



University of Pennsylvania Working Papers in Linguistics

Volume 26
Issue 1 *Proceedings of the 43rd Annual Penn
Linguistics Conference*

Article 15

10-1-2020

The Complexity of Optimizing Over Strictly Local Constraints

Nate Koser

Adam Jardine

Follow this and additional works at: <https://repository.upenn.edu/pwpl>

Recommended Citation

Koser, Nate and Jardine, Adam (2020) "The Complexity of Optimizing Over Strictly Local Constraints,"
University of Pennsylvania Working Papers in Linguistics: Vol. 26 : Iss. 1 , Article 15.
Available at: <https://repository.upenn.edu/pwpl/vol26/iss1/15>

This paper is posted at ScholarlyCommons. <https://repository.upenn.edu/pwpl/vol26/iss1/15>
For more information, please contact repository@pobox.upenn.edu.

The Complexity of Optimizing Over Strictly Local Constraints

Abstract

We show that in Optimality Theory (OT; Prince and Smolensky 1993), optimization over strictly local (SL; McNaughton and Papert 1971) constraints can generate fully regular patterns. We show that a set of stress constraints defined as SL but evaluated in parallel OT predicts an unattested “sour grapes”-type stress assignment pattern, in which iterative foot assignment occurs if and only if it generates a full parse. We show that this pattern is fully regular, thus demonstrating that SL constraints are not closed under optimization. Furthermore, while sour grapes has received attention in harmony (Padgett 1995, Wilson 2003, Wilson 2006, McCarthy 2010) and tone phenomena (Jardine 2016), the possibility of sour grapes-like stress has not previously been discussed.

The complexity of optimizing over strictly local constraints

Nate Koser and Adam Jardine*

1 Introduction

This paper formally investigates the generative capacity of the optimization process in Optimality Theory (OT; Prince and Smolensky 1993) given a very restrictive constraint set. In OT, a set of candidates produced from an input by a GEN function are evaluated against a ranked set of constraints CON, where each constraint is a function assigning violations to candidates. The candidate that performs best with respect to the highest ranked constraints is then selected as the winner, becoming the output for the input.

There are several ways to formally evaluate this process. One, we can ask what the level of complexity of the CON constraints are (Eisner 1997b, Potts and Pullum 2002, Jardine and Heinz 2016). Two, we can ask what the nature of the functions that can be described by OT grammars is (Eisner 1997a, Frank and Satta 1998, Riggle 2004, Buccola 2013). Three, we can examine the outputs of these functions as *phonotactic* patterns—or, more formally, as formal languages. This is particularly true in predictable stress patterns, where the only relevant input information is the length of the word.

In this paper, we study these output patterns given a very specific CON. More specifically, we show that even if we restrict CON to *strictly local* (SL) constraints (McNaughton and Papert 1971, Rogers and Pullum 2011), the optimization process can generate unattested phonotactic patterns beyond this level of complexity. While similar results have been shown for regular relations (Frank and Satta 1998, Gerdemann and Hulden 2012), our work focuses on even more restrictive classes of formal languages.

1.1 The Strictly Local class as a constraint definition language

We can study both phonotactic patterns and constraints in CON through the *sub-regular hierarchy* of formal languages, which has been shown to be relevant to phonotactic patterns and phonology in general (Johnson 1972, Kaplan and Kay 1994, Heinz 2010, Rogers et al. 2013). Three classes are relevant to our work. The first is the *regular* class, which is the set of formal languages that are, equivalently, describable by regular expressions, finite-state automata, or sentences of monadic-second order logic (Kleene 1951, Büchi 1960). The next largest strict superclass in this hierarchy is the *star-free* (SF) class, which is that describable by star-free generalized regular expressions, aperiodic finite-state automata, or first-order sentences with predecessor (McNaughton and Papert 1971). Importantly, studies of natural language stress patterns as formal languages have been shown them overwhelmingly to be SF (Rogers et al. 2013). The SL class, which lies at the bottom of the subregular hierarchy, is describable by *conjunctions of negative literals* (CNLS; Rogers et al. 2013), or a series of statements forbidding substrings. While an extremely restrictive class, the SL languages are highly relevant to phonology, as many phonotactic patterns are SL (Heinz 2010). For example, the stress constraint *CLASH is SL, as it prohibits sequences of adjacent stressed syllables. *CLASH returns a violation for every instance of $\acute{\sigma}\acute{\sigma}$ found in the string, so a string $\acute{\sigma}\acute{\sigma}\sigma$ would incur one violation, while a string $\acute{\sigma}\sigma\sigma$ would incur none.

