Proceedings of the Society for Computation in Linguistics

Volume 2 Article 41

2019

Formal Characterizations of True and False Sour Grapes

Caitlin Smith
University of Southern California, smithcm@usc.edu

Charlie O'Hara
University of Southern California, charleso@usc.edu

Recommended Citation

Smith, Caitlin and O'Hara, Charlie (2019) "Formal Characterizations of True and False Sour Grapes," *Proceedings of the Society for Computation in Linguistics*: Vol. 2 , Article 41.
DOI: https://doi.org/10.7275/vd79-kt51
Available at: https://scholarworks.umass.edu/scil/vol2/iss1/41

Formal Characterizations of True and False Sour Grapes

Caitlin Smith
University of Southern California
smithcm@usc.edu

Charlie O'Hara
University of Southern California
charleso@usc.edu

1 Introduction

The set of attested phonological input-output mappings is smaller than the set of all logically possible input-output mappings; attested phonological patterns appear to be bounded by computational complexity. The Subregular Hypothesis (Heinz 2011) claims that 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.

2 Background

Elgot & Mezei (1965) prove that all *regular* mappings can be decomposed into one left subsequential and one right subsequential mapping. A *subsequential* mapping can be described by a rule with an unbounded number of segments on at most one side of the rule's context, as in (1).

(1) Subsequential input-output mappings

a. Left subsequential: $X \rightarrow Y / A(B)_0 \_\_ C$
b. Right subsequential: $X \rightarrow Y / A \_\_ (B)_0 C$

Heinz & Lai (2013) define a class of mappings that is less computationally complex than regular mappings. These *weakly deterministic* mappings are those that can be decomposed into left and right subsequential functions that (1) do not change the number of symbols in a string and (2) do not introduce new symbols into a language's alphabet (set of symbols). Due to these restrictions, the first of two functions to apply to a string cannot use special symbols and/or changes in string length to specially mark up that string's intermediate form. This limits the types of input-output mappings that can be captured by such weakly deterministic functions.

3 True Sour Grapes Spreading

Wilson (2003) identifies a pathological pattern of feature spreading known as *sour grapes* spreading. In true sour grapes spreading, 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 phonological property borne by the trigger spreads to the edge of a domain or not at all. This pattern can be described by the rule in (2).

(2) True sour grapes spreading

$$U \rightarrow T / T(U,T)_0 \_\_ (U,T)_0\#$$

The sour grapes rule in (2) 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).

Proceedings of the Society for Computation in Linguistics (SCiL) 2019, pages 338-341.
New York City, New York, January 3-6, 2019

(3) True sour grapes spreading (decomposed)

Step 1, right subsequential:
a. $T \rightarrow T_B / \_\_ (U,T)_0 B$
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. $T_{\neg B}, T_B \rightarrow T / \_\_$

This intermediate markup on the trigger eliminates the need for a later rule such as (3c) to include information about the presence or absence of both triggers and blockers unboundedly far from any potential undergoers. For the left subsequential mapping, only information about the trigger (whether it is a successful trigger $T_{\neg B}$ or unsuccessful trigger T_B) is necessary for the rule in (3c) to determine if assimilation of an undergoer takes place.

While this sour grapes spreading 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. 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; it is regular, but not weakly deterministic. This point is reiterated by Jardine (2016), who argues that unbounded circumambient mappings, including true sour grapes spreading, cannot be decomposed into left and right subsequential mapping such that they fit the definition of weak determinism.

4 False Sour Grapes Spreading

We propose that there are attested spreading 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* spreading patterns, it is possible to use a markup strategy that is similar to the strategy used in (3) but does not introduce 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 (Bantu; Zambia) exhibits a sour-grapes-like pattern of unbounded progressive (rightward) tone spreading (Bickmore & Kula 2013; Kula & Bickmore 2015; Jardine 2016). The last high tone in the word spreads unboundedly to the right edge (HLLLL#⟶ HHHHH#), but any other high tone spreads only onto two additional tone bearing units (HLLLH#⟶HHHLH#). The data in (4) illustrate. (An acute accent indicates a high tone; a grave accent indicates a low tone.)

(4) a. /bá-ka-fik-a/ ⟶ [bá-ká-fík-á]
'they will arrive'
b. /tu-ka-páapaatik-a/ ⟶
[tù-kà-páápáátík-á] 'we will flatten'

c. /bá-ka-pat-a=kó/ ⟶ [bá-ká-pát-à=kó]
'they will hate a bit'
d. /bá-ka-londolol-a=kó/ ⟶
[bá-ká-lóndòlòl-à=kó]
'they will introduce them'

The tone spreading pattern in (4) can be described by the rules in (5).

(5) Copperbelt Bemba high tone spreading

a. $L \longrightarrow H / H(L)_0 \_\_ (L)_0 \#$
b. $L \longrightarrow H / H(L) \_\_$

These rules can be decomposed into right and left subsequential mappings. Crucially, and in contrast with the case of true sour grapes described in section 3, the mappings for Copperbelt Bemba need not introduce new symbols to the alphabet. Instead of marking up the final (successfully triggering) H in the word as $H_\#$ and any nonfinal (unsuccessfully 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 input HHLL and HLH substrings, leaving derived intermediate strings HHLL and HLH to uniquely represent successfully triggering and unsuccessfully triggering high tones. The left subsequential map can then transform these predictable substrings to their surface forms. The rules describing these mappings are provided in (6).

