

Vowel harmony with QFLFP

Eileen Blum¹

1 Introduction

Phonological transformations can be characterized as mathematical functions, which map an underlying representation to an attested surface form. Such transformations can be described using logical transductions and the expressivity of a logic determines the class of functions that it describes. For example, previous work has shown that Input Strictly Local (ISL) string functions are describable with quantifier-free (QF) first order (FO) logical transductions (Chandlee, 2014; Chandlee & Heinz, 2018). In addition, Chandlee and Jardine (2019a) showed that some tone processes are QF definable over two-tiered autosegmental representations (ARs), which make up the Autosegmental Input Strictly Local (AISL) function class. However, Chandlee and Jardine (2019a) claimed that unbounded spreading is not QF definable and thus not AISL. This paper will demonstrate a method for describing vowel harmony transformations with both unbounded spreading and blocking patterns over multi-tiered ARs using quantifier-free least fixed point (QFLFP) logical transductions. This paper will also illustrate that a blocking pattern is not describable with QFLFP unless the autosegmental association relation is translated into a function.

First, section 2 defines QF string transductions and the two relations needed to define autosegmental representations (ARs): ordering and association. QF transductions are defined using atomic formula that take terms as a function that maps an input structure to an output copy of the same type of structure. The structures analyzed in this paper are multi-tiered autosegmental representations, which utilize ordering functions between elements on a tier and association between elements on different tiers.

Section 3 outlines the representational assumptions held throughout this paper. This paper analyzes two vowel harmony patterns over multi-tiered ARs, which use a bottlebrush feature representation and binary vowel features. These ARs abstract away from consonants so they consist of a vowel tier with elements directly associated to features on separate tiers. Each feature tier consists of the + and - values of one feature. Autosegmental theory provides restrictions to allow only output ARs that obey Full Specification (FS), the No Crossing Constraint (NCC), and the Obligatory Contour Principle (OCP). Input ARs, on the other hand, are not subject to FS.

Section 4 introduces the vowel harmony patterns under discussion: unbounded spreading and blocking. In this section an unbounded spreading pattern from Akan is defined using QFLFP with a predecessor function and binary association relation. Spreading is characterized as an iterative process of associating multiple successive vowels to a single feature via the recursive *lfp* predicate. The Akan-like blocking pattern, however, is not QFLFP definable with a binary association relation.

Section 5 translates the binary association relation into a function. The association function allows association to be defined using terms, which adds the possibility for writing a QF predicate that defines “unspecified” vowels as elements on the vowel tier, which are associated to themselves. The recursive QFLFP definition of association to a particular feature can then refer to these input reflexive associations and overwrites them with association to the specified feature. In this way, the Akan-like blocking pattern is characterized as iterative spreading of a feature only to vowels that are “unspecified”, or associated to themselves, in the input.

Lastly, section 6 concludes and proposes a future extension of this work to investigate the QFLFP definition of a vowel harmony pattern with transparent vowels.

2 QF Logical Transductions

QF logic is precisely first-order (FO) logic without quantifiers (\exists or \forall). In QFFO logic, a term picks out a single element in a model's domain; (1) lists the mathematical notation for what is considered a term.

- (1) Terms (t):
 - a. a variable x, y, z, \dots
 - b. a function f applied to a term: $f(t)$

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

- (2) QF atomic formulas
 - a. a unary relation: $P(t)$
 - b. a binary relation: $R(t_1, t_2)$
 - c. a strict equivalence: $t_1 \approx t_2$

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 defined in (3).

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

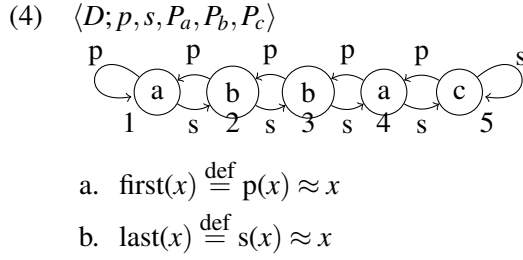
$$d_1^c <' d_2^e \stackrel{\text{def}}{=} \begin{array}{ll} x < y & \text{if } c \geq e \\ x \leq y & \text{if } c < e \end{array}$$

This order preservation allows the first copyset to be a copy of both the input elements and the ordering between them so that input ARs are mapped to an output AR.

The following sections will introduce the functions and relations needed to describe a vowel harmony transformation over multi-tiered autosegmental representations: order between elements on a tier and association between elements on different tiers.

