DM on Strings

Marina Ermolaeva
Daniel Edmiston

February 10, 2017

Introduction

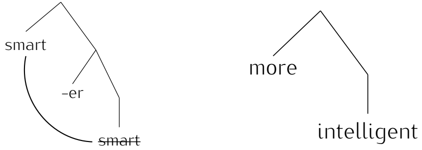
- **What this project is:**
  - Formalization of Distributed Morphology (DM) and the syntax-morphology interface
    - **What we do:**
    - Test a core assumption in DM
    - Probe linearization's place in the grammar
    - Solidify intuitions re: nature of operations

- **What this project is not:**
- Advocacy for a particular proposal
    - **What we don't do:**
    - Give in-depth, data-driven analysis

Introduction

- DM's core assumption: Syntax all the way down
  i.e. Morphology over binary trees

(1) e.g. English comparative

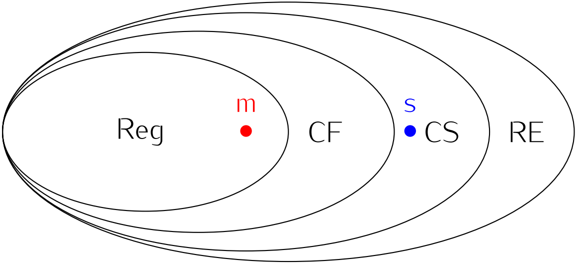


- '*smarter*' vs. '*more intelligent*': derived by morphology over *syntax*-like structure

Introduction

- BUT: Expressiveness of morphology (Karttunen et al. 1992, a.o.) vs. syntax (Shieber 1985) very different

(2) Morphology vs. Syntax



- If morphology is regular, strings should be sufficient! No need for trees!

But that means...

- If morphology runs on strings:
  (i) Linearization occurs BEFORE Morphology!
  (ii) DM needs to be recast

- This project is that recasting

- We propose the following architecture:
  1. Syntax: derivation over feature structures (FS)
  2. Linearization: strings of FSs
  3. Morphology: Finite-state transducer–strings of FSs to phonological information

Outline

Introducing DM

- Theoretical framework for Morphology (Halle & Marantz 1993)
- Morphology's role: Transfer information from syntax to phonology
- Morphology: between spell-out and PF

(3) Morphology's place in the Y-model

$$\begin{tikzcd}
 & \text{Syntactic derivation} \arrow[d] & \\
 & \text{Spell-out} \arrow[dl, "\textbf{Morph}"'] \arrow[dr] & \\
\text{PF} & & \text{LF}
\end{tikzcd}$$

Introducing DM

- Input to morphology is output from syntax
- Output of morphology is input to phonology
- Architecture of morphology: several (partially) ordered operations
    - **Vocabulary Insertion**
    - **Fusion/Fission**
    - **Impoverishment/Obliteration**
    - **Readjustment**
    - **Local Dislocation**

Vocabulary Insertion

- Vocabulary Insertion (VI)–DM's flagship operation
- Introduces phonological information to derivation

(4) Instance of VI: derivation of *worked*

[WORK] [+PAST] → 'work' 'ed'

Vocabulary Insertion

- VI: swaps morphosyntactic features for phonological info
- Operates on (potentially context-dependent) VI rules

(5) VI rules deriving *worked*

a. [WORK] → 'work'

b. [+PAST] → 'ed'

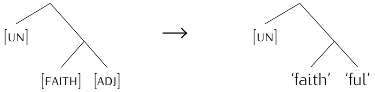
- Operates cyclically (inside-out)

Vocabulary Insertion

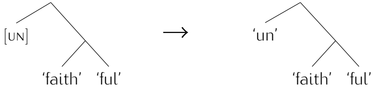
- Context-dependency + cyclicity

(6) Derivation for *unfaithful*

a.



b.



