCFG places an upper bound of m on the length of the tuples of strings. Here is a formal definition of MCFGs, from (Seki et al., 1991) :

Definition 26 An m-MCFG $G = \langle N, O, F, R, S \rangle$, where

N

N is a finite set of *non-terminal* symbols. For each $A \in N$, a positive integer $d(A)$ (the 'degree' of A) is associated with it.

O

Letting T (the terminal symbols) be a finite set of symbols disjoint from N, $O = \bigcup_{i=1}^{m} \langle T^* \rangle^i$. O is a set of sequences of strings over T.

F

F is a finite set of partial functions from $O \times O \times \ldots \times O$ to O, in other words, functions from sequences of sequences of strings, to sequences of strings. F_q is the subset of F in $[O^q \rightarrow O]$. F must satisfy the following conditions:

1. for each $f \in F_q$, there are positive integers $r(f)$, and $d_i(f)$ (for $1 \leq i \leq q$) which are not greater than m (the maximum length of sequences in O), such that f is a function from $(T^*)^{d_1(f)} \times (T^*)^{d_2(f)} \times \ldots (T^*)^{d_q(f)}$ to $(T^*)^{r(f)}$. Intuitively, $r(f)$ is the length of the sequence which is the output of f, and each $d_i(f)$ is the length of the i^{th} argument of f.
2. For $1 \leq h \leq r(f)$, let f^h denote the h^{th} component of f (that is, f^h on any input returns the h^{th} coordinate of f on the same input). Now let
$$\overline{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$$
and
$$X = \{x_{ij} | 1 \leq i \leq p \;\&\; 1 \leq j \leq d_i(f)\}$$

Here $\overline{x}_i$ is the i^{th} argument of f (which is a p-ary sequence of strings over T), and X is the set of all strings in each sequence of f's arguments. Now

$$f^h[\overline{x}_1, \ldots, \overline{x}_p] = \alpha_{h0} z_{h1} \alpha_{h1} z_{h2} \ldots z_{hv_h(f)} \alpha_{hv_h(f)}$$

where $\alpha_{hk} \in T^*$ $(0 \leq k \leq v_h(f))$ and $z_{hk} \in X$ $(1 \leq k \leq v_h(f))$. That is to say, each coordinate of the output of f is the concatenation of constant strings and strings in the input to f.

3. Moreover, each x_{ij} can only occur in at most *one* coordinate of the output of f, and then only once. (The non-copying condition)

R

R is a finite subset of $\bigcup_q (F_q \times N^{q+1})$. For $(f, A_0, \ldots, A_q) \in P$, write $A_0 \to f[A_1, \ldots, A_q]$. If $q = 0$, it is a *terminating rule* - then f is a nullary function, and denotes a sequence of strings over T. Moreover, if $A_0 \to f[A_1, \ldots, A_q]$ is a rule in R, then $d(A_0) = r(f)$, and for $1 \leq i \leq q$, $d(A_i) = d_i(f)$. That is, the sequences that are given as input to f must be in the domain of f, and the non-terminal the rule wants to assign the output of f to must be the right 'size'.

S

$S \in N$ is the start symbol, and $d(S) = 1$.

For $A \in N$, $L_G(A)$ is the smallest set such that:

1. if a terminating rule $A \to f[]$ is in R and $f(\emptyset) = \theta$, then $\theta \in L_G(A)$;

2. if $A \to f[A_1, \ldots, A_q] \in R$, for $1 \leq i \leq q$, $\theta_i \in L_G(A_i)$, and $f(\theta_1, \ldots, \theta_q)$ is defined, then $f(\theta_1, \ldots, \theta_q) \in L_G(A)$

The language generated by a MCFG G is $L(G) =_{\text{def}} L_G(S)$. As $d(S)$ was made equal to 1, $L(G)$ is a set of strings over T.

Example 6 Below is a MCFG for the language $\{ww|w \in \{a, b\}^*\}$. The non-terminals (N) are W and S (the start symbol), the terminals (T) are a and b, F contains A, B, con, and (ϵ, ϵ) such that $A, B \in [\langle T^* \rangle^2 \to \langle T^* \rangle^2]$, $con \in [\langle T^* \rangle^2 \to T^*]$, and $(\epsilon, \epsilon) \in [1 \to \langle T^* \rangle^2]$.

$A(x,y) = (ax, ay)$, $B(x,y) = (bx, by)$, $con(x,y) = xy$, and $(\epsilon,\epsilon)(\emptyset) = (\epsilon,\epsilon)$. The rules are the following:

$$\begin{array}{l} S \rightarrow con[W] \\ W \rightarrow A[W] \\ W \rightarrow B[W] \\ W \rightarrow (\epsilon,\epsilon)[] \end{array}$$

$L_G(W) = \{\langle w, w\rangle | w \in \{a,b\}^*\}$, and $L_G(S) = \{ww | w \in \{a,b\}^*\}$.
The string *baba* is derived in four steps:

$$\begin{array}{lcl} S & \rightarrow & con[W] \\ & \rightarrow & con[B[W]] \\ & \rightarrow & con[B[A[W]]] \\ & \rightarrow & con[B[A[(\epsilon,\epsilon)[]]]] \end{array}$$

Note that

$$\begin{array}{lcl} con(B(A((\epsilon,\epsilon)(\emptyset)))) & = & con(B(A(\langle\epsilon,\epsilon\rangle))) \\ & = & con(B(\langle a, a\rangle)) \\ & = & con(\langle ba, ba\rangle) \\ & = & baba \end{array}$$

For comparison, and because we already have a characterization of derivation languages for bare grammars, I provide a Bare Grammar characterization of MCFGs.

Definition 27
An MCFG $G = \langle N, O, F, R, S\rangle$ can be viewed as a bare grammar $G' = \langle V, Cat, Lex, \mathcal{F}\rangle$, where $V = T$, $Cat = N$, $Lex = \emptyset$, and $\mathcal{F} = R$, s.t. for $f = n_0 \rightarrow \sigma\,[n_1, \ldots, n_i] \in \mathcal{F}$, $f : \langle N \times O\rangle^i \rightarrow N \times O$ s.t. $f(\langle n_1, o_1\rangle, \ldots, \langle n_i, o_i\rangle) = \langle n_0, \sigma(o_1, \ldots, o_i)\rangle$ ('$n_0 \rightarrow \sigma\,[n_1, \ldots, n_i]$' is the *name* of a function which behaves as described above). For $f \in \mathcal{F}_0$, $f : 1 \rightarrow N \times O$, and $f = n_0 \rightarrow \sigma$. Thus $f(\emptyset) = \langle n_0, \sigma(\emptyset)\rangle$. $L(G') = \{o | \langle S, o\rangle \in Cl(Lex, \mathcal{F})\}$.

Let d be an arbitrary MCFG derivation. $fst(ev(d))$, the first component of $ev(d)$, is the category of $ev(d)$, and $snd(ev(d))$ is the string component of $ev(d)$.

