A mathematical demonstration that classic Optimality Theory is expressively weaker than ordered rewrite rules[1]

Brian Buccola
McGill University
brian.buccola@mail.mcgill.ca

MOT phonology workshop
University of Ottawa
March 16, 2013

1 Introduction

- Two major frameworks of generative phonology.

  1. *Rule–based serialism* (SPE; Chomsky and Halle, 1968): input is mapped to output via ordered, context–sensitive rewrite rules: $A \to B \ / \ C\_D$.
  2. *Constraint–based parallelism* (OT; Prince and Smolensky, 2004): input is mapped directly to those outputs that are optimal wrt. some set of ranked (totally ordered) constraints.

- Today's talk:

  1. Are there are input–output patterns that can be expressed in one framework but not the other?
  2. If so, are those patterns attested in natural language?

- First question is formal: if the two frameworks are expressively equivalent, then no difference in empirical coverage.

- Second question is empirical: if expressively different, then the one with better empirical coverage is preferable.

- Widespread intuition that *classic* OT (McCarthy, 2007), or traditional OT (Heinz, 2011), is expressively weaker than ordered rewrite rules.

- Specifically, there are attested, *phonologically opaque* patterns that are problematic for classic OT (Kager, 1999; McCarthy, 2007; Baković, 2007). However:

  – This claim has, to my knowledge, never been *formally* demonstrated.
  – Problematic in what sense? Are opaque patterns not expressible *at all*, or are they expressible in principle but require ad hoc, unmotivated constraints?

- Motivations for a formal demonstration of classic OT's inadequacy:

[1]This presentation reports on work completed for my second evaluation paper, which can be found at `people.linguistics.mcgill.ca/~brian.buccola/`. Many thanks are due to my supervisor, Morgan Sonderegger, for countless helpful discussions of this material, as well as to my two other committee members, Heather Goad and Brendan Gillon, for helpful comments on several earlier drafts. Thanks also to attendees of McGill's Ling Lunch series, especially Michael Wagner, who was the first to suggest using Canadian Raising for the proof presented here.

1. Prove that there are patterns which can be expressed by ordered rewrite rules but which cannot be expressed by *any* set of classic OT constraints, under *any* ranking.
   - Substantive difference in expressive power between classic OT and ordered rules.
   - Provides concrete way to compare frameworks.
2. Characterize the constraints which *are* needed to capture opaque patterns; compare them with proposed extensions to OT.
   - Provides insight into types of opacity, e.g., environment opacity requires x type of constraint, focus opacity y type.
   - May connect opacity with other phenomena requiring similar extensions/constraints.

- Use Canadian raising opacity for proof in this talk.
- Proof technique works for whole classes of counterbleeding on environment and counterfeeding on environment opacity (Baković, 2007).

Counterbleeding on environment	*Counterfeeding on environment*
Canadian raising/flapping	Isthmus Nahuat apocope/devoicing
Polish raising/devoicing	Bedouin Arabic raising/epenthesis
Bedouin Arabic deletion/palatalization	Lomongo gliding/deletion

Figure 1: Phonological patterns that are provably inexpressible by classic OT grammars, as defined in this talk.

- Overview:
  - Discuss intuitive notion of classic OT, especially classic faithfulness.
  - Look at Canadian raising opacity: easily captured by ordered rewrite rules, but intuitively problematic for classic OT.
  - Run–through of proof that Canadian raising opacity is not expressible by *any* classic OT grammar (as defined here).

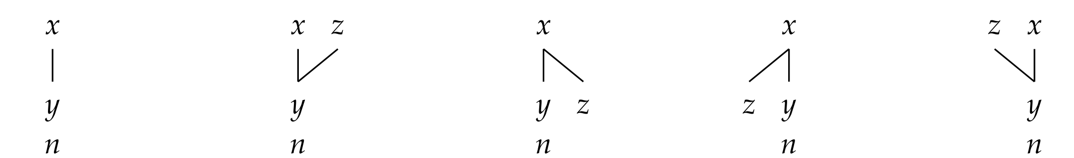
2 Classic OT

- Just two levels of representation: *input* (underlying) and *output* (surface).
- Input is mapped to that output that is optimal wrt. some set of ranked constraints.
- Potentially infinite set of constraints.
- Correspondence relation between segments in input and those in output.
- Two types of constraints:
  1. *Markedness*: penalize certain sequences of output segments, with no reference to input.
  2. *Faithfulness*: penalize certain *single* input–output segment pairs in correspondence, with no reference to any *other* segments in input or output.
- Faithfulness constraint assigns violation mark of 0 or 1 for each single segment pair $x \to y$ in correspondence.

