On the Expressivity of Optimality Theory versus Ordered Rewrite Rules

Brian Buccola

Department of Linguistics, McGill University
brian.buccola@mail.mcgill.ca

Abstract. I prove that there are phonological patterns which are expressible by ordered rewrite rules but not by any Optimality Theoretic (OT) grammar whose constraint set contains only markedness constraints and *single–state* faithfulness constraints, i.e. faithfulness constraints that assign violation marks to pairs of single input–output segments in correspondence, with no reference to other segments in the input or output. The intention is to capture formally the widespread intuition that certain *opaque* patterns, which are expressible by ordered rewrite rules, are problematic for *classic*, or traditional, OT.

1   Introduction

Generative phonology takes the phonetic form of an utterance to be the *surface* realization (or output) of an abstract, *underlying* phonological form (or input). Natural languages therefore exhibit input–output patterns, which the phonologist is tasked with describing in a formal, algorithmic way. Phonologists have largely described these patterns using two types of grammar: (i) *serial rule grammars*, e.g. as laid out in *The Sound Patterns of English* (SPE) [1], in which an input is mapped serially to intermediate outputs and finally to a terminal output via ordered, context–sensitive rewrite rules; and (ii) *parallel constraint grammars*, e.g. as in Optimality Theory (OT) [2], in which an input is mapped directly to that output which is optimal with respect to a set of ranked constraints, with no intermediate mappings (i.e. everything happens in parallel).

This paper addresses the following two questions: (i) Are there input–output patterns that can be expressed by one type of grammar but not by the other, and (ii) if so, are those patterns attested in natural language? The first is a formal question about the classes of input–output relations expressible by the two formalisms. The second is an empirical question that bears on whether phonologists should favor one type of grammar over another on the basis of empirical coverage.

Under the assumption that a rule may not rewrite within that part of a string which it has already rewritten,[1] the patterns expressible by ordered rewrite rules correspond exactly to the *regular relations* [3–5].[2] Without any restrictions, the patterns expressible by OT grammars include non–regular relations, i.e. patterns that are inexpressible by ordered rewrite rules [6, 7].[3]

What is not known is whether every regular relation is expressible by some OT grammar [10]. Put differently, the question is still open whether every pattern expressible by ordered rewrite rules is expressible by some OT grammar.

Nevertheless, there is a widespread intuition that certain *opaque* phonological patterns, which are expressible by ordered rewrite rules (hence, they are regular), are problematic for *classic* [11], or traditional [10], OT grammars, consisting of just two basic types of constraints (output markedness and input–output faithfulness). However, to my knowledge, no one has yet formally proved this claim. Such a proof could reach two possible conclusions: (i) there are regular relations that are completely inexpressible by classic OT grammars, or (ii) all regular relations are expressible in principle by classic OT, albeit sometimes only with ad hoc, linguistically unmotivated, but formally sound constraints.

In this paper I define classic OT grammars as OT grammars that contain only markedness and *single–state* faithfulness constraints, i.e. faithfulness constraints that assign violation marks to pairs of single input–output segments in correspondence, with no reference to other segments in the input or output. These constraints include many, if not most, of the standard constraints proposed in the OT phonology literature, e.g. IDENT, MAX, and DEP. I then prove that there are regular relations that cannot be expressed by any such OT grammar.

Section 2 reviews SPE–style and OT–style grammars, as well as the notion of opacity. In section 3, I prove, using data from Canadian English, that there are input–output patterns which are expressible by ordered rewrite rules but not by any classic OT grammar, as defined here. In section 4, I discuss several other cases of opacity that are likewise provably expressible by ordered rewrite rules but not by classic OT grammars; and I discuss cases of opacity that seem, in principle, to be expressible by classic OT grammars. Section 5 concludes.

---

[1] Nowhere in this paper is this assumption violated. Henceforth, when I write "ordered rewrite rules," I mean rule–based grammars in which this assumption is in place.

[2] The definition of a regular relation is not important for this paper; for details, see [4].

[3] Under certain assumptions, however, the patterns expressible by OT grammars lie within the regular relations [8–10]. These assumptions are (i) that both the constraints and the function GEN mapping inputs to sets of output candidates are regular, and (ii) that there is an upper bound on the number of violation marks assigned by any constraint. Without *either* assumption, OT can describe non–regular relations. I thank an anonymous referee for helping me to clarify these points.

2   Preliminaries

In this section I lay out the architectures of SPE–style, rule–based phonologies
and OT–style, constraint–based phonologies, and I explain the notion of phono-
logical opacity.[4]

2.1   Rule–Based Phonology

In an SPE–style, rule–based grammar, inputs are mapped to outputs via ordered,
context–sensitive rewrite rules of the form

$$A \rightarrow B \ / \ C \relbar D$$

read as, "A is rewritten as B whenever A occurs immediately after C and imme-
diately before D."[5] A is the *focus* of the rule, while $C \relbar D$ is the *environment*.
The focus and environment together are called the *input description*. Under a
single such rule, an input like $/CAD/$ is mapped to the output $[CBD]$.[6] If a rule
does not effect any change from input to output, then it applies *vacuously*.

Since rules simply map strings to strings, it is possible for ordered rules to
interact in the following ways (adapted from [12]).[7]

Definition 1. $\mathcal{R}_1$ *feeds* $\mathcal{R}_2$ *iff* $\mathcal{R}_1$ *creates part of* $\mathcal{R}_2$*'s input description.*

Definition 2. $\mathcal{R}_1$ *bleeds* $\mathcal{R}_2$ *iff* $\mathcal{R}_1$ *removes part of* $\mathcal{R}_2$*'s input description.*

Consider, for example, the rule

$$\mathcal{R} : \text{aɪ} \rightarrow \text{ʌɪ} \ / \ \relbar \text{t}$$

---

[4] Many phonologists take *features*, rather than *segments*, to be phonological primitives.
That is, rules and constraints are assumed to target natural classes of segments,
rather than individual segments or arbitrary sets of segments. In principle, this means
that feature–based phonologies are expressively more restrictive than segment–based
ones. However, in this paper the distinction is immaterial: the patterns considered
here are expressible by feature–based rules (hence, also by segment–based rules), and
they are not expressible by segment–based constraints (hence, neither by feature–
based constraints). Accordingly, I assume that rules and constraints generally target
features, but in the proof in section 3, I allow constraints to target arbitrary (sets
of) segments, demonstrating that even with such added power, classic OT grammars
still cannot express the relevant patterns.

