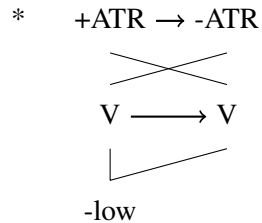# Vowel harmony with QFLFP

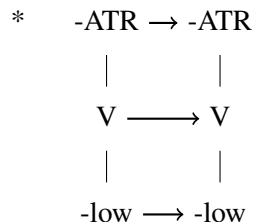*Eileen Blum[1]*

1 Introduction

2 Multi-tiered ARs

This paper investigates the utility of Quantifier-free Least Fixed Point logic (QFLFP) for describing vowel harmony transformations over multi-tiered autosegmental representations (ARs). The multi-tiered ARs in this paper utilize a version of Hayes (1990)'s bottlebrush theory of vowel feature representations such that each feature is represented on a separate tier. Assuming binary vowel features means that each feature tier includes both the + and - values of a particular feature. Associations connect the segmental and feature tiers such that each vowel is directly associated to an element on multiple feature tiers; thus "the CV tier... resembles the spine of a bottlebrush with features branching off"(Hayes, 1990). The ARs used throughout this paper abstract away from consonants and thus the "spine of the bottlebrush"(Hayes, 1990) only includes vowels because vowel harmony is assumed to affect only vowel features.

Much work with ARs includes assumptions about (im)possible autosegmental structures: the Obligatory Contour Principle (OCP)(Leben, 1973) and the No Crossing Constraint (NCC) (Goldsmith, 1976; Sagey, 1986). The investigation into logical descriptions of vowel harmony transformations carried out here will provide insights into the necessity of these two constraints on both input (underlying) and output (surface) structures. Examples of structures that violate each of these constraints are shown in (1)-(2) below.

(1)   Violates NCC

$$\begin{array}{cc} * & \text{+ATR} \rightarrow \text{-ATR} \\ & \diagdown\!\!\!\!\!\diagup \\ & \text{V} \longrightarrow \text{V} \\ & \text{-low} \end{array}$$

(2)   Violates OCP

$$\begin{array}{cc} * & \text{-ATR} \rightarrow \text{-ATR} \\ & | \qquad | \\ & \text{V} \longrightarrow \text{V} \\ & | \qquad | \\ & \text{-low} \longrightarrow \text{-low} \end{array}$$

The NCC states that association lines between the segmental tier and a feature tier never cross. Odden (1994) adds that the NCC can only evaluate the association between the segmental and one featural tier at a time. The representation in (**??**) violates the NCC because +ATR precedes -ATR,

but is associated to a vowel that is preceded by a vowel associated to -ATR; this configuration creates visually crossed association lines.

the OCP stipulates that successive featural elements must be distinct. The representation in (**??**) violates the OCP because on both the ATR and low feature tiers there are two identical successive features, -ATR and -low respectively. The OCP in conjunction with FS results in representations where multiple vowels are associated to a single feature rather than having multiple successive iterations of the same feature each associated to a single vowel. An example representation of an Akan word that satisfies all of the AR properties discussed here is shown in (**??**).

Both the NCC and the OCP have also been derived via a concatenation operation (∘) that merges autosegmental "graph primitives"(Jardine & Heinz, 2015, p. 1). An autosegmental graph primitive consists of an element on the segmental tier, the elements on each feature tier and the associations between the featural and segmental tiers. The concatenation operation combines a finite set of adjacent graph primitives to generate a fully specified AR. For example, the AR in (**??**) is derived from the set of graph primitives in (3). Each primitive in (3) is concatenated with a single adjacent primitive. If two adjacent primitives share an identical feature those two features are merged into one feature with two associations, as in (**??**). The merging of identical adjacent features essentially prevents surface ARs from having multiple iterations of a feature and crossed associations, thus satisfying both the OCP and the NCC. However, if two segmental elements are associated to the exact same feature and a different element intervenes then both iterations of that feature will occur in the surface AR because only adjacent primitive elements are concatenated and can thus be merged. This dissertation will show that an intervening element can be a vowel associated to the same feature with a different value or a domain boundary. It will further show that a domain boundary primitive may include that boundary on both segmental and feature tiers.

(3) Concatenation of adjacent autosegmental graph primitives

<div align="center">

-ATR   -ATR
| |
t ∘ i ∘ e
| |
-low   -low

</div>
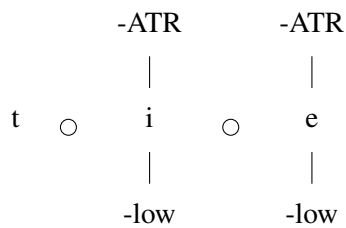
This paper will also provide insights into the necessity of the assumption that surface ARs obey Full Specification (FS)(Clements, 1976), which is held in my current ongoing work with multi-tiered ARs of vowel harmony. An example of a structure that violates FS is in (**??**).

(4) Violates FS

<div align="center">

*   -ATR
|
V ⟶ V
└──┘
-low

</div>
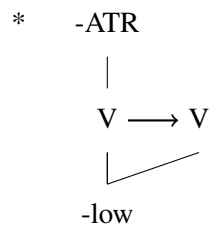
FS means that each featural element must be associated to at least one vowel on the segmental tier and each vowel on the segmental tier must be associated to at least one element on each featural tier. FS crucially allows vowels to be associated to multiple featural tiers as is necessary for each vowel feature to occupy its own tier. The hypothetical representation in (4) straighforwardly violates FS because there is a vowel that is not associated to any feature on the ATR tier. While both vowels are associated to a single -low feature, the second vowel is not associated to any feature on the ATR tier. Since vowel harmony patterns will be analyzed, it will be assumed that consonants cannot be associated to vowel features and that FS and vowel harmony in general ignore consonantal elements on the segmental tier.