(7) VI rules deriving *unfaithful*

a. [UN] → 'un'

b. [FAITH] → 'faith'

c. [ADJ] → 'ful' / {[FAITH], [FRUIT], ...} ___

Fusion

- Fusion–Pre-VI operation
- Targets certain pairs of feature-sets

(8) *'worse'* derived by Fusion

[BAD] [CMPR] → 'worse'

- VI operates as usual on fused feature set

(9) [BAD, CMPR] → 'worse'

Readjustment

- Readjustment: Post-VI operation
- Change phonological information in certain contexts

(10) Readjustment rule for English ablaut

a. /ei/ → /ɯ/ / X __ Y[PAST], X = $\sqrt{\text{SHAKE}}$, $\sqrt{\text{TAKE}}$, ...

- Captures changes in roots–but not true suppletion

Local Dislocation

- "Movement after syntax" (Embick & Noyer 2001)

(11) Input: $(_{Conjunct1}$ X Y) **-que** $(_{Conjunct2}$ W Z)
Surface: $(_{Conjunct1}$ X Y) t $(_{Conjunct2}$ W-**que** Z)

- Sensitive to (morpho)phonology

(12) a. circum-que ea loca
around-and those places
'and around those places'

b. in rēbus-que
in things-and
'and in things'

Regular Languages

- Hypothesis: Morphology can be described with regular relations (e.g. Koskenniemmi 1983)–we adapt methods from phonology

- Our follow up hypothesis: Strings are sufficient for morphology (Kaplan & Kay 1994)

- What are strings? What are regular languages/relations?

Regular Grammar

- Regular languages–describable by regular grammars

(13) Regular grammar G_R

$$N = \{S, A\}$$
$$\Sigma = \{a\}$$
$$P = \{\ S \to aA,\ A \to aA,\ A \to \epsilon\ \}$$
$$S \in N$$

(14) Derivation of aaaa by G_R

i $S \to aA$

ii $A \to aA$ (giving aaA)

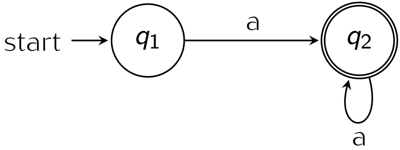
iii $A \to aA$ (giving aaaA)

iv $A \to a$ (giving aaaa)

Representing Regular Languages

- Different ways to represent RLs
- Regular expressions:
  - a^+
  - $a^*(b)c^*$
  - $a\backslash b$ c
  - ...

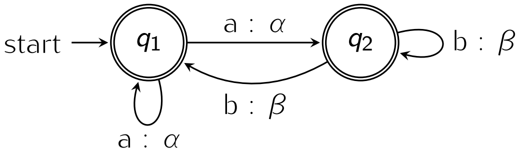
(15) FSA for a^+



Representing Regular Relations

- FST: representation of regular relation

(16) Roman → Greek FST



- Claim: Syntax/Morphology interface is (describable by) an FST

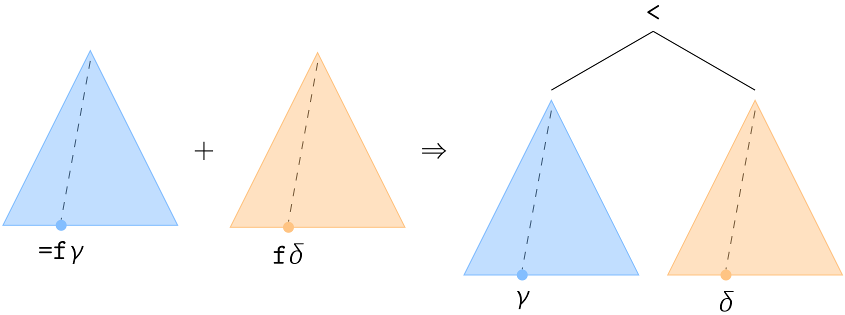
Minimalist Grammars

- A set of syntactic features:

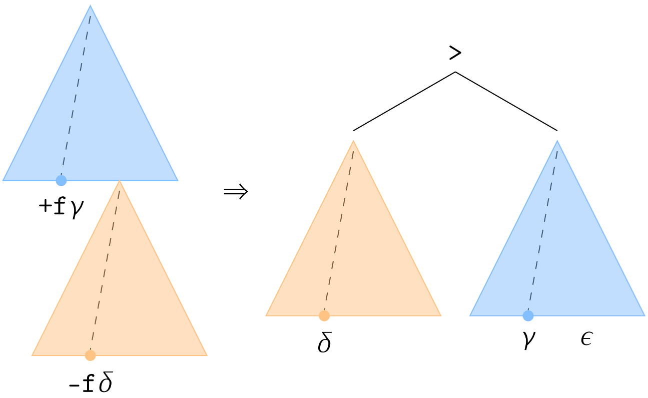
$$
\begin{array}{lll}
F = & \textit{Base} \cup & \text{(category features)} \\
 & \{\texttt{=f} \mid \texttt{f} \in \textit{Base}\} \cup \{\texttt{f=} \mid \texttt{f} \in \textit{Base}\} \cup & \text{(selectors)} \\
 & \{\texttt{+f} \mid \texttt{f} \in \textit{Base}\} \cup & \text{(licensors)} \\
 & \{\texttt{-f} \mid \texttt{f} \in \textit{Base}\} & \text{(licensees)}
\end{array}
$$

- A set of lexical items:
  $Lex \subset \Sigma^* \times F^*$,
  where Σ is a set of phonological units
- Two generating functions: **merge** and **move**

Merge



Move



Assembling morphological words

- Head movement: $[_{XP} X^0 \ldots [_{YP} \ldots Y^0 \ldots]] \rightarrow [_{XP} Y^0\text{–}X^0 \ldots [_{YP} \ldots]]$
- Lowering: $[_{XP} X^0 \ldots [_{YP} \ldots Y^0 \ldots]] \rightarrow [_{XP} \ldots [_{YP} Y^0\text{–}X^0 \ldots]]$
  - Embick & Noyer 2001: Lowering only applies after all syntactic movement.
- Mirror Theory (Brody 1997, Kobele 2002): *strong* and *weak* nodes.

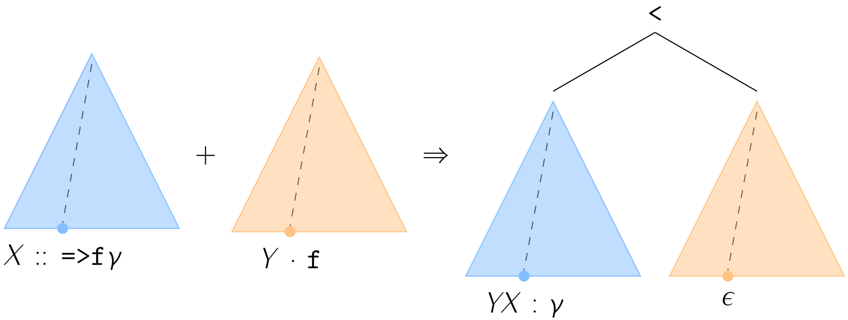
  `=f` (normal **merge**)

  `=>f` (strong node; **merge** with Head Movement)

  `<=f` (weak node; **merge** with Lowering)
- Syntax inserts *boundary symbols* (#) between morphological words.

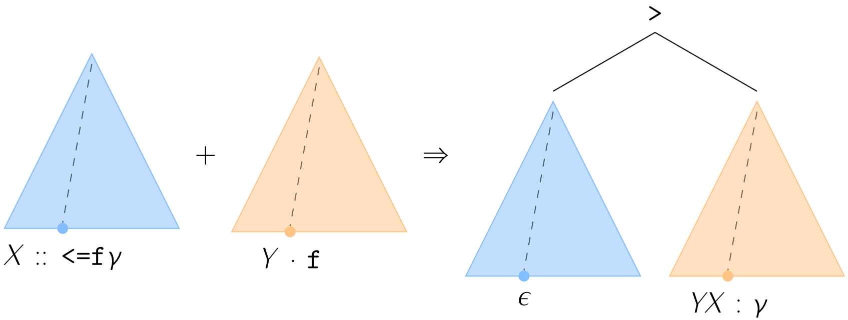
Head Movement

- The selecting expression must be nonderived;
- The selected expression must be a non-mover.



