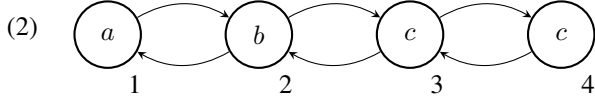
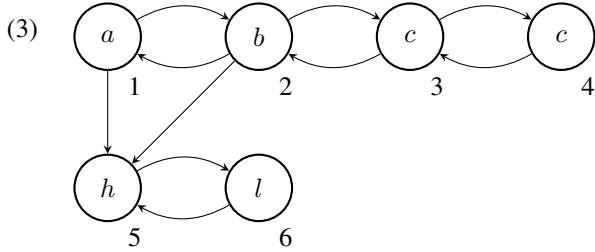


Strings on each tier are made of one or more symbols that assume a given position on that tier. For instance, in the string *abcc* shown in (2), the symbol *a* occupies position 1, *b* position 2, the first *c* position 3, and the second *c* position 4. In the diagram in (2), the top, right-pointing arrows represent the successor function and the bottom, left pointing ones represent the predecessor.



Symbols in a string have properties, such as the property of being a *c* or the property of occupying position 3, or that of being a *c* in position 3. It is these properties that unary relations refer to. As such, *a* and *b* are labels of positions 1 and 2, respectively. The order functions predecessor (*p*) and successor (*s*) take positions as inputs and outputs. Thus, given the string *abcc*, $p(2) = 1$ and $s(2) = 3$. That is, the predecessor of position 2 is position 1 and the successor of position 2 is position 3.

We then extend string models to autosegmental representations by adding an association function α that connects two distinct strings. The below shows a string *hl* and a string *abcc* in which *h* has been associated to both *a* and *l*.



In (3), $\alpha(1) = 6$ and $\alpha(2) = 6$. Note that with association as a function, and not a relation as is usual (see, e.g., Kornai 1995) our ability to represent contours seems limited, but functions allow us to capture long-distance patterns. In §4, we discuss how to derive both long-distance patterns and contours using functions.

We can use logical transductions to define input-output mappings in the following way. For example, consider an association rule like the one represented in (3), where the symbol *h* in the string *hl* on a given tier associates to *a* or a *b* that immediately follows an *a* on another tier. Note that no association exists between the symbols of the string *hl* and *abcc* in the input but upon its application, the association rule outputs a partially associated structure between the autosegments as shown in (4).

$$(4) \quad \begin{array}{ccc} a b c c & & a b c c \\ & \rightarrow & \vee \\ h l & & h l \end{array}$$

To capture the association rule in (4) using logical transductions, we need an input signature $S_i = \{P_a, P_b, P_h, P_l, p\}$ and an output signature $S_o = \{P_a, P_b, P_h, P_l, p, \alpha'\}$. Transductions list the conditions that need to be true (on the right of the equal sign) for x (a variable of the timing tier representing TBUs) and y (a variable of the tonal tier representing tones) to be associated on the surface. Surface associations are represented as $\alpha'(x) \approx y$. The transduction in (5) is written in the logical language of the input signature from which basic predicates are drawn. These basic predicates can be connected with the standard boolean connectives such as conjunction (\wedge), disjunction (\vee), negation (\neg), and implication (\rightarrow).

$$(5) \quad \alpha'(x) \approx y \stackrel{\text{def}}{=} \underbrace{(P_a(x) \wedge P_h(y))}_{(i)} \vee \underbrace{(P_b(x) \wedge P_h(y) \wedge P_a(p(x)))}_{(ii)}$$

The transduction above has two disjunctions and reads as follows: x and y are associated in the output if x is an *a* and y is a *h* in the input (i.e. if x and y satisfy $P_a(x) \wedge P_h(y)$) or if x is a *b*, y is a *h* and the predecessor of x is an *a* (i.e. if they satisfy $P_b(x) \wedge P_h(y) \wedge P_a(p(x))$). This guarantees that *h* will only be associated to an *a* and to a *b* that is preceded by an *a*. Since *l* was not referenced in the definition, it does not participate in the association process in (4). We now turn to cases with iterative applications of a rule and how they are QFLFP-definable.

2.2 Quantifier-Free Least Fixed Point Operators QFLFP operators (Chandlee & Jardine, 2019b) are made up of two parts, a quantifier-free (QF) part and a least fixed point (LFP) part. As the name suggests, a quantifier-free logic does not use quantifiers in its syntax; that is, when new variables are introduced, they are neither bound with \exists nor with \forall , they remain free. For this reason, QF logic can only refer to a local, fixed-size window around a given position and not to positions that are unboundedly far away. Lindell & Chandlee (2016) shows that QF transductions correspond exactly to Input Strictly Local functions, which are computable with deterministic finite state machines that only reference a local position in the input. When extended with the least fixed point logic (Libkin, 2004), the resulting QFLFP corresponds to a subset of subsequential functions, namely the Output Strictly Local (OSL) functions (Chandlee & Jardine, 2019b). OSL functions capture the iterative character of output-based locality, where targets of a given process are within a local window relative to their triggers in the output.

We follow Koser et al. (2019) in using Rogers (1997)’s implicit definitions as a shorthand for the full QFLFP formalism (Chandlee & Jardine, 2019b). Implicit definitions simply reference the object being defined in its own definition.

To illustrate, let’s consider tone spreading in Ndebele, where a H tone spreads unboundedly to any number of TBUs until the antepenultimate syllable (Sibanda, 2004; Hyman, 2011) as shown in (6).

(6) Ndebele Tone Spread (adapted from Hyman 2011; bolding added to highlight H-toned TBUs.)

Input	Output	Gloss
a. /ú-ku-hlek-a/	ú-kú -hlek-a	‘to laugh’
b. /ú -ku-hlek-is-a/	ú-kú-hlé k-is-a	‘to amuse (make laugh)’
c. /ú -ku-hlek-is-an-a/	ú-kú-hlé k-ís -an-a	‘to amuse each other’

The formula in (7) and (8) are disjuncts of a QFLFP transduction for the unbounded spread in Ndebele, where x and y are variables ranging over positions on the timing and melody tiers, respectively. Each part (a disjunct in this case) is a condition of association, one of which needs to be true before x can be associated to y in the output.

$$(7) \quad \alpha(x) \approx y$$

The first disjunct shown in (7) above states that a tone and a TBU must be associated to each other in the input for them to be associated in the output as shown in (9-i).

$$(8) \quad \alpha'(p(x)) \approx y \wedge \neg(\text{pen}(x) \vee \text{last}(x))$$

The second disjunct in (8) states that the predecessor of the TBU must be associated to the tone in the output and that the TBU can neither be the penult nor the last TBU on the timing tier, as shown in (9-ii).

$$(9) \quad \begin{array}{ccccccc} \mu \mu \mu \mu \mu & & \mu \mu \mu \mu \mu & & \mu \mu \mu \mu \mu & & \mu \mu \mu \mu \mu \\ | & \xrightarrow{(i)} & | & \xrightarrow{(ii)} & | & \xrightarrow{(ii)} & | \\ \text{H} & & \text{H} & & \text{H} & & \text{H} \\ \text{(input)} & & & & & & \text{(output)} \end{array}$$

So, for a tone to be associated to a TBU, either the disjunct in (7) or that in (8) must be true. The full definition is shown in (10) below.

$$(10) \quad \alpha'(x) \approx y \stackrel{\text{def}}{=} \underbrace{(\alpha(x) \approx y)}_{(i)} \vee \underbrace{(\alpha'(p(x)) \approx y \wedge \neg(\text{pen}(x) \vee \text{last}(x)))}_{(ii)}$$

Note that because the second disjunct refers to the output, any string position after H is guaranteed to be outputted as H, regardless of how far away it is from the trigger in the input. This is what it means for the iterative application of a rule to be QFLFP-definable, a restrictive (because it makes no use of quantifier), yet sufficiently expressive to model long distance processes. In the sections below, we elaborate on how to incorporate multi-tiered representation into QFLFP logic, in order to derive, among other processes, the UC processes which have been established to be properly regular over strings (see Jardine 2016 for details). Furthermore, the section also highlights the importance of underlying associations in UC processes.

In the Ndebele example and the examples that follow in §3, knowing which tone-TBUs are underlyingly associated is key in determining where a given span starts and where it ends. In §3, we present analyses of different long distance tone processes using the schema as in (10). The processes include, in addition to the unbounded spreading in Ndebele (Sibanda, 2004; Hyman, 2011; Chandlee & Jardine, 2019a) discussed above, the unbounded anticipatory upstep in Amo (Hyman, 1979, 2011), UTP in Luganda (Hyman & Katamba, 2010) and Ternary Tone Spread in Copperbelt Bemba (Bickmore & Kula, 2013). The boldings in the different examples are our own addition.

3 Analyses

The transductions in each of the cases presented below capture the tone process of interest in that language. Again, transductions list the conditions that need to be true for x (a variable of the timing tier representing TBUs) and y (a variable of the tonal tier representing tones) to be associated on the surface. $p(x)$ still indicates the immediate predecessor of x , and $s(x)$, its successor. $\alpha(x)$ (without apostrophe) indicates the element to which x is associated in the input, and $\alpha'(x)$ (with apostrophe) indicates the element to which x is associated in the output. Custom predicates like $first(x)$, $pen(x)$ and $last(x)$, respectively indicates that x is the *first*, *penult* and *last* element on its tier (Koser et al., 2019).

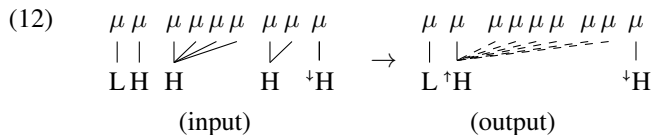
In the analyses below, we assume that any tones that are not associated to some TBU in the output are deleted. This is for expositional simplicity, and is not crucial to the analysis. To see how deletion is accomplished in logical formalisms, see, e.g., Strother-Garcia (2017).

3.1 Unbounded Anticipatory Upstep (UAU) in Amo Amo (Benue-Congo, Nigeria) has an anticipatory upstep process whereby a H tone upsteps in anticipation to a phonemic downstep that can be unboundedly far ahead (Hyman, 2011, 1979) as shown in (11). This anticipatory upstep affects a whole series of H tones preceding the anticipated ${}^4\text{H}$. We assume that the downstepped H (${}^4\text{H}$) has either phonologized or is derived through a downstep process whose output feeds the current anticipatory upstep. The data is presented in (11).

(11) Anticipatory Upstep in Amo (Hyman, 1979, 2011)

Input to UAU	Output	Gloss
a. kité úkóómí fínáwà	kité úkóómí fínáwà	‘the place ...of the animal’
b. kité úkóómí fíká ⁴ lé	kì ⁴ té úkóómí fíká ⁴ lé	‘the place ...of the monkey’

Based on Hyman (1979)’s observation that the upstep affects a whole series of Hs, we assume the structure in (12), where there are several underlying Hs preceding the ${}^4\text{H}$ but on the surface, it is a single H that multiply associates while undergoing upstep. This assumption is mainly motivated by the fact that there is no report of there being several upsteps before the downstep. That is, if the anticipatory upstep only raises the series of preceding Hs one step up, it must then be the case that all preceding Hs act like a single H. This reading leads to a crucial assumption we make about the Amo data, that OCP is enforced among adjacent surface H tones (Leben, 1978) that are realized at the same pitch level. This provides a motivation for the assumption that the non-associated tones delete on the surface. Furthermore, Odden (1986) argues that any OCP enforcing language must have an explicit mechanism to collapse adjacent identical tones into a single tone. Under this view, a tone collapsing (or deleting) mechanism will have to exist in Amo and may be independently motivated but we abstract away from showing it here.



Because we assumed that downstep is readily accessible to the upstep process, the logical characterization of the anticipatory upstep uses signatures extended with ${}^+$ and 4 , indicating upstep and downstep, respectively. The input signature of Amo is thus $S_{a/i} = \{L, {}^4\text{H}, \text{H}, p, s, \alpha\}$ and its output signature is $S_{a/o} = \{L', {}^4\text{H}', {}^+\text{H}', \text{H}', p', s', \alpha'\}$, where we use H , ${}^4\text{H}$, as clearer notational variants of the unary predicates P_{H} , $P_{{}^4\text{H}}$, etc. The anticipatory character of the process clearly shows that the trigger is

unboundedly to the right, so the function reads symbols from right-to-left. The transduction for this process has two parts, the first of which has three (3) disjuncts: the first part defines the underlying associations that survive in the output and the tone that is multiply associated, while the second part defines when an H upsteps on the surface.

The first disjunct of the first part, shown in (17)a, is repeated in (13) below.

$$(13) \quad (L(y) \vee {}^{\uparrow}H(y) \vee L(s(y))) \wedge \alpha(x) \approx y$$

According to (13), for x and y to be associated in the output, they must be underlyingly associated (i.e. $\alpha(x) \approx y$) and y must either be L or ${}^{\uparrow}H$ or the successor of y must be L (i.e. $L(y) \vee {}^{\uparrow}H(y) \vee L(s(y))$). The second disjunct in (14) requires that x and y be underlyingly associated (i.e. $\alpha(x) \approx y$) and that the successor of y be ${}^{\uparrow}H$ (i.e. ${}^{\uparrow}H(s(y))$). According to this disjunct, only the last H of any unbroken stretch of Hs will survive in the output. This is the part that enforces the Obligatory Contour Principle (OCP).

$$(14) \quad {}^{\uparrow}H(s(y)) \wedge \alpha(x) \approx y$$

The third and last disjunct shown in (15) requires that the position x is associated to, be labelled H (i.e. $H(\alpha(x))$) and that the successor of y be ${}^{\uparrow}H$ (i.e. ${}^{\uparrow}H(s(y))$), and last but not least, that the successor of x be associated to y in the output (i.e. $\alpha'(s(x)) \approx y$). Because it refers to the output, the latter part encodes iterative association, which results in multiple association.

$$(15) \quad H(\alpha(x)) \wedge {}^{\uparrow}H(s(y)) \wedge \alpha'(s(x)) \approx y$$

The second part of the transduction, shown in (17)b and repeated in (16), defines an upstep y as one whose immediate successor is a ${}^{\uparrow}H$ (i.e. ${}^{\uparrow}H(s(y))$). When the stated condition is met, this second part upsteps the multiply associated H tone that precedes the ${}^{\uparrow}H$, yielding a multiply associated ${}^{\uparrow}H$.

$$(16) \quad {}^{\uparrow}H(y) = H(x) \wedge {}^{\uparrow}H(s(y))$$

The complete transduction for the anticipatory upstep in Amo is presented in (17) below.

$$(17) \quad \text{a. } \alpha'(x) \approx y \stackrel{\text{def}}{=} ((L(y) \vee {}^{\uparrow}H(y) \vee L(s(y))) \wedge \alpha(x) \approx y) \vee \\ ({}^{\uparrow}H(s(y)) \wedge \alpha(x) \approx y) \vee \\ (H(\alpha(x)) \wedge {}^{\uparrow}H(s(y)) \wedge \alpha'(s(x)) \approx y)$$

$$\text{b. } {}^{\uparrow}H(y) = H(x) \wedge {}^{\uparrow}H(s(y))$$

Note also that $L'(x) = L(x)$, ${}^{\uparrow}H'(x) = {}^{\uparrow}H(x)$, and $H'(x) = H(x) \wedge L(s(y))$.

A derivation for the example in (11)-b is given in (18) with a reduced number of syllables for simplicity. Note that the unassociated tones deleted in the output AR in (18), as discussed above.

$$(18) \quad \begin{array}{ccccccc} \mu & \mu & \mu & \mu & \mu & \mu & \\ | & / & | & | & | & & \\ \text{LH} & \text{HH} & {}^{\uparrow}\text{H} & & & & \end{array} \rightarrow \begin{array}{ccccccc} \mu & \mu & \mu & \mu & \mu & \mu & \\ | & & | & | & | & & \\ \text{LH} & \text{HH} & {}^{\uparrow}\text{H} & & & & \end{array} \rightarrow \begin{array}{ccccccc} \mu & \mu & \mu & \mu & \mu & \mu & \\ | & & \backslash & | & | & & \\ \text{LH} & \text{HH} & {}^{\uparrow}\text{H} & & & & \end{array} \rightarrow \begin{array}{ccccccc} \mu & \mu & \mu & \mu & \mu & \mu & \\ | & & \backslash & \backslash & | & | & \\ \text{LH} & \text{HH} & {}^{\uparrow}\text{H} & & & & \end{array}$$

(Input)

$$\rightarrow \begin{array}{ccccccc} \mu & \mu & \mu & \mu & \mu & \mu & \\ | & \backslash & \backslash & \backslash & | & | & \\ \text{LH} & \text{HH} & {}^{\uparrow}\text{H} & & & & \end{array} \rightarrow \begin{array}{ccccccc} \mu & \mu & \mu & \mu & \mu & \mu & \\ | & \backslash & \backslash & \backslash & | & | & \\ \text{LH} & \text{H} & {}^{\uparrow}\text{H} & {}^{\uparrow}\text{H} & & & \end{array} \rightarrow \begin{array}{ccccccc} \mu & \mu & \mu & \mu & \mu & \mu & \\ | & \backslash & \backslash & \backslash & | & | & \\ \text{L} & & & & {}^{\uparrow}\text{H} & {}^{\uparrow}\text{H} & \end{array}$$

(output)

This successfully creates a span of upstepped H tones exactly in the case in which they are followed by a downstepped H tone. Note that this is crucially accomplished using the OCP, which allows all TBUs associated to H tones preceding a downstepped H to associate to a single, upstepped H.

3.2 Unbounded Circumambient Spread in Luganda UTP is a non-myopic long distance process triggered by (at least) two H tones that can be arbitrarily far away from each other. The tonally unspecified TBUs in-between the two tones are targets to this process by getting realized as H tones (see (19-d)) (Hyman & Katamba, 2010). The H realization of tonally unspecified TBUs creates an H tone plateau that starts with the first tone and ends with the second. The double triggers in both directions and their unbounded distance with each other is what earned UTP its UC process designation (Jardine, 2016).

(19) UTP in Luganda (Hyman & Katamba, 2010)

Input	Ouput	Gloss
a. /kitabo/	kitabo	‘book’
b. /mutéma/	mutéma	‘chopper’
c. /kisikí/	kisikí	‘log’
d. /mutéma+bisikí/	mutémábisikí	‘log chopper’
e. Unattested:	*mutéma+bisikí	–

The UTP transduction (shown in (23)) has three disjuncts that specify the different conditions that need to be met for x and y to be associated in the output. And because these are disjunctions, they do not have to be true at the same time. We give the first disjunct below.

(20) $\alpha(x) \approx y \wedge first(y)$

The first disjunct in (20) states that x and y must be associated in the input, with the condition that y must be the first element on its tier, the tone tier. This disjunct retains any underlying association to the first tone, but crucially does not preserve an association to the last tone.

(21) $\alpha'(p(x)) \approx y \wedge \neg(last(y) \wedge (last(\alpha(x))))$

The second disjunct in (21) requires that the predecessor of x be associated to y given that the position to which x is associated in the input, is not last. This conjunct is what captures the actual substance of UTP. It requires of some x and y that are not associated in the input to be associated if the predecessor of x is associated to y , where neither y nor the position to which x is associated are last. The requirement that y can not be the last element guarantees that two H tones will be needed before creating associations between pairs of elements that are not associated in the input.

(22) $last(\alpha(x)) \wedge first(y)$

This last disjunct requires that x be associated to the first y if the position to which it was associated to is last in the input. Note that this last disjunct fixes the fact that the disjunct in (21) excludes the TBU that is underlyingly associated to the last of the tones to avoid a many-to-one association between the two tones and a single TBUs. The full transduction for UTP is shown in (23) and a derivation in (24). Similar to the way we handled non-associated surface Hs in Amo, we assume that the second of the two tones involved in UTP deletes through an OCP enforcing mechanism.

(23) $\alpha'(x) \approx y \stackrel{\text{def}}{=} \underbrace{(\alpha(x) \approx y \wedge first(y))}_{(i)} \vee \underbrace{(\alpha'(p(x)) \approx y \wedge \neg(last(y) \wedge (last(\alpha(x)))))}_{(ii)} \vee \underbrace{(last(\alpha(x)) \wedge first(y))}_{(iii)}$

(24)
$$\begin{array}{ccccccc} \mu & \mu & \mu & \mu & \mu & & \\ | & | & & & & & \\ \text{H} & \text{H} & & & & & \end{array} \xrightarrow{(i)} \begin{array}{ccccccc} \mu & \mu & \mu & \mu & \mu & & \\ | & | & | & | & | & & \\ \text{H} & & \text{H} & & & & \end{array} \xrightarrow{(ii)} \begin{array}{ccccccc} \mu & \mu & \mu & \mu & \mu & & \\ | & | & | & | & | & & \\ \text{H} & & \text{H} & & & & \end{array} \xrightarrow{(ii)} \begin{array}{ccccccc} \mu & \mu & \mu & \mu & \mu & & \\ | & | & | & | & | & & \\ \text{H} & & \text{H} & & & & \end{array} \xrightarrow{(iii)} \begin{array}{ccccccc} \mu & \mu & \mu & \mu & \mu & & \\ | & | & | & | & | & & \\ \text{H} & & \text{H} & & & & \end{array}$$

3.3 Ternary Spread in Copperbelt Bemba Another UC tone process is found in Copperbelt Bemba, where the H tone spreads unboundedly to the right unless there is another underlying H that can be arbitrarily far to the right (Bickmore & Kula, 2013). In this case, the first H only spreads up to three TBUs at most, obeying the OCP. Since the language has its own OCP enforcing mechanism, which consists of leaving one unassociated TBU between the two tones on the surface, no additional OCP-enforcing mechanism is needed. The full QFLFP transduction for the ternary spread, given in (28), has three disjuncts whose meaning is given in prose below.

$$(25) \alpha(x) \approx y \wedge last(y)$$

The first disjunct in (25) states that x must be associated to the non-last y in the input, a condition that preserves underlying associations to the last tone.

$$(26) \alpha'(p(x)) \approx y \wedge last(y)$$

The second disjunct in (26) requires that the predecessor of x be associated to the last y in the output. This captures the unbounded spreading of the last tone by referring to the output and keeping track of the tone the preceding TBU attaches to.

$$(27) \alpha(x) \approx y \vee \alpha(p(x)) \approx y \vee \alpha(p(p(x))) \approx y \wedge \neg(s(x) \approx s(y))$$

The last conjunct above captures the ternary spread part of the Copperbelt Bemba process. It states that x and y are associated in the output if they are associated in the input or if the predecessor of x is associated to y or the predecessor of its predecessor is, such that it is not the case that the immediate successors of both x and y are associated. The first part of this disjunct limits to three the maximum number of TBUs that can be associated to a non-final tone (thus the idea of the ternary spread), while the last part of it enforces OCP by leaving at least one unassociated TBU between two multiply associated tones.

$$(28) \alpha'(x) \approx y \stackrel{\text{def}}{=} \underbrace{(\alpha(x) \approx y \wedge last(y))}_{(i)} \vee \underbrace{(\alpha'(p(x)) \approx y \wedge last(y))}_{(ii)} \vee \underbrace{((\alpha(x) \approx y \vee \alpha(p(x)) \approx y \vee \alpha(p(p(x))) \approx y) \wedge \neg(s(x) \approx s(y)))}_{(iii)}$$

The full transduction is shown in (28) and a derivation in (29).

$$(29) \begin{array}{ccc} \begin{array}{c} \mu \mu \mu \mu \mu \mu \\ | \quad | \\ \text{H} \quad \text{H} \\ \text{(input)} \end{array} & \text{a.} & \begin{array}{c} \xrightarrow{(i)} \begin{array}{c} \mu \mu \mu \mu \mu \mu \\ | \quad | \\ \text{H} \quad \text{H} \end{array} \xrightarrow{(ii)} \begin{array}{c} \mu \mu \mu \mu \mu \mu \\ | \quad | \\ \text{H} \quad \text{H} \end{array} \xrightarrow{(ii)} \begin{array}{c} \mu \mu \mu \mu \mu \mu \\ | \quad | \\ \text{H} \quad \text{H} \end{array} \end{array} \\ & \text{b.} & \begin{array}{c} \xrightarrow{(iii)} \begin{array}{c} \mu \mu \mu \mu \mu \mu \\ | \quad | \\ \text{H} \quad \text{H} \end{array} \xrightarrow{(iii)} \begin{array}{c} \mu \mu \mu \mu \mu \mu \\ | \quad | \\ \text{H} \quad \text{H} \end{array} \rightarrow \begin{array}{c} \mu \mu \mu \mu \mu \mu \\ | \quad | \\ \text{H} \quad \text{H} \end{array} \end{array} \end{array}$$

In the derivation, the second H spreads unboundedly to the last TBU, while the first H only spreads twice. Since the ternary spreading respects OCP, it always leaves one unassociated TBU between the two tones, even if this means spreading once or not at all. The transduction in (28) captures this through $\neg(s(x) \approx s(y))$.

