COMPUTING CYCLIC PHONOLOGY IN ARmenian

Hossep Dolatian

Stony Brook University

January 17, 2020
Decomposing...

- Title: *Computing Cyclic phonology in Armenian*
- Subtitle: *Logical structure of the morphology-phonology interface*
Decomposing...
  - Title: *Computing Cyclic phonology in Armenian*
  - Subtitle: *Logical structure of the morphology-phonology interface*

What is phonology?
  - What principles control the alternation in the pronunciation of morphemes

Two halves:
Decomposing...
  - Title: *Computing Cyclic phonology in Armenian*
  - Subtitle: *Logical structure of the morphology-phonology interface*

What is phonology?
  - What principles control the alternation in the pronunciation of morphemes

Two halves:
  1. Empirical + theoretical: What does Armenian phonology tell you?
  2. Computational + theoretical: How can we compute phonology with formal logic
Results + Takeaway

- Empirical + theoretical:
  - Armenian is under-documented but rich in phonological interactions
  - Focus on a bracketing paradox and how to ‘solve’ it
  - Requires cyclicity + head-operations + prosodic constituents

- Computational + theoretical:
  - Formalizing those principles with logic-based graph-to-graph transductions
  - Uncover implicit aspects of theory
  - Provide unified framework to examine expressivity of different processes
  - Isolate parts of theory that are computationally simple (local) vs. complex (global)
Empirical + theoretical:
- Armenian is under-documented but rich in phonological interactions
- Focus on a bracketing paradox and how to ‘solve’ it
- Requires cyclicity + head-operations + prosodic constituents

* Extract the principles behind the data
  - Morphological Structure
  - Prosodic Structure
  - Rule domains
  - Interactionism & Cyclicility

Computational + theoretical:
- Formalizing those principles with logic-based graph-to-graph transductions
- Uncover implicit aspects of theory
- Provide unified framework to examine expressivity of different processes
- Isolate parts of theory that are computationally simple (local) vs. complex (global)
Results + Takeaway

- Empirical + theoretical:
  - Armenian is under-documented but rich in phonological interactions
  - Focus on a bracketing paradox and how to ‘solve’ it
  - Requires cyclicity + head-operations + prosodic constituents
* Extract the principles behind the data
  - Morphological Structure
  - Prosodic Structure
  - Rule domains
  - Interactionism & Cyclicity
- Computational + theoretical:
  - Formalizing those principles with logic-based graph-to-graph transductions
  - Uncover implicit aspects of theory
  - Provide unified framework to examine expressivity of different processes
  - Isolate parts of theory that are computationally simple (local) vs. complex (global)
Table of Contents

 Armenian
  - Preview
  - Distribution
  - Head operations
  - Cyclic or counter-cyclic?
  - Variation in prosody
  - Wrap-up

 Computation
  - What’s the link?
  - Computing morpho-phono
  - Computing with logic
  - Computing morpho-phonology with logic

 End
Syllable-counting plurals

- Plural allomorphy in simplex words

\[
\begin{array}{c|c|c}
\sigma & \sigma \text{-er} & \sigma\sigma^+ & \sigma\sigma^+ \text{-ner} \\
pág & \text{pag-ér} & \text{panág} & \text{panag-nér} \\
& \text{‘yards’} & \text{‘armies’} & \text{‘armies’} \\
akarág & \text{akarag-nér} & & \text{farms}
\end{array}
\]
Syllable-counting plurals

- Plural allomorphy in simplex words

  \[ \sigma \quad \sigma\text{-}er \quad \mid \quad \sigma\sigma^+ \quad \sigma\sigma^+-\text{ner} \]
  
  \[ \text{pág} \quad \text{pag-ér} \quad \text{‘yards’} \quad \mid \quad \text{panág} \quad \text{panag-nér} \quad \text{‘armies’} \]
  
  \[ \text{akarág} \quad \text{akarag-nér} \quad \text{farms} \]

- Stress does not matter – everything has final stress

- Mnemonic:
  - short-allomorph → smaller syllables
    - PL → \text{-}er / \#\sigma _
      - pag pag-er ‘yards’
    - PL → \text{-}ner / elsewhere
      - panag panag-ner ‘armies’
**Compound plurals**

- Compounds: concatenate 2 stems with -a-
  
  \[ \text{	extit{t}f\text{\`a}r} + \text{	extit{s}ırd} \text{ ‘evil + heart’} \quad \text{ants\textit{r}év} + \text{	extit{t}f\text{úr}} \text{ ‘rain + water’} \]
  
  \[ \text{	extit{t}f\text{ar-a-s}ırd} \text{ ‘evil-hearted’} \quad \text{ants\textit{rev-a-t}f\text{úr}} \text{ ‘rain-water’} \]

- Final stress on compound \(\sim\) 1 PWord
- Plural?
Compounds: concatenate 2 stems with -a-

\[\text{efs} + \text{hrd} \quad \text{‘evil + heart’} \quad \text{antsr} + \text{fr} \quad \text{‘rain + water’}\]

\[\text{ef} + \text{hrd} \quad \text{‘evil-hearted’} \quad \text{antsr} + \text{fr} \quad \text{‘rain-water’}\]

- Final stress on compound \(\sim\) 1 PWord
- Plural?
  - STEM2 is monosyllabic but compound is polysyllabic...
Compound plurals

- Compounds: concatenate 2 stems with -a-
  \[ \text{tʃá}r + \text{sírd} \quad \text{‘evil + heart’} \quad \text{antsrév} + \text{tʃúr} \quad \text{‘rain + water’} \]
  \[ \text{tʃar-a-sírd} \quad \text{‘evil-hearted’} \quad \text{antsrev-a-tʃúr} \quad \text{‘rain-water’} \]

- Final stress on compound \(\sim\) 1 PWord

- Plural?
  - STEM2 is monosyllabic but compound is polysyllabic...
  - Count entire compound

\[ \text{tʃar-a-sírd} \quad \text{‘evil-hearted’} \]
\[ \text{tʃar-a-sírd-ner} \quad \text{‘evil-hearted ppl’} \]
Compound plurals

- Compounds: concatenate 2 stems with -a-
  - ţfăr + sîrd ‘evil + heart’
  - ţfar-a-sîrd ‘evil-hearted’
  - ţfăr-a-sîrd -ner ‘evil-hearted ppl’
  - anțsrév + ţfûr ‘rain + water’
  - anțsrév-a-ţfûr ‘rain-water’
  - anțsrév-a-ţfûr -er ‘rain-waters’

- Final stress on compound ~ 1 PWord

- Plural?
  - Stem2 is monosyllabic but compound is polysyllabic...
  - Count entire compound
  - Only count Stem2
Plural paradox

- Bracketing paradox: two contradictory constituencies
  - **SEM**: Plural has semantic scope over compound
  - **PHON**: Plural should count the *entire* compound

\[\text{\textit{tfar-a-sird-ner}} \text{ ‘evil-hearted ppl’}\]

Transparent Plural

\[\text{\textit{tfar-a-sird-ner}}\]

\[\text{\textit{tfar-a-sird-ner}} \text{ ‘rain-waters’}\]

Paradoxical Plural

\[\text{\textit{tfar-a-sird-ner}}\]

\[\text{\textit{tfar-a-sird-ner}}\]
Plural paradox

- Bracketing paradox: two contradictory constituencies
  - **SEM**: Plural has semantic scope over compound
  - **PHON**: Plural should count the *entire* compound

\[
\text{\textit{t}s\textit{ar}-a-\textit{sird}-\textit{ner}} \quad \text{'evil-hearted ppl'} \quad \text{\textit{antsrev}-a-\textit{t}s\textit{ur}-\textit{er}} \quad \text{‘rain-waters’}
\]

**Transparent Plural**

\[
\text{\textit{t}s\textit{ar}} \quad \textit{a} \quad \textit{sird} \quad \textit{ner}
\]

**Paradoxical Plural**

\[
\text{\textit{antsrev}} \quad \textit{a} \quad \textit{t}s\textit{ur} \quad \textit{er}
\]
Questions

- Data:
  - Transparent plural: $\tilde{t}far-a-sird-ner$
    - ‘evil-hearted people’
  - Paradoxical plural: $\tilde{a}ntsrev-a-tfur-er$
    - ‘rain-waters’

- Question: Why get transparent plurals vs. paradoxical plurals?
Questions

- Data:

<table>
<thead>
<tr>
<th>Transparent plural</th>
<th>Paradoxical plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>tfar-a-sird-ner</td>
<td>antsrev-a-tfur-er</td>
</tr>
<tr>
<td>‘evil-hearted people’</td>
<td>‘rain-waters’</td>
</tr>
</tbody>
</table>

- Question: Why get transparent plurals vs. paradoxical plurals?

- A lot of tools for paradoxes...

<table>
<thead>
<tr>
<th>Counter-cyclic</th>
<th>Cyclic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rebracketing</td>
<td>Head</td>
</tr>
<tr>
<td>Late Adjunction paradigms</td>
<td>Prosodic operations</td>
</tr>
<tr>
<td>Sproat ‘85</td>
<td>Hoeksema ‘84</td>
</tr>
<tr>
<td>Newell ‘05</td>
<td>Nespor &amp; Vogel ‘87</td>
</tr>
<tr>
<td>Stump ‘01</td>
<td></td>
</tr>
</tbody>
</table>

→ Head-operations + prosodic constituents!
● Paradox is *essentially* about SEMANTIC endocentricity\(^1\)

\(^1\)Allen (1979)’s ‘IS A’ relation; well-known generalization in Armenian (Vaux, 1998).
Paradox is essentially about SEMANTIC endocentricity\(^1\)

Is the compound endocentric?

\[ \uparrow \]

Is it hyponymic?

\[ \uparrow \]

Is compound a type of STEM2?

\[ \sqrt{\text{Paradoxical}} \quad \times \quad \text{Transparent} \]

\[ \begin{array}{c}
\text{rain-water} \quad \text{IS A} \quad \text{water} \\
\text{antsrev-a-t\text{fur-er}} \end{array} \quad \begin{array}{c}
\text{evil-hearted} \quad \text{IS not A} \quad \text{heart} \\
\text{t\text{far-a-sird-ner}} \end{array} \]

\[ \text{t\text{fur-er}} \]

\(^1\)Allen (1979)’s ‘IS A’ relation; well-known generalization in Armenian (Vaux, 1998).
**Compound classifications**

- How robust is the paradox? How robust is endocentricity?

\[^2\text{Donabédian, 2004}\]
Compound classifications

- How robust is the paradox? How robust is endocentricity? Very!\(^2\)
- Hyponymic → endocentric → paradoxical

<table>
<thead>
<tr>
<th>Nominal</th>
<th>(X + N = N)</th>
<th>(\text{antsrev} + \text{t}f\text{ur})</th>
<th>‘rain + water’</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(X + \underline{N}-er)</td>
<td>(\text{antsrev-a-} \text{t}f\text{ur-er})</td>
<td>‘rain water(s)’</td>
</tr>
</tbody>
</table>

\(^2\)(Donabédian, 2004)
Compound classifications

- How robust is the paradox? How robust is endocentricity? Very!\(^2\)
- Hyponymic → endocentric → paradoxical

**Nominal**

\[
\begin{align*}
X + N &= N & \text{antsrev} + t\text{fur} & \text{‘rain + water’} \\
X + N\text{-er} &= \text{antsrev-a-t\text{fur}-er} & \text{‘rain water(s)’}
\end{align*}
\]

- Non-hyponymic → exocentric → transparent

**Possessive**

\[
\begin{align*}
X + N &= A & t\text{far} + sird & \text{‘evil + heart’} \\
X + N\text{-er} &= t\text{far-a-sird-ner} & \text{‘evil-hearted (people)’}
\end{align*}
\]

**Deverbal**

\[
\begin{align*}
X + V_{\text{root}} &= N/A & \text{antsrev} + \text{per-el} & \text{‘rain + to bring’} \\
X + V\text{-er} &= \text{antsrev-a-per-ner} & \text{‘rain-bearing’}
\end{align*}
\]

- Doesn’t matter what’s POS of STEM2 or semantic relationship (appendix)

\(^2\)(Donabédian, 2004)
**INTERIM SUMMARY**

- Simple facts:
  - Compounds are polysyllabic
  - Plural formation is syllable-counting allomorphy
- Twist:
  - Exocentric plurals are formed by counting *entire* compound
  - Endocentric plurals are formed by counting *only* STEM2

\[ \text{tfar} + \text{sird} \quad 'evil + heart' \quad \text{antsrev} + \text{tfur} \quad 'rain + water' \]
\[ \text{tfar-a-sird-ner} \quad 'evil-hearted (people) \quad \text{antsrev-a-tfur-er} \quad 'rain-water(s)' \]
Interim summary

- Simple facts:
  - Compounds are polysyllabic
  - Plural formation is syllable-counting allomorphy
- Twist:
  - Exocentric plurals are formed by counting *entire* compound
  - Endocentric plurals are formed by counting *only* STEM2

\[\text{tfar} + \text{sird} \quad \text{‘evil + heart’} \quad \text{antsrev} + \text{tfur} \quad \text{‘rain + water’}\]
\[\text{tfar}-\text{a-sird-ner} \quad \text{‘evil-hearted (people)’} \quad \text{antsrev}-\text{a-tfur-er} \quad \text{‘rain-water(s)’}\]

- Ultimately about semantics:
  - is compound type of STEM2?
- But how do you *generate* the paradoxical plural?
  - Head-operation over *semantic head*
Head-marking

- PL is a head-operation
  - $PL(X + h) = X + PL(h)$
    - $PL \rightarrow -er / [\sigma]_h$
    - $PL \rightarrow -ner /$ elsewhere

- Semantic Head $h$:
  - If simplex/exocentric, then $h=\text{WORD}$
  - If endocentric, then $h=\text{STEM2}$
**Head-marking**

- **PL** is a **head-operation**
  - $\text{PL}(X + h) = X + \text{PL}(h)$
  - $\text{PL} \rightarrow \text{-er} / [\sigma]_{h} _{\_}$
  - $\text{PL} \rightarrow \text{-ner} / \text{elsewhere}$

- **Semantic Head $h$:**
  - If simplex/exocentric, then $h=\text{WORD}$
  - If endocentric, then $h=\text{STEM2}$

---

Exo

‘evil-hearted’

Input $\tilde{t}\tilde{f}ar + \text{sird} + \text{PL}$

1. Stems $\tilde{t}\tilde{f}\acute{a}r$ $\text{sírd}$

---
**Head-marking**

- PL is a **head-operation**
  - $PL(X + h) = X + PL(h)$
  - $PL \rightarrow -er \ / \ [\sigma]_h$
  - $PL \rightarrow -ner \ / \ elsewhere$

- **Semantic Head $h$:**
  - If simplex/exocentric, then $h=\text{WORD}$
  - If endocentric, then $h=\text{STEM2}$

---

Exo

‘evil-hearted’

Input $\widehat{tʃar} + \widehat{sírd} + PL$

1. Stems $\widehat{tʃár} \ \widehat{sírd}$
2. Combine $\widehat{tʃar-a-sírd}$
**Head-marking**

- **PL is a head-operation**
  - \( PL(X + h) = X + PL(h) \)
  - \( PL \rightarrow -er / [ \sigma ]_h \_ \)
  - \( PL \rightarrow -ner / \text{elsewhere} \)

- **Semantic Head \( h \):**
  - If simplex/exocentric, then \( h=\text{WORD} \)
  - If endocentric, then \( h=\text{STEM2} \)

---

**Exo**

‘evil-hearted’

Input \( \hat{t}far + \hat{sird} + PL \)

1. Stems \( \hat{t}f\acute{a}r \quad \hat{s}ird \)
2. Combine \( \hat{t}far-a-sírd \)
2.5 \( h? \) \( [\hat{t}far-a-sírd]_h \)
**Head-marking**

- **PL is a head-operation**
  - $PL(X + h) = X + PL(h)$
  - $PL \rightarrow -er / [\sigma]_h$
  - $PL \rightarrow -ner /$ elsewhere

- **Semantic Head $h$:**
  - If simplex/exocentric, then $h=$WORD
  - if endocentric, then $h=$STEM2

---

**Exo**

‘evil-hearted’

Input  $\tilde{t}sar + sird + PL$

1. Stems  $\tilde{t}s\acute{a}r$  sírd
2. Combine  $\tilde{t}sar-a-sírd$
2.5 $h?$  $[\tilde{t}sar-a-sírd]_h$
3. Add PL  $\tilde{t}sar-a-sírd-nér$
**Head-marking**

- **PL is a head-operation**
  - $PL(X + h) = X + PL(h)$
  - $PL \rightarrow -er$ / $\sigma_h$  
  - $PL \rightarrow -ner$ / elsewhere