(a) No violations (b) One total violation

Figure 2: Ident–IO(low) assigns 0 or 1 to each single input–output segment pair in correspondence.

- May *not* take context into account, i.e., may not "look behind" or "look ahead" at other segments in input or output.
- e.g., may *not* assign 1 to $x \to y$ only when x (or y) precedes (or follows) some other segment.



(a) Valid (b) Invalid (c) Invalid (d) Invalid (e) Invalid

Figure 3: What a faithfulness constraint can "look at" when assigning n violations to $x \to y$.

- In my paper,[2] I model this behavior formally in terms of finite state transducers (Riggle, 2004), but for our purposes this level of precision is fine.
- Without further restrictions, this version of OT allows, e.g., anti–faithfulness constraints, but that's OK: proof of expressive weakness will still work, even with that added power.

3 Canadian raising opacity

- Canadian English:

(1) **Canadian raising data**

a.	/raɪt/	→	[rʌɪt]	"write"
b.	/raɪd/	→	[raɪd]	"ride"
c.	/raɪtər/	→	[rʌɪɾər]	"writer"
d.	/raɪdər/	→	[raɪɾər]	"rider"

- Can capture all four patterns with two rules.

(2) **Canadian raising rules**

a. Raising: aɪ → ʌɪ / _ t
b. Flapping: t, d → ɾ / ˈV_V

[2]See footnote 1.

(3) **Rule–based derivations**

		Raising		Flapping	
a.	/raɪt/	→	rʌɪt	→	[rʌɪt]
b.	/raɪd/	→	raɪd	→	[raɪd]
c.	/raɪtər/	→	rʌɪtər	→	[rʌɪɾər]
d.	/raɪdər/	→	raɪdər	→	[raɪɾər]

- OT: capture raising as interaction between a markdedness constraint that penalizes [aɪ] followed a voiceless obstruent, and a faithfulness constraint that penalizes /aɪ/→[ʌɪ].
  - *aɪt, Ident–IO(low)
- Capture flapping as interaction between a markedness constraint that penalizes intervocalic [t, d], and a faithfulness constraint that penalizes /t, d/→[ɾ].
  - *ˈV{t,d}V, Ident–IO(sonorant)

(4) **Canadian raising constraints** (one possible ranking)
*aɪt >> *ˈV{t,d}V >> Ident–IO(low) >> Ident–IO(sonorant)

(5) **Attempted OT analysis of Canadian Raising**

	*aɪt	*V{t,d}V	Id(low)	Id(sonorant)
"write" /raɪt/				
a. raɪt	1			
→ b. rʌɪt@			1	
"ride" /raɪd/				
→ c. raɪd@				
d. rʌɪd			1	
"writer" /raɪtər/				
e. raɪtər	1	1		
f. rʌɪtər		1	1	
→ g. raɪɾər				1
h. rʌɪɾər@			1	1
"rider" /raɪdər/				
i. raɪdər		1		
j. rʌɪdər		1	1	
→ k. raɪɾər@				1
l. rʌɪɾər			1	1

- Note: "@" denotes *actual* output of given input, and "→" denotes *winning* candidate.
- "write", "ride", "rider" are correctly analyzed, but "writer" is not.
- [raɪɾər] is a more optimal output of /raɪtər/ than attested form, [rʌɪɾər].
- Violations assigned to [rʌɪɾər] for /raɪtər/ are a proper *superset* of those assigned to [raɪɾər]: [rʌɪɾər] is *harmonically bounded* by [raɪɾər], wrt. /raɪtər/.
  - No reranking of these four constraints can possibly make [rʌɪɾər] more optimal than [raɪɾər] for /raɪtər/.

- But what about a different constraint set? Is this analysis salvageable at all, even with ad hoc constraints?
- Quick example: chain–shift opacity /a/→[e], /e/→[i], /a/↛[i] can be expressed, e.g., with an (undominated) constraint that penalizes just /a/→[i] (Western Basque; de Rijk, 1970).
- It would be nontrivial to know that *not all* cases of opacity can be handled with such ad hoc constraints.
  - The fact that some patterns are totally inexpressible, no matter the constraint set or ranking, is not immediately obvious.