Lowering

- The selecting expression must be nonderived;
- The selected expression must be a non-mover.



MGs over feature structures

- Let Σ be a finite set of phonological units and M a finite set of (privative) features;
- Then the set of *feature structures* $FS = \mathcal{P}(M) \times (\Sigma \cup \{\epsilon, None\})$, where $None$ is the empty exponent;
- Let $fs = \langle x, y \rangle \in FS$. We define $feat(fs) = x$ and $exp(fs) = y$;
- $Lex \subset \{fs \mid fs \in FS \,\&\, exp(fs) = None\} \times F^*$.

John :: `d -k`	$\langle \{\mathrm{D}, \textsc{john}, 3, \textsc{sg}\}, None \rangle$ :: `d -k`
walk :: `=d v`	$\langle \{\mathrm{V}, \textsc{walk}\}, None \rangle$ :: `=d v`
-s :: `<=v +k t`	$\langle \{\mathrm{T}, \textsc{prs}, 3, \textsc{sg}\}, None \rangle$ :: `<=v +k t`

Underspecification and shorthand

- Morphological rules operate on *underspecified* feature structures:

  $FS_U = \mathcal{P}(M) \times (\Sigma \cup \{\epsilon, \mathit{None}, \ldots\})$,

  where … stands for "any exponent".
- When discussing specific examples, we will sometimes use *informal* shorthand:

  $\langle$XYZ$\rangle$, where XYZ is whatever information is needed to identify a set of feature structures in a given context.

Rewrite rules

- Morphological rules are of the form $A \rightarrow B \ / \ C\_\_D$

  such that $A = A_1, \ldots, A_m$ and $B = B_1, \ldots, B_n$ are sequences of underspecified feature structures, and
  C, D are regular expressions over $FS_U \cup \{\#\}$;

- Let r be such a rule.

  r is *purely morphological* iff
  $exp(A_1) = \ldots = exp(A_m) = exp(B_1) = \ldots = exp(B_n) = None$;

  r is *feature-preserving* iff
  $\bigcup_{i=1}^{m} feat(A_i) = \bigcup_{j=1}^{n} feat(B_j)$;

  r is *set-preserving* iff
  $feat(A_1) = \ldots = feat(A_m) = Feat(B_1) = \ldots = Feat(B_n)$

Rule classes

- A rule r of the form $A \rightarrow B \ / \ C\_\_D$ is...

  ... a *fusion rule*, iff $|A| = 2$, $|B| = 1$, and r is feature-preserving and purely morphological;

  ... a *fission rule*, iff $|A| = 1$, $|B| = 2$, and r is feature-preserving and purely morphological;

  ... an *impoverishment rule*, iff $|A| = |B| = 1$, and $feat(B_1) \subset feat(A_1)$, and r is purely morphological.
- Captures insight that impoverishment is different from other operations!

Rule classes

- Bobaljik 2015: morphosyntactic features are *rewritten* by VI;
- For now: add phonological material, keep morphosyntactic features.
- A rule r of the form $A \to B \; / \; C\_\_D$ is...