4 Discussion

In this paper we present a characterization of unbounded circumambient tone processes using the Quantifier-Free Least Fixed Point operators, which correspond to a subset of subsequential functions, namely the output strictly local functions (Chandlee & Jardine, 2019b). UC processes are demonstrably properly regular (Jardine, 2016), and in order to capture them using a logic that corresponds to a subregular class in the Chomsky Hierarchy, we need to find a way to increase the expressivity of our system without making it overly powerful. One way, presented in this paper, to achieve this is to enrich the representation by adopting autosegmental representations (Goldsmith, 1976) over string representations.

However, using autosegmental representations as association functions seems to suggest that we can only capture multiple associations of tones or contours, but not both. This is because functions only allow for either many-to-one or one-to-many associations. Using relations instead of functions would allow for both types of associations, but relations will make key parts of the syntax used in the above transductions, impossible. For instance, a predicate like $H(\alpha(x))$, which points to the position to which x is associated can not be expressed using a relation. One way to get around this problem is to allow for two different association functions, one that enforces the one-to-many associations, deriving multiply associated tones and another that enforces the many-to-one associations, deriving contours tones. Having a different function for each of these processes supports the idea, argued for in Zoll (2003), that multiple association and contour formation are two distinct phenomena. While it is possible for the two functions to interact, analysing the nature of their interactions is beyond the scope of this paper.

The current results suggest that ARs do not only ‘*resolve certain formal and substantive problems*’ in generative phonology as argued by Goldsmith (1976:2), they also and most importantly have a bearing on the expressivity of the processes they model. As such, they provide a straightforward way of extending the notion of subsequentality to the otherwise non-subsequential tone processes without increasing the expressive power of the logic needed for that purpose. Furthermore, the subregular hypothesis put forth for phonology by Heinz (2018) and according to which phonological processes belong to restrictive and well-defined areas of regular stringsets and maps, seems to hold even for complex tone processes. This combination of QFLFP logic and ARs seems to carve out a new class of functions within the subregular region. This class includes the complex UC processes discussed above (i.e UTP in Luganda and the Ternary Spread in Copperbelt Bemba) in addition to local processes.

Despite the clearly more expressive character of this new class, it is unclear what the upper bound to its expressivity is; that is, a full investigation of the other patterns it can capture is yet to be conducted. One way proposed in Koser et al. (2019) to constrain its expressivity, is to allow either the predecessor or the successor functions (but not both) in the recursive bit of the QFLFP logic.¹ This was shown to prevent the logic from generating the undesirable structure with *massively crossing association lines* (Koser et al., 2019:9). Whether further restrictive measures are necessary is left for future work. Recent work by Dolatian & Rawski (2020) pursue a very similar class of functions, the Multi-Input Strictly Local (MISL) class of functions, with an empirical coverage that appears to be similar to the one carved out by the QFLFP+AR system. Furthermore, Jardine (to appear) defines an OCP-driven class of formal grammars called the Melody Local (ML) grammars. Current work is studying how the MISL and ML grammars can be combined to create a restricted class of autosegmental functions. Work along these lines will further shed light on our understanding of the right characterization for tone and the interplay between representation and computation.

5 Conclusion

In conclusion, enriching the representation by shifting from a string-based representation to ARs allows us to use the same logical power for both segmental and tonal processes. Although this work shows that QFLFP over ARs is strictly more expressive than the subsequential class that QFLFP corresponds to (Chandlee & Jardine, 2019b), how much more expressive is an open question. While QFLFP logic is restrictive in that recursion can only be defined locally, future work will characterize exactly the range of processes definable by QFLFP over ARs. A related question is whether this class can distinguish Copperbelt Bemba from ‘true’ sour-grapes spreading (Pater, 2018; O’Hara & Smith, 2019). Regardless, it is clear that computational study confirms the notion from phonological theory that representation matters.

