Investigating phonological abstraction through feature induction

*Features in Phonology, Morphology, Syntax: What are they? Universitetet i Tromsø, October 31 2013*

Aleksei Nazarov, University of Massachusetts at Amherst

anazarov@linguist.umass.edu
Overview

- Introduction
  - should grammars always refer to features?
  - approach from perspective of machine learning
Overview

- Introduction
  - should grammars always refer to features?
  - approach from perspective of machine learning
- Computational simulation: how does a learner abstract over domains of application?
  - model, data, method
  - results: grammars with features in some constraints only
Overview

- Introduction
  - should grammars always refer to features?
  - approach from perspective of machine learning
- Computational simulation: how does a learner abstract over domains of application?
  - model, data, method
  - results: grammars with features in some constraints only
- Discussion: implications of grammars referring to features as well as other units
Introduction: background

- Features help generalize over domains of application of rules or constraints

- Phonology: features generalize over segment/phoneme categories

  E.g., /-z/ → [-s] / [p,t,k,f,θ,s,ʃ,tʃ]_ ⇒
  
  /-z/ → [-s] / [-voice]_
Introduction: background

- Question:
  Is it always advantageous (both for the analyst and the speaker) to state every rule or constraint in the grammar in terms of features?

- In other words: is it unreasonable for grammar to refer to sound event through levels of abstraction other than features?

(Not counting prosodic units, suprasegmentals)
Introduction: background

- Phonology: canonical answer is “yes”
Introduction: background

- Phonology: canonical answer is “yes”
- Chomsky & Halle (1968):
  - adapting categorical versions of phonetic features is most economical hypothesis of representation
Introduction: background

- Phonology: canonical answer is “yes”
- Chomsky & Halle (1968):
  - adapting categorical versions of phonetic features is most economical hypothesis of representation
  - establishes preference for phonetically natural rules
    (see Chomsky & Halle 1968, Postal 1968, Kenstowicz & Kisseberth 1979 for more)
Introduction: background

- Phonology: canonical answer is “yes”
- Chomsky & Halle (1968):
  - adapting categorical versions of phonetic features is most economical hypothesis of representation
  - establishes preference for phonetically natural rules
    (see Chomsky & Halle 1968, Postal 1968, Kenstowicz & Kisseberth 1979 for more)

- Models with richer representations lead to longer grammars, therefore are disfavored
Introduction: empirical issue

- Phonological patterns may apply to groups of segments, or to single segments.
Introduction: empirical issue

- Phonological patterns may apply to groups of segments, or to single segments.

- English (Jensen 1993, Mielke 2007):
  - sibilants [s,z,ʃ,ʒ,ʧ,ʤ] may not precede [s,z] word-finally: *[bʌs-s, bʌz-z, pætʃ-s, peɪdʒ-z]

<table>
<thead>
<tr>
<th>p</th>
<th>t</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>d</td>
<td>g</td>
</tr>
<tr>
<td>f</td>
<td>θ</td>
<td>s</td>
</tr>
<tr>
<td>v</td>
<td>ð</td>
<td>z</td>
</tr>
<tr>
<td>m</td>
<td>n</td>
<td>ŋ</td>
</tr>
<tr>
<td>w</td>
<td>l</td>
<td>j</td>
</tr>
</tbody>
</table>

Red: disallowed before [s,z] word-finally
Introduction: empirical issue

- Phonological patterns may apply to groups of segments, or to single segments.

- English (Jensen 1993, Mielke 2007):
  - only [s] may start a three-consonant word-initial cluster: [strit], *[ftrit, ntrit, ŋtrit]
  - Red: disallowed before [s,z] word-finally
  - Purple: allowed as C1 in word-initial CCC

| p | t | k | Red: disallowed before [s,z] word-finally |
| b | d | g | Purple: allowed as C1 in word-initial CCC |
| f | θs | ʃ | |
| v | ðz | ž | |
| m | n | ŋ | |
| w | l | j | |
Introduction: empirical issue

- Phonological patterns may apply to groups of segments, or to single segments.