Example 7 The MCFG in example 6 above gives us the following grammar:
$G = \langle\{a,b\}, \{S, W, X\}, \emptyset, \{S \rightarrow con[W], W \rightarrow A[W], W \rightarrow B[W],$

$W \rightarrow (\epsilon, \epsilon)[]\}\rangle$.
Below is a derivation tree for the string *baba*.

$$d = S \rightarrow con[W](W \rightarrow B[W](W \rightarrow A[W](W \rightarrow (\epsilon, \epsilon)[](\emptyset))))$$

$$ev(d) = \langle S, baba \rangle$$

4.5.2 MTG Embedding in MCFGs

Throughout this section I follow closely (Michaelis, 1998). The general idea of the embedding is to, for an arbitrary NRC-MTG G with m licensee features, represent each expression by a $m+1$-tuple, which contains the features of the initial chain and the non-initial chains. Each of these $m+1$-tuples is taken as a non-terminal symbol. The SMC guarantees that each successful derivation never has a subderivation with more than one non-initial chain headed by a particular $\texttt{-}l_i$ feature. Thus each of the non-initial coordinates in the $m+1$-tuple are taken to be a non-initial chain headed by $\texttt{-}l_i$, for $1 \leq i \leq m$. The merge and move operations are further decomposed into cases, depending on the particular feature the merged/moved item is headed with. From the sets defined in definition 28 (from (Michaelis, 1998)),[11]our non-terminal symbols are built; $suf(Cat)$ is defined to be the set of feature strings that are suffixes of a lexical item, and $suf(\texttt{-}l_i)$ the subset of that set which begin with $\texttt{-}l_i$.

Definition 28
$suf(Cat) =$
$\{\kappa \in Cat_G^* | \exists \kappa' \in Cat_G^* \; \exists \sigma_0, \sigma_1, \sigma_2 \in \Sigma^* \; \exists \cdot \in str \; (\sigma_0, \sigma_1, \sigma_2) \cdot \kappa'\kappa \in Lex_G\}$
$suf(\texttt{-}l_i) = \{\kappa \in suf(Cat) | \kappa = \epsilon \vee \exists \lambda \in Cat^* \; \kappa = \texttt{-}l_i\lambda\}$

[11]Why doesn't this work for arbitrary MTGs? $suf(Cat)$ doesn't suffice to describe the possible features of an initial chain. Consider the following example with a simple grammar for the context-free language $\{a^n b^n | n \in \mathbf{N}\}$:

Example 8 The lexicon consists of the following four lexical items:
E_1. $(\epsilon, \epsilon, \epsilon)$ ::: A
E_2. $(\epsilon, \text{a}, \epsilon)$ ::: =A A -A
E_3. $(\epsilon, \epsilon, \epsilon)$ ::: A= B
E_4. $(\epsilon, \text{b}, \epsilon)$ ::: =B +A B

This grammar derives the string aabb.

In order to allow for the feature percollation given rise to by the c-merge function, I define the set of possible sequences of inherited licensee features. For $n = |Lex|$, I define $inhrt(Cat)$ to be the set of sequences in $suf(\text{-l}_{i_1}) \cdot \ldots \cdot suf(\text{-l}_{i_n})$. Note $inhrt(Cat)$ is finite as it is the finite concatenation of finite sets.

Definition 29 Given a MTG G with $|licensee| = m$, an enumeration of the features in $licensee$ and $c \in base$ a distinguished feature, its MCFG image G' is the m+3-MCFG $\langle N, O, F, R, S\rangle$ below. Essentially, we split the MTG merge and move operations into two parts. F deals with the string components, and R with the features.

N

every $n \in N$ is either 'S' or an $m+1$-tuple $\langle \hat{\mu}_0, \mu_1, \ldots, \mu_m\rangle$, where $\hat{\mu}_0 = \langle \mu_0, a_\mu\rangle$, with $\mu_0 \in suf(Cat) \cdot inhrt(Cat), \mu_i \in suf(-l_i) \cdot inhrt(Cat)$, and $a_\mu \in \{s, w\}$.
Intuitively, a_μ records the strength (s, w) of the initial chain. μ_0 records the features of the initial chain, and μ_i those of the non-initial chains. Most of the non-terminals defined in this manner are useless, and will be ignored in examples.

O

$O = \Sigma^* \cup \langle \Sigma^* \rangle^{m+3}$

1.	c-merge2(E_2, E_1)	=	E_5	=	$(\epsilon, \text{a}, \epsilon)$ ::: A -A
2.	c-merge2(E_2, E_5)	=	E_6	=	$(\epsilon, \text{aa}, \epsilon)$ ::: A -A -A
3.	s-merge2(E_3, E_6)	=	E_7	=	$(\epsilon, \epsilon, \epsilon)$ ::: B, (aa) -A -A
4.	c-merge2(E_4, E_7)	=	E_8	=	$(\epsilon, \text{b}, \epsilon)$ ::: +A B, (aa) -A -A
5.	move2(E_8)	=	E_9	=	$(\epsilon, \text{b}, \epsilon)$ ::: B, (aa) -A
6.	c-merge2(E_4, E_9)	=	E_{10}	=	$(\epsilon, \text{bb}, \epsilon)$ ::: +A B, (aa) -A
7.	move1(E_{10})	=	E_{11}	=	$(\text{aa}, \text{bb}, \epsilon)$ ::: B

$suf(Cat)$ for the above grammar is the set

$$\{\langle \text{A}\rangle, \langle \text{=A}, \text{A}, \text{-A}\rangle, \langle \text{A}, \text{-A}\rangle, \langle \text{-A}\rangle, \langle \text{A=}, \text{B}\rangle, \langle \text{B}\rangle, \langle \text{=B}, \text{+A}, \text{B}\rangle, \langle \text{+A}, \text{B}\rangle, \epsilon\}$$

But to represent expression E_6 we would need a non-terminal with the shape $\langle \text{A}, \text{-A}, \text{-A}\rangle$, which is not in $suf(Cat)$ because of the feature inheritance in the definition of the c-merge functions. Moreover, as the number of licensee features a particular derived expression may have in this grammar is unbounded, no finite extension of $suf(Cat)$ will suffice to capture all the possible suffixes of chains in this grammar.

F

$con \in \left[\langle\Sigma^*\rangle^{m+3} \to \Sigma^*\right]$,
$cmerge1 \in \left[\langle\Sigma^*\rangle^{m+3} \times \langle\Sigma^*\rangle^{m+3} \to \langle\Sigma^*\rangle^{m+3}\right]$, as are $cmerge2$, $smerge1$, and, for all $1 \leq j \leq m$, $smerge2j$,
for $1 \leq j \leq m$, $move1j \in \left[\langle\Sigma^*\rangle^{m+3} \to \langle\Sigma^*\rangle^{m+3}\right]$, as are $move2jk$, for all $1 \leq j, k \leq m$,
and $\langle\sigma_0, \sigma_1, \sigma_2, \epsilon, \ldots, \epsilon\rangle \in \left[1 \to \langle\Sigma^*\rangle^{m+3}\right]$, for every $(\sigma_0, \sigma_1, \sigma_2) \cdot \phi \in Lex_G$, for $\cdot \in \{::, :::\}$,

s.t.