The SL class is highly relevant to studying the nature of CON. In OT, *markedness* constraints evaluate the ill-formedness or improper arrangement of phonological elements in an output candidate. The content of markedness constraints is not arbitrary. In fact, they are overwhelmingly negative, meaning that they ban certain structures (McCarthy and Prince 1993, Jardine and Heinz 2016). Examples of negative constraints include *CODA, which is violated when the coda position of a syllable is occupied, and the aforementioned *CLASH. These constraints do not require

*The authors would like to thank the helpful comments and suggestions of Bruce Tesar, the members of the PhonX reading group at Rutgers, and audiences at PhoNE and PLC.

structures; instead, they assign violations if those structures are absent. We can formalize the generalization that markedness constraints should be negative with a *Constraint Definition Language* (CDL) (Eisner 1997b, Potts and Pullum 2002, de Lacy 2011) for markedness constraints. A CDL is a language for defining and interpreting the set of constraints in CON. Thus, a CDL limits CON to constraints that can be written with that CDL—constraints beyond the expressive power of the selected constraint definition language will not appear in CON. For example, the original definition of ALIGN-type constraints in McCarthy (2003) uses first order logic. But if the selected CDL is less powerful than first order, constraints such as ALIGN cannot be written. Thus, a CDL makes testable empirical and typological predictions about the nature of CON. de Lacy states that: “An explicit CDL is both useful and ultimately essential to a complete Optimality Theory.” (p.1494). As will be shown in §2, the constraints used here are familiar stress constraints that have been proposed in the literature and behave in the same way but are defined with a CDL that is SL.

1.2 Overview of results

However, we show that even when a CDL reduces CON to SL constraints, OT can produce patterns that are properly regular—that is regular, but not SF. To the best of our knowledge, these patterns are unattested. This means that, even if we have a specific CDL based on computational complexity, optimization can generate patterns beyond this level of complexity. In this specific case, it also means that SL constraints can generate pathological patterns beyond the complexity of most attested stress patterns.

Furthermore, the particular pattern we generate is a novel example of a “sour grapes” (Padgett 1995, Wilson 2003, 2006) pattern. Sour grapes patterns are those in which an iterative process applies if and only if it can apply to the entire word—if it is somehow blocked, then it does not apply at all (thus “sour grapes”). These patterns are thought to be unattested, yet have been shown to be generated under some OT theories of spreading. Sour grapes patterns have received attention in harmony (Padgett 1995, Wilson 2003, 2006, McCarthy 2010) and tone phenomena (Jardine 2016, Pater 2018), but the possibility of sour grapes-like stress has not previously been discussed in the literature. Using SL constraints, we derive a pattern in which iterative foot building occurs if and only if a full parse is created—otherwise only a single foot is constructed. Not only is this not SF, but it shows that sour grapes-style application is pervasive in OT grammars, even though it is largely unattested in natural language phonology.

The paper is structured as follows: §2 introduces the constraint set that generates the sour grapes pattern. §3 explains the sour grapes pattern in detail, showing how it is generated in parallel OT. §4 gives evidence of the non-SL nature of the pattern with a proof showing that it is both not SF and describable with an FSA. §5 discusses the implications of this work and §6 concludes.

2 Constraints

Here, we define a set of constraints for stress patterns given a SL CDL. The idea of a CDL for CON is to have some statement about the level of complexity allowed in its constraints. Having a restriction on the complexity of CON of a certain level of logical complexity is a statement that the phonology has access to that level of logical power. Assuming that CNLs are the building blocks of markedness constraints gives us a principled theory of (a part of) CON, and is a strong prediction that markedness requirements are local only.

Stress assignment has received much attention in the OT literature. Constraints proposed for analyses of stress vary widely in their logical complexity. Definitions for ALIGN (McCarthy and Prince 1993) constraints, for example, which align prosodic categories such as feet and prosodic words (Nespor and Vogel 1986), employ some kind of first order logic, as they employ quantification over categories. The ALIGN-type constraints are defined as follows (McCarthy 2003):

ALIGN(Cat1, Edge1; Cat2, Edge2; Cat3):

\forall Cat1 \exists Cat2 such that Edge1 of Cat1 and Edge2 of Cat2 coincide. Assign one violation mark \forall Cat3 that intervenes between Edge1 of Cat1 and the nearest Edge2 of some Cat2

The definition of the constraint makes explicit reference to first order logic quantifiers \forall and \exists .

Other stress constraints are simpler, such as LAPSE (Selkirk 1984)—a ban on adjacent unstressed syllables. This is SL, as it simply forbids the sequence $\sigma\sigma$, where “ σ ” represents an unstressed syllable.

Here, we restrict our stress constraints to SL complexity by defining them with CNLs. We define the syntax of CNLs as follows. For an alphabet Σ of symbols we denote by Σ^* the set of all (potentially empty) strings over Σ and for any strings u and w we denote by uw their concatenation. A string $u \in \Sigma^*$ is a *substring* of another string $w \in \Sigma^*$ if $w = w_1uw_2$ for any $w_1, w_2 \in \Sigma^*$.

Definition 1 (Syntax of CNLs) For an alphabet Σ , A conjunction of negative literals (CNL) is a sequence

$$\neg w_1 \wedge \neg w_2 \wedge \dots \wedge \neg w_n$$

where each w_i is a string in Σ^* .