  - P-base cross-linguistic database of phonological classes (Mielke 2007):
    - 13 patterns encoded as applying to one segment
    - 11 additional cases (apply to all segments but one) found by manual search of languages starting with A alone
Introduction: empirical issue

- One-segment classes may be represented as intersections of a number of features
Introduction: empirical issue

- One-segment classes may be represented as intersections of a number of features
  - e.g., [s] is equivalent to [+ant,-voice,+strid]
Introduction: empirical issue

- One-segment classes may be represented as intersections of a number of features
  - e.g., [s] is equivalent to [+ant,-voice,+strid]

\[ \begin{array}{llll}
  p & t & k & \text{Red: [+anterior]} \\
  b & d & g & \\
  f & \emptyset & s & \text{\[\text{\[]}} \\
  v & \emptyset & z & 3d3 \\
  m & n & \text{\[\eta]} & \\
  w & l & j & \\
\end{array} \]
Introduction: empirical issue

- One-segment classes may be represented as intersections of a number of features
  - e.g., [s] is equivalent to [+ant,-voice,+strid]

Red: [+anterior]
Blue: [-voice]
Introduction: empirical issue

- One-segment classes may be represented as intersections of a number of features
  - e.g., [s] is equivalent to [+ant,-voice,+strid]
Introduction: always features?

- Featural representation of one-segment class will always be longer and more complex.

- Is it desirable (for analyst/speaker) to represent one-segment classes in this way?
Introduction: always features?

- Featural representation of one-segment class will always be longer and more complex

- Is it desirable (for analyst/speaker) to represent one-segment classes in this way?
  - If features are *a priori* specified as building blocks of grammars: yes
  - Is this still the case when this *a priori* assumption is taken away?
I will approach this question in terms of machine learning

Given a choice between representing a pattern in terms of segments and in terms of features:

- How will data containing both one-segment and multi-segment patterns be learned?

- Learning algorithm not explicitly instructed to aim for a certain level of abstraction
Introduction: machine learning

- Possible outcomes:
  
  1. The grammars have constraints referring only to segments

  2. The grammars have constraints referring only to features

  3. The grammars have constraints referring to both features and segments
Introduction: assumptions

- Essential assumptions for this simulation:

1. Atomic segment units are available to the language user:
   - active in on-line processing of speech (Jesse et al. 2007, Nielsen 2011)
   - active in phonological processes, e.g., consonant OCP (Coetzee & Pater 2008 and references therein)
Introduction: assumptions

- Essential assumptions for this simulation:

  2. Phonological features are learned:
     - assuming universal features, the same feature is realized differently across languages (Cho & Ladefoged 1999)
     - therefore, phonetic information cannot be sufficient for mapping perception/articulation to features
Introduction: assumptions

- Essential assumptions for this simulation:

  2. Phonological features are learned:
     - contextual information must be used
     - grammar contains contextual information
     - use contextual information from grammar (rather than contextual information outside of grammar)

      (see Mielke (2004) on learning features from phonological patterns)
Introduction: assumptions

- Consequences of these assumptions:

  1. Segment-to-feature mapping must be learned simultaneously with grammar

  2. Constraints/rules referring to features gradually become available during grammar learning process
Introduction: assumptions

- Non-essential working assumptions:
  - Features are induced only from contextual information: no phonetic content
    (Substance-free phonology: Morén 2006, 2007 (and many others))
  - All phonological constraints are induced instead of innate
    (see Hayes & Wilson 2008 on constraint induction)
Introduction: summary

- Question: Is it always advantageous (both for the analyst and the speaker) to state every constraint in the grammar in terms of features?

- Crucial empirical phenomenon: one-segment patterns
- Learning one-segment and multi-segment patterns: all-feature grammars as outcome?

- Preview: segment/feature grammars obtained
Simulation: overview

- Machine learning simulation based on paradigm established by Hayes & Wilson (2008):
  - phonotactic constraint-based grammar is built up from positive data
  - violable constraints selected and weighted to optimally predict the attested data
Simulation: overview

- Departure from Hayes & Wilson’s learner:
  - features are not built into the model, but induced at intermediate stages of grammar learning

- Questions:
  - will features be learned at all?
  - will all constraints in grammars learned by this procedure always use features?
Simulation: model

- Maximum Entropy model

  - probability distribution over possible representations based on weighted violable constraints (à la OT/Harmonic Grammar)

  - constraints weighted to make this distribution maximally similar to what is observed

(see Appendix for more)
Simulation: model

- Regularization:
  