- **Semantic Head $h$:**
  - If simplex/exocentric, then $h=\text{WORD}$
  - If endocentric, then $h=\text{STEM2}$

---

**Exo**

‘evil-hearted’

<table>
<thead>
<tr>
<th>Input</th>
<th>$\hat{t}_{far} + s\hat{ir}d + PL$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Stems</td>
<td>$\hat{t}_{f\acute{a}r} s\acute{ir}d$</td>
</tr>
<tr>
<td>2. Combine</td>
<td>$\hat{t}_{far-a-s\acute{ir}d}$</td>
</tr>
<tr>
<td>2.5 $h$?</td>
<td>$[\hat{t}_{far-a-s\acute{ir}d}]_h$</td>
</tr>
<tr>
<td>3. Add PL</td>
<td>$\hat{t}_{far-a-s\acute{ir}d-n\acute{e}r}$</td>
</tr>
</tbody>
</table>

**Endo**

‘rain-water(s)’

<table>
<thead>
<tr>
<th>Input</th>
<th>$\hat{ants}<em>{rev} + \hat{t}</em>{fur} + PL$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{ants}_{rev}$</td>
<td>$\hat{t}_{fur}$</td>
</tr>
<tr>
<td>$\hat{ants}<em>{rev} - \hat{t}</em>{fur}$</td>
<td>$\hat{t}_{fur}$</td>
</tr>
<tr>
<td>$\hat{ants}<em>{rev} - \hat{t}</em>{fur}$</td>
<td>$\hat{t}_{fur}$</td>
</tr>
<tr>
<td>$\hat{ants}<em>{rev} - \hat{t}</em>{fur}$</td>
<td>$\hat{t}_{fur}$</td>
</tr>
</tbody>
</table>
**Head-marking**

- **PL is a head-operation**
  - \( \text{PL}(X + h) = X + \text{PL}(h) \)
  - \( \text{PL} \rightarrow -er / [\sigma]_h _- \)
  - \( \text{PL} \rightarrow -ner / \text{elsewhere} \)

- **Semantic Head \( h \):**
  - If simplex/exocentric, then \( h=\text{WORD} \)
  - If endocentric, then \( h=\text{STEM2} \)

---

**Exo**

‘evil-hearted’

- **Input**  
  - \( \tilde{t}\text{far} + \text{sird} + \text{PL} \)

1. **Stems**  
  - \( \tilde{t}\text{far} \quad \text{sird} \)
2. **Combine**  
  - \( \tilde{t}\text{far-a-sird} \)
2.5 **\( h? \)**  
  - \( [\tilde{t}\text{far-a-sird}]_h \)
3. **Add PL**  
  - \( \tilde{t}\text{far-a-sird-nér} \)

**Endo**

‘rain-water(s)’

- **Input**  
  - \( \text{antsrev} + \tilde{t}\text{fur} + \text{PL} \)

- **antsrév + \text{t}fúr**
Head-marking

- **PL is a head-operation**
  - $PL(X + h) = X + PL(h)$
  - $PL \rightarrow -er / \{ \sigma \}_h$
  - $PL \rightarrow -ner /$ elsewhere

- **Semantic Head $h$:**
  - If simplex/exocentric, then $h =$WORD
  - if endocentric, then $h =$STEM2

---

**Exo**

- ‘evil-hearted’
  - Input: $\hat{t}far + sird + PL$
  - 1. Stems: $\hat{t}far$ sîrd
  - 2. Combine: $\hat{t}far-a-sîrd$
  - 2.5 $h?$: $[\hat{t}far-a-sîrd]_h$
  - 3. Add PL: $\hat{t}far-a-sîrd-nér$

**Endo**

- ‘rain-water(s)’
  - Input: $\hat{antsrev} + \hat{tfur} + PL$
  - 1. Stems: $\hat{antsrev}$ $\hat{tfur}$
  - 2. Combine: $\hat{antsrev-a-tfur}$
  - 2.5 $h?$: $[\hat{antsrev-a-tfur}]_h$
  - 3. Add PL: $\hat{antsrev-a-tfur}$
**Head-marking**

- **PL is a head-operation**
  - \( \text{PL}(X + h) = X + \text{PL}(h) \)
  - \( \text{PL} \rightarrow -er / [\sigma]_h \_
  - \( \text{PL} \rightarrow -ner / \text{elsewhere} \)

- **Semantic Head \( h \):**
  - If simplex/exocentric, then \( h=\text{WORD} \)
  - If endocentric, then \( h=\text{STEM2} \)

---

<table>
<thead>
<tr>
<th>Exo</th>
<th>Endo</th>
</tr>
</thead>
<tbody>
<tr>
<td>'evil-hearted'</td>
<td>'rain-water(s)'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>Exo</th>
<th>Endo</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{far} + \text{sírd} + \text{PL} )</td>
<td>( \text{antsrev} + \text{fur} + \text{PL} )</td>
<td></td>
</tr>
<tr>
<td>1. Stems</td>
<td>( \text{far} \quad \text{sírd} )</td>
<td>( \text{antsrév} + \text{fúr} )</td>
</tr>
<tr>
<td>2. Combine</td>
<td>( \text{far-a-sírd} )</td>
<td>( \text{antsrev-a-fúr} )</td>
</tr>
<tr>
<td>2.5 ( h )?</td>
<td>( [\text{far-a-sírd}]_h )</td>
<td>( \text{antsrev-a-[fúr]}_h )</td>
</tr>
<tr>
<td>3. Add PL</td>
<td>( \text{far-a-sírd-nér} )</td>
<td></td>
</tr>
</tbody>
</table>
**Head-marking**

- **PL is a head-operation**
  - PL\(X + h) = X + PL(h)
  - \(PL \rightarrow -er \)/ \([\sigma]_h\) 
  - \(PL \rightarrow -ner \)/ elsewhere

- **Semantic Head \(h\):**
  - If simplex/exocentric, then \(h=\text{WORD}\)
  - if endocentric, then \(h=\text{STEM2}\)

<table>
<thead>
<tr>
<th>Exo</th>
<th>Endo</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘evil-hearted’</td>
<td>‘rain-water(s)’</td>
</tr>
<tr>
<td>Input</td>
<td>(\hat{t}f\hat{a}r + s\hat{ir}d + PL)</td>
</tr>
<tr>
<td>1. Stems</td>
<td>(\hat{t}f\hat{a}r) (\hat{s}\hat{ir}d)</td>
</tr>
<tr>
<td>2. Combine</td>
<td>(\hat{t}f\hat{a}r-a-\hat{s}\hat{ir}d)</td>
</tr>
<tr>
<td>2.5 (h?)</td>
<td>([\hat{t}f\hat{a}r-a-\hat{s}\hat{ir}d])_h</td>
</tr>
<tr>
<td>3. Add PL</td>
<td>(\hat{t}f\hat{a}r-a-s\hat{ir}d-n\hat{e}r)</td>
</tr>
</tbody>
</table>
Irregular morphology

- Head-marking tends to be all-or-nothing:
  - If one Infl. rule is head-marking, all Infl. rules are (Stump, 1995)
  - Endocentric $\rightarrow$ paradoxical plural + irregular morphology
Head-marking tends to be all-or-nothing:

- If one Infl. rule is head-marking, all Infl. rules are (Stump, 1995)
- Endocentric → paradoxical plural + irregular morphology

‘mother’

majr

PL majr-er
Head-marking tends to be all-or-nothing:
  - If one Infl. rule is head-marking, all Infl. rules are (Stump, 1995)
  - Endocentric → paradoxical plural + irregular morphology

'mother'

\[ \text{majr} \]
PL \[ \text{majr}-\text{er} \]
Reg GEN \[ \text{majr}-\text{i} \]
IRREGULAR MORPHOLOGY

- Head-marking tends to be all-or-nothing:
  - If one Infl. rule is head-marking, all Infl. rules are (Stump, 1995)
  - Endocentric $\rightarrow$ paradoxical plural $+$ irregular morphology

‘mother’

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>majr</td>
<td>majr-er</td>
</tr>
<tr>
<td>PL</td>
<td></td>
</tr>
<tr>
<td>Reg GEN</td>
<td>majr-i</td>
</tr>
<tr>
<td>Irreg GEN</td>
<td>mor</td>
</tr>
</tbody>
</table>
Irregular morphology

- Head-marking tends to be all-or-nothing:
  - If one Infl. rule is head-marking, all Infl. rules are (Stump, 1995)
  - Endocentric → paradoxical plural + irregular morphology

<table>
<thead>
<tr>
<th>English</th>
<th>Arabic</th>
</tr>
</thead>
<tbody>
<tr>
<td>'mother'</td>
<td>مادر</td>
</tr>
<tr>
<td>'seal + mother'</td>
<td>'god-mother'</td>
</tr>
<tr>
<td>majr</td>
<td>gənk-a-[majr]h</td>
</tr>
<tr>
<td>majr-er</td>
<td>gənk-a-majr-er</td>
</tr>
<tr>
<td>majr-i</td>
<td>gənk-a-majr-i</td>
</tr>
<tr>
<td>mor</td>
<td></td>
</tr>
</tbody>
</table>

Semantic Head h is also Morphological Head
Irregular morphology

- Head-marking tends to be all-or-nothing:
  - If one Infl. rule is head-marking, all Infl. rules are (Stump, 1995)
  - Endocentric → paradoxical plural + irregular morphology

<table>
<thead>
<tr>
<th>'mother'</th>
<th>'seal + mother'</th>
</tr>
</thead>
<tbody>
<tr>
<td>majr</td>
<td>gənk-a-[majr]_{h}</td>
</tr>
<tr>
<td>PL</td>
<td>majr-er</td>
</tr>
<tr>
<td>Reg GEN</td>
<td>majr-i</td>
</tr>
<tr>
<td>Irreg GEN</td>
<td>mor</td>
</tr>
</tbody>
</table>
Head-marking tends to be all-or-nothing:

- If one Infl. rule is head-marking, all Infl. rules are (Stump, 1995)
- Endocentric $\rightarrow$ paradoxical plural + irregular morphology

<table>
<thead>
<tr>
<th>'mother'</th>
<th>'seal + mother'</th>
<th>'country + mother'</th>
</tr>
</thead>
<tbody>
<tr>
<td>majr</td>
<td>$\emptyset$majr$\emptyset_h$</td>
<td>$\emptyset$kar-majr$\emptyset_h$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PL</th>
<th>majr-er</th>
<th>kar-majr-ner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reg GEN</td>
<td>majr-i</td>
<td>kar-majr-i</td>
</tr>
<tr>
<td>Irreg GEN</td>
<td>mor</td>
<td>kar-mor</td>
</tr>
</tbody>
</table>
Irregular morphology

- Head-marking tends to be all-or-nothing:
  - If one Infl. rule is head-marking, all Infl. rules are (Stump, 1995)
  - Endocentric $\rightarrow$ paradoxical plural $+$ irregular morphology

<table>
<thead>
<tr>
<th></th>
<th>‘mother’</th>
<th>‘seal + mother’</th>
<th>‘country + mother’</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>majr</td>
<td>$g\emptyset k-a-[majr]_h$</td>
<td>$[kayak-a-majr]_h$</td>
</tr>
<tr>
<td>PL</td>
<td>majr-er</td>
<td>$g\emptyset nk-a-majr-er$</td>
<td>$kayak-a-majr-ner$</td>
</tr>
<tr>
<td>Reg GEN</td>
<td>majr-i</td>
<td>$g\emptyset nk-a-majr-i$</td>
<td>$kayak-a-majr-i$</td>
</tr>
<tr>
<td>Irreg GEN</td>
<td>mor</td>
<td>$g\emptyset nk-a-mor$</td>
<td></td>
</tr>
</tbody>
</table>

$\rightarrow$ Semantic Head $h$ is also Morphological Head $m$
  - Morphological Head $= \text{locus of irregular inflection}$
**Long-story short:**
- Armenian compounds use head-marking
- Semantic + Morphological Heads

<table>
<thead>
<tr>
<th>COUNTER-CYCLIC</th>
<th>CYCLIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rebracketing</td>
<td>Head</td>
</tr>
<tr>
<td>Late Adjunction</td>
<td>Prosodic operations</td>
</tr>
<tr>
<td>Head-based paradigms</td>
<td>constituents</td>
</tr>
<tr>
<td>Sproat ‘85</td>
<td>Hoeksema ‘84</td>
</tr>
<tr>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>Newell ‘05</td>
<td>Nespor&amp;Vogel ‘87</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
## Interim Heads

- **Long-story short:**
  - Armenian compounds use head-marking
  - Semantic + Morphological Heads

### COUNTER-CYCLIC  Cyclic

<table>
<thead>
<tr>
<th>Rebracketing</th>
<th>Late Adjunction</th>
<th>Head-based paradigms</th>
<th>Head operations</th>
<th>Prosodic constituents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sproat ‘85</td>
<td>Newell ‘05</td>
<td>Stump ‘01</td>
<td>Hoeksema ‘84</td>
<td>Nespor&amp;Vogel ‘87</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
<td>?</td>
<td>✓</td>
<td>?</td>
</tr>
</tbody>
</table>

- **Analysis is cyclic:**
  - Morphemes are added based *exactly* on their meaning
  - But what about if you add morphemes in the ‘wrong’ order
Cyclic or Counter-cyclic?

- Cyclic vs. counter-cyclic?
  - Exocentric: add PL after compounding
  - Endocentric: add STEM1 ‘late’

**Exo:** ‘evil-hearted’

**Input**

1. Stems
   -  ṭfár sírd

2. Combine
   -  ṭfár-á-sírd

3. Add PL
   -  ṭfár-á-sírd-nér
Cyclic or Counter-cyclic?

- Cyclic vs. counter-cyclic?
  - Exocentric: add PL after compounding
  - Endocentric: add STEM1 ‘late’

Exo: ‘evil-hearted’
Input $tʃar + sird + PL$
1. Stems $tʃár sírd$
2. Combine $tʃar-a-sírd$
Cyclic or Counter-cyclic?

- Cyclic vs. counter-cyclic?
  - Exocentric: add PL after compounding
  - Endocentric: add STEM1 ‘late’

Exo: ‘evil-hearted’

Input \( t\dot{f}ar + sird + PL \)

1. Stems \( t\dot{f}\acute{a}r \ sírd \)
2. Combine \( t\dot{f}ar-a-sírd \)
3. Add PL \( t\dot{f}ar-a-sírd-nér \)
Cyclic or Counter-cyclic?

- Cyclic vs. counter-cyclic?
  - Exocentric: add PL after compounding
  - Endocentric: add STEM1 ‘late’

Exo: ‘evil-hearted’

Input $t\bar{s}ar + s\acute{r}d + PL$

1. Stems $t\bar{s}\acute{a}r$ $s\acute{r}d$
2. Combine $t\bar{s}ar-a-s\acute{r}d$
3. Add PL $t\bar{s}ar-a-s\acute{r}d-n\acute{e}r$
Cyclic or Counter-cyclic?

- Cyclic vs. counter-cyclic?
  - Exocentric: add PL after compounding
  - Endocentric: add STEM1 ‘late’

Exo: ‘evil-hearted’

Input: \( \text{tʃ} \text{ar} + \text{sird} + \text{PL} \)

1. Stems: \( \text{tʃá} \text{r} \text{síd} \)
2. Combine: \( \text{tʃ} \text{ar-a-sírd} \)
3. Add PL: \( \text{tʃ} \text{ar-a-sírd-nér} \)

Eno: ‘rain-water(s)’

Input: \( \text{antsrév} + \text{tʃur} + \text{PL} \)

1. Stems: \( \text{antsrév} \text{tʃúr} \)

- Works for the paradox, but not for compounds in general

\( \rightarrow \) Compounds are formed cyclically because of compound phonology!
Cyclic or Counter-cyclic?

- Cyclic vs. counter-cyclic?
  - Exocentric: add PL after compounding
  - Endocentric: add STEM1 ‘late’

Exo: ‘evil-hearted’
Input: $\tilde{t}far + sird + PL$
1. Stems: $\tilde{t}far$ $sird$
2. Combine: $\tilde{t}far-a-sird$
3. Add PL: $\tilde{t}far-a-sird-néř$

Endo: ‘rain-water(s)’
Input: $\tilde{ants}rev + \tilde{t}fur + PL$
1. Stems: $\tilde{ants}rév$ $\tilde{t}úr$
2. Combine: $\tilde{ants}rev-a-\tilde{t}fur$-ér
3. Add PL: $\tilde{ants}rév$ $\tilde{t}fur-ér$
Cyclic or Counter-cyclic?