For example, if $\Sigma = \{\acute{\sigma}, \sigma\}$, then $\neg\acute{\sigma}\acute{\sigma}$ is a CNL.

Definition 2 (Semantics of CNLs) For an alphabet Σ and a CNL $C = \neg w_1 \wedge \neg w_2 \wedge \dots \wedge \neg w_n$, we say a string u satisfies C iff no w_i is a substring of u . We say u violates C if it does not satisfy C .

To continue with the example $\neg\acute{\sigma}\acute{\sigma}$, the string $\sigma\sigma\acute{\sigma}$ satisfies $\neg\acute{\sigma}\acute{\sigma}$ because it does not include $\acute{\sigma}\acute{\sigma}$ as a substring, but the string $\sigma\acute{\sigma}\acute{\sigma}$ violates it. Interpreting $\acute{\sigma}$ as stressed syllables, $\acute{\sigma}\acute{\sigma}$ is thus the constraint CLASH.

We refer the reader to Rogers et al. (2013) on the fact that CNLs describe exactly the SL class of regular languages. While the definition of the semantics of CNLs is boolean—they are either satisfied or not—it is straightforward to extend their interpretation to OT markedness constraints, which keep track of the *number* of violations. In this case, we can count the violations of u for a CNL constraint C by the number of times some w_i in C appears as a substring in u .

Turning to the set of constraints at hand, we adopt a simplified metrical structure including syllables and feet. An example is given in Fig. 1.

$$(\acute{\sigma}\sigma)(\acute{\sigma}\sigma)\sigma$$

Figure 1: Example of metrical structure

Syllables can be stressed ($\acute{\sigma}$) or unstressed (σ); we do not distinguish primary and secondary stress. Syllables are parsed into *feet*. Parentheses denote the boundaries of feet—“(” is a left foot boundary and “)” is a right foot boundary. Two *binary* feet are shown in Fig. 1. The next level of structure is the prosodic word; we assume each string is equal to a prosodic word. (For more on metrical structure and the prosodic hierarchy, see Nespor and Vogel (1986) and Selkirk (1984).) We assume culminativity—GEN will not produce candidates with no feet or no stresses. GEN will also not produce candidates with ill-formed feet (e.g., a sequence like “() σ ”) or feet larger than two syllables (e.g. ternary feet).

We thus define our constraints as CNLs over the alphabet $\Sigma = \{(\cdot), \sigma, \acute{\sigma}\}$, reflecting the metrical structure in Fig. 1. Each constraint is either a well-known stress constraint or has a known analogue in the stress literature. The constraint set is given in Table 1.

The constraint FTBIN is a ban on unary feet. FTBIN has been part of stress analyses in OT since McCarthy and Prince (1993). In theories of stress that refer to feet, some rankable constraint referring to degenerate feet is necessary to generate both languages that only employ binary feet and languages that create unary feet in order to fully parse all syllables.

The constraints IAMB and TROCHEE are constraints on the form of feet. Whether a language uses iambs or trochees will partly determine where stress occurs in words of that language. Also of note is that IAMB and TROCHEE are *foot antagonists*. The two constraints, working together, can limit creation of feet in a word to one with the proper ranking. This property of the foot form constraints plays a key role in generating the pathological sour grapes pattern discussed here, but is also important in analyses of attested single-stress systems, where there is only one foot in a word.

Constraint	Definition	Description
FTBIN	$\neg (\acute{\sigma})$	violated by unary feet
IAMB	$\neg (\acute{\sigma}\sigma) \wedge (\acute{\sigma})$	violated by trochees and unary feet
TROCHEE	$\neg (\sigma\acute{\sigma}) \wedge (\acute{\sigma})$	violated by iambs and unary feet
* σF	$\neg \sigma(\sigma \wedge \neg \sigma(\acute{\sigma}))$	violated by nonfinal unparsed syllables
* $F\sigma$	$\neg (\sigma)\sigma \wedge \neg \acute{\sigma})\sigma$	violated by noninitial unparsed syllables

Table 1: Constraint set

Theories of stress also need some way to align feet with prosodic word edges and with each other. When feet iterate throughout a word, the iteration is typically directional—it proceeds right-to-left or left-to-right—with the initial foot anchored to the edge from which iteration occurs. In OT, this is typically done with ALIGN (McCarthy and Prince 1993) constraints. Here, however, ALIGN is not available, as it employs first order logic and the constraint definition language in this theory is limited to CNLs. In order to achieve alignment of feet, the constraints * σF and * $F\sigma$ are introduced. These constraints are similar to the constraints * $F\acute{\sigma}$ and * $F\sigma$ discussed in McCarthy (2003), but are defined as CNLs here. By militating against unparsed-syllable gaps, * σF and * $F\sigma$ can motivate placement of feet. The OT tableau in Fig. 2 shows how this occurs.