  - Optimization of constraint weights constrained by L2 prior (Hastie et al. 2009):
    
    - keeps sum of constraint weights as small as possible

    - encourages more general constraints: one general constraint with larger weight yields smaller sum of weights than several specific constraints with smaller weights
Simulation: model

- Information gain:
  - Value which estimates how much a constraint will improve the current grammar (bring it closer to predicting the observed data)
  
  - Information gain of a constraint correlates with how accurately it captures a (sub)pattern in the data

(see Appendix for more)
Simulation: model

- Constraints:
  - phonotactic constraints against two- and three-element sequences of word-boundaries, segments or features
  - examples: *#m, *km, *u[labial]u
Simulation: model

- Constraints:
  - selected probabilistically based on information gain:
    - start with random seed constraint
      (subject to information gain threshold)
      e.g. *#pi
    - seed constraint repeatedly manipulated until this does not lead to increase in information gain
      e.g. *#pi → *#mi → *#m
Simulation: model

- Features found by clustering information gain of closely related constraints
  
  - Intuition: a feature denotes a class of segments that participates in the same phonological pattern
Simulation: model

- Features found by clustering information gain of closely related constraints
  - Implementation:
    a feature denotes a class of segments which yields high-valued constraints when inserted in the same context

<table>
<thead>
<tr>
<th>i</th>
<th>a</th>
<th>u</th>
<th>p</th>
<th>t</th>
<th>k</th>
<th>b</th>
<th>d</th>
<th>g</th>
<th>m</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>*#_</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.002</td>
<td>0.002</td>
<td>0.002</td>
<td>0.002</td>
<td>0.002</td>
<td>0.015</td>
<td>0.015</td>
</tr>
</tbody>
</table>
Simulation: model

- Features found by clustering information gain of closely related constraints

  - Cluster analysis (Mixture of Gaussians, Everitt 2011) divides same-context constraints into high and low information gain value clusters (whenever appropriate)
Simulation: model

- Features found by clustering information gain of closely related constraints
  - Focus segments extracted from cluster of high information-value constraints
  - Feature label assigned to these segments
    (phonetics not taken into account - labels are arbitrary)

<table>
<thead>
<tr>
<th></th>
<th>i</th>
<th>a</th>
<th>u</th>
<th>p</th>
<th>t</th>
<th>k</th>
<th>b</th>
<th>d</th>
<th>g</th>
<th>m</th>
<th>n</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>*#_</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.002</td>
<td>0.002</td>
<td>0.002</td>
<td>0.002</td>
<td>0.002</td>
<td>0.002</td>
<td>0.015</td>
<td>0.015</td>
<td>0.015</td>
</tr>
</tbody>
</table>
Simulation: data

- Nature of data to consider:
  - both one-segment and multi-segment patterns must be present
  - single segment in one-segment pattern must be representable as intersection of segment classes appealed to in multi-segment patterns

\[
\begin{array}{ccc}
p & t & k \\
b & d & g \\
m & n & \eta \\
\end{array}
\] ✔

\[
\begin{array}{ccc}
p & t & k \\
b & d & g \\
m & n & \eta \\
\end{array}
\] ✘
Simulation: data

- **Example: English** (Jensen 1993, Mielke 2007)

\[
\begin{array}{ccc}
\text{p} & \text{t} & \text{k} \\
\text{b} & \text{d} & \text{g} \\
\text{f} & \text{θ} & \text{ʃ} \\
\text{v} & \text{ð} & \text{ʒ} \\
\text{m} & \text{n} & \text{ŋ} \\
\text{w} & \text{l} & \text{j}
\end{array}
\]

Red: disallowed before [s,z] word-finally
Simulation: data

- Example: English (Jensen 1993, Mielke 2007)

\[\begin{array}{cccc}
p & t & k & \text{Red: disallowed before [s,z] word-finally} \\
b & d & g & \\
f & \theta & s & \text{Blue: allowed as C3 in word-final CCC} \\
v & \partial & z & \partial \\
m & n & \eta & \\
w & \eta & l & j \end{array}\]
Simulation: data

- Example: English \( (\text{Jensen 1993, Mielke 2007}) \)

- Red: disallowed before \([s, z]\) word-finally
- Blue: allowed as C3 in word-final CCC
- Purple: allowed as C1 in word-initial CCC
Simulation: data