2.1 Ordering with QF

In order to ensure that QF formulas use terms, which pick out an element in the model, the binary ordering relation used in previous work with ARs is changed to a function. Rather than represent ordering between elements in a string with the binary successor relation ($<$) QF formulas utilize a predecessor function ($p(x)$) or a successor function ($s(x)$). For example, for the string in (4) the predecessor of 5 is 4 and so $p(5) = 4$, $p(p(5)) = 3$, etc. In addition, the successor of 1 is 2, so $s(1) = 2$, $s(s(1)) = 3$, etc. Predecessor and successor must also be total functions, which means they are defined for every element in the domain so the predicate in (4a) defines the first element as that which is its own predecessor and the predicate in (4b) defines the last element as its own successor.



The string in (4) is defined over a signature that consists of the predecessor and successor functions and the unary relation that labels elements with letters of the alphabet: $\{p, s, P_{\sigma \in \Sigma}\}$, $\Sigma = \{a, b, c\}$.

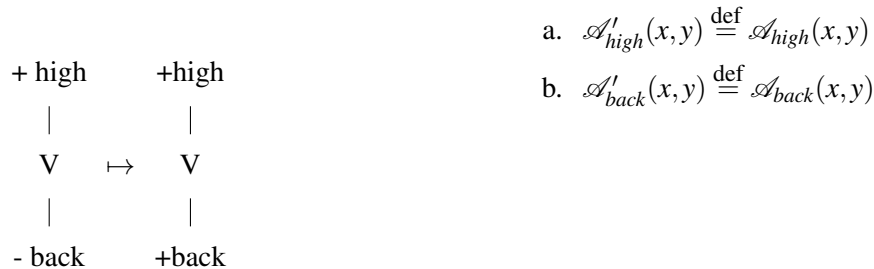
ARs consist of more than one string of disjoint sets of elements and these strings are connected by association. So each tier in a multi-tiered AR is essentially a string of elements with labels of a certain type. For example, in the ARs used here the vowel tier consists of elements labeled with $P_V(x)$ to represent vowels, while each feature tier consists of elements labeled with the + or - value of a particular feature. The predecessor and successor functions are used to define the order between elements on each tier. The following subsection discusses the association relation between tiers.

2.2 Association relation

In addition to ordering between elements on a tier, Chandlee and Jardine (2019a) introduced an antisymmetric binary association relation $\mathcal{A}(x, y)$ to describe autosegmental associations between elements on different tiers. Association was introduced as a binary relation due to the possible configurations of tone-TBU associations. In tone languages, it is possible for a single tone to be associated to multiple TBUs, but in the case of contour tones it is also possible for a single TBU to be associated to more than one tone. The possibility for multiple association in either direction rules out the possibility of representing tone-TBU associations as a function, which must pick out a single element when applied to another element. In principle the binary association relation could be defined symmetrically, but I follow Chandlee and Jardine (2019a) in defining it antisymmetrically. Thus for ARs of tone, x is always evaluated as an element on a tonal tier and y as an element on a tier of tone-bearing units (TBUs).

This same association relation can be used for multi-tiered ARs of vowel harmony patterns. In order to do so, a separate association relation must be defined between the vowel tier and each feature tier. For example, to describe the transformation in (5) the association between a vowel and a feature on the high tier must be defined separately from the association between a vowel and a feature on the back tier. Both associations are defined for the output in terms of the input.

(5) Separate association relations for each feature tier



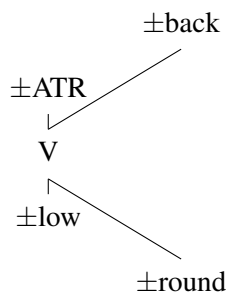
Since both output featural associations are also present in the input, the QF definitions of both the input and output associations are identical. The output association relation is denoted with '.

This section established the mathematical definitions that will be used to describe vowel harmony transformations as QF logical transductions. It introduced QF logic as well as the ordering functions and binary association relation needed to define multi-tiered ARs using QF logic. However, on its own QF is not sufficient to describe the vowel harmony transformations discussed in this paper. The definitions presented here along with the discussion in the next section provide the basis for understanding the logical descriptions of both unbounded spreading and blocking transformations using QFLFP discussed in sections 4 and 5.