- Cyclic vs. counter-cyclic?
  - Exocentric: add PL after compounding
  - Endocentric: add STEM1 ‘late’

Exo: ‘evil-hearted’

Input
1. Stems $\hat{t}s\acute{a}r \ s\acute{i}rd$
2. Combine $\hat{t}s\acute{a}r-a-s\acute{i}rd$
3. Add PL $\hat{t}s\acute{a}r-a-s\acute{i}rd-n\acute{e}r$

ENDO: ‘rain-water(s)’

Input
1. Stems $\hat{a}ntsr\acute{e}v \ t\acute{f}\acute{u}r$
2. Combine $\hat{a}ntsr\acute{e}v-a-t\acute{f}\acute{u}r-\acute{e}r$
3. Add PL $\hat{a}ntsr\acute{e}v \ t\acute{f}\acute{u}r-\acute{e}r$
Cyclic or Counter-cyclic?

- Cyclic vs. counter-cyclic?
  - Exocentric: add PL after compounding
  - Endocentric: add STEM1 ‘late’

Exo: ‘evil-hearted’
Input $\hat{t}f\acute{a}r + s\acute{i}rd + \text{PL}$
1. Stems $\hat{t}f\acute{a}r$ $s\acute{i}rd$
2. Combine $\hat{t}f\acute{a}r-a-s\acute{i}rd$
3. Add PL $\hat{t}f\acute{a}r-a-s\acute{i}rd-n\acute{e}r$

ENDO: ‘rain-water(s)’
Input $\text{antsrev} + \hat{t}f\acute{u}r + \text{PL}$
1. Stems $\text{antsr}\acute{e}v$ $\hat{t}f\acute{u}r$
3. Add PL $\text{antsr}\acute{e}v$ $\hat{t}f\acute{u}r\acute{e}r$
2. Combine $\text{antsrev-a-}\hat{t}f\acute{u}r\acute{e}r$

- Works for the paradox, but not for compounds in general
  → Compounds are formed cyclically because of compound phonology!
Cyclic compounding

- Armenian shows stem-level vs. word-level strata
  - Some processes apply in derivation but not inflection
  - e.g., destressed high vowel reduction

- Reduction is stem-level, not word-level
  - \text{tS\'ur} 'water'
  - \text{azniv} 'sincere'
  - \text{Der:} \text{tS\'r-aj\'in} 'aquatic'
  - \text{Inf:} \text{tSur-\' ov} 'with water'
  - \text{azniv-e} 'from sincere (one)'

- Reduction applies in compounds too!
  - \text{tS\'ur + p\'os} 'water + hole'
  - \text{azniv + sird} 'sincere+heart'
  - \text{tS\'r-a-p\'os} 'water-hole'
  - \text{azniv-a-s\'ird} 'sincere-hearted'
  - \text{tS\'r-a-pos} -\text{\'er}
    - \text{azniv-a-s\'ird} -\text{\'er}
Cyclic compounding

- Armenian shows stem-level vs. word-level strata
  - Some processes apply in derivation but not inflection
  - e.g., destressed high vowel reduction
- Reduction is stem-level, not word-level
  
  \[
  \begin{align*}
  \text{tʃúr} & \quad \text{‘water’} & \text{azniv} & \quad \text{‘sincere’} \\
  \text{Der} \quad \text{tʃər-aʃín} & \quad \text{‘aquatic’} & \text{aznəv-utjun} & \quad \text{‘sincerity’} \\
  \text{Inf} \quad \text{tʃur-óv} & \quad \text{‘with water’} & \text{azniv-ə} & \quad \text{‘from sincere (one)’}
  \end{align*}
  \]
Cyclic compounding

- Armenian shows stem-level vs. word-level strata
  - Some processes apply in derivation but not inflection
  - e.g., destressed high vowel reduction
- Reduction is stem-level, not word-level

\[
\begin{align*}
\text{tʃúr} & \quad \text{‘water’} & \text{azniv} & \quad \text{‘sincere’} \\
\text{Der} \quad \text{tʃəɾ-ajín} & \quad \text{‘aquatic’} & \text{aznəv-utjun} & \quad \text{‘sincerity’} \\
\text{Inf} \quad \text{tʃur-óv} & \quad \text{‘with water’} & \text{azniv-e} & \quad \text{‘from sincere (one)’}
\end{align*}
\]

- Reduction applies in compounds too!

\[
\begin{align*}
\text{tʃúr} + \text{pós} & \quad \text{‘water + hole’} & \text{azniv} + \text{sird} & \quad \text{‘sincere + heart’} \\
\text{tʃəɾ-a-pós} & \quad \text{‘water-hole’} & \text{aznəv-a-sírd} & \quad \text{‘sincere-hearted’}
\end{align*}
\]
Cyclic compounding

- Armenian shows stem-level vs. word-level strata
  - Some processes apply in derivation but not inflection
    - e.g., destressed high vowel reduction
- Reduction is stem-level, not word-level

\[
\begin{align*}
\text{Der} & \quad tf\text{\textae}r-aj'\text{i}n & \text{‘aquatic’} & \quad azn\text{\textae}v-utjun & \text{‘sincerity’} \\
\text{Inf} & \quad tf\text{\textae}r-\text{\textoe}v & \text{‘with water’} & \quad azniv-e & \text{‘from sincere (one)’}
\end{align*}
\]

- Reduction applies in compounds too!

\[
\begin{align*}
\text{Der} & \quad tf\text{\textae}r-a-p\text{\textae}s & \text{‘water-hole’} & \quad azn\text{\textae}v-a-s'\text{\textae}rd & \text{‘sincere-hearted’} \\
\text{Inf} & \quad tf\text{\textae}r-a-p\text{\textoe}s-\text{\textae}r & \quad azn\text{\textae}v-a-s'\text{\textae}rd-ner
\end{align*}
\]

- A lot of other stem-level processes apply in compounding too
**Cyclic compounding**

- So what?
  - The paradox cannot be counter-cyclic
Cyclic compounding

- So what?
  - The paradox cannot be counter-cyclic

- Why?
  - (2) Stems must combine cyclically to trigger stem-level phonology
  - (3) PL is added later and triggers word-level phonology

Exo

‘sincere-hearted’

Input: azniv + sird + PL

1. Stems: aznív s´ird
2. Combine: aznəv-a-s´ird
3. Add PL: aznəv-a-sird-nér

Endo

‘water-hole(s)’

Input: tʃur + pos + PL

1. Stems: tʃúr p´os
2. Combine: tʃər-a-pós
3. Add PL: tʃər-a-pos-ér
Cyclic compounding

- So what?
  - The paradox cannot be counter-cyclic

- Why?
  - (2) Stems must combine cyclically to trigger stem-level phonology
  - (3) PL is added later and triggers word-level phonology

Exo

- ‘sincere-hearted’
- Input: azniv + sird + PL
- 1. Stems: aznív sîrd
- 2. Combine: aznəv-a-sîrd
- 3. Add PL: aznəv-a-sîrd-nér

Endo

- ‘water-hole(s)’
- Input: ðfur + pos + PL
- 1. Stems: ðfur pós
- 2. Combine: ðfur-a-pós
- 3. Add PL: ðfur-a-pos-ér

- Head-operation: PL can check if STEM2 is the head or not
Lots of theories for paradoxes because...
- Armenian doesn’t work with counter-cyclic theories
- Armenian uses cyclic head-operations

<table>
<thead>
<tr>
<th>Counter-cyclic</th>
<th>Cyclic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rebracketing</td>
<td>Head</td>
</tr>
<tr>
<td>Late Adjunction</td>
<td>operations</td>
</tr>
<tr>
<td>Head-based paradigms</td>
<td>Prosodic constituents</td>
</tr>
<tr>
<td>Sproat ‘85</td>
<td>Hoeksema ‘84</td>
</tr>
<tr>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Newell ‘05</td>
<td>Nespor &amp; Vogel ‘87</td>
</tr>
<tr>
<td>X</td>
<td>?</td>
</tr>
</tbody>
</table>
**Interim theory**

- Lots of theories for paradoxes because...
  - Armenian doesn’t work with counter-cyclic theories
  - Armenian uses cyclic head-operations

<table>
<thead>
<tr>
<th><strong>Counter-cyclic</strong></th>
<th><strong>Cyclic</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Rebracketing</td>
<td>Head</td>
</tr>
<tr>
<td>Late Adjunction</td>
<td>Prosodic operations</td>
</tr>
<tr>
<td>Head-based paradigms</td>
<td>constituents</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sproat ‘85</th>
<th>Newell ‘05</th>
<th>Stump ‘01</th>
<th>Hoeksema ‘84</th>
<th>Nespor&amp;Vogel ‘87</th>
</tr>
</thead>
<tbody>
<tr>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>?</td>
</tr>
</tbody>
</table>

- Twist: variation data also shows use for *prosodic constituents*
Old data...

- \( \text{t}f\text{ar} + \text{sird} \) ‘evil + heart’
- \( \text{antsrev} + \text{t}f\text{ur} \) ‘rain + water’
- \( \text{t}f\text{ar-a-sird-ner} \) ‘evil-hearted’
- \( \text{antsrev-a-t}f\text{ur-er} \) ‘rain-water(s)’
Interim Heads

• Old data...
  \(t\text{far} + s\text{ird}\) ‘evil + heart’  \(a\text{ntsrev} + t\text{fur}\) ‘rain + water’
  \(t\text{far-a-sird-ner}\) ‘evil-hearted’  \(a\text{ntsrev-a-t\text{fur-er}}\) ‘rain-water(s)’

• Twist:
  • All you saw were 3-syllabic compounds
  • Bisyllabic endo compounds show variation

Exo
  \(k\text{ar} + d\text{af-el}\) ‘stone + to carve’
  \(k\text{ar-daf}\) ‘stone carver’

  \(k\text{ar-daf-ner}\)
Old data...

\( \tilde{t}_f\text{ar} + \text{sird} \) ‘evil + heart’ \( \text{antsrev} + \tilde{t}_f\text{ur} \) ‘rain + water’

\( \tilde{t}_f\text{ar-a-sird-ner} \) ‘evil-hearted’ \( \text{antsrev-a-\tilde{t}_f\text{ur-er}} \) ‘rain-water(s)’

Twist:

- All you saw were 3-syllabic compounds
- Bisyllabic endo compounds show variation

Exo

\( \text{kar} + \text{daf-el} \) ‘stone + to carve’ \( \text{xat}\tilde{f} + \text{kar} \) ‘cross + stone’

\( \text{kar-daf} \) ‘stone carver’ \( \text{xat}\tilde{f}-\text{kar} \) ‘cross-stone’

\( \text{kar-daf-ner} \)
**Interim Heads**

- Old data...

  \[
  \text{\textit{tʃar}} + \text{\textit{sird}} \quad \text{‘evil + heart’} \quad \text{\textit{antsrev}} + \text{\textit{tʃur}} \quad \text{‘rain + water’} \\
  \text{\textit{tʃar-a-sird-ner}} \quad \text{‘evil-hearted’} \quad \text{\textit{antsrev-a-tʃur-er}} \quad \text{‘rain-water(s)’}
  \]

- Twist:
  - All you saw were 3-syllabic compounds
  - Bisyllabic endo compounds show variation

  **Exo**
  - \text{\textit{kar}} + \text{\textit{dəf-el}} \quad \text{‘stone + to carve’} \quad \text{\textit{xatʃ} + \text{\textit{kar}}} \quad \text{‘cross + stone’}
  - \text{\textit{kar-داعش}} \quad \text{‘stone carver’} \quad \text{\textit{xatʃ-kar}} \quad \text{‘cross-stone’}
  - \text{\textit{kar-داعش-نر}}

  **Endo**
  - \text{\textit{xatʃ-کار}} \quad \text{‘cross-stone’} \quad \text{\textit{xatʃ-کار-نر}}

  → Bisyllabic endo compounds *optionally* get transparent plurals
Prosodic Heads

- Bisyllabic endo compounds optionally get transparent plurals
- Analysis: Prosodic Heads $p$
  1) Map $p$ \((...)_h \rightarrow (...)_p\)
  2) Optional restructuring \(#\sigma (\sigma)_p \rightarrow (\sigma \sigma)_p\)
  3) Count $\sigma$’s in $p$  
     PL \(\rightarrow -er / (\sigma)_p \_\)  
     PL \(\rightarrow -ner / \text{elsewhere}\)
Prosodic Heads

- Bisyllabic endo compounds *optionally* get transparent plurals
- Analysis: Prosodic Heads $p$
  1) Map $p$  
     \[(...)_h \rightarrow (...)_p\]
  2) Optional restructuring \[\#\sigma (\sigma)_p \rightarrow (\sigma\sigma)_p\]
  3) Count $\sigma$’s in $p$  
     \[PL \rightarrow -er / (\sigma)_p \_\]
     \[PL \rightarrow -ner / elsewhere\]

Exo:
- ‘evil-hearted’
- ‘stone-carver’
- $\textit{t}f\text{ar} + \textit{sird} + PL$
- $\textit{kar} + \textit{daf-} + PL$
Prosodic Heads

- Bisyllabic endo compounds *optionally* get transparent plurals
- Analysis: Prosodic Heads $p$
  1) Map $p$ $(...)_h \rightarrow (...)_p$
  2) Optional restructuring $\#\sigma \ (\sigma)_p \rightarrow (\sigma \sigma)_p$
  3) Count $\sigma$’s in $p$ $PL \rightarrow -er / (\sigma)_p$
- Exo: ‘evil-hearted’ ‘stone-carver
  $\widehat{t\text{far}} + s\text{ird} + PL \quad \text{kar} + daf- + PL$
1. Stems $\widehat{t\text{far}} + s\text{ird} \quad \text{kar} + daf$
Prosodic Heads

- Bisyllabic endo compounds optionally get transparent plurals
- Analysis: Prosodic Heads $p$
  1) Map $p$  \((...)_h \rightarrow (...)_p\)
  2) Optional restructuring  \#\sigma ( \sigma )_p \rightarrow ( \sigma \sigma )_p\)
  3) Count $\sigma$'s in $p$  PL $\rightarrow -er / ( \sigma )_p _-$
      PL $\rightarrow -ner / elsewhere$

Exo: ‘evil-hearted’ ‘stone-carver
   \(\widehat{t\text{far}} + \text{sird} + \text{PL}\)  kar + daf- + PL
1. Stems \(\widehat{t\text{far}} + \text{sird}\)  kar + daf
2. Combine \(\widehat{t\text{far-a-sird}}\)  kar-daf
Prosodic Heads

- Bisyllabic endo compounds optionally get transparent plurals
- Analysis: Prosodic Heads $p$
  1) Map $p$ $(...)_h\rightarrow (...)_p$
  2) Optional restructuring $\#\sigma (\sigma)_p \rightarrow (\sigma \sigma)_p$
  3) Count $\sigma$’s in $p$ PL $\rightarrow -er / (\sigma)_p$

Exo:

‘evil-hearted’ ‘stone-carver’

$t\hat{f}ar + sird + PL$ $kar + daf- + PL$

1. Stems $t\hat{f}ar + sird$ $kar + daf$
2. Combine $t\hat{f}ar-a-sird$ $kar-daf$

$h$? $[t\hat{f}ar-a-sird]_h$ $[kar-daf]_h$
Prosodic Heads

- Bisyllabic endo compounds *optionally* get transparent plurals
- Analysis: Prosodic Heads $p$
  1) Map $p$  
     \((...)_h\) $\rightarrow$ \((...)_p\)
  2) Optional restructuring  
     \(\#\sigma (\sigma)_p\) $\rightarrow$ \((\sigma \sigma)_p\)
  3) Count $\sigma$’s in $p$  
     PL $\rightarrow$ -er / \((\sigma)_p\) _
     PL $\rightarrow$ -ner / elsewhere

Exo:  
‘evil-hearted’  ‘stone-carver’
\(\hat{t}\)far + sird + PL  kar + da\$- + PL

1. Stems  
\(\hat{t}\)far + sird  kar + da\$

2. Combine  
\(\hat{t}\)far-a-sird  kar-daf

  $h$?  
  \([\hat{t}\)far-a-sird$]_h\)  \([kar-daf]_h\)

  $p$?  
  \((\hat{t}\)far-a-sird$)_p\)  \((kar-daf)_p\)
**Prosodic Heads**

- Bisyllabic endo compounds *optionally* get transparent plurals
- Analysis: Prosodic Heads $p$
  1) Map $p$  \[(...)_h \rightarrow (...)_p\]
  2) Optional restructuring \[\#\sigma (\sigma)_p \rightarrow (\sigma \sigma)_p\]
  3) Count $\sigma$’s in $p$  PL $\rightarrow -er / (\sigma)_p -$ PL $\rightarrow -ner /$ elsewhere

**Exo:**

‘evil-hearted’

‘stone-carver’