A notable effect of FS along with the NCC is that they prevent what have been called gapped structures (Archangeli & Pulleyblank, 1994; Ringen & Vago, 1998). A gapped structure is one in which a feature appears to have skipped over a vowel that it could potentially be associated to. FS would prevent gapped structures in which the "skipped" vowel is not associated to anything on that particular feature's tier. The NCC would prevent gapped structures in which the surrounding two vowels are associated to a single feature and the intervening "skipped" vowel is associated to a different feature on the same tier.

3   QF Logical Transductions

Phonological transformations can be described using logical transductions, which characterize functions. The expressivity of a logic determines the class of functions that it describes. For example, previous work has shown that Input Strictly Local (ISL) functions are describable with quantifier-free (QF) logical transductions (Chandlee, 2014; Chandlee & Heinz, 2018). QF logic is precisely first-order (FO) logic without quanitfiers ($\exists$ or $\forall$). In this logic, a term picks out a single element in a model's domain, so a variable is considered a term and a function applied to a term is also a term.

(5)   Terms:

     a. a variable $x, y, z, ...$

     b. a function f applied to a term $t : f(t)$

QF transductions define an output string in terms of the elements and relations of the input string using atomic formulas that take terms.

(6)   QF atomic formulas

     a. a unary relation P(t)

     b. a binary relation R(t$_1$, t$_2$) and

     c. $t_1 \approx t_2$

For an input alphabet Σ and an output alphabet Γ a logical transduction consists of a copyset C, a set of unary output condition predicates, and set of unary predicates to label the output elements. The copyset contains a copy of each input element, which is given a label via the unary predicates $P_\gamma^C(x)$. The unary output condition predicates define the conditions under which the copy of an input element is present in the output. Lastly, QF transductions are order-preserving as a result of the definition below.

(7)   Let $<$ (and $\leq$) be the transitive (and reflexive) closure of p; build p' such that its transitive closure is $<'$, defined as follows: for all $c, e \in C$

$$d_1^c <' d_2^e \stackrel{\text{def}}{=} \begin{array}{ll} x < y & if c \geq e \\ x \leq y & if c < e \end{array}$$

3.1 Order over strings

If we change our binary ordering relation between elements in a string to a function that applies to an element then we can represent the successor relation ($\lhd$) using a predecessor function (p(x)). Thus the ordering between elements is characterized by recursively picking out elements and their predecessors. For example, for the string in (8) the predecessor of 5 is 4 and so p(5) = 4, p(p(5)) = 3, etc. In order to make predecessor a total function, the first element is defined as that which is its own predecessor.

(8)   Ordering in a string with the predecessor function
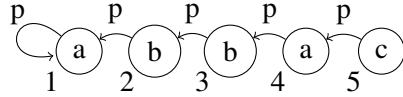
(9)   first(x) $\stackrel{\text{def}}{=}$ p(x) $\approx x$

The string in (8) is thus defined over a signature that consists of the predecessor function and the unary relation that labels elements with letters of the alphabet: {p, $P_{\sigma \in \Sigma}$}, Σ = {a, b, c}. The predecessor function can also be used to define the order between elements on a tier in autosegmental representations.

3.2 Association Relation

In addition to ordering between elements on a tier, Chandlee and Jardine (2019) introduced the binary association relation $\mathscr{A}(x,y)$ between elements on different tiers. For ARs of tone, x is evaluated as an element on a tonal tier and y as an element on the TBU tier. In order to use this relation for multi-tiered ARs of vowel harmony, a separate association relation must be defined between the vowel tier and each feature tier.

(10) $\mathscr{A}_F(x,y)$

(11) $\mathscr{A}_G(x,y)$

4   Unbounded Spreading

4.1   Blocking

5   Association Function

5.1   Blocking

5.2   Transparency

5.2.1   The No Crossing Constraint.

6 References

Archangeli, D., & Pulleyblank, D. (1994). *Grounded phonology* (Vol. 25). MIT Press.

Chandlee, J. (2014). *Strictly local phonological processes* (PhD thesis). University of Delaware.

Chandlee, J., & Heinz, J. (2018). Strict locality and phonological maps. *Linguistic Inquiry*, *49*(1), 23–60.

Chandlee, J., & Jardine, A. (2019). Autosegmental input strictly local functions. In *Transactions of the association for computational linguistics* (Vol. 7, pp. 157–168). MIT Press.

Clements, G. (1976). Vowel harmony in non-linear generative phonology: An autosegmental model. Bloomington, Indiana University Linguistics Club.

Goldsmith, J. (1976). *Autosegmental phonology* (PhD thesis). Massachusetts Institute of Technology.

Hayes, B. (1990). Diphthongization and coindexing. *Phonology*, *7*(1), 31–71.

Jardine, A., & Heinz, J. (2015). A concatenation operation to derive autosegmental graphs. In *Proceedings of the 14th annual meeting on the mathematics of language (mol 2015)* (pp. 139–151). Chicago, USA: Association for Computational Linguistics.

Leben, W. (1973). *Suprasegmental phonology* (PhD thesis). Massachusetts Institute of Technology.

Odden, D. (1994). Adjacency parameters in phonology. *Language*, *70*(2), 289–330.

Ringen, C., & Vago, R. (1998). Hungarian vowel harmony in optimality. *Phonology*, *15*, 393–416.

Sagey, E. (1986). *The representation of features and relations in non-linear phonology* (PhD thesis). Massachusetts Institute of Technology.