4 Run–through of the proof

- To prove: there are input–output patterns that can be expressed by ordered rewrite rules but not by any classic OT grammar, i.e., not by *any* set of constraints, under *any* ranking.
- Specifically: there is *no* set of constraints that can be ranked so that both
  1. [rʌɪɾər] is the optimal output of /raɪtər/.
  2. [raɪɾər] is the optimal output of /raɪdər/.
- How to deal with a potentially infinite set of constraints, and a potentially infinite set of output candidates? Two simplifications:
  1. Suffices to consider just the infinite set of all possible "classic" constraints (not every subset).
     Reason. If the infinite set cannot be successfully ranked, then neither can any subset, because if a subset *could* be so ranked, then so could the infinite set: just rank the rest below the successful ones.
  2. Suffices to consider just the two output candidates [raɪɾər] and [rʌɪɾər] for both inputs /raɪtər/ and /raɪdər/.
     Reason. Each candidate is the actual output of exactly one of the two inputs. If there is always at least one non–actual form that's optimal under every possible ranking of constraints, then no extra candidates could possibly make both actual outputs optimal. Intuitively, adding more candidates can never make a losing candidate suddenly win.
- Recap: just two output candidates; try to rank infinite set of classic constraints.
- Highest constraint on which two candidates differ determines winner and loser.
- Need a ranking that makes [rʌɪɾər] "win" for /raɪtər/ and makes [raɪɾər] "win" for /raɪdər/.
- Make a table showing what all of the logically possible helpful constraints look like.
- Let w = "win", l = "lose", and d = "draw" (cf. Elementary Ranking Conditions of Prince, 2002).
- That leaves 9 logically possible constraint types.[3]

[3] I say "types" because there are, in principle, infinitely many constraints that act like m_1, like m_2, etc. For simplicity, though, I'll refer to each m_i as if it were a constraint.

(6) **All logically possible constraint types**

		m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9
/raɪtər/										
a.	raɪɾər	*d*	*w*	*l*	*d*	*d*	*w*	*w*	*l*	*l*
b.	rʌɪɾər@	*d*	*l*	*w*	*d*	*d*	*l*	*l*	*w*	*w*
/raɪdər/										
c.	raɪɾər@	*d*	*d*	*d*	*w*	*l*	*w*	*l*	*w*	*l*
d.	rʌɪɾər	*d*	*d*	*d*	*l*	*w*	*l*	*w*	*l*	*w*

- Only care about constraints that make at least one desired output win; can disregard m_1, m_2, m_5, and m_7.

(7) **Logically possible helpful constraint types (reduced)**

		m_3	m_4	m_6	m_8	m_9
/raɪtər/						
a.	raɪɾər	*l*	*d*	*w*	*l*	*l*
b.	rʌɪɾər@	*w*	*d*	*l*	*w*	*w*
/raɪdər/						
c.	raɪɾər@	*d*	*w*	*w*	*w*	*l*
d.	rʌɪɾər	*d*	*l*	*l*	*l*	*w*

- m_6 and m_9 are both markedness constraints, but do not alone suffice.

  Reason. One must be ranked over the other, but if $m_6 >> m_9$, then [rʌɪɾər] is a more optimal output of /raɪdər/ than [raɪɾər], contra fact; and similar for if $m_9 >> m_6$.

- Thus, *at least one* of m_3, m_4, and m_8 is required.

- Claim: Each of m_3, m_4, m_8 is like the "invalid", non–classic faithfulness constraints from earlier.

- Proof (two cases to check):

  Case 1. None is a markedness constraint: letter assignments are different across both inputs, despite output candidate sets being exactly the same.

  Case 2. None is a classic faithfulness constraint. Only difference between two output candidates is second segment, which for both candidates is in correspondence with /a/. Thus, any classic faithfulness constraint must assign same letters across *both* inputs, but none of m_3, m_4, or m_8 does so.

  More precisely, consider an arbitrary faithfulness constraint c. Suppose that c assigns n violations for aɪ→aɪ and k violations for aɪ→ʌɪ. Whether $n = k$ or $n > k$ or $n < k$, c must assign same letters across both inputs.

- m_3 corresponds, e.g., to a constraint that penalizes aɪ→aɪ, but only when /aɪ/ precedes /t/.

- m_4 corresponds, e.g., to a constraint that penalizes aɪ→ʌɪ, but only when /aɪ/ precedes /d/.

- m_8 assigns a violation whenever m_3 *or* m_4 does.

- These three constraints must all "look" at segments other than the two correspondents under consideration (aɪ→aɪ and aɪ→ʌɪ).