$con[(\pi_0, \ldots, \pi_{m+2})] = \pi_0 \ldots \pi_{m+2}$

$cmerge1[(\sigma_0, \ldots, \sigma_{m+2}), (\tau_0, \ldots, \tau_{m+2})]$
$= (\sigma_0\sigma_2\tau_0, \tau_1\sigma_1, \tau_2, \kappa_1, \ldots, \kappa_m)$

$cmerge2[(\sigma_0, \ldots, \sigma_{m+2}), (\tau_0, \ldots, \tau_{m+2})]$
$= (\sigma_0, \tau_1\sigma_1, \sigma_2\tau_0\tau_2, \kappa_1, \ldots, \kappa_m)$

$smerge1[(\sigma_0, \ldots, \sigma_{m+2}), (\tau_0, \ldots, \tau_{m+2})]$
$= (\tau_0\tau_1\tau_2\sigma_0, \sigma_1, \sigma_2, \kappa_1, \ldots, \kappa_m)$, where $\kappa_i = \sigma_i$ if $\sigma_i \neq \epsilon$, and τ_i otherwise

$smerge2j[(\sigma_0, \ldots, \sigma_{m+2}), (\tau_0, \ldots, \tau_{m+2})]$
$= (\sigma_0, \sigma_1, \sigma_2, \kappa_1, \ldots, \kappa_m)$, where $\kappa_i = \sigma_i$ if $\sigma_i \neq \epsilon$, τ_i otherwise, and $\kappa_j = \tau_0\tau_1\tau_2$

$move1j[(\sigma_0, \ldots, \sigma_{m+2})]$
$= (\sigma_{j+2}\sigma_0, \ldots, \sigma_{j+1}, \epsilon, \sigma_{j+3}, \ldots, \sigma_{m+2})$

$movejk[(\sigma_0, \ldots, \sigma_{m+2})]$
$= (\sigma_0, \sigma_1, \sigma_2, \kappa_1, \ldots, \kappa_m)$, where $\kappa_i = \sigma_i, \kappa_j = \epsilon$, and $\kappa_k = \sigma_j$

$\langle\sigma_0, \sigma_1, \sigma_2, \epsilon, \ldots, \epsilon\rangle[] = (\sigma_0, \sigma_1, \sigma_2, \epsilon, \ldots, \epsilon)$

R

1. (a) $\langle\langle\phi, w\rangle, \epsilon, \ldots, \epsilon\rangle \longrightarrow \langle\sigma_0, \sigma_1, \sigma_2, \epsilon, \ldots, \epsilon\rangle[]$, for every

$(\sigma_0, \sigma_1, \sigma_2) :: \phi \in Lex_G$

(b) $\langle\langle\phi, s\rangle, \epsilon, \ldots, \epsilon, \rangle \longrightarrow \langle\sigma_0, \sigma_1, \sigma_2, \epsilon, \ldots, \epsilon\rangle[]$, for every $(\sigma_0, \sigma_1, \sigma_2) ::: \phi \in Lex_G$

2. $S \longrightarrow con[\langle\langle c, y\rangle, \epsilon, \ldots, \epsilon\rangle]$, for $y \in \{s, w\}$ and $c \in base$ the distinguished feature

3. (a) $T_0 \longrightarrow cmerge1[T_1, T_2]$
where $T_1 = \langle\hat{\mu}_0, \mu_1, \ldots, \mu_m\rangle$, $T_2 = \langle\hat{\nu}_0, \nu_1 \ldots, \nu_m\rangle$ s.t. $\mu_0 =$ =zγ, $a_\mu = w$, $\nu_0 = z\delta$, and for $1 \leq i \leq m$ $\mu_i = \epsilon$ if $\nu_i \neq \epsilon$ and $T_0 = \langle\hat{\kappa}_0, \kappa_1 \ldots, \kappa_m\rangle$, where $\hat{\kappa}_0 = \langle\gamma\delta, w\rangle$, and for $1 \leq i \leq m$ $\kappa_i = \mu_i$ if $\nu_i = \epsilon$, else $\kappa_i = \nu_i$

(b) $T_0 \longrightarrow cmerge2[T_1, T_2]$
where $T_1 = \langle\hat{\mu}_0, \mu_1, \ldots, \mu_m\rangle$, $T_2 = \langle\hat{\nu}_0, \nu_1 \ldots, \nu_m\rangle$ s.t. $\mu_0 =$ =zγ, $a_\mu = s$, $\nu_0 = z\delta$, and for $1 \leq i \leq m$ $\mu_i = \epsilon$ if $\nu_i \neq \epsilon$ and $T_0 = \langle\hat{\kappa}_0, \kappa_1 \ldots, \kappa_m\rangle$, where $\hat{\kappa}_0 = \langle\gamma\delta, s\rangle$, and for $1 \leq i \leq m$ $\kappa_i = \mu_i$ if $\nu_i = \epsilon$, else $\kappa_i = \nu_i$

(c) $T_0 \longrightarrow smerge1[T_1, T_2]$
where $T_1 = \langle\hat{\mu}_0, \mu_1 \ldots, \mu_m\rangle$, $T_2 = \langle\hat{\nu}_0, \nu_1 \ldots, \nu_m\rangle$ s.t. $\mu_0 =$ z=γ, $\nu_0 = z$, and for $1 \leq i \leq m$ $\mu_i = \epsilon$ if $\nu_i \neq \epsilon$ and $T_0 = \langle\hat{\kappa}_0, \kappa_1 \ldots, \kappa_m\rangle$, where $\hat{\kappa}_0 = \langle\gamma, a_\mu\rangle$, and for $1 \leq i \leq m$ $\kappa_i = \mu_i$ if $\nu_i = \epsilon$, else $\kappa_i = \nu_i$

(d) $T_0 \longrightarrow smerge2j[T_1, T_2]$
where $T_1 = \langle\hat{\mu}_0, \mu_1, \ldots, \mu_m\rangle$, $T_2 = \langle\hat{\nu}_0, \nu_1, \ldots, \nu_m\rangle$ s.t. $\mu_0 =$ z=γ, $\nu_0 = z$-$l_j\delta$, $\mu_j = \nu_j = \epsilon$, and for $1 \leq i \leq m$ $\mu_i = \epsilon$ if $\nu_i \neq \epsilon$ and $T_0 = \langle\hat{\kappa}_0, \kappa_1 \ldots, \kappa_m\rangle$, where $\kappa_0 = \gamma$, $\kappa_j =$ -$l_j\gamma$, and for $1 \leq i \leq m$ $\kappa_i = \mu_i$ if $\nu_i = \epsilon$, else $\kappa_i = \nu_i$

(e) $T_0 \longrightarrow move1j[T_1]$
where $T_1 = \langle\hat{\mu}_0, \mu_1, \ldots, \mu_m\rangle$, s.t. $\mu_0 =$ +$l_j\gamma$, $\mu_j =$ -l_j, and $T_0 = \langle\hat{\kappa}_0, \kappa_1, \ldots, \kappa_m\rangle$, where $\hat{\kappa}_0 = \langle\gamma, a_\mu\rangle$, $\kappa_j = \epsilon$, and for $1 \leq i \leq m$ $\kappa_i = \mu_i$

(f) $T_0 \longrightarrow move2jk[T_1]$
where $T_1 = \langle\hat{\mu}_0, \mu_1, \ldots, \mu_m\rangle$, s.t. $\mu_0 =$ +$l_j\gamma$, $\mu_j =$ -l_j-$l_k\delta$, $\mu_k = \epsilon$, and $T_0 = \langle\hat{\kappa}_0, \kappa_1, \ldots, \kappa_m\rangle$, where $\hat{\kappa}_0 = \langle\gamma, a_\mu\rangle$, $\kappa_j = \epsilon$, $\kappa_k =$ -$l_k\delta$ and for $1 \leq i \leq m$ $\kappa_i = \mu_i$

4. Nothing is in R if not by virtue of the above

4.5.3 An Example Grammar Re-revisited

The 'who dreams' grammar of §1 is here translated into a 5-MCFG by means of the algorithm above. For reasons of space, not all non-terminals are given; only those which are used in the derivation will be explicit.
The following enumeration is used:

$$\begin{bmatrix} 1 & \rightarrow & \texttt{-k} \\ 2 & \rightarrow & \texttt{-wh} \end{bmatrix}$$

and take c as our distinguished feature.
First, corresponding to each MTG lexical item we get a non-terminal, and a terminating rule:

$(\epsilon, \text{who}, \epsilon) :: \texttt{n -wh}$

1. $T_1 = \langle\langle \texttt{n -wh}, w\rangle, \epsilon, \epsilon\rangle$
2. $T_1 \longrightarrow \langle \epsilon, \text{who}, \epsilon, \epsilon, \epsilon\rangle[]$

$(\epsilon, \epsilon, \epsilon):::\texttt{=n d -k}$

1. $T_2 = \langle\langle \texttt{=n d -k}, s\rangle, \epsilon, \epsilon\rangle$
2. $T_2 \longrightarrow \langle \epsilon, \epsilon, \epsilon, \epsilon, \epsilon\rangle[]$

$(\epsilon,\text{dream},\epsilon) :: \texttt{d= v}$

1. $T_3 = \langle\langle \texttt{d= v}, w\rangle, \epsilon, \epsilon\rangle$
2. $T_3 \longrightarrow \langle \epsilon, \text{dream}, \epsilon, \epsilon, \epsilon\rangle[]$

$(\epsilon,\text{-s},\epsilon):::\texttt{=v +k i}$

1. $T_4 = \langle\langle \texttt{=v +k i}, s\rangle, \epsilon, \epsilon\rangle$
2. $T_4 \longrightarrow \langle \epsilon, \text{-s}, \epsilon, \epsilon, \epsilon\rangle[]$

$(\epsilon, \epsilon, \epsilon) :: \texttt{=i +wh c}$

1. $T_5 = \langle\langle \texttt{=i +wh c}, w\rangle, \epsilon, \epsilon\rangle$
2. $T_5 \longrightarrow \langle \epsilon, \epsilon, \epsilon, \epsilon, \epsilon\rangle[]$

The following non-terminals and non-terminating rules are also in the MCFG.

1. (a) $T_6 = \langle\langle \text{d -k -wh}, s\rangle, \epsilon, \epsilon\rangle$
   (b) $T_6 \longrightarrow cmerge2[T_2, T_1]$
2. (a) $T_7 = \langle\langle \text{v}, w\rangle, \text{-k -wh}, \epsilon\rangle$
   (b) $T_7 \longrightarrow smerge21[T_3, T_6]$
3. (a) $T_8 = \langle\langle \text{+k i}, s\rangle, \text{-k -wh}, \epsilon\rangle$
   (b) $T_8 \longrightarrow cmerge2[T_4, T_7]$
4. (a) $T_9 = \langle\langle \text{i}, s\rangle, \epsilon, \text{-wh}\rangle$
   (b) $T_9 \longrightarrow move212[T_8]$
5. (a) $T_{10} = \langle\langle \text{+wh c}, w\rangle, \epsilon, \text{-wh}\rangle$
   (b) $T_{10} \longrightarrow cmerge1[T_5, T_9]$
6. (a) $T_{11} = \langle\langle \text{c}, w\rangle, \epsilon, \epsilon\rangle$
   (b) $T_{11} \longrightarrow move12[T_{10}]$
7. (a) S
   (b) $S \longrightarrow con[T_{11}]$

The rules above are applied in the obvious way to yield an element of category S.

terminating rules

$$L(T_1) = (\epsilon, \text{who}, \epsilon, \epsilon, \epsilon)$$
$$L(T_2) = L(T_5) = (\epsilon, \epsilon, \epsilon, \epsilon, \epsilon)$$
$$L(T_3) = (\epsilon, \text{dream}, \epsilon, \epsilon, \epsilon)$$
$$L(T_4) = (\epsilon, \text{-s}, \epsilon, \epsilon, \epsilon)$$

non-terminating rules

$$L(T_6) = L(T_1)$$
$$L(T_7) = (\epsilon, \text{dream}, \epsilon, \text{who}, \epsilon)$$
$$L(T_8) = (\epsilon, \text{dream -s}, \epsilon, \text{who}, \epsilon)$$
$$L(T_9) = L(T_{10}) = (\epsilon, \text{dream -s}, \epsilon, \epsilon, \text{who})$$
$$L(T_{11}) = (\text{who}, \text{dream -s}, \epsilon, \epsilon, \epsilon)$$
$$L(S) = L(G) = (\text{who dream -s})$$

4.5.4 *NRC-MTL* $\subseteq$ *MCFL*

In this section I prove the inclusion of the languages defined by MTGs satisfying the NRC in the languages defined by MCFGs. I show that for any NRC compliant MTG G, the MCFG G' gotten from it via the above shares its weak generative power. First, some definitions:

In $\Gamma(G)$ there are derivation trees which are not subderivations of any successful derivation. As these are irrelevant for my purposes, and only complicate matters, I define the relevant derivations of G, $R(G)$:

Definition 30 For $d \in \Gamma(G)$, $d \in R(G)$ iff

1. $ev(d) = (\alpha, \beta, \gamma) \cdot c$, for c a distinguished feature, and $\cdot \in \{::, :::\}$
2. there is some d' s.t. $ev(d') = (\alpha, \beta, \gamma) \cdot c$, and $d \in subderiv(d')$

$d \in R(G)$ is said to be in $R_n(G)$ iff $d \in \Gamma_n(G)$.