3 Representational Assumptions

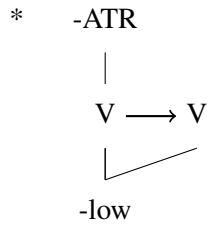
This paper explores the use of Quantifier-free Least Fixed Point logic (QFLFP) for describing vowel harmony transformations over multi-tiered autosegmental representations (ARs). Autosegmental representations (ARs) of tonal patterns generally consist of two tiers: the TBU and segmental tiers (Goldsmith, 1976; Jardine, 2016, 2017), but vowel harmony patterns refer to subsegmental features, which will be represented using multiple featural tiers. 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. The ARs used here abstract away from consonants under the assumption that they do not participate in vowel harmony and thus are not associated to vowel features; the "spine of the bottlebrush" (Hayes, 1990) only includes vowels. Association relations are represented by straight lines that connect elements (segments and features) on different tiers. Where a tier consists of multiple elements, the successor ordering relation between elements on that tier is represented with horizontal arrows. An example is shown in (6).

(6) Multi-tiered AR

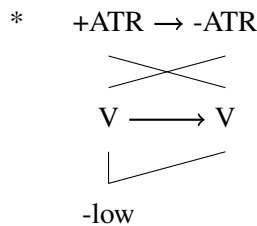


Developing logical descriptions of vowel harmony transformations provides insights into the necessity of certain restrictions on both input (underlying) and output (surface) structures. Use of ARs requires discussion of at least some of the basic representational assumptions that will be investigated in this paper. This paper assumes *Full Specification* (FS) (Clements, 1976) for surface, but not underlying representations. The *Obligatory Contour Principle* (OCP) (Leben, 1973) and the *No Crossing Constraint* (NCC) (Goldsmith, 1976; Sagey, 1986) are assumed for surface representations of spreading. Examples of structures that violate each of these constraints are shown in (7)-(9) below.

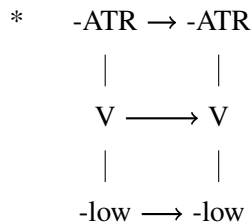
(7) Violates FS



(8) Violates NCC



(9) Violates OCP



First, assuming FS for surface structures means that each output element on a feature tier must be associated to at least one output element on the vowel tier and each output element on the vowel tier must be associated to at least one output 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 AR in (7) straightforwardly violates FS because there is a vowel that is not associated to any feature on the ATR tier. Since this paper examines vowel harmony patterns I assume that consonants can not to be associated to vowel features and so FS and vowel harmony in general ignore consonants.

Second, 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 vowel and one featural tier at a time. The representation in (8) 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.

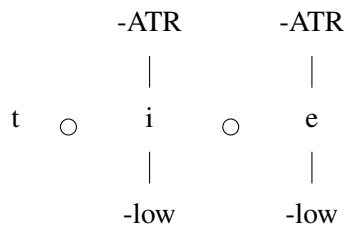
A notable effect of FS along with the NCC on surface representations is that they prevent what have been called gapped structures (Archangeli & Pulleyblank, 1994; Ringen & Vago, 1998). A gapped structure is one in which a feature appears to have skipped over a vowel that it could

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.

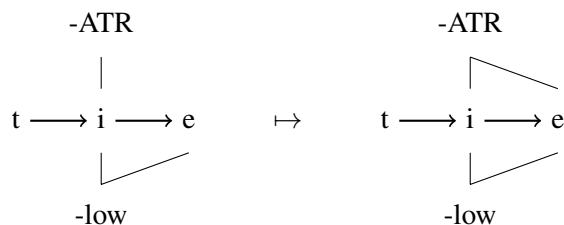
Lastly, the OCP stipulates that successive featural elements must be distinct. The representation in (9) violates the OCP because on both the ATR and low feature tiers there are two identical successive features, -ATR and -low respectively. The OCP in conjunction with FS results in representations in which multiple vowels are associated to a single feature rather than having multiple successive iterations of the same feature value on a tier 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 (11).

On the surface, 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 vowel tier, the elements on each feature tier and the associations between the feature and vowel tiers. The concatenation operation combines a finite set of adjacent graph primitives to generate a fully specified AR. For example, the AR in (11) is derived from the set of graph primitives in (10). Each primitive in (10) 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 (11). 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. An intervening element can be a vowel associated to the same feature with a different value.

(10) Concatenation of adjacent autosegmental graph primitives



(11) Surface AR satisfies FS, NCC, and OCP



In the transformation in (11), the initial consonant cannot be associated to a vowel feature in either the underlying AR (on the left) or the surface AR (on the right). While the consonant is ordered

with respect to the vowels, FS does not require the consonant to be associated to any element on either feature tier. FS also does not apply to the underlying representation, in which the second vowel is not associated to any ATR feature. On the surface, *tie* satisfies FS because the ATR feature has spread so that each vowel is associated to a feature on each of the featural tiers and all features are associated to at least one vowel. Both the underlying and surface ARs of *tie* also satisfy both the NCC and the OCP because there is only one of each feature. The features are represented on separate tiers so association lines cannot cross and there is nothing else on those tiers that could violate the OCP.