$t\hat{f}ar + \text{sird } + \text{PL}$

$\hat{t}\hat{f}ar + \text{sird}$

$kar + \text{daf- } + \text{PL}$

$kar + \text{daf}$

1. Stems

2. Combine $\hat{t}\hat{f}ar-a-\text{sird}$

$h? \quad [t\hat{f}ar-a-\text{sird}]_h$

$p? \quad (t\hat{f}ar-a-\text{sird})_p$

$kar-daf$

$[kar-daf]_h$

$(kar-daf)_p$

size?
Prosodic Heads

- Bisyllabic endo compounds optionally get transparent plurals
- Analysis: Prosodic Heads $p$
  1) Map $p$
     $\ldots)_h$ $\rightarrow (\ldots)_p$
  2) Optional restructuring
     $\#\sigma ( \sigma )_p$ $\rightarrow ( \sigma \sigma )_p$
  3) Count $\sigma$’s in $p$
     PL $\rightarrow -er / ( \sigma )_p$ _
     PL $\rightarrow -ner /$ elsewhere

Exo: ‘evil-hearted’ ‘stone-carver
\[
\text{\textasciitilde t}f\text{ar} + \text{sird} + \text{PL} \quad \text{kar} + \text{daf-} + \text{PL}
\]
1. Stems $\text{\textasciitilde t}f\text{ar} + \text{sird}$ $\text{kar} + \text{daf}$
2. Combine $\text{\textasciitilde t}f\text{ar-a-sird}$ $\text{kar-daf}$
   $h?$ $[\text{\textasciitilde t}f\text{ar-a-sird}]_h$ $[\text{kar-daf}]_h$
   $p?$ $(\text{\textasciitilde t}f\text{ar-a-sird})_p$ $(\text{kar-daf})_p$
   size?
3. Add PL $\text{\textasciitilde t}f\text{ar-a-sird-ner}$ $\text{kar-daf-ner}$
Prosodic Heads

- Bisyllabic endo compounds optionally get transparent plurals
- Analysis: Prosodic Heads \( p \)
  1) Map \( p \) \((...)_h \rightarrow (...)_p\)
  2) Optional restructuring \#\( \sigma \) \((\sigma)_p \rightarrow (\sigma \sigma)_p\)
  3) Count \( \sigma \)'s in \( p \) \(PL \rightarrow -er / (\sigma)_p \_\)
      \(PL \rightarrow -ner / \) elsewhere

Endo: ‘rain-water’ ‘cross-stone
  antsrev + t\(\text{fur}\) + PL xat\(\text{f}\) + kar + PL

1. Stems antsrev + t\(\text{fur}\) xat\(\text{f}\) + kar
Prosodic Heads

- Bisyllabic endo compounds *optionally* get transparent plurals
- Analysis: Prosodic Heads \( p \)

  1) Map \( p \) \( (\ldots)_h \rightarrow (\ldots)_p \)
  2) Optional restructuring \( \#_\sigma (\sigma)_p \rightarrow (\sigma \sigma)_p \)
  3) Count \( \sigma \)'s in \( p \) \( \text{PL} \rightarrow -er / (\sigma)_p \) \( -ner / \text{elsewhere} \)

Endo:

- ‘rain-water’
  
  \( \text{antsrev} + \text{t}fur + \text{PL} \)

- ‘cross-stone’
  
  \( \text{xatf} + \text{kar} + \text{PL} \)

1. Stems \( \text{antsrev} + \text{t}fur \)

2. Combine \( \text{antsrev}-a-\text{t}fur \)

\( \text{xatf-kar} \)
Prosodic Heads

- Bisyllabic endo compounds *optionally* get transparent plurals
- Analysis: Prosodic Heads $p$
  1) Map $p$ $\langle \ldots \rangle_h \rightarrow \langle \ldots \rangle_p$
  2) Optional restructuring $\#\sigma \langle \sigma \rangle_p \rightarrow \langle \sigma \sigma \rangle_p$
  3) Count $\sigma$’s in $p$ $PL \rightarrow -er / \langle \sigma \rangle_p$
      $PL \rightarrow -ner / \text{elsewhere}$

ENDO: ‘rain-water’ ‘cross-stone’

antsrev + tjur + PL xatf + kar + PL

1. Stems antsrev + tjur xatf + kar
2. Combine antsrev-a-tjur xatf-kar
   h? antsrev-a-[tjur]_h xatf-[kar]_h
Prosodic Heads

- Bisyllabic endo compounds optionally get transparent plurals

Analysis: Prosodic Heads $p$

1) Map $p$  
\[ (\ldots)_h \rightarrow (\ldots)_p \]

2) Optional restructuring  
\[ \#\sigma (\sigma)_p \rightarrow (\sigma \sigma)_p \]

3) Count $\sigma$’s in $p$  
\[
\begin{align*}
\text{PL} & \rightarrow -er / (\sigma)_p \\
\text{PL} & \rightarrow -ner / \text{elsewhere}
\end{align*}
\]

ENDO:  
‘rain-water’  
‘cross-stone’

\[
\begin{align*}
\text{antsrev} + \text{t}f\text{ur} + \text{PL} & \rightarrow \text{xatf} + \text{kar} + \text{PL} \\
\text{nton} & \rightarrow \text{xatf-kar} \\
\text{nton} & \rightarrow \text{xatf-[kar]}_h \\
\text{nton} & \rightarrow \text{xatf-(kar)}_p
\end{align*}
\]
Prosodic Heads

- Bisyllabic endo compounds *optionally* get transparent plurals
- Analysis: Prosodic Heads \( p \)
  1) Map \( p \) \( (\ldots)_h \rightarrow (\ldots)_p \)
  2) Optional restructuring \#\( \sigma (\sigma)_p \rightarrow (\sigma \sigma)_p \)
  3) Count \( \sigma \)'s in \( p \)\PL \rightarrow -er / (\sigma)_p _
      \PL \rightarrow -ner / elsewhere

Endo:

- ‘rain-water’
  \( \text{antsrev} + \text{t} \text{fur} + \text{PL} \)
  \( \text{xatf} + \text{kar} + \text{PL} \)

1. Stems
   \( \text{antsrev} + \text{t} \text{fur} \)
   \( \text{xatf} + \text{kar} \)

2. Combine
   \( \text{antsrev-a-t} \text{fur} \)
   \( \text{xatf-kar} \)
   \( \text{h?} \)
   \( \text{antsrev-a-[t} \text{fur}]_h \)
   \( \text{xatf-[kar]}_h \)
   \( \text{p?} \)
   \( \text{antsrev-a-(t} \text{fur})_p \)
   \( \text{xatf-(kar)}_p \)
   \( \text{size?} \)
   \( \text{xatf-(kar)}_p \)
   \( (\text{xatf-kar})_p \)
Prosodic Heads

- Bisyllabic endo compounds *optionally* get transparent plurals
- Analysis: Prosodic Heads $p$
  1) Map $p$ 
     $\text{(...)}_h \rightarrow \text{(...)}_p$
  2) Optional restructuring 
     $\#(\sigma \sigma)_p \rightarrow (\sigma \sigma)_p$
  3) Count $\sigma$’s in $p$ 
     PL $\rightarrow$ -er / $(\sigma)_p$
     PL $\rightarrow$ -ner / elsewhere

ENDO: 
‘rain-water’
‘cross-stone’

$\text{antsrev + tʃur + PL}$

1. Stems $\text{antsrev + tʃur}$

2. Combine $\text{antsrev-a-tʃur}$
   h? $\text{antsrev-a-[tʃur]}_h$
   p? $\text{antsrev-a-(tʃur)}_p$
   size? $\text{antsrev-a-(tʃur)}_p$

3. Add PL $\text{antsrev-a-tʃur-er}$

$\text{xatʃ + kar + PL}$

$\text{xatʃ + kar}$

$\text{xatʃ-kar}$

$\text{xatʃ-[kar]}_h$

$\text{xatʃ-(kar)}_p$

$\text{xatʃ-(kar)}_p$

$\text{xatʃ-kar-er}$

$\text{xatʃ-kar-ner}$
Prosodic Heads

- Bisyllabic endo compounds *optionally* get transparent plurals
- Analysis: Prosodic Heads \( p \)

1) Map \( p \) \( \rightarrow (\ldots)_h \rightarrow (\ldots)_p \)
2) Optional restructuring \( \#\sigma (\sigma)_p \rightarrow (\sigma \sigma)_p \)
3) Count \( \sigma \)'s in \( p \) \( \rightarrow -er / (\sigma)_p \)
   PL \( \rightarrow -ner / \) elsewhere

Endo:

- ‘rain-water' \( \text{antsrev} + \text{tʃur} + \text{PL} \)
- ‘cross-stone' \( \text{xatʃ} + \text{kar} + \text{PL} \)

1. Stems \( \text{antsrev} + \text{tʃur} \)
   \( \text{xatʃ} + \text{kar} \)
2. Combine \( \text{antsrev-a-tʃur} \)
   \( \text{xatʃ-kar} \)
   \( \text{h?} \) \( \text{antsrev-a-[tʃur]}_h \)
   \( \text{xatʃ-[kar]}_h \)
   \( \text{p?} \) \( \text{antsrev-a-(tʃur)}_p \)
   \( \text{xatʃ-(kar)}_p \)
   \( \text{size?} \) \( \text{xatʃ-(kar)}_p \)
   \( (\text{xatʃ-kar})_p \)
3. Add PL \( \text{antsrev-a-tʃur-er} \)
   \( \text{xatʃ-kar-er} \)
   \( \text{xatʃ-kar-ner} \)
What is $p$?

- Prosodic Head $p$: `rain-water(s)`
  - Foot ($F$ or $\Sigma$)
  - Prosodic Word ($\omega$ or $w$)
  - Prosodic Stem ($s$)

- Foot ($F$ or $\Sigma$)
- Prosodic Word ($\omega$ or $w$)

- FYI: Can't be a Foot or PWord → intermediate PStem (appendix)

\[ \text{\textit{antsrev-a-} \textit{tfur} -er} \quad \text{\textit{xatf-} \textit{kar} -er} \quad \text{\textit{xatf-kar} -ner} \]
What is $p$

- Prosodic Head $p$:
  - ‘rain-water(s)’
  - ‘cross-stone(s)’

- But what is $p$?
  - Prosodic Word ($\omega$ or $w$)
  - Foot ($F$ or $\Sigma$)

FYI: Can’t be a Foot or PWord $\rightarrow$ intermediate PStem (appendix)
What is $p$?

- Prosodic Head $p$:
  'rain-water(s)'
  ‘cross-stone(s)’

- But what is $p$?
  Prosodic Word ($\omega$ or $w$)

- FYI: Can’t be a Foot or PWord $\rightarrow$ intermediate PStem (appendix)
• In simplex words, PL is simple syllable-counting
  \[ \text{pag-er} \; \text{‘yards’} \quad \text{panag-ner} \; \text{‘armies’} \]

• In compounds, PL counts \( \sigma \)’s in either Compound or STEM2
  \[ \text{\( \hat{t} \text{far} + \text{sird} \)} \; \text{‘evil + heart’} \quad \text{ant\( \hat{s} \text{rev} + \text{\( \hat{t} \text{fur} \)} \)} \; \text{‘rain + water’} \]
  \[ \text{\( \hat{t} \text{far-a-sird-ner} \)} \; \text{‘evil-hearted’} \quad \text{ant\( \hat{s} \text{rev-a-\( \hat{t} \text{fur-er} \)} \)} \; \text{‘rain-water(s)’} \]

• Bracketing paradox:
  - **SEM**: PL scopes over Compound
  - **PHON**: But sometimes PL looks *inside* the compound into STEM2
Wrap-up

- PL is head-marking
  - \( PL(X + H) = X + PL(H) \)
  - \( PL \rightarrow -er / [\sigma]_h _\)
  - \( PL \rightarrow -ner / \text{elsewhere} \)
Wrap-up

- PL is head-marking
  - $\text{PL}(X + H) = X + \text{PL}(H)$
    - $\text{PL} \rightarrow -er / [\sigma]_h$ _
    - $\text{PL} \rightarrow -ner /$ elsewhere
- Necessarily cyclic because of stratal phonology
- Dissects different types of (nearly) isomorphic heads

<table>
<thead>
<tr>
<th>Semantic Head</th>
<th>Morphological Head</th>
<th>Prosodic Head</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endocentric</td>
<td>Irregular Infl</td>
<td>Bisyllabic minimality</td>
</tr>
</tbody>
</table>
Wrap-up

- PL is head-marking
  - \( PL(X + H) = X + PL(H) \)
  - \( PL \rightarrow -er / [\sigma]_h \ _ \)
  - \( PL \rightarrow -ner / \text{elsewhere} \)

- Necessarily cyclic because of stratal phonology
- Dissects different types of (nearly) isomorphic heads

<table>
<thead>
<tr>
<th>Semantic Head</th>
<th>Morphological Head</th>
<th>Prosodnic Head</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endocentric</td>
<td>Irregular Infl</td>
<td>Bisyllabic minimality</td>
</tr>
</tbody>
</table>

- Unique combinations of some theories, exclusion of others

<table>
<thead>
<tr>
<th>COUNTER-CYCCLIC</th>
<th>CYCLIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rebracketing</td>
<td>Head</td>
</tr>
<tr>
<td>Late Adjunction</td>
<td>Prosodic operations</td>
</tr>
<tr>
<td>Head-based paradigms</td>
<td>constituents</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sproat ‘85</th>
<th>Newell ‘05</th>
<th>Stump ‘01</th>
</tr>
</thead>
<tbody>
<tr>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hoeksema ‘84</th>
<th>Nespor &amp; Vogel ‘87</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
# Table of Contents

- **Armenian**
  - Preview
  - Distribution
  - Head operations
  - Cyclic or counter-cyclic?
  - Variation in prosody
  - Wrap-up

- **Computation**
  - What’s the link?
  - Computing morpho-phono
  - Computing with logic
  - Computing morpho-phononology with logic

- **End**
What did you see?... data on a bracketing paradox

\( \text{antsr}év + \text{tśúr} \) ‘rain + water’ \( \text{tśár} + \text{sird} \) ‘evil+heart’

\( \text{tśər-a-pos-ér} \) ‘rain-water’ \( \text{aznəv-a-sird-ner} \) ‘evil-hearted’

‘Phonology’?
Transition

- What did you see?... data on a bracketing paradox
  \( \text{antsrév} + \text{tfúr} \) ‘rain + water’ \( \text{tfár} + \text{sird} \) ‘evil+heart’
  \( \text{tfár-a-pos-ér} \) ‘rain-water’ \( \text{aznəv-a-sird-ner} \) ‘evil-hearted’
  
- ‘Phonology’?
    = alternations in the pronunciation of morphemes
    = what principles governed that alternation
Principles in morphophonology are based on...

1. Morphology: how to make a compound?
2. Prosody: what are the syllables?
3. Rule-domains: why did the /i/ disappear?
4. Derivation: what happens first?

But...
Principles in morphophonology are based on...

1. Morphology: how to make a compound?
2. Prosody: what are the syllables?
3. Rule-domains: why did the /i/ disappear?
4. Derivation: what happens first?

But...

- How are these principles combined?
- How are these combinations computed?
Rough typology of morphophonology

- A lot of different theories, but differences boil down to...
  - Representation: What are the Morphology + Prosody + Rule Domains
  - Organization: How are they combined and factorized
Rough Typology of Morphophonology

- A lot of different theories, but differences boil down to...
  - Representation: What are the Morphology + Prosody + Rule Domains
  - Organization: How are they combined and factorized
- Interactionist (Kiparsky, 1982) vs. Non-Interactionist (SPE)
A lot of different theories, but differences boil down to...
  - Representation: What are the Morphology + Prosody + Rule Domains
  - Organization: How are they combined and factorized

Interactionist (Kiparsky, 1982) vs. Non-Interactionist (SPE)

Input: \textit{root}

\textit{add}                   \\
\downarrow

\begin{align*}
\text{Morphology} \quad \text{Prosody} \quad \text{Phonology}
\end{align*}

\text{parse}                   \\
\downarrow

\text{apply} \quad \text{add}

\text{produce}               \\
\downarrow

Output: \textit{root}++
A lot of different theories, but differences boil down to...
  - Representation: What are the Morphology + Prosody + Rule Domains
  - Organization: How are they combined and factorized

Interactionist (Kiparsky, 1982) vs. Non-Interactionist (SPE)

Input: \( \text{root} \)

\[
\begin{align*}
\text{Morphology} & \quad \text{add} \\
\text{Prosody} & \quad \text{parse} \\
\text{Phonology} & \quad \text{apply} \\
\text{Output: root++} & \quad \text{produce}
\end{align*}
\]

Input: \( \text{root} \)

\[
\begin{align*}
\text{Morphology} & \quad \text{add} \\
\text{Prosody} & \quad \text{parse} \\
\text{Phonology} & \quad \text{apply} \\
\text{Output: root++} & \quad \text{produce}
\end{align*}
\]
Rough typology of morphophonology

- A lot of different theories, but differences boil down to...
  - Representation: What are the Morphology + Prosody + Rule Domains
  - Organization: How are they combined and factorized
- Interactionist (Kiparsky, 1982) vs. Non-Interactionist (SPE)

```
Input: root
       ↓ add
Morphology       ↖
       ↓ parse
Prosody
       ↓ apply
Phonology
       ↓ produce
Output: root++
```

```
Input: root
       ↓ add
Morphology
       ↓ parse
Prosody
       ↓ apply
Phonology
       ↓ produce
Output: root++
```
Rough typology of morphophonology

- Interactionist vs. Non-Interactionist

Input: \( root \)

- add
- parse
- apply
- produce

Morphology \( \rightarrow \)

Prosody \( \rightarrow \)

Phonology \( \rightarrow \)

Output: \( root^{++} \)

Input: \( root \)

- add
- parse
- apply
- produce

Morphology \( \rightarrow \)

Prosody \( \rightarrow \)

Phonology \( \rightarrow \)

Output: \( root^{++} \)

root or more?

