Distributed Morphology over strings

THE UNIVERSITY OF CHICAGO

Marina Ermolaeva
mermolaeva@uchicago.edu
Daniel Edmiston
danedmiston@uchicago.edu

Introduction

What we are doing:

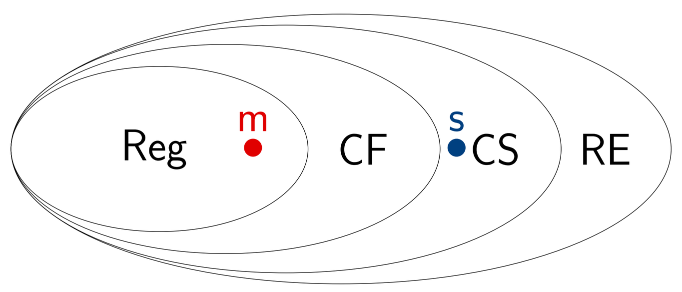
- Formalizing Distributed Morphology (DM) and Syntax/Morphology interface
- Testing core assumption in DM (and thus probing linearization's place)
- Solidifying intuitions re: nature of DM's operations

What we are proposing:

- Syntax: derivation over feature structures (FSs)
- Linearization: pre-morphology flattening of trees of FSs
- Morphology: regular relation mapping strings of FSs to strings of morphophonemes

Motivation

- **Claim of DM**: "Syntax all the way down", i.e. morphology over trees; This predicts morphology behaves like syntax.
- However, work in NLP treats morphology with finite-state methods (e.g. Karttunen et al. 1992); syntax cannot be done this way (Shieber 1985)



(morphology vs. syntax)

- If we eliminate binary trees from morphology, we get for free that morphology appears **(at most) regular**
- What would DM look like without trees (i.e. over strings)?

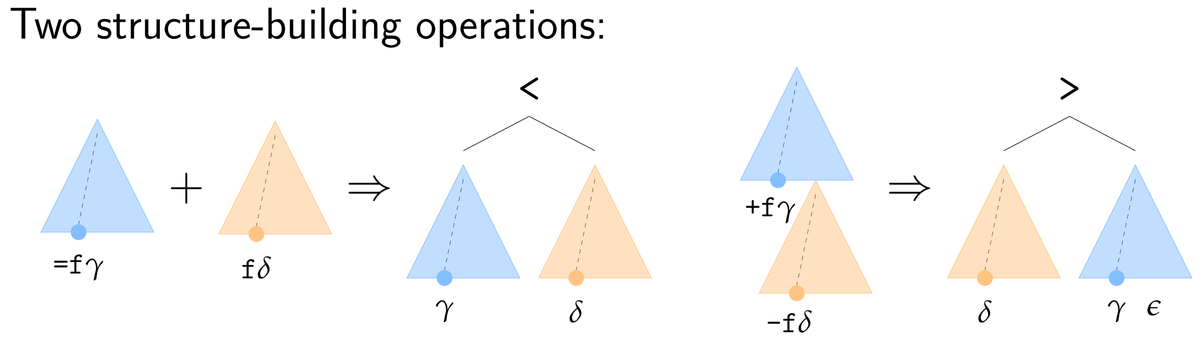
Syntactic assumptions

Framework: Minimalist Grammars (MGs, Stabler 1997)

- A set of **syntactic features**:

$Syn = base$	(categories)
$\cup\ \{\texttt{=f} \mid \texttt{f} \in base\}$	(selectors)
$\cup\ \{\texttt{+f} \mid \texttt{f} \in base\}$	(licensors)
$\cup\ \{\texttt{-f} \mid \texttt{f} \in base\}$	(licensees)

- A **lexicon**: $Lex \subset \Sigma^* Syn^*$, where Σ is a set of pronounced segments
- Two structure-building operations:



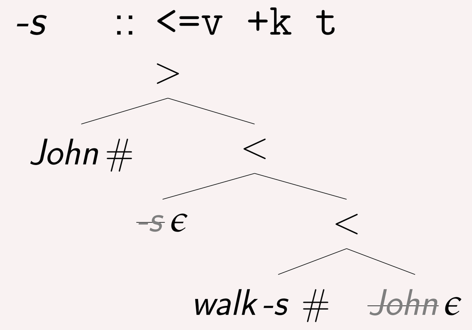
(*merge*) (*move*)