[5] The more familiar notation in formal language theory is: $CAD \rightarrow CBD$.

[6] By convention, symbols between forward slashes are inputs (underlying forms), and
those between square brackets are terminal outputs (surface forms). I write non–
terminal, i.e. intermediate, outputs with no brackets at all. These conventions also
apply to OT inputs and output candidates.

[7] Rather than creating (removing) part of $\mathcal{R}_2$'s input description, Baković [12] writes
that $\mathcal{R}_1$ "creates (removes) additional (potential) inputs" to $\mathcal{R}_2$. The two formula-
tions are equivalent. I prefer the former because it facilitates defining feeding and
bleeding on environment and focus (below).

where the vowel /aɪ/ before /t/ raises to [ʌɪ], and the rule

$$\mathcal{F} : \text{t, d} \rightarrow \text{ɾ} \;/\; \text{V} \underline{\quad} \text{V}$$

where /t/ or /d/ between two vowels surfaces as a flap, [ɾ]. If $\mathcal{F}$ is ordered before $\mathcal{R}$, then an input like /raɪtər/ is mapped by $\mathcal{F}$ to raɪɾər, which $\mathcal{R}$ maps vacuously to [raɪɾər]. $\mathcal{F}$ bleeds $\mathcal{R}$ because if $\mathcal{F}$ were absent, then $\mathcal{R}$ would map /raɪtər/ to [rʌɪtər]. The application of $\mathcal{F}$ before $\mathcal{R}$ removes part of $\mathcal{R}$'s input description ($\underline{\quad}$ t), so that $\mathcal{R}$ no longer applies non–vacuously.

Feeding and bleeding can both be further subcategorized (adapted from [13]).

Definition 3. $\mathcal{R}_1$ feeds on $\mathcal{R}_2$'s focus *iff $\mathcal{R}_1$ feeds $\mathcal{R}_2$ by creating $\mathcal{R}_2$'s focus. $\mathcal{R}_1$* feeds on $\mathcal{R}_2$'s environment *iff $\mathcal{R}_1$ feeds $\mathcal{R}_2$ by creating part of $\mathcal{R}_2$'s environment.*

Definition 4. $\mathcal{R}_1$ bleeds on $\mathcal{R}_2$'s focus *iff $\mathcal{R}_1$ bleeds $\mathcal{R}_2$ by removing $\mathcal{R}_2$'s focus. $\mathcal{R}_1$* bleeds on $\mathcal{R}_2$'s environment *iff $\mathcal{R}_1$ bleeds $\mathcal{R}_2$ by removing part of $\mathcal{R}_2$'s environment.*

The example above, with $\mathcal{F}$ ordered before $\mathcal{R}$, is one of bleeding on environment: the change effected by $\mathcal{F}$, i.e. mapping the /t/ in /raɪtər/ to [ɾ], removes part of $\mathcal{R}$'s environment ($\underline{\quad}$ t).

By convention, when I write that $\mathcal{R}_1$ "feeds (bleeds)" $\mathcal{R}_2$, it is implied that $\mathcal{R}_1$ is ordered *before* $\mathcal{R}_2$, even though strictly speaking, feeding (bleeding), as defined here, is independent of rule ordering. When a feeding (bleeding) rule is ordered *after* the rule it feeds (bleeds), the following terms are used.

Definition 5. $\mathcal{R}_1$ counterfeeds (counterbleeds) $\mathcal{R}_2$ *iff $\mathcal{R}_1$ both feeds (bleeds) and is ordered after $\mathcal{R}_2$.*

In the example above, if is ordered $\mathcal{F}$ *after* $\mathcal{R}$, then $\mathcal{F}$ counterbleeds on $\mathcal{R}$'s environment.

These notions of rule interaction have been useful in characterizing *phonological opacity*. Kiparsky [14, 15] was the first to identify the phenomenon of opacity, defining it as follows, where "process" can be construed as a rewrite rule.

Definition 6. *A process $\mathcal{P}$ of the form $A \rightarrow B \;/\; C \underline{\quad} D$ is* opaque *to the extent that there are surface representations (outputs) of the form (i) A in the environment $C \underline{\quad} D$, or (ii) B derived by $\mathcal{P}$ in environments other than $C \underline{\quad} D$.*

The idea is that opaque phonological generalizations are (i) generalizations that appear not to hold true of a surface form, or (ii) generalizations that are true of a surface form, but the motivation for their application is obscured.

These two types of opacity are typically associated with counterfeeding and counterbleeding rule ordering, respectively [11, 12, 16]. In counterfeeding, a later rule creates part of the input description of an earlier rule, such that the earlier rule seems not to have applied to the surface form, even though it matches the rule's input description. In counterbleeding, a later rule removes part of the input description of an earlier rule, such that the earlier rule seems, on the surface, to have applied without satisfying its input description.

/raɪtər/	*aɪt	*VtV	ID(low)	ID(son)
a. raɪtər	1	1	0	0
b. rʌɪtər	0	1	1	0
c. raɪɾər	0	0	0	1
d. rʌɪɾər	0	0	1	1

Fig. 1. An example OT tableau

For example, if rule $\mathcal{F}$ from above is ordered after $\mathcal{R}$, giving rise to a counterbleeding on environment rule interaction, then the input /raɪtər/ is mapped by $\mathcal{R}$ to rʌɪtər, which $\mathcal{F}$ maps to [rʌɪɾər]. $\mathcal{R}$ is then opaque in the sense of (ii): [ʌɪ] occurs in an environment other than ＿ t. In such a case, I will often abstract away from the rules and say that the input–output pattern /raɪtər/ $\rightarrow$ [rʌɪɾər] is opaque.

Although opacity is defined here in terms of rules, a natural question is whether opaque input–output patterns like /raɪtər/ $\rightarrow$ [rʌɪɾər] are expressible by other formalisms, like OT grammars. In the next section, I describe the architecture of OT grammars, for which such patterns have been problematic.