(6) Copperbelt Bemba high tone spreading (decomposed)

Step 1, right subsequential:
a. $L \longrightarrow H/H\_\_(L)_0\#$
b. $L \longrightarrow H/H\_\_H$
c. $L \longrightarrow H/HL\_\_(L)_0H$

Step 2, left subsequential:
d. $L \longrightarrow H/HHLL(L)_0\_\_$
e. $L \longrightarrow H/HHL\_\_$
f. $L \longrightarrow H/(H,LL,\#)H\_\_(H,\#)$

This markup 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 potential trigger of spreading. This allows the first subsequential mapping to mark up information about blockers that may be unboundedly far from the potential trigger on symbols that are local to that 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. As a result, the input-output mapping for Copperbelt Bemba tone spreading can be classified as weakly deterministic.

However, 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 such markup would result in incorrect neutralization of underlying contrasts. A successful markup strategy must distinguish blocked triggers T_B from unblocked triggers $T_{\neg B}$. Assume X is a substring of length n that is made up of symbols in a language's alphabet and is used to mark up unblocked triggers $T_{\neg B}$ (7a). Because in true sour-grapes spreading pre-blocker symbols will all surface faithfully, the underlying string $/X(U)_0B/$ must map to $[X(U)_0B]$. This creates a challenge, because substring X will not appear uniquely as an intermediate markup for unblocked $T_{\neg B}$. The non-uniqueness of substring X prevents a left subsequential function from capturing unbounded spreading, because whether a post-X undergoer U remains faithful (7b) or maps to T (7c) depends on whether a blocker B follows U.

(7) True sour grapes spreading

Step 1, right subsequential:
a. $T(U,T)_{n-1} \longrightarrow X/\_\_(U,T)_0\#$

Step 2, *not* left subsequential:
b. $U \longrightarrow U/X(U,T)_0\_\_(U,T)_0B$
c. $U \longrightarrow T/X(U,T)_0\_\_(U,T)_0\#$

We therefore draw a distinction between cases of attested, weakly deterministic false sour grapes and unattested, non-weakly deterministic true sour grapes. We claim that sour-grapes-like patterns of spreading are only attested if they involve zones of predictability, rendering their mappings weakly deterministic.

5 Conclusion

This paper identifies a distinction in the computational complexity of different types of sour-grapes-like patterns of spreading. We show that while *true sour grapes* spreading is classified as a regular but not weakly deterministic input-output mapping, what we call *false sour grapes* spreading can be classified as weakly deterministic. We introduce the idea that a *zone of predictability*, a predictable substring that occurs local to a potential trigger of spreading, can be utilized in a special markup strategy that lowers the computational complexity of an input-output mapping. As exemplified by Copperbelt Bemba tone spreading, cases of false sour grapes involve the presence of zones of predictability that can be used to distinctly mark up successful and unsuccessful triggers of unbounded spreading in the application of subsequential functions.

In identifying a special markup strategy that relies on a zone of predictability, we propose that the class of weakly deterministic mappings encompasses more patterns of spreading than previously assumed. These include cases of false sour grapes spreading while still excluding unattested true sour grapes spreading. By utilizing this special intermediate markup strategy, we have essentially exploited a loophole in the definition of weak determinism in order to smuggle information into intermediate representations using predictable substrings. In doing so, we provide new insight into how computational complexity interacts with information theoretic notions of predictability.

Acknowledgments

For helpful questions, comments, and discussion we thank Eric Baković, Reed Blaylock, Hayeun Jang, Karen Jesney, Andrew Lamont, Anna Mai, Adam McCollum, Eric Meinhardt, Arthur Santana, Rachel Walker, Yifan Yang, and audiences at SCAMP 2018 and AMP 2018.

References

Bickmore, Lee S., & Kula, Nancy C. (2013) Ternary Spreading and the OCP in Copperbelt Bemba. *Studies in African Linguistics*, 42(2), 101–132.

Elgot, C. C., & Mezei, J. E. (1965) On relations defined by generalized finite automata. *IBM Journal of Research and Development*, 9, 47–68.

Heinz, Jeffrey. (2011) Computational Phonology – Part I: Foundations. *Language and Linguistics Compass*, 5/4, 140–152.

Heinz, Jeffrey, & Lai, Regine. (2013) Vowel Harmony and Subsequentiality. In A. Kornai & M. Kuhlmann (Eds.), *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)* (pp. 52–63). Association for Computational Linguistics.

Jardine, Adam. (2016) Computationally, tone is different. *Phonology*, 33(2), 247–283.

Kula, Nancy C., & Bickmore, Lee S. (2015). Phrasal phonology in Copperbelt Bemba. *Phonology*, 32, 147–176.

Wilson, Colin. (2003) Analyzing unbounded spreading with constraints: marks, targets, and derivations. Unpublished ms., University of California Los Angeles.