# Distributed Morphology over Strings

Marina Ermolaeva and Daniel Edmiston
University of Chicago

**Introduction:** This research constitutes a fully formalized theory of Morphology by adapting the Distributed Morphology (DM) framework to work over strings. That the morphological module should operate over strings is desirable, since it is assumed that most (arguably all) morphological processes can be modelled with (i.e. have the weak-generative capacity of) regular languages. As is, DM is typically depicted as operating on trees, meaning its strong-generative capacity is above regular. We therefore claim that the generating mechanism of standard DM is too powerful. By constraining DM to operating on strings, we restrict the strong-generative capacity of the morphological module to that of regular languages, providing an immediate explanation for the "regular"ity of morphological phenomena in natural language.

Assuming the standard Y-model, the fact that standard DM operates on trees assumes that the "flattening" of the derivation for PF takes place post morphology. We push this flattening of the structure to above the morphological module, between the syntax and morphology.

**Distributed Morphology over Strings:** Following Trommer (1999), we assume that the basic unit of syntactic computation is a *feature structure* (FS), and that sequences of FSs act as inputs for morphology. We use Minimalist Grammars (Stabler 1997) as our syntactic framework, importing the notion of *node strength* from Mirror Theory (Brody 1997, Kobele 2002) to implement Head Movement and Lowering as mechanisms for constructing morphological words. An FS $F$ is defined as a pair $\langle M, E\rangle$, where $M = feat(F)$ is a subset of some finite set of features (including syntactic category labels) and $E = exp(F) \in (\Sigma \cup \{None, \epsilon\})$, where $\Sigma$ is a finite set of phonemes. We take $FS_{Syn}$ to be the (finite) set of FSs output by the syntactic component of the grammar; the exponent of each FS in $FS_{Syn}$ is assumed to be the placeholder value *None*–to be replaced by Vocabulary Insertion (VI).

As in standard DM, our operations are context-dependent. Context-dependent *rewriting rules* have the form $A \rightarrow B/C\_\_D$, where $A$, $B$, $C$, $D$ are regular expressions over some alphabet. With a few caveats, such rules (and ordered finite sequences thereof) have been shown to define regular relations on strings (Kaplan & Kay 1994). Assuming that morphology is regular, it should be possible to state morphological rules in a way compatible with this format and represent them with a finite-state transducer. We take $A$, $B$ to be sequences of FSs and $C$, $D$ to be regular expressions over FSs. As a proof of concept, we define VI and Readjustment as follows. VI rewrites a single node as a sequence of FSs which can be thought of as *morphophonemes* and can be further manipulated by Readjustment rules:

(1) A rule $r$ of the form $A \rightarrow B/C\_\_D$ is a *VI* rule iff:
$|A| = 1$, and $|B| \geq 1$;
$exp(A_i) = None$ for $1 \leq i \leq |A|$, and $exp(B_j) \neq None$ for $1 \leq j \leq |B|$;
$\bigcup_{1\leq i\leq|A|} feat(A_i) = \bigcup_{1\leq j\leq|B|} feat(B_j)$; ($r$ is *feature-preserving*)
$feat(A_1) = ... = feat(A_{|A|}) = feat(B_1) = ... = feat(B_{|B|})$. ($r$ is *set-preserving*)

(2) A rule $r$ of the form $A \rightarrow B/C\_\_D$ is a *Readjustment* rule iff:
$exp(A_i) \neq None$ for $1 \leq i \leq |A|$, and $exp(B_j) \neq None$ for $1 \leq j \leq |B|$;
$r$ is feature-preserving and set-preserving.