In this section I have outlined representational assumptions that will be utilized throughout this paper. These assumptions are adapted from the autosegmental literature and from my previous work with multi-tiered ARs of vowel harmony. The multi-tiered ARs used here assume a bottlebrush feature representation such that one tier consists of only vowels, which are each associated to features on different tiers. Each feature tier consists only of the + and - values of one feature. Surface ARs are assumed to obey FS, the NCC, and the OCP, but underlying ARs are subject only to the NCC and the OCP. Future extensions of this paper may further investigate the necessity of these assumptions when extending QFLFP to describe vowel harmony transformations with transparent vowels.

4 Vowel harmony patterns

Vowel harmony is considered a process of assimilation that affects only the vowels in a word. Different theories suggest different mechanisms for vowel harmony assimilation, but this paper focuses on the logical description of transformations and seeks only to probe their computational complexity. In this paper computational complexity is investigated via the expressive power of the logic used to describe vowel harmony transformations. This section will discuss the logical description of two transformations which form the basis of many attested vowel harmony patterns: unbounded spreading and blocking.

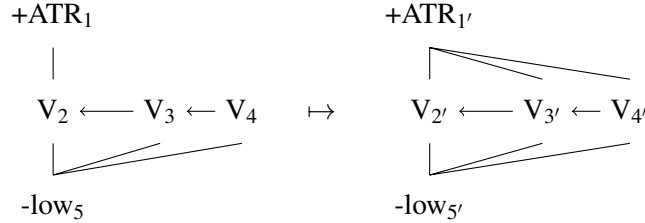
In order to describe these vowel harmony patterns the QF FO logic defined in section 2 is supplemented with a least fixed point (lfp) operator (Libkin, 2012) to describe feature spreading. This operator applies to the set of things denoted by the binary variable, which allows a formula in the definition for an output association, for example, to iteratively apply associations following a single underlying association. The *lfp* operator introduces a binary set variable, which takes two arguments. Using the *lfp* operator with QF logic requires that a formula within the scope of the *lfp* has at most two free variables. This supplemented QF logic is called QFLFP.

4.1 Unbounded spreading

A transformation in which no elements, orders, or associations change from the input to the output can be easily described using QF logic, but few vowel harmony patterns work this way. One pattern that is commonly discussed in autosegmental literature is unbounded spreading. Unbounded spreading occurs in languages with full harmony and in languages where some words undergo full harmony and others do not, like Akan. In vowel harmony terms, an unbounded rightward spreading pattern consists of an input with some number of vowels and only the first one is associated to a feature. That feature then spreads so that the output structure is fully specified. In other words, the output structure consists of the same number of vowels all associated to that same single feature. In Akan, the feature that spreads in fully harmonic words is either [+ATR] or [-ATR] and ATR spreading

interacts with low vowels, but this interaction will not be discussed until section 4.2. The ARs in (12) illustrate unbounded rightward spread of +ATR among -low vowels.

(12) Unbounded ATR Spreading



In (12) each element is represented with a subscript number, the labels are represented either by the feature's name and value or by a V, for vowels. The elements in the output AR (denoted with ') are all also present in the input AR and so the label they receive can be easily defined using QF logic, as in (13). The first definition ($P'_D(x)$) specifies that each domain element in the input is present in the output. The other three definitions label each output element with its input label.

(13) $\langle D; p, \mathcal{A}, P_V, P_{+ATR}, P_{-ATR}, P_{+low}, P_{-low} \rangle$

$$\begin{aligned} P'_D(x) &\stackrel{\text{def}}{=} P_D(x) & P'_V(x) &\stackrel{\text{def}}{=} P_V(x) \\ P'_{+ATR}(x) &\stackrel{\text{def}}{=} P_{+ATR}(x) & P'_{-low}(x) &\stackrel{\text{def}}{=} P_{-low}(x) \end{aligned}$$

Similarly, the output associations between vowels and the [-low] feature are all present in the input as well, so I can write the definition for association between vowels and low features as in (14).

$$(14) \quad \mathcal{A}'_{low}(x, y) \stackrel{\text{def}}{=} \mathcal{A}_{low}(x, y)$$

However, the current QF version of FO logic does not provide a way to define the spreading of a [+ATR] feature to an unbounded number of vowels, as in (12). Chandlee and Jardine (2019a) argue that unbounded tone spread is not A-ISL because the output association is not QF definable. In order to define an unboundedly spreading association, a least-fixed point (lfp) operator must be introduced (Libkin, 2012). The *lfp* operator introduces a recursive binary variable, which allows the evaluation of the association relation to apply iteratively over successive vowels. In this way, unbounded feature spreading can be defined as the iterative association of a feature from one underlying vowel onto all successive vowels. The unbounded spreading of +ATR from left to right in (12) can thus be defined using quantifier-free least fixed point logic (QFLFP), as in (15).