  ... a *VI rule*, iff $|A| = 1$, $|B| \geq 1$, $exp(A_1) = None$, $exp(B_j) \neq None$ for $1 \leq j \leq |B|$, and r is set-preserving;

  ... a *readjustment* rule, iff $exp(A_i) \neq None$ for $1 \leq i \leq |A|$, $exp(B_j) \neq None$ for $1 \leq j \leq |B|$, and r is set-preserving;

Instances

- DM rules (and our feature structures) are *underspecified*;
- FSTs can only operate over unanalyzable elements;
- We need to collect all instantiations of each rule.
- Let $fs \in FS_U$.
  Then $inst(fs) = \{x \mid x \in FS \;\&\; feat(x) \supseteq feat(fs) \;\&\; (exp(x) = exp(fs) \text{ or } exp(fs) = \ldots)\}$;
- Let X be a regular expression over underspecified feature structures.
  Then $inst(X) = X[x_1 \mapsto \bigcup inst(x_1), \ldots, x_n \mapsto \bigcup inst(x_n)]$, where $\{x_1, \ldots, x_n\}$ is the set of all feature structures in X.

Building a transducer

- Assume that our grammar is a *sequence* of rules $r_1, \ldots, r_k$.
- Let r be the rewrite rule $A \rightarrow B \ / \ C\_\_D$

  such that $A = A_1, \ldots, A_m \in FS_U$ and $B = B_1, \ldots, B_n \in FS_U$;
- Then $batch(r)$ is the set of all rules $a \rightarrow b \ / \ inst(C)\_\_inst(D)$ where $a = a_1, \ldots, a_m \in FS$ and $b = b_1, \ldots, b_n \in FS$

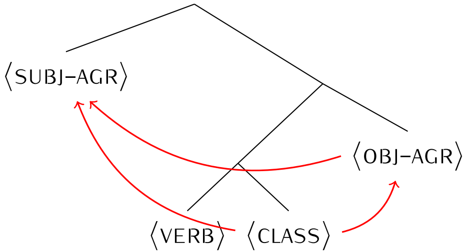
  such that $a_i \in inst(A_i)$ for $1 \leq i \leq m$, and
  $feat(b_j) = feat(B_j) \cup (\bigcup_{i=1}^{m} feat(a_i) \setminus \bigcup_{i=1}^{m} feat(A_i))$, and
  $exp(b_j) = exp(B_j)$ for $1 \leq j \leq n$.
- Kaplan & Kay 1994:
  - left-to-right, right-to-left, or simultaneous application of single rules;
  - simultaneous application of a rule set as *batch rules*;
  - ordered rules as *transducer composition*.

Cyclicity

- Bobaljik 2000: VI proceeds cyclically from the root outwards, *deleting* features it expresses;
    - Outwards-sensitive allomorphy is conditioned only by morphosyntactic features;
    - Inwards-sensitive morphology is conditioned only by morphophonological features.
- Counter-examples both ways: Deal & Wolf 2013, Gribanova & Harizanov 2015;
- VI rules need to be ordered, but not necessarily by depth of embedding.
- Which of these effects can be modelled by regular relations on strings?

Rule ordering: agreement in Itelmen

- Bobaljik 2000: outwards-sensitive allomorphy.



- Order VI rules by the category of the nodes they apply to:

  VERB > CLASS > OBJ–AGR > SUBJ–AGR

Local Dislocation: clitics in Latin

- Embick & Noyer 2001: a head can interact with a linearly adjacent element in its complement.

(17) a. circum-que ea loca
around-and those places
'and around those places'

b. in rēbus-que
in things-and
'and in things'

Local Dislocation: clitics in Latin

- Input to morphology:

  $\#\langle \textsc{que}\rangle\#\langle \textsc{circum}\rangle\#\langle \textsc{ea}\rangle\#\langle \textsc{loca}\rangle\#$

  $\#\langle \textsc{que}\rangle\#\langle \textsc{in}\rangle\#\langle \textsc{rēbus}\rangle\#$
- Rule ordering: N > D > P > Conj
- "Light" prepositions form a unit with the adjacent word:

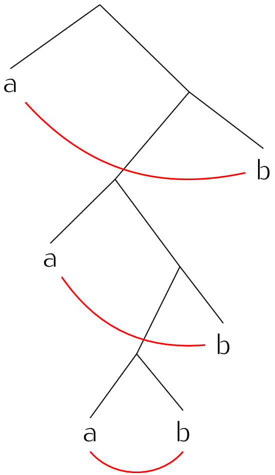
  $\langle \textsc{in}\rangle \rightarrow \textit{in}\star$