	/ $\sigma\sigma\sigma\sigma\sigma\sigma$ /	* σF	* $F\sigma$
☞ a.	($\sigma\sigma$)($\sigma\sigma$)($\sigma\sigma$) σ		*
b.	σ ($\sigma\sigma$)($\sigma\sigma$)($\sigma\sigma$)	*!	
c.	($\sigma\sigma$) σ ($\sigma\sigma$)($\sigma\sigma$)	*!	*
d.	($\sigma\sigma$)($\sigma\sigma$) σ ($\sigma\sigma$)	*!	*

Figure 2: Tableau for * σF and * $F\sigma$

* σF is violated when a footed syllable is the successor to an unparsed syllable. * $F\sigma$ is the opposite case—violation occurs when a footed syllable is immediately followed by a syllable that is unfooted. Both constraints have two conjuncts—one for when the footed syllable is stressed, and one for when it is unstressed. In standard theories of metrical structure, an unfooted syllable cannot be stressed, and so conjuncts addressing such cases are not part of the definition of the constraint. Note also that these constraints are not sensitive to a left or right edge - candidates (d) and (e) suffer equal violations of both * σF and * $F\sigma$ for breaking a string of feet up with an unparsed syllable. The two candidates differ in where the unparsed syllable goes, but their violation profiles are identical. Though it will be seen that these two constraints are crucial in generating the sour grapes pattern, any theory of stress in OT must have some constraint in CON that can place feet as in Fig. 2.

Analysis in OTWorkplace (Prince et al. 2007-2017) reveals a typology of ten unique languages for these five constraints. Present in the typology are four attested single-stress patterns, four sour grapes patterns, and two patterns that display ambiguity (more than one optimal output). The four attested patterns (initial, penultimate, and final stress) account for 68% of the languages present in Gordon (2002)'s typology of quantity insensitive stress. That such a level of typological coverage is achieved suggests that these constraints are a subset of a reasonable CON for analyses of stress. As an example, the comparative OT tableau (Prince 2002) in Fig. 3 shows how an initial stress pattern—attested in Nenets (Decsy 1966)—arises from the interaction of the constraints.

input	winner	loser	*σF	FTBIN	TROCH	IAMB	*Fσ
3syll	(σσ)σ	σ(σσ)	W			L	
2syll	(σσ)	(σσ)			W		L
4syll	(σσ)σσ	(σσ)(σσ)				W	L

Figure 3: Tableau for initial stress pattern

The ranking of *σF and TROCH in the highest stratum ensures the left-edge orientation and trochaic shape of feet in the language. In the next lowest stratum, IAMB exerts foot antagonistic influence, preventing the creation of additional trochees beyond the first. These factors combine to limit the language to a single, fixed stress on the initial syllable. This is the attested initial stress pattern of Nenets. The next sections focus on the capacity of the constraint set to generate the pathological sour grapes stress pattern.

3 Sour grapes stress patterns

This section examines one of the four “sour grapes” languages in the typology. Sour grapes phenomena have been identified in the realm of harmony and spreading (Padgett 1995, Wilson 2003, 2006, McCarthy 2010) and tonal phenomena (Jardine 2016). In sour grapes spreading, if some feature can not spread over the entire word, then the candidate with no spreading at all is selected as optimal instead. This involves the interaction of a markedness and faithfulness constraint—the markedness constraint (AGREE) requiring agreement on some feature in adjacent segments, and the faithfulness constraint IDENTIO(F) penalizing input-output changes in the relevant feature.

Here, however, there is a sour grapes stress pattern that arises from a constraint set containing *only* markedness constraints limited to the complexity of CNLs—no faithfulness constraints are present. As we will show, this pattern is not only beyond the complexity of CNLs (that is, it is not SL), it is not even SF, although it is regular. This illustrates that the process of optimization over candidates in OT can generate patterns of much higher logical complexity than the constraints themselves. The sour grapes pattern is given here:

(σσ)
 (σσ)σ
 (σσ)(σσ)
 (σσ)σσσ
 (σσ)(σσ)(σσ)
 (σσ)σσσσσ
 ...

Figure 4: The sour grapes stress pattern

The language starts with a normal right-to-left trochaic parse up to the four-syllable form, but then in the five syllable form the pattern breaks—only one foot is created. The six-syllable form returns to a full parse of syllables to feet, but the seven-syllable form again only builds one foot. This is the “sour grapes” pattern—the language wants to build binary feet all the way to the end of the word. If this cannot be done, as in odd-numbered-syllable forms, then the bare minimum is done instead—only one foot is created.¹ No further feet are “spread” to the right.