- add
- tree? set?
- recursive?
- trigger?
Rough typology of morphophonology

- Interactionist vs. Non-Interactionist
  - Input: \( \text{root} \)
    - Morphology: add
    - Prosody: parse
    - Phonology: apply
    - Output: \( \text{root}^{++} \)

- A lot of choices in representation
Rough typology of morphophonology

- Interactionist vs. Non-Interactionist
  - Input: \( \text{root} \)
    - Add
    - Morphology
    - Parse
    - Prosody
    - Apply
    - Phonology
    - Produce
    - Output: \( \text{root}++ \)

- A lot of choices in representation
- A lot of mix-matching between extremes
• Interactionist vs. Non-Interactionist

Input: \textit{root}

- add
- parse
- apply
- produce

Output: \textit{root++}

\begin{itemize}
  \item Morphology
  \item Prosody
  \item Phonology
\end{itemize}

\begin{itemize}
  \item add
  \item tree? set?
  \item recursive?
  \item trigger?
\end{itemize}

\begin{itemize}
  \item A lot of choices in representation
  \item A lot of mix-matching between extremes
  \item Do different representations+organizations have different computations?
\end{itemize}
Generic meaning: how to compute or *determine* something
What is computation

- Generic meaning: how to compute or determine something

Technical Meaning
1. What algorithms do we need?  
   Applied to Linguistics  
   What structures do we want?

- Background (1): descriptive + theoretical morphophonology
- Ultimate goal: learning the grammar (5)
- Major stepping stone (2): definition
What is computation

- Generic meaning: how to compute or determine something

**Technical Meaning**
1. What algorithms do we need?
2. How are they defined?

**Applied to Linguistics**
- What structures do we want?
- How do we generate them?

Focus on (1-3)

Background (1): descriptive + theoretical morphophonology

Ultimate goal: learning the grammar (5)

Major stepping stone (2): definition

But how do we define structure in the first place?

Then (3) generative capacity ...

Linguistic processes are often 'local' (=respect adjacency)
What is computation

- Generic meaning: how to compute or *determine* something

Technical Meaning

1. What algorithms do we need?  
2. How are they defined?  
3. How measurably complex are they?

Applied to Linguistics

What structures do we want?  
How do we generate them  
What’s their generative capacity?

Focus on (1-3)  
Background (1): descriptive + theoretical morphophonology  
Ultimate goal: learning the grammar (5)  
Major stepping stone (2): definition  
But how do we define structure in the first place?  
Then (3) generative capacity ...  
linguistic processes are often ‘local’ (=respect adjacency)
What is computation

- Generic meaning: how to compute or *determine* something

**Technical Meaning**
1. What algorithms do we need?
2. How are they defined?
3. How measurably complex are they?
4. How can we implement them?

**Applied to Linguistics**
- What structures do we want?
- How do we generate them?
- What’s their generative capacity?
- How do we test them?
What is computation

- Generic meaning: how to compute or *determine* something

**Technical Meaning**
1. What algorithms do we need?
2. How are they defined?
3. How measurable complex are they?
4. How can we implement them?
5. Use it

**Applied to Linguistics**
1. What structures do we want?
2. How do we generate them?
3. What’s their generative capacity?
4. How do we test them?
5. How can we learn grammars?
What is computation

- Generic meaning: how to compute or *determine* something

Technical Meaning

1. What algorithms do we need?
2. How are they defined?
3. How measurably complex are they?
4. How can we implement them?
5. Use it

- Focus on (1-3)
- Background (1): descriptive + theoretical morphophonology
- Ultimate goal: learning the grammar (5)

Applied to Linguistics

- What structures do we want?
- How do we generate them
- What’s their generative capacity?
- How do we test them?
- How can we learn grammars?
What is computation

- Generic meaning: how to compute or determine something

Technical Meaning
1. What algorithms do we need? What structures do we want?
2. How are they defined? How do we generate them?
3. How measurably complex are they? What’s their generative capacity?
4. How can we implement them? How do we test them?
5. Use it How can we learn grammars?

- Focus on (1-3)
- Background (1): descriptive + theoretical morphophonology
- Ultimate goal: learning the grammar (5)
- Major stepping stone (2): definition
  - But how do we define structure in the first place?
- Then (3) generative capacity ...
  - linguistic processes are often ‘local’ (=respect adjacency)
• **Finite-State Transducers:** State-of-the-art in computational morpho/phono
  ‣ efficient, versatile, and works in practice but designed for *linear* systems
**Finite-State Transducers:** State-of-the-art in computational morpho/phonology
  - efficient, versatile, and works in practice but designed for *linear* systems

- Theory is hierarchical
**Finite-State Transducers:** State-of-the-art in computational morpho/phonology

- efficient, versatile, and works in practice but designed for linear systems

**Theory is hierarchical**

... so any hierarchy must be flattened with boundaries

$\#(w(f\hat{\text{ants}}re)_f.v -LV a-LV. (s(s\hat{t}fur)_f)_s)_w\#$
FSTs and boundaries

- FSTs must reduce hierarchies to linear systems
  \[ #(w(s(\text{f\text{a\text{n\text{t\text{s\text{.\text{r\text{e\text{)}}f.v}}}} -_{LV} a_{LV}. (s(s(\text{f\text{u\text{r\text{)}}f}}}_s)w \# \]

- Still works, but clunky: gotta keep track of right number of boundaries

---

I like playing with the clunky parts
FSTs and Boundaries

- FSTs must reduce hierarchies to linear systems
  \[ \#(w(s(f\text{ants.re})f).v-LV a-LV. (s(stfur)f)_s)_w \# \]
- Still works, but clunky: gotta keep track of right number of boundaries
- Iconic: is it an faithful recap of linguistic theory?
  - Something seems ‘local’ in the tree but long-distance in the linear string
  - Unclear if expressivity of the linear system and of the hierarchical system 
    match

\[^3\]I like playing with the clunky parts
FSTs and Boundaries

- FSTs must reduce hierarchies to linear systems
  
  $$\#(w(s(f\text{ants.re})_f.v-\text{LV} a-\text{LV}. (s(stfur)_f)_s)_w\#$$

- Still works, but clunky: gotta keep track of right number of boundaries

- Iconic: is it an faithful recap of linguistic theory?
  
  - Something seems ‘local’ in the tree but long-distance in the linear string
  - Unclear if expressivity of the linear system and of the hierarchical system match

- General clunkiness happens in other areas of computational morpho/phono\(^3\)
  
  - reduplication (Dolatian und Heinz, 2018)
  - templates (Dolatian und Rawski, 2020)
  - tone (Rawski und Dolatian, 2020)
  - ...

\(^3\)I like playing with the clunky parts
Computing with logic

- How can we computationally define linguistic structure?
Computing with logic

- How can we computationally define linguistic structure?
  - *formal logic* over graph-to-graph transductions
- Why *formal logic*?

Higher the complexity, harder to learn

A lot of morpho is *QF* based on choices in organization
Computing with logic

- How can we computationally define linguistic structure?
  - *formal logic* over graph-to-graph transductions
- Why *formal logic*?
  - versatile enough to directly encode linguistic representation
  - explicit enough to test
  - systematic enough to measure complexity
- Logical complexity?
How can we computationally define linguistic structure?

→ *formal logic* over graph-to-graph transductions

Why *formal logic*?

- versatile enough to directly encode linguistic representation
- explicit enough to test
- systematic enough to measure complexity

Logical complexity?

- $\text{QF} < \text{FO} < \text{MSO}$
- Higher the complexity, harder to learn
- A lot of morphophonology is QF *based* on choices in organization
What I did?
- Used Armenian data as a case study
- Designed an interactionist morphophonology based on the data
- Defined the morphophonological system using formal logic
Computing with logic

- What I did?
  - Used Armenian data as a case study
  - Designed an interactionist morphophonology based on the data
  - Defined the morphophonological system using formal logic
- We can then examine how much....
  - can be done with it
  - *choices* in representation and organization affect generative capacity
Example

- /ban/ → [bánd] ‘banned’
- FYI: stay high-level, nitty-gritty notation in appendix + thesis for Armenian

Input: `root`

- **Morphology**
  - `add`

- **Prosody**
  - `parse`

- **Phonology**
  - `apply`

Output: `root++`
• \(/\text{ban}/ \rightarrow [\text{b\text{\text{\‘}}\text{\text{\‘}}}\text{\text{\‘}}\text{\text{\‘}}\text{\‘}}\text{\text{\‘}}] \text{‘banned’}

• FYI: stay high-level, nitty-gritty notation in appendix + thesis for Armenian

\text{Input: root}

\begin{array}{c|c|c|l}
\text{Cycle 1} & \text{MORPHO} & \text{Add } \emptyset & \text{ban-}\emptyset \\
\hline
\text{PROSO} & \text{Syllabify} & .\text{ban.} \\
& \text{Map PWord} & (.\text{ban.})_w \\
\hline
\text{PHONO} & \text{Apply SLevel} & \\
& \text{Apply stress} & \text{bán} \end{array}

\text{Output: root++}
**Example**

- `/ban/ → [b̃and] ‘banned’`
- FYI: stay high-level, nitty-gritty notation in appendix + thesis for Armenian

**Input:** `root`

### Cycle 1
- **Morphology**
  - Morphology
  - **parse**
  - **Prosody**
    - **parse**
    - **Phonology**
      - **produce**

### Input

<table>
<thead>
<tr>
<th>Cycle</th>
<th>MORPHO</th>
<th>PROSO</th>
<th>PHONO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Add $\emptyset$</td>
<td>Syllabify</td>
<td>Apply SLevel</td>
</tr>
<tr>
<td></td>
<td>ban-$\emptyset$</td>
<td>.ban.</td>
<td>(.ban.)$_w$</td>
</tr>
</tbody>
</table>

### Output

<table>
<thead>
<tr>
<th>Cycle</th>
<th>MORPHO</th>
<th>PROSO</th>
<th>PHONO</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Add -d</td>
<td>Syllabify</td>
<td>Nasalize</td>
</tr>
<tr>
<td></td>
<td>bán-d</td>
<td>.bán-d</td>
<td>bánd</td>
</tr>
</tbody>
</table>

**Output:** `root++`
What does the input look like?

- Theory: Words have a lot of hidden morphological structure
- Logic: A word is a graph of nodes + edges

simple root structure

```
<table>
<thead>
<tr>
<th>MRoot</th>
</tr>
</thead>
<tbody>
<tr>
<td>morph</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>ban</td>
</tr>
</tbody>
</table>
```
What does the input look like?

- Theory: Words have a lot of hidden morphological structure
- Logic: A word is a graph of nodes + edges

**simple root structure**

- ‘Morpho+phono’ labels + relations
  
  1. Domain: node indexes
  2. Labels: $\text{MRoot}(x), \text{morph}(x), ..., s(x)$
  3. Relations: $\text{MDom}(x, y), \text{succ}(x)$

```
\begin{align*}
\text{MRoot}_5 & \\
\downarrow & \\
\text{morph}_4 & \\
\downarrow & \\
b_1 & \leftarrow a_2 \leftarrow n_3
\end{align*}
```
How to add more morphology: /ban-/ → ban-Ø

- Theory: Add a zero suffix

\[
\begin{array}{c|c}
\text{Input} & \text{Morphology} \\
\hline
\text{MRoot} & \text{MRoot} \\
\text{morph} & \text{morph} \\
\text{ban} & \text{ban} \quad \emptyset \\
\end{array}
\]
How to add more morphology: \(/ban-/ \rightarrow ban-\emptyset\)

- Theory: Add a zero suffix
- Logic: generate new nodes as output correspondents of the input
- ... Done with graph-to-graph functions with logic (appendix)

\[
\begin{array}{c}
\text{Input} \rightarrow \text{Morphology} \\
\end{array}
\]

```
\begin{aligned}
\text{MRoot}_{0.5} & \quad \downarrow m \\
\text{morph}_{0.4} & \quad \downarrow m \\
\text{b}_{0.1} & \quad \downarrow \quad \text{a}_{0.2} & \quad \downarrow \text{n}_{0.3}
\end{aligned}
```

```
\begin{aligned}
\text{MRoot}_{1.5} & \quad \downarrow m \\
\text{morph}_{1.4} & \quad \downarrow m \\
\text{b}_{1.1} & \quad \downarrow \quad \text{a}_{1.2} & \quad \downarrow \text{n}_{1.3}
\end{aligned}
```
Implicitness in Prosody + Phonology

- Recall: ban → bánd

**Input:** root

**Morphology**

**Prosody**

**Phonology**

**Output:** root++

<table>
<thead>
<tr>
<th>Cycle 1</th>
<th>MORPHO</th>
<th>PROSO</th>
<th>PHONO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Ø</td>
<td>Syllabify</td>
<td>Map PWord</td>
<td></td>
</tr>
<tr>
<td>ban-Ø</td>
<td>.ban.</td>
<td>(.ban.)w</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cycle 2</th>
<th>MORPHO</th>
<th>PROSO</th>
<th>PHONO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add -d</td>
<td>Syllabify</td>
<td>Apply WLevel</td>
<td></td>
</tr>
<tr>
<td>bán-d</td>
<td>.bán-d</td>
<td>bánd</td>
<td></td>
</tr>
</tbody>
</table>

**Output:** bánd
Implicitness in Prosody + Rules

- Implicit step after morphology: Based on the morphology...
  - How do we know what to parse? (MStems? MWords? ...)
  - How do we know what sets of rules to apply? (SLevel? WLevel?...)
  → what triggers the right prosody/rules?