2.2   Optimality Theoretic Phonology

In an OT grammar, an input like /raɪtər/ is first fed into a function GEN that generates an infinite set of *output candidates*. In practice, every possible output is a candidate. This set is then filtered down via a potentially infinite set of strictly, totally ordered constraints until a unique output candidate remains. The entire process can be visualized using an OT *tableau.*

For example, Fig. 1 presents a tableau with four output candidates for the input /raɪtər/ and four constraints, ranked left to right from highest to lowest. The process begins with the leftmost constraint, *aɪt, which is an output–markedness constraint that assigns a violation for each occurrence of the sequence [aɪt] in an output candidate. Candidate (a) thus incurs a violation of 1, while the other candidates incur 0 violations; hence, candidate (a) is eliminated. The next constraint, *VtV, which assigns a violation mark for each intervocalic [t] in an output candidate, eliminates candidate (b). Next, IDENT–IO(low), an input–output faithfulness constraint which penalizes each occurrence of a low vowel like /aɪ/ mapping to a non–low vowel like [ʌɪ], eliminates candidate (d). At this point, only candidate (c) remains and is therefore the optimal output. The last constraint, IDENT–IO(sonorant), which penalizes mapping a non–sonorant segment like /t/ to a sonorant one like [ɾ], effectively does no work here.

Note that in this tableau the opaque input–output pattern /raɪtər/ $\rightarrow$ [rʌɪɾər] is *not* optimal. Moreover, the violations assigned to [rʌɪɾər] are a proper superset of those assigned to [raɪɾər]. Consequently, no reranking of these particular constraints can possibly make [rʌɪɾər] more optimal than [raɪɾər]. This is the sort of reasoning that underlies the intuition that opacity is problematic for OT.

I now give a formal characterization of OT grammars, following [8, 17]. If inputs and output candidates are taken to be strings of symbols over Σ, then GEN is a relation over $\Sigma^* \times \Sigma^*$: it pairs input strings with (sets of) output candidate strings.[8] An OT grammar can then be defined formally as follows.

Definition 7. *An OT grammar is a tuple* $\langle \Sigma, \text{GEN}, C, >_C \rangle$, *where* $\text{GEN} \subseteq \Sigma^* \times \Sigma^*$; *C is a set of functions from* GEN *to* $\mathbb{N}$; *and* $>_C$ *is a strict, total order on C.*

A relation $>_{\mathcal{G}}$ over GEN defines optimality relative to two input–output pairs: $\langle i, o \rangle >_{\mathcal{G}} \langle i, o' \rangle$ means that $\langle i, o \rangle$ is more optimal than $\langle i, o' \rangle$ in the OT grammar $\mathcal{G}$. In such a case, I will often write that o is more optimal than o' with respect to i (in $\mathcal{G}$).

Definition 8. *Given an OT grammar* $\mathcal{G} = \langle \Sigma, \text{GEN}, C, >_C \rangle$, *with* $\langle i, o \rangle$, $\langle i, o' \rangle$ $\in$ GEN, $\langle i, o \rangle >_{\mathcal{G}} \langle i, o' \rangle$ *iff there is some* $c_j \in C$ *such that* $c_j(\langle i, o \rangle) < c_j(\langle i, o' \rangle)$, *and for each* c_k *such that* $c_k >_C c_j$, $c_k(\langle i, o \rangle) = c_k(\langle i, o' \rangle)$.

A single output candidate for some input is then optimal just in case it is more optimal than every other candidate with respect to that input.

Definition 9. *Given an OT grammar* $\mathcal{G} = \langle \Sigma, \text{GEN}, C, >_C \rangle$, $\langle i, o \rangle \in$ GEN *is optimal in* $\mathcal{G}$ *iff for each* $\langle i, o' \rangle \in$ GEN *(o different from o'),* $\langle i, o \rangle >_{\mathcal{G}} \langle i, o' \rangle$.

Turning now to the constraint set, an OT constraint can be represented by a weighted finite state transducer (FST) [7–9], called a *finite state OT constraint*. This FST essentially "reads" or "processes" pairs of input–output strings, one input–output symbol pair at a time, transitioning from state to state and assigning 0 or 1 violations to each pair of corresponding input–output symbols. The symbol ϵ here denotes the empty string. (It should not be confused with the IPA symbol ɛ, which denotes an open–mid front vowel.)

Definition 10. *A finite state OT constraint is a tuple* $\langle Q, \Sigma, \delta, q_0, F \rangle$, *where:*

1. *Q is a non–empty set of states;*
2. Σ *is a set of symbols;*
3. δ *is a transition function from* $Q \times \Sigma^\epsilon \times \Sigma^\epsilon$ *to* $\{0, 1\} \times Q$, *where* $\Sigma^\epsilon = \Sigma \cup \{\epsilon\}$;
4. $q_0 \in Q$ *is the unique start state;*
5. $F \subseteq Q$ *is the non–empty set of final states.*

The transition function δ takes in a triple $\langle q_j, i, o \rangle$ consisting of some initial state $q_j \in Q$, an input symbol $i \in \Sigma^\epsilon$, and an output symbol $o \in \Sigma^\epsilon$ and returns a pair $\langle n, q_k \rangle$ consisting of some number of violations $n \in \{0, 1\}$ and some terminal state $q_k \in Q$. For ease of exposition, I will write members of δ as $\langle q_j, i, o, n, q_k \rangle$ instead of $\langle \langle q_j, i, o \rangle, \langle n, q_k \rangle \rangle$ and say that δ assigns n violations to the pair $i \to o$ (on the transition from q_j to q_k).

---

[8] Recall that Σ^* denotes the set of all strings over the symbols in Σ, including the empty string.
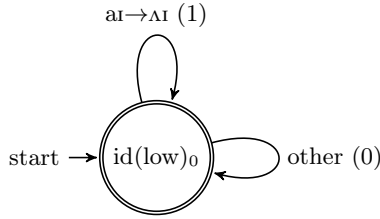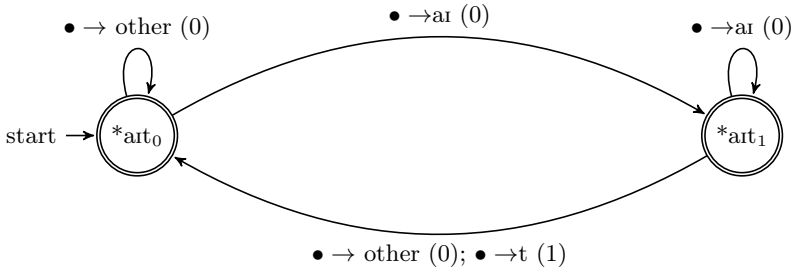
Fig. 2. A visualization of **id(low)**

Fig. 3. A visualization of ***aɪt**

Consider, for example, the faithfulness constraint IDENT–IO(low), which, to repeat, penalizes each mapping of a low vowel like /aɪ/ to a non–low vowel like [ʌɪ]. I assume that diphthongs like [aɪ] and [ʌɪ] are single units rather than segments composed of two vowels. Formally, this means aɪ and ʌɪ are each single symbols in Σ. IDENT–IO(low) can then be represented by the single–state FST

$$\mathbf{id(low)} = \langle \{\mathrm{id(low)}_0\}, \{\mathrm{aɪ}, \mathrm{ʌɪ}, \dots\}, \delta, \mathrm{id(low)}_0, \{\mathrm{id(low)}_0\}\rangle$$

where δ assigns 1 to the pair aɪ→ʌɪ and 0 to every other pair. This FST is conventionally visualized as in Fig. 2. Thus, an input–output pair like ⟨raɪtər, raɪtər⟩ incurs 0 total violations, whereas ⟨raɪtər, rʌɪtər⟩ incurs 1 total violation.