- **Example: English** (Jensen 1993, Mielke 2007)

  - Red: disallowed before [s,z] word-finally

  - Blue: allowed as C3 in word-final CCC

  - Purple: allowed as C1 in word-initial C

- **Other examples like this found in, e.g., Yoruba** (Pulleyblank 1988)
Simulation: data

- The actual data used for the simulations was a toy language which shared the crucial properties of these examples:

\[
\begin{array}{ccc}
p & t & k \\
b & d & g \\
m & n & ə
\end{array}
\]

Red: no nasals word-initially
Simulation: data

- The actual data used for the simulations was a toy language which shared the crucial properties of these examples:

  \[
  \begin{array}{c}
  \text{Red: no nasals word-initially} \\
  \text{Blue: no labials between high vowels [i,u]} \\
  \end{array}
  \]

  \[
  \begin{array}{cccc}
  p & t & k & \\
  b & d & g & \\
  m & n & \text{ŋ} & \\
  \end{array}
  \]
Simulation: data

- The actual data used for the simulations was a toy language which shared the crucial properties of these examples:

<table>
<thead>
<tr>
<th>p</th>
<th>t</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>d</td>
<td>g</td>
</tr>
<tr>
<td>m</td>
<td>n</td>
<td>ɲ</td>
</tr>
</tbody>
</table>

Red: no nasals word-initially
Blue: no labials between high vowels [i,u]
Purple: no [m] word-finally
Simulation: data

- The actual data used for the simulations was a toy language which shared the crucial properties of these examples:

  \[
  \begin{array}{ccc}
  \text{Red: no nasals word-initially} & \text{Blue: no labials between high vowels [i,u]} & \text{Purple: no [m] word-finally} \\
  \text{b} & \text{d} & \text{g} \\
  \text{m} & \text{n} & \text{n}
  \end{array}
  \]

- All possible CVCVC forms obeying these restrictions present in input to the learner
Simulation: procedure

- Initial state: no constraints, features unavailable
- All potential representations (given in segments) equally probable
Simulation: procedure

- Initial state: no constraints, features unavailable
- All potential representations (given in segments) equally probable

- All CVCVC sequences over toy language inventory are potential representations
- Observed forms have no initial nasals, no labials between high Vs, no final [m]

possible: ... padam padan ... nitun ditun dibun
observed: ... padan ... ditun
Simulation: method

- Step 1: Find a group of constraints which forms a local peak in gain value

  e.g., \{\#m,\#n,\#\eta\}

  These have higher information gain than, e.g., \#p, am, n:

  \#p, am, n ban (more) observed forms in the data and bring the empty grammar less close to predicting the observed data
Simulation: method

- Step 2: Find all possible contexts that can be made from these constraints.

The constraints \{*#m,*#n,*#ntag\} can be factored into the following contexts

* #_
* _m
* _n
* _etag
Simulation: method

- Step 3: for every context, find if there is a cluster of segments which yields a high information gain value when inserted in that context; assign feature labels to those clusters

\[
\begin{array}{cccccccccccc}
  & i & a & u & p & t & k & b & d & g & m & n & n \\
  * \#_ & 0.001 & 0.001 & 0.001 & 0.002 & 0.002 & 0.002 & 0.002 & 0.002 & 0.002 & 0.015 & 0.015 & 0.015 \\
\end{array}
\]

\[ [m, n, \eta] \Rightarrow [\text{nasal}] \]
Simulation: method

- Step 4: add the selected constraints to the grammar, and optimize their weights

Grammar:

*#m: 0 → *#m: 6
*#n: 0 → *#n: 6
*#ŋ: 0 → *#ŋ: 6
Simulation: method

- Steps 1-4 repeated until final goal is reached (observed data have at least 95% total likelihood)

- Features induced at step 3 available for use in constraints at next occurrence of step 1

  - Once *#m, *#n, *#ŋ are in the grammar, and the feature label [nasal] = [m, n, ŋ] is induced,

  - the constraint *#[nasal] becomes available
Simulation: method

- E.g., *[nasal] has high information gain value (not in current grammar, tightly fits data pattern)

- If selected and weighted, *[nasal] takes away all the weight of *[m], *[n], *[ŋ]
- zero weight equivalent to absence from grammar
Simulation: method