Modification: syntax assembles morphological words

- Lowering and Head Movement (HM) as *merge* with concatenation of heads
- Mirror Theory (Brody 1997, Kobele 2002): **strong** and **weak** nodes
- Three subtypes of selector features:
  =f (normal *merge*)
  =>f (strong node; *merge* + HM)
  <=f (weak node; *merge* + Lowering)
- Boundary symbols (#) separate words
- Morphology after syntax: =>f and <=f only select **non-moving expressions**

Example: toy grammar

John :: d -k
walk :: =d v
-*s* :: <=v +k t



yield: # *John* # *walk* -*s* #

Separating syntax and phonology

Modification: Lexical items no longer contain phonological information

- **Feature structures**:
  $FS = \mathcal{P}(M) \times (\Sigma \cup \{\epsilon, none\})$, where *none* denotes the "placeholder" exponent and M is a finite set of morphological features;
  For $s = \langle x, y\rangle \in FS$, $feat(s) = x$ and $exp(s) = y$.
- **Redefining lexicon**:
  $Lex \subset \{s \mid s \in FS\ \&\ exp(s) = none\}\ Syn^*$
- Feature structures after VI correspond to **morphophonemes**.

Example: FS-based grammar

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

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

$\left\langle {\{\text{T}, \textsc{prs}, 3, \textsc{sg}\} \atop none} \right\rangle$ :: <=v +k t

Morphological rules...

- Morphological rules operate on **underspecified feature structures**:
  $FS_U = \mathcal{P}(M) \times (\Sigma \cup \{\epsilon, none, ?\})$,
  where ? stands for "any exponent".
- "Context-sensitive" rewriting rules that do not overwrite their own output define **regular relations** over strings (Kaplan & Kay 1994)

Analogy: phonological rules

A rule in feature matrix notation is equivalent to a set of rules over atomic symbols. For instance, $[-syl, +voi] \to [-voi]\ /\ \_\_\ [-voi]$ abbreviates multiple rules:
$b \to p\ /\ \_\_\ (p \mid t \mid k)$, $d \to t\ /\ \_\_\ (p \mid t \mid k)$, ...

- **Rewriting rule format** for morphological rules:

structural description		structural change	/	left context		right context
A	$\to$	B	/	C	$\_\_$	D
$A_1, ..., A_m \in FS_U$		$B_1, ..., B_n \in FS_U$		(regular expressions over $FS_U \cup \{\#\}$)		

- A rule is **purely morphological** iff $exp(A_1) = ... = exp(A_m) = exp(B_1) = ... = exp(B_n) = none$;
  feature-preserving iff $\bigcup_{i=1}^{m} feat(A_i) = \bigcup_{j=1}^{n} feat(B_j)$;
  set-preserving iff $feat(A_1) = ... = feat(A_m) = feat(B_1) = ... = feat(B_n)$.

Rule class	$\lvert\mathbf{A}\rvert$	$\lvert\mathbf{B}\rvert$	Properties
fusion	2	1	feature-preserving, purely morphological
fission	1	2	feature-preserving, purely morphological
impoverishment	1	1	$feat(B_1) \subset feat(A_1)$, purely morphological
VI	1	≥ 1	$exp(A_1) = none$, $exp(B_j) \neq none$ for $j \in [1..\lvert B\rvert]$, set-preserving
readjustment	≥ 0	≥ 0	$exp(A_i) \neq none$ for $i \in [1..\lvert A\rvert]$, $exp(B_j) \neq none$ for $j \in [1..\lvert B\rvert]$, set-preserving

Example: DM rules for English morphology

Fusion:
$\left\langle {\{\textsc{Adj}, \textsc{bad}\} \atop none} \right\rangle \left\langle {\{\textsc{Cmpr}\} \atop none} \right\rangle \to \left\langle {\{\textsc{Adj}, \textsc{bad}, \textsc{Cmpr}\} \atop none} \right\rangle$