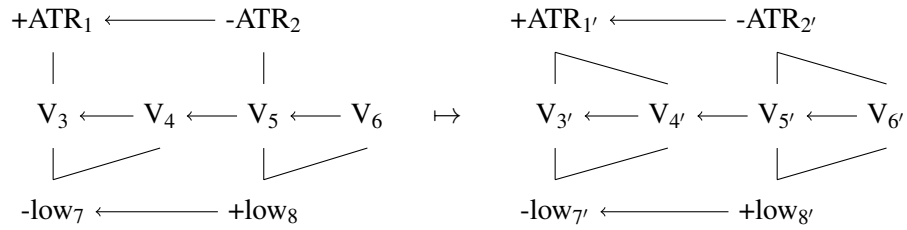
$$(15) \quad \mathcal{A}'_{ATR}(x, y) \stackrel{\text{def}}{=} [lfp \mathcal{A}_{ATR}(x, y) \vee R(x', p(y'))](x, y)$$

QFLFP requires a maximum of two free variables so the *lfp* operation requires reference to only two elements: an element on the relevant feature tier and an element on the vowel tier. As mentioned above, the x is evaluated as a featural element and y as a vocalic element; so according to the formula in (15) an output association will be drawn between the [+ATR] feature and a vowel if that vowel is underlyingly associated to the [+ATR] feature or if the preceding vowel has been associated to that feature.

4.2 Blocking

In Akan, a [+low] vowel blocks the spread of ATR. In autosegmental terms, this means that an underlying [+ATR] feature, for example, spreads only to [-low] vowels and does not associate to a [+low] vowel. The multi-tiered ARs used throughout this paper do not provide a straightforward way to illustrate such an interaction between feature tiers unless the [+low] vowel is also associated to [-ATR] in the underlying form. The lfp operator allows a feature to spread its underlying association from one vowel to multiple vowels, but if an underlyingly [+low, -ATR] vowel blocks the spread of [+ATR] the transformation will look like (16).

(16) Blocking



As before, each element in the domain is labeled by defining the unary output predicates and the same number of elements are present in the input as are in the output. However, in order to describe when spreading is blocked the formula must refer either to the second featural element on the ATR tier labeled with [-ATR] and associated to a vowel in the input (2) or to vowels that are not associated in the input AR (4 and 6). A QFLFP definition of the transformation in (16) is attempted in (17).

(17) $\langle D; p, s, \mathcal{A}, P_V, P_{+ATR}, P_{-ATR}, P_{+low}, P_{-low} \rangle$

$$\begin{aligned}
 P'_D(x) &\stackrel{\text{def}}{=} P_D(x) & P'_V(x) &\stackrel{\text{def}}{=} P_V(x) \\
 P'_{+ATR}(x) &\stackrel{\text{def}}{=} P_{+ATR}(x) & P'_{-ATR}(x) &\stackrel{\text{def}}{=} P_{+ATR}(x) \\
 P'_{+low}(x) &\stackrel{\text{def}}{=} P_{+low}(x) & P'_{-low}(x) &\stackrel{\text{def}}{=} P_{+low}(x) \\
 \mathcal{A}'_{ATR}(x, y) &\stackrel{\text{def}}{=} [lfp \mathcal{A}_{ATR}(x, y) \vee R(x', p(y')) \wedge \dots] \\
 \mathcal{A}'_{low}(x, y) &\stackrel{\text{def}}{=} [lfp \mathcal{A}_{low}(x, y) \vee R(x', p(y')) \wedge \dots]
 \end{aligned}$$

In order to refer to a second element on the ATR feature tier, the formula must introduce a third free variable, but the binary set variable introduced by the lfp operator only allows for two. In addition, the binary association relation provides a means by which to refer to elements on different tiers that are associated to one another, but there is not way to refer to unassociated elements. The alternative discussed in the next section involves drawing an analogy between the association and ordering relations and once again changing the model signature over which multi-tiered ARs are defined in QFLFP.