This pattern is pathological. If a language assigns stress by iterating in one direction, the pattern should not vary drastically between words of even-syllable length and words of odd-syllable length. As far as we are able to determine, no such stress pattern exists. However, the tableau in Fig. 5

¹In other sour grapes phenomena, the “bare minimum” is not done—no spreading occurs *at all* in nasal-spreading sour grapes processes with blocking, for instance. In theories of stress that assume culminativity, however, this is not an option—at least one stress, and therefore one foot, must be present.

shows how this pattern emerges from a ranking of the five constraints given in Table 1. A Hasse diagram is given in Fig. 6.

input	winner	loser	* σF	FTBIN	TROCH	* $F\sigma$	IAMB
3syll	($\acute{\sigma}\sigma$) σ	$\sigma(\acute{\sigma}\sigma)$	W			L	
3syll	($\acute{\sigma}\sigma$) σ	($\acute{\sigma}$)($\sigma\acute{\sigma}$)		W	W	L	
2syll	($\acute{\sigma}\sigma$)	($\sigma\acute{\sigma}$)			W		L
4syll	($\acute{\sigma}\sigma$)($\acute{\sigma}\sigma$)	($\acute{\sigma}\sigma$) $\sigma\sigma$				W	L
5syll	($\acute{\sigma}\sigma$) $\sigma\sigma\sigma$	($\acute{\sigma}\sigma$)($\acute{\sigma}\sigma$) σ					W

Figure 5: Tableau for sour grapes pattern

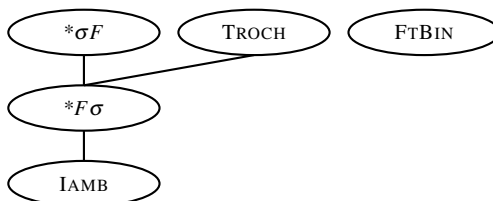


Figure 6: Hasse diagram for sour grapes pattern

Having $*\sigma F$ in the top stratum gives the language its left-to-right orientation. FTBIN prevents the creation of any unary feet—only binary feet can be built. The four-syllable winner-loser pair is revealing—a binary foot is added, as it fully parses all syllables and satisfies $*F\sigma$. For this sour grapes pattern, $*F\sigma$ is the spreading constraint. But instead of spreading a feature or a tone, it looks to “spread” the parse by bringing all syllables into feet. However, in a longer odd-parity form, adding binary feet cannot parse all syllables. This is seen in the five syllable winner-loser pair. If maximal creation of binary feet does not better satisfy $*F\sigma$, then foot-antagonist IAMB prevents the creation of additional trochees. Spreading of the parse is blocked.

An important part of what allows this pattern to appear in the typology is the presence of the pseudo-alignment constraints $*F\sigma$ and $*\sigma F$. These constraints encourage the spread of feet because they are most satisfied when every syllable is parsed to a foot. While this is important for capturing attested patterns that exhibit an exhaustive parse, their role in generating sour grapes patterns raises questions. $*F\sigma$ and $*\sigma F$ were posited and defined as CNLs in the hopes of avoiding issues that have been identified with the more complex ALIGN-schematic constraints (McCarthy 2003, Kager 2001, 2005, Buckley 2009, Hyde 2012). But if these CNL constraints, when fed through the optimization process in OT, can generate patterns of much higher complexity, then it is possible that no CDL for constraints is “simple enough” to avoid prediction of pathological patterns.

This sour grapes pattern is unique for two reasons. First, it is derived by the interaction of markedness constraints only, where previously identified sour grapes patterns occur as the result of a markedness-faithfulness interaction. However, as metrical feet are not considered part of the input in OT, there is no input-output relation to be faithful to. The pattern arises solely from output restrictions. Secondly, that the pattern is derived from a set of SL constraints indicates that SL patterns are not closed under optimization.

4 Formal Characterization

4.1 Sour grapes stress is regular

Regular languages are those that can be described by a finite state automaton (FSA; see, e.g. Hopcroft et al. 2006). An FSA corresponding to the sour grapes-like pattern is given in Fig. 7.