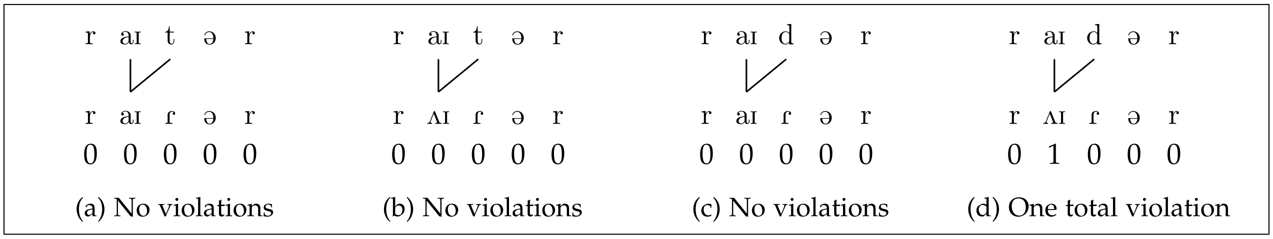


Figure 4: What m_4 must "look" at when determining violation mark for the pairs aɪ→aɪ and aɪ→ʌɪ.

5 Concluding remarks

- Classic OT, as defined here, consists of just two levels of representation and of just two types of constraint (markedness and faithfulness).
- Specifically, a classic faithfulness constraint assigns violation marks based on *single* input–output segment pairs; no "looking ahead/behind" allowed.
- Two patterns of Canadian raising are expressible by ordered rewrite rules but not by *any* set of classic OT constraints, under *any* ranking.
- Proof technique works for many other cases of environment opacity, but *not* (in general) for focus opacity (recall Basque chain–shifts).
  - Conjecture: focus opacity, but not environment opacity, is in general expressible by classic OT grammars.
  - Substantive formal difference in expressive power between classic OT, as defined here (and any more restrictive version) and ordered rules.
  - If the opaque patterns are attested, then difference in empirical coverage.
- In my paper, I make the notion of "looking ahead/behind" (taking context into account) more precise by formalizing OT constraints as finite state transducers (FSTs).
- Classic (valid) faithfulness constraints = input–*dependent*, *single*–state FSTs.
- Non–classic ones = input–*dependent*, *multi*–state FSTs.
- Note: If classic faithfulness constraints = input–dependent FSTs with $\leq$ 2 states, then Canadian raising is expressible, among other opaque patterns (see Fig. 1).
- Multistate faithfulness constraints, i.e., constraints that can look ahead/behind one (or more) segments, have already been proposed for analyses of other phonological data.[4]
  - Positional faithfulness: e.g., Ident–IO(high), but only in onsets (Beckman, 1998).
  - Anchor constraints: e.g., don't delete/epenthesize at an edge (McCarthy and Prince, 1995, 1999).
  - Multiple correspondence (McCarthy and Prince, 1995, 1999).
- Future directions: explore division between environment and focus opacity, look at other phenomena that require contextual faithfulness.

[4]Thanks to Heather Goad for bringing these examples to my attention.

References

Baković, Eric. 2007. A revised typology of opaque generalizations. *Phonology* 24:217–259.

Beckman, Jill. 1998. Positional faithfulness. Doctoral Dissertation, University of Massachusetts Amherst.

Chomsky, Noam, and Morris Halle. 1968. *The sound pattern of English*. The MIT Press.

Heinz, Jeffrey. 2011. Computational phonology—part II: Grammars, learning, and the future. *Language and Linguistics Compass* 5:153–168.

Kager, René. 1999. *Optimality theory*. Cambridge University Press.

McCarthy, John. 2007. *Hidden generalizations: Phonological opacity in optimality theory*. Equinox Publishing Ltd.

McCarthy, John, and Alan Prince. 1995. Faithfulness and reduplicative identity. In *University of Massachusetts Occasional Papers in Linguistics*, ed. Jill Beckman, Laura Walsh Dickey, and Suzanne Urbanczyk, volume 18, 249–384. GLSA Publications.

McCarthy, John, and Alan Prince. 1999. Faithfulness and identity in Prosodic Morphology. In *The prosody–morphology interface*, ed. René Kager, Harry van der Hulst, and Wim Zonneveld, 218–309. Cambridge University Press.

Prince, Alan. 2002. Entailed ranking arguments. ROA–500, Rutgers Optimality Archive, `http://roa.rutgers.edu/`.

Prince, Alan, and Paul Smolensky. 2004. *Optimality Theory: Constraint interaction in Generative Grammar*. Blackwell Publishers.

Riggle, Jason. 2004. Generation recognition and learning in finite state Optimality Theory. Doctoral Dissertation, UCLA.

de Rijk, Rudolph. 1970. Vowel interaction in Bizcayan Basque. *Fontes Linguae Vasconum* 2:149–167.