In this section I have shown that an unbounded spreading transformation over multi-tiered ARs can be described using QFLFP and a binary association relation, but not a blocking transformation. Unbounded spreading QFLFP definable because the lfp operator introduces a binary set variable, which allows exactly two variables to be free within the scope of the lfp predicate. The binary set

variable takes two terms, which allows the output association to be recursively defined. In other words, the QFLFP definition of spreading is an iterative process of association from successive vowels to a single feature. However, a blocking transformation is not QFLFP definable using the binary association relation because the lfp predicate can have only two free variables (x and y) and there is no way to refer to a second input feature without introducing a third variable. In addition, there is no way to refer to unassociated vowels so the lfp predicate cannot specify onto which vowels the feature spreads.

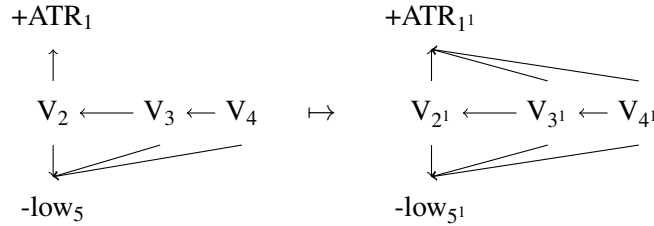
5 Association Function

Unlike with tone, vowel harmony does not allow a vowel to be associated to more than one element on a feature tier. In other words, vowel harmony patterns do not include ARs in which a single vowel is associated to multiple elements on a feature tier; multiple association can only be unidirectional. As a result, the binary autosegmental association relation introduced by Chandlee and Jardine (2019a) could be reformulated as a function (α), which associates elements on the vowel tier to an element on a feature tier. The analogy could be drawn here between association and ordering because the binary ordering relations were also changed to functions when moving from FO to QFFO logic. Again, a separate association function must be defined for each feature tier. Since featural associations are all the same type of function ($\alpha_F(x) \approx y$) they will work in the same way. In order to evaluate $\alpha_F(x) \approx y$, the variable x is evaluated as an element on the vowel tier and y as an element on a feature tier. The association function will be represented with arrows from a vowel to a feature, as in (18).

5.1 Unbounded Spreading

The Akan patterns discussed in the previous section will now be reanalyzed using the new association function. First, the same unbounded spreading pattern as in (12) is reproduced here in (18) using the association functions, α_{ATR} and α_{low} .

(18) Association as a unidirectional function



Again, each element in the output also appears in the input so the unary predicates can be easily defined with QF, as in (19). The output functions in (20) are defined in terms of the input associations by using the lfp operator and binary variable introduced in the previous section, which allows the formula to have exactly two free variables.

$$(19) \quad \langle D; p, \alpha, P_V, P_{+ATR}, P_{-low} \rangle$$

$$\begin{array}{ll} P'_D(x) \stackrel{\text{def}}{=} P_D(x) & P'_V(x) \stackrel{\text{def}}{=} P_V(x) \\ P'_{+ATR}(x) \stackrel{\text{def}}{=} P_{+ATR}(x) & P'_{-low}(x) \stackrel{\text{def}}{=} P_{-low}(x) \end{array}$$

$$(20) \quad \begin{array}{l} \text{a. } \alpha'_{ATR}(x) \approx y \stackrel{\text{def}}{=} [\text{Ifp } \alpha_{ATR}(x) \approx y \vee R(p(x'), y)](x, y) \\ \text{b. } \alpha'_{low}(x) \approx y \stackrel{\text{def}}{=} \alpha_{low}(x) \approx y \end{array}$$

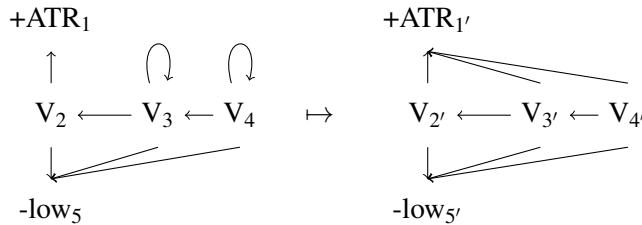
Unbounded ATR spreading is thus described using the association function as in (20a), which states that an element on the vowel tier is associated to an element on the ATR tier in the output copy if it is associated to that element on the ATR tier in the input or if its predecessor is associated to the same element on the ATR tier in the output. This description is true for 2', 3', and 4', which are all associated to 1'.