¹ From the perspective of implicit definitions, the recursive part is the part of the definition that refers to the output.

References

- Bickmore, Lee S & Nancy C Kula (2013). Ternary spreading and the ocp in copperbelt bemba. *Studies in African Linguistics* 42:2, 101–132.
- Chandlee, Jane & Adam Jardine (2019a). Autosegmental input strictly local functions. *Transactions of the Association for Computational Linguistics* 7, 157–168.
- Chandlee, Jane & Adam Jardine (2019b). Quantifier-free least fixed point functions for phonology. *Proceedings of the 16th Meeting on the Mathematics of Language*, 50–62.
- Dolatian, Hossep & Jonathan Rawski (2020). Multi-input strictly local functions for templatic morphology. *Proceedings of the Society for Computation in Linguistics (SCiL)* 282–296.
- Goldsmith, John (1976). *Autosegmental Phonology*. Ph.D. thesis, Massachusetts Institute of Technology.
- Heinz, Jeffrey (2018). The computational nature of phonological generalizations. Hyman, Larry & Frans Plank (eds.), *Phonological Typology*, De Gruyter Mouton, 126–195.
- Heinz, Jeffrey & Regine Lai (2013). Vowel harmony and subsequentiality. Kornai, Andras & Marco Kuhlmann (eds.), *Proceedings of the 13th Meeting on Mathematics of Language*, Sofia, Bulgaria.
- Hyman, Larry M (1979). A reanalysis of tonal downstep. *Journal of African Languages and Linguistics* 1:1, 9–30.
- Hyman, Larry M (2011). Tone: Is it different. *The handbook of phonological theory* 2.
- Hyman, Larry M & Francis X Katamba (2010). Tone, syntax, and prosodic domains in luganda. *UC Berkeley PhonLab Annual Report* 6:6.
- Jardine, Adam (2016). Computationally, tone is different. *Phonology* 33:2, 247–283.
- Jardine, Adam & Jeffrey Heinz (2016). Locality and learning over autosegmental representations. Talk, Linguistic Society of America, Washington, DC.
- Jardine, Adam (to appear). Melody learning and long-distance phonotactics in tone. *Natural Language & Linguistic Theory*.
- Kornai, András (1995). *Formal Phonology*. Garland Publication.
- Koser, Nathan, Christopher Oakden & Adam Jardine (2019). Tone association and output locality in non-linear structures. *Supplemental Proceedings AMP 2019*.
- Leben, William R (1978). The representation of tone. *Tone*, Elsevier, 177–219.
- Libkin, Leonid (2004). *Elements of Finite Model Theory*. Berlin: Springer-Verlag.
- Lindell, Steven & Jane Chandlee (2016). A logical characterization of strictly local functions. Paper presented at the Fourth Workshop on Natural Language and Computer Science (NLCS '16), Columbia University, New York, NY.
- Mohri, Mehryar (1997). Finite-state transducers in language and speech processing. *Computational Linguistics* 23:2, 269–311.
- Odden, David (1986). On the role of the obligatory contour principle in phonological theory. *Language* 353–383.
- O'Hara, Charlie & Caitlin Smith (2019). Computational complexity and sour-grapes-like patterns. *Proceedings of the Annual Meetings on Phonology*, vol. 7.
- Pater, Joe (2018). Substance matters: a reply to Jardine (2016). *Phonology* 35:1, 151–156.
- Rogers, James (1997). Strict It2 : Regular :: Local : Recognizable. Retoré, Christian (ed.), *Logical Aspects of Computational Linguistics: First International Conference, LACL '96 Nancy, France, September 23–25, 1996 Selected Papers*, Springer Science & Business Media, 366–385.
- Sibanda, Galen (2004). Verbal phonology and morphology of ndebele. berkeley, ca: University of california dissertation.
- Strother-Garcia, Kristina (2017). Imdlawn Tashlhiyt Berber syllabification is quantifier-free. *Proceedings of the first annual meeting of the Society for Computation in Linguistics*, vol. 1, 145–153.
- Zoll, Cheryl (2003). Optimal tone mapping. *Linguistic Inquiry* 34:2, 225–268.