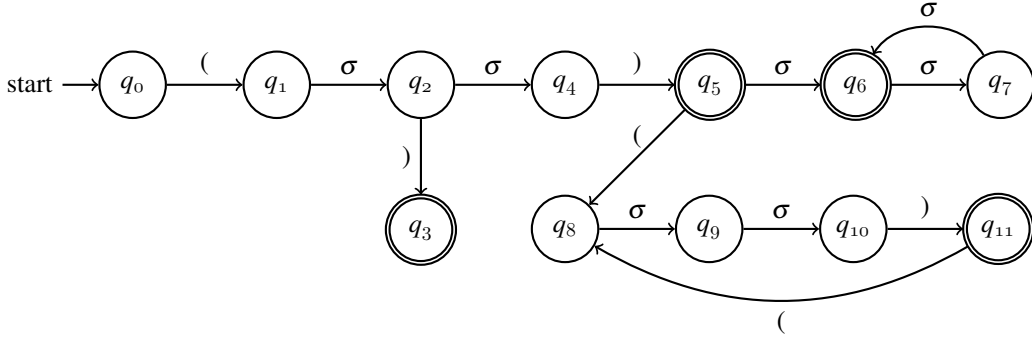


Figure 7: FSA for sour grapes pattern

The initial state is marked with an arrow labeled “start,” and final states are marked with double circles. In the FSA below and in the following discussion, we use the alphabet $\Sigma = \{ (,), \sigma \}$, which conflates the distinction between stressed and unstressed syllables. While the stressed-unstressed distinction is important in generating the specific pattern described above, it is not necessary for studying the general properties of the pattern, and will simplify the discussion. Additionally, the FSA and proof given below will hold for any left-to-right sour grapes-type pattern, regardless of whether the specific language employs trochees or iambs. As an example, Fig. 8 shows how the FSA accepts the well-formed strings $(\sigma\sigma)(\sigma\sigma)$ and $(\sigma\sigma)\sigma\sigma\sigma$ and rejects the ill-formed string $*(\sigma\sigma)\sigma\sigma$; accepting states are highlighted in boxes for clarity.

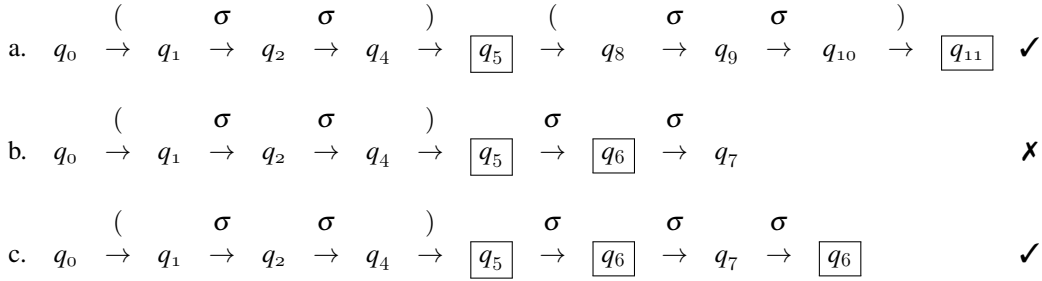


Figure 8: Processing of four-syllable (above) and five-syllable (below) strings by sour-grapes FSA

From the initial state, only strings that begin with a left foot boundary will be read. The first accepting state is q_3 . This accounts for monosyllabic forms in the language. Reading a single binary foot leads to q_5 , which accepts disyllabic forms. As all strings in the language other than the monosyllable begin with a binary foot, all derivations in Fig. 8 arrive in state q_5 .

If the string is longer, there are two paths. In an odd-parity form such as 8c, all remaining symbols in the string will be syllables only. Reading of the third syllable leads to accepting state q_6 . From here, if another syllable is read state q_7 is reached. q_7 is not accepting, as being in this state means an even number of unparsed syllables has been read. After reading another syllable, the FSA loops back to q_6 , where an odd-number unparsed syllable string can be accepted again. The derivation for 8c is complete. Notice that once in state q_6 , there are no transitions that take a foot boundary as input—only odd-length forms consistent with the sour grapes pattern will be accepted by the machine. In 8b for example, an even number of unparsed syllables after the initial binary foot lands on a non-accepting state, and so the string is rejected by the FSA.

From q_5 , if a left foot boundary is read instead as in 8a, the lower branch beginning with q_8 is reached. This branch, ending in accepting state q_{11} , reads additional binary feet and leaves the option for further binary feet open via the edge between q_{11} and q_8 . Once this lower branch is reached, only full binary feet are accepted. No other string will land on an accepting state. As such, this portion of the FSA accounts for even-number-syllable forms. The FSA accepts all and only the

strings of a left-to-right sour grapes stress pattern.

This FSA thus describes the following set of strings:

$$L = \{(\sigma\sigma), (\sigma\sigma)\sigma, \quad (1)$$

$$(\sigma\sigma)(\sigma\sigma), (\sigma\sigma)\sigma\sigma\sigma,$$

$$(\sigma\sigma)(\sigma\sigma)(\sigma\sigma), (\sigma\sigma)\sigma\sigma\sigma\sigma, \dots\}$$

We now show that this L is not SF.

4.2 Sour grapes stress is not star free

In order to prove that the sour grapes pattern presented above is not SF, it is necessary to reference the following theorem:

Theorem 1 (Rogers and Pullum 2011) *A language L is Star-Free iff it is non-counting, that is, iff there exists some $n > 0$ such that for all strings u, v, w over Σ , if $uv^n w$ occurs in L then $uv^{n+i} w$, for all $i \geq 1$, occurs in L as well.*

Since this principle must hold for **all** $i \geq 1$ at some (any) $n > 0$, proving that sour grapes is not SF is a matter of finding two classes of counter-examples to this theorem—one for any odd n and one for any even n . Doing so shows that substituting any even or odd number for n (so, any integer) will also fail to meet the requirements of the theorem. This will prove that the sour grapes-like pattern seen here is fully regular, and that SL patterns are not closed under optimization.