Vocabulary Insertion:
$\left\langle {\{\textsc{take}\} \atop none} \right\rangle \to \left\langle {\{\textsc{take}\} \atop /\text{t}/} \right\rangle \left\langle {\{\textsc{take}\} \atop /\text{eɪ}/} \right\rangle \left\langle {\{\textsc{take}\} \atop /\text{k}/} \right\rangle$

Readjustment:
$\left\langle {\{\textsc{take}\} \atop /\text{eɪ}/} \right\rangle \to \left\langle {\{\textsc{take}\} \atop /\text{ʊ}/} \right\rangle\ /\ \_\_ \left(\left\langle {\{\textsc{take}\} \atop ?} \right\rangle\right)^* \left\langle {\{\text{T}, \textsc{Pst}\} \atop ?} \right\rangle$

... as regular relations

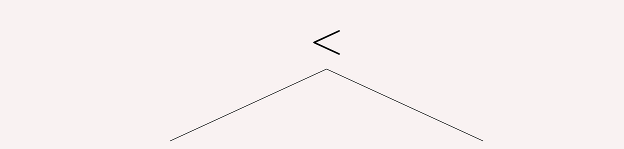
- Regular relations manipulate strings of unanalyzable symbols
- **Instances**: regular expressions over fully specified feature structures
  For $fs \in FS_U$, $inst(fs) = \{x \mid x \in FS\ \&\ feat(x) \supseteq feat(fs)\ \&\ (exp(x) = exp(fs)$ or $exp(fs) = ?)\}$;
  For any regular expression X over $FS_U \cup \{\#\}$, $inst(X) = X[x_1 \mapsto \bigcup inst(x_1), ..., x_n \mapsto \bigcup inst(x_n)]$,
  where $\{x_1, ..., x_n\}$ is the set of all feature structures in X.
- **Rule instantiations**:
  For any rule $r = A \to B\ /\ C\_\_D$ ($|A| = m, |B| = n$),
  $batch(r)$ is the set of all rules $a \to b\ /\ inst(C)\_\_inst(D)$ such that
  $a = a_1, ..., a_m \in FS$ and $b = b_1, ..., b_n \in FS$;
  $a_i \in inst(A_i)$ for $i \in [1..m]$;
  $feat(b_j) = feat(B_j) \cup (\bigcup_{i=1}^{m} feat(a_i) \setminus \bigcup_{i=1}^{m} feat(A_i))$
  $exp(b_j) = exp(B_j)$ for $j \in [1..n]$.
- **Kaplan & Kay 1994**:
  Simultaneous application of a rule set as **batch rules**;
  Ordered rules as **composition** of regular relations.

Cyclicity as rule ordering

- Bobaljik 2000: VI proceeds cyclically **from the root outwards** and **deletes** features it expresses;
  - Outward sensitivity only to morphosyntactic features;
  - Inward sensitivity only to morphophonological features.

  Counter-examples: Deal & Wolf 2013, Gribanova & Harizanov 2015
- Adger 2003: **Hierarchy of Projections** (HoP)
  Clausal: C ⟩ T ⟩ (Neg) ⟩ (Perf) ⟩ (Prog) ⟩ (Pass) ⟩ v ⟩ V
  Nominal: D ⟩ (Poss) ⟩ n ⟩ N
  HoP can be built into syntactic features to control *merge*.
- Simulating cyclicity effects on strings:
  Ordering of VI rules follows HoP, allowing for possible mismatches;
  Expressed features (provisionally) remain part of the representation.

Example: phonologically conditioned allomorphy in English



$\left\langle {\{\text{D}, \textsc{indef}\} \atop none} \right\rangle \# \quad \left\langle {\{\text{N}, \textsc{apple}, 3, \textsc{sg}\} \atop none} \right\rangle \#$

Ordered VI rules:

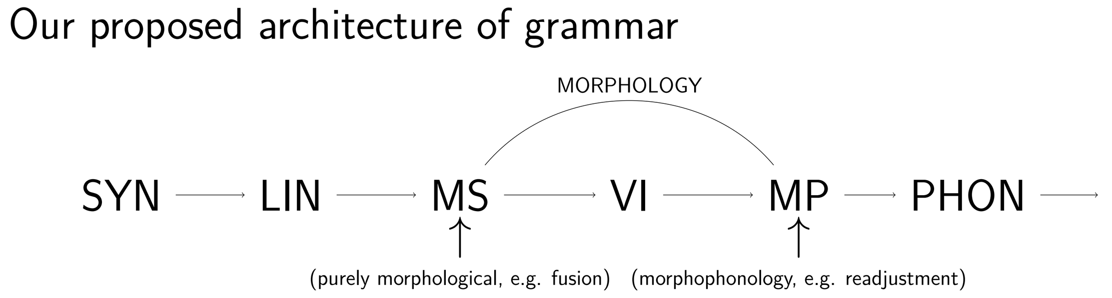
$\left\langle {\{\text{N}, \textsc{apple}, 3, \textsc{sg}\} \atop none} \right\rangle \to \left\langle {\{\text{N}, \textsc{apple}, 3, \textsc{sg}\} \atop /\text{æ}/} \right\rangle \left\langle {\{\text{N}, \textsc{apple}, 3, \textsc{sg}\} \atop /\text{p}/} \right\rangle \left\langle {\{\text{N}, \textsc{apple}, 3, \textsc{sg}\} \atop /\text{l̩}/} \right\rangle$

$\left\langle {\{\text{D}, \textsc{indef}\} \atop none} \right\rangle \to \left\langle {\{\text{D}, \textsc{indef}\} \atop /\text{ə}/} \right\rangle \left\langle {\{\text{D}, \textsc{indef}\} \atop /\text{n}/} \right\rangle\ /\ \_\_\ \# \left(\left\langle {\{\ \} \atop /\text{æ}/} \right\rangle \middle| \left\langle {\{\ \} \atop /\text{e}/} \right\rangle \middle| ... \right)$

$\left\langle {\{\text{D}, \textsc{indef}\} \atop none} \right\rangle \to \left\langle {\{\text{D}, \textsc{indef}\} \atop /\text{ə}/} \right\rangle\ /\ \_\_\ \# \left(\left\langle {\{\ \} \atop /\text{b}/} \right\rangle \middle| \left\langle {\{\ \} \atop /\text{d}/} \right\rangle \middle| ... \right)$

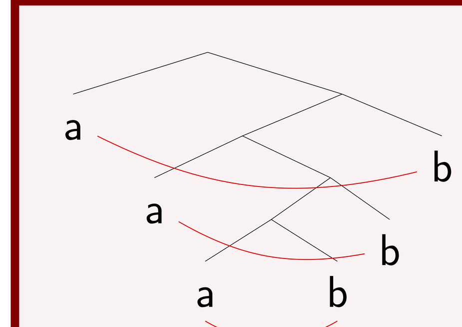
Conclusion

- Our proposed architecture of grammar

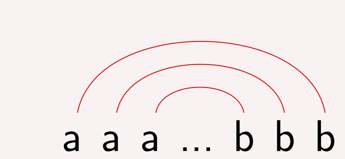


- Syntax/Morphology interface modelled by regular relations
- Morphology works over strings: binary trees not needed
- Linearization/flattening happens **pre**-morphology
- Predicts we should never see context-free morphology
- Our formalization helps make intuitions re: DM operations concrete

Abstract example: context-free morpho(phono)logy



Unbounded nested dependencies: **incompatible** with DM over strings



Any natural language examples?

References: Adger, David. 2003. Core syntax: a minimalist approach. ▪ Bobaljik, Jonathan D. 2000. The ins and outs of contextual allomorphy. ▪ Brody. Michael. 1997. Mirror theory. ▪ Deal, Amy R., and Matthew Wolf. 2013. Outward-sensitive phonologically-conditioned allomorphy in Nez Perce. ▪ Gribanova, Vera, and Boris Harizanov. 2015. Locality and directionality in inward-sensitive allomorphy: Russian and Bulgarian. ▪ Kaplan, Ronald M., and Martin Kay. 1994. Regular models of phonological rule systems. ▪ Karttunen, Lauri, Ronald M. Kaplan, and Annie Zaenen. 1992. Two-level morphology with composition. ▪ Kobele, Gregory M. 2002. Formalizing Mirror theory. ▪ Shieber, Stuart M. 1985. Evidence against the context-freeness of natural language. ▪ Stabler, Edward P. 1997. Derivational minimalism.