<table>
<thead>
<tr>
<th>Input</th>
<th>/ban/</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycle 1</td>
<td></td>
</tr>
<tr>
<td>Morpho</td>
<td>Add ø</td>
</tr>
<tr>
<td>Proso</td>
<td>Syllabify</td>
</tr>
<tr>
<td>Phono</td>
<td>Map PWord</td>
</tr>
<tr>
<td>Cycle 2</td>
<td></td>
</tr>
<tr>
<td>Morpho</td>
<td>Add -d</td>
</tr>
<tr>
<td>Proso</td>
<td>Syllabify</td>
</tr>
<tr>
<td>Phono</td>
<td>Apply WLevel</td>
</tr>
</tbody>
</table>

Output: bánd
Implicitness in Prosody + Rules

- Settings: Examine morphology to discover the trigger of the parsing/rules
  - Information is encapsulated in the derivation
  - Process (stress...) is separate from (sometimes non-local) trigger
  - Trigger is knowing what’s already parsed + the active cophonology

<table>
<thead>
<tr>
<th>Input</th>
<th>Cycle 1</th>
<th>Cycle 2</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>MORA</td>
<td>MORPHO</td>
<td>MORPHO</td>
<td></td>
</tr>
<tr>
<td>Add Ø</td>
<td>EXAMINE</td>
<td>EXAMINE</td>
<td></td>
</tr>
<tr>
<td>ban-Ø</td>
<td>ban-d</td>
<td>ban-d</td>
<td>bând</td>
</tr>
<tr>
<td>Proso</td>
<td>Syllabify</td>
<td>Syllabify</td>
<td>ban.</td>
</tr>
<tr>
<td>Map PWord</td>
<td></td>
<td></td>
<td>(.ban.)w</td>
</tr>
<tr>
<td>Phono</td>
<td>Apply SLevel</td>
<td>Apply WLevel</td>
<td>bán</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nasalize</td>
<td>bând</td>
</tr>
</tbody>
</table>

Input: /ban/
Encapsulation of Parsing Instructions

- Examine input for relevant info → encapsulate into a SETTINGS constant
  1. Parse:MStem:solo(SETTINGS): topmost node is an MStem which dominates an MRoot but has no PWord
     → we must parse topmost MStem into a new PWord

\[
\text{Morphology} \quad \rightarrow \quad \text{Encapsulation}
\]

<table>
<thead>
<tr>
<th>SETTINGS</th>
<th>Parse:MStem:solo</th>
</tr>
</thead>
</table>

\[
\begin{array}{c}
\text{MStem}_{0.8} \\
\text{MRoot}_{0.5} \\
\text{morph}_{1.4} \\
b_{0.1} \rightarrow a_{0.2} \rightarrow n_{0.3}
\end{array}
\quad
\begin{array}{c}
\text{MStem}_{1.8} \\
\text{MRoot}_{1.5} \\
\text{morph}_{1.4} \\
b_{1.1} \rightarrow a_{1.2} \rightarrow n_{1.3}
\end{array}
\]

\[
\begin{array}{c}
\text{V}_{0.7} \\
morph_{0.6} \\
\text{V}_{1.7} \\
morph_{1.6}
\end{array}
\]
Encapsulation of Rule Instructions

- Examine input for relevant info → encapsulate into a SETTINGS constant

2. **Domain:Cophon:SLevel**(SETTINGS): the input’s topmost morphological node is an MStem [→] we must apply the stem-level rules

<table>
<thead>
<tr>
<th>Morphology</th>
<th>→</th>
<th>Encapsulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETTINGS</td>
<td>Parse:MStem:solo</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Domain:Cophon:SLevel</td>
<td></td>
</tr>
</tbody>
</table>

```
MStem0.8
  \downarrow m
MRoot0.5
  \downarrow m
morph_{1.4}
  \downarrow m
b0.1 \rightarrow a0.2 \rightarrow n0.3

MStem1.8
  \downarrow m
MRoot1.5
  \downarrow m
morph_{1.4}
  \downarrow m
b1.1 \rightarrow a1.2 \rightarrow n1.3
```
Role of encapsulation

- SETTINGS Encapsulation factorizes the parsing/rules from their trigger
- Some of the info on the trigger can be locally deduced, some cannot
- SETTINGS is a descriptive shorthand, but it does affect generative capacity

<table>
<thead>
<tr>
<th>Morphology</th>
<th>→</th>
<th>Encapsulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETTINGS</td>
<td>Parse:MStem:solo</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Domain:Cophon:SLevel</td>
<td></td>
</tr>
</tbody>
</table>

Diagram:

```
<table>
<thead>
<tr>
<th>MStem0.8</th>
<th>MStem1.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRoot0.5</td>
<td>MRoot1.5</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>morph1.4</td>
<td>morph1.4</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>b0.1</td>
<td>b1.1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>a0.2</td>
<td>a1.2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>n0.3</td>
<td>n1.3</td>
</tr>
</tbody>
</table>
```
With the SETTINGS, generate the right prosody
  ▪ Theory: syllabify and parse the input MStem into a new PWord

\[
\begin{align*}
\text{Encapsulation} & \quad \rightarrow \quad \text{Prosody} \\
\text{MStem} & \quad \text{MStem} \quad \text{PWord} \\
\text{MRoot} & \quad \text{MRoot} \quad F \\
morph & \quad \text{morph} \quad \sigma \\
\varnothing & \quad \varnothing \\
\text{ban} & \quad \text{ban}
\end{align*}
\]
Prosody

- With the SETTINGS, generate the right prosody
  - Logic: generate syllables + a PWord from an underlying MStem... because the SETTINGS tells me I have an unparsed MStem
  - \( p(x, y) \) is prosodic dominance

Encapsulation → Prosody

\[
\begin{align*}
\text{SETTINGS} & \quad \text{Parse: MStem: solo} \\
& \quad \text{Domain: Cophon: SLevel}
\end{align*}
\]

\[
\begin{align*}
\text{MStem}_{0.8} & \quad \text{MRoot}_{0.5} \\
& \quad \text{morph}_{1.4} \quad \text{morph}_{0.6} \\
& \quad \text{b}_{0.1} \quad \text{a}_{0.2} \quad \text{n}_{0.3}
\end{align*}
\]

\[
\begin{align*}
\text{PWord}_{4.8} & \quad \text{F}_{3.2} \\
& \quad \text{\sigma}_{2.2} \\
& \quad \text{b}_{1.1} \quad \text{a}_{1.2} \quad \text{n}_{1.3}
\end{align*}
\]
• In output, Morpho + Prosody are ‘matched’ (Selkirk, 2011)

Prosody

Settings

$\text{Parse: MStem: solo}$

$\text{Domain: Cophon: SLevel}$

MStem$_{1.8}$

MRoot$_{1.5}$

$\text{morph}_{1.4}$

b$_{1.1}$

$\leftarrow$ a$_{1.2}$

$\leftarrow$ n$_{1.3}$

morph$_{1.6}$

$v_{1.7}$

PWord$_{4.8}$

$\overrightarrow{F}_{3.2}$

$\leftarrow$ $\overrightarrow{\sigma}_{2.2}$
• Phono rules are triggered by specific morphological constituents
  ▶ Theory: apply stem-level rules (stress)

**Prosody**  \[\rightarrow\]  **Phonology**

<table>
<thead>
<tr>
<th>Prosody</th>
<th>Phonology</th>
</tr>
</thead>
<tbody>
<tr>
<td>MStem</td>
<td>MStem</td>
</tr>
<tr>
<td>MRoot</td>
<td>MRoot</td>
</tr>
<tr>
<td>(v)</td>
<td>(v)</td>
</tr>
<tr>
<td>(F)</td>
<td>(F)</td>
</tr>
<tr>
<td><strong>morph</strong></td>
<td><strong>morph</strong></td>
</tr>
<tr>
<td><strong>ban</strong></td>
<td><strong>bán</strong></td>
</tr>
<tr>
<td><strong>∅</strong></td>
<td><strong>∅</strong></td>
</tr>
<tr>
<td><strong>ban</strong></td>
<td><strong>bán</strong></td>
</tr>
</tbody>
</table>
Rule domains

- Logic: apply stress because the SETTINGS says that the topmost morphological node is an MStem \[\rightarrow\] apply stem-level
- stress placement is local, but finding its trigger isn’t without SETTINGS

Prosody

SETTINGS

\[\text{Parse: MStem: solo} \]
\[\text{Domain: Cophon: SLevel} \]

Match

MStem_{0.8}

\[\text{MRoot}_{0.5} \]
\[\text{morph}_{1.4} \]
\[\text{b}_{0.1} \leftarrow \text{a}_{0.2} \leftarrow \text{n}_{0.3} \]

\[\text{MRoot}_{0.5} \leftarrow \text{v}_{0.7} \]
\[\text{morph}_{0.6} \]
\[\text{PWord}_{0.11} \]
\[\text{F}_{0.10} \]
\[\sigma_{0.9} \]
Rule domains

- Logic: apply stress because the SETTINGS says that the topmost morphological node is an MStem \([\rightarrow]\) apply stem-level
- stress placement is local, but finding its trigger is not without SETTINGS

\[\rightarrow\text{Phonology}\]

\[
\begin{array}{c}
\text{SETTINGS} \\
\text{Parse:MStem:solo} \\
\text{Domain:Cophon:SLevel}
\end{array}
\]

\[
\begin{array}{c}
\text{MStem}_{1.8} \\
\text{MRoot}_{1.5} \\
\text{morph}_{1.4} \\
b_{1.1} \xrightarrow{p} \hat{a}_{1.2} \xrightarrow{p} n_{1.3} \\
\end{array}
\]

\[
\begin{array}{c}
\text{MStem}_{1.8} \\
v_{1.7} \\
morph_{1.6} \\
PWord_{1.11} \\
F_{1.10} \xrightarrow{p} \sigma_{1.9}
\end{array}
\]
Cycle continues to make [bând] with a larger PWord + word-level rules

Input: root

- add
- examine
- parse
- apply
- produce

trigger
add

Output: root++
Cycle 2 [bând]: word-level rules + prosodic wrapping (Truckenbrodt, 1999)
Wrap-up: Coverage

- Logic provides a coherent formalism to compute the morphology-phonology interface because it’s flexible!
- Armenian was my illustrative case-study but it works for any language.
Wrap-up: coverage

- Logic provides a coherent formalism to compute the morphology-phonology interface because it’s flexible!
- Armenian was my illustrative case-study but it works for any language

<table>
<thead>
<tr>
<th>Morphology</th>
<th>Prosody</th>
<th>Rule Domains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affix linearization</td>
<td>Syllabification</td>
<td>Morpheme-triggered</td>
</tr>
<tr>
<td>Zero</td>
<td>Generating syllables</td>
<td>MCons-triggered</td>
</tr>
<tr>
<td>Prefix</td>
<td>Syllable ordering</td>
<td>PCons-triggered</td>
</tr>
<tr>
<td>Suffix</td>
<td>Resyllabification</td>
<td></td>
</tr>
<tr>
<td>Mobile Affix</td>
<td>Tiers over syllables</td>
<td></td>
</tr>
<tr>
<td>Affix Allomorphy</td>
<td>Mapping</td>
<td></td>
</tr>
<tr>
<td>Inwards-sensitive</td>
<td>Generating P-Cons</td>
<td></td>
</tr>
<tr>
<td>Outwards-sensitive</td>
<td>Misaligning P-Cons</td>
<td></td>
</tr>
<tr>
<td>Phono-conditioned</td>
<td>Restructuring P-Cons</td>
<td></td>
</tr>
<tr>
<td>Morpho-conditioned</td>
<td>Recursive Prosody</td>
<td></td>
</tr>
<tr>
<td>Tiers over dominance</td>
<td>Generating</td>
<td></td>
</tr>
<tr>
<td>Compounding</td>
<td>Flattening</td>
<td></td>
</tr>
<tr>
<td>Formation</td>
<td>Compound prosody</td>
<td></td>
</tr>
<tr>
<td>Head-marking</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Expressivity of the Interface**

- Logical formulae were written in MSO
  - Easier to describe and write
- But how much of it can be reduced from MSO to FO or QF? A lot!

<table>
<thead>
<tr>
<th>Morphology</th>
<th>Prosody</th>
<th>Rule Domains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affix linearization</td>
<td>Syllabification</td>
<td>Morpheme-triggered</td>
</tr>
<tr>
<td>Zero</td>
<td>Generating syllables</td>
<td>MCons-triggered</td>
</tr>
<tr>
<td>Prefix</td>
<td>Syllable ordering</td>
<td>PCons-triggered</td>
</tr>
<tr>
<td>Suffix</td>
<td>Resyllabification</td>
<td></td>
</tr>
<tr>
<td>Mobile Affix</td>
<td>Tiers over syllables</td>
<td></td>
</tr>
<tr>
<td>Affix Allomorphy</td>
<td>Mapping</td>
<td></td>
</tr>
<tr>
<td>Inwards-sensitive</td>
<td>Generating P-Cons</td>
<td></td>
</tr>
<tr>
<td>Outwards-sensitive</td>
<td>Misaligning P-Cons</td>
<td></td>
</tr>
<tr>
<td>Phono-conditioned</td>
<td>Restructuring P-Cons</td>
<td></td>
</tr>
<tr>
<td>Morpho-conditioned</td>
<td>Recursive Prosody</td>
<td></td>
</tr>
<tr>
<td>Root allomorphy</td>
<td>Generating</td>
<td></td>
</tr>
<tr>
<td>Tiers over dominance</td>
<td>Flattening</td>
<td></td>
</tr>
<tr>
<td>Compounding</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Formation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Head-marking</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

54
Expressivity of the Interface

- Logical formulae were written in MSO
  - Easier to describe and write
- But how much of it can be reduced from MSO to FO or QF
  - With settings encapsulation + interactionism
  - (is allomorphy trigger within a finite bound)

<table>
<thead>
<tr>
<th>Morphology</th>
<th>Prosody</th>
<th>Rule Domains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affix linearization</td>
<td>Syllabification</td>
<td>Morpheme-triggered</td>
</tr>
<tr>
<td>Zero</td>
<td>Generating syllables</td>
<td>MCons-triggered</td>
</tr>
<tr>
<td>Prefix</td>
<td>Syllable ordering</td>
<td>PCons-triggered</td>
</tr>
<tr>
<td>Suffix</td>
<td>Resyllabification</td>
<td></td>
</tr>
<tr>
<td>Mobile Affix</td>
<td>Tiers over syllables</td>
<td></td>
</tr>
<tr>
<td>Affix Allomorphy</td>
<td>Mapping</td>
<td></td>
</tr>
<tr>
<td>Inwards-sensitive</td>
<td>Generating P-Cons</td>
<td></td>
</tr>
<tr>
<td>Outwards-sensitive</td>
<td>Misaligning P-Cons</td>
<td></td>
</tr>
<tr>
<td>Phono-conditioned</td>
<td>Restructuring P-Cons</td>
<td></td>
</tr>
<tr>
<td>Morpho-conditioned</td>
<td>Recursive Prosody</td>
<td></td>
</tr>
<tr>
<td>Tiers over dominance</td>
<td>Generating</td>
<td></td>
</tr>
<tr>
<td>Compounding</td>
<td>Flattening</td>
<td></td>
</tr>
<tr>
<td>Formation</td>
<td>Compound prosody</td>
<td></td>
</tr>
<tr>
<td>Head-marking</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Expressivity of the Interface**

- Logical formulae were written in MSO
  - Easier to describe and write
- But how much of it can be reduced from **MSO** to **FO** or **QF**
  - No **SETTINGS** encapsulation + interactionism
  - (is the morpho trigger within finite bound or not)

<table>
<thead>
<tr>
<th>Morphology</th>
<th>Prosody</th>
<th>Rule Domains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affix linearization</td>
<td>Syllabification</td>
<td>Morpheme-triggered</td>
</tr>
<tr>
<td>Zero</td>
<td>Generating syllables</td>
<td>MCons-triggered</td>
</tr>
<tr>
<td>Prefix</td>
<td>Syllable ordering</td>
<td>PCons-triggered</td>
</tr>
<tr>
<td>Suffix</td>
<td>Resyllabification</td>
<td></td>
</tr>
<tr>
<td>Mobile Affix</td>
<td>Tiers over syllables</td>
<td></td>
</tr>
<tr>
<td><strong>Affix Allomorphy</strong></td>
<td>Mapping</td>
<td></td>
</tr>
<tr>
<td>Inwards-sensitive</td>
<td>Generating P-Cons</td>
<td></td>
</tr>
<tr>
<td>Outwards-sensitive</td>
<td>Misaligning P-Cons</td>
<td></td>
</tr>
<tr>
<td>Phono-conditioned</td>
<td>Restructuring P-Cons</td>
<td></td>
</tr>
<tr>
<td>Morpho-conditioned</td>
<td>Recursive Prosody</td>
<td></td>
</tr>
<tr>
<td><strong>Tiers over dominance</strong></td>
<td>Generating</td>
<td></td>
</tr>
<tr>
<td>Compounding</td>
<td>Flattening</td>
<td></td>
</tr>
<tr>
<td>Formation</td>
<td>Compound prosody</td>
<td></td>
</tr>
</tbody>
</table>
### Expressivity of the Interface

- Logical formulae were written in MSO
  - Easier to describe and write
- But how much of it can be reduced from **MSO** to **FO** or **QF**
  - No SETTINs encapsulation + no interactionism
  - (may need to find long-distant grandmothers or granddaughters)

<table>
<thead>
<tr>
<th>Morphology</th>
<th>Prosody</th>
<th>Rule Domains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affix linearization</td>
<td>Syllabification</td>
<td>Morpheme-triggered</td>
</tr>
<tr>
<td>Zero</td>
<td>Generating syllables</td>
<td>MCons-triggered</td>
</tr>
<tr>
<td>Prefix</td>
<td>Syllable ordering</td>
<td>PCons-triggered</td>
</tr>
<tr>
<td>Suffix</td>
<td>Resyllabification</td>
<td></td>
</tr>
<tr>
<td>Mobile Affix</td>
<td>Tiers over syllables</td>
<td></td>
</tr>
<tr>
<td>Affix Allomorphy</td>
<td>Mapping</td>
<td></td>
</tr>
<tr>
<td>Inwards-sensitive</td>
<td>Generating P-Cons</td>
<td></td>
</tr>
<tr>
<td>Outwards-sensitive</td>
<td>Misaligning P-Cons</td>
<td></td>
</tr>
<tr>
<td>Phono-conditioned</td>
<td>Restructuring P-Cons</td>
<td></td>
</tr>
<tr>
<td>Morpho-conditioned</td>
<td>Recursive Prosody</td>
<td></td>
</tr>
<tr>
<td>Tiers over dominance</td>
<td>Generating</td>
<td></td>
</tr>
<tr>
<td>Compounding</td>
<td>Flattening</td>
<td></td>
</tr>
<tr>
<td>Formation</td>
<td>Compound prosody</td>
<td></td>
</tr>
<tr>
<td>Head-marking</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table of Contents

