Vowel Harmony in QFLFP

Eileen Blum[1]

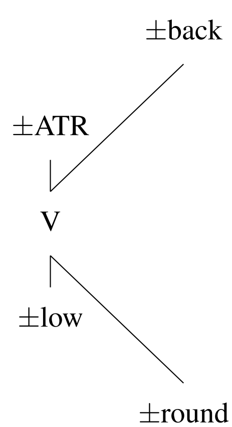
1 Introduction

Last time we talked about vowel harmony transformations with neutral vowels. We saw how QFLFP can be used to describe spreading and blocking over strings. Dine, then introduced two-tiered autosegmental representations of tone and showed that we can also use QFLFP to describe autosegmental spreading (Chandlee & Jardine, 2019; Goldsmith, 1976). Today, I will combine these two approaches to show that QFLFP can be used to describe spreading and blocking of vowel features over multi-tiered autosegmental representations.

1.1 Multi-tiered ARs of vowel harmony

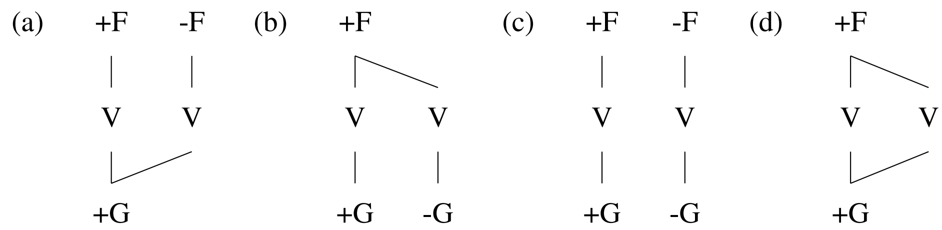
Unlike for tone, we will represent vowel harmony patterns using autosegmental representations (ARs) with more than two tiers. Ignoring consonants, we can assume there is one tier, which consists of vowels and each feature (i.e. ATR, back, low, round, etc.) is represented on a separate tier with both its + and - minus values. Assuming a bottlebrush feature configuraton, each feature tier is connected to the vowel tier via association, as shown in (1).

(1) Bottlebrush configuration with a multi-tiered AR



A single vowel is thus necessarily associated to multiple feature tiers. Assuming both the Obligatory Contour Principle (OCP) and the No Crossing Constraint (NCC), multi-tiered ARs of vowel harmony can utilize multiple association, but only in one direction; a single vowel cannot be associated to both the + and - value of a feature. The possible surface configurations of association that include both + and - values on a tier are shown in (2).

(2) Possible vowel harmony ARs



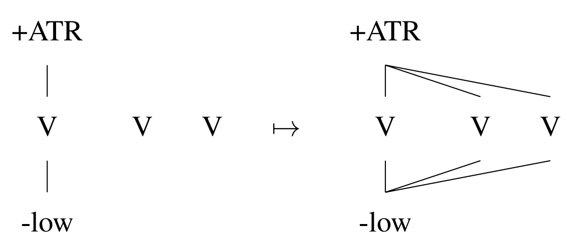
In (2a) two vowels are associated to a single +G feature and in (2b) two vowels are associated to a single +F feature. However, if we assume both the OCP and the NCC then on any feature tier if both the + and - values of a feature are present they must be associated to different vowels, as in (2a-c). Lastly, (2d) shows full spreading where both vowels are associated to a single feature on each tier.

2 Association Relation

2.1 Unbounded spreading

Following Dine's logical characterization of ARs, we could define each element on each tier and a binary association relation that holds between them. For example, a vowel harmony pattern like the one in Akan can be described as an ATR feature spreading from the left until it is blocked by a +low vowel (Clements, 1976). A word with no +low vowels will show unbounded feature spreading, as in (3).

(3) Unbounded Feature Spreading



Unbounded feature spreading is straightforwardly captured with QFLFP because all surface features are found underlyingly and the associations can be said to spread iteratively. We thus define a unary relation for vowels and each feature (separate ones for + and - values).

(4) $\langle \mathrm{D}; \mathrm{p}, \mathrm{s}, \mathscr{A}, \mathrm{P}_V, \mathrm{P}_{+ATR}, \mathrm{P}_{-ATR}, \mathrm{P}_{+low}, \mathrm{P}_{-low} \rangle$

$\mathrm{P}'_V(x) \overset{\text{def}}{=} \mathrm{P}_V(x)$

$\mathrm{P}'_{+ATR}(x) \overset{\text{def}}{=} \mathrm{P}_{+ATR}(x)$ $\quad\quad$ $\mathrm{P}'_{-ATR}(x) \overset{\text{def}}{=} \mathrm{P}_{-ATR}(x)$

$\mathrm{P}'_{+low}(x) \overset{\text{def}}{=} \mathrm{P}_{+low}(x)$ $\mathrm{P}'_{-low}(x) \overset{\text{def}}{=} \mathrm{P}_{-low}(x)$

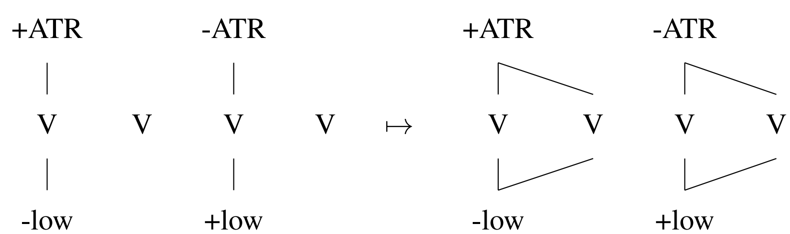
We can also define two association relations, between the vowel tier and each feature tier.

$\mathscr{A}'_{ATR}(x,y) \overset{\text{def}}{=} [lfp\mathscr{A}(x',y') \vee R(x',p(y'))](x,y)$
$\mathscr{A}'_{low}(x,y) \overset{\text{def}}{=} [lfp\mathscr{A}(x',y') \vee R(x',p(y'))](x,y)$

2.2 Blocking

However, Akan ATR harmony is blocked by a +low vowel so we need to add a restriction to these association definitions. If we stipulate that +low vowels are underlyingly specified for ATR, we will see something like (5).

(5) Blocking



Again, each feature and all the vowels only appear in the output structure if they are present in the input structure. The idea that we want to capture is that a feature spreads from one vowel onto successive *unassociated* vowels and stops when it reaches another vowel that is *underlyingly associated*. Is it possible to incorporate blocking into the definitions of the binary association relation?

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

$P'_V(x) \overset{\text{def}}{=} P_V(x)$
$P'_{+ATR}(x) \overset{\text{def}}{=} P_{+ATR}(x)$ $P'_{-ATR}(x) \overset{\text{def}}{=} P_{-ATR}(x)$
$P'_{+low}(x) \overset{\text{def}}{=} P_{+low}(x)$ $P'_{-low}(x) \overset{\text{def}}{=} P_{-low}(x)$
$\mathscr{A}'_{ATR}(x,y) \overset{\text{def}}{=} [lfp\mathscr{A}(x',y') \vee (R(x',p(y'))\wedge$

$\mathscr{A}'_{low}(x,y) \overset{\text{def}}{=} [lfp\mathscr{A}(x',y') \vee (R(x',p(y'))\wedge$

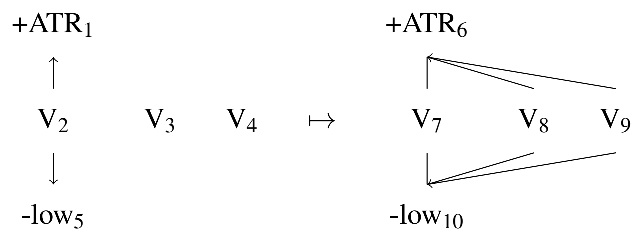
3 Association as a function

3.1 Unbounded spreading

An alternative approach would be to describe the associations between the vowel tier and each feature tier as a function. Because multiple association is possible with multi-tiered ARs, association can only be a function if it is unidirectional. If you look back at the possible structures in (2), in which direction could a function that holds between tiers operate?

A separate association function (let's call it α) must be defined for each feature tier, but since they are all the same type of function ($\alpha(x) \approx y$) they will work in the same way. An example transformation for Akan is given in (7) using the unidirectional association function represented with arrows from vowels to a feature. In order to evaluate $\alpha(x) \approx y$, the variable x would be used for an element on the vowel tier and y for an element on a feature tier.

(7) Association as a unidirectional function



We can use the same definitions for unary P relations as before. Can we also define the output associations for unbounded spreading in (7) in terms of the input associations?

(8) $\langle$D; p, s, α, P_V, P_{+ATR}, P_{-ATR}, P_{+low}, $P_{-low}\rangle$

$P'_V(x) \stackrel{\text{def}}{=} P_V(x)$

$P'_{+ATR}(x) \stackrel{\text{def}}{=} P_{+ATR}(x)$ $P'_{-ATR}(x) \stackrel{\text{def}}{=} P_{-ATR}(x)$

$P'_{+low}(x) \stackrel{\text{def}}{=} P_{+low}(x)$ $P'_{-low}(x) \stackrel{\text{def}}{=} P_{-low}(x)$

$\alpha'_{ATR}(x) \approx y \stackrel{\text{def}}{=}$

$\alpha'_{low}(x) \approx y \stackrel{\text{def}}{=}$

Let's evaluate these functions over the input structure according to our definitions and see if the results match the output structure in (7). Recall that in order for α to be a function there must be only one y for each x. Multiple xs can map to the same y, but not vice versa.

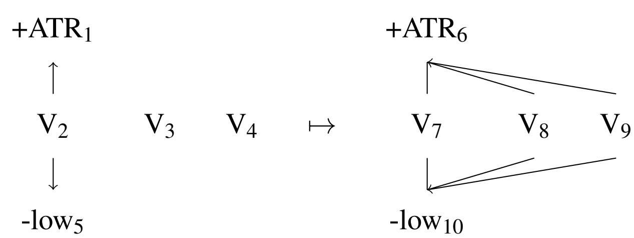
$\alpha_{ATR}(2) \approx$ $\alpha_{ATR}(3) \approx$ $\alpha_{ATR}(4) \approx$

$\alpha_{low}(2) \approx$ $\alpha_{low}(3) \approx$ $\alpha_{low}(4) \approx$

3.2 Total Function

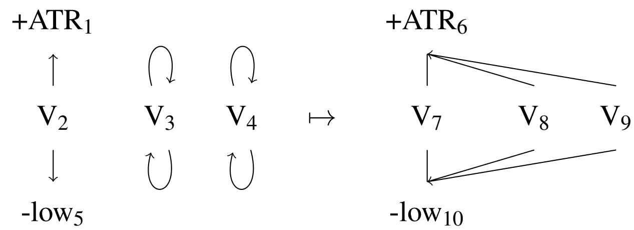
Remember the string relations that we are now considering as functions: predecessor and successor. In the current system, these are both total functions over strings because each element in the input string has a predecessor and a successor, including the first and last one respectively. In other words, the first element in a string is its own predecessor and the last element in a string is its own successor. Could we extend this idea to ARs? For ARs, how could we make association a total function? Let's look at (7) again repeated here in (9).

(9) Association as a unidirectional function



First, all elements on the vowel tier must be associated to something. Notice the input AR is underspecified because the vowels 2 and 3 are not associated to any features. We can solve this in the same way that we did for strings. We can say that unspecified vowels in the input are associated to themselves and then our definitions for output association will overwrite the input associations to spread the feature to all vowels. This will make the unbounded spreading ARs look like (10)

(10) Association as a total function

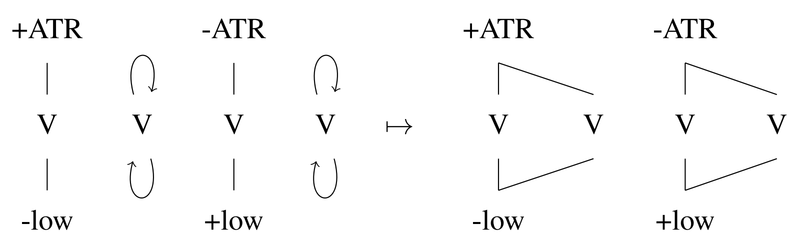


Including looping associations in the input will now allow us to refer to unspecified vowels, which we could not do with the binary association relation.

3.3 Blocking

Lastly, we cannot forget that Akan has blocking. The major advantage of defining association as a function rather than a binary relation is that we can use QFLFP to describe feature spreading with blocking. We can now write the complete AR transformation for blocking by adding to (5), as in (11).

(11) Blocking



The question remains whether or not we can define the association function using QFLFP to incorporate blocking. Using QFLFP we can now describe a transformation that spreads a feature association from one vowel only to an adjacent unspecified vowel. First, we can define an unspecified element in the same way that we defined the first and last elements in a string. How would you define the unspec(x) predicate?

$\text{unspec}_{ATR}(x) \overset{\text{def}}{=}$

$\text{unspec}_{low}(x) \overset{\text{def}}{=}$

Next, we can define the unary relations for each element in the same way we have been so far.

(12) $\langle$D; p, s, α, P_V, P_{+ATR}, P_{-ATR}, P_{+low}, $P_{-low}\rangle$

$P'_V(x) \overset{\text{def}}{=} P_V(x)$
$P'_{+ATR}(x) \overset{\text{def}}{=} P_{+ATR}(x)$ $\quad$ $P'_{-ATR}(x) \overset{\text{def}}{=} P_{-ATR}(x)$
$P'_{+low}(x) \overset{\text{def}}{=} P_{+low}(x)$ $\quad$ $P'_{-low}(x) \overset{\text{def}}{=} P_{-low}(x)$

Lastly using the predicate unspec(x) defined above, how can we define the two association functions needed to describe the Akan vowel harmony pattern with blocking?

$\alpha'_{ATR}(x) \approx y \overset{\text{def}}{=}$

$\alpha'_{low}(x) \approx y \overset{\text{def}}{=}$

4 References

Chandlee, J., & Jardine, A. (2019). Autosegmental input strictly local 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.