The analogy with ordering does not end at making association a function, though. Section 2.1 explains that the ordering functions are necessarily total functions, which means they are by defined for each element in the domain. This is made possible by defining external predicates, which define the first element in a string as its own predecessor and the last element in a string as its own successor. In the same way, the association function is defined for all elements on the vowel tier such that underlyingly unassociated vowels are associated to themselves. These “unspecified” vowels are also defined with an external predicate for each association function, as in (21).

$$(21) \quad \begin{array}{l} \text{a. } \text{unspec}_{ATR}(x) \stackrel{\text{def}}{=} \alpha_{ATR}(x) \approx x \\ \text{b. } \text{unspec}_{low}(x) \stackrel{\text{def}}{=} \alpha_{low}(x) \approx x \end{array}$$

The final transformation for an unbounded rightward ATR spreading pattern thus looks like (22). The unary predicates are defined logically as in (23) and association functions are defined in (24).

$$(22) \quad \text{Association as a total function}$$



$$(23) \quad \langle D; p, \alpha, P_V, P_{+ATR}, P_{-low} \rangle$$

$$\begin{array}{ll} P'_D(x) \stackrel{\text{def}}{=} P_D(x) & P'_V(x) \stackrel{\text{def}}{=} P_V(x) \\ P'_{+ATR}(x) \stackrel{\text{def}}{=} P_{+ATR}(x) & P'_{-ATR}(x) \stackrel{\text{def}}{=} P_{-ATR}(x) \\ P'_{+low}(x) \stackrel{\text{def}}{=} P_{+low}(x) & P'_{-low}(x) \stackrel{\text{def}}{=} P_{-low}(x) \end{array}$$

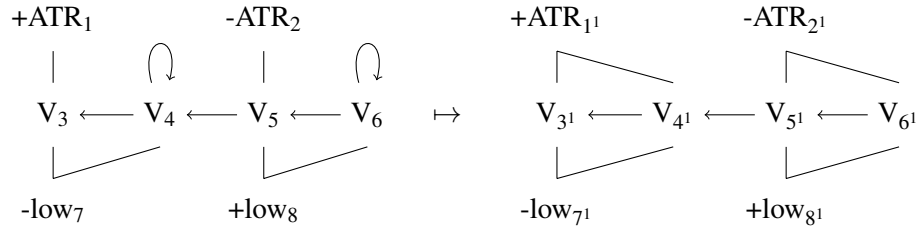
- (24) a. $\alpha'_{ATR}(x) \approx y \stackrel{\text{def}}{=} [lp\alpha_{ATR}(x) \approx y \vee R(p(x'), y')](x, y)$
 b. $\alpha'_{low}(x) \approx y \stackrel{\text{def}}{=} \alpha_{low}(x) \approx y](x, y)$

The definition of $(\alpha'_{ATR}(x) \approx y)$ effectively overwrites any input associations that are not included in the lp formula in (24a), but the $unspec_F(x)$ predicates will be useful in the next subsection.

5.2 Blocking

I proposed the association function as an alternative to the binary association relation, which could not describe a feature spreading pattern with blocking, like Akan. In the previous section I explained that the formula which defines the output ATR associations must refer to either a second ATR feature or the underlyingly unspecified vowels. The association function and $unspec$ predicates allow a QFLFP formula to refer directly to elements on the vowel tier which are both associated and unassociated to an ATR feature. In this way, QFLFP can be used to describe the blocking pattern of Akan if the blocking [+low] vowel is also underlyingly associated to [-ATR].

(25) Blocking



For this reason, the input AR in (25) has two underlying ATR associations. In the input 3 associated to 7 and 1, and the [+ATR] feature spreads so that 4 is also associated to 1 in the output. However, 5 is associated to 2 and 8 and so it cannot also be associated to 1, thus blocking the spread of [+ATR].

The definitions of the blocking transformation in (25) look similar to the definitions of the unbounded spreading transformation in (22), except for the output condition on ATR association. All of the input elements are copied into the output so the output conditions for the unary predicates are easily defined using QF, as in (26).

- (26) $\langle D; p, \alpha, P_V, P_{+ATR}, P_{-ATR}, P_{+low}, P_{-low} \rangle$
- a. $P'_D(x) \stackrel{\text{def}}{=} P_D(x)$ $P'_V(x) \stackrel{\text{def}}{=} P_V(x)$
 b. $P'_{+ATR}(x) \stackrel{\text{def}}{=} P_{+ATR}(x)$ $P'_{-ATR}(x) \stackrel{\text{def}}{=} P_{-ATR}(x)$
 c. $P'_{+low}(x) \stackrel{\text{def}}{=} P_{+low}(x)$ $P'_{-low}(x) \stackrel{\text{def}}{=} P_{-low}(x)$
- (27) a. $\alpha'_{ATR}(x) \approx y \stackrel{\text{def}}{=} [lp(\alpha_{ATR}(x) \approx y \wedge \neg x \approx y) \vee (R(p(x'), y') \wedge unspec_{ATR}(x'))](x, y)$
 b. $\alpha'_{low}(x) \approx y \stackrel{\text{def}}{=} \alpha_{low}(x) \approx y](x, y)$