  $\star\# \rightarrow \emptyset$
- Displacement via copy and deletion:

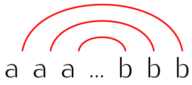
  $\emptyset \rightarrow \langle \textsc{que}\rangle \; / \; \#\langle \textsc{que}\rangle\#(\backslash\#)^{*}\_\_\#$

  $\langle \textsc{que}\rangle \rightarrow \emptyset \; / \; \#\_\_\#$

Morphology from another dimension

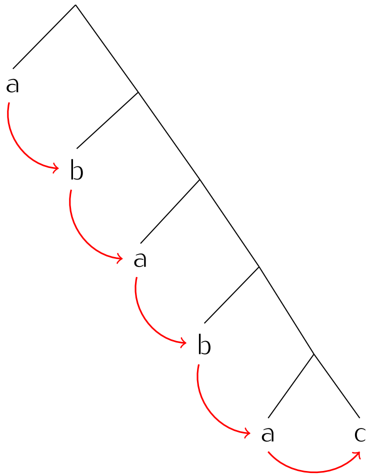


- Unbounded nested dependencies:



- Incompatible with the core assumption;
- Any natural language examples?

Unfinished business



- Rules depending on each other's output:

  $\langle a \rangle \rightarrow a \; / \; \_\_ \; (b \mid c)$

  $\langle b \rangle \rightarrow b \; / \; \_\_ \; a$

- Impossible to order the rules...
- ... or apply them as a batch.

English examples

(18) Derivation of *John walks*–VI only

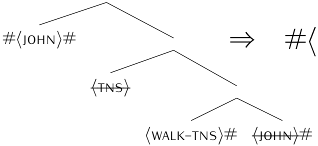
Lexicon:

$\langle\{\text{D}, \textsc{john}, 3, \textsc{sg}\}, \textsc{none}\rangle :: \texttt{d -k}$

$\langle\{\text{V}, \textsc{walk}\}, \textsc{none}\rangle :: \texttt{=d v}$

$\langle\{\text{T}, \textsc{pres}, 3, \textsc{sg}\}, \textsc{none}\rangle :: \texttt{<=v +k t}$

Syntax:



$\Rightarrow \#\langle\textsc{john}\rangle\#\langle\textsc{walk}\rangle\langle\textsc{tns}\rangle\#$

VI Rules:

$\left\langle \begin{matrix}\{\textsc{john}\},\\ \textit{None}\end{matrix} \right\rangle \longrightarrow \left\langle \begin{matrix}\{\textsc{john}\},\\ /\text{j}/\end{matrix} \right\rangle \left\langle \begin{matrix}\{\textsc{john}\},\\ /\text{o}/\end{matrix} \right\rangle \left\langle \begin{matrix}\{\textsc{john}\},\\ /\text{n}/\end{matrix} \right\rangle$

$\left\langle \begin{matrix}\{\textsc{walk}\},\\ \textit{None}\end{matrix} \right\rangle \longrightarrow \left\langle \begin{matrix}\{\textsc{walk}\},\\ /\text{w}/\end{matrix} \right\rangle \left\langle \begin{matrix}\{\textsc{walk}\},\\ /\text{a}/\end{matrix} \right\rangle \left\langle \begin{matrix}\{\textsc{walk}\},\\ /\text{l}/\end{matrix} \right\rangle \left\langle \begin{matrix}\{\textsc{walk}\},\\ /\text{k}/\end{matrix} \right\rangle$

$\left\langle \begin{matrix}\{\textsc{tns}\},\\ \textit{None}\end{matrix} \right\rangle \longrightarrow \left\langle \begin{matrix}\{\textsc{tns}\},\\ /\text{s}/\end{matrix} \right\rangle$

English examples

(19) Derivation of *worse*–Fusion + VI

Lexicon:

$\langle\{\text{Adj}, \textsc{bad}\}, \textsc{none}\rangle$ :: `adj`

$\langle\{\text{Cmpr}, \textsc{more}\}, \textsc{none}\rangle$ :: `=adj adj`

Syntax:

...