- Armenian
  - Preview
  - Distribution
  - Head operations
  - Cyclic or counter-cyclic?
  - Variation in prosody
  - Wrap-up

- Computation
  - What’s the link?
  - Computing morpho-phono
  - Computing with logic
  - Computing morpho-phononology with logic

- End
Final Wrap

- Title: *Computing Cyclic phonology in Armenian*
- Subtitle: *Logical structure of the morphology-phonology interface*
Final wrap

- Title: *Computing Cyclic phonology in Armenian*
- Subtitle: *Logical structure of the morphology-phonology interface*
- *Phonology*: what principles control the alternation of morphemes
Final Wrap

• Title: Computing *Cyclic phonology in Armenian*
• Subtitle: *Logical structure of the morphology-phonology interface*
• *Phonology*: what principles control the alternation of morphemes
• *Cyclic Phonology*: Empirical takeaway + results:
  - Armenian phonology shows multiple hidden structures
  - Morphology + Prosody + Rule domains + Interactionist Cyclicity
  - How they interact reinforces with some theories, falsifies others
Final Wrap

- *Computing Cyclic phonology*: Computational takeaway + results
  - Hidden structure can be encoded with formal logic
  - Computational explicitness uncovers implicit aspects of theory
  - Gives you more tools to your analytical toolbox
Final wrap

- *Computing Cyclic phonology*: Computational takeaway + results
  - Hidden structure can be encoded with formal logic
  - Computational explicitness uncovers implicit aspects of theory
  - Gives you more tools to your analytical toolbox

<table>
<thead>
<tr>
<th>Goals in computational linguistics</th>
<th>Me</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. What structures do we want to represent</td>
<td>✓ – empirical work</td>
</tr>
</tbody>
</table>

I didn’t do it – Jeff+co do though
• Computing Cyclic phonology: Computational takeaway + results
  ▪ Hidden structure can be encoded with formal logic
  ▪ Computational explicitness uncovers implicit aspects of theory
  ▪ Gives you more tools to your analytical toolbox

Goals in computational linguistics

1. What structures do we want to represent
   ✔ – empirical work
2. How do we generate them
   ✔ – transductions
Final wrap

- *Computing Cyclic phonology*: Computational takeaway + results
  - Hidden structure can be encoded with formal logic
  - Computational explicitness uncovers implicit aspects of theory
  - Gives you more tools to your analytical toolbox

<table>
<thead>
<tr>
<th>Goals in computational linguistics</th>
<th>Me</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. What structures do we want to represent</td>
<td>✔ – empirical work</td>
</tr>
<tr>
<td>2. How do we generate them</td>
<td>✔ – transductions</td>
</tr>
<tr>
<td>3. What’s their generative capacity?</td>
<td>✔ ~ kinna</td>
</tr>
<tr>
<td>4. How do we test them?</td>
<td>...</td>
</tr>
<tr>
<td>5. How can we learn grammars</td>
<td>...</td>
</tr>
</tbody>
</table>

3 Generative capacity:
- What’s the maximum/minimal logical power you need?
- What happens without *settings* encapsulation or without interactionism?
  → Mostly QF based on preliminary

4 I didn’t do it – but it’s human/machine readable enough for Prolog

5 I didn’t do it – Jeff+co do though
Thanks to

<table>
<thead>
<tr>
<th>School</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>family</td>
</tr>
<tr>
<td>LAU</td>
<td>friends + faculty</td>
</tr>
<tr>
<td>UD</td>
<td>friends + faculty</td>
</tr>
<tr>
<td>SB</td>
<td>friends + faculty</td>
</tr>
</tbody>
</table>

See y’all SB folk in 2-3 weeks
More info on....

1. types of compounds w.r.t. the paradox [63]
2. role of morphology vs. semantics [69]
3. prosodic constituent in compounding [69]
4. what does logic look like [85]
5. logical hierarchies [72]
6. expressivity of quantifiers in morphophonological structure [74]
Nominal compound: X+N=N

- Hyponymic → Endocentric → Paradoxical Plural
  
  \[
  \text{antsrev} + \text{fur} \quad \text{‘rain + water’} \\
  \text{antsrev-a-fur-er} \quad \text{‘rain water(s)’}
  \]

- Doesn’t matter what’s relationship between STEM1 & STEM2

<table>
<thead>
<tr>
<th>Type</th>
<th>Example</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 of 1</td>
<td>arev + joy</td>
<td>‘sun + ray’</td>
</tr>
<tr>
<td></td>
<td>arev-joy-er</td>
<td>‘sun-ray(s)’</td>
</tr>
<tr>
<td>2 made from 1</td>
<td>medaks + kork</td>
<td>‘silk + carpet’</td>
</tr>
<tr>
<td></td>
<td>medaks-kork-er</td>
<td></td>
</tr>
<tr>
<td>2 in 1</td>
<td>kedin + xorfi</td>
<td>‘ground + pit’</td>
</tr>
<tr>
<td></td>
<td>kedn-xorfi-er</td>
<td>‘ditch(es)’</td>
</tr>
</tbody>
</table>
Nominal compound:

- Nominal compound: X+N=N
  - Hyponymic → Endocentric → Paradoxical Plural
    
    \[\text{antsrev} + \text{t}fur\]  \[\text{rain} + \text{water}\]
    \[\text{antsrev-}a-\text{t}fur\text{-er}\]  \[\text{rain water(s)}\]

- Doesn’t matter what’s POS of STEM1

<table>
<thead>
<tr>
<th>POS</th>
<th>Example</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>don + dzar</td>
<td>‘holiday + tree’</td>
</tr>
<tr>
<td></td>
<td>don-a-dzar-er</td>
<td>‘Christmas tree(s)’</td>
</tr>
<tr>
<td>Adj</td>
<td>nax + hajr</td>
<td>‘first + father’</td>
</tr>
<tr>
<td></td>
<td>nax-a-hajr-er</td>
<td>‘forefather(s)’</td>
</tr>
<tr>
<td>INF</td>
<td>kordz-el + gerb</td>
<td>‘to work + manner’</td>
</tr>
<tr>
<td></td>
<td>kordz-el-a-gerb-er</td>
<td>‘strategy’</td>
</tr>
<tr>
<td>V root</td>
<td>afxad-il + varts</td>
<td>‘to work + reward’</td>
</tr>
<tr>
<td></td>
<td>afxad-a-varts-er</td>
<td>wage(s)</td>
</tr>
</tbody>
</table>
• Possessive compound: X+N=A
  - Non-hyponymic → Exocentric → Transparent Plural
    \[ t\text{far} + s\text{ird} \quad \text{‘evil + heart’} \]
    \[ t\text{far}-a-s\text{ird}-\text{ner} \quad \text{‘evil-hearted (people)’} \]
• Doesn’t matter what’s POS of STEM1
  - Adj
    \[ t\text{etev} + k\text{ajl} \quad \text{‘light + footstep’} \]
    \[ t\text{etev}-a-k\text{ajl}-\text{ner} \quad \text{‘light-footed (people)’} \]
  - Noun
    \[ a\text{rjun} + k\text{ujn} \quad \text{‘blood + color’} \]
    \[ a\text{rjun}-a-k\text{ujn}-\text{ner} \quad \text{‘blood-colored (people)’} \]
  - V root
    \[ x\text{eyt}-e\text{l} + t\text{sajn} \quad \text{‘to strangle + voice’} \]
    \[ x\text{eyt}-a-t\text{sajn}-\text{ner} \quad \text{‘strangled-voiced (people)’} \]
Deverbal compound

- Verbs and deverbal compounds
  
  Verb | Deverbal compound?  
  --|---
  √  + TH + T/Agr | X + V_{root}  
  *per*  -e   -l | *antsrev-a*  *per*  
  ‘to bring’ | ‘rain + to bring’ = ‘rain-bearing’

- Deverbal compounds ~ English synthetic compounds
  - Argument structure
  - BUT no overt suffix
  - Bare verbal root, similar to Romance compounds

- What about the Plural Paradox?
Deverbal compound

- Deverbal compound: $X + V_{\text{root}} = N/A$
  - Non-hyponymic $\rightarrow$ Exocentric $\rightarrow$ Transparent
- Doesn’t matter what’s POS of STEM1
  
<table>
<thead>
<tr>
<th>POS</th>
<th>Example</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>manug + varz-el</td>
<td>‘child + ti instruct’</td>
</tr>
<tr>
<td></td>
<td>mang-a-varz-ner</td>
<td>‘school-teacher(s)’</td>
</tr>
<tr>
<td>Adj</td>
<td>lav + des-nel</td>
<td>‘good + to see’</td>
</tr>
<tr>
<td></td>
<td>lav-a-des-ner</td>
<td>‘optimist(s)’</td>
</tr>
<tr>
<td>V root</td>
<td>hajhoj-el + sir-el</td>
<td>‘to swear + to love’</td>
</tr>
<tr>
<td></td>
<td>hajhoj-a-ser-ner</td>
<td>‘lover(s) of swearing’</td>
</tr>
</tbody>
</table>
Deverbal compound

- Deverbal compound: $X + V_{\text{root}} = N/A$ ~50% lexicon
  - Non-hyponymic $\rightarrow$ Exocentric $\rightarrow$ Transparent
- Type of argument structure doesn’t matter

<table>
<thead>
<tr>
<th>Argument Structure</th>
<th>Example</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obj + Active</td>
<td>hoy + kordz-el</td>
<td>‘earth + to work’</td>
</tr>
<tr>
<td></td>
<td>hox-a-kordz-ner</td>
<td>‘farmer(s)’</td>
</tr>
<tr>
<td>Adv + Intrans</td>
<td>jergar + dev-el</td>
<td>‘long + to last’</td>
</tr>
<tr>
<td></td>
<td>jergar-a-dev-ner</td>
<td>‘long-lasting (things)’</td>
</tr>
<tr>
<td>Subj + Passive</td>
<td>vodn + gox-vil</td>
<td>‘foot + to be trodden’</td>
</tr>
<tr>
<td></td>
<td>vodn-a-gox-ner</td>
<td>‘foot-trodden (things)’</td>
</tr>
<tr>
<td>Instr + Passive</td>
<td>jergat + kam-vil</td>
<td>‘iron + to be nailed’</td>
</tr>
<tr>
<td></td>
<td>jergat-a-kam-ner</td>
<td>‘iron-nailed (things)’</td>
</tr>
<tr>
<td>Adv + Passive</td>
<td>təzvar + mars-vil</td>
<td>‘hard + to be digested’</td>
</tr>
<tr>
<td></td>
<td>təzvar-a-mars-ner</td>
<td>‘indigestible (things)’</td>
</tr>
</tbody>
</table>
Is it morphological?

- My story: plural counting depends on semantic heads
- Alternative:
  - Not semantics, just morphology
  - it’s a morphological quirk of nominal compounds
  - PL counts $\sigma$ of $\text{STEM2}$ iff nominal compound

<table>
<thead>
<tr>
<th>$\text{STEM2}$</th>
<th>Nominal</th>
<th>Possessive</th>
<th>Deverbal</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{\text{t}}\text{fur}$</td>
<td>$\text{sird}$</td>
<td>$\text{kordz-el}$</td>
<td>‘water’</td>
</tr>
<tr>
<td>$\text{Comp PL}$</td>
<td>$\text{antsrev-a-}$</td>
<td>$\text{t}\text{far-a-sird-ner}$</td>
<td>$\text{hats-a-kordz-ner}$</td>
</tr>
</tbody>
</table>

- Looks attractive, but won’t work
Is it morphological?

- Alternative: PL counts $\sigma$ iff nominal compound
- Doesn’t work because...

1. Not every nominal compound gets a paradoxical plural
   - Semantic bleaching: if meaning is opaque, get transparent plural
     
     \[
     \begin{array}{cccc}
     \text{STEM1} & \text{STEM2} & \text{Comp} & \text{PL} \\
     \widehat{dz\emptyset x-e-l} & xod & \widehat{dz\emptyset x-a-xod} & \widehat{dz\emptyset x-a-xod-ner} \\
     \end{array}
     \]
     
     ‘to smoke’ ‘grass’ ‘tobacco’
   - Find variation when compound has transparent vs. opaque metaphorical meanings

2. Adjectival compounds are hyponymic and paradoxically pluralized
   - Rare, but exist and getting slowly more popular
     
     \[
     \begin{array}{cccc}
     \text{STEM1} & \text{STEM2} & \text{Comp} & \text{PL} \\
     derev & xid & derev-a-xid & derev-a-xid-er \\
     \end{array}
     \]
     
     ‘hole’ ‘dense’ ‘dense with leaves’

3. More shifts in loanwords, lexicalization, etc.
Is it morphological?

- Alternative: PL counts $\sigma$ iff nominal compound
- Misses the fact that semantics and plural go hand-in-hand
  - Some pairs of $\text{STEM1} + \text{STEM2}$ could get either an endo vs. exo meaning
  - Plural matches the meaning

<table>
<thead>
<tr>
<th>STEM1</th>
<th>STEM2</th>
<th>COMP</th>
<th>TransPL</th>
<th>ParaPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>garmir</td>
<td>tev</td>
<td>garmr-a-tev</td>
<td>garmr-a-tev-ner</td>
<td>garmr-a-tev-er</td>
</tr>
<tr>
<td>‘red’</td>
<td>‘wing’</td>
<td>‘red-winged’</td>
<td>‘red-winged things’</td>
<td>‘red wings’</td>
</tr>
</tbody>
</table>
**Generative capacity**: what can a formalism and what it can’t

<table>
<thead>
<tr>
<th>Finite-state transducers</th>
<th>Logical transductions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rational</td>
<td>Monadic-Second Order</td>
</tr>
<tr>
<td>WD</td>
<td>First-Order FO</td>
</tr>
<tr>
<td>L-Seq</td>
<td>Quantifier-Free QF</td>
</tr>
<tr>
<td>R-Seq</td>
<td></td>
</tr>
<tr>
<td>L-OSL</td>
<td></td>
</tr>
<tr>
<td>ISL</td>
<td></td>
</tr>
<tr>
<td>R-OSL</td>
<td></td>
</tr>
</tbody>
</table>

• Informally...
  • FST: what info do the states keep track of
  • Logic: how do you use quantifiers $\exists$
**Logical hierarchy**

- Logic: how do you use quantifiers $\exists(x)$
- Formally...
  - MSO: Quantify over variables + sets $\exists x, \exists X$
  - FO: Quantify over variables $\exists x$
  - QF: No quantifiers $\exists x$
- Informally...
  - MSO: More long-distance: does $x$ exist and precede me?
  - FO: mild long-distance: does $x$ exist?
  - QF: No long-distance info
Distance of information

- Ways to find information
  - **Contextless**: /t/ becomes a [s] → \( t \rightarrow s/ _\)
  - **Adjacent**: /t/ becomes a [s] before an /i/: \( t \rightarrow s/ _i \)
  - **Close**: /t/ becomes a [s] before a segment+/i/: \( t \rightarrow s/ _\Sigma i \)
  - **Distant**: /t/ becomes a [s] if there is an /i/: \( t \rightarrow s/ _\ldots i \text{ or } i.. \)

- Ways to compute such information
  - **Contextless** \( \phi s(x_1) \overset{\text{def}}{=} t(x) \)
  - **Adjacent** \( \phi s(x_1) \overset{\text{def}}{=} t(x) \land \exists y[\text{succ}(x, y) \land i(y)] \)
  - **Close** \( \phi s(x_1) \overset{\text{def}}{=} t(x) \land \exists y, z[\text{succ}(x, y) \land \text{succ}(y, z) \land i(z)] \)
  - **Distant** \( \phi s(x_1) \overset{\text{def}}{=} t(x) \land \exists y[i(y)] \)

- Contextless is local because QF
- But... how can you ‘tell’ that Adjacent+Close should be local too?
  → how do you find \( y, z \) w.r.t \( x \)
Functions vs. Relations

- Convert relations $R(x, y)$ into functions $F_l: R(x) = y$ and $F_r: R(y) = x$:

- Example:

  \[
  t_{0.1} \lessdot i_{0.2} \lessdot t_{0.3} \lessdot a_{0.4}
  \]