Proposition 5
If $d \in R(G)$ then each licensee feature $\texttt{-}l$ is the first feature in at most one chain $c \in ev(d)$.

Proof: Assume it were not the case. Then for some $d \in R(G)$, there are distinct chains c_1, c_2 in $ev(d)$ s.t. both begin with the same $\texttt{-}l \in licensee$. This leads immediately to the contradiction that $d \notin R(G)$: $ev(d) \neq (\alpha, \beta, \gamma)\cdot c$, as there are at least two unchecked features: $\texttt{-}l$ on c_1 and on c_2. Nor is d a subderivation of some successful derivation d', as to check either $\texttt{-}l$ feature, d' must contain a subderivation d'' of which d is a subderivation, whose root is *move*, and whose immediate proper subderivation t_1 is such that the initial chain of $ev(t_1)$ begins with $\texttt{+}l$. This cannot happen, as $ev(t_1)$ is not in the domain of *move* (it violates the SMC). □

Now we are ready to show the containment of NRC-MTGs by Multiple Context-Free Grammars. Theorem 2 shows that the language generated by a MTG G is a subset of the language generated by its MCFG counterpart. The idea is to establish that for every MTG derivation d, and value $ev(d) = (\sigma_1, \sigma_2, \sigma_3)\cdot\gamma_0, (\sigma_i)\texttt{-}l_i\gamma_i, \ldots, (\sigma_j)\texttt{-}l_j\gamma_j$, there is some MCFG derivation T with corresponding value $ev(T) = \langle \mu, s\rangle$, such that

1. $\mu = \langle\langle\gamma_0, x\rangle, \delta_1, \ldots, \delta_m\rangle$, and

2. $s = \langle \sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_i, \ldots, \sigma_j, \ldots, \sigma_{m+3} \rangle$

Proposition 6 establishes the link between μ and $ev(d)$, stating that every δ_i in μ records the features of the appropriate chain in $ev(d)$, and that every chain in $ev(d)$ has its features recorded in the appropriate position in μ. Proposition 6 also ensures that every chain has its phonetic features recorded in the appropriate position in s. Proposition 7 shows it to be the case that the only phonetic features in s are those had by $ev(d)$, by showing that only positions in s which correspond to non-empty positions in μ can be non-empty.

Proposition 6
for G an NRC-MTG, and G' its associated MCFG by the transform above, for each $d \in R(G)$, there exists some $T \in \Gamma(G')$ s.t., for $s = \langle s_1, \ldots, s_{m+3} \rangle = snd(ev(T))$, $n = \langle \hat{\mu}_0, \mu_1, \ldots, \mu_m \rangle = fst(ev(T))$,

1. the initial chain of $ev(d)$ is $(s_1, s_2, s_3) :: \mu_0$ if $a_\mu = w$ and $(s_1, s_2, s_3) ::: \mu_0$ otherwise, and
2. for each $1 \leq i \leq m$, if $\mu_i \neq \epsilon$ there is a non-initial chain c in $ev(d)$, s.t. $c = (s_{i+3})\mu_i$, and
3. for every non-initial chain $c = (\alpha)\text{-}l_i\gamma$ in $ev(d)$, $\mu_i = \text{-}l_i\gamma$ and $s_{i+3} = \alpha$.

Proof: by induction on the height of d.

basis: $d \in R_0(G)$
Then $d = ev(d)$, and is a lexical item. Wlg, let it be $(\sigma_0, \sigma_1, \sigma_2) :: \gamma$. By definition, $T = \langle\langle\gamma, w\rangle, \epsilon, \ldots, \epsilon\rangle \longrightarrow \langle\sigma_0, \sigma_1, \sigma_2, \epsilon, \ldots, \epsilon\rangle[] \in R$, and $s = snd(ev(T)) = \langle\sigma_0, \sigma_1, \sigma_2, \epsilon, \ldots, \epsilon\rangle$.

induction: assume for $d \in R_{n-1}(G)$ the claim holds. Show for $d \in R_n(G)$ the same holds.
There are six cases, corresponding to the six cases of the generating functions. I set out only one in detail, the other cases are not interestingly different...
$d = smerge2(d', d'')$
$ev(d') = (\alpha, \beta, \gamma) \cdot x\text{=}\ \rho, \phi_1, \ldots, \phi_n$
$ev(d'') = (\zeta, \eta, \theta) \cdot\cdot x\text{-}l_i\delta, \psi_1, \ldots, \psi_k$
$ev(d) = (\alpha, \beta, \gamma) \cdot \rho, \phi_1, \ldots, \phi_n, (\zeta\eta\theta) \cdot \text{-}l_i\delta, \psi_1, \ldots, \psi_k$

By the induction hypothesis, $\langle ev(d'), d' \rangle$ and $\langle ev(d''), d'' \rangle$ correspond to $\langle s', T' \rangle$ and $\langle s'', T'' \rangle$ respectively. Then d corresponds to $T = n \longrightarrow smerge2i[n', n''](T', T'')$. Here $n' = \langle \hat{\mu}_0, \mu_1, \ldots, \mu_m \rangle = fst(ev(T'))$ and $n'' = \langle \hat{\nu}_0, \nu_1, \ldots, \nu_m \rangle = fst(ev(T''))$, where $\hat{\mu}_0 = \langle x\text{=}\,\rho, a_\mu \rangle$, $\hat{\nu}_0 = \langle x\text{-}l\delta, a_\nu \rangle$, and $n = \langle \hat{\kappa}_0, \kappa_1, \ldots, \kappa_{i-1}, \text{-}l_i\delta, \kappa_{i+1}, \ldots,$
$\kappa_m \rangle$, where $\kappa_0 = \langle \rho, a_\mu \rangle$ and for $1 \leq j \leq m$, $\kappa_j = \mu_j$ if $\nu_j = \epsilon$, and $\kappa_j = \nu_j$ otherwise. That for no j are both ν_j and μ_j not equal to ϵ is given by proposition 5. By looking back to part R3d in definition 29, one verifies that $T \in R$. Now,

1. By the definition of *smerge2i*, $s_1 = s'_1 = \alpha$, $s_2 = s'_2 = \beta$, and $s_3 = s'_3 = \gamma$. Thus the initial chain of $ev(d) = (s_1, s_2, s_3) \cdot \rho$.
2. Let k such that $1 \leq k \leq m$ be arbitrary. If $k = i$, then $s_{i+3} = s''_1 s''_2 s''_3 = \zeta\eta\theta$, $\kappa_i = \text{-}l_i\delta$, and we see there is a non-initial chain in $ev(d)$ such that it is $(s_{i+3})\kappa_i$. If $k \neq i$, then either $\kappa_k = \mu_k$ or $\kappa_k = \nu_k$, in both cases $(s_{i+3})\kappa_k$ is a non-initial chain in $ev(d)$ (because it is a non-initial chain in either $ev(d')$ or $ev(d'')$).
3. Let $c = (\chi)\text{-}l_j\xi$ be a non-initial chain in $ev(d)$. Either $j = i$, in which case $\chi = s_{i+3}$ and $\text{-}l_j\xi = \kappa_i$, or $j \neq i$, in which case c was a non-initial chain in either $ev(d')$ or $ev(d'')$. Then by assumption $\chi = s_{j+3}$ and $\text{-}l_j\xi = \kappa_j$.