We note an important property of L in Ex. 1. No string $(\sigma\sigma)\sigma^n$ for an even value of n will appear in the language—even-syllable forms always parse all syllables to feet. Using $(\sigma\sigma)\sigma^n$ with even n as the target for the $uv^{n+i} w$ part of Theorem 1, the following proof can be formulated:

Theorem 2: The pattern of L is not SF.

Proof: Let $uv^1 w$ be the string $(\sigma\sigma)\sigma\sigma\sigma \in L$ such that $u = (\sigma\sigma)$, $v^1 = \sigma$, and $w = \sigma\sigma$. Set i to 1. For any odd value of n , $|v^n w|$ will be an odd-number string of syllables, and $|v^{n+1} w|$ will be an even-number string of syllables, meaning the string $uv^{n+1} w$ will also be even. For example, if $n = 1$, $u = (\sigma\sigma)$, $v^2 = \sigma\sigma$, and $w = \sigma\sigma$. The string $uv^2 w$ over Σ is $(\sigma\sigma)\sigma\sigma\sigma\sigma$ and $(\sigma\sigma)\sigma\sigma\sigma\sigma \notin L$. Thus L fails Thm. 1 for any odd n .

Let $uv^2 w$ be the string $(\sigma\sigma)\sigma\sigma\sigma \in L$ such that $u = (\sigma\sigma)$, $v^2 = \sigma\sigma$, and $w = \sigma$. Set i to 1. For any even value of n , $|v^n w|$ will be an even-number string of syllables, and $|v^{n+1} w|$ will be an odd-number string of syllables, meaning the string $uv^{n+1} w$ will also be odd. For example, if $n = 2$, $u = (\sigma\sigma)$, $v^3 = \sigma\sigma\sigma$, and $w = \sigma$. The string $uv^3 w$ over Σ is $(\sigma\sigma)\sigma\sigma\sigma\sigma$ and $(\sigma\sigma)\sigma\sigma\sigma\sigma \notin L$. Thus L also fails Thm. 1 for any even n .

This demonstrates that for all $i \geq 1$, it is not the case that there is an odd n or even n such that if $uv^n w$ is a string of L then $uv^{n+i} w$ is a string of L for all $i \geq 1$, in contradiction of Thm. 1. This proves that Thm. 1 does not hold for the sour grapes-style pattern. It proves that this pattern is not SF and—taken together with the FSA provided in Fig. 7—proves that it is properly regular.

5 Discussion

Theorem 2 thus shows that SL constraints are not closed under optimization in classical, parallel OT. It is important to situate this general result within the constellation of other formal language work done on optimization.

Frank and Satta (1998:§5) show that constraints defined as regular relations without a bound on their constraint violations can produce non-regular relations. Similarly, Gerdemann and Hulden (2012:§5) show that an OT grammar with a set of four basic constraints is beyond description by regular relations.

This is thematically similar to the result presented here. But while Frank and Satta (1998)'s line of investigation has to do with relations, here the focus is formal languages. We show that optimization in OT can take an SL language and return a fully regular language. That this result for SL languages has an analogue in regular relations suggests that the lack of complexity class closure may be an unavoidable reality of optimization in classical OT.

This is a particularly pressing problem for stress analyses. Studying the stress pattern database of Heinz (2007), Rogers et al. (2013) show that all but one pattern in the database are sub-SF. The one counterexample they cite, Cairene Arabic (Graf (2010) argues that the same for Creek) is controversial: whether or not it is regular depends on the presence of iterative secondary stress. This is not the case for the sour grapes pattern discussed here: it is the very nature of the iteration that it is not regular. Thus, SL constraints under optimization generate an unattested pattern that is clearly more complex than most stress patterns.

Finally, the previously unreported ability of OT to generate a sour grapes pathology with respect to stress, and without faithfulness constraints, suggests that sour grapes-style application is a very general issue for OT grammars. This paper suggests a line of future work studying the general conditions under which OT constraints generate sour grapes patterns. This is particularly of interest given the prominence of sour grapes patterns in recent theoretical literature.

6 Conclusion

This paper has shown that SL patterns are not closed under optimization. Limiting the constraint definition language for markedness constraints to conjunctions of negative literals with the successor ordering relation, a set of five CNL constraints implemented for analyses of stress generate a “sour grapes”-like pattern that is properly regular. A formal language theory approach reveals this previously unknown property of the optimization process in OT.

The line of inquiry here has revealed an interesting property of optimization in Optimality Theory, and suggests that more profitable research in this domain remains to be done. It also calls efforts to limit the CDL to some relatively low level of logical power into question. If optimization over candidates with SL-only constraints can generate a factorial typology with pathological regular patterns, then it is unclear what stands to be gained by proposing a CDL that is maximally simple.