- Relations:
  - $\text{succ}(x, y)$ is true for $\{(1,2), (2,3), (3,4)\}$

- Functions
  - $F_L: \text{succ}(x)$ is defined as
    - $F_L: \text{succ}(1) = 2$
    - $F_L: \text{succ}(2) = 3$
    - $F_L: \text{succ}(3) = 4$
  - $F_R: \text{succ}(x)$ is defined as
    - $F_R: \text{succ}(2) = 1$
    - $F_R: \text{succ}(3) = 2$
    - $F_R: \text{succ}(4) = 3$
Relations into Functions

- Original relations with quantifiers
  Contextless: $\phi_s(x_1) \equiv t(x)$
  Adjacent: $\phi_s(x_1) \equiv t(x) \land \exists y \left[ \text{succ}(x, y) \land i(y) \right]$ 
  Close: $\phi_s(x_1) \equiv t(x) \land \exists y, z \left[ \text{succ}(x, y) \land \text{succ}(y, z) \land i(z) \right]$ 
  Distant: $\phi_s(x_1) \equiv t(x) \land \exists y \left[ i(y) \right]$ 

- Now with functions unless when quantifier+relations are unavoidable
  Contextless: $\phi_s(x_1) \equiv t(x)$
  Adjacent: $\phi_s(x_1) \equiv t(x) \land \exists y \left[ \text{succ}(x, y) \land i(y) \right]$ 
  $\phi_s(x_1) \equiv t(x) \land i(\text{FL:succ}(x))$
  Close: $\phi_s(x_1) \equiv t(x) \land \exists y, z \left[ \text{succ}(x, y) \land \text{succ}(y, z) \land i(z) \right]$ 
  $\phi_s(x_1) \equiv t(x) \land i(\text{FL:succ}(\text{FL:succ}(x))))$
  Distant: $\phi_s(x_1) \equiv t(x) \land \exists y \left[ i(y) \right]$
If a relation isn’t 1-to-1 in one direction, then can’t make a function
e.g., morpheme-to-segment morphological dominance

Relation:
- $\text{MDom}(x, y)$ is true for $\{(5,1), (5,2), (5,3), (5,4)\}$

Breaking $\text{MDom}(x, y)$ into two relations or functions
- $F_L:\text{MDom}(x)$ is defined as a one-to-many relation
  - $F_L:\text{prec}(5) = \{1, 2, 3, 4\}$
- $F_R:\text{MDom}(x)$ is defined as a 1-to-1 function
  - $F_R:\text{prec}(1) = 5$
  - $F_R:\text{prec}(2) = 5$
  - $F_R:\text{prec}(3) = 5$
  - $F_R:\text{prec}(4) = 5$
(Un)avoidable long-distance in morphology

- Input

  \[\text{MRoot} \quad \text{MRoot}_5 \]

  \[
  \xymatrix{
  \varnothing_1 \ar[r] & p_2 \ar[r] & \varnothing_3 \ar[r] & l_4
  }
  \\
  \xymatrix{
  \varnothing \ar[r] & \varnothing_1 \ar[r] & p_2 \ar[r] & \varnothing_3 \ar[r] & l_4
  }
  \\
  \]

- Deterministic: segment-to-root dependence is local
  - Give a segment \([p]\) the label \(+F\) if it is part of a root
  - \(\phi^{+F}(x_1) \overset{\text{def}}{=} p(x) \land \exists y [\text{MDom}(y, x) \land \text{MRoot}(y)]\)
  - \(\phi^{+F}(x_1) \overset{\text{def}}{=} p(x) \land \text{MRoot}(F_L: \text{MDom}(x))\)

- Nondeterministic: root-to-segment dependence is nonlocal
  - Give an MRoot the label \(+F\) if it dominates a \([p]\) segment
  - \(\phi^{+F}(x_1) \overset{\text{def}}{=} \text{MRoot}(x) \land \exists y [\text{MDom}(x, y) \land p(y)]\)
Locality in morphology-phonology

- What relation can/can’t be made into a function?
  → What type of dependence is local or not

1. Segment successor
2. Morphological dominance
   - Finding mothers
     ★ What’s the morpheme of a segment?
   - Finding finitely-bounded daughters...
   - Binary branching for morpho constituents to binary daughters
   - Ternary branching for compounds
   - Not n-ary branching like from morphemes to segments

3. Prosodic dominance
   - Finding mothers
     ★ Is this syllable part of a foot or PWord?
   - Finding finitely-bounded daughters...
   - finite syllable size: PDom:syll_nuc(x, y)...
   - Not iterative footing PDom:PWord_foot(x, y)
$p = \text{Foot?}$

- $p = \text{Foot?}$

\[ \begin{array}{c}
\text{w} \\
\text{F} \quad \text{F} \\
\text{xatʃ} \quad \text{kar} \\
\text{-er} \\
\end{array} \quad \begin{array}{c}
\text{w} \\
\text{F} \\
\text{xatʃ - kar} \\
\text{-ner} \\
\end{array} \]

- Nope!... Armenian has initial secondary stress

→ Above words *always* have two feet + $p$ is higher

\[ \begin{array}{c}
\text{w} \\
\text{p} \\
\text{F} \quad \text{F} \\
\text{xatʃ} \quad \text{kár} \\
\text{-er} \\
\end{array} \quad \begin{array}{c}
\text{w} \\
\text{p} \\
\text{F} \quad \text{F} \\
\text{xatʃ} \quad \text{kár} \\
\text{-ner} \\
\end{array} \]
$p = \text{PWord?}$

- $p = \text{a recursive PWord?}$

- Nope because stem-level processes apply across these ‘word’ boundaries’
$p = \text{PWord?}$

- Stem-level processes apply in Der + Comp but not inflection
  \[
  \begin{align*}
  \hat{t\dot{s}úr} & \quad \text{‘water’} & \quad \text{‘water + hole’} \\
  \text{Der} \quad \hat{t\dot{s}ər-ajín} & \quad \text{‘aquatic’} & \quad \hat{t\dot{s}ər-a-pós} & \quad \text{‘water-hole’} \\
  \text{Inf} \quad \hat{t\dot{s}ur-óv} & \quad \text{‘with water’} & \quad \hat{t\dot{s}ər-a-pos-ér}
  \end{align*}
  \]

- Weird if $p$ is a word-boundary because stem-level processes apply \textit{across} it
What is $p$

$p \neq F$

$p \neq W$

$F < p < W$
Prosodic Stems

- $p = \text{Prosodic Stem } s$?
- Tradition: PWord is smallest morphologically-derived constituents
- But: some agglutinative languages show need for a smaller one

\[
\text{Intonational Phrase (ι)} \quad \text{Intonational Phrase (ι)}
\]

\[
\text{Phonological Phrase (φ)} \quad \text{Phonological Phrase (φ)}
\]

\[
\text{Prosodic Word (ω or PWord)} \quad \text{Prosodic Word (ω or PWord)}
\]

\[
\text{Foot (F or Σ)} \quad \text{Foot (F or Σ)}
\]
What does it look like?

- Example: English plural
- Input-output:
  1. string of segments
  2. with features

Input: s   a   p   z
Output: s   a   p   s
What does it look like?

- Example: English plural
- Input-output:
  1. string of segments
  2. with features
  3. ordered

\[
\text{Input} \quad \begin{array}{c}
 s & \rightarrow & a & \rightarrow & p & \rightarrow & z \\
< & & < & & < & & < \\
\end{array}
\]
\[
\text{Output} \quad \begin{array}{c}
 s & \rightarrow & a & \rightarrow & p & \rightarrow & s \\
< & & < & & < & & < \\
\end{array}
\]
What does it look like?

- Example: English plural
- Input-output:
  1. string of segments
  2. with features
  3. ordered
  4. stand in correspondence

Input

\[ S_{0.1} \rightarrow a_{0.2} \rightarrow p_{0.3} \rightarrow z_{0.4} \]

Output

\[ S_{1.1} \rightarrow a_{1.2} \rightarrow p_{1.3} \rightarrow S_{1.4} \]
How does it work?

- What’s in the input?

  Input

  \[ \text{Input} \]

  \[ \text{s}_{0.1} \rightarrow \text{a}_{0.2} \rightarrow \text{p}_{0.3} \rightarrow \text{z}_{0.4} \]

- Specifically...

  1. Domain: \( \{ 1, 2, 3, 4 \} \)
  2. Labels \( L \): \( a(x), p(x), s(x), z(x), b(x) \)....
      - \( s(x) = \text{true} \) for \( x \in \{1\} \)
      - \( a(x) = \text{true} \) for \( x \in \{2\} \)
      - \( p(x) = \text{true} \) for \( x \in \{3\} \)
      - \( \ldots \)
  3. Relations \( R \): \( \text{succ:seg}(x, y) \)
      - \( \text{succ:seg}(x, y) = \text{true} \) for \( x \in \{(1, 2), (2, 3), (3, 4)\} \)
As for the output...

Input

\[\begin{align*}
S_{0.1} &\rightarrow a_{0.2} &\rightarrow p_{0.3} &\rightarrow z_{0.4}
\end{align*}\]

Output

\[\begin{align*}
S_{1.1} &\leftarrow a_{1.2} &\leftarrow p_{1.3} &\leftarrow S_{1.4}
\end{align*}\]

Define correspondences using predicates and output functions
Define correspondences using predicates and output functions

1. Predicates = short-hand

\[\begin{align*}
\text{Predicates} & = s(x) \vee p(x) \vee \ldots \\
\text{Post-voiceless} & = \exists y \left[ \text{succ}(y, x) \land \text{voiceless}(y) \right]
\end{align*}\]

Output functions = create output correspondents for every \( \ell \in \mathcal{L} - \{s, z\} \):

\[\begin{align*}
\phi_{\ell}(x_1) & = \ell(x) \\
\phi_{\text{succ}}(x_1, y_1) & = \text{succ}:\text{seg}(x, y) \\
\phi_z(x_1) & = z(x) \land \neg \text{post-voiceless}(x) \\
\phi_s(x_1) & = s(x) \lor [z(x) \land \text{post-voiceless}(x)]
\end{align*}\]

No extrinsic ordering – everything must be simultaneously true

\[\begin{align*}
\text{Input} & \quad s_{0.1} \quad \rightarrow \quad a_{0.2} \quad \rightarrow \quad p_{0.3} \quad \rightarrow \quad z_{0.4}
\end{align*}\]
• Define correspondences using predicates and output functions
  1. Predicates = short-hand
     ★ $\text{voiceless}(x) \overset{\text{def}}{=} s(x) \lor p(x) \lor \ldots$

Input

\[
\begin{array}{cccc}
S_{0.1} & \rightarrow & a_{0.2} & \rightarrow \rightarrow \\
& & & \rightarrow \rightarrow \\
& & & p_{0.3} \\
& & & \rightarrow \rightarrow \\
& & & z_{0.4}
\end{array}
\]
Define correspondences using predicates and output functions

1. Predicates = short-hand
   - \( \text{voiceless}(x) \overset{\text{def}}{=} s(x) \lor p(x) \lor \ldots \)
   - \( \text{post\_voiceless}(x) \overset{\text{def}}{=} \exists y[\text{succ}(y, x) \land \text{voiceless}(y)] \)

Input

\[
\begin{array}{cccc}
S_{0.1} & \xrightarrow{<} & a_{0.2} & \xrightarrow{<} & p_{0.3} & \xrightarrow{<} & z_{0.4}
\end{array}
\]
• Define correspondences using predicates and output functions
  1. Predicates = short-hand
     ★ \( \text{voiceless}(x) = s(x) \lor p(x) \lor \ldots \)  
     ★ \( \text{post\_voiceless}(x) = \exists y[\text{succ}(y, x) \land \text{voiceless}(y)] \)
  2. Output functions = create output correspondents

Input
\[ S_{0.1} \rightarrow a_{0.2} \rightarrow p_{0.3} \rightarrow z_{0.4} \]

Output
Make it do something

- Define correspondences using predicates and output functions
  1. Predicates = short-hand
     - \( \text{voiceless}(x) \overset{\text{def}}{=} s(x) \lor p(x) \lor \ldots \)
     - \( \text{post\_voiceless}(x) \overset{\text{def}}{=} \exists y [\text{succ}(y, x) \land \text{voiceless}(y)] \)
  2. Output functions = create output correspondents
     - for every \( \text{lab} \in L - \{s, z\} \):
       \( \phi_{\text{lab}}(x_1) \overset{\text{def}}{=} \text{lab}(x) \)

---

Input

- S0.1 → a0.2 → p0.3 → Z0.4

Output

- a1.2 → p1.3
Define correspondences using predicates and output functions

1. Predicates = short-hand
   - \( \text{voiceless}(x) \defeq s(x) \lor p(x) \lor \ldots \)
   - \( \text{post\_voiceless}(x) \defeq \exists y[\text{succ}(y,x) \land \text{voiceless}(y)] \)

2. Output functions = create output correspondents
   - for every \( \text{lab} \in L - \{s, z\} \):
     - \( \phi_{\text{lab}}(x_1) \defeq \text{lab}(x) \)
     - \( \phi_{z}(x_1) \defeq z(x) \land \neg \text{post\_voiceless} \)

Input

\[
\begin{array}{c}
\text{S0.1} \quad \text{a0.2} \quad \text{p0.3} \quad \text{Z0.4}
\end{array}
\]

Output

\[
\begin{array}{c}
\text{a1.2} \quad \text{p1.3}
\end{array}
\]
Define correspondences using predicates and output functions

1. Predicates = short-hand
   - $\text{voiceless}(x) \eqdef s(x) \lor p(x) \lor \ldots$
   - $\text{post\_voiceless}(x) \eqdef \exists y[\text{succ}(y, x) \land \text{voiceless}(y)]$

2. Output functions = create output correspondents
   - for every $\text{lab} \in L - \{s, z\}$:
     - $\phi_{\text{lab}}(x) \eqdef \text{lab}(x)$
     - $\phi_z(x) \eqdef z(x) \land \neg \text{post\_voiceless}$
     - $\phi_s(x) \eqdef s(x) \lor [z(x) \land \text{post\_voiceless}(x)]$

Input

\[ \text{S0.1} \, \text{S1.1} \, \text{a0.2} \, \text{a1.2} \, \text{p0.3} \, \text{p1.3} \, \text{z0.4} \, \text{s1.4} \]

Output
• Define correspondences using predicates and output functions
  1. Predicates = short-hand
     ★ \( \text{voiceless}(x) \overset{\text{def}}{=} s(x) \lor p(x) \lor \ldots \)
     ★ \( \text{post\_voiceless}(x) \overset{\text{def}}{=} \exists y[\text{succ}(y, x) \land \text{voiceless}(y)] \)
  2. Output functions = create output correspondents
     ★ for every \( \text{lab} \in L - \{s, z\} \):
       \( \phi_{\text{lab}}(x_1) \overset{\text{def}}{=} \text{lab}(x) \)
     ★ \( \phi_z(x_1) \overset{\text{def}}{=} z(x) \land \neg \text{post\_voiceless} \)
     ★ \( \phi_s(x_1) \overset{\text{def}}{=} s(x) \lor [z(x) \land \text{post\_voiceless}(x)] \)

Input

\[
\begin{array}{cccc}
S_{0.1} & \rightarrow & a_{0.2} & \rightarrow & p_{0.3} & \rightarrow & Z_{0.4}
\end{array}
\]

Output

\[
\begin{array}{cccc}
S_{1.1} & \leftarrow & a_{1.2} & \leftarrow & P_{1.3} & \leftarrow & S_{1.4}
\end{array}
\]
Make it do something

- Define correspondences using predicates and output functions
  1. Predicates = short-hand
     - \( \text{voiceless}(x) \defeq s(x) \lor p(x) \lor \ldots \)
     - \( \text{post\_voiceless}(x) \defeq \exists y[\text{succ}(y, x) \land \text{voiceless}(y)] \)
  2. Output functions = create output correspondents
     - for every \( \text{lab} \in L - \{s, z\} \):
       \( \phi_{\text{lab}}(x_1) \defeq \text{lab}(x) \)
     - \( \phi_{z}(x_1) \defeq z(x) \land \neg \text{post\_voiceless} \)
     - \( \phi_{s}(x_1) \defeq s(x) \lor [z(x) \land \text{post\_voiceless}(x)] \)
     - \( \phi_{\text{succ\_seg}}(x_1, y_1) \defeq \text{succ\_seg}(x, y) \)
     - * No extrinsic ordering – everything must be simultaneously true

Input

\[
\begin{array}{c}
S_{0.1} \xrightarrow{<} a_{0.2} \xrightarrow{<} p_{0.3} \xrightarrow{<} Z_{0.4}
\end{array}
\]

Output

\[
\begin{array}{c}
S_{1.1} \xrightarrow{<} a_{1.2} \xrightarrow{<} p_{1.3} \xrightarrow{<} S_{1.4}
\end{array}
\]


[Selkirk 2011] Selkirk, Elisabeth: The syntax-phonology interface. In: