Charlie O'Hara & Caitlin Smith
University of Southern California

Computational Complexity and Sour-Grapes-Like Patterns

The set of attested phonological input-output mappings is smaller than the set of all logically possible input-output mappings; attested phonological patterns appear bounded by **computational complexity**. The Subregular Hypothesis (Heinz 2011) claims all attested phonological mappings are a proper subset of the class of regular input-output mappings. The formal characterization of that subset is the subject of ongoing study. We propose that the class of **weakly deterministic** mappings (Heinz & Lai 2013) is larger than previously assumed, and as a result encompasses all attested phonological patterns (c.f. Jardine 2016). However, this expanded weakly deterministic class is still smaller than the class of all regular mappings. Crucially, sour grapes spreading, an unattested pattern described by Wilson (2003), is shown to be regular but not weakly deterministic. True sour grapes can be contrasted with what we claim are cases of **false sour grapes**, attested sour-grapes-like patterns that we propose are less computationally complex than true sour grapes.

In true sour grapes, a potential undergoer U that is preceded at any distance by a trigger T assimilates to the trigger (TUU#→TTT#). However, if a blocker B appears anywhere after a trigger, any potential undergoers do not assimilate to the trigger (TUB#→TUB#). In other words, a property spreads to the edge of a domain or not at all. This is captured by the rule in (1).

(1) True sour grapes: $U \rightarrow T/T(U,T)_0\_\_(U,T)_0\#$

Elgot & Mezei (1965) prove that all regular mappings can be decomposed into one left subsequential and one right subsequential mapping. Subsequential rules are those that have an unbounded number of segments on only one side of the rule's context, as in (2).

(2a) Left subsequential: $X \rightarrow Y/A(B)_0\_\_C$ (2b) Right subsequential: $X \rightarrow Y/A\_\_(B)_0C$

The true sour grapes rule in (1) is regular and can be decomposed into left and right subsequential mappings. The presence or absence of a blocker unboundedly far from a trigger can first be marked on the trigger by a right subsequential mapping, as in (3a-b). This eliminates the need for a later rule to include information about both triggers and blockers unboundedly far from any potential undergoers. For the left subsequential mapping, only information about the trigger (whether it is $T_{\neg B}$ or T_B) is necessary for rule (3c) to determine if assimilation of a undergoer takes place.

(3) Step 1 (Right Subsequential): (a) $T \rightarrow T_B/\_\_(U,T)_0B$ (b) $T \rightarrow T_{\neg B}/\_\_(U,T)_0\#$
Step 2 (Left Subsequential): (c) $U \rightarrow T/T_{\neg B}(U,T_{\neg B})_0\_\_$ (d) $T_{\neg B},T_B \rightarrow T/\_\_$

While true sour grapes is a regular pattern, Heinz & Lai (2013) define the subregular class of **weakly deterministic** mappings as those that can be decomposed into a left subsequential and a right subsequential mapping, such that neither mapping is string-length-increasing, nor adds additional symbols to the language's alphabet (set of symbols). The markup strategy used in (3) to capture true sour grapes is thus not weakly deterministic, as it introduces the symbols T_B and $T_{\neg B}$ to the language's alphabet. From this, Heinz & Lai claim that sour grapes spreading is an unattested phonological pattern because of its computational complexity.

However, we propose that there are attested patterns that resemble true sour grapes, but are crucially different in that they can be represented by a weakly deterministic mapping. For these **false sour grapes** patterns, it is possible to use a markup strategy similar to that in (3) without introducing new symbols to a language's alphabet. Under this approach, information is smuggled into an intermediate representation using **predictable substrings** of the symbols already in a language's alphabet. This strategy is available whenever the first of two subsequential mappings involves neutralization of an input contrast on symbols local to the trigger.

For example, Copperbelt Bemba exhibits a sour-grapes-like pattern of tone spreading (Kula & Bickmore 2015; Jardine 2016). The last high tone in the word spreads unboundedly to the right edge

Charlie O'Hara & Caitlin Smith
University of Southern California

(HLLLL#→HHHHH#), but any other high tone spreads only onto two additional tone bearing units (HLLLH#→HHHLH#). The rules that produce this pattern are provided in (4).