Importantly, finite-state machines operate on strings of symbols drawn from some finite alphabet; in such an implementation, FSs have to be treated as atoms. In other words, underspecification notation in terms of feature bundles generally used to state rewriting rules serves as an "abbreviatory convention" (Kaplan & Kay 1994:350) and needs to be translated into the language of unanalyzable segments. In our system, this alphabet (the complete set of fully specified FSs) is guaranteed to be finite, as the sets of features and exponents are finite by definition. Each underspecified rule can then be converted (via a well-defined procedure) into a set of *instances*–fully explicit regular relations over the alphabet of FSs. For instance, assuming the set of morphological features encoding English verbal inflection {±PAST, ±FIN(ITE)}, the root alternation in *take/took* can be captured via a combination of VI and Readjustment (3), represented as the regular rewriting rules in (4). We use $\langle M, ...\rangle$ as shorthand for $F_1|...|F_n$, where $\{F_1, ..., F_n\}$ is the set of all FSs with the feature bundle $M$, regardless of the exponent:

(3) a. [TAKE] → teɪk

b. eɪ → ʊ / X__Y [+PAST, +FIN], where X__Y ∈ {SHAKE, TAKE, ...}

(4) a. $\left\langle \begin{matrix}\{\text{TAKE}\},\\ \textit{None}\end{matrix}\right\rangle \rightarrow \left\langle \begin{matrix}\{\text{TAKE}\},\\ \text{t}\end{matrix}\right\rangle \left\langle \begin{matrix}\{\text{TAKE}\},\\ \text{eɪ}\end{matrix}\right\rangle \left\langle \begin{matrix}\{\text{TAKE}\},\\ \text{k}\end{matrix}\right\rangle$

b. $\left\langle \begin{matrix}\{\text{TAKE}\},\\ \text{eɪ}\end{matrix}\right\rangle \rightarrow \left\langle \begin{matrix}\{\text{TAKE}\},\\ \text{ʊ}\end{matrix}\right\rangle / \_\left[\left\langle \begin{matrix}\{\text{TAKE}\},\\ ...\end{matrix}\right\rangle\right]^* \left\langle \begin{matrix}\{+\text{PAST}, +\text{FIN}\},\\ ...\end{matrix}\right\rangle$

The regular relations encoding rules are then combined via composition into a single regular relation, following the procedure laid out in Kaplan & Kay 1994.

Composition requires all rules to be ordered, whereas rule ordering in standard DM is underspecified. This requirement is actually a blessing, as with fully instantiated rule-orderings we can mimic cyclicity effects which usually rely on the tree notion of embedding. Since syntactic category labels are accessible for morphology, restrictions on phonologically conditioned allomorphy can be handled by stating that the order of VI rules mirrors the order of syntactic projections. For example, with English DPs of the form $[_{DP}D[_{NP}N]]$ we can capture the *a/an* allomorphy by having rules that apply to FSs containing N precede rules that target FSs containing D. In other words, cyclicity effects–run from most embedded to least embedded on trees–can be simulated on strings by establishing feeding and counter-feeding relations between rules.

**Discussion:** Representing DM as described above allows us to capture formally certain intuitions about the different operations. First, using the notion of a feature-preserving rule allows us to capture the intuition that Impoverishment is more "powerful" than other operations (Harley 2008). Furthermore, explicit mention of the bipartite feature structure allows us to fully distinguish between pure morphology and morphologically-conditioned phonology. Pre-VI operations (e.g. fission) operate on morphosyntactic features in $feat(F)$ (i.e. are not set-preserving), while VI and operations afterwards (e.g. Readjustment) operate only on $exp(F)$ in any FS $F$, and as such are set-preserving.

**Conclusion:** Above, we defined a portion of DM over strings and gave an example as a proof of concept. We also introduced how cyclicity effects over trees can be modeled with strings by rule orderings, one example where the added generative capacity of trees can be done away with. Assuming that we can capture the same empirical coverage running DM over strings as is caught by DM over trees, it is our contention that DM should be run on strings, as it serves as a direct explanation for the robust generalization that morphology appears to be regular, and allows for efficient parsing and generation of surface forms (Kaplan & Kay 1994).

**References:** BRODY, M. (1997). Mirror Theory. HARLEY, H. (2008). When is a syncretism more than a syncretism? Impoverishment, metasyncretism, and underspecification. KAPLAN, R. M., & KAY, M. (1994). Regular models of phonological rule systems. KOBELE, G. M. (2002). Formalizing Mirror Theory. STABLER, E. P. (1997). Derivational minimalism. TROMMER, J. (1999). Morphology consuming syntax' resources: Generation and parsing in a minimalist version of Distributed Morphology.