References

- Buccola, Brian. 2013. On the expressivity of Optimality Theory versus ordered rewrite rules. In *Proceedings of Formal Grammar 2012 and 2013*, volume 8036 of *Lecture Notes in Computer Science*, 142–158. Berlin: Springer-Verlag.
- Büchi, J. Richard. 1960. Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 6:66–92.
- Buckley, Eugene. 2009. Locality in metrical typology. *Phonology* 26:389–435.
- Decsy, Gyula. 1966. Yurak chrestomathy. *Uralic and Altaic Series* 50.
- Eisner, Jason. 1997a. Efficient generation in primitive Optimality Theory. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, 313–320. Madrid.
- Eisner, Jason. 1997b. What constraints should OT allow? Talk handout, Linguistic Society of America, Chicago. ROA#204-0797. Available at <http://roa.rutgers.edu/>.
- Frank, Robert, and Giorgio Satta. 1998. Optimality theory and the generative capacity complexity of constraint violability. *Computational linguistics* 24:277–299.
- Gerdemann, Dale, and Mans Hulden. 2012. Practical finite state optimality theory. In *Proceedings of the 10th International Workshop on FSMNLP*, 10–19. ACL.
- Gordon, Matthew. 2002. A factorial typology of quantity-insensitive stress. *Natural Language & Linguistic Theory* 20:491–552.
- Graf, Thomas. 2010. Logics of phonological reasoning. Master's thesis, UCLA.
- Heinz, Jeffrey. 2007. The Inductive Learning of Phonotactic Patterns. Doctoral dissertation, UCLA.
- Heinz, Jeffrey. 2010. Learning long-distance phonotactics. *LI* 41:623–661.

- Hopcroft, John, Rajeev Motwani, and Jeffrey Ullman. 2006. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, third edition.
- Hyde, Brett. 2012. Alignment constraints. *NLLT* 30:789–836.
- Jardine, Adam. 2016. Computationally, tone is different. *Phonology* 33(2):385–405.
- Jardine, Adam, and Jeffrey Heinz. 2016. Markedness constraints are negative: an autosegmental constraint definition language. *Proceedings of the 51st CLS*.
- Johnson, Douglas. 1972. Formal aspects of phonological description.
- Kager, Rene. 2001. Rhythmic directionality by positional licensing. *HILP-5 ROA*-514.
- Kager, Rene. 2005. Rhythmic licensing: an extended typology. In *Proceedings of the 3rd International Conference on Phonology*. Seoul: The Phonology Morphology Circle of Korea. 5?31.
- Kaplan, Ronald, and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20:331–378.
- Kleene, Stephen. 1951. Representation of events in nerve nets and finite automata. Project RAND Research Memorandum RM-704.
- de Lacy, Paul. 2011. Markedness and faithfulness constraints. In *The Blackwell Companion to Phonology*. Blackwell.
- McCarthy, John. 2010. Autosegmental spreading in optimality theory. In *Tones and features*, 195–222. Berlin & Boston: De Gruyter Mouton.
- McCarthy, John J. 2003. OT constraints are categorical. *Phonology* 20:75 – 138.
- McCarthy, John J., and Alan Prince. 1993. Generalized alignment. *Yearbook of Morphology 1993* 79–153.
- McNaughton, R., and S. Papert. 1971. Counter-free automata.
- Nespor, M., and I. Vogel. 1986. *Prosodic phonology*. Dordrecht: Foris.
- Padgett, Jaye. 1995. Partial class behavior and nasal place assimilation. In *Proceedings of the 1995 Southwestern Workshop on Optimality Theory (SWOT)*, 145–183. Tuscon: Department of Linguistics, University of Arizona.
- Pater, Joe. 2018. Substance matters: a reply to Jardine (2016). *Phonology* 35:151–156.
- Potts, Christopher, and Geoffrey K. Pullum. 2002. Model theory and the content of OT constraints. *Phonology* 19:361–393.
- Prince, Alan. 2002. Entailed ranking arguments.
- Prince, Alan, and Paul Smolensky. 1993. Optimality theory: constraint interaction in generative grammar. *Ms, Rutgers University*.
- Prince, Alan, Bruce Tesar, and Nazzare Merchant. 2007-2017. *OTWorkplace*. <https://sites.google.com/site/otworkplace/>.
- Riggle, Jason. 2004. Generation, Recognition, and Learning in Finite State Optimality Theory. Doctoral dissertation, UCLA.
- Rogers, James, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. 2013. Cognitive and sub-regular complexity. In *Formal Grammar*, volume 8036 of *Lecture Notes in Computer Science*, 90–108. Springer.
- Rogers, James, and Geoffrey Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language, and Information* 20(3):329–342.
- Selkirk, Elisabeth. 1984. *Phonology and Syntax: The relation between sound and structure*. Cambridge: MIT Press.
- Wilson, Colin. 2003. Analyzing unbounded spreading with constraints: marks, targets, and derivations.
- Wilson, Colin. 2006. Unbounded spreading is myopic.

Department of Linguistics
 18 Seminary Place 08901
 New Brunswick, NJ
nate.koser@rutgers.edu
adam.jardine@rutgers.edu