A markedness constraint like *aɪt, which penalizes each occurrence of the sequence [aɪt] in an output candidate, can be represented by the two–state FST

$$\mathbf{*aɪt} = \langle \{\mathrm{*aɪt}_0, \mathrm{*aɪt}_1\}, \{\mathrm{aɪ}, \mathrm{ʌɪ}, \mathrm{t}, \dots\}, \delta, \mathrm{*aɪt}_0, \{\mathrm{*aɪt}_0, \mathrm{*aɪt}_1\}\rangle$$

where δ assigns 1 whenever the sequence [aɪt] occurs in an output, regardless of the input, and 0 otherwise. This FST is visualized as in Fig. 3, where • stands for any symbol from the symbol set (or the empty string).

Of note is that the violations assigned by a markedness FST are independent of the input (hence the • on every transition). By contrast, the violations assigned by a faithfulness FST depend on the input. This can be formalized as follows.[9]

Definition 11. *Given an OT grammar* $\mathcal{G} = \langle \Sigma, \text{GEN}, C, >_C \rangle$, *a constraint* $c \in C$ *is a* markedness constraint *iff it can be represented by a finite state constraint* $\langle Q, \Sigma, \delta, q_0, F \rangle$ *such that, if* $\langle q_j, \alpha, \beta, 1, q_k \rangle \in \delta$, *then for any* $\gamma \in \Sigma^\epsilon$ *such that* $\langle q_j, \gamma, \beta, n, q_k \rangle \in \delta$, *it follows that* $n = 1$. *An FST that represents a markedness constraint is a* finite state markedness constraint.

Definition 12. *Given an OT grammar* $\mathcal{G} = \langle \Sigma, \text{GEN}, C, >_C \rangle$, *a constraint* $c \in C$ *is a* faithfulness constraint *iff it can be represented by a finite state constraint* $\langle Q, \Sigma, \delta, q_0, F \rangle$ *such that* $\langle q_j, \alpha, \beta, 1, q_k \rangle \in \delta$ *and* $\langle q_j, \gamma, \beta, 0, q_k \rangle \in \delta$, *for some* $\alpha, \beta, \gamma \in \Sigma^\epsilon$ *and for some* $q_j, q_k \in Q$. *An FST that represents a faithfulness constraint is a* finite state faithfulness constraint.

Moreover, whereas (finite state) markedness constraints may have one or more states, standard faithfulness constraints like IDENT, MAX, and DEP have just one state: when assigning a violation mark to an input–output symbol pair $a \to b$, these constraints do not "look ahead" or "look behind" at other symbols in the input or output. I call this sort of constraint a *single–state faithfulness constraint*.

Definition 13. *Given an OT grammar* $\mathcal{G} = \langle \Sigma, \text{GEN}, C, >_C \rangle$, *a constraint* $c \in C$ *is a* single–state faithfulness constraint *iff it can be represented by a finite state faithfulness constraint* $\langle Q, \Sigma, \delta, q_0, F \rangle$ *such that* $|Q| = 1$.

The definition of a classic OT grammar can now be stated as follows.[10,11]

Definition 14. *A* classic OT grammar *is an OT grammar* $\mathcal{G} = \langle \Sigma, \text{GEN}, C, >_C \rangle$ *such that (i) if* $\langle i, o \rangle \in \text{GEN}$, *then* $\langle i, o' \rangle \in \text{GEN}$, *for each* $o' \in \Sigma^*$; *and (ii) each* $c \in C$ *is either is a markedness constraint or a single–state faithfulness constraint.*

3   The Proof

In this section I prove the following claim.

---

[9] When there is no confusion, I will write "markedness (faithfulness) constraint" rather than "finite state markedness (faithfulness) constraint" to refer to the FST representing some OT constraint.

[10] The requirement on GEN in (i) is intended to capture the assumption, stated above, that every possible output of a given input is considered to be a candidate. See the discussion of *freedom of analysis* in [16, p. 20]. This requirement also ensures that the relevant candidates are in competition in order for the proof in section 3 to proceed, though any weaker requirement that ensures this would suffice, too.

[11] Some phonologists might object that the single–state restriction on faithfulness constraints in (ii) is *not enough*, since, for example, faithfulness constraints penalizing arbitrary input–output symbols are allowed. However, if, as I prove in section 3, even this relatively powerful version of classic OT is incapable of expressing certain regular relations, then so is any more restrictive version of it.

	Input		Output	
a.	/raɪt/	$\rightarrow$	[rʌɪt]	"write"
b.	/raɪd/	$\rightarrow$	[raɪd]	"ride"
c.	/raɪtər/	$\rightarrow$	[rʌɪɾər]	"writer"
d.	/raɪdər/	$\rightarrow$	[raɪɾər]	"rider"

Fig. 4. Canadian raising data

		$\mathcal{R}$		$\mathcal{F}$	
a.	/raɪt/	$\rightarrow$	rʌɪt	$\rightarrow$	[rʌɪt]
b.	/raɪd/	$\rightarrow$	raɪd	$\rightarrow$	[raɪd]
c.	/raɪtər/	$\rightarrow$	rʌɪtər	$\rightarrow$	[rʌɪɾər]
d.	/raɪdər/	$\rightarrow$	raɪdər	$\rightarrow$	[raɪɾər]

Fig. 5. Rule–based derivations of Canadian raising

Claim. There is a relation R such that (i) R is regular, and (ii) there is no classic OT grammar $\mathcal{G}$ such that each $\langle i, o \rangle \in R$ is optimal in $\mathcal{G}$.

The patterns I will use are the Canadian raising patterns already discussed, presented in Fig. 4. I first show that these patterns are expressible by ordered rewrite rules; hence, they are regular, proving part (i) of the claim. I then show, less trivially, that the patterns cannot be expressed by any classic OT grammar, proving part (ii) of the claim. The main result is proved as Theorem 2.