□

Proposition 7
For G the image of some NRC-MTG G', and for every $ev(T) =$
$\langle \langle \hat{\mu}_0, \mu_1, \ldots, \mu_m \rangle, s \rangle$, s.t. $T \in \Gamma(G)$, $s_i = \epsilon$ if $\mu_{i-3} = \epsilon$, for $4 \leq i \leq m+3$

Proof: By induction on the height of the derivation tree.

base case: $T \in \Gamma_0(G)$
Then $T = \langle \langle \phi, y \rangle, \epsilon, \ldots, \epsilon \rangle \longrightarrow \langle \sigma_1, \sigma_2, \sigma_3, \epsilon, \ldots, \epsilon \rangle[]$, and
$snd(ev(T)) = \langle \sigma_1, \sigma_2, \sigma_3, \epsilon, \ldots, \epsilon \rangle$ by construction.

inductive step: assume for $T \in \Gamma_{n-1}(G)$ for $4 \leq i \leq m+3$, $s_i = \epsilon$ if $\mu_{i-3} = \epsilon$. Show the same holds for $T \in \Gamma_n(G)$. As the cases are very similar, I show only one.
$T = n \longrightarrow move1i[n'](T')$. Note that n differs from n' (n' is $fst(ev(T'))$) at $\hat{\mu}_0$, and at μ_i. We are here concerned with only μ_i, which is now

ϵ. We must verify that $s_{i+3} = \epsilon$. As defined in F, $move1i$ maps any $s \in \langle\Sigma^*\rangle^{m+3}$ to that s' which differs from s in particular in that $s'_{i+3} = \epsilon$.

□

Theorem 2
for G an NRC-MTG, and G' its associated MCFG, if $\ell \in L_c(G)$ for $c \in base_G$ a distinguished feature, then $\ell \in L(G')$.

Proof: Let $\ell \in L_c(G)$, and let $d \in \Gamma(G)$ be a derivation of $(\sigma_0, \sigma_1, \sigma_2) \cdot c$, s.t. $\sigma_0\sigma_1\sigma_2 = \ell$. As per proposition 6, there is a $T \in \Gamma(G')$ s.t. $ev(T) = \langle n, s\rangle$, where $n = \langle\hat{\mu}_0, \mu_1, \ldots, \mu_m\rangle$, with $\hat{\mu}_0 = \langle c, y\rangle$, and $1 \leq i \leq m$, $\mu_i = \epsilon$. Thus n is in the domain of $S \longrightarrow con[n]$, hence $S \longrightarrow con[n](T) \in \Gamma(G')$, and $ev(S \longrightarrow con[n](T)) = \langle S, con(s)\rangle$. By proposition 7, $s = \langle\tau_0, \tau_1, \tau_2, \epsilon, \ldots, \epsilon\rangle$. By proposition 6, $\tau_0 = \sigma_0, \tau_1 = \sigma_1, \tau_2 = \sigma_2$, so $con(s) = \ell$, and thus $\ell \in L(G')$
□

Theorem 3 shows that the language generated by the MCFG counterpart of a MTG is a subset of that MTG. The general strategy is the same as the one used in proving theorem 2 - proposition 8 below is the mirror image of proposition 6. Proposition 7 applies 'as is'.

Proposition 8
for G an NRC-MTG, and G' its associated MCFG, for each $T \in \Gamma(G')$, there exists some $d \in \Gamma(G)$ s.t. for $s = snd(ev(T))$, and $n = \langle\hat{\mu}_0, \mu_1, \ldots, \mu_n\rangle = fst(ev(T))$, such that

1. the initial chain of $ev(d)$ is $(s_1, s_2, s_3) \cdot \mu_0$
2. for each non-initial chain c in $ev(d)$, there is an i s.t. $c = (s_{i+2})\mu_i$, and
3. for each $1 \leq i \leq m$, if $\mu_i \neq \epsilon$, then for some non-initial chain c in $ev(d)$, $c = (s_{i+2})\mu_i$

Proof: by induction on the height of T.

basis: $T \in \Gamma_0(G')$
Then $T = n \longrightarrow \langle\sigma_0, \sigma_1, \sigma_2, \epsilon, \ldots, \epsilon\rangle$, where $n = \langle\langle\phi, y\rangle, \epsilon, \ldots, \epsilon\rangle$, and by construction there is some $\ell = (\sigma_1, \sigma_2, \sigma_3) \cdot \phi \in Lex_G$, and thus also in $\Gamma_0(G)$.

induction: assume that the claim holds for every $T \in \Gamma_{n-1}(G')$. Show for $T \in \Gamma_n(G)$ the same holds.

There are six cases, corresponding to the six cases of the generating functions. I set out only one in detail, the other cases are not interestingly different...

Let $T = n \longrightarrow move1i[n'](T')$. Then $ev(T') = \langle n', s' \rangle$, and $ev(T) = \langle n, move1i(s') \rangle$, where $n' = \langle \langle +l_i\gamma, y \rangle, \mu_1, \ldots, \mu_{i-1}, -l_i, \mu_{i+1}, \ldots, \mu_m \rangle$, $n = \langle \langle \gamma, y \rangle, \mu_1, \ldots, \mu_{i-1}, \epsilon, \mu_{i+1}, \ldots, \mu_m \rangle$, $s' = \langle \sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{i+3}, \ldots, \sigma_{m+3} \rangle$, and $move1i(s') = \langle \sigma_{i+3}\sigma_1, \sigma_2, \sigma_3, \ldots, \epsilon, \ldots, \sigma_{m+3} \rangle$. By hypothesis, $d' \in \Gamma(G)$ corresponds to T', whence $ev(d') = (\sigma_1, \sigma_2, \sigma_3) \cdot +l_i\gamma, c_1, \ldots, c_k$, such that for each non-initial chain c_j, there is some p, $1 \leq p \leq m$, s.t. $c_j = (\sigma_{p+3})\mu_p$, and that every $\mu_p \neq \epsilon$ is related suchly to some c_j. In particular, μ_i corresponds to some $c_f = (\sigma_{i+3})-l_i$. Thus, $d' \in dom(move1)$. By definition of $move1$, $move1(d') = (\sigma_{i+3}\sigma_1, \sigma_2, \sigma_3) \cdot \gamma, c_1, \ldots, c_{f-1}, c_{f+1}, \ldots, c_k$. We see immediately that for every $c_j \in move1(d')$, it is related to the same μ_p as before (the only μ_p that changed was μ_i, which now equals ϵ), and, likewise, to every $\mu_p \neq \epsilon$ in n, there corresponds the same c_j as before (only c_f was deleted).

□

Theorem 3
for G an NRC-MTG, and G' its associated MCFG, if $\ell \in L(G')$ then $\ell \in L_c(G)$, for $c \in base_G$ the distinguished feature.

Proof: Let $\ell \in L(G')$. Then $\langle S, \ell \rangle \in Cl(Lex, \mathcal{F}')$, and so there is some $S \longrightarrow con[n](T) \in \Gamma(G')$, where $n = fst(ev(T)) = \langle \langle c, y \rangle, \epsilon, \ldots, \epsilon \rangle$. By proposition 7, $snd(ev(T)) = \langle \sigma_1, \sigma_2, \sigma_3, \epsilon, \ldots, \epsilon \rangle$, and $\ell = con(s) = \sigma_1\sigma_2\sigma_3$. By proposition 8 there is some $d \in \Gamma(G)$ s.t. $ev(d) = (\sigma_1, \sigma_2, \sigma_3) \cdot c$. Thus, $\sigma_1\sigma_2\sigma_3 = \ell \in L_c(G)$. □

Corollary 1
for G an NRC-MTG, G' its associated MCFG, and $c \in base_G$ a distinguished feature, $L(G') = L_c(G)$.

Proof: By Theorem 1, if $\ell \in L_c(G)$ then $\ell \in L(G')$. By Theorem 2, if $\ell \in L(G')$ then $\ell \in L_c(G)$. Thus $\ell \in L(G')$ iff $\ell \in L_c(G)$, and so $L(G') = L_c(G)$. □

4.6 The MTG Merge Fragment

In this section, the question is raised as to how much expressive power we obtain by employing a non-standard spellout function over our trees. To this end, I consider the formalism without the movement operation, as phrasal restructuring seems tangential in determining the power of the spellout function. The merge fragment of MTGs (i.e. when $\mathcal{F} = \{merge\}$, or, equivalently, when no lexical item has a movement feature) contains grammars that define non-context-free languages. For example, the grammar below defines the language $\{a^i b^j c^i d^j | i, j \in \mathbf{N}\}$ which is not context-free. It is simple to show that every context-free language (1-MCFL) is definable by a grammar in the merge fragment of MTGs, as well as that the languages definable by such MTGs are contained in the class of languages definable by 3-MCFGs. The question remains, however, as to just how powerful the merge fragment actually is. The following grammar is believed to be illustrative of context sensitive languages definable by the merge fragment of MTGs. The intuition is that one can link two grammars for $\{a^n b^n | n \in \mathbf{N}\}$ by adding a lexical item that c-selects sentences of the one, and is in turn c-selected by a 'starting' lexical item of the other. If the only strong node in the new grammar is this linking lexical item, then the MW is pronounced between the two linked sentences in reverse order, yielding a crossing dependency. The lexicon consists of the following eight expressions:

E_1.	$(\epsilon, \mathrm{b}, \epsilon)$ :: =B D= B	E_6.	$(\epsilon, \mathrm{c}, \epsilon)$ :: =Z A= C
E_2.	$(\epsilon, \mathrm{b}, \epsilon)$ :: D= B	E_7.	$(\epsilon, \mathrm{c}, \epsilon)$ :: =C A= C
E_3.	$(\epsilon, \mathrm{d}, \epsilon)$ :: D	E_8.	$(\epsilon, \mathrm{a}, \epsilon)$ :: A

E_4.	$(\epsilon, \epsilon, \epsilon)$ ::: =B Z
E_5.	$(\epsilon, \epsilon, \epsilon)$ ::: Z

$E_1 - E_3$ constitute the lower grammar, and $E_6 - E_8$ the higher. $E_4 - E_5$ are the linking lexical items. A derivation of *aabccd* follows:

1.	s-merge1(E_2, E_3)	=	E_{11}	=	$(\mathrm{d}, \mathrm{b}, \epsilon)$ :: B
2.	c-merge2(E_4, E_{11})	=	E_{12}	=	$(\epsilon, \mathrm{b}, \mathrm{d})$ ::: Z
3.	c-merge1(E_6, E_{12})	=	E_{13}	=	$(\epsilon, \mathrm{bc}, \mathrm{d})$ :: A= C
4.	s-merge1(E_{12}, E_8)	=	E_{14}	=	$(\mathrm{a}, \mathrm{bc}, \mathrm{d})$ ::: C
5.	c-merge1(E_7, E_{14})	=	E_{15}	=	$(\mathrm{a}, \mathrm{bcc}, \mathrm{d})$ ::: A= C
6.	s-merge1(E_{15}, E_3)	=	E_{16}	=	$(\mathrm{aa}, \mathrm{bcc}, \mathrm{d})$ :: C

5 Extensions

The mirror principle found in (Baker, 1985) does not require that the order of application of GF-changing processes be realized as *suffix* order on the finite verb, only that the syntactic and morphological processes proceed in tandem (with the conclusion being that they are two sides of the same coin). According to such an interpretation of the mirror principle, the syntactic complement relation might mirror morphological constituency (of which suffix order is a special case) by allowing the edge of alignment to be specified lexically. This could be implemented immediately in the chain based MTGs by adding a new kind of selection feature ($c'selection = \{\texttt{==}b | b \in base\}$), and the following pair of c′-merge rules:

$$\frac{(s,t,u) :: \texttt{==}f\gamma, \alpha_1, \ldots, \alpha_k \qquad (v,w,x) \cdot f\nu, \iota_1, \ldots, \iota_l}{(suv, tw, x) :: \gamma\nu, \alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l}\text{c}'\text{-merge1}$$

$$\frac{(s,t,u) ::: \texttt{==}f\gamma, \alpha_1, \ldots, \alpha_k \qquad (v,w,x) \cdot f\nu, \iota_1, \ldots, \iota_l}{(s, tw, uvx) ::: \gamma\nu, \alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l}\text{c}'\text{-merge2}$$

The above rules merely allow for a member of a MW to be spellt out to the right (c-merge) or to the left (c′-merge) of its complement. We can see immediately that this extension does not affect the results obtained for the NRC fragment of MTGs. However, with the above modification, it is straightforward to create a grammar for the copy language in the merge-fragment of MTGs. Thus, adding the option of prefixation or suffixation seems to give the merge fragment more similarity to TAGs, and other 2-MCFGs. To implement this extension in the tree-based version of MTGs, it is easiest to add yet another set into the tuple of a MTT, the interpretation of which is that a node is in this set iff it is ordered before its complement at spellout (it is a ‘prefixing’ node). We adjust the definitons of spellout and of c-merge accordingly.

It is a short step from allowing the syntactic complementation relation to be ambiguous between morphological left-edge alignment and right-edge alignment (rather, the syntactic complementation relation *is* the morphological constituency relation), to attempting to implement the morphological processes (e.g. reduplication, Ablaut, truncation, infixation) in the syntax. We let the phonetic part of an expression be interpreted as a function over strings, and we reinterpret the operation of concatenation in morphological words as function application. The most conservative extension of this type is to only reinterpret concatenation as application in the morphological word:

$$\frac{(s,t,u) :: \texttt{=}f\gamma, \alpha_1, \ldots, \alpha_k \qquad (v,w,x) \cdot f\nu, \iota_1, \ldots, \iota_l}{(suv, t(w), x) :: \gamma\nu, \alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l}\text{c-merge1}$$

$$\frac{(s,t,u) ::: \texttt{=}f\gamma, \alpha_1, \ldots, \alpha_k \qquad (v,w,x) \cdot f\nu, \iota_1, \ldots, \iota_l}{(s, t(w), uvx) ::: \gamma\nu, \alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l}\text{c-merge2}$$

Though less obvious, such an extension does not affect our language theoretic results even with arbitrary morphological functions,[12] though it is obvious that, with arbitrary functions, the complexity of MTGs without the NRC is not even recursively enumerable.

Pushing in a slightly different direction, we might keep the 'suffixation' interpretation of the mirror principle, but separate the morphological word-forming relation from direction of syntactic branching. Again, this could be implemented lexically with the addition of 'MW forming' selection features ($\texttt{=}\hat{f}, \hat{f}\texttt{=}$). This move pushes us further towards allowing both left- and rightward movement (somewhat analogously to pre-minimalist theories which parameterized the spec - head and head - complement orders) (with $\texttt{+}f$ for leftward and $f\texttt{+}$ for rightward movement). On a more linguistic note, one might try accounting for cliticization by allowing moved items to form morphological words with their attractors ($\texttt{-}\hat{f}$).

Finally, there have been various proposals for eliminating the copy-delete theory of movement in favour of simply a copy theory (Chomsky, 1998; Pesetsky, 2000), where 'deletion' takes place during spellout . This is similar in spirit to, but requires more machinery (in the form of chains) than, a multiple domination theory (Gärtner, 2002), where one position is chosen to be the surface position of the moved phrase. Both theories require additional complications at (say) the S-M interface (i.e. which copy gets spellt out, or which position is the surface position). However, mirror theory already has the requisite machinery, and thus can replace movement with multiple domination at no extra cost. The strength features used in determing the

[12]MTGs meeting the NRC only allow finitely many combinations of functions (i.e. only provide each function with a finite number of possible inputs). If every possible morphological word (functional output) is represented by a unique sequence of lexical items, then the standard MTG can simulate the function-MTG.

spellout positions of MWs might here be made to serve double duty - the spellout position of a chain (or multiply dominated phrase) is in the specifier of the highest strong node immediately dominating it. This modification of the theory presented herein might have interesting consequences for learning as the learner can bring additional evidence to bear regarding the strength features of lexical items.

References

Abels, Klaus. 2001. Move? ms. University of Connecticut, Storrs.

Baker, Mark. 1985. The mirror principle and morphosyntactic explanation. *Linguistic Inquiry*, 16:373–416.

Baker, Mark. 1988. *Incorporation: a theory of grammatical function changing*. MIT Press, Cambridge, Massachusetts.

Brody, Michael. 1997. Mirror theory. ms. University College London.

Brody, Michael. 2000. On the status of representations and derivations. ms. University College London.

Chomsky, Noam. 1995. *The Minimalist Program*. MIT Press, Cambridge, Massachusetts.

Chomsky, Noam. 1998. Minimalist inquiries: the framework. MIT. Forthcoming.

Gärtner, Hans-Martin. 2002. *Generalized Transformations and Beyond: Reflections on Minimalist Syntax*. Akademie Verlag, Berlin.

Gerdemann, Dale. 1994. Parsing as tree traversal. In *Proceedings of COLING 94, vol. I*, pages 396–400.

Hale, John and Edward P. Stabler. 2001. Notes on unique readability. ms. UCLA.

Harkema, H. 2001. A Characterization of Minimalist Grammars. In P. de Groote, G.F. Morrill, and C. Retoré, editors, *Logical Aspects of Computational Linguistics (LACL 2001)*, volume 2099 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, Berlin, Heidelberg, Germany.

Joshi, Aravind. 1985. How much context-sensitivity is necessary for characterizing structural descriptions. In D. Dowty, L. Karttunen, and A. Zwicky, editors, *Natural Language Processing: Theoretical, Computational and Psychological Perspectives*. Cambridge University Press, NY, pages 206–250.

Keenan, Edward L. and Edward P. Stabler. forthcoming. *Bare Grammar*. CSLI Publications, Stanford University. Cambridge University Press, NY.

Kobele, Gregory M., Travis Collier, Charles Taylor, and Edward P. Stabler. 2002. Learning mirror theory. In *Proceedings of the Sixth International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+6),* Venezia.

Kobele, Gregory M. and Jason Kandybowicz. 2001. A normal form theorem for minimalist grammars. ms. UCLA.

Michaelis, J. 1998. Derivational Minimalism is Mildly Context-Sensitive. In M. Moortgat, editor, *Logical Aspects of Computational Linguistics, (LACL '98)*, volume 2014 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, Berlin, Heidelberg, Germany.

Michaelis, J. 2001. Transforming Linear Context-Free Rewriting Systems into Minimalist Grammars. In P. de Groote, G.F. Morrill, and C. Retoré, editors, *Logical Aspects of Computational Linguistics (LACL 2001)*, volume 2099 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, Berlin, Heidelberg, Germany.

Michaelis, Jens and Marcus Kracht. 1997. Semilinearity as a syntactic invariant. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*, pages 37–40, NY. Springer-Verlag (Lecture Notes in Computer Science 1328).

Pesetsky, David. 2000. *Phrasal movement and its kin*. MIT Press, Cambridge, Massachusetts.

Seki, Hiroyuki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. 1991. On multiple context-free grammars. *Theoretical Computer Science*, 88:191–229.

Stabler, Edward P. 1994. The finite connectivity of linguistic structure. In C. Clifton, L. Frazier, and K. Rayner, editors, *Perspectives on Sentence Processing*. Lawrence Erlbaum, NJ, pages 303–336.

Stabler, Edward P. 1997. Derivational minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*. Springer-Verlag (Lecture Notes in Computer Science 1328), NY, pages 68–95.

Stabler, Edward P. 2001. Recognizing head movement. In Philippe de Groote, Glyn Morrill, and Christian Retoré, editors, *Logical Aspects of Computational Linguistics*, Lecture Notes in Artificial Intelligence, No. 2099. Springer, NY, pages 254–260.

Stabler, Edward P. and Edward L. Keenan. 2000. Structural similarity. In A. Nijholt, G. Scollo, T. Rus, and D. Heylen, editors, *Algebraic Methods in Language Processing, AMiLP 2000*, University of Iowa. Revised version forthcoming in *Theoretical Computer Science*.

Travis, Lisa. 1984. *Parameters and effects of word order variation*. Ph.D. thesis, Massachussets Institute of Technology, Cambridge, Massachusetts.