$\#\langle\textsc{more}\rangle \quad \langle\textsc{bad}\rangle\# \quad \Rightarrow \quad \#\langle\textsc{more}\rangle\langle\textsc{bad}\rangle\#$

Fusion: $\langle\textsc{more}\rangle\langle\textsc{bad}\rangle \rightarrow \langle\textsc{more}, \textsc{bad}\rangle$

VI Rules:

$$\left\langle \begin{matrix}\{\textsc{more},\textsc{bad}\},\\ \textit{None}\end{matrix} \right\rangle \rightarrow \left\langle \begin{matrix}\{\textsc{more},\textsc{bad}\},\\ /\text{w}/\end{matrix} \right\rangle \left\langle \begin{matrix}\{\textsc{more},\textsc{bad}\},\\ /\text{r̩}/\end{matrix} \right\rangle \left\langle \begin{matrix}\{\textsc{more},\textsc{bad}\},\\ /\text{s}/\end{matrix} \right\rangle$$

English examples

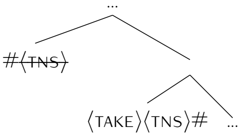
(20) Derivation of *took*–VI + Readjustment

Lexicon:

$\langle\{\text{V, TAKE}\}, \text{NONE}\rangle$ :: `v`

$\langle\{\text{T, PST}\}, \text{NONE}\rangle$ :: `<=v t`

Syntax:



$\Rightarrow \quad \#\langle\text{TAKE}\rangle\langle\text{TNS}\rangle\#$

VI Rules:

$$\left\langle \begin{matrix}\{\text{TAKE}\},\\ \textit{None}\end{matrix} \right\rangle \rightarrow \left\langle \begin{matrix}\{\text{TAKE}\},\\ /\text{t}/\end{matrix} \right\rangle \left\langle \begin{matrix}\{\text{TAKE}\},\\ /\text{ei}/\end{matrix} \right\rangle \left\langle \begin{matrix}\{\text{TAKE}\},\\ /\text{k}/\end{matrix} \right\rangle$$

$$\left\langle \begin{matrix}\{\text{TNS}\},\\ \textit{None}\end{matrix} \right\rangle \rightarrow \left\langle \begin{matrix}\{\text{TNS}\},\\ \epsilon\end{matrix} \right\rangle / \left\langle \begin{matrix}\{\text{TAKE}\},\\ /\text{k}/\end{matrix} \right\rangle \_\_$$

Readjustment:

$$\left\langle \begin{matrix}\{\text{TAKE}\},\\ /\text{ei}/\end{matrix} \right\rangle \rightarrow \left\langle \begin{matrix}\{\text{TAKE}\},\\ /\text{ɯ}/\end{matrix} \right\rangle / \_\_ \left\langle \begin{matrix}\{\text{TAKE}\},\\ /\text{k}/\end{matrix} \right\rangle \left\langle \begin{matrix}\{+\text{PAST},+\text{FIN}\},\\ \epsilon\end{matrix} \right\rangle$$

Conclusion

(21) Architecture of grammar

Syn

$\downarrow$

Lin

$\downarrow$

MS $\Leftarrow$ purely morphological, e.g. fusion

$\downarrow$

morphology VI

$\downarrow$

MP $\Leftarrow$ morphophonology, e.g. readjustment

$\downarrow$

Phon

$\downarrow$

PF

Conclusion

- Syntax/Morphology interface modelled by FST
- FST (morphology) works over strings–binary trees not needed
- Linearization PRE-morphology!
- Formalization like this helps make concrete intuitions, e.g. impoverishment vs. other operations
- Where to go from here:
  - Refine details, e.g. VI ordering rules
  - Address subregularity?