All four patterns in Fig. 4 can be captured by a rule $\mathcal{R}$ of raising /aɪ/ to [ʌɪ] before underlying /t/

$$\mathcal{R} : \text{aɪ} \rightarrow \text{ʌɪ} / \underline{\quad}\text{t}$$

ordered before a rule $\mathcal{F}$ of flapping (changing intervocalic /t/ or /d/ to [ɾ])

$$\mathcal{F} : \text{t, d} \rightarrow \text{ɾ} / \text{V}\underline{\quad}\text{V}$$

where V stands for any vowel, e.g. /aɪ/, /ʌɪ/, and /ə/. Fig. 5 shows how the four patterns are expressed by the interaction of these two rules.

Recall that the pattern /raɪtər/ $\rightarrow$ [rʌɪɾər] is the crucial opaque pattern: /aɪ/ raises to [ʌɪ] because of the underlying /t/ ($\mathcal{R}$'s environment), but /t/ is changed to [ɾ] by $\mathcal{F}$. Thus, $\mathcal{F}$ removes the environment that motivates the application of $\mathcal{R}$, obscuring, on the surface, the reason for raising.

These patterns are intuitively problematic for classic OT for the following reason: given the evidence from /raɪdər/ that [raɪɾər] is an unmarked output, it is always more optimal to map /raɪtər/ to [raɪɾər], incurring just one faithfulness violation (t→ɾ), than to map /raɪtər/ to [rʌɪɾər], incurring two faithfulness violations (t→ɾ, aɪ→ʌɪ). (This explains why in Fig. 1 of the previous section, [raɪɾər] is a more optimal output of /raɪtər/ than [rʌɪɾər].)

Informally, the patterns /raɪtər/ $\rightarrow$ [rʌɪɾər] and /raɪdər/ $\rightarrow$ [raɪɾər] are conflicting. More formally, we have the following theorem.

Theorem 1. *Let $\mathcal{G} = \langle \Sigma, \text{GEN}, C, >_C \rangle$ be a classic OT grammar such that $\langle \text{raɪtər, rʌɪɾər} \rangle$ and $\langle \text{raɪdər, raɪɾər} \rangle$ are members of* GEN. *Then it is not the case both that $\langle \text{raɪtər, rʌɪɾər} \rangle$ is optimal in $\mathcal{G}$ and that $\langle \text{raɪdər, raɪɾər} \rangle$ is optimal in $\mathcal{G}$.*

Proof. The proof is by contradiction. Assume both that $\langle \text{raɪtər, rʌɪɾər} \rangle$ is optimal in $\mathcal{G}$ and that $\langle \text{raɪdər, raɪɾər} \rangle$ is optimal in $\mathcal{G}$. Then for each $\langle \text{raɪtər, } o \rangle \in$ GEN (with o different from [rʌɪɾər]), $\langle \text{raɪtər, rʌɪɾər} \rangle >_{\mathcal{G}} \langle \text{raɪtər, } o \rangle$; and for each $\langle \text{raɪdər, } o \rangle \in$ GEN (with o different from [raɪɾər]), $\langle \text{raɪdər, raɪɾər} \rangle >_{\mathcal{G}} \langle \text{raɪdər, } o \rangle$.

Since $\langle \text{raɪtər, rʌɪɾər} \rangle$ and $\langle \text{raɪdər, raɪɾər} \rangle$ are members of GEN, then so are $\langle \text{raɪtər, raɪɾər} \rangle$ and $\langle \text{raɪdər, rʌɪɾər} \rangle$. It follows both that $\langle \text{raɪtər, rʌɪɾər} \rangle >_{\mathcal{G}} \langle \text{raɪtər, raɪɾər} \rangle$ and that $\langle \text{raɪdər, raɪɾər} \rangle >_{\mathcal{G}} \langle \text{raɪdər, rʌɪɾər} \rangle$. By the definition of optimality, the following statements are derived.

$$\text{There is a } c_j \in C \text{ such that } c_j(\langle \text{raɪtər, rʌɪɾər} \rangle) < c_j(\langle \text{raɪtər, raɪɾər} \rangle). \quad (1)$$

$$\text{For all } c' \in C \text{ such that } c' >_C c_j, c'(\langle \text{raɪtər, rʌɪɾər} \rangle) = c'(\langle \text{raɪtər, raɪɾər} \rangle). \quad (2)$$

$$\text{There is a } c_k \in C \text{ such that } c_k(\langle \text{raɪdər, raɪɾər} \rangle) < c_k(\langle \text{raɪdər, rʌɪɾər} \rangle). \quad (3)$$

$$\text{For all } c' \in C \text{ such that } c' >_C c_k, c'(\langle \text{raɪdər, raɪɾər} \rangle) = c'(\langle \text{raɪdər, rʌɪɾər} \rangle). \quad (4)$$

Recall that, as defined here, classic OT grammars contain only markedness and single–state faithfulness constraints. Thus, to prove the theorem, it must be shown that at least one of c_j, c_k is neither of these types. There are four cases to consider.

CASE 1. Suppose that both c_j and c_k are markedness constraints. Then c_j assigns m violation marks to [rʌɪɾər] and n violations to [raɪɾər], regardless of input; hence, from (1) it follows that $m < n$. Similarly, c_k assigns p violation marks to [rʌɪɾər] and q violations to [raɪɾər], regardless of input; hence, from (3) it follows that $q < p$.

It cannot be the case that c_j and c_k are the same constraint, for then $m = p$ and $n = q$, deriving a contradiction. Thus, c_j and c_k are different constraints, with one ranked above the other. Suppose that $c_j >_C c_k$. Then it follows from (4) that

$$c_j(\langle \text{raɪdər, raɪɾər} \rangle) = c_j(\langle \text{raɪdər, rʌɪɾər} \rangle)$$

and hence $n = m$. But it was already established that $m < n$, so it cannot be that $c_j >_C c_k$.

Now suppose that $c_k >_C c_j$. Then it follows from (2) that

$$c_k(\langle \text{raɪtər, rʌɪɾər} \rangle) = c_k(\langle \text{raɪtər, raɪɾər} \rangle)$$

and hence $p = q$. But it was already established that $q < p$, so it cannot be that $c_k >_C c_j$. We have reached a contradiction, so c_j and c_k cannot both be markedness constraints. At least one of them is a single–state faithfulness constraint.

CASE 2. Suppose that c_j is a single–state faithfulness constraint and that c_k is a markedness constraint. From (1) it follows that the FST representing c_j assigns 1 violation to the pair aɪ→aɪ and 0 violations to the pair aɪ→ʌɪ, because the two inputs are the same and the two output candidates differ only in having [aɪ] or [ʌɪ] as the second segment. Thus, it follows that

$$c_j(\langle \text{raɪdər}, \text{rʌɪɾər} \rangle) < c_j(\langle \text{raɪdər}, \text{raɪɾər} \rangle)$$

since, again, the two inputs are the same and the two output candidates differ only in the second segment.

Since c_j and c_k are different constraints, one must outrank the other. It cannot be that $c_j >_C c_k$, for it was just established that

$$c_j(\langle \text{raɪdər}, \text{rʌɪɾər} \rangle) \neq c_j(\langle \text{raɪdər}, \text{raɪɾər} \rangle)$$

which contradicts (4). Suppose, then, that $c_k >_C c_j$. Since by hypothesis c_k is a markedness constraint, then from (3) it follows that c_k assigns fewer violations to [raɪɾər] than to [rʌɪɾər], regardless of input, so that

$$c_k(\langle \text{raɪtər}, \text{raɪɾər} \rangle) \neq c_k(\langle \text{raɪtər}, \text{rʌɪɾər} \rangle)$$

But this contradicts (2). We have reached a contradiction: if c_j is a single–state faithfulness constraint, then c_k cannot be a markedness constraint.

CASE 3. Suppose that c_j and c_k are both single–state faithfulness constraints. From (3) it follows that the FST representing c_k assigns 1 violation to the pair aɪ→ʌɪ and 0 violations to the pair aɪ→aɪ. Thus, it follows that

$$c_k(\langle \text{raɪtər}, \text{raɪɾər} \rangle) < c_k(\langle \text{raɪtər}, \text{rʌɪɾər} \rangle)$$

Since this statement is different from (1), c_j and c_k must be different constraints and hence ranked one over the other. It cannot be that $c_k >_C c_j$, for the above statement contradicts (2). However, as was established in case 2, if c_j is a single–state faithfulness constraint, then it cannot be that $c_j >_C c_k$. We have reached a contradiction: if c_k is a single–state faithfulness constraint, then c_j must be a markedness constraint.

CASE 4. Suppose that c_j is a markedness constraint and that c_k is a single–state faithfulness constraint. From (1) it follows that c_j assigns fewer violations to [rʌɪɾər] than to [raɪɾər], regardless of input; hence, from the statement derived in case 3 concerning c_k as a single–state faithfulness constraint, it follows that c_j and c_k are different constraints, with one ranked above the other.

It cannot be that $c_j >_C c_k$, for otherwise

$$c_j(\langle \text{raɪdər}, \text{rʌɪɾər} \rangle) < c_j(\langle \text{raɪdər}, \text{raɪɾər} \rangle)$$

which contradicts (4). Thus, $c_k >_C c_j$. However, as already established in case 3, if c_k is a single–state faithfulness constraint, then it cannot be that $c_k >_C c_j$.

We have reached a contradiction and have exhausted all cases: there are no markedness or single–state faithfulness constraints that satisfy (1–4). Hence, it is not the case both that ⟨raɪtər, rʌɪɾər⟩ is optimal in $\mathcal{G}$ and that ⟨raɪdər, raɪɾər⟩ is optimal in $\mathcal{G}$.

$\square$

From Theorem 1, together with the rule–based analysis above, the claim at the start of this section follows easily.

Theorem 2. *There is a relation R such that (i) R is regular, and (ii) there is no classic OT grammar $\mathcal{G}$ such that each $\langle i, o \rangle \in R$ is optimal in $\mathcal{G}$.*

Proof. Let $R = \{\langle$raɪtər, rʌɪɾər$\rangle, \langle$raɪdər, raɪɾər$\rangle\}$. For part (i), since there are ordered rewrite rules ($\mathcal{R}$ and $\mathcal{F}$ above) that map /raɪtər/ to [rʌɪɾər] and map /raɪdər/ to [raɪɾər], it follows that R is regular. Part (ii) follows from Theorem 1.

$\square$

4   Discussion

In the previous section, I proved that Canadian raising, which is an example of counterbleeding on environment opacity, is expressible by ordered rewrite rules but not by classic OT grammars. I now show that there are other examples of counterbleeding on environment opacity, as well as examples of counterfeeding on environment opacity, that are likewise inexpressible by classic OT grammars. Though the patterns are all different, I discuss several formal properties that they share.[12]

4.1   Other Examples of Counterbleeding on Environment Opacity

Polish. Consider the data from Polish in Fig. 6 [12, 18, 19]. All four patterns can be captured by a rule $\mathcal{R}$ of raising /o/ to [u] before voiced segments (/l, b/), ordered before a rule $\mathcal{D}$ of devoicing word–final obstruents, i.e. mapping word–final /z, b/ to [s, p]. $\mathcal{D}$ counterbleeds on $\mathcal{R}$'s environment because $\mathcal{D}$ removes part of $\mathcal{R}$'s environment (__ l, b) by mapping /b/ to [p].

The patterns /ʒwob/ → [ʒwup] and /ʒwop/ → [ʒwop] cannot be expressed by any classic OT grammar. The proof is essentially the same as that of Theorem 1, and the reason is that the Canadian raising patterns and the Polish patterns are formally almost identical. In Canadian raising, /aɪ/ surfaces as [ʌɪ] before /t/, which (along with /d/) surfaces as [ɾ]. In Polish, /o/ surfaces as [u] before /b/, which (along with /p/) surfaces as [p]. The difference is that in Canadian raising, /t, d/ surface as a third segment, [ɾ], whereas in Polish, /b, p/ surface as [p].

---

[12] In each of the following examples, I employ a hypothetical input that is not actually part of that language's lexicon, as far as I know. Nonetheless, I take such absences to simply be accidental lexical gaps. Moreover, the formal result that these sets of input–outputs patterns are expressible by ordered rules but not by classic OT still holds; the empirical question of whether the patterns are attested is a separate issue.

	Input		Output	
a.	/sol/	→	[sul]	"rubble"
b.	/gruz/	→	[grus]	"salt"
c.	/ʒwob/	→	[ʒwup]	"crib"
d.	/ʒwop/	→	[ʒwop]	hypothetical

Fig. 6. Polish data

	Input		Output	
a.	/ħaːkimiːn/	→	[ħaːkʲmiːn]	"ruling (masc. pl.)
b.	/ħaːkmiːn/	→	[ħaːkmiːn]	hypothetical

Fig. 7. Bedouin Arabic data 1

Bedouin Arabic 1. Consider the data from Bedouin Arabic in Fig. 7 [11]. Both patterns are expressible by a rule $\mathcal{P}$ of palatalizing /k/ to [kʲ] before /i/, ordered before a rule $\mathcal{D}$ of deleting /i/ (mapping /i/ to ϵ). $\mathcal{D}$ counterbleeds on $\mathcal{P}$'s environment because $\mathcal{D}$ removes part of $\mathcal{P}$'s environment (__ i) by deleting /i/. These two patterns cannot be expressed by any classic OT grammar, and the reason, again, is that these patterns are formally similar to those in Canadian raising: /k/ surfaces as [kʲ] before /i/, which (along with ϵ) surfaces as ϵ.

4.2 Examples of Counterfeeding on Environment Opacity

Isthmus Nahuat. Consider the data from Isthmus Nahuat in Fig. 8 [16, 19, 20]. All four patterns are expressible by a rule $\mathcal{D}$ of devoicing /l/ to [l̥] word–finally, ordered before a rule $\mathcal{A}$ of apocope, i.e. deleting /i/ word–finally. $\mathcal{A}$ counterfeeds on $\mathcal{D}$'s environment because $\mathcal{A}$ creates part of $\mathcal{D}$'s environment (__ #, where # denotes a word boundary) by deleting a word–final /i/ (mapping i# to ϵ#).

The patterns /ʃikɑkíli/ → [ʃikɑkíl] and /ʃikɑkíl/ → [ʃikɑkíl] are not expressible by any classic OT grammar. The proof, omitted here, is essentially the same as that of Theorem 1 for counterbleeding on environment opacity, using ⟨ʃikɑkíli, ʃikɑkíl⟩, ⟨ʃikɑkíli, ʃikɑkíl⟩, ⟨ʃikɑkíl, ʃikɑkíl⟩, and ⟨ʃikɑkíl, ʃikɑkíl⟩ as the relevant members of GEN. Informally, the reason that these patterns are inexpressible is the following: given the evidence from /ʃikɑkíli/ that [ʃikɑkíl] is an unmarked output, it is always more optimal to map /ʃikɑkíl/ to [ʃikɑkíl], incurring no faithfulness violations, than to map /ʃikɑkíl/ to [ʃikɑkíl], incurring one faithfulness violation (l → l̥). Or, in terms of the formalism presented here, /i/ surfaces as ϵ word–finally, and /l/ surfaces as [l̥], but only if /l/ does not precede a word–final /i/; and this latter condition on the identity of /l/ can be expressed only by a multistate faithfulness constraint.

Bedouin Arabic 2. Fig. 9 presents more data from Bedouin Arabic [11]. Both patterns can be captured by a rule $\mathcal{R}$ of raising /a/ to [i] in an open syllable,

	Input		Output	
a.	/támi/	$\rightarrow$	[tám]	"it ends"
b.	/tájoːl/	$\rightarrow$	[tájoːl̥]	"shelled corn"
c.	/ʃikɑkíli/	$\rightarrow$	[ʃikɑkíl]	"put it in it"
d.	/ʃikɑkíl/	$\rightarrow$	[ʃikɑkíl̥]	hypothetical

Fig. 8. Isthmus Nahuat data

	Input		Output	
a.	/gabr/	$\rightarrow$	[gabur]	"grave"
b.	/gabur/	$\rightarrow$	[gibur]	hypothetical

Fig. 9. Bedouin Arabic data 2

ordered before a rule $\mathcal{E}$ of epenthesis, i.e. inserting a [u] (mapping ϵ to [u]) between two consonants like /br/. $\mathcal{E}$ counterfeeds on $\mathcal{R}$'s environment because $\mathcal{E}$ creates part of $\mathcal{R}$'s environment (__ bV) by inserting a vowel, [i], after [b]. These two patterns cannot be expressed by any classic OT grammar, and the reason is that these patterns are formally similar to those in Isthmus Nahuat. In Bedouin Arabic, ϵ surfaces as [u] between /br/, and /a/ surfaces as [i], but only when /a/ does not precede a consonant cluster.

Lomongo. Consider, finally, the data from Lomongo in Fig. 10 [12, 19, 21]. Both patterns can be captured by a rule $\mathcal{G}$ of prevocalic gliding, i.e. mapping /o/ to [w] before a vowel, ordered before a rule $\mathcal{D}$ of intervocalic obstruent deletion, i.e. deleting /b/ between two vowels. $\mathcal{D}$ counterfeeds on $\mathcal{G}$'s environment because $\mathcal{D}$ creates part of $\mathcal{G}$'s environment (__ V) by deleting a prevocalic /b/, i.e. by mapping /bV/ to just [V]. These patterns are not expressible by any classic OT grammar, and the reason, once more, is that these patterns are similar to those in Isthmus Nahuat. In this case, /b/ surfaces as ϵ between two vowels, and /o/ surfaces as [w], but only when /o/ does not precede a prevocalic /b/.

Summary. Fig. 11 summarizes all of the sets of input–output candidate pairs that have been discussed in this paper, with the empty string symbol (ϵ) included where appropriate. Each of the six sets exhibits the following three properties: (i) the two inputs differ by exactly one segment x; (ii) the two output candidates differ by exactly one segment y; and (iii) x is different from y, i.e. they occupy different string positions. (In Fig. 11, the differing segments are in bold.) Formally, then, these sets are essentially the same; hence, it is not surprising that in all six examples the relevant input–output patterns cannot be made optimal by classic OT grammars, as defined here. Moreover, from this perspective it emerges that the difference between the counterbleeding on environment examples and the counterfeeding on environment examples boils down to which two (of the four possible) input–output candidate pairs are supposed to be optimal.