The formula in (27a) states that an element on the vowel tier is associated to an element on the ATR tier in the output if it is associated to that ATR feature in the input and is not itself, or if the vocalic element is unspecified and its predecessor is associated to that same element on the ATR tier in the

output. Adding “ $\text{unspec}_{ATR}(x')$ ” to the second disjunct in the lfp predicate prevents the [+ATR] feature from spreading to the third vowel (5), which is already underlyingly associated to [-ATR]. Thus in principle, the [+low] vowel has blocked the spread of [+ATR], but only because it is also [-ATR]. In practice, writing separate definitions for the associations between the vowel tier and each feature tier prevents a process on one feature tier from referencing elements on a different feature tier.

In this section I have shown that describing both unbounded spreading and blocking transformations over multi-tiered ARs is possible using an association function ($\alpha_F(x) \approx y$). The association function is similar to the predecessor and successor ordering functions, $p(x)$ and $s(x)$ respectively, because it redefines a binary relation as a function which takes a term and outputs a term. Autosegmental association can be defined as a function for ARs of vowel harmony because it is not possible for a vowel to be associated to both the + and - minus of a feature so multiple association is only possible in one direction. Thus $\alpha_F(x) \approx y$ operates unidirectionally so x variables are evaluated as elements on the vowel tier and y variables are evaluated as elements on a feature tier. I defined predicates that allow QFLFP formulas to make reference to unspecified vowels, which are elements on the vowel tier that are associated to themselves. The association function is not quite a total function because it is defined for every element on the vowel tier, but not for elements on feature tiers.

6 Conclusion

This paper outlines a method for describing vowel harmony transformations of both unbounded spreading and blocking patterns over multi-tiered ARs using quantifier-free least fixed point logic (QFLFP). It reviews some previous work on both logical descriptions of phonological transformations and autosegmental theory then extends these to vowel harmony transformations over multi-tiered autosegmental representations (ARs). The QFLFP logical descriptions used throughout this paper reveal the necessity of using functions rather than binary relations in order to refer to terms, which pick out elements in the domain. First, the successor ordering relation is translated to a predecessor function then the use of functions is extended to ARs when the binary association relation is also translated into an association function. Unlike with tone, vowel harmony patterns allow autosegmental association to be defined as a function because multiple association is only possible in one direction; no vowel can be associated to both the + and - value of a feature.

Using the association function makes a blocking pattern over multi-tiered ARs QFLFP definable where the binary association relation did not. In order to refer to a feature on a different tier—as descriptions of blocking often do—or a second feature on the same tier, a FO formula would have to introduce a third variable, which cannot be done without a quantifier. As a function, though, association refers to terms which allows for a definition of what have been called unspecified vowels as elements with a looping association; in other words elements on the vowel tier that are associated to themselves. Being able to refer to “unspecified” vowels effectively circumvents the problem of not using quantifiers and maintains the restriction of only having two free variables within the scope of an lfp predicate.

Chandlee and Jardine (2019b) claim that unbounded spreading is not AISL because it is not QF definable, but this paper shows that it is QFLFP definable. It is thus possible—and other recent work with QFLFP provides further evidence—that QFLFP describes a superset of the ISL class of functions established by Chandlee (2014). These findings provide valuable insight into the computational complexity of vowel harmony and phonological transformations in general, which warrant further research.

Future investigations into the expressivity of QFLFP with respect to vowel harmony patterns will include developing a description of a transformation with transparent vowels. In particular, investigating the proposed underlying forms of vowel harmony transformations with transparent vowels could provide important insights into the utility of the representational assumptions held throughout this paper. More specifically, future work includes an investigation of whether or not transformations with transparent vowels require that the No Crossing Constraint (NCC) be violable.

7 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. (2019a). Autosegmental input strictly local functions. In *Transactions of the association for computational linguistics* (Vol. 7, pp. 157–168). MIT Press.
- Chandlee, J., & Jardine, A. (2019b). Quantifier-free monadic least-fixed point transductions and sequential functions.
- 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. (2016). *Locality and non-linear representations in tonal phonology* (PhD thesis). University of Delaware.
- Jardine, A. (2017). The local nature of tone association patterns. *Phonology*, 34(2), 385–405.
- 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.
- Libkin, L. (2012). *Elements of finite model theory*. Springer.
- 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.