- Reset to 0 because of regularization prior:
  - higher weight on one constraint is better than lower weights on three constraints combined

- This effect occurs when the candidates punished by a new constraint are a strict superset of those punished by individual existing constraints:
  - *[#nasal] versus *#m, *#n, *#ŋ
  - *[hi][labial][hi] versus *ibi, *ibu, *umi ...
Simulation: method

- Reset to 0 does not happen when feature-based constraint and segment-based constraint are homonymous:
  \[ *[\text{labial, nasal}]# = *m# \]
Simulation: method

- Reset to 0 does not happen when feature-based constraint and segment-based constraint are homonymous:
  - *[labial,nasal]# = *m#

- Homonymous feature-based constraint has lower information gain (*repeats existing constraint*)
  - *[lab,nas]# less likely to be selected

- Even when it is selected, no reset to 0
  - *m# retains some weight next to *[lab,nas]#
Simulation: results

- 31 out of 32 runs yielded grammars referring to both segments and features

- Most frequent grammar:
  \[*\#[nasal], *\[high][labial][high], *m# *

- One all-feature grammar:
  \[*\#[nasal], *\[hi][labial][hi], *[labial,nasal]# *

- All other grammars were variations of the most frequently observed grammar (see Appendix)
Simulation: results

- The learner strongly prefers a segmental representation for the one-segment pattern, and a featural representation for the multi-segment patterns.

- By extrapolation, languages with at least one one-segment pattern are expected not to represent that one-segment pattern (entirely) in terms of features.
Discussion

- Machine learning simulation shows:
  - when *a priori* assumption of all-feature grammars is lifted:
  - despite bias in favor of generalization,
  - one-segment patterns not represented in terms of features

- This is because features are more efficient *only* for multi-segment patterns
Discussion

- These results show that:
  - features can be learned in a bottom-up fashion from phonological patterns
  - grammars that represent one-segment patterns without features emerge despite bias towards generalization (from regularization)
Discussion

- These results show that:
  - features can be learned in a bottom-up fashion from phonological patterns (see also Archangeli et al. 2012)
  - grammars that represent one-segment patterns without features emerge despite bias towards generalization (from regularization)

(Procedure relies only on structural factors: these methods may also be applied to other domains of language, e.g., syntax)
Discussion: implications

- Implication for (phonological) analysis:
  - when a (phonological) pattern is analyzed, it is not trivial that it is stated in terms of features
  - rather, question of appropriate level of abstraction asked for every pattern
Discussion: implications

- Implication for (phonological) analysis:
  - when a (phonological) pattern is analyzed, it is not trivial that it is stated in terms of features
  - rather, question of appropriate level of abstraction asked for every pattern

- Why would level of abstraction matter?
Discussion: implications

- There are psycholinguistic techniques to probe into levels of abstraction:
  - Bach testing (Halle 1978)
  - Priming (Jesse et al. 2007)
  - Talker adaptation (McQueen et al. 2006, Nielsen 2011)

- Ergo: level of abstraction in hypothesized rules/constraints matters empirically

- Important direction for future research
Discussion: implications

- Another consequence of grammars with both featural and lower-order descriptions:
  - same sound event may be described at different levels of abstraction
    e.g., [m] or [labial, nasal]
  - this means: multiple autonomous levels of representation for sounds
Discussion: implications

- This property is reminiscent of models such as
  - Turbidity (Goldrick 2001)
  - Abstract Declarative Phonology (Bye 2006)
  - Colored Containment (Van Oostendorp 2004, 2008)
  - Bidirectional Phonology (Boersma 2007)

- Grammars with multiple levels of abstraction need little extension to have the extra power of such models (Nazarov 2012, 2013)

- Another direction for further investigation
Conclusion

- Are features always better for representing phonological patterns?
- Investigation through machine learning of features:
  - no: one-segment patterns favor representation by segment units

- Grammars which refer both to features and lower-order units (segments) are worthy of consideration by speakers and analysts
Thank you!
Acknowledgements

- Many thanks to:
  - Kristine Yu
  - Brian Dillon
  - Tom Roeper
  - Joe Pater
  - John Kingston
  - John McCarthy
  - participants of the UMass Sound Seminar and the UMass Phonology Reading Group
References


References


References


**Nazarov**, A. 2013. *Phonological opacity as differential classification of sound events*. Talk given at the University of Amsterdam on 1/10/2013.