	Input		Output	
a.	/obina/	→	[oina]	"you (sg.) dance"
b.	/oina/	→	[wina]	hypothetical

Fig. 10. Lomongo data

Language	Inputs		Outputs	
Canadian English	raɪtər	raɪdər	raɪɾər	rʌɪɾər
Polish	ʒwob	ʒwop	ʒwop	ʒwup
Bedouin Arabic 1	ħaːkimiːn	ħaːkemiːn	ħaːkemiːn	ħaːkʲemiːn
Isthmus Nahuat	ʃikakíli	ʃikakílɛ	ʃikakílɛ	ʃikakílɛ
Bedouin Arabic 2	gabɛr	gabur	gabur	gibur
Lomongo	obina	oɛina	oɛina	wɛina

Fig. 11. A summary of all the input–output candidate pairs in this paper

Notably, the proof technique used in this paper (finding two conflicting input–output patterns) does not seem to work for cases of focus opacity. To give just one example of counterfeeding on focus opacity, in Western Basque [22], /a/ raises to [e], and /e/ raises to [i], but /a/ does not raise to [i]. To express the latter generalization in OT, it suffices to posit an undominated, single–state faithfulness constraint that assigns 1 to the pair a→i and 0 to every other pair. Such a constraint may seem intuitively ad hoc, yet it is formally sound.

More generally, the reason that focus opacity seems unproblematic for OT from this paper's perspective is twofold: (i) classic faithfulness constraints, as defined here, are single–state faithfulness constraints that can penalize arbitrary, single input–output segment pairs; and (ii) in focus opacity, all change occurs at a single focus. Thus, it suffices to posit a single–state faithfulness constraint that assigns 1 violation to those focus changes that are undesired, like a→i, and 0 to those that are desired. Moreover, taking into account Fig. 11 and the three properties of environment opacity mentioned above, it comes as no surprise that the focus opacity of Western Basque works differently: it necessarily lacks at least one of the three properties. Specifically, if the first two properties hold, i.e. the relevant inputs differ by exactly one segment x and the relevant output candidates differ by exactly one segment y, then the third property, according to which x and y occupy different string positions, must not hold, because in focus opacity the segments that change (a, e, i) all occupy the same string position.

5   Conclusion

In this paper I defined a classic OT grammar as any OT grammar that contains only markedness constraints, i.e. constraints representable by an FST that is input–*independent*, and single–state faithfulness constraints, i.e. constraints representable by an FST that is both input–*dependent* and *single*–state. I then

proved, using data from Canadian English, that there are input–output patterns which can be expressed by SPE–style, ordered, context–sensitive rewrite rules, but which cannot be expressed by *any* classic OT grammar (Theorem 1). Hence, there are regular relations that classic OT grammars, as defined here, cannot express (Theorem 2). I also demonstrated that several other cases of counterbleeding on environment opacity, as well as several cases of counterfeeding on environment opacity, are likewise expressible by ordered rewrite rules but not by classic OT. Lastly, I argued that focus opacity, unlike environment opacity, seems unproblematic for classic OT grammars that allow arbitrary single–state faithfulness constraints.

Regarding empirical coverage, assuming that the Canadian raising data are attested, then classic OT, as defined here, undergenerates. Assuming also that natural language phonology is strictly subregular (see [5]), then, since I did not place any upper bound on the number of violations that any constraint may assign, classic OT also overgenerates: it can express unattested, non–regular patterns (see footnote 3). Ordered rewrite rules, under the same assumptions, only overgenerate.[13] However, whether or not this serves as a basis to favor rules over classic OT remains to be seen: it could be, for example, that the classes of patterns by which classic OT grammars over– and undergenerate are in some formal sense smaller (or more manageable) than the class of patterns by which rules overgenerate.

References

1. Chomsky, N., Halle, M.: The Sound Pattern of English. The MIT Press (1968)
2. Prince, A., Smolensky, P.: Optimality Theory: Constraint Interaction in Generative Grammar. Blackwell Publishers (2004)
3. Johnson, C.D.: Formal Aspects of Phonological Description. Mouton (1972)
4. Kaplan, R.M., Kay, M.: Regular models of phonological rule systems. Computational Linguistics 20(3), 331–378 (1994)
5. Heinz, J.: Computational phonology—part I: Foundations. Language and Linguistics Compass 5, 140–152 (2011)
6. Gerdemann, D., Hulden, M.: Practical finite state optimality theory. In: Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing, pp. 10–19 (2012)
7. Riggle, J.: Generation Recognition and Learning in Finite State Optimality Theory. PhD thesis, UCLA (2004)
8. Frank, R., Satta, G.: Optimality Theory and the generative complexity of constraint violability. Computational Linguistics 24(2), 307–315 (1998)

---

[13] I thank an anonymous referee for raising these points.

9. Karttunen, L.: The proper treatment of optimality in computational phonology. In: Proceedings of the International Workshop on Finite State Methods in Natural Language Processing, Association for Computational Linguistics, pp. 1–12 (1998)

10. Heinz, J.: Computational phonology—part II: Grammars, learning, and the future. Language and Linguistics Compass 5(4), 153–168 (2011)

11. McCarthy, J.: Hidden Generalizations: Phonological Opacity in Optimality Theory. Equinox Publishing Ltd. (2007)

12. Baković, E.: Opacity and ordering. In: Goldsmith, J.A., Riggle, J., Yu, A.C.L. (eds.) The Handbook of Phonological Theory, 2nd edn., Blackwell Publishers (2011)

13. McCarthy, J.: Sympathy and phonological opacity. Phonology 16, 331–399 (1999)

14. Kiparsky, P.: Historical linguistics. In: Dingwall, W.O. (ed.) A Survey of Linguistic Science. University of Maryland Linguistics Program, College Park (1971)

15. Kiparsky, P.: Abstractness, opacity, and global rules. In: Fujimura, O., Smith, D.L. (eds.) Three Dimensions of Linguistic Theory, TEC (1973)

16. Kager, R.: Optimality Theory. Cambridge University Press (1999)

17. Graf, T.: Reference–set constraints as linear tree transductions via controlled optimality systems. In: de Groote, P., Nederhof, M.-J. (eds.) Formal Grammar 2010/2011. LNCS, vol. 7395, pp. 97–113. Springer, Heidelberg (2012)

18. Bethin, C.Y.: Phonological rules in the nominative singular and genitive plural of the Slavic substantive declension. PhD thesis, University of Illinois, Champaign–Urbana (1978)

19. Kenstowicz, M., Kisseberth, C.: Generative Phonology: Description and Theory. Academic Press (1979)

20. Law, H.: Morphological structure of Isthmus Nahuat. International Journal of American Languages 24, 108–129 (1958)

21. Hulstaert, G.: Grammaire du Lomongo. Musée royal de l'Afrique centrale (1961)

22. de Rijk, R.: Vowel interaction in Bizcayan Basque. Fontes Linguae Vasconum 2(5), 149–167 (1970)