(4a) $L \rightarrow H / H(L)_0 \_\_ (L)_0\#$ (4b) $L \rightarrow H / H(L) \_\_$

The rules in (4) can be decomposed into right and left subsequential mappings, and crucially, these mappings need not introduce new symbols to the alphabet. Instead of marking up the final (triggering) H in the word as $H_\#$ and any nonfinal (non-triggering) H as H_H in the first subsequential mapping, we can mark up these tones using **predictable substrings** of symbols already in the alphabet: HHLL for $H_\#$LLL and HLH for H_HLL. The right subsequential map can also transform all underlying HHLL and HLH substrings, leaving derived intermediate strings HHLL and HLH to uniquely represent triggering and non-triggering high tones. The left subsequential map can then transform these predictable substrings to their surface forms. These mappings are in (5).

(5) Step 1 (right subseq.): (a) $L \rightarrow H / H\_(L)_0\#$ (b) $L \rightarrow H / H\_H$ (c) $L \rightarrow H / HL\_(L)_0H$
Step 2 (left subseq.): (d) $L \rightarrow H / HHLL(L)_0\_$ (e) $L \rightarrow H / HHL\_$ (f) $L \rightarrow H / (L,\#)H\_(H,\#)$

This strategy is successful because every high tone spreads onto at least the two following tone bearing units, neutralizing the contrast between H and L in those positions. There is thus a **zone of predictability** local to the trigger of spreading. This allows the first subsequential mapping to mark up information about blockers that are unboundedly far from the trigger on symbols that are local to the trigger. This markup can carry the same type of information as markups $T_{\neg B}$ and T_B in (3) while using no special symbols outside of a language's alphabet.

There is no markup strategy using only a language's alphabet that captures true sour grapes spreading, in which there is no **zone of predictability** local to either the trigger or blocker. Any markup strategy must distinguish pre-blocker triggers T_B from other triggers $T_{\neg B}$. However, in true sour grapes spreading, there is no way to mark up triggers and undergoers in a way that does not undesirably neutralize contrasts between underlying symbols. For any substring *x* used to mark up pre-blocker triggers $T_{\neg B}$, the underlying string /*x*B/ (the markup substring followed by a blocker) must surface faithfully as [*x*B]. Thus, substring *x* must map to *x* before a blocker, rendering *x* unavailable to uniquely represent $T_{\neg B}$ in the intermediate string. This is illustrated in (6).

(6) Step 1 (Right subsequential): (a) $T(U,T)_{n-1} \rightarrow x / \_\_(U,T)_0\#$
Step 2 (**Not** left subsequential): (b) $U \rightarrow T / x(U,T)_0 \_\_ \mathbf{(U,T)_0}\#$ (c) $x \rightarrow T_n / \_\_(U,T)\#$

We therefore claim that sour-grapes-like patterns of spreading are only attested if they involve **zones of predictability**, rendering their mappings **weakly deterministic**. We refer to Copperbelt Bemba tone spreading and other such patterns as **false sour grapes** spreading. This designation can be extended to other false sour grapes spreading patterns, including Tuyuca nasal harmony (Barnes 1996) and Tutrugbu ATR harmony (Essegbey & McCollum 2017). True sour grapes patterns involving no zones of predictability, on the other hand, are correctly predicted not to be attested.

References

Barnes, J. (1996) Autosegments with Three-Way Lexical Contrasts in Tuyuca. *International Journal of American Linguistics*, 62, 31–58. **Elgot, C., & Mezei, J.** (1965). On relations defined by generalized finite automata. *IBM Journal of Research and Development*, 9, 47–68. **Essegbey, J., & McCollum, A.** (2017). *Initial prominence and progressive harmony in Tutrugbu*. Ms., U. Florida & UCSD. **Heinz, J.** (2011) Computational Phonology – Part I: Foundations. *Language and Linguistics Compass*, 5/4, 140–152. **Heinz, J., & Lai, R.** (2013) Vowel Harmony and Subsequentiality. In *Proceedings of MoL 13* (pp. 52–63). **Jardine, A.** (2016) Computationally, tone is different. *Phonology*, 33, 247–283. **Kula, N., & Bickmore, L.** (2015) Phrasal phonology in Copperbelt Bemba. *Phonology*, 32, 147–176. **Wilson, C.** (2003). *Analyzing unbounded spreading with constraints: marks, targets, and derivations*. Ms., UCLA.