Appendix: Maximum Entropy model

- Observed distribution \( p \)
  \[ p(x) = \frac{\text{count}(x)}{\sum_{y \in \Omega} \text{count}(y)} \]

- Predicted distribution \( q \): based on harmony scores \( H \) for every candidate
  \[ H(x) = \sum (w_i \times C_i(x)) \]
  \[ q(x) = \frac{e^{H(x)}}{\sum_{y \in \Omega} e^{H(y)}} \]

\( \Omega \) stands for the set of possible representations
Appendix: Maximum Entropy model

- Objective of the model: manipulate weights to minimize K-L divergence of observed distribution from predicted distribution

\[
D_{KL}(t \| w) = \sum [ t(x) \ast \ln(t(x) / w(x))] \\
\text{Obj} = \min \left[ \sum_{w} D_{KL}(p \| q) + \sum_{w \in W} \left[ (w - \mu)^2 / 2\sigma \right] \right] \\
\text{regularization term;} \quad \mu = 0 \text{ and } \sigma = 10,000
Appendix: Information gain

- Let C* be a proposed new constraint, and w* its weight
- Let q’ be the distribution predicted by the current grammar augmented with C* with weight w*

Information gain: maximum descent in K-L divergence of observed from predicted when C* is added to the grammar

\[ G(w^*, C^*) = \max_{w^*} [ D_{KL}(p || q) - D_{KL}(p || q') ] \]

(L2 regularization with \( \mu = 0 \) and \( \sigma = 10,000 \) added to this maximization also)
Appendix: Results

- Word-initial pattern:
  - 26 grammars: represented by *#[nasal]
  - 3 grammars: *#[nasal], *#[nasal]V
  - 3 grammars:

(42) the three runs at which the word-initial restriction was represented by non-overlapping constraints

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Traditional notation</th>
<th>Weight</th>
<th>Constraint</th>
<th>Traditional notation</th>
<th>Weight</th>
<th>Constraint</th>
<th>Traditional notation</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>*#m</td>
<td>*#m</td>
<td>2.68</td>
<td>*#{ŋ}</td>
<td>*#[nasal, -labial]</td>
<td>2.78</td>
<td>*#{ŋ}</td>
<td>*#[nasal, -labial]</td>
<td>3.37</td>
</tr>
<tr>
<td>*#{ŋ}</td>
<td>*#[nasal, -labial]</td>
<td>1.12</td>
<td>*#{ŋ}</td>
<td>*#[nasal, -coronal]</td>
<td>2.78</td>
<td>*#m</td>
<td>*#m</td>
<td>2.68</td>
</tr>
<tr>
<td>*#{ŋ} {aiu}</td>
<td>*#[nasal, -labial]V</td>
<td>1.12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*#{ŋ}</td>
<td>*#[nasal, -labial]</td>
<td>1.12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix: Results

- Word-medial pattern:
  - Combination of one or more of the following constraints:

(43) a survey of all 18 constraints attested in the final grammars which represented (part of) the word-medial pattern

- E.g.: *\{iu\} \{pb\}\{iu\}, *\{iu\} \textit{m}\{iu\}
Appendix: Results

- Word-final pattern:
  - 28 grammars: only *m#
  - 1 grammar: only *[nasal,labial]#
  - 3 grammars:

(44) the three runs (not counting run 23) at which the word-final restriction was not solely represented with the constraint *m#

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Traditional notation</th>
<th>Weight</th>
<th>Constraint</th>
<th>Traditional notation</th>
<th>Weight</th>
<th>Constraint</th>
<th>Traditional notation</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>*m#</td>
<td>*m#</td>
<td>2.29</td>
<td>*m#</td>
<td>*m#</td>
<td>2.27</td>
<td>*m#</td>
<td>*m#</td>
<td>2.15</td>
</tr>
<tr>
<td>*{aiu}m#</td>
<td>*Vm#</td>
<td>0.05</td>
<td>*{aiu}m#</td>
<td>*Vm#</td>
<td>0.16</td>
<td>*{aiu}m#</td>
<td>*Vm#</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Aleksei Nazarov, University of Massachusetts at Amherst